

Primjena Interneta stvari za upravljanje proizvodnjom plastenika

Toljan, Marko

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:134468>

Rights / Prava: [Attribution 4.0 International](#) / [Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-04-24**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Marko Toljan

Zagreb, 2021. godina.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

Doc. dr. sc. Miro Hegedić

Student:

Marko Toljan

Komentor:

Dr. sc. Davor Kolar

Zagreb, 2021. godina.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem svojim mentorima na ukazanom povjerenju i izdvojenom vremenu. Također zahvaljujem i ostalim profesorima i zaposlenicima fakulteta s kojima sam se susreo za vrijeme studija.

Hvala i svim mojim prijateljima i kolegama koji su bili uz mene i sa mnom dijelili radost i teškoće studiranja.

Velika zahvala i mojim roditeljima te braći i sestri na neizmjerljivoj podršci i ljubavi koju su mi čitavog života pružali.

Posebna zahvala pripada mojoj zaručnici Dori, koja je bila uz mene svo ovo vrijeme, pružala bezuvjetnu podršku i davala snagu u najtežim trenucima.

Marko Toljan



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske radove studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa: 602-04/21-6/1	
Ur. broj: 15-1703-21	

DIPLOMSKI ZADATAK

Student: **MARKO TOLJAN**

Mat. br.: 0035201294

Naslov rada na hrvatskom jeziku: **Primjena Interneta stvari za upravljanje proizvodnjom plastenika**

Naslov rada na engleskom jeziku: **Application of the Internet of Things to manage greenhouse production**

Opis zadatka:

Uloga plastenika je izvansezonski uzgoj biljaka pri čemu je potrebno kontinuirano održavanje specifičnih klimatskih uvjeta unutar takvih prostora. U većini njih, primjenjuje se ručno upravljanje glavnim parametrima kao što su npr. temperatura i vlaga te vizualni nadzor istih. Nedostatak takvog upravljanja i nadzora je nemogućnost regulacije parametara proizvodnje (klimatskih uvjeta) u stvarnom vremenu što utječe na učinkovitost i kvalitetu proizvodnje. Primjenom tehnologije Interneta stvari (eng. Internet of Things - IoT), moguće je automatizirati proces proizvodnje plastenika na način da se svi ključni parametri prikupljaju putem senzora, a primjenom neizrazite logike učinkovitije i kvalitetnije upravlja procesom u stvarnom vremenu.

U skladu s prethodno navedenim, u radu je potrebno:

1. Detaljno opisati problematiku praćenja stanja i upravljanja proizvodnjom plastenika.
2. Predložiti automatizirano IoT rješenje za praćenje stanja i upravljanje proizvodnjom plastenika.
3. Razviti inteligentni model upravljanja proizvodnjom primjenom neizrazite (fuzzy) logike temeljen na podacima.
4. Na konkretnom primjeru prikazati funkcionalnosti razvijenog rješenja.
5. Na temelju analize sprovedenog istraživanja izvesti zaključak te predložiti smjernice za daljnje istraživanje.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:
6. svibnja 2021.

Rok predaje rada:
8. srpnja 2021.

Predviđeni datum obrane:
12. srpnja do 16. srpnja 2021.

Zadatak zadao:
doc. dr. sc. Miro Hegedić

Komentor:
dr. sc. Davor Kolar

Predsjednica Povjerenstva:
prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS TABLICA.....	VI
POPIS KRATICA	VII
SAŽETAK.....	VIII
SUMMARY	IX
1. UVOD	1
2. UPRAVLJANJE PARAMETRIMA U PLASTENIKU	2
2.1. Parametri važni za razvoj biljaka	2
2.2. Napredne metode upravljanja.....	6
2.2.1. Agrivi.....	6
2.2.2. WaterBit	9
2.2.3. SupPlant.....	11
3. NEIZRAZITA LOGIKA	14
3.1.1. Neizraziti skupovi i funkcije pripadnosti	14
3.1.2. Lingvističke varijable	16
3.1.3. Operacije s neizrazitim skupovima [16].....	17
3.1.4. Neizraziti sustavi.....	18
4. PREGLED KORIŠTENE OPREME	20
4.1. Računalna sklopovska oprema, hardware	20
4.1.1. Mikrokontroleri.....	20
4.1.2. Senzori	23
4.1.3. Aktuatori.....	25
4.2. Software	26

5.	SUSTAV ZA UPRAVLJANJE I PRAĆENJE STANJA U PLASTENIKU	28
5.1.	Definiranje strukture sustava.....	28
5.2.	Prikaz strukture baze podataka.....	29
5.3.	Prikaz i opis upotrebe web stranice za nadzor	31
5.4.	Sučelje za upravljanje aktuatorima na daljinu.....	35
5.5.	Sklop za prikupljanje podataka sa senzora.....	36
5.6.	Pretpostavke rada zasebnih podsustava.....	38
5.6.1.	<i>Regulacija vlažnosti zemlje.....</i>	<i>38</i>
5.6.2.	<i>Regulacija temperature zraka</i>	<i>39</i>
5.6.3.	<i>Regulacija vlažnosti zraka</i>	<i>39</i>
5.7.	Objašnjenje predviđenih načina upravljanja	40
5.7.1.	<i>Ručno upravljanje.....</i>	<i>40</i>
5.7.2.	<i>Automatsko praćenje ručno postavljenih granica.....</i>	<i>42</i>
5.7.3.	<i>Neizrazito upravljanje.....</i>	<i>45</i>
5.8.	Modeliranje FIS-a	46
5.8.1.	<i>Funkcije pripadnosti</i>	<i>47</i>
5.8.2.	<i>Pravila neizrazitog sustava upravljanja.....</i>	<i>51</i>
5.9.	Izrađeni model za demonstraciju rada sustava	54
6.	ZAKLJUČAK	58
	LITERATURA.....	60

POPIS SLIKA

Slika 1.	Uređaj za grijanje zraka [2]	2
Slika 2.	Sustav za prirodnu ventilaciju [3]	3
Slika 3.	Sjenila na krovu plastenika [4]	3
Slika 4.	Raspršivanje maglice za hlađenje [6]	4
Slika 5.	Navodnjavanje kišenjem [8]	5
Slika 6.	Lokalizirano navodnjavanje „kap na kap“ [9]	5
Slika 7.	Primjer mikro prskalice	6
Slika 8.	Unos kulture i pripadajuće površine [11]	7
Slika 9.	Nadzorna ploča s ključnim pokazateljima [11]	8
Slika 10.	WaterBit Carbon™ - modularna jedinica za prikupljanje podataka [13]	10
Slika 11.	Sučelje za postavljanje rasporeda navodnjavanja [13]	10
Slika 12.	Prikaz korisničkog sučelja za pregled podataka [13]	11
Slika 13.	Senzor promjene promjera ploda (lijevo) i dendrometar (desno) [15]	12
Slika 14.	Senzor temperature lista (lijevo) i meteorološka stanica (desno) [15]	12
Slika 15.	Tjedni pregled izmjerenih parametara [15]	13
Slika 16.	Upravljanje upozorenjima o ekstremnim uvjetima [15]	13
Slika 17.	Funkcije pripadnosti prikazanih skupova	15
Slika 18.	Funkcija pripadnosti s karakterističnim točkama [16]	15
Slika 19.	Različite funkcije pripadnosti [16]	16
Slika 20.	Operacije s neizrazitim skupovima [16]	18
Slika 21.	Tijek procesa u neizrazitom sustavu	19
Slika 22.	Atmel ATmega328 mikrokontroler [18]	20
Slika 23.	NodeMCU DEVKIT 1.0 ESP-12E [19]	21
Slika 24.	NodeMCU ESP32 [21]	22
Slika 25.	DHT11 [25]	23
Slika 26.	Mjerni elementi u modulu, termistor (gore) i kondenzator (dolje)	24
Slika 27.	SEN0193 kapacitivni senzor	25
Slika 28.	4-kanalni 5V relej [29]	26
Slika 29.	Shematski prikaz međusobnih odnosa entiteta	29
Slika 30.	Struktura baze	30

Slika 31.	Tablica t_lokacije	30
Slika 32.	Tablica t_senzori	31
Slika 33.	Tablica s izmjerenim vrijednostima	31
Slika 34.	Naslovna strana	32
Slika 35.	Forma za promjenu granica	33
Slika 36.	Filtriranje i prikaz podataka (1).....	33
Slika 37.	Filtriranje i prikaz podataka (2).....	34
Slika 38.	Izvještaj u PDF formatu	35
Slika 39.	Sučelje za upravljanje aktuatorima.....	36
Slika 40.	Kod za slanje GET zahtjeva pomoću prekidača.....	36
Slika 41.	Arduino kod za čitanje i slanje podataka sa senzora	37
Slika 42.	PHP skripta za unos podataka u bazu.....	38
Slika 43.	Shema ručne kontrole pumpe za navodnjavanje	40
Slika 44.	Arduino kod za ručno upravljanje	41
Slika 45.	Arduino kod za čitanje stanja releja	41
Slika 46.	PHP skripta za provjeru stanja releja	42
Slika 47.	Arduino kod za automatsko upravljanje.....	43
Slika 48.	Brojač za određivanje stanke u programu	44
Slika 49.	PHP skripta za čitanje zadnjeg unosa iz baze.....	44
Slika 50.	Arduino kod za raščlanjivanje stringa	45
Slika 51.	Arduino kod za neizravno upravljanje	46
Slika 52.	<i>Fuzzy Logic Designer</i>	47
Slika 53.	<i>Membership function editor</i> – funkcija pripadnosti za vlažnost zraka.....	48
Slika 54.	Funkcije pripadnosti – temperatura zraka	49
Slika 55.	Funkcije pripadnosti – vlažnost tla.....	49
Slika 56.	Funkcije pripadnosti – ventilator.....	49
Slika 57.	Funkcije pripadnosti – difuzor	50
Slika 58.	Funkcije pripadnosti – pumpa	50
Slika 59.	Funkcije pripadnosti – grijač	50
Slika 60.	<i>Rule Editor</i> – pravila sustava.....	52
Slika 61.	<i>Rule Viewer</i> – vizualni prikaz inferencije	53
Slika 62.	Model platenika	54

Slika 63.	Konzola s gumbima.....	55
Slika 64.	Konzola – 3D model, Catia	55
Slika 65.	Teglica sa senzorom i prskalicama.....	56
Slika 66.	Prikaz opreme u modelu.....	57

POPIS TABLICA

Tablica 1. Pinovi ESP-12E modula [20]	21
Tablica 2. Specifikacije DHT11 modula [27]	24
Tablica 3. Rasponi vrijednosti u <i>fuzzy</i> modelu	51
Tablica 4. Ulazi i izlazi za rubne uvjete	53

POPIS KRATICA

Kratika	Značenje kratice
ET	Evaporperspiracija (eng. <i>evaporperspiration</i>)
IDE	Integrirano razvojno okruŹje (eng. <i>Integrated development environment</i>)
IoT	Internet stvari (eng. <i>Internet of Things</i>)
LVDT	Linearni varijabilni diferencijalni transformator (eng. <i>Linear Variable Differential Transformer</i>)
SoC	Sustav na čipu (eng. <i>System on a Chip</i>)
VPD	Deficita vlaŹnosti zraka (eng. <i>vapor pressure deficit</i>)
WAN	Širokopojasna mreŹa (eng. <i>Wide-Area Network</i>)

SAŽETAK

U radu su detaljno opisani problematika praćenja stanja zraka i tla u sustavu plastenika te razrađeni zahtjevi i preduvjeti za kontrolu tog stanja i upravljanje samim funkcioniranjem plastenika. Kroz rad je prikazan razvoj niskobudžetnog rješenja, prvo kroz sustava za nadzor, a zatim i nadogradnjom tog sustava funkcionalnostima potrebnim za upravljanje. Predložene su tehnologije i komponente pogodne za rješavanje navedenog problema, a razvijena je i platforma za korištenje naprednih modela upravljanja pomoću neizrazite (fuzzy) logike. Za demonstraciju prijedloga je izrađena maketa plastenika sa svim funkcionalnostima.

Ključne riječi: neizrazito upravljanje, fuzzy, internet stvari, upravljanje plastenikom, Arduino

SUMMARY

The thesis describes in detail the challenges of monitoring of air and soil parameters in the greenhouse system and elaborates the requirements and prerequisites for controlling said condition and managing the functioning of the greenhouse. The paper presents the development of a low-budget solution, first through the monitoring system, and then by upgrading that system with the functionalities needed for the function of control. Technologies and components suitable for solving this problem have been proposed, and a platform for the use of advanced control models using fuzzy logic has been developed. To demonstrate the proposal, a model of a greenhouse with all the functionalities was made.

Keywords: fuzzy control, Internet of Things, greenhouse control, Arduino

1. UVOD

Internet stvari (eng. *Internet of Things, IoT*) je skup tehnologija koje za cilj imaju povezivanje različitih predmeta u cjelinu koju se s jednog mjesta može nadzirati i njome upravljati, ili je pak to upravljanje automatizirano na temelju podataka koji se prikupljaju. Primjena interneta stvari omogućava znatan tehnološki napredak u mnogim sustavima što konačno dovodi do više dodane vrijednosti i nižih troškova.

Jedno područje u kojem svakako ima prostora za tehnološki napredak i upotrebu suvremenih digitalnih tehnologija je poljoprivreda. U ovom radu je odabran specifičan dio poljoprivredne proizvodnje, a to je proizvodnja u zaštićenim objektima, točnije plastenicima. Plastenici se posebno ističu po visokim zahtjevima jer se svi parametri kojima se upravlja moraju kontinuirano držati van raspona koji su prisutni u prirodi, a najčešće se njima upravlja ručno, tj. bez eksplicitno izražene logike upravljanja. U ovom diplomskom radu, teorijski i praktično je prikazana mogućnost automatizacije upravljanja radom plastenika primjenom IoT-a na način da se pomoću senzora mjere i prikupljaju ključni parametri procesa. Prikupljeni i pohranjeni podaci iskorišteni su kao ulazni podaci za upravljanje radom uz primjenu modela neizrazite logike.

U ranijim poglavljima je opisan način funkcioniranja klasičnih plastenika, tehnologije koje se koriste za upravljanje te okvirni zahtjevi poljoprivredne proizvodnje. Također je dan pregled već prisutnih rješenja koja primjenom digitalnih tehnologija nude unaprjeđenja, od sustava za praćenje svih aktivnosti u sustavu, sustava za nadzor i pomoć pri odlučivanju do rješenja za inteligentno upravljanje. Kao početak praktičnog dijela je dan pregled korištene opreme uz objašnjenje načina rada i detaljne specifikacije. Središnji dio rada se bavi razvojem predloženog rješenja, počevši s pregledom ranije razvijenog sustava za prikupljanje podataka, zatim opisom implementacije predložene opreme te konačno razvojem modela upravljanja, s posebnim fokusom na neizrazito upravljanje. Zaključno, analizirani su rezultati rada te uočene mogućnosti za unaprjeđenje.

2. UPRAVLJANJE PARAMETRIMA U PLASTENIKU

Najzastupljeniji oblik proizvodnje povrtnarskih i cvjećarskih kultura je proizvodnja u zaštićenim prostorima (plastenicima i staklenicima). Neke od prednosti koje se tim načinom proizvodnje ostvaruju su veći ukupni prinosi zbog proizvodnje van sezone te ranija berba zbog optimiziranja uvjeta [1].

2.1. Parametri važni za razvoj biljaka

Mikroklimatski uvjeti u plasteniku utječu na rast i razvoj biljaka, pojavu bolesti i štetnika te visinu prinosa.

- Temperatura zraka

Izlazak temperature van granica optimalnih vrijednosti dovodi do sporijeg razvoja, a dugotrajno održavanje nepovoljne temperature u konačnici može dovesti i do ugibanja. Nagle promjene temperature također negativno utječu na razvoj. Iz tog razloga, moderni plastenici se redovito opremaju sustavima za grijanje i recirkulaciju zraka te za ventilaciju (provjetravanje) što je i prikazano u slikama u nastavku [1].



Slika 1. Uređaj za grijanje zraka [2]



Slika 2. Sustav za prirodnu ventilaciju [3]

Sam ulazak topline se može djelomično spriječiti sjenilima što prikazuje slika 3..



Slika 3. Sjenila na krovu plastenika [4]

Za odvođenje viška topline se osim ventilacije može koristiti i latentna toplota isparavanja vode jer jedan grama vode isparavanjem spušta temperaturu jednog kubičnog metra zraka za $2,5\text{ }^{\circ}\text{C}$ [5].

Slika 4. prikazuje raspršivanje fine maglice iznad biljaka s ciljem rashlađivanja.



Slika 4. Raspršivanje maglice za hlađenje [6]

- Vlažnost zraka

Relativna vlažnost zraka također ima velik utjecaj u razvoju, kako na sam rast, tako i na pojavu bolesti. Moguća je potreba za dodatnim orošavanjem kao i za smanjenjem vlažnosti ventilacijom [1].

- Sastav zraka

Udio CO_2 u zraku je važan za odvijanje procesa fotosinteze, udio kisika je također bitan za neke procese, gnojdbom dolazi do otpuštanja različitih plinova pa je provjetravanjem potrebno održavati kvalitetu zraka, a posebnim uređajima se može i podešavati udio određenih plinova u zraku [1].

- Vlažnost tla

U proizvodnji u zaštićenim prostorima se na udio vode u zemlji gotovo uvijek mora utjecati navodnjavanjem. Načini navodnjavanja se mogu podijeliti u 4 skupine, svaka sa svojim prednostima i nedostacima [7]:

- Površinsko navodnjavanje – najjeftinija i najučestalija metoda kod koje se voda dovodi slobodnim tokom te se širi (ili stagnira) površinom i infiltrira u tlo.
- Navodnjavanje kišenjem – voda se raspršuje te iz zraka ravnomjerno pokriva čitavu površinu. Dobro podešen sustav može smanjiti utrošak vode. Takav sustav je

učinkovit i po pitanju količine opreme po jedinici površine, ali vlaženje listova biljaka povećava rizik razvoja bolesti.



Slika 5. Navodnjavanje kišenjem [8]

Slika 5. prikazuje finu maglicu koja ujednačeno prekriva prostor.

- Lokalizirano navodnjavanje – voda se dovodi samo na uži prostor na kojem se razvija glavna masa korijena te se ostvaruje ušteda utroška vode, omogućuje doziranje gnojiva i ostalih kemikalija te se smanjuje razvoj neželjenih organizama (korova i gljivica). Ograničenje predstavlja visok trošak opreme te njezina osjetljivost na nečistoće u vodi.



Slika 6. Lokalizirano navodnjavanje „kap na kap“ [9]

Slika 6. jasno pokazuje lokalnu promjenu boje tla uslijed većeg udjela vode, ali i crijeva koja moraju doseći svaku pojedinu biljku.

- Podzemno navodnjavanje – pomoću kapilarnosti se voda dovodi iz podzemnih kanala do područja korijena.

Kombinacija prednosti lokalnog i navodnjavanja kišenjem se može postići mikro prskalicama koje se također postavljaju u blizini svake biljke, ali pomoću njihovih mlazova je moguće još točnije dozirati vodu što je bitno kod kultura koje trebaju učestalo navodnjavanje manjim količinama. Slika 7. prikazuje primjer mikro prskalice.



Slika 7. Primjer mikro prskalice

2.2. Napredne metode upravljanja

S obzirom na to da se tema rada dotiče razvoja sustava inteligentnog upravljanja plastenikom, u ovom potpoglavlju će biti navedeni i opisani neki postojeći sustavi koji za cilj imaju unaprjeđenje proizvodnje pomoću modernih tehnologija.

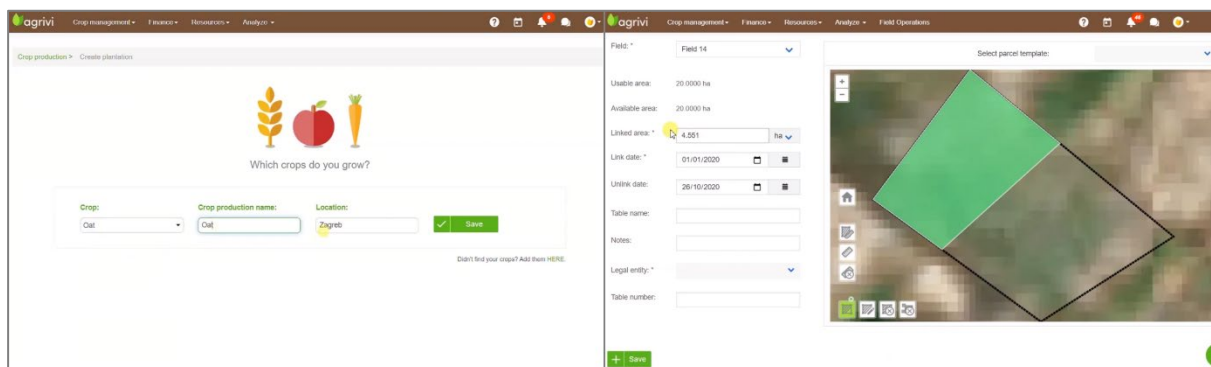
2.2.1. *Agrivi*

Tvrtka Agrivi je specijalizirana za razvoj ERP sustava za poljoprivredu. Neka od rješenja koja nude su:

- *Farm enterprise* - digitalna transformacija operacija u poljoprivredi

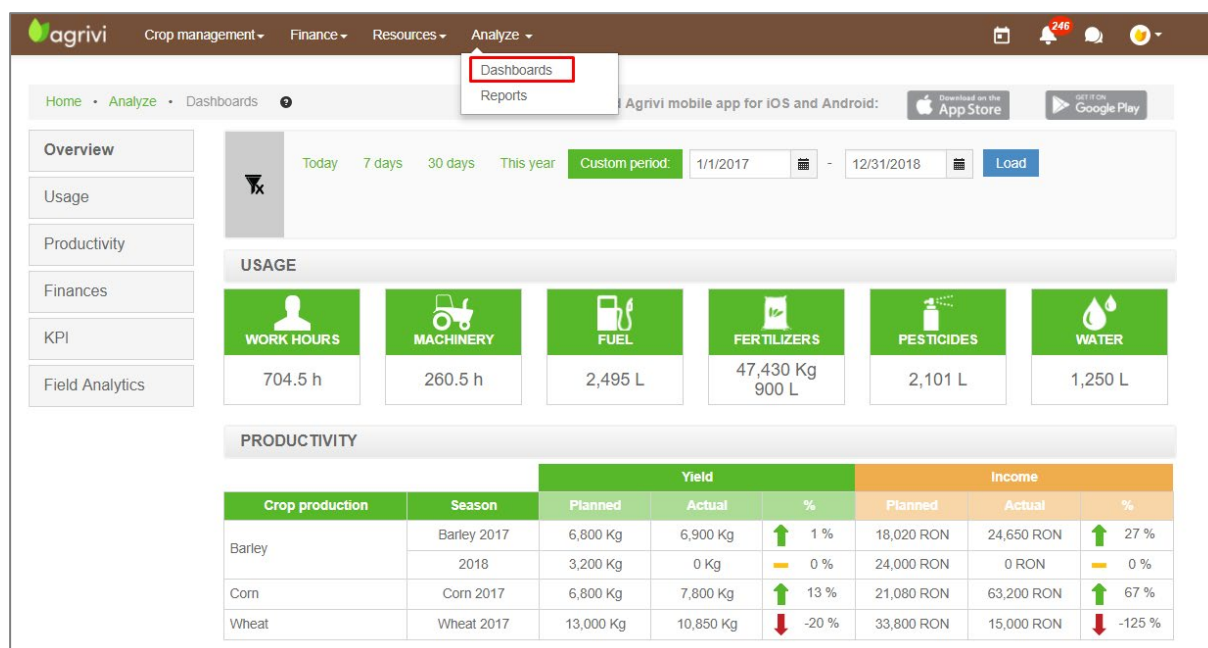
Ovaj sustav omogućuje prebacivanje svih aktivnosti vezanih uz poslovanje poljoprivrednog sustava u digitalni format. Moguće je u sustav unijeti fakture i radne naloge i na temelju njih

jednostavno voditi evidencije o prskanju, gnojidbi i ostalim važnim podacima te raditi analize poslovanja kako bi se optimizirali troškovi. Sustav omogućava učinkovito planiranje usjeva pomoću pregleda svih plodoreda tijekom vremena, uvid u polja u stvarnom vremenu pomoću satelitskih prikaza, vremenskih izvješća, alarma rizika od štetočina i praćenja napretka usjeva. Sustav također pojednostavljuje administraciju farmi jer su svi podaci spremljeni na jednom mjestu, dostupni putem interneta i lako se mogu koristiti za generiranje izvještaja [10].



Slika 8. Unos kulture i pripadajuće površine [11]

Na slici iznad je prikazano sučelje pomoću kojeg je na satelitskom prikazu polja moguće označiti područje kojeg zauzima određena kultura ili samo ručno unijeti iznos površine koja će se pokriti. Grafički unos je detaljniji te također može upozoriti na preklapanje kultura, tj. zauzetost površine. Osim same površine, unosi se i datum početka kulture na temelju kojeg se dobivaju preporuke o potrebnim aktivnostima (prskanje, gnojidba...). Datume aktivnosti je moguće unijeti i ručno, a zadani raspored se može spremati kao predložak za buduće korištenje. Unosom datuma berbe/žetve se zatvara krug i uspjeh uroda kulture se može usporediti s rezultatima na ostalim površinama te se mogu donijeti preporuke za rotaciju kultura u budućnosti. Vizualni pregled rezultata i parametara je dostupan u obliku nadzornih ploča (eng. *dashboard*), kao što prikazuje slika 9. [10].



Slika 9. Nadzorna ploča s ključnim pokazateljima [11]

Nadzorne ploče pružaju brz vizualni pregled uspješnosti poslovanja uz širok raspon ključnih pokazatelja uspješnosti. Sastoje se od nekoliko odjeljaka: *Overview* – kratki pregled najznačajnijih pokazatelja, *Usage* – pregled utrošenih radnih sati zaposlenika i strojeva te utroška goriva, vode i ostalih poljoprivrednih dodataka, *Productivity* - razlike u postotku između planiranog i stvarnog prinosa za cijelu zasađenu površinu, *Finances* - pregleda prodaje, troškova i kapitalnih troškova i *Field analytics* – analiza uspješnosti pojedinih polja [10].

- IoT integracija

Sa sustavom *Farm Enterprise* je moguće povezati različite IoT sustave za nadzor procesa u poljoprivrednim gospodarstvima. Npr., *IoT Fleet* je platforma za nadzor geolokacije poljoprivredne mehanizacije i pomoću nje je moguće dobiti podatke o lokaciji radnih strojeva, pređenoj udaljenosti, brzini te utrošenom radnom vremenu. Osim toga, pomoću platforme se može odrediti radijus kretanja strojeva uz alarme u slučaju da stroj napusti zadano područje. *IoT Meteo* je sustav za nadzor mikroklimatskih uvjeta na poljima u stvarnom vremenu i omogućava praćenje temperature i vlage zraka, brzinu i smjer vjetra, solarnu radijaciju te druge podatke o mikroklimi na polju. Omogućava i optimizaciju navodnjavanja tako da preciznije prati oborine i isparavanja tla te izrađuje učinkovite planove navodnjavanja. Precizni podaci o mikroklimatskim uvjetima na poljima omogućuju visoku točnost predviđanja pojave bolesti i štetnika te povećavaju točnost odluka o zaštiti bilja. Analizom povijesnih mikroklimatskih

podataka o polju omogućuje usporedbu meteoroloških uvjeta kroz prethodne sezone i analizu utjecaja klimatskih uvjeta na prinos [12].

- *Agriculture Supply Chain* – platforma za održivi lanac opskrbe usjevima

Agriculture Supply Chain je softversko rješenje za upravljanje poljoprivrednim lancem nabave s ciljem podizanja kvalitete proizvoda, sigurnosti hrane i sljedivosti, namijenjeno industriji hrane i pića koja radi direktan otkup poljoprivrednih proizvoda od proizvođača. Ideja platforme je da otkupljivači povežu dobavljače u jedinstvenu mrežu kojom je jednostavnije upravljati i donositi odluke koje mogu koristiti svim uključenim stranama. Neki od načina postizanja tog cilja su:

- Digitalizirani proces nabave
- Izravno pružanje agronomskih savjeta za unapređenje kvalitete
- Praćenje poljoprivrednih praksa na poljima.

Pomoću ove platforme, otkupljivač može pratiti odnos između planiranih i stvarnih prinosa po poljoprivrednom gospodarstvu, usjevu i polju u stvarnom vremenu. Na taj način može si osigurati kontinuiranu opskrbu i izbjeći probleme s isporukom proizvoda. Pomoću platforme korisnik može analizirati klimatski otisak svog lanca nabave, upotrebu zemljišta i vode, utjecaj na okoliš i poslovanje dobavljača. Također, može kontrolirati i osigurati postizanje korporativnih ciljeva održivosti [12].

2.2.2. *WaterBit*

Tvrtka WaterBit je razvila skalabilno rješenje niske kompleksnosti za kontrolu navodnjavanja poljoprivrednih površina. Sustav prikuplja podatke o stanju tla te na temelju njih automatski podešava parametre navodnjavanja. Osnovu sustava predstavljaju modularne jedinice za prikupljanje podataka (Slika 10.) koje preko središnjeg čvora šalju podatke na cloud server [13].



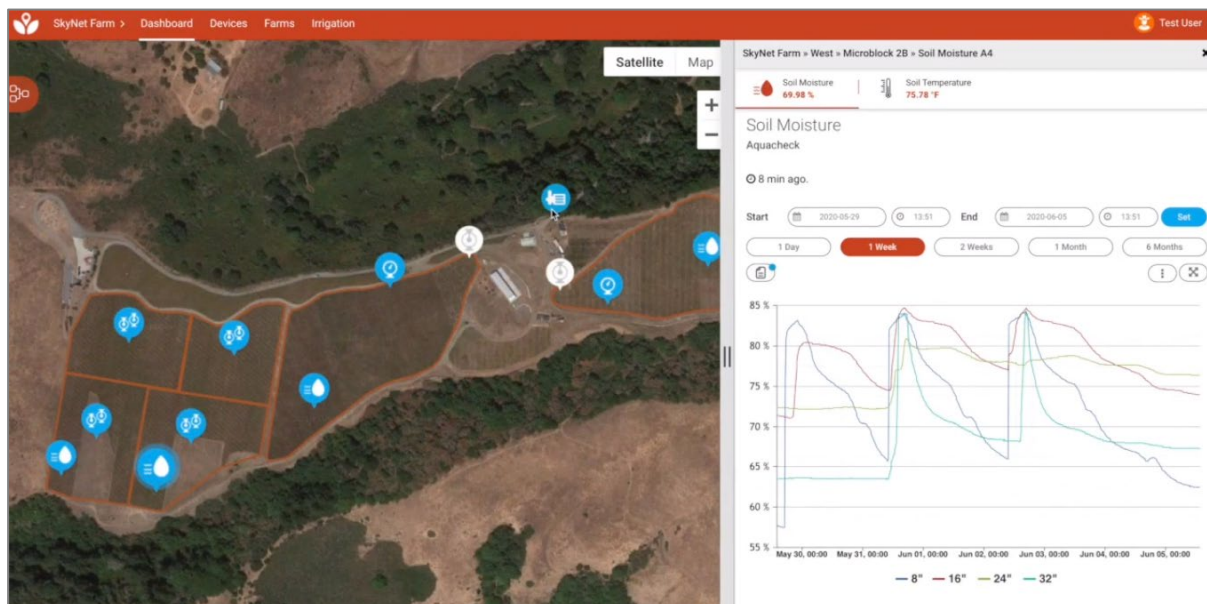
Slika 10. WaterBit Carbon™ - modularna jedinica za prikupljanje podataka [13]

Kao što se vidi na slici, jedinica za prikupljanje podataka je opremljena fotonaponskom ćelijom koja joj omogućava potpunu autonomiju, moguće ju je postaviti na lokaciju bez potrebe za provlačenjem žica za napajanje ili mijenjanjem baterija. Komunikacija s čvorom se odvija preko LoRa WAN (eng. *Long-Range Wide-Area Network*) protokola i jedan čvor je dovoljan za sve jedinice unutar 2km (uz izravnu liniju pogleda). Na jedinicu se spajaju senzori za vlažnost i temperaturu tla, a moguće je spojiti i senzore pritiska i protoka vode pa je, osim stanja tla, moguće mjeriti i potrošnju vode. Također je moguće spojiti i jedinice s ventilom za upravljanje navodnjavanjem te ih grupirati kako bi se različitim intenzitetom navodnjavalo različite dijelove površine pod kontrolom. Navodnjavanje se može paliti ručno, putem unaprijed definiranog rasporeda ili prema graničnim vrijednostima parametara, a više detalja se vidi na slici ispod [14].

SkyNet Farm > Dashboard Devices Farms Irrigation Test User >						
Scheduled Automated - Soil Moisture		CSV Scheduling			Download Irrigation History	
		Import Download Active Download Template			7 days Scheduled Applied	
Pump/Valve/Block	Duration	Start Irrigation	Repeat	Repeat Settings	Ending Date	Action
Microblock 1A	00:30	03/22/2020 11:30 PM	Every N Days	1 day	—	
Microblock 1A	00:30	03/23/2020 11:30 AM	Every N Days	1 day	—	
Pump Station	08:00	06/04/2020 08:30 AM	Days of Week	S M T W T F S	—	
East Block Valve	04:00	06/05/2020 06:00 AM	Days of Week	S M T W T F S	07/21/2020	
Microblock 1B	01:15	06/05/2020 09:15 AM	Days of Week	S M T W T F S	07/03/2020	
East Block Valve	01:30	06/10/2020 12:00 PM	none	---	—	
Delete All Add Row Save						

Slika 11. Sučelje za postavljanje rasporeda navodnjavanja [13]

Na slici 12. je prikazana struktura korisničkog sučelja. Na karti se označuju lokacije na kojima se određeni elementi nalaze te je moguće odabrati pojedinačni element ili sve elemente unutar definiranog područja. Podaci se mogu filtrirati prema vremenskom razdoblju (točno određeni raspon ili dan, tjedan, mjesec...). Na slici se vidi povijesni prikaz vlažnosti tla, a linije različite boje predstavljaju vrijednosti za senzore na različitim dubinama.



Slika 12. Prikaz korisničkog sučelja za pregled podataka [13]

2.2.3. *SupPlant*

Napredniji sustav predstavlja tehnologija tvrtke SupPlant koja nudi uštedu vode potrebne za navodnjavanje uz porast prinosa pomoću analize podataka s biljaka, tla te meteoroloških senzora. Prikupljeni podaci se uspoređuju s ranije zabilježenim podacima i od drugih korisnika iz baze (1500 jedinstvenih ciklusa rasta). Senzori su postavljeni na pet mjesta, tri na biljci (plod, list, deblo/stabljika) i dva u tlu (plitko i duboko) [15].



Slika 13. Senzor promjene promjera ploda (lijevo) i dendrometar (desno) [15]

Slika 13. prikazuje LVDT (eng. *Linear Variable Differential Transformer*) senzor, odnosno senzor koji registrira linearni pomak pomoću čega se bilježi rast ploda te dendrometar, senzor promjene promjera debla. Njihova kombinacija omogućava praćenje rasta ovisno o okolišnim uvjetima.



Slika 14. Senzor temperature lista (lijevo) i meteorološka stanica (desno) [15]

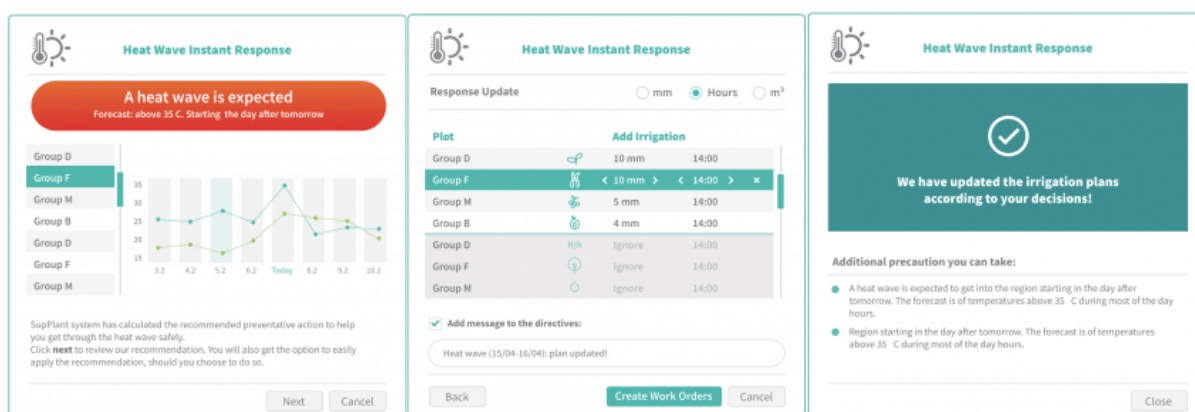
Na slici 14. se mogu vidjeti senzor temperature lista te meteorološka stanica za mjerenje temperature i vlažnosti zraka, razine Sunčevog zračenja, brzine vjeta i količine padalina. Upravo su ti parametri važni za određivanje količine vodene pare koja se ispušta iz tla i biljaka u atmosferu (ET, eng. *evaporperspiration*), i deficita vlažnosti zraka (VPD, eng. *vapor pressure*

deficit), razlike između trenutne i maksimalne zasićenosti zraka vodenom parom. Ta dva pokazatelja omogućuju modeliranje potrebe biljaka za vodom. Također se mjeri potrošnja vode te se u korisničkom sučelju dobiva pregled prošlih navodnjavanja te preporuka za idućih 8 dana. Sve navedene parametre je moguće odabrati za grafički prikaz. (Slika 15.)



Slika 15. Tjedni pregled izmjerenih parametara [15]

Uz preporuku za navodnjavanje, sustav automatski izdaje upozorenja kod slučajeva izlaska parametara van normalnih vrijednosti. Upozorenja su vezana uz razinu vode u tlu, naprezanja biljaka te vremenske uvjete. Sustav radi prognozu kretanja parametara u budućnosti te omogućava odabir automatske ili ručne prilagodbe navodnjavanja s obzirom na nove parametra što prikazuje slika 16. [15].



Slika 16. Upravljanje upozorenjima o ekstremnim uvjetima [15]

U ovom poglavlju su objašnjeni osnovni zahtjevi u poljoprivrednoj, posebice plasteničkoj proizvodnji, te opisani neki od sustava za poboljšavanje takve proizvodnje. U radu će biti prikazana oprema i metode uz koje će se jedan takav sustav pokušati izraditi.

3. NEIZRAZITA LOGIKA

S porastom složenosti, opisivanje sustava pomoću egzaktnih matematičkih funkcija sve je veći izazov. Bez obzira na to, ljudi imaju sposobnost razumijevanja složenih sustava, primjerice vožnje automobila, bez da znaju jasno izraziti njegov dinamički model. Teorija neizrazitih skupova pokušava ta implicitna znanja preslikati u skupove koji mogu biti interpretirani programskim jezicima [16].

3.1.1. Neizraziti skupovi i funkcije pripadnosti

Prema klasičnoj teoriji skupova, skup se definira kao kolekcija objekata koji dijele neko zajedničko svojstvo. Pripadnost objekta klasičnom skupu određena je egzaktno odnosno ili je objekt izvan skupa ili unutar (0 ili 1). Kada želimo pomoću funkcije izraziti pripadnost objekta skupu, onda funkcija pripadnosti objekta x skupu Z , $\mu_Z(x)$ ima samo dvije vrijednosti te je definirana s:

$$\mu_Z(x) = \begin{cases} 1, & \text{za } x \in Z \\ 0, & \text{za } x \notin Z \end{cases} \quad (1)$$

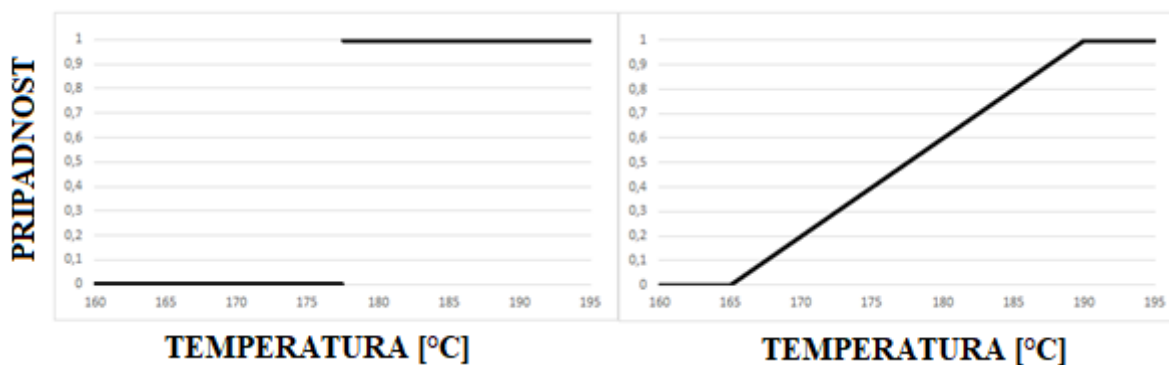
Prikazani način je pogodan za izražavanje pripadnosti brojeva skupu brojeva, npr. sasvim je jasno koji brojevi pripadaju skupu pozitivnih brojeva, ali kad se na isti način pokušaju izraziti opažanja u stvarnom svijetu, izraziti skupovi vrlo nepovoljni za granične slučajeve. Sljedeći primjer prikazuje definiranje skupova dovoljno i nedovoljno vruće pećnice:

$$\mu_{(vruća)}(x) = \begin{cases} 1, & \text{za } x \geq 180 \text{ } ^\circ\text{C} \\ 0, & \text{za } x < 180 \text{ } ^\circ\text{C} \end{cases} \quad (2)$$

Za pećnicu zagrijanu na $150 \text{ } ^\circ\text{C}$ i $250 \text{ } ^\circ\text{C}$ je sasvim jasno kojem skupu pripadaju, ali vrlo je teško opravdati svrstavanje pećnice zagrijane na $179 \text{ } ^\circ\text{C}$ u skup dovoljno vrućih pećnica ako je pećnica zagrijana na $180 \text{ } ^\circ\text{C}$ proglašena nedovoljno zagrijanom. Pomoću neizrazitog skupa, moglo bi se taj problem formulirati na sljedeći način:

$$\mu_{(vruća)}(x) = \begin{cases} 1, & \text{za } x \geq 190 \text{ } ^\circ\text{C} \\ \frac{x - 165}{25}, & \text{za } 165 \leq x < 190 \text{ } ^\circ\text{C} \\ 0, & \text{za } x < 165 \text{ } ^\circ\text{C} \end{cases} \quad (3)$$

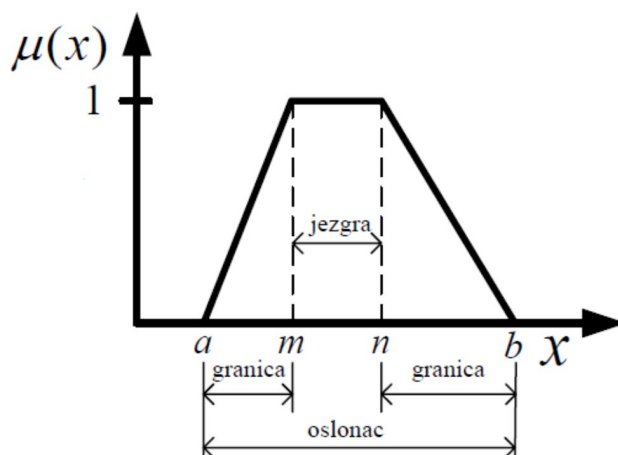
U ovom slučaju, pećnicama zagrijanim na 179 i 180 cm će biti pridružene pripadnosti od 0,56 i 0,6 što su vrijednosti koje znatno više odgovaraju stvarnim opisima.



Slika 17. Funkcije pripadnosti prikazanih skupova

Na slici se može vidjeti razliku između visina i pripadajućih iznosa pripadnosti prema dvama definiranim skupovima. Upravo ti grafovi predstavljaju funkcije pripadnosti, matematičke funkcije koje prikazuju istinitost tvrdnje.

U općenitom primjeru, većina funkcija pripadnosti se može izraziti kao specijalni slučaj trapezoidne funkcije pripadnosti, kakvu prikazuje slika 18. [16].

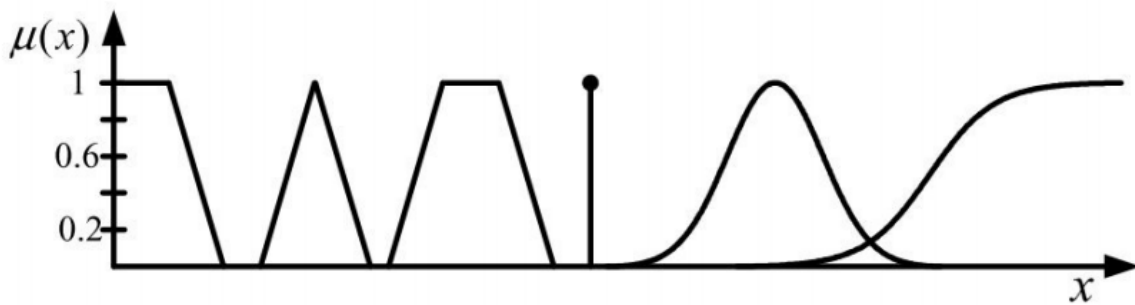


Slika 18. Funkcija pripadnosti s karakterističnim točkama [16]

Skup kojeg prikazuje funkcija sa slike glasi:

$$\mu_{(\text{visok})}(x) = \begin{cases} 0 & \text{ako } x < a; x > b \\ \frac{x-a}{m-a} & \text{ako } a \leq x \leq m \\ 1 & \text{ako } m \leq x \leq n \\ \frac{b-x}{b-n} & \text{ako } n \leq x \leq b \end{cases} \quad (4)$$

U slučaju kada je $m = n$ trapezoidna funkcija postaje trokutna funkcija pripadnosti, za $a = m$ i $b = n$ trapezoidna funkcija postaje pravokutna funkcija pripadnosti, a za $a = m = n = b$ dobiva se tzv. *singleton*. Različite funkcije pripadnosti su prikazane na slici 19. Visina neizrazitog skupa A je najveća vrijednost funkcije pripadnosti, oslonac neizrazitog skupa je izraziti skup koji sadrži sve elemente univerzalnog skupa U za koje je funkcija pripadnosti neizrazitog skupa A veća od nule, a jezgra neizrazitog skupa A je izraziti skup koji sadrži sve elemente univerzalnog skupa U za koje je funkcija pripadnosti neizrazitog skupa A jednaka jedinici. [16]



Slika 19. Različite funkcije pripadnosti [16]

3.1.2. Lingvističke varijable

Jezične varijable su varijable koje kao vrijednosti poprimaju riječi ili rečenice. Npr. 'visina' je jezična varijabla koja može poprimiti jezične vrijednosti 'nisko', 'visoko', 'vrlo visoko', itd. Jezična varijabla zapisuje se na sljedeći način:

$$[x, T(x), X, M], \quad (5)$$

gdje su:

- x – ime jezične varijable,

- $T(x)$ – prostor jezičnih vrijednosti koje varijabla x može poprimiti, tj. prostor neizrazitih skupova definiranih nad područjem X (npr. nizak, visok),
- X – kvantitativno područje razmatranja varijable x nad kojim ona poprima lingvističke vrijednosti (npr. 15 – 30 °C),
- M – semantička funkcija koja povezuje prostor $T(x)$ s područjem X (funkcije pripadnosti pojedinih jezičnih vrijednosti) [16].

3.1.3. Operacije s neizrazitim skupovima [16]

Kako bi se donijele odluke na temelju ulaznih varijabli i funkcija pripadnosti, provode se operacije sa skupovima. Neke od osnovnih operacija su:

- Jednakost - neizraziti skupovi A i B su jednaki ako za svaki $x \in U$ vrijedi:

$$\mu_A(x) = \mu_B(x) \quad (6)$$

- Sadržaj (inkluzija) – neizraziti skup B sadrži skup A ($A \subseteq B$) ako za svaki $x \in U$ vrijedi:

$$\mu_A(x) \leq \mu_B(x) \quad (7)$$

- Presjek – presjek neizrazitih skupova ($A \cap B$)

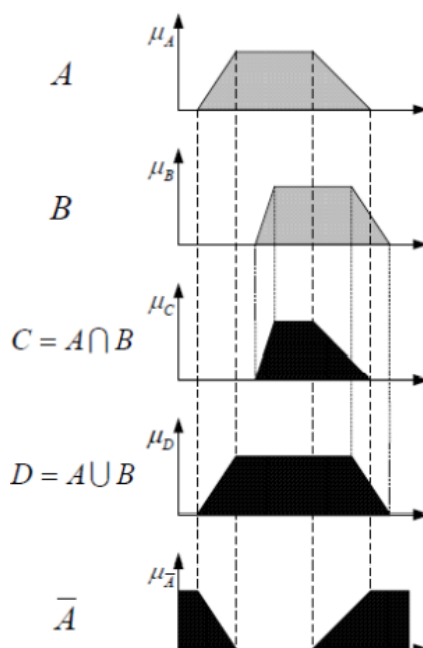
$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \quad x \in U \quad (8)$$

- Unija – unija neizrazitih skupova ($A \cup B$)

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \quad x \in U \quad (9)$$

- Komplement – komplement neizrazitog skupa ($\bar{A}$) svaki $x \in U$ vrijedi:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (10)$$



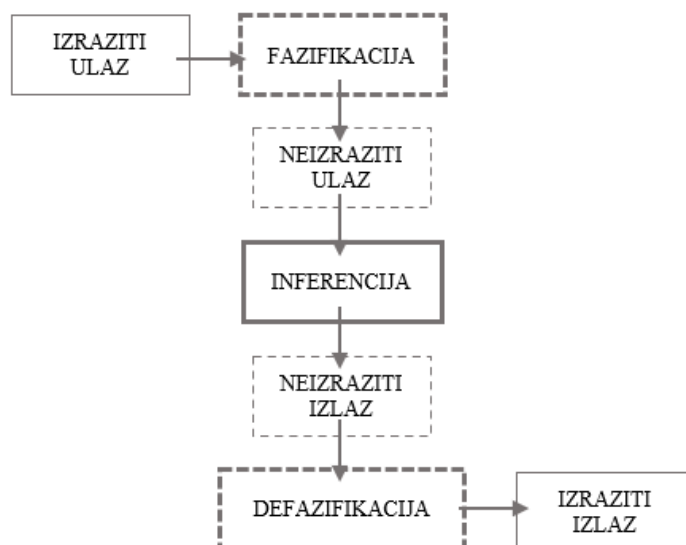
Slika 20. Operacije s neizrazitim skupovima [16]

3.1.4. Neizraziti sustavi

Neizraziti skupovi, kojima se opisuju neprecizni pojmovi, i neizrazita logika, koja se fokusira na lingvističke varijable s ciljem omogućavanja aproksimativnog rezoniranja s nepreciznim propozicijama, čine osnovu neizrazitih sustava. Osnovu svih neizrazitih sustava čine sljedeća tri postupka:

1. Omekšavanje ili fazifikacija (eng. *fuzzyfication*) - proces transformiranja ulaznih izrazitih vrijednosti u stupnjeve pripadnosti ulaznih neizrazitih skupova. Središnji dio tog postupka je redom određivanje ulaznih i izlaznih (jezičnih) varijabli, određivanje područja definicije navedenih varijabli, te određivanje broja i oblika funkcija pripadnosti koje prekrivaju područje definicije pojedinih jezičnih varijabli.
2. Inferencija ili proces odlučivanja - proces transformiranja ulaznih neizrazitih skupova u neizrazite izlazne skupove. Središnji dio tog postupka je određivanje baze ponašanja u obliku IF – THEN pravila. Baza pravila ponašanja predstavlja formalizirani oblik znanja eksperta ili operatera.
3. Izoštavanje ili defazifikacija (eng. *defuzzyfication*) - postupak transformiranja neizrazitih izlaznih skupova u izrazite izlazne vrijednosti. Taj postupak je nužan kod neizrazitih regulatora s obzirom da izvršni član (aktuator) može raditi samo s izrazitim vrijednostima [16].

Na slici je prikazan tijek procesa s sva tri postupka u neizrazitom sustavu.



Slika 21. Tijek procesa u neizrazitom sustavu

Ovo poglavlje služi kao teorijska podloga za razvoj naprednog sustava upravljanja, a detalji same izrade će biti objašnjeni kasnije.

4. PREGLED KORIŠTENE OPREME

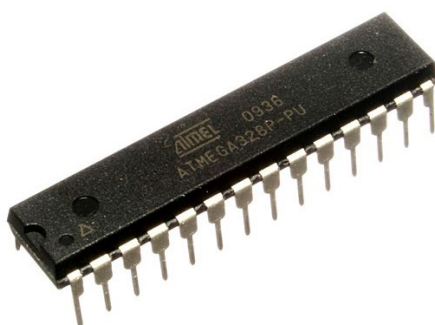
U ovom poglavlju će ukratko biti opisani programski jezici i paketi korišteni pri razvoju softverske strane te elektroničke komponente upravljačkog sustava.

4.1. Računalna sklopovska oprema, hardware

Sustav prikupljanja podatak i upravljanja je izrađen na Arduino platformi. Prednost Arduina je to što su sve komponente otvorenog tipa pa mogućnosti razvoja kompatibilnih komponenti nije ograničena patentima. Iz tog razloga je moguće pronaći brojne komponente po izuzetno povoljnim cijenama.

4.1.1. Mikrokontroleri

Mikrokontroler je vrsta mikročipa koji objedinjuje procesorsku jedinicu, memoriju i I/O (ulaz/izlaz) sučeljem bez potrebe za spajanjem vanjskih krugova. Primarna ideja mikrokontrolera je samodostatnost i niska cijena i zbog toga su često korišteni kao jezgra za izvršavanje logike u integriranim (eng. *embedded*) sustavima [17]. Pod pojmom „mikrokontroler“ će u radu biti obuhvaćeni kompletni mikrokontrolerski moduli opisani u nastavku.

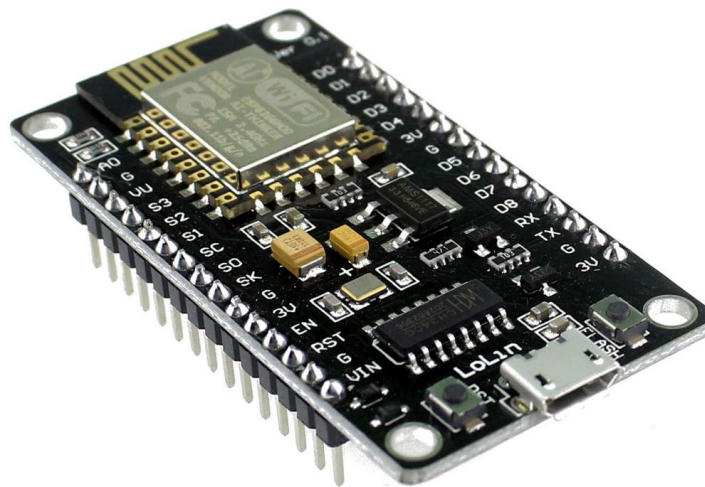


Slika 22. Atmel ATmega328 mikrokontroler [18]

Jedan od najčešće korištenih mikrokontrolera otvorenog tipa, ATmega328, je prikazan na slici 22.

- NodeMCU DEVKIT 1.0 ESP-12E (ESP8266)

ESP-12E je razvojna pločica na bazi ESP8266 SoC-a (eng. *System on a Chip*) tvrtke Espressif Systems s mikrokontrolerskim mogućnostima i Wi-Fi odašiljačem/prijamnikom. NodeMCU je naziv firmwarea prvotno nastalog za razvoj ESP8266 pločica koji je proširen i na ESP32.



Slika 23. NodeMCU DEVKIT 1.0 ESP-12E [19]

Modul se na računalo spaja preko micro-USB sučelja te se na njega prenosi kod. Modul podržava i niz internet protokola (TCP/IP) pa je moguće spajanje na lokalne bežične mreže. Mikroprocesor je 32-bitni Xtensa LX106 sa RISC arhitekturom, radnim taktom od 80 MHz, 32 KiB radne memorije, a modul sadrži vanjskih 4 MB flash memorije. Radni napon je 3.3V, a prosječna potrošnja oko 80mA. Tablica 1. sadrži popis pinova modula [20].

Tablica 1. Pinovi ESP-12E modula [20]

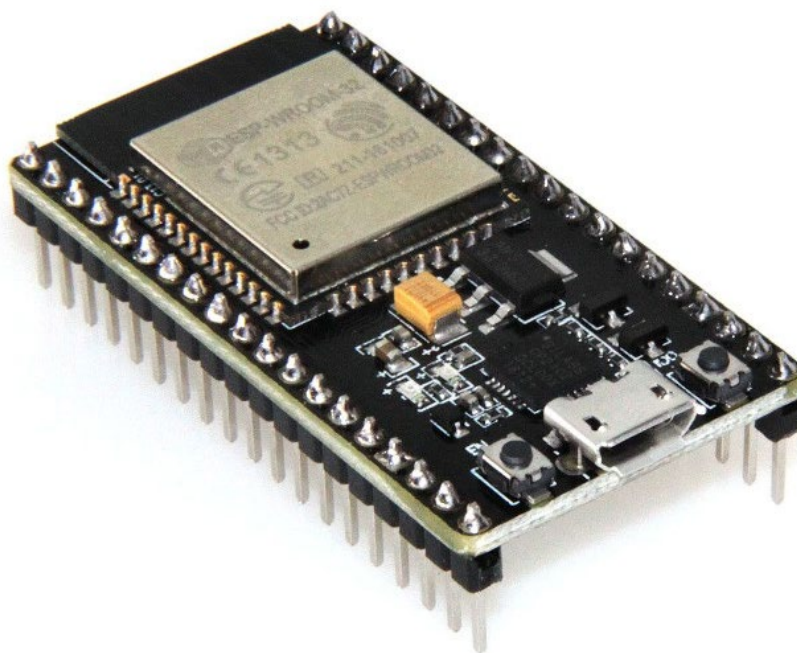
Vrsta	Naziv	Opis
Napajanje	Micro-USB, 3.3V, GND, Vin	
Upravljanje	EN, RST	Pinovi i gumb za resetiranje mikrokontrolera
Analogni	A0	Analogno mjerenje napona u rasponu 0-3.3V
GPIO	GPIO1 - GPIO16	16 pinova za općenitu upotrebu

SPI	SD1, CMD, SD0, CLK	4 pina za SPI serijsku komunikaciju
UART	TX, RX	UART sučelje se može koristiti za bežični prijenos programa na mikrokontroler

- NodeMCU ESP32 (ESP32-D0WDQ6)

ESP32 je nasljednik ESP8266, a neka od poboljšanja koja donosi su:

- viši radni takt procesora
- podrška dvojezgrenog procesuiranja
- više GPIO pinova
- Bluetooth 4.2, Bluetooth LE
- ugrađeni senzor dodira i senzor Hallovog efekta
- sekundarni procesor za rad u režimu niske potrošnje



Slika 24. NodeMCU ESP32 [21]

Korišteni mikroprocesor je Tensilica Xtensa 32-bit LX6 s radnim taktom od 160 MHz (do 240), ugrađenih 448 KB radne memorije, 512 KiB SRAM memorije, a modul sadrži 4 MB ugrađene flash memorije. Raspored pinova je sličan kao i kod ESP-12E, samo što ESP32 zbog dva 12-

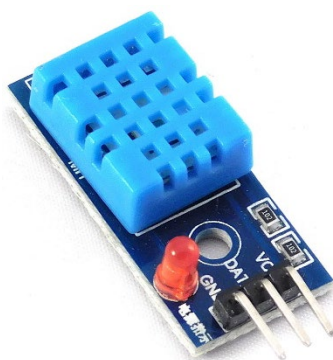
bitna A/D konvertera naspram jednog 10-bitnog kod ESP-12E raspolaže s dodatnih 14 analognih pinova [22].

4.1.2. Senzori

U elektronici, senzori su uređaju koji vanjski podražaj iz okoline bilježe i pretvaraju u električni signal, uglavnom kao promjenu napona. Danas je zbog želje za detaljnim razumijevanjem svakog procesa teško naći sustav koji u sebi ne sadrži neki oblik senzora [23]. Brojne vrste senzora se može podijeliti i prema [24]:

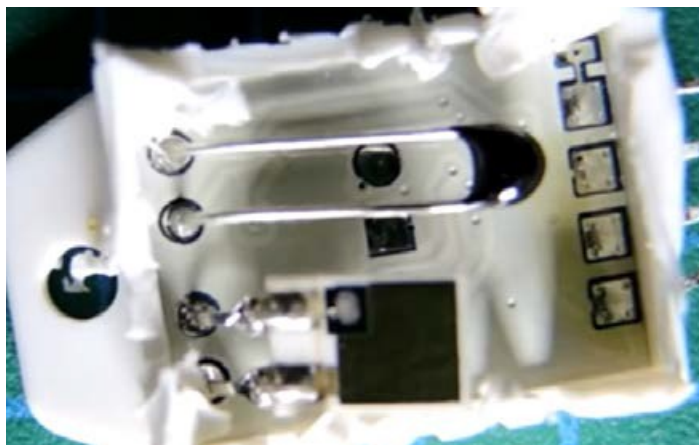
- Mjernoj veličini – akustični, kemijski, električni, magnetski, mehanički, toplinski...
 - Tehničkim karakteristikama – osjetljivost, točnost, preciznost, mjerni raspon, rezolucija...
 - Načinu detekcije – biološki, kemijski, magnetski, električni, mehanički, toplinski...
 - Vrsti pretvaranja – biološki, kemijski i fizikalni
 - Materijalu senzora – organski, neorganski, vodiči, izolatori...
 - Području primjene – kućanstvo, proizvodnja, znanstvena istraživanja...
-
- DHT11 senzor temperature i vlažnosti zraka

Za mjerenje temperature i vlažnosti zraka u plasteniku, korišten je jedan digitalni mjerni modul, DHT11. Modul sadrži dvije pločice, veću, s kondenzatorom za filtriranje i tri pina te manju na koju su spojeni mjerni elementi.



Slika 25. DHT11 [25]

Dva su mjerna elementa, tj. senzora. Jedan je termistor, kojem se otpornost smanjuje s porastom temperature, a drugi kapacitivni senzor vlažnosti koji je prevučen higroskopskim slojem te se zbog više ili manje vodene pare u zraku mijenja njegov kapacitet. Izgled i raspored mjernih elementa prikazuje slika 26. [26].



Slika 26. Mjerni elementi u modulu, termistor (gore) i kondenzator (dolje)

Tablica 2. prikazuje radne karakteristike i raspon iskoristivosti DHT11 modula.

Tablica 2. Specifikacije DHT11 modula [27]

Radni napon	3,5 – 5V
Radna struja	0,3 mA čitanje, 0.06 mA stanje pripravnosti
Raspon temperature	0 – 50 °C
Raspon vlažnosti	20 – 90 %
Preciznost	±2 °C, ±5 %
Frekvencija uzorkovanja	1 Hz

Postoji i preciznija verzija DHT11 senzora, DHT22.

- SEN0193 senzor vlažnosti tla

SEN0193 je kapacitivni senzor što znači da udio vlage u mediju u kojem se nalazi određuje preko promjene unutarnjeg kapaciteta. Radni napon je 3.3-5V, ali s nižim naponom su

primijećene poteškoće u radu. To je prednost u odnosu na neke druge vrste senzora koje za određivanje koriste električni otpor između dvaju polova kroz koje teče struja jer na taj način dolazi do korozije, i to vrlo brzo. Na **Error! Reference source not found.** je zelenom bojom označena optimalna dubina na koju bi senzor trebao biti zataknut, a crvenom maksimalna dubina.



Slika 27. SEN0193 kapacitivni senzor

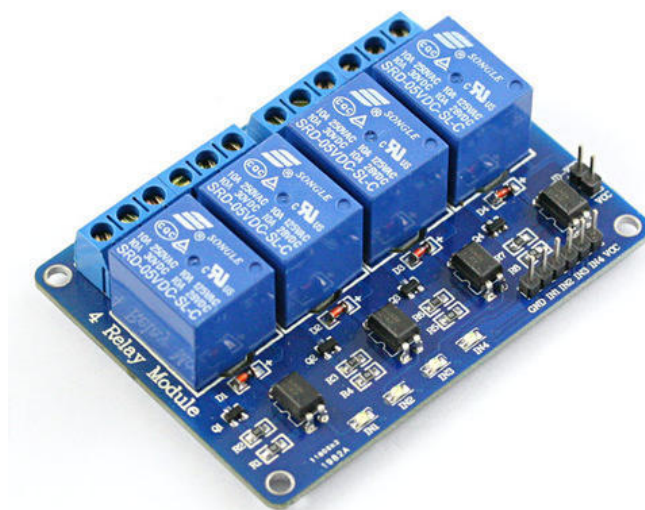
Senzor je analogni što znači da ga je potrebno umjeriti da bi se izmjerene vrijednosti mogle iskazati kao stvarna vlažnost tla u postotku. Izmjerena vrijednost ovisi o više faktora, temperaturi i vrsti tla, kemijskom sastavu, rahlosti itd., a ne samo udjelu vode u tlu tako da senzor uvelike smanjuje robusnost modela.

4.1.3. Aktuatori

Dok senzori služe za pribavljanje informacija o stanju okoline, svrha aktuatora je izvođenje zadataka si ciljem utjecaja na okolinu, a na temelju signala dobivenog sa signala. Važno svojstvo aktuatora je mogućnost kontrole pomoću signala niske snage [28].

- 5V relej

Relej je elektronički uređaj koji služi za kontrolu uređaja veće snage pomoću signala male snage. Najčešće se sastoji od elektromagneta koji, ovisno o pobudi, otvara ili zatvara strujni krug koji prolazi kroz njega.



Slika 28. 4-kanalni 5V relej [29]

Slika 28. prikazuje modul s 4 kanala, ali u projektu su korišteni i moduli sa samo jednim kanalom. Izvedba s 4 kanala je povoljnija zbog jedinstvenog napajanja. Nominalni napon za aktivaciju je 5V, ali i 3.3V logika je dovoljna ako je modul spojen na vanjski izvor 5V. Maksimalna struja koju može kontrolirati je 10A/30V DC i 10A/250V AC [30].

4.2. Software

Za izradu rada je korišteno nekoliko programa koji će u nastavku biti nabrojani i kratko opisani.

- Microsoft Visual Studio

Visual Studio je Microsoftov IDE (eng. *Integrated development environment*) koji podržava rad u svim istaknutim programskim jezicima, a u radu je korišten najviše zbog izrade web formi i integracije s bazom podataka. Visual Studio Community Edition verzija programa je besplatna za korištenje.

- Arduino

Kao što je ranije navedeno, čitav rad je izrađen na Arduino platformi. Pod pojmom Arduino se obuhvaćaju i hardverske komponente i programski jezik za njihovo programiranje. Sintaksa programskog jezika Arduino je zapravo istovjetna C/C++, jedina razlika je u bibliotekama zbog ograničenja resursa u Arduino primjenama, tj. raspolaže se s puno manje memorije. U ovom projektu se programiranje izvodilo u Arduino IDE programu. Programski kod koji se prenosi

na mikroprocesor se naziva sketch, a minimalni sadržaj sketcha su samo dvije funkcije, *setup()* za inicijalizaciju varijabli i biblioteke te *loop()* koja se ponavlja dok god pločica radi.

- Matlab

Matlab je programski paket korišten primarno za numeričku analizu, ali razni dodatni alati (eng. *toolbox*) omogućuju široku primjenu u raznim područjima, poput regulacije i umjetne inteligencije. *Toolbox* korišten u ovom radu je *Fuzzy Logic Toolbox*, a korištenje je detaljnije opisano u potpoglavlju 5.8. *Fuzzy Logic Toolbox*, kao i većina ostalih, nije besplatan, a korištena je probna verzija. Matlab također označava i programski jezik, ali ovdje se misli samo na programsko sučelje, jer jezik u radu nije niti korišten, već samo vizualno sučelje korištenog *toolboxa*.

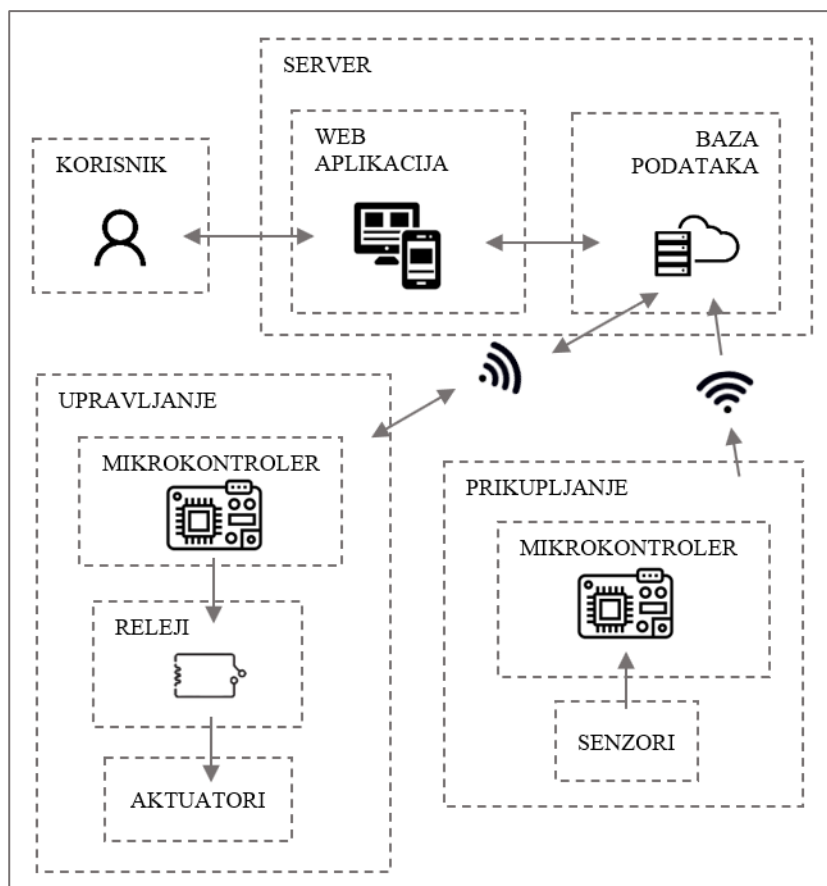
U ovom poglavlju su prikazani alati korišteni za izradu sustava, neke od ključnih specifikacija te kratki opisi njihovog načina rada. Sljedeće je na redu opisivanje razvoja web aplikacije, slaganje komponenti u cjelinu te programiranje sustava za upravljanje.

5. SUSTAV ZA UPRAVLJANJE I PRAĆENJE STANJA U PLASTENIKU

Kroz ranije kolegije na studiju je razvijen sustav za prikupljanje podataka sa senzora u bazu podataka te njihov strukturiran prikaz na web stranici (<http://projekt-host.xyz>). U ovom poglavlju će biti opisani njegovi glavni elementi i tehnologije korištene za izradu, dodane funkcionalnosti te razvoj fizičke strane sustava, elektroničkih komponenti.

5.1. Definiranje strukture sustava

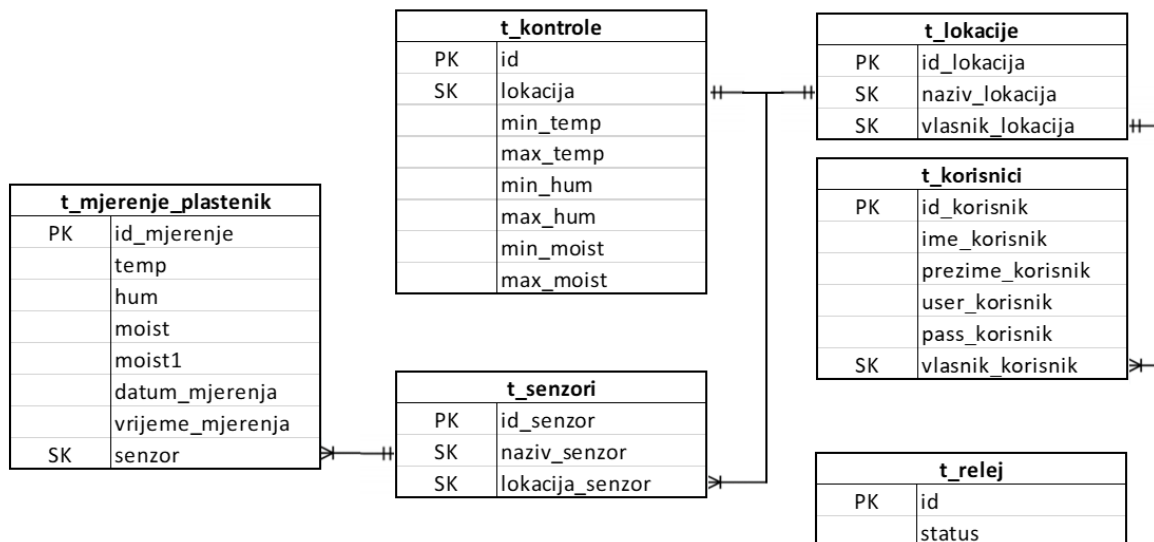
Funkcionalnosti razvijenog sustava su se mogle ostvariti na brojne načine i pri definiranju strukture su se nudile razne mogućnosti. Korištena platforma NodeMCU je odabrana primarno zbog niske cijene, ali i zato što je u radu korištena baza podataka u oblaku i samo je potrebno ostvariti komunikaciju s njom, ništa se ne izvodi lokalno. Da je postojala potreba za isključivo lokalnim spremanjem podataka, puno bolji izbor platforme bi bio *Raspberry Pi*, računalo sa svim mogućnostima osobnog računala ili laptopa, ali u formatu malo većem od mikrokontrolera. Kontrola parametara bez pristupa internetu, u zatvorenom krugu, se mogla ostvariti i samo pomoću NodeMCU-a, uz komunikaciju različitih mikrokontrolera u lokalnoj mreži. Ipak, s obzirom da je pokušaj bio razviti sustav s navedenim funkcionalnostima, ali uz minimalne troškove, odabrano je rješenje u kojem je moguće s neograničenim brojem mikrokontrolera neovisno pristupati bazi i spremati i čitati podatke.



Slika 29. Shematski prikaz međusobnih odnosa entiteta

5.2. Prikaz strukture baze podataka

Za potrebe projekta je korištena MySQL baza podataka jer je za nju puno lakše povezivanje s Arduinoom. myasp.net nudi alat phpMyAdmin za administriranje baze. phpMyAdmin je grafičko sučelje koje preko PHP naredbi izvodi operacije nad bazom te omogućava promjenu strukture baze i uređivanje podataka iz web preglednika što je vrlo značajna pogodnost za sustav kakvog se u projektu razvija.



Slika 30. Struktura baze

Vrlo važno za ispravno funkcioniranje aplikacije u budućnosti je od samog početka voditi računa o mogućnosti proširenju, tj. skalabilnosti baze i čitavog sustava. Za očekivati je da neće zauvijek biti korišteni identični senzori, niti po vrsti, niti po količini. Zato je potrebno tablice u bazi strukturirati tako da je za dodavanje ili oduzimanje komponenti potrebno obavljati što manje promjene.

Npr., za dodavanje nove lokacije je potrebno spojiti potrebne senzore s novim mikrokontrolerom, na njega prenijeti kod i jednostavno dodati novi red s novom lokacijom u tablicu `t_lokacije` i novi red s novim senzorom u `t_senzori` i sva nova mjerenja s novih senzora će se moći prikazivati na način kao i ona s prijašnjih, a za njihovo prikaz je potrebno samo odabrati novu lokaciju iz padajućeg izbornika koji će biti prikazan kasnije.

id_lokacija	naziv_lokacija	vlasnik_lokacija
1	Plastenik	vlasnik1
2	Soba	vlasnik2

Slika 31. Tablica `t_lokacije`

id_senzor	naziv_senzor	lokacija_senzor
1	senzor1	plastenik
2	senzor2	soba

Slika 32. Tablica t_senzori

	id_mjerenje	temp	hum	moist	datum_mjerenja	vrijeme_mjerenja	senzor
☐ Edit Copy Delete	1	21.96	34	51	2020-07-15	00:00:00	senzor1
☐ Edit Copy Delete	2	27.72	36	98	2020-07-15	00:00:00	senzor2
☐ Edit Copy Delete	3	30.23	80	37	2020-07-15	01:00:00	senzor1
☐ Edit Copy Delete	4	20.87	34	13	2020-07-15	01:00:00	senzor2
☐ Edit Copy Delete	5	19.03	60	79	2020-07-15	02:00:00	senzor1
☐ Edit Copy Delete	6	26.50	45	76	2020-07-15	02:00:00	senzor2
☐ Edit Copy Delete	7	23.49	86	2	2020-07-15	03:00:00	senzor1
☐ Edit Copy Delete	8	30.30	49	93	2020-07-15	03:00:00	senzor2
☐ Edit Copy Delete	9	22.31	30	75	2020-07-15	04:00:00	senzor1

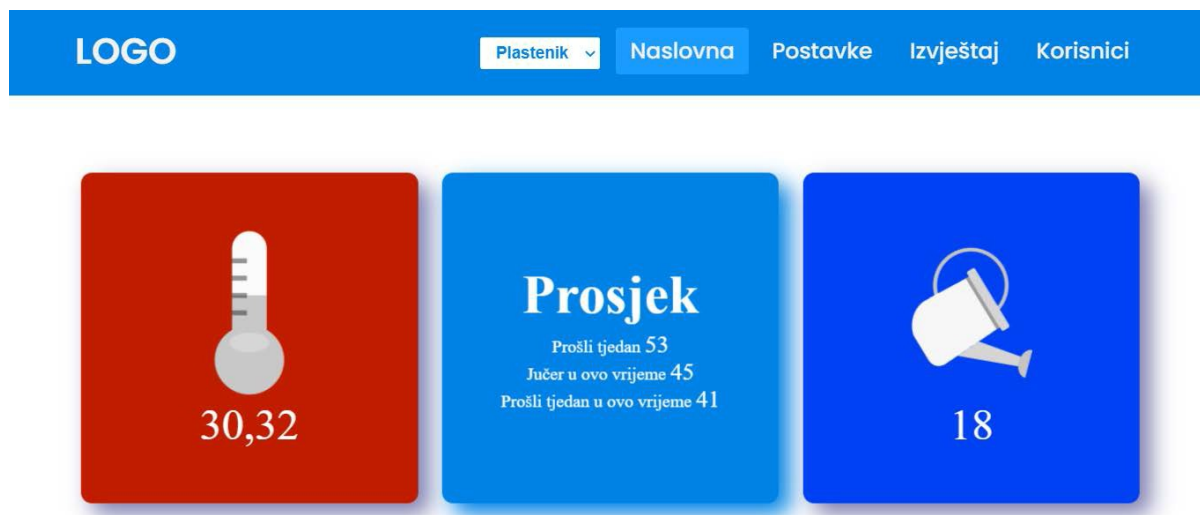
Slika 33. Tablica s izmjerenim vrijednostima

Može se vidjeti kako tablica t_senzori sadrži stupac s nazivom lokacije na kojoj je određen sensor postavljen, a upravo preko te kolone uskladištena procedura odabire ispravne podatke za prikaz. Logika za prikaz podataka za određenu lokaciju je sljedeća:

- padajući izbornik se popunjava vrijednostima iz tablice t_lokacije
- odabrana vrijednost iz padajućeg izbornika se prosljeđuje uskladištenoj proceduri kao ulazni parametar tname
- procedura preko parametra odabire senzor iz tablice t_senzori
- procedura preko odabranog senzora odabire sve podatke za prikaz

5.3. Prikaz i opis upotrebe web stranice za nadzor

Ideja web stranice je da se korisniku odmah po otvaranju da jasna informacija o stanju tri mjerena parametra (temperatura, vlažnost zraka i vlažnost tla), a to je postignuto dinamičkom promjenom boja elemenata stranice ovisno o odnosu definiranih graničnih vrijednosti i izmjerenih parametara.



Slika 34. Naslovna strana

Elementi poprimaju crvenu boju u slučaju da je vrijednost izmjerenog parametra iznad gornje definirane granice, plavu boju ako je izmjerena vrijednost ispod donje granice, a zelenu ako vrijednost pada unutar granica. Na taj način se korisnik odmah upozorava na potrebu za poduzimanjem određene akcije, npr. prozračivanje ili zalijevanje. S druge strane svakog elementa se nalaze prosječne vrijednosti za određena razdoblja koje daju kontekst vrijednostima s prednje strane, tj. omogućuju dodatan stupanj prosudbe. Pomoću padajućeg izbornika je moguće promijeniti lokaciju, tj. povući podatke s drugog seta senzora što pojednostavljuje proširenje aplikacije.

Sljedeća mogućnost kojom korisnik raspolaže je promjena graničnih vrijednosti po potrebi, npr. s obzirom na promjenu vremenskih uvjeta, promjenu kultura unutar plastenika ili zbog stečenih novih saznanja u vezi uzgoja.

LOGO

Plastenik ▾

Naslovna

Postavke

Izveštaj

Korisnici

TEMPERATURA

Min: 16
Max: 23

Spremi

Spremi sve

Minimum:

Maksimum:

VLAŽNOST ZRAKA

Min: 25
Max: 50

Minimum:

Maksimum:

VLAŽNOST TLA

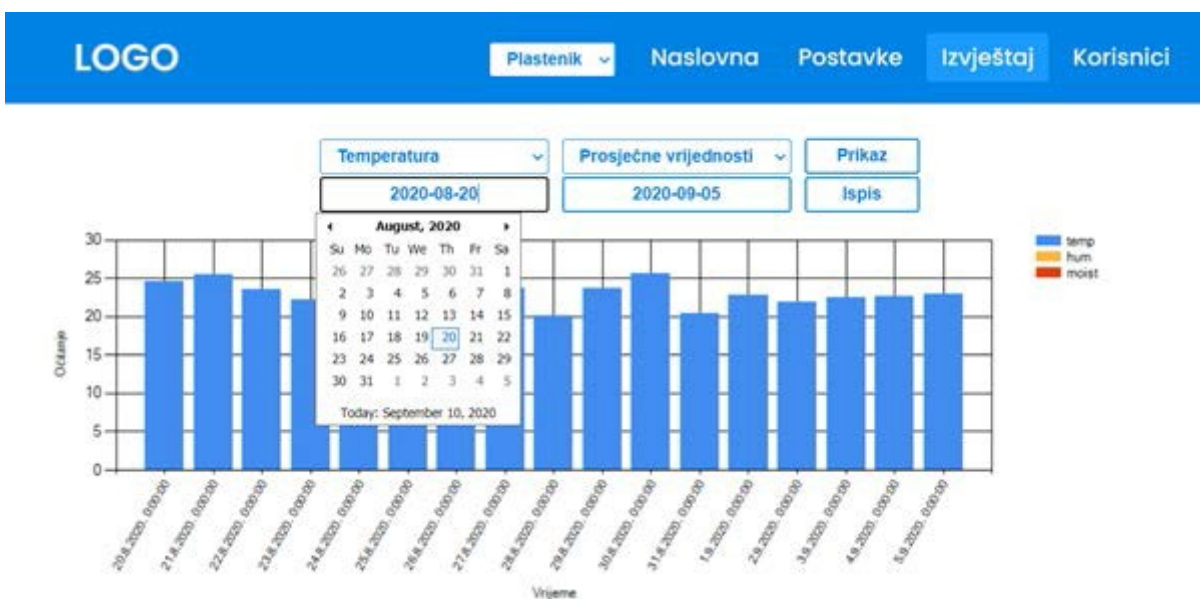
Min: 38
Max: 60

Minimum:

Maksimum:

Slika 35. Forma za promjenu granica

Treća mogućnost je filtriranje podataka i njihov grafički prikaz, a to je ujedno i najvažnija funkcija za pomoć pri donošenju odluka jer omogućuje prepoznavanje uzoraka i izvlačenje znanja iz naizgled zamršenog skupa podataka.



Slika 36. Filtriranje i prikaz podataka (1)



Slika 37. Filtriranje i prikaz podataka (2)

Pomoću padajućih izbornika i tekstualnih polja je moguće unijeti parametre za uskladištene procedure i pozadinski kod čime se može u polju za prikaz promijeniti promatrani vremenski interval, mjereni parametar koji se prikazuje te vrstu grafa.

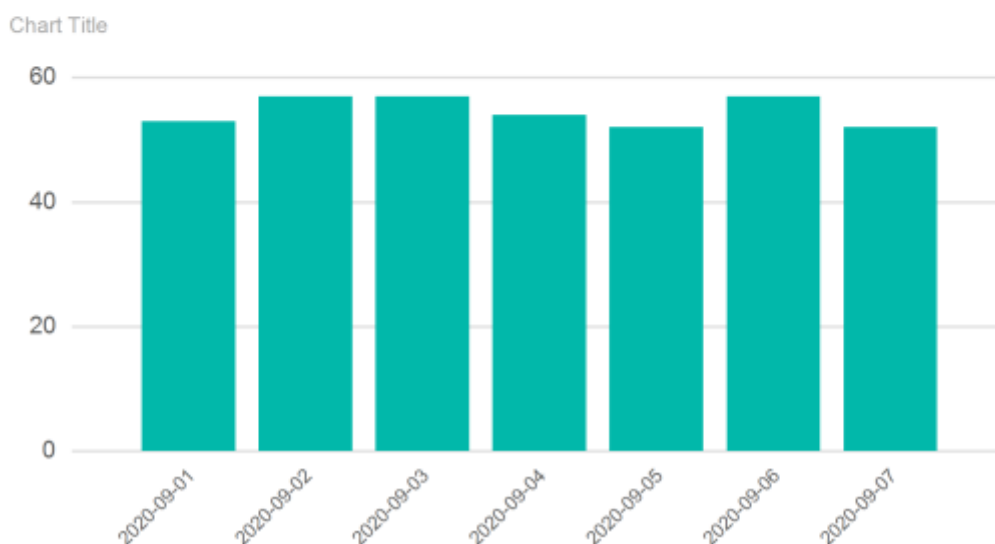
Uz prikaz podataka na samoj web stranici je moguće i prikazati podatke u PDF formatu pomoću RDLC priključka za Visual Studio. Takav način prikaza podataka je vrlo koristan kad postoji potreba za ispisom podataka na pisaču, zbog vođenja dokumentacije ili nekog drugog razloga.

IZVJEŠTAJ - Plastenik

TEMPERATURA VLAŽNOST ZRAKA VLAŽNOST TLA

2020-09-01 - 2020-09-07

TEMPERATURA	VLAŽNOST ZRAKA	VLAŽNOST TLA	DATUM MJERENJA
22.72	53	51	2020-09-01
21.89	57	52	2020-09-02
22.43	57	55	2020-09-03
22.7	54	53	2020-09-04
22.94	52	51	2020-09-05
23.53	57	42	2020-09-06
22.77	52	42	2020-09-07

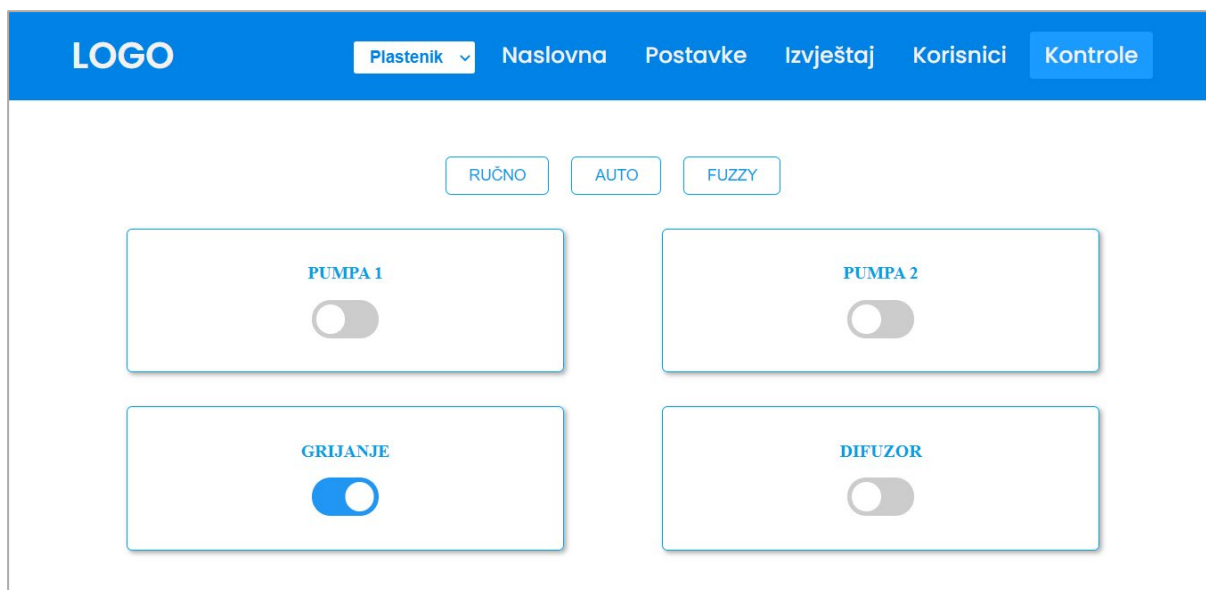


Slika 38. Izvještaj u PDF formatu

PDF izvještaj se također generira dinamički tj. parametri se prenose iz polja za filtriranje. U zaglavlje izvještaja se prenosi naziv lokacije za koju se prikazuju podaci, ističe se varijabla koju se prikazuje te se ispisuju granični datumi, a u glavnom dijelu se s obzirom na odabran način prikaza (vrsta grafa) mijenja atribut *visibility* te tako postiže različita vrsta grafa.

5.4. Sučelje za upravljanje aktuatorima na daljinu

Kao nova funkcionalnost web aplikacije u ovom radu u odnosu na sustav razvijen na kolegiju *Informacijski sustavi* je stranica za ručni način upravljanja aktuatorima, čiji će način rada biti objašnjen u nastavku.



Slika 39. Sučelje za upravljanje aktuatorima

Na slici se mogu vidjeti dostupne kontrole. Tri gumba na vrhu služe za odabir načina upravljanja, a ostali prekidači za kontrolu pojedinih aktuatora. Pritiskom na kontrolu, šalje se JSON GET zahtjev koji sadrži URL PHP skripte za promjenu vrijednosti u bazi. Tu vrijednost mikrokontroler čita i koristi za definiranje stanja releja.

```
document.getElementById('auto').addEventListener('click', function () {  
    var url = "https://projekt-host.xyz/api/update.php?id=6&status=auto";  
    $.getJSON(url, function (data) {  
        console.log(data);  
    });  
});
```

Slika 40. Kod za slanje GET zahtjeva pomoću prekidača

Prikazan je JavaScript kod za slanje GET zahtjeva. JavaScript je korišten kako bi se, korištenjem AJAX metoda (u ovom slučaju *getJSON*), naredbe mogle izvršavati bez potrebe za osvježavanjem stranice (kao što je slučaj za ASP.NET). Za svaki prekidač je korišten isti kod, jedina je razlika u atributu u funkciji *getElementById* jer se tu definira na koji prekidač se funkcija odnosi, te u parametrima u URL-u PHP skripte.

5.5. Sklop za prikupljanje podataka sa senzora

Kao što je navedeno u ranijim poglavljima, način rada sustava je dizajniran tako da se podaci sa senzora prikupljaju u bazu podataka neovisno o ostaloj opremi kojom korisnik raspolaže,

bilo da se radi o prikupljanju informativnog karaktera, samo za nadzor stanja tla i vremenskih uvjeta, bilo za podršku donošenju odluka oko korištenja klasičnih, „ručnih“ sustava, ili kao što je slučaj u ovom radu, za vođenje automatskog sustava upravljanja. Za demonstraciju takvog načina rada, podaci sa senzora se prikupljaju i šalju u bazu preko jednog mikrokontrolera, a upravljanje se izvršava preko drugog. U nastavku je prikazan dio koda za prikupljanje i slanje podataka u bazu.

```
int sensorValue = analogRead(34)/4;
int sensorValue1 = analogRead(35)/4;

float h = dht.readHumidity();
float t = dht.readTemperature();

Serial.print("connecting to ");
Serial.println(host);

WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
}

String url = "/api/insert.php?time=" + String(Time) + "&date=" +
    String(Date) + "&temp=" + String(t) + "&hum=" + String(h)
    + "&moist=" + String(sensorValue) + "&moist1=" +
    String(sensorValue1) + "&senzor=senzor1";

Serial.print("Requesting URL: ");
Serial.println(url);

client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
```

Slika 41. Arduino kod za čitanje i slanje podataka sa senzora

Na samom početku se može vidjeti deklariranje varijabli u koje se spremaju vrijednosti očitane se senzora te pozivanje funkcija za čitanje tih vrijednosti. Sljedeći korak je uspostavljanje veze s klijentom, odnosno spajanje na WiFi mrežu.

Nakon toga se formira string koji se pridodaje URL-u servera i šalje na server u obliku HTTP GET zahtjeva. Primjer formiranog stringa je npr.:

http://www.projekt-host.xyz/api/insert.php?time=11:10:31&date=2021-06-21&temp=21.10&hum=34.50&moist=913 &moist1=725&senzor=senzor1

Dio stringa „*insert.php*“ predstavlja naziv PHP skripte na serveru koja čita dobiveni string (odnosno njegov dio iznad znaka ?) te parametre sadržane u njemu unosi u bazu. Korak unošenja u bazu se izvodi pomoću SQL upita sadržanog u PHP skripti.

```
if (isset($_GET['temp']) && isset($_GET['hum']) && isset($_GET['time'])
&& isset($_GET['date']) && isset($_GET['moist']) &&
isset($_GET['moist1']) && isset($_GET['senzor'])) {

    $temp= $_GET['temp'];
    $hum= $_GET['hum'];
    $moist= $_GET['moist'];
    $moist1= $_GET['moist1'];
    $senzor= $_GET['senzor'];

    $time = $_GET['time'];
    $date = $_GET['date'];

    $filepath = realpath (dirname(__FILE__));
    require_once($filepath."/db_connect.php");

    $db = new DB_CONNECT();

    $result = mysql_query("INSERT INTO t_mjerenje_plastenik
(temp,hum,moist,moist1,datum_mjerenja,vrijeme_mjerenja,senzor)
VALUES ('$temp', '$hum', '$moist', '$moist1', '$date', '$time', '$senzor')");
```

Slika 42. PHP skripta za unos podataka u bazu

U isječku koda se može vidjeti spremanje primljenih parametara u varijable i provjera njihovog primitka, zatim spajanje s bazom pomoću druge skripte, *db_connect.php*, a nakon uspješnog spajanja, slanje SQL upita.

5.6. Pretpostavke rada zasebnih podsustava

Na svaki od mjerenih parametara je moguće utjecati pomoću aktuatora, ako to dozvoljava dostupnost i pristupačnost potrebnih uređaja. U radu su korišteni samo vrlo jeftini i jednostavni uređaji tako da npr. nije moguće isušivati zrak jer su takvi uređaji na višoj razini kompleksnosti.

5.6.1. Regulacija vlažnosti zemlje

Za navodnjavanje zemlje u teglicama se mora voditi računa da zemlja nikad nije suha, tj. da korijen biljke uvijek bude u dodiru s vlažnom zemljom, ali isto tako se ne smiju događati razdoblja u kojima je zemlja duže vremena vrlo vlažna jer u tom slučaju može doći do truljenja, a i biljke dio kisika dobivaju iz zemlje što nije moguće ako je prisutno previše vode.

Kod ručnog navodnjavanja, iskustvo govori koliko često i koliko intenzivno je potrebno zalijevati zemlju, a i pogledom na tlo te dodirrom se dobivaju važne informacije. Pri korištenju sustava razvijenog u okviru ovog rada, neke od ovih informacija nisu dostupne. Kod navodnjavanja na daljinu, zemlju se ne može vidjeti pa je intenzitet zalijevanja vrlo nizak kako zabunom ne bi došlo do prekomjernog zalijevanja. Same pumpe su niskog protoka, a i voda prolazi kroz prskalice tako da se protok dodatno smanjuje pa je za zalijevanje potpuno suhe zemlje potrebno oko jedne minute aktivnosti pumpe. Time je omogućeno da se zalijevanje odvija u koracima i da se može provjeriti učinak jednog intervala na vrijednost vlažnosti sa senzora (puno teže bi bilo upaliti zalijevanje tri puta po tri sekunde nego tri puta po 15 sekundi, puno je manji prostor za pogrešnu procjenu). U slučaju automatskog zalijevanja, nije dostupan niti vizualni pregled stanja, a i iskustvo je teško pretočiti u numerički format jer se u stvarnosti češće radi o osjećaju nego o jasnom planu. Iz tog razloga se kod automatskih načina konstantno provjeravaju vrijednosti parametara i poduzimaju mjere korekcije.

5.6.2. *Regulacija temperature zraka*

Za podizanje temperature zraka, ranije opisani grijač postavljen na dno plastenika je i više nego dovoljan. Ipak, kako bi se izbjeglo prekomjerno lokalno zagrijavanje, grijač se koristi u paru s ventilatorom koji ubrzava širenje i miješanje toplog zraka. S obzirom na to da se u radu koristi plastenik vrlo malih dimenzija, zamjena ukupne količine zraka je lako izvediva pomoću ventilatora. Za svrhu regulacije temperature zraka, ta metoda nije naročito korisna jer je ventilacijom moguće samo izbaciti topli zrak i uvesti zrak sobne temperature. U radu nije predviđeno hlađenje zraka kojeg se uvodi, ali je zato moguće rashladiti zrak unutar plastenika dovodenjem vodene pare, pomoću difuzora. Doprinos ventilacije je i u tome da se zamjenom zraka može nadoknaditi iskorišteni ugljikov dioksid.

5.6.3. *Regulacija vlažnosti zraka*

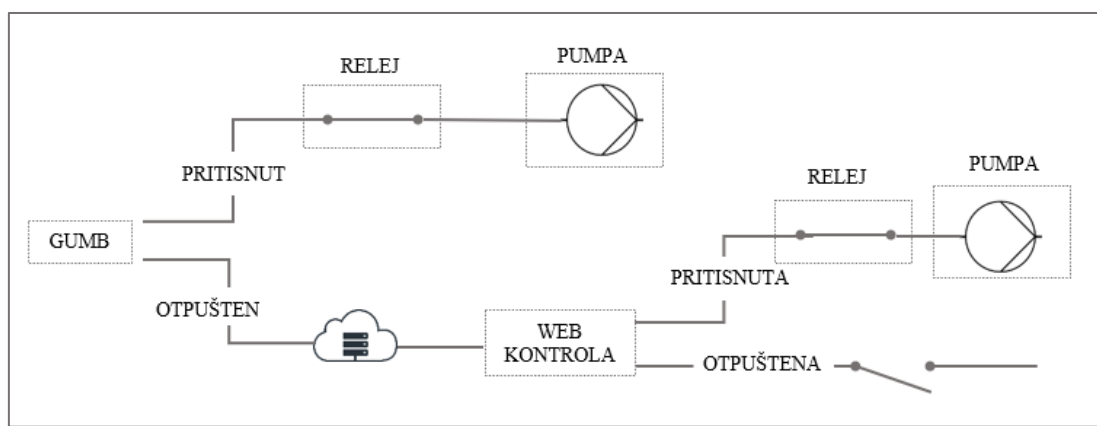
Ventilacijom je također moguće postići nižu relativnu vlažnost zraka jer zrak u većini klima najčešće ne doseže vrijednosti koje su optimalne za razvoj biljaka. S druge strane, za podizanje vlažnosti je potrebno u zrak dovoditi vodenu paru na drugačiji način, a u ovom radu je odabran difuzor. Budući da je gustoća vodene maglice viša od gustoće zraka, izražena je tendencija gotovo neprimjetnog rasta vlažnosti pri vrhu plastenika pa se za homogenizaciju u paru s difuzorom koristi ventilator koji pospješuje miješanje zraka i pare.

5.7. Objašnjenje predviđenih načina upravljanja

Kako bi se korisniku omogućila što veća razina kontrole nad platenikom, zamišljena su tri stupnja autonomije upravljanja.

5.7.1. Ručno upravljanje

Ručni način upravljanja je moguć pomoću fizičkih tipki na konzoli na bazi platenika te preko kontrola u web aplikaciji. Upravljanje fizičkim tipkama nije zamišljeno kao primarni način upravljanja (jer je ideja rada upravljanje na daljinu), ali je ipak dostupna i ta mogućnost kao rezerva u slučaju prekida veze sa serverom.



Slika 43. Shema ručne kontrole pumpe za navodnjavanje

Slika 43. ugrubo prikazuje logiku paljenja pumpe u ručnom načinu rada. Budući da postoje dva prekidača za kontrolu pumpe (fizički i u web aplikaciji), potrebno je izbjeći koliziju između njih, tj. osigurati da je naredba prema pumpi jednoznačna. U stanju dok niti jedna kontrola nije pritisnuta, sustav čeka naredbu s obje, a prioritet ima ona koja se aktivira prva.

Ako se pritisne kontrola u web aplikaciji, bit će poslan JSON GET zahtjev prema bazi podataka koji će promijeniti vrijednost u tablici `t_relej`. S druge strane, mikrokontroler provjerava je li pritisnut fizički gumb, i ako nije, šalje HTTP GET zahtjev prema bazi te čita vrijednost iz tablice `t_relej`. Očitana vrijednost se koristi kao argument u funkciji za kontrolu releja. Čitanje vrijednosti za sve predviđene web kontrole traje oko jedne sekunde i to vrijeme predstavlja kašnjenje odziva pumpe na promjenu kontrole. Ako je fizički gumb pritisnut, korak čitanja vrijednosti iz baze se preskače sve dok se gumb ne otpusti. Isječak koda za upravljanje pumpom je prikazan ispod (zbog nepotpune kompatibilnosti korištenog mikrokontrolera i knjižnice za čitanje prekidača, atributi „*isReleased*“ i „*isPressed*“ su zamijenjeni).

```

gumb_p1.read();
if (gumb_p1.isReleased()) { // gumb pritisnut

    digitalWrite(p1Pin, LOW);

}
if (gumb_p1.isPressed()) { // gumb pušten

    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
        Serial.println("connection failed");
        return;
    }

    Call_database("1");
    if (led == "off") {
        digitalWrite(p1Pin, 1);
        Serial.println("p1 is off..!");
    }
    else if (led == "on") {
        digitalWrite(p1Pin, 0);
        Serial.println("p1 is on..!");
    }
    Serial.println();
    delay(100);
}

```

Slika 44. Arduino kod za ručno upravljanje

Funkcija „*Call_database*“ je zadužena za slanje zahtjeva prema bazi. Atribut „1“ označava ID releja u tablici *t_relej*, odnosno aktuatora kojim se upravlja, a preko njega se dobiva linija u tablici koju je potrebno pročitati. Vrijednosti u tablici su „on“ i „off“, a preko njih funkcija određuje koji atribut da proslijedi na pin na kojeg je priključen kanal releja. U prikazanom slučaju, radi se o pinu *p1Pin*, a na njega je spojena pumpa 1.

```

void Call_database(String id) {

    String result;
    HTTPClient http;

    http.begin("http://www.projekt-host.xyz/api/read.php?id=" + id);
    int httpCode = http.GET();

    if (httpCode > 0) {

        result = http.getString();

    }
}

```

Slika 45. Arduino kod za čitanje stanja releja

U isječku koda funkcije *Call_database* se može vidjeti njezin glavni element, a to je pozivanje URL-a PHP skripte za čitanje podataka. Ispod je prikazan isječak koda PHP skripte *read.php* koja bazi šalje SQL upit za čitanjem retka prema odgovarajućem ID parametru.

```
$db = new DB_CONNECT();

if (isset($_GET["id"])) {
    $id = $_GET['id'];

    $result = mysql_query("SELECT *FROM t_relej WHERE id =
'id'");

    if (!empty($result)) {

        if (mysql_num_rows($result) > 0) {

            $result = mysql_fetch_array($result);

            $led = array();
            $led["id"] = $result["id"];
            $led["status"] = $result["status"];

            $response["success"] = 1;

            $response["led"] = array();

            array_push($response["led"], $led);

            echo json_encode($response);
```

Slika 46. PHP skripta za provjeru stanja releja

Svi ostali aktuatori se kontroliraju na identičan način, jedina je razlika u vrijednosti ID parametra.

5.7.2. Automatsko praćenje ručno postavljenih granica

Drugi način upravljanja ne zahtijeva intervenciju korisnika nego sustav sâm pokušava zadržati parametre unutar unaprijed zadanih raspona. Kao granične vrijednosti se koriste iste one kao i za vizualna upozorenja na naslovnoj stranici (Slika 35.), sadržane na stranici *Postavke*, što znači da korisnik može podesiti iznose parametara koje želi ostvariti.

Sustav radi tako da očita granične parametre i zadnji unos sa senzora iz baze, usporedi njihove iznose te na temelju njihovog odnosa donese odluku o potrebi za aktivacijom aktuatora. Pravila za odlučivanje glase:

- ako je stvarna temperatura niža od granične, aktivira se grijač
- ako je stvarna vlažnost zraka niža od granične, aktivira se difuzor
- ako je stvarna vlažnost tla niža od granične, aktivira se pumpa
- ako je stvarna temperatura viša od granične ili kao je stvarna vlažnost zraka viša od granične, aktivira se ventilator

```
void loop() {  
  
    ReadData();  
    ReadDataGranice();  
  
    if (actmoist1 > moist1Granica)  
    {  
        startZal = millis();  
        while (millis() - startZal <= zal)  
        {  
            digitalWrite(p1Pin, LOW);  
            Serial.println("pumpa 1 on");  
  
            delay(50);  
        }  
  
        digitalWrite(p1Pin, HIGH);  
        Serial.println("pumpa1 off");  
    }  
}
```

Slika 47. Arduino kod za automatsko upravljanje

U prikazanom kodu se može vidjeti početak automatskog ciklusa s jednom petljom za kontrolu pumpe. Na samom početku se čitaju podaci iz baze, zatim se provjerava odnos očitanih vrijednosti za svaki parametar i onda se izvodi petlja za paljenje pumpe. Varijabla *zal* zapravo služi kao brojač koji definira duljinu izvođenja petlje, a njezin iznos je određen pri deklaraciji varijable na početku programa. Funkcija *millis()* vraća vrijednost u milisekundama koja predstavlja vrijeme proteklo od početka pokretanja koda na mikrokontroleru (mjeri se pomoću ugrađenog kristalnog oscilatora s devijacijom oko 2,5 sekundi po danu), a u *while* petlji se od trenutne vrijednost funkcije *millis()* konstantno oduzima njezina vrijednost na samom početku izvođenja petlje zabilježena u varijabli *startZal*. Ta razlika predstavlja proteklo vrijeme, a usporedbom s varijablom *zal* se dobiva vrijeme kad se petlja treba prekinuti. Prednost korištenja funkcije *millis()* u odnosu na klasičnu funkciju *delay()* kojom se inače određuju trajanja je ta što *delay()* zaključava program dok ne istekne, a to bi značilo da se ne bi moglo prebacivati između načina upravljanja trenutačno, već s odmakom definiranim iznosom funkcije *delay()*. I

za ostale parametre koje se kontrolira je kod vrlo sličan, jedino što se mijenja su varijable brojača.

Na kraju svakog automatskog ciklusa je postavljena pauza kako bi se efekti poduzetih koraka staložili i osiguralo točnije očitavanje sa senzora. Pauza je opet izvedena pomoću *millis()* funkcije.

```
cekanje = millis();  
while (millis() - cekanje <= 15000) {  
    delay(50);  
}
```

Slika 48. Brojač za određivanje stanke u programu

Podaci se iz baze čitaju na vrlo sličan način kao što su se u prethodnom poglavlju mijenjali, pomoću HTTP GET zahtjeva, samo što je u ovom slučaju drugačiji SQL upit u PHP skripti. Na primjeru se može vidjeti dio skripte s upitom za čitanje zadnjeg unosa u bazi.

```
$sql= "SELECT `temp`,`hum`,`moist`,`moist1` FROM  
`t_mjerenje_plastenik` ORDER BY id_mjerenje desc LIMIT 1;";  
$actual = $conn->query($sql);  
  
if ($actual->num_rows > 0) {  
    // output data of each row  
    while($row = $actual->fetch_assoc()) {  
        $acttemp = ($row["temp"]);  
        $acthum = ($row["hum"]);  
        $actmoist1 = ($row["moist"]);  
        $actmoist2 = ($row["moist1"]);  
    }  
}
```

Slika 49. PHP skripta za čitanje zadnjeg unosa iz baze

Za razliku od prošlog primjera gdje se čitala vrijednost samo jedne ćelije, u slučaju čitanja više podataka je potreban dodatan korak raščlanjivanja (eng. *parsing*) poruke koju se dobiva GET zahtjevom. PHP skripta je podešena tako da podatke šalje kao niz odvojen zarezima u formatu *temp,hum,moist1,moist2* (npr. 25,40,714,925) pa je potrebno svaki član zasebno spremiti kao varijablu. Za taj zadatak se koristi nova funkcija *StringReceive()* koja je prikazana ispod, a iz komentara u kodu se mogu vidjeti objašnjenja pojedinih koraka.

```
void StringReceive1()
{
    String strArr[4]; //broj varijabli

    int stringStart = 0;
    int arrayIndex = 0;

    for (int i = 0; i < payload1.length(); i++) {
        //provjera koji je znak zarez
        if (payload1.charAt(i) == ',') {
            //pražnjenje stringa
            strArr[arrayIndex] = "";
            //spremanje znakova prije zareza u array
            strArr[arrayIndex] = payload1.substring(stringStart,
i);
            //postavljanje nove početne točke za provjeru
            stringStart = (i + 1);
            arrayIndex++;
        }
    }

    //spremanje zasebnih stringova
    String value1 = strArr[0];
    String value2 = strArr[1];
    String value3 = strArr[2];
    String value4 = strArr[3];
    //pretvaranje iz string u float
    acttemp = value1.toFloat();
    acthum = value2.toInt();
    actmoist1 = value3.toInt();
    actmoist2 = value4.toInt();
}
```

Slika 50. Arduino kod za raščlanjivanje stringa

5.7.3. Neizrazito upravljanje

Najnapredniji i potpuno automatizirani način upravljanja podiže razinu detaljnosti kontrole parametara tako što se umjesto ručno definiranih pravila za aktivaciju koriste fuzzy logika. Iako postoji način izravne komunikacije Arduina i Matlaba, odabran je vanjski alat [31] za pretvaranje FIS datoteke u Arduino jezik. Budući da se u ovom slučaju radi o nekoliko stotina linija koda, puni kod će biti priložen na kraju rada, a u ovom poglavlju će biti prikazane samo ključne funkcije.

```
ReadData();

g_fisInput[0] = acttemp;
g_fisInput[1] = acthum;
g_fisInput[2] = 0;

g_fisOutput[0] = 0;
g_fisOutput[1] = 0;
g_fisOutput[2] = 0;
g_fisOutput[3] = 0;

fis_evaluate();

unsigned long startfan = millis();
Serial.println("");
Serial.println("fan on");
while (millis() - startfan <= (1000 * g_fisOutput[0]))
{
    digitalWrite(fPin, LOW);
    delay(50);
}
digitalWrite(fPin, HIGH);
Serial.println("fan off");
```

Slika 51. Arduino kod za neizravno upravljanje

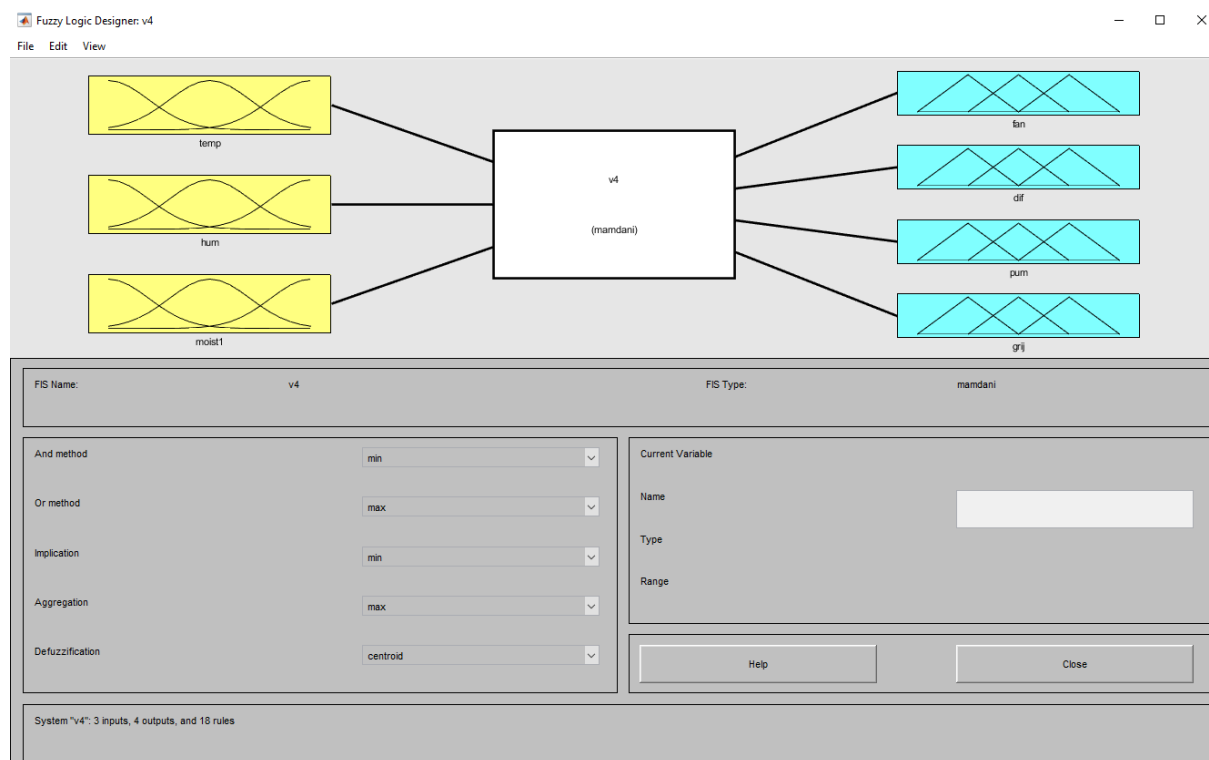
I ovaj način upravljanja, kao i prethodni, počinje s čitanjem podataka sa senzora iz baze. Zatim se očitane vrijednosti spremaju u ulazne varijable fuzzy funkcije, *g_fisInput []*. S obzirom da je prikazan dio koda za kontrolu ventilatora, koriste se samo dva parametra na koje ventilator utječe, a treći (vlažnost tla) se zanemaruje. Izlazne varijable se postavljaju na nulu i poziva se funkcija *fis_evaluate()* koja zapravo pretvara ulaze u izlaze. *g_fisOutput [1]* je izlazna varijabla koja nam treba za slučaj ventilatora i nju se koristi kao referentnu vrijednost u brojaču (pomnoženu s 1000 jer je njezin iznos u sekundama, a brojač radi u milisekundama). Za kontrolu grijača i difuzora se na isti način koriste sljedeće dvije izlazne varijable, a za kontrolu pumpi je postupak malo drugačiji.

S obzirom da su u Matlab modelu pretpostavljena identična pravila za oba regulacijska kruga za navodnjavanje i predviđen samo jedan ulaz za vlažnost tla, za kontrolu svake pumpe se ne može koristiti izlaz dobiven u prvom izvođenju funkcije *fis_evaluate()*, već se postupak definiranja varijabli i izvođenja funkcije ponavlja jednom za svaku pumpu.

5.8. Modeliranje FIS-a

Za modeliranje neizrazitog sustava je odabran Matlab i njegov *Fuzzy logic toolbox*. Izrada počinje definiranjem vrste neizrazitog sustava, a odabran je mam, te odabirom broja ulaznih i

izlaznih varijabli u *Fuzzy logic designeru*. Mamdani je vrsta neizrazitih sustava pogodna za korištenje lingvističkih varijabli dok Sugeno koristi jednodimenzionalne funkcije pripadnosti (*singleton*) pa je inferencija brža što ga čini pogodnijim za adaptivne sustave koji uče iz velikog broja podataka.



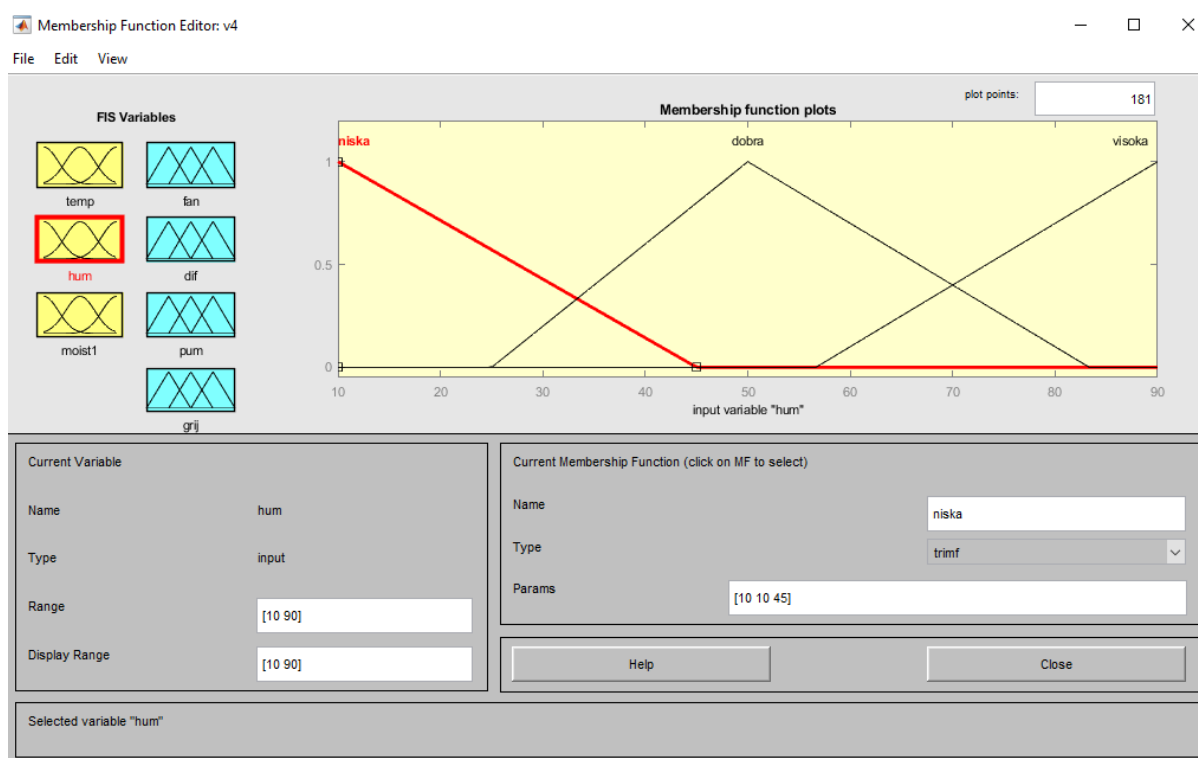
Slika 52. *Fuzzy Logic Designer*

Kao što je vidljivo sa slike, definirana su samo tri ulazna parametra što ne odgovara zadanom planu kontroliranja 4 parametra (temperature, vlažnosti zraka i vlažnosti tla na dva mjesta). Međutim, s obzirom da je pretpostavljeno da su karakteristike obaju lokacija identične (svojstva zemlje, protok pumpe), a ne postoji njihova međuovisnost, za više lokacija će se podaci više puta provesti kroz model. Ostale postavke u prozoru se tiču naprednijeg podešavanja koraka inferencije i u ovom radu nisu korištene, već su ostavljene osnovne postavke (centroidna metoda defuzifikacije).

5.8.1. *Funkcije pripadnosti*

Sljedeći korak je podešavanje funkcija pripadnosti. Funkcija pripadnosti svakog parametra ima za cilj definirati raspone koji će što točnije dočaravati stvarne doživljaje korisnika. Iz tog je razloga za određivanje funkcija pripadnosti konzultiran stručnjak za plasteničku proizvodnju koji je pomogao odrediti granice raspona. Dvostrukim klikom na bilo koji od parametara se

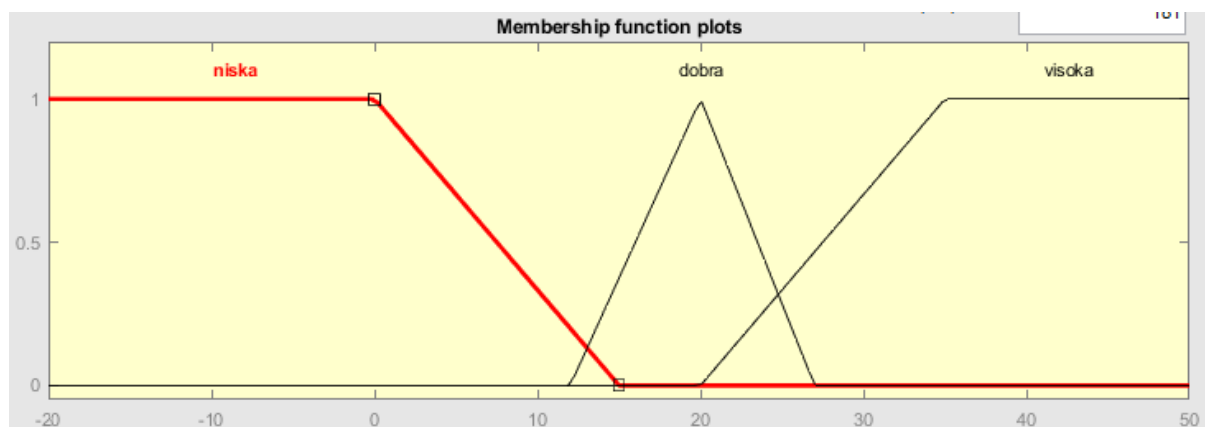
otvara sučelje za uređivanje funkcija pripadnosti. Može se vidjeti kako su funkcije pripadnosti pretežito trokutaste, a to je odabrano zbog jednostavnosti, zato što su trokutaste funkcije definirane samo trima parametrima (vidljivo sa slike 53. pod poljem „Params“) dok je kod drugih oblika broj parametara znatno veći (npr. 9 za trapezoidnu). Drugi faktor kod izbora je i pogodnost za buduću optimizaciju, jer jednostavnije funkcije brže konvergiraju [32]. Iznimka su funkcije pripadnosti za nisku i visoku temperaturu koje su trapezoidalne kako bi se svom rasponu ekstremnih temperatura pridružila vrijednost pripadnosti 1 (temperature ispod 0 i iznad 35) te funkcija za nisku vlažnost tla (trapezoidalna ispod 550). Osim oblika funkcija pripadnosti, potrebno je odabrati i njihov broj, a za sve varijable je taj broj 3, kako bi se stručnjaku dalo dovoljno široke raspone i omogućilo njihovo jasno određivanje



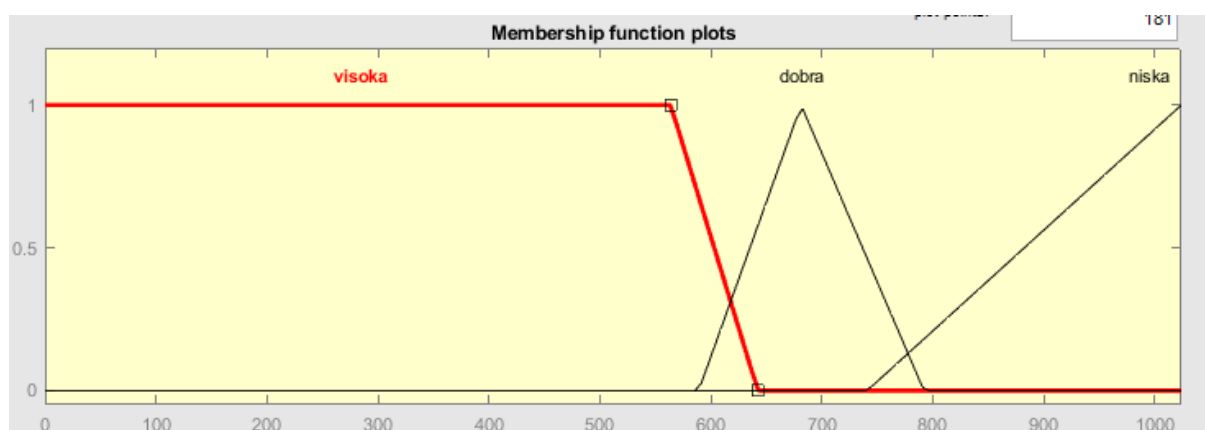
Slika 53. *Membership function editor* – funkcija pripadnosti za vlažnost zraka

Nakon određivanja funkcija pripadnosti za mjerne parametre, moraju se odrediti i funkcije pripadnosti za odzive, za aktuatore. Izlaz *fuzzy* sustava za aktuatore je trajanje u sekundama. Postupak je identičan, jedino se za određivanje vrijednosti ne koristi isključivo znanje konzultiranog stručnjaka, jer on znanje o korištenoj opremi niti nema, već je djelovanje pojedinih uređaja za upravljanje parametrima zabilježeno kvantitativno te je stručnjak na temelju tih podataka dao prijedloge o potrebnom djelovanju, točnije, vremenima aktivacije.

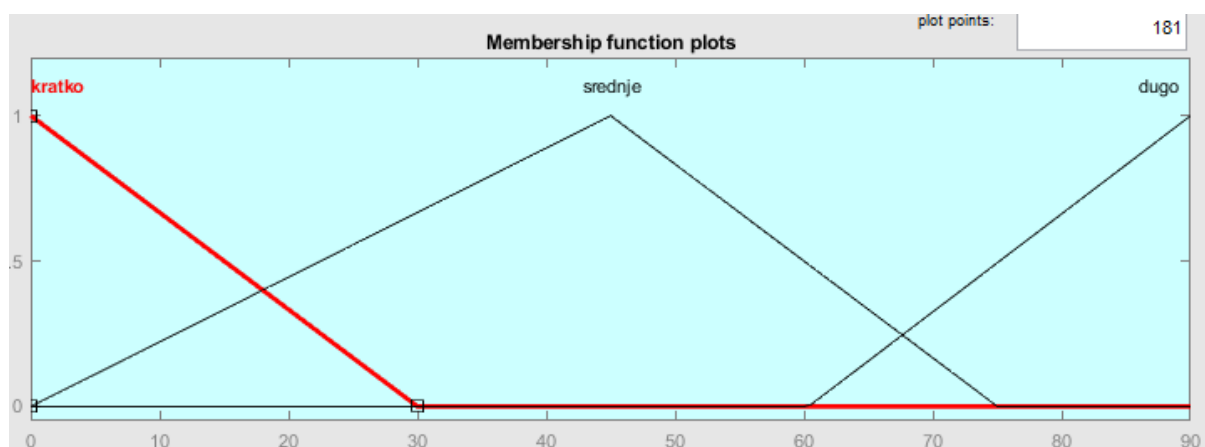
U nastavku će biti prikazane sve funkcije pripadnosti korištene u radu.



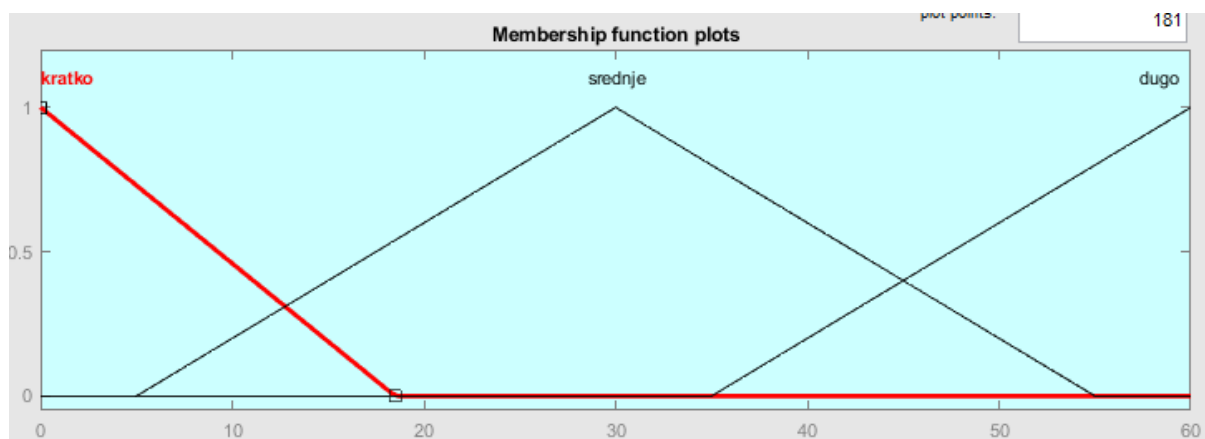
Slika 54. Funkcije pripadnosti – temperatura zraka



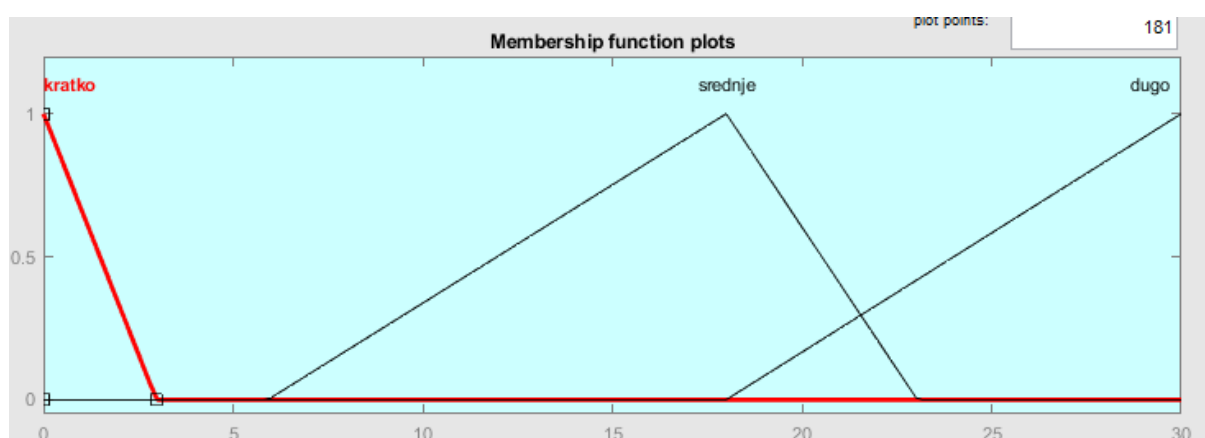
Slika 55. Funkcije pripadnosti – vlažnost tla



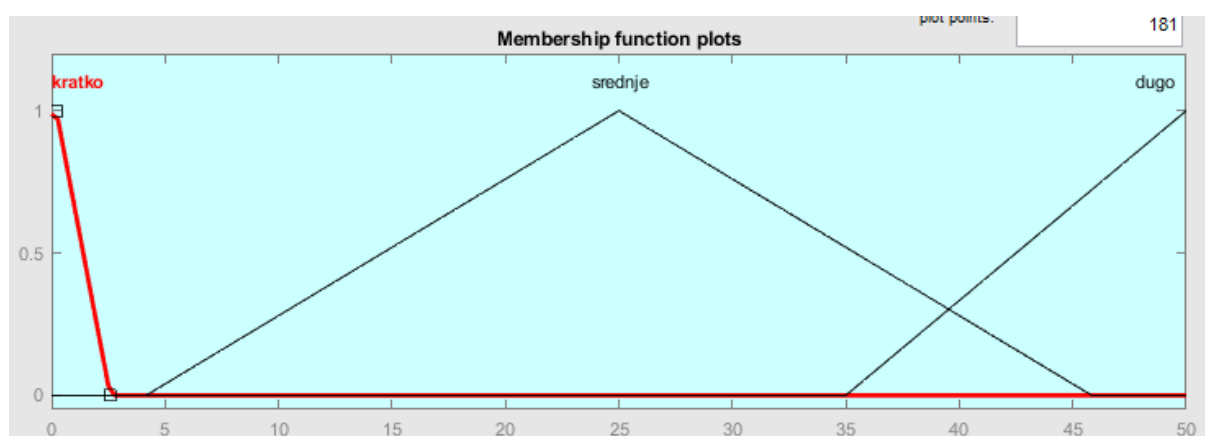
Slika 56. Funkcije pripadnosti – ventilator



Slika 57. Funkcije pripadnosti – difuzor



Slika 58. Funkcije pripadnosti – pumpa



Slika 59. Funkcije pripadnosti – grijač

Tablica prikazuje vrijednosti svih karakterističnih točaka vidljivih na ranijim slikama.

Tablica 3. Rasponi vrijednosti u *fuzzy* modelu

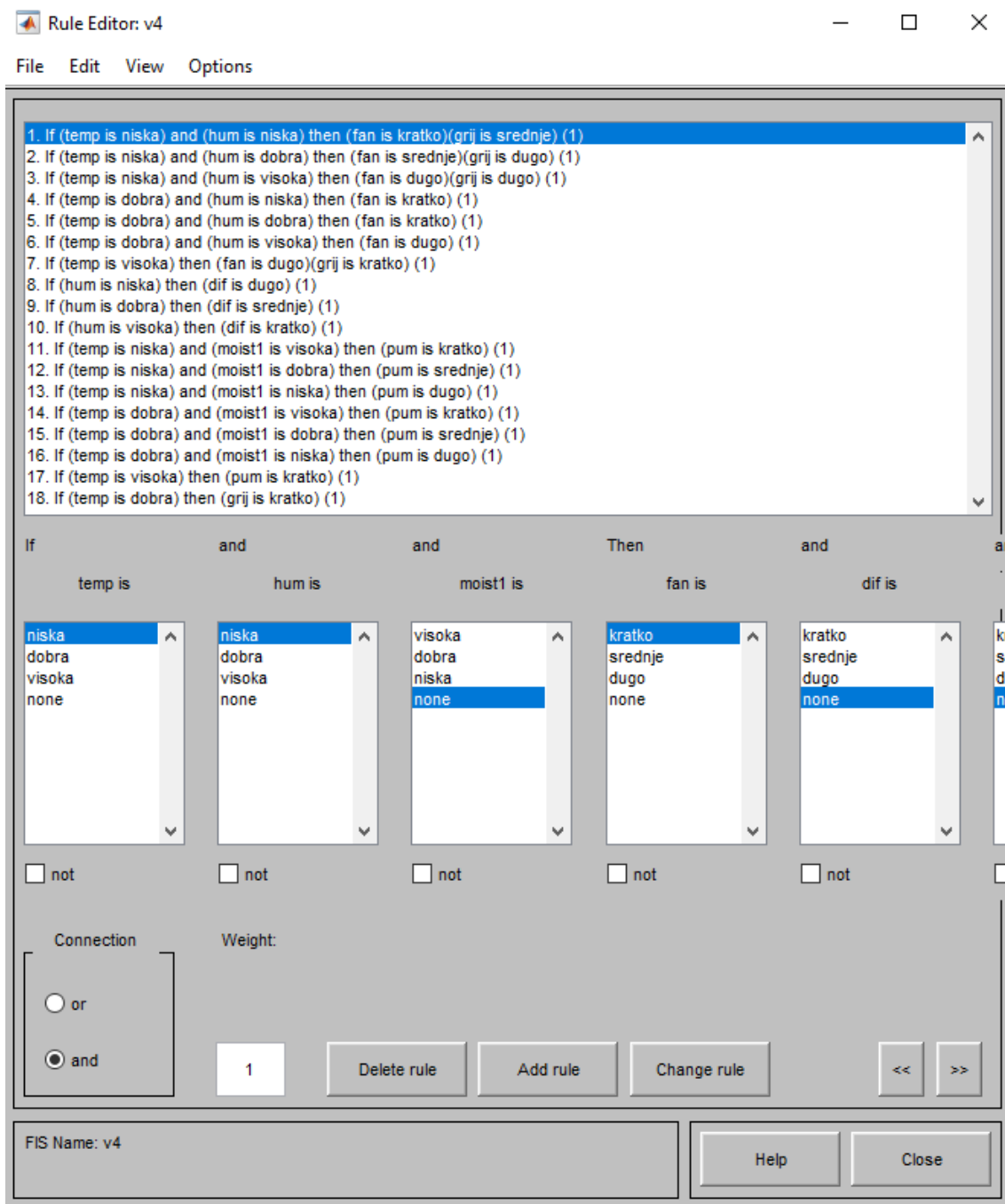
PARAMETAR	LINGVISTIČKA VARIJABLA	RASPON		
		NIŽA VRIJEDNOST	SREDINA	VIŠA VRIJEDNOST
Temperatura zraka	Niska	-20	0	15
	Dobra	11	20	27
	Visoka	20	35	50
Vlažnost zraka	Niska	10	/	45
	Dobra	25	50	83
	Visoka	57	/	90
Vlažnost tla	Niska	0	554	640
	Dobra	590	680	790
	Visoka	740	/	1024
Ventilator	Kratko	0	/	30
	Srednje	0	45	75
	Dugo	60	/	90
Difuzor	Kratko	0	/	18,5
	Srednje	5	30	55
	Dugo	35	/	60
Pumpa	Kratko	0	/	3
	Srednje	6	18	23
	Dugo	18	/	30
Grijač	Kratko	0	/	2,5
	Srednje	4	25	46
	Dugo	35	50	71

5.8.2. Pravila neizrazitog sustava upravljanja

Nakon što su definirane funkcije pripadnosti, potrebno je urediti pravila za inferenciju kako bi sustav na temelju očitanih vrijednosti i preračunatih pripadnosti izračunao pripadnosti za izlaze. Sučelje *Rule Editor* daje tu mogućnost, a više se može vidjeti na slici 60. Sva pravila su u formatu *if-then*, a nekoliko je glavnih pravila koje se pratilo pri izradi:

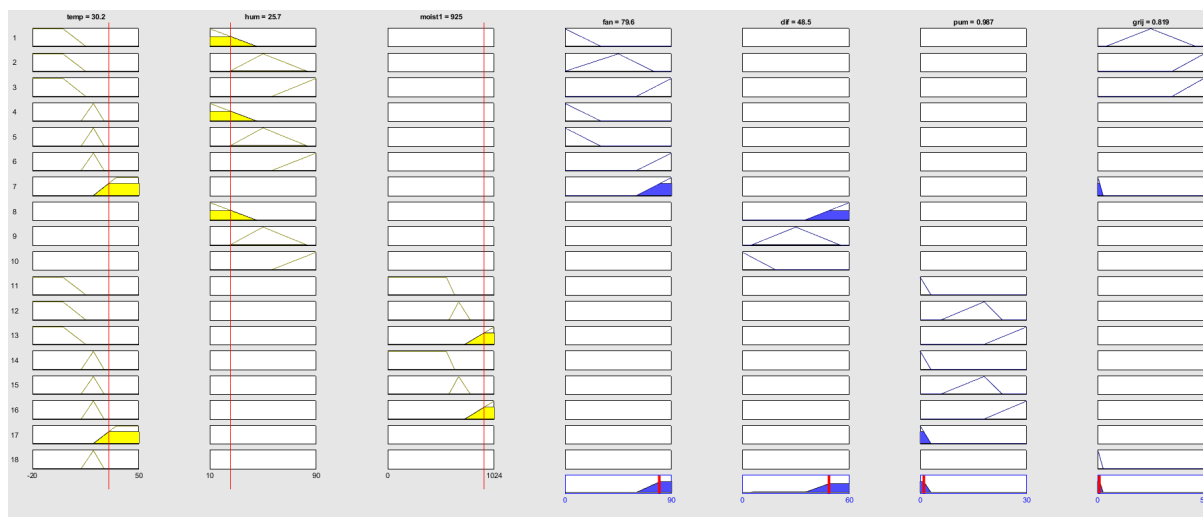
- grijač se koristi za podizanje temperature
- difuzor se koristi za podizanje vlažnosti zraka
- ventilator se koristi za smanjivanje temperature i vlažnosti zraka

- pumpa se koristi za podizanje vlažnosti tla
- zalijevanje se ne smije koristiti pri visokim temperaturama



Slika 60. Rule Editor – pravila sustava

Sučelje *Rule Viewer* daje vizualni prikaz donošenja odluka na temelju ulaznih varijabli, koristeći operacije s neizrazitim skupovima prema funkcijama zadanim funkcijama pripadnosti, a odluka za jedan slučaj, tj. jedan set ulaznih varijabli je prikazana na sljedećoj slici.



Slika 61. Rule Viewer – vizualni prikaz inferencije

U sljedećoj tablici su numerički prikazani svi rubni uvjeti sustava, kao i odzivi sustava za te slučajeve. Također je prikazan i slučaj sa slike 61.

Tablica 4. Ulazi i izlazi za rubne uvjete

Temperatura	Vlažnost zraka	Vlažnost tla	Ventilator	Difuzor	Pumpa	Grijač
30,2	25,7	925	79,6	48,5	1,0	0,8
-20	10	0	9,7	51,9	0,9	25
-20	10	1024	9,7	51,9	26,1	25
-20	90	0	80,4	6,0	0,9	45,2
-20	90	1024	80,4	6,0	26,1	45,2
50	10	0	80,4	51,9	0,9	0,7
50	10	1024	80,4	51,9	0,9	0,7
50	90	0	80,4	6,0	0,9	0,7
50	90	1024	80,4	6,0	0,9	0,7

Iz odziva sustava su jasno vidljiva neka od ranije objašnjenih pravila. Npr. pri visokim temperaturama je pumpa kratko aktivna iako je vlažnost tla vrlo niska.

5.9. Izrađeni model za demonstraciju rada sustava

Kako bi se prikazao rad svega navedenog ranije, komponente su smještene u model plastenika. Model je izrađen od drveta, a za stranice je korišten pleksiglas debljine 2 mm jer pruža dovoljno dobra mehanička svojstva i otpornost na udarce.

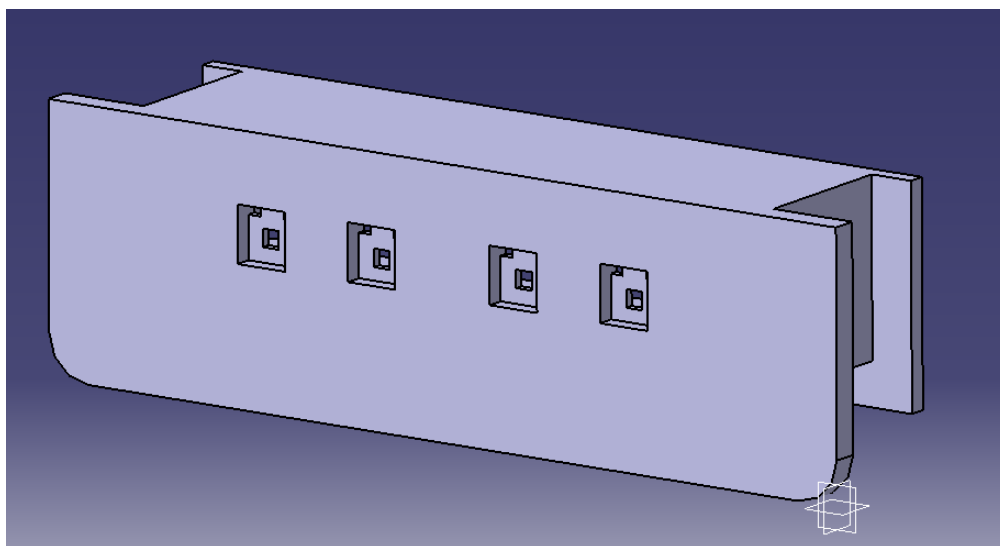


Slika 62. Model plastenika

Model sadrži dvije police kako bi se istovremeno mogle pratiti dvije kulture. Na stranici se može vidjeti konzola s gumbima za ručno upravljanje opremom. Konzola je izrađena od PLA polimera, procesom 3D ispisa, jer gotova rješenja za potrebe ovog rada nije bilo moguće pronaći. Ispod su priloženi bliži pogled te 3D model komada.



Slika 63. Konzola s gumbima



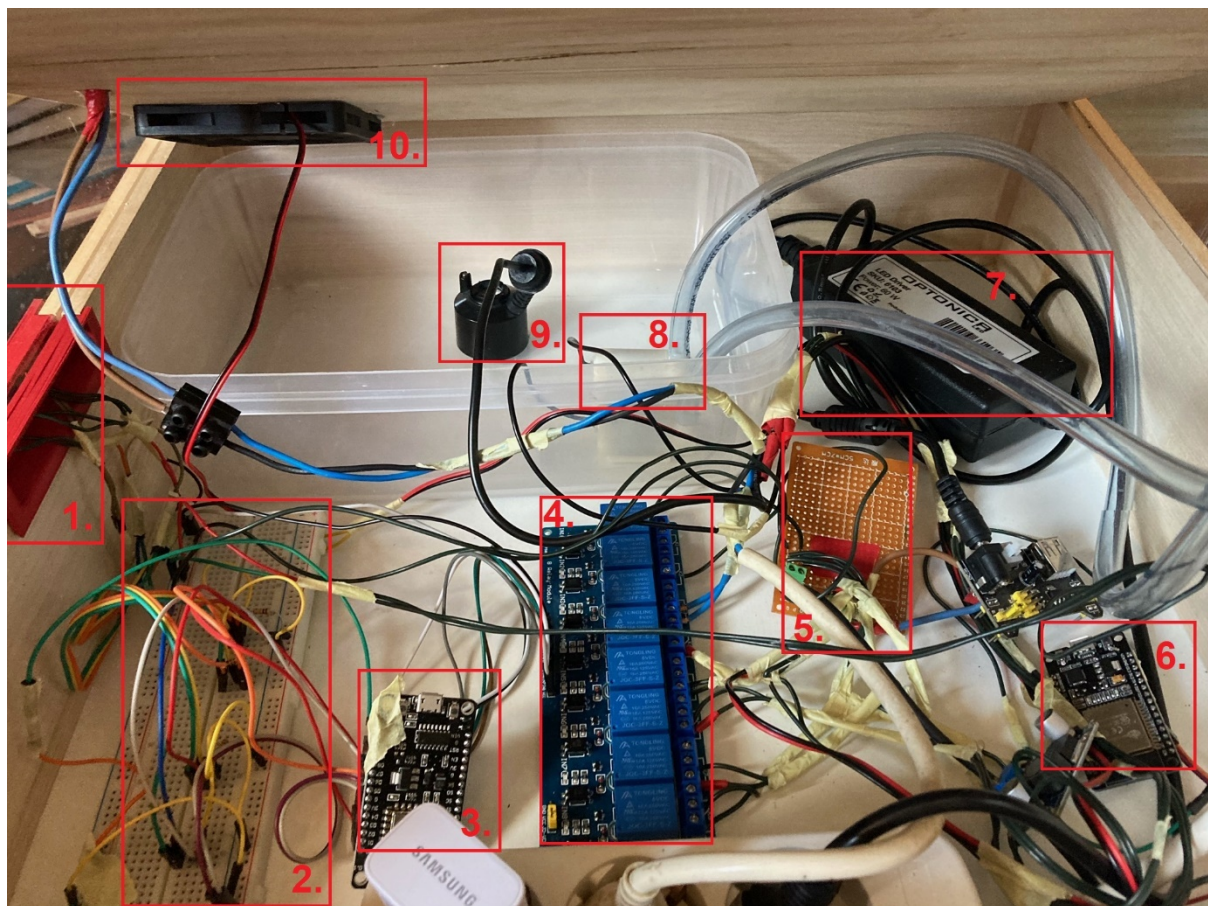
Slika 64. Konzola – 3D model, Catia

Na slici 62. se također mogu vidjeti teglice sa zemljom koje se koriste za testiranje rada sustava za navodnjavanje, a bliži pogled, sa vidljivim senzorom za vlažnost tla te prskalicama spojenim na crijevo od pumpe, se može vidjeti na slici ispod.



Slika 65. Teglica sa senzorom i prskalicama

Na slici 62. se može vidjeti i grijač, kojeg bi inače bilo idealno smjestiti van modela, kako ne bi bio izložen nepogodnim uvjetima atmosferskim uvjetima, te onda dovod zraka usmjeriti preko njega, ali to je bio prezahtjevan pothvat za ovaj rada. Izravno ispod grijača se nalazi ventilator koji pospješuje cirkulaciju toplog zraka, a može ga se vidjeti, kao i ostatak opreme koja se nalazi ispod donje police, na slici ispod.



Slika 66. Prikaz opreme u modelu

Oprema vidljiva na slici je:

1. Konzola za gumbe – stražnja strana
2. Eksperimentalna pločica – spajanje većine kablova za komunikaciju te krugovi za upravljanje gumbima
3. ESP12-E – mikrokontroler za upravljanje
4. Relejna ploča
5. Razvodna pločica za napajanje – na nju su spojena napajanja od 5V i 12V
6. ESP 32 – mikrokontroler za prikupljanje podataka
7. Napajanje 12V
8. Pumpa
9. Difuzor
10. Ventilator

6. ZAKLJUČAK

Preljevanje koncepata Industrije 4.0 u svakodnevni život je porasla osviještenost krajnjih korisnika o prednostima koje digitalizacija donosi, ali i dalje je prepreka visok trošak kojeg transformacije sustava predstavlja. U ovom radu su prikazane cjenovno vrlo pristupačne komponente te je postavljena osnova za daljnji rad na proširenju funkcionalnosti sustava. Pristup razvoju je bio takav da se omogući lakše proširivanje sustava, s fokusom na mogućnosti modularne nadogradnje.

Trenutno stanje sustava omogućava prikupljanje podataka s više lokacija na jedno mjesto, njihov pregled koji omogućava dobivanje informacija o sustavu pod nadzorom i njihovo korištenje za donošenje odluka. Razvijeni načini upravljanja su upotrebljivi, pogotovo ručno upravljanje koristeći logiku korisnika, a i automatski načini rada uspješno izvršavaju zadatak, dok im točnost ovisi o agrotehničkom znanju programera pa bi se na podizanje točnosti trebalo dodatno usredotočiti.

Naravno, izrađeni sustav nije ograničen isključivo na poljoprivrednu proizvodnju, već se razvijeni okvir može prilagođavati i drugim potrebama. Održavanje proizvodnih sustava je možda najbliže područje s obzirom na zahtjeve, a korištenje specifičnih senzora (npr. akcelerometara, akustičnih senzora, itd.) bi se moglo iskoristiti za nadzor stanja strojeva i za prediktivno i preskriptivno održavanje. Neizrazito upravljanje je također vrlo primjenjivo za takve zadatke pa bi sustav u kojeg je moguće jednostavno implementirati gotov fuzzy model bio od velike koristi.

Nekoliko je smjerova u kojima bi daljnji razvoj mogao ići. Nadogradnja koju bi se najlakše primijenilo je unaprijeđenje modela neizrazitog upravljanja. Opsežno su dokumentirani primjeri korištenja strojnog učenja i neuronskih mreža za podešavanje funkcija pripadnosti i pravila, a bilježenje implicitnog znanja korisnika kroz upotrebu sustava je rješenje koje bi najviše unaprijedilo sustav. Jedna od nadogradnja bi svakako bilo podizanje autonomije sustava po pitanju pristupa energetske mreže i internetu jer su poljoprivredni sustavi često na udaljenim lokacijama. Problemu opskrbe energijom se može pristupiti s dvije strane. Jedna je optimizacija koda i korištenje rješenja za vrlo nisku potrošnju energije, kroz metode dubokog sna gdje se sustav pali samo kad je potrebno izvršiti neku radnju, a ostatak vremena je *offline*. Druga strana je korištenje alternativnih izvora energije, npr. pomoću fotonaponskih ćelija. Pristup internetu je također moguće riješiti na više načina. Jedan način je priključivanje na GSM mrežu, pomoću

modula za SIM karticu. Drugi način je međusobno umrežavanje mikrokontrolera s obzirom da NodeMCU ima mogućnost otvaranja lokalne mreže. Umrežavanjem više mikrokontrolera je moguće ostvariti domet daleko izvan dometa jednog mikrokontrolera. Tu je i mogućnost korištenja drugih načina komunikacije, npr. ranije opisani LoRa protokol, ali i smještanje baze podataka na Raspberry Pi i pristup internetu preko njega. Izvođenje *fuzzy* logike na serveru bi doprinijelo efikasnosti sustava, a i omogućilo bi korištenje puno kompleksnijih modela za upravljanje.

LITERATURA

- [1] N. Parađiković i Ž. Kraljićak, Zaštićeni prostori - plastenici i staklenici, 2008.
- [2] »Heating system in greenhouse: the advantages of combiTERM,« [Mrežno]. Available: <https://www.pericoli.com/EN/news/35/Heating-system-in-greenhouse-the-advantages-of-combiTERM.html>. [Pokušaj pristupa 14. 3. 2021.].
- [3] »Natural Ventilation Greenhouse,« [Mrežno]. Available: https://www.gothicarchgreenhouses.com/Natural_Ventilation.htm. [Pokušaj pristupa 14. 3. 2021.].
- [4] »Shade Cloth - Aluminet Shade Cloth,« [Mrežno]. Available: https://www.gothicarchgreenhouses.com/shade_cloth.htm. [Pokušaj pristupa 14. 3. 2021.].
- [5] V. P. Sethi i S. S. K., »Survey of cooling technologies for worldwide agricultural,« *Solar Energy*, svez. 81, br. 12, pp. 1447-1459, 2007.
- [6] »Greenhouse Fog Cooling,« [Mrežno]. Available: <https://www.aisgreenworks.com.au/climate-systems/high-pressure-fog/>. [Pokušaj pristupa 14. 3. 2021.].
- [7] R. Lešić, Povrćarstvo, 2002..
- [8] »Water-saving irrigation technology in greenhouse,« [Mrežno]. Available: <http://www.enhuawei.com/index.php/article/detail/735>. [Pokušaj pristupa 14. 3. 2021.].
- [9] C.-u. D. I. o. P. S. i. t. Greenhouse. [Mrežno]. Available: <https://www.istockphoto.com/photo/close-up-drip-irrigation-of-pepper-seedlings-in-the-greenhouse-gm579747734-99704359>. [Pokušaj pristupa 14. 3. 2021.].
- [10] »Products - Agrivi 360,« [Mrežno]. Available: <https://www.agrivi.com/products#agrivi-360>. [Pokušaj pristupa 13. 6. 2021.].
- [11] »Using AGRIVI,« [Mrežno]. Available: <https://agrivi.zendesk.com/hc/en-us/categories/115000464773-Using-AGRIVI>. [Pokušaj pristupa 13. 6. 2021.].
- [12] »Products - Agrivi IOT,« [Mrežno]. Available: <https://www.agrivi.com/products#agrivi-360>. [Pokušaj pristupa 13. 6. 2021.].

- [13] »WaterBit - Products,« [Mrežno]. Available: <https://www.waterbit.app/products/>. [Pokušaj pristupa 13. 6. 2021.].
- [14] »WaterBit Dashboard,« [Mrežno]. Available: https://www.waterbit.com/wp-content/uploads/2019/01/WaterBit_DS_Dashboard.pdf. [Pokušaj pristupa 14. 6. 2021.].
- [15] »Smart watering technology - SupPlant,« [Mrežno]. Available: <https://supplant.me/>. [Pokušaj pristupa 13. 6. 2021.].
- [16] J. Kasać, Neizrazita logika, Materijali za predavanja.
- [17] J. Sanchez i M. P. Canton, Microcontroller Programmin, 2007..
- [18] [Mrežno]. Available: <https://en.wikipedia.org/wiki/File:ATMEGA328P-PU.jpg>. [Pokušaj pristupa 15. 3. 2021.].
- [19] [Mrežno]. Available: <https://esp8266-shop.com/product/nodemcu-esp8266-esp-12e/>. [Pokušaj pristupa 15. 3. 2021.].
- [20] »ESP-12E WiFi Module Datasheet,« [Mrežno]. Available: https://components101.com/asset/sites/default/files/component_datasheet/ESP12E%20Datasheet.pdf.
- [21] [Mrežno]. Available: <https://diyprojects.io/deal-esp32-generic-development-board/#.YE7FOTqSnIU>. [Pokušaj pristupa 15. 3. 2021.].
- [22] »ESP32-WROOM-32 Datasheet,« [Mrežno]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.
- [23] J. Fraden, Handbook of Modern Sensors Physics, Designs, and Applications, 2004..
- [24] R. M. White, »A Sensor Classification Scheme,« 1987..
- [25] [Mrežno]. Available: <https://nettigo.eu/products/dht11-sensor-module>. [Pokušaj pristupa 15. 3. 2021.].
- [26] »DHTxx family Humidity Sensors,« [Mrežno]. Available: <https://wiki.metropolia.fi/display/sensor/Capacitive+Humidity+Sensors>. [Pokušaj pristupa 15. 3. 2021.].

- [27] »DHT11 Product Manual,« [Mrežno]. Available: https://components101.com/asset/sites/default/files/component_datasheet/DHT11-Temperature-Sensor.pdf.
- [28] H. Janocha, *Actuators: Basics and Applications*, 2004..
- [29] [Mrežno]. Available: <https://www.indiamart.com/proddetail/4-channel-5v-relay-board-19179450812.html>. [Pokušaj pristupa 15. 3. 2021.].
- [30] »5V 5-Pin Relay Datasheet,« [Mrežno]. Available: <https://components101.com/switches/5v-relay-pinout-working-datasheet>. [Pokušaj pristupa 15. 3. 2021.].
- [31] [Mrežno]. Available: http://www.makeproto.com/projects/fuzzy/matlab_arduino_FIST/index.php.
- [32] W. Dongri, »Twelve Considerations in Choosing between,« *IEEE World Congress on Computational Intelligence*, 2012.