Automatizacija u kući primjenom koncepta IoT

Žirovec, Anđelko

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:122:993676

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

University North Digital Repository



Sveučilište
Sjever

Završni rad br. 403/EL/2017

Automatizacija u kući primjenom koncepta IoT

Anđelko Žirovec, 5560/601

Varaždin, srpanj 2017. godine



Sveučilište
Sjever

Odjel za Ime odjela

Završni rad br. 403/EL/2017

Automatizacija u kući primjenom koncepta IoT

Student : Anđelko Žirovec

5560/601

Mentor

Josip Srpak, dipl. ing el.

Varaždin, srpanj 2017. godine

Sveučilište Sjever
Sveučilišni centar Varaždin
104. brigade 3, HR–42000 Varaždin

UNIVERSITY NORTH

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL: Odjel za elektrotehniku

PRISTUPNIK: Anđelko Žirovec | MATIČNI BROJ: 5560/601

DATUM: 30.06.2017. | KOLEGIJ: Automatizacija strojeva i uređaja

NASLOV RADA: Automatizacija u kući primjenom koncepta IoT

NASLOV RADA NA ENGL. JEZIKU: Home automation using the IoT concept

MENTOR: Josip Srpak | ZVANJE: predavač

ČLANOVI POVJERENSTVA:

1. Miroslav Horvatić, dipl.ing., predavač
2. Dunja Srpak, dipl.ing., predavač
3. Josip Srpak, dipl.ing., predavač
4. mr.sc.Ivan Šumiga, viši predavač - rezervni član
5.

Zadatak završnog rada

BROJ: 403/EL/2017

OPIS

U završnom radu potrebno je opisati koncepciju pametne kuće ostvarene primjenom mreže mikroupravljača zaduženih za nadzor i upravljanje procesima u kući. Prikupljeni se podaci (prekidači tipke, senzori, trošila ..) putem MQTT protokola prosljeđuju na servis u oblaku gdje im se opet može pristupiti s ciljevim upravljanja ili nadzora odnosno uvidom u povijest prikupljenih podataka. U slučaju prekida komunikacije svaki element sustava treba nastaviti raditi autonomno prema zadanim algoritmima.

U radu je potrebno:
- opisati osnovne elemente i sklopove od kojih se sustav sastoji,
- opisati problematiku implementacije sustava do funkcionalnosti,
- objasniti komunikacijska i programska rješenja,
- izraditi maketu kuće na kojoj se može demonstrirati rad sustava,
- izvršiti testiranje sustava komunikacije, nadzora i upravljanja, lokalno i putem web sučelja,
- komentirati i analizirati dobivene rezultate.

ZADATAK URUČEN: 04.07.2017. | POTPIS MENTORA: Josip Srpak

REPUBLIKA HRVATSKA
SVEUČILIŠTE SJEVER

Predgovor

Prvi primjerak koncepta interneta stvari bio je Coca Cola automat 1980-te godine na Carnegie Melon sveučilištu koji je imao mogućnost daljinskog informiranja o dostupnosti pića i temperaturi. Internet stvari prvi se put službeno imenuje 1999. godine. IoT je dobio odskočnu dasku razvojem tehnologije koja čini Internet lako dostupnim, a uređaje moćnim i jeftinim.
2008.-2009.godine broj spojenih uređaja izjednačio se s brojem ljudi koji žive na kugli zemaljskoj.

Danas, svjetska se populacija bliži brojci od 7,5 milijardi, a broj uređaja povezanih s internetom 22 milijarde.

Predviđa se da će do 2020. godine biti umreženo 50 milijardi uređaja koji će olakšavati svakodnevni život. Sve se više investira u razvoj sigurnosti i zaštitu privatnosti interneta stvari uz zadržavanje fleksibilnost i čari lakoće korištenja te nove tehnologije. Internet stvari objedinjuje mobilne operacijske sustave, web aplikacije, senzore i aktuatore na različitim platformama poput Arduina, računala na jednoj pločici poput Raspberry PI , komunikacijske protokole poput HTTP, CoAP, MQTT, Bluetooth, … [34]

Sažetak

U završnom je radu obrađena ideja daljinsko upravljive kuće koja ima automatiziranu rasvjetu i grijanje. Sustav se sastoji od mreže WiFi mikroupravljača. Mikroupravljači na temelju prikupljenih informacija od senzora temperature, vlage, senzora kiše, senzora pokreta, tipkala nadziru i upravljaju izvršnim elementima. Prikupljene se informacije spremaju u bazu podataka pa se u svakom trenutku može vršiti uvid u grafički pregled bliže povijesti podataka. Mikroupravljačima se upravlja daljinski preko bežične lokalne mreže postavljene u zvijezda topologiju. Web sučelje kojim se vrši nadzor i upravljanje nalazi se na lokalnom serveru. Pristup sučelju je preko Internet preglednika u lokalnoj mreži. Komunikacija između nadzorno-upravljivog sučelja i mikroupravljača vrši se MQTT protokolom. MQTT protokol zahtjeva prisutnost MQTT posrednika. U radu je testiran rad lokalno instaliranih Mosquitto i HiveMQ posrednika ali i javnih posrednika u oblaku. U slučaju prekida veze svaki element ima mogućnost autonomnog nastavka obavljanja svoje zadaće prema prethodno definiranim algoritmima. U radu su objedinjena znanja iz računalnih mreža, programskih jezika, elektrotehnike i računalnih alata kako bi se postigla funkcionalnost sustava. Demonstracija sustava objedinjuje cijeli koncept od makete kuće, komunikacije, nadzora i upravljanja, bilo lokalno bilo preko web sučelja.

KLJUČNE RIJEČI:

ESP8266 modul, relejni modul, senzor temperature i vlage, senzor kiše, senzor pokreta, , MQTT protokol, XAMPP multiplatforma, MariaDB baza podataka, Apache server, HTML /JavaScript /CSS web stranica, HiveMQ posrednik, Mosquitto posrednik.

Popis korištenih kratica

A/D	Analog-to-digital converter – pretvorba analognog signala u digitalni
AC/DC	Alternating Current / Direct Current - izmjenična struja / istosmjerna struja
AP	Access point – pristupna točka
BLOB	Binary large object – datoteka većeg kapaciteta spremljena u bazi podataka
CPU	Central processing unit - središnja jedinica za obradu
CSV	Comma-separated values - dokument koji sadrži samo tekstualne informacije bez ikakvog vizualnog uređivanja
DCV(VDC)	Direct current voltage - istosmjerni napon
FTP	File Transfer Protocol – mrežni protokol
GND	Ground - Naponska točka od 0 V
GPL	General Public License – javna licenca opće namjene
HTML	HyperText Markup Language - obilježavanje hiper-tektualnih dokumenata.
HTTP	Hypertext Transfer Protocol – mrežni protokol prijenosa informacija
I2C	Inter-Integrated Circuit – sabirnica za prijenos informacija na tiskanim pločama
ID	identification - identifikacijska oznaka
IDE	Integrated development environment - razvojno okruženje
IEEE	Institute of Electrical and Electronics Engineers – organizacija zadužena za standarde
IoT	Internet of Things - internet objekata
IP	Internet Protocol – mrežni protokol
IPV4	Internet Protocol version 4 - IP mrežni protokol verzije 4
IR	InfraRed - infracrveno
LAN	Local area network - lokalna mreža
LWT	Last Will and Testament – posljednja želja MQTT klijenta
M2M	Machine-to-Machine - komunikacija između dva uređaja
MAC	Media Acces Control - fizička adresa mrežnog uređaja
MQTT	Message Queue Telemetry Transport – protokol za razmjenu M2M poruka
OS	Operating system – operacijski sustav
OUT	Output - Izlazni pin
PCB	Printed-Circuit-Board Antennas – antena otisnuta na tiskanoj pločici mikroupravljača
PDF	Portable document format – vrsta portabilnog digitalnog dokumenta
PIR	Passive infrared sensor - infracrveni senzor pokreta
PULL-UP	Otpornik vezan između signalnog pina i + pol napajanja VCC
QoS	Quality of service - razina kvalitete usluge MQTT poruke
SDIO	Standard input output – ulazno izlazni pin mikroupravljača
SPI	Serial Peripheral Interface Bus – komunikacija serijskom sabirnicom
SPIFFS	SPI Flash File System – SPI datotečni sustav
SQL	Structured Query Language – programski jezik baze podataka
SSID	Service Set Identifier – identifikacijska oznaka bežične mreže
STA	Station – način rada wifi uređaja
TCP	Transfer control protocol – mrežni protokol koji brine o točnosti prijenosa
TLS	Transport Layer Security – sigurnosni mrežni protokol
UART	Universal asynchronous receiver/transmitter – univerzalni asinkroni predajnik/prijemnik
UDP	User Datagram Protocol – mrežni protokol za prijenos informacija, brži ali manje pouzdan od TCP-a.
URL	Uniform Resource Locator – adresa web stranice
USB	Universal Serial Bus – univerzalna serijska sabirnica
VCC	IC power supply pin - naponska točka napajanja
webSocket	Komunikacisjki protokol
WEP	Wired Equivalent Privacy – sigurnosni protokol pristupa bežičnoj mreži
WiFi	Wireless - Bežična radio-frekvencijska mreža
WPA/WPA2	Wi-Fi Protected Access - sigurnosni protokol pristupa bežičnoj mreži
XAMPP	XAMPP - Cross-Patform, Apache, MariaDB/MySQL, PHP, Pearl Multi platforma koja služi za testiranje i razvoj lokalnih dinamičkih web stranica
XML	Extensible Markup Language – označni jezik koji definira skup pravila za kodiranje dokumenata u formatu koji je čitljiv i računalima i ljudima

Sadržaj

1. IoT (Internet of things) 5
2. Elementi sustava pametne kuće 7

2.1. IoT 7
2.2. MQTT 9
2.3. ESP8266 14
Općenito 14

Programiranje 15

2.4. XAMPP 18
Apache 18

MariaDB 19

phpMyAdmin 20

2.5. SENZORI 21
DHTxx senzor 21

PIR senzor 22

Senzor kiše 23

Senzor inteziteta svjetlosti 24

Relej 25

3. Realizacija IoT okružanja 26

3.1. MQTT 26
3.2. ESP8266 28
NodeMCU DEVKIT V1.0 28

Arduino IDE 29

Povezivanje na MQTT broker 34

ESP Spremanje i učitavanje zadnje spremljenih postavki 36

ESP Povezivanje na lokalnu mrežu 38

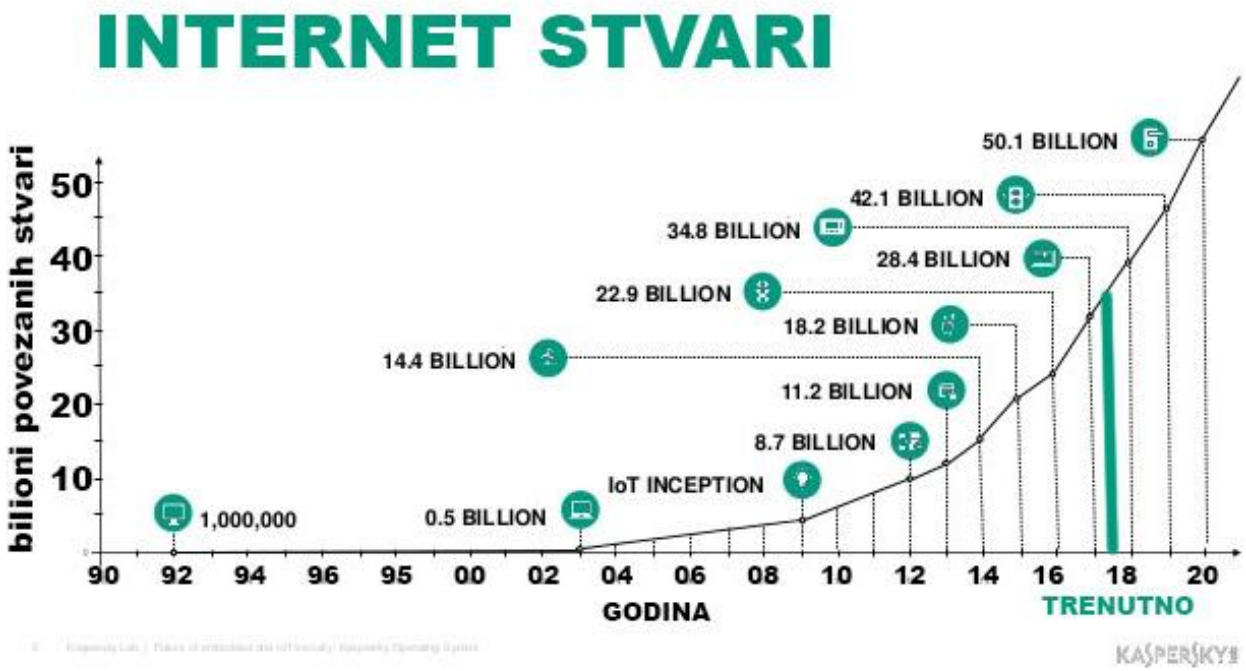
3.3. APACHE server i web stranica 40
Apache server 40

Nadzor i upravljanje putem web stranice 42

4. Zaključak 46
5. Literatura 48

1. IoT (Internet of things)

IoT ili Internet stvari je jedan od najvećih trendova posljednjih godina. Laka dostupnost internetu te jeftini moćni mikroupravljač s ugrađenom WiFi tehnologijom omogućili su procvat interneta stvari. Ideja je povezati svakodnevne predmete poput ručnih satova, frižidera, rasvjete, fitness pomagala na Internet. To bi omogućilo razmjenu informacija svakodnevnih stvari i njihovu dostupnost s bilo kojeg mjesta. Takav pristup omogućuje i direktnu međusobnu komunikaciju stvari što se naziva M2M (Machine to Machine) komunikacija. IoT je svoju primjenu pronašao u industriji, medicini, trgovini, kućnoj automatizaciji, prijevoznim sredstvima i drugo. Svoju ulogu jednako ispunjava kako u poslovnoj tako i u privatnoj dimenziji života. Već 2008. godine broj spojenih stvari na Internet prelazio je ukupan broj ljudi na planeti.
Do 2020. godine prema predviđanjima bi se ta brojka mogla popeti na 50 milijardi spojenih stvari. Uskoro će Internet stvari postati nezamjenjiva svakodnevica. Kompanije poput Googla, Apple i Samsunga ulažu ogromne svote novca u razvijanje te tehnologije. Prema predviđanjima bi do 2020. godine 250 miliona vozila trebalo biti opremljeno Internetom stvari, bilo da se radi o automatiziranoj vožnji ili pak drugim funkcijama. (Slika 2.1.1.)



Slika 2.1.1. Broj IoT stvari prikazan u vremenu [6]

IoT sustav u pametnim kućama podiže kvalitetu života ljudi pružajući im udobnost, dostupnost, sigurnost i jednostavnost življenja. Nekad skupa tehnologija koja se smatrala luksuzom polako ulazi u sve sfere društva i postaje standardom.. Tome je uzrok pojeftinjenje i laka dostupnost IoT opreme. Nema puno hrvatskih tvrtki koje se bave tim područjem. Neke od

njih su Smartcom, Electra-c i Matel koje trenutno dominiraju sustavima IoT automatizacije na domaćem tržištu. Od stranih tvrtki u hrvatskoj je vodeća Siemens svojim Synco Living Home Automation System-om. Te tvrtke uglavnom uvode IoT sustave automatizacije u stambene i poslovne zgrade, hotele, industriju, itd. Manjim dijelom tek zadiru u svijet obiteljskih kuća. Uvođenjem takvih sustava pokušava se postići jednostavnost korištenja, pouzdanost rada, ekonomičnost. IoT sustav u kućnoj automatizaciji pruža više prostora i vremena za privatni život jer će neke od poslova obavljati umjesto ljudi uz uštedu energenata. Nadzor i kontrolu vrši se svakim trenutkom uvidom u trenutno stanje te bližu povijest. Tome pripomaže i daljinski način rada.

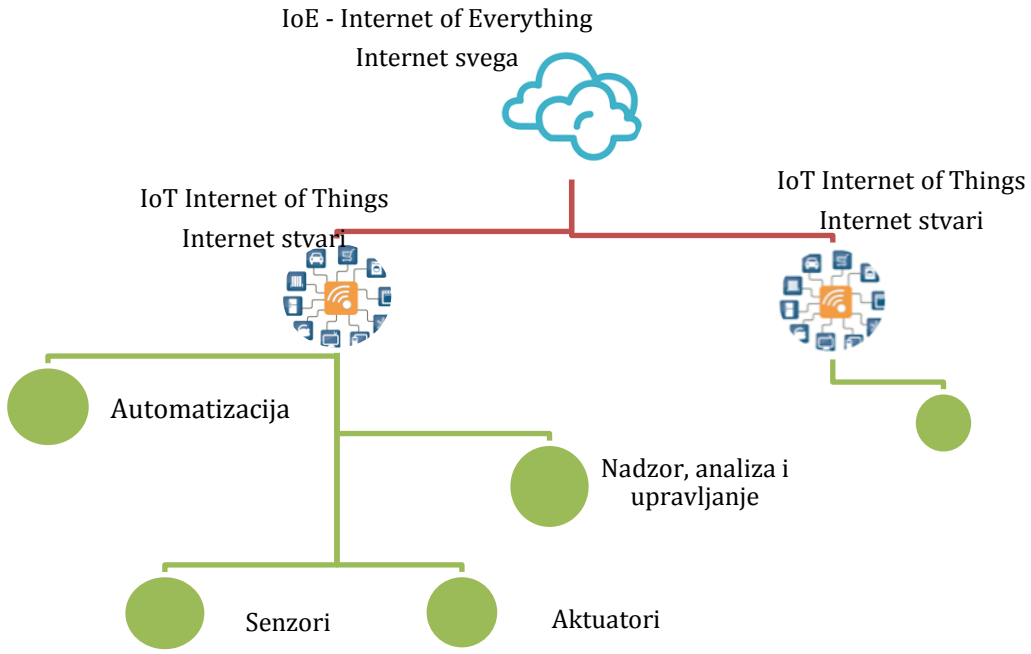
Neposredno prije ustajanja uključuje se sustav za grijanje vode, pristavlja se voda za kavu. Odlaskom iz kuće isključuju se svjetla, gase kućanski aparati, uključuje sigurnosni sustav. Pri povratku preko mobilne aplikacije uključuje se grijanje ili hlađenje prostorija, hladnjak dojavljuje stanje namirnica. Prolaskom kroz sobe svijetla se sama uključuju, svira unaprijed odabrana glazba, stvara željeni ugođaj. Sjenila na prozorima automatskim kontroliraju unos svjetlosti i toplinskog zračenja u kuću. Automatsko navodnjavanje održava život u vrtu. Početak kiše aktivira prozorsku zaštitu koja ih automatski zatvara. Odlaskom na spavanje sustav automatski smanjuje grijanje, isključuje sva svijetla i zaključava ulazna vrata. To su samo neke od funkcija koje se uvode u domove sustavima IoT tehnologije. [7] [8] [9] [14] [15]

2. Elementi sustava pametne kuće

2.1. IoT

OSNOVNI POJMOVI:

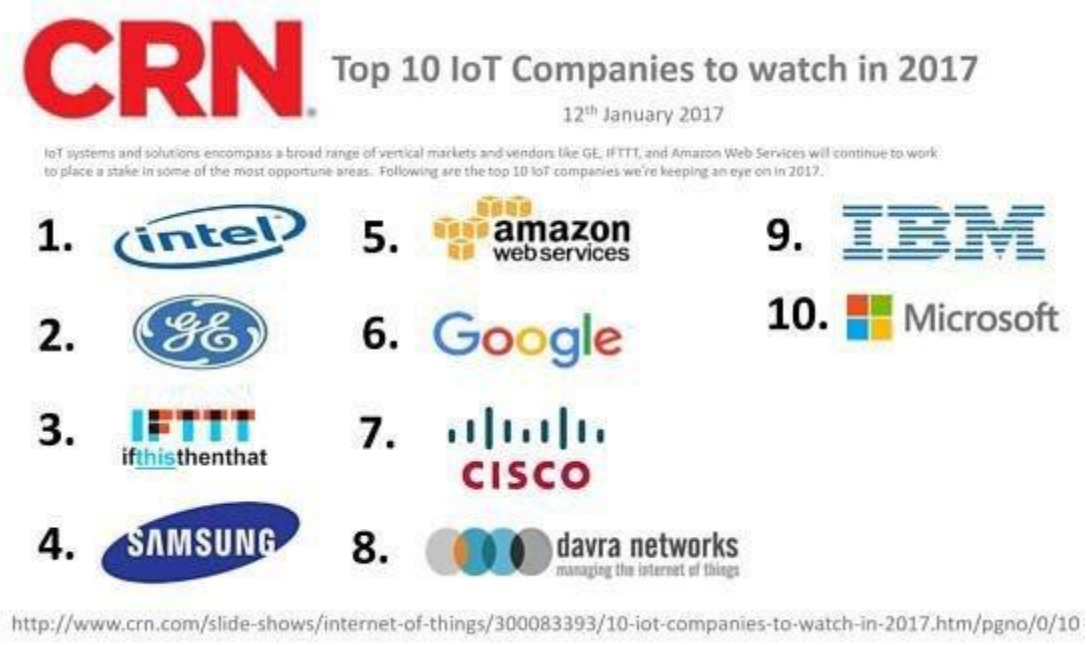
- **Internet stvari:** mreža povezanih stvari na Internet koji imaju sposobnost očitavanja informacija svoje okoline pomoću instaliranih senzora te razmjenu tih informacija.
- **IoT stvar:** bilo koja stvar povezana na Internet čiji je nadzor i upravljanje moguć s udaljene lokacije.
- **IoT ekosustav:** Zajednički naziv za sve komponente IoT sustava koje omogućuju poslovnom svijetu, vladinim organizacijama, i ostalim korisnicima povezivanje na njihove IoT uređaje, uključujući daljinsko upravljanje, kontrolne ploče, mrežni dio, analizu, spremanje podataka, i sigurnost. (Slika 2.1.2.) [8] [15]



Slika 2.1.2. Prikaz IoT arhitekture

IoT se koristi u industriji bilo koje namjene, transportu, agrokulturi, logistici, bankama, naftnim platformama, rudnicima, uslužnim djelatnostima, medicini, i drugo.

Svakim danom je sve više tvrtki čija je djelatnost Internet stvari a neke su navedene niže u tablici. (Slika 2.1.3.) [21]



Slika 2.1.3.Popis IoT tvrtki [21]

IoT platforme služe poput mostova za povezivanje IoT uređaja i omogućuju razmjenu informacija internetom pomoću transportnih protokola, te IoT uređaja i sustava za pohranu, obradu, analizu, nadzor i upravljanje.

Poznatije IoT platforme su:

- Amazon Web Services
- Microsoft Azure
- ThingWork IoT Platform
- IBM Watson
- Cisco IoT Cloud Connect
- Salesforce IoT Cloud
- Oracle Integrated Cloud
- GE Predix

Kako je sve više povezanih uređaja i stvari, zahvaljujući internetu stvari, tako se javlja i potreba za osiguravanjem sigurnosti i privatnosti bilo da se radi o poslovnim, vladinim, ili pak privatnim IoT ekosustavima. Sve je više cyber napada koji ciljaju upravo na IoT. Najviše se napada bilježi u sigurnosnim sustavima nadzora ali nisu ni ostali IoT sustavi pošteđeni. Zbog toga ima dosta tvrtki čija je primarna djelatnost zaštititi podatke i pouzdanosti takvih sustava.

Prednosti:

- Informacije – što je više informacija to je lakše napraviti ispravne odluke, smanjiti štetu i uštedjeti na vremenu i novcu.
- Praćenje: - računalni sustavi prate dostupnost, lokaciju, rok trajanja, i slične osobine te na temelju tih informacija povećavaju efikasnost i kvalitetu.
- Vrijeme: IoT štedi vrijeme daljinskim upravljanjem i nadzorom, lokatorima određuje najbrže rute i drugo.
- Novac: Kao rezultat svega iznad navedenog dobivamo financijsku uštedu u svim aspektima korištenja.

Nedostatci:

- Kompatibilnost: IoT još uvijek nije do kraja standardiziran
- Kompleksnost: ukoliko kompleksniji sustavi nisu dobro realizirani i imaju prikrivene bugove, može doći do ozbiljnijih problema.
- Privatnost i sigurnost: IoT je još uvijek nova i ranjiva tehnologija izložena hakerskim napadima bilo da se radi o krađi informacija ili pak manipulacijom samog sustava.

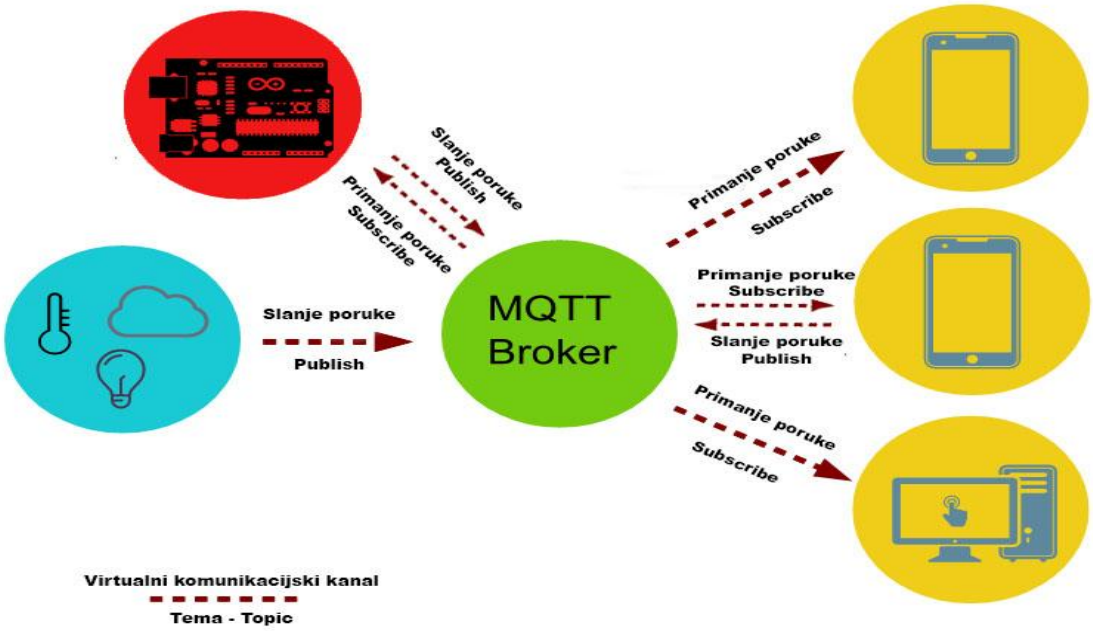
Uz sve nedostatke, IoT će se nastaviti razvijati velikom brzinom. Za par godine će nešto o čemu smo nekad mislili da je nemoguće postati svakodnevica upravo zbog IoT tehnologije. [15][14]

2.2. MQTT

MQTT (Message Queuing Telemetry Transport) je M2M (machine to machine) komunikacijski protokol koji služi za razmjenu informacija između IoT članova. Njegovo rođenje bilježi se 1999. godine. Od te godine započinje eksponencijalni rast razvoja IoT tehnologije. Komunikacija može biti između jednog klijenta i drugog klijenta ili pak jednog klijenta prema više klijenata. Komunikacija je anonimna jer klijent kad šalje poruku ne zna kome sve šalje poruku, a kad primi poruku on ne zna od koga je primio poruku. Klijent je na MQTT sustav povezan samo s posrednikom i za povezivanje na njega klijent treba imati poznatu informaciju o IP adresi posrednika i porta na kojem se komunikacija obavlja. Najčešće je to port 1883 za MQTT TCP komunikaciju odnosno 8883 za MQTT TLS (Transport Layer Security) komunikaciju te 80/443 za MQTT websocket komunikaciju. MQTT se nalazi u aplikacijskom sloju TCP/IP modela. Protokol je standardiziran od strane OASIS tehničkog odbora što je omogućilo jednostavnu implementaciju široke namjene IoT uređaja, raznih platformi i operacijskih sustava.

Osnovni pojmovi:

- **MQTT posrednik** : (eng broker) često se naziva i serverom i on je srce samog protokola. Ovisno o izvedbi može se nositi s tisućama klijenata istovremeno obrađujući i prosljeđujući poruke istodobno održavajući kontinuitet i pouzdanosti komunikacije. Njegova je zadaća primiti poruku, filtrirati po temama i proslijediti poruku klijentima koji zadovoljavaju kriterij pretplate na temu. Također sprema poruke pa ponavlja slanje ukoliko je došlo do gubitaka same informacije ili pak prekida veze. MQTT posrednici su Mosquitto, EMQttd, VerneMQ, HiveMQ … (Slika 2.2.1.)
- **MQTT klijent**: (eng. Client) je pojam koji se koristi za bilo koji uređaj koji ima mogućnost spajanja na MQTT broker preko žičane ili bežične mrežne veze te ima mogućnost primanja odnosno slanja MQTT poruke. Klijent mora imati TCP/IP stog koji je esencijalan za korištenje samog MQTT protokola. Uređaj može biti mikroupravljač siromašnijih performansi (8 bitni Arduino) pa sve do moćnih računalnih servera. MQTT Pub/Sub knjižnice za klijente dostupne su u velikom broju programskih jezika kao što su C, C#, C++, G0, iOS, Java, JavaScript, .Net, Android, Arduino…



Slika 2.2.1 MQTT posrednik : (eng broker)

Kontrolni paket	Smjer komunikacije	Opis
CONNECT	klijent > posrednik	Klijent zahtjeva povezivanje na posrednik
CONNACK	posrednik > klijent	Posrednih potvrđuje povezivanje
PUBLISH	klijent > posrednik posrednik > klijent	Objava poruke
PUBACK	klijent > posrednik posrednik > klijent	Potvrda objave poruke
PUBREC	klijent > posrednik posrednik > klijent	Poruka primljena
PUBREL	klijent > posrednik posrednik > klijent	Poruka proslijeđena
PUBCOMP	klijent > posrednik posrednik > klijent	Objava završena
SUBSCRIBE	klijent > posrednik	Zahtjev za pretplatom na temu
SUBACK	posrednik > klijent	Prihvaćena pretplata
UNSUBSCRIBE	klijent > posrednik	Zahtjev za ukidanje pretplate
UNSUBACK	posrednik > klijent	Prihvaćeno ukidanje pretplate
PINGREQ	klijent > posrednik	PING zahtjev
PINGRESP	posrednik > klijent	PING odgovor
DISCONNECT	klijent > posrednik	Prekidanje veze s posrednikom

Tablica 2.2.1. Prikaz kontrolnih poruka MQTT protokola

Usporedba pošta i MQTT

MQTT se može zamisliti kao slanje pisma poštom budući da ima dosta sličnost. MQTT posrednik (broker) ima sličnu ulogu kao i poštanski ured a MQTT klijent može se zamisliti ko korisnika usluga pošte. Ukoliko klijent želi poslati poruku on to čini tako da u omotnicu stavi sadržaj poruke s naznačenom temom poruke. Takva omotnica stiže u poštanski ured. Posrednik će pregledati poruku odrediti njenu temu, pogledati dali ima zainteresiranih klijenata za tu poruku te ukoliko ima, poruka će biti multiplicirana i poslana svim zainteresiranim korisnicima. Ukoliko korisnik želi primiti poruku on mora do znanja dati poštanskom uredu da je on zainteresiran za neku od tema na način da se u poštanskom uredu pretplati na temu (topic). Ukoliko je pretplata obavljena, korisnik će primati sva pisma koje su vezana uz pretplaćenu temu. Komunikacija može biti u jednom smjeru (korisnik ima mogućnost primanja pisma ili samo slanja pisma) ili dvosmjerna (korisnik ima mogućnost i primanja i slanja pisma).

Osnovna načela MQTT protokola:

1. Objavi/Pretplati : - (eng. Publish /Subscribe) MQTT protokol zasniva se na te dvije osnovne funkcije. Objavom se poruka šalje u komunikacijski kanal i u njenom sadržaju mora biti unesena tema (topic) te poruke. Ukoliko drugi uređaj želi primiti poruku on se pretplati na određenu temu i tada će sve poruke s tom temom biti proslijeđene njemu.
2. Tema i pretplaćivanje : Tema je svojstvo poruke koje mora biti navedeno u svakoj poruci. Na temelju teme broker kasnije filtrira i vrši slanje zainteresiranim korisnicima. Svaki korisnik koji je zainteresiran pretplaćuje se na temu i tada će primiti sve poruke vezane uz tu temu. Ukoliko koristimo hijerarhiju tema, za njihovu pretplatu koristimo zamjenske znakove (eng. wildcard). Zamjenski znakovi se mogu koristiti isključivo kod pretplata, ne vrijede kod objava. Postoje dva zamjenska znaka:
    - + za jednu razinu hijerarhije
    - # za više razina hijerarhije

   Uzmimo za primjer objave u sljedeće teme „ESP8266_1/dnevna/temperatura“ „ESP8266_1/hodnik/rasvjeta“ i „ESP8266_1/predsoblje/temperatura“. Pretplatom na “ESP8266_1/+/temperatura“ primat će se poruke o temperaturi i u dnevnoj i u predsoblju. Ukoliko formiramo pretplatu na sljedeći način ESP8266_1/# primat će se sve poruke vezane uz ESP8266_1.

3. Kvaliteta usluge : MQTT ima tri razine kvalitete usluge. Ukoliko je važno da poruka pouzdano stigne na odredište definira se viši QoS . Posljedica toga je veće zauzimanje mrežnog prostora i latencija poruke.
    - QoS 0 – poruke se dostavljaju na najbolje iskorištavanje mrežnog prostora ali može doći do gubitka poruke.
    - QoS 1 – poruka pouzdano stiže na odredište, moguće dupliciranje poruke
    - QoS 2 – poruka pouzdano stiže na odredište uz nemogućnost pojavljivanje duplikata
4. Zadržane poruke: - (eng. retained messages) MQTT broker nakon prosljeđivanja kod sebe zadržava poruke i čuva ih spremljene. Ukoliko se novi pretplatnik spoji na broker i pretplati sve će mu zadržane poruke s tematikom za koju je zainteresiran biti proslijeđene.
5. Zastavica čistog procesa (Clean session flag) – ukoliko je zastavica postavljena tada će sve pretplate klijenta biti poništene prilikom prekida veze s brokerom.

6. Opcija zamolbe – (eng last will and testament" LWT") klijent može od brokera tražiti da mu prilikom iznenadnog prekida veze izvrši automatsko slanje poruke zamolbe što je dosta važno u sigurnosnim sustavima u kojima svaki prekid veze senzora mora biti zabilježen. Korisna metoda kojom se detektira prekid veze kod bežične komunikacije itd.

Prednosti :

- omogućuje povezanost raznih senzora i aktuatora sa centralnom jedinicom
- jednostavan za implementaciju u rješavanju problema komunikacije.
- Pošiljatelj poruke niti ne zna kome je sve poslao poruku jer je fokus poruke na njenom sadržaju i jedini je uvjet da je definirana tema same poruke koju će kasnije primatelj koristiti kao pristupnicu za primanje poruke. Primatelj poruke također ne zna tko je pošiljatelj.
- Posrednik rješava probleme kod pucanja veze.
- Šalju se samo poruka i tema pa je veličina tog paketa minimalna što štedi energiju (omogućuje dulji rad uređaja s baterijskim izvorom napajanja) i omogućuje rad u mrežama malih propusnosti, visoke latencije i nestabilnih veza.
- Praćenje komunikacije omogućava svijest o stanju povezanosti uređaja i reagiranje porukom zamolbe.
- Kroz različite razine QoS poruke se zadržavaju na brokeru.
- Sadržaj poruke može biti u bilo kojem čitljivom formatu.

Poznatije implementacije MQTT protokola :

- Facebook Messenger – 2011. godine je tvrtka Facebook implementirala MQTT u svoju mobilnu aplikaciju ali ne postoje podaci u kojoj mjeri te u koju svrhu.
- IECC Scalable - sustav za kontrolu i signalizaciju
- EVRYTHNG - IoT platforma
- Amazon Web Services – Amazon IoT
- The OpenStack – open source IoT platforma
- Adafruit IO – IoT platforma Adafruit open source hardware kompanije

Izvor: [11] [15] [22]

2.3. ESP8266

Općenito

ESP8266 je SoC (System on a Chip) integrirani sklop niske cijene dizajniran za prostorno i energetske ograničene mobilne platforme. Ima ugrađenu sposobnost WiFi veze koji nudi mogućnost umrežavanja pametnih sustava ili pak djeluje kao samostalna WiFi stanica. Kada ESP djeluje kao samostalna jedinica tada se njegov programski kod direktno pokreće s vanjske memorije uz korištenje priručne cache memorije za poboljšanje performansi u te namjene. No ESP može imati i namjenu WiFi adaptera koji omogućuje umrežavanje gotovo svih mikroupravljača svojim jednostavnim načinom perifernog serijskog povezivanja (SPI/SDIO ili I2C/UART). Svojom cijenom od 5 $,ugrađenom WiFi komunikacijom te moćnim performansama u odnosu na dimenzije, približio se širokom broju ljudi a njegova primjena je raznolika. Znatno je jeftiniji od tradicionalnog Arduina s WiFi adapterom čija je cijena oko 40 $ uz gotovo iste mogućnosti. Zbog toga je jako cijenjen u sustavima kućne automatizacije. [13] [32]

Namjena:

- Automatizacija stambenih zgrada
- Pametne utičnice i rasvjeta
- Umrežavanje sustava
- Industrijska WiFi kontrola
- Baby monitor
- IP kamere
- Telemetrijski sustav senzora
- Pametni nosivi uređaji (narukvice, satovi, ..)
- Sigurnosne ID oznake
- WiFi sustavi pozicioniranja, i drugo

Osobine:

- 32-bit RISC CPU: Tensilica Xtensa L106 80 MHz
- 64 KiB RAM za instrukcije, 96 KiB RAM za podatke
- vanjska QSPI flash memorija: 512 KiB do 4 MiB
- IEEE 802.11 b/g/n Wi-Fi (20 / 17 / 14 dBm)
- WEP ili WPA/WPA2 autentifikacija, ili spajanje na nezaštićenu otvorenu mrežu
- 16 GPIO ulazno izlaznih pinova opće namjene s mogućnošću pulsno širinske modulacije
- 1 analogni ulaz
- integriran TCP/IP protokol stog koji omogućuje rad IPv4, TCP/UDP/HTTP/FTP protokola
- operabilno napajanje 3.3 V (min 3.0 ~ max 3.6 V)
- podržava STA/AP/STA+AP načine rada
- podržava Smart Link funkciju za Android i iOS uređaje
- periferne sabirnice SDIO 2.0, (H) SPI, UART, I2C, I2S, IR daljinsko upravljanje,
- STBC, 1x1 MIMO, 2x1 MIMO
- načini rada duboki san <10uA (buđenje < 2ms), isključen < 5uA, mirovanje < 1.0mW
- operabilna temperatura -40C ~ 125C
- certificiran od FCC, CE, TELEC, WiFi Alliance, i SRRC organizacija
- dimenzije 24x16 mm [17] [18] [19] [20]

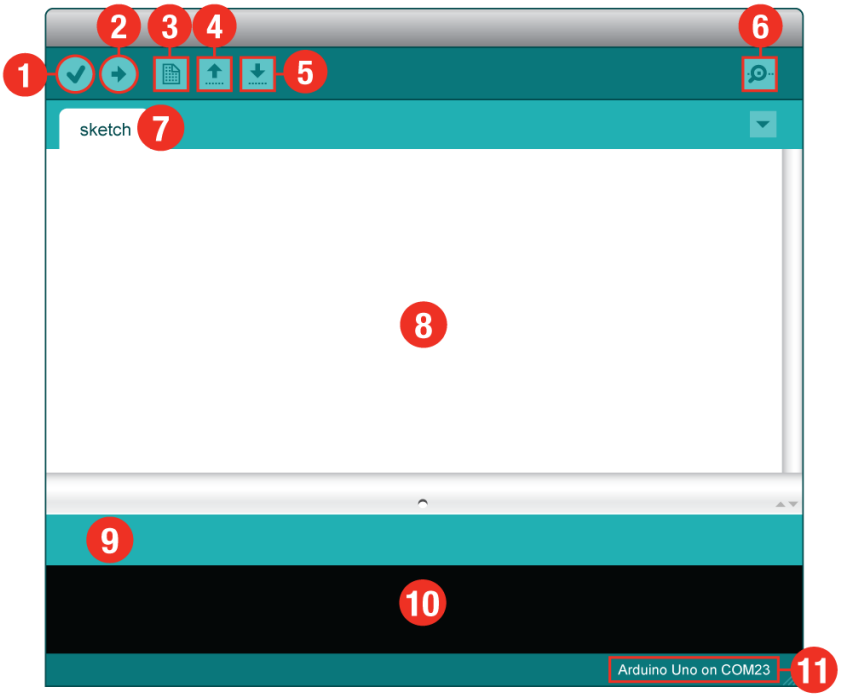
Programiranje

Za programiranje samog mikroupravljača potreban je još izvor napajanja koji će opskrbljivati ESP sa strujom od najmanje 300 mA, ali uvijek valja računati na manju rezervu pa bi preporučen izvor napajanja morao davati 3.3V i 500mA. Osim napajanja potreban je modul kojim će preko računala prebaciti serijskom komunikacijom kompajlirani program u mikroupravljač. U tu svrhu može se upotrijebiti i Arduino (npr. Arduino Uno) ili pak samostalne niskobudžetne i lako dostupne module kompatibilne s ESP8266 čipom poput „FTDI USB /UART programmer“. Na tržištu postoje i razvojne pločice koje već na sebi imaju ugrađen ESP8266 čip, napajanje, komunikaciju, tipkala te spojne pinove za programiranje pa je za njihovo korištenje potrebno samo spojiti USB na računalo što je idealno rješenje za razvojne svrhe i testiranja. Takav pristup korišten je u ovom završnom radu upotrebom NodeMCU 1.0 razvojne pločice. O njoj slijedi više informacija u praktičnom dijelu rada.

Spojenim hardwareom na računalo potreban je još samo razvojni alat kojim ćemo programirati mikroupravljač.

ARDUINO IDE (Arduino Integrated Development Environment)

je besplatni razvojni alat koji služi za razvijanje Arduino mikroupravljača. Kompatibilan je s ES8266 čipom zahvaljujući neprofitabilnoj društvenoj zajednici koja je osigurala podržavanje ESP8266 modula u Arduino IDE razvojnom okruženju. Sam razvojni alat je kreiran u java programskom kodu ali se za pisanje programa koji se koristi u mikroupravljačima upotrebljava C/C++ prilagođen Arduino okruženju. Sadrži uređivač programskog koda koji znatno olakšava pisanje funkcionalnog programskog koda. Pored njega sadrži i jednostavni mehanizam kompajliranja programskog koda u jezik razumljivom mikroupravljaču koji se pokreće jednostavnim klikom na ikonu za kompajliranje i prijenos. Na dnu samog razvojnog alata nalazi se područje konzole u kojem nas Arduino IDE porukama obavještava o statusu trenutne akcije. Te poruke uvelike pomažu kod otklanjanja sintaksnih grešaka. Arduino IDE sadrži i serijski monitor pa se može preko njega primati ali i slati podatci preko serijske sabirnice. Serijski način omogućuje nadzor i upravljanje mikroupravljača. (Slika 2.3.1.)



1.verifikacija
2.kopajliranje i prijenos na mikroupravljač
3. nova skica
4. otvori skicu iz datoteke
5. spremi skicu
6. serijski monitor
7. Ime taba skice
8. područje programskog koda
9. statusna poruka
10. konzola
11. naziv spojenog mikroupravljača i porta

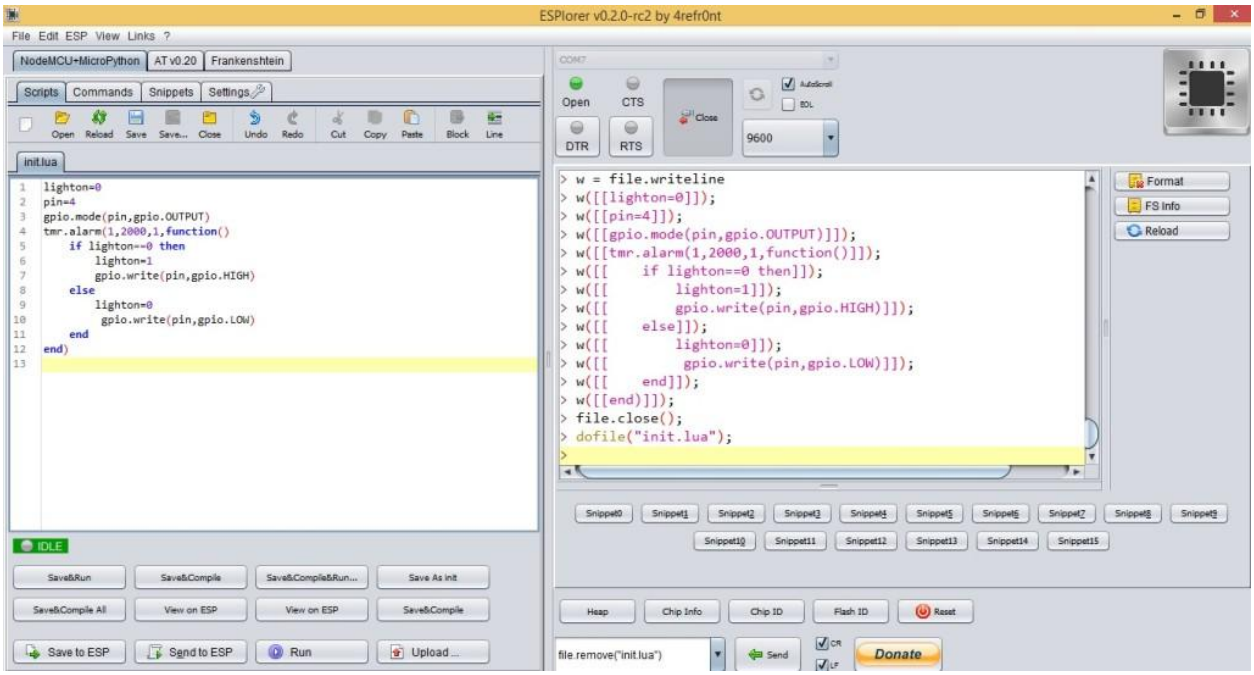
Slika 2.3.1. Prikaz radnog prozora Arduino IDE alata

Arduino IDE sadrži mnoštvo knjižnica koje se mogu ubaciti u programski kod. Knjižnice imaju primjere koji demonstriraju rad pa je njihovo razumijevanje svedeno na najviši mogući nivo. Knjižnice se dodaju „**Sketch → Include library**“ a primjerima se pristupa „**File → Examples**“

ESPlorer

Lua je moćan, pristupačan, lako usvojiv i brz programski jezik a dizajnirali su ga 1993. godine Roberto Ierusalimschy, Luiz Henrique de Figueiredo, i Waldemar Celes, članovi grupe računalne grafičke tehnologije na Papinskom katoličkom sveučilištu u Rio de Janeiro, Brazil. Koristi se u pisanju video igara te sustava raznih namjena. Dizajniran je kako bi zadovoljio sve veće potrebe za prilagodbom u svijetu softverskih aplikacija. Dizajneri LUA-e usredotočili su se na poboljšanje brzine, prenosivosti, proširivosti, fleksibilnosti i jednostavnosti korištenja u razvoju.

Programskim jezikom LUA je podržano programiranje ESP8266 mikroupravljača. Za pisanje programa mikroupravljača potreban je razvojni alat ESPlorer. ESPlorer ima na svojoj lijevoj strani uređivač teksta koji olakšava pisanje naredbi. Na desnoj strani ima serijski monitor kroz koji pruža nadzor i upravljanje samim modulom u razvoju. (Slika 2.3.2.) ESP8266 mora imati u svojoj memoriji snimljeni mikroprogram (firmware) koji će primati LUA naredbe. Te se naredbe pišu u ESPloreru i zatim spremaju u datoteke s ekstenzijom „ .lua “ u vanjsku memoriju samog modula. Datoteka init.lua se pokreće prva kod uključivanja ESP8266 čipa, a ukoliko se želi da koja druga datoteka s naredbenim blokovima bude pokrenuta mora se u init.lua datoteci ubaciti naredba „dofile('filename.lua'); “ [27]



Slika 2.3.2. Izgled ESPlorer razvojnog alata [27]

2.4. XAMPP

XAMPP je besplatna i jednostavna multi-platforma čijom instalacijom se dobiva na korištenje Apache , mySQL, PHP, phpMyAdmin i još mnoge druge aplikacije, ovisno o verziji, koje uvelike pomažu kod razvoja i testiranja dinamičkih web stranica. Podržava rad na Linux, Windows, Solaris, macOS operacijskim sustavima. Nije namijenjena komercijalnoj upotrebi jer nema sve sigurnosne postavke kao web server. Upravo iz tog razloga, izuzetno je koristan za testiranje i razvoj u lokalnom okruženju. [12]

X - označava simbolični znak za multi-platformu

A - Apache

M – mySQL od 2015. godine zamijenjen s MariaDB

P – PHP

P – Perl

Apache

Apache HTTP Server ili kraće Apache je najpopularniji i najkorišteniji web server koji služi kao domaćin za oko 60 % aktivnih web stranica. Besplatan je web server otvorenog tipa za čiji se razvoj brine „Apache Software Foundation“ zajednica. Najčešće je korišten na Linux podlozi ali je prilagođen za rad i na drugim platformama poput Windows, Unix, Mac OS X, FreeBSD, Novell NetWare, i druge. Ukoliko nije drugačije podešen, radi na portovima HTTP : 80 i HTTPS: 443. Apache podržava mnoge programske jezike poput Perl-a, Python-a, TCL-a i PHP-a. Također podržava razne autentifikacijske sheme. Virtualni hosting dozvoljava ovom HTTP serveru distribuciju više web stranica istovremeno jednom instalacijom. [23]

MariaDB

MariaDB je relacijska baza što znači da sadrži tablice s podacima koji su međusobno povezani relacijama, tj. zajedničkim podatkom. U primjeru ispod nalaze se dvije tablice. Prva se zove „Proizvod“ i sadrži podatke o opisu proizvoda i njegovoj cijeni. Druga tablica, „Skladišna_evidencija“, pokazuje podatke o proizvodu na skladištu. Međusobno ih povezuje šifra artikla. (tablica 2 i tablica 3).

šifra_artikla	opis	cijena
A416	Čavli kutija	30 kn
C923	Olovka, kom	7 kn

Tablica 2.4.1. „Proizvod“

pozicija	operacija	šifra_artikla	količina
3804	1	**A416**	10
3804	2	**C923**	15

Tablica 2.4.2. Skladišna evidencija

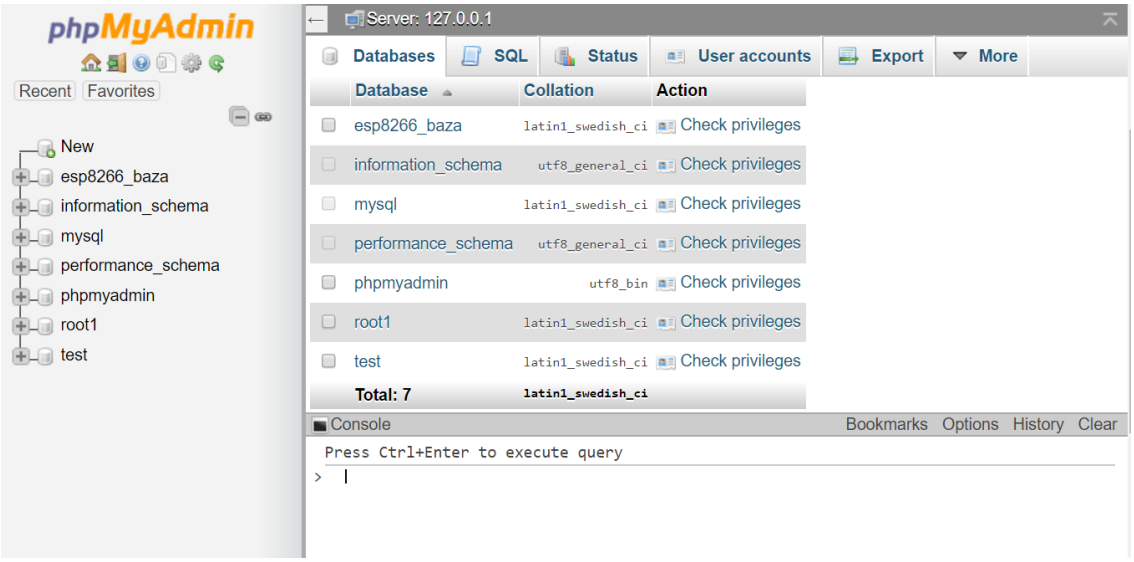
MariaDB je nasljednik slavne mySQL baze podataka i dostupna je pod GPL licencom. Dizajnirali su je razvojni programeri i članovi zajednice „MariaDB Foundation“ odcijepivši se od Oracleovog mySQL-a. Prati industrijske standarde u svrhu poboljšanja pa su sigurnosne zakrpe i nadogradnje uvijek u skladu s aktualnim stanjem. Određeni testovi pokazuju da je MariaDB brža od mySQL sustava. MariaDB je potpuno kompatibilan s mySQL relacijskom bazom.

phpMyAdmin

phpMyAdmin je besplatan alat otvorenog koda pisan u PHP programskom jeziku čija je namjena olakšati administrativni nadzor, upravljanje i održavanje baze podataka. Omogućuje kreiranje, modifikaciju ili brisanje relacijski baza, tablica, polja ili stupaca, izvršavanje SQL upita i uređivanje korisničkog pristupa. (Slika 2.4.1)

Mogućnosti :

- Web sučelje
- MySQL i MariaDB upravitelj bazom podataka
- Uvoz podataka iz CSV-a i SQL-a
- Izvoz podataka u raznim formatima: CSV, SQL, XML, PDF (preko TCPDF knjižnice), ISO/IEC 26300 - OpenDocument Text i Spreadsheet, Word, Excel, LaTeX i drugi.
- Više-serversko administrativno upravljanje
- Kreiranje PDF grafičkog prokaza baze podataka.
- Kreiranje kompleksnih upita pomoću Query-by-Example (QBE)
- Globalno traženje bazom podataka ili nekim njenim dijelom.
- Transformacija spremljenih podataka u bilo koji format koristeći zadane funkcije poput prikaza BLOB podataka (Binary Large OBject- slika, zvuk, ili drugi multimedijski objekti) kao grafičke slike ili kao link za preuzimanje.
- Nadzor rada, povezivanja, procesa, CPU/memorijskog korištenja servera preko grafova u realnom vremenu
- Rad na raznim operativnim sustavim



Slika 2.4.1. Prikaz phpMyAdmin sučelja

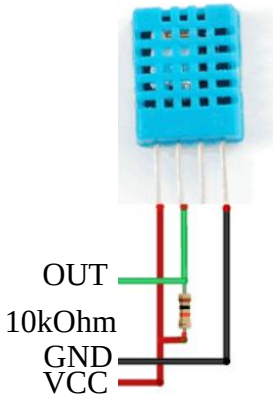
2.5. SENZORI

DHTxx senzor

U ovu skupinu senzora spadaju DHT11-21-22-33-44 senzori. Poznatiji su još po sinonimima:

- DHT11 = RHT01
- DHT21 = RHT02= AM2301 = HM2301
- DHT22 = RHT03= AM2302
- DHT33 = RHT04 = AM2303
- DHT44 = RHT05

Jeftini i senzori malih dimenzija namijenjeni su mjerenju temperature i vlage zraka u ne previše zahtjevnim i agresivnim okruženjima ali zahvalno obavljaju svoj posao u kućnoj automatizaciji i sličnim funkcijama. Duljina žice je do 20 m. Sastoje se od A/D pretvornika koji na izlaznom „OUT“ pinu daje digitalnu vrijednost temperature i vlage, termistora koji je osjetnik temperature te osjetnika vlage. Prvi pin slijeva se spaja na 3-5 V istosmjernog radnog napona. Drugi pin slijeva je digitalni izlazni pin s kojeg se čitaju podaci i on se spaja na mikroupravljač. Poželjno je spojiti 10 kOhm pull-up otpornik između OUT i VCC pina kao što je prikazano na slici 2.5.1. Treći pin se ne koristi i njega jednostavno izostavimo. Četvrti pin slijeva je GND pin koji se spaja na 0V. Preporuča se dodavanja 100nF kondenzatora između GND i VCC zbog filtriranja smetnji. Podatci se čitaju uz pomoć DHT knjižnice koja pomoću naredbi **dht.readHumidity()** i **dht.readTemperature()** omogućuje dohvat podataka. Minimalna pauza između čitanja je 2s. [1]



Slika 2.5.1. Shema spajanja DHTxx senzora

PIR senzor

Pasivni infracrveni senzor je malih dimenzija, jeftin i dugotrajan uređaj male potrošnje energije. Može se naći prema nazivima PIR(Passive Infrared), „Pyroelectric“ ili „IR senzor pokreta“. (Slika 2.5.2.) IR senzor detektira kretanje, tj. prostornu promjenu tijela koje emitira infracrvena zračenja–toplinu. Što je tijelo više zagrijano to je IR zračenje veće. Standardno napajanje je 3.3 ili 5 VDC no može podnijeti napon do 12 VDC . Senzor ima integriran BISS0001 čip koji se brine da analogna očitavanja osjetnika zračenja pretvara u digitalni izlaz. Izlaz može davati signal mikroupravljaču ili pak biti spojen, preko tranzistora, na relej i slično. Detektor pokreta je podijeljen na dvije polovice koje se međusobno poništavaju. Kada čovjek ili životinja dođe u „vidno“ područje senzora javlja se razlika u vrijednostima IR zračenja i senzor će na svom izlazu postaviti vrijednost od 3.3 V. Može se detektirati kretanje tijela koje emitira toplinu ali ne i saznati koliko je tijela prisutno. To će stanje držati još neko vrijeme a zatim će se ponovno vratiti u početno stanje detektiranja. Vrijeme držanja Tx, visoke razine napona na izlazu, može se namjestiti na ugrađenom potenciometru od 5 to 200 sekundi. a osjetljivost Sx 3 do 7 m. Na širinu vidnog polja, osjetljivost i domet uvelike utječe leća koja se nalazi na samom osjetniku topline te njegovo pozicioniranje. Kvalitetnim postavljanjem, ugađanjem i odabirom leće može se izbjeći neželjena detektiranja kućnih ljubimaca i drugih smetnji. [2]



Slika 2.5.2 PIR senzor [31]

Senzor kiše

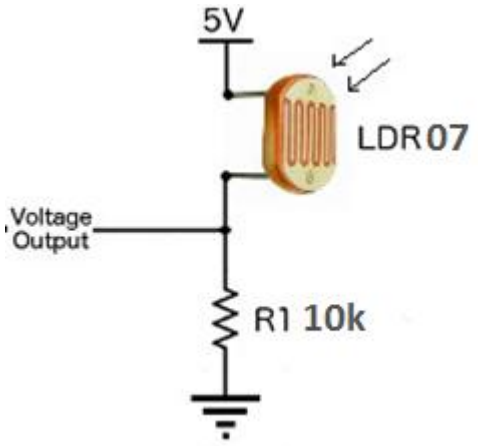
Napajan na 5 VDC, senzor kiše može raditi u dva načina. U jednom načinu rada služi kao sklopka koja će na svom digitalnom izlazu naponom od 5 V signalizirati detekciju padanja kiša. Osjetljivost se u tom slučaju regulira na ugrađenom potenciometru. U drugom načinu rada analogni izlaz postavlja vrijednost od 0 - 5 VDC ovisno o intenzitetu kiše. (Slika 2.5.3.) [4]



Slika 2.5.3. Senzor kiše [33]

Senzor inteziteta svjetlosti

Jedan od načina mjerenja intenziteta svjetlosti je preko fotootpornika. Ukoliko je količina svjetlosti veća, fotootpornik će poprimiti manju vrijednost otpora i obrnuto. Spajanjem u strujni krug dobiva se naponsko dijelilo koje će u idealnim uvjetima, ovisno o količini svjetlosti, u točci Vout davati napon od 0 VDC za potpuni mrak odnosno 5 V za najjači intenzitet svjetlosti. (Slika 2.5.4.) [3]

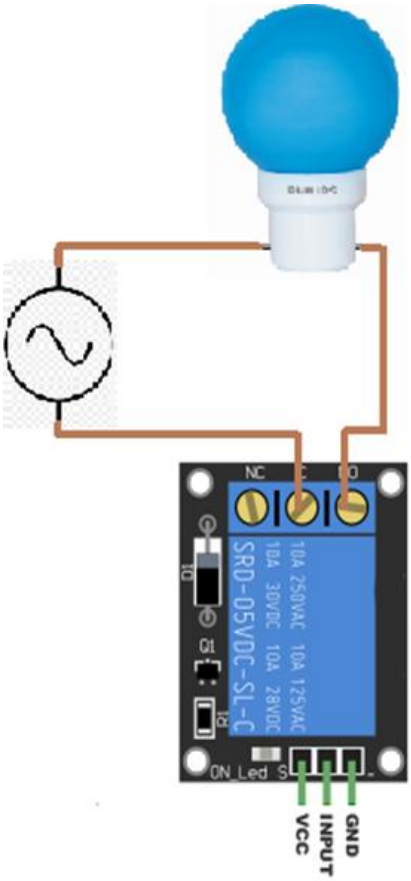


Slika 2.5.4 Shema spajanja fotootpornika [28]

Model	Vmax (VDC)	Pmax (mW)	Temperatura okoline (°C)	maksimalna spektralna osjetljivost (nm)
LDR07	150	100	-30 ~ +70	540

Tablica 2.5.1 karakteristika fotootpornika LDR

Relej



Slika 2.5.5 Relej SONGLE SRD_05VDC-SL-C

Relej je vrsta prekidača čije je stanje (uključeno ili isključeno) upravljano električnim nabojem, tj. energijom elektromagneta ili nekoga drugog mehanizma. Signalni pin pri određenoj vrijednosti automatski zatvara ili otvara kontakte energetskog dijela. Za SONGLE SRD_05VDC-SL-C ta je vrijednost 5 VDC. Vrijeme isklopa je 5 milisenkundi. Vrijeme uklopa je 10 milisekundi . Maksimalna struja koja može poteći energetskim kontaktima je 10A kod 240 VAC. Relej može raditi u temperaturnom opsegu okoline od -25°C do +70°C. Vrijeme života je 10^5 operacija. Slika 2.5.5 prikazuje oznake pinova te shemu spajanja energetskog dijela SONGLE SRD_05VDC-SL-C releja na potrošač. [5]

3. Realizacija IoT okružanja

3.1. MQTT

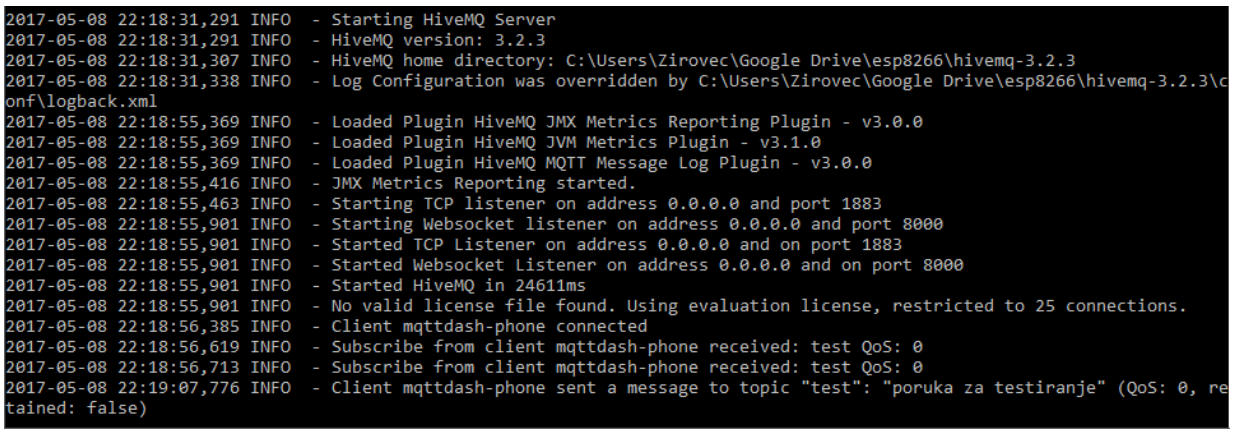
U svrhu korištenja MQTT protokola za razmjenu poruka potreban je posrednik (broker) na kojeg se zatim spajaju klijenti. U ovom radu isprobane su dvije verzije MQTT posrednika : Mosquito (Ubuntu okruženje) i HiveMQ broker (Windows okruženje). Za upravljanje i nadzor ESP8266 mikroupravljača je korištena web stranica. Web stranica je MQTT asinkroni klijent koji se spaja na posrednik putem websocketa i na taj način razmjenjuje poruke.

HiveMQ je MQTT posrednik za Windows okruženje. HiveMQ posrednik može se preuzeti s njihove službene stranice. Besplatna verzija omogućuje testiranje i upotrebu posrednika u edukacijske svrhe 6 mjeseci i maksimalno 25 istovremenih klijenata. Za sve što je iznad tih kriterija potrebna je kupovna licenca. Preuzimanjem posrednika otvara se mapa koja sadrži HiveMQ datoteke. Kako bi omogućili websocket treba se locirati i otvoriti, nekim do tekstualnih uređivača, konfiguracijska datoteka conf.xml koja se nalazi u ovom direktoriju „\hivemq-verzija\conf\conf.xml“ . U dio konfiguracije „listeners“ dodaje se konfiguraciju za websocket:

```
<listeners>
        <tcp-listener>
            <port>1883</port>
            <bind-address>0.0.0.0</bind-address>
        </tcp-listener>

        <websocket-listener>
            <port>8000</port>
            <bind-address>0.0.0.0</bind-address>
            <path>/mqtt</path>
            <subprotocols>
                <subprotocol>mqttv3.1</subprotocol>
            </subprotocols>
            <allow-extensions>true</allow-extensions>
        </websocket-listener>
    </listeners>
```

Spremi se konfiguracija a pokretanje MQTT posrednika vrši se tako da se otvori datoteka „\hivemq-verzija\bin\run.bat“ desnim klikom → pokreni kao administrator. (Slika 3.1.1.)



Slika 3.1.1. HiveMQ posrednik

Mosquitto je besplatni open-source posrednik. Standardno može istovremeno manipulirati s 1024 spojenih klijenata ali se uz određena podešena taj broj može popeti na 100K, nešto manje uz omogućen SSH protokol. Postoji i verzija za Windows OS ali onda nema mogućnost upotrebe websocketa. Na linux okruženju instalira se pomoću sljedećih naredbi preko terminala:

```
sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
sudo apt-get update
sudo apt-get install mosquitto
sudo apt-get install mosquitto-clients
```

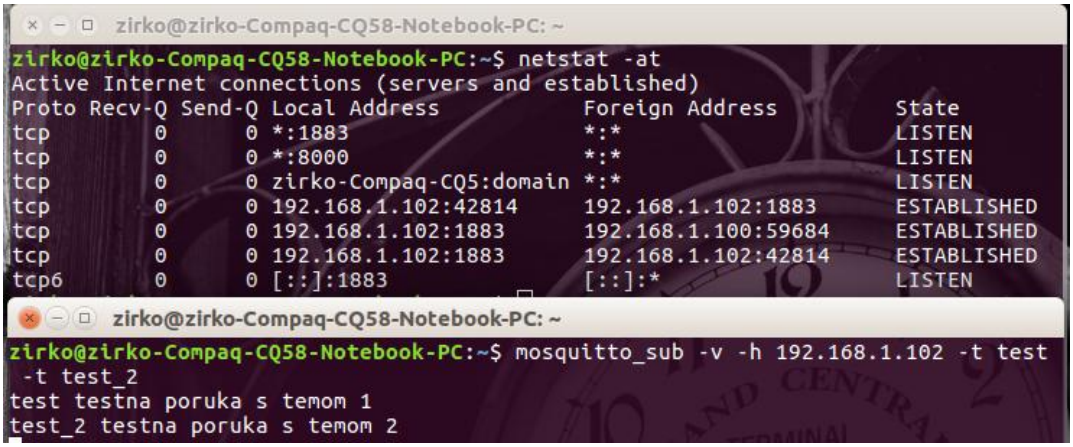
Nakon što je instaliran biti će automatski pokrenut. Uređenjem konfiguracijske datoteke „\etc\mosquitto\mosquitto.conf“ omogućuje se websocket na portu 8000.

```
include_dir /etc/mosquitto/conf.d
listener 1883
listener 8000
protocol websockets
```

Nakon spremanja konfiguracijske datoteke potrebno je ponovno pokrenuti Mosquitto posrednik i dodati željeni port u firewall. Operaciju činimo sljedećim naredbama

```
sudo systemctl restart mosquitto
sudo ufw allow 8000
```

Preostalo je samo još testirati rad Mosquitto posrednika.



Slika 3.1.2 Testiranje Mosquitto posrednika

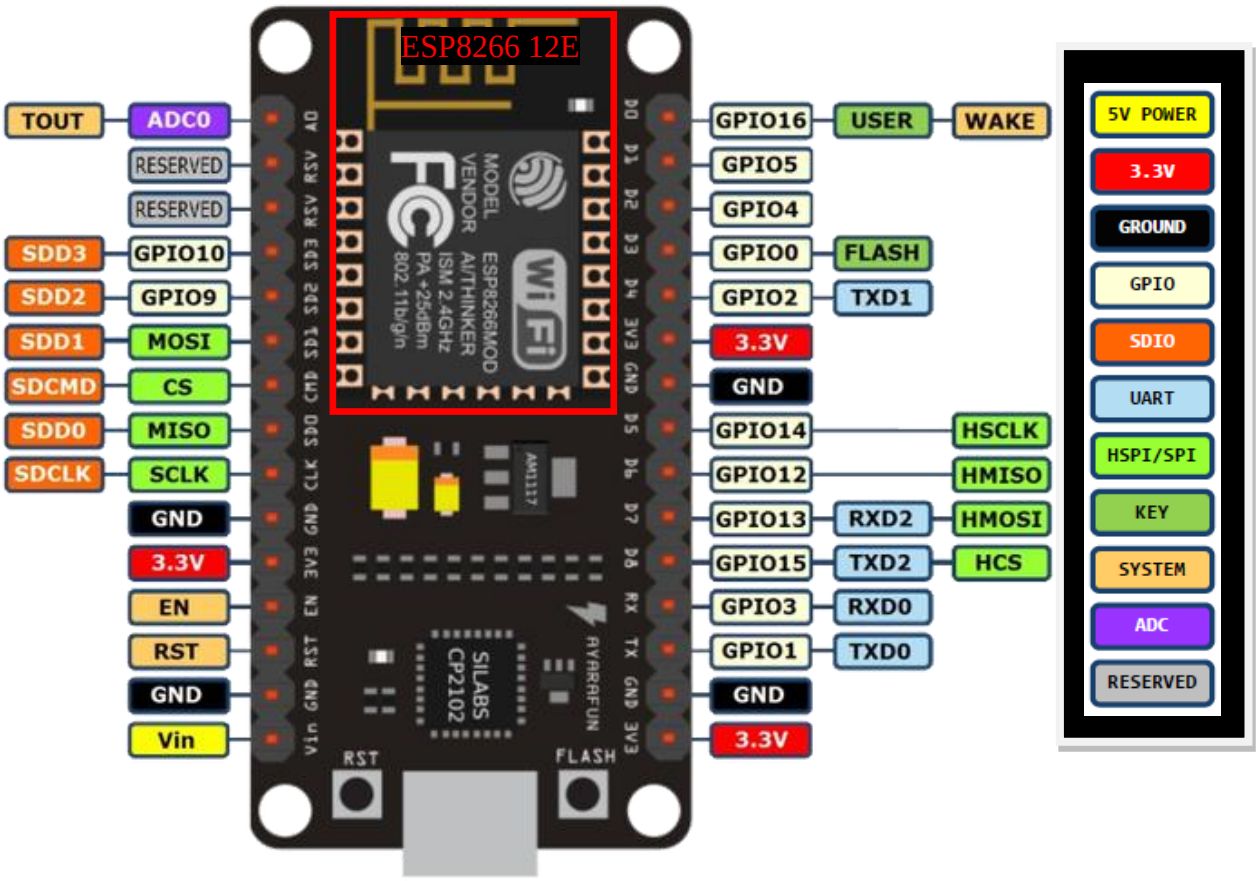
Na slici 3.1.2 vidi se kako je uspostavljena komunikacija klijenta i posrednika. Klijent, koji je indirektno lokalno spojen na posrednika preko WiFi routera, je poslao dvije poruke. Prva poruka je imala temu „test“, a druga „test_2“. Na slici je vidljivo kako su oba dvije poruke proslijeđene. Mosquitto posrednik pravilno obavlja svoj rad. [25]

3.2. ESP8266

NodeMCU DEVKIT V1.0

Postoje tri verzije NodeMCU razvojne pločice: v0.9 , v1.0, i v3. U ovom radu korištena je v1.0 razvojna pločica, još nazivana i v2, koja na sebi sadrži ESP8266 12E SoC. ESP 12E verzija ima 32bit procesor, 4MB memorije za pohranu, 128 kByta radne memorije, 10 ulazno izlaznih pinova opće namjene a svaki ima PWM mogućnost. Uobičajena struja koju pinovi daju na izlaz je 6mA a maksimalna vrijednost koju mogu podnijeti je 12mA.

Na razvojnoj pločici dimenzija 2.5400×4.8260cm nalaze se: mikroupravljač, regulator napona, „flash“ i „reset“ tipkalo , ugrađena plava LED svjetleća dioda, „USB to UART“ komunikacija, spojni pinovi te ostale komponente koje omogućavaju razvoj i testiranja. Već ugrađena „USB to UART“ komunikacija omogućuje povezivanje razvojnog programa s modulom. Na računalu je potrebno imati instaliran „USB to UART“ upravljački program ili eng. driver. Razvojna pločica ima ugrađen regulator napona koji omogućava 5V napajanje preko microUSB kabela. Tim kabelom ujedno i vršimo snimanje programa u modul ili se koristi za upravljačko nadzornu serijsku komunikaciju. Raspored pinova i naziva vidi na slici 3.2.1. [16] [29]



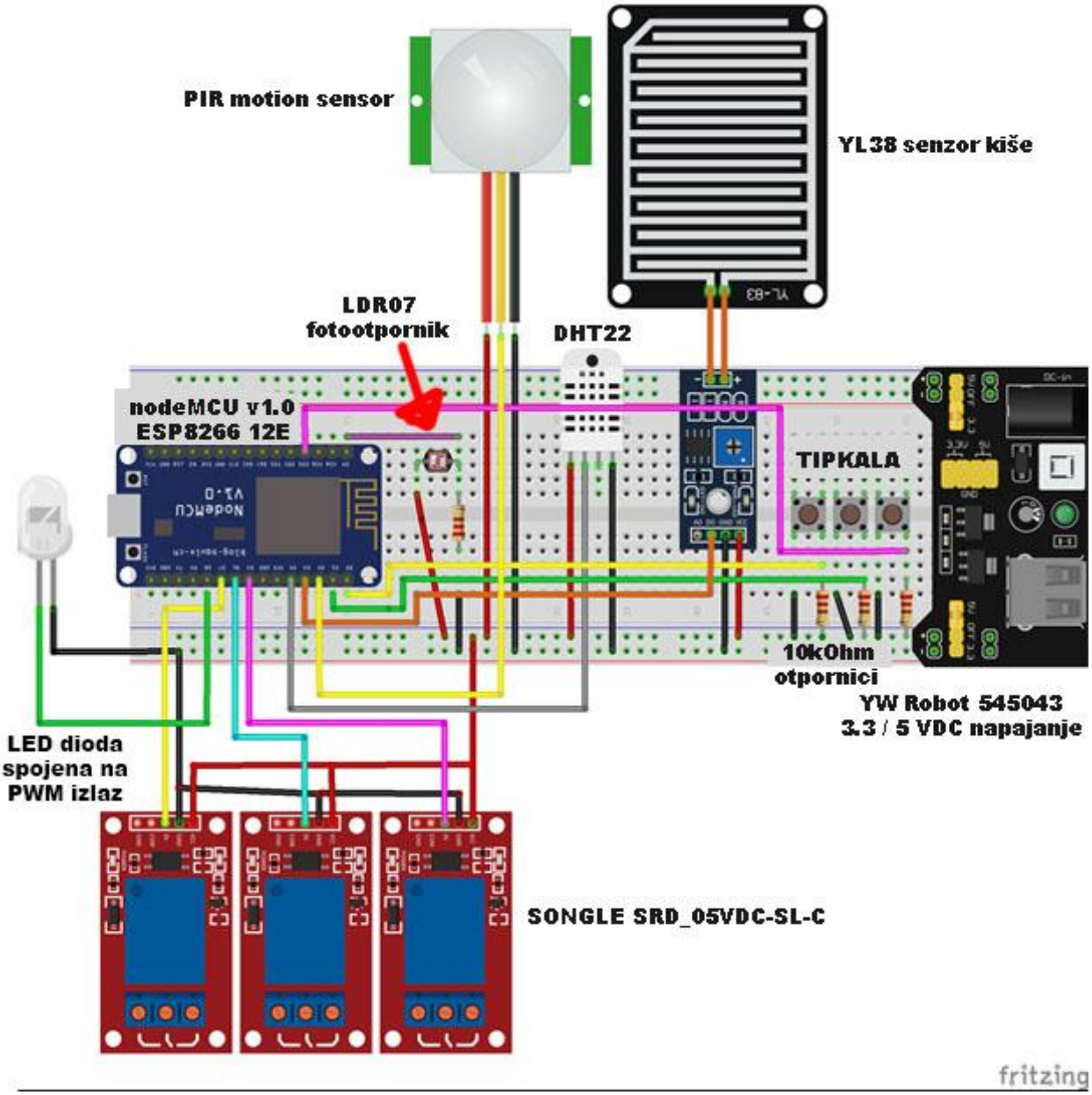
Slika 3.2.1 Raspored i oznake pinova [29]

Arduino IDE

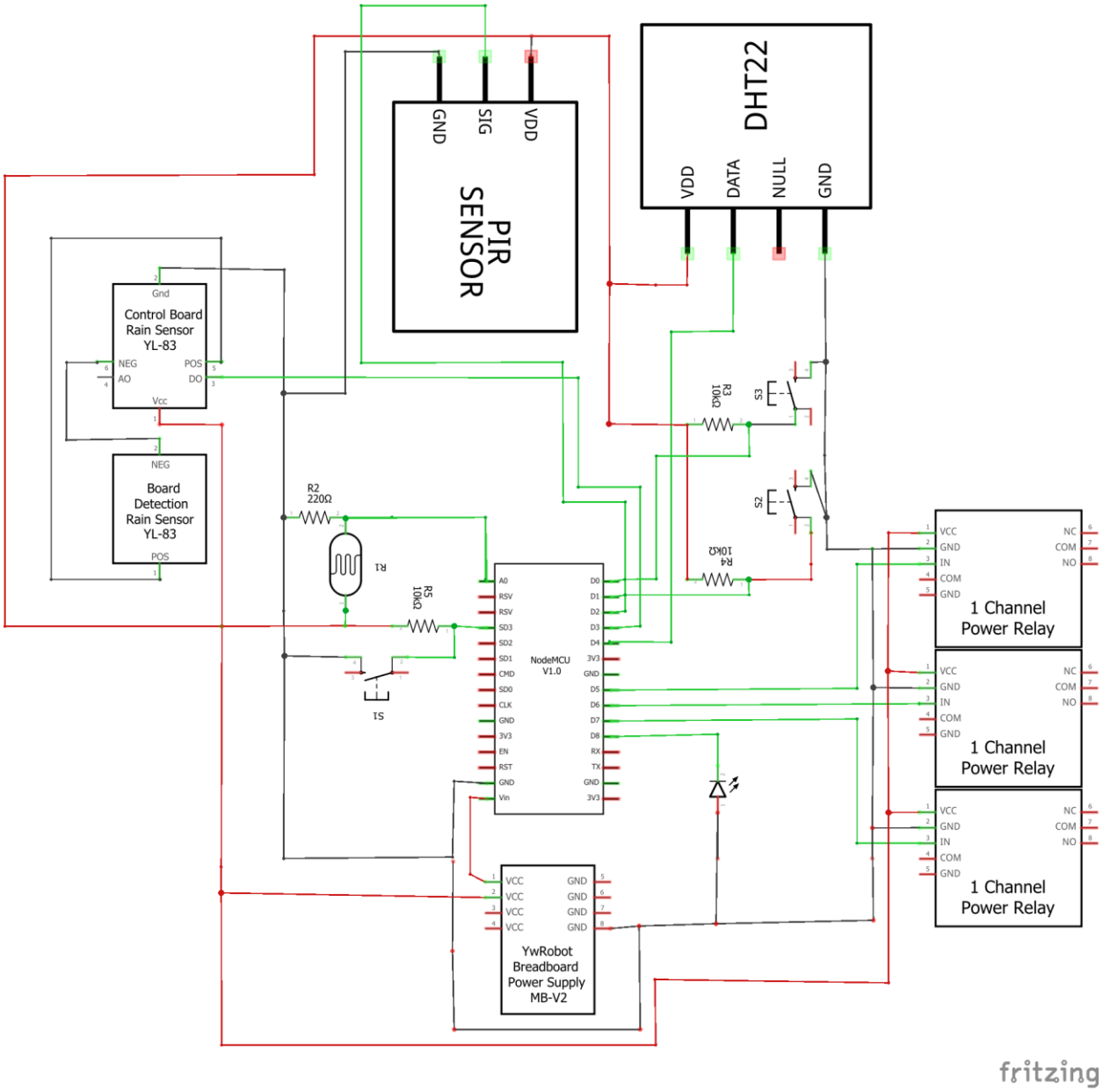
Ukoliko se želi koristiti Arduino IDE za programiranje ESP mikroupravljača mora ga se najprije prilagoditi mikroupravljaču. Otvori se program i klikne na „**File → Preferences**“ . U polje “**Aditional Boards Manager URLs**“ upiše se sljedeče „http://arduino.esp8266.com/stable/package_esp8266com_index.json“. Zatim se otvori „**Tools → Board → Board Manager**“, upiše ESP8266 i instalira. Zatim se ponovno otvara „**Tools → Board“** i izabire mikroupravljač **Generic ESP8266 Module.**

Slika 3.2.1 prikazuje shemu spajanja elemenata na modul u svrhu testiranja na testnoj pločici.

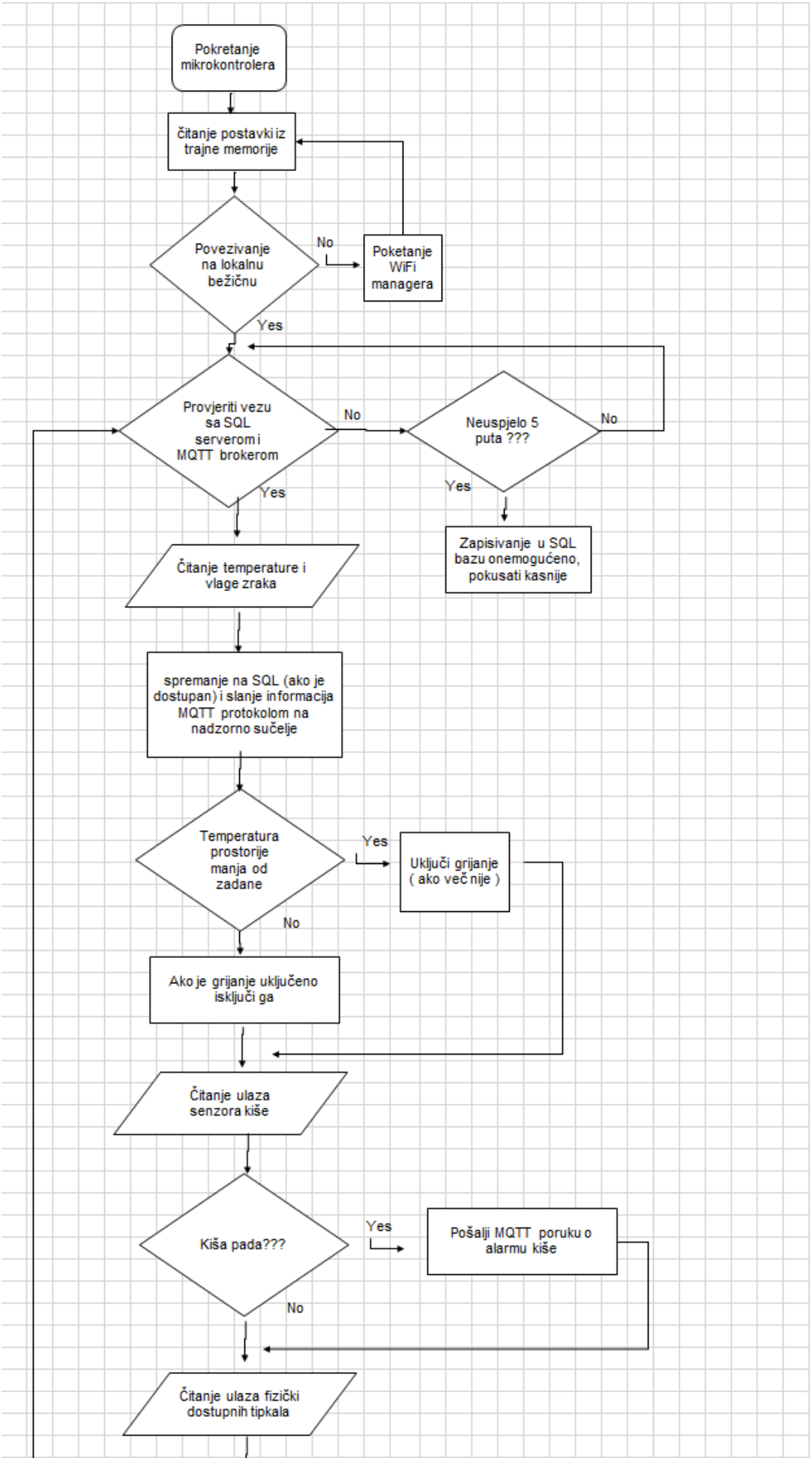
Slika 3.2.2 prikazuje shemu strujnog kruga mikroupravljača i spojenih elemenata. Shema ožičenja i shema strujnog kruga crtane su Fritzing programom. [30]

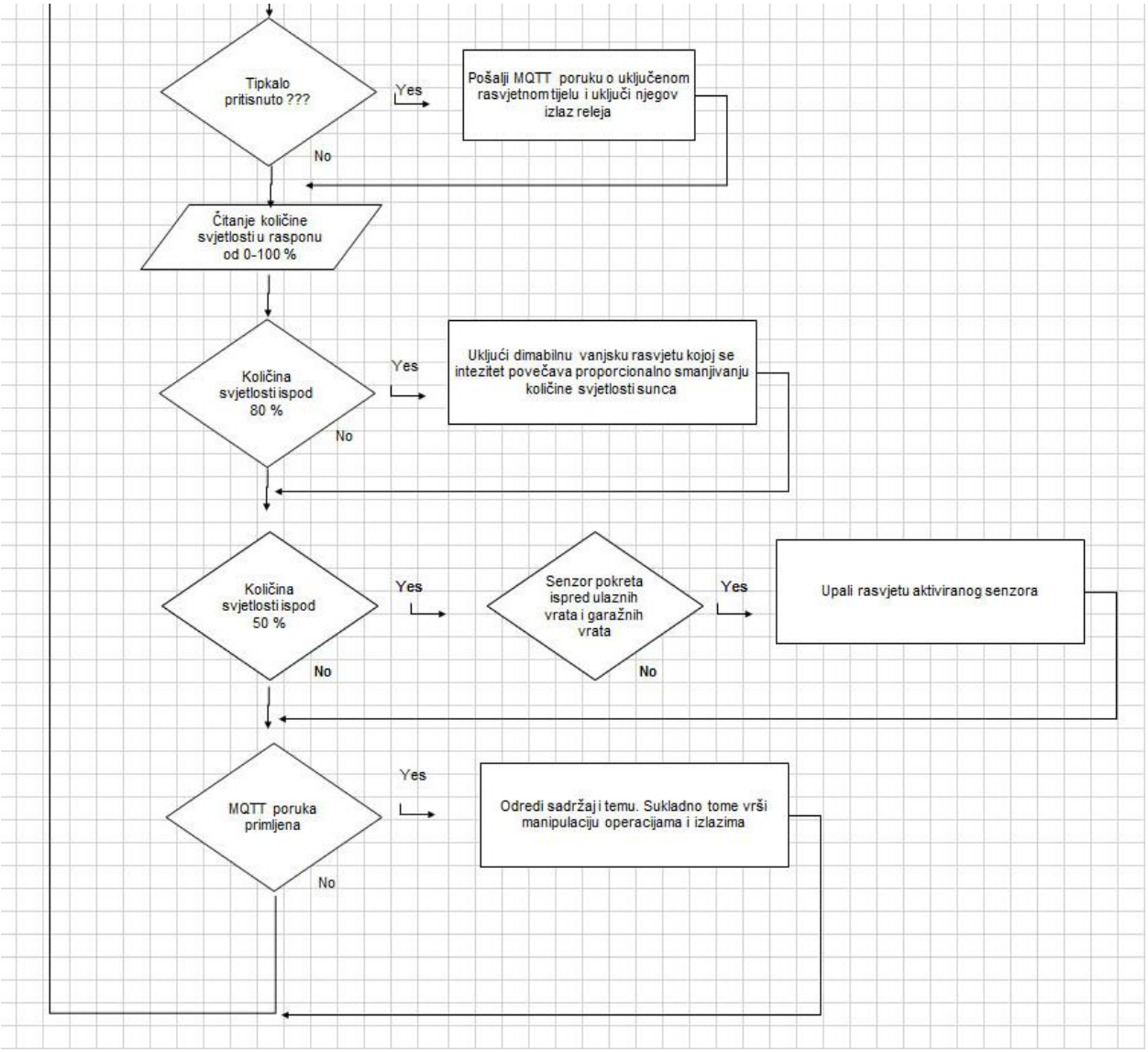


Slika 3.2.2 Shema spajanja



Slika 3.2.3. Shema strujnog kruga


Pokretanje mikrokontrolera
Čitanje postavki iz trajne memorije
Povezivanje na lokalnu bežičnu
No
Poketanje WiFi managera
Yes
Provjeriti vezu sa SQL serverom i MQTT brokerom
No
Neuspjelo 5 puta ???
No
Yes
Yes
Zapisivanje u SQL bazu onemogućeno, pokusati kasnije
Čitanje temperature i vlage zraka
spremanje na SQL (ako je dostupan) i slanje informacija MQTT protokolom na nadzorno sučelje
Temperatura prostorije manja od zadane
Yes
Uključi grijanje (ako već nije)
No
Ako je grijanje uključeno isključi ga
Čitanje ulaza senzora kiše
Kiša pada???
Yes
Pošalji MQTT poruku o alarmu kiše
No
Čitanje ulaza fizički dostupnih tipkala



Slika 3.2.4 Pojednostavljena blok shema tijeka programa mikroupravljača

Povezivanje na MQTT broker

PubSubClient.h knjižnica olakšava povezivanje na MQTT posrednik pozivanjem gotovih naredbi. U početnom postavljanju mora se definirati klijent koji će se spojiti na posrednik naredbama WiFiClient espClient i PubSubClient client(espClient); Iz trajne memorije dohvate se informacije o adresi MQTT posrednika i portu te naredbom client.setServer(mqtt_serverR, mqtt_portT); pridodaju te informacije klijentu. Nakon što su dohvaćeni svi potrebni parametri programskim blokom 3.2.1, u nastavku, povezuje se na MQTT posrednik. Ukoliko posrednik nije povezan modul poziva funkciju reconnect() (programski blok 3.2.2) kojom se povezuje na MQTT posrednik. Uspostavljanjem veze potrebo je poslati ID klijenta kako bi se omogućila komunikacija. Nakon uspješno uspostavljene komunikacije klijentu je omogućeno slanje poruka.

```
1  if (!client.connected() ) {
2    reconnect();
5  }
6  else
7    client.loop();
```

Programski blok 3.2.1

```
void reconnect() {

  while (!client.connected()) {
    String clientId = "ESP8266_1-ID";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      client.subscribe("ESP_1/Relay");
    }
    else {
      Serial.print("greska, rc=");
      Serial.print(client.state());
      delay(5000);
    }
  }
}
```

Programski blok 3.2.2

Ukoliko se žele primati poruke mora se posredniku dati do znanja zainteresiranost za određene teme. Osim toga mora se definirati funkcija koja će vršiti dohvat i kontrolu nad porukama. *client.setCallback(callback);* daje do znanja klijentu da je funkcija callback odgovorna za primanje poruka.

```
1 void callback(char* topic, byte* payload, unsigned int length) {
2 char poruka[50]= {};
3 if (strcmp(topic, "ESP_1/Relay") == 0)
4   {
5     for (int i = 0; i < length; i++) {
6      Serial.print((char)payload[i]);
7       poruka[i]=(char)payload[i];
8       }
9  if (strcmp(poruka, "LEDON") == 0) digitalWrite(BUILTIN_LED, LOW);
10 else if (strcmp(poruka, "LEDOFF") == 0)digitalWrite(BUILTIN_LED, HIGH);
11 }
12 }
```

Programski blok 3.2.3

Primanjem poruke filtriraju se samo one koje u tom trenutku imaju težinu, tj. dohvaćaju se samo informacije iz željenih tema i na njih se reagira (redak 3, 9 i 10 programskog bloka 3.2.3).

ESP Spremanje i učitavanje zadnje spremljenih postavki

Za rad modula potrebne su neke od postavki koje su unaprijed inicijalno definirane za prvo pokretanje, no imaju mogućnost promjene u toku rada na željene vrijednosti. Te su postavke spremljene u datoteci config.json u permanentnoj memoriji modula. Da bi se moglo baratati datotekama u stalnoj memoriji potrebno je postaviti datotečni sustav. Datotečni sustav tip je pohranjivanja i organiziranja računalnih datoteka na medij za pohranu podataka. SPI Flash File System jedan je od datotečnih sustava koji se koristi na mikroupravljačima. U programskom bloku 3.2.4, u nastavku, nalazi se jedan od načina na koji se čitaju podaci iz datoteke. Naredbom *SPIFFS.begin()* pokreće se algoritam datotečnog sustava, zatim provjeri dali datoteka *„config.json“* uopće i postoji naredbom *SPIFFS.exists("/config.json").* Naredbom *SPIFFS.open("/config.json", "r")* otvara se datoteka te preko međupohrane (eng. buffer) dohvaća njen sadržaj, zatim se prilagođava željenoj formi i prosljeđuje informacije na daljnje izvođenje programskog ciklusa.

```
1    if (SPIFFS.begin()) {
2    if (SPIFFS.exists("/config.json")) {
3      delay(20);
4      File configFile = SPIFFS.open("/config.json", "r");
5      if (configFile) {
6        size_t size = configFile.size();
7        std::unique_ptr<char[]> buf(new char[size]);
8        configFile.readBytes(buf.get(), size);
9        delay(20);
10        DynamicJsonBuffer jsonBuffer;
11       JsonObject& json = jsonBuffer.parseObject(buf.get());
12        delay(20);
13        json.printTo(Serial);
14        delay(20);
15        if (json.success()) {
16          strcpy(mqtt_server, json["mqtt_server"]);
17          strcpy(mqtt_port, json["mqtt_port"]);
18          strcpy(mqtt_token, json["mqtt_token"]);
19          strcpy(inTempChar, json["inTempChar"]);
20          strcpy(inRefChar, json["inRefChar"]);
21
22          if(json["ip"]) {
23            Serial.println("setting custom ip from config");
24            strcpy(static_ip, json["ip"]);
25            strcpy(static_gw, json["gateway"]);
26            strcpy(static_sn, json["subnet"]);
27          }
28        else Serial.println("prilagodena adresa nije pronadena");
29          }
30          else Serial.println("neuspjelo otvaranje datoteke");
31      }
32    }
33  }
34 else Serial.println("neuspjeli dohvat datoteke");
```

Programski blok 3.2.4. Čitanje postavki iz datoteke u trajnoj memoriji

Prilikom rada modula neke od postavki će se zasigurno mijenjati češće, neke rjeđe, a neke će ostati nepromijenjene. Kad se postavke pomjerene one moraju biti spremljene u konfiguracijsku datoteku kako bi modul kod sljedećeg pokretanja znao prethodno stanje. Spremanje postavki vrši se prebacivanjem željenih informacija u međupohranu (redak 1 -10). Datoteka se otvara naredbom *SPIFFS.open("/config.json", "w")* . Parametar -*w* daje dozvolu za pisanje. Naredbom *json.printTo(configFile)* zapisuju se postavke iz međupohrane u datoteku. Nakon zapisivanja zatvara se datoteka naredbom *configFile.close().* U nastavku se nalazi programski blok 3.2.5 koji prikazuje jedan od načina spremanja postavki u datoteku.

```
1     DynamicJsonBuffer jsonBuffer;
2     JsonObject& json = jsonBuffer.createObject();
3           json["mqtt_server"] = mqtt_server;
4           json["mqtt_port"] = mqtt_port;
5           json["mqtt_token"] = mqtt_token;
6           json["inTempChar"] = inTemp;
7           json["inRefChar"] = inRef;
8           json["ip"] = WiFi.localIP().toString();
9           json["gateway"] = WiFi.gatewayIP().toString();
10          json["subnet"] = WiFi.subnetMask().toString();
11
12    File configFile = SPIFFS.open("/config.json", "w");
13    if (!configFile) Serial.println("neuspjelo");
14    json.prettyPrintTo(Serial);
15    json.printTo(configFile);
16    configFile.close();
17    delay(10);
```

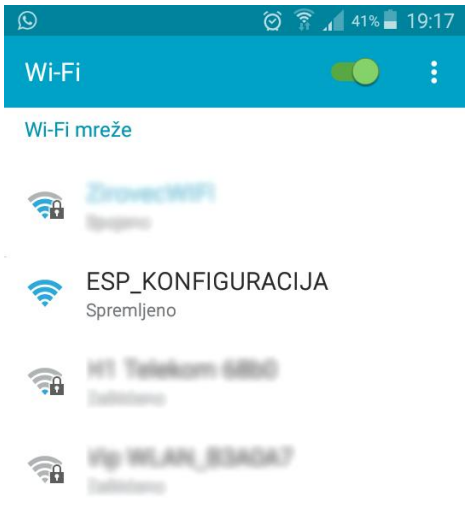
Programski blok 3.2.5. Spremanje postavki u datoteku u trajnoj memoriji

ESP Povezivanje na lokalnu mrežu

Modul je potrebno povezati na lokalnu mrežu bežičnom komunikacijom na lokalni usmjerivač (eng. router) koji ima mogućnost WiFi komunikacije kako bi omogućili telemetriju i upravljanje. Domet klasične kućne bežične WiFi komunikacije na 2.4 Ghz je prema IEEE 802.11(a/b/g/n) standardima 100-150 metara na otvorenom prostoru. Iskustveno, raspon uspješne komunikacije kućnom opremom je ostvariv u krugu od 20-70 m i ta udaljenost znatno ovisi o fizičkim preprekama poput poprečnih zidova koji mogu umanjiti snagu signala i do 25%. Pristup uređaju ostvaruje se kriptiranom prijavom prema WEP (Wired Equivalent Privacy) i/ili WPA/WPA2 (Wi-Fi Protected Access) modelu sigurnosti.

Nakon što se modul uključi na izvor napajanja, on će nakon par sekundi biti spreman za rad. ESP modul će se pokušati povezati na usmjerivač prema zadnje spremljenim postavkama. Ukoliko povezivanje ne uspije ESP se prebacuje u **A**ccess**P**oint način rada što znači da će omogućiti spajanje na sebe pomoću bežične komunikacije.

U postavkama mreže bilo kojeg WiFi uređaja komunikacije odabere se mreža „ESP_KONFIGURACIJA“ (Slika 3.2.5) čija je pristupna lozinka „**admin**“. Nakon uspostavljanja uspješne komunikacije otvorit će se web stranica koja omogućuje unos potrebnih podataka. Ukoliko se web stranica ne otvori automatski, pokrenemo web pretraživač i u adresnu traku upisujemo „**192.168.4.1**“. Na meniju web stranice odabere se „Postavke WiFi mreže“. (Slika 3.2.6.)

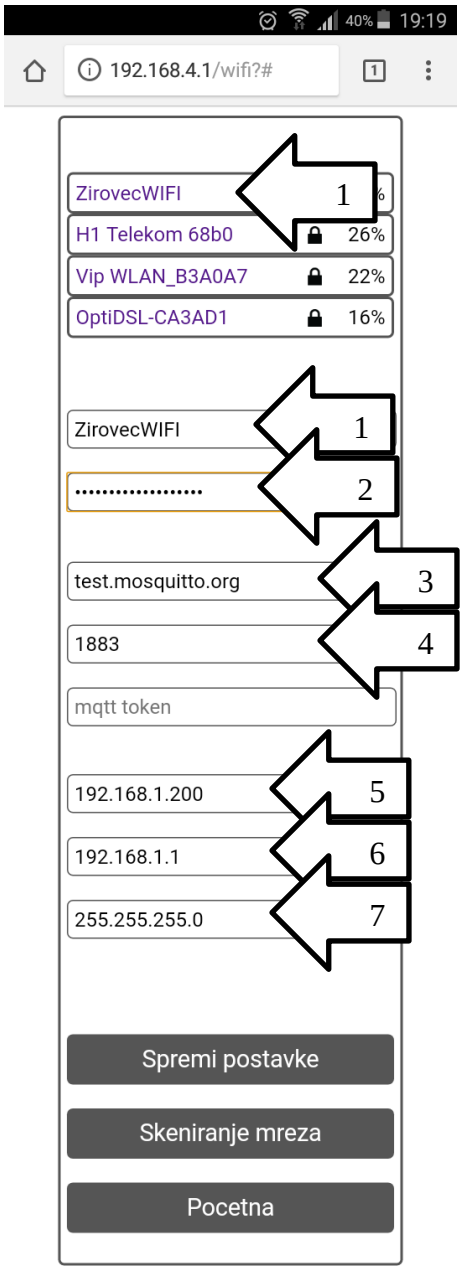




Slika 3.2.5 Wireless mobilno povezivanje Slika 3.2.6. ESP konfiguracijsko sučelje

Pokrenuti prozor nudi mjesta za unos podataka. Na vrhu je popis dohvaćenih WiFi mreža i klikom na željenu mrežu automatski se popunjava prvi prozor koji sadržava ime mreže (SSID) . U sljedeći prozor upisuje se lozinka prethodno odabrane mreže. Nakon toga dolazi okno za upis adrese MQTT posrednika (IP adresa ili URL) te porta na kojem radi. Standardno je to 1883 port. Sljedeće je okno za upis IP adrese koja će biti dodijeljena ESP modulu, pa nakon njega mrežni

izlaz (eng. default gateway) i mrežna maska (eng. subnet mask). Pravilnim upisom podataka potvrđuje se unos pritiskom na gumb „Spremi postavke“. Ukoliko povezivanje nije bilo uspješno postupak se ponavlja. Ukoliko su podaci točni, povezivanje će biti uspješno i ESP modul prelazi iz **AccessPoint** načina rada u **Station** način rada i direktno povezivanje na sam ESP modul više nije moguć, već će sva sljedeća komunikacija teći preko usmjernika. (Slika 3.2.7.)



Slika 3.2.7. Unos podataka za povezivanje na lokalnu mrežu

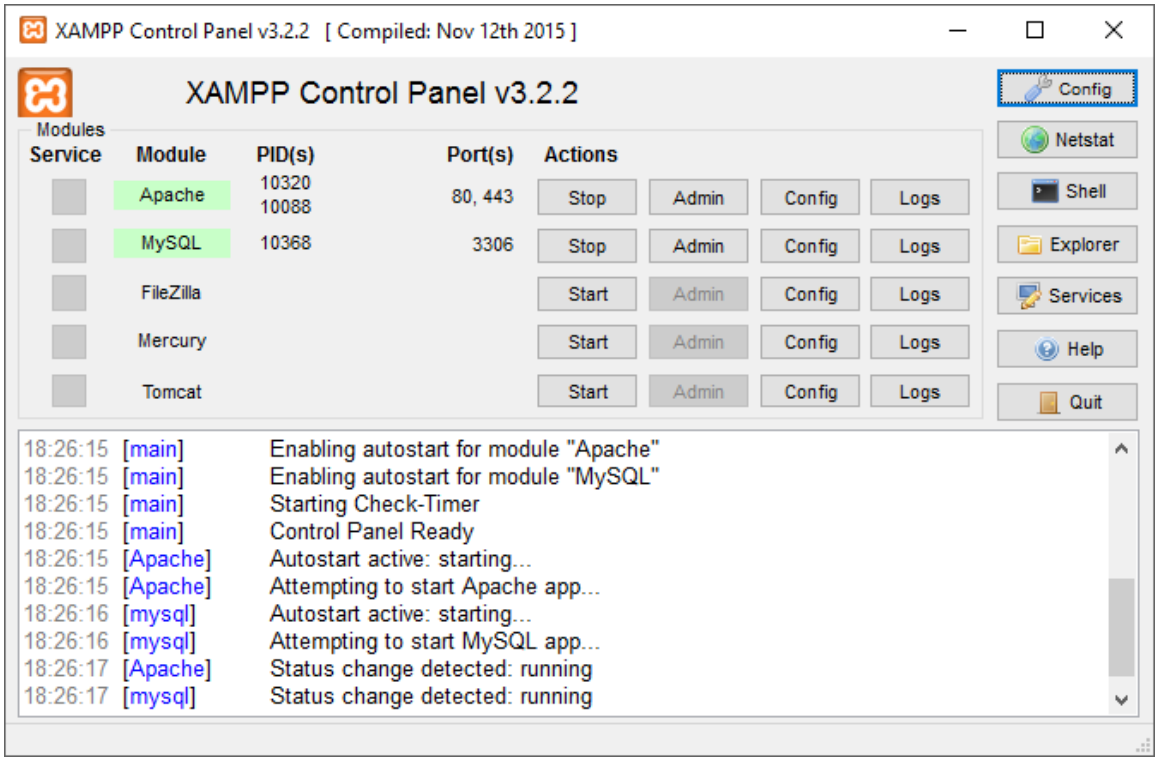
3.3. APACHE server i web stranica

Apache server

XAMPP multiplatforma radi na više operacijskih sustava, između ostalog i na Windows te na Linux. Prikladna verzija preuzme se sa službene XAMPP stanice : „https://www.apachefriends.org/download.html „

Windows:

Nakon preuzimanja datoteke, jednostavnim dvostrukim klikom pokreće se instalacijska datoteka te slijede upute i u par koraka je multi-platforma instalirana. Za pravilan rad aplikacija mora biti instalirana u -*root* direktoriju particije operacijskog sustava. Pokreće se dvostrukim klikom na ikonu „XAMPP Control Panel“ (Slika 3.3.1.)



Slika 3.3.1. XAMPP sučelje

Da bi mogli pristupiti web stranici preko drugih računala potrebno je preurediti konfiguracijsku datoteku httpd-xampp.conf koja se nalazi u direktoriju „\xampp\apache\conf\extra\httpd-xampp.conf“ Apache servera. [12]

```
DocumentRoot "C:/xampp/htdocs"
<Directory "C:/xampp/htdocs">
    Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Require all granted
</Directory>
```

Također, ukoliko je potrebno, omogućuje se siguran prolaz u firewall-u.

Linux: Nakon preuzimanje željene verzije otvara se direktorij u kojem je instalacijska datoteka. U tom direktoriju otvara se i terminal te preko naredbe u nastavku pokreće instalacija.

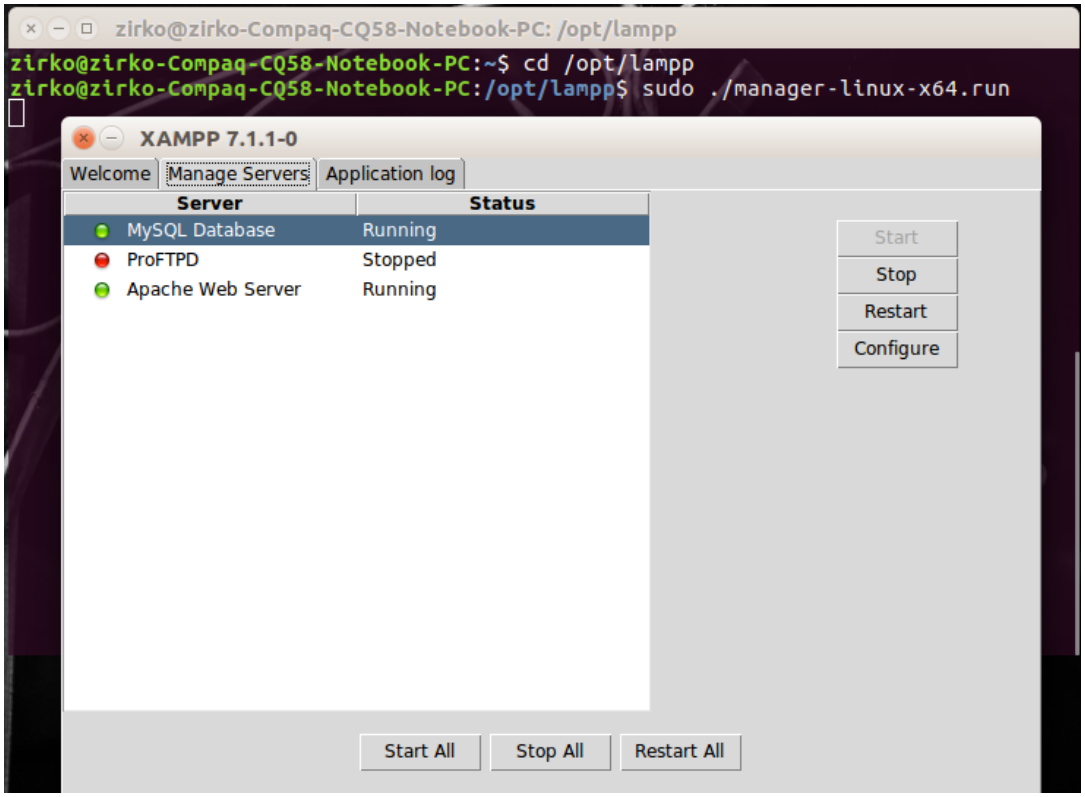
```
sudo ./xampp-linux-x64-5.6.24-1-installer.run
```

Prema uputama instalacije u par koraka završava radnja instalacije. Podcrtani dio naredbe je varijabilan i ovisi o verziji koja je ranije preuzeta. Otvaranjem terminala, odabire se direktorij u kojem je instaliran XAMPP naredbom:

```
cd /opt/lampp
```

Pokrećetanj se vrši naredbom iz terminala:

```
sudo ./manager-linux.run (ili manager-linux-x64.run)
```



Slika 3.3.2. LAMPP platforma na Linux okruženju

U slučaju da je onemogućeno kreiranje mapa i datoteka potrebno je podesiti dozvole pisanja i čitanja sljedečom naredbom:

```
sudo chmod 777 -R /opt/lampp/htdocs/
```

Da bi mogli pristupiti web stranici preko drugih računala potrebno je preurediti konfiguracijsku datoteku httpd-xampp.conf koja se nalazi u direktoriju „\xampp\apache\conf\extra\httpd-xampp.conf“ Apache servera kao u nastavku.

```
Alias /bitnami/ "/opt/lampp/apache2/htdocs/"
Alias /bitnami "/opt/lampp/apache2/htdocs"

<Directory "/opt/lampp/apache2/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
```

Također, ukoliko je potrebno, omogućuje se siguran prolaz u firewall-u.

Nadzor i upravljanje putem web stranice

Nadzor i kontrola sustavom kojim upravlja modul vrši se preko lokalne web stranice smještene na Apache serveru. Web stranici se može pristupiti s bilo kojeg uređaja (računalo, tableta, Smartphonea,...) koji je direktno (na istom računalu) ili indirektno (preko usmjerivača) mrežno spojen na Apache server. Otvaranjem web sučelja, prvo se od modula zahtijevaju ažurni podatci pa se već nakon nekoliko trenutaka dobivaju informacije koje su vezane uz temperaturu i relativnu vlagu prostorije, količini svjetlosti okoline kuće, trenutnom stanju rasvjete pojedinog dijela kuće, indikator pokreta na ulaznom i garažnom dijelu, alarm kiše i dima. U prostoru za podešavanje grijanja upisuje se prvo podatak o željenoj temperaturi prostorije, te stupanj uključenja grijanja ispod željene temperature. Pritiskom na gumb „Potvrdi“ te se informacije šalju i spremaju u modul. Tako će npr. za unijete vrijednosti 27 i 1 temperatura uključenja grijanja biti 26 °C a isključenja postignutim 27 °C. Slika 3.3.3 u nastavku prikazuje web sučelje za kontrolu i upravljanje u internet pregledniku.



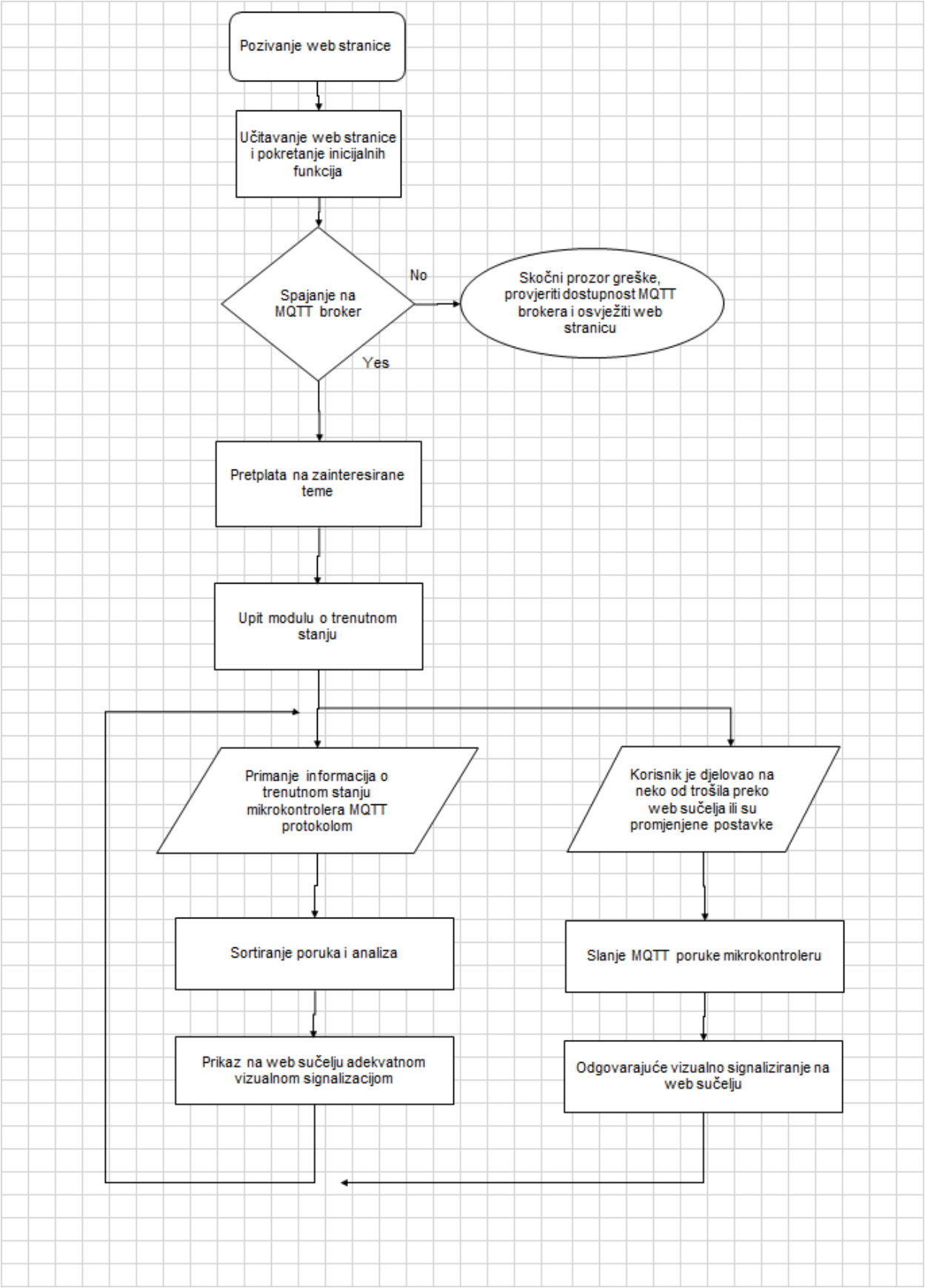
Slika 3.3.3. Nadzor i upravljanje putem web stranice

Da bi sve to bilo moguće potreban je funkcionalan dio programskog koda koji će rješavati MQTT komunikaciju i prema njoj manipulirati stanjima na web sučelju. Taj je dio pisan u javaScript programskom kodu (vidi Programski kod blok 1). Povezivanje na posrednik kreće od kreiranja naredbe klijenta (redak 1-2) kojem je definirana adresa posrednika, websocket porta, te njegove jednoznačne identifikacijske šifre. Prema standardima MQTT protokol vrši komunikaciju između klijenata na 1883 portu, međutim prema web sučeljima ta komunikacija mora ići preko web socketa koji prethodno moraju biti konfigurirani na posredniku. Od 3-7 retka napisana je funkcija kojom je omogućeno slanje poruka posredniku. Funkcija prima parametre : **''sadržaj poruke''**, **''tema''**, **''QOS''.** Primjer slanja poruke nalazi se u retku broj 11. U dijelu programskog koda od 8-13 retka definirane su opcije pretplate na željene teme. Time se do znanja daje posredniku zainteresiranost za određene teme i zaposlili ga da šalje poruke vezane uz te teme. Svi su potrebni parametri upisani i naredbom `client.connect(options);` započinje povezivanje i komuniciranja s posrednikom. Funkcija `client.onMessageArrived` je aktivna po primitku poruke. U toj funkciji vrši se daljnje filtriranje poruka kako bi se pravilno moglo

reagirati na željene poruke i u potpunosti iskoristiti dobivenu informaciju. U primjeru od 15-18 retka vrši se provjera teme poruke (" `ESP_1/inTemp` "). U toj se temi šalju podatci o vrijednosti zadane temperature grijanja. Ukoliko posrednik proslijedi tu informaciju ona će biti prikazana na web stranici.

```
1   var client = new Messaging.Client("192.168.1.102", 8000,
2                "WebClient_ID_" + parseInt(Math.random() * 100, 10));
3   var publish = function (payload, topic, qos) {
4             var message = new Messaging.Message(payload);
5             message.destinationName = topic;
6             message.qos = qos;
7             client.send(message);
8   var options = {
9        timeout: 3,onSuccess: function () {
10            client.subscribe('ESP_1/inTemp', {qos: 0});
11            publish('updateState','ESP_1/Relay',0);},
12        onFailure: function (message) {
13            alert("Connection failed: " + message.errorMessage);}};
14   client.connect(options);
15   client.onMessageArrived = function (message) {
16        if(message.destinationName === "ESP_1/inTemp" ) {
17        document.getElementById("inTempID").value =
18                message.payloadString;}}
```

Programski kod blok 3.3.1.



Slika 3.3.4 Blok dijagram nadzora i upravljanja preko web sučelja

4. Zaključak

Predak kućne automatizacije su sustavi industrijske automatizacije koje čine skupa oprema i zahtijevaju velika znanja iz tih područja. Razvojem tehnologije ta se granica polako pomiče jer je hardver sve jeftiniji a informacije su lako dostupne što je približilo sustave automatizacije ne samo profesionalnom industrijskom okruženju već i amaterskom hobi osmišljavanju projekata. Razina tehnologije koja se infiltrira u živote ljudi svakim se danom sve više povećava pa nije bilo potrebno dugo čekati da se uvuče u domove. Cijene će vremenom padati pa sustavi najmodernije današnje tehnologije, koji se ugrađuju u domove, kroz neko vrijeme neće biti luksuz već realnost svakodnevnog života.

Cilj ovog rada bio je postići upravljivo i energetsko štedljivo okruženje koje će nadzor i kontrolu približiti ljudima. Mikroupravljači s ugrađenom bežičnom komunikacijom omogučili su ugradnju sustava na postojeće instalacije uz iziskivanje tek minimalnih prilagodba. Cjelokupni je sustav smješten na lokalnoj razini. Lokalna bežična mreža je jedina ovisnost koju sustav ima da bi se iskoristio njegov puni potencijal ali su ugrađeni algoritmi koji manipuliraju procesima u slučaju prekida veze kako bi se spriječila nemogućnost izvršenja funkcionalnosti. Kao nadzorno upravljačka jedinica korišteno je web sučelje čijom je implementacijom stečeno znanje u području HTML/CSS/javaScript te PHP web programiranja. Ta komponenta daje jednostavan pristup nadzoru i upravljanju sustavom budući da ne zahtjeva posebne aplikacije a samo korištenje je lako razumljivo. Web sučelje smješteno je na Apache serveru. Njegova konfiguracija i postavljanje zahtijevalo je poznavanje računalnih i sigurnosnih alata te znanja iz računalnih mreža. Podaci se spremaju u kreiranu lokalnu MariaDB bazu podataka. Njeno kreiranje i korištenje zahtijevalo je poznavanje područja baza podataka i SQL upita. Upotrebom mikroupravljača, senzora, indikatora te ostalih izvršnih elemenata utvrđeno je znanje iz građe računala, senzorike, automatskog upravljanja procesima, elektrotehnike te Java i C++ programskih jezika. Obuhvaćen je koliko teoretski toliko i praktični dio sustava automatizacije pa daje dobar temelj za daljnje bavljenje tim velikim područjem koje je ovim radom tek dotaknuto. Izrada sustava za automatizaciju doma je iznimno složen i dugotrajan postupak koji uz nepravilno projektiranje ili nepažnjom može dovesti do nefunkcionalnosti sustava te u krajnjem slučaju materijalnu štetu i ugrožavanje zdravstvene sigurnosti.

Osim što rad pokazuje snagu kućne automatizacije otvara se veliko pitanje o sigurnost, privatnosti, dostupnosti informacija te mogućnosti neželjenog vanjskog djelovanja na takve sustave. Mikroupravljači često ni nemaju dovoljnu snagu za implementaciju nekih od sigurnosnih protokola pa je komunikacija između korisnika i samog sustava često izložena napadima. Razvojem IoT tehnologije područje sigurnosti igrat će veliku ulogu.

Daljnji razvoj, nove spoznaje te investicije u svim nabrojenim područjima tehnologije pametne kuće i općenito Interneta stvari učiniti će pristupačnim i široko rasprostranjenim. Vrlo vjerojatno će ta tehnologija kroz neko vrijeme biti asistent u svakodnevnom životu.

U Varaždinu, _____ *Potpis:* _____

5. Literatura

Knjige se navode:

[1] Adafruit Industries – DHTxx datasheet.pdf
[2] Adafruit Industries - PIR-passive-infrared-proximity-motion-sensor.pdf
[3] LDR07-ETC.pdf
[4] RAIN SENSOR MODULE - OpenHacks.pdf
[5] SRD-05VDC-SL-C-Datasheet.pdf
[6] THE FUTURE OF EMBEDDED AND IoT SECURITY: KASPERSKY OPERATING SYSTEM 2016.
[7] The Definitive Guide to the Internet of Things for Business, 2nd Edition.pdf Syed Zaeem Hosain, CTO, Aeris , 2016.
[8] What is IoT and how dows ARM fit in . The Guardian 18.11.2016.
[9] IoT beyound sensors and passive devices – The rise of smart objects.pdf, CES 2015.
[10] 15 IoT facts you should know.pdf, Konstant infosolutions 06.2016.
[11] MQTT A Practical Guide.pdf , Satya Sankar Sahoo 2014.
[12] INSTALLING, CONFIGURING, AND DEVELOPING WITH XAMPP Dalibor D. Dvorski, March 2007 Canada – Ontario
[13] IoT with Arduino Marco Schwartz 2016.
[14] What is IoT , BUSINESS INSIDER, 19.12.2016.
[15] Building Smarter PlanetSolutions with MQTT and IBM WebSphere MQ Telemetry, International technical support organization , 09.2012.
[16] NODEMCU_DEVKIT_V1.0.PDF , 12.05.2015.
[17] ESP8266EX Datasheet, Espressif 2017.
[18] ESP8266 Pin List.xlsx Espressif 15.11.2014.
[19] Espressif Products Ordering Information.pdf Espressif 30.06.2017.
[20] ESP8266 Technical Reference Espressif 25.05.2017.

Internet izvori:

[21] http://www.crn.com/slide-shows/internet-of-things/300083393/10-iot-companies-to-watch-in-2017.htm dostupan 2.5.2017.
[22] https://www.ibm.com/developerworks/community/blogs/5things/entry/5_things_to_know_about_mqtt_the_protocol_for_internet_of_things?laen dostupan 02.05.2017.
[23] https://kompjuteras.com/apache-http-server-instalacija-i-osnovna-konfiguracija/ dostupan 04.05.2017.
[24] https://e27.co/advantages-disadvantages-internet-things-20160615/ dostupan 6.5.2017.
[25] http://www.steves-internet-guide.com/install-mosquitto-broker/ dostupan 8.5.2017.
[26] https://github.com/adafruit/Adafruit_Sensor dostupan 25.5.2017.
[27] https://www.14core.com/programming-your-esp8266-using-esplorer-guide/ dostupan 25.5.2017
[28] http://cactus.io/hookups/sensors/light/ldr/hookup-arduino-to-ldr-sensor dostupan 25.5.2017.
[29] https://iotbytes.wordpress.com/nodemcu-pinout/ dostupan 27.5.2017.

[30] http://fritzing.org/home/
[31] https://www.adafruit.com/product/189 dostupan 24.6.2017.
[32] http://espressif.com/en/products/hardware/esp8266ex/resources dostupan 27.6.2017
[33] https://www.amazon.com/uxcell%C2%AE-Sensor-Rainwater-Detection-3-3V-5V/dp/B00FZRKLIE dostupan 24.6.2017.
[34] https://thefutureofsmart.wordpress.com/2014/10/04/the-first-iot-device-a-coke-machine/

Popis slika

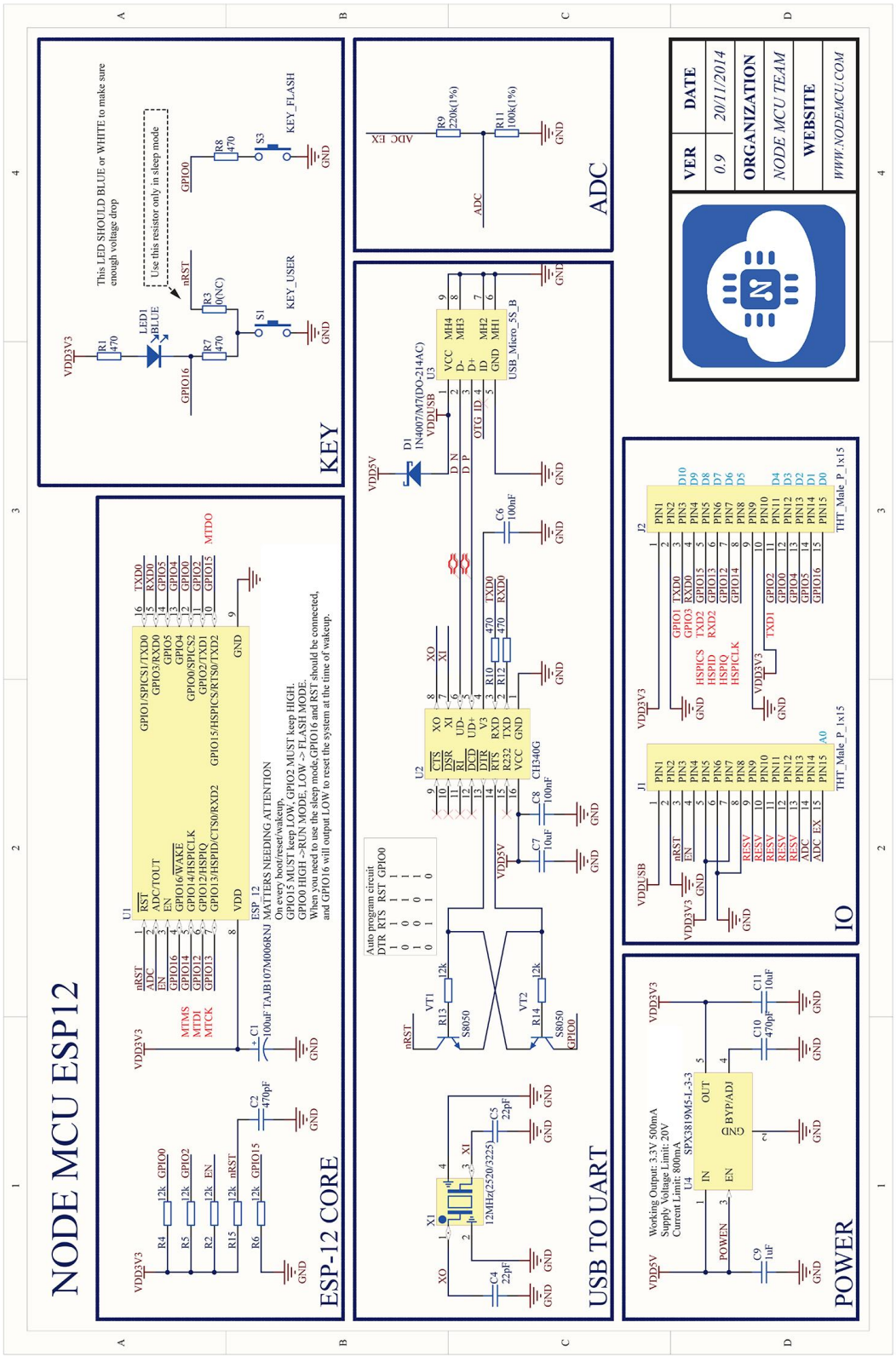
Slika 2.1.1. Broj IoT stvari prikazan u vremenu
Slika 2.1.2. Prikaz IoT arhitekture
Slika 2.1.3. Popis IoT tvrtki
Slika 2.2.1 MQTT posrednik : (eng broker)
Slika 2.3.1. Prikaz radnog prozora Arduino IDE alata
Slika 2.3.2. Izgled ESPlorer razvojnog alata
Slika 2.4.1. Prikaz phpMyAdmin sučelja
Slika 2.5.1. DHTxx senzor
Slika 2.5.2 PIR senzor
Slika 2.5.3. Senzor kiše
Slika 2.5.4 Shema spajanja fotootpornika
Slika 2.5.5 Relej SONGLE SRD_05VDC-SL-C
Slika 3.1.1. HiveMQ posrednik
Slika 3.1.2 Testiranje Mosquitto posrednika
Slika 3.2.1 Raspored i oznake pinova
Slika 3.2.2 Shema spajanja
Slika 3.2.3 Wireles mobilno povezivanje
Slika 3.2.4. ESP konfiguracijsko sučelje
Slika 3.2.5. Unos podataka za povezivanje na lokalnu mrežu
Slika 3.2.6. Pokretanje XAMPP
Slika 3.2.7. LAMPP platforma na Linux okruženju
Slika 3.3.1. Nadzor i kontrola putem web stranice
Slika 3.3.2. LAMPP platforma na Linux okruženju
Slika 3.3.3. Nadzor i upravljanje putem web stranice
Slika 3.3.4 Blok dijagram nadzora i upravljanja preko web sučelja

Popis tablica:

Tablica 2.2.1. Prikaz kontrolnih poruka MQTT protokola
Tablica 2.4.1. Proizvod
Tablica 2.4.2. Skladišna evidencija
Tablica 2.5.1 Karakteristika fotootpornika LDR

PRILOG:

ESP8266 12E STRUJNA SHEMA



Sveučilište Sjever

IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Anđelko Žirovec *(ime i prezime)* pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom Automatizacija u kući primjenom IoT koncepta *(upisati naslov)* te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)
Anđelko Žirovec
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, Anđelko Žirovec *(ime i prezime)* neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom Automatizacija u kući primjenom IoT koncepta i mikroupravljača ESP8266 *(upisati naslov)* čiji sam autor/ica.

Student/ica:
(upisati ime i prezime)
Anđelko Žirovec
(vlastoručni potpis)