Automatizacija proizvodnje u plasteniku

Osredek, Luka

Undergraduate thesis / Završni rad

2017

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:122:227051

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-28**

Repository / Repozitorij:

University North Digital Repository

Sveučilište
Sjever

Završni rad br. 401/EL/2017

Automatizacija proizvodnje u plasteniku

Luka Osredek, 5558/601

Varaždin, srpanj 2017. godine

Sveučilište
Sjever

Odjel za Elektrotehniku

Završni rad br. 401/EL/2017

Automatizacija proizvodnje u plasteniku

Student

Luka Osredek, 5558/601

Mentor

Ivan Šumiga, mr. sc.

Varaždin, srpanj 2017. godine

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

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL: Odjel za elektrotehniku

PRISTUPNIK: Luka Osredek | MATIČNI BROJ: 5558/601

DATUM: 30.06.2017. | KOLEGIJ: PROCESNA INSTRUMENTACIJA

NASLOV RADA: Automatizacija proizvodnje u plasteniku

NASLOV RADA NA ENGL. JEZIKU: Automation of production in greenhouse

MENTOR: mr.sc. Ivan Šumiga, dipl.ing. | ZVANJE: viši predavač

ČLANOVI POVJERENSTVA:
1. Miroslav Horvatić, dipl.ing., predavač
2. Dunja Srpak, dipl.ing., predavač
3. mr.sc.Ivan Šumiga, viši predavač
4. mr.sc.Matija Mikac, viši predavač, zamjenski član
5.

Zadatak završnog rada

BROJ: 401/EL/2017

OPIS

Za veće prinose u proizvodnji biljaka po jedinici prostora i više berbi godišnje izgrađuju se plastenici. Na rast i razvoj biljaka, pojavu bolesti i štetnika te visinu prinosa uz prihranu, najveći utjecaj imaju mikroklimatski uvjeti kao što su: temperatura, svjetlost, vlažnost zraka i tla. Zato je realizacija automatizacije - nadzora i upravljanja mikroklimatskim uvjetima u plasteniku od velike koristi poljoprivrednim proizvođačima..

U radu je potrebno:
- opisati senzore pomoću kojih će se pratiti mikroklimatski parametri plastenika: temperatura, vlažnost zraka, vlažnost tla te jakost svijetla,
- opisati aktuatore kojima se mikroklimatski uvjeti prilagođuju zahtjevima pojedine biljke (grijanje ili hlađenje, osvjetljenje, navodnjavanje),
- realizirati prikupljanje podataka sa senzora, njihovu obradu i upravljanje radom izvršnih članova (pomoću mikroračunala Arduino UNO),
- realizirati komunikaciju mikroračunala s nadređenim sustavom bežičnom vezom (WiFi modulom) te spremanje podataka (u MySQL bazu podataka),
- na temelju prikupljenih podataka realizirati prikaz i obradu podataka te upravljanje izvršnim članovima plastenika lokalno i putem web sučelja,
- izraditi maketu plastenika i izvršiti testiranje realiziranih funkcija.

ZADATAK URUČEN | POTPIS MENTORA

REPUBLIKA HRVATSKA
SVEUČILIŠTE SJEVER

Predgovor

Zahvaljujem se mentoru mr.sc. Ivanu Šumigi, dipl. ing. na konzultacijama tijekom izrade završnog rada, te kod procesa izrade rada. Hvala Vam što ste mi pomogli da ovaj rad učinim uspješnim.

Zahvaljujem profesorima Sveučilišta Sjever u Varaždinu, koji su prenijeli znanje na predavanjima i na laboratorijskim vježbama te mi olakšali izradu završnog rada.

Zahvaljujem se također svojoj obitelji na potpori i mogućnosti obrazovanja te svima koji su mi pružili potporu tijekom trajanja obrazovanja na Sveučilišta Sjever.

Sažetak

U ovom završnom rada opisan je koncept i realizacija automatskog sustava proizvodnje u plasteniku , korištenjem Arduino UNO razvojne platforme. Opisan je programski alat Node-red za izradu web sučelja, WiFi bežična komunikacija, ESP8266 modul za bežičnu komunikaciju te MQTT komunikacijski protokol. Detaljno su opisane korištene upravljačke komponente te njihove funkcije. Nakon razrade ideje, napravljena je maketa automatskog sustava proizvodnje u plasteniku.

Maketa se sastoji od Arduino UNO platforme kao upravljačke jedince, ESP8266 WiFi modula, relejnog modula, LED žarulje, temperaturnog senzora, foto otpornika, senzora vlage zraka , senzora vlažnosti tla, Pletierov-og elementa kao grijača, ventilatora, MOSFET tranzistora.

Kako bi se omogućila bežična komunikacija korišteni je WiFi modul ESP8266 koji pomoću MQTT protokola, koji je detaljno opisan i objašnjen, komunicira s web serverom.

Nakon ulaska u web sučelje, te pokretanjem automatskog sustava proizvodnje u plasteniku korisnik dobiva uvid u trenutne vrijednosti očitanih sa senzora. Sustav ima automatsko i ručno upravljanje.

Kada je uključeno automatsko upravljanje izvršni elementi se pokreću prema zadanim parametrima te je korisnik preko web sučelja obavješten koji izvršni element je pokrenut. Kada je uključeno ručno upravljanje korisnik može pokretati izvršne članove bez obzira na parametre postavljene kod automatskog upravljanja. Izvršne elemente sustava čine solenoidni ventil za navodnjavanje, LED svijetla za rasvjetu te Pletierov element montiran na hladnjak i ventilator za upuhivanje zraka koji služe za grijanje i hlađenje.

KLJUČNE RIJEČI:, Arduino UNO, WiFi modul ESP8266,fotootpornik, mikrokontroler , relejni modul, temperaturni senzor, senzor vlage zraka , senzor vlage tla ,MQTT protokol, Node-red programski alat ,

Popis korištenih kratica

PWM Pulse Width Modulation

Modulacija širine impulsa

DC Istosmjerna struja

AC Izmjenična struja

VCC Positive supply voltage

Pozitivni polaritet napona

GND Ground

Masa

LED Light emitting diode

Svjetleća dioda

PCB Printed circuit board

Tiskana pločica

MOSFET metal–oxide–semiconductor field-effect transistor

metal-oksidni-poluvodič tranzistor sa efektom polja

TTL Transistor-Transistor Logic

Tranzistor-tranzistor logika

USB Universal serial bus

Univerzalna serijska sabirnica

IDE Integrated development environmenta

Razvojna okolina programskog jezika

LDR Light Dependent Resistor

Foto otpornik

SoC System on Chip

Sustav na čipu

Sadržaj

1. Uvod 5
2. Automatizacija proizvodnje u plastenicima 7
2.1. Mikroklimatski uvjeti 7
2.2. Automatizacija proizvodnje 8
2.3. Komercijalna rješenja za automatsko upravljanje plastenicima 8
3. Bežične mreže i Wi-Fi komunikacija 10
3.1. WiFi modul ESP8266 10
3.2. ESP8266 ESP-01 11
3.3. MQTT protokol 12
4. Realizacija sustava 16
4.1. Arduino razvojna platforma 16
4.2. Senzor relativne vlažnosti zraka i temperature zraka DHT22 18
4.3. Senzor vlažnosti tla 19
4.4. LDR otpornik 20
4.5. Relejni modul 21
5. Maketa sustava 23
5.1. Upravljačka jedinica 23
5.2. Realizacija grijanja i hlađenja 24
5.3. Navodnjavanje i rasvjeta 26
5.4. Realizacija programskog koda 29
5.5. Realizacija web sučelja pomoću Node-Red programskog alata 38
5.6. Prikaz rada sustava automatskog upravljanja plastenikom 45
6. Zaključak 49
7. Literatura 50
Popis slika 51

1. Uvod

Kako bi ostvarili veće prinose biljaka po jedinci prostora i više berbi godišnje poljoprivrednici se odlučuju na izgradnju plastenika. Najveći utjecaj na rast i razvoj biljaka u plasteniku ima mikroklima na koju utječu čimbenici kao temperatura, vlažnost zraka, vlažnost tla, svijetlost. Ovim završnim radom htjelo se kako pokazati automatizacija proizvodnje biljaka u plastenicima može pridonijeti većim prinosima te biti od velike koristi poljoprivrednicima. U svijetu postoji komercijalan primjena automatiziranih sustava plastenika. Pošto svaka kultura zahtjeva drugačije mikroklimatske uvjete automatizirani plastenici projektiraju se prema tim uvjetima. U ovom završnom radu izradit ćemo sustav koji će se lako prilagoditi uvjetima različitih kultura.

Za upravljanje procesom automatizacije odabrana je Arduino UNO platforma koja se kasnije prema potrebi može nadograditi ili prilagoditi našim potrebama. Kao ulazne elemente u sustavu automatskog upravljanja plastenikom korišteni su DHT22 digitalni senzor temperature i relativne vlažnosti zraka, senzor vlažnosti tla, KY-18 LDR modul za mjerenje jakosti svijetlosti.

Kako bi se omogućilo udaljeno nadgledanje i upravljanje procesima koristi se WiFi modul ESP8266 u izvedbi ESP01 koji se spaja na Arduino UNO platformu, te oni međusobno komuniciranju pomoću serijske komunikacije. ESP01 je jeftini WiFi modul za Arduino platformu koji je temeljen na Espressif-ovm ESP8266 integriranom krugu koji daje Wi-Fi SoC(Sustav na čipu) rješenje kako bi zadovoljio potrebe korisnika. ESP01 uz priključenu antenu ima domet skoro preko 150 metara. Kako bi se moglo komunicirati između web servera i Arduino UNO platforme odlučeno je da će se koristiti MQTT protokol. MQTT protokol temelji se na arhitekturi publish/subscribe u kontrastu s HTTP-ovim request/response. Publish/subscribe služi za pokretanje događaja i omogućava slanje poruka klijentima. Da bi MQTT protokol funkcionirao potreban je MQTT posrednik koji prosljeđuje poruke između uređaja. U našem slučaju korišten je jednostavni Mosquitto posrednik(broker) instalirani na lokalno računalo. Vizualizacija i upravljanje preko web-sučelja realizirano je pomoću Node-RED programskog alata. Node-RED je grafički programski alat koji je razvio IBM razvojni tim za potrebe korisnika koji razvijaju IoT(Internet stvari) aplikacije. Nakon što su odabrani svi potrebni sklopovski i programski elementi krećemo u realizaciju makete automatiziranog plastenika.

Kao izvršne članove u sustavu koriste se LED svijetla za rasvjetu. LED rasvjeta emitira valne duljine od 300-840nm koje pomažu biljkama da raste kada nema dosta svijetlosti kroz dan. LED rasvjetom upravlja relejni modul koji pali i gasi rasvjetu prema očitanjima senzora te podešenim parametrima sustava. Za navodnjavanje biljaka koristi se kap po kap navodnjavaju.

Cijev za kap po kap navodnjavanje spojena je na solenoidni ventil kojim upravlja relejni modul koji upravlja navodnjavanjem prema očitanjima sa senzora te podešenim parametrima Održavanje konstante temperature izvedeno je pomoću Peltierovog elementa kao grijača montiranog na hladnjaku te ventilatora. Prema očitanima vrijednostima uključuje se grijač te ventilator za grijanje ili samo ventilator za hlađenje. Kako je potrebno često uključivati/isključivati grijanje/hlađenje, umjesto relejnog modula korišteni su MOSFET tranzistor koji također imaju mogućnost PWM modulacijom regulirati brzinom vrtnje ventilatora. Nakon izrade i spajanja makete izradit ćemo programski kod koji će upravljati sustavom. Za analizu podataka korištena je MySQL baza podataka u koju se spremaju vrijednosti dobivene sa senzora.

2. Automatizacija proizvodnje u plastenicima

Kako bi se ostvarili viši prinosi kulture koja se uzgaja, koristi se proizvodnja u zaštićenom prostoru. Kao zaštićeni prostor za uzgoj različitih kultura koriste se staklenici ili u većoj mjeri plastenici.

„Plastenici su velike aluminijske ili čelične konstrukcije pokrivene polietilenskom folijom. Nosiva konstrukcija visokih tunela postavlja se direktno u zemlju, a prilikom izgradnje plastenika montira se na betonske temeljne stupove“. [2]

Slika 2.1 Plastenik[28]

2.1. Mikroklimatski uvjeti

U plasteniku su jedan od najvažnijih čimbenika mikroklimatski uvjeti. Oni utječu na rast i razvoj biljaka, pojavu bolesti i štetnika te visinu prinosa.[3]Da bi kvaliteta i prinos kultura bio visok potrebno je smanjiti oscilacije u temperaturi, pošto one dovode do usporenog rasta ili ugibanja .Kako u razvoju biljaka veliku ulogu ima i relativan vlažnost zraka, kulturama koje su osjetljive na moramo osigurati provjetravanje plastenika.Za rast te fotosintezu biljaka potrebna je sunčeva svijetlost valne duljine 400-700 nm.

2.2. Automatizacija proizvodnje

Jedan od rastućih trendova u poljoprivredi je uvođenje koncepta IOT u proizvodnji u plasteniku. Danas više nego ikad je veća važnost povećati kvalitetu i količinu prinosa. IoT odnosi se na koncept bežičnog povezivanja različitih senzora i uređaja uz korištenje mogućnosti Interneta. U našem kontekstu, sustav kontinuirano prati i automatizira plastenik u kojem uzgajamo različite poljoprivredne kulture.

Sustav nadzire klimatske uvjete kao što su temperatura, svjetlost, vlaga i vlažnost tla koristeći senzore te bežičnom mrežom šalje podatke u oblak(cloud). U automatskim sustavima nadzora, korektivne mjere poduzimaju se na temelju trenutnih mjernih podataka. Drugim riječima, sustav je dovoljno pametan da poduzme nezavisne radnje upravljane podacima, smanjujući ljudske intervencije. Instalacije za grijanje, hlađenje, osvjetljenje i dr. kontrolirano je računalom koje omogućuje maksimalnu učinkovitost.

Slika 2.2 Prikaz parametara praćenih u plasteniku[10]

2.3. Komercijalna rješenja za automatsko upravljanje plastenicima.

Growtronix[9]sustav može pratiti gotovo svaki aspekt plastenika. Sustav je modularan u dizajnu. To znači da sepostaviti Growtronix module gdje je to potrebno. Na primjer, može se imati temperaturni senzor smješten u jednom dijelu plastenika i utičnicu koja kontrolira ventilator koji se nalazi negdje drugdje u plasteniku. Svi sustavi tvrtke Growtronix počinju s osnovnim sustavom Growtronix. Osnovni sustav uključuje Growtronix softver koji se učitava na računalo sa Windows operacijskim sustavom. Modul Network Interface Modul Growtronix (NIM) realizira komunikaciju između računala i hardvera Growtronix.

Slika 2.3 Growtronix komercijalni modul[8]

Sensaphone[7]sustav daljinski prati uvjete u plasteniku. To je jednostavno, prikladno, ekonomično rješenje idealno za manje plastenike, komercijalne proizvođače i maloprodaju. Sustav nadzire stanje 24 sata dnevno, uključujući unutarnju i vanjsku temperaturu, vlagu, ventilaciju, ugljični dioksid, nestanak struje i sigurnost za najviše osam lokacija. Kada sustav prepozna potencijalni problem, odmah može upozoriti do osam osoba s prilagođenim telefonskim pozivima. Unutarnja sigurnosna kopija baterije omogućuje 24 sata kontinuiranog nadzora i upozorenja u slučaju prekida napajanja. Svaka je jedinica zapečaćena u zatvorenom prostoru kako bi se zaštitila od vlage, prljavštine i kemikalija koje se obično nalaze u plasteniku. Operatori mogu dobiti status svakog praćenog stanja na instalacijskom mjestu ili telefonom.

3. Bežične mreže i Wi-Fi komunikacija

Kako bi povezali jedno ili više računala bežično u računalnu mrežu koriste se bežične mreže. Veze između računala uspostavljaju se pomoću elektromagnetskih signala ili valova.[4]
Osnovni elementi bežične mreže su:

- čvor bežične mreže
- bazna stanica
- bežične veze

Wi-Fi ili WiFi je tehnologija za bežično lokalno povezivanje s uređajima na temelju IEEE 802.11 standarda.[5] Uređaji koji koriste Wi-Fi tehnologiju uključuju osobna računala, konzole za video igre, pametne telefone, digitalni fotoaparate, tablet računala i mnoge druge. Kompatibilni uređaji Wi-Fi mogu se povezati s internetom putem WLAN-a i bežične pristupne točke.
Kao nositelj digitalnog signala poruke koristi se visokofrekventni signal od 2.4 Ghz, 3.7 GHz ili 5 GHz. Wi-Fi kao temelj za prepoznavanje uređaja koristi MAC (media access control) adresu. Pristup uređaju ostvaruje se kriptiranom prijavom prema WEP (Wired Equivalent Privacy) ili WPA (Wi-Fi Protected Access) modelu. WEP koristi 40 bit-nu enkripciju, a WPA 128 bit-nu enkripciju.

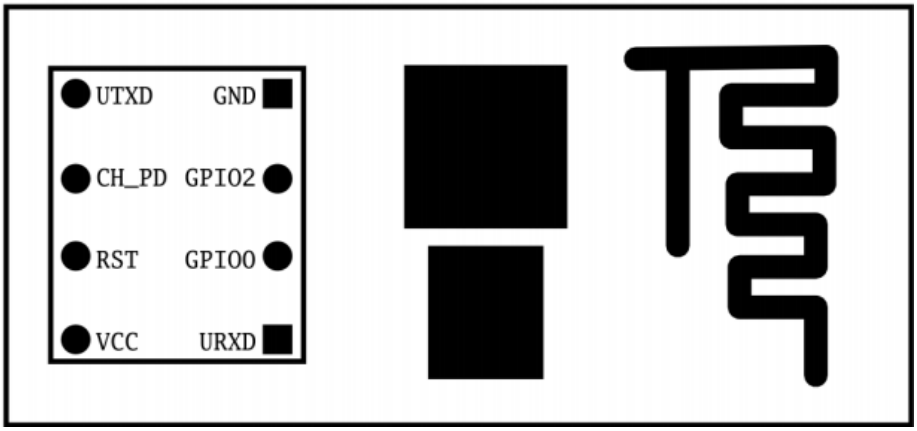
3.1. WiFi modul ESP8266

U izradi makete plastenika za potrebe ostvarivanja bežične komunikacije korišten je Espressif-ov ESP8266 integrirani krug[18].ESP8266 nudi integrirano Wi-Fi SoC(Sustav na čipu) rješenje kako bi zadovoljio potrebe korisnika za učinkovitijim korištenjem energije, kompaktnim dizajnom i pouzdanim performansama u IoT industriji. Navedeni čip dolazi s 32-bitnim mikrokontrolerom sa ugrađenom, radnom memorijom, EEPROM-om te podrškom za 802.11 b/g/n protokol za bežičnu komunikaciju, uz integrirani TCP/IP stog .

ESP8266 može služiti za pohranjivanje te izvršavanje samostalnih aplikacija ili preuzimanje i izvršavanje svih Wi-Fi mrežnih funkcija neke druge aplikacije. Svaki modul ESP826 dolazi s unaprijed programiranim firmware-om sa skupom AT komandi.

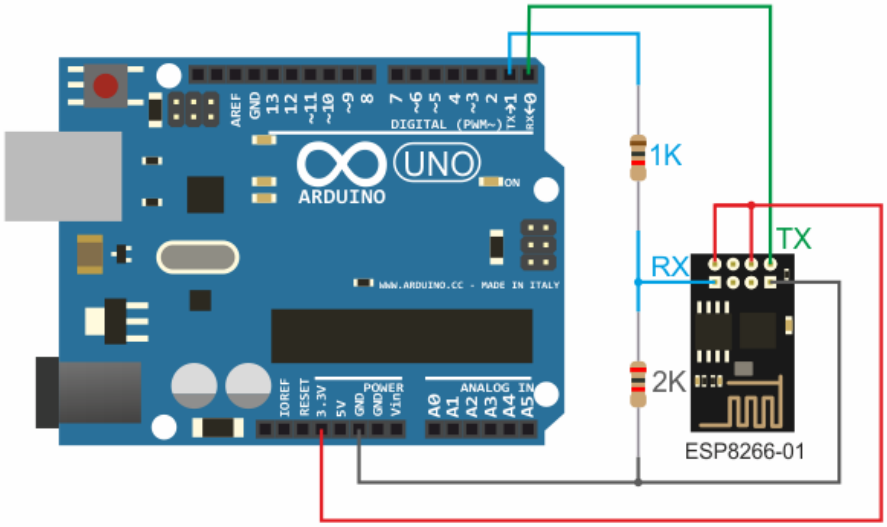
3.2. ESP8266 ESP-01

Modul koji je korišten jedan je od naj rasprostranjenih modula naziva ESP-01 koji uz ESP8266 SoC na pločici sadrži i vanjsku Flash memoriju od 1MB te 8 pinova i PCB antenu dometom od skoro 150 metara. Radi na naponu od 3,3V.[30]



Slika 3.1 Prikaz arhitekture ESP8266 ESP01 modula[27]

Serijska komunikacija između Arduino UNO modula i ESP8266 WiFi modula postiže se međusobnim spajanjem RX/TX pinova.Veze između Arduino UNO modula i ESP-a su vrlo jednostavne: ESP-Rx se spaja s Arduino Tx, ESP-Tx se spaja s Arduino Rx pinom. Međutim, svi ESP-8266 rade na 3.3V, dok Arduino izlazi koriste 5V. Prije spajanja, potrebno je osigurati prilagodbu ovih napona ili se može oštetiti ESP. Da bi se prilagodio napon, korišteno je naponsko dijelilo izvedeno pomoću otpornika od 2kΩ i od 1kΩ.



Slika 3.2 Spajanje ESP8266 modula sa Arduino UNO platformom[29]

Za spajanje korištena je WiFiEsp knjižnica čije su značajke :

- API kompatibilni s standardnom Arduino WiFi knjižnicom.
- Koristiti AT naredbe standardnog ESP firmwarea
- Podržava hardverske i softversko serijske priključke.

Programski kod korišten za spajanje Arduino UNO platforme preko ESP8266 modula na bežičnu mrežu dan je u nastavku.

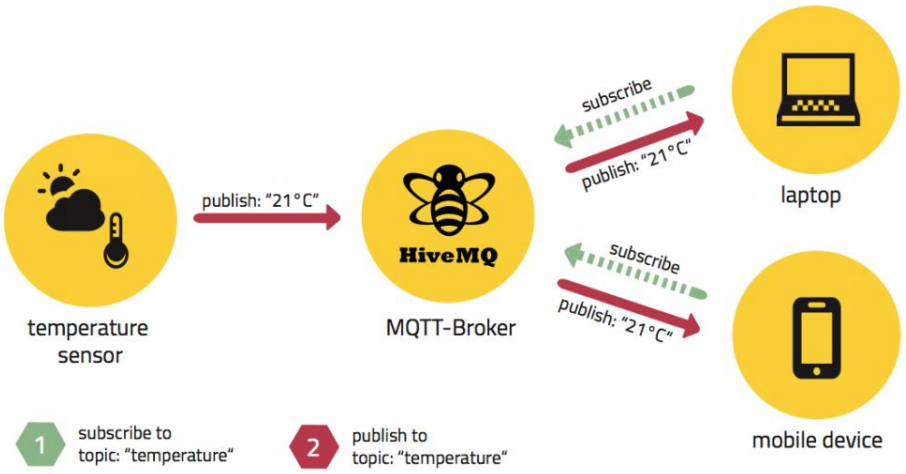
```
void connectWiFi()
{
// Provjera ako je prisutian WIFI modul
if (WiFi.status() == WL_NO_SHIELD) {
//Serial.println("WiFi modul nije prisutan");
while (true);
}
// Pokušaj spajanja na WIFi
while ( status != WL_CONNECTED) {
//Serial.print("Attempting to connect to WPA SSID: ");
// Spajanje na WPA/WPA2 network
status = WiFi.begin(ssid, pass);
}
}
```

3.3. MQTT protokol

Za potrebe praćenja naftovoda kroz pustinju 1999. godine Andy Stanford-Clark i Arlen Nipper razvijaju MQTT(MQ Telemetry Transport). Postavljeni ciljevi kod razvijanja protokola bili su brzina prijenosa, mala potrošnja baterije te na kraju smanjenje cijene sustava, pošto se u to vrijeme koristila satelitska veza koja je bila skupa.[21]MQTT smanjuje mrežno opterećenje te zahtjeve manje resursa od uređaja pri komunikaciji. Ovaj pristup čini MQTT protokol vrlo dobro prilagođen za komunikaciju između uređaja.

Protokol koristi arhitekturu publish/subscribe u kontrastu s HTTP-ovim request/response. Publish/subscribe služi za pokretanje događaja i omogućava slanje poruka klijentima. Središnja komunikacijska točka je MQTT posrednik(broker) on je zadužen za slanje svih poruka između pošiljatelja i prijemnika.

MQTT protokol temelji se na nekoliko osnovnih pojmova, a sve u cilju osiguranja isporuke poruke.MQTT protokol temelji se na načelu objavljivanja poruka i pretplata na teme, što se obično naziva modelom objavljivanja / pretplate. Klijenti se mogu pretplatiti na teme koje se odnose na njih i time primati sve poruke koje se objavljuju na tim temama. Alternativno, klijenti mogu objavljivati poruke temama, čime one postaju dostupne svim pretplatnicima na njih.



Slika 3.3 Primjer MTT komunikacije[21]

Poruke u MQTT-u objavljuju se na teme. Teme su usmjerivačka informacija posredniku(brokeru). Tema je UTF-8 niz, a sastoji od jedne ili više razina. Svaka razina teme odvaja se kosom crtom.

Slika 3.4 Primjer imenovanja tema[21]

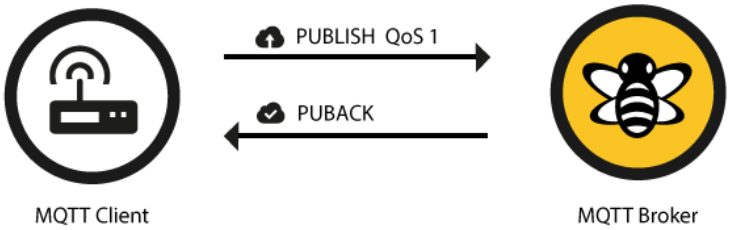
Klijent se, prijavljuje za primanje određenih poruka pretplatom na temu. Pretplate mogu biti eksplicitne, što ograničava poruke koje se primaju na određenu temuili mogu upotrijebiti oznake zamjenskih znakova, kao što su znak „#“ ili znak „+“ za primanje poruka za različite povezane teme. Na primjer tema za broj golova u nogometu glasila bi sport/nogomet/golovi. Klijent ako želi može se pretplatiti na točno tu temu te će dobiti broj golova u nogometu, ali može i koristiti znak „+“ ili „#“ kao na primjer sport/+/golovi te će dobivati informaciju za sve sportove koji su na mjestu „+“ znaka.

Razina kvalitete servisa (Quality of Service Level)[22]-predstavlja razinu garancije isporuke objavljenih poruka. Može poprimiti 3 vrijednosti:

Vrijednost QoS0 (At most once)- poruka je poslana samo jednom bez povratne informacije od primatelja te, bez spremanja i ponovnog slanja od strane pošiljatelja.

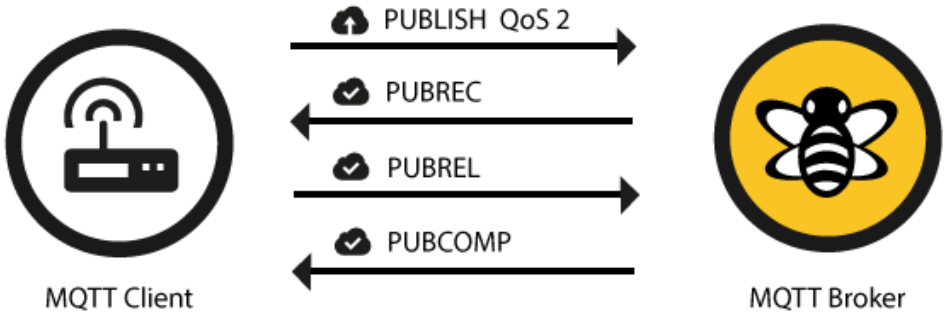
Slika 3.5 QoS 0 razina kvalitete servisa[22]

Vrijednost Qos1 (at-least-once delivery)- ova razina garantira da će poruka biti dostavljena primatelju barem jednom, ali može i više puta. Pošiljatelj prvo sprema poruku, a zatim ju šalje posredniku koji poruku prosljeđuje svim pretplatnicima i vraća potvrdu o slanju poruke u obliku Puback paketa koji sadrži identifikator paketa objavljene poruke (packetId). Nakon što primi potvrdu o uspješnoj objavi, pošiljatelj iz spremnika briše poruku koju je spremio prije same objave



Slika 3.6 QoS 1 razina kvalitete servisa[22]

Vrijednost QoS2-(exactly-once delivery)-najviša, ali ujedno i najsporija razina kvalitete servisa garantira da je svaka poruka primljena točno jednom. Jamstvo osigurava dva toka podataka između pošiljatelja i prijemnika.



Slika 3.7 QoS 2 razna kvalitete servisa[22]

Kada se klijent spaja na poslužitelj, može obavijestiti poslužitelja da ima oporuku(will) ili poruku koja bi se trebala objaviti na određenoj temi ili temama u slučaju neočekivanog isključivanja.

Općenito, MQTT klijent može istodobno biti izdavač(publisher) i pretplatnik(subscireber). MQTT klijent je bilo koji uređaj od mikrokontrolera do punopravnog poslužitelja koji ima MQTT knjižnicu i povezuje se s MQTT brokerom preko bilo koje vrste mreže.

Pandan MQTT klijent je MQTT posrednik, koji je središnji dio modela objavi-pretplati. Ovisno o konkretnoj implementaciji, broker može komunicirati s tisućama istodobno povezanih MQTT klijenata. Brokeri su prvenstveno odgovorni za primanje svih poruka, filtriranje, odlučivanje tko je zainteresiran, a zatim slanje poruke svim pretplatnicima.

U našem slučaju kao posrednika korišten je Mosquitto posrednik[23], koji je open-source MQTT broker, te podržava MQTT V3.1 protokol i dostupan je za više operacijskih sustava.

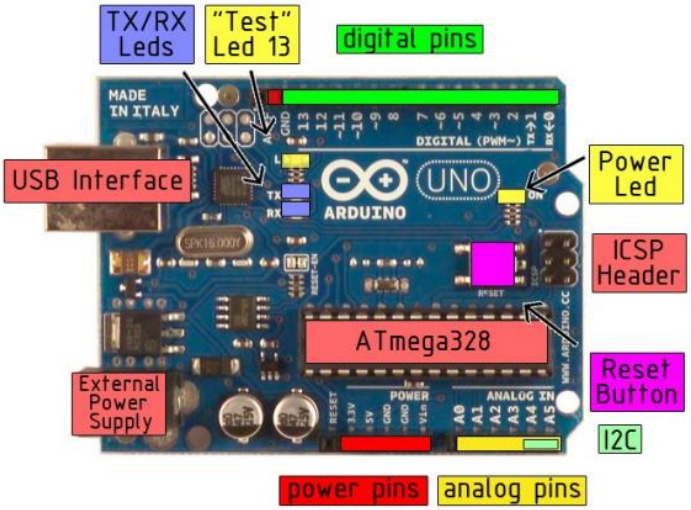
4. Realizacija sustava

4.1. Arduino razvojna platforma

Arduino je open-source platforma otvorenog koda temeljena na jednostavnoj razvojnoj pločici s ulazno/izlaznim pinovima i jednostavnim korisničkim sučeljem.Kao jezgru sadrži mikrokontroler, U/I pinove, USB priključak za napajanje i upravljanje.

Programiranje platforme je vrlo jednostavno i izvodi se iz grafičkog IDE-a(Razvojna okolina programskog jezika) koji postoji za Windowse i Linux, u programskom jeziku koji je sličan C-u. Napravljen je od open source hardverskih komponenti koje su spojene na 8-bitni Atmel AVR mikrokontroler ili 32-bitni ARM procesor. Trenutno postoji 16 različitih modela, a svi se mogu proširivati sa tzv. shield-ovima, gotovim PCB-ovima koji imaju različite funkcije.

Arduino Uno je razvojna platforama temeljena na ATmega328 mikrokontroleru. Ima 14 ulaznih/izlaznih digitalnih pinova (od kojih se 6 može koristiti kao PWM izlaz), 6 analognih ulaza, USB priključak, utičnicu za napajanje, ICSP(in-circuit serial programming) zaglavlje i gumb za resetiranje.[33] Arduino Uno može se napajati preko USB priključka ili s vanjskim napajanjem. Arduino UNO može raditi na vanjskom napajanju od 6V do 20V DC. Ako se napaja s manje od 7V, 5V pin može isporučiti manje od pet volti, te rad sustava može postati nestabilan. Ako se koristi više od 12V,regulator napona se može pregrijati i oštetiti ploču. Preporučeni raspon je 7 do 12 volti. Svaki od 14 digitalnih pinova na Arduino Uno platformi može se koristiti kao ulaz ili izlaz, koristeći `pinMode()` funkciju. Rade na 5 volti. Svaki pin može pružiti ili primiti maksimalno 40 mA i ima unutarnji pull-up otpornik (standardno isključen) od 20-50 kΩ.

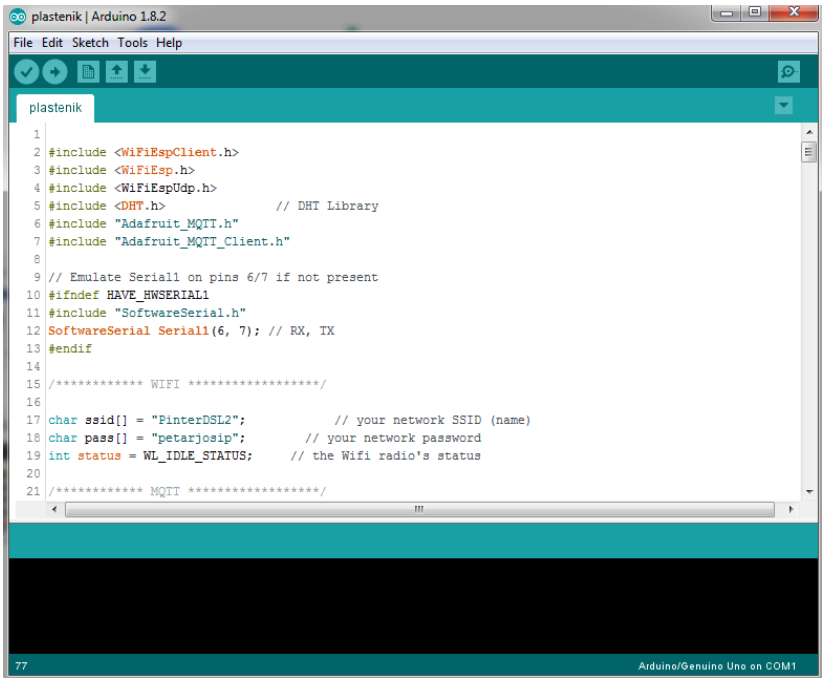


Slika 4.1 Prikaz komponenti Arduino Uno R3 razvojne platforme[20]

Neki pinovi imaju specijalizirane funkcije:

- Serijska komunikacija : pinovi 0 (RX) i 1 (TX). Koriste se za primanje (RX) i slanje (TX) TTL serijskih podataka. Ti pinovi su spojeni na odgovarajuće pinove ATmega8U2 USB-to-TTL serijskog čipa.
- Vanjski prekidi : pinovi 2 i 3. Mogu se konfigurirati tako da pokreću prekid na niskoj razini, porast ili pad brida ili promjenu vrijednosti. Koristi funkciju `attachInterrupt ()` za upravljanje prekidima
- PWM: pinovi 3, 5, 6, 9, 10 i 11. 8-bitni PWM izlaz s kojim se upravlja funkcijom `analogWrite ()`.
- SPI: pinovi 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ovi pinovi podržavaju SPI komunikaciju pomoću SPI knjižnica.

Arduino programsko sučelje (IDE) omogućuje pisanje programa i njihovo prenošenje na Arduino ploču. Sadrži uređivač teksta za pisanje koda, područje za poruke, tekstnu konzolu, alatnu traku s gumbima za uobičajene funkcije i niz izbornika. Programi pisani pomoću Arduino Software (IDE) nazivaju se skice. Skice napisane u uređivaču teksta spremaju se uz nastavak .ino.



Slika 4.2 Arduino IDE razvojno sučelje

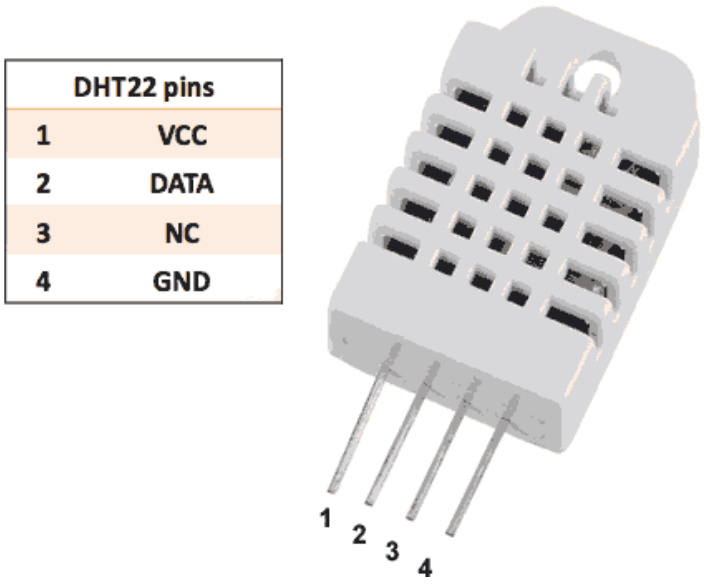
Osnovna struktura Arduino programskog jezika dosta je jednostavna te se sastoji od najmanje dvije potrebne funkcije a to su `void setup()` i `void loop()`.

`void setup()` funkciju poziva se jednom i to na početku kada se program pokreće prvi puta. U njoj inicijaliziramo funkcije pinova ili na primjer započinjemo serijsku komunikaciju. Funkcija mora biti uključena u program

`void loop()` funkciju poziva se nakon `void setup()` funkcije te se kao što joj ime govori izvršava cijelo vrijeme u petlji. U ovom dijelu programa definiraju sefunkcije koje će se izvršavati u petlji te tako upravljati Arduino UNO razvojnom platformom.

4.2. Senzor relativne vlažnosti zraka i temperature zraka DHT22

DHT22je kapacitativni senzor relativne vlažnosti zraka i temperature zraka koji na svojem izlazu daje 40-bitni digitalni signal[19]. Osjetilo temperature i osjetilo relativne vlažnosti zraka spojeno je na 8-bitni mikrokontroler te OTP memoriju preko kojih se vrši kalibriranje svakog očitanja.DHT22 senzor ima vlastiti Single Wire protokol koji se koristi za prijenos podataka. Ovaj protokol zahtijeva precizno mjerenje vremena i vremenske dijagrame za dobivanje podataka iz senzora. Međutim, ne mora se brinuti mnogo o ovim vremenskim dijagramima jer se kod korištenja DHT22 senzora preuzima i uključuje knjižnica koja brine o svemu (`#include dht.h`).



Slika 4.3 DHT22 dijagram pinova[19]

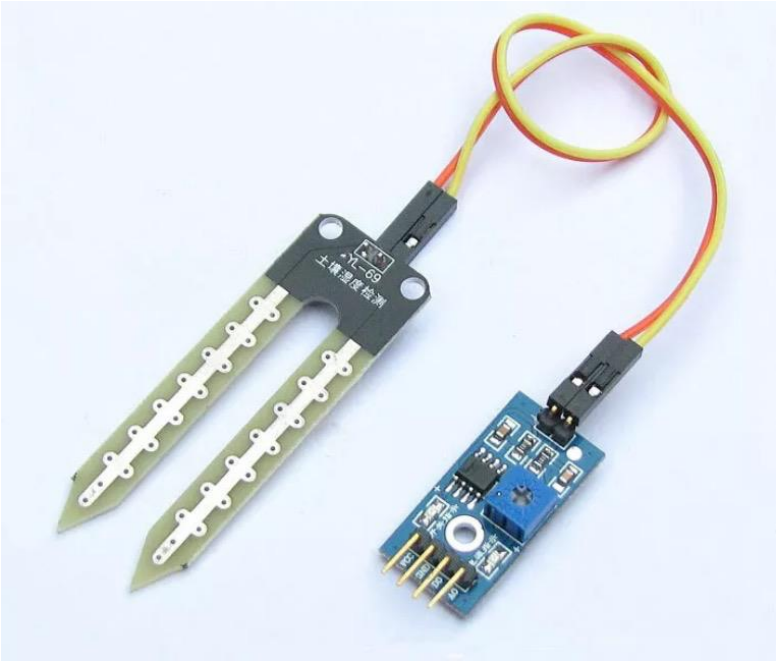
Senzor može mjeriti temperaturu u opsegu od -40°C do 125°C uz točnost od 0.5°C , te relativnu vlažnost zraka od 0-100% uz točnost od 2%. Radi na naponu od 3.3V do 6V(preporučeno 5V) uz prosječnu struju od 300uA. Nakon što spojimo VCC pin na +5V te ,GND pin na masu, pin DATA spaja se na odabrani digitalni pin Arduina, za koji se koristi pull-up otpornik od 4.7kΩ do 10kΩ kako bi cijelo vrijeme DATA pin bio u stanju logičke jedinice te tako omogućio komunikaciju između senzora i Arduino mikroračunala.

Nakon što se u programu uključi knjižnica, definira se pin koji je spojeni na DATA pin DHT22 modula pomoću naredbe #define DHTPIN 2,te model DHT modula pomoću naredbe

#define DHTTYPE DHT22. Nakon toga slijedi inicijalizacija komunikacije između Arduino i DHT22 modula pomoću DHT dht(DHTPIN, DHTTYPE);naredbe. U dijelu void setup() prije početka mjerenja izvršava se naredba dht.begin();.

4.3. Senzor vlažnosti tla

Vrsta senzora koji se koristi je otpornički senzor[24]. Koristi dvije sonde kako bi struja prolazila kroz tlo, a zatim pročitamo otpor kako bi dobili razinu vlage. Senzor vlažnosti tla sastoji se od logičkog modula, te od sonde koja mjeri količinu vode u tlu . Sonda se sastoji od dvije metalne šipke odvojene na fiksnoj udaljenosti pomoću izolacijskog materijala. Sondu se spaja na logički modul.



Slika 4.4 Senzor vlažnosti tla[30]

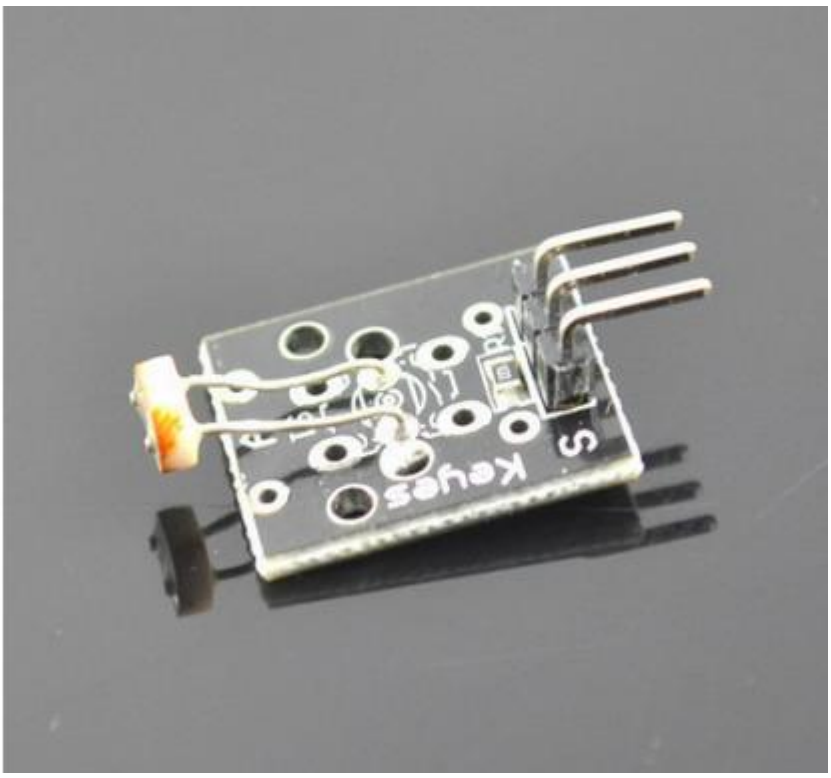
Na VCC stezaljke logičkog modula priključuje se napon od 3.3V-5V, a na stezaljke GND masu. Rezultate mjerenja dobijemo preko analogno izlaza A0 koji je spojeni na analogni pin Arduino UNO platforme. Rezultati mjerenja se prikazuju brojčano od 0-1023, i mogu se skalirati od 0%-100%. Na modulu također postoji digitalni izlaz koji se aktivira kada je vrijednost mjerenja jednaka ili veća od zadana vrijednosti koja se može namjestiti pomoću potenciometra od 10kΩ. Usporedbu vrijednosti vrši komparator LM393. Nedostatak ovog senzora je moguća pojava korozije kod učestalog korištenja.

4.4. LDR otpornik

Foto otpornici, također poznati kao svjetlosni ovisni otpornici (LDR) su svjetlosno osjetljivi uređaji koji se najčešće koriste za označavanje prisutnosti ili odsutnosti svjetlosti ili za mjerenje intenziteta svjetlosti.[34]

U mraku, njihova otpornost je vrlo visoka, ponekad do 1MΩ, ali kada je LDR senzor izložen svjetlosti, otpor padne dramatično, pa čak i do nekoliko ohma, ovisno o intenzitetu svjetla. Koriste se u mnogim aplikacijama, ali ponekad ih mijenjaju drugi uređajima poput fotodioda i fototranzistora.

S obzirom na korištene materijale za izradu, foto otpornici se mogu podijeliti u dvije vrste. Intrinzični i ekstrinzični. Intrinzični foto otpornici koriste čiste materijale poput silicija ili germanija. Ekstrinzični foto otpornici izrađeni su od materijala dopiranog s nečistoćama, koji se nazivaju i dopandi. Bez obzira na to koji materijali se koriste u izradi , obje vrste pokazuju smanjenje otpora kada su osvijetljene. Što je veći intenzitet svjetla, veći je pad otpora.



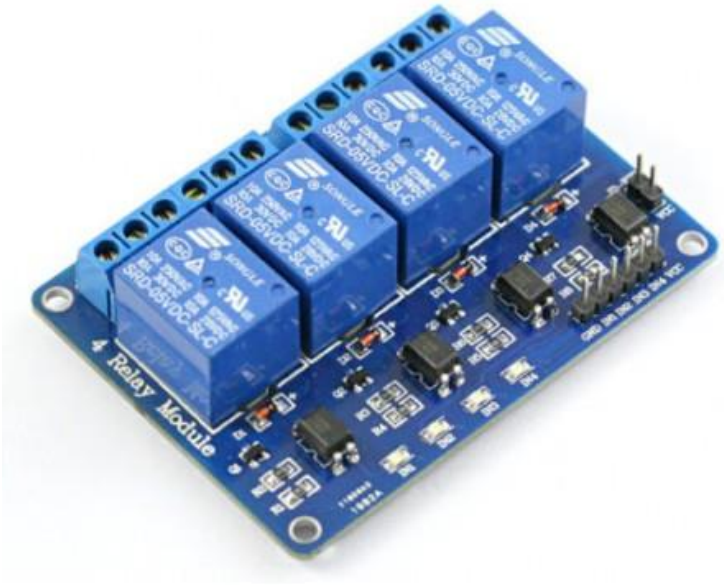
Slika 4.5 LDR modul[15]

Modul koji je korišten naziva je KY-18[14].U seriju s foto otpornikom spojen je otpornik vrijednosti 10kΩ te skupa čine djelitelj napona. Modul ima 3 stezaljke VCC na koju je spojen napon od 5V, GND stezaljku na koju je spojena masa , te stezaljku signal koja se spaja na analogni ulaz Arduino UNO modula.

4.5. Relejni modul

Releji se uključuju preko više vrsta kontakata i osnovni im je zadatak da preko svojih kontakata i pomoćnih strujnih krugova djeluju na druge uređaje radi upravljanja, mjerenja, signalizacije, zaštite nekih postrojenja i nekih dijelova postrojenja.[36] Relejske kontakte pomiče namot elektromagnetnom silom. U našem konkretnom slučaju relej pali i gasi rasvjetu i navodnjavanje.

Krug koji energizira zavojnicu je potpuno galvanski odvojen od upravljačkog dijela. To je razlog zašto možemo upravljati relejom pomoću 5Viz Arduino sustava, a drugi kraj može pokrenuti uređaj napajan s 230V koji je potpuno izoliran od 5V Arduino UNO sklopa.



Slika 4.6 Četvero kanalni relejni modul[18]

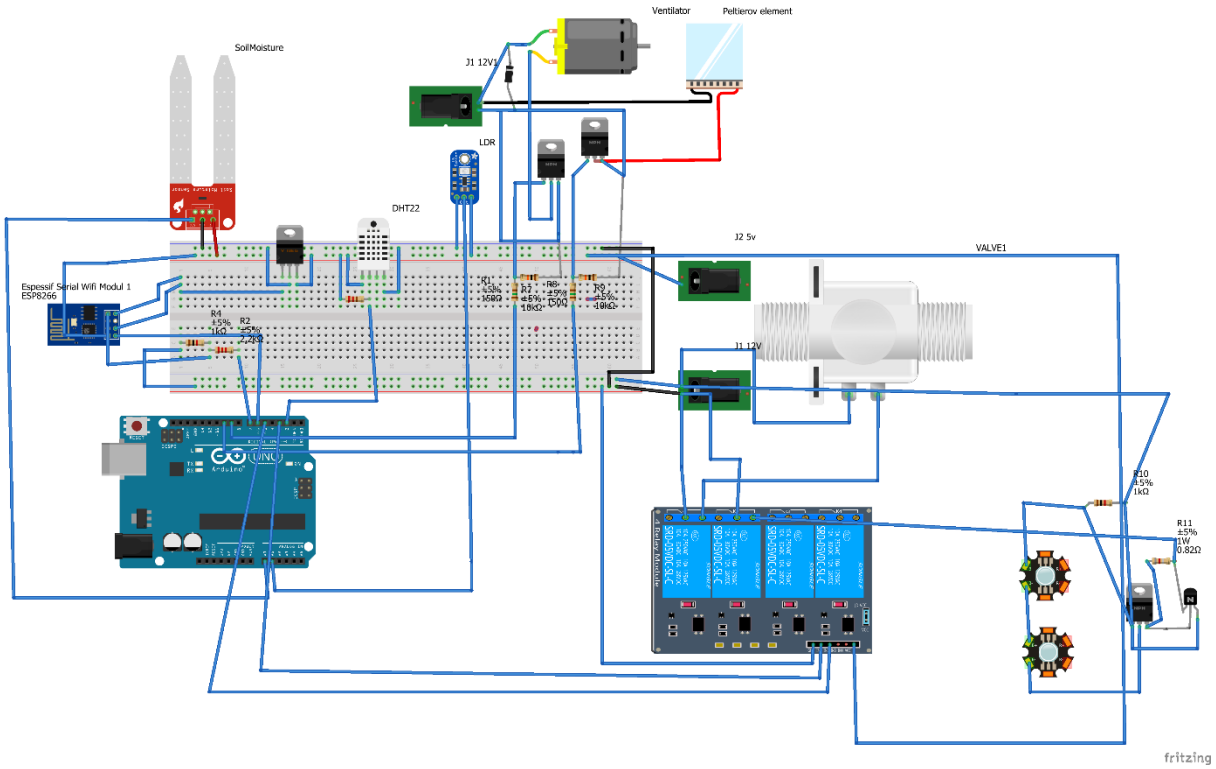
Ovaj modul dolazi s optoizolatorom koji ima ulogu električnog odvajanja strujnog kruga mikrokontrolera od strujnog kruga elektromagneta releja. Optički izolirano znači da zraka infracrvene svjetlosti povezuju strujni krug mikrokontrolera sa strujnim krugom elektromagneta releja. U ovom modulu za optičku izolaciju koristi se integrirani krug PC817.

Za kompletnu optičku izolaciju, VCC pin spaja se na Arduino +5 volti. Spaja se Arduino digitalni izlazni pinovi ,u ovom slučaju na "IN1", "IN2" stezaljke. Zatim se zasebnih +5V spaja

na JD-Vcc stezaljku i Arduino GND na masu. Ovo će napajati strujom tranzistor i zavojnicu releja.

5. Maketa sustava

Maketa sustava sastoji se od upravljačke jedinice, grijanja, hlađenja, navodnjavanja te rasvjete.



Slika 5.1 Shema spajanja sustava automatskog upravljanja proizvodnje u plasteniku

5.1. Upravljačka jedinica

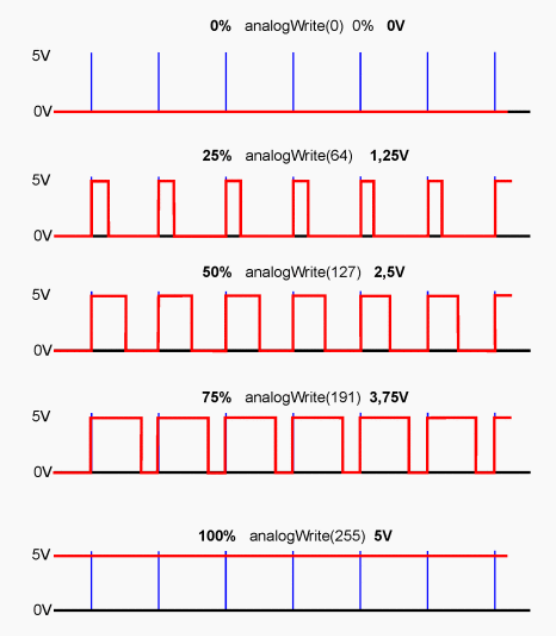
Upravljačka jedinica sastoji se od Arduino UNO razvojne platforme. Arduino UNO razvojna platforma upravlja WiFi modulom, relejnim modulom te MOSFET tranzistorima koji su korišteni za PWM modulaciju upravljanja te kao sklopke za ON/OFF upravljanje. Za napajanje Arduino UNO platforme korišteni je istosmjerni napon od 12V, a za ostalu elektroniku istosmjerni napon od 5V.

Kako WiFi modul ESP8266 radni na naponskoj razini od 3.3V, korišten je regulator napona LD1117 da bi snizio naponsku razinu od 5V na naponsku razinu od 3.3V.LD1117 prema svojim specifikacijama može pružiti struju od 1A što je dovoljno pošto je maksimalan struja koju WiFi modul ESP8266 troši 400mA.Shema sustava prikaza je na slici 5.1

5.2. Realizacija grijanja i hlađenja

Da bi se održavala temperatura konstantom te da ne bi previše oscilirala korišteno je grijanje i hlađenje plastenika. Grijanje je izvedeno pomoću hladnjaka na kojeg je pričvršćena topla strana Peltierov-og elementa te su smještena u cijev kroz koju se upuhuje zrak pomoću ventilatora koji radi na 12V DC. Pošto se kod grijanje i hlađenja često uključuje te isključuje Peltierov element, te radi potrebe PWM modulacije za regulaciju brzine vrtnje ventilatora umjesto releja korišten je IRLR2705PBF N-FET-logic level MOSFET tranzistor maksimalne struje 28A te napona 55V DC.

PWM je tehnika koja se koristi za dobivanje analognih vrijednosti, brzim uključivanjem i isključivanjem DC napona tako da uređaji koji primaju vrijednost s pinova vide samo srednju vrijednost[1]. Na Arduino PWM pinovima ovom tehnikom, se može dobiti vrijednost izlaznih napona između 0 i 5Va ne samo granične vrijednosti.

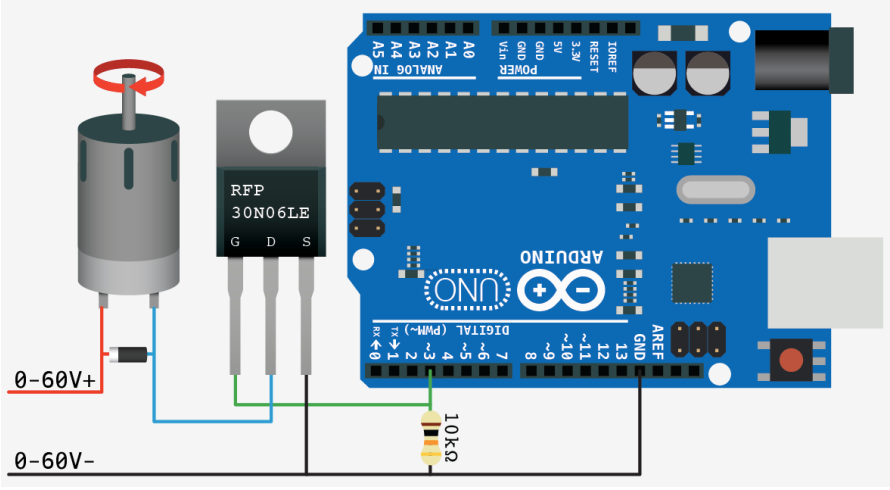


Slika 5.2 PWM modulacija [1]

Kao grijač korišteni je Peltierov element montiran na hladnjak. Peltierov element je termo-električni element koji se sastoji od velikog broja poluvodičkih kristala p- i n-tipa, a koji su u naizmjeničnom rasporedu poslagani između dva sloja keramike koji služe kao izolatori[11]. Princip rada Peltierov-og elementa temelji se na istoimenom efektu. Element koji je korišteni u izradi makete je TEC SP1848-27145.

Slika 5.3 Peltierov element montiran na hladnjak te smješten u cijev

Kako se za grijanje i hlađenje uključuju velike DC struje koje mogu uništiti Arduino UNO moraju se koristi relejni moduli ili MOSFET tranzistori .Uzimajući u obzir često uključivanje i isključivanje grijača te korištenje PWM modulacije, umjesto relejnog modula korišteni su IRLR2705PBF N-FET-logic level MOSFET tranzistori koji zapravo služe kao sklopke. Shema spajanja može se vidjeti na slici 5.4 .



Slika 5.4 Prikaz spajanja MOSFET tranzisora sa Arduino UNO platformom[9]

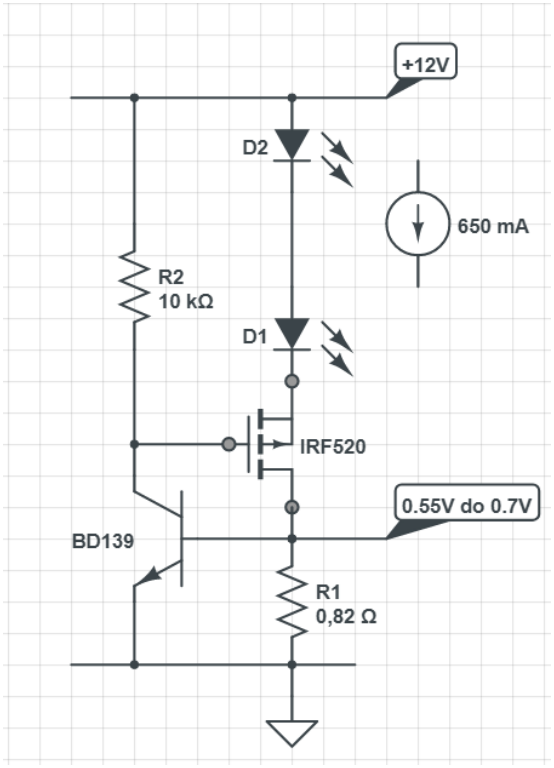
Gate pin MOSFET-a spojen je na jedan od PWM pinova ARDUINO UNO platforme. Kada se postavi pin Arduino UNO platforme na visoku naponsku razinu počinje teći struja između source i drain pina tranzistora te se pokreće uređaj koji želimo kontrolirati. Source pin tranzistora spojen je na GND Arduino UNO platforme. Drain pin tranzistora spojen je na negativnu stezaljku uređaja kojom se upravlja. Pozitivna stezaljaka uređaja spojena je na pozitivni DC napon. Između gate pina tranzistora te Arduino UNO pina dodan je pull-down otpornik koji osigurava nisku naponsku razinu kada Arduino UNO ne šalje signal visoko naponske razine. Da bi zaštitio MOSFET tranzistor paralelno se spaja dioda koja ne dopušta prolazak struja od uređaja prema tranzistoru.

5.3. Navodnjavanje i rasvjeta

Kako bi se osiguralo navodnjavanje u plasteniku korišten je solenoidni ventil kojim se upravlja pomoću relejnog modula. Na ulaz solenoidnog ventil spojeno je crijevo preko 1/2" priključnice na vodovodnu mrežu. Na izlaz solenoidnog ventil pomoću priključnice sa 1/2" na 3/4" na koju je priključena spojnica na koju se montira cijev za kap po kap navodnjavanje.

Slika 5.5 Spajanje navodnjavaj kap na pak sa solenoidnim ventilom

Kako bi se produžila svijetlost za rast biljaka korištene su 2 LED žarulje snage 3W spojene u seriju koje se pale kada padne mrak. LED žarulje emitiraju svijetlost valne duljine od 380-840 nm koju biljke koriste za rast. Kako LED žaruljama za ispravan rad treba omogućiti konstantnu struju od 700mA, a mi koristimo napajanje konstantnog napona od 12V DC, potreban je LED driver konstantne struje. Za tu namjenu izrađen je jednostavni LED driver od IRF520 N-FET tranzistora, BD139 NPN tranzistora ,otpornika od 10KΩ kao pull-otpornika, te 0.82Ω otpornika koji služi kao osjetilo struje. Shema spajanja rasvjete je prikazana na slici 5.6.



Slika 5.6 Shema spajanja LED rasvjete

Glavni strujni krug čine LED diode, IRF520 i R1. Kada previše struje teče kroz R1, BD139 će se početi uključivati, što počinje isključivati IRF520. Isključivanjem IRF520 smanjuje se struja kroz LED i R1. Stoga je stvorena povratna veza, koja kontinuirano prati struju i održava je na točno zadanoj vrijednosti u svakom trenutku.

Nakon spajanja svih komponenti u plastenik su posađene 3 sadnice paprike. Tu završnu maketu možemo vidjeti na slici ispod.

Slika 5.7 Maketa plastenika sa posađenim sadnicama paprike

5.4. Realizacija programskog koda

Nakon što su odabrane komponente međusobno povezane napisan je programski kod za upravljanje procesima koji su potrebni da bi sustav funkcionirao. Programiranje Arduino UNO razvojne platforme vrši se u programskom okruženju Arduino IDE. Dosta je slično C programskom jeziku ali ima svoje specifičnosti pa je potrebno proučiti pripadajuću dokumentaciju.

Na početku programa kao i kod C programskog jezik a moramo uključiti knjižnice koje se koriste u izradi programskog koda. Prije nego se uključe pojedine knjižnice je potrebno preuzeti te instalirati u Arduino IDE programskom okruženju. Knjižnice se koriste kao definicije kako bi pojedine komponente funkcionirale ispravno. U ovom slučaju potrebno je uključiti knjižnice za Wi-Fi modul ESP8266 , DHT22 osjetilo temperature te Adafruit MQTT knjižnicu.[20]

```
#include <WiFiEspClient.h>
#include <WiFiEsp.h>
#include <WiFiEspUdp.h>
#include <DHT.h>                // DHT Library
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
```

Kako su kod Arduino UNO razvojne platforme pinovi za serijsku komunikaciju na pinu 0 i 1 koji su također potrebni kod testiranja i praćenja rada programa pomoću Serial Monitora, korištena je knjižnica SoftwareSerial kako bi simulirali RX/TX pinove na nekom drugom digitalnom pinu Arduino UNO razvoje platforme u našem slučaju odabrani su digitalni pinovi 6 i 7.

```
#ifndef HAVE_HWSERIAL1
#include "SoftwareSerial.h"
SoftwareSerial Serial1(6, 7); // RX, TX
#endif
```

Radi povezivanja na Wi-Fi pomoću ESP8266 Wi-Fi modula potrebno je definirati ime, te lozinku mreže na koju se želi povezati. Također je potrebno definirati IP adresu, te port MQTT brokera.

```
/************ WIFI ******************/

char ssid[] = "PinterDSL2";             // ime mreže
char pass[] = "petarjosip";         // lozinka mreže
int status = WL_IDLE_STATUS;
```

```
/************ MQTT ******************/

#define AIO_SERVER      "192.168.0.100"
#define AIO_SERVERPORT  1883



/************ ESP i MQTT ******************

WiFiEspClient espClient;
Adafruit_MQTT_Client mqtt(&espClient, AIO_SERVER, AIO_SERVERPORT);
```

Kako bi se u daljnjem programskom kodu koristili pinovi Arduino UNO razvojne platforme potrebno je definirati pinove koji se koriste. Pošto u se koristi DHT22 osjetilo temperature, potrebno je radi inicijalizacije definirati koji pin, te koji je model DHT osjetila korišten.

```
/************ DHT & pin settings ******************/

#define DHTPIN 2

#define DHTTYPE DHT22   // DHT 22

#define LDR_PIN 1 // PIN LDR senzora
#define soilm_pin 0 //PIN Senzora vlažnosti tla
#define LAMP 3
#define PUMP 4
#define HEATER 9
#define FAN 10
DHT dht(DHTPIN, DHTTYPE);
```

Prije nego se definiraju konstante te varijable koje će se koristi u daljnjem programu potrebno je radi knjižnice koju koristimo za komunikaciju MQTT protokolom definirati teme na koje će se objavljivati poruke.

```
/*****************************Feeds ******************************/

Adafruit_MQTT_Subscribe  humslider  =  Adafruit_MQTT_Subscribe(&mqtt,
"humslider", MQTT_QOS_1);

Adafruit_MQTT_Subscribe  tempslider  =  Adafruit_MQTT_Subscribe(&mqtt,
"tempslider", MQTT_QOS_1  );


Adafruit_MQTT_Subscribe  soilslider  =  Adafruit_MQTT_Subscribe(&mqtt,
"soilslider",MQTT_QOS_1 );

Adafruit_MQTT_Subscribe svijetslider = Adafruit_MQTT_Subscribe(&mqtt,
"svijetslider" ,MQTT_QOS_1 );
```

```
Adafruit_MQTT_Subscribe gumb = Adafruit_MQTT_Subscribe(&mqtt,  "gumb"
,MQTT_QOS_1 );
```

Također potrebno se pretplatiti na teme preko kojih će se primati poruke.

```
Adafruit_MQTT_Publish hum = Adafruit_MQTT_Publish(&mqtt,  "hum");

Adafruit_MQTT_Publish tem = Adafruit_MQTT_Publish(&mqtt, "temp");

Adafruit_MQTT_Publish ldr = Adafruit_MQTT_Publish(&mqtt,  "light");

Adafruit_MQTT_Publish soilm = Adafruit_MQTT_Publish(&mqtt,  "soilm");

Adafruit_MQTT_Publish     prom     =     Adafruit_MQTT_Publish(&mqtt,
"promjena");
```

Nakon definicije pinova, te tema potrebnih za MQTT komunikaciju, definirane su pojedine varijable koje će se koristi u daljnjem programu .

```
/****************************** Setup*********************/

float zad_temp = 18;
int mrak= 40;
int suho_tlo = 30;
int zad_vla = 60;

long  sampleTimingSeconds  =  240;  //Ciklus  u  sekundama  nakon  kojeg
čitamo //                        //vrijednost senzora
long startTiming = 0;
long elapsedTime = 0;
float light ;
float soilMoist;
float humidity;
float temperature;
int Auto = 1;
```

Nakon definiranja svih varijabli potrebnih u programskom kodu u petlji `void setup()` inicijalizirani su parametri potrebni za ispravan rad programa. Petlja se izvršava samo jednom prilikom prvog pokretanja programa. U petlji se definiraju pinovi kao izlazni, pokreće serijska komunikaciju, spajanje na Wi-Fi mrežu, uspostavljanje MQTT komunikacije, inicijalno očitanje podatka sa senzora te se pokreće tajmer.

```
void setup()
{
  pinMode(LAMP, OUTPUT);
  pinMode(PUMP, OUTPUT);
  pinMode(HEATER,OUTPUT);
```

```
  pinMode(FAN,OUTPUT);
  Serial.begin(115200);  //Serial monitor za debugiranje
  Serial1.begin(9600);   //Inicjializacija serijske komunikacije ESP
modula
  WiFi.init(&Serial1); // Inicijalizacija ESP modula


  dht.begin();
  connectWiFi();
  MQTTsubscribe();
  mqtt.will("promjena","prekinuto"); //poruku koju šaljemo u slučaju
                          //prekida MQTT veze
  MQTT_connect();

  readSensorsAndPublish ();
  startTiming = millis();



}



/*************************WiFi&MQtt ******************************/

void connectWiFi()
{
  // Provjera ako je prisutian WIFI modul
  if (WiFi.status() == WL_NO_SHIELD) {
    //Serial.println("WiFi modul nije prisutan");
    while (true);
  }

  // Pokušaj spajanja na WIFi
  while ( status != WL_CONNECTED) {
    //Serial.print("Attempting to connect to WPA SSID: ");
   // Serial.println(ssid);
    // Spajanje na WPA/WPA2 network
    status = WiFi.begin(ssid, pass);
  }

  //Serial.println("Spojeni se na mrežu:");

 // Serial.println(WiFi.localIP());
}
```

Funkcija `MQTTsubscribe()` pretplaćuje se na teme s kojih se želi primati poruke .

```
void MQTTsubscribe()
{
  mqtt.subscribe(&tempslider);
  mqtt.subscribe(&humslider);
  mqtt.subscribe(&soilslider);
  mqtt.subscribe(&svijetslider);
  mqtt.subscribe(&gumb);
```

```
}

/**********************looppetalja  ****************************/
```

Nakon inicijalizacije parametara u `void setup()` petlji, potrebno je u petlji `void loop()` definirati programski kod koji će se ponavljati ciklički. U daljnjem programskom kodu definirane su funkcije koje će se koristi u `void loop()` petlji. Unutar petlje pozivana je funkcija `MQTT_connect()` koja održava MQTT komunikaciju između Arduino UNO platforme i MQTT brokera, funkcija `sub()` koja provjerava da li je primljena nova poruke sa pretplaćenih tema. Unutar petlje korišten je tajmer koji svake 4 minute pokreće funkciju `readSensorsAndPublish()` te ona očitava podatke sa senzora pa ih nakon očitanja objavljuje na teme koje se definirane pomoću MQTT protokola, te funkciju `autoControlPlantation()` koja prema zadanim parametrima upravlja izvršnim članovim sustava.

```
void loop()
{

 elapsedTime = millis()-startTiming;

 MQTT_connect();

 sub();



 if (elapsedTime > (sampleTimingSeconds*1000))
  {
    readSensorsAndPublish ();
    autoControlPlantation();
    startTiming = millis();
  }

 if(! mqtt.ping()) {
    mqtt.disconnect();
  }

if( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network
    status = WiFi.begin(ssid, pass);
  }
}



/*****************************MQTT spajanje ************************/

void MQTT_connect() {
  int8_t ret;
```

```
  // Stop if already connected.
  if (mqtt.connected()) {
    return;
  }

  //Serial.print("Connecting to MQTT... ");

  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for
connected
       // Serial.println(mqtt.connectErrorString(ret));
      //  Serial.println("Retrying MQTT connection in 5 seconds...");
        mqtt.disconnect();
        delay(5000);  // wait 5 seconds
        retries--;
        if (retries == 0) {
          // basically die and wait for WDT to reset me
          while (1);
        }
  }
  Serial.println("MQTT Connected!");
  prom.publish("Spojeno");
 }
/*****************************Senzori*********************************/
```

Kako bi se očitala vrijednost sa LDR senzora potrebno je napisati funkciju koja svaki puta kada je pozvana očitava 10 puta vrijednost koristeći `analogRead` naredbu s pina na koji je spojen te izračunava srednju vrijednost. Nakon toga se pomoću naredbe `map` skalira od 0% do 100%.

```
int getLumen(int anaPin)
{
  int anaValue = 0;
  for(int i = 0; i < 10; i++)
  {
    anaValue += analogRead(anaPin);
    delay(50);
  }

  anaValue = anaValue/10;
  anaValue = map(anaValue, 1023, 0, 0, 100);

  return anaValue;
}
```

Za očitavanje vrijednosti senzora vlažnosti tla korištena je slična funkcija kao i za LDR senzor. Razlika je u tome što ovdje ne skalira vrijednosti od 0 do 1023 zbog toga što je procijenjeno kada senzor daje vrijednost 300 zapravo suho tlo, a kada daje vrijednost 1000 tlo najveće vlažnosti te su zbog toga te vrijednosti skalirane od 0% do 100%.

```
int getSoilMoist(int anaPin)
{
 int anaValue2 = 0;
 anaValue2 = analogRead(anaPin);

 anaValue2 = map(anaValue2, 1000, 300, 0, 100);
  return anaValue2;
  }
```

Funkcija `readSensorsAndPublish` poziva funkcije za očitavanje senzora te se poziva nakon što odbroji tajmer koji je postavljen. Kada očita vrijednosti sa senzora objavljuje ih pomoću naredbe publish na točno određenu temu koju MQTT broker prosljeđuje serveru te se prikazuje na web sučelju.

```
void readSensorsAndPublish(void)
{
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  light = getLumen(LDR_PIN);
  soilMoist = getSoilMoist(soilm_pin);

   tem.publish(temperature);
          delay(10);
   hum.publish(humidity);
          delay(10);
   ldr.publish(light);
          delay(10);
   soilm.publish(soilMoist);
          delay(10);}
```

U `void loop` funkciji cijelo vrijeme se poziva funkcija `void sub` koja provjerava da li je primljena nova poruku s pretplaćenih tema.

Ako primi poruku, provjerava sadržaj poruke te dalje izvršava funkcije koje su određene . Slanje poruka sa web sučelja omogućeno je samo kada je isključeno automatsko upravljanje.

```
void sub(){
   Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(5000))) {

    if (subscription == &tempslider) {
      prom.publish("Vrijednost promijenjena");
      zad_temp= atoi((char *)tempslider.lastread);
    }

    if (subscription == &humslider) {
      prom.publish("Vrijednost promijenjena");
       zad_vla = atoi((char *)humslider.lastread);
```

```
if (subscription == &soilslider) {
     prom.publish("Vrijednost promijenjena");
     suho_tlo = atoi((char *)soilslider.lastread);

}

 if (subscription == &svijetslider) {
     prom.publish("Vrijednost promijenjena");
     mrak = atoi((char *)svijetslider.lastread);

}

if (subscription == &gumb) {

                if (strcmp((char *)gumb.lastread, "ONA") == 0) {
                   Auto = 1;
                   prom.publish("Auto ON");
                   }

                  if (strcmp((char *)gumb.lastread, "OFFA") == 0) {
                  Auto = 0;
                prom.publish("AUTO OFF");
                  }
                  if (strcmp((char *)gumb.lastread, "ONN") == 0) {
                  digitalWrite(PUMP, LOW);
                prom.publish("PUMP ON");
                  }
                  if (strcmp((char *)gumb.lastread, "OFFN") == 0) {
                    digitalWrite(PUMP, HIGH);
                  prom.publish("PUMP OFF");
                  }
                  if (strcmp((char *)gumb.lastread, "ONS") == 0) {
                    digitalWrite(LAMP, LOW);
                  prom.publish("LAMP ON");
                  }
                  if (strcmp((char *)gumb.lastread, "OFFS") == 0) {
                    digitalWrite(LAMP, HIGH);
                  prom.publish("LED OFF");
                  }

                  if (strcmp((char *)gumb.lastread, "ONG") == 0) {
                    digitalWrite(HEATER, HIGH);
                  analogWrite(FAN,64);
                    prom.publish("HEAT ON");
                  }
                  if (strcmp((char *)gumb.lastread, "OFFG") == 0) {
                    digitalWrite(HEATER, LOW);
                  analogWrite(FAN,0);
                    prom.publish("HEAT OFF");
                  }
            }

      }
 }
```

Kada je uključeno automatsko upravljanje tajmer koji je postavljen, nakon isteklog vremena poziva funkciju `autoControlPlantation()` koja kontrolira izvršne članove prema postavljenim parametrima.

```
void autoControlPlantation()
{

if(Auto == 1){

     if(l < mrak)
           {digitalWrite(LAMP, LOW);
             prom.publish("LED ON");
                       }

     else {digitalWrite(LAMP, HIGH);
            prom.publish("LED OFF");
                    }


       if (soilMoist < suho_tlo && l < mrak)
            {digitalWrite(PUMP, LOW);
               prom.publish("PUMP ON");

                  }
      else{
            digitalWrite(PUMP, HIGH);
              prom.publish("PUMP OFF");    }
         delay(10);


  if (t < zad_temp-0.5  || t > zad_temp+0.5){void heaterfan();}



  else{digitalWrite(HEATER, LOW);
  digitalWrite(FAN, LOW);}
  delay(10);

    }
}
```

Funkcija `heaterfan()` poziva se kada je temperature plastenika 0.5℃ veća ili manjaod željene. Prema algoritmu se pali grijač i regulira brzina vrtnje ventilatora ako je potrebno zagrijavati, te ako je potrebno hladiti pali se samo ventilator.

```
void heaterfan()
{


if( t < zad_temp - 3 )
  {digitalWrite(HEATER,HIGH);
```

```
            analogWrite(FAN,255);
prom.publish("HEAT on");


  }

 if( t < zad_temp - 2  )
        {digitalWrite(HEATER,HIGH);
         analogWrite(FAN,127);
         delay(10);
prom.publish("HEAT on");
       }

 if( t < zad_temp - 0.6  )
        {digitalWrite(HEATER,HIGH);
         analogWrite(FAN,64);
         delay(10);
prom.publish("HEAT On");
          }


        if( t > zad_temp +0.6 )
        {digitalWrite(HEATER,LOW);
         analogWrite(FAN,64);
         delay(10);
prom.publish("COLLING ON");
       }
if( t > zad_temp +2 )
        {digitalWrite(HEATER,LOW);
         analogWrite(FAN,125);
         delay(10);
prom.publish("COLLING ON");

       }
if( t > zad_temp +3 )

        {

          digitalWrite(HEATER,LOW);
          delay(10);
          analogWrite(FAN,255);
prom.publish("COLLING ON");
}
     }
```

5.5. Realizacija web sučelja pomoću Node-Red programskog alata

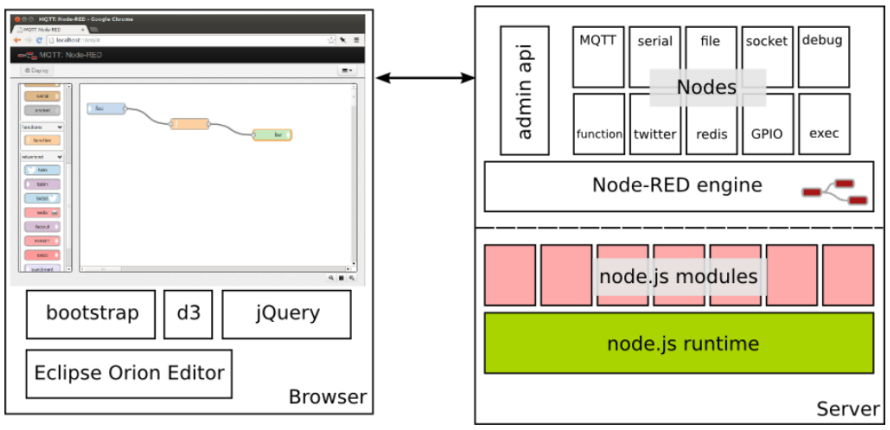
U izradi web sučelja za upravljanje i vizualizaciju korišteni je alat Node-RED.[35]

Node-RED je alat za izgradnju Internet of Things (IoT) aplikacija s naglaskom na pojednostavljenju "spajanja" dijelova koda za obavljanje zadataka. Koristi vizualni programski pristup koji programerima omogućuje povezivanje unaprijed definiranih blokova koda, poznatih

kao čvorovi(nodes), kako bi obavili zadatak. Spojeni čvorovi, obično kombinacijom ulaznih čvorova, čvorova za obradu i izlaznih čvorova , povezani zajedno, čine tokove(flows).

Node-RED je stvoren kao open source projekt skupine IBM Emerging Technologies.U početku su stvorili Node-RED kao alat za sebe jer su radili na IoT projektima tražeći jednostavniji proces povezivanja sustava i senzora pri izradi proof-of-concept modela za kupce.Prva verzija Node-RED-a objavljena je kao open source projekt krajem 2013. godine. Do danas je Node-red skupio aktivnu zajednicu korisnika i programera, koja sa skupinom IBM Emerging Technologies radi na poboljšanju Node-RED-a. Zbog svoje jednostavnosti i brzine pogodan je za pokretanja na uređajima kao Raspberry Pi-u ili u oblaku na servisima kao IBM Bluemix, Amazon Web Services, Microsoft Azure.

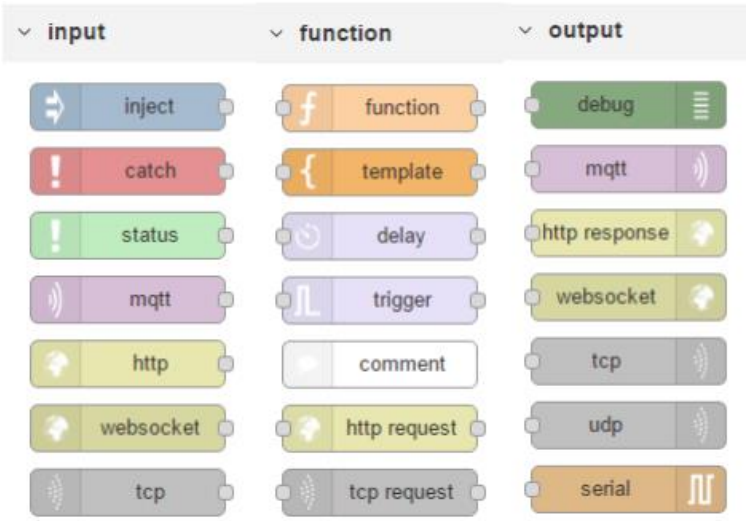
ArhitekturaNode-RED-a je bazirana na node.js platformi. Node.js je platforma za izradu brzih i skalabilnih aplikacija, te je izgrađena na „Google V8 JavaScript engine-u“ koji izvodi kod. Aplikacije pisane u Node.js-u pogodne su zbog njihove asinkronosti: jedna radnja ne blokira druge radnje. Node-RED dolazi sa velikim izborom čvorova, no ako korisnik želi neke nove funkcionalnosti , može kreirati male module pomoću JavaScript-a i HTML-a. Korisnik može te module ubaciti u Node-RED pomoć Node Package Manager (NPM). NPM je upravitelj paketa za JavaScript.



Slika 5.8 Arhitektura Node-RED programskog alata

Čvorovi u NR sastoje se od HTML datoteke i jedne JavaScript datoteke. HTML-datoteka definira svojstva čvora, a datoteka JavaScript je odgovorna za način funkcioniranja čvora. Node-

Red nudi veliki izbor pred instaliranih čvorova koji se mogu koristiti, ali moguće je izraditi vlastite čvorove s određenom funkcionalnošću i izgledom.

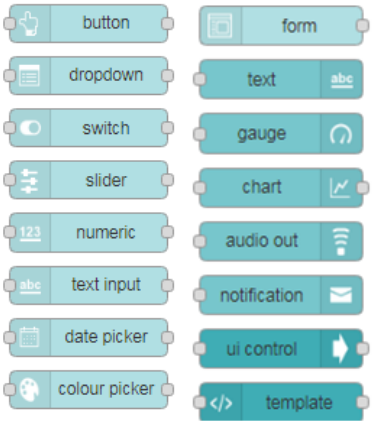


Slika 5.9 Ulazni čvorovi Node-red Programskog alata

Ulazni čvorovi imaju samo priključak za izlaz i mogu se koristiti samo za pokretanje toka. Izlazni čvorovi imaju samo priključak za ulaz i mogu se koristiti samo za prekid toka. Procesni čvorovi imaju ulazni priključak na koji dolaze podaci koji se obrađuju te se šalju na izlani priključak prema daljnjim čvorovima. Uz predinstalirane čvorove za vizualizaciju podataka u stvarnom vremenu koristi se modul „node-red-dashboard“ .

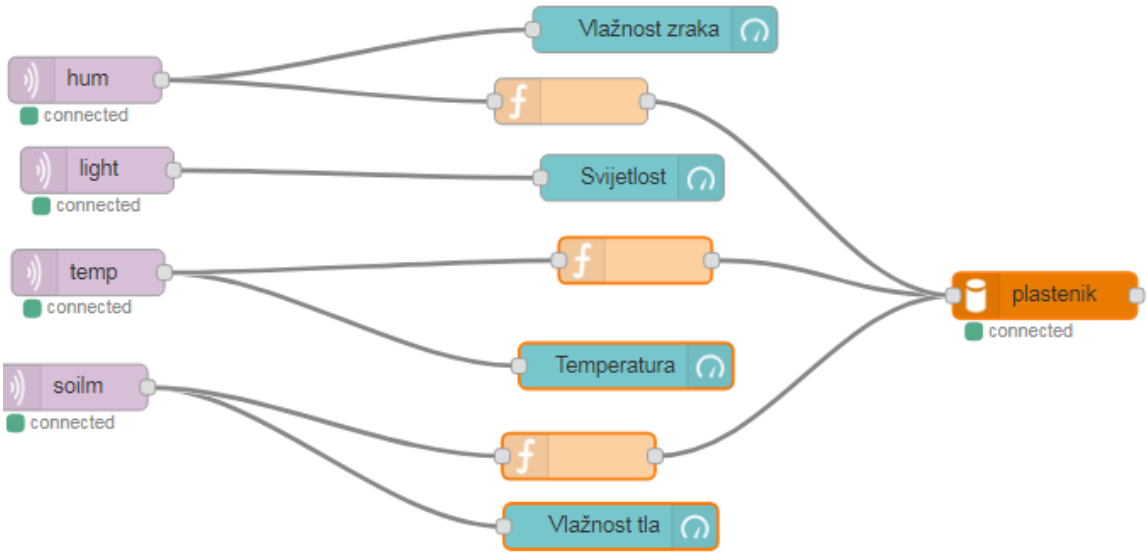
Tok se definira kao skup čvorova koji imaju svoj nezavisni radni ciklus, koji se pokreće paralelno s drugim tokovima.

Koristeći Node-RED vizualni alat za programiranje izrađeno je web sučelje za upravljanje. Da bi se vizualizirali očitani podaci koje očitavamo sa senzora koristimo modul „node-red-dashboard“ . Modul node-red-dashboard sadrži razno ulazne i izlazne čvorove koji su prikazani na slici 5.10.



Slika 5.10 Prikaz čvorova node-red-dashboard modula

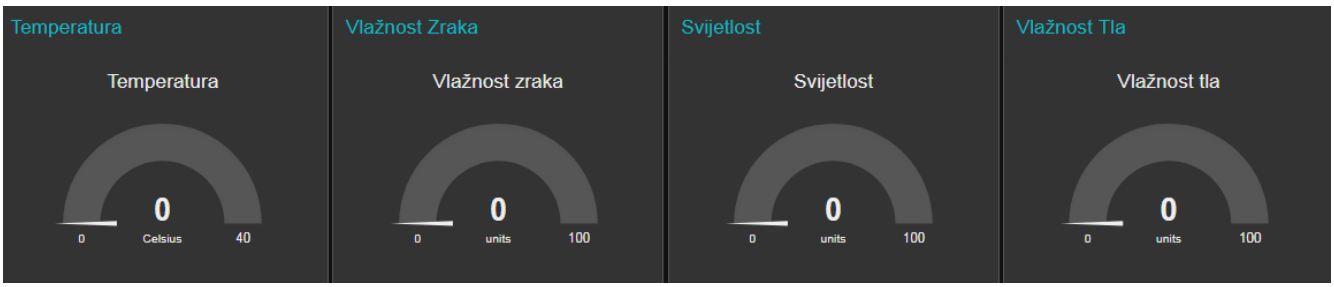
Da bi se vizualizirali te spremili podatci koji se očitavaju sa senzora korišteni su sljedeći čvorovi povezani na način koji je prikazan na slici 5.11.



Slika 5.11 Programski tok vizualizacije dobivenih vrijednosti te spremanja u bazu podataka

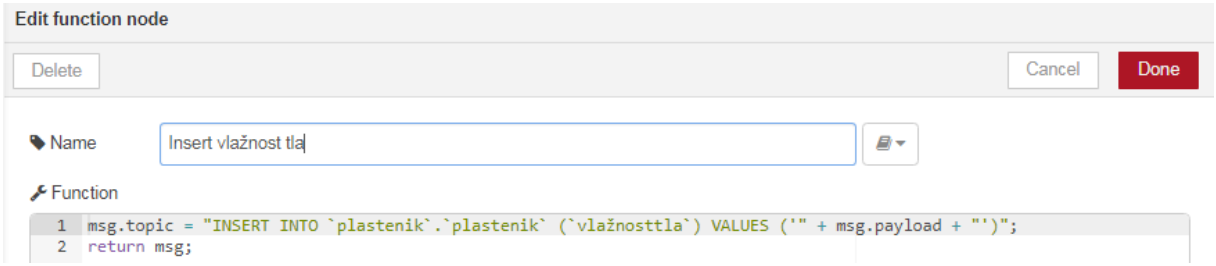
Prvi čvor s lijeve strane na slici je MQTT ulazni čvor koji preko MQTT brokera prima poruke poslane sa Arduino Uno platforme pomoću WiFi modula. MQTT ulazni čvorovi spajaju se na IP adresu MQTT brokera preko kojega se šalju poruke, u našem slučaju to je adresa 192.168.0.101 to jest adresa lokalnog računala na kojem je pokrenuti . Svaki čvor ima definiranu temu za koju prima poruke koje se šalju sa Arduino Uno platforme.

Nakon što primi poruku MQTT ulazni čvor poruku prosljeđuje „gauge“ čvoru koji skalira te vizualizira vrijednost koju je dobio. Na slici 5.12 može se vidjeti kako to izgleda.



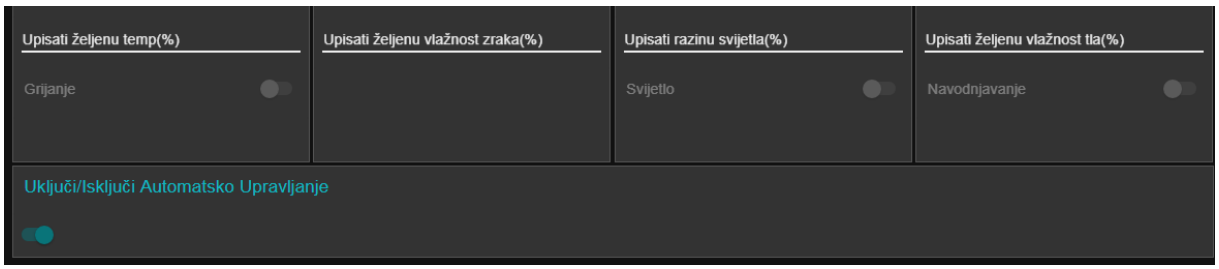
Slika 5.12 Vizualizacija dobivenih vrijednosti

Kako bi očitane podatke spremili za daljnju obradu korištena je MySQL baza podataka. Koristeći MySQL modul unutar Node-RED programskog jezika uspostavlja se spajanje na MySQL bazu podataka te se pomoću jednostavnog MySQL koda dodaju vrijednosti u bazu.



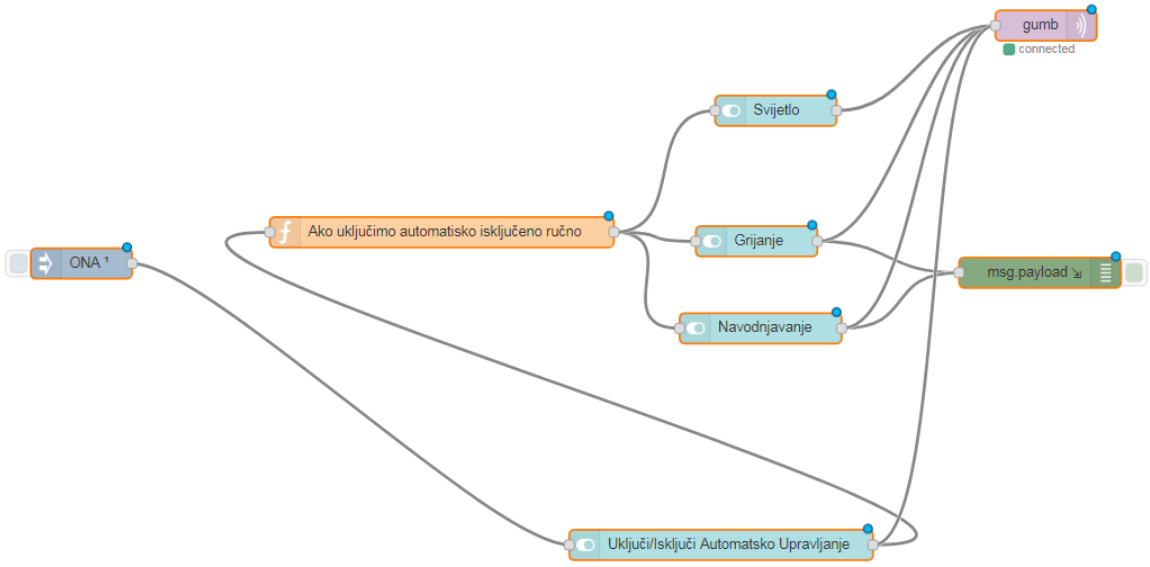
Slika 5.13 Programski kod za SQL bazu podataka

Nakon što su podaci vizualizirano sljedeći korak je upravljanje.



Slika 5.14 Sučelje za ručno upravljanje sustavom

Web sučelje omogućuje korisniku da odabere želi li automatsko ili ručno upravljanje. Ukoliko želi automatsko upravljanje može mijenjati parametre upravljanja. Ukoliko želi ručno upravljanje može zasebno uključiti ili isključiti pojedine izvršne članove. Čvorovi i tok kojima je to ostvareno prikazani su na slici 5.15.

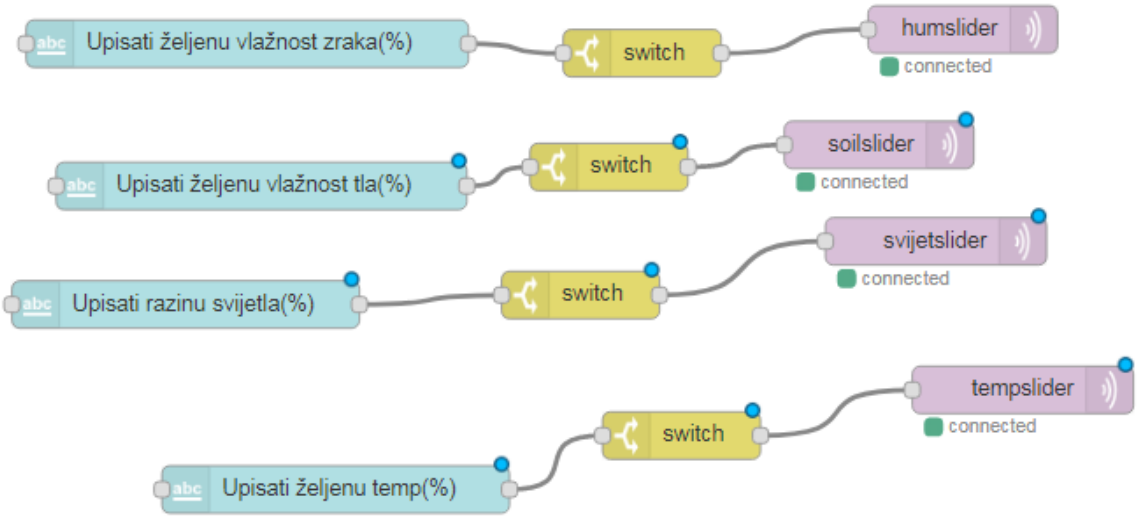


Slika 5.15 Prikaz toka ručnog upravljanja sustavom

Slika prikazuje tok u kojem su korišteni čvorovi „switch“ iz „node-red-dashboard“ modula za upravljanje izvršnim članovima, te MQTT izlazni čvor koji šalje poruke na Arduino UNO platformu. Kada se prvi puta pokreće sustav postavljeno je automatsko upravljanje. Kako bi isključili mogućnost ručnog upravljanja tijekom automatskog, korištena je jednostavna funkcija pisanu u JavaScript programskom jeziku koja onemogućuje „switch“ čvorove.

```
if (msg.payload == "ONA"){
    msg.enabled = false;
    }
else msg.enabled = true;
return msg;
```

Kako bi tijekom automatskog upravljanja korisnik mogao mijenjati parametre upravljanja korišteni je „text-field“ čvor u koji se upisuje željena vrijednost te pomoću MQTT izlaznog čvora šalje na Arduino UNO platformu. Kako bi spriječili slanje krive vrijednosti između ta dva čvora korišteni je „switch“ čvor iz osnovnog modula koji MQTT izlaznom modulu jedino šalje vrijednosti iz zadanog opsega.



Slika 5.16 Prikaz toka mjenjanja vrijednosti parametara

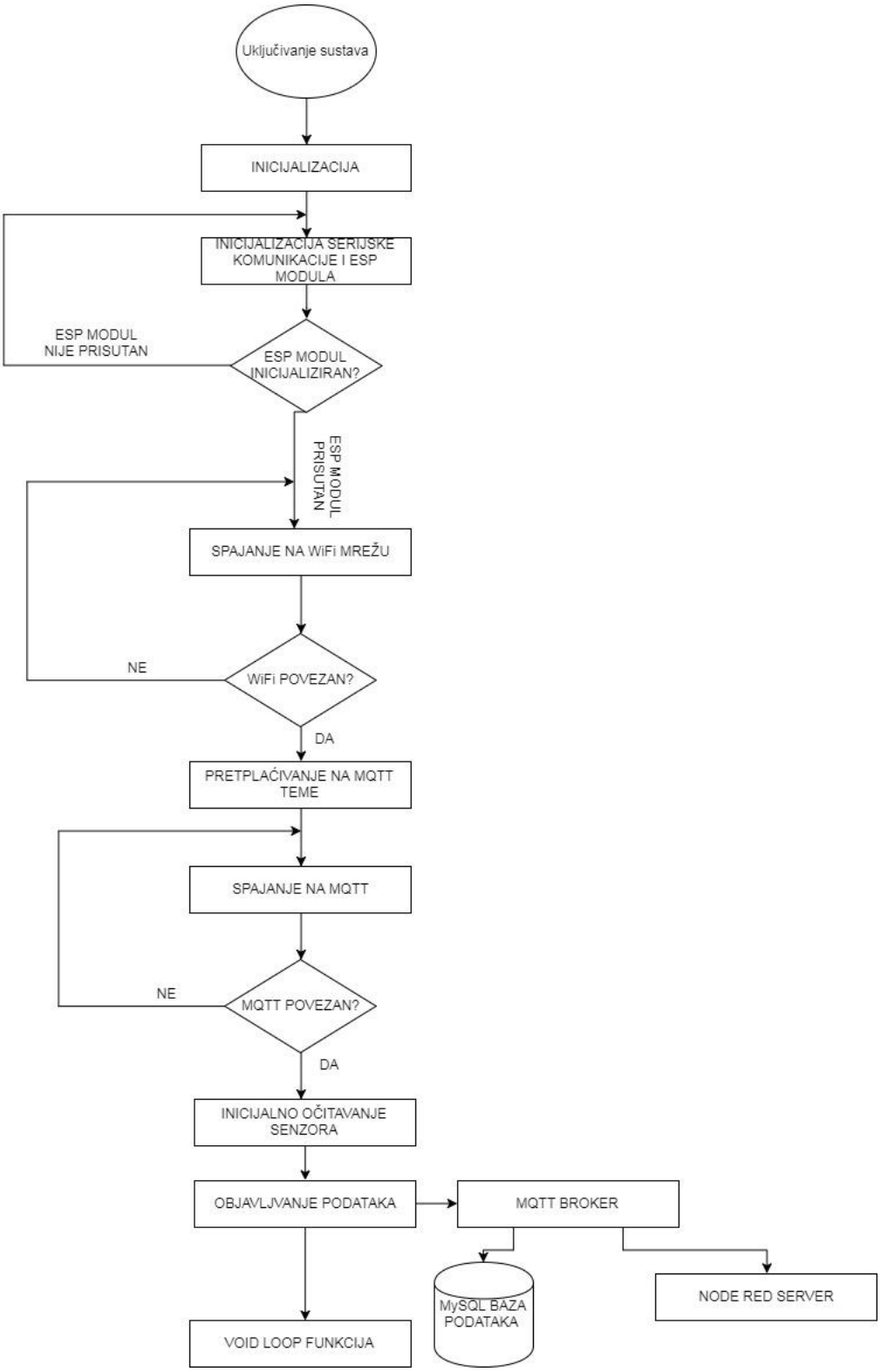
Kako bi korisnik bio obaviješteni u kakvom su stanju izvršni članovi korišten još jedan MQTT ulazni čvor koji je spojen na „template“ čvor. Template čvor sadrži JavaScript kod koji prima poruke te ih prikazuje na web sučelju. Između njih je „switch“ čvor koji usmjerava pojedine poruke na ispravan „template čvor„. JavaScript kod je prikazan u daljnjem tekstu.

```
{{msg.payload}}
<script>
    (function(scope) {
        console.log('Position 1');
        console.dir(scope);
        console.log(scope.msg);
        scope.$watch('msg.payload', function(data) {
            console.log('Position 2');
            console.dir(data);
        });
    })(scope);
</script>
```

Također kako bi korisnik bio obaviješteni o trenutnim promjenama, na primjer kada se u ručnom načinu rada uključi neki izvršni član korišten je „notification“ čvor koji na 5 sekundi prikazuje poruku koju mu proslijedi MQTT ulazni čvor.

5.6. Prikaz rada sustava automatskog upravljanja plastenikom

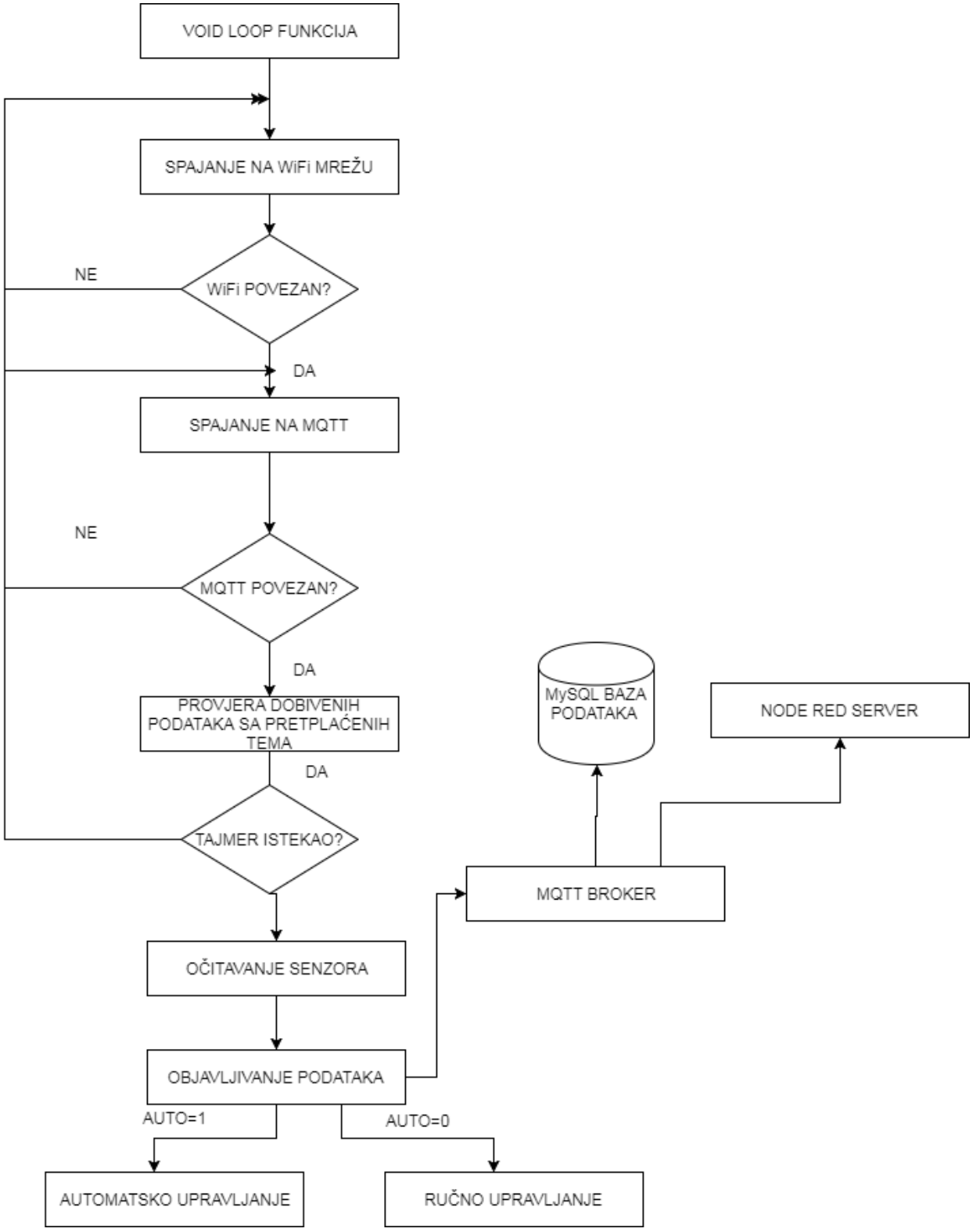
Rad sustava pobliže je objašnjen dijagramom toka.



Slika 5.17 Dijagram toka početka rada sustava

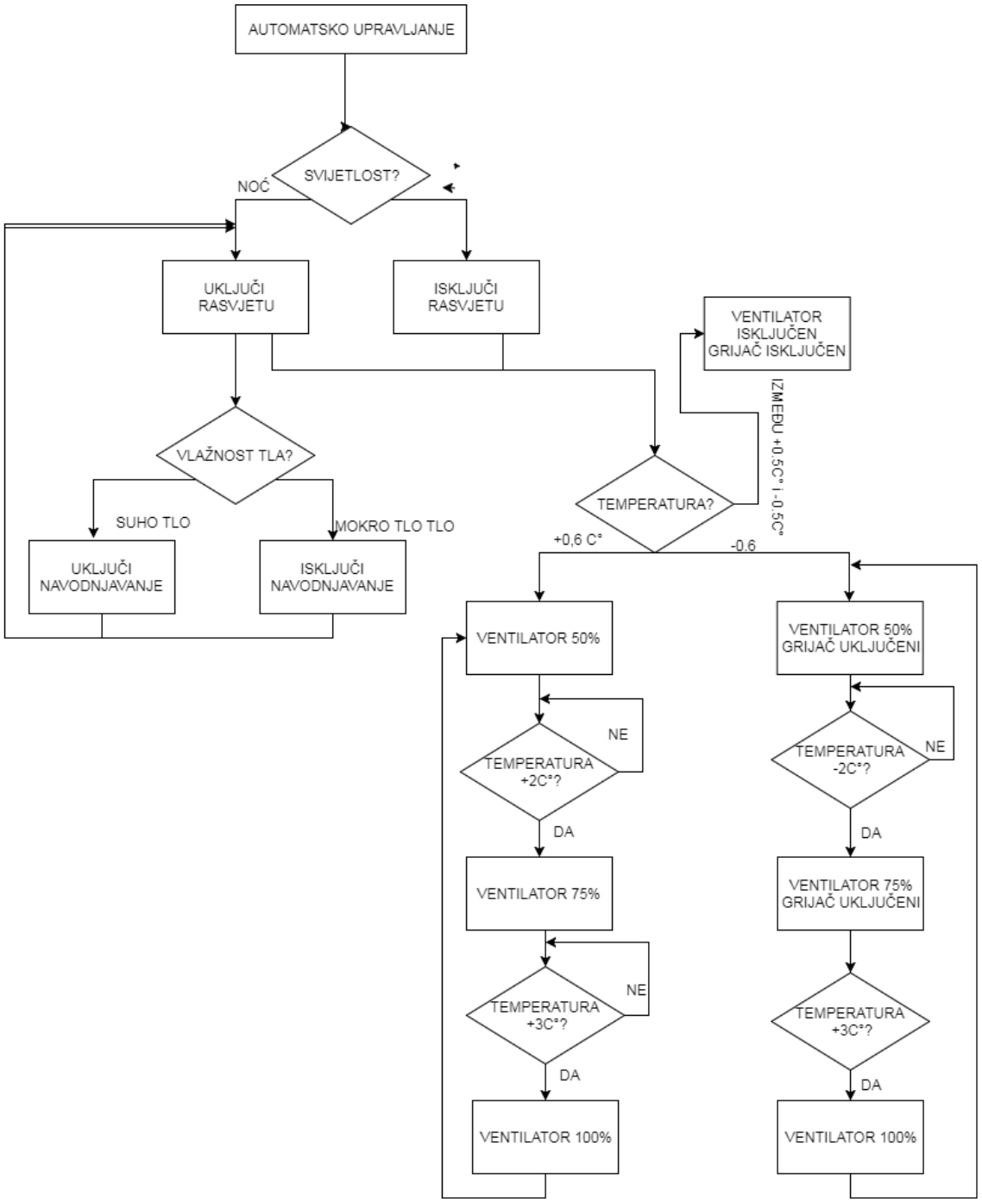
Kada se sustav prvi puta pokreće na početku se inicijaliziraju potrebne varijable i funkcije za daljnji rad sustava. Nakon inicijalizacije varijabli i funkcija, inicijalizira se serijska komunikacija te provjerava da li je ESP modul prisutan. Ako je prisutan spaja se na WiFi mrežu. Nakon

uspješnog spajanja slijedi pretplaćivanje na teme te spajanje na MQTT brokera. Nakon toga očitavaju se inicijalne vrijednosti senzora koje se dalje šalju u MySQL bazu podataka te na Node Red server gdje se vizualiziraju. U nastavku se počinje izvršavati funkcija void loop čiji se dijagram toka može vidjeti na slici 5.18.



Slika 5.18 Blok dijagram loop funkcije

U funkciji se cijelo vrijeme provjerava da li je spojeni na WiFi mrežu te MQTT server. Nakon toga provjerava se da li su primljene poruke sa pretplaćenih tema. Poslije se provjerava da li je istekao postavljeni tajmer, kada tajmer istekne očitavaju se senzori te objavljuju podaci u bazu podataka i Node Red server. Ukoliko je uključeno automatsko upravljanje sustav uključuje i isključuje izvršne članove prema zadanim parametrima. Blok dijagram se može vidjeti na slici 5.19.



Slika 5.19 Blok dijagram automatskog upravljanja

Na slici vidimo kako se prema zadanim parametrima isključuju te uključuju izvršni članovi.

Ako je umjesto automatskog upravljanja odabrano ručno izvršni članovi se uključuju te isključuju prema dobivenim naredbama sa web sučelja.

6. Zaključak

Kako će potreba za svježim voćem i povrćem rasti tako će se i poljoprivrednici s obzirom na nepredvidive vanjske klimatske uvjete početi više baviti proizvodnjom u plastenicima. Kako bi u takvoj proizvodnji povećali prinose početi će koristi sustave automatskog upravljanje proizvodnje pomoću neke vrste mikrokontrolera. Kako je u ovom radu korišteni Arduino UNO sustav koji je prihvatljiv cijenom te jednostavnošću dobivena je relativno moćnu te jeftina platforma koju je moguće nadograditi. Korištenjem ESP8266 modula koji je također cjenovno prihvatljiv dobivena je mogućnost udaljenog nadgledanja procesa što će uštedjeti vrijeme poljoprivrednicima koji imaju velike površine. Node-Red programskim alatom dobiveno je jednostavno, a moćno web sučelje koje nije teško savladati te se lako modificira prema potrebama korisnika. Kako bi sustav bio još bolji te davao veće prinose možemo uključiti senzor koji mjeri CO_2 u zraku kako bi pratili i poboljšali kvalitetu zraka.

Projektiranjem sustava automatskog upravljanja proizvodnjom u plastenicima stečena su nova znanja i iskustva iz područja programiranja Arduino platforme te automatskog upravljanja sustavom.

Kako bi nadogradili sustav u sljedećim verzijama moguće je korištenje ESP8266 ESP32 verzije koja cijenom i snagom može zamijeniti Arduino UNO ali u ovom trenutku nije bila dostupna. Također kao nadogradnju može se dodati web kameru koju je također moguće integrirati u web sučelje te tako pratiti događaje u plasteniku.

U Varaždinu _____ _____

7. Literatura

[1] https://cecaradlovacki.wordpress.com/2015/04/10/sta-je-pwm[01.07.2017]
[2] https://bib.irb.hr/datoteka/776357.Mogunost_proizvodnje_malina_i_kupina_u_zatienom_prostoru_Antonija_Bakovi.pdf[13.07.2017]
[3] http://www.obz.hr/hr/pdf/poljoprivredni_info_pult/2010/Za%C5%A1ti%C4%87eni%20prostori-plastenici%20i%20staklenici.pdf[13.07.2017]
[4] https://repozitorij.unipu.hr/islandora/object/unipu%3A199/datastream/PDF/view[13.07.2017]
[5] https://en.wikipedia.org/wiki/Wi-Fi[05.07.2017]
[6] https://mro.massey.ac.nz/bitstream/handle/10179/3231/02_whole.pdf[18.6.2017]
[7] http://www.sensaphone.com/industries/greenhouse.php [11.05.2017]
[8] http://www.growtronix.com/cart/1-large_default/growtronix-base-system.jpg[11.05.2017]
[9] https://www.growtronix.com/cart/blog/how-growtronix-works-n5[11.05.2017]
[10] http://www.sensaphone.com/images/application_guides/new/alIMb.jpg [11.05.2017]
[11] https://hr.wikipedia.org/wiki/Termoelektri%C4%8Dni_generator [01.07.2017]
[12] http://bildr.org/2012/03/rfp30n06le-arduino/[22.04.2017]
[13] http://bildr.org/blog/wp-content/uploads/2012/03/rfp30n06le-arduino-motor-400x218.png[22.04.2017]
[14] https://tkkrlab.nl/wiki/Arduino_KY-018_Photo_resistor_module
[15] https://tkkrlab.nl/w/images/thumb/2/27/Arduino_KY-018_Photo_resistor_module.PNG/400px-Arduino_KY-018_Photo_resistor_module.PNG[25.05.2017]
[16] https://www.arduino.cc/[25.05.2017]
[17] https://communicating99.files.wordpress.com/2017/04/kit1.png[25.05.2017]
[18] http://www.esp8266.com/wiki/doku.php?id=esp8266-module-family[22.04.2017]
[19] https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf[28.04.2017]
[20] https://learn.adafruit.com/adafruit-io/mqtt-api[28.03.2017]
[21] http://www.hivemq.com/blog/how-to-get-started-with-mqtt[28.03.2017]
[22] http://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels[28.03.2017]
[23] https://mosquitto.org/[28.03.2017]
[24] https://learn.sparkfun.com/tutorials/soil-moisture-sensor-hookup-guide[22.04.2017]
[25] http://www.electroschematics.com/6519/simple-soil-moisture-sensor-arduino-project/[22.05.2017]
[26] http://www.electroschematics.com/wp-content/uploads/2015/02/DHT22-PinOut.png[22.05.2017]
[27] http://rancidbacon.com/files/kiwicon8/ESP8266_WiFi_Module_Quick_Start_Guide_v_1.0.4.pdf[22.04.2017]
[28] http://www.pfos.unios.hr/upload/documents/Za%C5%A1ti%C4%87eni%20prostori-plastenici%20i%20staklenici.pdf[29.06.2017]
[29] http://remotexy.com/img/help/start_arduino_esp_cloud/start_arduino_esp_cloud_scheme.png[22.04.2017]
[30] https://rydepier.files.wordpress.com/2015/05/img_1911.png?w=840[05.07.2017]
[31] https://coderwall-assets-0.s3.amazonaws.com/uploads/picture/file/1165/relay.jpg[05.07.2017]
[32] http://www.microchip.ua/wireless/esp01.pdf [05.07.2017]
[33] https://www.farnell.com/datasheets/1682209.pdf[05.07.2017]
[34] https://en.wikipedia.org/wiki/Photoresistor[05.07.2017]
[35] https://nodered.org/[05.07.2017]
[36] http://www.hobbyist.co.nz/?q=interfacing-relay-modules-to-arduino[05.07.2017]

Popis slika

Slika 2.1 Plastenik[28] 7
Slika 2.2 Prikaz parametara praćenih u plasteniku[10] 8
Slika 2.3 Growtronix komercijalni modul[8] 9
Slika 3.1 Prikaz arhitekture ESP8266 ESP01 modula[27] 11
Slika 3.2 Spajanje ESP8266 modula sa Arduino UNO platformom[29] 11
Slika 3.3 Primjer MTT komunikacije[21] 13
Slika 3.4 Primjer imenovanja tema[21] 13
Slika 3.5 QoS 0 razina kvalitete servisa[22] 14
Slika 3.6 QoS 1 razina kvalitete servisa[22] 14
Slika 3.7 QoS 2 razna kvalitete servisa[22] 15
Slika 4.1 Prikaz komponenti Arduino Uno R3 razvojne platforme[20] 16
Slika 4.2 Arduino IDE razvojno sučelje 17
Slika 4.3 DHT22 dijagram pinova[19] 18
Slika 4.4 Senzor vlažnosti tla[30] 19
Slika 4.5 LDR modul[15] 20
Slika 4.6 Četvero kanalni relejni modul[18] 21
Slika 5.1 Shema spajanja sustava automatskog upravljanja proizvodnje u plasteniku 23
Slika 5.2 PWM modulacija [1] 24
Slika 5.3 Peltierov element montiran na hladnjak te smješten u cijev 25
Slika 5.4 Prikaz spajanja MOSFET tranzisora sa Arduino UNO platformom[9] 25
Slika 5.5 Spajanje navodnjavaj kap na pak sa solenoidnim ventilom 26
Slika 5.6 Shema spajanja LED rasvjete 27
Slika 5.7 Maketa plastenika sa posađenim sadnicama paprike 28
Slika 5.8 Arhitektura Node-RED programskog alata 39
Slika 5.9 Ulazni čvorovi Node-red Programskog alata 40
Slika 5.10 Prikaz čvorova node-red-dashboard modula 41
Slika 5.11 Programski tok vizualizacije dobivenih vrijednosti te spremanja u bazu podataka 41
Slika 5.12 Vizualizacija dobivenih vrijednosti 42
Slika 5.13 Programski kod za SQL bazu podataka 42
Slika 5.14 Sučelje za ručno upravljanje sustavom 42
Slika 5.15 Prikaz toka ručnog upravljanja sustavom 43
Slika 5.16 Prikaz toka mijenjana vrijednosti parametara 44
Slika 5.17 Dijagram toka početka rada sustava 45

Slika 5.18 Blok dijagram loop funkcije.....46
Slika 5.19 Blok dijagram automatskog upravljanja.....47

Sveučilište
Sjever

SVEUČILIŠTE
SJEVER

IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

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

Ja, Luka Osredek pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor završnog rada pod naslovom „Automatizacija proizvodnje u plasteniku" te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student:
Luka Osredek

(vlastoručni potpis)

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

Ja,Luka Osredek neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog rada pod naslovom „Automatizacija proizvodnje u plasteniku" čiji sam autor.

Student:
Luka Osredek

(vlastoručni potpis)