

# **Modelovanje autonomnog sistema za kompaktan uzgoj biljaka**

Modelovanje i simulacije

ALEKSANDAR STOJANOVIĆ RN97-2018

8. januar 2020.



# Sadržaj

<b>1</b>	<b>Priprema</b>	<b>7</b>
1.1	Definisanje problema . . . . .	7
1.2	Plana izrade projekta . . . . .	7
1.2.1	Merenje temperature i regulacija . . . . .	10
1.2.2	Merenje vlažnosti zemlje i ambijenta . . . . .	10
1.2.3	Regulacija svetlosnog ciklusa . . . . .	11
1.2.4	Regulacija rada ventilatora . . . . .	12
1.2.5	Zalivanje biljke . . . . .	13
1.2.6	Logovanje . . . . .	13
1.3	Hardverski dizajn . . . . .	14
1.4	Softverski dizajn . . . . .	14
1.4.1	Pokretanje i princip rada . . . . .	14
1.4.2	Temperatura . . . . .	15
1.4.3	Vlaga zemlje . . . . .	15
1.4.4	Obrada učitanih vrednosti . . . . .	16
1.4.5	Ispis na ekran . . . . .	16
<b>2</b>	<b>Izgradnja modela</b>	<b>19</b>
2.1	Konceptualizacija modela . . . . .	19
2.2	Kolekcija podataka . . . . .	19
2.2.1	Regulacija temperature . . . . .	19
2.2.2	Regulacija zalivanja . . . . .	20
2.2.3	Regulacija svetlosnog ciklusa . . . . .	21
2.3	Prevodjenje modela . . . . .	21
2.4	Verifikacija . . . . .	21
2.5	Validacija . . . . .	21
<b>3</b>	<b>Izvršavanje simulacije</b>	<b>23</b>
3.1	Dizajn eksperimenta . . . . .	23
3.2	Izvršavanje i analiza . . . . .	23
3.3	Dodatna izvršavanja . . . . .	23

<b>4</b>	<b>Implementacija</b>	<b>25</b>
4.1	Dokumentacija i izveštaj . . . . .	25
4.2	Implementacija . . . . .	25

# Uvod

## Svrha rada

Ovaj rad je stvoren sa ciljem da opiše jednostavno i pristupačno rešenje za proizvodnju hrane u veštačkim uslovima.

U ovom radu prezentovaću svoj pristup dizajniranja ovakvog sistema. Detaljno cu analizirati principe rada individualnih podsistema koji sačinjavaju ovu jedinicu i simulirati njihov rad.

## Uzgoj u zastvorenom prostoru

Ovakav vid proizvodnje sa urbanizacijom postaje sve popularniji. Kako gradovi postaju sve veći, potreba za organskom hranom raste te se ljudi okreću alternativnim metodama uzgoja.

Kada je reč o zatvorenim prostorima, u glavnom se misli na kontrolisano okruženje koje ima za cilj da olakša razvoj biljke pa kasnije i samih plodova. Ovo se postiže adekvatnim planiranjem što podrazumeva razmatranje i ispunjavanje svih uslova neophodnih za rast.

## Tradicionalan ili zatvoren uzgoj

Dok nam tradicionalan pristup uzgoju olakšava logistiku i nudi dosta pogodnije mogućnosti za ekspanzije, zatvoren pristup pruža kompletno kontrolu nad samim okruženjem. Pored toga biljka je kompletno izolovana od negativnih spoljašnjih faktora kao što su:

- paraziti,
- naglih oscilacija temperature,
- prevelike količine padavina.

Sama činjenica da je biljka u izolovanom okruženju nam omogućava da bliže pratimo njen razvoj. Ovo posebno dolazi do izražaja kod uočavanja problema u ranim fazama. Pored otkrivanja problema omogućava nam optimizaciju procesa uzgoja što dovodi do efikasnije proizvodnje.

# Glava 1

## Priprema

### 1.1 Definisanje problema

Dakle, naš cilj je izrada autonomne jedinice koja radi bez čovekovog prisustva. Kako bismo to postigli moramo se osloniti na nekakvu upravljačku jedinicu koja će biti zadužena za kontrolu celokupnog sistema uz oslonac na senzore.

Glavna prepreka je limitirana količina prostora koja nam je na raspolaganju. Kada je reč o uzgoju u zatvorenim prostorima podrazumeva se da nam je sam prostor jako važan resurs i potrebno je iskoristiti ga što efikasnije. Tek kada je prostor pravilno iskorišćen možemo započeti izradu ostalih delova sistema.

Da bismo prostor koristili efektivno bitno je da unapred definišemo funkcionalnosti našeg sistema:

- 
1. Merenje i regulacija temperature,
  2. Merenje vlažnosti zemlje i ambijenta,
  3. Regulacija svetlosnog ciklusa,
  4. Regulacija brzine ventilatora,
  5. Zalivanje biljke,
  6. Logovanje
- 

Imajući ove funkcije na umu možemo odrediti grub plan projekta. U sledećoj tački ćemo da detaljno definisati svaku od ovih funkcija i opisati hardver koji ćemo koristiti za izradu. Cilj nam je da hardver zauzme sto manje prostora kako bi ostalo što više prostora za rad sa biljkom.

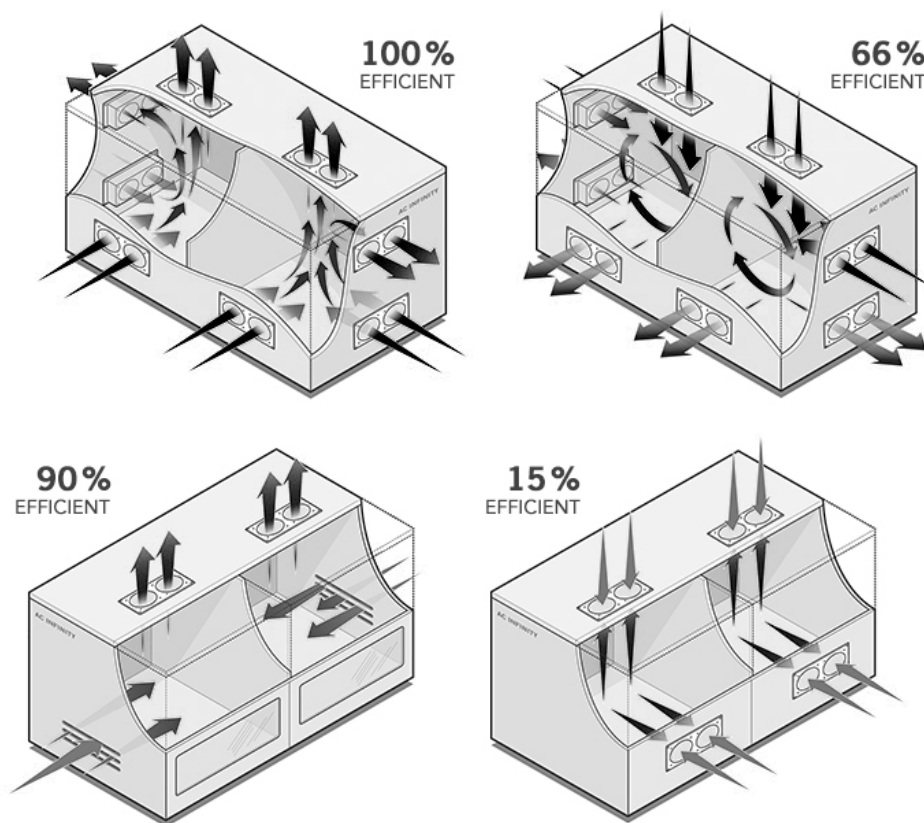
### 1.2 Plana izrade projekta

Sistem će biti smešten u kabinet dimenzija 35x35x60 koji je izradjen od metala. Ova veličina će podržati biljku srednjih veličina. Neophodno je da

unutrašnjost kabineta bude obložena reflektivnim materijalom koji će dobiti da biljka iskoristi pun potencijal svetla. Pored reflektivnog materijala, unutrašnje ivice sistema je potrebno obložiti gumom kako bismo efektivno izolovali sistem od sredine.

Na vrh(plafon) kabineta neophodno je ugraditi nekakvo rešenje za fiksiranje svetla u vidu kuka. Za kuke se kače sajle koje će držati svetlosnu fiksaturu. Na poledjini kabineta izbušićemo dve rupe za kačenje ventilatora koji će obezbediti cirkulaciju. Donji otvor će služiti za unos svežeg vazduha dok se na gornji fiksira filter vazduha radi kontrole mirisa. Razlog ovoga je sledeći:

Ako pogledamo sledeće slike<sup>1</sup>



Možemo uvideti da je najbolja konfiguracija ventilatora ona koja vazduh uvlači pri dnu, a topao izbacuje pri vrhu.

Što se kontrolera tiče, budući da ovakav sistem zahteva kontinualan rad sto podrazumeva dugoročno opterećenje, Arduino Uno<sup>2</sup> je idealno rešenje jer

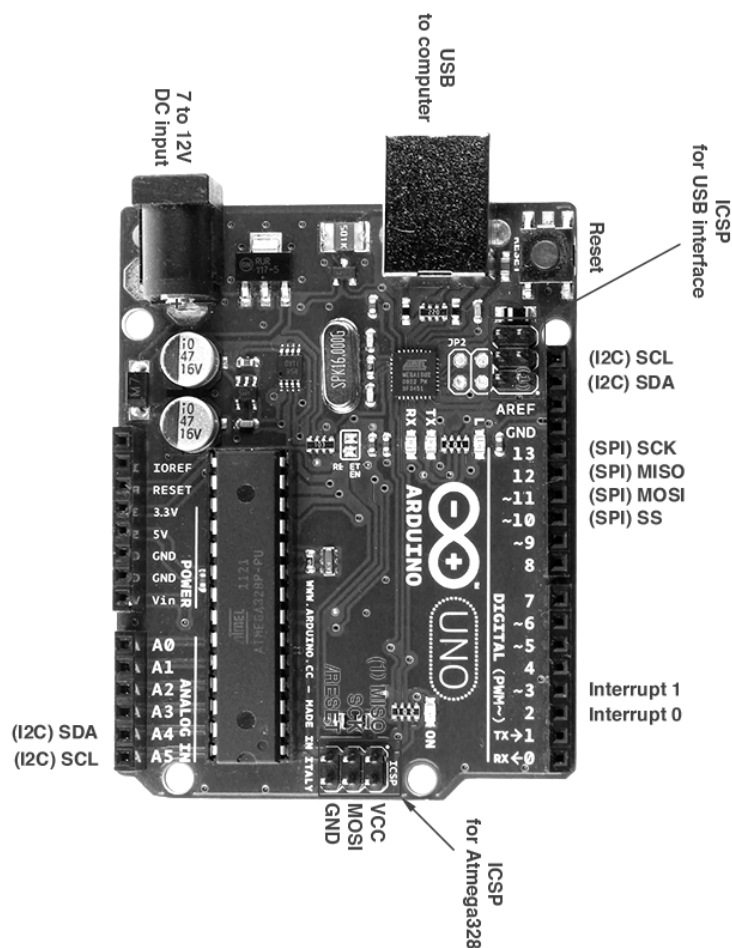
<sup>1</sup>AC infinity je kompanija koja se bavi solucijama za hlađenje, više na: <https://www.acinfinity.com/pages/technical-guides/cabinet-cooling-and-ventilation.html>

<sup>2</sup>Arduino Uno je open-source rešenje u vidu kontrolera za IoT projekte koji pruža



nudmi stabilnost pod dugoročnim radom i jednostavnu integraciju senzora. Pored stabilnosti i male veličine ako pogledamo njegov dijagram možemo uočiti da imamo izlaze od 3.3V i 5V što je idealno za napajanje senzora i releja.

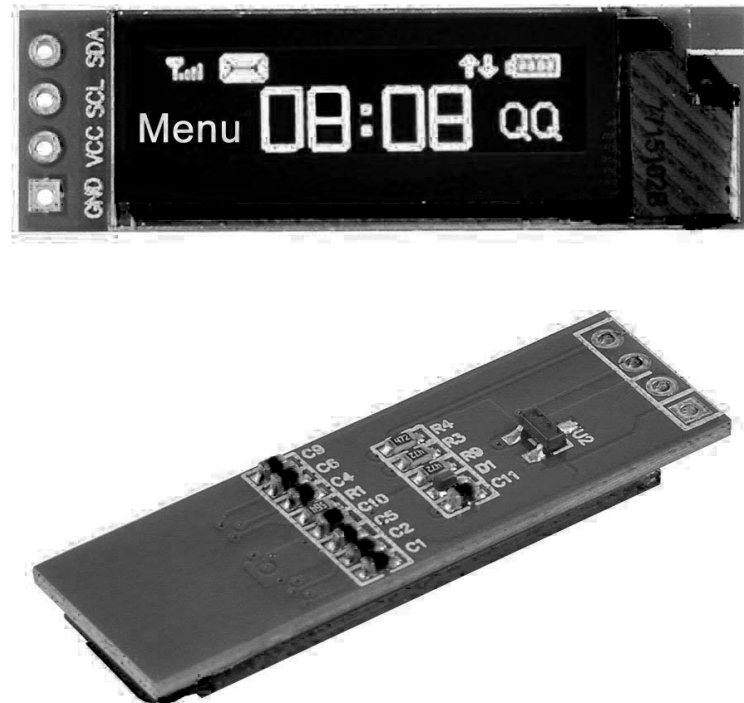
Kontroler će biti smešten izvan sistema sa spoljašnje strane unutar odeljka zajedno sa protobordom za povezivanje svih komponenti i releja. Ovo će predstavljati mozak sistema odakle ćemo rutirati kablove. Kao odeljak, plastična razvodna kutija je dimenzija 20cm x 20cm x 5cm je sasvim dovoljna za naše potrebe.



Kao monitor za feedback sistema koristimo mali I2C<sup>3</sup> OLED ekran 128x32 veličine 0.91 inča kao dugoročno rešenje dok će serijski port biti primarno korišćen u početnim fazama izrade.

mnoštvo mogućnosti. Više na: <https://www.arduino.cc/en/Guide/Introduction>

<sup>3</sup>I2C protokol služi za serijsku komunikaciju sa mikrokontrolerima. Više na: <https://i2c.info/>



### 1.2.1 Merenje temperature i regulacija

Čest problem sa kompaktnim zatvorenim baštama je visoka temperatura.

Merenje temperature je krucijalan korak jer je to jedan od glavnih faktora okruženja. Različite biljke zahtevaju različite uslove poput povećane vlage, stoga neophodno je koristiti adekvatan hardver.

DS18B20<sup>4</sup> je senzor po izboru iz razloga što je fabrički izolovan, ima dosta širi opseg vrednosti od očekivanog unutar sistema i napaja se sa 3.3V ili 5.5V.

Regulaciju rešavamo korišćenjem ulaznih i izlaznih ventilatora(vise o regulaciji temperature će biti).

### 1.2.2 Merenje vlažnosti zemlje i ambijenta

Praćenjem vlažnosti zemlje nam omogućava da automatizovano zalivamo biljku u zavisnosti od njenih potreba. Za razliku od fiksnih ciklusa zalivanja

<sup>4</sup>Digitalni senzor toplote koji nudi 9-bit do 12-bit vrednosti toplote izražene u celizijusu. Vise na: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

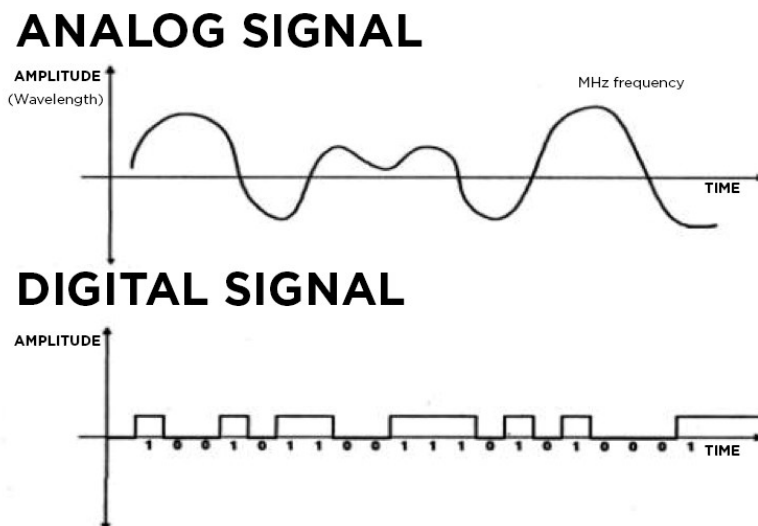
kod kojih može doći do preteranog navodnjavanja ovde se mehanizam za navodnjavanje aktivira samo kada je to potrebno. Ovo sprečava prekomerno zalivanje sto dovodi do raznih problema poput:

- Nedostatka kiseonika u zemlji za korenov sistem,
- gljivično oboljenje korena,
- nedostatak hranjivih materija usled prekomernog ispiranja

Za vlagu ambijenta koristimo HR202 senzor vlage u vazduhu, a za zemljiste koristimo kapacitivni senzor LDTR-WG0236.

Ovaj senzor vraća analognu vrednost što znaci da dobijamo kontinualan signal velike preciznosti.

Ako pogledamo sledeci grafik:



Ovo znači da ćemo uvek dobijati neku vrednost vlage za razliku od digitalnih koji za neku vrednost X vraća 0 ako je izmerena manja ili 1 ako je veća. Tačnu vrednost valge za aktivaciju ćemo utvriditi merenjem.

### 1.2.3 Regulacija svetlosnog ciklusa

Različite biljke zahtevaju specifične svetlosne cikluse. Stoga, moramo konfigurisati naš kontroler po parametrima biljke kako bismo joj pružili optimalne uslove. Interval osvetljenja je konfigurisan softverski i zavisno od istog kontroler utiče na relej.

Što se izbora tehnologije svetla tiče u opticaju imamo sledeće:

- LED - Light emitting diode
- HID - High-intensiti discharge

- Flourescent(CFL)

Tabela 1.1: Karakteristike različitih tehnologija

	CFL	HID	LED
Inicijalna cena	Nisko	Srednje	Nisko/skupo
Potrošnja	Nisko	Srednje/Visoko	Nisko
Efikasnost	Osrednje	Dobro	Dobro/Odlično
Udaljenost	Mala	Srednja	Srednja/Velika

U sledećoj tabeli<sup>5</sup> ćemo detaljno pogledati razliku u proizvodnji toplote između različitih tipova sijalica.

Tabela 1.2: Proizvodnja toplote

Tehnologija	Gubitak putem radijacije[%]	Toplotni gubitak[%]
Inkandescentne sijalice	81 - 86	5 - 6
fluorescentne sijalice	30 - 32	44
HID(živa)	62 - 65	16 - 22
HID(sodijum)	57 - 74	7 - 20
HID(metal halogene)	47.3 - 63.3	10 - 23
LED	0 - 0.2	80 - 88

Zbog efikasnosti i povoljne cene koristićemo LED svetla za sistem. Ova tehnologija je najbolja opcija za nas jer imamo dovoljno prostora a pored toga je jako isplativa. Same dimenzije ovih fikstura su male pa ih to čini jos povoljnijim.

Kada gledamo snagu svetla, preporučeno je koristiti minimum

$$35Wpo1ft^2,$$

gde je optimalno 50W a maksimalno 80W. Ako pogledamo prostor sistema koji iznosi  $1225cm^2$  što je  $1.40ft^2$ , biće dovoljno koristiti LED svetlo snage 150W (efektivno oko 100W) sa ugrađenim hladnjakom.

#### 1.2.4 Regulacija rada ventilatora

Ventilatori nam koriste za razmenu vazduha sa okolinom. Sa druge strane, kako utičemo na njihovu brzinu jedinica će se brže odnosno sporije hladiti.

Da bismo pronašli odgovarajuću veličinu ventilatora za sistem moramo odrediti zapreminu samog prostor sa kojim radimo:

<sup>5</sup>Vise o generaciji toplote na: budući da researchgate ima užasne linkove, rad ce biti na kraju.

$$35cm * 35cm * 60cm = 73500cm^3 = 2.595628ft^3$$

Iz ovoga vidimo da su nam dovoljno obicni 12V ventilatori. Budući da se na izlaz kači filter vazduha koji u sebi sadrži dva 60mm ventilatora koji po specifikaciji pomeraju 38.35cfm a nasa jedinica ima samo 2.5, na ulaz možemo staviti jedan ventilator od 120mm koji cemo uključivati u slučaju da nam temperatura predje dozvoljenu. Ovo radimo sa ciljem da unutar jedinice stvorimo negativni pritisak (izbacuje se vise nego ulazi) kako bi obezbedili da nam vazduh izlazi samo na predvidjenom mestu a ne kroz sitne pukotine.

Prebacivanjem izduva sa 9V na 12V i ukljucivanjem usisa na 5V vrši se brža razmena toplote dok se temperature ne vrati na dozvoljeni nivo.

### 1.2.5 Zalivanje biljke

Zalivanje biljke je jedan od elementarnih zahteva koje moramo ispuniti. Za ovo se koristi pumpa za vodu od 12V koja je potopljena u kanistar sa vodom. Pumpa se aktivira putem releja kada senzor vlage u zemlji detektuje niske količine vlage. Pre puštanja u rad potrebno je konfigurisati senzor. Ako senzor nije adekvatno kalibrisan može doći do preteranog zalivanja ili čak prelivanja vode unutar sistema.

Od velikog značaja je ravnomerna distribucija vode unutar saksije kako bi senzor stvarao sto realniju sliku o stanju unutar iste. Rešenje za ovo je postavljanje cevi po obodu saksije. Cev je neophodno izbusiti ravnomerno tako da je jednak razmak i veličina rupa.

### 1.2.6 Logovanje

Logovanje nam omogućava detaljnu analizu procesa i samog rada naše mašine ako se korektno implementira. Znatno olakšava otkrivanje greške ili kvara, pomaže u rešavanju i služi kao output sistema.

Generičan modul za sd kartice se koristi ovde koji ima sledeci pinout:

- 
- VCC - 3.3V
  - VCC - 5V
  - MOSI
  - CLK
  - MISO
  - GND - ground

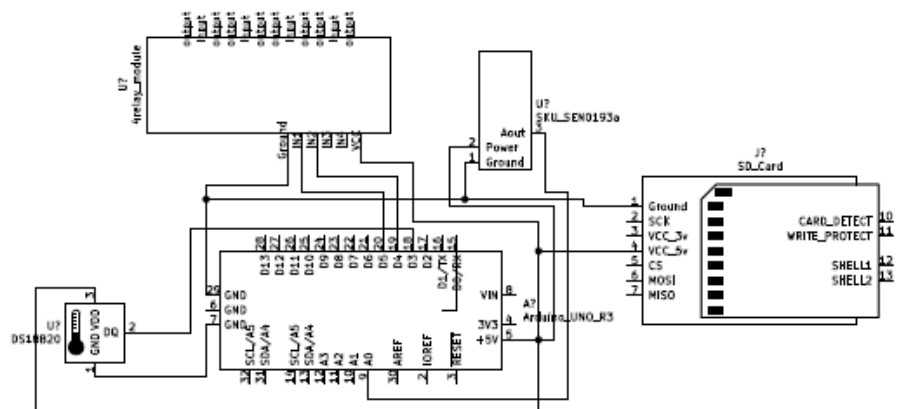
Ovo ga čini idealnim za naše uslove. Na kartici će se beležiti svaki input koji sistem primi zajedno sa akcijom koja je izvršena kroz vreme.

Ovakvo logovanje nam omogućava da pratimo rad mašine uz oslonac na grafike kroz vreme. Budući da imamo punu kontrolu nad formatiranjem podataka, možemo se postarati da izlaz bude kompatibilan sa nekim od popularnih softverskih alata.

U sledećim glavama ćemo definisati primer funkcije za logovanje zajedno sa formatom podataka.

### 1.3 Hardverski dizajn

Posto smo odredili komponente koje odgovaraju nasim uslovima možemo staviti grubu šemu sistema. Budući da softver(Escheme) ne podržava breadboard(protoboard), veze poput 5V i zemlje sa arduina sam kačio na isti izlaz. U praksi ovo je urađeno "proširivanjem" protobordom.



Ulaze i izlaze modula kačimo direktno na kontroler. Na relej se kači svetlo, pumpa za vodu i ventilatori(na jedan terminal kačimo 9V, a na drugi 12V).

### 1.4 Softverski dizajn

#### 1.4.1 Pokretanje i princip rada

U inicijalnom setup petlji kontrolera inicijalizujemo sve potrebne biblioteke i promenljive.

Glavna petlja će pratiti sledeći sablon rada:

- Hvatanje temperature,
  - hvatanje vlage zemlje,
  - pokretanje pomoćne funkcije koja određuje akcije u vidu zalivanja ili regulacije toplote,
  - zapisivanje podataka i akcija,
  - ispis na ekran.
- 

### 1.4.2 Temperatura

Hvatanje temperature vrši se pozivom funkcije `read_temperature()` koja će od senzora zatražiti trenutnu temperaturu. Ako je doslo do greske vratiće staru vrednost u protivnom biće vraćena očitana.

```
/*
 * Funkcija za citanje temperature preko senzora
 * @return float - temp
 */
float read_temperature(){
    //Var za temperaturu
    float tempC = sensors.getTempCByIndex(0);

    //Trazimo temperaturu
    Serial.print("Hvatam temperaturu");
    sensors.requestTemperatures();
    Serial.println("uspeh");

    Serial.print("Temperatura za device 1 (index 0) je: ");
    Serial.println(tempC);

    if(tempC != DEVICE_DISCONNECTED_C) return tempC;

    //Exception
    else Serial.println("Greska prilikom citanja!");
    return null;
}
```

### 1.4.3 Vlaga zemlje

Što se vlage zemlje tiče proces je isti, funkcija izgleda ovako:

```

/*
 * Funkcija za citanje vlage zemlje preko senzora
 * @return float - soil_val
 */
float read_soil_moisture(){

    Serial.print("Hvatam vlagu");

    //citanje vlaznosti
    int soil_val = analogRead(SENSOR_PIN);

    //konverzija
    soil_val = map(soil_val, 550, 0, 0, 100);

    //ispis
    Serial.print("Vlaga: ");
    Serial.print(soil_val);

    return soil_val;
}

```

#### 1.4.4 Obrada učitanih vrednosti

Obrada vrednosti vrši se pozivom funkcije `process_inputs`

```

/*
 * Funkcija za kontrolu toka i logovanja
 * @param temp_val - temperatura
 * @param soil_val - vlaga
 */
void process_inputs(float temp_val, float soil_val){

    if(temp_val >= 30) digitalWrite(relay1, HIGH);
    else digitalWrite(relay1, LOW);

    if(temp_val <= 20) digitalWrite(relay1, LOW);

}

```

#### 1.4.5 Ispis na ekran

Ova funkcija prvo konvertuje vrednosti u stringove i smešta ih u char niz. Nakon što su smešteni u svoje buffere pozivaju se funkcije za ispis.

```

/*

```



```
* Funkcija za ispis na ekran
* @param temp_val – temperatura
* @param soil_val – vlaga
*/
void output_values(float temp_val, float soil_val){

    //Pravimo char array za drawStr funkciju
    String temp_string = String(temp_val);
    String soil_string = String(soil_val);

    char temp_buffer[5];
    temp_string.toCharArray(temp_buffer, 5);

    char soil_buffer[5];
    soil_string.toCharArray(soil_buffer, 5);

    //Ispis na ekran
    temp_out(temp_buffer);
    delay(2000);
    soil_out(soil_buffer);
}
```



## Glava 2

# Izgradnja modela

Sistem se može posmatrati kao kontrolni sistem

Kada smo definisali plan sistema možemo započeti izradu modela. U ovoj glavi posvetićemo se parametrima i njihovoj korelaciji unutar sistema. Kada su nam veze jasne, lako možemo napraviti model za simulaciju.

### 2.1 Konceptualizacija modela

Kako bismo stvorili koncept modela moramo se vratiti na funkcije našeg sistema. Analizom funkcija i konkretnog hardvera pronalazimo matematički model koji predstavlja temelj naše simulacije.

### 2.2 Kolekcija podataka

Podatke možemo prikupiti preko samih proizvođača komponenti za koje smo se opredelili. Čitanjem tehničke dokumentacije ili samih specifikacija proizvoda možemo stvoriti gruba očekivanja kako će se sistem ponašati u realnim uslovima.

Bitno je imati na umu da vrednosti koje nam proizvođač da uglavnom nisu merodavne pa je neophodno, ukoliko je to moguće, istraživanjem utvrditi realne performanse komponente. Ukoliko nije moguće naći iste, kako bi izbegli dodatne troškove, moramo uračunati do 30% devijacije od datih.

#### 2.2.1 Regulacija temperature

Za početak neophodno je odrediti izvore toplote unutar sistema. U našem slučaju to je LED svetlo od 150W koje, bez obzira na led tehnologiju, ipak proizvodi toplotu.

Sada moramo uvesti termin BTU<sup>1</sup> kako bismo lakše ordredili. Ako znamo da je  $1 \text{ Watt} = 3.41 \text{ BTU}$ . Svetlo koje koristimo iznosi 150W pa opterećenje sistema iznosi 341 BTU/h. Sledi računanje razlike ulazne i izlazne teperature vazduha. Ako pogledamo rezultate testa sistema bez aktivnog hladjenja:

Tabela 2.1: Zagrevanje - bez aktivne ventilacije

Trajanje[minut]	Temperatura[°C]	Vreme
0	25.3	9:48
15	27.4	10:03
30	29.1	10:18
60	29.1	10:48

Možemo uočiti da se temperatura sa sobne od 25.3°C popela na 29.1°C gde je stagnirala. Ako oduzmemo ove vrednosti dobićemo povećanje temperature of 4°C.

Ako pogledamo sledeću formulu:

$$CFM = BTU / (1.08 * \Delta T)$$

Pa iz ovoga dobijamo da nam je potreban CFM = 78.9. Naši ventilatori iznose manje ali testom je utvrđeno sledeće:

Tabela 2.2: Zagrevanje - aktivan izdub na 40% snage

Trajanje[minut]	Temperatura[°C]	Vreme
0	25.1	11:48
15	26.5	12:03
30	27.1	12:18
60	27.1	12:48

Uključivanjem ventilatora na samo 40% postićemo sniženje temperature za 2°C. Ako iskoristimo pun potencijal istog možemo održati sobnu temperaturu.

Za naše potrebe temperatura od 20-30[°C] je sasvim prihvatljiva što znači da je solucija za hladjenje adekvatna.

### 2.2.2 Regulacija zalivanja

Zalivanje vrši se putem pumpe za manje fontane. Po specifikaciji proizvođača protok vode iznosi  $Q = 150l/h$ . Testiranjem je utvrđeno sledeće:

Testiranjem je utvrđena tačnost specifikacije i sada nam ostaje da odredimo dužinu vremenskog perioda za koji će relej držati pumpu otvorenom.

---

<sup>1</sup>British thermal unit - Po definiciji to je količina energije potrebna da se zagreje jedna funta (0.454 kg) vode tj. jedna desetina UK Galona od 39°F (Farenhajta) na 40 °F, tj. od 3,8 °C (Celzijusa) do 4,4°C. Vise na: <https://sr.wikipedia.org/wiki/Btu>

Tabela 2.3: Rad pumpe za vodu

Trajanje[sekunde]	Količina vode u posudi[l]
0	0
15	0.5
30	1.2
60	2.4

Saksija koju ćemo koristiti ovde iznosi 1.5l što znači da ako pogledamo tabelu, možemo uočiti da je 10s sasvim dovoljno da se potopi zemlja unutar saksije.

Ovo nije optimalna količina vode ali nam pruža sigurnost od preliivanja. Pumpa će se aktivirati opet ukoliko vlažnost zemlje ne odgovara našem standardu.

### 2.2.3 Regulacija svetlosnog ciklusa

Svetlosni ciklus je definisan parametrima biljke. Biljke se razlikuju po dužini svetlosnog intervala, stoga mi ćemo uzeti "12-12"interval.

Imajući na umu da je svetlo glavni izvor toplotne energije unutar sistema, neophodno je isključiti ventilatore tokom "mračnog"perioda kako bi temperatura ostala u poželjnom opsegu.

## 2.3 Prevodjenje modela

### 2.4 Verifikacija

### 2.5 Validacija



## Glava 3

# Izvršavanje simulacije

3.1 Dizajn eksperimenta

3.2 Izvršavanje i analiza

3.3 Dodatna izvršavanja





## Glava 4

# Implementacija

### 4.1 Dokumentacija i izveštaj

### 4.2 Implementacija