

# Primjena interneta stvari u upravljanju sustavom centralnog grijanja

---

Vanja, Kristijan

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Bjelovar University of Applied Sciences / Veleučilište u Bjelovaru**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:144:723316>

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

Download date / Datum preuzimanja: **2024-09-16**



Repository / Repozitorij:

[Repository of Bjelovar University of Applied Sciences - Institutional Repository](#)



VELEUČILIŠTE U BJELOVARU  
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

**PRIMJENA INTERNETA STVARI U UPRAVLJANJU SUSTAVOM  
CENTRALNOG GRIJANJA**

Završni rad br. 14/RAČ/2023

Kristijan Vanja

Bjelovar, listopad 2023.



Veleučilište u Bjelovaru  
Trg E. Kvaternika 4, Bjelovar

**1. DEFINIRANJE TEME ZAVRŠNOG RADA I POVJERENSTVA**

Student: **Kristijan Vanja**

JMBAG: **0314018054**

Naslov rada (tema): **Primjena interneta stvari u upravljanju sustavom centralnog grijanja**

Područje: **Tehničke znanosti**

Polje: **Računarstvo**

Grana: **Programsko inženjerstvo**

Mentor: **Krunoslav Husak, dipl. ing. rač.**

zvanje: **predavač**

**Članovi Povjerenstva za ocjenjivanje i obranu završnog rada:**

1. **Krešimir Markota, mag. ing. comp., predsjednik**
2. **Krunoslav Husak, dipl. ing. rač., mentor**
3. **Danijel Radočaj, mag. inž. meh., član**

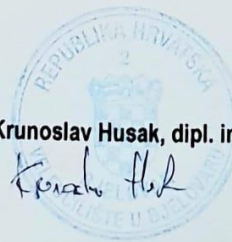
**2. ZADATAK ZAVRŠNOG RADA BROJ: 14/RAČ/2023**

U sklopu završnog rada potrebno je:

1. Analizirati i opisati Arduino razvojnu okolinu i usporediti ju s drugim razvojnim okolinama.
2. Analizirati i opisati ESP8266 mikrokontroler i usporediti ga s drugim sličnim mikrokontrolerima.
3. Predložiti i opisati komponente korištene za izradu hardverske podrške sustava interneta stvari za upravljanje sustavom centralnog grijanja.
4. Izraditi hardversko rješenje interneta stvari za upravljanje sustavom centralnog grijanja.
5. Izraditi softversko rješenje interneta stvari za upravljanje sustavom centralnog grijanja.

Datum: 22.09.2023. godine

Mentor: **Krunoslav Husak, dipl. ing. rač.**



## *Zahvala*

Zahvaljujem svojoj obitelji, djevojci i prijateljima na potpori i svojoj pomoći koju su mi pružili tijekom mog studiranja, kako financijskoj tako i psihičkoj. Od obitelji posebno se moram zahvaliti stricu Željku Štefančiću koji je i sam programski inženjer te mi je uvijek bio spreman pomoći s bilo kakvim poteškoćama koje sam imao tijekom studija. Posebno se zahvaljujem svom mentoru, profesoru Krunoslavu Husaku, dipl.ing.rač. i voditelju studija smjera računarstvo profesoru Tomislavu Adamoviću, mag.ing.el. koji su mi najviše izašli u susret brojnim satima konzultacija i svojom profesionalnom pomoći koju su mi pružili tijekom mog akademskog obrazovanja. Zahvaljujem se svim ostalim profesorima koji su vodili predmete koje sam pohađao tijekom svog akademskog obrazovanja, na izuzetnoj susretljivosti i na kvalitetnom obrazovanju.

# Sadržaj

<b>1.</b>	<b>Uvod.....</b>	<b>1</b>
<b>2.</b>	<b>INTERNET STVARI.....</b>	<b>2</b>
	2.1 <i>Kako internet stvari funkcionira.....</i>	2
	2.2 <i>Povijest interneta stvari.....</i>	3
	2.3 <i>Primjena interneta stvari.....</i>	3
<b>3.</b>	<b>RAZVOJNE OKOLINE.....</b>	<b>8</b>
	3.1 <i>Arduino razvojna okolina.....</i>	8
	3.1.1    Arduino komponente.....	8
	3.1.2    Arduino integrirana razvojna okolina.....	10
	3.2 <i>Raspberry Pi.....</i>	14
	3.2.1    Čarobnjak za izradu projekta.....	16
	3.2.2    Prednosti razvojnog okruženja.....	17
<b>4.</b>	<b>MIKROKONTROLERI (MCU).....</b>	<b>19</b>
	4.1 <i>ESP8266.....</i>	19
	4.2 <i>Nordic Semiconductor nRF52 Series.....</i>	21
	4.2.1    Bluetooth Low Energy.....	22
	4.2.2    Zigbee.....	22
	4.3 <i>RP2040.....</i>	23
<b>5.</b>	<b>IoT sustav za upravljanje centralnog grijanja.....</b>	<b>24</b>
<b>6.</b>	<b>HARDVERSKO RJEŠENJE.....</b>	<b>26</b>
	6.1 <i>Lolin NodeMCU ESP8266 12 – E V3.....</i>	26
	6.2 <i>GY-BME280 modul.....</i>	27
	6.3 <i>SRD-05VDC-SL-C.....</i>	28
<b>7.</b>	<b>SOFTVERSKO RJEŠENJE.....</b>	<b>29</b>
	7.1 <i>Korisnička Angular web aplikacija.....</i>	29
	7.2 <i>MQTT posrednik.....</i>	32
	7.2.1    MQTT komponente.....	32
	7.3 <i>Go aplikacijsko programsko sučelje.....</i>	33
	7.4 <i>Izrada programskog rješenja mikrokontrolera.....</i>	34
<b>8.</b>	<b>ZAKLJUČAK.....</b>	<b>38</b>

9.	LITERATURA .....	39
10.	OZNAKE I KRATICE.....	43
11.	SAŽETAK.....	45
12.	ABSTRACT .....	46

## 1. Uvod

U današnjem svijetu ne možemo zamisliti dan bez korištenja interneta i uređaja koji su povezani na njega. Pojam IoT (engl. *Internet of Things*) ili „Internet Stvari“ odnosi se na kolektiv mrežno povezanih uređaja i tehnologije koji komuniciraju međusobno i sa udaljenim serverima. Zahvaljujući sve jeftinijim računalnim čipovima sada imamo milijarde uređaja povezanih na internet. Internet stvari integrira „stvari“ koje svakodnevno koristimo, s internetom kako bi nam poboljšao kvalitetu života.

Tema ovog završnog rada je „Primjena interneta stvari u upravljanju sustava centralnog grijanja“. Kako se u zimskim mjesecima budimo rano i u hladnoći, uključujemo grijanje kako bi se kuća zgrijala dovoljno da ne bude prehladno dok se spremamo za odlazak na posao, školu ili fakultet. Prije odlaska iz kuće, grijanje se gasi ili smanjuje na određenu temperaturu radi manje potrošnje te se doma opet vraćamo u hladnu kuću. Tijekom dana se mijenja vanjska temperatura što potencijalno utječe na to želimo li da nam centralno grijanje bude uključeno ili ne, također moguće je da se nećemo vratiti u vrijeme u kojem se najčešće vraćamo. Izradom ovog rada, korištenjem interneta stvari, riješiti ćemo gore navedene probleme udaljenim uključivanjem, isključivanjem ili namještanjem intervala koji automatski odlučuju o radu sustava centralnog grijanja.

Za izradu rada koristi se mikroracunalo NodeMCU ESP8266 – 12E koje se bavi učitavanjem temperature, vlažnosti i pritiska zraka koristeći temperaturni senzor BME280, ovisno o očitanoj temperaturi i postavkama zadanim od strane korisnika, relej SRD-05VDC-SL-C uključuje ili isključuje sustav centralnog grijanja.

Web aplikacija koja je napisana u Angular okvirniku, za koji je korišteno integrirano razvojno okruženje Visual Studio Code, korisnici mogu unijeti postavke koje odlučuju o radu sustava centralnog grijanja. Ukoliko korisnik ne želi ručno paliti ili gasiti grijanje, može namjestiti intervale paljenja i gašenja odabirom vremenskog razdoblja od – do te željene temperature.

Nakon što korisnik odabere postavke za svoj sustav centralnog grijanja, one se šalju na aplikacijsko programsko sučelje koje je napisano u programskom jeziku Go, nakon čega se podaci obrađuju, zapisuju u bazu te šalju na mikroracunalo pomoću posrednika MQTT (engl. Message Queing Telemetry Transport). Za programiranje mikroracunala korišteno je integrirano razvojno okruženje Arduino te programski jezik C. Mikroracunalo i programsko sučelje koriste MQTT protokol za razmjenu poruka koji radi na principu objavljivanja i preplate.

## 2. INTERNET STVARI

Internet je tehnologija koja je promijenila ljudsku svakodnevicu, kako u poslovnom tako i u privatnom svijetu te je to nešto bez čega se danas ne može zamisliti ljudski život. Mnogi svjetski stručnjaci internet stvari nazivaju „Novom velikom stvari“ zbog široke primjene koju nam ta tehnologija nudi te je u zadnjih nekoliko godina postala jedna od najbitnijih tehnologija 21. stoljeća. Pretpostavlja se da će internet stvari s vremenom koristiti u većini kućanstava i poslovnih objekata s ciljem poboljšanja i pojednostavljenja ljudskih života te rasta svjetske ekonomije. Internet stvari podrazumijeva pojam velike količine uređaja poput raznih senzora, aktuatora, mikroračunala, uređaja za pohranu podataka, uređaja za komunikaciju s korisnicima i ostalih koji su spojeni na internet te obrađuju prikupljene podatke u realnom vremenu. Primjer takvih uređaja uključuje mobilne telefone, termostate, rasvjetne sustave, sigurnosne kamere, automobile, gradove, čak i životinje. Obzirom na to da se s navedenih uređaja očitava velika količina podataka, potrebna je „Big Data“ tehnologija kako bi se si ti podaci mogli prikupiti, spremati i analizirati. Za analizu podataka koriste se alati poput logaritama za strojno učenje, vizualni prikaz podataka i analitički modeli za predviđanje[1, 2].

### 2.1 Kako internet stvari funkcionira

Temelj interneta stvari je IP (engl. *Internet Protocol*) i TCP (engl. *Transfer Control Protocol*). Ovi protokoli tvore bazu za komunikaciju između brojnih uređaja interneta stvari koji su međusobno povezali pomoću interneta. Podaci s uređaja se prikupljaju, procesiraju i prosljeđuju, žičano i najčešće bežično koristeći:

- Ethernet
- Wi-Fi
- Bluetooth
- Radio frekvenciju (RFID (engl. *Radio-frequency identification*))
- Komunikaciju bliskog polja (NFC (engl. *Near Field Communcation*))

Brojni mrežni standardi omogućavaju podacima da dosegnu pravu „stvar“ te tako povezuju fizički i digitalni svijet[1].



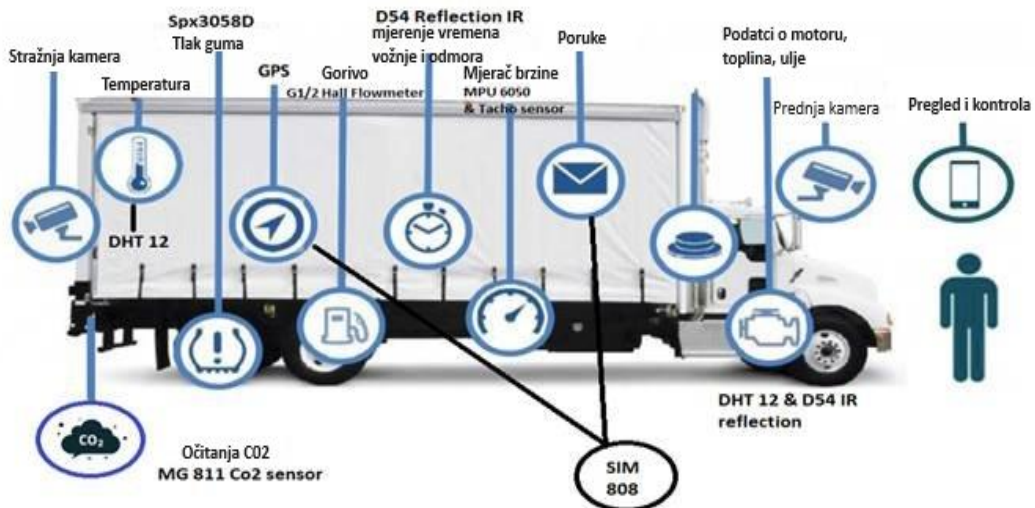
## 2.2 Povijest interneta stvari

Prvi primjeri međusobno povezanih elektroničkih uređaja postoje od 19. stoljeća i izuma telegrafa, no porijeklo interneta stvari potječe krajem 1960-ih. Tada je skupina znanstvenika prvi puta smislila način međusobnog povezivanja računala, primjer toga je ARPANET (engl. *Advanced Research Projects Agency Network*) te je on preteča današnjeg interneta. Krajem 1970-ih svjetske vlade, poduzeća i potrošači počeli su istraživati načine povezivanja osobnih računala, te su u 1980-im komunikaciju uspjeli uspostaviti lokalnom mrežom (LAN (engl. *Local Area Network*)). Sredinom 1990-ih internet je proširio sposobnost komunikacije između računala na globalnom nivou. Britanski znanstvenik Kevin Ashton je 1997. je krenuo istraživati tehnologiju radio frekvencijske identifikacije (RFID), koja bi omogućila povezivanje fizičkih uređaja pomoću mikročipova i bežičnih signala, te je tada prvi puta koristio izraz „Internet stvari“. Nedugo nakon toga došlo je do velikog napretka u tehnologiji, pojavili su se pametni mobilni uređaji, mogućnost veće procesorske moći i poboljšani programski algoritmi što je omogućilo lakše i brže pribavljanje, spremanje, procesiranje i dijeljenje podataka. U isto vrijeme pojavili su se profinjeniji senzori koji su mogli mjeriti, pokret, temperaturu i vlažnost zraka, smjer vjetra, zvuk, svjetlost, vibracije i mnoge druge podatke[1].

## 2.3 Primjena interneta stvari

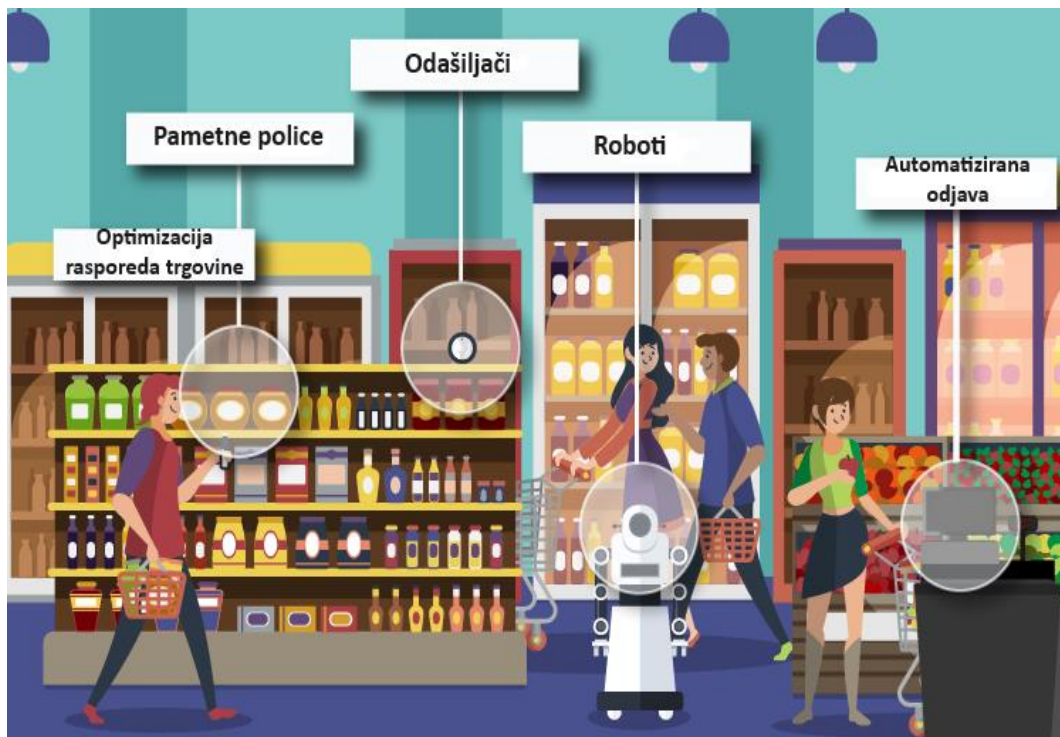
Industrijski uređaji interneta stvari se mogu koristiti za nadzor uređaja za proizvodnju kako bi se pratile performanse, otkrile pogreške i optimizirali procesi proizvodnje. Takav primjer vidimo u sensorima koji se koriste kako bi pratili razni uvjeti poput temperature i vlažnosti u proizvodnom pogonu time omogućavajući optimalne uvjete za osjetljive proizvode. Senzori se također koriste za praćenje fizičkog stanja opreme kako bi ju proizvođači brže i lakše mogu provjeriti i po potrebi zamijeniti, što tvrtkama omogućava smanjenje troškova i bolje upravljanje radnim vremenom. Uređaji se također koriste za praćenje inventara, kvalitete završenih proizvoda i upravljanje zalihama.

Internetom stvari beneficira sistem transporta i logistike. Automobili, kamioni, brodovi i vlakovi koji prevoze robu mogu biti preusmjereni ovisno o podacima poput zastoja prometa i vremenskim uvjetima. Uređaji za kontrolu temperature mogu se koristiti za nadzor robe poput hrane i pića, cvijeća, zdravstvenih proizvoda, kako bi se ostvarili optimalni uvjeti. Na slici 2.1 možemo vidjeti primjer korištenja interneta stvari u kamionu [2, 3].



Slika 2.1: Korištenje IoT uređaja u transportu [4]

Maloprodajne tvrtke koriste internet stvari kako bi pratile stanje inventara na policama i u skladištu, praćenje kretnje kupaca kako bi se optimizirao raspored trgovine i poboljšalo iskustvo kupaca optimiziranjem rasporeda u trgovini. Slika 2.2 prikazuje korištenje interneta stvari u trgovini [2, 3].



Slika 2.2: Pametna trgovina[5]

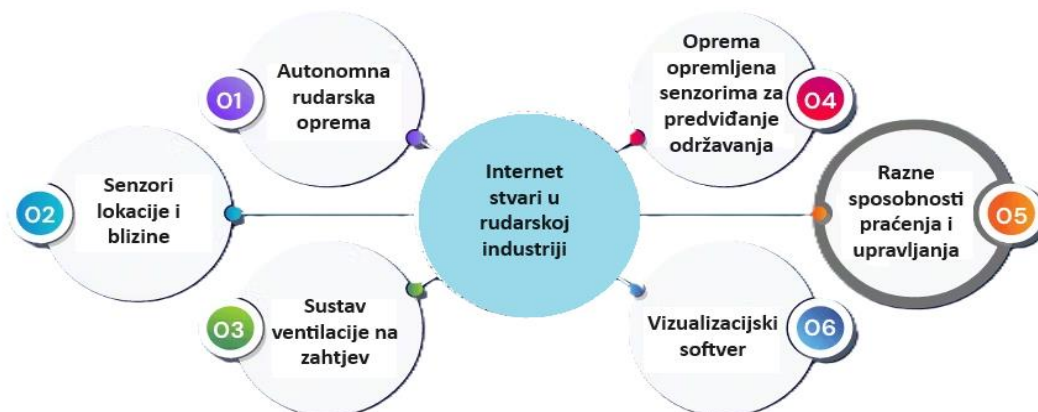
Uređaji za praćenje razne opreme u zdravstvenom sektoru omogućavaju brže pružanje zdravstvene njege. Također uređaji interneta stvari mogu se koristiti za praćenje i prikupljanje podataka o vitalnim znakovima pacijenata poput pulsa srca, razine glukoze u

stvarnom vremenu. Praćenjem dobivamo obrazac kojim možemo otkriti potencijalne zdravstvene probleme prije nego do njih dođe. Na slici 2.3 prikazano je korištenje uređaja interneta stvari u zdravstvu [2, 3].



Slika 2.3: Pametno liječenje [6]

Generalna industrijska sigurnost također može biti poboljšana korištenjem uređaja interneta stvari. Zaposlenici koji rade u opasnim okruženjima poput kemijskih i energetske postrojenja, rudnika, naftnih i plinskih polja mogu saznati o opasnim događajima koji bi mogli imati negativan utjecaj na njih. Primjer toga može biti curenje plina, kojeg senzor može detektirati. Na slici 2.4 prikazane su svrhe korištenja interneta stvari u rudarskoj industriji [2, 3].



Slika 2.4: Povećanje mogućnosti u rudarskoj industriji [7]

Automobilaska industrija također ima značajne prednosti u korištenju uređaja interneta stvari. Senzori za detektiranje kvarova se koriste tijekom proizvodnje ili vožnje

automobila te detaljima o kvaru obavještavaju proizvođača ili vozača čak i prije nego što dođe do kvara. Novija vozila povezivanjem i komunikacijom s raznim uređajima imaju sposobnost autonomne vožnje. Na slici 2.5 prikazano je korištenje interneta stvari u automobilskoj industriji kako bi vozilo komuniciralo s objektima u okruženju [8].



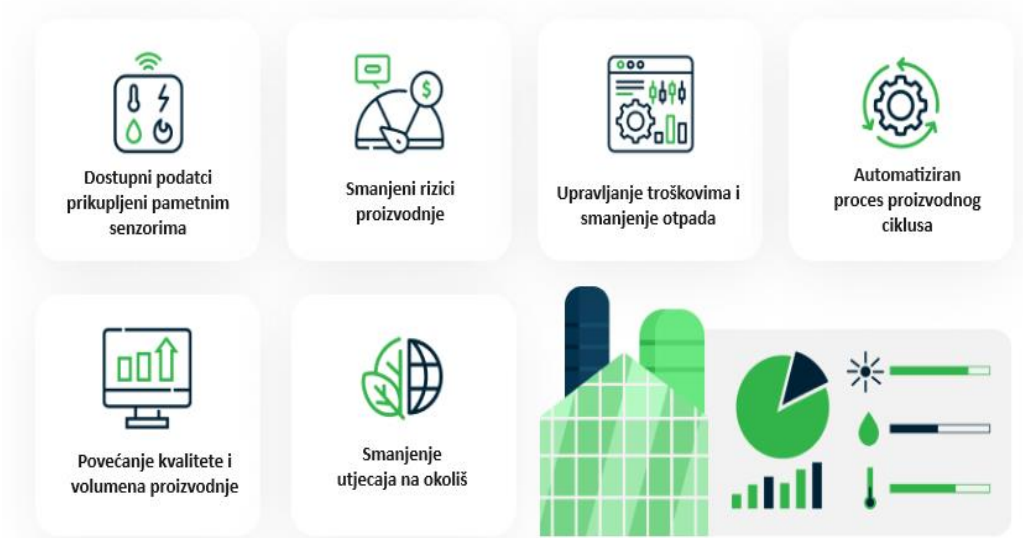
*Slika 2.5: Komunikacija automobila s okolinom [9]*

Javni sektor beneficira tako da ih uređaji mogu obavijestiti o prekidima rada vodenih, strujnih ili kanalizacionih usluga kako bi se takve situacije što prije razriješile. Internet stvari se također koriste za izradu pametnih gradova i zajedno s umjetnom inteligencijom analiziraju podatke i informacije te ih spremaju na servere. Tako pametni gradovi upravljaju rasvjetom, prate kvalitetu zraka i vode, odlučuju gospodarenjem otpada i mnogih drugih stvari kako bi poboljšali energetske učinkovitost i smanjili ovisnost o ne obnovljivim izvorima energije. Raznolikost primjene interneta stvari u javnom sektoru prikazana je na slici 2.6 [10].



*Slika 2.6: IoT uređaji u pametnom gradu [11]*

Internet stvari oblikuju agrikulturu prikupljanjem velike količine podataka kao što su vremenski uvjeti, kvaliteta tla, bilježenje napretka rasta usjeva i slično. Također korištenjem pametnih uređaja omogućava se veća kontrola i povećava standard kvalitete usjeva, osim toga mogu se koristiti i za nadzor životinja i njihovog zdravlja te izdvojiti bolesne životinje od zdravih kako bi zaustavili širenje zaraze. Slika 2.7 prikazuje korisnost interneta stvari u agrikulturi [12].



Slika 2.7: Pobljšanja u agrikulturi koristeći IoT sustave [12]

### **3. RAZVOJNE OKOLINE**

U današnjem svijetu gdje postoji veliki interes za tehnologiju i razvoj sustava poput interneta stvari radi olakšavanja ljudske svakodnevnice, također postoji i mnogo razvojnih okolina kako bi se olakšao rad ljudima koji se bave izradom takvih sustava. U sljedećim odlomcima navest ćemo nekoliko takvih okolina te navesti njihove prednosti i mane.

#### **3.1 Arduino razvojna okolina**

Arduino je platforma osmišljena za jednostavno i fleksibilno kreiranje elektroničkih prototipova. Ključan je alat za razvoj interaktivnih objekata i okruženja te ga koriste djeca, hobisti, umjetnici i najčešće programeri. Upravo ta raznolikost ljudi koji ga koriste, omogućila je napredak Arduina. Glavne prednosti Arduino okoline u usporedbi s drugim sličnim platformama su jednostavno korištenje koje omogućuju brojne upute korištenja koje su detaljno opisane kako bi ju mogli koristiti amateri, ali je i dovoljno fleksibilna za profesionalno korištenje. Što je više korisnika, više je i povratnih informacija koje omogućuju novitete pa samim time i poboljšanje potrebama korisnika i tržišta. Što se tiče samog funkcioniranja Arduino hardware i software dijela, također se prije svega izdvaja jednostavnost i dostupnost. Arduino pločice funkcioniraju na principu ulazne i izlazne informacije. Na primjer, to znači da je moguće da pločica učita svjetlo na senzoru, otisak prsta na zaslonu ili pak primitak poruke, a istu će tu ulaznu informaciju iskoristiti za reakciju kroz primjerice automatsku aktivaciju zaslona, svjetlosnu obavijest ili čak radnju poput automatske objave na društvenoj mreži[13].

##### **3.1.1 Arduino komponente**

Arduino svojim korisnicima nudi kompletne potrebne komponente za rad i to čak 3 vrste: Arduino Starter Kit, Arduino Oplà IoT Kit te Arduino Sensor kit. Za sami početnički komplet pripremljena je pločica UNO, elektroničke komponente te priručnik koji u 15 poglavlja objašnjava i pomaže korisniku u radu s ovim kompletom. Na slici 3.1. su prikazani navedeni kompleti.



a) *Arduino Starter*



b) *Arduino Oplà*



c) *Arduino Sensor*

*Slika 3.1: Arduino razvojna okruženja [14]*

Arduino je do sada proizveo preko 100 hardverskih proizvoda kao što su navedeni kompleti, pločice, pokretači, nosači te brojne druge komponente. Najpoznatije vrste Arduino hardverskih proizvoda su iz reda Nano, MKR i Classic. Na slikama 3.2, 3.3 i 3.4 možemo vidjeti Arduino pločice.



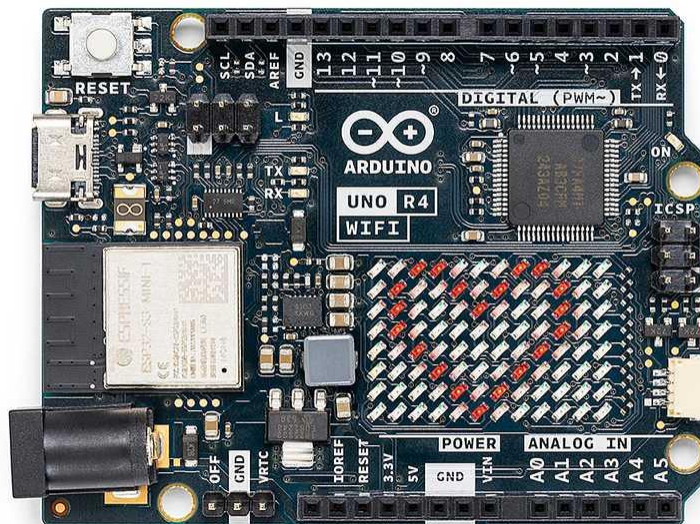
*Slika 3.2: Arduino Nano razvojna pločica [14]*

Najpopularnija pločica skupine Nano svakako je Arduino Nano 33 IoT koja se ističe jednostavnošću i niskom cijenom. Glavni dio svakako je SAMD21 mikrokontroler, mogućnost Wi-Fi i Bluetooth povezivanja uz pomoć u-blox, NINA-W10 modula.



*Slika 3.3: Arduino MKR razvojna pločica [14]*

Sve pločice skupine MKR opremljene su Cortex-M0 32-bit SAMD21 procesorom. Najčešće korištena pločica, jednostavna i cjenovno prihvatljiva, je Arduino MKR Wi-Fi 1010 pločica. Omogućava Wi-Fi povezivanje iz ureda ili kuće, ali i Bluetooth povezivanje.



Slika 3.4: Arduino Uno R4 razvojna pločica [14]

Među ponudom skupine Classic pločica svakako najpoznatija je Arduino UNO, a potom i Leonardo i Mikro pločice. Ove pločice dio su brojnih uspješnih projekata već godinama.

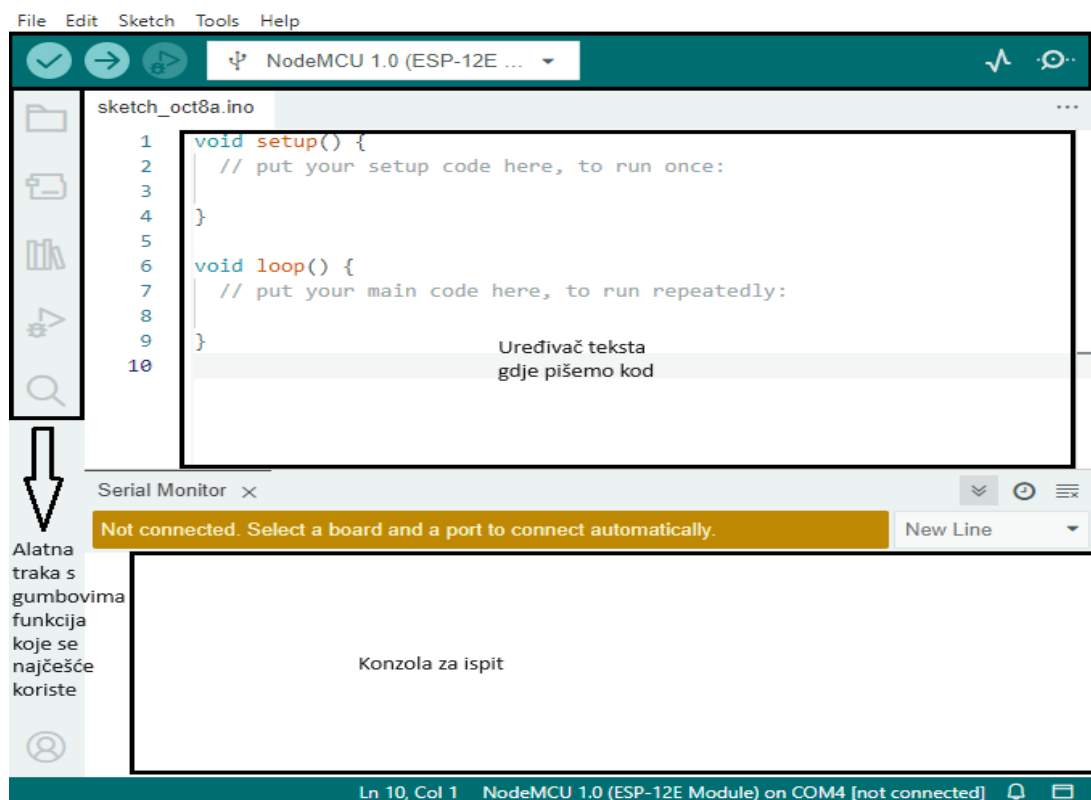
Kod Arduino UNO pločica svakako treba spomenuti Arduino UNO R4 Wi-Fi koji je čista kombinacija snage i mogućnosti, ali kombinacije noviteta uz RA4M1 i ESP32-S3 mikroprocesorima. Uz ugrađene mogućnosti korištenja Wi-Fi i Bluetooth tehnologija, omogućava unaprjeđenje postojećih projekata kroz bežično povezivanje, ali uz to pokriva sve potrebe novog projekta od početka do kraja. Također na samoj pločici integriran je 12x8 matrix LED (engl. *Light Emitting Diode*) zaslon na kojem se mogu ispisati razni podaci [14].

### 3.1.2 Arduino integrirana razvojna okolina

Arduino je programska razvojna okolina otvorenog izvora, što znači da je potpuno besplatna. Softver je moguće pokrenuti na operacijskim sustavima Windows, Linux i Mac, ali se također može koristiti i kao online web editor koji ne zahtjeva dodatnu instalaciju razvojne okoline, no zahtjeva instalaciju Arduino razvojnog agenta. Dizajniran je za lagano



povezivanje hardvera i softvera te je vrlo jednostavan za korištenje. Jedna od prednosti je ta što je Arduino programski jezik baziran na C++ programskom jeziku i stoga ga je vrlo jednostavno za naučiti. Samom instalacijom Arduino softvera ali i instaliranjem dodatnih razvojnih ploča putem upravitelja pločama dobivamo i jako dobre primjere koda gdje je svaka funkcionalnost objašnjena u komentarima. No kao što sve ima svoje prednosti tako ima i svoje mane, naime ovom integriranom programskom okruženju fali funkcionalnost poput automatskog dovršavanja koda, ne podržava jezike poput Pythona, Jave ili Rusta koji bi bili primjereniji za izradu kompleksnijih rješenja. Pokretanjem okruženja može se primijetiti što sve čini Arduino integrirano razvojno okruženje idealnim za razvoj. Sučelje sadrži uređivač teksta, gdje se može pisati kod, područje za poruke, konzolu za ispis, alatnu traku s gumbovima za funkcije koje se najčešće koriste te razne druge postavke u glavnoj traci. Na slici 3.5 prikazano je Arduino korisničko sučelje[15].

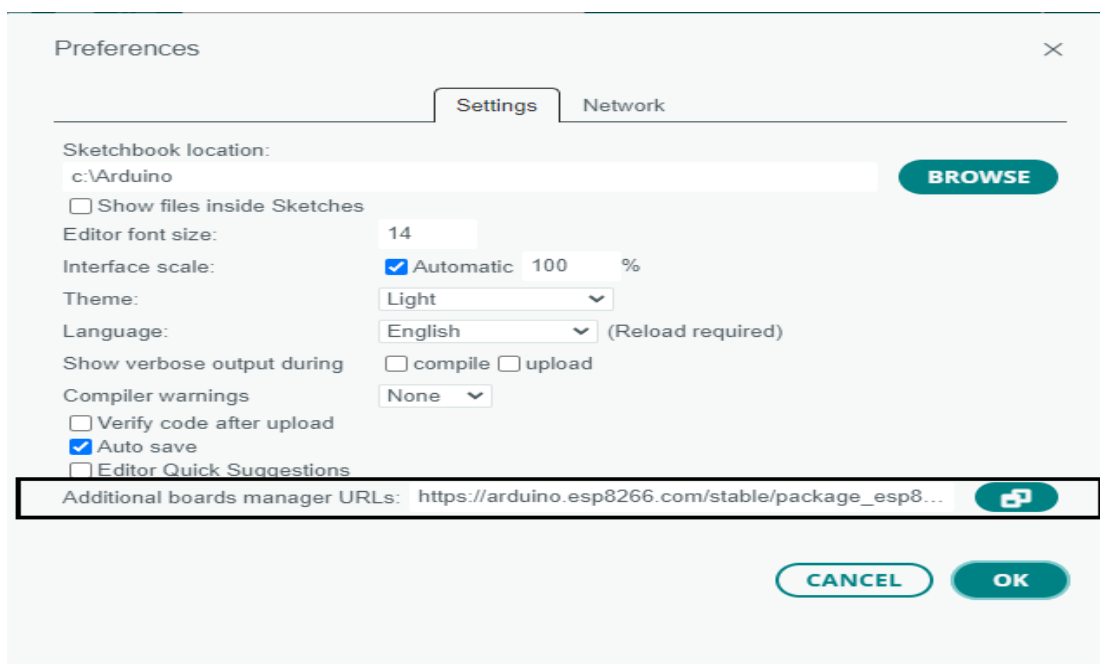


Slika 3.5: Prikaz Arduino korisničkog sučelja

Programski kod se piše u takozvanim „skicama“ koje imaju ekstenziju to jest tip datoteke .ino, iako je programski jezik koji se koristi za razvoj C i C++. Kako bi se napisani programski kod zapisao na mikroračunalo, u padajućem izborniku treba odabrati razvojnu ploču te pritisnuti gumb „Upload“ koji će automatski napisani kod prevesti u

strojni. Ukoliko je okruženje pravilno namješteno i u programskom kodu nema grešaka, kod će se uspješno zapisati u fleš memoriju mikroračunala, u protivnom će se u konzoli crvenom bojom dobiti ispisana greška s njezinim detaljima. Za prenošenje koda koristi se UART (engl. *Universal asynchronous reciever-transmitter*) protokol.

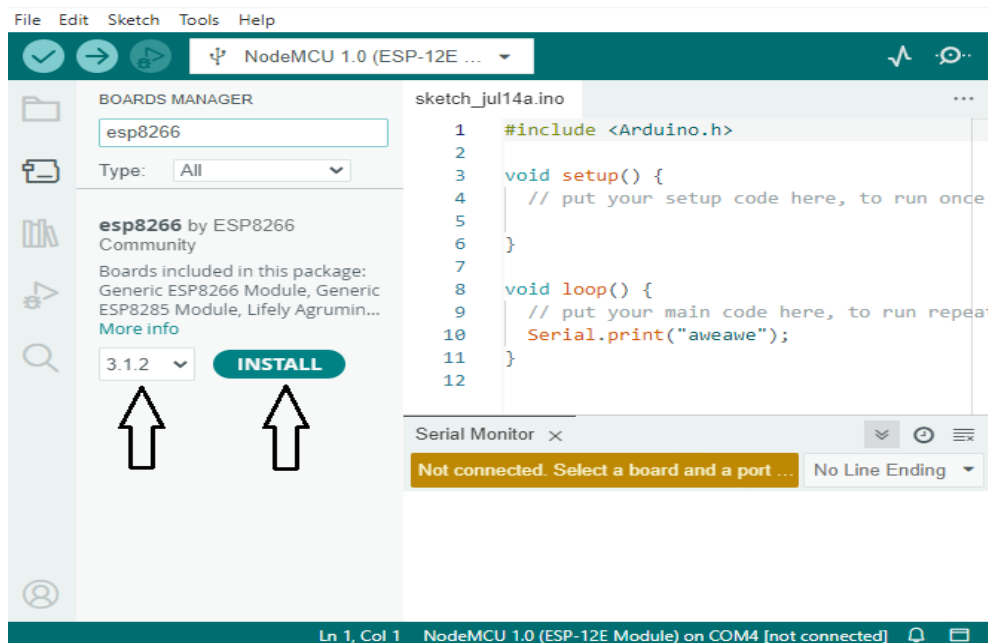
U Arduino integriranom programskom okruženju Arduino AVR pločice nemaju potrebe za dodatnom instalacijom i konfiguracijom, no kako postoji mnoštvo razvojnih pločica, neke od njih zahtijevaju dodatnu instalaciju jezgre i potencijalno njezinih biblioteka. Implementacijom funkcionalnosti upravitelja pločicama i upravitelja bibliotekama postiže se kompatibilnost s razvojnim okruženjem. Klikom na „file“ padajući izbornik u glavnoj traci, te klikom na „preferences“ dobiva se novi prozor u kojem se mogu konfigurirati razne postavke, no ono što je najbitnije nalazi se pod nazivom „Additional boards manager URLs“. Kako bi se razvojna pločica treće strane mogla instalirati, u ulazno polje moramo zalijepiti poveznicu koja vodi do JSON datoteke. Navedeno možemo vidjeti na slici 3.6.



Slika 3.6: Arduino upravitelj dodatnim pločicama

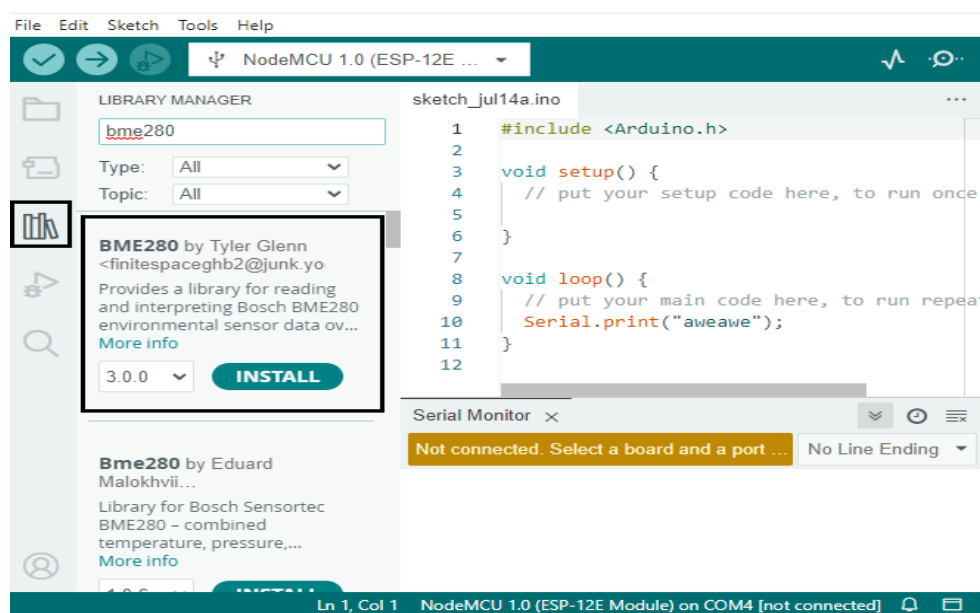
Nakon unesenog URL-a dodatne razvojne pločice i spremanjem preferenci pritiskom tipke ok, na alatnoj traci s gumbovima za najčešće korištene funkcije pritiskom na drugu ikonicu odozgo ulazi se u upravitelj pločica. U ulazno polje napisati naziv dodatne pločice, odabrati željenu verziju i pritisnuti gumb za instalaciju s čime je dodana dodatna jezgra,

nakon čega se može započeti programiranjem razvojne pločice. Instalacija je prikazana na slici 3.7.



Slika 3.7: Instalacija dodatne razvojne pločice

Koristeći upravitelj bibliotekama mogu se instalirati dodatne biblioteke koje donose razne prednosti te olakšavaju rad sa senzorima, aktuatorima i brojnim drugim pločicama.



Slika 3.8: Instalacija dodatnih biblioteki

## 3.2 Raspberry Pi

Raspberry Pi je serija malih računala koju je razvila istoimena zaklada u Ujedinjenom Kraljevstvu. Radi se o seriji koja koristi jednu pločicu, a prvotno je zamišljena za pomoć pri nastavi informatike u školama te unaprjeđivanju znanja i mogućnosti. Na tržištu je od 2012. godine. I Arduino i Raspberry Pi među prvim su izborima kako onih koji rade projekte iz hobija, pa tako i profesionalaca. Svaki pri tome ima svoje prednosti i nedostatke i odabiru između njih uglavnom će presuditi svrha za koju trebaju poslužiti.

Za početak, važno je razumjeti osnovnu razliku, a to je da se Arduino ploče gledaju kao mikrokontroleri, a ne potpuna računala, dok se kod Raspberry Pi ploča radi o mikroprocesorima s vlastitim operacijskim sustavom. Arduino ploče djeluju na principu prepoznavanja neke jednostavnije ulazne informacije koju će u skladu s kodom registrirati i na nju automatski reagirati. Pri tome se, dakle, radi o nekim jednostavnijim radnjama i to ograničenom broju istih. S druge strane, Raspberry Pi može istovremeno pokrivati i više naredbi, ali i one kompleksnije. Pri tome se ističe i još jedna bitna razlika, a to je da je Raspberry Pi direktno povezan na internet i na taj način može ažurirati napredak u radu, dok se Arduino pločica mora spojiti dodatnim uređajima kako bi se povezala s internetom.

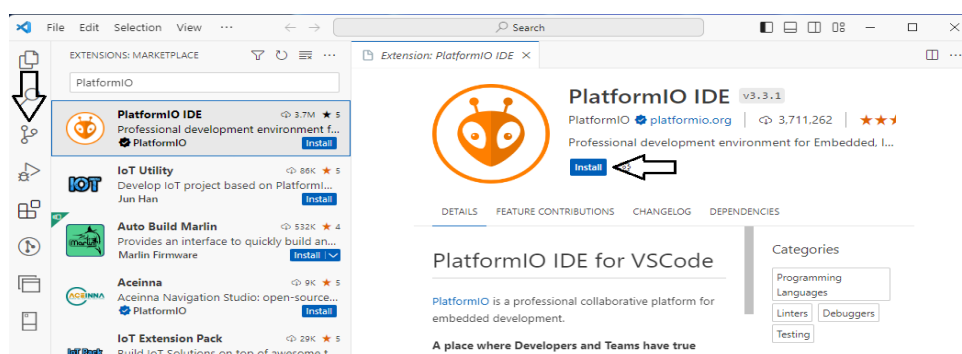
Arduino je otvorenog tipa, odnosno, i hardverski i softverski dizajn je otvorenog tipa i dostupan svima, dok se kod Raspberry Pi radi o zatvorenom tipu koda. Ono što, također, može biti presudno je i cijena, pa tako slobodno možemo reći da je Arduino cjenovno jeftiniji, dok se kod Raspberry Pi mikroprocesora radi o dosta većim cijenama. Raspberry Pi svojom znatno većom memorijom i snagom procesora dakako više odgovara zahtjevnijim projektima, ali pri tome troši znatno veću količinu energije za razliku od Arduina.

Ukratko se može reći da se odabir zasniva na potrebama korisnika i ideji za projekt. Usporedba se može svesti na nekoliko jednostavnih prekretnica u odabiru, a to su cijena, kompleksnost (mogućnost izvršavanja više radnji istovremeno kod Raspberry Pi), utrošak energije, snaga, memorija. Na slici 3.9. prikazana je razlika između razvojnih okolina Raspberry Pi i Arduino razvojnih pločica[16].



Slika 3.9: Razlika između razvojnih okolina Arduino i Raspberry Pi [16]

Za razvijanje Raspberry Pi pločice se može koristiti PlatformIO, vrhunski besplatni alat otvorenog izvora licenciran od strane MIT-a koji je moguće instalirati na raznim platformama. Najčešće se koristi kao dodatak Microsoftovom uređivaču teksta Visual Studio Codu koji pruža najbolje performanse te je većini korisnika najjednostavniji za korištenje, obzirom na to da podržava jako puno programskih jezika te postoji velika mogućnost da ga se koristi i za neki drugi razvoj. Koriste ga mnogi ljudi diljem svijeta, na što ukazuju ocjene i broj preuzimanja ekstenzije kao što je prikazano na slici 3.10.

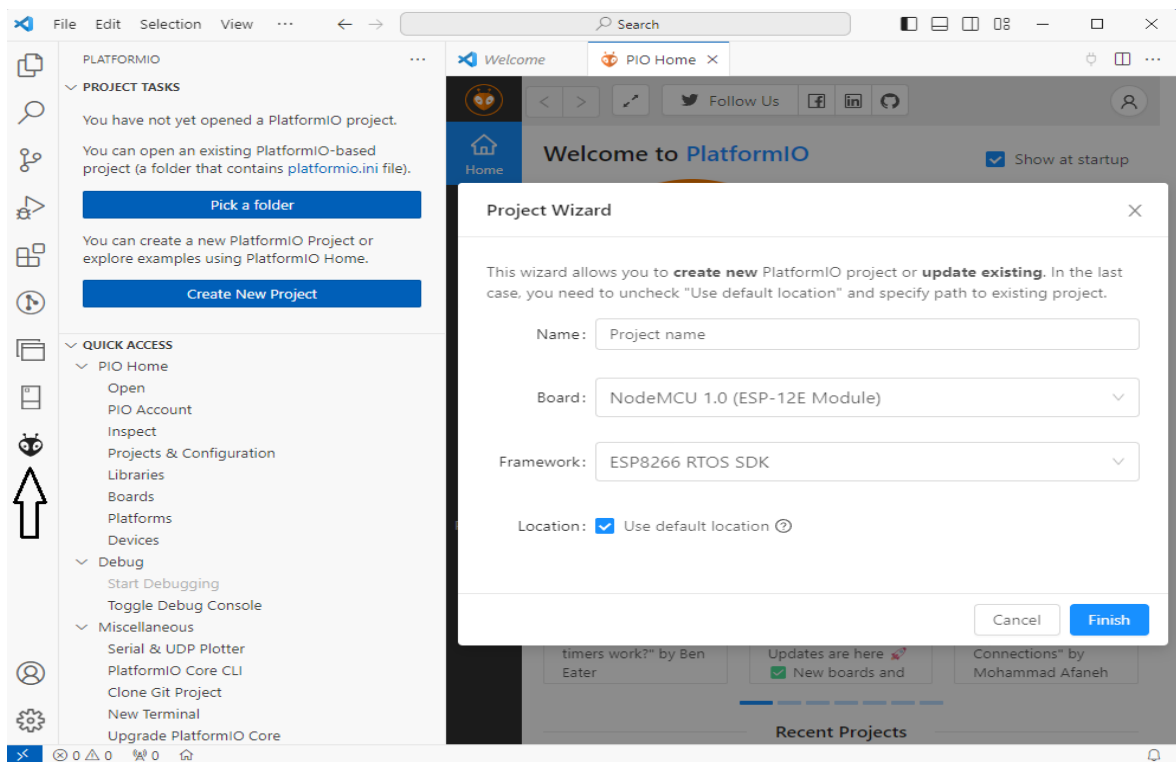


Slika 3.10: Instalacija PlatformIO dodatka u Visual Studio Codu

PlatformIO možemo instalirati kao dodatak mnogim drugim popularnim razvojnim okruženjima i uređivačima teksta.

### 3.2.1 Čarobnjak za izradu projekta

Instaliravanjem dodatka PlatformIO unutar Visual Studio Coda može se početi s razvijanjem projekta. Pritiskom na ikonu dodatka, odabirom padajućeg izbornika PIO Home i njegove stavke Open, dobije se prozor gdje se pomoću čarobnjaka može izraditi projekt. U prozoru se može unijeti naziv projekta, odabrati više od tisuću ploča koje se mogu koristiti za razvoj projekta te okvirnik koji želimo koristiti. Navedeno je prikazano na slici 3.11.

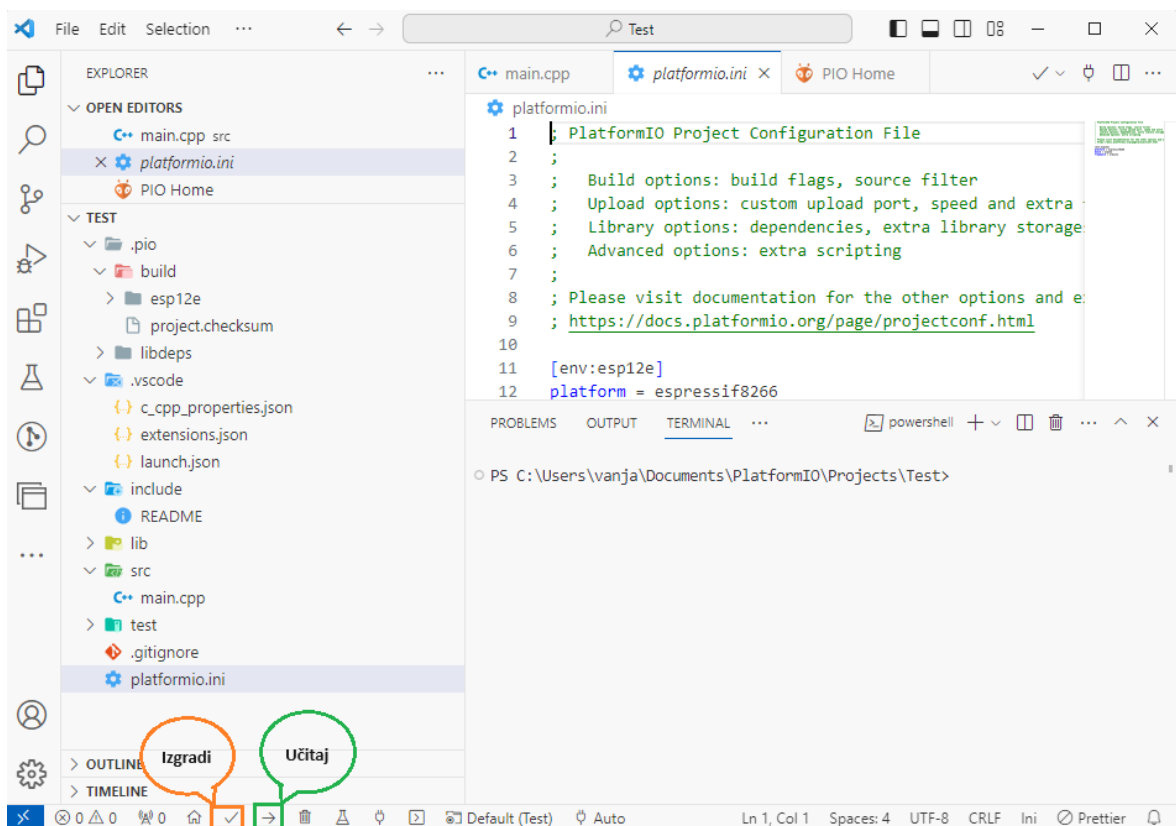


Slika 3.11: Kreiranje projekta pomoću čarobnjaka

Nakon unosa željenih postavki, preuzeti će se sve potrebno razvijanje projekta, nakon čega možemo početi pisati kod. Ovdje možemo vidjeti razliku u odnosu na korištenje Arduina, gdje mnoge pločice moramo dodavati sami, također kod u PlatformIO se piše u datoteci s nastavkom .ccp.

### 3.2.2 Prednosti razvojnog okruženja

Ideja ovog integriranog razvojnog okruženja je programerima i sistemskim inženjerima pružiti moderno okruženje koje podržava profinjeno otklanjanje grešaka, jedinično testiranje, automatsku analizu koda i udaljen razvoj. Podržava brojne biblioteke i okvirkne što dodatno olakšava proces razvoja projekata korištenjem ugrađenih funkcionalnosti. Za razliku od Arduina, traka s najčešće korištenim funkcijama nalazi se na samom dnu kao što možemo vidjeti na slici 3.12.



Slika 3.12: Alatna traka s često korištenim funkcijama

Ovaj moćni alat, za razliku od Arduina posjeduje inteligentan sustav koji funkcionira tijekom pisanja C i C++ koda, naime sustav može prepoznati vaše namjere te predložiti automatsko dovršavanje koda. Također, prepoznaje greške i predlaže akcije koje bi se mogle poduzeti za njihovo ispravljanje. Okruženje ima ugrađeno sučelje naredbenog retka koje je napisano u Pythonu te je ono baza cijelog ekosustava. Obzirom na to, uz instalaciju dodatka moramo instalirati i programski jezik Python koji se također mora dodati u varijable puta operacijskog sustava. Sučelje naredbenog retka može se instalirati i bez

korištenja integriranog razvojnog okruženja, stoga ga možemo koristiti i na uređajima poput Raspberry Pi, BeagleBone i mnogim drugima[17]. Logo programskog jezika Python možemo vidjeti na slici 3.13



*Slika 3.13: Logo programskog jezika Python[18]*



## 4. MIKROKONTROLERI (MCU)

Mikrokontroler je integrirani krug koji se može programirati za obavljanje specifičnih računalnih operacija pri čemu omogućuje kontrolu nad upravljanjem uređajima i procesima. Svaki mikrokontroler sadrži središnju jedinicu za procesiranje (CPU (engl. *Central Processing Unit*)), memoriju i ulazno/izlazne konektore. Mikrokontroleri imaju mogućnost rada uz minimalan utrošak energije, a osim toga mogu varirati i u veličini što znači da pojedini mikrokontroleri mogu biti smješteni u malene uređaje. Otkriće mikrokontrolera omogućilo je masovniju proizvodnju i jeftinije cijene te je zbog toga od njihove pojave 1970-ih prošlo vrlo malo vremena dok nije postao komercijalno dostupan, kako među proizvođačima tako i među korisnicima. Primjerice, tako su postali vrlo važan dio tehnologije u proizvodnji automobila (brisači vjetrobrana kontrolirani senzorima, kontrola motora i mnogo drugih), osim u automobilskoj industriji mikrokontroleri mogu se pronaći u brojnoj bijeloj tehnici, mobilnim uređajima. Još jedno poboljšanje mikrokontrolera i DIY(engl. *Do It Yourself*) uslijedilo je 1990-ih godina gdje mikrokontroleri dobivaju novu vrstu memorije EEPROM (engl. *Electrically Erasable Programmable Read Only Memory*), koja je riješila problem ograničenja u brisanju i ponovnom zapisivanju koda. Samim time njihova izrada je postala znatno jeftinija. Primjena mikrokontrolera se često pronalazi u sustavima interneta stvari kako bi kontrolirali i nadzirali razne sustave i uređaje[19, 20].

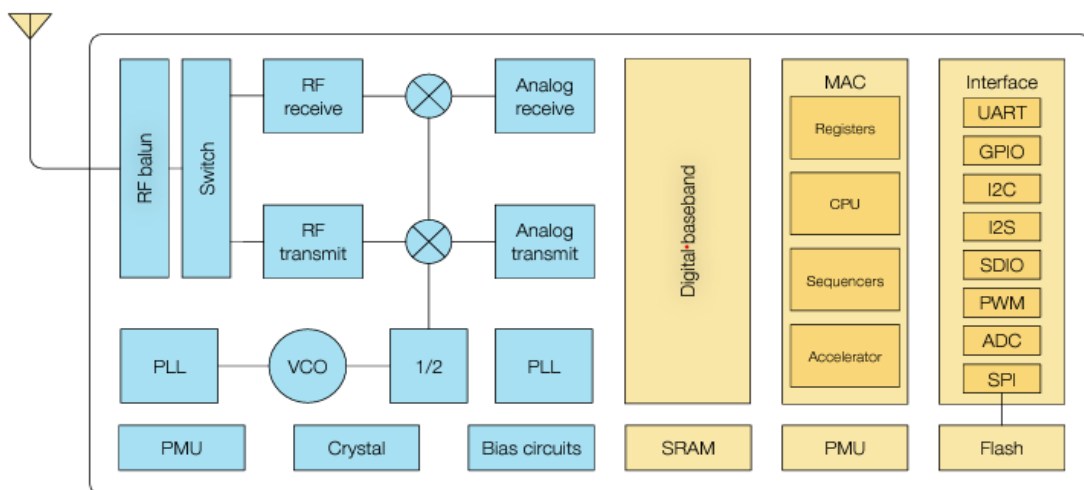
### 4.1 ESP8266

Ovaj vrlo isplativ Wi-Fi mikrokontroler specijaliziran je za korištenje u sustavima interneta stvari u koje se često integrira. Istaknute prednosti ovog mikrokontrolera su zavidna razina izdržljivosti koja je omogućena širokom spektrom radne temperature što ga čini idealnim za korištenje u industrijskim okruženjima. ESP8266 postiže rad uz nisku potrošnju energije omogućenu kroz tri stanja rada, a to su: stanje aktivnosti, spavanja i dubokog spavanja. Zbog ugrađene sposobnosti za Wi-Fi komunikaciju IEEE 802.11 b/g/n standardom može se koristiti zasebno, ali i kao most koji povezuje druge uređaje koji sami po sebi nemaju sposobnost mrežnog povezivanja. Na slici 4.1 možemo vidjeti mikrokontroler ESP8266-12E Wi-Fi.



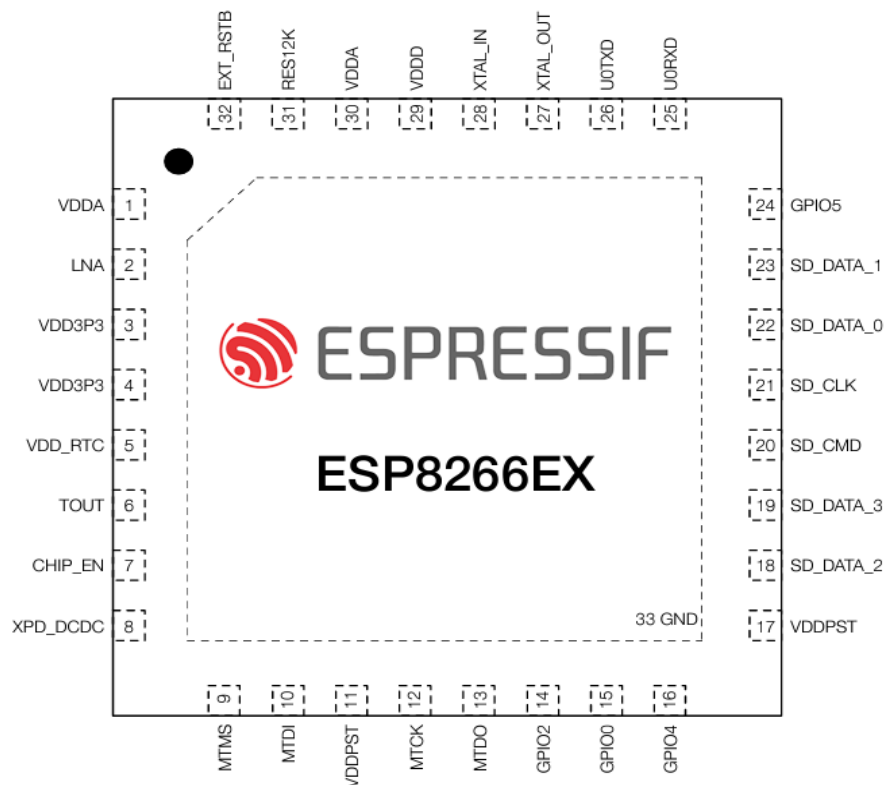
Slika 4.1: ESP8266 Wi-Fi modul [21]

Zbog navedenih sposobnosti mrežnog povezivanja primjena ESP8266 najčešće se može pronaći u kućnoj automatizaciji i mnogim DIY projektima. Procesor L106, baziran na Transilica Diamond Standard 106Micro koristi 32 bitnu RISC (engl. Reduced Instruction Set Computer) arhitekturu te postiže maksimalnu brzinu ciklusa od 160Mhz. RTOS (engl. *Real – Time Operating System*) i Wi-Fi sustav korisnicima omogućuje da tijekom razvoja iskoriste čak oko 80% procesorske snage. U mikrokontroler integrirana je radna memorija koju uključuje: 32KiB (KibiByta) instrukcijske memorije i predmemorije i predmemorije, 80 KiB korisničke memorije i 16KiB sistemske memorije. 80. Slika 4.2 prikazuje funkcionalni dijagram ESP8266EX mikrokontrolera.



Slika 4.2: Funkcionalni dijagram ESP8266EX mikrokontrolera [22]

Ovaj mikrokontroler koristi 17 GPIO (engl. General Purpose Input/Output) pinova koji se različitim komunikacijskim protokolima kao što su SDIO (engl. Secure Digital Input/Output), SPI (engl. Serial Peripheral Interface), I<sup>2</sup>C (engl. Inter - Integrated Circuit), I<sup>2</sup>S (engl. Inter - Integrated Sound), UART, PWM (engl. Pulse-Width Modulation), IR (engl. Infra Red) i ADC (engl. Analog-to-Digital Converter) mogu koristiti za dodatnu pohranu podataka, povezivanje motora, raznih zaslona, senzora, aktuatora i mnogih drugih uređaja. Raspored svih pinova iz ptičje perspektive može se vidjeti na slici 4.3 [21, 22].



Slika 4.3: Raspored pinova ESP8266EX [22]

## 4.2 Nordic Semiconductor nRF52 Series

Seriya ovih naprednih i visokokvalitetnih mikrokontrolera koristi Arm Cortex-M4 procesor frekvencije 64 MHz. Svi modeli ove serije podržavaju UART, SPI i TWI protokole, dok u onim naprednijima također možemo naći SPI, PWM, I2S i slične. Možemo ih pronaći u različitim primjenama ali se najčešće koriste u ugradbenim sustavima i potrošačkoj elektronici poput naprednih nosivih uređaja, pametnih kućnih uređaja, profesionalnoj rasvjeti, osobnim uređajima sa svrhom praćenja vježbi i osobnog

zdravlja, udaljenu kontrolu uređaja i slično. Glavne značajke serije ovih mikrokontrolera su Bluetooth Low energy, Bluetooth 5, Bluetooth Mesh, Thread, Zigbee, ANT (engl. *Advanced and Adaptive Network Technology*). Za nRF52 seriju možemo koristiti nRF5 SDK (engl. *Software Development Kit*) i nRF Connect SDK alate za razvoj. Također, koristi NFC (engl. *Near Field Communication*) tehnologiju za buđenje uređaja iz stanja sna ili za sigurnosno povezivanje.

#### 4.2.1 Bluetooth Low Energy

Ova tehnologija iznimno je bitna za uređaje koji koriste mikrokontroler iz serije nRF5 2 stoga što troši jako malo energije tijekom prijenosa podataka, dok pokušava održati sličan domet za komunikaciju. Ovaj bežični protokol operira na frekvenciji od 2.4 GHz s propusnošću do 1.4 Mbps, no pojavom tehnologije Bluetooth 5.2 najveća moguća propusnost povećana je na 2 Mbps. Ova tehnologija je iznimno sigurna jer koristi razne značajke kako bi osigurala privatnost, povjerljivost i integritet.

#### 4.2.2 Zigbee

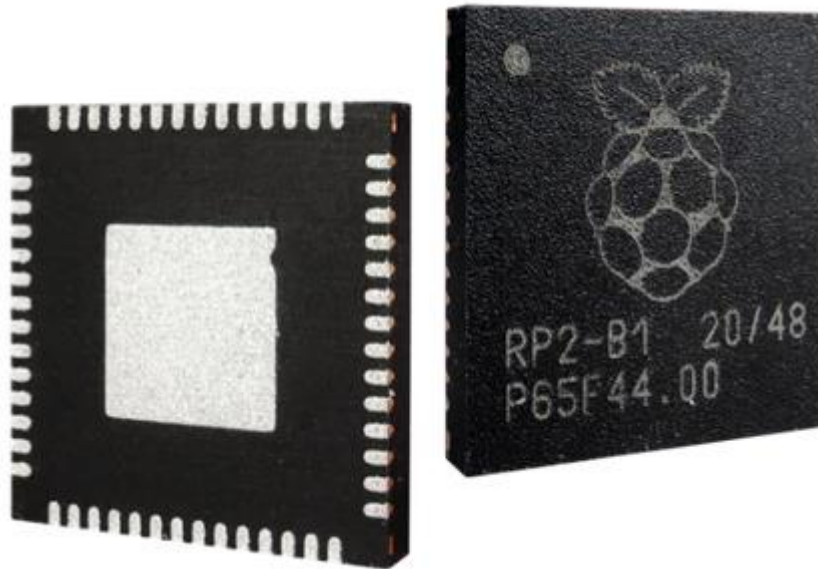
Zigbee je komunikacijski protokol koji se koristi za niskoenergetsku komunikaciju unutar osobne mreže. Koristi se za rješenja u pametnim kućama, nadziranje zgrada i ostalim rješenjima koja mogu koristiti beneficije niske propusnosti. Niska potrošnja ovog protokola ograničava domet odašiljanja na 10 do 100 metara u liniji pogleda, a domet ovisi o karakteristikama okruženja u kojem se uređaj nalazi. Domet se može produžiti korištenjem mesh umrežavanja. Najčešće se koristi u rješenjima gdje je bitna stavka dugoročnost baterije i sigurno umrežavanje[23]. Slika 4.4 prikazuje mikrokontroler nRF52 serije.



Slika 4.4: Mikrokontroler nRF52 [24]

### 4.3 RP2040

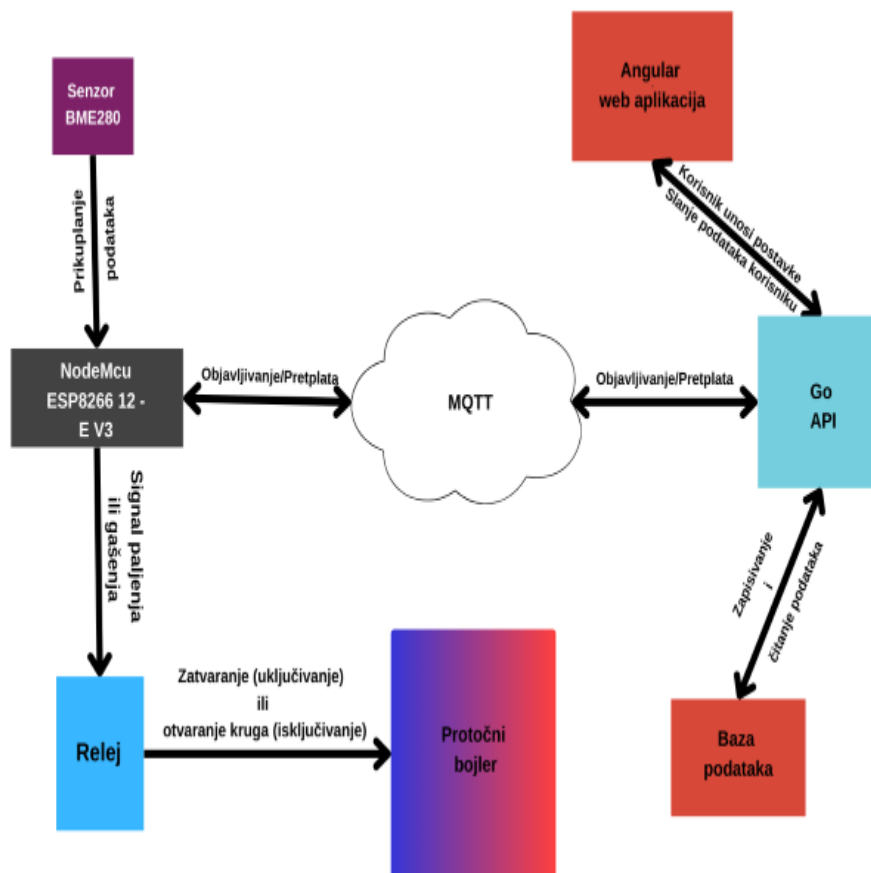
RP2040 je mikrokontroler razvijen od tvrtke Raspberry Pi kojeg krasi karakteristike poput visokih performansi, malog troška energije i jednostavnog korištenja. Slika 4.5 prikazuje navedeni mikrokontroler.



Slika 4.5: Mikroprocesor RP2040[25]

Ključne specifikacije ovog mikrokontrolera su dvojezgreni procesor Dual ARM Cortex-M0+ koji radi na frekvenciji od 133MHz, 264KB SRAM(engl. *Static Random Access Memory*) integrirane memorije smještene u 6 različitih banki kako bi se omogućilo paralelno čitanje, 30 multifunkcionalnih GPIO od kojih 6 služe za podršku za vanjsku memoriju do 16MB, hardver za često korištenu periferiju, USB(engl. *Universal Serial Bus*) kontroler i programabilne IO(engl. *Input/Output*) za dodatnu periferiju. RP2040 također ima integrirani senzor za mjerenje temperature koji mora biti kalibriran od strane korisnika. Ovaj mikrokontroler se može naći na razvojnim pločicama Raspberry Pi Pico [25].

## 5. IoT sustav za upravljanje centralnog grijanja



Slika 5.1: Blok diagram izrađenog sustava centralnog grijanja

Slika 5.1 prikazuje blok shemu funkcioniranja izrađenog rješenja. U sljedećim poglavljima biti će navedene komponente koje su korištene za izradu hardverskog rješenja te njihova povezanost softverskim rješenjem kojima je zajednička svrha upravljanje sustavom centralnog grijanja. Za upravljanje cijelim sustavom odabrana je varijanta, u poglavlju 4.1. opisanog mikrokontrolera, ESP - 12E. Ovo je varijanta ESP8266 mikrokontrola koja u sebi ima ugrađen Wi-Fi modul, navedeni modul je integriran u Lolin NodeMCU 12 – E V3 razvojnu pločicu kojom je omogućeno lakše korištenje, razvijanje i povezivanje s ostatkom hardverskih komponenti. Jedna od tih komponenti je kompaktni senzor visoke kvalitete i niske potrošnje BME280 koji se nalazi na pločici GY-BME280 radi lakše integracije u sam projekt te prikuplja podatke o temperaturi, vlazi i atmosferskom pritisku. Jedna od najbitnijih stvari u svakom IoT projektu je upravljanje jednostavnim uređajima poput žarulja, ventilatora, ali i onih kompleksnijih poput sustava

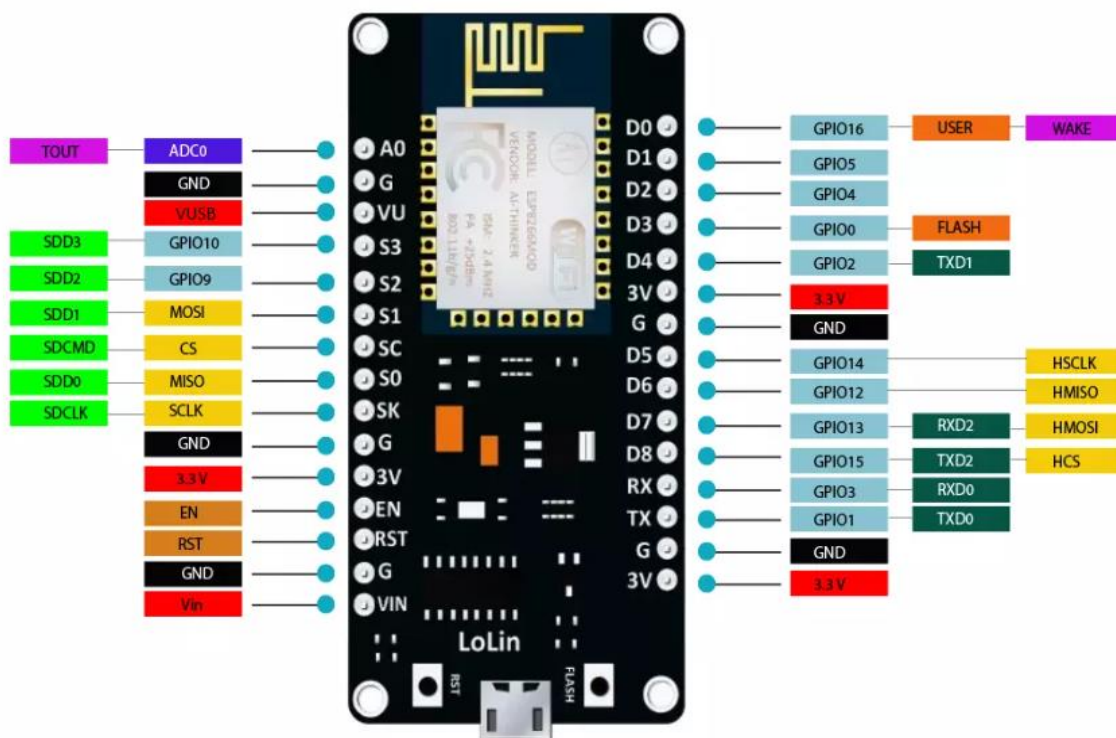
za centralno grijanje. U izrađenom projektu, za kontroliranje uključivanja i isključivanja koristi se relej SRD-05VDC-SL-C koji ovisno o danom signalu otvara ili zatvara strujni krug protočnog bojlera. Kako hardver sam po sebi nema primjenu bez dodatnog softverskog rješenja, navedeni će biti i slojevi programskog rješenja. Pločice su programirane u integriranom razvojnom okruženju Arduino koje je opisano u poglavlju 3. Kako je okolinu potrebno namjestiti, ESP8266 potrebno je dodati kao dodatnu razvojnu pločicu i instalirati ju putem upravitelja pločama. Također, potrebno je instalirati biblioteke Adafruit BME280 i MQTT Client koje omogućavaju lakšu izradu programskog koda. MQTT protokol koji je standard slanja poruka za uređaje interneta stvari. Kako su resursi na ESP8266 mikrokontroleru ograničeni, on je zadužen za jednostavne stvari poput spajanja na mrežu, očitavanja i objavljivanja podataka, dok je za kompleksniju obradu zadužen API (engl. Application Programming Interface), napisan u programskom jeziku Go. Pozadinska aplikacija obrađuje podatke i sprema ih u bazu podataka, šalje podatke na mikrokontroler te odgovara na upite korisničke aplikacije napisane u JavaScript okvirniku Angularu.

## 6. HARDVERSKO RJEŠENJE

Hardversko rješenje sastoji se od pločica Lolin NodeMCU 12 – E V3 koje je povezano s pločicom senzora BME280 te relejom SRD-05VDC-SL-C koji je spojen na protočni bojler i time zatvara strujni krug.

### 6.1 Lolin NodeMCU ESP8266 12 – E V3

Pločica dolazi s integriranim gumbovima za ponovo pokretanje i brisanje memorije. Također ima ugrađen LED indikator koji je povezan na pin D0. Pomoću ugrađenog CH340G USB u UART pretvornika mikrokontroler se može programirati, isto tako pomoću njega podaci se mogu čitati na računalu. Ova pločica sadrži četiri pina za napajanje, tri 3.3V pina koji služe kao izvor napajanja uređajima povezanim na pločicu i jedan VIN (engl. *Voltage Input*) pin koji služi za napajanje pločice. Kako ESP8266 radi na naponu od 2.5 do 3.3V, tako uz nju dolazi i regulator napona koji će napon do 5V spustiti na 3.3V na izlazu. Razvojna pločica ima mnogih drugih prednosti te je stoga jedna od najpopularnijih. Na slici 6.1 vidimo raspored i namjenu pinova razvojne pločice Lolin NodeMCU ESP8266 12 – E V3 [26].

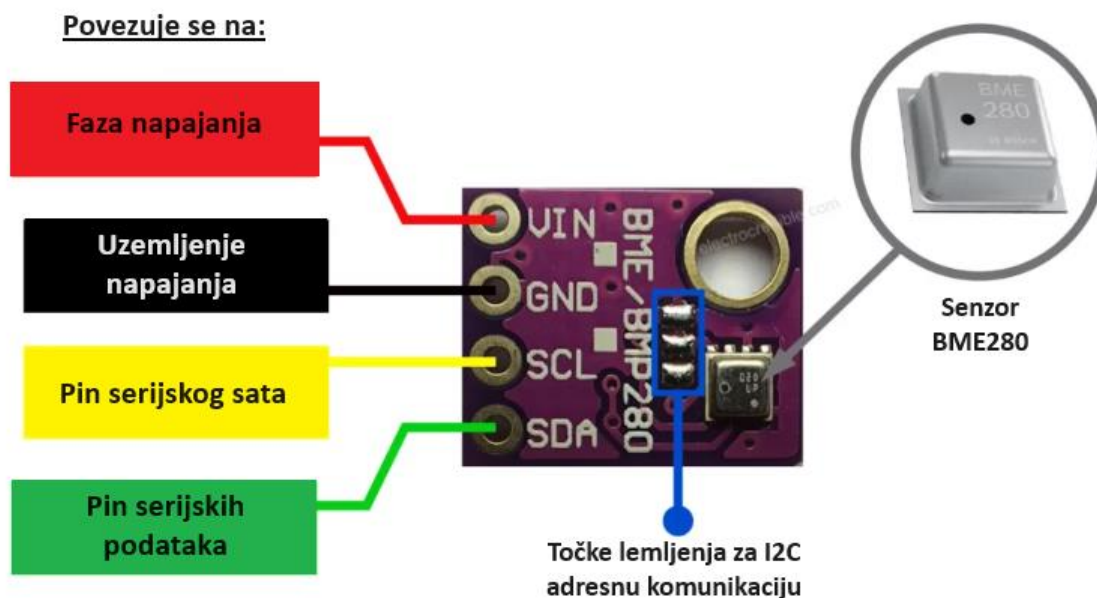


Slika 6.1: Razvojna pločica Lolin NodeMCU 12 – E V3 i njeni pinovi [27]



## 6.2 GY-BME280 modul

GY-BME280 Modul koristi 4 pina za povezivanje s razvojnim pločicama, ti pinovi su VIN (engl. *Voltage Input*), GND (engl. *Ground*), SCL (engl. *Serial Clock*) i SDA (engl. *Serial Data*). VIN i GND pinovi se koriste za napajanje senzora, dok SCL i SDA služe za razmjenu podataka. Modul na sebi koristi senzor BME280 koji je razvijen od strane Njemačke tvrtke Bosch Sensortech. Senzor je napravljen za primjenu u mobilnim, nosivim i uređajima gdje su bitni parametri poput veličine i male potrošnje energije, tako je primjenu našao i u razvoju IoT rješenja. Kako je temperatura najbitniji podatak koji se mjeri u sustavima centralnog grijanja te jedan od podataka na temelju kojeg se odlučuje o uključivanju i isključivanju sustava, odabran je senzor sa sposobnošću mjerenja podataka poput temperature, vlažnosti i atmosferskog pritiska s velikom preciznošću. Temperaturu može mjeriti u rasponu od -40 do 85 °C, vlažnost od 0 do 100 %, pritisak od 300 do 1100 hPa i indirektno može izmjeriti nadmorsku visinu. Na slici 6.2 vidimo prikaz pinova i senzor BME280 integriran na modulu GY-BME280 [28].



Slika 6.2: Svojstva pločice GY-BME280 [28]

### 6.3 SRD-05VDC-SL-C

Releji spadaju u skupinu uređaja koju zovemo aktuatorima. Aktuatori su dijelovi uređaja ili uređaji koji energiju, najčešće električnu, pretvaraju u fizičku silu. Ovaj relej ima 2 načina upravljanja strujnim krugom, normalno otvoreni i normalno zatvoreni koji rade na naponu od 120 do 240V. Relej se nalazi na dodatnoj pločici kako bi se olakšala povezivost s NodeMCU pločicom, koja se ostvaruje pomoću 3 pina. Navedeni pinovi su uzemlje, 5V VCC (engl. *Voltage Common Collector*) koji pločici daje napajanje i signalni pin pomoću kojeg upravljamo zatvaranjem i otvaranjem strujnog kruga. U normalno otvorenom načinu, strujni krug je otvoren sve dok relej ne primi signal visokog napona čime se strujni krug zatvori i ostaje zatvoren sve dok ne primi signal niskog napona. Normalno zatvoreni način funkcionira suprotno od normalno otvorenog. Pločica dolazi s ugrađenim LED diodama koje emitiraju crveno i zeleno svjetlo. Crvena dioda prikazuje da je pločica radi, dok zelena prikazuje je li strujni krug zatvoren. Opisanu pločicu možemo vidjeti na slici 6.3 [29].



Slika 6.3: Pločica releja SRD-05VDC-SL-C

## 7. SOFTVERSKO RJEŠENJE

U ovom poglavlju biti će navedeni slojevi programskog rješenja i njihova primjena. Korisnici upravljaju sustavom grijanja putem Angular web aplikacije gdje mogu vidjeti prikaz nekretnina, njezinih katova, soba i na kraju uređaja koji se u njima nalaze. Korisnik tako može vidjeti učitane podatke sa senzora svakog uređaja, je li uređaj kontrolni uređaj te stanje releja. Za svaki kontrolni uređaj korisnik može postaviti intervale grijanja, točnije vrijeme u danu kada se grijanje uključuje a kada uključuje. Kako svaka stambena jedinica ima svoju fizičku adresu, kada korisnik odabere pojedinu zgradu, može vidjeti i vanjske vremenske uvjete. Nakon što korisnik odabere postavke podaci se šalju na API koji je napisan u Go programskom jeziku gdje se podaci obrađuju, zapisuju ili čitaju iz baze te šalju na mikrokontroler koji pali ili gasi sustav. Mikrokontroler je isprogramiran u programskom jeziku Arduino, zadužen je za učitavanje podataka sa senzora i upravljanje relejom, točnije centralnim grijanjem, također odašilje učitane podatke na API.

### 7.1 Korisnička Angular web aplikacija

Angular je platforma zasnovana na TypeScript programskom jeziku koja omogućava izradu SPA (engl. *Single Page Application*) web stranica. TypeScript je programski jezik baziran na JavaScript jeziku, no nudi nam pogodnosti poput korištenja tipova podataka. Na prikazu koda 7.1 možemo vidjeti primjer TypeScript objekta.

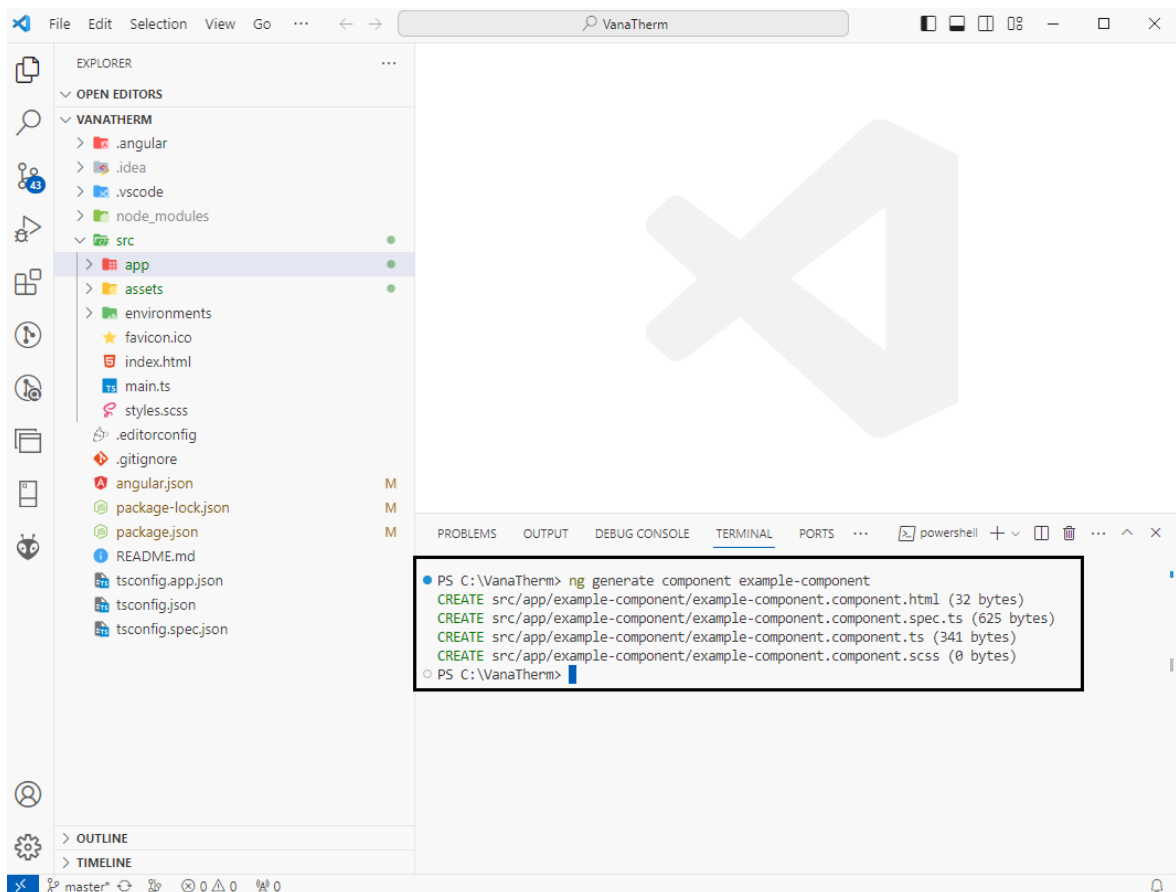
*Programski kod 7.1: Kreiranje interface objekta s TypeScriptom*

---

```
export interface User {
  id?: number;
  firstName?: string;
  lastName?: string;
  gender?: boolean;
  birthDate?: Date;
  username?: string;
  email?: string;
  password?: string;
}
```

---

Ovaj okvirnik baziran je na MVC (engl. *Model-View-Controller*) strukturi te je pomoću brojnih dobro integriranih biblioteka vrlo olakšan razvoj značajki web aplikacija poput navigacije, integracije formi, klijentsko-serverske komunikacije i mnogih drugih. Za funkcioniranje ovog okvira potrebno je instalirati Node.js pomoću kojeg Angular učitava sve potrebne ovisnosti poput CSS (engl. *Cascade Style Sheet*) i JavaScript datoteka. Dolazi s vrlo moćnim CLI (engl. *Command Line Interface*) kojim se izrađuje Angular aplikacija pomoću naredbe `ng new`, također omogućava vrlo laganu izradu web aplikacija pomoću naredbi za izgradnju komponenti i pokretanja aplikacije. Angular server prati svaku promjenu te na naredbu spremanja datoteka ponovo izgrađuje aplikaciju kako bi promjene odmah bile vidljive na aplikaciji. Svaka komponenta sastoji se od TypeScript, CSS, HTML (engl. *Hypertext Markup Language*) i testne datoteke te ju vrlo laganom možemo napraviti pokretanjem naredbe `ng generate component` kao što je vidljivo na slici 7.1.



Slika 7.1: Kreiranje komponente naredbom `ng generate component`

Pomoću naredbenog retka mogu se napraviti i servisi, servis se koristi za izvršavanje koda koji se može pozvati bilo gdje u aplikaciji. Vrlo bitna funkcionalnost svake web

aplikacije je dohvaćanje podataka, stoga ovaj okvirnik ima ugrađen HttpClient servis pomoću kojeg programeri s lakoćom mogu implementirati komunikaciju između web aplikacije i pozadinske aplikacije. Nakon što korisnik unese svoje podatke u web aplikaciji, poziva se metoda login koja komunicira s pozadinskom aplikacijom, nakon čega ona provjerava postoji li taj korisnik u bazi podataka te vraća odgovarajući odgovor ovisno o tome je li korisnik registriran. Nakon potvrdnog odgovora, korisnik je prijavljen u aplikaciju gdje može koristiti razne funkcionalnosti poput kreiranja zgrada u kojima se nalaze uređaji, vidjeti podatke o temperaturi, vlazi i pritisku koji su učitani s uređaja, također može upravljati kontrolnim uređajima namještanjem intervala ili pokretanjem i zaustavljanjem preko kliznog gumba. U programskom kodu 7.2. prikazana je lakoća pozivanja s pozadinskom aplikacijom.

*Programski kod 7.2: Servis i pozivi prema pozadinskoj aplikaciji*

---

```
import { HttpClient, HttpParams } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { environment } from 'src/environments/environment';
import { User } from './user.model';

@Injectable({
  providedIn: 'root'
})
export class UserService {
  constructor(private http: HttpClient) { }

  login(emailOrUsername: string = '', password: string = '') {
    const params = new HttpParams()
      .set((emailOrUsername.includes('@') ? 'email' : 'username'),
        emailOrUsername)
      .set('password', password);

    return this.http.get<User>(`${environment.api}authUser`, { params });
  }

  signup(user: User) {
    return this.http.post<User>(`${environment.api}signupUser`, user);
  }
}
```

---

## 7.2 MQTT posrednik

MQTT je posrednik koji je zadužen za komunikaciju između uređaja interneta stvari. Baziran je na modelu objave i pretplate te je zadužen za slanje i primanje podataka u mreži koja je ograničena resursima. MQTT protokol je postao standard slanja podataka u IoT sustavima zbog svoje efikasnosti, tako najmanje poruke mogu biti veličine 2 bajta.

Ovaj vrlo sigurni protokol za komunikaciju koristi SSL (engl. *Secured Socket Layer*) protokol kako bi se osiguralo slanje osjetljivih podataka. Skalabilnost ovog protokola je jedan od bitnijih stavki, može se implementirati za komunikaciju između brojnih uređaja dok troši vrlo malo električne energije. MQTT olakšava kriptiranje poruka, provjeru uređaja i korisnika koristeći moderne protokole za provjeru.

### 7.2.1 MQTT komponente

- **Klijent:** bilo koji uređaj od servera do mikrokontrolera koji pokreće MQTT biblioteku i njome komunicira putem interneta.
- **Broker:** pozadinski sistem čija je glavna funkcionalnost zaprimanje i filtriranje poruka te njihovo slanje pretplaćenim i identificiranim klijentima. Poruke se filtriraju pomoću takozvanih tema, koje su organizirane hijerarhijski poput direktorija datoteka.
- **Povezanost:** klijent započinje povezanost slanjem poruke brokeru kojom zahtjeva povezivanje, nakon čega broker provjerava i potvrđuje vezu.

Primjena MQTT protokola u ovom projektu može se pronaći u komunikaciji između mikrokontrolera i pozadinske aplikacije [30]. MQTT logo prikazan je na slici 7.2.



Slika 7.2: Logo MQTT posrednika [31]

## 7.3 Go aplikacijsko programsko sučelje

Go je programski jezik tvrtke Google koji je napravljen kao alternativa jeziku C++ s namjenom za jednostavnije korištenje, ali sa svim bitnim karakteristikama koje možemo pronaći u drugim jezicima poput C++a, Pythona i JavaScripta. Prednosti korištenja ovog jezika su brzo vrijeme prevođenja i izvršavanja koda, automatsko brisanje nekorištenih objekata u memoriji, portabilnost, sigurnost memorije, posebno rukovanje greškama, veliki broj ugrađenih biblioteka i slično. Kreiranje strukture u programskom jeziku Go prikazano je na programskom kodu 7.3.

*Programski kod 7.3: Definiranje strukture Weather u programskom jeziku Go*

---

```
type Weather struct {
ID          int    `json:"id"`
Main        string `json:"main"`
Description string `json:"description"`
Icon        string `json:"icon"`
}
```

---

Nedostaci su apstraktan kod, provođenje strogih pravila i korištenje ne konvencionalnog objektno orijentiranog programiranja. U gore navedenom kodu možemo vidjeti primjer strukture programskog jezika Go. Kod za kreiranje web servera i osluškivanje poziva prikazano je u kodu 7.4.

*Programski kod 7.4: Pokretanje web servera i osluškivanje dolazećih poziva*

---

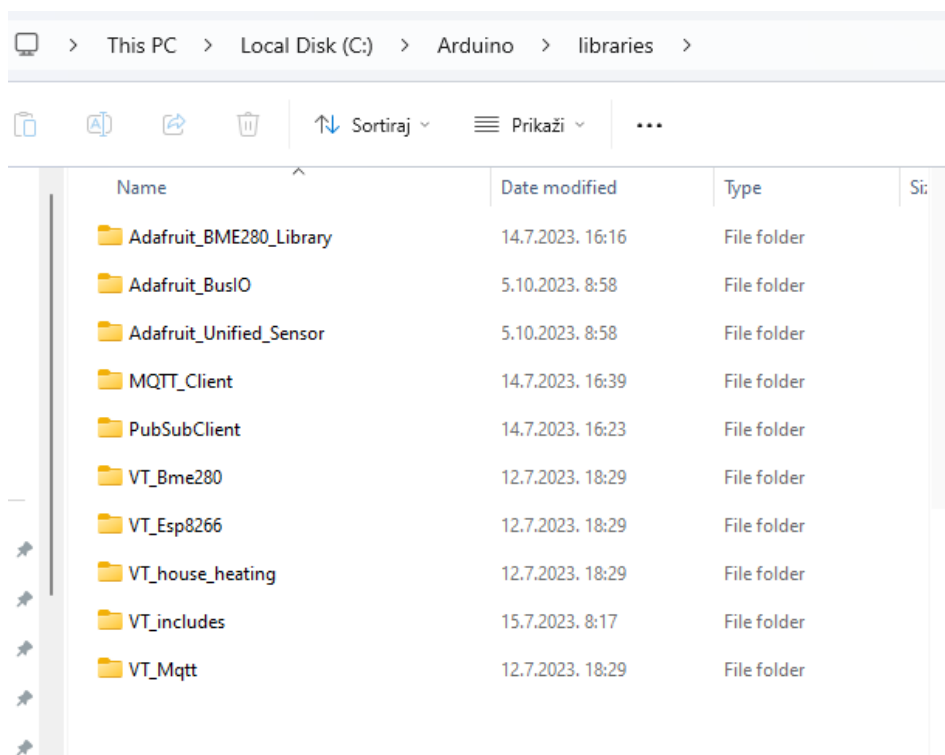
```
cors := cors.New(cors.Options{
    AllowedOrigins: []string{"*"},
    AllowedMethods: []string{"GET", "POST", "PUT", "DELETE", "OPTIONS"},
    AllowedHeaders: []string{"*"},
    Debug:         false,
    AllowCredentials: true
})
mux := http.NewServeMux()
handler := cors.Handler(mux)
mux.HandleFunc("/bme280/getData", iot.getData)
mux.HandleFunc("/authUser", iot.authenticateUser)
http.ListenAndServe(":8080", handler)
```

---

Ovaj jezik također omogućuje vrlo jednostavno pokretanje u svrhe korištenja pozadinske aplikacije, time sa svega par linija koda počinje slušati dolazne zahtjeve. U gornjem kodu pokrenuta pozadinska aplikacija omogućava CORS (engl. *Cross – Origin Resource Sharing*) mehanizam koji omogućava razmjenu podataka između različitih domena, nakon čega osluškuje zahtjeve pomoću funkcije HandleFunc. Također, može se primijetiti da osluškuje zahtjeve koji dolaze s web aplikacije ali i one poslane s mikrokontrolera. Komunikacija između API i mikrokontrolera odvija se pomoću MQTT posrednika.

## 7.4 Izrada programskog rješenja mikrokontrolera

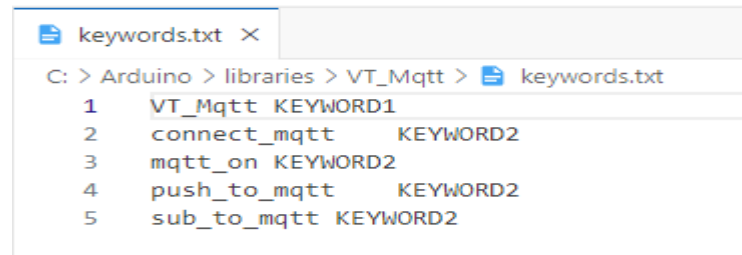
Programsko rješenje mikrokontrolera izrađeno je u Arduino integriranoj razvojnoj okolini. Za namještanje okoline korišteni su koraci navedeni u potpoglavlju 3.1. ovog dokumenta. Arduino također podržava izradu vlastitih biblioteka što je velika prednost jer se tako sav kod ne mora pisati u jednoj datoteci. Biblioteke moraju biti smještene na putanji gdje se nalaze ostale instalirane biblioteke kao što možemo vidjeti na slici 7.3 gdje su biblioteke s prefiksom VT izrađene za potrebe ovog projekta.



Slika 7.3: Izrađene biblioteke s prefiksom VT



Svaka biblioteka mora sadržati tekstualnu datoteku keywords koja omogućuje prepoznavanje koda i značenja funkcionalnosti. Primjer keywords datoteke možemo vidjeti na slici 7.4.



```
keywords.txt
C: > Arduino > libraries > VT_Mqtt > keywords.txt
1 VT_Mqtt KEYWORD1
2 connect_mqtt KEYWORD2
3 mqtt_on KEYWORD2
4 push_to_mqtt KEYWORD2
5 sub_to_mqtt KEYWORD2
```

Slika 7.4: Datoteka keywords i njezine ključne riječi

Tako u datoteci imamo ključne riječi poput KEYWORD1 koja označuje klasu i KEYWORD2 koja označuje funkciju dok su riječi koje se nalaze ispred kodnih riječi, nazivi.

Programski kod 7.5: Uključivanje izrađenih VT biblioteka

---

```
#include <VT_Esp8266.h>
```

---

Primjer koda 7.5 prikazuje lakoću uključivanja izrađenih biblioteka u glavnu datoteku programskog koda, koji se radi vrlo lako s obzirom na to da se putanja izrađenih biblioteka nalazi u mapi Arduino i podmapi biblioteke.

Programski kod 7.6: Postavljanje početnih parametara

---

```
void setup() {
  esp.connect_to_WiFi(SSID_NAME, PWD);

  esp.set_relay_pin(RELAY_PIN);
  esp.set_interval_in_sec(5);

  esp.set_led_pin(LED_BUILTIN);
  esp.set_led_status(false, true);

  mqtt.connect_mqtt(device);
  mqtt.sub_to_mqtt(deviceID"/relayOn");
}
```

---

U programskom kodu 7.6 vidljiv je primjer setup funkcije koja je korištena za početno postavljanje postavki mikrokontrolera. U poziv funkcije `connect_to_WiFi` prosljeđujemo 2 ulazna parametra, konstante `SSID_NAME` i `PWD` koje označavaju naziv mreže i zaporku na koju će se mikrokontroler spojiti, s obzirom da su navedeni parametri osjetljivi podaci, konstante su smještene u posebnoj datoteci kako bi mogle biti izostavljene tijekom verzioniranja koda.

Funkcijom `set_relay_pin` postavljamo pin koji će se koristiti za slanje signala koji kontrolira isključivanje i uključivanje releja, točnije pin koji će odlučivati o radu sustava centralnog grijanja. Sljedeća funkcija, `set_interval_in_sec` postavlja vremenski razmak u kojim će se očitavati vrijednosti sa senzora BME280. Funkcije `set_led_pin` i `set_led_status` namještaju pin i status led diode ukoliko ona postoji, drugi parametar `set_led_status` funkcije govori je li dioda koja se koristi dioda ugrađena u mikrokontroler. Zadnje dvije metode povezuju mikrokontroler s MQTT posrednikom i pretplaćuju ga na pristizanje novih vrijednosti teme „deviceID/relayOn“, učitavanjem vrijednosti preplate odlučuje se o aktivnosti sustava centralnog grijanja.

*Programski kod 7.7: Provjera rada senzora BME280*

---

```
bool VT_Bme280::is_available() {  
  
    if (!_bme.begin(_bme_adress)) {  
        return false;  
    }  
  
    return true;  
}
```

---

Učitavanje vrijednosti sa senzora je vrlo bitno u ovom projektu, stoga je napravljena funkcija prikazana u kodu 7.7. koja provjerava je li senzor dostupan, ovisno o tome šalje se podatak dostupnosti na API koji vrijednost sprema u bazu podataka te dostupnost senzora prikazuje krajnjem korisniku. Također, ukoliko je senzor nedostupan program ne može prikupiti potrebne podatke te se stoga sustav centralnog grijanja isključuje. Kada se senzor ponovo aktivira ili zamijeni novim, mikrokontroler automatski odlučuje hoće li sustav ostati isključen ili će se uključiti.

## Programski kod 7.8: Učitavanje podataka sa senzora i objavljivanje na MQTT

---

```
if (bme.is_available()) {
  if (bme.is_read_interval_reached()) {
    float t = bme.get_temperature();
    float h = bme.get_humidity();
    float p = bme.get_pressure();
    float a = bme.get_altitude();

    float data[] = { t, h, p, a };

    bme.print_sensor_values(data);

    if (!(isnan(t) && isnan(p) && isnan(h))) {
      mqtt.AddBme280DataToJson(data);
      mqtt.push_to_mqtt(MqttDevicePath);
    }

    if (esp.is_led_on()) {
      for (int i = 0; i < 3; i++) {
        esp.chng_pin_mode_w_delay(LED_BUILTIN, LOW, true, MILISECONDS,
          200, 100, "Palim", "Gasim");
      }
    }
    mqtt.recieve_subscription_values();
  }
}
```

---

U programskom kodu 7.8 istekom intervala ispisujemo vrijednosti na serijski monitor, dohvaćamo podatke o temperaturi, pritisku i vlazi i nadmorskoj visini te ih objavljujemo putem MQTT posrednika, nakon kojeg preplaćena pozadinska aplikacija obrađuje podatke i zapisuje ih u bazu. Tijekom učitavanja temperature uključuje se i isključuje LED dioda ugrađena u mikrokontroler te se ispisuju poruke „Palim“ i „Gasim“ na serijski monitor. Funkcionalnost paljenja i gašenja LED diode napravljena je isključivo kako bi korisnici u svom domu mogli vidjeti učitavaju li se vrijednosti sa senzora.

## 8. ZAKLJUČAK

Tema završnog rada je udaljeno i pametno upravljanje sustavom centralnog grijanja koristeći tehnologiju interneta stvari. Nakon opće spoznaje o tome što je internet stvari, povijesti tehnologije te njenoj primjeni, zaključeno je da se već i koristi za izradu rješenja poput onih koji su tema ovog rada. Izradom rada analizirane su razvojne okoline poput Arduina i Raspberry Pi i u radu korištene okoline ESP8266 te integriranih programskih okruženja najčešće korištenih za njihov razvoj kao što su Arduino IDE i PlatformIO.

Korištenjem razvojne pločice ESP8266 NodeMCU 12 – E koja ima mnoštvo pinova, spajanjem komponenti poput senzora BME280 i releja SRD-05VDC-SL-C izrađeno je hardversko rješenje koje je isprogramirano u Arduino programskom jeziku, koristeći Arduino IDE koji olakšava i ubrzava razvoj jednostavnošću korištenja i brojnim bibliotekama.

Nakon spajanja hardverskog rješenja na Wi-Fi mrežu, komunikacija s Go API pozadinskom aplikacijom izvršava pomoću posrednika MQTT koji je danas standard i ključ komunikacije svakog IoT sustava te je vrlo lagan za namještanje i korištenje. Za prikaz informacija i interakciju s korisnikom napravljena je web aplikacija u vrlo poznatom, naprednom i snažnom okvirniku Angularu koji je napravljen da se korisničko sučelje napravi s lakoćom i što kvalitetnije.

Izrada projektnog djela završnog rada optimizirala je korištenje sustava centralnog grijanja, povećala njegovu efikasnost i olakšala njegovo korištenje udaljenim upravljanjem. Sustav je napravljen kao nešto što će se koristiti i podržava daljnje razvijanje integriranjem raznih uređaja ali i dodavanjem novih programskih rješenja kao što su Big data i umjetne inteligencije radi lakšeg i kvalitetnijeg prikupljanja, ali i obrade podataka.

## 9. LITERATURA

[1] Samuel Greengard. Internet of Things [Online]. 2023.

Dostupno na:

<https://www.britannica.com/science/Internet-of-Things> (03.10.2023)

[2] IBM. What is internet of things? [Online]

Dostupno na:

<https://www.ibm.com/topics/internet-of-things> (03.10.2023)

[3] Oracle. What is IoT? [Online]

Dostupno na:

<https://www.oracle.com/internet-of-things/what-is-iot/> (03.10.2023)

[4] ResearchGate. IoT in Logistic Transportation [Online] 2019

[https://www.researchgate.net/figure/IoT-in-Logistic-Transportation\\_fig2\\_332228542](https://www.researchgate.net/figure/IoT-in-Logistic-Transportation_fig2_332228542)

(05.10.2023)

[5] 42Gears Team. How IoT is changing retail industry [Online] 2019

Dostupno na:

[https://www.researchgate.net/figure/IoT-in-Logistic-Transportation\\_fig2\\_332228542](https://www.researchgate.net/figure/IoT-in-Logistic-Transportation_fig2_332228542)

(04.10.2023)

[6] Karan Shah. IoT in healthcare: Applications, benefits, challenges & Future! [Online] 2022

Dostupno na:

<https://www.solutelabs.com/blog/iot-in-healthcare> (04.10.2023)

[7] Kamal R. The Ultimate Use Cases Of IoT in The Mining Industry [Online] 2021

Dostupno na:

<https://www.intuz.com/blog/usecases-of-iot-in-mining-industry> (05.10.2023)

[8] Biz4intellia. Application Of IoT in Automotive Industry | Future of Automobiles [Online]

Dostupno na:

<https://www.biz4intellia.com/blog/iot-applications-in-automotive-industry/> (05.10.2023)

[9] Technology Inovators. IoT: A revolution in the automobile industry [Online]

Dostupno na:

<https://www.technology-innovators.com/iot-a-revolution-in-the-automobile-industry/>

(05.10.2023)

- [10] Justyna Matuszak. The rise of IoT in Smart Cities [Online] 2023  
Dostupno na:  
<https://knowhow.distrelec.com/internet-of-things/the-rise-of-iot-in-smart-cities/>  
(05.10.2023)
- [11] Omrania. Smart Cities Need Culture and Community, Not Just data [Online] 2022  
Dostupno na:  
<https://omrania.com/insights/smart-cities-need-culture-and-community-not-just-data/>  
(06.10.2023)
- [12] Alexey Shalimov. IoT in agriculture: 9 technology use cases for smart farming (and challenges to consider) 2023  
Dostupno na:  
<https://easternpeak.com/blog/iot-in-agriculture-technology-use-cases-for-smart-farming-and-challenges-to-consider/> (06.10.2023)
- [13] e-elektro.blogspot. Što je Arduino?  
Dostupno na:  
<http://e-elektro.blogspot.com/2014/06/sto-je-arduino.html> (06.10.2023)
- [14] Arduino. Arduino Hardware [Online] 2022  
Dostupno na:  
<https://www.arduino.cc/en/hardware> (06.10.2023)
- [15] Arduino. What is Arduino? [Online] 2018  
Dostupno na:  
<https://www.arduino.cc/en/Guide/Introduction> (07.10.2023)
- [16] Les Pounder. Raspberry Pi vs Arduino: Which Board is best? [Online] 2020  
Dostupno na:  
<https://www.tomshardware.com/features/raspberry-pi-vs-arduino> (07.10.2023)
- [17] PlatformIO. What is PlatformIO? 2023  
Dostupno na:  
<https://docs.platformio.org/en/latest/what-is-platformio.html> (07.10.2023)
- [18] Python. The Python Logo [Online]  
Dostupno na:  
<https://www.python.org/community/logos/> (07.10.2023)
- [19] Ben Lutkevich. Microcontroller (MCU) [Online] 2019  
Dostupno na:  
<https://www.techtarget.com/iotagenda/definition/microcontroller> (08.10.2023)

- [20] Mithun Subbaroybhat. A Complete Guide to Microcrotrollers [Online] 2021  
Dostupno na:  
<https://uk.rs-online.com/web/content/discovery/ideas-and-advice/microcontrollers-guide>  
(08.10.2023)
- [21] Randomnerdtutorials. Getting Started with ESP8266 NodeMCU development Board [Online]  
<https://randomnerdtutorials.com/getting-started-with-esp8266-wifi-transceiver-review/>  
(09.10.2023)
- [22] Espressif. ESP8266EX Datasheet [Online] 2023  
Dostupno na:  
[https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf) (09.10.2023)
- [23] ElectronicWings. Introduction of nRF52 Series [Online]  
Dostupno na:  
<https://www.electronicwings.com/nordic-nrf52-series/introduction-nrf52-series>  
(09.10.2023)
- [24] Symmetry Electronics. An Introduction to the Nordic nRF52 Series [Online] 2015  
Dostupno na:  
<https://www.symmetryelectronics.com/blog/an-introduction-to-the-nordic-nrf52-series/>  
(09.10.2023)
- [25] Robocraze. RP2040 Microcontroller Chip [Online]  
Dostupno na:  
<https://robocraze.com/blogs/post/raspberry-pi-rp2040-ic-microcontroller-chip>  
(10.10.2023)
- [26] randomnerdstutorial. Getting Started with ESP8266 NodeMCU development Board [Online]  
Dostupno na:  
<https://randomnerdtutorials.com/getting-started-with-esp8266-wifi-transceiver-review/>  
(10.10.2023)
- [27] David Watson. Introduction to NodeMCU V3 [Online] 2018  
Dostupno na:  
<https://www.theengineeringprojects.com/2018/10/introduction-to-nodemcu-v3.html>  
(10.10.2023)
- [28] Abhilekh Das. Humidity sensor BME280 [Online] 2022

<https://electrocredible.com/bme280-pinout-specs-applications/> (10.10.2023)

[29] Scott Campbell. How to set up a 5V relay on the Arduino [Online]

Dostupno na:

<https://www.circuitbasics.com/setting-up-a-5v-relay-on-the-arduino/> (10.10.2023)

[30] AWS. What is MQTT? [Online]

Dostupno na:

<https://aws.amazon.com/what-is/mqtt/> (10.10.2023)

[31] MQTT. MQTT Specification [Online]

Dostupno na:

<https://mqtt.org/mqtt-specification/> (10.10.2023)



## 10. OZNAKE I KRATICE

IoT (engl. *Internet of Things*)

MQTT (engl. *Message Queing Telemetry Transport*)

IP (engl. *Internet Protocol*)

TCP (engl. *Transfer Control Protocol*)

RFID (engl. *Radio-frequency identification*)

NFC (engl. *Near Field Communcation*)

ARPANET (engl. *Advanced Research Projects Agency Network*)

LAN (engl. *Local Area Network*)

LED (engl. *Light Emitting Diode*)

UART (engl. *Universal asynchronous reciever-transmitter*)

CPU (engl. *Central Processing Unit*)

DIY (engl. *Do It Yourself*)

EEPROM (engl. *Electrically Erasable Programmable Read-Only Memory*)

RISC (engl. *Reduced Instruction Set Computer*)

RTOS (engl. *Real – Time Operating System*)

IEEE (engl. *Institute od Electrical and Electronics Engineers*)

GPIO (engl. *General Purpose Input/Output*)

SPI (engl. *Serial Peripheral Interface*)

SDIO (engl. *Secure Digital Input/Output*)

I<sup>2</sup>C (engl. *Inter-Integrated Circuit*)

I<sup>2</sup>S (engl. *Inter - Integrated Sound*),

PWM (engl. *Pulse - Width Modulation*)

IR (engl. *Infra Red*)

ADC (engl. *Analog – to – data – converter*)

SPI (engl. *Serial Peripheral Interface*)

TWI (engl. *Two – Wire Interface*)

SDK (engl. *Software Development Kit*)

SRAM (engl. *Static Random Access Memory*)

USB (engl. *Universal Serial Bus*)

IO (engl. *Input/Output*)

VIN (engl. *Voltage Input*)

API (engl. *Application Programming Interface*)

MVC (engl. *Model – View – Controller*)

CSS (engl. *Cascading Style Sheets*)

CLI (engl. *Command - Line Interface*)

HTML (engl. *HyperText Markup Language*)

CORS (engl. *Cross Origin Resource Sharing*)

## 11. SAŽETAK

**Naslov:** Primjena interneta stvari u upravljanju sustavom centralnog grijanja

Glavni cilj ovog rada je razvoj i primjena hardverske i softverskog rješenja za udaljeno upravljanje sustavom centralnog grijanja koristeći IoT tehnologiju. Koristeći MQTT protokol komunikacija se odvija između dvije najbitnije stavke razvoja ovog projekta, hardverskog i softverskog rješenja. Bazni element hardverskog rješenja Lolin NodeMCU ESP8266 12 – E V3 prikuplja podatke o temperaturi, vlazi i pritisku pomoću senzora BME280 nakon čega podatke šalje MQTT posrednikom prema pozadinskoj aplikaciji. Pozadinska aplikacija napisana u Go programskom jeziku, dobivene podatke zaprima, obrađuje i sprema u bazu podataka te ovisno o zadanim postavkama grijanja šalje povratnu informaciju mikrokontroleru koji pomoću releja uključuje ili isključuje grijanje. Pozadinska aplikacija također surađuje s korisničkom web aplikacijom koja je napisana u vrlo kompleksnom ali izvrsnom okvirniku Angularu gdje vizualno pokazuje podatke koje korisnici mogu izmjenjivati sa svrhom upravljanja sustavom centralnog grijanja. Rješenje je testirano u stvarnim uvjetima gdje su ostvareni zadovoljavajući rezultati i ostvaren je cilj, a to je kvalitetnije i udaljeno upravljanje sustavom centralnog grijanja pomoću IoT tehnologije.

**Ključne riječi:** Udaljeno upravljanje sustavom centralnog grijanja, IoT, MQTT, Lolin NodeMCU ESP8266 12 – E V3, BME280.

## 12. ABSTRACT

**Title:** Application of Internet of Things in Central Heating System Management

Main goal of this work is development and application of hardware and software solution for remote central heating system management using IoT technology. Communication of two most important parts of this development, hardware and software, is done with MQTT protocol. Base element of hardware solution Lolin NodeMCU ESP8266 12 – E V3 collects temperature, humidity and pressure data using sensor BME280 after which data is sent to backend application using MQTT broker. Backend application written in Go programming language collects, processes and saves given data to database and depending on heating settings sends feedback to microcontroller which then turns heating on or off. Backend application also cooperates with web application written in Angular Framework which visually displays data that can be modified by users with purpose of managing central heating system. Solution is tested in real conditions where it achieved satisfying results and achieved main goal of high quality remote central heating system management using IoT technology.

Keywords: Remote central heating system management, IoT, MQTT, Lolin NodeMCU ESP8266 12 – E V3, BME280

## IZJAVA O AUTORSTVU ZAVRŠNOG RADA

Pod punom odgovornošću izjavljujem da sam ovaj rad izradio/la samostalno, poštujući načela akademske čestitosti, pravila struke te pravila i norme standardnog hrvatskog jezika. Rad je moje autorsko djelo i svi su preuzeti citati i parafraze u njemu primjereno označeni.

Mjesto i datum	Ime i prezime studenta/ice	Potpis studenta/ice
U Bjelovaru, <u>19. 10. 2023</u>	Kristijan Vanja	Kristijan Vanja

U skladu s čl. 58, st. 5 Zakona o visokom obrazovanju i znanstvenoj djelatnosti, Veleučilište u Bjelovaru dužno je u roku od 30 dana od dana obrane završnog rada objaviti elektroničke inačice završnih radova studenata Veleučilišta u Bjelovaru u nacionalnom repozitoriju.

Suglasnost za pravo pristupa elektroničkoj inačici završnog rada u nacionalnom repozitoriju

Kristijan Vanja  
ime i prezime studenta/ice

Dajem suglasnost da tekst mojeg završnog rada u repozitorij Nacionalne i sveučilišne knjižnice u Zagrebu bude pohranjen s pravom pristupa (zaokružiti jedno od ponuđenog):

- a) Rad javno dostupan
- b) Rad javno dostupan nakon \_\_\_\_\_ (upisati datum)
- c) Rad dostupan svim korisnicima iz sustava znanosti i visokog obrazovanja RH
- d) Rad dostupan samo korisnicima matične ustanove (Veleučilište u Bjelovaru)
- e) Rad nije dostupan

Svojim potpisom potvrđujem istovjetnost tiskane i elektroničke inačice završnog rada.

U Bjelovaru, 19.10.2023

Kristijan Vanja  
potpis studenta/ice