

Modelovanje autonomnog sistema za kompaktan uzgoj biljaka

Modelovanje i simulacije

ALEKSANDAR STOJANOVIĆ RN97-2018

28. januar 2020.

Sadržaj

1	Priprema	7
1.1	Definisanje problema	7
1.2	Plana izrade projekta	8
1.2.1	Jedinica	8
1.2.2	Kontroler	9
1.2.3	Monitor	10
1.2.4	Merenje temperature i regulacija	11
1.2.5	Merenje vlažnosti zemlje i ambijenta	11
1.2.6	Regulacija rada ventilatora	12
1.2.7	Svetlo i zagrevanje	13
1.2.8	Zalivanje biljke	15
1.2.9	Logovanje	15
1.3	Hardverski dizajn	16
1.4	Softverski dizajn	16
1.4.1	Pokretanje i princip rada	16
1.4.2	Temperatura	17
1.4.3	Vlaga zemlje	18
1.4.4	Obrada učitanih vrednosti	18
1.4.5	Ispis na ekran	19
2	Izgradnja modela	21
2.1	Konceptualizacija modela	23
2.2	Kolekcija podataka	24
2.2.1	Regulacija zalivanja - pilotni projekat	24
2.3	Prevodjenje modela	25
2.3.1	Definisanje parametara okruženja	25
2.3.2	Izrada logike	26
2.4	Verifikacija	28
2.4.1	Sistem za kontrolu ventilatora	28
2.4.2	Obračun gubitaka	28
2.5	Validacija	29

3	Izvršavanje simulacije	31
3.1	Dizajn eksperimenta	31
3.2	Izvršavanje i analiza	31
3.3	Dodatna izvršavanja	32
4	Implementacija	33
4.1	Dokumentacija i izveštaj	33

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 kada je jedinka najosetljivija što se tiče uticaja okoline.

Pored otkrivanja problema omogućava nam optimizaciju procesa uzgoja što dovodi do efikasnije proizvodnje. Pod ovim se podrazumeva uzgoj bilja vrhunskog kvaliteta i visokog prinosa u celogodišnjoj proizvodnji bez ograničenja u pogledu sezone ili optimalnih uslova gajenja, kako u atmosferi tako i u zoni ukorenjavanja gajenih biljaka. Precizna, manuelna ili automatska kontrola temperature i relativna vlažnost vazduha, provetravanje, ishrana i navodnjavanje i održavanje idealnog vodnog i vazdušnog režima zemljišta ili supstrata doprinosi eliminisanju ili maksimalnom smanjenju potencijalnog rizika.

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.

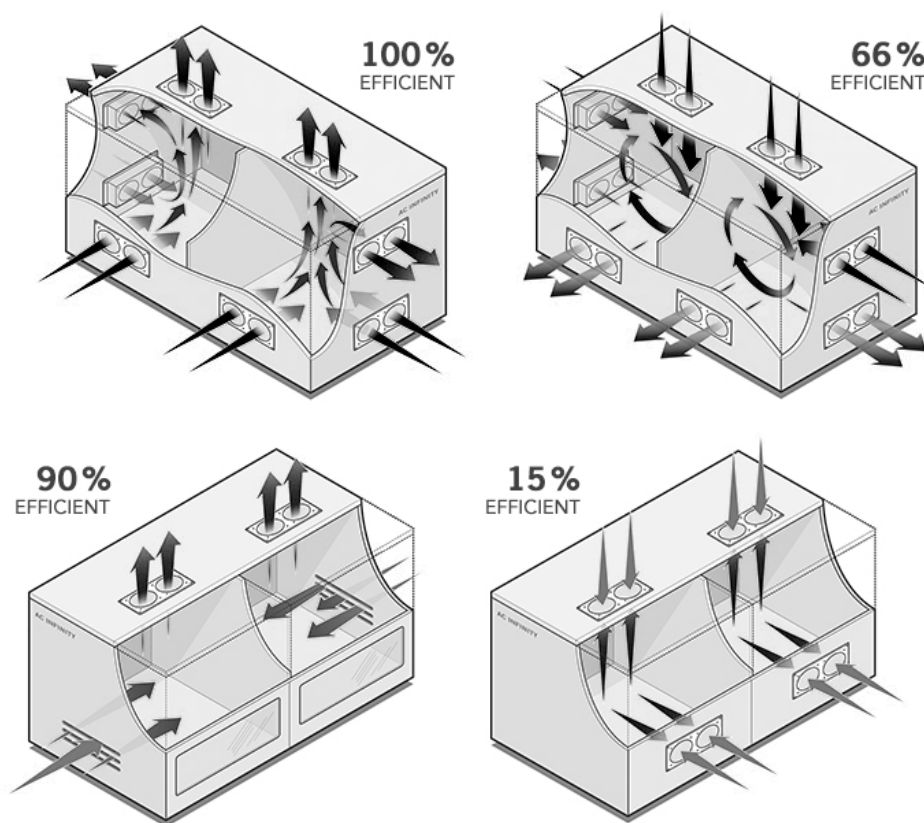
1.2 Plana izrade projekta

1.2.1 Jedinica

Sistem će biti smešten u kabinet dimenzija 35x35x60 koji je izradjen od metala. Ova veličina ce 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 ce 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 slike¹



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

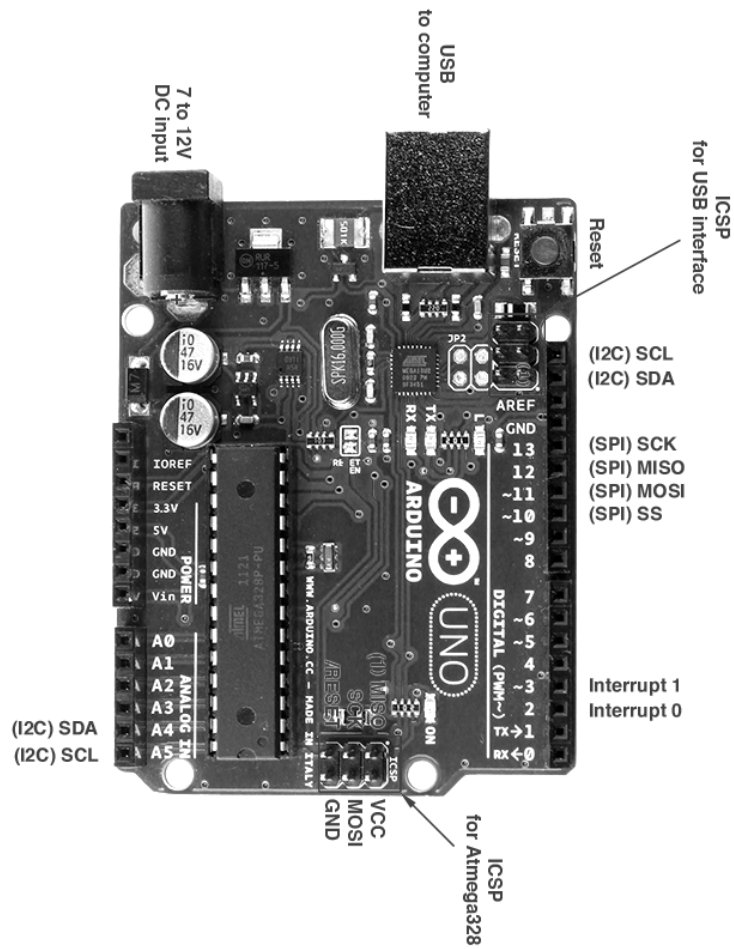
Možemo uvideti da je najbolja konfiguracija ventilatora ona koja vazduh uvlači pri dnu, a topao izbacuje pri vrhu.

1.2.2 Kontroler

Što se kontrolera tiče, budući da ovakav sistem zahteva kontinualan rad sto podrazumeva dugoročno opterećenje, Adruino Uno² je idealno rešenje jer nudi 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 sto 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.

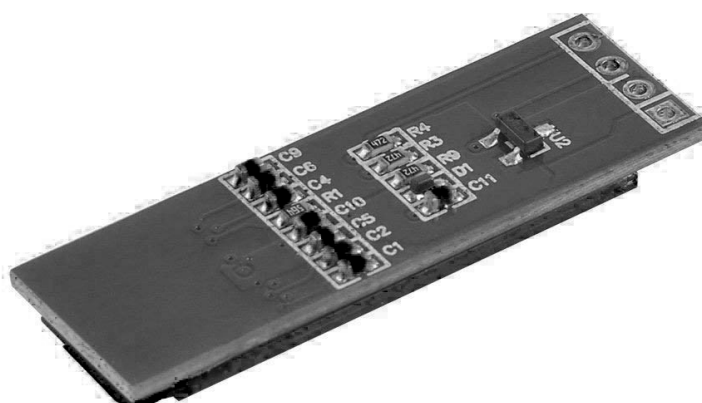
²Arduino Uno je open-source rešenje u vidu kontrolera za IoT projekte koji pruža mnoštvo mogućnosti. Više na: <https://www.arduino.cc/en/Guide/Introduction>



1.2.3 Monitor

Kao monitor za feedback sistema koristimo mali I2C³ OLED ekran 128x32 veličine 0.91 inča kao dugoročno rešenje dok će serijski port biti primarno korišćen u početnim fazama izrade.

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



1.2.4 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.

DS18B20⁴ 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 kasnije u radu).

1.2.5 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

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

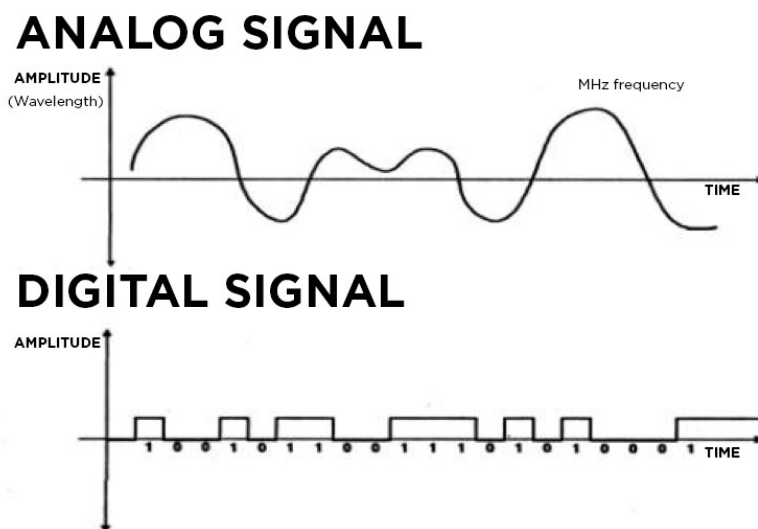
kod kojih može doći do preteranog navodnjavanja ovde se mehanizam za navodnjavanje aktivira samo kada je to potrebno. Ovo sprečava prekomerno zalivanje što dovodi do raznih problema poput:

- 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 znači 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.

1.2.6 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:

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

Iz ovoga vidimo da su nam dovoljno obični 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 ćemo uključivati u slučaju da nam temperatura predje dozvoljenu. Ovo radimo sa ciljem da unutar jedinice stvorimo negativni pritisak (izbacuje se više nego sto ulazi) kako bi obezbedili da nam vazduh izlazi samo na predviđenom mestu a ne kroz sitne pukotine.

Prebacivanjem izduva sa 9V na 12V i uključivanjem usisa na 5V vrši se brža razmena toplote dok se temperature ne vrati na dozvoljeni nivo. Kasnije u radu ćemo zaći dublje na ovu temu kada nam budu poznati izvori toplote.

1.2.7 Svetlo i zagrevanje

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 zavisan od istog kontroler utiče na relej.

Što se izbora tehnologije svetla tiče u optičaju 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 tabeli⁵ ćemo detaljno pogledati razliku u proizvodnji toplote između različitih tipova sijalica.

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 još povoljnijim.

Kada gledamo snagu svetla, preporučeno je koristiti minimum

$$35Wpo1ft^2,$$

⁵Vise o generaciji toplote na kraju rada.

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

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 ugradjenim hladnjakom.

Sada moramo uvesti termin BTU⁶. Ako znamo da je $1\text{ Watt} = 3.41\text{ BTU}$. Svetlo koje koristimo iznosi 100W pa opterećenje sistema iznosi 341 BTU/h. Sledi računanje razlike ulazne i izlazne teperature vazduha. Ako pogledamo rezultate testa sistema bez aktivnog hladjenja:

Tabela 1.3: 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 $\Delta T = 4^\circ\text{C}$.

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:

Uključivanjem ventilatora na samo 40% postizemo sniženje temperature za 2°C . Ako iskoristimo pun potencijal istog možemo održati sobnu temperaturu. Za naše potrebe temperatura od $20\text{-}30^\circ\text{C}$ je sasvim prihvatljiva što znači da je solucija za hladjenje adekvatna.

⁶British 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 $3,8^\circ\text{C}$ (Celzijusa) do $4,4^\circ\text{C}$. Vise na: <https://sr.wikipedia.org/wiki/Btu>

Tabela 1.4: Zagrevanje - aktivan izduv 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

1.2.8 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.

1.2.9 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
-

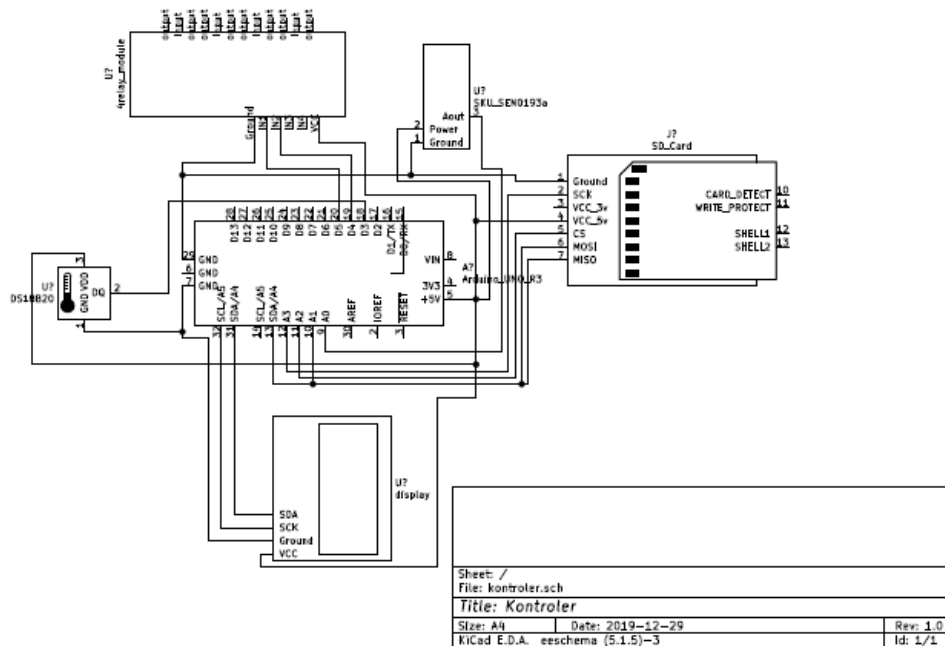
Ovo ga čini idealnim za naše uslove jer nam nudi mogućnost napajanja preko 3.3V ili 5V.

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 Hardvesrki dizajn

Posto smo odredili komponente koje odgovaraju nasim uslovima možemo stovriti grubu šemu sistema. Budići da softver(Eeschema) ne podržava breadboard(protoboard), veze poput 5V i zemlje sa arduina sam kačio na isti izlaz.



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 neophodne za rad. Kompleksnija rešenja zahtevaju re-implementaciju određenih drajvera zbog memorijskog prostora na Arduino

kontroleru. Naš sistem nije velikog razmera i iz tog razloga prostor nije limitirajući faktor.

Kada je kontroler spreman za rad pokreće se glavna loop petlja koja će kroz svreme kontinualno vršiti kolekciju podataka i procesira ih. Glavna petlja ce 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);

    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){

    // Aktivacija pumpe
    if(soil_val < SOIL_MOISTURE_TRESHOLD) digitalWrite(relay4, LOW);

    // Palimo 12V
    if(temp_val >= 30){
        digitalWrite(relay1, HIGH);
        digitalWrite(relay2, LOW);
    }
}
```

```

    }

    // Gasimo oba
    if (temp_val <= 20) {
        digitalWrite(relay1, HIGH);
        digitalWrite(relay2, HIGH);
    }

    // Palimo 9V
    else {
        digitalWrite(relay1, HIGH);
        digitalWrite(relay2, 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);

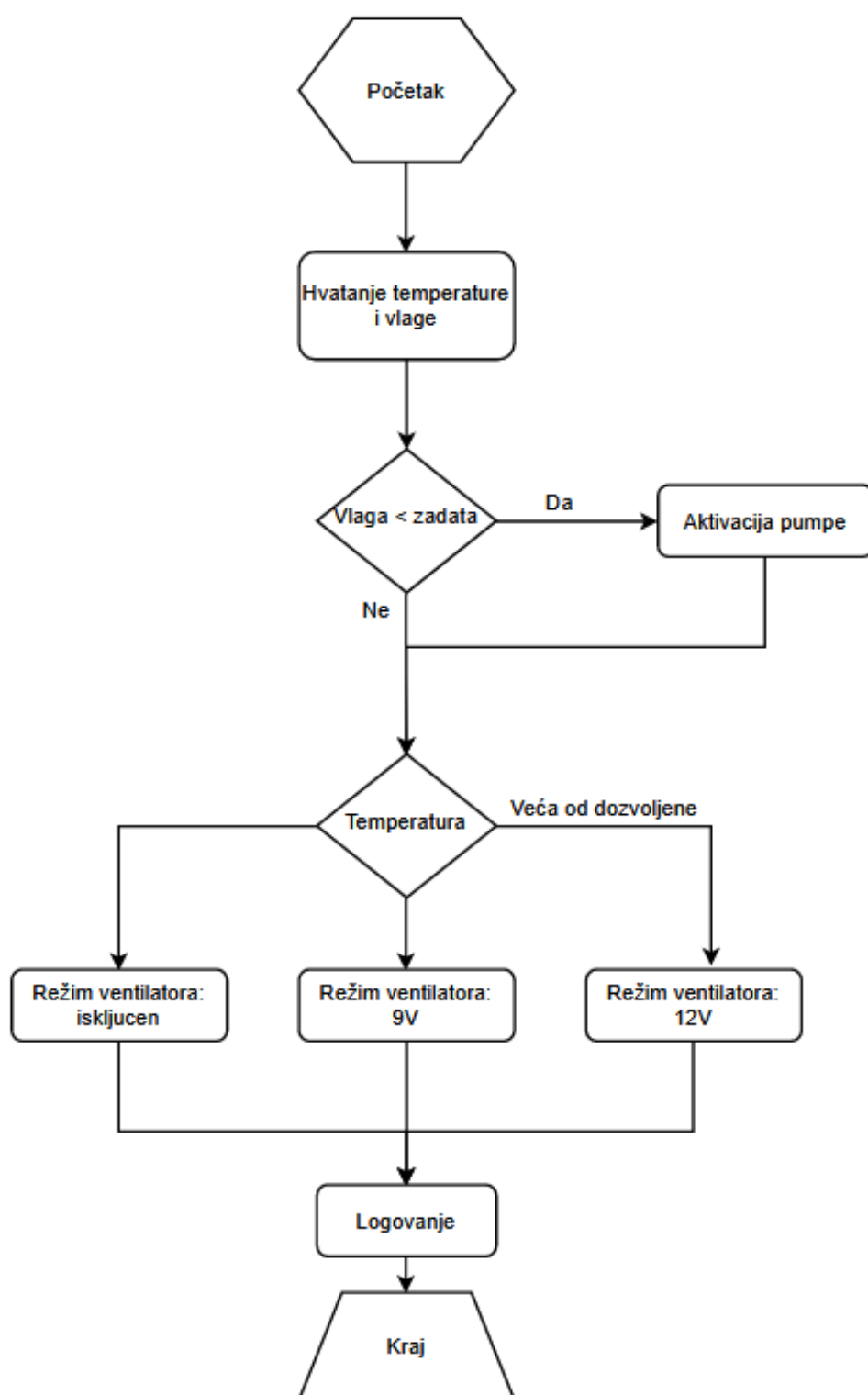
    //Ispis na ekran
    temp_out(temp_buffer);
    delay(2000);
    soil_out(soil_buffer);
}

```


Glava 2

Izgradnja modela

Kada smo definisali plan sistema možemo započeti izradu modela. Za početak odredjujemo tok operacija koje naš sistem treba da izvrši. Imajući na umu da modelujemo kontroler sistema, možemo ga posmatrati kao automat. Po definiciji automata, mi modelujemo prelaze stanja koji su elementi nekog konačnog skupa stanja Q . Stoga, osnovni tok operacija se može predstaviti flowchart-om koji izgleda ovako:

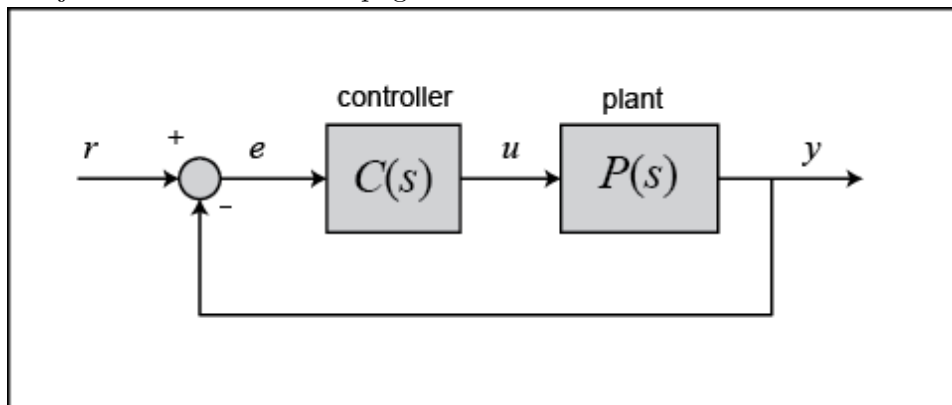


Ovim smo predstavili jedan ciklus sistema koji se izvršava kontinualno.

Sada možemo izraditi koncept modela.

2.1 Konceptualizacija modela

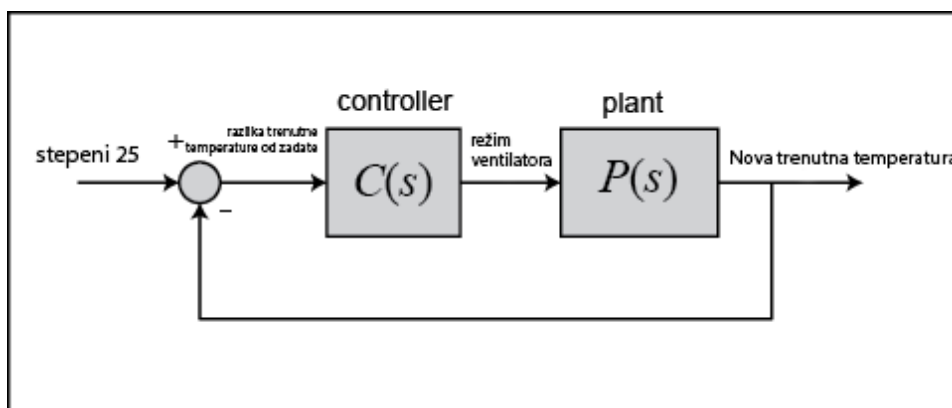
Kako bismo stvorili koncept modela moramo se vratiti na funkcije našeg sistema. Analizom funkcija i konkretnog hardvera pronalazimo uslove i ograničenja sistema koji modelujemo. Mi smo ranije u radu prešli rad određenih komponenti koje su od značaja za rad sistema, sada određujemo njihovu korelaciju unutar sistema. Ako pogledamo sliku PID kontrolera:



zaključicemo da je naš sistem idealan primer istog.

Sa leve strane imamo promenljivu r koja predstavlja idealan output sistema. Naš model će se fokusirati na regulaciju temperature stoga mi želimo da nam r bude vrednost unutar opsega 20-30°C. Uzećemo 25°C kao fiksnu vrednost.

Plant ili $P(s)$ nam predstavlja sistem koji želimo kontrolisati. Možemo primetiti da on ima ulaznu vrednost(u) koja nam utiče na izlaz y . Da bi sistem izvršio validaciju on će izlazni signal da uporedi sa zadatom vrednosti. Rezultat te operacije(e) se prosledjuje kao input kontroleru $C(s)$ koji onda na osnovu njega utiče na sistem. U našem slučaju sistem će izgledati ovako:



Sada sledi kolekcija podataka gde nam je cilj eliminacija što više nepoznatih faktora.

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. U ovom radu, prilikom planiranja mi smo izvršili kolekciju i obradu podataka da bi stvorili plan stoga nam preostaje prevodjenje modela i izrada simulacije. Jedino što nam preostaje jeste sistem za zalivanje koji ne utiče direktno na ambijent mašine, njega ćemo posebno simulirati putem takozvanog pilotnog projekta.

2.2.1 Regulacija zalivanja - pilotni projekat

Za projekat sam pronašao saksiju adekvatnih veličina za sistem. Saksija je napunjana sa $2/3$ zemlje i $1/3$ perlita čija je uloga da pomogne oko drenaže (ako za to bude potrebe). Do saksije je sprovedeno crevo po obodu sa sitnim rupicama za zalivanje. Sada preostaje razmatranje pumpe:

Budući da je naš sistem relativno malih razmera koji podržava jednu biljku neće nam biti potrebna velika pumpa. Ako se okrenemo akvaristici možemo pronaći sasvim idealnu pumpu za projekat. Zalivanje vrši se putem pumpe za manje akvarijume. Po specifikaciji proizvođača protok vode iznosi $Q = 150l/h$. Testiranjem je utvrđeno sledeće:

Tabela 2.1: Rad pumpe za vodu

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

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

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.

2.3 Prevodjenje modela

Kao alat za simulaciju korišćemo Matlab verziju R2018a uz oslonac na Simulink za grafičko predstavljanje podataka. Prevodjenje modela može se podeliti na definisanje parametara okruženja i izradu logike kontrolera.

2.3.1 Definisanje parametara okruženja

U ovom koraku kreiramo model podataka za simulaciju

Za početak je neophodno odrediti ukupno površinu okruženja sa kojom radimo. Ako se vratimo na specifikaciju jedinice možemo uvideti da su dimenzije $30 * 30 * 60$ stoga se može modelovati na sledeći način:

```
% Definisanje okruzenja

% length = 35 cm
len = 35;
% width = 35 cm
wid = 35;
% height = 60 cm
ht = 60;

% ukupna površina
wallArea = 2*len*ht + 2*wid*ht
```

Da bismo odredili gubitke sa okolinom, moramo odrediti termodinamička svojstva naseg okruženja. Nas konkretno zanima izolacija, a može se predstaviti kao razlika debljine zida i koeficijenta propusta:

```
% -----
% Definisanje izolacije
% -----
kWall = 0.038*3600; % jedinica za vreme je h

% debljina zida
LWall = .02;

% ukupno
RWall = LWall/(kWall*wallArea);
```

Pored ovoga ostaju nam promenljive poput protoka vazduha, gustine i mase koje se računaju na sledeci način:

```
% c = cp of air (273 K) = 1005.4 J/kg-K
c = 1005.4;
```

```
Mdot = 3538; % jedinica h za ukupan protok vazduha
```

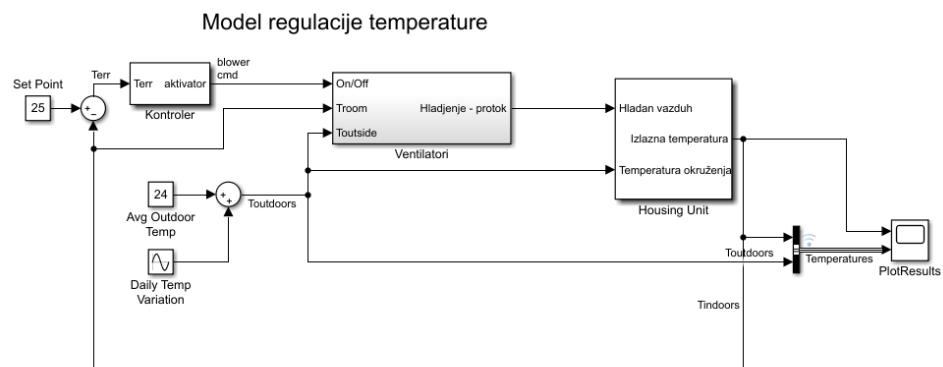
```
% gustina vazduha na 0 nadmorske visine = 1.2250 kg/m^3
densAir = 1.2250;
```

```
% ukupna masa
M = (len*wid*ht*densAir);
```

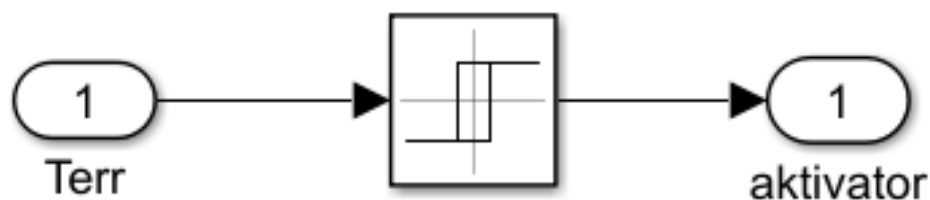
```
% TinIC = inicijalna temperatura = 24 deg C
TinIC = 24;
```

2.3.2 Izrada logike

Sada prelazimo na Simulink deo modela. Za naš slučaj potrebna su nam tri podsistema unutar našeg PID kontrolera za regulaciju rada. Simulacija je osmišljena tako da smo uzeli rad ventilatora na 70% efektivne snage kako bismo dobili realno sliku sistema. Celokupam sistem se može modelovati na sledeći način:



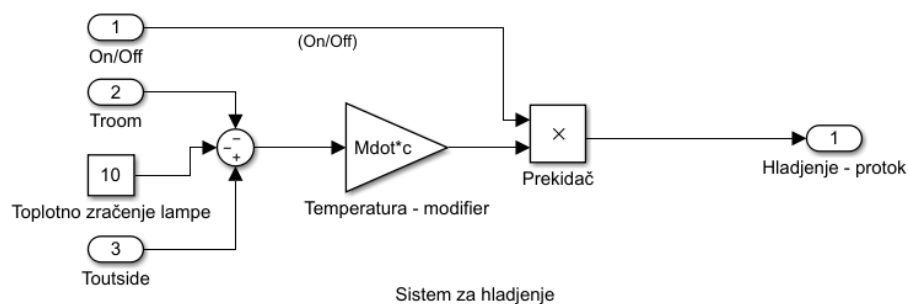
Podlašavamo željenu temperaturu korišćenjem Set Point konstante koju ćemo porediti sa izlazom sistema. Greška tog poredjenja ulazi u nas prvi podsistem koji predstavlja logicku kapiju za pokretanje regulacije. Ako pogledamo dizajn ovog sistema:



Kontroler - Aktivator ventilacije

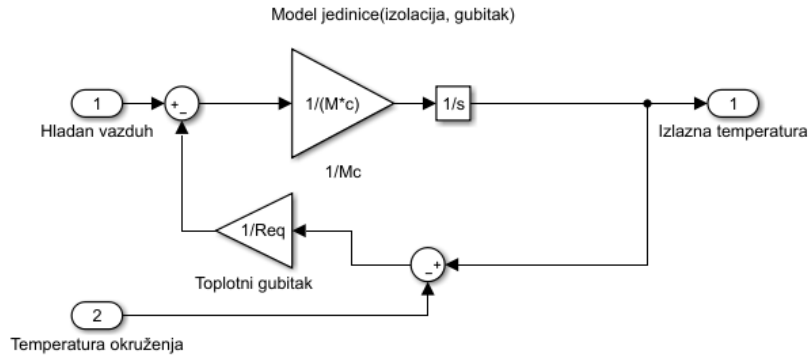
Vidimo da je on zapravo običan relej sistem koji se aktivira na određenom toplotnom rangu. U našem slučaju dozvolili smo varijaciju od 3 stepena. Ovaj relej daje signal sledecem sistemu koji služi za razmenu toplote sa okruženjem tako što vraća 1 za slučaj da je potrebna korekcija ili 0 ako je temperatura u dozvoljenim granicama.

Sada prelazimo na sistem za hladjenje:



Uzimamo trenutnu vrednost okruženja na koju dodajemo zagrevanje pomoću lampe. Od te vrednosti oduzimamo trenutnu spoljnu temperaturu koju će ventilatori ubacivati. Nakon toga računamo toplotni modifier (promenu imajući u vidu karakteristike vazduha i protoka ventilatora). Prekidač samo množi ova dva ulaza što znači da ako je aktivator u isključenom režimu output vazduha je 0.

Sledeći sistem je zadužen za gubitke toplote kroz radijaciju.



Ovo se postiže uzimajući u obzir temperaturu vazduha koji se ubacuje i izolaciju sistema. Toplotni gubitak se računa tako što od izlaza sistema oduzmemo toplotnu vrednost okruženja i provučemo je kroz filter koji dobijenu vrednost deli sa izolacijom. Ovo će se primenjivati pri svakom ciklusu.

Nakon ovoga ostaje nam samo da vratimo izlaznu vrednost na početak ciklusa radi računanja greške i ispišemo je na ekran.

2.4 Verifikacija

Ovde ćemo se pozabaviti ispravnosti naseg modela. Proći ćemo kroz sisteme i verifikovati da li se implementacija poklapa sa matematičkim formulama. Ovaj korak je krucijalan za simulaciju jer obezbeđuje ispravnost rezultata.

Za verifikaciju simulacije moramo sagledati simulaciju sa stanovišta podsistema. Budući da je kontroler jako jednostavan njega ćemo ignorisati.

2.4.1 Sistem za kontrolu ventilatora

Ako pogledamo podsistem za ubacivanje vazduha, promene temperature možemo izračunati sledećom formulom:

$$\frac{\delta Q}{\delta t} = (T_{room} - T_{current} - kT_{heatgain}) * M_{Dot} * c$$

M_{Dot} - protok vazduha

Proverom vidimo da se naš dizajn poklapa sa formulom, stoga možemo reći da zadovoljava uslove.

2.4.2 Obračun gubitaka

Sada možemo preći na podsistem vezan za obračun temperature u jedinici. Ovde primenjujemo dve formule:

$$\frac{\delta Q}{\delta t}_{loses} = \frac{(T_{room} - T_{out})}{Req}$$

gde je

$$\frac{\delta T_{room}}{\delta t} = \frac{1}{M_{air} * c} * \left(\frac{\delta Q_{ventilatora}}{\delta t} - \frac{\delta Q_{loses}}{\delta t} \right)$$

Req - vrednost izolacije jedinice

M_{air} - količina vazduha u jedinici

c - termalni kapacitet vazduha pri konstantnom pritisku.

Ako prodjemo kroz šemu vidimo da je i ovaj deo sistema korektno modelovan.

2.5 Validacija

Validnost modela ocenjuje se pomocu nivoa validnosti. Nivo validnost predstavlja tačnost sistema i utvrđuje se ponavljanjem jednog skupa ishoda. Potrebno je imati na umu da ni jedan sistem se ne može savršeno prezentovati, samo je potrebno da nivo poverenja(tačnost) modela ispunjava kriterijum.

Ukoliko sistem zadovoljava naše potrebe, možemo reći da je važeći ali da bismo to utvrdili neophodno je izvršiti inicijalne simulacije. Nakon analize rezultata ukoliko nisu ispunjeni kriterijumi neophodno je vratiti se na model sistema i detaljno analizirati tok.

Glava 3

Izvršavanje simulacije

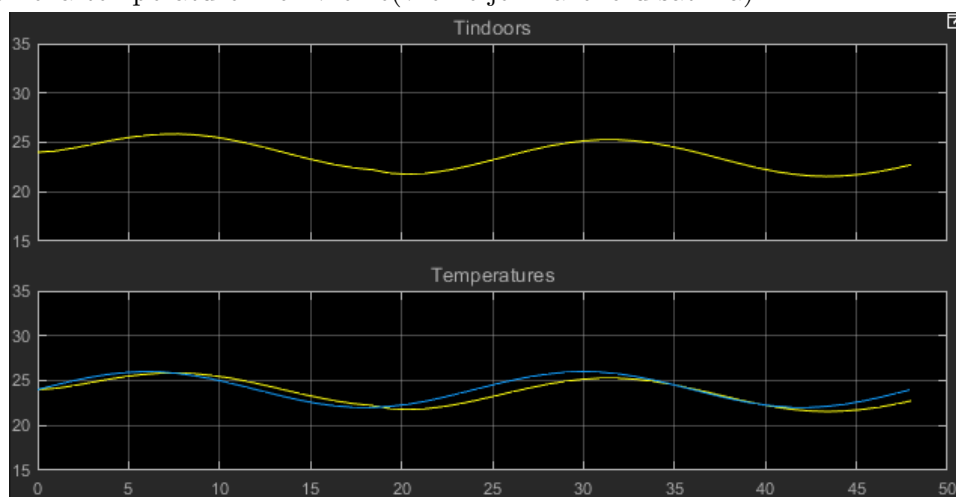
Budući da smo za sisteme zalivanja i svetlosnog ciklusa koristili pilotne projekte jer su po prirodi jako jednostavni nije potrebno izvršiti simulaciju rada. Iz tog razloga fokusiraćemo se samo na simulaciju sistema za regulisanje temperature.

3.1 Dizajn eksperimenta

Eksperiment je dizajniran po specifikaciji projekta tako da smo

3.2 Izvršavanje i analiza

Pokretanjem šeme iz Simulink-a dobijamo sledeću sliku gde je prikazana promena temperature kroz vreme (vreme je izraženo u satima):



Kao što možemo videti iz rezultata, izlaz sistema odgovara našim potrebama. Budući da smo zadovoljni sa rezultatom nije potrebno vršiti izmene.

3.3 Dodatna izvršavanja

Glava 4

Implementacija

4.1 Dokumentacija i izveštaj

Bibliografija

- [1] www.agropress.org - Koje su prednosti plasteničke proizvodnje?
<https://bit.ly/35PbbkT>
- [2] General Study for using LED to replace traditional lighting devices -
Y.K. Cheng, K.W.E.Cheng,
<https://www.researchgate.net/publication/224694534>
- [3] Thermal and Energy Management Based on Bimodal Airflow-
Temperature Sensing and Reinforcement Learning - Zhen Zhang, Cheng
Ma, Rong Zhu,
<https://www.researchgate.net/publication/321344620>
- [4] Thermal Model of a House
<https://ch.mathworks.com/help/simulink/slref/thermal-model-of-a-house.html>