PRIMJENA MQTT PROTOKOLA U PROCESU KONTROLE KVALITETE PROIZVEDENIH ELEKTRONIČKIH KOMPONENTI

Bartolić, Srećko

VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

PRIMJENA MQTT PROTOKOLA U PROCESU KONTROLE KVALITETE PROIZVEDENIH ELEKTRONIČKIH KOMPONENTI

Srećko Bartolić

Zagreb, siječanj 2019.

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada".

U Zagrebu, 05.01.2019.

Predgovor

Zahvaljujem se mentoru, profesoru dr.sc. Goranu Đambiću.

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme završnog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

Internet stvari (engl. *Internet of Things*, skraćeno IoT) je skup tehnologija, odnosno koncept koji postoji već duže vrijeme, no tek je u zadnjih nekoliko godina naglo porasla primjena IoT tehnologija. Razlog je većinom jeftinija proizvodnja uređaja i računala malih dimenzija i male potrošnje energije. Ujedno, utjecala je i potreba za pametnijom automatizacijom industrije, što se na internetu može pronaći pod nazivom Industrija 4.0 ili kao četvrta industrijska revolucija. IoT se tako u primjeni može naći u poljoprivredi, domovima pa čak i u cijelim gradovima.

U proizvodnom procesu jako je bitno imati što više informacija u pravo vrijeme, a razvojem IoT-a prikupljanje i obrada podataka postalo je brže, jeftinije i pristupačnije. IoT uređaji generiraju velike količine podataka iz kojih je moguće izvesti informacije te ih upotrijebiti za poboljšanje proizvodnje i održavanja. Vrijednost IoT podataka prepoznale su i velike tvrtke kao Google, Microsoft, IBM i Amazon koje su svoje usluge u oblaku (engl. *Cloud services*) prilagodili i IoT zahtjevima.

Moguću upotrebu IoT-a u industriji ovaj rad će pokazati na primjeru održavanja stabilne temperature u procesu proizvodnje elektroničkih komponenti. Nakon što je elektronička komponenta proizvedena potrebno ju je elektronički ispitati kako bi se provjerila ispravnost, kod ispitivanja vrlo je bitna stabilna temperatura kako bi mjerenja bila što preciznija.

Korištenjem IoT-a takav sustav za održavanje temperature je robustan, malih dimenzija, jednostavan, jeftin i modularan, te ga je uz male izmjene moguće primijeniti i na ostale procese proizvodnje. Podaci prikupljeni takvim sustavom dostupni su online unutar usluga u oblaku za razmjenu podataka sa aplikacijama, spremanje podataka, te obradu podataka pomoću prediktivne analitike.

Prototip sustava izrađen je pomoću Raspberry Pi i ESP8266 uređaja koji međusobno komuniciraju preko *Message Queuing Telemetry Transport* (skraćeno MQTT) protokola u stvarnom vremenu (engl. *real-time*). MQTT protokol dizajniran je za uređaje sa malom potrošnjom energije i za prijenos male količine podataka.

Ključne riječi: Internet stvari, IoT, Usluge u oblaku, Raspberry Pi, ESP8266, MQTT.

Internet of Things (short IoT) is a set of technologies or a concept that has existed for a long time, but only in the last few years has the application of the IoT grown rapidly. The reason for this growth is mostly cheap manufacturing of small-sized devices and computers with low power consumption. But there was also a need for a smarter automation of industry, which can be found on the Internet under Industry 4.0 or as the fourth industrial revolution. IoT can be used in agriculture, homes and even cities.

In the production process, it is very important to get information in the right time, and IoT's development of data collection and processing has become faster, cheaper and more accessible. Thus, IoT devices generate large amounts of data from which information can be derived that relate to improved production and maintenance. The value of IoT data was recognized by large companies such as Google, Microsoft, IBM and Amazon, which adapted their Cloud services to IoT.

Possible use of IoT in the industry will be illustrated by the example of maintaining a stable temperature in the process of manufacturing electronic components. Once the electronic component has been manufactured it is necessary to electronically examine it. It is very important to have a stable temperature for testing to make the measurements as precise as possible.

By using IoT, such temperature maintenance system is robust, small in size, simple, inexpensive and modular, and with small modifications it can be applied to other manufacturing processes. Data collected through such a system is easily available online within Cloud service data storage, which can than be used in data processing with predictive analytics.

The system prototype is made with Raspberry Pi and ESP8266 devices that communicate with each other via a real-time MQTT protocol. The MQTT protocol is designed for low energy devices and for transferring small amounts of data.

Keywords: Internet of Things, IoT, Cloud services, Raspberry Pi, ESP8266, MQTT.

Sadržaj

1. Uvod

Internet stvari skup je tehnologija koje su posljednjih godina dobile maha te se danas mogu pronaći u svim dijelovima ljudskih života. IoT koristi mnogo raznih tehnologija te je ponekad teško snaći se i odabrati pravi set za neki problem. Zbog toga su u ovom radu pobrojane i opisane najvažnije IoT tehnologije današnjice. Velika količina podataka koje generiraju IoT uređaji potrebno je spremiti i obraditi kako bi se iz njih izvukla neka korist. Optimalan i jeftin način za dobivanje vrijednosti iz podataka je korištenje brojnih IoT usluga u oblaku koje u ponudi imaju razne servise za razmjenu, pohranu i analizu podataka.

Kroz drugo poglavlje opisano je što je to Internet stvari te njegove primjene u industriji i općenito u ljudskim životima. Također su pobrojane i opisane tehnologije koje se koriste unutar pojma IoT uključujući hardver, softver i komunikacijske protokole. Dan je i kratki osvrt na podatke koje generiraju IoT uređaji te njihova obrada, a navedeni su i nedostaci IoT tehnologija.

Opisuje se najvažniji protokol za razmjenu podataka između IoT uređaja, a to je MQTT. U trećem poglavlju ukratko su opisane karakteristike protokola te njegove programske implementacije.

Ovome radu cilj je analizirati *online* MQTT posrednike za prikupljanje i distribuciju poruka odnosno podataka unutar komunikacije IoT uređaje, detaljna analiza IoT usluga u oblaku, odnosno usporedbu cijena osnovnih IoT servisa u ponudi kod najvećih IT kompanija današnjice. Uz analizu, cilj je izraditi i prototip sustava za kontrolu temperature primjenom IoT tehnologije i MQTT protokola. Kompletna analiza i izrada praktičnog rada opisana je u četvrtom poglavlju.

Peto poglavlje se bavi povezivanjem komponenti sustava, njegovim testiranjem, te analizom podataka prikupljenih testiranjem.

2. Internet stvari (IoT)

Što je Internet stvari? Stručnjaci se zapravo ne mogu složiti što sve spada pod stvari i gdje je granica. Ja bih IoT okarakterizirao kao mrežu povezanih uređaja (stvari) koja sadrži hardver (elektroniku), softver (programe), senzore, pokretače (engl. *actuator*) i komunikacijske sposobnosti radi prikupljanja i razmjene podataka. Pod uređaje možemo podrazumijevati vozila, bijelu tehniku, nosive uređaje, instrumente te ostalu općenitu tehniku koja do sada nije bila povezana. Internet u nazivu podrazumijeva sposobnost uređaja povezivanjem na Internet radi razmjene podataka, ali i daljinskog nadzora i kontrole istih.

Sam naziv Internet stvari skovan je 1999. godine [1]. Spomenimo, da se pod prvu povezanu tehniku smatra modificirani Coca Cola automat 1982. godine koji je na Carnegie Mellon University-u preko interneta javljao da li ima pića u automatu i kolika im je temperatura [2]. No, prvi ozbiljniji IoT uređaji počeli su se pojavljivati tek 2000.-tih godina, dok je recimo zadnjih desetak godina IoT postao „razorna“ (engl. *disruptive*) tehnologija. [1][3][4]

2.1. Primjena

IoT proizvode možemo pronaći u svakom dijelu ljudskih života, a primjenu IoT tehnologije podijelio bih na dvije osnovne: komercijalnu i industrijsku. Slika dolje (Slika 2.1) vizualno prikazuje komercijalnu i industrijsku granu. [1][3]

U komercijalna područja pripadaju primjene poput automatizacije domova (pametne kuće), transporta i medicine, te svakodnevna primjena.

Automatizacija domova je dosta razgranato područje koje uključuje pametnu bijelu tehniku (npr. Google Nest[1] ili robotski usisavači), uređaje za asistenciju (npr. Amazon Echo ili Google Home[2]), te uređaje i senzore za upravljanje energijom (prekidači, utičnice i žarulje).

U transportu IoT povezuje sve sudionike prometa: vozila, infrastrukturu i korisnike. Sve je veća komercijalna zainteresiranost za pametne i samovozeće automobile u kojima prednost imaju Google, Tesla, Volvo, pa i Uber. Automobili mogu svojim senzorima, ali i informacijama koje dobiju od pametnih cesta (mjerenje brzine, količine prometa i stanja na

[1] https://nest.com/, studeni 2018.
[2] https://store.google.com/gb/product/google_home, studeni 2018.

cestama) dati upozorenja vozačima ili samovozećim automobilima i time uvelike smanjiti prometne nesreće i povećati ugodnost i smanjiti vrijeme putovanja [6].

IoT je u medicini pripomogao u boljoj dijagnostici i bržoj reakciji u nesretnim slučajevima. Daljinsko nadziranje zdravlja povećalo je kvalitetu života pacijenata, a posebice starijih osoba. Senzori za otkucaje srca ili mjerenje krvnog tlaka bežično su spojeni sa centralnim sustavom koji prikuplja informacije za bolju dijagnostiku ili u slučaju problema alarmira medicinsko osoblje.

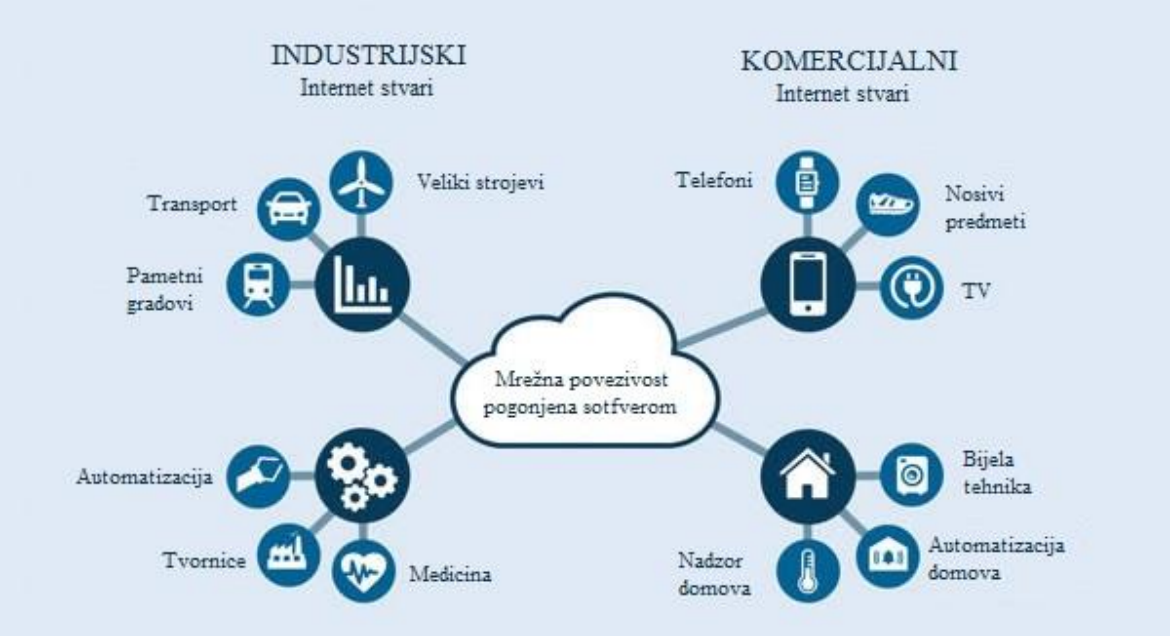
Sa sve pristupačnijim i jeftinijim elektroničkim komponentama i senzorima pametne naprave sve više ulaze i u svakodnevnicu ljudskih života. Tako su pametni satovi, narukvice, ogrlice i prstenje sve popularniji u svim dobnim skupinama.

Industrijska područja pokrivaju automatizaciju tvornica, poljoprivreda i pametne gradovi.

Automatizacijom tvornica IoT je ušao i u područje proizvodnje gdje omogućava bolje proizvode, bržu proizvodnju i dinamički i realni odziv na promjene kombinacijom umreženih strojeva, senzora i kontrolnog sustava. IoT pridonosi i sigurnosti u proizvodnji, održavanju strojeva i infrastrukture, te optimizaciji potrošnje i upravljanje energijom u realnom vremenu.

U poljoprivredi IoT uglavnom služi za prikupljanje podataka o temperaturi, kiši, vlažnosti zraka, vjetru i tlu te time poljoprivrednicima pruža točne podatke i u realnom vremenu. Takvi podaci pomažu kod donošenja pravilnih odluka koje povećavaju kvalitetu i kvantitetu usjeva, a smanjuju potrošnju vode, gnojiva i pesticida.

IoT tehnologija u širokoj primjeni tek je zaživjela u gradovima, gdje se IoT koristi za praćenje kvalitete zraka i vode, za nadziranje tunela i mostova ili upravljanje javnom rasvjetom.



Slika 2.1 Industrijska i komercijalna IoT primjena[3]

2.2. Tehnologije

Različite primjene IoT-a zahtijevaju i različit pristup i tehnologije za najbolju iskoristivost. Tehnologije bih podijelio na pet osnovnih kategorija: komunikacija, protokoli, hardver, softver i podaci. Komunikacija uređaju omogućuje da bude dio IoT mreže, protokoli služe za standardiziranu razmjenu podataka, hardver je obično minijaturan i kompaktan, a softver što jednostavniji. Za razvijanje IoT sustava isprva su korištene tada dostupne tehnologije, no u zadnje vrijeme razvijaju se tehnologije sa posebnom namjenom za IoT. Hardver postaje sve dostupniji, minijaturniji i jeftiniji, a posebno se gleda na energetsku učinkovitost (što manja potrošnja energije je bolja jer se većina IoT uređaja napaja iz baterijskih izvora). Štednja energije odražena je ne samo u hardveru nego i softveru i komunikacijskim tehnologija.

2.2.1. Hardver

Hardver bih podijelio na mrežni hardver, pločice za izradu prototipova, profesionalni IoT uređaje, te na čipove za komunikaciju. [7][8]

[3] Preuzeto sa https://www.i-scoop.eu/internet-of-things/, studeni 2018.

Komunikacijski čipovi dolaze već certificirani za bežičnu komunikaciju i s podrškom za komunikacijske protokole i time olakšavaju ugradnju uz ostali hardver. Najčešći komunikacijski protokoli za ovakve čipove su Bluetooth, Wi-Fi, ZigBee ili RF koje ću opisati u kasnijem poglavlju. Microchip[4] i Texas Instruments[5] samo su od nekih poznatijih proizvođača ovakvih čipova.

Mrežni uređaji su samostalne jedinice visoke snage i procesorske moći koji poslužuje više manjih IoT uređaja kojima omogućuju međusobnu komunikaciju ili povezivanje s vanjskim svijetom (Internetom). IoT uređaji mogu komunicirati raznim protokolima, mrežni hardver podržava sve, a prema Internetu se komunicira sa samo jednim standardom (obično MQTT). Moguće je i filtriranje ili predprocesiranje podataka, a u zadnje vrijeme sve je potrebnija sigurnosna zaštita.

Svaki IoT uređaj potrebno je prvo razviti i testirati, a za to nam služe razne pločice za izradu prototipova. Neke od najpoznatijih su Arduino[6] i Raspberry Pi te njihove brojne derivacije. Većina pločica su gotovi sustavi na koje je potrebno samo spojiti senzore kako bi dobili željenu funkcionalnost. Senzora postoji svih vrsta poput senzora za temperaturu, vlagu, tlak, svjetlost, ultrazvučnih, infracrvenih, pokreta, te mikrofona, kamera i mnogih drugih. Naravno uz senzore moguće je priključiti i pokretače koji će na zahtjev odraditi neku radnju. Pod pokretače ubrajamo motore svih vrsta, klizne staze, releje i ventile, te brojne druge izlazne uređaje. [9]

Pod profesionalnim hardverom smatram uređaje koji su visoke kvalitete i pouzdanosti da se mogu upotrebljavati svakodnevno. Neke i od prije spomenutih razvojnih pločica su dovoljno kvalitetne da se mogu uvrstiti u ovu skupinu. Ovakvi proizvodi trebali bi uključivati i zaštitne kutije sa raznim razinama otpornosti, te standardizirane priključke za senzore i pokretače. Jedan od najpoznatijih proizvođača IoT uređaja je Libelium[7] koji nudi kompletan ekosustav proizvoda. Ponuda im uključuje razvojne pločice, senzore i mrežni hardver. IoT uređaji u njihovoj ponudi se mogu primijeniti u medicini, poljoprivredi, pametnim gradovima i industriji. Na te uređaje spajaju se senzori za promet, parkiranje, osvjetljenje, održavanje voda i otpada, zagađenje, vatru, NFC naplata, temperaturu, ozon, kvalitetu vina, navodnjavanje, UV radijaciju i mnogi drugi. [7][10][11]

[4] https://www.microchip.com/, studeni 2018.
[5] http://www.ti.com/, studeni 2018.
[6] https://www.arduino.cc/, studeni 2018.
[7] http://www.libelium.com/products/, prosinac 2018.

2.2.2. Softver

Pojam softver u IoT-u ima nekoliko značenja s obzirom gdje je instaliran. Softver može biti instaliran na hardver koji čini IoT senzore i uređaje, zatim IoT prolaze (engl. *gateway*) za prikupljanje podataka i sustave u oblaku za obradu i distribuiranje podataka. [12][13]

Softver za *gateway* uređaje čini operacijski sustav, obično neka distribucija Linux operacijskog sustava, plus programi za prikupljanje i agregiranje podataka te slanje istih na pospremanje. Osim za prikupljanje podataka ovaj softver uključuje i sigurnosne programe kao vrlo važan aspekt današnjice te programe za nadziranje statusa samih IoT senzora i pokretača.

IoT softver u oblaku je specijaliziran za prikupljanje, agregiranje, obradu i vizualizaciju velike količine podataka prikupljen o IoT uređaja i senzora. Da je IoT postao velik dio današnjice i Internata, daje dokaz da usluge u oblaku nude svi veliki internetski divovi kao Google, Amazon, IBM i Oracle.

Softver za uređaje i senzore dosta ovisi o tipu hardvera. Oni malo snažniji mogu pokretati neke od distribucija Linux operativnog sustava, dok malo slabiji pokreću neki od *real-time* operacijskih sustava, a najslabiji poput mikrokontrolera izvršavaju samo program koji je upisan u memoriju. Programi za mikrokontrolere se izrađuju u nekom od programskih alata poput Atmel Studio, Arduino Software ili Platform-IO, programski jezik je većinom C/C++ koji se prevodi u strojni jezik. *Real-time* operacijski sustavi (engl. *Real-time operating system*, skraćeno RTOS) koji nemaju sve opcije pravog operacijskog, ali isto tako i nemaju velike hardverske zahtjeve. Takvi operacijski sustavi većinom donose prednost okidanja događaja i višenitno (engl. *multithreading*) izvršavanje programa. Neki od najpoznatijih su Tiny OS, Contiki, LiteOS, FreeRTOS i RIoT. [13]

2.3. Komunikacija

IoT ne bi bio IoT bez komunikacije koja omogućava spajanje na Internet, pri tome upotrebljava se cijeli niz komunikacijskih i telekomunikacijskih tehnologija. Neke tehnologije upotrebljavaju se za komunikaciju između IoT uređaja međusobno ili sa IoT *gatewayem* dok se drugi upotrebljavaju za razmjenu podataka prema Internetu. IoT *gatewayi* obično upotrebljavaju tehnologije više razine, snage i udaljenosti poput 2G/3G/4G i satelitske veze za bežičnu komunikaciju ili pak WAN ili LAN za žičnu. No nekako

komunikacijske tehnologije koje označavaju IoT su Bluetooth[8], Wi-Fi[9], ZigBee[10], NFC[11], RFID[12] pa čak i GSM[13]. [12][14]

Slika 2.2 IoT komunikacijske tehnologije[14]

Dakako postoji još čitav niz komunikacijskih tehnologija koje nismo nabrojali poput Z-Wave[15], WiMax ili komunikacije preko kućanske strujne mreže. Slika gore (Slika 2.2) pokazuje IoT komunikacijske tehnologije s obzirom na udaljenost djelovanja.

2.4. Podaci

Daljinsko upravljanje i nadziranje velika je prednost koju je donijela IoT *disruptivna* tehnologija, no još jedna skrivena prednost jesu podaci. Kako je sve više i više uređaja spojeno na IoT tako se generira sve više podataka od tih IoT uređaja i količina raste eksponencijalno. Podaci uključuju ne samo očitanja senzora nego i statuse uređaja i ostale

[8] https://www.bluetooth.com/, prosinac 2018.
[9] https://www.wi-fi.org/, prosinac 2018.
[10] https://www.zigbee.org/what-is-zigbee/, prosinac 2018.
[11] https://nfc-forum.org/what-is-nfc/about-the-technology/, prosinac 2018.
[12]https://www.electronics-notes.com/articles/connectivity/rfid-radio-frequency-identification/standards-iec-iso-epcglobal.php, prosinac 2018
[13] https://www.etsi.org/technologies-clusters/technologies/mobile/2g, studeni 2018.
[14] Preuzeto sa https://www.postscapes.com/internet-of-things-technologies/, prosinac 2018.
[15] https://z-wavealliance.org/z-wave-alliance-overview/, studeni 2018.

meta podatke. Analizom podataka dobivamo izvješća, grafikone, vizualizacije i upozorenja kako bismo pratili spojene uređaje i vizualno predočili podatke. Analiza nam služi otkrivanju uzoraka, detektiranje anomalija, ili izrada prediktivnih modela, te odrađivanje akcija prema određenim pravilima.

IoT uređaji imaju mali kapacitet memorije općenito tako da se podaci ako ih želimo spremati ne mogu pohranjivati lokalno. Podaci se zbog toga šalju preko IoT *gatewaya* gdje se spremaju i obrađuju i/ili se šalju dalje na udaljenu lokaciju. Na udaljenoj lokaciji (serveru) nije samo problem kapaciteta za spremanje podataka nego i sama raznolikost podataka te briga oko sigurnosti i privatnosti podataka. Zbog raznolikosti podataka koje smo već spomenuli, očitanja senzora, statusa uređaja i ostalih, često ne postoji jedno rješenje za sve. Prema tipu analize podataka podaci se mogu kratkotrajno pospremati na brze memorije ili se trajno spremaju na velike diskove. Prema tome se razlikuju i tehnologije baza za pospremanje podataka koje su većinom NoSQL baze tipa MongoDB, ili neke od Apache baza, ali mogu i uključivati baze specifične za temporalne tipove podataka kao Prometheus, Graphite ili InfluxDB koje su idealne za IoT očitanja senzora. [15][16]

2.4.1. Big Data

Big Data je izraz koji se povezuje sa velikom količinom, raznolikošću i brzinom generiranja podataka. Količina podataka koji se generiraju na Internetu porasla je drastično pojavom socijalnih mreža a spajanjem IoT uređaja na Internet količina podataka još je skočila. Kako je imati pravodobnu i ispravnu informaciju dobivenu iz na oko kaotičnih i nepovezanih podataka velika prednost, *Big Data* je našao mjesto u raznim sferama ljudskih života. Tako *Big Data* analize možemo naći u državnoj upravi, proizvodnji, medicini, edukaciji, medijima, osiguranju, sportu, znanosti i drugima. Do pojave IoT-a *Big Data* analize su se bavile većinom podacima generiranim ljudima i proizvodile predviđanja za planiranje kapaciteta, praćenje kupaca, zaštite zarade. IoT podaci se razlikuju u tome što su to masivni podaci generirani od strane senzora koji zahtijevaju *real-time* analizu i promjenu informacija na prediktivno održavanje, optimizaciju troškova i procesa. Zbog toga se *Big Data* način pospremanja, rukovanja i analize podataka trebao promijeniti. Prema studiji koju proveli iz Gartner-a[16] previđaju da bi do 2020. IoT uređaja moglo biti oko 26 milijardi što je oko 4 puta više nego mobitela, tableta i PC-a u istom predviđanju. Isto tako previđaju da bi IoT mogao

[16] https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016, studeni 2018.

donesti godišnji promet od 300 milijardi što je jako profitabilno za kompanije koje se time bave [5]. Pro prije svega u svoj toj silnoj količini podataka potrebno je imati pravovaljane i dobro analizirane podatke kako bi oni mogli donesti vrijednost. Zbog toga IoT zahtjeva i zahtijevati će od kompanija koje se bave *Big Data* analiza da promijene svoje postojeće procese, alate i tehnologije kako bi se prilagodile zahtjevima IoT-a.

Prva stvar koja nam pada na pamet kad se spomene *Big Data* i IoT je pospremanje podataka. Veliko povećanje generiranih podataka zahtjeva prilagodbu podatkovnih centara i njihovu strukturu. Zbog toga većina kompanije prelazi na model platforma kako usluga (engl. *Platform-as-a-Service,* skraćeno PaaS) te ne koriste svoju infrastrukturu za pospremanje podataka već prelaze na usluge u oblaku. Lokalizirano pospremanje podataka zahtjeva konstantno povećanje kapaciteta kako se povećava količina podataka, dok PaaS nudi fleksibilnost, skalabilnost i sigurnost.

Sam način pospremanja podataka je također drugačiji, većina kompanija koristi Hive[17] ili Hadoop[18] za pospremanje podataka, ali NoSQL baze podataka su pogodnije za pospremanje IoT podataka zbog njihove velike propusnosti i niske latencije. Takvi tipovi baza ujedno podržavaju veliku fleksibilnost i nestrukturirane podatke baš kao i IoT podaci, te s time omogućuju lagano dodavanje novih tipova izvora podataka. [15]

2.4.2. Cloud usluge

Pospremanje podataka u oblak razlikuje se od lokalnog po tome što se podaci spremaju u logičke cjeline. To znači da su fizički podaci spremljeni na nekoliko servera, a ponekad i na nekoliko lokacija, tu spada i cijela infrastruktura od lokacija, zgrada, energije do ljudi koji održavaju i čuvaju objekte. Smještaj podataka na nekoliko lokacija, često na različitim tektonskim pločama, je velika prednost za dostupnost podataka tijekom nesreća i prirodnih katastrofa. Ljudi i/ili kompanije mogu kupiti ili iznajmiti skladišni prostor za sve vrste podataka, a ne moraju se brinuti o infrastrukturi. Pristup podacima omogućen je preko web sučelja, aplikacijskog programskog sučelja (engl. *application programming interface,* skraćeno API) i stolne (engl. *desktop*) aplikacije. Najveće prednosti usluga u oblaku su cijena (hardver, softver, infrastruktura, održavanje, energija), fleksibilnost i samoposluživanje (engl. *self-service,* koje omogućuje sa nekoliko klika miša zakup novih resursa ili otpuštanje neiskorištenih), globalna pokrivenost (velika raširenost podatkovnih centara omogućuje

[17] https://hive.apache.org/, prosinac 2018.
[18] https://hadoop.apache.org/

pokrivenost cijelog svijeta sa kvalitetnom uslugom), produktivnost (samoposluživanje omogućuje da se IT timovi bave sa poslovnim ciljevima), performanse (podatkovni centri stalno se dograđuju na novi i kvalitetniji hardver) te sigurnost (usluge u oblaku zaštićene su širokim spektrom politika, tehnologija i kontrola). [17]

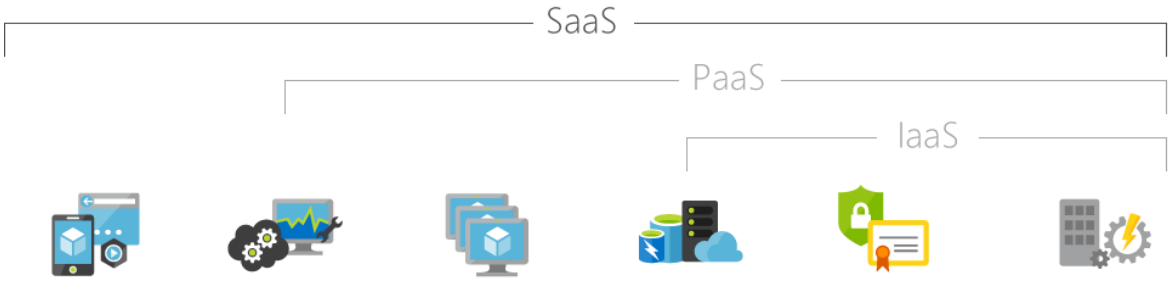
Kako bi se pokrio što veći raspon potreba korisnika postoji tri tipa usluga u oblaku u smislu ugradnje odnosno implementacije.

Javne usluge koje održavaju kompanije specijalizirane su za tu vrstu usluge poput Microsofta, Googlea ili Amazona. Sav hardver, softver i infrastruktura je u njihovom vlasništvu i održavanju. Korisnici pristupaju servisima i podacima preko internet preglednika.

Privatne usluge se odnose na resurse koje koristi jedna kompanija ili organizacija. U većini slučajeva svi resursi su fizički locirani lokalno kod korisnika. No neke kompanije se opredjeljuju da im treća strana održava privatni oblak.

Hibridne usluge su kombinacija pospremanja podataka lokalno i upotreba javnih usluga u oblaku. Taj način je koristan kada osjetljive podatke želite zadržati kod sebe a usluge u oblaku upotrebljavati za analize i pospremanje ostalih podataka. Hibridni pristup pruža dosta fleksibilnosti ali sve više velikih kompanije prelazi na kompletne javne usluge.

Postoji i razni tipovi usluga u oblaku uz koje se vežu pojmovi infrastruktura kao usluga (engl. *Infrastructure-as-a-service*, skraćeno IaaS), softver kao usluga (engl. *Software-as-a-service*, skraćeno SaaS) i računalstvo bez poslužitelja (engl. *serverless*), te već spomenuti PaaS. Slika dolje (Slika 2.3) pokazuje IaaS, PaaS i SaaS usluge u međusobnom odnosu.



Slika 2.3 Vrste usluga u oblaku[19]

[19] Preuzeto sa https://azure.microsoft.com/en-in/overview/what-is-saas/, prosinac 2018.

2.5. Nedostaci

IoT je dosta složen koncept koji povezuje puno drugih tehnologija iz svih područja IT i elektrotehničke industrije. Povezano sa velikom složenosti i relativno novim konceptom sigurno ima i nekoliko nedostatka i kriticizma koje ću ukratko dotaknuti u ovom poglavlju [18][19].

Stručnjaci / Radna snaga – u svijetu je veliki nedostatak stručnjaka i radne snage za IoT tehnologije zbog širokog spektra korištenih tehnologija.

Upravljanje i sigurnost podataka – većina kompanija koristi usluge u oblaku te su njihovi podaci u rukama pružatelja usluga u oblaku. Zbog toga se postavlja pitanje koliko su ti podaci sigurni od zlouporabe što od pružatelja usluga, što od treće strane koja može do podataka doći nelegalno ili pak kupiti podatke od pružatelja usluga.

Privatnost – senzori u pametnoj kući ili automobilima skupljaju danonoćno podatke te se postavlja pitanje koliko je narušena privatnost korisnika pametnih usluga od pružatelja istih.

Nehomogenost tehnologija – velika raznolikost tehnologija i mala količina priznatih standarda otežava ulazak u IoT granu tehnologija i suradnju sa ostalim pružateljima IoT usluga. Većina proizvođača razvija kompletan asortiman tehnologija od hardvera, komunikacijskih protokola do softvera.

Utjecaj na okoliš – pošto bi se uskoro količina IoT uređaja mogla brojati u milijardama postavlja se pitanje što sa reciklažom te ogromne količine senzora i popratnih uređaja.

Sloboda korištenja komercijalnih uređaja – kada se IoT kompanija ugasi ili je kupljena od druge strane što sa uređajima, programima ili serverima u oblaku. Događa se da nova kompanija jednostavno prestane podržavati tehnologije kompanije koju je kupila.

Terminologija – različite tehnologije i sva mjesta na kojima se IoT primjenjuje imaju svoje skraćenice i terminologije, a i svaka IoT kompanija osmišljava svoje nazive na tehnologije koje mogu imati istu svrhu kao i kod drugih proizvođača.

Gubitak radnih mjesta – postavlja se pitanje što sa svim radnim mjestima koje će zamijeniti IoT tehnologije kao u recimo pametnim gradovima i automatiziranim tvornicama.

3. MQTT

MQTT[20], ukratko, je IoT komunikacijski protokol za razmjenu poruka između dva uređaja.

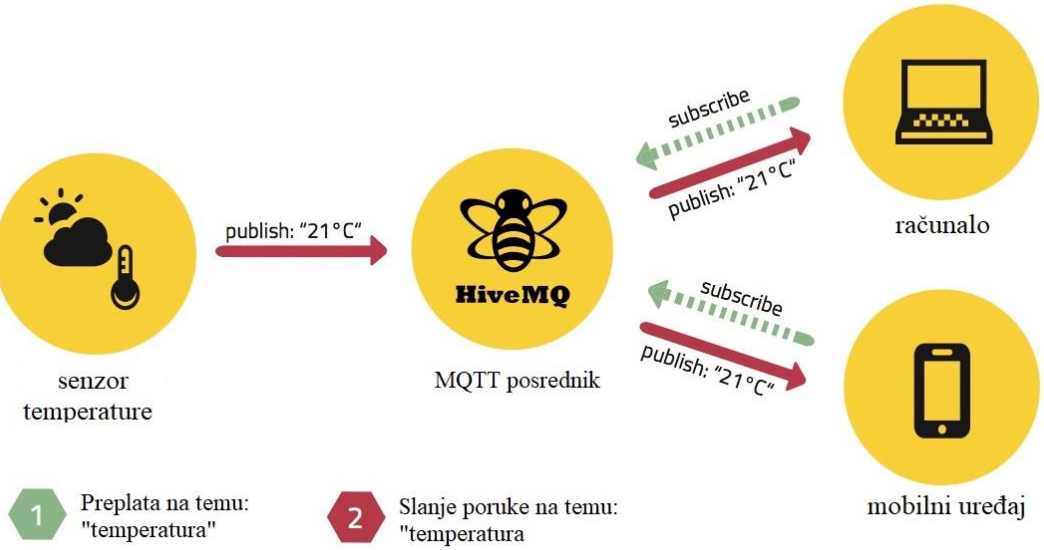
Dizajniran je kao iznimno lagan i jednostavan objava/pretplata (engl. *publish/subscribe*) protokol, te je koristan kod konekcija na udaljenim lokacijama gdje je potreba mala količina koda i/ili je mala propusnost bežične komunikacije.

MQTT je 1999. godine nastao u IBM-u, 2014. godine 3.1.1 verzija protokola priznata je kao standard od strane OASIS-a, 2016. godine kao ISO standard ISO/IEC 20922:2016, a od 2018. godine dostupna je 5.0 verzija. MQTT je prije bio poznat kao skupina SCADA protokola, no oba protokola danas više nisu u upotrebi. Protokol je besplatan i mnoge su ga kompanije uključile u svoje proizvode [20].

Publish/subscribe znači da se sustav sadrži od servera i više klijenata, server još ima naziv posrednik (engl. *broker*), a klijent može biti *publisher* koji šalje informacije ili *subscriber* koji se pretplaćuje na informacije određene teme. Informacije su organizirane u teme (engl. *topics*) koje imaju strukturu stabla, te se klijent pretplaćuje na određenu temu ili cijelu granu. Slika dolje (Slika 3.1) grafički pokazuje načelni rad p*ublish/subscribe* komunikacije. Kada *publisher* ima novu informaciju, on šalje kontrolnu poruku sa informacijom posredniku. Posrednik šalje informaciju svim klijentima koji su se pretplatili na tu temu. Tako *publisher* ne zna ništa o ostalim klijentima, dok pretplatnici ne moraju imati nikakve postavke u vezi *publishera*. Klijenti obično komuniciraju samo sa jednim serverom no sustav može sadržavati više posrednika koji međusobno razmjenjuju informacije. Tako npr. u tvornici imamo više posrednika za svaku zgradu koji šalju informacije na centralni server, a on pak može slati sve u oblak na daljnju obradu i distribuciju [21].

Jedna minimalna MQTT poruka može biti veličine od 2 bajta, dok velike poruke (engl. *Binary large object*, skraćeno BLOB) mogu ići i do 256 MB ako je potrebno. Postoji 14 različitih tipova kontrolnih poruka, vidljivo u tablici (Kontrolne_MQTT_poruke.xlsx) u prilogu. Osnovni tipovi poruka su *Connect/Disconnect* koje služe za otvaranje/zatvaranje konekcije od klijenta prema serveru, *Publish* za slanje informacija, *Subscribe/Unsubscribe* za pretplatu ili ukidanje pretplate na određenu temu/teme poruka, i *Ping* za nadgledanje kvalitete konekcije [22].

[20] http://mqtt.org/, studeni 2018.



Slika 3.1 Primjer razmjena poruka pomoću MQTT protokola[21]

Nivo kvalitete slanja poruke (engl. *Quality of Service*, skraćeno QoS) određuje kako će se poruka poslati i mora se navesti za svaku poruku poslanu preko MQTT protokola. Važno je pravilno odrediti nivo QoS za svaku poruku, jer ona određuje način komunikacije između klijenta i servera. Postoje tri QoS nivoa:

- QoS 0 (najviše jednom) – poruke se dostavljaju kako je to najbolje moguće unutar sustava, te je moguće da će se poruka izgubiti tijekom komunikacije.
- QoS 1 (barem jednom) – osigurano je da će poruke stići na odredište ali postoji mogućnost pojave duplikata poruka.
- QoS 2 (točno jednom) – osigurano je da će poruka stići točno jednom.

Povećanjem razine QoS dobiva se veća sigurnost isporuke poruka ali pod cijenu performansi. Odabirom nivoa QoS MQTT nudi fleksibilnost IoT uređajima kod odnosa slanja poruka i sistemskih zahtjeva [22].

Glavne negativnosti MQTT protokola su manjak sigurnosti jer nema ugrađenu mogućnost enkripcije, minimalne mogućnosti identifikacije pošto se korisničko ime i lozinka šalju kao običan tekst, i nedostatak interoperabilnosti jer su poruke u binarnom obliku bez informacija kako su kodirane što stvara probleme kod komunikacije između različitih platformi [23].

[21] Preuzeto sa https://www.hivemq.com/blog/how-to-get-started-with-mqtt/, studeni 2018.

3.1. Implementacije

MQTT je standardizirani protokol i za njegovu upotrebu potrebno ga je implementirati. Implementacija protokola se najviše razlikuje prema razini sustava kojoj je program namijenjen. IoT senzori većinom imaju malo resursa pa je program što laganiji, a serveri i *gatewayi* imaju više resursa pa implementacija može imati i više mogućnosti, dok sustavi u oblaku uz podršku za MQTT nude arhiviranje i analizu prikupljenih podataka. Uz već spomenute hardverske zahtjeve implementacije mogu podržavati i razne verzije protokola, kvalitetu servisa QoS te ostale softverske zahtjeve.

Tako se implementacija i korištenje MQTT protokola može pronaći u svim dijelovima industrije i ljudskih života gdje je potrebna brza i jednostavna izmjena poruka kao recimo u održavanju i nadgledanju tvornica, pametnim kućama, logistiki, proizvodnji i medicini. Kao primjer mogu navesti Facebook, koji je u svoj Messenger ugradio neke dijelove MQTT protokola, a Amazon, Microsoft i Adafruit nude usluge u oblaku sa podrškom za MQTT.

3.1.1. Programska rješenja

U ovome ću poglavlju kratko opisati i usporediti neka od programskih implementacija MQTT protokola sa namjenom korištenja na IoT senzorima i *gatewayima*. Glavna podjela programskih rješenja je prema njihovoj namjeni, da li su namijenjeni za MQTT klijente ili servere (posrednike), te prema programskom jeziku u kojem su napisani. Podržana je većina programskih jezika poput C/C++, Clojure, Delphi, Erlang, Elixir, Go, Haskell, Java, Javascript/Node.js, Lua, .NET, Objective-C, Perl, PHP, Python, Ruby i još neki. U tablici (Tablica 3.1) ispod nabrojat ću neke od poznatijih programskih implementacija te pobrojati i usporedi neke od njihovih karakteristika. Jedan od najpoznatijih posrednika je Mosquitto koji ću upotrijebiti u praktičnom dijelu radu [24][25][26].

Tablica 3.1 Implementacije MQTT protokola

Implementacija	Proizvođač	Open source	Programski jezik	Tip	MQTT 3.1.1	MQTT 5.0
MQTT Broker	Bevywise	Ne	C, Python	Posrednik	Da	Ne
wolfMQTT	wolfSSL	Da	C	Klijent	Da	Ne
VerneMQ	Octavo Labs	Da	Erlang	Posrednik	Da	Da
Paho	Eclipse	Da	C/C++, Java, Python, Go	Klijent	Da	Ne
Mosquitto	Eclipse	Da	C, Python	Posrednik	Da	Ne
M2Mqtt	Eclipse	Da	.Net	Klijent	Da	Ne
moquette	Andrea Selva	Da	Java	Posrednik	Da	Ne
JoramMQ	ScalAgent	Ne	Java	Posrednik	Da	Ne
HiveMQ	dc-square	Ne	Java	Posrednik	Da	Da
flespi	Gurtam	Ne	C	Posrednik	Da	Da
EMQ	EMQ	Da	Erlang	Posrednik	Da	Da
Adafruit IO	Adafruit	Da	Node.js,Python	Klijent	Da	Ne

3.1.2. Online posrednici

Kako sve više IoT senzora podržava Wi-Fi, moguće je povezati sustav direktno na Internet, a kao posrednik koristiti jednu od danas mnogih usluga u oblaku. U tome slučaju nije nam potreban lokalno *gateway*/*broker* koji financijski može biti dosta zahtjevan. Ipak, kao što sam već spomenuo, posrednici se mogu međusobno povezivati pa upotreba lokalnog i *online* posrednika zajedno nije isključena. *Online* posrednik, pak, u tome slučaju služi kao centralno mjesto za prikupljanje podataka te njihovu daljnju obradu i analizu, i eventualnu distribuciju na sljedeće posrednike u nizu. Kod većih ponuđača usluga u oblaku MQTT posrednici su u ponudi zajedno sa ostalim *Big Data* uslugama što zaokružuje cjelokupnu ponudu za IoT sustave prikupljanja i obrade podataka. IBM, Microsoft, Amazon, Oracle, Bosch i Google samo su neki od velikih ponuđača IoT usluga u oblaku i posrednika. U praktičnom dijelu rada detaljno ću analizirati sve veće IoT platforme i MQTT posrednike.

3.2. Alternativni protokoli

U ovome poglavlju ukratko ću nabrojati i opisati alternativne *real-time* protokole za razmjenu poruka u IoT sustavu [27].

CoAP[22] (engl. *Constrained Application Protocol*) je internet protokol za razmjenu podataka na čvorovima (IoT uređajima) i mrežama sa manjom propusnošću i sistemskim resursima. Protokol je dizajniran za uređaj-uređaj (engl. *machine-to-machine*, skraćeno M2M) komunikaciju poput pametnih kuća i tvornica. Dizajniran je da se spoji sa HTTP-om za integraciju sa Web-om a pri tome osigurava jednostavnost i male zahtjeve.

STOMP[23] (engl. *Simple Text Oriented Messaging Protocol*) je protokol baziran na razmjeni tekstualnih poruka (MQTT podsjetimo se razmjenjuje poruke u binarnom obliku) uglavnom za WebSocket komunikaciju. Uz normalnu *publish/subscribe* funkcionalnost nudi i sigurne blokove poruka preko *begin/publish/commit* sekvenci i pozitivne i negativne potvrde. Obično se koristi za komunikaciju između internetskih preglednika.

XMPP[24] (engl. *Extensible Messaging and Presence Protocol*) protokol otvorenog tipa za *real-time* komunikaciju čije implementacije pogone razne aplikacije uključujući aplikacije za *chat*, video i glasovne pozive, kolaboraciju te razmjenu poruka između uređaja.

AMQP[25] (engl. *Advanced Message Queuing Protocol*) je najpoznatiji protokol kada se radi o protokolima na bazi reda. Implementiran je u raznim posrednicima poput najpoznatijih RabbitMQ-a i HornetQ-a, a nudi razne distribucijske strategije. Najčešće se koristi kada potrebno implementirati komunikaciju složeniju od jednostavnih *publish/subscribe* metoda. AMQP je pouzdan i bogat mogućnostima, no nije posebno brz i zahtjevan je što se tiče resursa.

WAMP[26] (engl. *Web Application Messaging Protocol*) uz *publish/subscribe* nudi i *request/response* podršku kao i kompleksnije strategije dostave. Obično se koristi u distribuiranim i mikroservisnim aplikacijama. Dosta dobar protokol, ali ne toliko široko prihvaćen.

[22] http://coap.technology/, studeni 2018.
[23] https://stomp.github.io/, studeni 2018.
[24] https://xmpp.org/about/technology-overview.html, studeni 2018.
[25] https://www.amqp.org/, studeni 2018.
[26] https://wamp-proto.org/, studeni 2018.

4. Praktični rad

Za primjer upotrebe IoT tehnologija i MQTT protokola odlučio sam automatizirati jedan dio procesa unutar tvornice za izradu SMT (engl. *Surface-mount technology*) elektroničkih komponenti. Nakon što je SMT komponenta proizvedena potrebno je ispitati njenu kvalitetu radi kategoriziranja kod prodaje, a i daljnjeg usavršavanja procesa. Ispitivanje se vrši u nekoliko koraka i na nekoliko načina:

- optički – izgled komponente, metalizacije i slojeva komponente
- elektronički – ispitivanje elektroničkih svojstava komponente
- mehanički – ispitivanje izdržljivosti komponente na rubnim temperaturama, vlazi i vibracijama zadanim prema specifikaciji

Dio procesa koji ću unaprijediti je mjerenje elektroničkih svojstva SMT komponente.

Kod elektroničkog mjerenja najvažnije su dvije stvari, ispravni i kalibrirani instrumenti za mjerenje te stabilna temperatura mjerenja. Inače održavanje stabilne temperature mjerenja unutar stroja nije teško izvesti, sonda za mjerenje temperature unutar stroja je kablom direktno spojena na klimu, a crijeva od klime idu nazad na stroj. No problem se javlja kada imamo više takvih strojeva, a za svakoga želimo vidjeti radnu temperaturu i to na udaljenom mjestu kako bi procesni inženjer mogao vidjeti korisnost održavanja stabilne temperature. Problem u ovom pristupu je što moramo imati silan broj računala koja možda nisu skupa, ali veći problem je njihovo održavanje. Za rješavanje ovog problema predlažem upotrebu IoT tehnologija.

Kao IoT senzor na koji ću spojiti temperaturnu sondu upotrijebit ću ESP8266 modul koji će ujedno imati i ulogu MQTT klijenta. IoT pokretač i drugi MQTT klijent bit će također ESP8266 na koji ću spojiti klimu (u ovomu slučaju ventilator) kako bismo je mogli kontrolirati. MQTT posrednik će biti Raspberry Pi, drugi posrednik će biti CloudMQTT za globalni pristup podacima. Podatke iz usluge u oblaku će koristiti Android aplikacija koja će kao treći MQTT klijent moći promatrati temperaturu, ali i kontrolirati klimu. U narednim poglavljima ću dodatno opisati i analizirati, te izraditi svaki dio ovog IoT sustava.

4.1. Analiza online MQTT posrednika

Online posrednici služe za razmjenu MQTT poruka bez potrebe za lokalnim posrednikom. Opcije koje ovakvi posrednici nude su skromne i uključuju podešavanje korisnika, prava, tema i postavljanje mostova (engl. *bridge*) između više posrednika. Razlikuju se u cijeni korištenja, a besplatni servisi obično nude određen broj klijenata i/ili broj poruka mjesečno. Uz komercijalne posrednike postoje i javno dostupni koji se mogu upotrebljavati bez korisničkog računa, ali nije zagarantirana stabilnost sustava i privatnost poruka (sve poruke su javno dostupne i vidljive). Javni posrednici služe samo za testiranje rada sustava. Tri najpoznatija javna posrednika su Mosquitto, Eclipse (također Mosquitto implementacija) i HiveMQ, podaci za spajanje su u tablici ispod (Tablica 4.1) [28].

Tablica 4.1 Podaci MQTT javnih posrednika

Posrednik	**Lokacija**	**Portovi**	**Websocket**
Mosquitto	test.mosquitto.org	1883 / 8883 / 8884	8080 / 8081
Eclipse	iot.eclipse.org	1883 / 8883	ws://iot.eclipse.org:80/ws wss://iot.eclipse.org:443/ws
HiveMQ	broker.hivemq.com	1883	8000

Komercijalni posrednici su većinom sastavni dio kompletne IoT usluge, no tu su i *online* posrednici koji se samo time bave poput 84codes, Bevywise, dc-square i Adafruit. Većina posrednika koje ću navesti koriste jednu od implementacija MQTT protokola instaliranu u *online* okruženje. Okruženje u oblaku ne mora nužno biti u vlasništvu kompanije posrednika već se on može nalaziti u velikim servisima poput AWS-a, Google Cloud-a i Microsoft Azure-a.

84codes **CloudMQTT**[27] je isključivo MQTT posrednik i pri tome najjednostavniji za korištenje. Servis se sastoji od Mosquitto MQTT posrednika koji su pokrenuti na Amazon AWS servisima. Uz osnove MQTT posrednika u ponudi je i kreiranje mosta za razmjenu podataka između više posrednika.

[27] https://www.cloudmqtt.com/, listopad 2018.

Adafruit IO[28] je platforma u oblaku primarno namijenjena za hobi projekte. Njihov klijent podržava Android, micro:bit, Raspberry Pi i Adafruit uređaje. Uz osnovno prikupljanje i pospremanje podataka nude i vizualizaciju preko grafova, tablica i mjernih satova, te je moguće postavljati okidače na promjene u podacima.

Bevywise **MQTTRoute**[29] posrednik je dio Bevywise IoT platforme. IoT platforma se pokreće u jednom od servisa u oblaku, imaju podršku za Google Cloud, AWS i Azure. Kao i Adafruit nude grafički prikaz podataka te kreiranje okidača i obavijesti. Dok posrednik nudi korisnu opciju trajnog pospremanja podataka u MySQL, SQLite i MongoDB baze podataka.

Solace **PubSub+**[30] Cloud je *online* posrednik instaliran kao servis u jednom od već spomenutih *Cloud* servisa (AWS, Azure, Google). Nude mogućnost spajanja usluge sa lokalnim, *cloudom* ili drugim servisima, te infrastrukturu u oblaku sa vlastitim resursima te visoku dostupnost servisa.

Realnu usporedbu *online* posrednika je nemoguće napraviti zbog nedostupnosti cjenika pojedinih usluga (vidljivo u tablicama Cloud_MQTT_posrednici.xlsx u prilogu). No za osobnu uslugu ili za testiranje prototipova najprije bih odabrao CloudMQTT ili Adafruit IO servis zbog niske cijene i jednostavnosti. Za komercijalne i primjene koje generiraju veliku količinu poruka koje želimo obraditi te iz njih izvući zaključke odabrao bih jedan od velikih servisa u oblaku iz idućeg poglavlja.

4.2. Analiza IoT Cloud usluga

IoT usluge i servise u oblaku i servise koje ću u ovome poglavlju obraditi su Amazon AWS IoT, Google Cloud IoT, Microsoft Azure IoT, IBM Watson IoT Platform, Bosch IoT Platform i Oracle Cloud IoT. Svi ti servisi postoje i bez IoT sufiksa, a IoT označava da su servisi konfigurirani za primanje velikog broja podataka te njihovu obradu. Većina ovih servisa ne podržava MQTT protokol izravno već REST preko HTTP-a, MQTT je potrebno dodati kao posebnu dogradnju. Glavna razlika prema posrednicima iz prošlog poglavlja je što usluge u oblaku nude procesiranje podataka te primjenu prediktivne analitike nad podacima [29]. Cijene svakog spomenutog servisa i usluga nalaze se na označenim poveznicama kao i u tablicama (Cijene_Cloud_usluga.xlsx) u prilogu.

[28] https://io.adafruit.com/, listopad 2018.
[29] https://www.bevywise.com/mqtt-broker/, listopad 2018.
[30] https://cloud.solace.com/, listopad 2018.

Važan koncept koji se stvara iz *meta* podataka pojedinog izvora (uređaja) je digitalna kopija (engl. *Device clone* ili *Device shadow*). Digitalna kopija služi za čuvanje informacija o uređaju i lakše kontrole čitavog sustava koji se obično sastoji od stotina takvih uređaja.

Bosch IoT Platform odnosno Suite je skupina od nekoliko IoT servisa instaliranih u Bosch IoT Cloud ili Amazon AWS. Servisi u ponudi su IoT Hub, Gateway Software, Things, Remote Manager, Analytics, Insights, Permissions i Rollouts. Svaki od ovih servisa ima zasebne pakete i cijene. [30]

IoT Hub[31] je posrednik u oblaku, a IoT Gateway Software[32] je aplikacija posrednika koja se instalira na lokalni server. Bosch ne daje puno informacija o ovoj usluzi bez direktnog kontakta. IoT Things[33] služi za upravljanje digitalnim kopijama. IoT Remote Manager[34] kontrolira IoT uređaje. IoT Analytics[35] servis se bavi detekcijom anomalija u podacima za validaciju očitanja senzora te je koristan u prediktivnom održavanju. IoT Insights[36] ima ulogu prikupljanja i analize podataka, vizualizacije i pospremanja podataka. Pospremljeni podaci se mogu pretraživati upitima (NoSQL/MongoDB) te se mogu pomoću JSON ili CSV formata prebaciti u analitički alat poput Matlab-a, Excel-a i Tableu-a.

Oracle IoT Cloud uz spremanje i obradu poruka nudi i kreiranje prediktivne digitalne kopije, strojno učenje iz podataka, REST API za komunikaciju sa raznim uređajima i aplikacijama. Svi servisi su zajedno u ponudi, a za kompletnu uslugu u oblaku Oracle ne naplaćuje mjesečnu pretplatu već se odlučio za naplatu po korištenju. Jedinica za naplatu je OCPU (engl. *Oracle Compute Unit*) koja je definirana kao procesorski kapacitet jednak jednoj fizičkoj jezgri Intel Xeon procesora, svaka OCPU jedinica odgovara izvršavanju dviju niti na takvom procesoru. Također, koriste *Universal Credit* kao kupone koji se mogu kupiti na mjesečnoj razini i prema tome koristiti bilo koji servis dok se ne potroše kupljeni kuponi [31].

Oracle unutar IoT Cloud servisa nudi i aplikacije koje koriste prikupljene podatke te ih analiziraju uz pomoć prediktivne analitike kako bi unaprijedili procese. Aplikacije su

[31] https://www.bosch-iot-suite.com/hub/, prosinac 2018.
[32] https://www.bosch-si.com/iot-platform/iot-platform/gateway/software.html, prosinac 2018.
[33] https://www.bosch-iot-suite.com/things/, prosinac 2018.
[34] https://www.bosch-iot-suite.com/remote-manager/, prosinac 2018.
[35] https://www.bosch-iot-suite.com/analytics/, prosinac 2018.
[36] https://www.bosch-iot-suite.com/insights/, prosinac 2018.

namijenjene kao gotova usluga za pojedine grane poslovanja i proizvodnje. IoT Asset Monitoring Cloud koristi *real-time* podatke za nadgledanje kako bi poboljšali prediktivno održavanje, te za stvaranje digitalnih kopija. IoT Production Monitoring Cloud služi za nadzor tvornice, procesa, proizvoda i strojeva, te za stvaranje digitalne kopije cijelih tvornica i pojedinih procesa preko modela strojnog učenja. IoT Fleet Monitoring Cloud je kompletan nadzor svih vozila te njihovo održavanje i bolje predviđanje vremena dostave. IoT Connected Worker Cloud služi za nadzor ljudi odnosno njihovu sigurnost na radu preko nosivih i stacionarnih senzora [32].

IBM IoT Platform je dio Watson servisa, a služi za prikupljanje IoT podataka njihovu analizu, okidače i daljnju integraciju. IoT uređaji, *gatewayi*, i ostali izvori se spajaju na platformu preko MQTT ili HTTP protokola. Glavna prednost platforme je mogućnost korištenja prikupljenih i obrađenih podataka u vlastitim aplikacijama koje se pokreću na IBM Watson Cloud-u, ili nekom drugom oblaku ili servisu preko REST API-a. Cjenovna ponuda platforme se razlikuje prema modelu naplate koji mogu biti prema razmijenjenim podacima u megabajtima te prema analiziranim podacima u megabajtima (naplata prema korištenju) [33].

Microsoft Azure IoT sastoji se od dva osnovna servisa IoT Hub i IoT Central. Azure IoT Suite ima kvalitetnu integraciju sa *third-party* servisima poput SAP, Salesforce, Oracle, WebSphere i ostalih [34].

IoT Hub[37] se u osnovnoj inačici ponaša kao *online* spremanje podataka, dok u standardnoj verziji uključuje komunikaciju u oba smjera, digitalne kopije i IoT Edge. IoT Edge je naziv za Microsoft API za uređaje i *gatewaye* na bazi Microsoft i Linux operacijskih sustava, a može se kupiti kao zaseban servis no to je u osnovi IoT Hub standardni paket.

IoT Central[38] se nudi kao potpuno SaaS rješenje koje pojednostavljuje inicijalne postavljanje IoT uređaja i aplikacija bez znanja o oblaku. Central uključuje IoT Hub servis i automatsku skalabilnost sustava a korisnik nema potrebe znati da postoje oba servisa. Servis se naplaćuje po priključenom uređaju.

[37] https://azure.microsoft.com/en-us/services/iot-hub/, prosinac 2018.
[38] https://azure.microsoft.com/en-us/services/iot-central/, prosinac 2018.

Google Cloud IoT je kompletan set alata za spajanje, procesiranje, spremanje i analizu podataka koji dolaze od spojenih uređaja. Integracija sa ostalim Google servisima je dostupna, no to je ujedno i nedostatak jer nema servisa ostalih proizvođača, a i podrška za programske jezike je dosta limitirana. Glavni servis je Cloud IoT Core[39] (*online* posrednik) koji skuplja podatke preko MQTT i HTTP protokola te ih šalje na Cloud Pub/Sub servis [35].

Cloud Pub/Sub[40] je ulazni dio analize jer prikupljene podatke prosljeđuje na daljnju obradu. Obrada podataka sadržava čitav niz servisa poput Google BigQuery za jednostavne analize i Cloud Machine Learning Engine za strojno učenje. Pub/Sub ima jednostavnu naplatu po količini prenesenih podataka, bez naplate kreiranja i održavanja tema i pretplata [36].

Cloud IoT Edge[41] je usluga (trenutno u *alpha* verziji) *real-time* analize i donošenja odluka lokalno koristeći Google softver (TensorFlow Lite) i hardver.

Amazon IoT je skupina od nekoliko servisa za spajanje, nadzor i analizu IoT uređaja i podataka u oblaku. Cijela usluga je podijeljena na dva osnovna dijela, softver za uređaje i servise u oblaku. Amazon FreeRTOS je operacijski sustav za mikrokontrolere, a AWS IoT Greengrass je softver koji omogućuje lokalno sakupljanje, sinkroniziranje podataka i mogućnosti strojnog učenja na siguran način. Servisi su AWS IoT: Core, Device Management, Device Defender, Things Graph, Analytics, SiteWise i Events. To su redom servisi za spajanje uređaja, nadzor, sigurnost, analizu, industrijsku primjenu i konfiguraciju događaja. Velik broj usluga i servisa povećava složenost te AWS IoT ima veliku uzlaznu krivulju korištenja [37].

AWS IoT Core[42] je osnovni servis koji omogućuje sigurno i lako spajanje uređaja i njihovu interakciju sa aplikacijama u oblaku i ostalim uređajima. Jedna od osnovnih značajki IoT Core servisa je Device SDK koji omogućuje uređaju ili mobilnoj aplikaciji spajanje na AWS IoT Core preko sigurne komunikacije i preko raznih protokola (MQTT, HTTP, WebSockets), a podržava C, JavaScript i Arduino programske jezike. Sljedeće značajke su Device Gateway i Message Broker koje su u suštini *publish/subscribe* posrednici, uz

[39] https://cloud.google.com/iot-core/, prosinac 2018.
[40] https://cloud.google.com/pubsub/, prosinac 2018.
[41] https://cloud.google.com/iot-edge/, prosinac 2018.
[42] https://aws.amazon.com/iot-core/, prosinac 2018.

mogućnosti autentikacije i autorizacije te enkripcije svih točaka komunikacije, registar svih spojenih uređaja, i Device Shadow koji je drugi naziv za Device Clone odnosno digitalnu kopiju uređaja. Rules Engine procjenjuje poruke pristigle na AWS IoT Core te ih transformira ili prosljeđuje na drugi uređaj ili servis u oblaku. Servis naplaćuje samo usluge koje se koriste bez mjesečne pretplate, posebno se naplaćuju Connectivity, Messaging, Device Shadow, Registry i Rules Engine.

AWS IoT Device Management[43] je servis za nadzor, pretraživanje i kontrolu uređaja. Uređaje je moguće registrirati u skupinama, a također je moguće i slati akcije uređajima poput *reseta* ili *firmware updatea*.

AWS IoT Analytics[44] skuplja, obrađuje, pohranjuje i analizira podatke skalabilno te se svaka od tih akcija naplaćuje posebno. Moguće je uvesti i vlatite analize preko spremnika (engl. *docker*) koji se izvršavaju na AWS IoT Analytics.

Treba navesti kako svaki servis ima besplatan paket usluge te nakon njegove potrošnje kreće naplata. Kao što je vidljivo iz navedenog, naplata usluga je dosta složena i razgranata ali daje veliku fleksibilnost i mogućnost odabira samo onih usluga koje su nam potrebne.

U tekstu niže usporediti ću cijene usluga za nekoliko scenarija upotrebe IoT uređaja i komunikacije sa servisima u oblaku, te korištenje obrade i analize podataka. Za usporedbu ću uzeti Microsoft, IBM, Google i Amazon usluge zbog dostupnosti cjenika i kvota za izračun potrošnje [39]. Detaljni izračuni za svaki scenarij nalaze se u tablicama (Izračun_scenarija.xlsx) u prilogu.

Scenarij 1 (srednji broj uređaja / srednja veličina poruka): 1000 uređaja, veličina poruke 8 KB, 2 poruke u minuti. Dnevno 2.880.000 poruka, mjesečno 86.400.000 poruka i 675.000 MB podataka.

Cijena mjesečnih usluga po veličini: Amazon $183, Microsoft $250, IBM $844, Google $1350.

Scenarij 2 (mali broj uređaja / velika veličina poruka): 200 uređaja, veličina poruke 50 KB, 2 poruke u minuti. Dnevno 576.000 poruka, mjesečno 17.280.000 poruka i 843.750 MB podataka.

[43] https://aws.amazon.com/iot-device-management/, prosinac 2018.
[44] https://aws.amazon.com/iot-analytics/, prosinac 2018.

Mjesečna cijena za ovaj scenarij po uslugama: Amazon $175, Microsoft $500, IBM $1055, Google $1688.

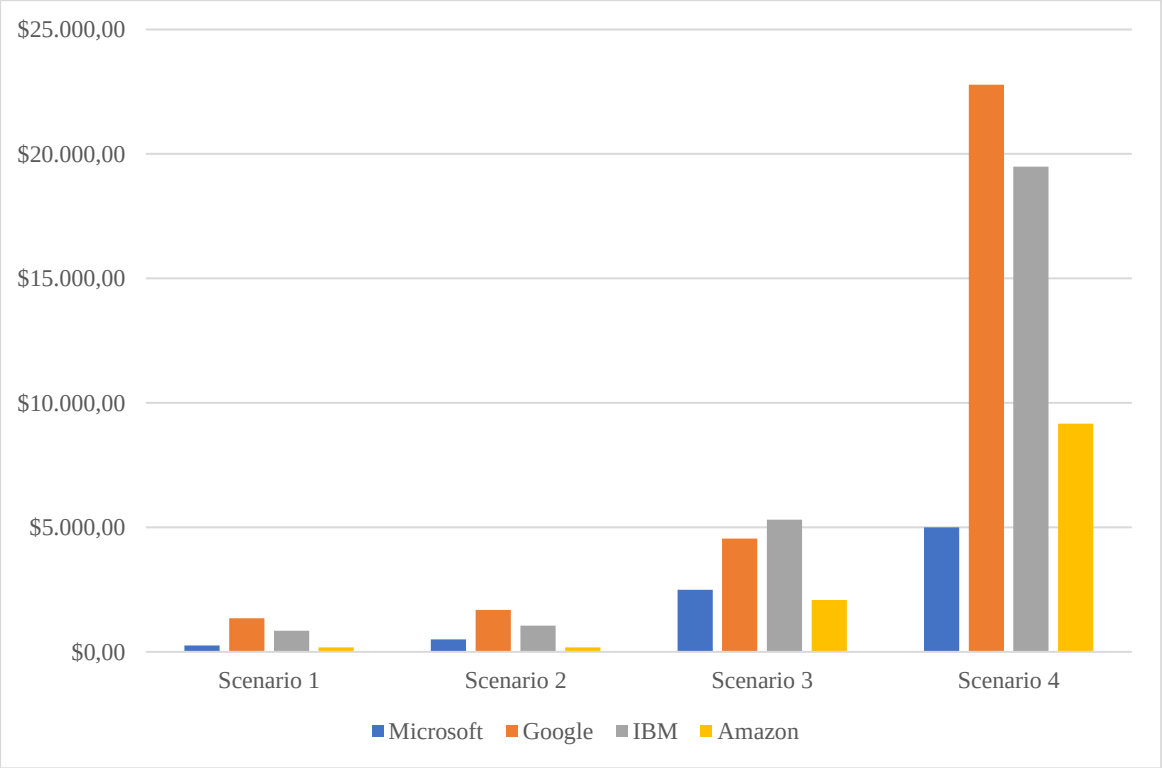
Scenarij 3 (srednji broj uređaja / mala veličina poruka / velik broj poruka u minuti): 1000 uređaja, veličina poruka 4 KB, 60 poruka u minuti. Dnevno 86.400.000 poruka, mjesečno 2.592.000.000 poruka i 10.125.000 MB podataka.

Mjesečna cijena za usluge: Amazon $2.084, Microsoft $2.500, Google $4.557, IBM $5.316.

Scenarij 4 (velik broj uređaja / mala veličina poruka / velik broj poruka u minuti): 10.000 uređaja, veličina poruka 4 KB, 30 poruka u minuti. Dnevno 432.000.000 poruka, mjesečno 12.960.000.000 poruka i 50.625.000 MB podataka.

I na kraju cijena mjesečnih usluga za scenarij 4: Microsoft $5.000, Amazon $9.171, IBM $19.491, Google $22.782.

Prema izračunima navedenih scenarija vidljivo je da u normalnim situacijama Amazon AWS daleko najisplativiji dok ga prati Microsoft Azure, a Google Cloud je u svim opcijama najskuplji. Jedino u ekstremnoj situaciji je Microsoft najjeftiniji zbog fiksnog mjesečnog plana. U obzir valja uzeti i detalje svake usluge u koje nismo ulazili poput broja uređaja koji se mogu spojiti u minuti, pa do brzine promjene indeksa i podataka o digitalnoj kopiji [38]. Graf ispod (Graf 4.1) pokazuje vizualno usporedbu izračuna po scenarijima.



Graf 4.1 Usporedba izračuna pojedinih servisa po scenarijima

U većini slučajeva odabrao bih Amazon AWS IoT zbog niske cijene i velikog izbora popratnih usluga i mogućnosti kombinacije, no da već posjedujem Microsoft Azure korisnički račun vjerojatno bih odabrao njihovu IoT Cloud uslugu.

4.3. Hardver

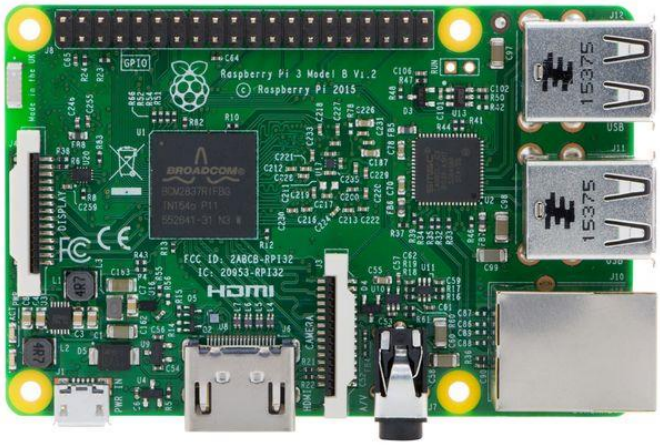
Za samu izradu i testiranje praktičnog dijela rada koristit ću pojednostavljenu verziju prije opisanog sustava. Za radno mjesto neću upotrijebiti pravi industrijski stroj već provizorno mjesto za testiranje sustava, a umjesto klime upotrijebiti ćemo ventilator.

Uz već nabrojane ESP8266 i Raspberry Pi, od hardvera upotrijebiti ću DHT22[45] digitalnu temperaturnu sondu, 5V ventilator, te nešto popratnih elektroničkih komponenti. DHT22 nije posebno precizna sonda no poslužiti će za svrhu demonstracije. Najzanimljiviji dijelovi hardvera su ESP8266 i Raspberry Pi koji su dosta poznati kod hobi projekata i izrade prototipova te će dalje biti opisani u sljedećim poglavljima.

[45]https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf, listopad 2018.

4.3.1. Raspberry Pi

Prvi Raspberry Pi (Slika 4.1) je kreiran 2012. godine od strane Raspberry Pi Foundation[46] obrazovno-dobrotvorne tvrtke. Raspberry Pi je mini računalo na koje možemo spojiti monitor ili TV, standardni miš i tipkovnicu, zvučnike i mikrofon, te raznu USB periferiju. Omogućuje nam pretraživanje Interneta, igranje, gledanje videa, crtanje, pisanje i programiranje. Nama najvažnija osobina je mogućnost interakcije sa vanjskim svijetom preko raznih senzora, motora i kamera koji se direktno ili indirektno spajaju na razne Raspberry Pi ulaze.



Slika 4.1 Raspberry Pi mini računalo[47]

Tijekom godina izašlo je nekoliko modela Raspberry Pi-a: model A sa 1 i 1+ verzijom, model B sa verzijama 1, 1+, 2, i posljednja 3, tu su još Compute Module i Zero verzija.

Hardverski Raspberry Pi koristi SoC Broadcom2836 koji ima quad-core ARM Cortex A7 na 900MHz procesor sa 1GB RAM-a, GPU (VideoCore IV), GPIO, I^2C i SPI komunikaciju, PWM te UART. Kontrola USB-a i mreže vrši se preko jednog LAN9514 kontrolora. Memorija računala je na Elpida EDB8132B4PB-8D-F čipu. Od izlaza imamo 4 USB porta, jedan 10/100 Mbit/s mrežni konektor, analogni 3.5 mm audio/video konektor, HDMI (1.3 & 1.4 verzije) konektor [40][41].

Za napajanje Raspberry Pi model B koristi +5.0V DC i preporučeno 2.5A izvor napajanja. Model B, obično, vuče oko 700 do 1000mA bez dodatnih uređaja. Korištenjem raznih sučelja potrošnja se povećava, te je vrlo važno imati odgovarajući izvor napajanja ali i kabel za napajanje [40].

[46] https://www.raspberrypi.org/, listopad 2018.
[47] Preuzeto sa https://www.chipoteka.hr/artikl/127339/raspberry-pi-3-model-b-ugradeni-wi-fi-i-bluetooth-8208000130, listopad 2018.

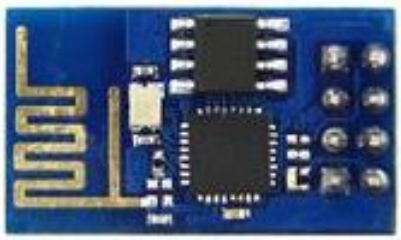
Operacijski sustav Raspberry Pi nalazi se na MicroSDHC kartici i brzina sustava dosta ovisi o kvaliteti iste. Većina operacijskih sustava koji se mogu pokrenuti bazirano je na Linux-u i njegovim distribucijama poput Raspbian-a (baziran na Debian-u), Ubuntu Mate, Snappy Ubuntu Core, Pidora, ali se mogu pokrenuti i neki drugi poput Windows 10 IOT Core i RISC OS.

Programiranje na većini operacijskih sustava je moguće, a podržana je većina programskih jezika. Neki od njih su C++, Python, Perl, Scratch i Ruby, a uz dodatke moguće je i programiranje u Javi, C#, Pascal i PHP-u. Ova široka raznolikost omogućuje kako početniku tako i profesionalcu izradu programa koji će se pokretati na Raspberry Pi-u.

Jedna od glavnih osobina Raspberry Pi-a (posebno za hobiste) je GPIO konektor koji omogućuje komunikaciju sa uređajima preko SPI, I^2C i UART protokola. Ostali pinovi se koriste kao programirani digitalni ulazi/izlazi. Nema analognih ulaza ni izlaza, osim preko specijalnog modula. GPIO konektor ima logički nivo od 3.3V, nije 5V tolerantan i nema zaštitu od prevelikog napona. Zbog toga je potrebna velika pažnja prilikom spajanja ostalih uređaja i korištenje pretvarača logičkog nivoa kako ne bi došlo do oštećenja ulaza/izlaza, a moguće i kompletnog računala.

4.3.2. ESP8266

ESP8266 (Slika 4.2) proizveo je pravi bum u hobi svijetu posebno vezano u IoT kada ga je 2014. godine proizvela kineska kompanija Espressif[48]. Od te godine izašlo je nekoliko verzija čipa i nekoliko klonova raznih proizvođača.



Slika 4.2 ESP8266-01 modul[49]

ESP8266 je SoC koji sadrži Tensilica L106 80MHz 32-bit mikrokontroler i Wi-Fi primopredajnik 802.11 b/g/n. Također, modul sadržava i 16 GPIO pinova, analogno-digitalni

[48] https://www.espressif.com/en/products/hardware/esp8266ex/overview, listopad 2018.
[49] Preuzeto sa https://nurdspace.nl/ESP8266#Translated_datasheet, listopad 2018.

pretvarač za analogni ulaz, PWM modulaciju, SPI, I^2C i UART komunikacijska sučelja. Naravno glavna prednost ovog čipa je puna Wi-Fi podrška uključujući potpuni TCP/IP sa DNS podrškom [42].

Točnije, Espressif je proizvođač SoC čipa, a postoje razne serije i proizvođači modula kojima je baza Espressif-ov čip. Najpoznatiji proizvođač takvih modula je Ai-Thinker[50] koji je proizveo cijelu seriju ESP modula. Poznati su još i SparkFun[51] i Adafruit[52] moduli, no daleko najpoznatiji je NodeMCU[53] modul kojeg je zbog *on-board* USB porta vrlo lako programirati. Kod modula nepoznatih proizvođača glavni je nedostatak što Wi-Fi modul nije legalno odobren od certifikacijske kuće pa ga je problem koristiti u komercijalne svrhe. Moduli se većinom razlikuju u veličini vanjske *flash* memorije i broju dostupnih GPIO pinova, ali i o *on-board* regulaciji napajanja i sučelju za programiranje.

ESP8266 modul se napaja isključivo 3.3V napajanjem, a potrebno je paziti na kvalitetu napajanja jer prilikom odašiljanja modul može povući i do 215mA struje. Modul ima i nekoliko načina rada (*shutdown, saving mode, deep sleep, standby*) pa je moguće korištenjem jednog od njih uštediti dosta energije [42].

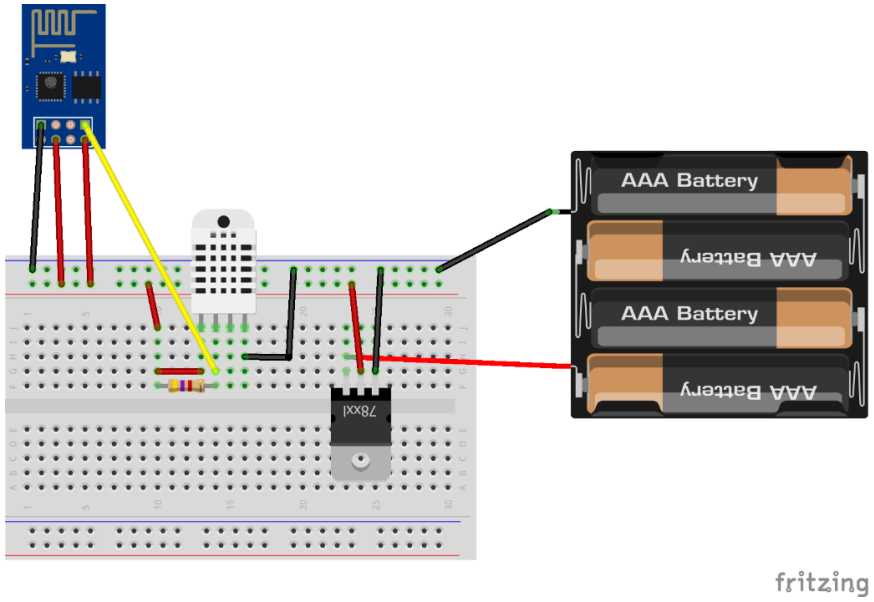
Postoji nekoliko SDK alata koji olakšavaju razvoj i programiranje ESP8266 modula. Službeni Espressif-ov je baziran na FreeRTOS-u, a postoje još ESP-Open-SDK baziran na GCC alatima. Tu su još razni SDK i sučelja na bazi Lua, Python, JavaScript, Forth, Lisp, Basic i C/C++ programskim jezicima. Te *firmware* na bazi i za NodeMCU, Arduino, PlatformIO, MicroPython, ESP8266 BASIC, Espruino, Mongoose OS te ostali.

4.3.3. Izrada tiskanih pločica

Za izradu senzorskog i pokretačkog modula odlučio sam se upotrijebiti bušenu pločicu (engl. *protoboard*) jer sam želio biti siguran u spojeve, makar se danas za izradu prototipova većinom koristi *breadboard* radi lakšeg mijenjanja sheme spajanja. Popis materijala za oba modula vrlo je kratak i dan je niže zajedno sa shemama pojedinog modula.

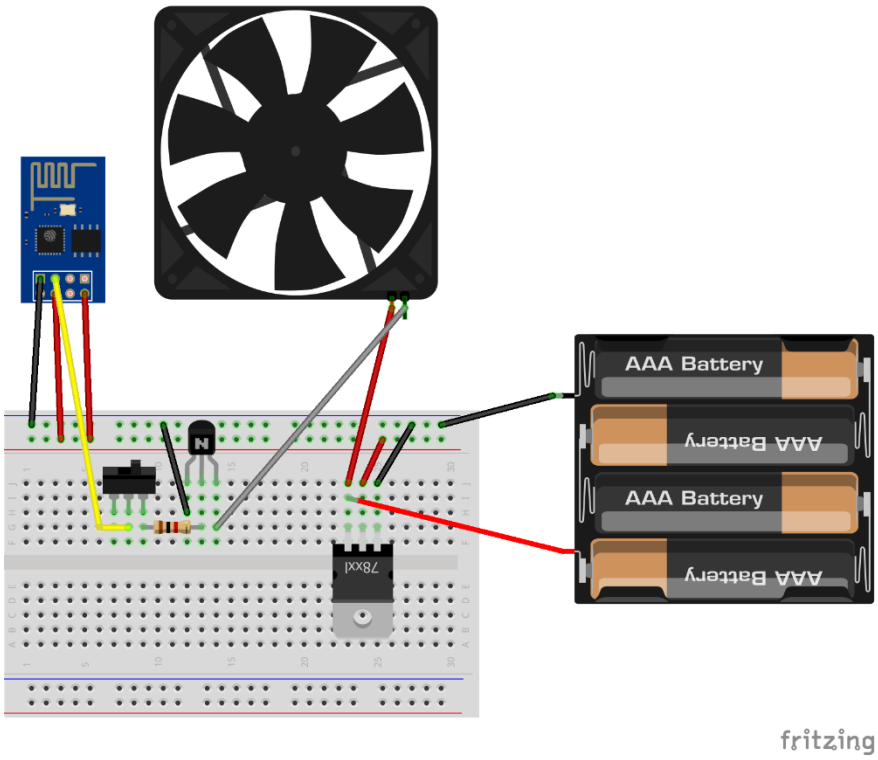
Senzorski modul (Slika 4.3) se sastoji od digitalnog temperaturnog senzora DHT22, ESP8266 modula, i 4.7kΩ otpornika za *pull-up* senzora, te regulatora napajanja sa 5V na 3.3V 1083-33.

[50] https://www.ai-thinker.com/product/esp8266/, listopad 2018.
[51] https://www.sparkfun.com/, listopad 2018.
[52] https://www.adafruit.com/, listopad 2018.
[53] http://www.nodemcu.com/index_en.html, listopad 2018.



Slika 4.3 Shema ESP8266 senzora

Pokretač (Slika 4.4) se sastoji od ESP8266 modula na koji je spojen 5V ventilator preko tranzistora BC547 NPN i otpornika od 1kΩ i prekidača na bazi tranzistora, te regulatora napajanja LM111733. Prekidač služi samo kod prvog pokretanja ESP8266 modula zato jer su I/O izlazi uključeni, pa prekidačem prekidamo dovod napona na tranzistor te time omogućujemo normalno pokretanje modula.



Slika 4.4 Shema ESP8266 pokretača

4.4. Programsko rješenje

Programska rješenja praktičnog rada sastoje se od dva C++ programa za klijente senzora i prekidača, i Android aplikacije za mobilni klijent. Za Raspberry Pi posrednik koristio sam gotovo rješenje u obliku Mosquitto[54] aplikacije, a kao *Cloud* posrednik koristio sam CloudMQTT rješenje.

4.4.1. C++ program za kontrolu i nadzor

Programi su napisani u C++ programskom jeziku koristeći Arduino IDE[55] programsko sučelje. Prije programiranja na oba ESP8266 modula instalirao sam Arduino Core[56] *firmware* pomoću programatora koji sam sam izradio. Za korištenje Arduino platforme odlučio sam se zbog prijašnjeg iskustva sa Arduino proizvodima, te mogućnosti korištenja Arduino biblioteka. Izvorni kôdovi oba programa nalaze se u prilogu (Senzor.ino i Kontrola.ino).

Program za senzor koristi četiri vanjske biblioteke:

- DHTesp[57] – biblioteka za komunikaciju sa DHT senzorom prilagođena za ESP8266
- ESP8266WiFi[58] – upravljanje Wi-Fi konekcijom
- PubSubClient[59] – *Publish/Subscribe* klijent za Arduino
- BlynkSimpleEsp8266[60] – Blynk je naziv za Android aplikaciju koja spaja razne IoT uređaje te vizualizira njihove podatke. Za spajanje uređaja koriste svoje servere i aplikacije. Ovu biblioteku koristim jer pruža mogućnost kreiranja rasporeda za vremensko izvršavanje zadataka

Slanje poruke odvija su u sva osnovna koraka:

1. Podešavanje PubSubClient-a:
    a. Potrebno je postaviti Wi-Fi klijent koji smo kreirali pomoću ESP8266WiFi biblioteke: `pubSubClient.setClient(wifiClient);`

[54] https://mosquitto.org/, listopad 2018.
[55] https://www.arduino.cc/en/main/software, listopad 2018.
[56] https://github.com/esp8266/Arduino, listopad 2018.
[57] https://github.com/dzindra/dhtesp, listopad 2018.
[58] https://github.com/ekstrand/ESP8266wifi, listopad 2018.
[59] https://github.com/knolleary/pubsubclient, listopad 2018.
[60] https://github.com/blynkkk/blynk-library, listopad 2018.

   b. Zatim postaviti IP adresu servera i port, u našem slučaju je to IP adresa Raspberry Pi posrednika, a port je 1883 standardan za MQTT komunikaciju:
```
IPAddress mqtt_server(192, 168, 43, 11);
pubSubClient.setServer(mqtt_server, 1883);
```
   c. Te uspostaviti konekciju sa serverom i predstaviti se s unikatnim imenom:
```
pubSubClient.connect(ID);
```
2. Izvršavanje u petlji pomoću BlynkSimpleEsp8266 biblioteke :
   a. Čitanje vrijednosti temperature sa DHT senzora pomoću DHTesp biblioteke
   b. Slanje poruke sa temperature na temu o temperaturi:
```
pubSubClient.publish(TOPIC, temperature);
```

Nedostatak programa je nemogućnost ponovnog uspostavljanja Wi-Fi konekcije kada ona jednom pukne ili odlazak u *Sleep* mod u intervalu kada se temperatura ne čita radi uštede energije.

Program za kontrolu također koristi četiri biblioteke no umjesto DHTesp biblioteke za DHT senzor koristi ArduinoJson[61] biblioteku koja služi za razmjenu i korištenje složenijih informacija. Pokretač osim što prima temperaturu senzora i prema njoj djeluje može i primati kontrolne informacije iz Android aplikacije za ručno paljenje/gašenje klime (ventilatora).

Ovaj program je nešto složeniji i ima korak više:

1. Podešavanje – ovaj korak uz ista podešavanja kao i kod senzora potrebno je dodati još:
   a. Pretplaćivanje na teme sa kojih želimo dobivati informacije:
```
pubSubClient.subscribe(SENSOR_TOPIC);
pubSubClient.subscribe(SETTINGS_TOPIC);
```
   b. Postavljanje naziva metode koja će se izvršavati kada klijet primi novi poruku:
```
pubSubClient.setCallback(callback);
```
2. Implementacija *callback* metode – ova metoda pretvara podatke pomoću ArduinoJson biblioteke s obzirom sa koje teme je poruka stigla:
   a. Ako je poruka stigla sa SENSOR teme pretvori pristiglu temperaturu iz teksta u decimalni broj
   b. A ako je poruka pristigla sa SETTINGS teme pretvori JSON podatke u postavke
3. Izvršavanje u petlji:

[61] https://github.com/bblanchon/ArduinoJson, listopad 2018.

a. Pomoću pristigle temperature i postavki te zadane maksimalne dopuštene temperature odredi da li klimu treba upaliti, ugasiti ili ostaviti staro stanje
b. Pošalji poruku sa informacijom u kojem je stanju klima:
`pubSubClient.publish(CONTROL_TOPIC, on);`

Program za kontrolu pati od istih nedostatka kao i program senzora. Oba još imaju problem što su npr. informacije o Wi-Fi postavkama ili MQTT temama upisane direktno u programu, a mogu se mijenjati.

4.4.2. Konfiguracija MQTT posrednika

Prvo ću konfigurirati CloudMQTT posrednik te nakon toga lokalni Mosquitto Raspberry Pi posrednik. Također, na lokalnom posredniku podesit ću most tako da sve teme i poruke putuju između lokalnog i *online* posrednika.

Podešavanje CloudMQTT posrednika:

1. Prvo je potrebno obaviti registraciju i otvaranje novog računa na CloudMQTT stranicama[27]
2. Kreirati ću novu instancu koristeći *„Cute Cat“* besplatan plan, te odabrati najbliži podatkovni centar
3. Odabirom nove kreiranje instance otvara se stranica sa važnim detaljima (*server*, *user*, *password*, *port*) koji će trebati kasnije
4. Dodati ću novog korisnika, to je zapravo identifikacijski ključ klijenta ili drugog posrednika, pod karticom *„Users & ACL“*
5. Na istoj kartici podesiti ću i nivo kontrole (ACL) za korisnika. # pod *„Topic“* znači za sve teme
6. Pod karticom *„Websocket UI“* nalazi se koristan alat za testiranje sustava pomoću kojega gledamo poruke koje dolaze u sustav, ali možemo i slati poruke na određenu temu
7. Konfiguracija je gotova za posrednik u oblaku (slike glavnih kartica nalaze se u prilogu)

Podešavanje Raspberry Pi posrednika nešto je složenije:

1. Raspberry Pi je potrebno prvo pripremiti, odnosno instalirati operacijski sustav. Ja sam se odlučio za Raspbian Lite[62] (službeni OS, verzija bez *desktopa* jer je Raspberry Pi ionako samo posrednik)
2. Nakon instalacije OS-a potrebno je spojiti se na Wi-Fi
    a. Pokretanje naredbe `sudo raspi-config` te nakon toga u izborniku *„Network options“* odabiremo opciju *„Wi-Fi“*
    b. U datoteci wpa-supplicant moramo unesti postavke Wi-Fi mreže
    ```
    sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
    ```
    c. Postavke imaju oblik
    ```
    network={
        ssid="ime_mreže"
        psk="lozinka_mreže"
    }
    ```
    d. Nakon spremanja datoteke Ctrl+X te Y, i Enter pokrećem rekonfiguraciju mrežnog adaptera naredbom
    ```
    wpa_cli -i wlan0 reconfigure
    ```
3. Zatim ću ažurirati operacijski sustav:
    ```
    sudo apt-get update
    sudo apt-get upgrade
    ```
4. Zbog veće sigurnosti i mogućnosti rada sustava bez dodatnog uređaja (Wi-Fi *routera*) bilo bi korisno da Raspberry Pi konfiguriramo kao pristupnu točku (skraćeno AP). Tada bi se klijenti spajali direktno na Raspberry Pi, no zbog složenosti konfiguracije AP točke to neću učiniti, te će se klijenti kao i Raspberry Pi spajati na mobilnu pristupnu točku.
5. Kao što sam već spomenuo kao MQTT lokalni posrednik koristiti ću Eclipse Mosquitto implementaciju. Za instalaciju ću pokrenuti sljedeće naredbe:
    ```
    sudo apt-get install mosquitto -y
    sudo apt-get install mosquitto-clients -y
    ```
6. Te podesiti Mosquitto konfiguraciju u datoteci:
    ```
    sudo nano /etc/mosquitto/mosquitto.conf
    ```
7. Konfiguracija:
    ```
    allow_anonymous false
    password_file /etc/mosquitto/pwfile
    ```

[62] https://www.raspberrypi.org/downloads/raspbian/, listopad 2018.

```
listener 1883
```

8. U konfiguraciju dodajem i postavke za most sa lokalnog na server u oblaku i obratno, postavke su vidljive niže (Kôd 4.1):

```
connection cloudmqtt
address m20.cloudmqtt.com:26669
topic # both
start_type automatic
remote_username „korisničko ime CloudMQTT računa“
remote_password „lozinka ime CloudMQTT računa“
remote_clientid raspberrypi
local_clientid raspberrypi
keepalive_interval 300
cleansession true
bridge_protocol_version mqttv311
bridge_cafile /etc/ssl/certs/ca-certificates.crt
bridge_insecure false
```

Kôd 4.1 Postavke mosta sa lokalnog na server u oblaku

9. Konfiguraciju je potrebno spremiti sa Crtl+X, nakon toga Y i Enter
10. Pošto sam u konfiguraciji odredio da klijenti za spajanje na MQTT posrednik moraju imati korisničko ime i lozinku moram iste konfigurirati:
    `sudo mosquitto_passwd -c /etc/mosquitto/pwfile raspberrypi`
    Time će se kreirati datoteka sa lozinkom, koju moramo dva puta upisati, za korisničko ime „raspberrypi“
11. Za kraj ću resetirati Raspberry Pi nakon čega se MQTT server sam pokrenuti:
    `sudo reboot`

4.4.3. Android aplikacija

Android aplikacija se sastoji od jedne glavne aktivnosti i MQTT pozadinskog servisa. Koristi se nekoliko glavnih vanjskih biblioteka:

- Google GSON[63] – mapiranje objekata u JSON, a služi za slanje kontrolnih postavki pokretaču
- MPAndroidChart[64] – za lijepe grafikone o temperaturi i stanju klime, koristim besplatni dio biblioteke

[63] https://github.com/google/gson, listopad 2018.
[64] https://github.com/PhilJay/MPAndroidChart, listopad 2018.

- Eclipse Paho[65] – Eclipse implementacija MQTT klijenta za većinu programskih jezika, uz slanje ujedno sadrži i pozadinski servis za pretplatu i primanje poruka

Od postavki aplikacije (AndroidManifest.xml) najvažnije su:

a. Prava aplikacije – naša aplikacija zahtijeva pristup internetu (Kôd 4.2):

```
<uses-permission android:name="android.permission.WAKE_LOCK"
/>
<uses-permission android:name="android.permission.INTERNET"
/>
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission
android:name="android.permission.READ_PHONE_STATE" />
```

Kôd 4.2 Prava mobilne aplikacije

b. Registracija MQTT servisa za prijem i slanje poruka:

```
<service
android:name="org.eclipse.paho.android.service.MqttService" />
```

Prikaz glavne aktivnosti sastoji se od nekoliko dijelova:

- Prikaz trenutne temperature koju mjeri senzor
- Prikaz stanja klime – ugašena/upaljena, ručni/automatski način rada
- Grafički prikaz temperature i stanja klime
- Prekidači za podešavanje moda i stanja klime

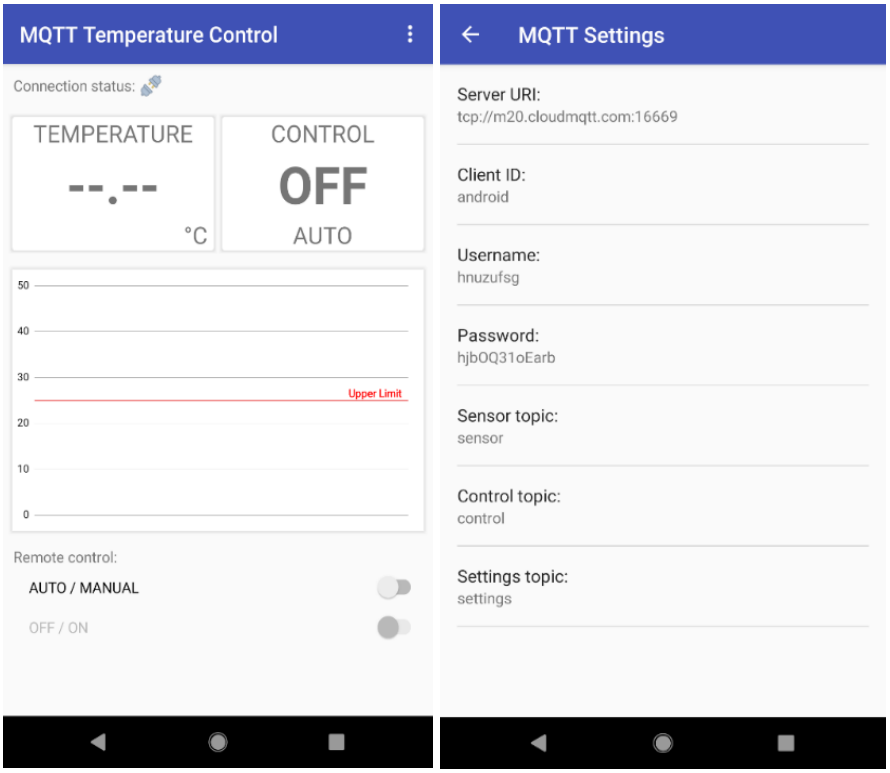
Za rad aplikacije potrebno je implementirati nekoliko metoda iz Paho MQTT servisa. Točnije potrebno je odrediti što će servis učiniti kada se pokušamo spojiti/isključiti sa CloudMQTT servisa, pretplatiti/isključiti pretplatu na neku temu, što napraviti kada želimo poslati ili stigne nova poruka. Kod implementacije svake navedene metode imamo mogućnost odrediti što se desi kada se akcija (metoda) uspješno ili neuspješno izvrši. Za razliku od prijašnjih C++ programa ovdje je programski podržano ponovno spajanje na servis u oblaku kada se promijene postavke ili se iz nekog razloga odspojimo sa mreže.

MPAndroidChart biblioteka se koristi tako da je nakon početnih postavki potrebno samo dodavati nove vrijednosti te će se biblioteka brinuti o njihovom dinamičkom iscrtavanju.

[65] https://www.eclipse.org/paho/, listopad 2018.

Glavna aktivnost kreira nove objekte na prvom pokretanju aplikacije te delegira pozive između iscrtavanja grafa, MQTT servisa i korisnikovih unosa.

Izgled aplikacije i osnovnih postavki prikazan je na slikama ispod (Slika 4.5), a izvorni kod aplikacije nalazi se u prilogu (mapa ZavrsniRadAndroid).



Slika 4.5 Izgled početnog ekrana mobilne aplikacije i ekrana MQTT postavki

5. Testiranje

Rad sustava ću testirati na privremenom radnom mjestu kombinirajući uređaje pri sobnoj temperaturi (24-25 °C). Promjenu radne temperature ću simulirati generatorom toplog zraka od otprilike 50-60 °C, pri tome za kontrolu temperature ću koristiti zasebni temperaturni senzor. Podaci koje ću prikupljati prilikom testiranja su: temperatura senzora sustava, temperatura kontrolnog senzora, vrijeme rada hlađenja (klime/ventilatora), vrijeme rada generatora toplog zraka. U obzir treba uzeti nekoliko parametara: brzinu razmjene MQTT poruka, brzina reakcije senzora, te vrijeme u kojem programska petlja senzora pokreće mjerenje temperature (10 s) i preciznost DHT22 senzora (+/- 0,5 °C). Na kraju, rad i reakciju sustava odrediti ću usporedbom podataka sustava i kontrolnih podataka (temperatura kontrolnog senzora i trajanje programske petlje).

5.1. Povezivanje uređaja

Prije pravog testa sustava sve uređaje treba spojiti na napajanje, uključiti, te provjeriti da li se razmjenjuju poruke između klijenata i posrednika. Zbog veće šanse da sve radi ispravno uređaje ću pokretati određenim redoslijedom:

1. Na CloudMQTT stranicama otvoriti ću *„Websocket UI“* karticu koju sam već prije spomenuli
2. Pokrećem pristupnu točku za Internet
3. Na napajanje ću priključiti Raspberry Pi, što automatski pokreće operacijski sustav i Mosquitto server
4. ESP8266 senzor spajam na napajanje
5. ESP8266 pokretač spajam na napajanje
6. Pokrećem Android aplikaciju i aktiviram konekciju

Programska petlja na senzoru se izvršava svakih 10 sekundi i stoga bi se svakih 10 sekundi trebala pojaviti poruka s temperaturom na *„Websocket UI“* kartici pod *„Received Messages“* i pri tome je *„Topic“* – *„sensor“*, a poruka vrijednost temperature, dok je druga tema *„control“* Na kartici *„Connections“* trebalo bi biti vidljivo da je spojen Raspberry Pi i Android aplikacija. Cijelim tijekom bi se na Android aplikaciji trebao osvježavati graf temperature i rada ventilatora. Preko aplikacije također testiramo ručno upravljanje

ventilatorom. Nakon uspješnog povezivanja i testiranja komunikacije krećem na testiranje rada sustava.

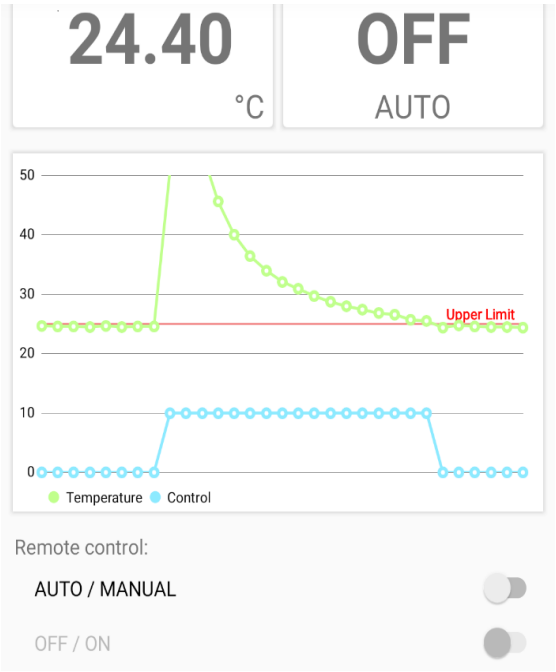
5.2. Testiranje rada sustava

Rezultati testiranja rada sustava prema već opisanom postupku dani su u tabličnom obliku (Podaci_prvog_testa.xlsx) u prilogu, dok je dio podataka prikazan u tablici niže (Tablica 5.1).

Tablica 5.1 Rezultati testiranja automatskog rada sustava

Redni broj mjerenja	**Temperatura senzora (°C)**	**Temperatura kontrolnog senzora (°C)**	**Ventilator upaljen/ugašen (1/0)**	**Vanjski generator temperature (1/0)**
…	…	…	…	…
8	24,6	35,7	0	1
9	50,9	53,6	1	1
10	64,3	67,2	1	0
…	…	…	…	…

Slika niže (Slika 5.1) prikazuje dio zaslona mobilne aplikacije nakon testa:



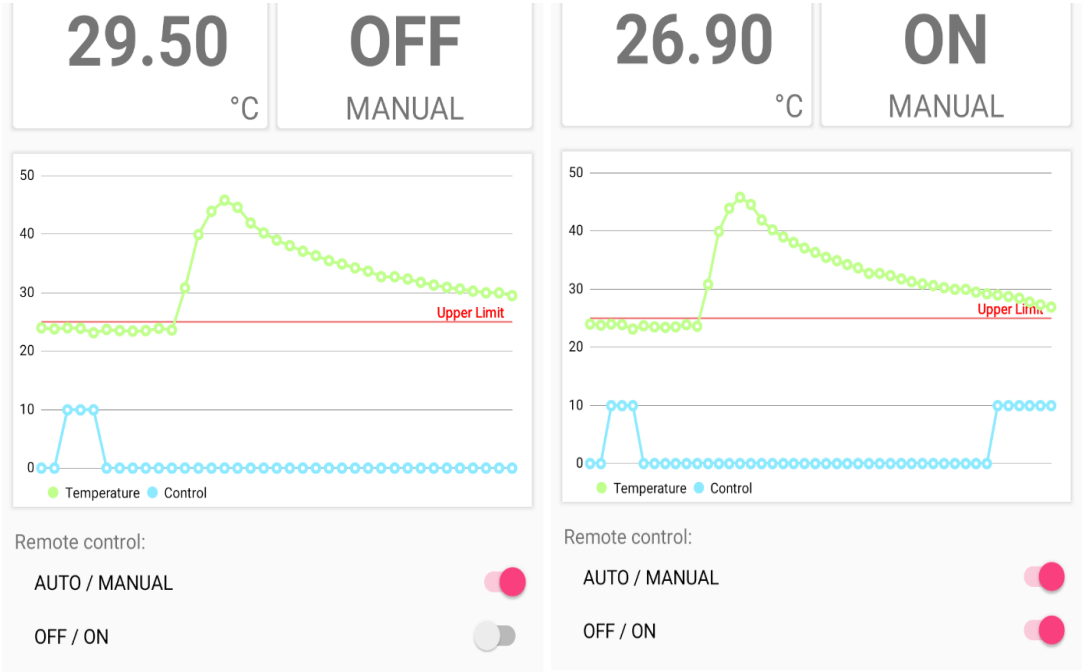
Slika 5.1 Dio zaslona mobilne aplikacije nakon prvog testa

Sljedeći test pokazuje rad sustava sa daljinskim upravljanjem preko mobilne aplikacije koja premošćuje normalan rad sustava. Kompletni podaci prikazani su u tablici (Podaci_drugog_testa.xlsx) u prilogu, a jedan dio podataka prikazan je u tablici niže (Tablica 5.2).

Tablica 5.2 Rezultati testa daljinskog upravljanja sustavom

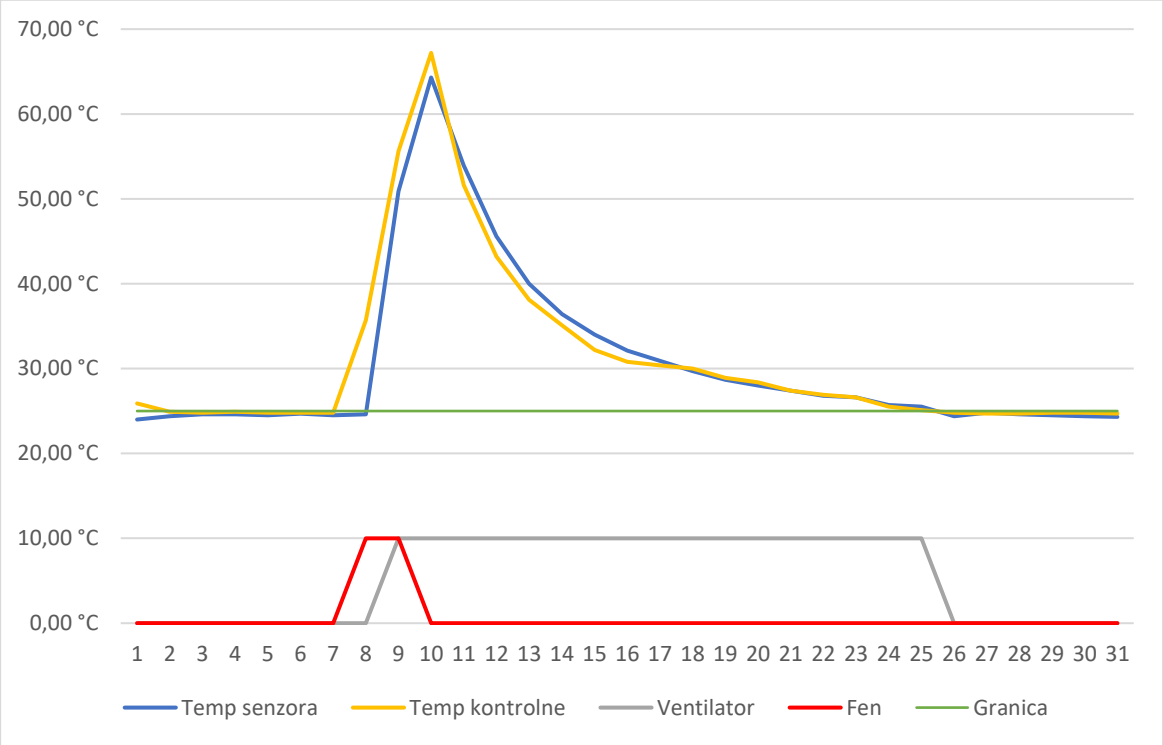
Redni broj mjerenja	**Temperatura senzora (°C)**	**Temperatura kontrolnog senzora (°C)**	**Ventilator upaljen/ugašen (1/0)**	**Vanjski generator temperature (1/0)**	**Ručno aktiviran/deaktiviran ventilator (1/0)**
…	…	…	…	…	…
2	23,1	23,4	1	0	1
3	23,7	23,6	1	0	1
…	…	…	…	…	…
90	30,8	31,2	0	1	0
100	39,9	40,2	0	1	0
…	…	…	…	…	…

Slike niže (Slika 5.2) prikazuju dio zaslona mobilne aplikacije drugog testiranja:



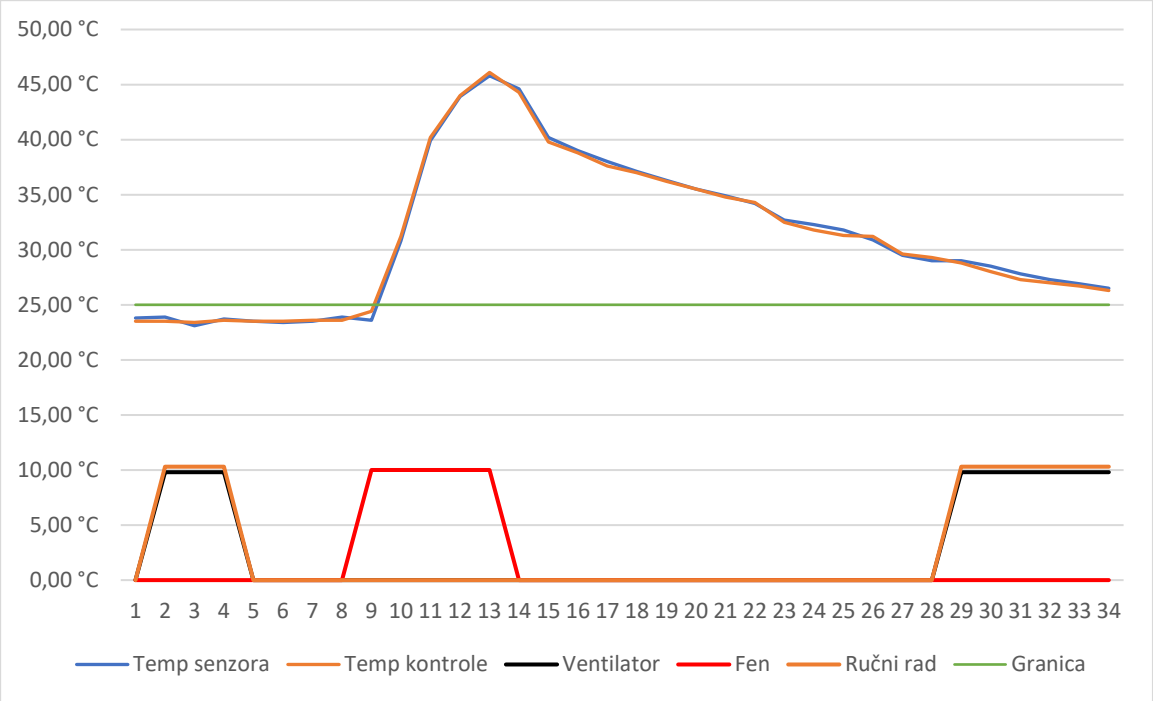
Slika 5.2 Dio zaslona mobilne aplikacije nakon 270 sekundi, te na kraju testa

5.3. Analiza rezultata



Graf 5.1 Analiza rezultata testiranja sustava u automatskom načinu rada

Kao što je vidljivo iz grafa gore (Graf 5.1) sustav radi ispravno, znači uključuje ventilator kada temperatura senzora prijeđe zadanu granicu (25 °C) i isključuje ga kada padne ispod granice. Razlike u linijama između temperature senzora i kontrolnog senzora su zbog razmaka u uzimanju uzoraka prilikom mjerenja. Također, iz grafa bi se moglo zaključiti da ventilator ima jako brzu reakciju no ona zapravo ovisi o razlici vremena kada su senzor i posrednik priključeni na napajanje. Ako je npr. posrednik upaljen 5 sekundi nakon senzora on će zapravo kasniti 5 sekundi sa reakcijom. Ako bi se vrijeme izvršavanja programske petlje smanjilo tada bi se smanjilo i vrijeme kašnjenja reakcije. Valja naglasiti da zbog prikaza na istom grafu vrijednosti ventilatora i fena (vanjski generator temperature) su isključen za 0.00 °C i uključen za 10.00 °C.



Graf 5.2 Analiza rezultata testiranja sustava daljinskim upravljanjem

Daljinsko upravljanje sustavom premošćuje normalan (automatski) rada sustava, a podizanje temperature normalno se odražava na očitanja senzora što je vidljivo iz grafa gore (Graf 5.2). Dok je ventilator uključen samo kada je aktiviran unutar mobilne aplikacije. Pad temperature, odnosno hlađenje sustava je normalno kada se ukloni dodatni izvor topline. Nešto malo strmija krivulja temperature vidljiva je kada se aktivirao ventilator za hlađenje. Tromost sustava prisutna je kao i u testu prije, no ona najviše ovisi o programskim ograničenjima.

Testiranje bih proglasio uspješnim, a ovaj prototip velikih ograničenja i nedostataka sasvim zadovoljavajućim za prikaz primjene IoT tehnologija i MQTT protokola u automatizaciji i poboljšanju industrije.

Zaključak

Internet stvari je relativno novi i zanimljiv koncept prema kojemu se dalje razvija ne samo Internet već i industrija, prateće tehnologije a i ljudski životi. IoT pokriva skup tehnologija od hardvera, softvera, komunikacije svih vrsta do servisa u oblaku i aplikacija. Obrada velike količine IoT podataka pomoću prediktivne analitike pomaže boljoj kvaliteti ljudskih života, ali i razumijevanju procesa kojima pridonosi poboljšanja i kvalitetnije održavanje.

Uz hardver malih dimenzija i niske potrošnje energije za kvalitetnu komunikaciju unutar IoT infrastrukture vrlo je bitan komunikacijski protokol. Uz bazni IP protokol vrlo je bitan i MQTT protokol za *real-time* razmjenu poruka. No IoT sadržava još nekoliko raznih žičnih, bežičnih i satelitskih protokola.

Upravo velik broj raznih tehnologija i manjak standarda jedan je glavnih nedostataka IoT koncepta. Jedan od većih nedostataka također je i manjak stručnjaka za IoT područje, dok se stvara bojazan za gubitak radnih mjesta nakon automatizacije.

Glavna prepreka za uvođenje IoT tehnologija je čisto vidljiva vrijednost, zbog toga je ovaj rad pokazao prednosti na primjeru kontrole temperature prilikom mjerenja kvalitete proizvedenih elektroničkih komponenti. Prototip sustava za kontrolu temperature izrađen je od lako dostupnih i jeftinih hardverskih komponenti te jednostavnih programa.

Testiranjem sustava zaključio sam da IoT jednostavno i jeftino unapređuje bilo koji proces, a svi podaci dostupni su sa bilo koje točke na svijetu. Prednost ovakvog sustava je velika modularnost jer se sustav može prenamijeniti za drugu upotrebu, a svaki dio sustava u slučaju kvara lako je zamijeniti. Svi prikupljeni podaci sustava mogu se analizirati u jednom od ponuđenih servisa u oblaku.

Ukratko, Internet stvari je moćan koncept sa velikim mogućnostima i primjena koje se još uvijek otkrivaju. Gotovo svakoj kompaniji korisno je i profitabilno da primijene barem jedan dio IoT tehnologije, a ako već nisu, trebale bi čim prije.

Popis kratica

ACL	*Access Control List*	
ADC	*Analog-to-Digital Converter*	Analogno-Digitalni pretvarač
AI	*Artificial Intelligence*	Umjetna inteligencija
AMQP	*Advanced Message Queuing Protocol*	
API	*Application Programming Interface*	
ARM	*Advanced RISC Machine*	
AWS	*Amazon Web Services*	
BAP	*Battery-Assisted Passive*	
BLE	*Bluetooth Low Energy*	Bluetooth niske energije
BLOB	*Binary Large Object*	Binarni veliki objekt
CPU	*Central Processing Unit*	
CoAP	*Constrained Application Protocol*	
DAC	*Digital-to-Analog Converter*	Digitalno-Analogni pretvarač
GCC	*GNU Compiler Collection*	
GPIO	*General Purpose Input/Output*	
GPS	*Global Positioning System*	
GPU	*Graphics Processing Unit*	
GSM	*Global System for Mobile Communications*	
HDMI	*High Definition Multimedia Interface*	
I^2C	*Inter-Integrated Circuit*	
ID	*Identity*	Identitet
IDE	*Integrated Development Environment*	
IEC	*International Electrotechnical Commission*	
IEEE	*Institute of Electrical and Electronics Engineers*	
IO	*Input-Output*	Ulaz-Izlaz
IP	*Internet Protocol*	
ISO	*International Organization for Standardization*	
IT	*Information Technology*	Informacijska tehnologija
IaaS	*Infrastructure-as-a-Service*	
IoT	*Internet of Things*	Internet stvari
LAN	*Local Area Network*	
LCD	*Liquid Crystal Display*	Ekran sa tekućim kristalom

M2M	*Machine-to-Machine*	
MB	*Megabyte*	Megabajt
MCU	*Microcontroller unit*	Mikrokontroler
MQTT	*Message Queuing Telemetry Transport*	
NFC	*Near Field Communication*	
OASIS	*Organization for the Advancement of Structured Information Standards*	
OS	*Operating system*	Operacijski sustav
PC	*Personal Computer*	Osobno računalo
PWM	*Pulse Width Modulation*	
PaaS	*Platform-as-a-Service*	
QoS	*Quality of Service*	
RAM	*Random Access Memory*	
RF	*Radio Frequency*	Radijska frekvencija
RFID	*Radio-frequency identification*	
RISC	*Reduced Instruction Set Computer*	
ROM	*Read-Only Memory*	
RTOS	*Real-time operating system*	
SCADA	*Supervisory Control and Data Acquisition*	
SDHC	*Secure Digital High Capacity*	
SDK	*Software Development Kit*	
SMT	*Surface-Mount Technology*	
SPI	*Serial Peripheral Interface*	
STOMP	*Simple Text Oriented Messaging Protocol*	
SaaS	*Software-as-a-Service*	
SoC	*System-on-Chip*	Sistem na čipu
TCP	*Transmission Control Protocol*	
TV	*Television*	Televizija
UART	*Universal Asynchronous Receiver/Transmitter*	
UI	*User Interface*	
USB	*Universal Serial Bus*	
UV	*Ultraviolet*	Ultraljubičasto
VM	*Virtual Machine*	Virtualni stroj
WAMP	*Web Application Messaging Protocol*	
WAN	*Wide Area Network*	
XMPP	*Extensible Messaging and Presence Protocol*	

Popis slika

Popis tablica

Popis grafova

Popis kôdova

Literatura

[1] WIKIPEDIA, https://en.wikipedia.org/wiki/Internet_of_things, studeni 2018.

[2] CARNEGIE MELLON UNIVERSITRY, The "Only" Coke Machine on the Internet, https://www.cs.cmu.edu/~coke/history_long.txt, studeni 2018.

[3] I-SCOOP, https://www.i-scoop.eu/internet-of-things, studeni 2018.

[4] IBM, https://www.ibm.com/cloud/garage/architectures/iotArchitecture/overview, studeni 2018.

[5] GARTNER, https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016, studeni 2018.

[6] WIRED, https://www.wired.com/story/guide-self-driving-cars/, studeni 2018.

[7] POSTSCAPES, https://www.postscapes.com/internet-of-things-hardware/, studeni 2018.

[8] POSTSCAPES, https://www.postscapes.com/companies/iot-hardware-companies/, studeni 2018.

[9] IOT BYTES, https://iotbytes.wordpress.com/popular-hardware-platforms-for-iot/, studeni 2018.

[10] LIBELIUM, http://www.libelium.com/, studeni 2018.

[11] DZONE, https://dzone.com/articles/everything-you-need-to-know-about-iot-hardware, studeni 2018.

[12] POSTSCAPES, https://www.postscapes.com/internet-of-things-technologies/, prosinac 2018.

[13] POSTSCAPES, https://www.postscapes.com/internet-of-things-software-guide/, studeni 2018.

[14] IBM, https://developer.ibm.com/articles/iot-lp101-connectivity-network-protocols/, prosinac 2018.

[15] IBM, https://developer.ibm.com/tutorials/iot-lp301-iot-manage-data/, studeni 2018.

[16] IBM, https://www.ibm.com/analytics/machine-learning, studeni 2018.

[17] MICROSOFT, https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/, studeni 2018.

[18] LINKEDIN, https://www.linkedin.com/pulse/pros-cons-internet-things-iot-bhaskara-reddy-sannapureddy, studeni 2018.

[19] DZONE, https://dzone.com/articles/iot-concerns, studeni 2018.

[20] MQTT, http://mqtt.org/faq, studeni 2018.

[21] HIVE MQ, https://www.hivemq.com/blog/how-to-get-started-with-mqtt/, studeni 2018.

[22] OASIS, MQTT Version 3.1.1, http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf, studeni 2018.

[23] DZONE, https://dzone.com/articles/mqtt-the-nerve-system-of-iot, studeni 2018.

[24] WIKIPEDIA, https://en.wikipedia.org/wiki/Comparison_of_MQTT_Implementations, studeni 2018.

[25] GITHUB, https://github.com/mqtt/mqtt.github.io/wiki/libraries, studeni 2018.

[26] GITHUB, https://github.com/mqtt/mqtt.github.io/wiki/server-support, studeni 2018.

[27] DEEPSTREAM HUB, https://deepstreamhub.com/blog/an-overview-of-realtime-protocols/, studeni 2018.

[28] DIY PROJECTS, https://diyprojects.io/8-online-mqtt-brokers-iot-connected-objects-cloud/, studeni 2018.

[29] DZONE, https://dzone.com/articles/the-best-iot-platforms-today, prosinac 2018.

[30] BOSCH, https://www.bosch-iot-suite.com/, prosinac 2018.

[31] ORACLE, https://cloud.oracle.com/iot, prosinac 2018.

[32] ORACLE, https://cloud.oracle.com/iot-apps, prosinac 2018.

[33] IBM,https://console.bluemix.net/docs/services/IoT/index.html#gettingstartedtemplate, prosinac 2018.

[34] MICROSOFT, https://docs.microsoft.com/en-us/azure/iot-fundamentals/, prosinac 2018.

[35] GOOGLE, https://cloud.google.com/solutions/iot-overview, prosinac 2018.

[36] GOOGLE, https://cloud.google.com/pubsub/docs/overview, prosinac 2018.

[37] AMAZON, https://aws.amazon.com/iot/?nc=sn&loc=1, prosinac 2018.

[38] MICROSOFT, https://docs.microsoft.com/en-ca/azure/iot-hub/iot-hub-devguide-quotas-throttling, studeni 2018.

[39] MEDIUM, https://medium.com/@iskerrett/price-comparison-of-iot-platform-vendors-b07ab4bbf0e, prosinac 2018.

[40] RAPBERRY PI, Readme, https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md, listopad 2018

[41] HACK A DAY, http://hackaday.com/2015/02/02/introducing-the-raspberry-pi-2/, listopad 2018

[42] NURDS, https://nurdspace.nl/ESP8266#Translated_datasheet, listopad 2018.

Prilog

Kontrolne_MQTT poruke.xlsx

Cloud_MQTT posrednici.xlsx

Cijene_Cloud_usluga.xlsx

Izračun_scenarija.xlsx

Podaci_prvog_testa.xlsx

Podaci_drugog_testa.xlsx

Senzor\Senzor.ino

Kontrola\Kontrola.ino

ZavrsniRadAndroid

Kreiranje_nove_instance.jpg, preuzeto sa https://customer.cloudmqtt.com/instance/create, siječanj 2019.

Detalji–Details.jpg, preuzeto sa https://api.cloudmqtt.com/console/25074301/details, siječanj 2019.

Korisnici_i_prava–Users_&_ACL.jpg, preuzeto sa https://api.cloudmqtt.com/console/25074301/users, siječanj 2019.

Websocket_UI.jpg, preuzeto sa https://api.cloudmqtt.com/console/25074301/websocket, siječanj 2019.

Konekcije–Connections.jpg, preuzeto sa https://api.cloudmqtt.com/console/25074301/connections, siječanj 2019.