Autonomni pametni staklenik

Šuran, Matija

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Istrian University of applied sciences / Istarsko veleučilište - Universita Istriana di scienze applicate**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:212:277495

Istarsko veleučilište – Università Istriana di scienze applicate

Matija Šuran

AUTONOMNI PAMETNI STAKLENIK

Završni rad

Pula, 2023.

Istarsko veleučilište – Università Istriana di scienze applicate

Matija Šuran

AUTONOMNI PAMETNI STAKLENIK

Završni rad

JMBAG: 0233008194

Studijski smjer: Preddiplomski stručni studij Mehatronike

Predmet: Projektiranje ugrađenih računalnih sustava

Mentor: Marko Turk, predavač

Pula, Rujan 2023.

IZJAVA

o korištenju autorskog djela

Ja, MATIJA ŠURAN dajem odobrenje Istarskom veleučilištu – Università Istriana di scienze applicate, kao nositelju prava iskorištavanja, da moj specijalistički završni rad pod nazivom

AUTONOMI PAMETNI STAKLENIK

koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, rujan 2023 godine

Potpis

Matija Šuran

Sažetak

Ovaj je završni rad inspiriran mogućnošću implementacije automatizacije u staklenicima kako bi oni bili autonomniji. U ovom radu bit će jednim djelom opisani staklenici i zašto su nam potrebni, u drugom dijelu bit će opisano zašto je automatizacija takvih sustava pogodna za čovjeka i biljke. U radu ćemo se dotaknuti svih bitnih elemenata za dobivanje maksimalnog uroda biljaka. Ti su elementi zalijevanje i prehrana biljaka, održavanje optimalne temperature i razine vlage unutar staklenika, praćenje potrošnje vode te upravljanje količinom svijetlosti. Rad se bazira upravljanjem Arduino mikrokontrolera uz sve potrebne senzore i upravljačke uređaje. Priložene će biti sheme samog sustava, i objašnjeni svi sustavi zasebno uz to pripadajući programski kod napravljen u Arduino IDE software-u.

Ključne riječi : automatizacija, staklenik, upravljanje, programski kod, arduino

Abstract

This work was inspired by the possibility of implementing automation in greenhouses to make them more autonomous. In this paper, greenhouses will be described in one part and why we need them. In the second part it will be described why the automation of such systems is convenient for humans and plants. In the paper, all the essential elements for obtaining the maximum yield of plants will be stated. These elements are watering and feeding of the plants, maintaining the optimal temperature and humidity level inside the greenhouse while monitoring water consumption, and managing the amount of light. The work is based on the control of the Arduino microcontroller with all the necessary sensors and control devices. Schemes of the system itself are attached, and all systems will be explained separately, along with the associated programming code created in the Arduino IDE software.

Keywords: automation, greenhouse, control, programming code, arduino

Sadržaj

1. UVOD

Sve većom potrebom proizvodnje prehrambenih proizvoda i u današnje vrijeme dostupnom tehnologijom automatizacija staklenika je privlačna i relativno lako izvediva ideja. Ljudi koriste staklenike već stoljećima za uzgoj raznih kultura povrća voća i ostalih biljaka. Nikad nije bilo jednostavnije optimizirati i pojednostavniti uzgoj bilja nego u današnje vrijeme.

Svaki dan se ljudi bore sa vremenskim neprilikama a tehnologija današnjice ako se pametno upotrijebi može značajno doprinijeti uštedi vremena i osigurati kvalitetnije proizvode u staklenicima uz senzorsko praćenje raznih faktora koji su bitni za rast kultura unutar staklenika. Automatizacija je na neki način počela biti standard u staklenicima ali naravno uvijek ima mjesta poboljšanju jer još uvijek ta automatizacija na neki način nije autonomna te i dalje postoji faktor čovjeka koji nadzire cijeli taj „automatizirani“ proces. Sljedeći korak je da sva ta automatizacija bude autonomna tj. čim više smanji ljudski faktor tijekom procesa rasta kulture koja se uzgaja u stakleniku.

U ovom radu objasniti će se koji su izazovi, problemi postoje kako u teoretskom dijelu tako i u praktičnoj primjeni autonomnih pametnih staklenika te ponuditi rješenje vezano za primjenu takvih autonomnih sustava staklenika. Nakon kratkog pregleda literature i postojećih implementacija pametnih staklenika, definiran je problem i metodološki je prikazan plan izrade rješenja kako bi se izradila i fizička implementacija autonomnog pametnog staklenika. Od projektiranja do odabira potrebnog hardvera i komponenti do razvoja programskog koda za postizanje željenih funkcionalnosti predloženog sustava.

2. STAKLENICI I POTREBA ZA AUTOMATIZACIJOM

Staklenici (Slika 1) su zatvoreni prostori u kojima se uzgajaju razne hortikulture u kontroliranim i zaštićenim uvjetima. U prošlosti su se koristila stakla za obložiti staklenik, u današnje vrijeme se koriste specijalizirane folije te polikarbonat da bi se postigao staklenički efekt te da bi se nasad unutar staklenika zaštitio od vanjskih uvjeta.

Slika 1. Prikaz staklenika

Izvor: Staklenik, https://hr.wikipedia.org/wiki/Staklenik (pristupljeno 19.9.2023.)

Prve zapise o ideji da se hrana uzgaja u kontroliranim uvjetima imamo od prije 2000 godina za vrijeme vladavine cara Tiberiusa Rimskim carstvom. U njegovo je vrijeme počeo uzgoj krastavaca u kontroliranim, uvjetima zbog toga jer je car želio uvijek imat krastavce u svojoj prehrani. U to su vrijeme po danu iznosili biljke krastavaca van na sunce tokom dana, dok su po noći vraćali biljke unutar zatvorenih prostorija kako bi bile biljke u toplijim uvjetima (Janick i Paris, 2022).

Prvi „moderni“ staklenici se pojavljuju u 16 stoljeću, pojavom egzotičnih biljaka na ovim prostorima. Pošto su te biljke zahtijevale drugačije uvjete za rast, njih su držali u takozvanim botaničkim vrtovima. Veliki je problem bio zadržati toplinu na tim prvim vrtovima, ali daljnjim razvojem stakla i ostalih materijala staklenici su postali neophodan alat za uzgoj bilja diljem Europe i svijeta tee su već u 17. stoljeću u Nizozemskoj i Engleskoj nastali staklenici kakve na neki način poznamo dan danas (Muijzenberg, 1980).

Povećanjem svjetskog stanovništva te sve većom potrebom za hranom a istodobno i nepredvidivim klimatskim promjenama staklenici su postali sve uobičajeniji sustav za uzgoj

raznih kultura voća povrća i biljaka. Automatizacija takvih staklenika je ključna za održivu budućnost opskrbnog hranidbenog lanca. Analiza učinaka različitih klimatskih faktora na ponašanje biljaka postaje gotovo nemoguća zbog promjenjivog i međusobno povezanog prirodnog okoliša. Najprikladnije rješenje za ovu situaciju je premještaj istraživanja u prostor za uzgoj biljaka s djelomičnom ili potpunom kontrolom okoliša, a takav dobro održavan prostor naziva se kontroliranim okolišnim staklenikom ili automatiziranim staklenikom (Tiwari, 2003). U današnje vrijeme se velikom brzinom počela razvijati industrija automatizacije radi velikih prednosti koje ona donosi (Allen, 2015).

2.1. Motivacija i svrha automatizacije staklenika

U drugom djelu dvadesetog stoljeća, distribucija i opskrba hrane su se naglo razvile te su mogle pratiti rast stanovništva te su mnoge regije u svijetu profitirale u tom razvitku, međutim, kako smo ušli u 21. stoljeće, situacija se počela mijenjati. Ubrzani rast populacije, ogromna potražnja za hranom, te novi pokazatelji koji ukazuju da trenutni način proizvodnje hrane utječe na klimatske promjene sa čime se pojavio novi rizik za poljoprivredu i prehrambeni sustav (Vermeulen S.J., 2012). Iz tog razloga automatizacija staklenika može promijenit način na koji se proizvodi hrana i smanjiti štetne emisije kako bi zadovoljili ideje većeg doprinosa hrane i smanjenja utjecaja na klimatske promjene. Zbog toga je bitno dignuti svijest poljoprivrednika te ih potaknuti da se moderniziraju kako bi bili uz korak budućim ograničenjima koji bi se mogla uvesti pretežno vezane uz proizvodnju hrane i njen utjecaj na okoliš.

Svrha automatizacije je da se proizvodnja optimizira, korištenjem moderne tehnologije i praćenjem proizvodnje, možemo u ovom slučaju prvenstveno uštediti na vodi, koja je osnovna namirnica biljkama. Navodnjavanje biljaka u staklenicima je jedna od najbitnijih stvari u proizvodnji, iako je često zanemareno i uzimano zdravo za gotovo. Često se odgovornost za zalijevanje prepušta manje iskusnim radnicima iako se pogreške u navodnjavanju često odražavaju na kvalitetu usjeva. Da bismo osigurali optimalne uvjete za rast biljaka, ključno je razumjeti faktore koji utječu na vlažnost tla (Newman, 2013). Praćenjem vlažnosti tla možemo odrediti koliko je biljci vode potrebno, tako da ima uvijek optimalnu količinu vode za rast. Na veće stakleničke sustave, zalijevanje biljaka koje traje tri sata ili četiri sata imamo veliku razliku u potrošnji vode, uz praćenje vlažnosti tla taj se proces može automatizirati te bi doprinijelo značajnoj uštedi vode.

Sljedeći korak je optimalna mikro klima u stakleniku. Naanašnji dan upravljanja mikroklimom na našim prostorima skoro pa i nema. Ručno održavanje idealnih okolišnih uvjeta unutar

staklenika je izuzetno zahtjevno i nepraktično. Samim time nedovoljno praćenje i održavanje rezultira manjom proizvodnjom usjeva, nižom kvalitetom i smanjenim prihodima. Za učinkovitu kontrolu, trenutno se koriste automatizirani sustavi kontrole poput mikroprocesora i računala kako bi se održavali okolišni uvjeti. Ovi automatizirani sustavi detektiraju i mjere okolišne parametre, uspoređuju ih s zadanim parametrima i kad je potrebno, aktiviraju odgovarajuće uređaje kako bi prilagodili uvjete da odgovaraju željenim parametrima (Radha i Igathinathane, 2007).

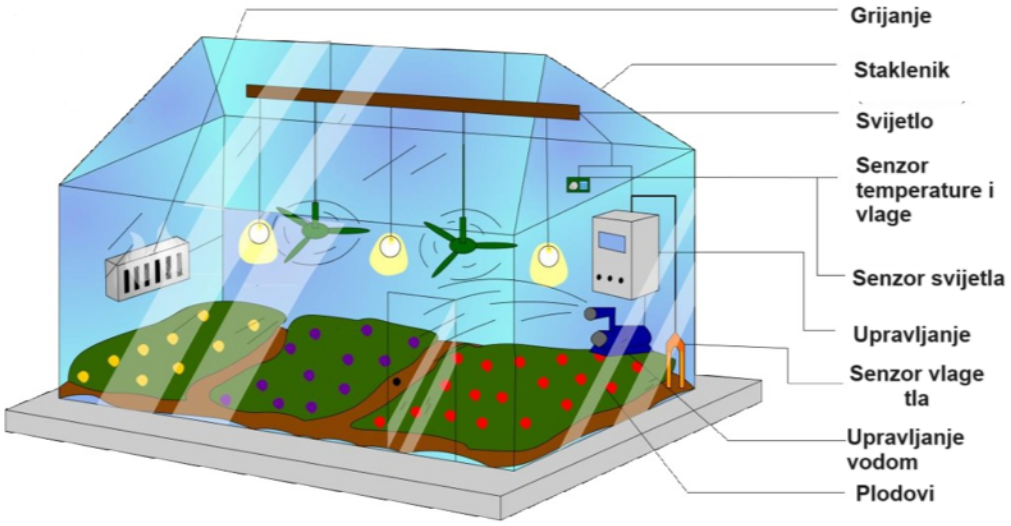
2.2. Prednosti Automatizacije Staklenika

Pri automatizaciji standardnog staklenika te pretvaranjem u automatizirani staklenik dobivamo pregršt prednosti od kojih je najznačajniji veći prinos proizvoda koji se uzgajaju u tom stakleniku te imamo smanjeni rizik pojave štetnika koji mogu uništiti kulturu. Uzgoj u zatvorenom prostoru osigurava da nepovoljni vremenski utjecaji ne ugrožavaju proizvodnju. Automatizacijom možemo pratiti kvalitetu tla te bolje iskoristiti sistem navodnjavanja da bi biljka imala optimalnu količinu hrane. Značajnija prednost je i smanjenje ljudske intervencije u cijeli proces zahvaljujući automatskim praćenjem i upravljanjem klimatskim faktorima unutar staklenika kako bi se optimizirali resursi potrebni za proizvodnju kulture unutar staklenika(Jermin Jeaunita, 2018).

2.3. Tehničke komponente automatizacije staklenika

Komponente koje su potrebne za automatizaciju staklenika uvelike ovise o površini nasada, o lokaciji staklenika te o kulturama koje se uzgajaju unutar staklenika. Naravno za veće staklenike sa više različitih kultura u nekim hladnijima regijama je potrebno daleko više komponenata te sve mora biti zaštićenije od vanjskih utjecaja u usporedbi sa nekim manjim staklenikom u regiji umjerene klime sa samo jednom kulturom.
Sam sustav kontrole treba se sastojati od senzora, aktuatora te upravljačke jedinice. Glavni izazov u prošlosti za automatizaciju je bio digitalni sustav za obradu svih podataka. U današnje vrijeme su sustavi upravljanja u realnom vremenu najznačajniji faktor ako imamo kompleksnu strukturu automatizacije. Za staklenike sa visokim stupnjem automatizacije su oni itekako potrebni, dok kod jednostavnijih sustava zbog ne tako naglih promjena parametara unutar staklenika mogu se koristiti i jednostavnije upravljačke jedinice. Napredni digitalni sustavi

omogućuju brzo izvršavanje zadanih pravila u vrlo kratkom vremenskom periodu. Zato je bitno razumijeti osnovne kontrolere koji se koriste za upravljanje staklenicima. Jedna od najboljih opcija je ugradnja kontrolera koji koristi RTOS (eng. Real-Time Operating System) (Pedro Ponce 2014).



Slika 2. Ilustracija tehničkih komponenti automatizacije staklenika

Izvor: Saha, 2017 https://www.semanticscholar.org/paper/Construction-and-Development-of-an-Automated-System-Saha-Jewel/4a04317a880ff332e18fa15bcf065727287786cb (pristupljeno 19.09.2023)

2.4. Utjecaj na poljoprivredu i održivost

Predviđanja Ujedinjenih Naroda govore o tome da bi do 2050. godine na kugli Zemaljskoj moglo biti 9 milijardi ljudi (Cohen, 2001), zbog toga se sve više počelo razmišljati o poljoprivredi u zatvorenim prostorima. Moderni staklenici rade kao jedan sustav stoga ih se naziva kontroliranom poljoprivrednom proizvodnjom u zatvorenom prostoru eng. Controlled Environment Agriculture (CEA) (Shamshiri, 2018). Plan je u budućnosti izgraditi što više takvih sustava kako bi se postigli optimalni uvjeti rasta za razne hortikulture koji bi smanjili troškove proizvodnje i povećali prinose. Prema planovima UN-a dvije trećine svjetskog stanovništva će 2050 godine živjeti u gradovima (Ritchie, 2018), tako da se već sada radi na planovima za proizvodnju hrane koji bi bili održivi u ruralnim sredinama sredinom ovog stoljeća. Zbog inovacija i razvoja u tehnologiji, posebno u granama senzorike i komunikacije biti će sve lakše prijeći iz klasičnog načina proizvodnje biljaka na novi sustav takozvanih tvornica biljaka i to sve u urbanim sredinama (Slika 3). Poljoprivreda u staklenicima doživljava veliku transformaciju zbog tehnološkog napretka, moderni će staklenici postati

visokotehnološke tvornice hrane, koje će sve više i više biti optimizirane s većom iskoristivošću prostora sa što manje otpada u smislu hrane (Shamshiri, 2018).

Slika 3. Prikaz jedne od tvornica biljaka u Parizu na vrhu zgrade

Izvor: https://www.bloomberg.com/news/features/2023-07-07/in-quest-for-food-security-cities-test-limits-of-urban-agriculture (pristupljeno 19.09.2023)

3. Metode i materijali

Ovim radom želi se riješiti problem kontroliranja uvjeta uzgoja jedne kulture. Cilj je izrada sustava kontrole uvjete unutar staklenika. Samim time, cilj je Arduino kontrolerom napraviti jednostavnu autonomnu automatizaciju za upravljanje uzgoja rajčice u stakleniku.

Naš kontroler ima zadatak očitati varijable unutar staklenika, odnosno očitati vlagu u zemlji, vlagu i temperaturu zraka unutar staklenika te očitati razine sunčeve svijetlosti te na osnovu toga donijeti odluku što je kulturi unutar nasada potrebno u određeno vrijeme tokom dana. Sustav će izmjerene podatke o temperaturi, vlazi, vlazi tla i intenzitetu svjetlosti prikazivati na LED ekranu kako bi korisnik mogao vidjeti stanje mikroklime i ostalih parametara unutar staklenika. Način rada kontrolera će biti predefiniran prema kulturi rajčice koja je zasađena unutar samog staklenika.

Podatci o željenim vrijednostima koje želimo postići su da moramo održavati temperaturu unutar staklenika između 24°C i 27°C preko dana, odnosno 14-17°C preko noći. Relativna vlaga zemlje trebala bi biti u rasponu između 60-80%, svjetlosti minimalno 12h unutar

staklenika (Nurhasanah, 2021). U tablici 1 prikazani su problemi koji nastaju ukoliko mikroklima kod uzgoja rajčice nije unutar granica.

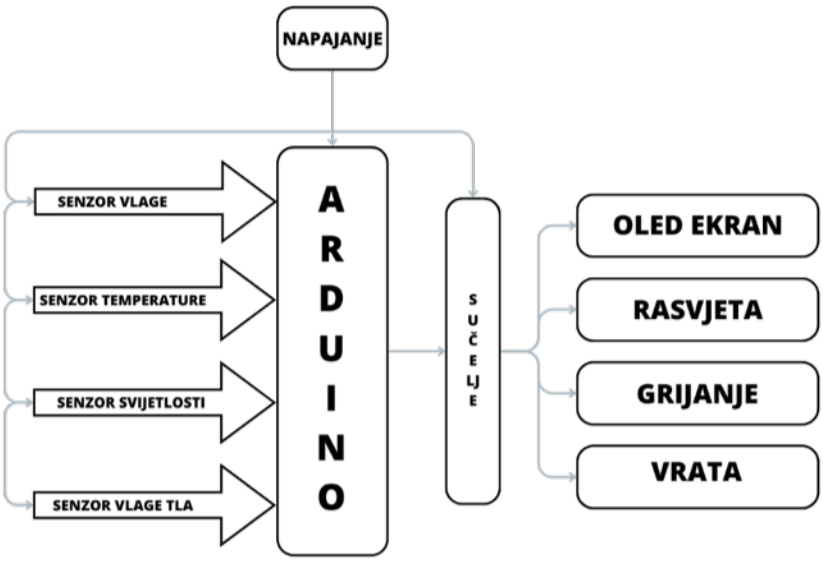
Problemi u nasadima uzrokovani klimatskim promjenama	
Temperatura	Problemi u nasadima uzrokovani klimatskim promjenama
Preko 35°C	Pelud ne sazrijeva, tako da nema oplodnje, cvijetovi otpadaju, plod ne raste dobro
Ispod 12°C	Utječe na rast biljke
Preko 35°C/Ispod 12°C	Smanjena oplodnja te nam plodovi dobivaju žutu nijansu
Relativna vlažnost	
Preko 80%	Gruda se pelud, povećava se mogućnost bolesti, i imamo pukotine na plodovima
Ispod 60%	Dehidracija peludi sprječava oplođivanje
Svjetlo	
Premalo svijetla	Slab rast biljke, usporeno cvjetanje, oprašivanje i zrenje

Tablica 1. Problemi u nasadima rajčice uzrokovani klimatskim promjenama
Izvor: Ponce P., 2014. Greenhouse Design and Control

Za izradu autonomne automatizacije staklenika odabrana je Arduino UNO razvojna pločica zbog dostupnosti samog kontrolera, zbog njegove cijene i mogućnosti koju nudi za tu vrijednost. Pod time se misli na broj ulaznih i izlaznih jedinica te procesorsku moć koje su nam dovoljne za napraviti upravljanje staklenikom u kojem raste samo jedna kultura. Uz sam Arduino UNO koristit će se tržišno lako dobavljivi senzori i releji, servo motori i LED ekran koji su svi kompatibilni s Arduino Uno mikro kontrolerom.

Za programiranje samog Arduina koristiti će se Arduino IDE aplikacija, te Fritzing software za prikaz sheme spajanja ulaznih i izlaznih uređaja na Arduino UNO.

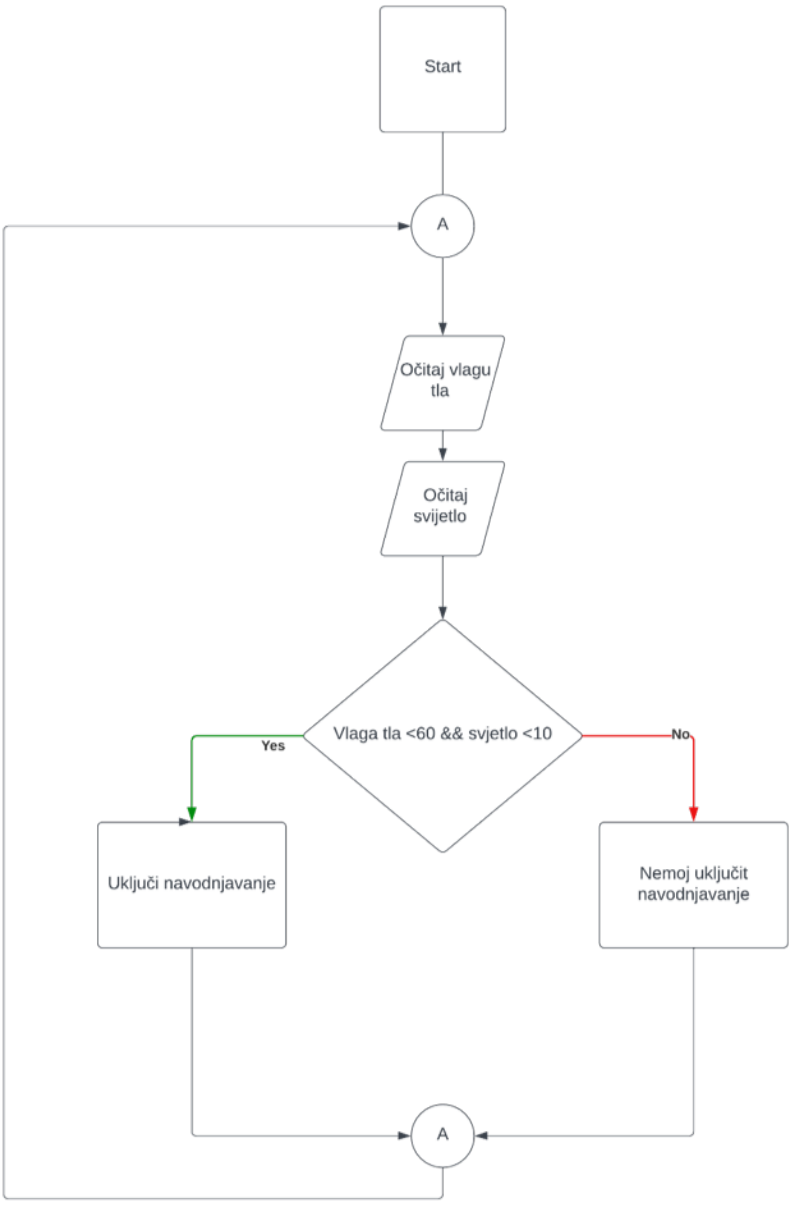
Kontroler te LCD ekran za prikaz trenutnih vrijednosti su zamišljeni da budu na pristupačnom mjestu na ulazu u staklenik zaklonjeni od vanjskih vremenskih uvjeta, senzor vlage tla bi spojili dvožilnim oklopljenim vodičem na početak nasada neposredno uz cijev za navodnjavanje, elektro ventil koji regulira navodnjavanje bio bi spojen na dovod vodovodne cijevi u stakleniku. Senzor svjetla bio bi montiran unutar staklenika ispod razine krova kako bi dobivali realna očitanja o dnevnoj svjetlosti. Servomotori bi upravljali vratima na početku i kraju staklenika, te bi se jedan relej spojio na Arduino kako bi upravljao grijalicom koja bi bila unutar prostora. Podaci bi se u realnom vremenu spremali u Arduino UNO te bi u realnom vremenu bili prikazivani na LED ekranu.



Slika 4. Dijagram konfiguracije sustava automatizacije staklenika

Izvor: Izradio autor

U dijagramu toka niže u Slici 5 prikazana je za primjer logika koja je potrebna za navodnjavanje.



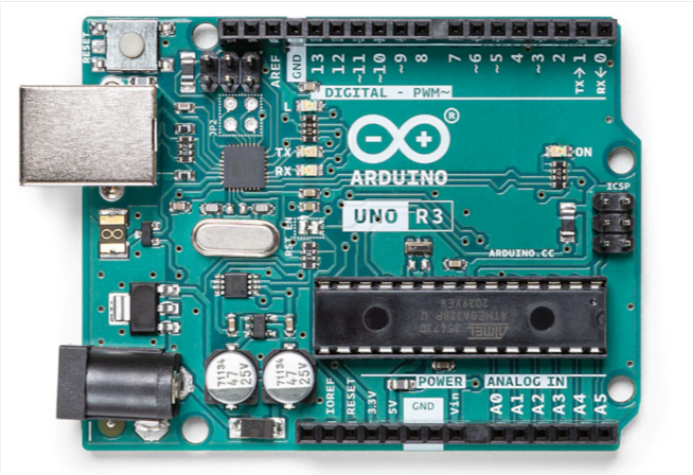
Slika 5. Dijagram toka upravljanja navodnjavanjem

Izvor: Izradio autor.

4. POTREBAN HARDWARE I KOMPONENTE

4.1 Arduino UNO R3

Arduino UNO je mikrokontrolerska ploča otvorenog koda koja se temelji na Mikročipu ATmega328P. Pločicu je razvio Arduino i prva verzija je izašla 2010 godine. Pločica ima 14 digitalnih ulazno izlaznih (u daljnjem tekstu I/O) pinova, od kojih je 6 sposobno za pulsno širinsku modulaciju (u daljnjem tekstu PWM modulacija) te ima i 6 analognih ulaznih pinova. Programiranje se vrši preko Arduino IDE softwarea te se na PC računalo spaja putem USB kabela tipa B.



Slika 6. Arduino UNO R3

Izvor: https://store.arduino.cc/products/arduino-uno-rev3 (pristupljeno 19.09.2023.)

Specifikacije Arduina UNO R3	
Mikrokontroler	ATmega328P
Radni Napon	5V
Ulazni Napon	7-12V
Ulazni napon granice	6-20V
Digitalni I/O pinovi	14 (od kojih 6 imaju mogućnost PWM izlaza)
PWM Digitalni I/O pinovi	6
Analogni ulazni pinovi	6
Dc struja za 3.3 V Pin	20 mA
Dc struja za 3.3 V Pin	50 mA
Flash memorija	32 KB (ATmega328P) od kojih se 0.5 KB koristi za bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Brzina procesora	16 MHz
LED ugrađeni	13
Duljina	68.6 mm
Širina	53.4 mm
Težina	25 g

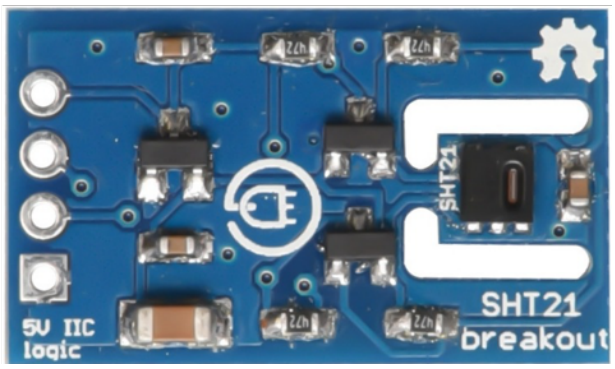
Tablica 2. Specifikacije Arduino UNO mikrokontrolera

Izvor: https://store.arduino.cc/products/arduino-uno-rev3 (pristupljeno 19.09.2023)

4.2 Senzor temperature i vlažnosti zraka SHT 20

Senzor temperature i vlažnosti zraka prikazan je na Slici 7 i je jedan od senzora koja nam treba u stakleniku. Za ovaj projekt odabran je SHT 20 senzor zbog njegovih svojstava koje udovoljavaju našim potrebama.

Ovaj je senzor temperature i relativne vlažnosti zraka vrlo popularan zbog svoje visoke preciznosti i pouzdanosti. Sam je senzor proizveden i kalibriran u Švicarskoj a pločica na kojoj je senzor smješten je hrvatski proizvod.



Slika 7. Senzor temperature i vlažnosti zraka

Izvor: https://soldered.com/learn/hum-sht21-sht20-temperature-and-humidity-sensor/ (pristupljeno 19.09.2023)

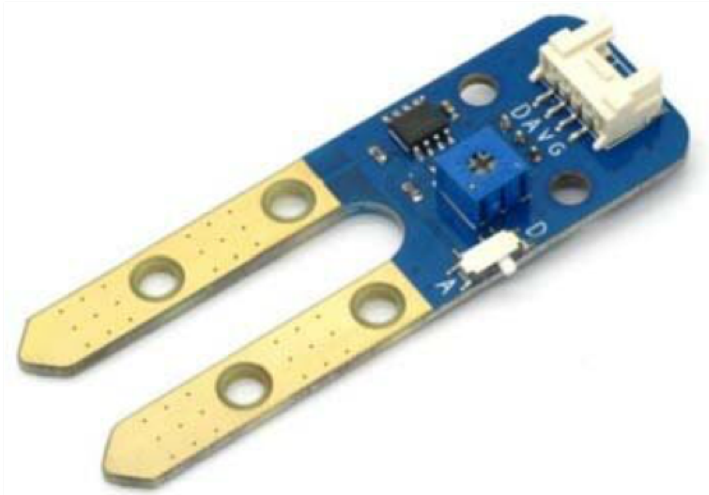
Specifikacije SHT20 senzor temperature i vlage	
Napon napajanja	5V
Relativna vlaga, raspon mjerenja	0-100%
Relativna vlaga, preciznost	0.04% (12 bit)
Temperatura, raspon mjerenja	`-40°C - 125°C
Temperatura, preciznost	0,01°C (14 bit)
Komunikacijsko šučelje	I2C
Dimenzije	25 mm x 15 mm

Tablica 3. Specifikacije SHT20 senzora

Izvor: https://soldered.com/learn/hum-sht21-sht20-temperature-and-humidity-sensor/ (pristupljeno 19.09.2023)

4.3. Senzor vlažnosti tla

Senzorom vlažnosti tla se koristimo za dobivanje točnih informacija o postotku vlage unutar zemlje u kojoj je posađena biljka. Ovaj senzor koristi dvije elektrode kroz koje prolazi struja. Očitavanjem električnog otpora dobivamo informaciju zasićenosti zemlje vodom. U slučaju ako je otpor manji znamo da je zemlja zasićenija vodom a ako je otpor veći znači da je zasićenost tla vodom manja.



Slika 8 Senzor vlažnosti tla

Izvor: https://www.robotshop.com/products/electronic-brick-soil-moisture-sensor (pristupljeno 19.09.2023)

Specifikacije senzora vlažnosti tla	
Napon napajanja	3.3V ili 5V
Izlazni naponski signal	0~4.2V
Struja	35mA

Tablica 4. Specifikacija senzora vlažnosti tla

Izvor: https://www.robotshop.com/products/electronic-brick-soil-moisture-sensor (pristupljeno 19.09.2023)

4.4. Senzor svjetlosti

Senzorom svjetlosti može se pratiti intenzitet i vrijeme trajanja dnevnog svjetla. Ovaj senzor odabran je radi njegove jednostavnosti, dostupnosti i preciznosti. Senzor svjetlosti sadrži otpornik ovisan o svjetlu (eng. LDR)[1]. princip rada mu se bazira na promjeni otpora ovisno o količini svjetlosti koju senzor dobiva. Što više svjetla senzor prima to će imati manji otpor, odnosno što manje svjetla dobiva to je otpor veći. Ovaj senzor koji koristimo ima digitalni i analogni izlaz zbog čega se može koristiti u više primjena i može ga se povezati na više načina sa mikro kontrolerom. Na samoj pločici senzora postoji mali potenciometar čijim podešavanjem možemo podesiti prag osjetljivosti digitalnog signala.

Slika 9. Jednostavni senzor svjetlosti

Izvor: https://soldered.com/hr/proizvod/jednostavni-senzor-svjetlosti/ (pristupljeno 19.09.2023)

[1] Soldered.com jednostavni senzor svijetlosti, n.d, https://soldered.com/hr/proizvod/jednostavni-senzor-svjetlosti/

Specifikacije jednostavnog senzora svijetlosti	
Razina logičkog napona	Isti kao i VCC (5V ako koristite 5V za izvor
Radni napon	3.3-5V
Komparator na ploči	LM393
Dimenzije	22mm x 22mm

Tablica 5. Specifikacije jednostavnog senzora svijetlosti

Izvor: https://soldered.com/hr/proizvod/jednostavni-senzor-svjetlosti/ (pristupljeno 19.09.2023)

4.5. Servomotor Micro Tower Pro SG90

Servo motor će u radu služiti kao motor koji će otvarati i zatvarati vrata staklenika. Ovom vrstom motora možemo precizno kontrolirati kut poluge u odnosu na koju je motor postavljen. Ovaj je servo motor primjeren za demonstraciju na koji način bi se vrata staklenika otvarala i zatvarala.

Slika 10. Servo motor TowerPro SG90

Izvor: https://soldered.com/hr/proizvod/jednostavni-senzor-svjetlosti/ (pristupljeno 19.09.2023)

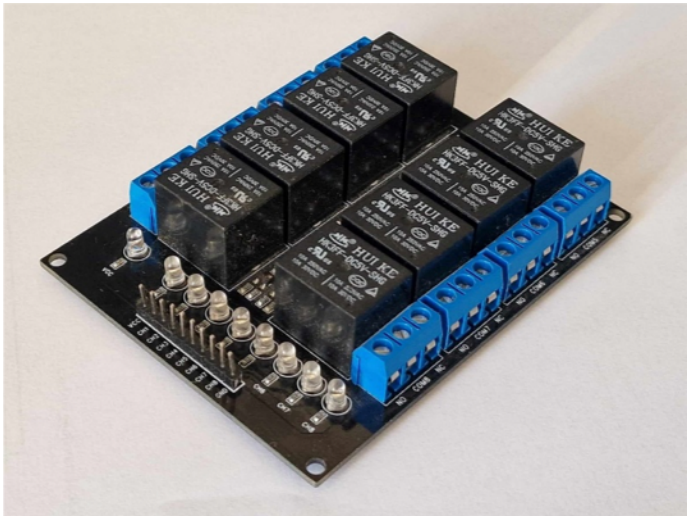
Specifikacije servomotora TowerPro SG90	
Kontrola	Analogna uz Servo library
Radni napon	5V
Brzina	60° za 0.1s
Težina	9g
Dimenzije	23mm x 12mm x 29mm

Tablica 6. Specifikacija Servo motora TowerPro SG90

Izvor: https://soldered.com/hr/proizvod/jednostavni-senzor-svjetlosti/ pristupljeno 19.09.2023)

4.6. Relej

Releji su jednostavni elektromehanički prekidači a način na koji funkcioniraju je taj da protokom struje kroz elektromagnet uključujemo ili isključujemo drugi strujni krug. Pogodni su zato jer malim strujama upravljanja možemo upravljati uređajima kojima je potrebna veća količina struje. U ovom slučaju koristi se pločicu razvijenu za arduino projekte koja ima na sebi 8 releja a za ovaj su projekt dostatna 3 što nam dopušta da ovim načinom postoji mogućnost nadogradnje sustava dodavanjem dodatnih uređaja potrebnih za održavanje mikroklime staklenika.



Slika 11. Relejni modul sa 8 releja

(Izvor autor)

Specifikacije relejnog modula	
Ulazni napon pobude releja	5V
Maksimalni DC napon i struja	30V, 10A
Maksimalni AC napon i struja	250V, 10A
Broj releja	8
Dimenzije	72mm x 91 mm x 20 mm

Tablica 7. Specifikacije relejnog modula

Izvor: Autor

4.7. Solenoidni ventil

Solenoidni ventili su elektromehanički upravljivi ventili. Princip rada mu je taj da protjecanjem struje kroz magnet stvaramo magnetsko polje koje utječe na jezgru te ju pomiče po jednoj osi. Pomicanjem jezgre u biti zatvaramo i otvaramo propusnost nekog fluida unutar ventila. Važno je napomenuti da ukoliko nema protjecanja struje na jezgru tada imamo utjecaj opruge koja

vraća jezgru u početni položaj. U našem slučaju ventil je u zatvorenom stanju ako nema pobude na elektromagnet.

Slika 12. Solenoidni ventil za vodu

Izvor: https://devito-promet.hr/proizvod/elektroventil-jednostruki-180o/ (pristupljeno 19.09.2023)

4.8. OLED ekran

OLED (eng. Organic Light Emiting Diodes) ekranom može se u realnom vremenu pratiti stanje sustava te podatke vezane za mikroklimu unutar staklenika. U ovom projektu koristi se maleni OLED od 0,91 inča dijagonale i rezolucije 128x32 piksela. Prednost OLED ekrana nad klasičnim LCD-om je ta što ima veći kontrast, manju potrošnju i manjih je dimenzija kada gledamo dubinu naspram LCD ekrana.

Slika 13. OLED 0.91

Izvor: https://www.smart-prototyping.com/0.91-inch-OLED-display pristupljeno 19.09.2023)

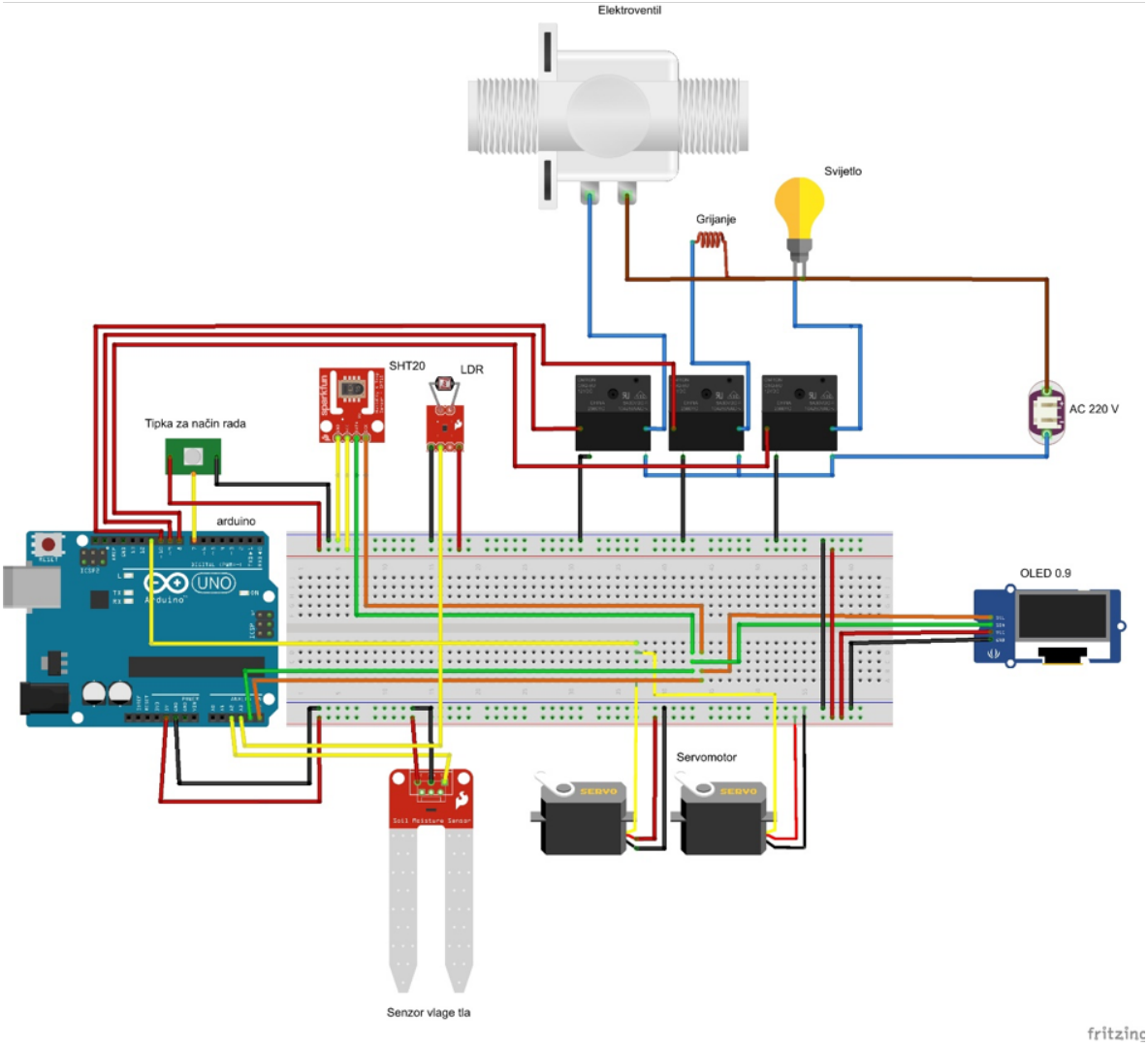
Specifikacije OLED ekrana	
Veličina ekrana	0.91 inča
Dimezije sa pločicom	12 x 38 mm
Boja slova	Bijela
Driver IC	SSD1306

Tablica 8. Specifikacije OLED ekrana

Izvor: https://www.smart-prototyping.com/0.91-inch-OLED-display (pristupljeno 19.09.2023)

4.8. Shematski prikaz sklopa

Shema spajanja izrađena je u Fritzing aplikaciji i prikazana je slikom 14.



Slika 14. Shema spajanja sklopa

Izvor: Izradio Autor

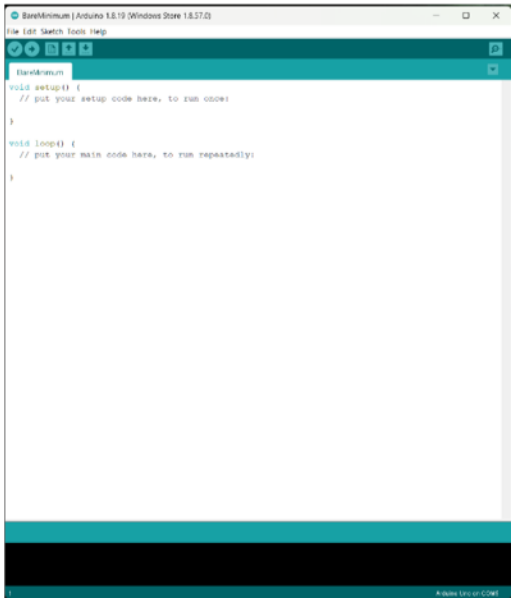
5. PROGRAMSKO RJEŠENJE

5.1. Korišteni softverski paketi

Za ovaj rad koristilo se nekoliko softverskih rješenja kako bi sklop postao funkcionalan. Najosnovniji softver je naravno Arduino IDE u kojem je cijeli kod napisan a korišten je i Fritzing kako bi se izradila shema gotovog sklopa te CANVU za dodatne slikovne prikaze.

5.1.1. Arduino IDE

Arduino IDE u biti sadrži uređivač teksta u kojemu se piše programski kod koji se upotrebljava u arduino mikro kontrolerima a osim toga sadrži područje za poruke te alatnu traku s gumbima za osnovne funkcije te niz ostalih izbornika.



Slika 15. Arduino IDE softver

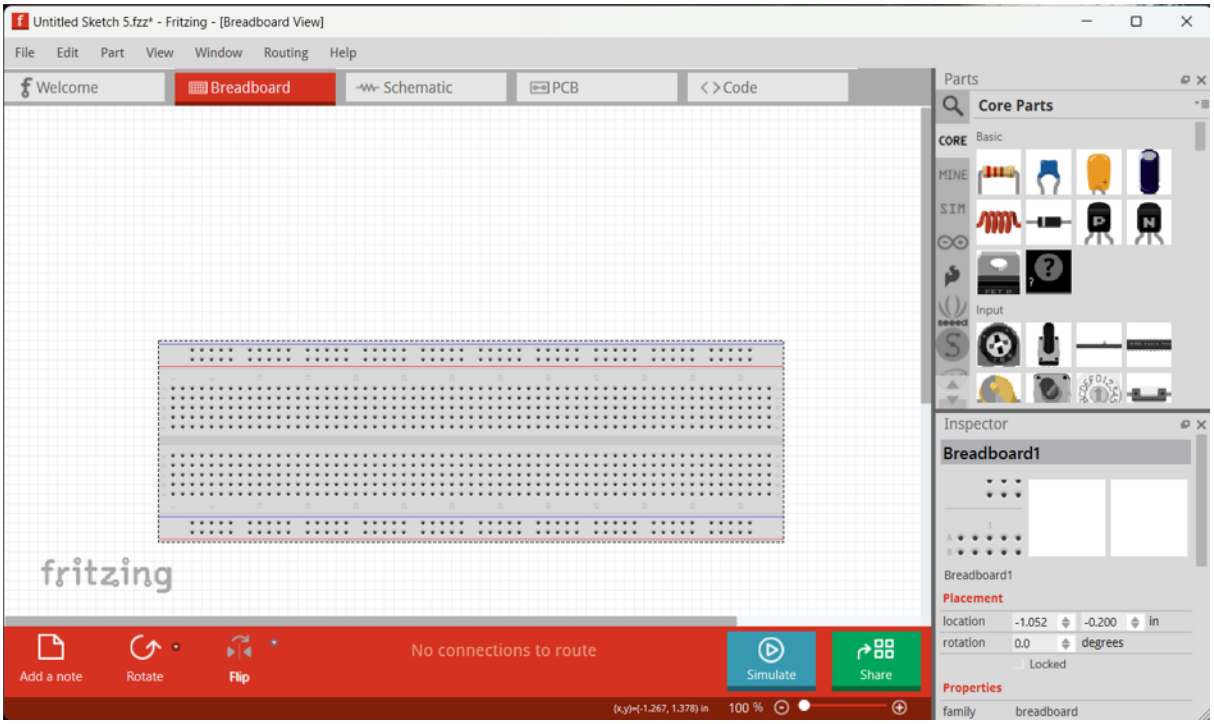
Izvor: autor

Programi izrađeni uz pomoć Arduino IDE se nazivaju skice. Skice su napisane u uređivaču teksta te se spremaju s ekstenzijom datoteke .ino. Na dnu prozora imamo područje u kojemu se vide poruke ukoliko ima nekih problema prilikom spremanja i izvoza datoteke u Arduino mikrokontrolera.

5.2.2. Fritzing

Software otvorenog koda koji je osmišljen kako bi pružio jednostavno projektiranje i dizajniranje elektroničkih sklopova. Pretežno se koristi za izradu električnih shema za pripremu

tiskanih pločica. Fritzing ima svoju biblioteku proizvoda a na Internetu ima mnogo korisnika koji baš za Fritzing rade dodatne pakete sklopova kako bi se proširila biblioteka samog softvera.



Slika 16. Fritzing software

Izvor: autor

5.2.3. Programski kod

Programski kod je nastao prema zahtjevima projekta i potpuna verzija za implementaciju priložena je u dodatku ovog rada. Na početku rada definiralo se da će ovaj sklop služiti za održavanje jedne kulture unutar staklenika a za tu kulturu odabrana je rajčica. Prema uzgoju rajčice predefinirani su i parametri unutar programskog koda. Rajčica je kultura koja mora imati jako puno svjetla tako da je definirano da u slučaju da ima manje od 12 sati svjetla dnevno, kontroler upali rasvjetu kako bi se zadovoljio taj uvjet. Biljke imaju dnevni i noćni režim rada tako da se to uzima u obzir.

Vezano uz temperaturu uzgoja, optimalna temperatura za rajčice je od 23°C do 27°C po danu odnosno 13°C do 15°C po noći. U slučaju da temperatura padne ispod zadanih vrijednosti pali se grijanje staklenika. Vlažnost tla je bitna jer rajčice imaju veliku potrebu za vodom tako da vlažnost treba održavati od 75 do 90% s tim da je poželjno zalijevati biljke pretežno noću. U kodu je definirano da kada vlažnost zemlje postigne 90%, sustav stane sa zalijevanjem. Ako temperatura u dnevnom režimu prijeđe 27°C otvaraju se vrata i to na način da se za svaki °C servomotor pomakne za 30°. To znači da ako temperatura dođe do 30°C vrata će biti maksimalno otvorena.

5.2.4. Postupak stvaranja koda za automatizaciju staklenika

Ovo poglavlje sadrži detaljni opis svih dijelova koda koji su napravljeni prema definiranim parametrima u prethodnom poglavlju.

Prije svega potrebno je definirati programske datoteke (eng. Libraries) koje nam služe za ispravan rad programa. Biblioteke su unaprijed napisane kako bi pojednostavnile razvojni proces arduino programiranja.

U ovom programu koristilo se 6 biblioteka.

- Wire.h biblioteka omogućava komunikaciju preko I2C protokola, te ju koristimo za komunikaciju sa OLED ekranom i SHT20 senzorom temperature i vlage.
- DFRobot_SHT20.h je dodana za podršku SHT20 senzora.
- Adafruit_SSD1306.h omogućava komunikaciju s led ekranima koji koriste SSD1306 kontroler kao slučaj s led ekranom korištenim u ovom radu.
- Adafruit_GFX.h je dio Adafruit GFX knjižnice te omogućuje ispis grafičkih elemenata na OLED ekranu.
- Fonts/FreeSans9pt7b.h je biblioteka koja učitava font koji ćemo naknadno koristiti na ekranu
- Servo.h biblioteka omogućuje kontrolu servomotora koji se koristi za otvaranje i zatvaranje vrata staklenika

```
1. #include <Wire.h>
2. #include "DFRobot_SHT20.h"
3. #include <Adafruit_GFX.h>
4. #include <Adafruit_SSD1306.h>
5. #include <Fonts/FreeSans9pt7b.h>
6. #include <Servo.h>
```

U linijama koda od 7 do 8 definiramo veličinu ekrana u pikselima, širinu od 128 piksela i visinu od 32 piksela, dok u liniji koda 9 vidimo makro definiciju kojom postavljamo I2C adresu koja će se koristiti za komunikaciju sa OLED ekranom

```
7. #define SCREEN_WIDTH 128
8. #define SCREEN_HEIGHT 32
9. #define SSD1306_I2C_ADDRESS 0x3C
```

U ovom se dijelu koda definiraju svi pinovi koje koristimo za određene komponente u projektu.

```
10. #define buttonPin 7   //pin na koji je spojen standby gumb
11. #define vlagaTlaPin A2 //pin za očitavanje sa senzora za vlagu tla
12. #define ldrPin A3    //pin za očitavanje senzora svijetla LDR
13. #define svjetloPin 8 // Pin na kojem je spojeno svjetlo preko releja
14. #define relejPin 9 // Pin za upravljanje relejem za navodnjavanje
```

```
15. #define grijanjePin 10//Pin koji upravlja relejem grijanja
16. #define servoPin 11 // Pin na koji su spojeni servo motori
```

Linija koda 17 inicijalizira objekt za upravljanje OLED ekranom pomoću Adafruit biblioteke. Parametri „SCREEN_WIDTH“ i „SCREEN_HEIGHT“ određuju širinu i visinu ekrana u pikselima, dok „&Wire“ označava komunikacijski kanal (I2C) koji će se koristiti za komunikaciju s ekranom. Ova linija u biti priprema ekran za prikazivanje informacija.
Linija 18 stvara objekt „sht20“ za komunikaciju sa samim senzorom. Kada je jednom inicijaliziran, možemo koristiti različite funkcije koje su dostupne za čitanje podataka sa senzora.
Linija 19 također stvara objekt „myservo“ kako bi mogli upravljati servomotorima.

```
17. Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire);
18. DFRobot_SHT20 sht20;
19. Servo myservo;
```

U liniji koda 20 definiramo početnu vrijednost gumba za standby tako da nakon priključenja arduina na napajanje, program se pokreće automatski a ako ga želimo zaustaviti moramo pritisnuti tipku.

```
20. int buttonState = 1;
```

Linija 21-23 su nam vezane uz prikaz podataka na OLED ekranu, najprije određujemo koja vrsta podataka se početno prikazuje na ekranu te nakon toga vidimo da postoje još tri podatka koja se izmjenjuju, linija 22 stavlja početno vrijeme na 0 kako bi se nakon toga mogli podaci prikazivati svake 2 sekunde kao što se vidi iz linije 23.

```
21. int trenutniPrikaz = 0; // Varijabla za praćenje trenutnog prikaza (0 -
    temperatura, 1 - vlažnost zraka, 2 - vlažnost tla, 3 - svjetlo)

22. unsigned long prethodnoVrijeme = 0; // Varijabla za pohranu vremena zadnjeg
    prikaza
23. const long interval = 2000; // Interval izmjene prikaza u milisekundama (2
    sekunde)
```

U liniji 24 definira se prag svjetla u lux-ima kako bi definirali noćni period

```
24. int pragSvjetla = 50;               // Prag svjetla koje smo definirali
    kao noćno vrijeme
```

Linija 25 nam definira trajanje svjetla koje nam je potrebno u stakleniku. U ovom slučaju stavljena je vrijednost 12 sati i to je definirano *unsigned long* varijablom koja nam označava da varijabla može sadržavat samo pozitive brojeve velike vrijednosti.

```
25. Unsigned long trajanjeSvjetla = 12 * 60 * 60 * 1000; // 12 sati u
    milisekundama
```

Linije 26 i 27 postavljaju pragove za vlažnost tla, koji se koriste za upravljanje sustavom navodnjavanja u našem programu.

```
26. int pragVlageTlaZaUkljucivanje = 60;  // Prag za uključivanje navodnjavanja
27. int pragVlageTlaZaIskljucivanje = 80; // Prag za isključivanje
    navodnjavanja
```

U sljedećoj liniji definira se varijablu kao tip podataka „bool“ koja može imati samo dvije vrijednosti, „"true" (istina) ili "false" (laž). Ova varijabla se koristi za praćenje stanja sustava navodnjavanja.

```
28. bool navodnjavanjeUkljuceno = false;
```

U sljedećoj liniji označavamo definiciju funkcije „setup()“. Funkcija se izvršava samo jednom i koristi se za postavljanje početnih uvjeta, inicijalizaciju pina i uređaja te izvođenje drugih radnji koje je potrebno obaviti samo jednom pri pokretanju uređaja.

```
29. void setup() {
```

U ovoj liniji inicijaliziramo serijsku komunikaciju sa brzinom od 9600 bita u sekundi. Ova nam funkcija služi za komunikaciju između Arduina i uređaja koji s njim komuniciraju putem serijskog porta.

```
30. Serial.begin(9600);
```

Linije od 31-34 postavljaju ulazne i izlazne režime rada za određene pinove na Arduinu. Kao što vidimo imamo dvije opcije INPUT koji definira ulazni pin, i OUTPUT koji definira izlazni pin.

```
31. pinMode(buttonPin, INPUT);     // buttonPin definiramo kao ulazni pin
32. pinMode(svjetloPin, OUTPUT); // svijetloPin definiramo kao izlazni pin
33. pinMode(relejPin, OUTPUT); // relejPin definiramo kao izlazni pin
34. pinMode(grijanjePin, OUTPUT); // grijanjePin definiramo kao izlazni pin
```

Ovdje je definirano da izlazi na pinove svietlo, navodnjavanje i grijane u početku izvođenja programa budu isključeni.

```
35. digitalWrite(svjetloPin, LOW);   // Inicijalno iskljuceno svjetlo
36. digitalWrite(relejPin, LOW);     // Inicijalno iskljucen relej za
    navodnjavanje
37. digitalWrite(grijanjePin, LOW);  //Inicijalno iskljucen relej za grijanje
```

Definicija da servomotor bude spojen na gore definirani pin.

```
38. Myservo.attach(servoPin)
```

Ovim linijama koda pokrećemo OLED ekran preko *begin* funkcije te da se poveže sa Arduinom preko određene I2C adrese. U slučaju da se ekran ne uspije povezati na Serial Monitoru unutar Arduino IDE dobivamo poruku da se ekran nije uspio spojiti.

```
39. //pokretanje OLED ekrana
40. if (!display.begin(SSD1306_I2C_ADDRESS)) {
41. Serial.println(F("SSD1306 allocation failed"));
42. }
```

Linije 43 do 50 se odnose na postavljanje i prikazivanje pozdravne poruke na OLED ekranu nakon što je ekran uspješno inicijaliziran.

```
43. display.clearDisplay();
44. display.setFont(&FreeSans9pt7b);
45. display.setTextSize(1);
46. display.setTextColor(SSD1306_WHITE);
47. display.setCursor(0, 20);
48. display.println("Dobrodosli!");
49. display.display();
50. delay(2000);
```

Linije 51 do 54 izvode inicijalizaciju senzora vlage i temperature tipa SHT20.

```
51. sht20.initSHT20();
52. delay(100);
53. sht20.checkSHT20();
54. }
```

Sljedeća linija označava početak loop funkcije takozvane petlje koja je glavna petlja u programu. Sve naredbe unutar ove funkcije se izvršavaju iznova u beskonačnoj petlji dokle god je arduino na napajanju.

```
55. void loop() {
```

Ovdje se provjerava stanje tipke za standby.

```
56. //Opcije rada standby ili radni automatski nacin
57. buttonState = digitalRead(buttonPin);
```

Pošto smo prije definirali da je stanje 1 ili HIGH program nastavlja izvršavanje koda nakon linije 69. Dodana je funkcija da na Serial monitoru vidimo promjenu stanja tipke te da nam ispiše u kojem je stanju izvođenje programa.

```
58. if (buttonState == LOW) {
59. Serial.println("Tipka pritisnuta! Pauza u izvodjenju programa...");
60. display.clearDisplay();
61. display.setCursor(0, 20);
62. display.setFont(&FreeSans9pt7b);
63. display.print("Standby...");
```

```
64. display.display();
65. while(digitalRead(buttonPin) == LOW) {
66. delay(100);
67. }
68. Serial.println("Izlaz iz pauze.");
69. } else {
```

Linija 70 definira trenutno vrijeme korištenjem funkcije *milis* kako bi program definirao početak izvođenja programa dok linija 71 računa trenutno vrijeme i prethodno vrijeme kako bi omogućio izmjenu prikaza podataka na OLED ekranu u točno definiranom vremenu.

```
70. // Provjera vremena za izmjenu prikaza
71. unsigned long trenutnoVrijeme = millis();
72. if (trenutnoVrijeme - prethodnoVrijeme >= interval) {
```

Ovdje je definirano da se brišu prijašnji podaci sa ekrana te postavljen početak gdje će se zapis početi ispisivati na ekranu te kojim fontom.

```
73. display.clearDisplay(); // Ažurirajte prikaz
74. display.setCursor(0, 20);
75. display.setFont(&FreeSans9pt7b);
```

Iz Senzora SHT20 očitavamo vrijednost temperature, pretvaramo *string* funkcijom u tekstualni oblik kako bi smanjili decimalu na 1 radi prikaza na OLED ekranu. To je napravljeno jer je ekran malih dimenzija pa da se dobije jasnije očitanje sa ekrana.

```
76. float temperatura = sht20.readTemperature();
77. String formatiranaTemperatura = String(temperatura, 1); // 1 znaèi jedna
    decimala
```

Isto tako je napravljeno i za očitanje vlažnosti zraka.

```
78. float vlaga = sht20.readHumidity();
79. String formatiranaVlaga = String(vlaga, 1);
```

Linija 80 iščitava analognu vrijednost dobivenu iz senzora vlage tla te u sljedećoj liniji pretvara očitane vrijednosti pomoću „map“ funkcije koje znamo da su u rasponu od 0 do 1023, u vrijednosti postotaka od 100 do 0. testiranjem utvrđeno da senzor ima problema oko krajnjih vrijednosti pa ih se na određeni način prilagođava kalibriranjem da pokazuje točna mjerenja.

```
80. int vlagaT = analogRead(vlagaTlaPin);
81. float VlagaTla = map(vlagaT, 190, 950, 100, 0);
82. String formatiranaVlagaTla = String(VlagaTla, 1);
```

Linija 83 sprema analognu vrijednost količine svjetla. U liniji 84 invertira se njegovo očitanje i onda se opet pomoću *map* funkcije određuje kako da analognu vrijednost prikažemo u luxima, uz pretvorbu podatka u tekstualni oblik bez decimala.

```
83. float svjetlo = analogRead(ldrPin);
84. svjetlo = 1023 - svjetlo;
85. float svjetloLux = map(svjetlo, 0, 1000, 1, 10000);
```

```
86. String formatiranoSvjetlo = String(svjetlo, 0);
```

U ovim linijama koda provjerava se koliko je vremena prošlo od kada postoji dnevna svjetlost a u slučaju da prag koji je definiran u luxima padne ispod dozvoljene vrijednosti u vremenskom periodu manjem od 12 sati, pali se svjetlo.

```
87. // Provjera svjetla i upravljanje svjetlom
88. if (svjetlo < pragSvjetla && trenutnoVrijeme - prethodnoVrijeme >=
    trajanjeSvjetla) {
89. digitalWrite(svjetloPin, HIGH); // Uključuje se  svjetlo
90. prethodnoVrijeme = trenutnoVrijeme; // Resetirajte vrijeme
91. } else if (svjetlo >= pragSvjetla) {
92. digitalWrite(svjetloPin, LOW); // Isključuje se svjetlo ako je svjetlina
    dovoljno visoka
93. }
```

Ovdje se definira da ako temperatura prijeđe 27 stupnjeva da se servomotor uključi i otvori vrata za 30 stupnjeva a ako temperatura raste onda za svaki stupanj temperature se otvara za još 30 stupnjeva, do maksimalnih 30 stupnjeva kad se otvori za 90 stupnjeva.

```
94. // Otvaranje i zatvaranje vrata staklenika preko servomotora
95. if (temperatura > 27) {
96. int kut = map(temperatura, 27, 30, 0, 90);
97. kut = constrain(kut, 0, 90);
98. myservo.write(kut);
99. } else {
100.myservo.write(0);
101.}
```

U linijama 103-109 upravlja se navodnjavanjem tako da se definira da se navodnjavanje pali samo ako je vlaga ispod 60 posto i ako je vani sumrak ili noć.

```
102.// Provjera vlage tla i upravljanje navodnjavanjem
103.if (VlagaTla < pragVlageTlaZaUkljucivanje && svjetlo < pragSvjetla &&
    !navodnjavanjeUkljuceno) {
104.digitalWrite(relejPin, HIGH); // Uključuje se relej za navodnjavanje
105.navodnjavanjeUkljuceno = true;
106.} else if (VlagaTla >= pragVlageTlaZaIskljucivanje || svjetlo >=
    pragSvjetla) {
107.digitalWrite(relejPin, LOW); // Isključuje se relej za navodnjavanje
108.navodnjavanjeUkljuceno = false;
109.}
```

Grijanje je definirano u linijama 110-121 na način da se pali u noćnom periodu samo ako je temperatura manja od 13 stupnjeva C, te da zagrijava do 15 stupnjeva C, a u dnevnom režimu da grije ako temperatura padne ispod 25 stupnjeva C te da se ugasi kad dostigne 25 stupnjeva Celzija.

```
110.// Paljenje i gašenje grijanja
111.if (temperatura < 13) {
112.if (svjetlo < 100 && temperatura < 15) {
113.digitalWrite(grijanjePin, HIGH);  // Uključuje se grijanje po noći
114.} else if (svjetlo >= 100 && temperatura < 25) {
115.digitalWrite(grijanjePin, HIGH);  // Uključuje se grijanje po danu
116.} else {
117.digitalWrite(grijanjePin, LOW);  // Isključuje se grijanje
118.}
119.} else {
120.digitalWrite(grijanjePin, LOW);  // Isključuje se grijanje
121.}
```

Linije koda 122-144 se tiču prikaza podataka na OLED ekranu, *switch* funkcijom pozivamo varijablu *trenutniPrikaz* kako bi definirali početni slučaj 0 te u tom slučaju se prikazuje temperatura, nakon *break* naredbe izlazimo iz *switch* petlje te nam se trenutni prikaz povećava za 1 i ponovno ulazimo u *switch* petlju gdje se sada prikazuje informacija iz *case* 2, i tako i za ostale 2 varijable. U programskom kodu ranije postavljen je *delay* od 2 sekunde tako da se prikaz osvježava svake dvije sekunde.

```
122.switch (trenutniPrikaz) {
123.case 0:
124.display.print( "Temp: ");
125.display.print(formatiranaTemperatura);
126.display.println("C");
127.break;
128.case 1:
129.display.print( "Vlaga: ");
130.display.print(formatiranaVlaga);
131.display.println("%");
132.break;
133.case 2:
134.display.print("Vlaga tla: ");
135.display.print(formatiranaVlagaTla);
136.display.println( "%");
137.break;
138.case 3:
139.display.print("Svjetlo: ");
140.display.print(formatiranoSvjetlo);
141.display.println("Lx");
142.break;
143.}

144.trenutniPrikaz = (trenutniPrikaz + 1) % 4; // Rotacija između 0, 1, 2 i 3
   prikaza podataka
```

Linijom koda 145 osvježavamo OLED display kako bi prikazali gore navedene podatke.

```
145.display.display();
```

Naposljetku, stavljena je odgoda izvođenja da se dobije 2 sekunde razmaka u prikazivanju podataka.

```
146.delay(2000)
147.}


148.}
149.}
```

6. ZAKLJUČAK

Tema automatizacije vezana za prehranu u današnje vrijeme je jedna od značajnijih tema kojima se mogu studenti posvetiti. U našim krajevima još nije toliko razvijena automatizacija proizvodnje hrane ali se sve više teži ka tome. Kroz ovaj rad spoznalo se mnogo značajnih informacija i saznanja o tome kako uspješno automatizatirati staklenik kako bi se uzgajalo voće i povrće ali potpomognuto modernom tehnologijom. Arduino je za ovakav projekt sasvim dostatan mikrokontroler te se nije ni u jednom trenutku pokazao kao nepovoljan za ovaj projekt. Tijekom testova uspješno je savladavao zadatke koji su mu zadani. Zadatak je sam po sebi kompleksan gdje postoji još mjesta za napredak i modernizaciju ovog sustava misleći na industrijsku primjenu. Ovaj osnovni sustav nam prikazuje trenutne vrijednosti mikroklime unutar staklenika i prati sve podatke kako bi mogao reagirati u potrebnom trenutku. Što se tiče poboljšanja sustava za neki budući razvoj svakako bi trebalo implementirati bežičnu povezanost s nekim udaljenim računalom na koji bi se mogli spremati podaci te bi se mogla napraviti bolja analiza učinkovitosti staklenika uz poboljšanje zaštićenosti elemenata i pločica koje se koriste u radu od vlage prašine i drugih vanjskih utjecaja. Tu se vidi da s cijenom dolazi i „osiromašenija“ oprema. Sklop je ispunio zadana očekivanja te odradio funkciju koja je zadana kao cilj ovog rada.

Popis literature

Allen, L. (2015). *Accelerated greenhouse growth to help feed the world*. https://www.farmprogress.com/vegetables/accelerated-greenhouse-growth-to-help-feed-the-world (Pristupljeno 19.09.2023.)

Cohen, J.E. (2001). *World population in 2050: assessing the projections*. https://www.semanticscholar.org/paper/World-population-in-2050%3A-assessing-the-projections-Cohen/28ac4827d549b04e9c42645fff62a515d58efed2 (Pristupljeno 23.09.2023.)

Janick, J., i Paris, H. (2022). *History of Controlled Environment Horticulture: Ancient Origins*. HortScience, 57(2), 236-238. https://doi.org/10.21273/HORTSCI16169-21

Jeaunita J., Sarasvathi, V., Harsha, M.S., Bhavani B.M. , Kavyashree, T. *An automated greenhouse system using agricultural Internet of Things for better crop yield*, https://doi.org/10.1049/cp.2018.1388 (Pristupljeno 23.09.2023)

Muijzenberg. (1980). *A history of greenhouses* / Erwin W.B. van den Muijzenberg. (Preliminary edition.). Institute for Agricultural Engineering, distributor, https://discovered.ed.ac.uk/permalink/44UOE_INST/iatqhp/alma9918480803502466 (pristupljeno 19.09.2023).

Newman, S. E. (2013). *Sustainable Commercial Greenhouse Production*.

Nurhasanah R. (2021). *Design and Implementation of IoT based Automated Tomato Watering System Using ESP8266* https://doi.org/10.1088/1742-6596/1898/1/012041 (Pristupljeno 22.09.2023)

Ponce, P. (2014). *Greenhouse Design and Control.* CRC Press. https://www.routledge.com/Greenhouse-Design-and-Control/Ponce-Molina-Cepeda-Lugo-MacCleery/p/book/9781138026292 (Pristupljeno 22.09.2023)

Radha, K., & Igathinathane, C. (2007). *Greenhouse Technology and Management*. Hyderabad, IND: Global Media.

Ritchie H., Roser M., *Urbanization* (2018). https://ourworldindata.org/urbanization#citation (Pristupljeno 20.09.2023)

Saha, T., Jewel, M. K. H., Mostakim, M. N., (2017). *Construction and Development of an Automated Greenhouse System Using Arduino Uno"*, International Journal of Information Engineering and Electronic Business (IJIEEB), Vol.9, No.3, pp.1-8, 2017. DOI: https://doi.org/10.5815/ijieeb.2017.03.01 (pristupljeno 19.09.2023)

Shamhsiri R. R. (2018), *Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture*.

Tiwari, G. (2003). *Greenhouse Technology for Controlled Environment.* Pangbourne, England: Alpha Science International Ltd https://www.researchgate.net/publication/316911414_Greenhouse_Technology_for_Controlled_Environment (pristupljeno 22.09.2023)

Vermeulen S.J., Campbell B.M., Ingram J.S.. *Climate change and food systems*. Annu Rev Environ Resour. 2012;21(37):195–222. DOI: https://10.1146/annurev-environ-020411-130608 (Pristupljeno 23.09.2023)

Popis slika

Popis tablica

DODATAK - Potpuni programski kod rješenja

```
#include <Wire.h>
#include "DFRobot_SHT20.h"
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Fonts/FreeSans9pt7b.h>
#include <Servo.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define SSD1306_I2C_ADDRESS 0x3C
#define buttonPin 7
#define vlagaTlaPin A2
#define ldrPin A3
#define svjetloPin 8 // Pin na kojem je spojeno svjetlo (relej ili tranzistor)
#define relejPin 9 // Pin za upravljanje relejem za navodnjavanje
#define grijanjePin 10//Pin koji upravlja relejem grijanja

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire);
DFRobot_SHT20 sht20;
Servo myservo;

int servoPin = 11;  // Pin na kojem je servo spojen
int trenutniPrikaz = 0; // Varijabla za praæenje trenutnog prikaza (0 -
temperatura, 1 - vlažnost zraka, 2 - vlažnost tla, 3 - svjetlo)
int buttonState = 0;
unsigned long prethodnoVrijeme = 0; // Varijabla za pohranu vremena zadnjeg
prikaza
const long interval = 2000; // Interval izmjene prikaza u milisekundama (2
sekunde)
unsigned long trajanjeSvjetla = 12 * 60 * 60 * 1000; // 12 sati u milisekundama
int pragVlageTlaZaUkljucivanje = 75;  // Prag za ukljuèivanje navodnjavanja
int pragVlageTlaZaIskljucivanje = 80; // Prag za iskljuèivanje navodnjavanja
int pragSvjetla = 500;                // Prag svjetla ispod kojeg æe se
navodnjavanje moæi uključiti (u Lux)
bool navodnjavanjeUkljuceno = false;

void setup() {
  Serial.begin(9600);

  pinMode(buttonPin, INPUT);
  pinMode(svjetloPin, OUTPUT);
  pinMode(relejPin, OUTPUT);
  pinMode(grijanjePin, OUTPUT);
  digitalWrite(svjetloPin, LOW);  // Inicijalno isključèeno svjetlo
  digitalWrite(relejPin, LOW);    // Inicijalno isključèen relej za navodnjavanje
  digitalWrite(grijanjePin, LOW); //Inicijalno isključèen relej za grijanje
  //pokretanje OLED ekrana
  if (!display.begin(SSD1306_I2C_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for (;;);
  }
  display.clearDisplay();
  display.setFont(&FreeSans9pt7b);
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
```

```
  display.setCursor(0, 20);
  display.println("Dobrodosli!");
  display.display();
  delay(2000);

  sht20.initSHT20();
  delay(100);
  sht20.checkSHT20();
}

void loop() {


  //Opcije rada standby ili radni automatski naèin

  buttonState = digitalRead(buttonPin);

  if (buttonState == LOW) {
    Serial.println("Tipka pritisnuta! Pauza u izvoðenju programa...");
    display.clearDisplay();
    display.setCursor(0, 20);
    display.setFont(&FreeSans9pt7b);
    display.print("Standby...");
    display.display();
    while(digitalRead(buttonPin) == LOW) {
      delay(100);
    }
    Serial.println("Izlaz iz pauze.");
  } else {

    // Provjera vremena za izmjenu prikaza
    unsigned long trenutnoVrijeme = millis();
    if (trenutnoVrijeme - prethodnoVrijeme >= interval) {
      // Ažurirajte prikaz
      display.clearDisplay();
      display.setCursor(0, 20);
      display.setFont(&FreeSans9pt7b);
      float temperatura = sht20.readTemperature();
      String formatiranaTemperatura = String(temperatura, 1); // 1 znaèi jedna
decimala
      float vlaga = sht20.readHumidity();
      String formatiranaVlaga = String(vlaga, 1);
      int vlagaT = analogRead(vlagaTlaPin);
      float VlagaTla = map(vlagaT, 190, 950, 100, 0);
      String formatiranaVlagaTla = String(VlagaTla, 1);
      float svjetlo = analogRead(ldrPin);
      svjetlo = 1023 - svjetlo;
      float svjetloLux = map(svjetlo, 0, 1000, 1, 10000);
      String formatiranoSvjetlo = String(svjetlo, 0);


      // Provjera svjetla i upravljanje svjetlom
     if (svjetlo < pragSvjetla && trenutnoVrijeme - prethodnoVrijeme >=
trajanjeSvjetla) {
        digitalWrite(svjetloPin, HIGH); // Uključuje se  svjetlo
        prethodnoVrijeme = trenutnoVrijeme; // Resetirajte vrijeme
      } else if (svjetlo >= pragSvjetla) {
        digitalWrite(svjetloPin, LOW); // Isključuje se svjetlo ako je svjetlina
dovoljno visoka
```

```
      }
      // Otvaranje i zatvaranje vrata staklenika preko servomotora
     if (temperatura > 27) {
         int kut = map(temperatura, 27, 30, 0, 90);
         kut = constrain(kut, 0, 90);
         myservo.write(kut);
      } else {
         myservo.write(0);
       }
      // Provjera vlage tla i upravljanje navodnjavanjem
     if (VlagaTla < pragVlageTlaZaUkljucivanje && svjetlo < pragSvjetla &&
!navodnjavanjeUkljuceno) {
        digitalWrite(relejPin, HIGH); // Uključuje se relej za navodnjavanje
        navodnjavanjeUkljuceno = true;
     } else if (VlagaTla >= pragVlageTlaZaIskljucivanje || svjetlo >= pragSvjetla)
{
        digitalWrite(relejPin, LOW); // Isključuje se relej za navodnjavanje
        navodnjavanjeUkljuceno = false;
       }
      // Paljenje i gašenje grijanja
       if (temperatura < 13) {
        if (svjetlo < 100 && temperatura < 15) {
            digitalWrite(grijanjePin, HIGH);  // Uključuje se grijanje po noći
        } else if (svjetlo >= 100 && temperatura < 25) {
            digitalWrite(grijanjePin, HIGH);  // Uključuje se grijanje po danu
        } else {
            digitalWrite(grijanjePin, LOW);  // Isključuje se grijanje
        }
    } else {
        digitalWrite(grijanjePin, LOW);  // Isključuje se grijanje
    }


      switch (trenutniPrikaz) {
        case 0:
          display.print("Temp: ");
          display.print(formatiranaTemperatura);
          display.println("C");
          break;
        case 1:
          display.print("Vlaga: ");
          display.print(formatiranaVlaga);
          display.println("%");
          break;
        case 2:
          display.print("Vlaga tla: ");
          display.print(formatiranaVlagaTla);
          display.println("%");
          break;
        case 3:
          display.print("Svjetlo: ");
          display.print(formatiranoSvjetlo);
          display.println("Lx");
          break;
      }

      trenutniPrikaz = (trenutniPrikaz + 1) % 4; // Rotacija između 0, 1, 2 i 3
prikaza podataka
```

```
      display.display();

      delay(2000);
    }


  }
}
```