



УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ” - ШТИП

ФАКУЛТЕТ ЗА ИНФОРМАТИКА

Вештачка интелигенција и роботика

Штип

РАТКО ИВАНОВ

**ИЗРАБОТКА И КОНТРОЛА НА МОДЕЛ НА ЕНЕРГЕТСКИ ЕФИКАСЕН ПАМЕТЕН
ДОМ СО ТЕХНИКИ НА ВЕШТАЧКА ИНТЕЛИГЕНЦИЈА**

-МАГИСТЕРСКИ ТРУД-

Штип, февруари, 2021

Комисија за оценка и одбрана

Ментор: проф. д-р Цвета Мартиновска-Банде

Факултет за информатика при Универзитетот „Гоце Делчев“ - Штип

Член/претседател: доц. д-р Наташа Стојковиќ

Факултет за информатика при Универзитет „Гоце Делчев“ - Штип

Член: доц. д-р Доне Стојанов

Факултет за информатика при Универзитет „Гоце Делчев“ - Штип

Февруари, 2021 год.

Благодарност

По повод изработката на овој магистерски труд сакам да изразам благодарност до сите оние што помогнаа овој труд да ја добие потребната содржина во која ќе биде презентираан.

Најпрво огромна благодарност упатувам до мојот ментор, проф. д-р Цвета Мартиновска-Банде за долгогодишната успешна соработка, за целокупната неизмерна поддршка која ми ја даваше и за огромното знаење кое несебично ми го пренесе во текот на моите постдипломски студии.

Голема благодарност сакам да упатам и до членовите на комисијата за одбрана доц.д-р Наташа Стојковиќ и доц.д-р Доне Стојанов кои со својот професионален ангажман придонесоа за изработката на овој магистерски труд.

Воедно, сакам да упатам и голема благодарност до моето семејство и моите најблиски за нивната морална поддршка, поттик и мотивација.

Ратко Иванов

Листа на објавени трудови

1. Ivanov Ratko, Ivanov Vlatko, Martinovska Bande Cveta (2020). Energy efficiency in Smart Home system. International Journal of Scientific and Research Publications, Volume 10, Issue 9, September 2020 ISSN 2250-3153 pp(683-688)
2. Ivanov Vlatko, Ivanov Ratko, Martinovska Bande Cveta (2020). Real time alert system in Smart Home. International Journal of Scientific and Research Publications, Volume 10, Issue 9, September 2020 ISSN 2250-3153 pp(676-682)

ИЗРАБОТКА И КОНТРОЛА НА МОДЕЛ НА ЕНЕРГЕТСКИ ЕФИКАСЕН ПАМЕТЕН ДОМ СО ТЕХНИКИ НА ВЕШТАЧКА ИНТЕЛИГЕНЦИЈА

КРАТОК ИЗВАДОК

Со напредокот на технологијата, постојано се стремиме да ги автоматизираме процесите што ќе го олеснат секојдневието и истовремено да ги намалат дневните трошоци. Така, денес постојат паметни домови кои имаат за цел да ги намалат дневните обврски и трошоци, а во исто време да ја зголемат безбедноста и удобноста на домот. Технологијата „Smart Home“ е општ поим кој се однесува на домови кои имаат удобности поврзани со комуникациските технологии, овозможувајќи автоматски или далечински управувач. „Паметен дом“ нуди можности за заштеда на енергија и намалување на трошоците, подобрување на квалитетот на животот и зголемување на безбедноста. Со техниките на вештачката интелигенција се добиваат големи придобивки и олеснувања во самиот дом. Автоматизацијата на домот обезбедува удобност, безбедност, заштеда на енергија, флексибилност и приспособливост кон идниот развој.

Клучни зборови: Енергетска ефикасност, паметен дом, вештачка интелигенција

MAKING AND CONTROL OF A MODEL OF ENERGY EFFICIENT SMART HOME WITH ARTIFICIAL INTELLIGENCE TECHNIQUES

ABSTRACT

With the advancement of technology, we are constantly striving to automate the processes that will make everyday life easier and at the same time reduce daily costs. Thus, today there are smart homes that aim to reduce daily obligations and costs, and at the same time increase the safety and comfort of home. "Smart Home" technology is a general term that refers to homes that have the comforts associated with communication technologies, providing automatic or remote control. "Smart Home" offers opportunities to save energy and reduce costs, improve the quality of life and increase security. Artificial intelligence techniques provide great benefits and relief at home. Home automation provides comfort, safety, energy savings, flexibility and adaptability to future development.

KEY WORDS: Energy efficiency, smart home, artificial intelligence

СОДРЖИНА:

| | |
|--|----|
| 1. Вовед | 1 |
| 2. Преглед на литература | 2 |
| 3. Интернет на нештата | 4 |
| 4.1. Историја на IoT..... | 5 |
| 4.2. Интернет на нештата во нашите животи..... | 11 |
| 4.2.1. Паметно здравство..... | 11 |
| 4.2.2. Транспорт..... | 11 |
| 4.2.2.1. Безбедност..... | 12 |
| 4.2.3. Индустија | 13 |
| 5. Паметен дом..... | 14 |
| 6. Заштеда на енергија..... | 16 |
| 7. Вештачка интелигенција..... | 18 |
| 8. Користени технологии | 20 |
| 8.1. Софтвер | 21 |
| 8.1.1. Angular..... | 21 |
| 8.1.2. Spring boot..... | 21 |
| 8.1.2.1. Зошто Spring boot е популарен | 22 |
| 8.1.3. Python..... | 22 |
| 8.1.4. ActiveMq..... | 23 |
| 8.1.4.1. Група на пораки..... | 23 |
| 8.1.4.2. Виртуелни дестинации | 24 |
| 8.1.5. Mqtt | 24 |
| 8.1.5.1. Модел објави – претплати (publish-subscribe)..... | 27 |
| 8.1.5.2. MQTT конекција..... | 28 |
| 8.1.5.3. Објави и претплати | 30 |
| 8.1.5.4. MQTT тема | 34 |
| 8.1.5.4.1. Wildcards | 35 |
| 8.1.5.5. Клиент | 35 |
| 8.1.6. WebSockets..... | 36 |
| 8.2. Хардвер | 36 |
| 8.2.1. NodeMCU..... | 36 |

| | |
|---|----|
| 9. Опис на проектот..... | 42 |
| 9.1. Енергетска аналитика | 42 |
| 9.2 Архитектура | 43 |
| 9.3. Изработка..... | 44 |
| 9.3.1 Софтвер | 44 |
| 9.3.1.1. Spring boot..... | 44 |
| 9.3.1.1.1. H2 | 44 |
| 9.3.1.1.2. Mqtt | 45 |
| 9.3.1.1.3. Rest API | 46 |
| 9.3.1.2. Front end Angular | 48 |
| 9.3.1.2.1. Гласовни команди..... | 51 |
| 9.3.1.3. Програмирање на NodeMCU..... | 51 |
| 9.3.1.4. Систем за предикција на температура со примена на LSTM невронска мрежа..... | 51 |
| 9.3.1.4.1 LSTM..... | 52 |
| 9.3.1.4.2 Имплементација на системот за предвидување на температура | 56 |
| 9.3.2. Хардвер | 61 |
| 9.3.2.1. Сензори и уреди | 61 |
| 9.3.2.1.1. Сензор за мерење на температура и влажност (DHT11) | 61 |
| 9.3.2.1.2. Reed switch..... | 62 |
| 9.3.2.1.3. LED | 63 |
| 9.3.2.1.4. Детектор на дожд (Rain detector) | 64 |
| 9.3.2.1.5. Отпорник на светлина (LDR) | 65 |
| 9.3.2.1.6. Серво мотори..... | 66 |
| 9.3.2.1.7. Сензор за влага во почвата..... | 67 |
| 9.3.2.1.8. Сензор за звук..... | 69 |
| 9.3.2.1.9. 3V DC Мотор..... | 70 |
| 9.3.3. Шематски приказ..... | 71 |
| 9.3.3.1 Поврзување на сензорот за температура и влажност, серво моторот и сензорот за детектирање на дожд со NodeMCU..... | 71 |
| 9.3.3.2 Поврзување на магнетен сензор, сензорот за детектирање на влага во почвата и серво мотор со NodeMCU..... | 73 |
| 9.3.3.3 Поврзување на отпорникот на светлина, сензорот за звук и лед диоди со NodeMCU | 75 |
| 9.3.3.4 Поврзување на 3V DC мотор, L293d motor driver со NodeMCU..... | 77 |

| | |
|---------------------------------|----|
| 10. Резултати и дискусија | 80 |
| 11. Заклучок | 84 |
| 12. Користена литература..... | 85 |

1. Вовед

Светската побарувачка на енергија постојано се зголемува. Додека многу сектори (на пример, транспорт, производна индустрија) веќе неколку години се обидуваат да ја намалат потрошувачката на енергија, намалувањето на потрошувачката на енергија во домовите е на самиот почеток. Ова произлегува барем делумно од фактот дека иако свеста и мотивацијата за заштеда на енергија во денешно време обично постојат кај сопствениците на куќи, адекватната технолошка поддршка за корисниците во голема мера недостасува. Ова се однесува пред сè на достапноста на посветени, сеопфатни системи кои поддржуваат енергетски ефикасно работење на дом или зграда. Имајќи ги предвид рапидно растечките трошоци за енергија, намалената потрошувачка на енергија има економски бенефиции, но се исплаќа и на макроскопско ниво, каде што треба да се исполнат националните и меѓународните еколошки цели и закони.

Во текот на изминатата деценија, паметните домови станаа новонастанато прашање во академските истражувања, како и во секторот за станбени згради. Примамливата визија за паметна контрола врз околината ги мотивира сопствениците на домови да ја интегрираат технологијата за автоматизација во своите домови со ветувачки ефекти на зголемена удобност, мир на умот и намалени оперативни трошоци.

Технологиите на паметен дом и вештачката интелигенција (ВИ) полека стануваат неразделни. Со обработката на податоците собрани од сензорите вештачката интелигенција станува еден од клучните фактори во паметните домови.

Целта на овој магистерски труд е изработка на модел на паметен дом и систем за контрола и следење на сензори и уреди со цел да се постигне енергетска ефикасност. Од техниките на вештачка интелигенција ќе се искористат говор во техниката текст за испраќање на гласовни команди, а од добиените мерења од сензорот за температура ќе се изврши предикција на температурата за наредните 7 дена. Со ова би се прикажал целиот процес од собирањето на податоците, анализирањето и прикажувањето на нив преку изработена веб-апликација.

2. Преглед на литература

Во последната декада паметните домови се една од најактуелните теми кај научниците. Објавените истражувања за паметен дом растат експоненцијално. Како дел од сферата на енергетската ефикасност во паметните домови се прегледани следниве литератури:

[1] Вуд Г. и Њубороу М. во 2003 година се фокусирале на паметни броила-прикажувачи и потрошувачката на енергетската енергија. Тие направиле теренска студија на 44 домови каде што се готвело, а 12 контролни домови биле следени 12 месеци со последователна интервенција на повратни информации биле тестирани три методи за влијание врз однесувањето:

1. Инфо пакет;
2. Индикатори за потрошувачка на електрична енергија;
3. Првите два метода заедно.

Имале четири групи со приближно еднаква големина кои биле следени вкупно 18 месеци.

Резултатите што ги постигнале биле:

Просечно домаќинство троши електрична енергија за готвење која изнесува 1300 kW/h на година, просечната заштеда со користење на индикаторите за потрошувачка на електрична енергија изнесувала 15%, додека кај оние на кои им било дадено само информации заштедиле само 3%. Домаќинствата на кои им било поставени само монитори заштедиле повеќе во просек 15.2% од оние кои:

1. Кои имале само претходни информации 3%;
2. Оние со монитори и претходни информации 8.9.

[2] Од друга страна тимот од научници (Бејтс, Хазас, Морли и др.) претставиле метод за енергетските услуги што се користи како идејна и практична врска помеѓу податоците за електрична енергија - податоците од сензорите и домашните практики иако (практиките се користат повеќе во смисла на значајни групи од активности во домот).

Нивните резултати биле дека податоците од сензорите и разделени податоци од електричната енергија можат да се користат во различен степен за да се идентификува кога и каде се користат енергетски услуги во домовите, но овие енергетски услуги се поврзани со повеќе практики, некои рутински, некои не.

Услугите за осветлување и готвење се идентификувале поедноставно преку специфични уреди (светла, шпорет). Добри аргументи изнесени за придобивките од услугите за потрошувачката на домаќинствата:

„Пристапот ориентиран кон услуги ќе бара значително повеќе сметководствени напори отколку оние ориентирани на ресурси. Тврдиме дека исплатата за овој напор е пренасочување од основни мерки за тоа колку енергија трошат апаратите, кон поинклузивно разбирање на пакетот од уреди, инфраструктури, практики, очекувања и ресурси што ја сочинуваат услугата. За истражувачите кои ги разгледуваат можностите за дизајн системот може веднаш да открие каде услугите подржуваат единични или повеќе практики, кои уреди функционираат во групи и зошто и да резултира во поширок поглед на можните реконфигурации на услугите.“

Уште еден труд кој укажува на вредноста на паметните домови е „Energy Management for Smart Homes“ [3]. Бежад Лашкари и Јуксијанг Чен направиле истражување во кое акцентот е ставен на комуникациските протоколи и носечки и неносечки сензори. Како заклучок, истакнуваат дека се очекува паметните домови да станат сè поважни во блиска иднина. Ова ќе ја интензивира побарувачката за ефикасно управување со енергијата со оглед на удобноста на корисниците. Според нив неколку други предизвици сè уште треба да се решат во контекст на паметни домови, како што се распоредување ефикасни сензорски технологии, препознавање и предвидување на активност, методи за складирање и анализа на податоци. Покрај тоа, идните паметни домови ќе коегзистираат со други паметни технологии и ќе бараат конвергенција на соодветните мрежи и услуги. Ова јасно ја илустрира мултидисциплинарната природа на оваа област и бара вклучување на голем број специјалисти во развојот на идните паметни домови.

Според „Oracle“ интернет на нештата претставува:

Интернетот на нештата (IoT) ја опишува мрежата на физички објекти - „нештата“ - кои се вградени во сензори, софтвер и други технологии заради поврзување и размена на податоци со други уреди и системи преку интернет. Овие уреди се движат од обични предмети за домаќинството до софистицирани индустриски алатки. Со повеќе од 7 милијарди поврзани IoT уреди денес, експертите очекуваат овој број да порасне на 10 милијарди до 2020 година и 22 милијарди до 2025 година. [4]

4.1. Историја на IoT

За целосно да се сфати идејата на IoT, прво треба да се открие како се развивал со текот на времето. Со поедноставни зборови, IoT е околина од меѓусебно поврзани уреди. Го користи интернетот да сподели податоци и може да се поврзе со скоро кој било уред на кој може да се помисли. Листата од IoT технологии што поддржува робусни мрежни перформанси вклучува разновидни развојни околинати, „облак“-овозможени платформи, со голем број комуникациски протоколи кои работат на различен хардвер. Ова се мобилни телефони, фитнес додатоци, сензори за движење, коли, апарати за паметни домови и ред други. Но, ова се алатки кои се достапни во сегашноста, постарите IoT уреди биле различни.

Првиот фиксен телефон на телеграфот бил развиен во **1830-‘40**. Овозможил директна комуникација и инспирирал понатамошен развој за слични технологии.

Првиот безжичен пренос на глас е изведен на 3 јуни **1900**, преку радио, туркајќи го работ на иновацијата. Оваа витална компонента на иднината на IoT околината била надополнета со компјутерите кои се појавија во педесеттите години од минатиот век.

1982 – Студентите на Универзитетот Кернеги Мелон (Carnegie Mellon University) ја измислиле АРПАНЕТ – поврзаната вендинг машина . Четири студенти од одделот на Факултетот за компјутерски науки инсталирале прекинувачи во машината вендинг. Тие броеа колку шишиња сок имаат останато во секој ред и за

колку долго. Ако наполнето шише остане долго време во машината, се означува како „ладно“. Сите овие податоци тогаш биле далечински достапни за клиентите преку интерфејс на прст (finger interface) [5]. Овој експеримент инспирирал многу пронаоѓачи ширум светот да создадат свои поврзани апарати.

1989 – Тим Бернерс-Ли ја предложил World Wide Web (интернетот каков што го знаеме). Сè започнало како мрежа за префрлување пакети, која што потоа растела и се развивала. Покрај тоа, глобалниот систем за позиционирање станал достапен за комерцијална употреба, владиниот систем на сателити бил надополнет со приватни, обезбедувајќи ги идните IoT системи со основна комуникација [6]. Од тогаш, пронаоѓачите имаат можност да развиваат меѓусебно поврзани уреди со олеснување. Први примери се:

1990 – На годишната изложба организирана од IBM, Џон Ромки (John Romkey) претставува тостер поврзан на интернет [7]. Се смета дека тостерот е првиот IoT уред. Бил поврзан со компјутер преку TCP/IP протокол;

1993 – Xcoffe е создаден во компјутерска лабораторија на Универзитетот Кембриџ. Двајца академици кои работеле во Тројанската соба во лабораторијата имале видео грабач (video-frame grabber) прикачен на еден од компјутерите. Одлучиле да поврзат камера контролирана од страна на сервер апликација што го набљудувала кафето и го фотографирала неколку пати во минута. Потоа, сликите се испратени до клиент апликацијата, да го известува корисникот со информации дали се наполнил садот;

1994 – “Една од првите камери на WWW”. Стив Ман(Steve Mann) создава нослива камера поврзана на интернет [8]. WearCam бил „експеримент за поврзаност” – ги пренесувала сликаните фотографии на компјутерот преку интернет;

1997 – на 13,14 октомври се одржал првиот меѓународен симпозиум за преносни компјутери. Универзитетот Кернеги Мелон (Carnegie Mellon University), веќе познат по својата машина за кола, MIT И Georgia Tech биле ко-домаќини на овој настан за да дискутираат за поддршката за перформанси на преносните

компјутери, механизмите за внесување, комуникациските способности и многу повеќе;

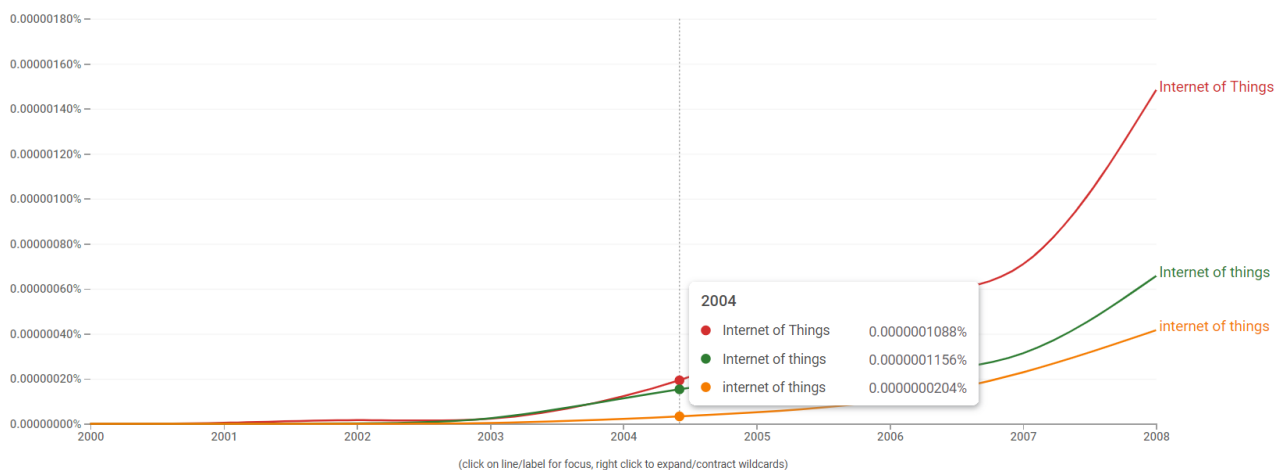
1998 – Воведување на IPv6. Овој протокол значително го зголеми просторот за адреси, кој беше имплементиран во 2017. Тој стана одличен додаток на IoT мрежата, кој веќе имаше повеќе уреди отколку вкупен број на човечка популација. До 1998 било тешко да се замислат и достигнат широки IoT мрежи бидејќи едноставно немало доволно IP адреси за да ги регистрираат сите поврзани уреди;

1999 – Кевин Ештон (Kevin Ashton) го воведува терминот “Интернет на Нештата”. Кевин, извршниот директор на Auto-ID Labs (Лаборатории за авто-идентификација) на MIT, го има искористено овој израз во својата презентација за Procter и Gamble, и сега целиот свет го користи [9]. Во тоа време идејата понекогаш била нарекувана „вграден интернет”, што не ја опишува соодветно. Во оваа презентација, Кевин ги поврзува идеите за RFID и Интернетот, што многу ги воодушевило слушателите поради ваквата иновативност. Auto-ID Labs лансирани. Подоцна истата година Кевин Ештон (Kevin Ashton), Дејвид Брок (David Brock) и Санџеј Шарма (Sanjay Sharma) ја спровеле идејата на Кевин за RFID поврзување. Воспоставиле лаборатории за авто-идентификација – мрежа на академски лаборатории за истражување, чија цел е развивање на нови технологии и нивна имплементација во доменот на трговијата. Како резултат, учествувале во развој на RFID-базиран идентификациски систем - Електронски код на производ (EPC Electronic Product Code), кој бил замислен да ги замени баркодските. ЕПЦ може да го обезбеди секој објект во светот со свој идентификациски таг. ЕПЦ е супер пример за можноста на IoT да поврзи физички објекти во хетерогена мрежа со помош на сензори и други безжични средства за комуникација;

2000 – LG го најавува интернет дигиталниот DIOS – првиот ладилник со достапност на интернет во светот [10]. Продажбата сепак не поминала добро, не бил прифатен на потрошувачкиот пазар, и неговото производство престанало, цената се сметала за превисока;

2004 - Терминот „Интернет на нештата“ станува популарен. Огромни издавачи, весници и списанија започнуваат да го споменуваат IoT во своите

натписи. На пример, Бостон глоуб има архива на своите публикации каде може да се најде натписот „Интернет на нештата“.



Слика 2 популарноста на интернет на нештата

Figure 2 IOT popularity

Старт-апи скокаат на следната голема работа: мали мрежни чипови“ од 25 октомври 2004 година. Написот зборува за магацин опремен со 600 безжични радио микрочипови. Ова била успешна мрежа на IoT мрежа која имала за цел да ги замени жичените контролери и да обезбеди простор за идни напредни апликации, како на пример софтвер за далечинско следење. Била предвидена ваквата високотехнолошка иднина, со вакви мрежи распространети низ други магацини, домови, продавници и други места.

2005 - Се појавува Nabaztag. Уредот поврзан со Интернет Nabaztag е мал робот за потрошувачка, произведен за да се поврзе со WiFi мрежи, да собира извештаи за временска прогноза, новости и промени на берзата и да ги чита на глас за сопственикот [11]. Овој амбиентален уред во форма на зајак му покажал на светот како Интернетот на нештата може нежно да се вгради во секојдневниот живот. Nabaztag бил ажуриран трипати низ својата историја, а во 2015 година неговите сервери и услуги на клиенти се откажани поради неможноста технолошки да напредуваат.

2008 - Прва меѓународна конференција за Интернет на нештата (IoT 2008). Настанот е прва конференција од ваков вид и се одржал во Цирих, Швајцарија. Биле опфатени теми за развој на Интернет на нештата, имено безжични комуникации со краток опсег, локализација во реално време, сензорски мрежи. Околу 250 индивидуални автори од 23 земји ги доставиле своите трудови за IoT за да бидат прегледани.

2009 - Google започна со тестирање на автомобили што управуваат самостојно. Првиот автомобил кој помина автономно возење беше Тојота Приус. На горната палуба имаше уред со сензор, кој имал можност да лоцира и открие пешаци, велосипедисти, работа на пат и други вредни предмети. Сега софтверот е доволно моќен за да го предвиди однесувањето на сите учесници во сообраќајот. St. Jude Medical станал усвојувач на IoT за здравствена заштита. Лансирал безжичен USB адаптер кој прима податоци од имплант на срце кај пациентот, па потоа безбедно ги пренесува податоците преку мобилните мрежи на лекарите. Кога било невозможно да се поврзи на мрежа, уредот вршел далечинско следење.

2010 - Кина ја бира IoT како клучна индустрија. Кинескиот премиер Вен Џиабао (Wen Jiabao) посакувал да се справи со финансиската криза и други прашања чувствителни на земјата. Тој одлучил да посвети дополнително внимание на IoT откако разговарал со истражувачот кој студирал во странство. Вен земал предвид дека Интернетот на нештата може да е совршена технологија за имплементација кај врвните стратешки индустрии. Идејата за далечинско управување и идентификување на различни објекти изгледала совршена за развојот на инфраструктурата и услугите на земјата. Ова е прв пат IoT да биде признат на владинско ниво. Компанијата Нест(Nest) започнува со производство на апарати за паметни домови. Поранешните инженери на Apple се вклучија во стартап што произлегол од идејата дека термостатите не ги исполнуваат барањата на пазарот. Првиот производ е Термостат за учење, кој може да ги научи навиките на корисниците и, како резултат, да го оптимизира распоредот за греење. Ова е првиот IoT термостат во светот придвижуван со помош на машинско учење.

2011 - Светски ден на IPv6. Настанот е организиран од страна на Facebook, Google, Yahoo, Akamai Technologies и Limelight Networks. Главната идеја на настанот била да се мотивираат провајдерите на Интернет, компаниите за веб-хостинг и слично, низ целата индустрија да се подготват за премин од IPv4 на IPv6 кога ќе се потполни просторот за IPv4. Повеќе од 400 учесници, вклучувајќи пребарувачи, социјални мрежи и мрежи за дистрибуција на содржина се приклучиле на експериментот.

2013 – Гугл(Google) најавува бренд паметни очила. „Гугл стакло“ е уред поставен на главата, произведен за да одговара на обликот на очилата. Се одликува со дисплеј кој има можност да прикажува информации без раце и модул за препознавање глас на природен јазик за поврзување со Интернет преку говорни команди. Сепак, проектот бил затворен се до 2017 година.

2017 – Microsoft лансира Azure IoT edge кој им овозможува на малите уреди да користат облак сервиси, дури и ако тие не се поврзани со облакот. Амазон нуди напредни карактеристики за безбедност и управување со уредот. Google го лансира Cloud IoT Core што им овозможува на уредите полесно да се поврзат со облак.

Судејќи според овие факти, може да се увиди дека главните технолошки лидери имаат сфатено дека Интернетот на нештата бара различен сет на алатки и услуги за правилно функционирање, и како резултат нуди облак платформи. Владите ширум светот почнуваат да размислуваат за безбедноста на IoT. Следејќи ги нарушувањата на сајбер безбедноста што им се случиле на различни компании, владите започнале да размислуваат за развој на законите за да се направат IoT-околините побезбедни. На пример, сенаторот Марк Ворнер го вовеле „Акт за подобрување на сајбер безбедноста на Интернет на нештата“ во американскиот Сенат. Актот сугерира да се забранат хард-кодирани лозинки и да се доставуваат закрпи до уредите, што претставува ниска граница за безбедност на IoT. Кripto-валутите се фокусираат на IoT. Блокчејн беше поврзан со IoT поради горенаведените безбедносни проблеми. Со помош на блокчејн уредите можат да разменуваат паметни договори во безбедна дистрибуирана околина. За да помогне да се прифати блокчејн и да се направи IoT безбеден, е формирана специјална



Слика 4 интернет на нештата во транспортот

Figure 4 IOT in transport

Собирањето и споделувањето податоци помеѓу уредите за донесување одлуки без човечка интеракција е она што го прави IoT. Кога станува збор за превозот, IoT е она што ќе им помогне на самостојните автомобили во иднината да комуницираат едни со други и да ве заштитат [15]-[16].

4.2.2.1. Безбедност

Безбедноста е честопати една од првите работи што на многумина им паѓа на ум кога разговараат за влијанието на IoT врз транспортот. Со можност за комуникација во реално време користејќи податоци од различни извори, возилата што управуваат самостојно ја зголемуваат безбедноста не само на луѓето во возилото, туку и на пешаците. Преку паметни сензори, транспортните возила можат да добијат информација за какви било несреќи и пречки во сообраќајот на тековните правци. Тие исто така можат да им помогнат на возачите доколку станат неспособни или имаат потреба од итна медицинска помош.

4.2.3. Индустирија

Индустиријата е област во која примената на IoT решенијата според повеќето прогнози ќе биде масовна. Благодарение на оваа технологија, голем број работни места ќе бидат автоматизирани, што значително ќе ја зголеми ефикасноста и продуктивноста (и на крајот профитот) [17].

General Electric во соработка со Intel создадоа IoT систем за индустриски постројки - Predix, [18] со кој планираат да ја направат индустиријата „паметна“. На пример, еден дел од овој систем се состои од сензори вградени во машини кои лесно го лоцираат дефектот (некои дури и предупредуваат однапред каде прво може да се појави дефектот), со што се постигнуваат значителни заштеди во областа на поправка и одржување на машините.



Слика 5 интернет на нештата во индустиријата

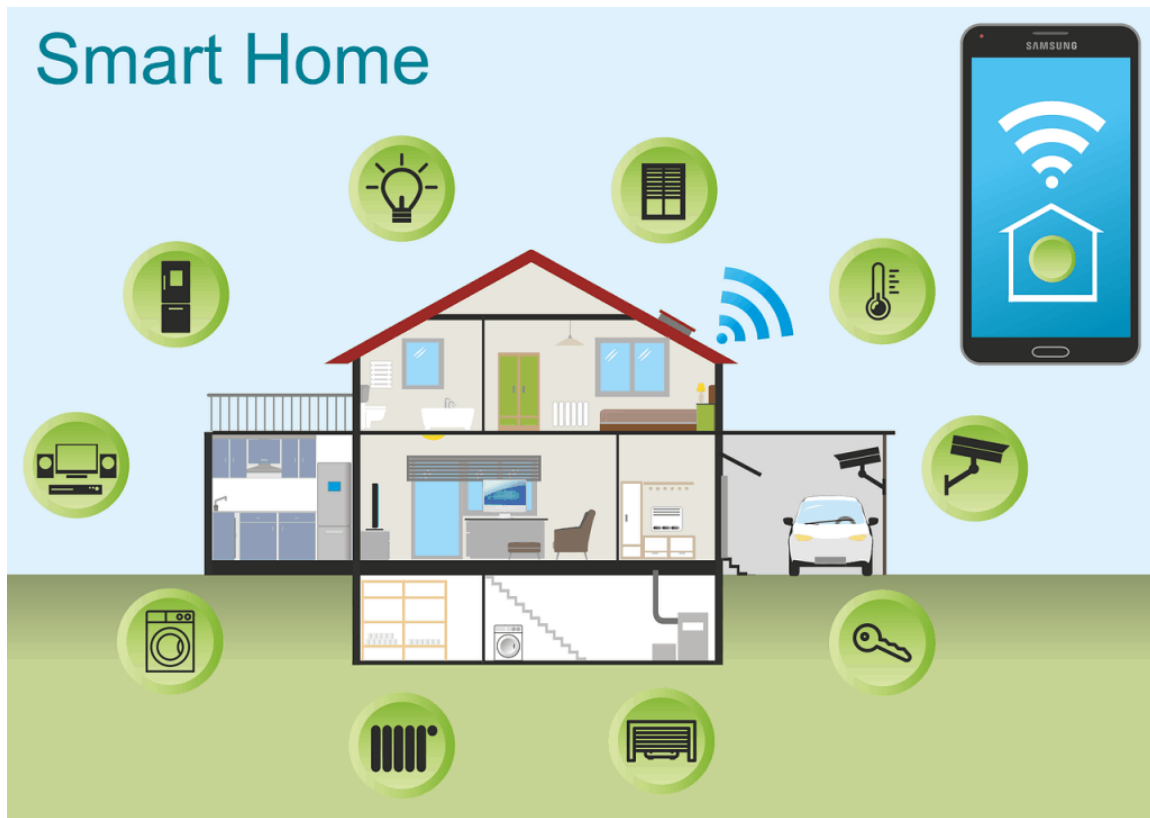
Figure 5 IOT in industry

5. Паметен дом

Подобрувањето на квалитетот на животот на лицата со попреченост и зголемениот процент на стари лица станува сè поважна задача за европското општество. Еден начин да се подобри квалитетот на животот е да се направи животната средина поудобно место за живеење со воведување паметна технологија во паметен дом.

Една од најдобрите дефиниции за технологијата на паметни домови е дека паметниот дом претставува интеграција на технологијата и услугите преку домашно вмрежување за подобар квалитет на живот. Некои други поими поврзани со паметен дом се свесна куќа, променлива куќа и амбиентална интелигенција.

Овие термини се користат за да се нагласи фактот дека домашното опкружување секогаш треба да биде во можност да одговори или да се прилагоди на нејзините различни жители и нивните променливи потреби. На пример, амбиенталната интелигенција е дефинирана како дигитално опкружување кое е чувствително, прилагодливо и реагира на присуството на луѓето. Паметен дом е место на живеење опремено со паметна технологија што овозможува услуги што го подобруваат човечкиот начин на живот, како на пр. безбедноста, енергетската ефикасност, забавата и слично.



Слика 6 пример на паметен дом
Figure 6 example of smart home

Паметен дом е систем што се состои од мрежа за податоци што ги поврзува и интегрира клучните електрични уреди и им овозможува да се контролираат од централен извор [19]-[20].

Паметните домови овозможуваат интеракција помеѓу контролираните елементи и токму тоа го прави системот различен од вообичаените системи за контрола на животната средина. Паметните системи користат електрична енергија, коаксијални кабли, телекомуникациски системи, инфрацрвени, интернет или радиофреквенции или комбинација на неколку други како комуникациски и контролни медиуми. Основната инсталација на паметен дом е лесно програмирана да ги задоволи сите потреби на поединецот, и затоа е многу прилагодена на принципите на дизајнот за секого. Паметните домашни системи вклучуваат

контрола на корисникот, функции за автоматска контрола преку интеракција помеѓу компонентите и врската и преку интеракција со разни медиуми за комуникација.

6. Заштеда на енергија

Голем дел од придобивките од паметниот дом доаѓаат на начини што не се толку очигледни. Паметните домови заштедуваат енергија и топлина, поради што се еколошки и можат да бидат дури и поевтини за живеење [21-22].

Како паметните домови успеваат да заштедат толку многу енергија?

Убавината на начинот на кој паметните домови заштедуваат енергија е нејзината елеганција. Тие не се обидуваат да ја ограничат вашата потрошувачка на енергија ниту пак се обидуваат да ве присилат сами да го сторите тоа. Наместо тоа, тие водат сметка за вашиот животен стил и се обидуваат да најдат начини да не трошите енергија кога не ви е потребна. Тие се дизајнирани да извлечат максимум од достапната енергија. Ако го земете предвид сето тоа, не е изненадување што паметните домови понекогаш можат да заштедат и до 30-40% енергија.

Првиот начин на кој паметните домови заштедуваат енергија е следење на топлината. Старите термостати ретко се способни правилно да водат сметка за топлината во вашиот дом. Паметните домови користат повеќе сензори за да ја измерат точната температура и затоа имаат подобро познавање како да ја загреат. Доколку го комбинираме ова со модерен систем за греење, тогаш лесно ќе го загрееме нашиот дом со изненадувачки мала количина на енергија. Паметните домови заштедуваат енергија со тоа што водат сметка кои простории ги користите и кога ги користите.

Системот ќе следи кои простории ги зафаќате и проверува дали температурата во нив е доволно топла кога ги користите. Домот повеќе нема да се загрева без потреба додека нема никој во него. Ниту пак ќе мора да се чека да се загрее доколку има некој во него.

Одлична карактеристика што треба да се додаде во паметниот дом се соларните панели. Со поставување соларни панели на кровот и поврзување на нив со паметен дом, се осигурува дека нивото на енергија во домот е секогаш

оптимално. Паметните домови прават разлика помеѓу енергијата што ја произведува вашиот дом и енергијата што ја купувате. Тие можат да бидат програмирани да претпочитаат да користат ваша сопствена енергија и да користат само дополнително доколку е потребно. На овој начин, вашиот дом ќе биде самоодржлив, што е секогаш одлично кога станува збор за еко-пријателство. Додавањето соларни панели на кровот е одличен начин да се биде еколошки подобри.

Тие се обидуваат да ги следат нашите животни навики и да го натераат нашиот дом да биде повеќе во склад со нив. Паметниот дом може да ви даде само информации и да се обиде да го оптимизира она што го имате во моментот. Од вас зависи дали ќе му дадете повеќе алатки за употреба и подобро опкружување за да ги користите.

Не може да се зборува за енергетска ефикасност ако немате пристојна изолација. Како по правило, домовите што биле изградени пред 1970-тите немаат добра изолација. Па дури и модерните имаат тенденција да имаат лоша изолација. Значи, она што треба да се направи е да се разгледа каква изолација имате и да ја надградите, ако е можно. Со правилно изолирање на вашиот дом, ќе го направите неговото загревање и ладење многу полесно и ефикасно. Некои згради имаат толку добра изолација и ефикасни соларни панели што произведуваат повеќе енергија отколку што им треба.

LED светлата се одличен начин за заштеда на енергија. Овие светилки користат многу помалку енергија отколку обичните сијалици. На тоа додадете го фактот дека можете лесно да ги поврзете со генератори на соларен погон и лесно ќе забележите дека тие се задолжителни за секоја еко-свесна личност. Тие се особено корисни на надворешните ламби бидејќи се прилично издржливи и можат да траат многу подолго од обичните сијалици. Нивната единствена лоша страна е што тие чинат повеќе од обичните сијалици.

Уште еден фактор за заштедата на енергија е користењето на евтината тарифа на струјата. Многу од луѓе не обрнуваат внимание на користењето на струја во евтината тарифа, но тоа може да донесе голема заштеда на енергија. Во

паметниот дом е вклучена функционалност која ве известува во кој период од денот можете да ја користите евтината струја.

Енергетската ефикасност носи големи придобивки во секојдневниот живот:

- 1) намалување на потребата за увоз на енергија;
- 2) поефикасно греење и ладење;
- 3) заштеда на пари;
- 4) подобра средина;
- 5) намалување на ризикот од разни болести предизвикани од штетни материји, кои се резултат на производство на енергија;
- 6) зголемување на бројот на работни места;

зголемување на употребата на обновливи извори на енергија

7. Вештачка интелигенција

Вештачката интелигенција е област од компјутерските науки чија цел е создавање на машини кои изведуваат функции што кога би се изведувале од луѓе би барале интелигенција.

Во 1943 година, неврофизиологот и кибернетичар Warren S. McCulloch и логичарот Walter Pitts смислиле математички модел за да се обидат да симулираат работа на централниот нервен систем на живите суштества [23]. Така, тие го развија првиот модел на вештачка невронска мрежа (engl. artificial neural network). Својот опис на вештачки неврон како творештво што прима податоци, ги обработува и ги проследува на други неврони, објавено во *The Bulletin of Mathematical Biophysics*.

По првиот модел на неврон, во 1958 година Rosenblatt претстави нов модел на вештачки неврон наречен *perception* [24]. Perceptron првично беше замислен како уред, но како што еволуираше, тој напредуваше во модел кој овозможува категоризација на влезните сигнали. Набргу по ова, Widrow и Hoff во 1960 година претставија нов модел на вештачки неврони наречен АДАЛИН [25]. АДАЛИН е еднослоен линеарен вештачки неврон базиран на невроните McCulloch и Pitts и користи метод за учење со најмал квадрат. (engl. least mean square).

Во меѓувреме во 1956г. John McCarthy на својата прва академска конференција ја претставува идејата за машините да функционираат како човечки суштества. John McCarthy, вклучувајќи ги и Marvin Minsky, Allen Newell и Herbert Simon, станале лидери во истражувачката областа на ВИ за многу наредни декади. Тие и нивните студенти напишале програми, кои за повеќето луѓе, биле вистинско чудо: компјутерите решавале теоретски проблеми во алгебра, докажувајќи логички теореми, зборувајќи на англиски јазик.

Тезата дали машините имаат иста способност да размислуваат и учат сами по себе е воведена од математичарот Алан Тјуринг. Тој преку серии од тестирања кои подоцна се наречени и (Тјуринг тестови) докажува дека е возможно машините да размислуваат и учат како човекот. Доказ за ова е победата на DeepBlue над шаховскиот велемајстор Гари Каспаров во 1997 година. Подоцна се појавуваа уште многу компјутерски софтвери кои успеваат да ги надминат луѓето со што го докажуваат напредокот на ВИ.

Областа на вештачката интелигенција е комбинација на биолошко знаење за човечкиот мозок и неговиот нервен систем и компјутерските науки и технологија. Тоа е дел од науката што се развива секој ден и открива нешто ново. Од самото воведување на пазарот таа беше причината за брзата промена во технолошката и деловната област.

Вештачката интелигенција е насекаде околу нас, таа е дел од нашите животи. Личните асистенти како што се Cortana на Microsoft [26], Siri [27] на Apple кои препознаваат гласови и извршуваат наредби преку нашите уреди па се до останатите апликации и алгоритми за длабоко учење се дел од нашето секојдневие. [28] На Интернационалното Првенство во една од најпопуларните видео игри Dota 2, OpenAI ги победи најдобрите светски играчи во оваа игра во натпревари 1 на 1. Подоцна во 2019 година успеа да ги победи шампионите во ова натпреварување во натпревари 5 на 5 (5 луѓе против 5 компјутери) и стана првиот софтвер кој успеал да победи Светски Шампион во видео игра.

8. Користени технологии

За да се постигне целта на овој проект се искористени неколку технологии.

За корисничкиот интерфејс преку кој можеме да ја следиме состојбата и да ги контролираме уредите и сензорите е искористена рамката за дизајн на апликации Angular, од причина што подржува објектно-ориентиран пристап за развој и е побрз, поскалабилен и помодерен.

За контрола, пресметка и анализа на податоците од сензорите е изработена апликација во Spring-boot кој е избран бидејќи се заснова на Java, која е еден од најпопуларните програмски јазици во светот. Покрај тоа, Spring Boot може да ви помогне брзо да изградите какви било апликации без да се грижите за нивната безбедна и правилна конфигурација.

Системот за предикција на температурата е изработен во програмскиот јазик Пајтон. Овој јазик е избран бидејќи е еден од најдобрите програмски јазици во развојот на софтвер, управување со инфраструктура и анализа на податоци. Тој е лесен за учење и синтаксата е изработена да биде читлива и едноставна.

Контролата на сензорите и уредите е обезбедена преку NodeMCU микроконтролер кој е главна алка во системот.

Како посредник на пораки помеѓу апликациите е избран ActiveMq бидејќи е водич во групата на брокери. Поддржува голема количина на клиенти и протоколи. Вклучува библиотеки во Java, C, C ++, C #, Руби, Перл, Пајтон, PHP.

За комуникација помеѓу апликациите е избран протколот Mqtt бидејќи е брз и енергетски ефикасен и е незаменлива алка од IoT индустријата.

Уште еден начин преку кој комуницираат апликациите е преку WebSockets. Тоа е постојана врска помеѓу клиентот и серверот.

8.1. Софтвер

8.1.1. Angular

Angular е рамка за дизајн на апликации и развојна платформа за создавање ефикасни и софистицирани апликации на една страница со употреба на HTML и TypeScript. Angular е напишан во TypeScript [29].

Архитектурата на angular апликација се потпира на одредени фундаментални концепти.

Основните градежни блокови се NgModules, кои обезбедуваат компилациски контекст за компонентите. NgModules собира поврзан код во функционални множества. Angular апликацијата е дефинирана од множество на NgModules.

Апликацијата секогаш има барем еден главен модул што овозможува подигнување и обично има многу повеќе модули за одлики.

- Компонентите дефинираат прегледи, кои се множество на елементи на екранот што Angular може да ги избере и да ги измени според логиката и податоците на програмата.
- Компонентите користат сервиси, кои обезбедуваат специфична функционалност, која не е директно поврзана со прегледите. Сервис провајдерите може да се инјектираат во компоненти како зависности, што го прави вашиот код модуларен повеќекратно употреблив и ефикасен.

8.1.2. Spring boot

Spring Boot е микро-рамка со отворен код што се користи за изградба на Spring апликации со помош на микросервиси. Создаден од Pivotal Software, Inc., може да се користи и за традиционални додавања на WAR и за самостојни Java апликации. Spring Boot им помага на развивачите веднаш да започнат со кодирање без да губат време на подготовка и конфигурирање на околината. За разлика од другите Java-рамки, тој обезбедува флексибилни конфигурации на XML, моќно обработувачко процесирање, трансакции со бази на податоци, лесен проток на работа, заедно со широк спектар на алатки за развој. [30]

Општо функционира со обезбедување на автоматка конфигурација со што може веднаш да се започне со било кој нов Spring проект.

8.1.2.1. Зошто Spring boot е популарен

Како прво, се заснова на Java, која е еден од најпопуларните програмски јазици во светот. Покрај тоа, Spring Boot може да ви помогне брзо да изградите какви било апликации без да се грижите за нивната безбедна и правилна конфигурација.

Предности на Spring Boot:

- Го намалува времето поминато на развој и ја зголемува вкупната ефикасност на тимот за развој.
- Помага при автоматско конфигурирање на сите компоненти
- Го олеснува создавањето и тестирањето на апликации базирани на Java, обезбедувајќи поставување за единечно и интеграциско тестирање.
- Помага да се избегне целата рачна работа за пишување на дупликат код , прибелешки и сложени конфигурации на XML.
- Доаѓа со вградени HTTP сервери како Jetty и Tomcat за тестирање веб-апликации
- Овозможува многу приклучоци што програмерите можат да ги користат за непречено и лесно работење со вградени и во-мемориски бази на податоци.

8.1.3. Python

Програмскиот јазик Пајтон е релативно нов програмски јазик кој се појавува во 1991 година. Од самиот почеток, се сметаше дека Пајтон ја пополнува празнината и дека е начин за пишување скрипти кои ги автоматизираат работите, или брзо ги прототипира апликациите што треба да се имплементираат на еден или повеќе други јазици. Меѓутоа со текот на годините Пајтон се покажа како еден од најдобрите програмски јазици во развојот на софтвер, управување со инфраструктура и анализа на податоци. Тој веќе не е јазик кој ја пополнува празнината туку е водечка сила во развојот на веб апликациите, аналитиката (Big Data) и машинското учење [31].

Успехот на Пајтон се заснова на неколку предности кои им ги нуди на почетниците и на експертите. Пајтон е лесен за учење, бројот на функции во самиот јазик е

скромнен, па за да се направи првата програма е потребно релативно малку вложено време и труд. Синтаксата е изработена да биде читлива и едноставна.

Многу од големите компании го користат Пајтон во своите проекти:

1. YouTube - „Пајтон е доволно брз за нашата страница и ни овозможува да произведуваме нови функционалности што може да се одржуваат во рекордни времиња, со минимум програмери“, - Куонг До, архитект на софтвер, YouTube.com. [32]
2. Google - „Пајтон е важен дел од Гугл од самиот почеток и останува така како што системот расте и се развива. Денес голем дел од инженерите на Гугл го користат Пајтон и бараме повеќе луѓе со вештини на овој јазик. - Питер Норвиг, директор за квалитет на пребарување во Google, Inc. [32]

8.1.4. ActiveMQ

ActiveMQ е посредник на пораки со отворен од, кој има целосна имплементација на JMS клиент. Дава опции за големи претпријатија кои овозможуваат комуникација со повеќе клиенти и сервери истовремено. Поддржува голема количина на клиенти и протоколи. Вклучува библиотеки во Java, C, C ++, C #, Руби, Перл, Пајтон, PHP.

Во моментот, најновата верзија е 5.16.0. Подолу се дадени некои од многуте напредни опции што ActiveMQ ги нуди покрај самата спецификација JMS, со што ги прави пораките што е можно подобри и лесни за употреба. [34]

8.1.4.1. Група на пораки

Група на пораки (енг. Message group) е состои од сите пораки кои што имаат иста вредност на полето (JMSXGroundID). ActiveMQ се грижи сите пораки од иста група да бидат испратени на еден JMS корисник (сопственик на групата на пораки) се додека е активен. Доколку тој корисник престанува со работа, тогаш ќе биде избран нов корисник. Оваа употреба на пораки е пожелна доколку сакате да го балансирате товарот на потрошувачите на некоја дестинација. Она што можеби не е очигледно на прв поглед е фактот дека сите пораки што стигнуваат до потрошувачот и сопственикот на групата пораки ќе дојдат точно по редоследот по кој се испратени. Овој метод го гарантира редоследот на обработка на пораките на истиот потрошувач. [35]

8.1.4.2. Виртуелни дестинации

Виртуелните дестинации ви овозможуваат да креирате логични дестинации кои многу лесно се мапираат на една или повеќе физички дестинации. Идејата за виртуелни дестинации е да обезбедат функционалност обезбедена од трајни потрошувачи, но на начин што избегнува некои проблеми со балансирање на оптоварување што тие ги контролираат. Имено, при исклучување на издржлив производител, ние не сакаме неговата работа да зависи од него, сакаме да има можност некој да продолжи да ја работи неговата работа.[36]

Да речеме дека постои тема `VirtualTopic.Topic`. Потрошувачот А може да се приклучи на неа како и на било која друга тема. Меѓутоа, потрошувачите Б и Ц можат да се спојат на редот `ConsumeQueue.VirtualTopic` (каде `ConsumeQueue` е име на редот кое е заедничко за А и Б), па преку него најизменично ќе ги примаат пораките од оригиналната тема. На овој начин, редот, односно темата, ќе се конзумира дури и по престанокот на еден од потрошувачите (Б или Ц). Во овој пример, редицата `ConsumeQueue` ќе се создаде автоматски.

8.1.5. Mqtt

Протоколот MQTT (Transport Queue Telemetry Transport) е развиен од Енди Стенфорд Кларк (IBM) и Ален Нипер (Еуротек) во 1999 година за да се следи нафтоводот среде пустина [37].

Нивната цел беше да развијат протокол што ќе овозможи ефикасна брзина на пренос на податоци со мала потрошувачка на батерија и пониска цена на целиот систем [34].

Развиениот протокол MQTT со архитектура за објавување и запишување се покажа како полесен и поекономичен од протоколот HTTP ((HyperText Transfer Protocol) со модел на Request – Response.

Моделот објавување-претплати е управуван од настани и им овозможува на клиентите да објавуваат пораки без да се грижат за нивната крајна дестинација. Конечната дестинација на објавената порака ја одредува посредникот MQTT кој врз основа на претплатените клиенти ги испраќа пораките и на тој начин ги ослободува клиентите од испраќање барања за пораки, за разлика од протоколот HTTP каде што клиентите треба да ги побараат потребните информации од серверот.

Протоколот MQTT се карактеризира и со мала големина на фиксно заглавје (fixed header) од само 2 бајти, што дополнително го намалува оптоварувањето на мрежата [36].

Според Степ Николас, протоколот MQTT нуди заштеда на батерија на мобилните уреди од околу 4,1% за еден ден, додека одржува стабилна врска наспроти HTTP протоколот [37]. Иако првичното поврзување на клиентот MQTT со серверот троши повеќе батерија, бидејќи универзалниот идентификатор на клиент е дополнително испратен, одржувањето на постојана врска при примање и објавување пораки се покажа поекономично за протоколот MQTT бидејќи HTTP врши повеќе поврзувања што многу почесто влијае на траењето на батеријата.

Табела 1 Перформансно тестирање на MQTT и HTTPS

Table 1 Performance test on MQTT and HTTPS

| Карактеристики | | 3G | | WiFi | |
|------------------|------------------------------|---------|---------|----------|---------|
| | | HTTPS | MQTT | HTTPS | MQTT |
| Примени пораки | Пораки/час | 1708 | 160278 | 3628 | 263314 |
| | Процент на батерија / час | 18.43% | 16.13% | 3.45% | 4.23% |
| | Процент на батерија / пораки | 0.01709 | 0.00010 | 0.000095 | 0.00002 |
| Испратени пораки | Пораки/час | 1926 | 21685 | 5229 | 23184 |
| | Процент на батерија / час | 18.79% | 17.80% | 5.44% | 3.66% |
| | Процент на батерија / пораки | 0.00975 | 0.00082 | 0.00104 | 0.00016 |

Очигледно е дека не може да се измери точната потрошувачка на енергија за сесиите MQTT и HTTP на лаптоп, па затоа е искористено Raspberry [43].

Во табела 2 е прикажани резултати од тест за потрошувачка на енергија за Mqtt и HTTP протокол. Кодот прави бесконечен број на MQTT или HTTP сесии.

За тестот е искористена 5V 2000mAh батерија поврзана за Raspberry.

Податоци кои се измерени:

1. Број на сесии додека батеријата не се ипразни
2. Колку време трае тестот
3. Просечна употреба на процесорот

Табела 2 Анализа на потрошувачка на енергија помеѓу Rest API и MQTT

Table 2 Analysis of power consumption between API and MQTT for rest

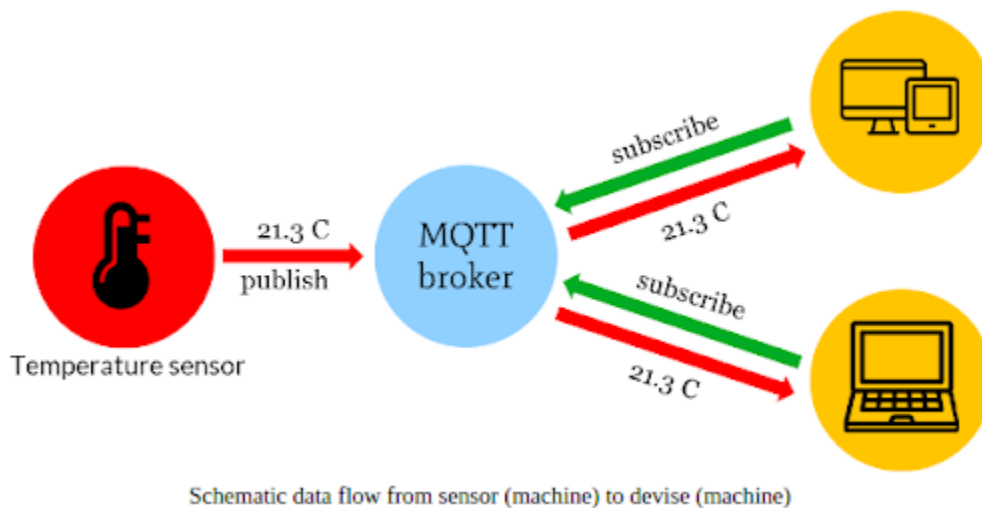
| Потрошувачка на енергија на Raspberry | 3Г модем интернет конекција | | Етернет интернет конекција | |
|---------------------------------------|-----------------------------|-----------------------|----------------------------|-----------------------|
| | Rest API via curl | Mosquitto MQTT client | Rest API via curl | Mosquitto MQTT client |
| Број на сесии | 9243 | 11728 | 20964 | 22864 |
| Време на тестот | 2ч. 39мин. | 2ч. 50мин. | 5ч. 53мин. | 6ч. 21мин. |
| mAh за сесија | 0.2164 | 0.17 | 0.0954 | 0.089 |
| Пораки во секунда | 0.97 | 1.15 | 0.99 | 0.98 |
| Користење на процесорот | 64 | 60 | 69 | 64 |

Дури и на краток рок, MQTT е до 22% енергетски поефикасен и 15% побрз од HTTP и не зависи од тоа дали типот на врска е 3G или етернет.

8.1.5.1. Модел објави – претплати (publish-subscribe)

Како што споменавме во претходниот дел, комуникацијата помеѓу клиентите во протоколот MQTT се постигнува со формулар за (publish – subscribe) објавување-претплата со централна улога на посредник MQTT кој ги следи претплатите на клиентите и делегира пораки во зависност од претплатите.

Темата (topic) ја претставува главната информација за посредникот за насочување на пораката до крајните клиенти. Од друга страна, кога клиентите се претплаќаат, тие мора да ја наведат темата на која што сакаат да се претплатат за да може посредникот да знае како да ги доставува сите пораки чија тема одговара на темите на кои клиентот се претплатил.



Слика 7 Шематски приказ на протокот на податоци од сензорот до одредена машина

Figure 7 Schematic data flow from sensor to machine

Кај овој начин на комуникација клиентите не треба да се познават меѓусебно и можат да комуницираат само преку одбрани теми преку што се избегнува зависноста помеѓу прозведувачите (producers) и потрошувачите (consumers).

Постојат клиенти што праќаат пораки кои се наречени објавувачи (publishers) и клиенти што примаат пораки кои се наречени претплатници (subscribers).

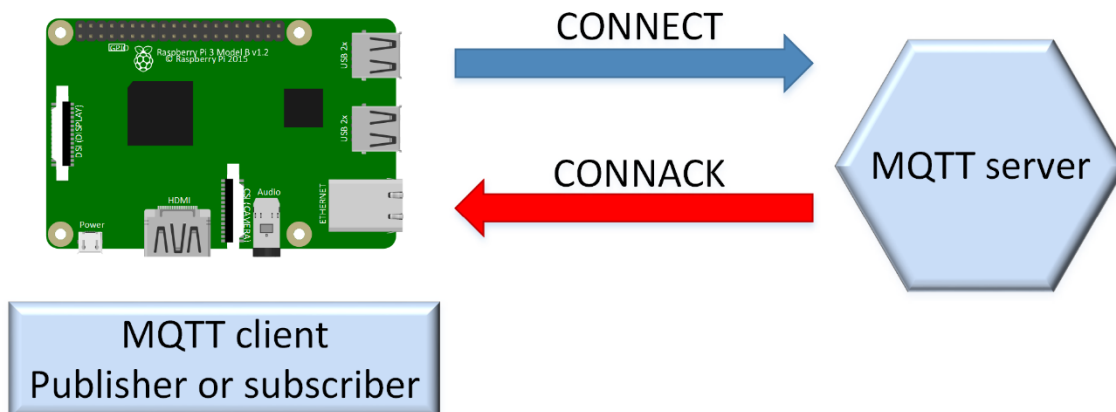
Одвојувањето на испраќачот од примачот може да се подели на три димензии:

1. Просторна димензија – објавувачот и претплатникот не треба да се познават еден со друг
2. Временска димензија – нема потреба да бидат активни во исто време
3. Синхронизациска димензија – операциите на објавувачот и претплатникот не се сопираат при објавување и примање пораки (клиентите работат ансихроно независен еден од друг)

8.1.5.2. MQTT конекција

Почетокот на секоја комуникација помеѓу клиентите е нивно поврзување со посредникот кој е центар на комуникација и е одговорен за примање на сите пораки, нивно филтрирање и проследување до сите запишани клиенти. При поврзување со брокерот, клиентите мора да ги наведат адресата на брокерот и портата 1883 која е резервирана исклучиво за MQTT комуникација. Бидејќи MQTT се заснова на транспортен или мрежен слој на TCP / IP, секој клиент, при поврзување со посредникот, отвора постојана TCP врска над која се отвора сесија на MQTT за објавување и примање пораки [43].

Самата врска е иницирана од клиентот кој испраќа Connect порака до брокерот MQTT на која тој одговара со порака ConnACK со статусен код што дава информации за успешноста на врската со клиентот.



Слика 8 Приказ на Connect / ConnACK

Figure 8 Representation of Connect / ConnACK

Доколку врската се загуби на кој било начин, брокерот MQTT може да ги зачува влезните пораки и да ги препрати на клиентот кога тој повторно ќе се поврзе. Кога се испраќа порака за поврзување, клиентот мора да дефинира единствен идентификатор (`clientId`), со кој брокерот ги разликува клиентите. Покрај уникатниот идентификатор, клиентот може да постави дополнителни знамиња кои ги прошируваат можностите на протоколот MQTT. [39]

Знамиња кои можат опционално да се постават се:

1. (`clean session flag`) - Знаме за чиста сесија – доколку е поставено ова знаме брокерот MQTT ќе ги избрише сите информации поврзани со претходната сесија. Во случај знамето да не е поставено, ќе се воспостави постојана сесија и брокерот ќе ги зачува сите претплати и пропуштени пораки и покрај исклучувањето.
2. Знамиња за корисничко име и лозинка – овозможуваат авторизација на корисникот преку поставување на корисничко име и лозинка.
3. (`will flag`) - Доколку ова знаме е поставено, брокерот ја зачувува пораката како (`will message`) и при губење на конекција ја испраќа пораката до клиентите кај кои е поставено (`will flag`).

Клиентите кои го имаат поставено ова знаме можат да постават период од траење на врската кој претставува најголем временски интервал помеѓу испраќањето на еден и почетокот на испраќање на друг MQTT пакет по кој конекцијата ќе биде прекината. Со цел да се избегне истекот на овој интервал, одговорноста на клиентот е да испрати едноставни пакети `Pingreq` само за да ја одржи врската жива. Брокерот на `Pingreq` пакетите одговара со `Pingresp` пакети кои потврдуваат дека тие функционираат нормално. Откако MQTT клиентот ќе испрати `Connect` пакет до посредникот, треба да добие `Connack` пакет за врската успешно да се воспостави.

`Connack` одговорот на посредникот содржи две информации:

1. (`session present flag`) - кога е поставено ова знаме, тоа значи дека веќе постои сесија помеѓу клиентот и посредникот, т.е. дека клиентот и брокерот веќе

имале запаметена интеракција пред тековната врска. Кога знамето не е поставено, се создава нова празна сесија. Доколку клиентот постави чисто знаме на сесијата за време на врската, ова знаме никогаш нема да биде поставено за врските на тој клиент со брокерот.

2. Повратен код - покажува дали врската е прифатена од посредникот, доколку е прифатена повратната вредност е 0.

Табела 3 Повратен код и објаснување

Table 3 Code and description

| Повратен код | Објаснување |
|--------------|--|
| 0 | Прифатена врска |
| 1 | Одбиена врска, неприфатлива верзија на протокол |
| 2 | Одбиена врска, идентификаторот е отфрлен |
| 3 | Одбиена врска, серверот е недостапен |
| 4 | Одбиена врска, погрешно корисничко име или лозинка |
| 5 | Одбиена врска, ако не сте овластен |

8.1.5.3. Објави и претплати

Кога клиентот кој претходно успешно воспоставил конекција одлучи да испрати порака мора покрај податоците кои сака да ги објави да наведе и тема (topic) на кој сака да ја испрати пораката. На овој начин MQTT посредникот ќе знае на кои клиенти да ја проследи добиената порака. Податоците кои се во пораката (payload) се испраќаат на посредникот во бајти бидејќи MQTT овозможува објава на податоци во било кој облик (бинарни, текстуални податоци, JSON или XML) [44]. Објавувањето на пораката вклучува испраќање на пакет за објава (publish packet) до посредникот.



Слика 9 „Publish“-порака за објава
 Figure 9 Representation of publish message

Овој пакет ги вклучува следниве атрибути:

1. Име на темата - едноставна низа карактери структурирани хиерархиски со црта напред како разграничувач што ги претставува главните информации до медијаторот за понатамошно рутирање на пораката.
2. QoS ниво на квалитет на сервисите (Quality of Service Level) – претставува нивото на гаранција за испорака на објавените пораки. Овој атрибут со два бита може да прима 3 вредности:
 - Вредност 0 (at-most-once delivery) - најниско ниво на квалитет на сервис каде пораката се доставува во согласност со можностите на протоколот за долниот слој (протокол TCP). Пораката се испраќа само еднаш без повратни информации од примачот и без зачувување и повторно испраќање од испраќачот.
 - Вредност 1 (at-least-once delivery) – ова гарантира дека пораката ќе биде успешно испратена најмалку една до примачот, но постои можност да биде испратена и повеќе пати. Испраќачот ја спрема пораката, а потоа ја испраќа до посредникот кој ја проследува до сите примачи и враќа потврда за испратените пораки во облик (Puback)

пакет кој содржи идентификатор (packetId). Откако ќе добие потврда дека пораката е успешно испратена испраќачот ја брише пораката интерно.

- Вредност 2 (exactly-once delivery) – ова е најспората опција. Гарантира дека пораката ќе биде испратена само еднаш. Испраќачот ја припрема пораката пред испраќање, откако ќе ја спреми ја испраќа до примачите, и за возврат добива потврда во пакет (Pubrec) со идентификациски пакет од испратената порака. По ова тој испраќа (Pubrel) пакет на кој посредникот одговара со (Pubcomb) пакет, и двата пакети содржат само идентификациски пакет од испратената порака. За да се осигура дека пораката ќе биде испратена само еднаш посредникот го памти идентификаторот на пораката се додека не прими (Pubrel) пакет по кој испраќачот нема повторно да ја испрати пораката.
3. Знаме за памтење на пораките (retain flag) – Ова знаме одредува дали посредникот ќе ја запамти пораката за одредена тема. Доколку ова знаме е поставено тогаш пораката ќе биде испратена на клиентите кои ќе се претплатат на таа тема.
 4. Податоци (payload) – содржина на пораката која се испраќа. Може да биде во облик на слика, текст со кој било метод на кодирање, шифрирани или бинарни податоци.
 5. Идентификатор на пакетот (packetId) – двабитен податок кој ја означува пораката во комуникацијата помеѓу клиентот и посредникот.
 6. Знаме за препознавање дуплицирани пораки (DUP flag) – Доколку е поставено ова знаме, тоа значи дека пораката повторно е испратена бидејќи испраќачот нема примено потврда за успешноста на објавување на пораката до посредникот. Ова знаме е релевантно само за пакетите на објава кои имаат QoS поголем од 0.

Откако пакетот на објава со пораката и темата стигне до посредникот, тој на основа на темата и запаметените претплатници ја проследува пораката. За да знае на кој да ја испрати пораката посредникот мора да им овозможи на клиентите кои сакаат да ја примаат пораката можност за претплата на темата која ги интересира.

Клиентите се претплатуваат на темата со испраќање на пакет за претплата (subscribe packet) до посредникот.



Слика 10 Приказ на „Subscribe“ порака

Figure 10 Representation of Subscribe message

Пакетот за претплата содржи идентификатор на пакетот и листа од претплати. Идентификаторот на пакетот го одредува пакетот во комуникацијата помеѓу посредникот и претплатникот, а листата на претплати содржи листа од теми на кои клиентите сакаат да се претплатат. Овој пакет мора да содржи најмалку една тема со QoS. Откако клиентот ќе го испрати пакетот за претплата со листа од теми, посредникот мора да врати со одговор во облик (Subpack) пакет кој ја потврдува успешноста на пораката.



Слика 11 Suback порака за потврда

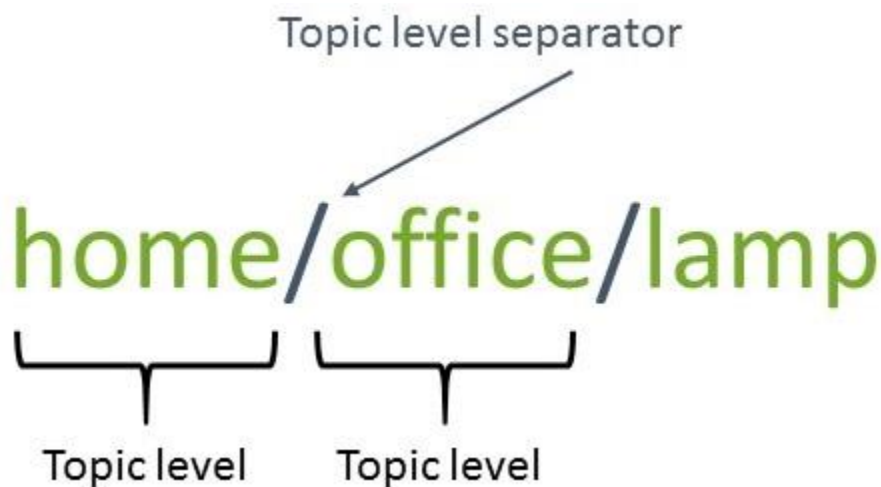
Figure 11 Suback acknowledgement message

Овој пакет го содржи истиот идентификатор на пакетот како и пакетот за претплата и листата од повратни кодови за секоја тема на која клиентот сака да се пријави.

Повратниот код може да добие 4 вредности: 0,1 и 2 за успешно воспоставена претплата со сервис 0,1 и 2 и вредност 128 доколку претплата е неуспешна. По успешно воспоставената претплата клиентот ќе ги добие сите пораки од посредникот каде темата на претплатникот е иста со темата на посредникот. Доколку клиентот не сака повеќе да прима пораки од одредена тема може едноставно да се откаже о испраќање на (unsubscribe packet) кој содржи идентификатор и листа на теми од кои клиентот сака да се одјави. Кога посредникот ќе добие порака за одјавување, тој одговара со Unsuback пакет со исти идентификатор за да ја потврди успешноста на одјавување.

8.1.5.4. MQTT тема

Темата е низа од знакови во UTF-8 формат кој посредникот го користи за филтрирање и проследување на пораките до клиентите. Темата се состои од една или повеќе делови кои се хиерархички одвоени со коса црта.



Слика 12 Mqtt тема

Figure 12 Mqtt topic

Секоја тема мора да содржи најмалку еден карактер, а имињата на нивоата можат да содржат празно место и се чувствителни на големи и мали букви. Клиентот не треба да креира тема пред да ја објави пораката затоа што брокерот автоматски ќе ја создаде доколку не постои во времето на испраќање на пораката.

8.1.5.4.1. Wildcards

Кога клиентот се претплатува на топик тој може да се претплати на бараната тема или да искористи (wildcard) за да се претплати на повеќе теми истовремено. Wildcard може да се користат само за претплатување на теми, но не и за објавување на пораки. Постојат два вида на wildcards:

1. (single-level) – како што наведува самото име оваа (wildcard) заменува едно ниво од темата. Знакот ' + ' претставува едно ниво од темата.

На пример:

Во претплатата на темата /smarthome/+/temperature спаѓаат и темите:

- ✓ /smarthome/sensors/temperature
 - ✓ /smarthome/outdoor/temperature
 - ✓ /smarthome/indoor/temperature
2. (multi-level) – (' # ') симболот претставува замена за било кое ниво од хиерархиски подредени теми. Доколку клиентот е претплати на темата /smarthome/sensors/# тој ќе добие известувања за темата smarthome/sensors и сите негови подтеми како што се smarthome/sensors/temperature, smarthome/sensors/humidity, smarthome/sensors/lights итн. Треба да се напомене дека симболот (' # ') мора да биде последен во името на темата и мора да биде одделен со сепаратор /.

8.1.5.5. Клиент

За имплементација на MQTT во апликација, потребно е да се вклучи програмска подршка во изработката на проектот. Програмската подршка е достапна во облик на клиентска библиотека која нуди интерфејс за поврзување на клиентите и посредниците (broker) и за користење на специфични функции на протоколот MQTT кои беа опишани во претходното поглавје. Листа од сите клиентски библиотеки за програмскиот јазик Јава може да се најде на официјалната страница за MQTT протоколот. [44]

Во овој труд се користи Eclipse PaHo [45] поради најдобрата документација и најголемиот број на имплементирани примери.

8.1.6. WebSockets

Веб сокет е постојана врска помеѓу клиентот и серверот. Веб сокетите обезбедуваат двонасочен, full-duplex комуникациски канал кој работи преку HTTP преку TCP / IP конекција

Во основа, протоколот WebSocket го олеснува пренесувањето на пораката помеѓу клиентот и серверот. [49]

Идејата за WebSockets се појавува поради ограничувањата на технологијата заснована на HTTP. Со HTTP, клиентот бара ресурс, а серверот одговара со бараните податоци. HTTP е строго еднонасочен протокол - клиентот мора да ги побара сите податоци испратени од серверот до клиентот. [50]

8.2. Хардвер

8.2.1. NodeMCU

NodeMCU е бесплатен фирмвер (Системски софтвер запишан во ROM, PROM или EPROM меморијата или на некое друго место во електрониката на компјутерот. Корисникот овој софтвер може да го чита, но не може да го промени.) и комплет за развој кој се користи за изработка на IoT производи [46]-[50]. Вклучува фирмвер што работи на ESP8266 Wi-Fi SoC од Espressif Systems и хардвер кој се базира на модулот ESP-12. Фирмверот го користи јазикот за кодирање Lua. Се базира на проектот eLua и е изграден на Espressif Non-OS SDK за ESP8266.

MCU претставува MicroController - што означува дека е компјутер на еден чип. Микроконтролерот содржи еден или повеќе процесори (процесорски јадра) заедно со меморија и програмирачки периферни уреди за влез / излез. Тие се користат за автоматизација на контролата на моторот во автомобилот, медицински уреди за, далечински управувачи, канцелариски машини, апарати, електрични алати, играчки итн.

GPIO

General-purpose input/output (GPIO) е стандарден интерфејс што се користи за поврзување на микроконтролери со други електронски уреди. GPIO може да се користат за прифаќање на влезни или испраќање на сигнали за излез, кои ги одредува самиот корисник.

Една важна работа што треба да се забележи во врска со ESP8266 е дека бројот GPIO не одговара на етикетата на плочата. На пример, D0 одговара на GPIO16 и D1 одговара на GPIO5.

Табела 4 Соодносот помеѓу етикети и број на GPIO

Table 4 Correspondence between labels and GPIO number

| Лабела | GPIO | Влез | Излез | Забелешка |
|--------|---------------|-------------------------------------|------------------------------|---|
| D0 | GPIO16 | Без можност за “interrupt” функција | Нема поддршка за PWM или I2C | Под висок напон на стартување, се користи за будење од „длабок сон“ |
| D1 | GPIO5 | OK | OK | Често се користи како SCL (I2C) |
| D2 | GPIO4 | OK | OK | Често се користи како SDA (I2C) |
| D3 | GPIO0 | Pulled up | OK | Поврзано со копчето за флешување |
| D4 | GPIO2 | Pulled up | OK | Под висок напон на стартување, |

| | | | | |
|----|---------------|----------------------------|--------|-------------------------------------|
| | | | | поврзано со “LED“ на плочата |
| D5 | GPIO14 | OK | OK | SPI (SCLK) |
| D6 | GPIO12 | OK | OK | SPI (MISO) |
| D7 | GPIO13 | OK | OK | SPI (MOSI) |
| D8 | GPIO15 | Поврзано со заземјување | OK | SPI (CS) |
| RX | GPIO3 | OK | RX пин | Под висок напон на стартување |
| TX | GPIO1 | TX пин | OK | Под висок напон на стартување |
| A0 | ADC0 | Аналоген влез | / | |

Следната табела ја покажува соодносот помеѓу етикетите на плочата и бројот на GPIO, како и кои пинови се најдобри за употреба во проектите и со кои треба да се внимава.

Пиновите означени со зелена боја се во ред за употреба. Оние што се обележани со жолто, се во ред за употреба, но треба да се обрне внимание затоа што може да имаат неочекувано однесување главно при подигање. Пиновите означени со црвена боја не се препорачуваат да се користат како влезови или излези.

GPIO поврзани со чипот за флешување

GPIO6 до GPIO11 обично се поврзани со флеш чипот во плочите ESP8266. Ова значи дека овие пинови не се препорачуваат за користење.

Пинови кои се користат при бутирање

ESP8266 може да се спречи да се подигне ако некои пинови се повлечат НИСКО или ВИСОКО. Следната листа ја покажува состојбата на следниве пинови:

- GPIO16: висок при подигање
- GPIO0: подигањето паѓа доколку се поврзе на НИСКО
- GPIO2: висок при подигање, подигањето паѓа доколку се поврзе на НИСКО
- GPIO15: подигањето паѓа доколку се поврзе на ВИСОКО
- GPIO3: висок при подигање
- GPIO1: висок при подигање, подигањето паѓа доколку се поврзе на НИСКО
- GPIO10: висок при подигање
- GPIO9: висок при подигање

Пинови ВИСОКИ при подигање

Постојат одредени пинови кои на излез имаат 3,3V сигнал кога ESP8266 ќе се подигне. Ова може да биде проблематично ако имате релеи или други периферни уреди поврзани со тие GPIO. Следните GPIO имаат ВИСОК сигнал при подигнување:

- GPIO16
- GPIO3
- GPIO1
- GPIO10
- GPIO9

Дополнително, другите GPIO, освен GPIO5 и GPIO4, при подигнување можат да добијат нисконапонски сигнал, што може да биде проблематично доколку се поврзани со транзистори или релеи.

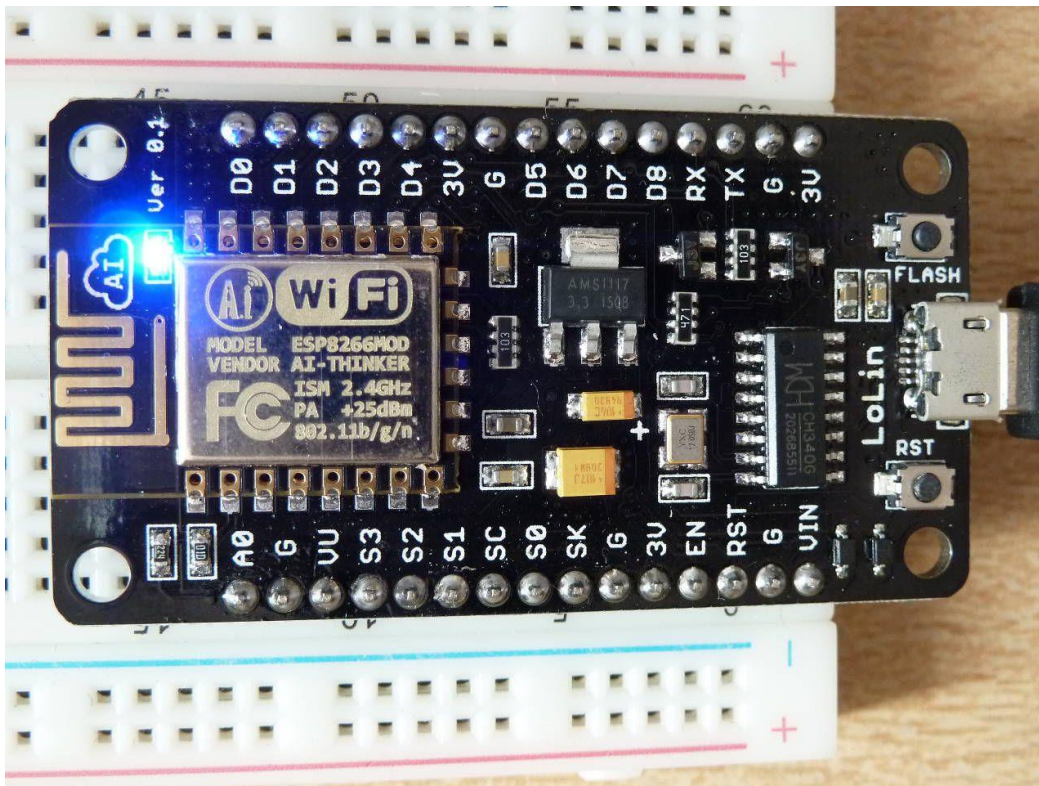
Аналоген влез

ESP8266 поддржува аналогно читање на само еден GPIO. Тој GPIO се нарекува ADC0 и обично е означен на плочата како A0.

Максималниот влезен напон на пинот ADC0 е 0 до 1V доколку се користи ESP8266. Доколку се користи табла за развој како комплет ESP8266 12-E NodeMCU, опсегот на влез на напон е од 0 до 3,3V бидејќи овие плочи содржат внатрешен разделувач на напонот.

LED на плочата

Повеќето од плочи за развој ESP8266 имаат вграден ЛЕД. Овој LED обично е поврзан со GPIO2.



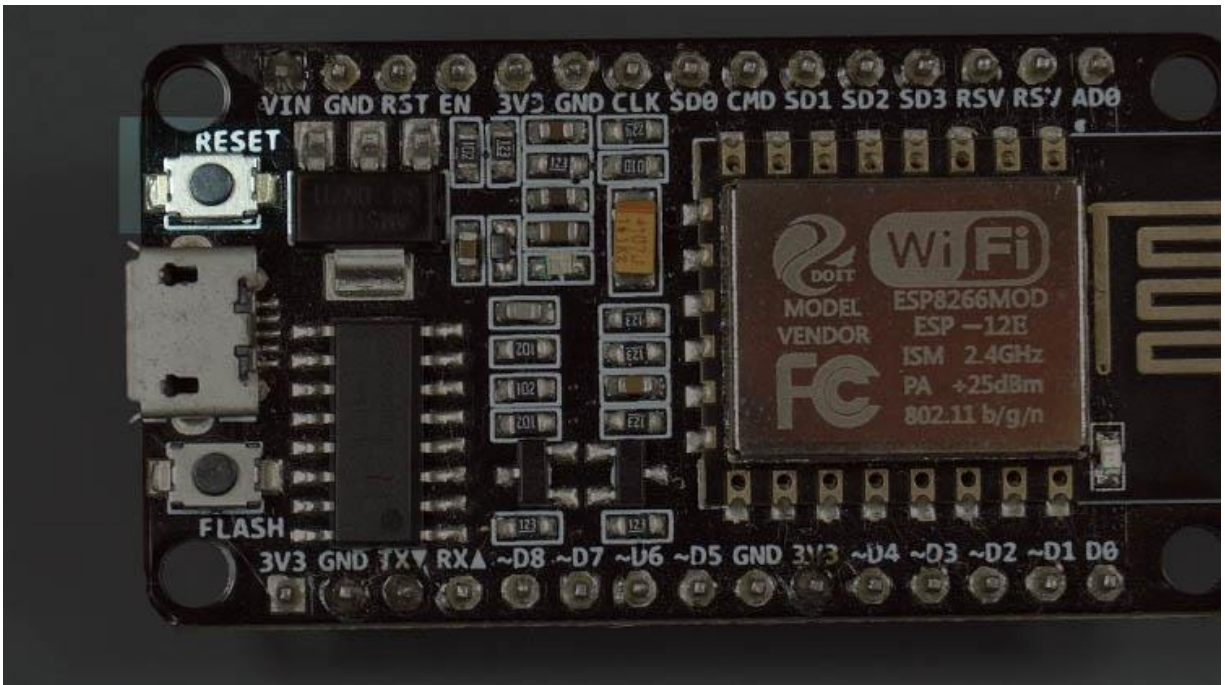
Слика 13 LED на плочата

Figure 13 LED on the board

Овој LED е поврзан со pull-down резистор, па доколку испратиме висок сигнал тогаш LED светилката ќе се исклучи.

RST пин

Кога RST пинот ќе добие низок сигнал тогаш ESP8266 се ресетира. Ова е исто како да сме го искористиле копчето за ресет кое се наоѓа на плочата.



Слика 14 RST пин

Figure 14 RST pin

GPIO16

GPIO16 може да се користи за будење на ESP8266 од “длабок сон”. За да се разбуди ESP8266 од длабок сон, GPIO16 треба да биде поврзан со RST пинот.

I2C (Inter-Integrated Circuit)

ESP8266 нема хардверски пинови I2C, но може да се имплементира во софтвер. Така, можете да користите какви било GPIO како I2C. Обично, следниве GPIO се користат како I2C пинови:

1. GPIO5: SCL
2. GPIO4: SDA

SPI (Serial Peripheral Interface)

Пиновите што се користат како SPI во ESP8266 се:

1. GPIO12: MISO
2. GPIO13: MOSI

3. GPIO14: SCLK
4. GPIO15: CS

PWM пинови

ESP8266 дозволува софтвер PWM во сите влезни и излезни пинови GPIO0 до GPIO16. PWM сигналите на ESP8266 имаат 10-битна резолуција.

Interrupt пинови

ESP8266 поддржува прекини во секое GPIO, освен GPIO16.

9. Опис на проектот

9.1. Енергетска аналитика

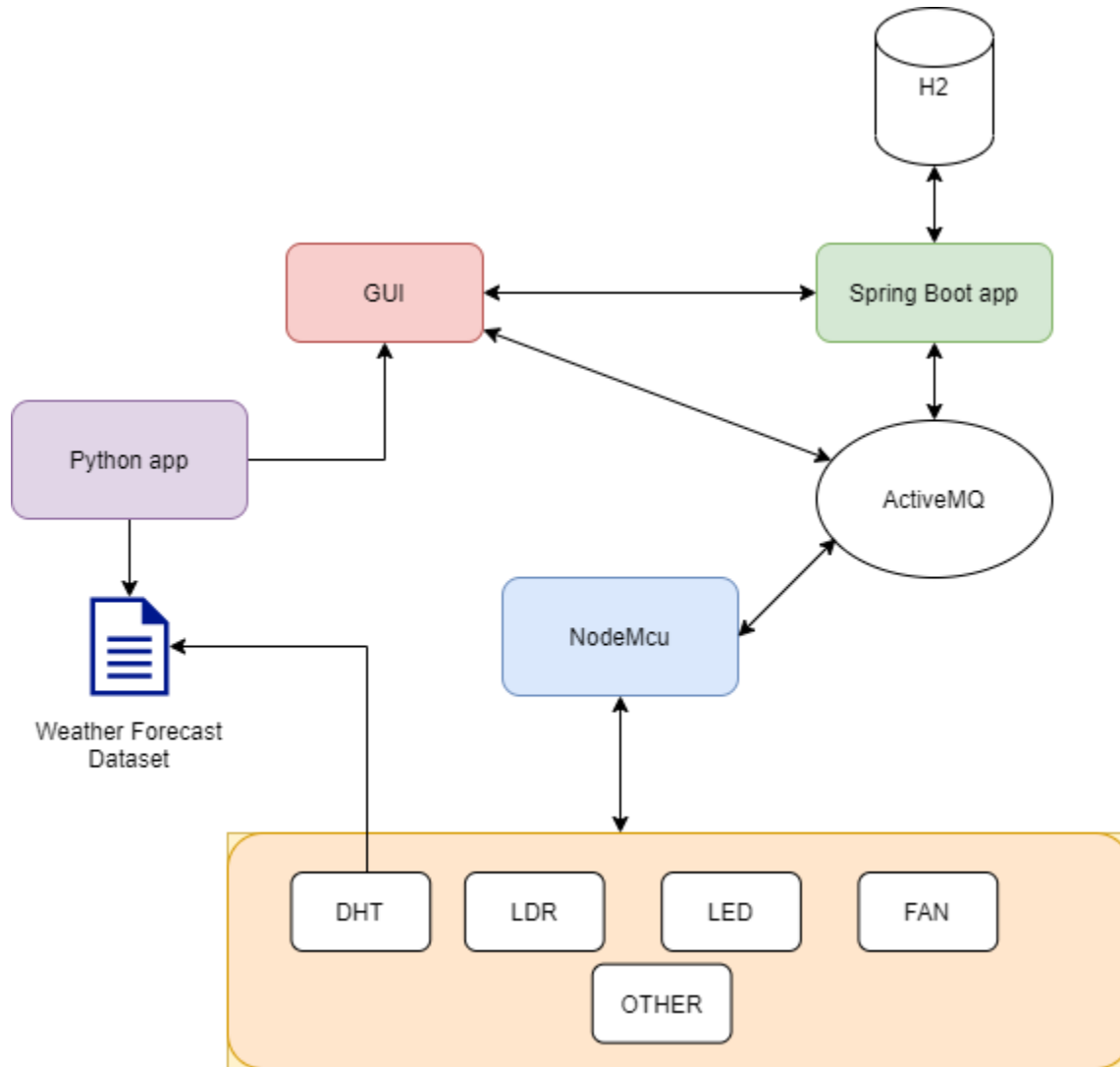
Во овој труд се опфатени неколку случаи за енергетска ефикасност во системот:

1. Доколку има доволно светлина во домот или во дворот системот автоматски ќе ги исклучи светилките во домот или дворот.
2. Доколку системот детектира дека прозорот е отворен а во исто време е вклучен системот за греење/ладење, тогаш системот автоматски ќе го исклучи системот за греење/ладење.
3. Доколку сензорот за детекција на дожд детектира дожд тогаш системот автоматски ги затвора прозорците.
4. Систем преку електронска пошта известува за времето кога е евтина тарифа на струјата.

Во случај да се изврши некој од горенаведените случаи и системот превземе акција тогаш се активира режимот за енергетска ефикасност кој го пресметува времетраењето на активноста на режимот и пресметува колку би била потрошената енергија за дадениот уред и ја брои во заштедената за тој ден. Овие податоци се зачувуваат во базата на податоци и се прикажуваат на графичкиот интерфејс. Цените кои се прикажани на графичкиот интерфејс се пресметани со помножување на заштедената енергија и цената на kW/h која според регулаторната комисија за енергетика и водни услуги на Република Северна Македонија која изнесува 4.44 денари за kW/h.[51]

9.2 Архитектура

Овој проект е составен од повеќе компоненти кои комуницираат меѓусебно со цел да се постигне лесно одржлив и енергетско ефикасен паметен дом.



Слика 15 Архитектура на проектор

Figure 15 Architecture of the project

На сликава е прикажана архитектурата на проектот. NodeMcu е поврзан со повеќе сензори и ги контролира истите сензори преку добивање на команди односно пораки на одредени теми на ActiveMq. Сензорот за температура и влажност своите мерења ги запишува во Excel документ. Апликацијата изработена во Пајтон го користи тој документ како датасет од податоци врз кои прави предикција на температурата за наредните денови и ги прави податоците достапни за

апликацијата изработена во Angular која ги прикажува добиените податоци. Преку Spring boot апликацијата можеме да ги контролираме и да ги превземеме вредностите од сензорите преку ActiveMq каде што NodeMcu ги испраќа на одредена тема. Од превземаните податоци се прави аналитика за заштедата на енергија. Податоците се складираат во h2 база. Во spring boot апликацијата имаме Rest API преку кое податоците стануваат достапни за Angular апликацијата. Angular апликацијата ги превзема податоците од останатите апликации и ги прикажува. Во неа имаме и дел од каде што можеме да ги контролираме уредите и да ја следиме нивната состојба. Исто така имаме и дел за контрола на сензори преку гласовни команди.

9.3. Изработка

За изработката на паметен дом креирани се неколку апликации со користење на следните технологии:

1. Spring boot
2. Angular
3. NodeMCU (Arduino IDE)
4. Python

9.3.1 Софтвер

9.3.1.1. Spring boot

Апликацијата креирана со Spring boot е наменета за контрола, пресметка и анализа на податоците од сензорите. Во оваа апликација ја имаме состојбата на секој сензор и уред зачувани во базата на податоци h2. Пресметките и аналитиката исто така се зачувуваат во h2 базата на податоци. Апликацијата комуницира со GUI преку REST API и WebSocket и комуницира со NodeMcu преку протокол Mqtt преку ActiveMq.

9.3.1.1.1. H2

За складирање на податоците се користи релационата база на податоци H2.

H2 ("Hypersonic 2") е релационен систем за управување со базата на податоци напишан на Java кој ги подржува стандардните SQL API и JDBC API, како и PostgreSQL ODBC. H2 се карактеризира со брзи перформанси и ниско барање на

меморија. H2 може да биде вграден во Java апликација или да се користи како сервер за база на податоци.

Може да биде користена како in-memory или податоците да се чуваат на диск. Кога станува збор за безбедноста H2 исто така има силни безбедносни карактеристики.

```
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>compile</scope>
</dependency>
```

9.3.1.1.2. Mqtt

Протоколот Mqtt е искористен за комуникација помеѓу Spring boot апликацијата и NodeMCU. Како функционира протколот Mqtt е објаснето во точка 8.1.5.

Како клиент е користена библиотеката Eclipse Paho.

```
<dependency>
  <groupId>org.eclipse.paho</groupId>
  <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
  <version>1.2.0</version>
</dependency>
```

Конекција:

```
public static IMqttClient mqttConnection() throws MqttException,
MqttSecurityException {
    String publisherId = UUID.randomUUID().toString();
    IMqttClient client = new MqttClient("tcp://0.0.0.0:1883", publisherId);
    MqttConnectOptions options = new MqttConnectOptions();
    options.setAutomaticReconnect(true);
    options.setCleanSession(true);
    options.setConnectionTimeout(10);
    client.connect(options);
    return client;
}
```

За користење на mqtt клиентот имаме искреирано mqtt controller во кој имаме методи:

1. sendToTopic (служи за испраќање на порака на одредена тема)

```
@GetMapping(value = "/sendToTopic")
private ResponseEntity<String> sendToTopic(@RequestParam String topic, @RequestParam String message)
    throws MqttPersistenceException, MqttException {
    IMqttClient client = mqttService.mqttConnection();
    MqttMessage msg = mqttService.createMessage(message);
    client.publish(topic, msg);
    return new ResponseEntity<>("SENT", HttpStatus.OK);
}
```

Слика 16 Контролер за испраќање пораки на тема
Figure 16 Controller for sending messages to topic

2. subscribeToTopic (претплатување на одредена тема)

```
@GetMapping(value = "/subscribeToTopic")
private ResponseEntity<String> subscribeToTopic(@RequestParam String topic)
    throws MqttPersistenceException, MqttException {
    IMqttClient client = mqttService.mqttConnection();
    mqttService.mqttSubscriber(client, topic);
    return new ResponseEntity<>("Subscribed", HttpStatus.OK);
}
```

Слика 17 Контролер за зачленување на тема
Figure 17 Controller for subscribing on topic

9.3.1.1.3. Rest API

За комуникација помеѓу Angular апликацијата и Spring boot апликацијата се искреирани неколку контролери со кои ќе се овозможи пристап до податоците за Angular апликацијата.

1. Device-controller (служи за креирање, превземање, променување и бришење на уредите)

| device-controller Device Controller | |
|--|---|
| POST | <code>/createDevice</code> createDevice |
| GET | <code>/getAllDevices</code> getAllDevices |
| GET | <code>/getDeviceById</code> getDeviceById |
| GET | <code>/hardReset</code> deleteAll |
| GET | <code>/updateDeviceById</code> updateDeviceById |

Слика 18 Контролер за уредите

Figure 18 Device Controller

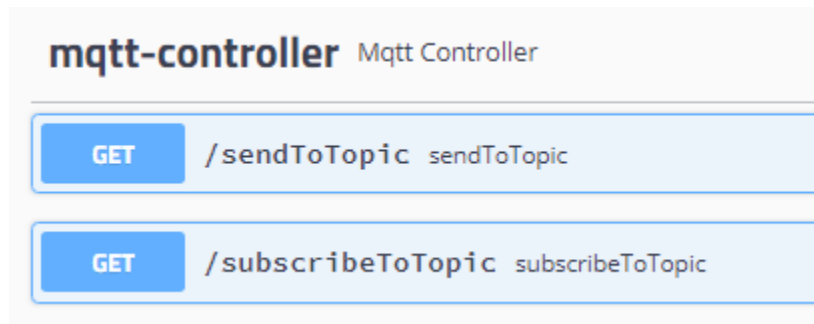
2. Energy-consumption-controller (служи за стартување и стопирање на заштедата на енергија исто и за превемање на податоците од заштедената енергија)

| energy-consumption-controller Energy Consumption Controller | |
|--|---|
| GET | <code>/getEnergyConsumptionByPeriod</code> getEnergyConsumptionFromTo |
| GET | <code>/startEnergyConsumption</code> startEnergyConsumption |
| GET | <code>/stopEnergyConsumption</code> stopEnergyConsumption |

Слика 19 Контролер за заштеда на енергијата

Figure 19 Energy consumption controller

3. Mqtt-controller (служи за испраќање пораки до одредена тема и претплатување)



Слика 20 Mqtt контролер

Figure 20 Mqtt controller

9.3.1.2. Front end Angular

Со цел да се изгради корисничкиот интерфејс на Smart Home платформата, се користи Angular, која ќе биде основната технологија на апликацијата. Од причина што подржува објектно-ориентиран пристап за развој, тој е побрз, поскалабилен и помодерен. HTML5, CSS3 со Java Script се основните блокови на апликацијата. За графички приказ на податоците во форма на графикони и графици, избрана е библиотеката chart.js. За дизајн, стил и целосен изглед за апликацијата се користи Bootstrap.

Апликацијата ги добива податоците со комуникација со Spring boot апликацијата преку Rest API и Web Sockets и истите ги прикажува.

На почетната страница е прикажана табела од состојбата на секој од уредите дали е активен или е исклучен.

Исто така има и копче за испраќање на гласовни команди.

Менито се состои од:

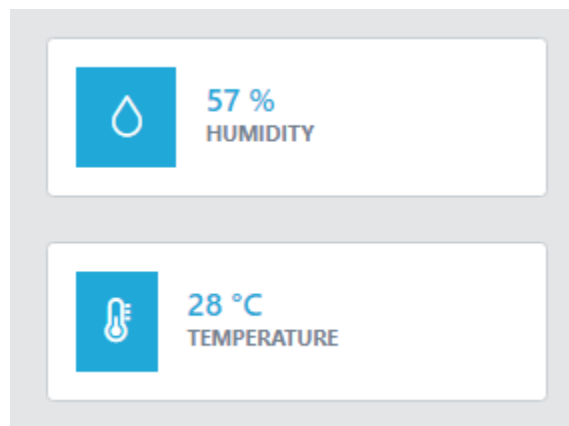
1. Модул за контрола на уредите каде што можеме да ги контролираме нашите уреди. На пр. Кога корисникот ќе ја вклучи сијалицата, апликацијата испраќа порака до Spring Boot апликацијата кој подоцна преку mqtt протокол ја испраќа командата до NodeMcu



Слика 21 Копчиња за контрола на уредите

Figure 21 Buttons for controlling devices

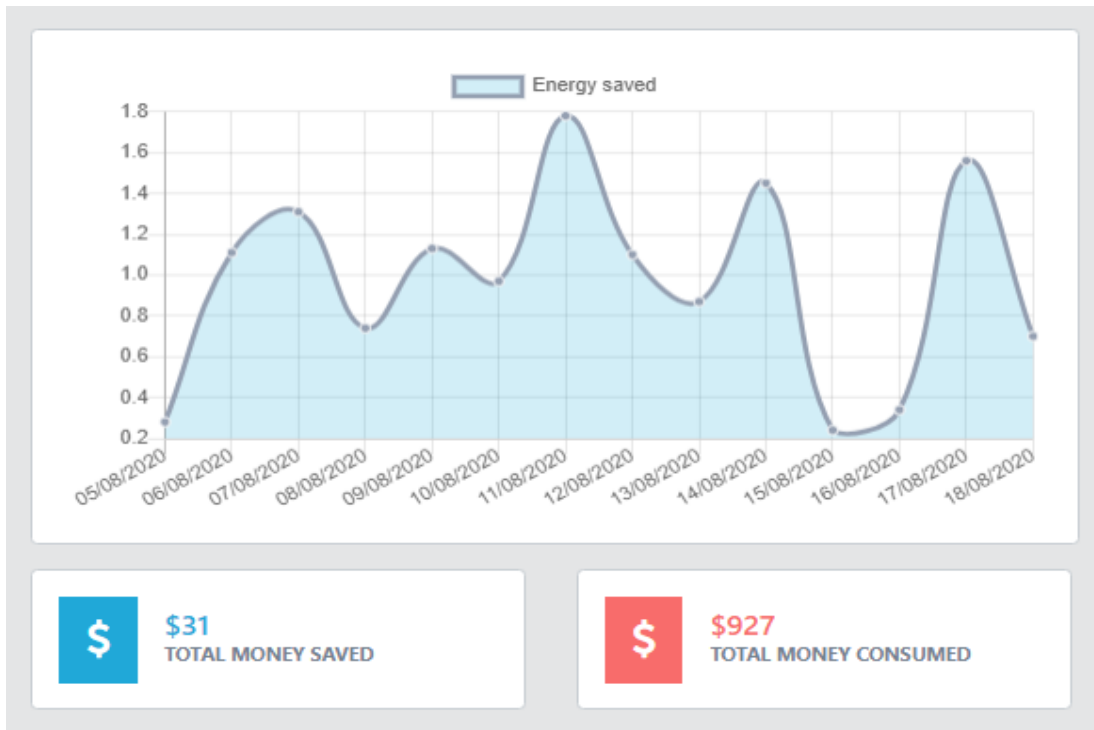
2. Модул за следење на состојбата на сензорите кој ги прикажува податоците превземени од сензорите преку mqtt протокол.



Слика 22 Приказ на моменталната температура и влажност

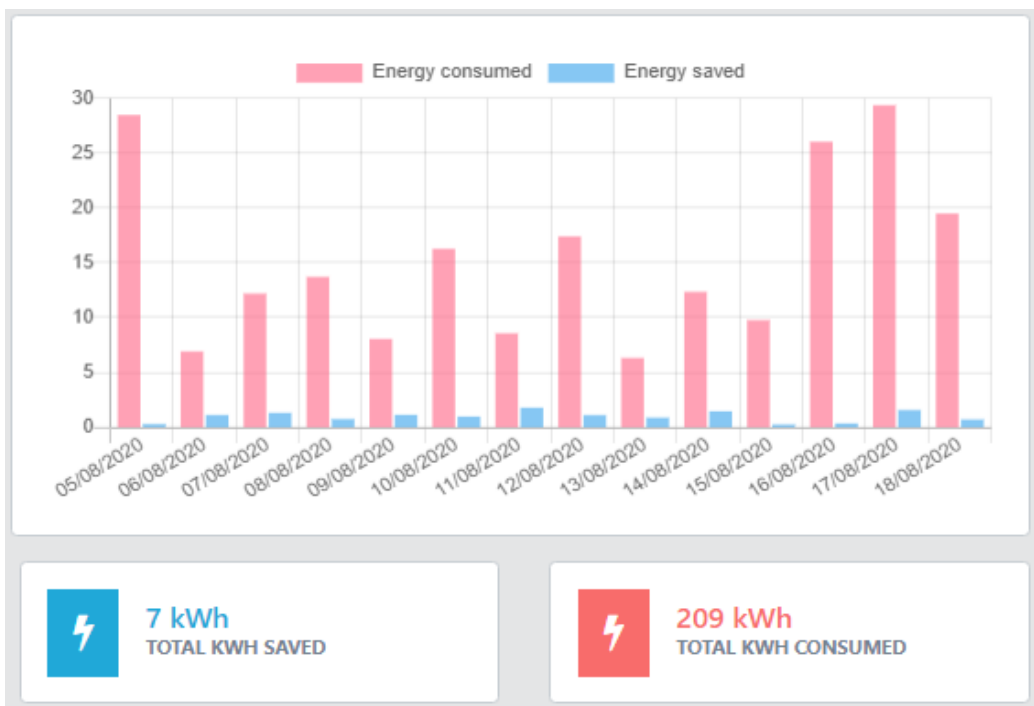
Figure 22 Representation of the current humidity and temperature values

3. Модул за енергетска ефикасност каде можеме да одбереме датум за кој сакаме да видиме два графици. Едниот ни покажува колку енергија и пари имаме потрошено во дадениот период, а другиот ни покажува колку енергија и пари имаме заштедено во дадениот период.



Слика 23 Приказ на резултатите од системот за енергетска ефикасност

Figure 23 Overview of the results of the system for energy efficiency



Слика 24 Приказ на резултатите од системот за енергетска ефикасност

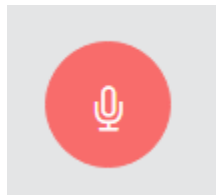
Figure 24 Overview of the results of the system for energy efficiency

9.3.1.2.1. Гласовни команди

Контролата на уредите преку гласовни команди е една од компонентите во паметниот дом.

Google има препознавање говор вградено во прелистувачот Chrome. Apple и другите производители на Android имаат гласовна контрола вградена во нивните мобилни телефони.

Сега, благодарение на annyang може да се додаде гласовна контрола на нашата веб-страница. Annyang е мала библиотека на JavaScript.



Слика 25 Копче за гласовни команди

Figure 25 Button for voice commands

Во оваа апликација имаме употребено гласовни команди за палење и гасење на сијалица и за отварање и затварање на гаражата.

9.3.1.3. Програмирање на NodeMCU

Апликацијата за контрола и следење на уредите е напишана во Arduino IDE. Ардуино интегрирано развојно опкружување (IDE) е крос-платформска апликација (за Windows, macOS, Linux) што е напишана користејќи функции од C и C ++. Се користи за пишување и пренесување на програми на Arduino компатибилни плочи. Arduino IDE ги поддржува јазиците C и C ++ користејќи посебни правила за структурирање на кодот.

Како библиотека за mqtt протокол се користи "EspMQTTClient.h".

9.3.1.4. Систем за предикција на температура со примена на LSTM невронска мрежа

Како дел од магистерската работа креиран е систем кој врз основа на податоците добиени од сензорот за температура прави предикција на температурата за

следните 7 дена. Апликацијата со која се реализира системот е изработена во програмскиот јазик Пајтон и користи LSTM невронска мрежа.

9.3.1.4.1 LSTM

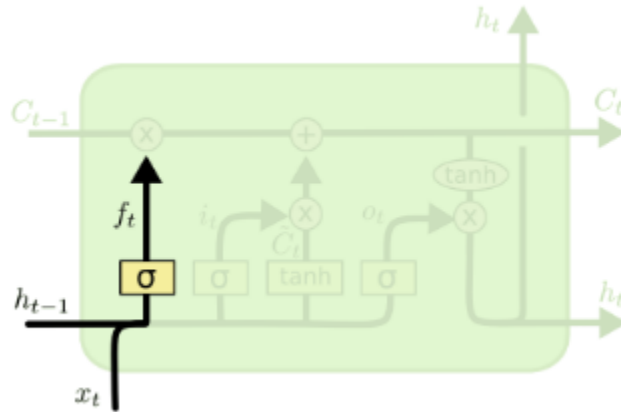
Долгорочната краткотрајна меморија (LSTM-Long short-term memory) е вештачка рекурентна невронска мрежа (RNN) што се користи во областа на длабоко учење. За разлика од стандардните еднонасочни (feedforward) невронски мрежи, мрежата LSTM има повратни врски кои им даваат на невроните можност да донесуваат одлуки засновани не само на тековните, туку и на претходните вредности. Не само што може да обработува индивидуални податоци (како што се слики), туку и цели секвенци на податоци (како што се говор или видео). LSTM невронските мрежи се воведени од Hochreiter & Schmidhuber (1997) и станале популарни за проекти како што се препознавање ракопис или препознавање говор, и погодни се за предвидување на временски серии. [52]

Основниот градежен блок на мрежата LSTM е ќелијата LSTM. Главната идеја на ќелијата LSTM е да ја измени внатрешната состојба. Состојбата претставува збир на информации во рамките на ќелијата. Ќелијата LSTM има можност да додава и одзема информации од состојбата преку три контролери (обично се нарекуваат порти) на протокот на информации во рамките на единицата LSTM:

1. Порти за заборавање
2. Влезни порти
3. Излезни порти

Првиот чекор во LSTM е да одлучиме кои информации ќе ги исфрлиме од состојбата на ќелијата. Оваа одлука е донесена од сигмоиден слој наречен слој за заборав. Во овој слој се пресметува h_{t-1} (излезен вектор) и x_t (влезен вектор) и како резултат се добива број помеѓу 0 и 1 за секој број во ќелијата C_{t-1} . Доколку се добие 1 претставува „целосно задржи го ова“, додека 0 претставува „целосно ослободи се од ова“.

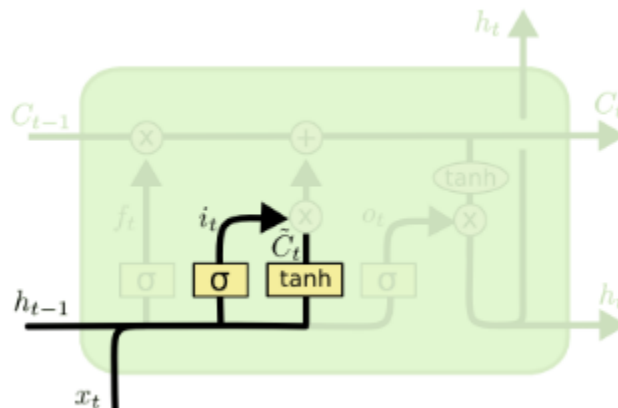
Слојот за заборав се состои од параметри W_f, b_f , каде f претставува заборав.



Слика 26 Слој за заборав
Figure 26 Forget gate layer

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

Следниот чекор е да се одлучи кои нови информации ќе ги зачуваме во ќелијата. Ова има два дела. Прво, сигмоидниот слој наречен „влезен слој на портата“ одлучува кои вредности ќе ги ажурираме. Следно, тан слојот создава вектор на нови вредности на кандидати, \tilde{C}_t , што може да се додаде на состојбата.



Слика 27 Влезен слој и тан слој
Figure 27 Input and tan layer

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

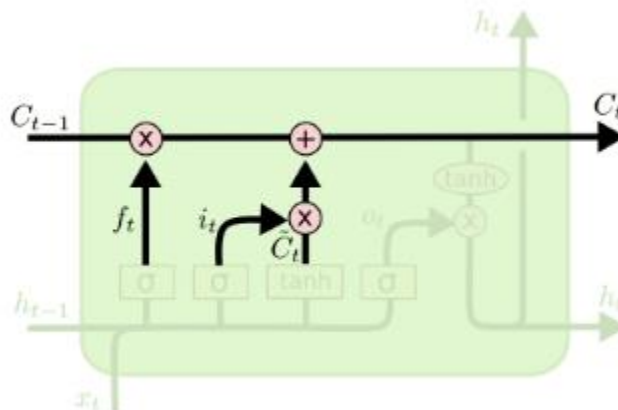
Влезниот слој се состои од параметри W_i, b_i , каде i претставува влез.

Мемориската ќелија ги зачувува информациите за скриените единици низ временските чекори. Се состои од параметри W_c, b_c , каде c претставува ќелија

Потоа се ажурира старата состојба на ќелијата, C_{t-1} , во нова состојба C_t .

Со помножување на старата состојба со f_t се забораваат вредностите за кои претходно имаме одлучено да се забораваат.

Потоа го додаваме $I_t * \tilde{C}_t$. Ова се новите вредности на кандидатите, намалени според тоа колку е одлучено да се ажурира секоја вредност.



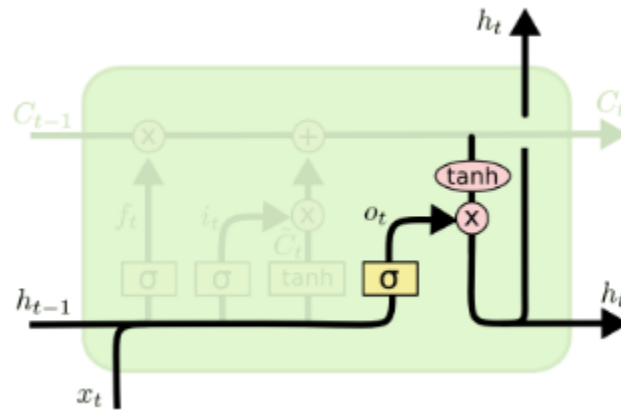
Слика 28 Ажурирање на нова состојба

Figure 28 Updating the new state

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Како последен чекор е излезниот слој. Овој излез ќе се заснова на состојбата на ќелијата, но ќе биде филтрирана верзија. Прво, се користи сигмоиден слој кој одлучува кои делови од состојбата на ќелијата ќе преминат кон излез. Потоа се користи тан (да се притиснат вредностите да бидат помеѓу -1 и 1) и да се помножи

со излезот на сигмоидната порта, така што на излез ќе се добијат посакуваните вредности.



Слика 29 Излезен слој

Figure 29 Output layer

Излезната порта дозволува каква информација ќе излезе на екранот и што ќе се пропагира напред како дел од новата скриена состојба. Излезниот слој се состои од параметри W_o, b_o , каде o претставува излез.

$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Поедноставно, ќелијата LSTM може да биде претставена на следниов начин. Влезните порти контролираат како новата вредност влегува во ќелијата, портата за забораване контролира колку време вредноста останува во ќелијата, а излезната порта го контролира степенот до кој се користи вредноста во ќелијата за пресметување на активирањето на излезот на единицата.

LSTM мрежите се погодни за класифицирање, обработка и правење предвидувања врз основа на податоците од временските серии, бидејќи може да има одложувања

на непознато времетраење помеѓу важните настани во временските серии. Една од предностите на LSTM во однос на RNN е токму релативната нечувствителност на должината на интервалот помеѓу овие настани. LSTM мрежите се развиени за решавање на проблемот со градиентното спуштање каде градиентите на грешката исчезнуваат експоненцијално брзо со големината на интервалот меѓу важните настани, што можат да се сретнат при обука на традиционалните RNN мрежи.[54]

Во теорија, класичните RNN можат да следат произволни долгорочни зависимости во влезните низи. Проблемот со RNN е кога тренираме со употреба на (back-propagation), градиентите што се размножуваат може да „исчезнат“ (т.е. може да имаат тенденција на нула) или да „експлодираат“ (може да имаат тенденција до бесконечност). RNN кои користат единици на LSTM делумно го решаваат проблемот со исчезнување на градиентот, бидејќи единиците LSTM овозможуваат градиентите да останат непроменети. Сепак, мрежите LSTM сè уште можат да страдаат од проблемот со експлодирање на градиентите.

9.3.1.4.2 Имплементација на системот за предвидување на температура
Апликацијата која прави предвидување на временската прогноза за наредните 7 дена е изработена во Пајтон.

Се користи база на податоци од температура која се полни преку сензорот за мерење на температура (DHT11) и на основа на тие податоци прави предикција за наредните 7 дена.

Прв чекор во проектот е читањето на базата со податоци и креирањето на DataFrame од тие податоци.

Подоцна со помош на библиотеката MinMaxScaler ги скалираме податоците да бидат помеѓу 0 и 1.

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
df1=scaler.fit_transform(np.array(df1).reshape(-1,1))
```

Слика 30 Скалирање на податоците

Figure 30 Scaling the data

```
print(df1)
[[0.50697507]
 [0.50508505]
 [0.50544505]
 ...
 [0.71055711]
 [0.70218702]
 [0.68463685]]
```

Слика 31 Печатење на податоците

Figure 31 Printing the data

Затоа што имаме само една база со податоци ќе ја поделиме на два дела од кои 35% ќе бидат податоци за тестот а 65% ќе бидат податоците за тренирање.

Бидејќи пресметката на предикцијата трае премногу долго, како пример е земена предикција која се базира на податоците од претходните 100 дена.

9.3.1.4.2.1. Keras

Керас е API на невронската мрежа, напишан во Python и работи врз TensorFlow, CNTK или Theano.[55-56]

Керас овозможува:

1. Лесно и брзо прототипирање (преку лесна употреба, модуларност и приспособливост).
2. Овозможува поддршка за создавање на конвулуциски и рекурентни невронски мрежи, како и нивна комбинација.
3. Може да работи и на процесорот CPU и на графичката GPU

Креирање на модел:

```
model=Sequential()  
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))  
model.add(LSTM(50,return_sequences=True))  
model.add(LSTM(50))  
model.add(Dense(1))  
model.compile(loss='mean_squared_error',optimizer='adam')
```

Слика 32 Креирање на модел

Figure 32 Creation of the model

Основната структура на податоци во рамките на Керас е модел. Моделот претставува начин на организирање на слоеви. Наједноставниот, и оној што се користи за создавање на овој труд, е секвенцијален модел (enɡl. Sequential), кој претставува линеарен куп слоеви. Додавањето на слоеви е едноставно, моделот содржи функција add() која како параметар го прима типот на слојот (LSTM, Dense...). Откако ќе го искреираме моделот се додава функцијата compile() со помош на која можеме да ја дефинираме функцијата за грешка (loss function). Функцијата за грешка претставува функција која настојува да ја оптимизира невронската мрежа (обично ја минимизира), а процесот на оптимизација е начин за прилагодување на тежините на врските во рамките на невронската мрежа. Со функцијата model.summary() можеме да направиме преглед како изгледа нашиот модел.

```
Model: "sequential"  
-----  
Layer (type)                Output Shape                Param #  
-----  
lstm (LSTM)                  (None, 100, 50)            10400  
-----  
lstm_1 (LSTM)                (None, 100, 50)            20200  
-----  
lstm_2 (LSTM)                (None, 50)                  20200  
-----  
dense (Dense)                (None, 1)                   51  
-----  
Total params: 50,851  
Trainable params: 50,851  
Non-trainable params: 0
```

Слика 33 Преглед на моделот

Figure 33 Model overview

Тренирањето на моделот можеме да го започнеме со функцијата `model.fit()`.

```
#Making the model trained using LSTM
model.fit(X_train,y_train,validation_data=(X_test,ytest),epochs=5,batch_size=64,verbose=1)

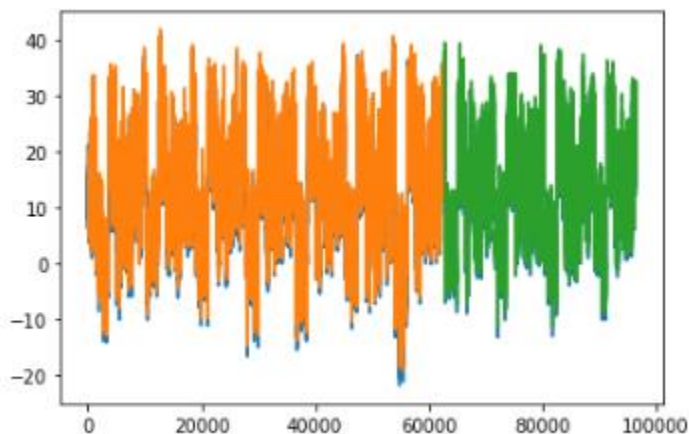
Epoch 1/5
979/979 [=====] - 154s 157ms/step - loss: 0.0031 - val_loss: 0.0013
Epoch 2/5
979/979 [=====] - 162s 165ms/step - loss: 9.6011e-04 - val_loss: 0.0011
Epoch 3/5
979/979 [=====] - 161s 164ms/step - loss: 5.9745e-04 - val_loss: 4.2755e-04
Epoch 4/5
979/979 [=====] - 166s 170ms/step - loss: 5.4890e-04 - val_loss: 3.9998e-04
Epoch 5/5
979/979 [=====] - 161s 164ms/step - loss: 5.2822e-04 - val_loss: 4.6301e-04
```

Слика 34 Тренирање на моделот

Figure 34 Training of the model

Откако ќе заврши тренирањето ја повикуваме функцијата `model.predict()`.

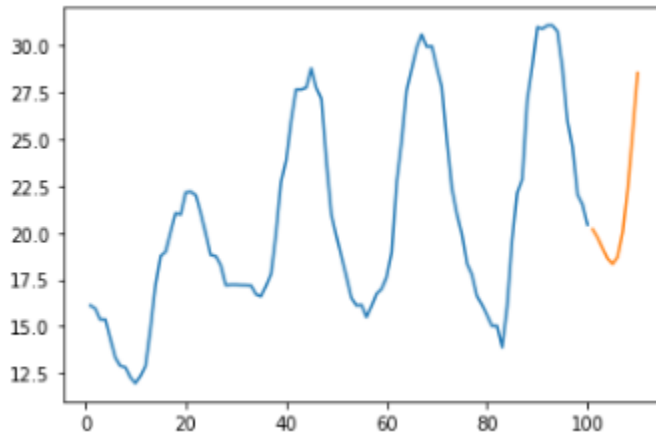
На следнава слика е прикажано како изгледа предикцијата, со портокалова боја се веќе постоечките податоци, а со зелена се предвидените податоци.



Слика 35 Резултати од предикцијата

Figure 35 Results of the prediction

На следнава слика е прикажан график каде со плава боја е исцртано како се движи температурата од базата на податоци, а со портокалово е исцртана предикцијата за следните 7 дена.



Слика 36 График на температурата и предикцијата за наредните 7 дена
 Figure 36 Temperature and prediction chart for the next 7 days

Откако ги имаме предвидените податоци со користење на Flask библиотеката која служи за креирање на веб апликации, креираме endpoint на кој можеме да ги добиеме податоците за предвидените 7 дена кои подоцна ќе бидат прикажани на Angular апликацијата.

```
@app.route('/forecast')
def test():
    getAllAnomalies = scaler.inverse_transform(anomalies.Temperature)
    np_array_to_list = getAllAnomalies.tolist()
    print(np_array_to_list)
    anomly = []
    anomly = [{'Date': d, 'Temperature': t} for d, t in zip(anomalies.index, np_array_to_list)]
    return jsonify(anomly);
```

Слика 37 Креирање на крајна точка
 Figure 37 Creation of the endpoint

9.3.2. Хардвер

9.3.2.1. Сензори и уреди

9.3.2.1.1. Сензор за мерење на температура и влажност (DHT11)

Влажноста и температурата на просторијата се мерат со сензорот DHT11. [57]

Сензорот има повеќе верзии и во зависност од верзијата има различни карактеристики, но сите DHT сензори имаат иста намена. Тој е составен од две компоненти кои се сместени во исто куќиште.

- Термистор – отпорник кој се менува во зависност од температурата
- Капацитивен хигрометар – уред преку кој се менува капацитетот во зависност од влажноста

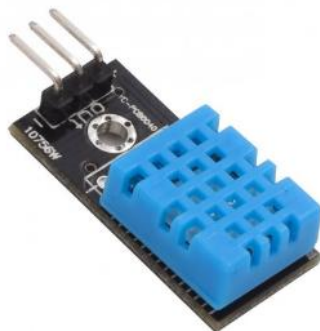
Сензорот автоматски ги чита двете вредности и ги пакува во еден дигитален сигнал кој може да се прочита.

Табела 5 Спецификации на сензорот DHT 11

Table 5 Specifications of the DHT 11 sensor

| Спецификации на DHT 11 | |
|------------------------|------------------|
| Работен напон | 3.5V до 5.5V |
| Излез | Сериски податоци |
| Температурен опсег | 0°C до 60°C |
| Опсег на влажност | 20% до 90% |
| Точност | ±2°C and ±2% |

Податоците од мерењата на овој сензор се запишуваат во excel документ врз кои подоцна ќе биде извршена предикција.



Слика 38 Сензор за мерење на температура и влажност (DHT11)

Figure 38 Temperature and humidity sensor (DHT11)

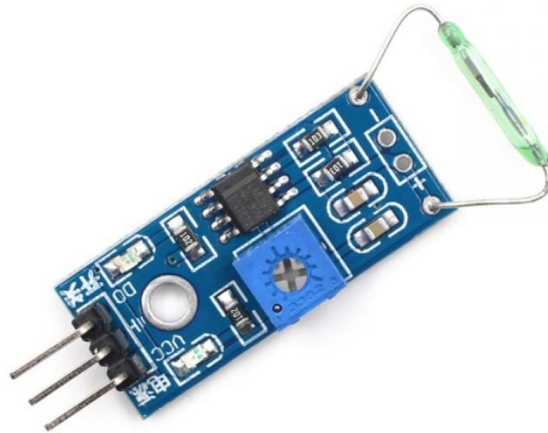
9.3.2.1.2. Reed switch

Reed switch е електричен прекинувач управуван од применето магнетно поле.

Овој сензор се состои од прекинувач, отпорници, кондензатор, потенциометар, компаратор LM393 IC, напојување и статусна ЛЕД диода во интегрално коло. Овој сензорски модул може да се користи во фотокопири, машини за перење, фрижидери, фотоапарати, кабинети за дезинфекција, врати, магнетика на прозорци, електромагнетни релеи, електронско мерење, мерачи на ниво, броила, водомери итн.

Рид прекинувач е пасивна компонента за електронско вклучување што се користи за контрола на протокот на електричната енергија во колото. Рид прекинувачот се состои од запечатен стаклен плик кој е исполнет со интерен гас наречен родиум и две црни еластични трски. Во нормални услови, двете трски се одделени во пликот, но кога магнетната материја ќе се приближи до стаклениот коверт, трските ќе се соберат заедно поради магнетното поле со што ќе се заврши електричното коло.

Во овој проект ќе биде искористен за отварање и затварање на прозорци. Со што можеме да ја контролираме и енергетската ефикасност [58].



Слика 39 Магнетен сензор (Reed Switch)

Figure 39 Reed Switch sensor

9.3.2.1.3. LED

Диода што емитира светлина (LED) е полупроводнички извор на светлина што емитира светлина кога струјата тече низ неа. Се користи за претставување на светлината. Светлината ќе биде контролирана преку системот и сензорите.[59]



Слика 40 ЛЕД

Figure 40 LED

9.3.2.1.4. Детектор на дожд (Rain detector)

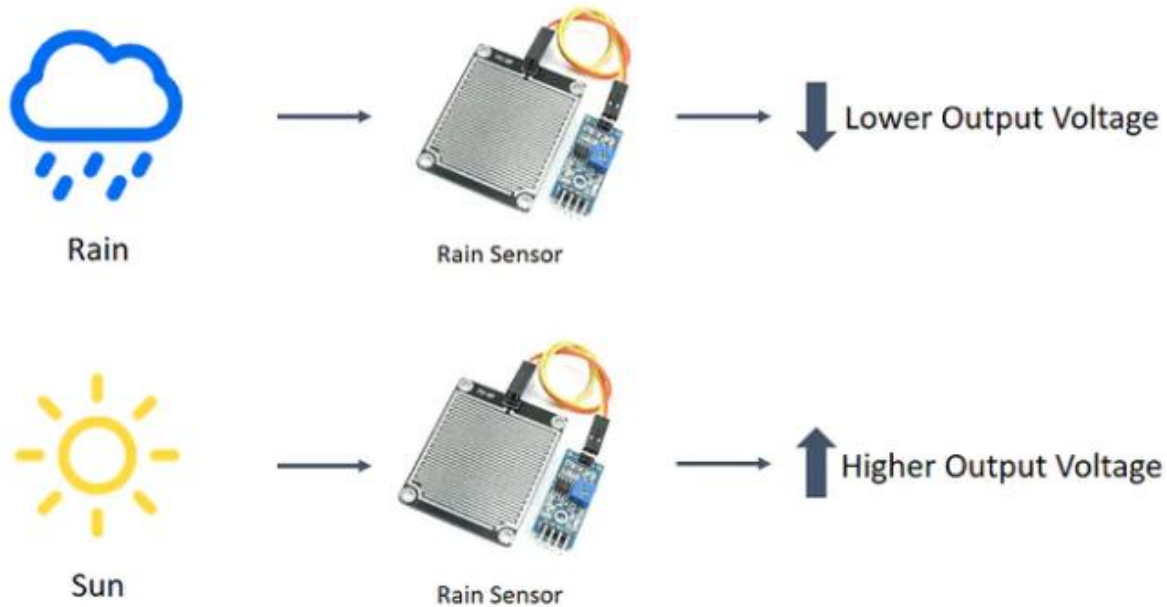
Сензорот за дожд е составен од плоча за откривање на дожд и компаратор кој управува со интелигенцијата. [60]



Слика 41 Сензор за детектирање на дожд

Figure 41 Rain detector

Сензорот делува како променлив отпор што ќе го смени статусот: отпорот се зголемува кога сензорот е влажен, а се намалува кога сензорот е сув.



Слика 42 Приказ како работи сензорот за детектирање на дожд

Figure 42 Demonstration of how the rain detection sensor works

Компараторот има 2 излеза поврзани со сензорот за дожд, дигитален излез (0/1) и аналоген излез (0 до 1023).

Во овој проект се користи детекција на дожд за автоматско затворање на прозорците и прилагодување на собната температура.

9.3.2.1.5. Отпорник на светлина (LDR)

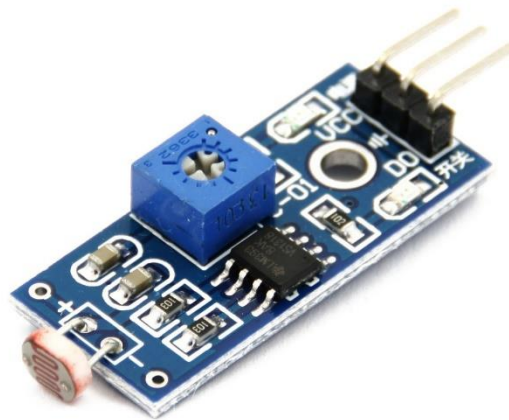
Отпорник на светлина е компонента која е чувствителна на светлина. Кога светлината паѓа врз неа, тогаш отпорот се менува.

Табела 6 Спецификации на отпорникот на светлина

Table 6 Specifications of the LDR

| Спецификации на отпорникот на светлина | |
|--|----------------------|
| Работен напон | 3.5V до 5.5V |
| Излез | Аналоген и дигитален |
| Осетливост | Подесувачка |

Во овој проект се користи за мерење на светлината во домот и надвор од него. Кога има доволно осветлување што може да го осветли домот или дворот, системот автоматски ќе ги исклучи светлата.[61]



Слика 43 Отпорник на светлина (LDR)

Figure 43 LDR Sensor

9.3.2.1.6. Серво мотори

Серво мотор е мал уред кој има излезно вретено. Ова вретено може да се постави во специфични аголни позиции со испраќање на серво кодиран сигнал. Сè додека кодираниот сигнал постои на влезната линија, моторот ќе ја задржи аголната позиција на вретеното. Ако кодираниот сигнал се смени, аголната позиција на вретеното се менува.[62]

Табела 7 Спецификации на серво моторот

Table 7 Specifications of the servo motor

| Спецификации на серво моторот | |
|-------------------------------|--|
| Работен напон | 3.5V до 5.5V |
| Излез | Дигитален излез (0 и 1, високо или ниско ниво) |
| Вртежен момент | 2.5kg/cm |
| Брзина | 0.1s/60° |
| Ротација | 0°-180° |

Серво моторите се користат во овој проект за отварање и затварање на гаражата.



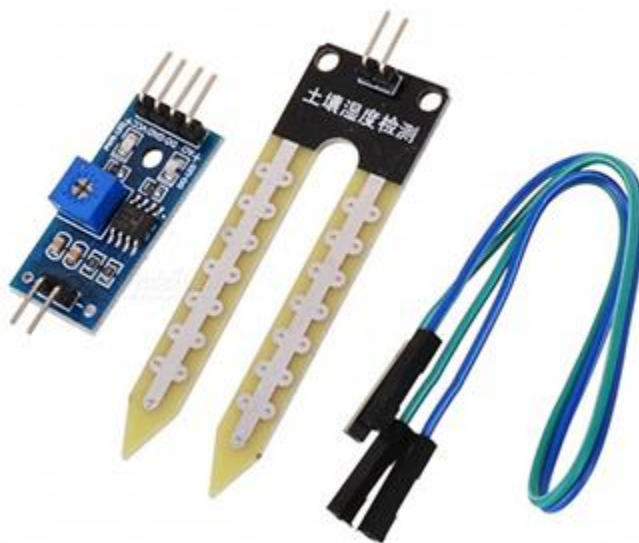
Слика 44 Серво мотор

Figure 44 Servo motor

9.3.2.1.7. Сензор за влага во почвата

Овој сензор ја мери волуметриската содржина на вода во почвата и ни го дава нивото на влага како излез. Сензорот е опремен и со аналоген и со дигитален излез, така што може да се користи и во аналоген и во дигитален режим.

Сензорот за влага во почвата се состои од две сонди кои се користат за мерење на волуметриската содржина на вода. Двете сонди дозволуваат струјата да помине низ почвата, а потоа се добива отпорна вредност за да се измери вредноста на влагата. [63]



Слика 45 Сензор за влага во почвата

Figure 45 Moisture sensor

Кога има повеќе вода, почвата ќе спроведе повеќе електрична енергија што значи дека ќе има помал отпор. Затоа, нивото на влага ќе биде поголемо. Сувата почва слабо спроведува електрична енергија, па кога ќе има помалку вода, тогаш почвата ќе спроведе помалку електрична енергија што значи дека ќе има поголема отпорност. Затоа, нивото на влага ќе биде помало.

За што се користи:

1. Земјоделство
2. Дворови / Градини
3. Истражување и предвидување на потенцијални суши, одрони, ерозии и сл.

9.3.2.1.8. Сензор за звук

Модулот за сензор за звук обезбедува лесен начин за откривање на звук и генерално се користи за откривање на интензитетот на звукот. Овој модул може да се користи за безбедност, прекинувач и мониторинг апликации.

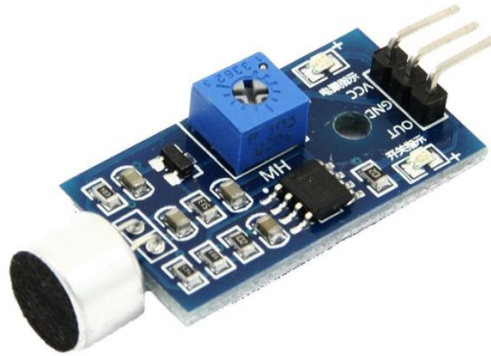
Модулот за сензор за откривање на звук го детектира интензитетот на звукот преку микрофон и се внесува во засилувач LM393. Кога сензорот открива звук, тој го обработува напонот на излезниот сигнал кој се испраќа до микроконтролер, а потоа ја извршува потребната обработка.

Се состои од потенциометар за прилагодување на зададената точка за нивото на звукот [64].

Табела 8 Спецификации на сензорот за звук

Table 8 Specifications of the sound sensor

| Спецификации на сензорот за звук | |
|----------------------------------|--|
| Работен напон | 3.5V до 5.5V |
| Излез | Дигитален излез (0 и 1, високо или ниско ниво) |
| Осетливост | подесувачка |



Слика 46 Сензор за звук

Figure 46 Audio sensor

Сензорот за звук се користи во овој проект за вклучување на LED светилката со ракоплескање.

9.3.2.1.9. 3V DC Мотор

Мотор со директна струја (DC) е вид на електричен уред што ја претвора електричната енергија во механичка енергија. DC моторите земаат електрична енергија преку директна струја и ја претвораат енергијата во механичка ротација.

DC моторите користат магнетни полиња што се појавуваат од генерираните електрични струи, што го напојува движењето на роторот фиксиран во рамките на излезната вретено. Вртежниот момент и брзината на движење зависат од електричниот влез и од дизајнот на моторот.



Слика 47 3V DC Мотор

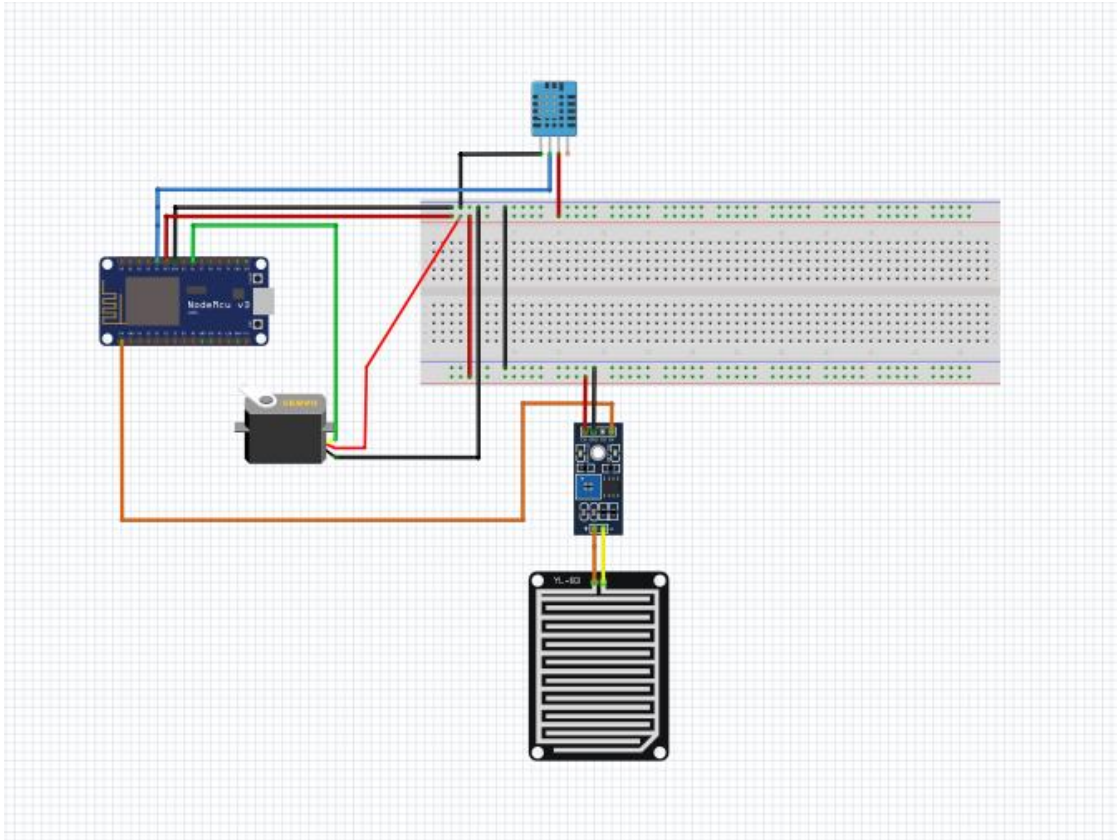
Figure 47 3V DC Motor

Во овој проект DC моторот е искористен како уред за греење и ладење. Контролата на моторот е овозможена со користење на L293d motor driver.

9.3.3. Шематски приказ

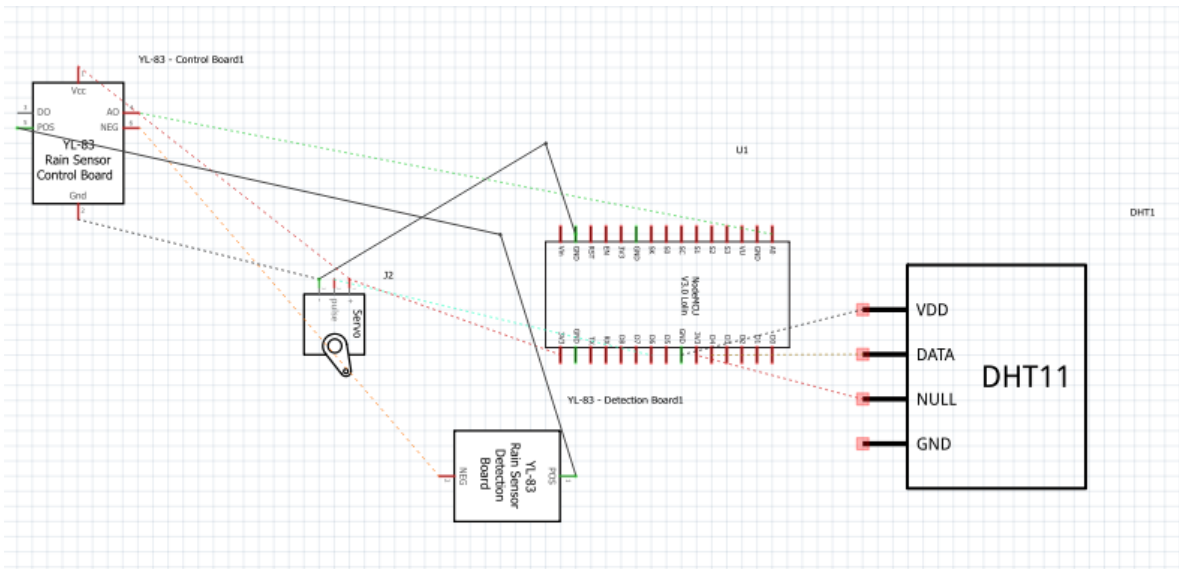
9.3.3.1 Поврзување на сензорот за температура и влажност, серво моторот и сензорот за детектирање на дожд со NodeMCU

На следнава шема е прикажано поврзувањето на сензорот за температура и влажност, сензорот за детекција на дожд и серво моторот. Како што може да се види од шемата, за сензорот за температура и влажност е искористен дигиталниот пин D4, заземјување и 3.3V напојување. За сензорот за детекција на дожд е искористен аналогниот пин A0, заземјување и напојување со вредност од 3.3V. Серво моторот е поврзан со дигиталниот пин D5, заземјување и напојување со вредност од 3.3V. Од NodeMCU се извлечени заземјувањето и напојувањето и се поврзани со жици на протобордот (развијачката плоча).



Слика 48 Поврзување на сензорот за температура и влажност, серво моторот и сензорот за детектирање на дожд со NodeMCU

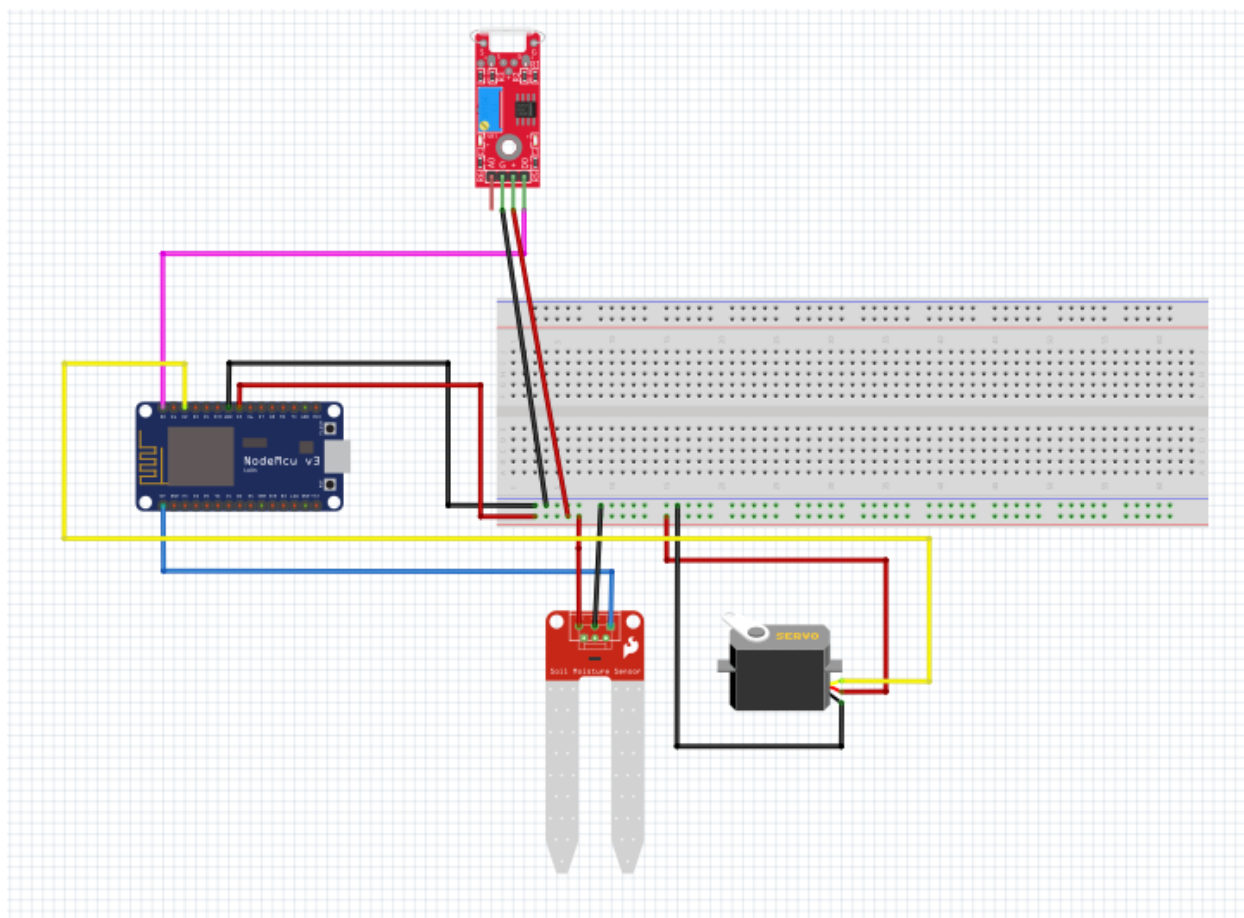
Figure 48 Connection of the dht11 module, servo motors and rain detector with NodeMCU



Слика 49 Шематски приказ на поврзување на сензорот за температура и влажност, серво мотор и сензорот за детектирање на дожд со NodeMCU
 Figure 49 Schematic representation of the connection between dht11 module, servo motor and rain detector with NodeMCU

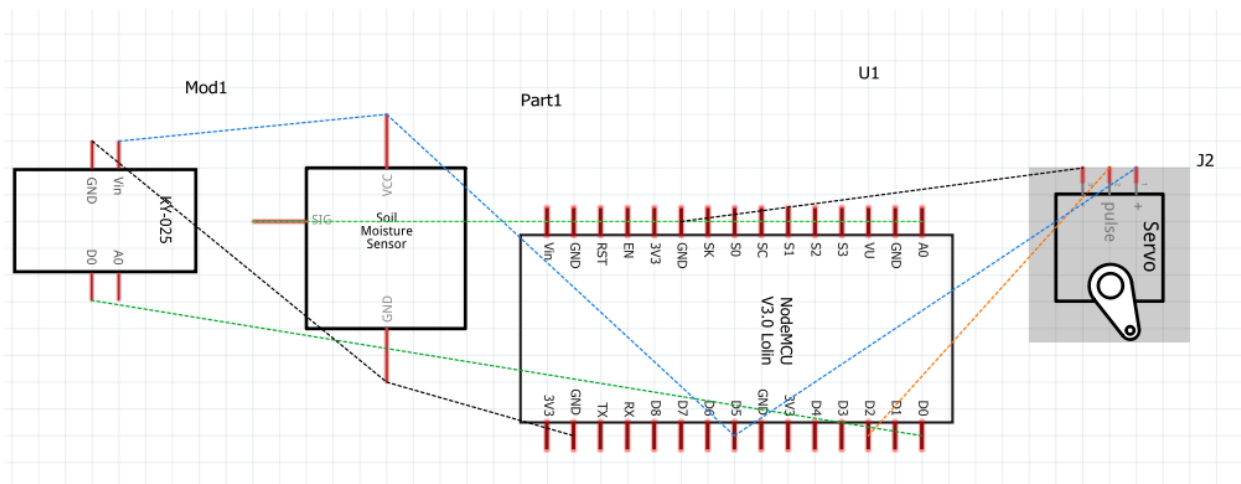
9.3.3.2 Поврзување на магнетен сензор, сензорот за детектирање на влага во почвата и серво мотор со NodeMCU

На следнава шема е прикажано поврзувањето на магнетен сензор, сензорот за детекција на дожд и серво моторот. Како што може да се види од шемата, за сензорот за магнетниот сензор е искористен дигиталниот пин D0, заземјување и 3.3V напојување. За сензорот за детекција на влага во почвата е искористен аналогниот пин A0, заземјување и напојување со вредност од 3.3V. Серво моторот е поврзан со дигиталниот пин D3, заземјување и напојување со вредност од 3.3V. Од NodeMCU се извлечени заземјувањето и напојувањето и се поврзани со жици на протобордот (развивачката плоча).



Слика 50 Поврзување на магнетен сензор, сензорот за детектирање на влага во почвата и серво мотор со NodeMCU

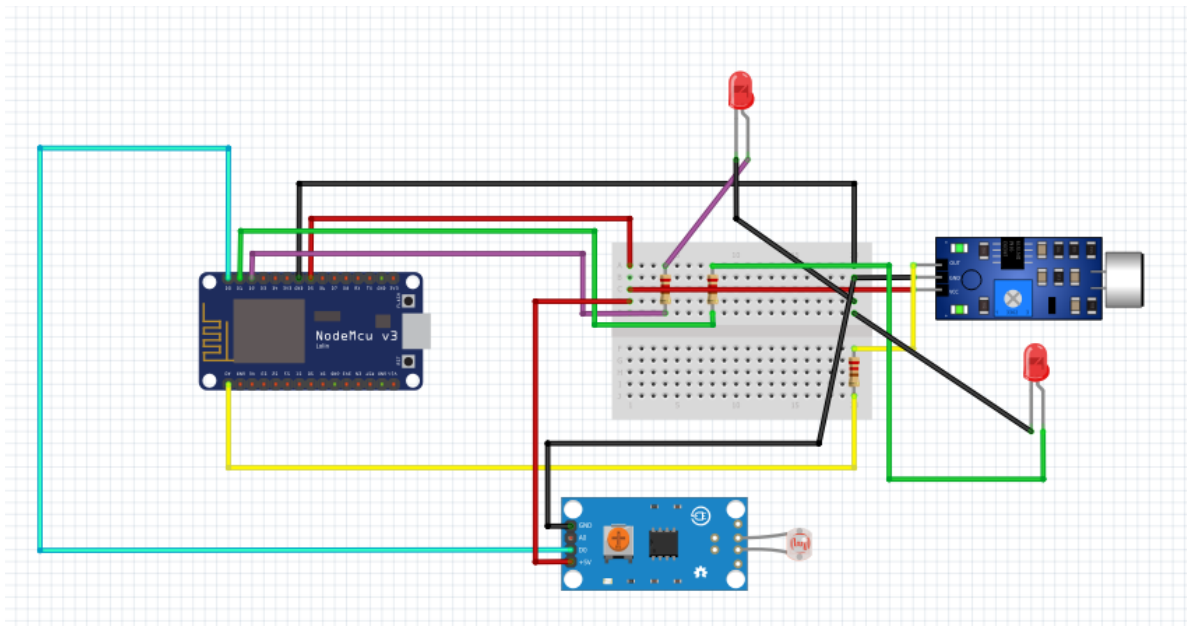
Figure 50 Connection of the reed switch module,moisture sensor and servo motor with NodeMCU



Слика 51 Шематски приказ на поврзување магнетен сензор, сензорот за детектирање на влага во почвата и серво мотор со NodeMCU
 Figure 51 Schematic representation of the connection between reed switch module, moisture sensor and Servo motor with NodeMCU

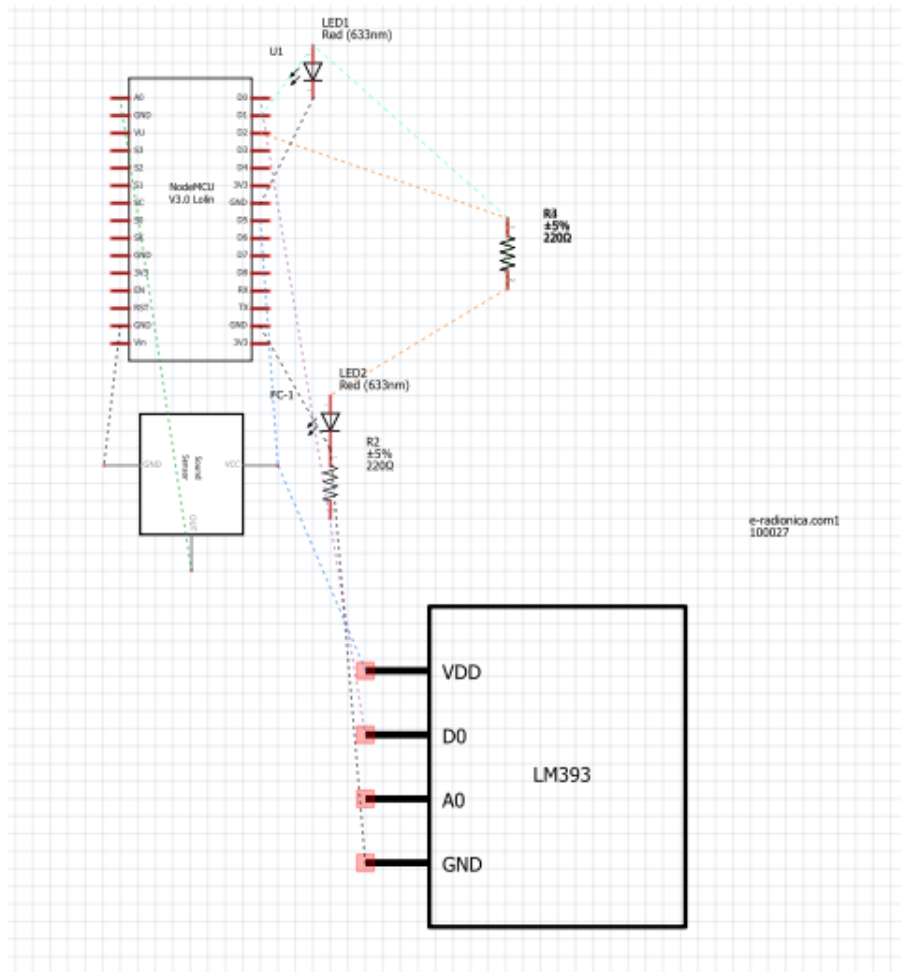
9.3.3.3 Поврзување на отпорникот на светлина, сензорот за звук и лед диоди со NodeMCU

На следнава шема е прикажано поврзувањето на отпорникот за светлина, сензорот за звук и лед диоди. Како што може да се види од шемата, за отпорникот на светлина е искористен дигиталниот пин D0, заземјување и 3.3V напојување. За сензорот за звук е искористен аналогниот пин A0, заземјување и напојување со вредност од 3.3V. Од NodeMCU се извлечени заземјувањето и напојувањето и се поврзани со жици на протобордот (развивачката плоча).



Слика 52 Поврзување на отпорникот на светлина, сензорот за звук и лед диоди со NodeMCU

Figure 52 Connection of the LDR module, audio module and LED with NodeMCU



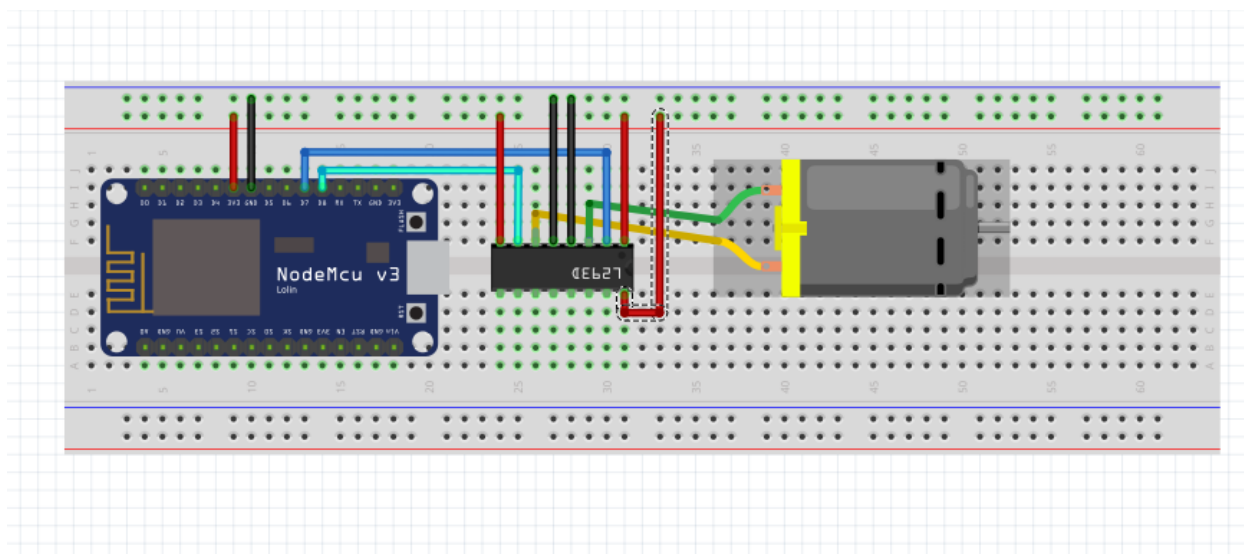
Слика 53 Шематски приказ на поврзување на отпорникот на светлина, сензорот за звук и лед диоди со NodeMCU

Figure 53 Schematic representation of the connection between LDR module, audio module and LED with NodeMCU

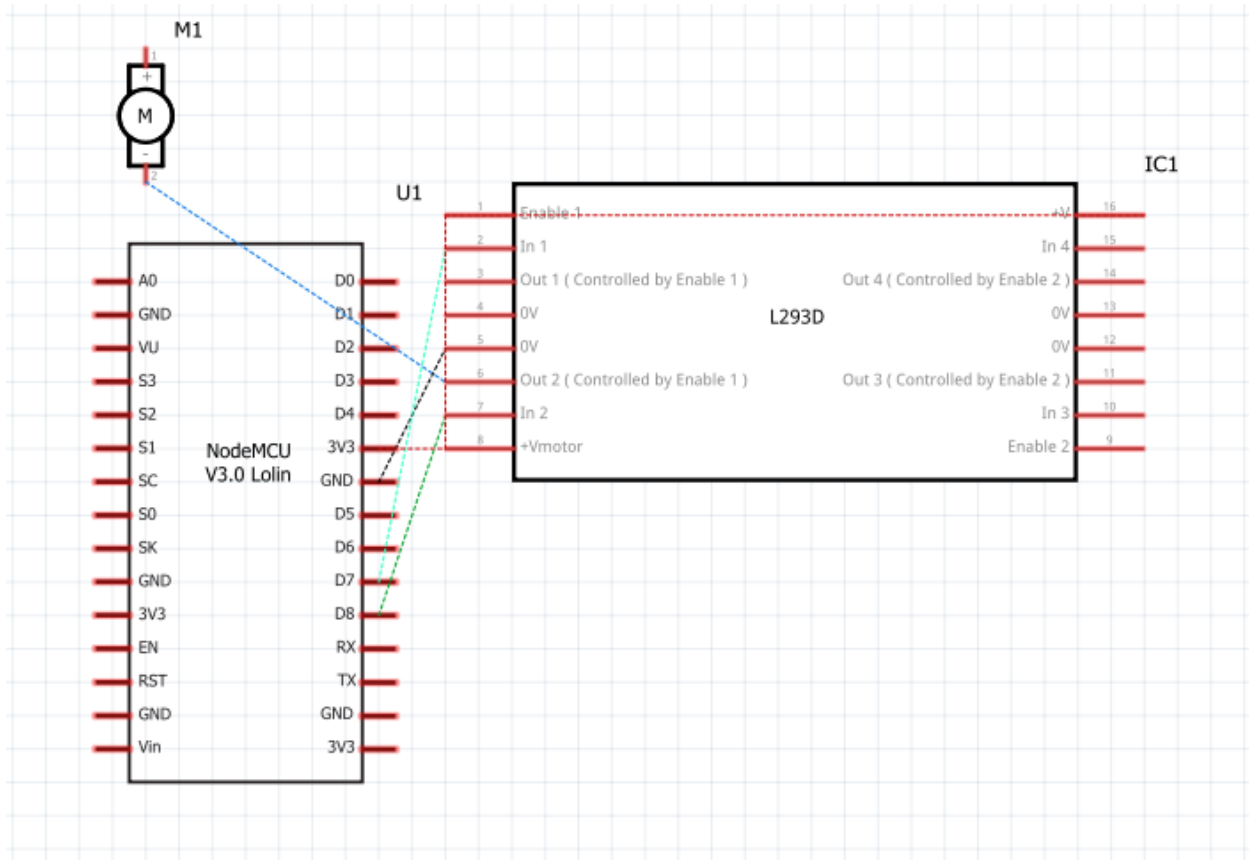
9.3.3.4 Поврзување на 3V DC мотор, L293d motor driver со NodeMCU

На следнава шема е прикажано поврзувањето на 3V DC мотор, L293d motor driver со NodeMCU. Со цел да се овозможи контрола на моторот е искористен L293d motor driver. Првиот пин е поврзан со 3.3V напојување. Вториот пин е поврзан со D7 пинот од NodeMCU. На третиот пин е поврзано напојувањето од моторот. Четвртиот и петиот пин се искористени за заземјување. Шестиот пин е поврзан со

заземјувањето на моторот. Седмиот пин е поврзан со D8 пинот од NodeMCU. Осмиот и шеснаесетиот пин се поврзани со 3.3V напојување.



Слика 54 Поврзување на 3V DC мотор, L293d motor driver со NodeMCU
Figure 54 Connection of the 3V DC motor, L293d motor driver with NodeMCU



Слика 55 Шематски приказ на поврзување на 3V DC мотор, L293d motor driver со NodeMCU

Figure 55 Schematic representation of the connection between the 3V DC motor, L293d motor driver with NodeMCU

10. Резултати и дискусија

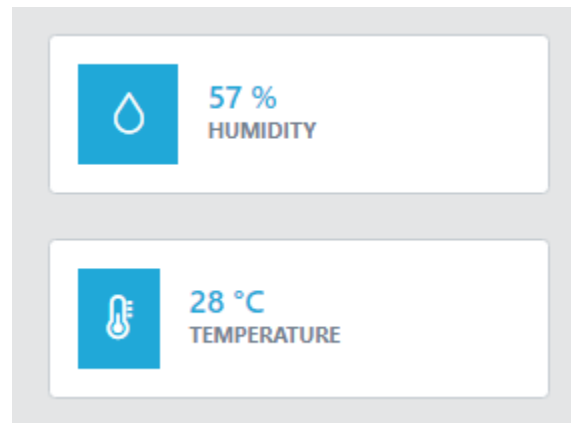
Во оваа глава ќе бидат наведени добиените резултати од изработениот проект.

1. Добиен е графички интерфејс преку кој можеме да ги контролираме уредите и да ја следиме нивната состојба.



Слика 56 Копчиња за контрола на уредите

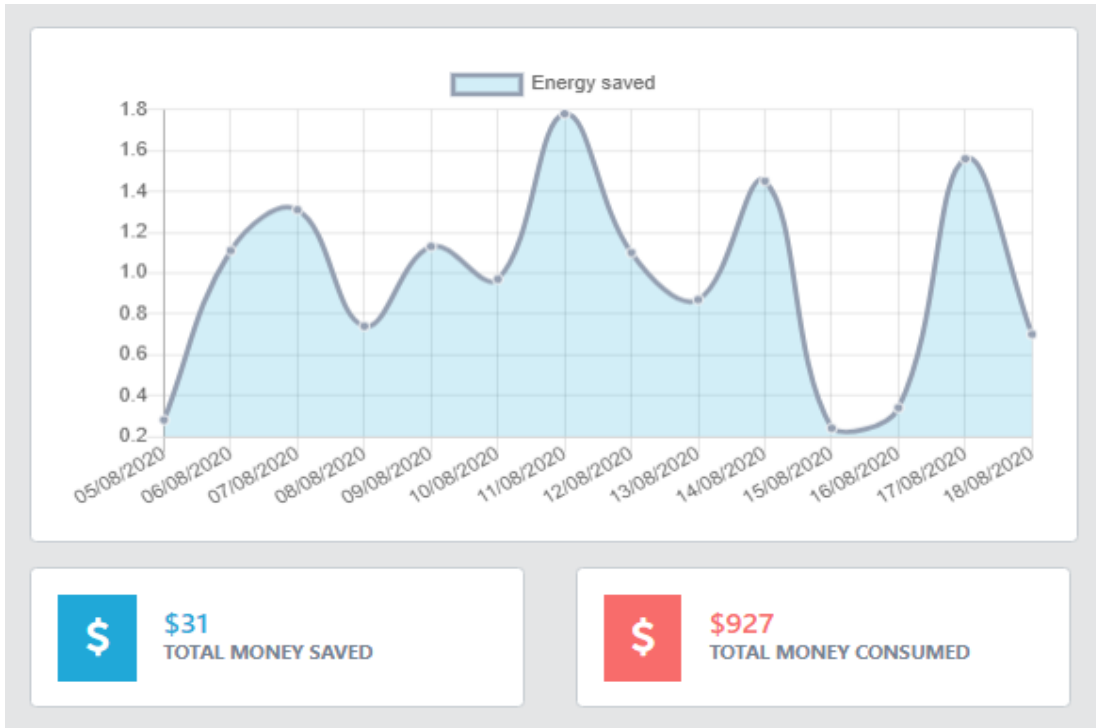
Figure 56 Buttons for controlling devices



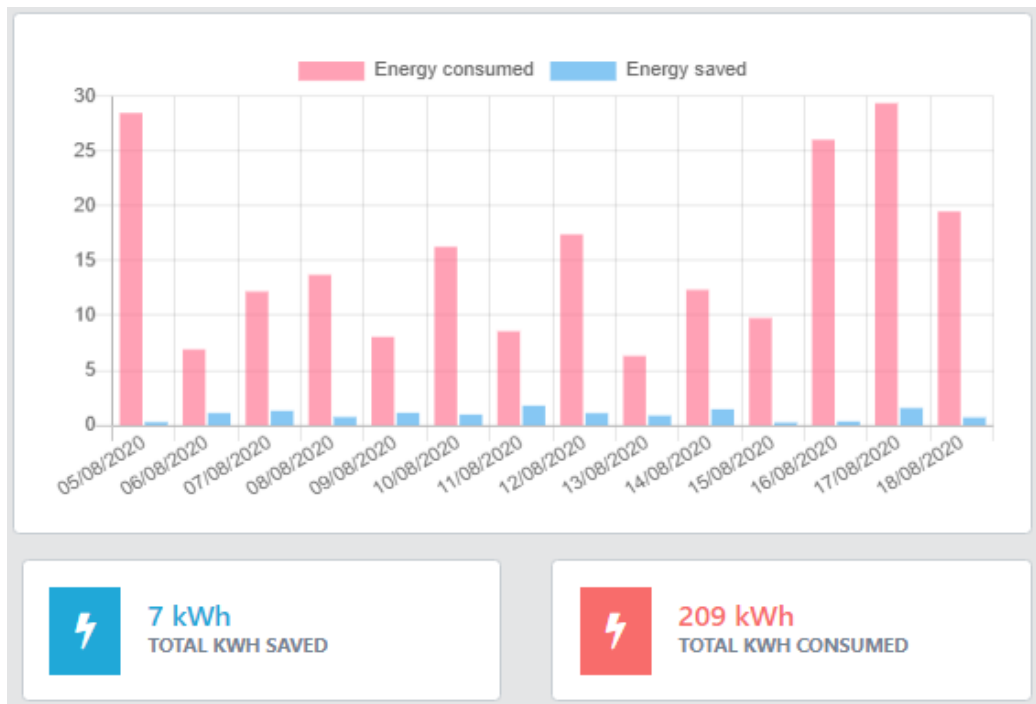
Слика 57 Приказ на моменталната температура и влажност

Figure 57 Representation of the current humidity and temperature values

2. Преку копче за препознавање на говор можеме да зададеме гласовни команди и да ги контролираме уредите.
3. Добиен е систем кој се грижи за контрола, пресметка и анализа на податоците од сензорите.



Слика 58 Приказ на резултатите од системот за енергетска ефикасност
 Figure 58 Overview of the results of the system for energy efficiency



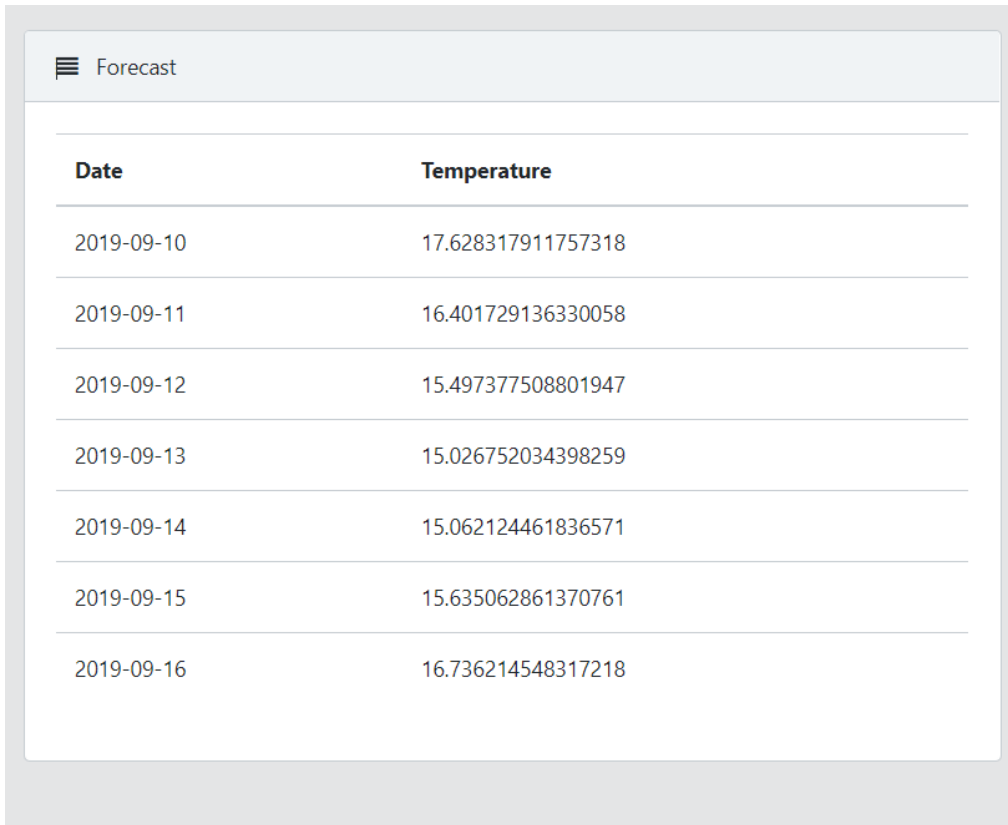
Слика 59 Приказ на резултатите од системот за енергетска ефикасност
 Figure 59 Overview of the results of the system for energy efficiency

4. Доколку има доволно светлина во домот или во дворот системот автоматски ги гаси светилките во домот или дворот со што се активира режимот за енергетска ефикасност.
5. Доколку сензорот за детекција на дожд детектира дожд тогаш системот автоматски ги затвора прозорците со што се активира режимот за енергетска ефикасност.
6. Корисникот ќе биде известен преку електронска пошта за времето кога започнува евтината тарифа на струјата.
7. Овозможена е контрола на лед диода преку ракоплескање.
8. Може да се следи состојбата на влажност на почвата во паркот.
9. За системот за предикција на температура е искористен сензорот за мерење на температура и влажност (DHT 11). Земени се податоци од работата на сензорот во нормални собни услови во времетраење од еден месец. Бидејќи искористениот алгоритам работи подобро со поголем број на податоци, останатиот дел од сетот од податоци е симулиран. Добиена е предикција за временската температура за наредните 7 дена.

На слика 35 е прикажано како изгледа предикцијата. Со портокалова боја се веќе постоечките податоци, а со зелена се предвидените податоци.

На слика 36 е прикажан график каде со плава боја е исцртано како се движи температурата од базата на податоци, а со портокалово е исцртана предикцијата за следните 7 дена.

Резултатите односно предикциите се прикажани на графичкиот интерфејс во форма на табела.



The image shows a screenshot of a web application interface. At the top left, there is a hamburger menu icon followed by the text "Forecast". Below this is a table with two columns: "Date" and "Temperature". The table contains seven rows of data, each representing a date from 2019-09-10 to 2019-09-16 and a corresponding numerical temperature value.

| Date | Temperature |
|------------|--------------------|
| 2019-09-10 | 17.628317911757318 |
| 2019-09-11 | 16.401729136330058 |
| 2019-09-12 | 15.497377508801947 |
| 2019-09-13 | 15.026752034398259 |
| 2019-09-14 | 15.062124461836571 |
| 2019-09-15 | 15.635062861370761 |
| 2019-09-16 | 16.736214548317218 |

Слика 60 Приказ на резултатите од предикцијата

Figure 60 Overview of the results of the forecast

11. Заклучок

Примамливата визија за паметна контрола врз околината ги мотивира сопствениците на домови да ја интегрираат технологијата за автоматизација во своите домови со ветувачки ефекти на зголемена удобност, мир на умот и намалени оперативни трошоци.

Голем дел од придобивките од паметниот дом доаѓаат на начини што не се толку очигледни. Паметните домови заштедуваат енергија и топлина, поради што се еколошки и можат да бидат дури и поевтини за живеење.

Во овој магистерски труд е прикажан процесот на изработка на систем за енергетска ефикасност кој се состои од неколку апликации со кои можеме да ги видиме придобивките на паметните домови. Беа искористени уреди за: мерење на температура и влажност, магнетен сензор, детектор на дожд, отпорник на светлина, серво мотори, сенор за влага во почвата, сензор за звук и LED диоди.

Од резултатите може да се заклучи дека имплементирањето на паметните домови во секојдневниот живот, благодарение на напредокот на технологијата ни овозможува поубав, покомотен, посигурен и поевтин живот.

12. Користена литература

1. Wood, G., & Newborough, M. 2003. Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design. *Energy and Buildings*, 35, p821-841
2. Bates, O., A. Clear, A. Friday, M. Hazas and J. Morley (2012). Accounting for Energy-Reliant Services within Everyday Life at Home. *Pervasive Computing*. J. Kay, P. Lukowicz, H. Tokuda, P. Olivier and A. Krüger, Springer Berlin Heidelberg. 7319: 107-124.
3. Behzad Lashkari.; Yuxiang Chen.; Petr Musilek; Energy Management for Smart Homes - State of the Art, 21 August 2019
4. <https://www.oracle.com/internet-of-things/what-is-iot.html> . (Access Date: 03.02.2020)
5. Ithiel de Sola Pool, *Technologies Without Boundaries*, Eli M. Noam, ed. (Cambridge, Mass.: Harvard University Press, 1990), p. 56.
6. Tim Berners-Lee with Mark Fischetti, *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*, Harper San Francisco, 1999
7. Evolution of IoT in Smart Vehicles: An Overview, Keertikumar M., Shubham M., R.M. Banakar
8. <http://wecam.org/bio.htm> (Access Date: 10.04.2020)
9. Internet of Things (IoT) and its Challenges for Usability in Developing Countries, Egemen Hopali, Özalp Vayvay - January 2018
10. <https://www.gapintelligence.com/blog/are-we-there-yet/> (Access Date: 22.06.2020)
11. Can your pet rabbit read your email?: a critical analysis of the Nabaztag rabbit , Chung-Ching Huang, Jeffrey Bardzell
12. Yamazaki, T. Beyond the smart home. In *Proceedings of the 2006 International Conference on Hybrid Information Technology*, Cheju Island, South Korea, 9–11 November 2006.
13. Dishman, E. Inventing wellness systems for aging in place. *Computer* 2004, 5, 34–41. doi: 10.1109/MC.2004.1297237.
14. Chan, M.; Campo, E.; Estève, D.; Fourniols, J.-Y. Smart homes - Current features and future perspectives. *Maturita* 2009, 64, 90–97. doi: 10.1016/j.maturitas.2009.07.14.
15. Nallapaneni Manoj Kumar; Archana Dash; - The Internet of Things: An Opportunity for Transportation and Logistics. 2017 IEEE International Conference on Inventive

16. Intel, olution Blueprint Internet of Things, Building an Intelligent Transportation System with the Internet of Things (IoT) <http://www.intel.com/iot>. (Access Date: 07.02.2020)
17. Li Da Xu, Wu He, Shancang - LInternet of Things in Industries: A Survey. IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 10, NO. 4, NOVEMBER 2014
18. Predix - <https://www.predix.io/> (Access Date: 11.02.2020)
19. Georgios Kavallieratos; Nabin Chowdhury; Stephen Wolthusen; -Threat Analysis for Smart Homes. Future Internet 2019, 11, 207; doi:10.3390/fi11100207
20. Meensika Sripan; Xuanxia Lin; Ponchan Petchlorlean; Mahasak Ketcham; - Research and Thinking of Smart Home Technology. International Conference on Systems and Electronic Engineering (ICSEE'2012) December 18-19, 2012 Phuket (Thailand)
21. Christian Reinisch; Mario J. Kofler; Felix Iglesias; Wolfgang Kastner; - ThinkHome Energy Efficiency in Future Smart Homes. EURASIP Journal on Embedded Systems Volume 2011, Article ID 104617
22. W. Kastner, G. Neugschwandtner, S. Soucek, and H. M. Newman, "Communication systems for building automation and control," Proceedings of the IEEE, vol. 93, no. 6, pp. 1178– 1203, 2005.
23. The First Computational Theory of Mind and Brain: A Close Look at Mcculloch and Pitts's "Logical Calculus of Ideas Immanent in Nervous Activity", Gualtiero Piccinini - August 2004
24. F. Rosenblatt, Psychol. Rev. 65 (1958) 386–408
25. B. Widrow, M.E. Hoff, Jr., Adaptive Switching Circuits, u: IRE WESCON Convention Record, Vol. 4, pp. 96–104, The Institute of Radio Engineers, Inc., New York, SAD, 1960.
26. Cortana Microsoft - <https://support.microsoft.com/en-us/topic/what-is-cortana-953e648d-5668-e017-1341-7f26f7d0f825> (Access Date: 22.02.2020)
27. Siri apple - <https://www.apple.com/siri/> (Access Date: 15.03.2020)
28. OpenAi - <https://openai.com/projects/five/> (Access Date: 21.03.2020)
29. Angular - <https://angular.io/docs> (Access Date: 04.04.2020)
30. „Spring,“ Pivotal Software, Inc, 2016. : <https://spring.io/> (Access Date: 05.04.2020)
31. Python - <https://www.python.org/doc/essays/blurb/>

32. Python - <https://www.python.org/about/quotes/>
33. ActiveMq - <http://activemq.apache.org/>
34. Qusay I. Sarhan; Idrees S. Gawdan; - Java message service based performance comprasion of apache activemq and apache Apollo brokers. - Science Journal of University of Zakho 5(4), 307-312, Dec.-2017
35. Ahuja, S. P.; Mupparaju, N.; - (2014). Performance Evaluation and Comparison of Distributed Messaging Using Message Oriented Middleware. Computer and Information Science, 7(4), 9-20
36. Peter R. Egli (2013.), „MQTT - MQ Telemetry Transport for Message“, <http://www.slideshare.net/PeterREgli/mq-telemetry-transport>
37. Stephen Nicholas (2012.), „Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on Android“, <http://stephendnicholas.com/archives/1217>
38. Oasis (2014), „MQTT Version 3.1.1 OASIS Standard“, <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>
39. Mqtt - <https://github.com/mqtt/mqtt.github.io/wiki>
40. Paho - <https://www.eclipse.org/paho/files/javadoc/index.html>
41. Dipa Soni; Ashwin Makwana; - A survey on mqtt: A protocol of internet of things IOT.
42. Luzuriaga, J. E., Cano, J. C., Calafate, C., Manzoni, P., Perez, M., & Boronat, P. (2015, September). Handling mobility in IoT applications using the MQTT protocol. In Internet Technologies and Applications (ITA), 2015 (pp. 245-250). IEEE.
43. Performance test - <https://flespi.com/blog/http-vs-mqtt-performance-tests>
44. Qigang Liu; Xiangyang Sun; - Research of Web Real-Time Communication Based on Web Socket
45. D. Sheiko, “Persistent Full Duplex Client-Server Connection via Web Socket,” 2010. <http://dsheiko.com/weblog/persistent-full-duplex-client-server-connection-via-web-socket>
46. Yogendra Singh Parihar; Scientist D; - Internet of Things and Nodemcu - A review of use of Nodemcu ESP8266 in IoT products. 2019 JETIR June 2019, Volume 6, Issue 6.
47. R. Muniz, J. Diaz, F. Nuno, M. J. Prieto, and A. M. Pernia, “A Smart Power Meter to Recharge Electric Vehicles in Communal Parking Areas,” IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3448–3454, Apr. 2019

48. "ESP8266 Overview | Espressif Systems."
<https://www.espressif.com/en/products/hardware/esp8266ex/overview>
49. "Espruino on ESP8266 WiFi." - <https://www.espruino.com/EspruinoESP8266>
(Access Date: 17.05.2020)
50. Overview NodeMCU documentation - <https://nodemcu.readthedocs.io/en/master/>
51. <https://www.erc.org.mk/pages.aspx?id=153> (Access Date: 21.05.2020)
52. Bašić B.D., Čupić M., Šnajder J., „Umjetne neuronske mreže“; Zagreb; svibanj 2008
53. Afan Galih Salman; Yaya Heryadi; - Weather forecasting using merged Long Short-Term Memory Model (LSTM) and Autoregressive Integrated Moving Average (ARIMA) Model - July 2018 Journal of Computer Science 14(7):930-938
54. Xingjian Shi; Zhourong Chen; Hao Wang; - Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting
55. Keras - <https://keras.io/> (Access Date: 15.08.2020)
56. Keras - <https://www.infoworld.com/article/3336192/what-is-keras-the-deep-neural-network-api-explained.html> (Access Date: 22.08.2020)
57. DHT11 - <https://www.adafruit.com/product/386> (Access Date: 26.05.2020)
58. Reed Switch - <https://www.arrow.com/en/research-and-events/articles/the-reed-switch-ingeniously-simple-sensing> (Access Date: 19.07.2020)
59. LED - <https://www.ledsmagazine.com/leds-ssl-design/materials/article/16701292/what-is-an-led> (Access Date: 20.07.2020)
60. Rain Sensor - <https://www.instructables.com/Arduino-Modules-Rain-Sensor/>
(Access Date: 24.07.2020)
61. LDR Sensor - <https://maker.pro/arduino/tutorial/how-to-use-an-ldr-sensor-with-arduino> (Access Date: 02.08.2020)
62. Servo Motor - <https://www.instructables.com/Interfacing-Servo-Motor-With-NodeMCU/> (Access Date: 03.08.2020)
63. Moisture Sensor - <https://www.electronicwings.com/nodemcu/soil-moisture-sensor-interfacing-with-nodemcu> (Access Date: 10.08.2020)
64. Sound Sensor - <https://www.elprocus.com/sound-sensor-working-and-its-applications/> (Access Date: 12.08.2020)