



Šolski center Celje

Srednja šola za kemijo, elektrotehniko in računalništvo

NADGRADNJA NADZORA MIKROKLIME V PROSTORU S KRMILNIKOM ARDUINO

RAZISKOVALNA NALOGA

AVTORJA

Andraž Špan
Rok Lešnik

MENTOR

Andrej Grilc

Celje, marec 2019

KAZALO VSEBINE

KAZALO SLIK.....	4
ZAHVALA.....	5
POVZETEK	6
ABSTRACT	6
1 UVOD	7
1.1 OPIS SNOVANJA IN GRADNJE SISTEMA V PREJŠNJEM ŠOLSKEM LETU	7
1.2 PREDSTAVITEV PROBLEMA.....	9
1.3 HIPOTEZE	10
1.4 RAZISKOVALNE METODE	11
2 TOPOLOGIJA SISTEMA	13
2.1 SERVERSKA OMARICA.....	13
2.1.1 SPLET.....	13
2.1.2 RUTER	14
2.1.3 POE SWITCH.....	14
2.1.4 RASPBERRY PI	14
2.2 KRMILNI SISTEM	14
2.2.1 ARDUINO	14
2.2.2 TEMPERATURNI SENZOR	15
2.3 VREMENSKA POSTAJA	16
3 STANJE SISTEMA PRED NADGRADNJO	17
4 SNOVANJE REŠITEV.....	19
5 NADGRADNJA STROJNEGA DELA	21
5.1 TEMPERATURNI SENZOR	21
5.2 USB KONEKTORJI	21
5.3 NAPAJALNIK ZA VREMENSKO POSTAO.....	25
5.4 VREMENSKA POSTAJA	26
5.5 UPS.....	27
6 PROGRAM.....	28
6.1 ARDUINO PROGRAM	28
6.1 RASPBERRY PI PROGRAM	32
6 SPLETNA STRAN	35
8 MOŽNOSTI NADGRADNJE V PRIHODNOSTI.....	37
8.1 NEREALIZIRANI OZ. OVRŽENI PREDLOGI UPORABNIKOV	38

9 UGOTOVITVE.....	39
10 ZAKLJUČEK	40
VIRI IN LITERATURA.....	41
PRILOGE	42
IZJAVA.....	42
NAVODILA ZA UPORABO	43

KAZALO SLIK

Slika 1: Pogled na strešne kupole na strehi.....	7
Slika 2: Arduino.....	8
Slika 3: Vremenska postaja ob obilnem sneženju	9
Slika 4: Nalaganje Arduino programa	10
Slika 5: Snovanje rešitev	11
Slika 6: Prototip za testiranje delovanja.....	12
Slika 7: Temperaturni senzor DHT22	15
Slika 8: Vremenska postaja.....	16
Slika 9: Pogled v dozo	17
Slika 10: Pogled na streho pred nadgradnjo	18
Slika 11: Merjenje hitrosti vetra	19
Slika 12: Merjenje moči motorja z merilnikom Fluke.....	20
Slika 13: Izpis na serijskem vmesniku	20
Slika 14: Temperaturni senzor v prostoru	21
Slika 15: Shema vezja za USB konektor	22
Slika 16: Tiskano vezje za USBkonektor.....	22
Slika 17:Rrezkanje tiskanega vezja	23
Slika 18: Izrezkano tiskano vezje	23
Slika 19: Priprava na premaz z kolofonijo	24
Slika 20:Spajkanje USB konektorja	24
Slika 21: Testiranje delovanja USBkonektorja	25
Slika 22: Napajalnik	25
Slika 23: Končni izgled vremenske postaje	26
Slika 24: Izbran UPS	27
Slika 25: Definiranje spremenljivk za program	29
Slika 27: Zagon programa	30
Slika 26: Zaznavanje pritiska tipke in pošiljanje UDP sporočila.....	30
Slika 28: Zagonski del programa vremenske postaje	31
Slika 29: Zaznavanje dežja in vetra.....	31
Slika 30: Priprava programa za odpiranje in zapiranje ob določeni uri.....	33
Slika 31: Trenutno napisan program za zapiranje kupol v primeru dežja ali vetra	33
Slika 32: Primer izpisa dogajanja v programu.	34
Slika 33: Domača stran spletne strani	35
Slika 34: Zavihek nadzor kupol	35
Slika 35: Zavihek omogočenost sistema	36
Slika 36: Predlogi	37
Slika 37: Sestanek z uporabniki	38
Slika 38: Upravljanje serverja	40

ZAHVALA

Pri izdelavi raziskovalne naloge se zahvaljujeva najinemu mentorju, Andreju Grilcu, univ.dipl.inž.el. za pomoč pri snovanju naloge. Zahvaljujeva se ravnatelju, g. Mojmirju Klovarju za pridobitev materiala, namembo sredstev in dovoljenje za vgradnjo sistema v šolsko infrastrukturo. Zahvaljujeva se tudi tehničnim delavcem šole za pomoč in potrpežljivost pri nastajanju naloge in preizkušanju sistema.

Posebej se zahvaljujeva tudi razredniku, prof. Gregorju Kramerju, ki nama je omogočil izostajanje od pouka in pomagal z nasveti o delovanju krmilja, in Janku Holobarju za nasvete in pomoč pri izdelavi tiskanega vezja.

Zahvala gre tudi dr. Tanji Jelenko, za lektoriranje naloge ter prof. Ireni Sojč za pregled povzetka v angleščini.

POVZETEK

Cilj raziskovalne naloge je bil nadgraditi nadzor mikroklima v prizidku šole. V šolskem letu 2017/2018 sva izvedla popolno avtomatizacijo strešnih kupol, kar sva v preteklem šolskem letu opisala tudi v raziskovalni nalogi: "Nadzor mikroklima v prostoru z mikrokrmlnikom Arduino". Tekom delovanja sistema so se pojavile manjše pomanjkljivosti in dodatne želje uporabnikov po spremembah delovanja avtomatike. Pomanjkljivosti v programu sva odpravila, upoštevala pa sva tudi želje uporabnikov ter dodala nekaj tehnoloških dodatkov. Vse to vam želiva predstaviti v najini raziskovalni nalogi.

Na počutje v prostoru vpliva veliko dejavnikov. Ena glavnih pa sta temperatura in kakovost zraka, kar je še posebej pomembno ko govorimo o učilnicah, kjer se vsakodnevno zamenja ogromno dijakov, se zavemo resnosti problema. Za to sva se s sošolcem odločila za raziskovalno nalogo v kateri bi razvila iz izvedla nadgradnjo nadzora mikroklima v prizidku šole.

ABSTRACT

The aim of this research work was to study the upgrading of the microclimate control system in the classrooms of the school building extension. Based on the research, we planned and carried out the complete automation of the roof domes in the classrooms of the school building extension, in the school year 2017/2018. We used the Arduino microcontroller.

There are many factors influencing students feeling well in the classroom. As the most important factors are certainly temperature and air quality me and my classmate decided to do a research task on how to develop and carry out the upgrading of the microclimate control system in the classroom with the Arduino controller.

1 UVOD

Na osnovi raziskovalne naloge, ki sva jo izvedla preteklo šolsko leto (2017/2018) sva izvedla nadgradnjo avtomatizacije strešnih kupol s pomočjo mikrokrmilnika Arduino. Uporabniška izkušnja sistema je bila zadovoljiva vendar so imeli uporabniki veliko želj po spremembah delovanja nekaterih funkcij. Opazila ter kasneje odpravila pa sva tudi nekaj pomenljivosti v strojni opremi sistema. Z uporabniki ter ravnateljem sva se nekajkrat sestala da smo dorekli vse spremembe ter želje. Pojavilo se je tudi nekaj nerealnih predlogov, kar pa bova opisala v nadaljevanju.



Slika 1: Pogled na strešne kupole na stehi

1.1 OPIS SNOVANJA IN GRADNJE SISTEMA V PREJŠNJEM ŠOLSKEM LETU

V drugem letniku smo pri predmetu "Upravljanje s programljivimi napravami (UPN)" razmišljali, kako bi lahko izboljšali funkcionalnost naše šole, da bi sledili tehnološkim trendom. Prišla sva do ideje, da bi v novem prizidku šole izvedli avtomatizacijo strešnih kupol glede na temperaturo, zunanje vremenske vplive in časovni interval. Poleg tega pa bi imel uporabnik na voljo še ročni nadzor preko tipk. Pričela sva z izdelavo delovnega načrta, seznamom potrebnih komponent in načrtom dela.

Z delom sva pričela novembra leta 2017. Najino raziskovanje se je začelo z zbiranjem koristnih informacij, ki so nama kasneje pomagale pri izdelavi celotnega krmilja. Na začetku sva se usmerila k merjenju temperature in vlage v prostoru. Ugotovila sva, da je kakovost zraka v učilnicah zelo slaba. V nadaljevanju sva spremljala navade prezračevanja v tem delu šole in ugotovila, da kupole ostajajo odprte. Opazila sva tudi, da so kupole odprte kljub neugodnim vremenskim razmeram, kar predstavlja nevarnost za poplavo ter poškodbo šolske opreme.

Prav tako sva ugotovila, da se dijaki "igrajo" z njimi, saj so tipke prosto dostopne in jih lahko upravlja kdorkoli, kar predstavlja problem ob slabem vremenu. Ugotovila sva tudi, da ostanejo odprte čez noč, čez vikend ...

Najprej sva določila stanje in zapisala ugotovitve, nato sva se lotila še zahtev ravnatelja in učiteljev, ki poučujejo v tistem delu šole. Prišla sva do zanimivih in inovativnih idej, ter tudi varnostnih in uporabnih zahtev, ki sva jih kasneje vključila v program.



Slika 2: Arduino

Pri izdelavi sva za nadzor posamezne kupole ter vremenske postaje izbrala mikrokrmlnik Arduino Leonardo ETH, program zanj sva pisala v programskem okolju Arduino IDE (Arduino Integrated Development Environment), program za Raspberry Pi, ki nadzira celoten sistem pa je napisan s programskim jezikom Python.

V prejšnji raziskovalni nalogi sva si postavila naslednje hipoteze, vse pa sva tudi potrdila:

- Kakovost zraka je brez aktivnega nadzora slabša.
- Aktiven nadzor mikroklima z mikrokontrolerjem Arduino bo bistveno izboljšal kakovost zraka v prostoru.
- Za še boljši nadzor nad mikroklimo bi bil potreben nadzor tudi nad ogrevalnim sistemom.
- Aktiven nadzor z mikrokontrolerjem Arduino bo izboljšal tudi varnost opreme in infrastrukture.

1.2 PREDSTAVITEV PROBLEMA

Glavni del vsakega sistema za nadzor klime v prostoru so senzorji, ki posredujejo podatke krmilju. Merilo lahko marsikaj, povezava z centralnim krmilnikom pa je lahko izvedena na mnogo načinov, izbrala sva povezavo preko UTP Ethernet kabla, preko katerega se mikrokontrolerji tudi napajajo.

Kot je bilo že nekajkrat omenjeno so se pojavile želje ter zahteve po nadgradnji sistema, ki sva jih opravila. Uporabniki so želeli popoln ročni nadzor nad sistemom, zahtevali pa so tudi natančna navodila za uporabo.

Senzorji za zaznavanje temperature so zaradi namestitve na neprimerenem mestu v prostoru (pod stropom z namenom, da jih dijaki nebi poškodovali) odčitavali temperaturo, ki ni bila realna. Senzor za hitrost vetra na vremenski postaji ni bil pravilno umerjen, senzor za dež pa je zaznaval stanje "dež" tudi, ko ni deževalo, ampak je bilo megleno ali pa je snežilo (kljub grelcu se sneg ni stopil).



Slika 3: Vremenska postaja ob obilnem sneženju

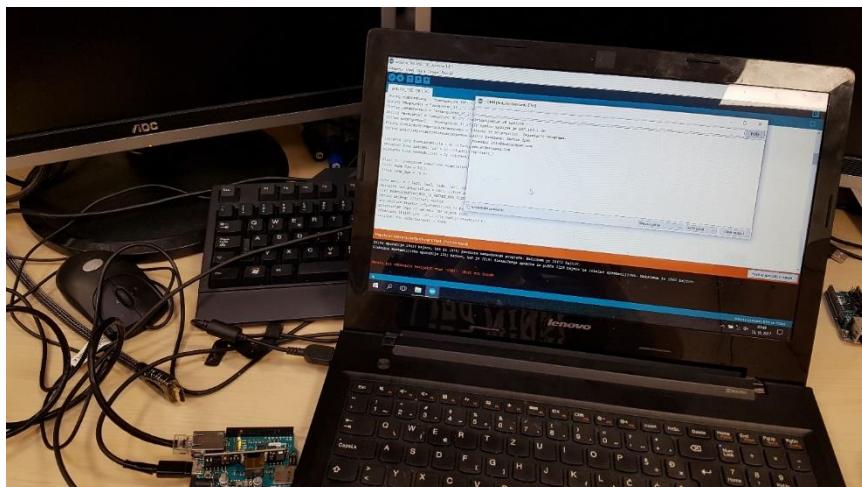
Pojavljal se je problem pri nadzoru posamezne kupole in pri nadgradnji te, saj je vedno znova potrebno odpreti ohišje krmilja, kar je zelo zamudno.

Težava se je pojavila, ko je prišlo do izpada električne energije. Po ponovni vzpostavitvi napajanja se program na RasberryPi-ju ni samodejno zagnal, kar pomeni, da se kupole v primeru vremenskih vplivov niso samodejno zaprle. Hkrati se je pojavila potreba po UPS-u, saj smo želeli, da se v primeru izpada napajanja vse kupole samodejno zaprejo.

1.3 HIPOTEZE

Pred začetkom izdelave raziskovalne naloge sva si postavila naslednje hipoteze, ki jih bova skušala v nadaljevanju naloge potrditi oz. ovreči:

- Po izpolnitvi uporabniških želj in zahtev bo uporabniška izkušnja boljša, hkrati se bo izboljšalo mnenje o sistemu.
- Premik senzorja temperature bo omogočil zaznavanje realnejše vrednosti temperature.
- Poenostavitev programa bo sprostila prostor na mikrokrmlniku.
- UPS bo izboljšal varnost sistema oz. prostora.
- Vgradnja USB priključkov bo olajšala morebitne posodobitve programa v prihodnosti ter lokalni nadzor delovanja.
- Zapiranje ob 20. uri bo preprečilo morebitne nevšečnosti zaradi vremena.

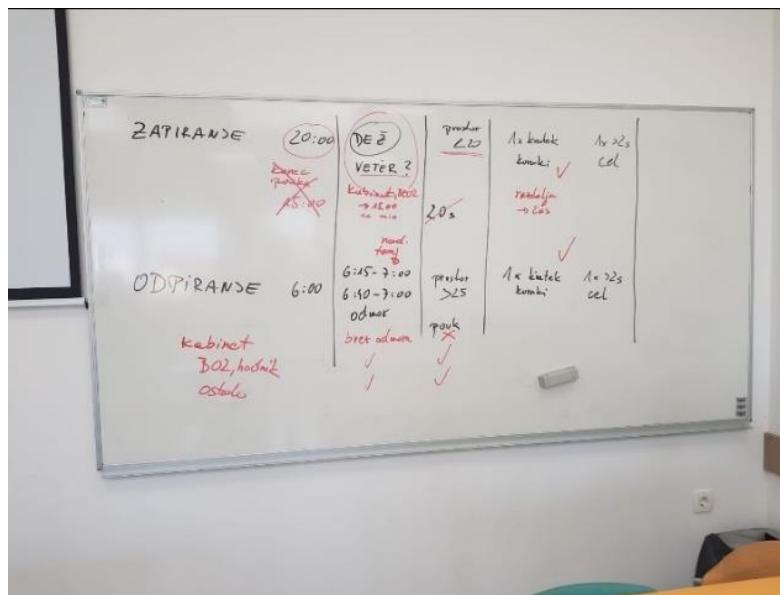


Slika 4: Nalaganje Arduino programa

1.4 RAZISKOVALNE METODE

Na začetku sva si postavila cilj ugotoviti čim bolj realno stanje sistema, preučiti, kako se sistem obnaša v različnih situacijah in od uporabnikov pridobiti informacije o morebitnih težavah ki jih zaznavajo in slišati njihove predloge za nadgradnjo. Nekajkrat sva imela sestanek z ravnateljem in profesorji da smo uskladili vse potrebne podrobnosti glede delovanja, kot so npr. odpiranje ob določeni uri, uporaba senzorjev ...

Pred začetkom nadgradnje sva uporabnike prosila, da so rešili kratko anketo o zadovoljstvu s sistemom. Isto anketo sva jim posredovala tudi na koncu in rezultati so pričakovani (prva hipoteza).

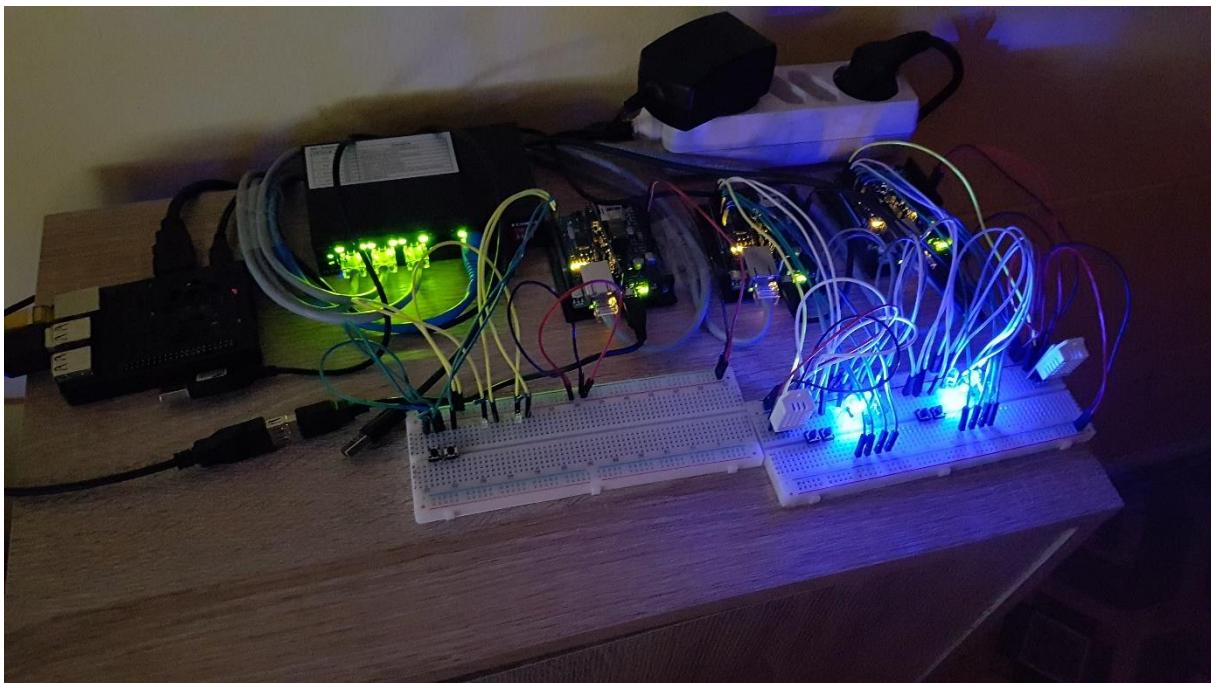


Slika 5: Snovanje rešitev

Izmerila sva temperaturo v prostoru in jo nato primerjala s temperaturo, ki jo je izpisoval Arduino in ugotovila, da je razlika med njima zelo velika. Več je opisano v poglavju *Snovanje rešitev*.

Na podoben način sva izmerila tudi hitrost vetra, vendar je bilo s tem nekaj manj težav, saj je bilo potrebo hitrost le pravilno izmeriti in nato senzor umeriti na pravilno hitrost.

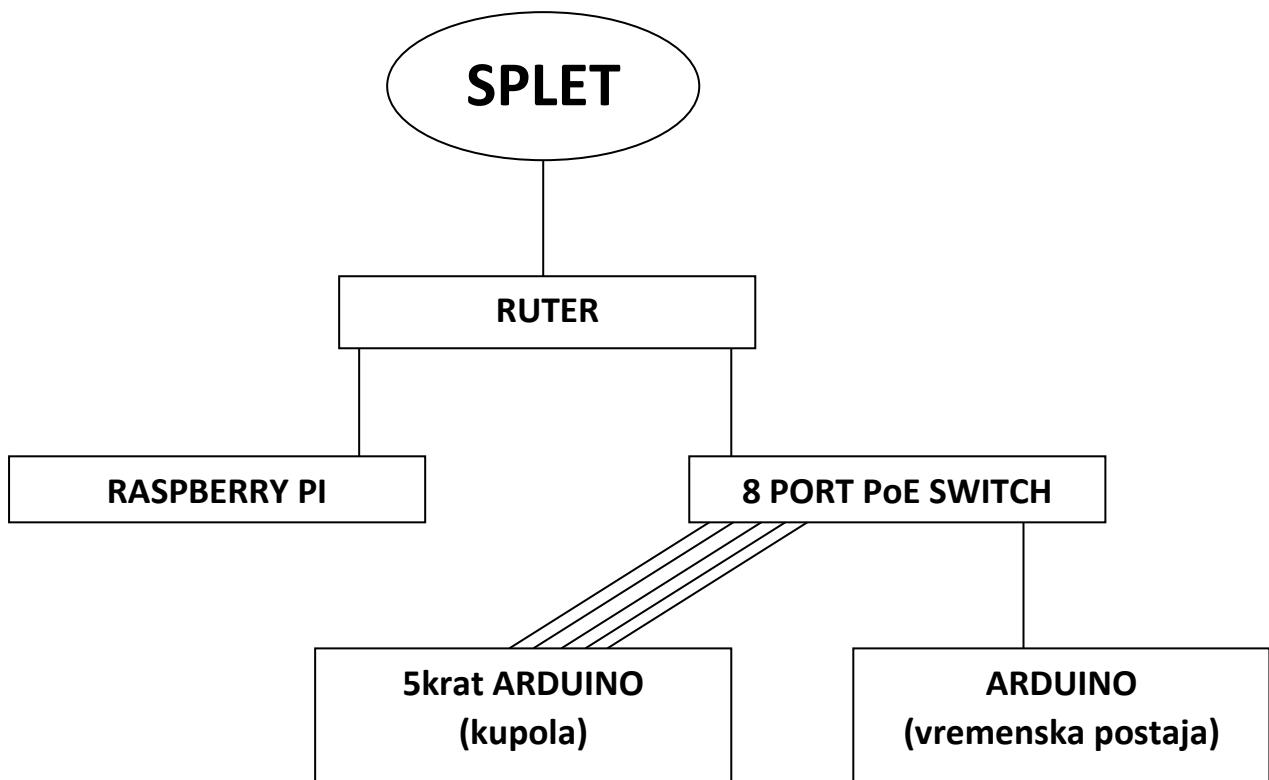
Preden sva program prenesla v sistem, sva posodobitev simulirala na prototipni ploščici z uporabo tipkal, led sijalk in temperaturnega senzorja. Tipkala so predstavljala tipke v učilnici in senzor za veter in dež, led sijalke so predstavljale stanje posamezne kupole, odprto stanje je simulirala zelena, zaprto pa rdeča led sijalka. Temperaturni senzor pa je predstavljal dejanski senzor temperature v prostoru.



Slika 6: Prototip za testiranje delovanja

2 TOPOLOGIJA SISTEMA

Sistem je zasnovan na naslednji način.



2.1 SERVERSKA OMARICA

Serverska omarica je nameščena na steni v kabinetu (BE-01). V njej se nahajajo povezava do spletu, ruter, PoE switch, napajalnik za vremensko postajo in Raspberry Pi. Omarica je v steno pritrjena s posebnimi vijaki za suho montažno steno in zaklenjena s ključem. Napajanje je izvedeno preko lastnega inštalacijskega odklopnika v razdelivcu.

2.1.1 SPLET

Sistem je povezan v svetovni splet preko UTP cat5e vodnika. Povezava je namenjena oddaljenemu dostopu do nadzora sistema in dostopanju do njegovih podatkov. Dostop je trenutno mogoč le lokalno.

[2.1.2 RUTER](#)

Za uporabo sva izbrala Linksys E900, ki je bil cenovno ugoden in je zadostil potrebam sistema.

[2.1.3 POE SWITCH](#)

Za uporabo sva izbrala switch Tenda TEF1108P, ki je 8 portno namizno stikalo s katerim sva omogočila mrežno povezavo in komunikacijo vseh z VLAN podporo, prav tako pa sva s tem poskrbela za napajanje vseh mikrokontrolerjev v sistemu. Ker uporabljava industrijski PoE je napetost približno 50V, kar pomeni zanemarljive izgube na vodnikih. To v primerjavi z ločenimi napajalniki v krmilni enoti pomeni večjo energetsko učinkovitost.

[2.1.4 RASPBERRY PI](#)

Uporabila sva računalnik Raspberry Pi 3 je zmogljiva različica računalniškega sistema v velikosti kreditne kartice. Raspberry zaganja in shranjuje podatke prek kartice microSD.

Ima lastno napajanje preko micro USB prikučka.

[2.2 KRMILNI SISTEM](#)

Krmilni sistem je sestavljen iz mikrokontrolerja arduino, temperaturnega senzorja ter dvokanalnega releja za nadzor kupol.

[2.2.1 ARDUINO](#)

Za mikrokontroler sva uporabila Arduino Leonardo ETH, saj je to edini Arduino, ki je trenutno na trgu z vgrajenim internetnim konektorjem RJ45 in ima možnost napajanja preko internetnega kabla (Power over Ethernet–PoE).

Ima 20 digitalnih vhodov oziroma izhodov, od teh se jih 7 lahko uporablja kot PWM izhod, 12 pa kot analogni vhod. 16 MHz oscilator, RJ45 povzavo, microUSB konektor ter čitalec mikro SD kartic, kar nama omogoča dodatno shrambo za program in knjižice (naštete so le za najin projekt potrebne informacije).

2.2.2 TEMPERATURNI SENZOR

S temperaturnim senzorjem meriva temperaturo, senzor sicer omogoča še merjenje vlage vendar sva ugotovila, da za šolske prostore ustreza le merjenje temperature, saj to najbolj vpliva na klimo v razredu. V osnovi sva že lela digitalni senzor. Izbirala sva med DHT11 in DHT22. Sta ena redkih senzorjev, ki sta digitalna in cenovno ugodna, posledično pa tudi zelo razširjena. DHT11 ima veliko odstopanje pri merjenju temperature, za to sva se odločila za DHT22, njegovo odstopanje je le +/- 0,5 °C. Največja razlika med senzorjem je po podatkih proizvajalca v odstopanju pri merjenju vsebnosti vlage v prostoru, ta podatek pa za naju ni bil pomemben. Senzor ima temperaturno območje od -40 °C do 80 °C, kar pomeni, da je senzor primeren za najino območje uporabe.

Porabi 2.5 mA, podatke pa pošlje vsaki 2 sekundi. Ima 4 pine, od tega en ni povezan nikamor in je namenjen le fiksiranju. Senzor lahko neposredno prispajkamo na vezje ali pa ga povežemo s tremi vodniki. Uporabila sva slednjo opcijo saj se senzor v prostoru nahaja nekaj metrov stran od krmilne enote.



Slika 7: Temperaturni senzor DHT22 (<https://core-electronics.com.au/dht22-temperature-sensor-module.html>)

2.3 VREMENSKA POSTAJA

Vremenska postaja deluje na 24V AC/DC in porabi 0,2A toka. Njene mere so 80x240x200 mm, tehta pa 1,5 kg. Sestavljena je iz ohišja, lastnega vezja, nosilca, vetrne turbine in površine senzorja za dež na vrhu ohišja. Omogoča merjenje vetra do 35m/s in dežja. V primeru vetra ali dežja se aktivira preklopni kontakt. Senzor za dež deluje na principu prevodnosti površine senzorja. Postaja omogoča nastavitev občutljivosti ter zakasnitve.

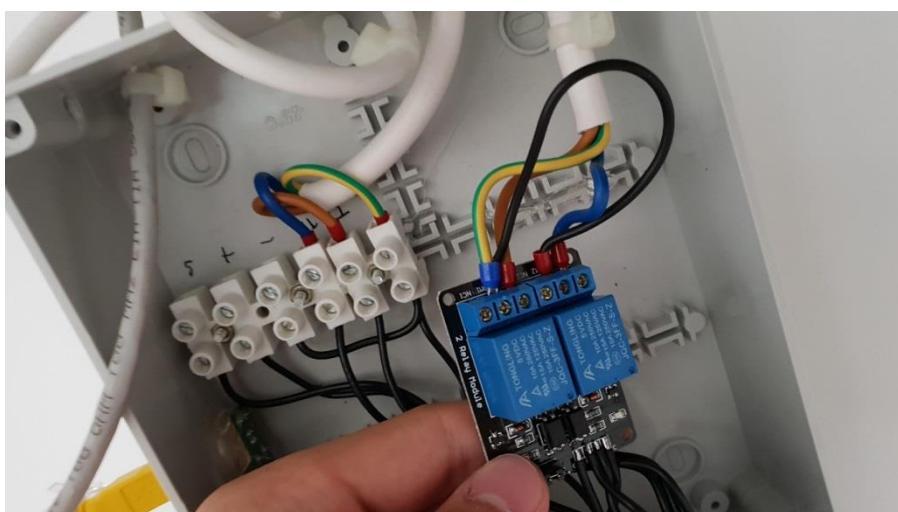


Slika 8: Vremenska postaja (<https://www.windowdrives.com/shop/elektrozubehoer/wind-regenmelder/534/windowmaster-wind-und-regensor-wla-330>)

3 STANJE SISTEMA PRED NADGRADNJO

Najprej sva preverila, kakšno je bilo stanje sistema pred nadgradnjo. Ugotovila sva naslednje:

Vremenska postaja ni delovala idealno, saj je že ob najmanjši rosi sporočala, da dežuje, prav tako so se težave pojavile ob sneženju, saj grelci v vremenski postaji niso stalili snega in je vremenska postaja stalno sporočala, da dežuje, kljub temu da je že prenehalo snežiti. Druga težava se je pojavila pri senzorju vetra, saj je vremenska postaja oddajala vrednost vetra kot analogno napetost od 0 V do 10 V, kjer 0 V pomeni 0 m/s in 10V 35 m/s.



Slika 9: Pogled v dozo

Glede na te podatke in upoštevajoč delilnik napetosti, ki je potreben, ker Arduino podpira napetost na analognem vhodu le do 5V ter 10-bitni ADC pretvornik, ki se nahaja v Arduinu sva izračunala mejno vrednost hitrosti vetra. Ker ni nujno, da je meritnik linearen, natančnost ni popolna, potrebna je kalibracija meritnika hitrosti vetra. Vremenska postaja, ki za delovanje potrebuje 24 V enosmerne napetosti, je bila napajana preko laboratorijskega napajalnika, nastavljenega na 24 V. Prvotno je bila naročena vremenska postaja, ki se napaja preko 230 V izmenične napetosti, vendar smo dobili postajo, ki deluje na 24 V. Zaradi tega je bil laboratorijski napajalnik začasna rešitev, predvidena pa je bila menjava z 24 V napajalnikom.

Senzorji temperature v prostorih so se nahajali pod stropom, saj nas je skrbelo, da bi dijaki poškodovali senzorje. Kasneje smo ugotovili, da senzorji odčitavajo občutno previsoke temperature, zaradi česar bo potrebna umeritev senzorjev ali premik senzorjev na drugo lokacijo, kjer bo mogoče realnejše odčitavanje temperature.

Arduino krmilje se nahaja v električni dozi, na katero je privijačen pokrov. Če sva želela posodobiti program, je bilo dozo potrebno odpreti, kar se je izkazalo za zamudno. Preučila sva možnost nalaganja programa na Arduino preko spletja, vendar se je ta rešitev izkazala za nepraktično, saj bi bilo potrebno prav tako fizično pritisniti reset tipko na Arduino.

Odločila sva se, da na stranico doze vgradiva USB vtičnico, kar bi omogočilo hitro nalaganje novih programov in lažje preverjanje delovanja programa v primeru težav, saj Arduino izpisuje določene "debuging" parametre preko serijskega vmesnika. V sistemu še ni bil vgrajen UPS, ki bi zagotavljal napajanje, potrebno za zapiranje kupol v primeru izpada električne energije. Načrt za vgradnjo UPS-a je bil že izdelan, čakamo na nakup, saj trenutno ni ustreznih finančnih sredstev. Program je večinoma deloval brez večjih težav. Občasno so se pojavile težave z odpiranje kupol ob dežju. Uporabnike je motilo to, da so se kupole večkrat odpirale in zapirale. Program je bil napisan tako, da se kupole odprejo, ko temperatura preseže 25 stopinj Celzija in da se kupole zaprejo, ko temperatura pade pod 20 stopinj Celzija. Uporabniki so se pritoževali, da se kupole neprekiniteno odpirajo in zapirajo. Glede na program in testiranje je to nemogoče. Najverjetnejši vzrok je bilo napačno odčitavanje temperature, ki bi ga lahko povzročilo pihanje vetra. Program na RaspberryPi se trenutno v primeru izpada električne energije ne zažene samodejno, kar je potrebno urediti. Spletna stran, ki sva jo naredila za upravljanje s kupolami na daljavo, je funkcionalna, potrebno je urediti vizualno podobo, beleženje IP in MAC naslova obiskovalca ter prijavno stran z uporabniški imeni in gesli, ki bodo potrebni za dostop do administrativne spletne strani. Po pregledu stanja sva si izdelala seznam stvari, ki jih je potrebno popraviti oziroma spremeniti.



Slika 10: Pogled na streho pred nadgradnjo

4 SNOVANJE REŠITEV

Pri iskanju rešitev sva se najprej osredotočila na najbolj nujne spremembe. Najprej je bilo potrebno doseči natančnejše meritve temperature, vetra ipd. kar pomeni, da je bilo potrebno umeriti senzorje.

Najprej sva se lotila senzorjev za temperaturo. Po podatkih proizvajalca imajo tipala oziroma senzorji za temperaturo odstopanje +/- 2°C. Pri meritvah, ki sva jih izvedla 29. novembra sva ugotovila, da je odstopanje precej večje kot navaja proizvajalec. Seveda pa temperature nisva merila pri senzorju, temveč na višini človeškega telesa v sedečem položaju. Sklepala sva, da je za odstopanje kriv položaj senzorja, saj je bil nameščen tik pod strop.

	29. 11.		9. 12.		18. 12.	
PROSTOR	TEMPERATURA	IZPISANO	TEMPERATRURA	IZPISANO	TEMPERATURA	IZPISANO
BE-01	22	25,4	22	22,9	23	23,7
BE-02	22	23,9	21	21,1	22	22,9
BE-05	20	22,7	21	20,9	21	21,7
BE-06	21	22,8	20	20,6	21	21,6
HODNIK	19	21,3	20	20,4	19	19,3

Iz preglednice lahko razberemo, da se je odčitavanje temperature (9. in 18. december) precej izboljšalo, zanimivo pa je, da senzorji vedno prikazujejo temperaturo, ki je višja od dejanske.

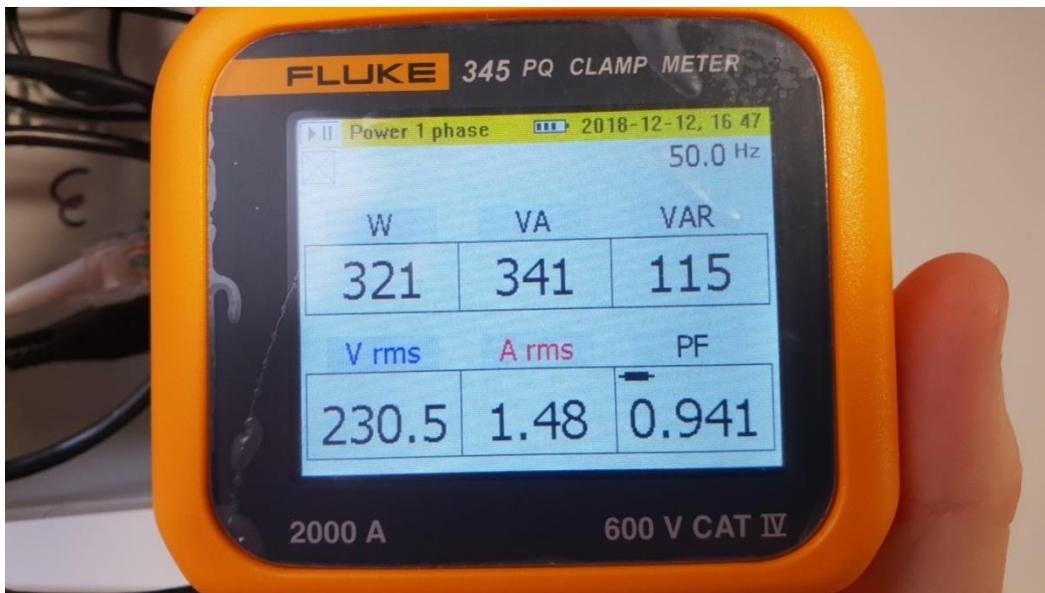
Pri merjenju vetra sva ugotovila, da senzor dejansko ni umerjen tako, kot navaja proizvajalec, tako da sva ga morala umeriti sama. Čakala sva na primeren, vetroven dan, da sva lahko izvedla potrebne meritve. Uporabila sva prenosni merilec vetra, ki hitrost izpisuje v m/s. Vremenska postaja pa veter izpisuje kot napetost od 0 do 10V. Po definirjanju novih vrednosti v programu in dogovoru, da je za kupole nevaren veter 13,8 m/s (50 km/h), se kupole sedaj zaprejo pri vetru večjem od 473 enot.



Slika 11: Merjenje hitrosti vetra

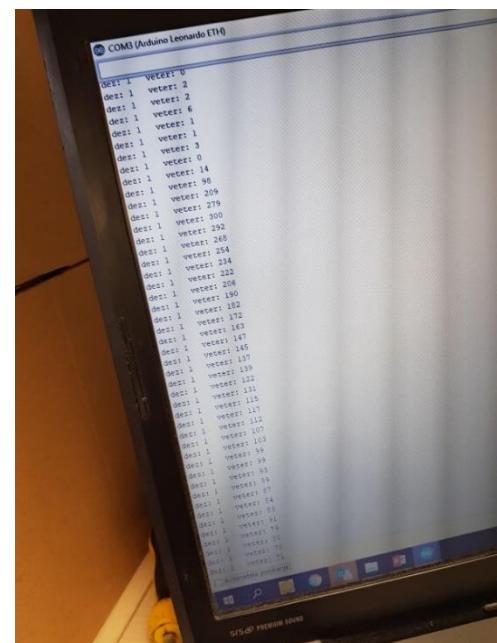
Po ugotovitvi, da bi sistem postal zanesljivejši z namestitvijo UPS-a sva s pomočjo merilnika Fluke analizirala moč, ki jo porabijo kupole. Merila sva na hodniku, saj sta na isti dovod priključena dva motorčka, za dve kupoli, ki ju hkrati odpiramo. Sledil je izračun, da sva dobila potrebne podatke o moči UPS-a, ki bi bil primeren za sistem.

Ugotovila sva, da se v primeru izpada napetosti program na raspberry-ju ne zažene samodejno, kar je predstavljalo določene težave. Programsko sva napako odpravila.



Slika 12: Merjenje moči motorja z merilnikom Fluke

Po določenem času sva ugotovila, da sistem kljub temu da ne dežuje, zaznava stanje dež, kar pomeni, da je nekaj narobe z vremensko postajo. Na strehi sva preverila delovanje releja, izmerila vhodno in izhodno napetost itd. ter ugotovila, da vse deluje brezhibno. Prišla sva do spoznanja, da gre najbrž za prevelik padec napetosti na vodniku (vremenska postaja za javljanje padavin uporablja 1 in 0, torej 1 je 5V, 0 pa 0V) in se odločila, da najprej na arduinu vremenske postaje spremeniva program, namesto digitalnega vhoda sva uporabila analognega in definirala, da je stanje dežuje, ko je napetost nad 2,5V, ne dežuje pa pod to mejo. Rešitev ni pomagala, tako da sva morala premakniti vso krmilje za vremensko postajo na streho. Tako sva rešila težavo. UTP kabel, preko katerega se pošiljajo podatki, pa je bil že predhodno nameščen.



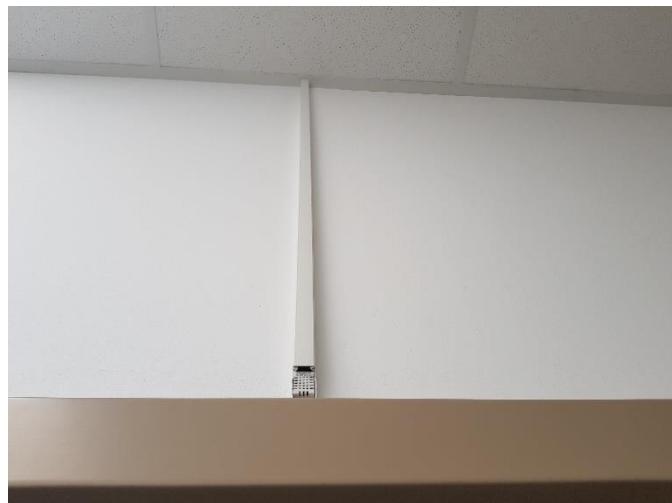
Slika 13: Izpis na serijskem vmesniku

5 NADGRADNJA STROJNEGA DELA

V sistemu je bilo potrebno opraviti kar nekaj nadgradnje strojnega dela.

5.1 TEMPERATURNI SENZOR

Pri prvi točki ne gre za nadgradnjo, temveč le za premaknitev senzorjev temperature na drugo, nižjo lokacijo, s čimer sva dosegla realnejšo vrednost temperature. Senzor je bilo potrebno odklopiti iz Arduina, narediti nove kable, saj je dolžina vodnikov sedaj bistveno daljša, namestiti PVC kanalček na steno in montirati senzor vanj. Namestitev se je izvedla v vsakem prostoru, vendar se razlikuje od prostora do prostora. Še vedno pa sva pazila, da dijaki senzorjev ne morejo zlahka uničiti.

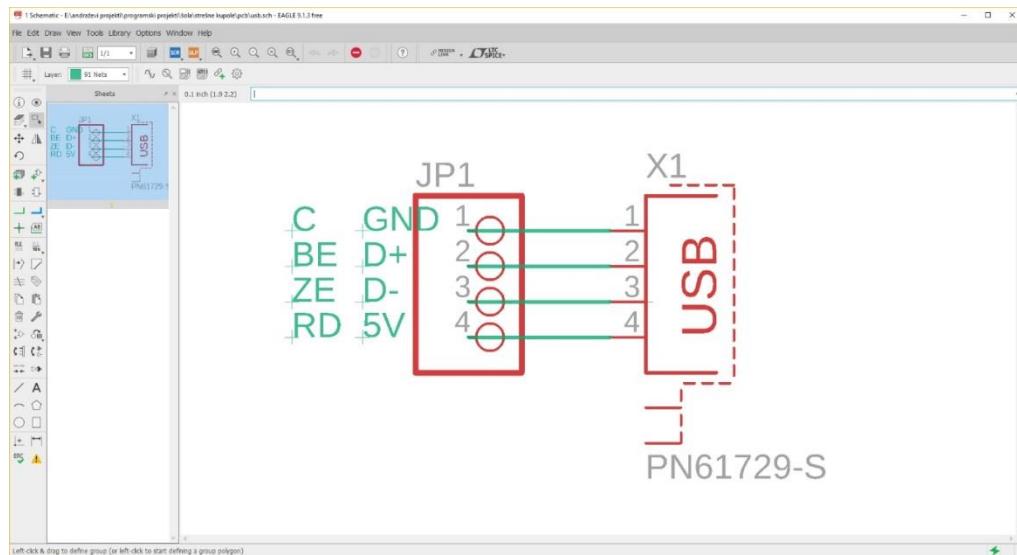


Slika 14: Temperaturni senzor v prostoru

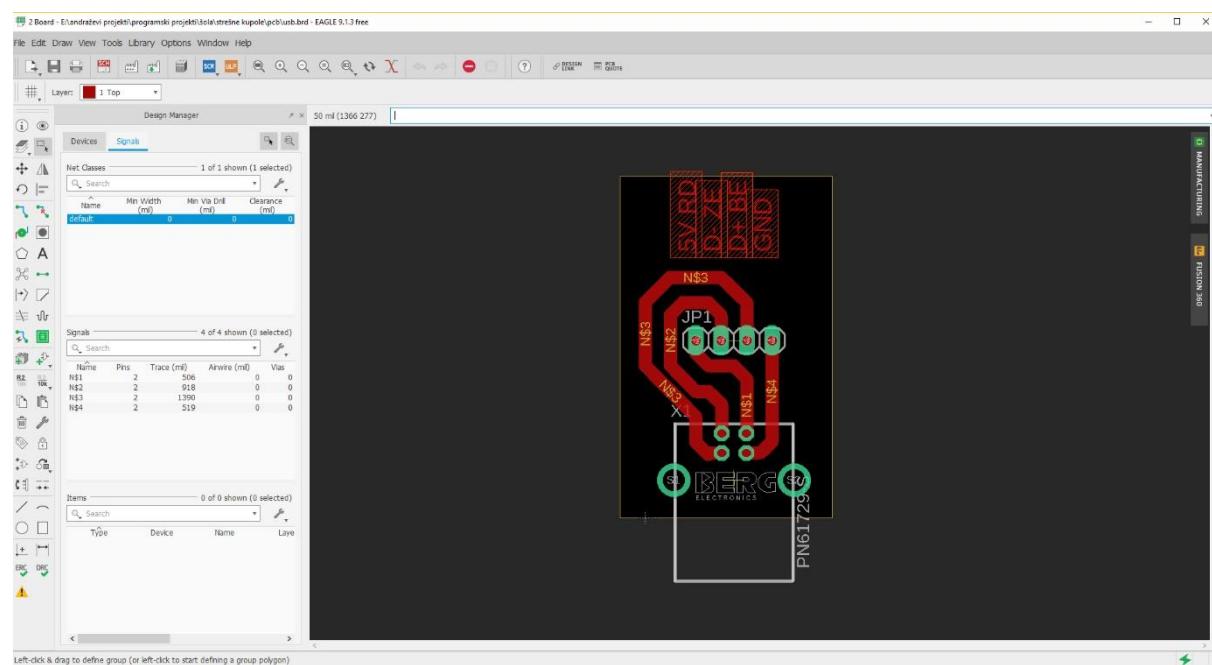
5.2 USB KONEKTORJI

Med posodabljanjem programa sva naletela na težavo, ki je bila problem predvsem s časovnega vidika. Ob vnovičnem nalaganju programa, preverjanju izpisov senzorja ipd. je bilo potrebno odpreti vsako dozo, v kateri je nameščen Arduino, da sva se lahko povezala nanj. Najprej sva poskušala to rešiti tako, da bi podatke prejemala preko interneta, vendar to za njen model ni bilo na voljo, ta funkcija pa ni niti uradno potrjena s strani proizvajalca. Nato sva se odločila, da bova na doze vgradila USB konektorje, ki bodo omogočali prenos podatkov. Iz starih oz. odpisanih tiskalnikov, monitorjev in podobnih računalniških naprav sva pobrala USB konektorje, ki sva jih kasneje preko vezja, ki sva ga sproektirala v programu Eagle in izdelala s pomočjo CNC rezkalnega stroja, povezala na Arduino. Potrebovala sva še mini USB kabel, na eni strani sva odščipnila konektor in ga namestila na vezje, drugo stran pa povezala

na Arduino. V dozi sva izrezala luknjo, v katero sva namestilo konektor in ga zalila z vročo pištolo. Rešitev se je izkazala za dobro.

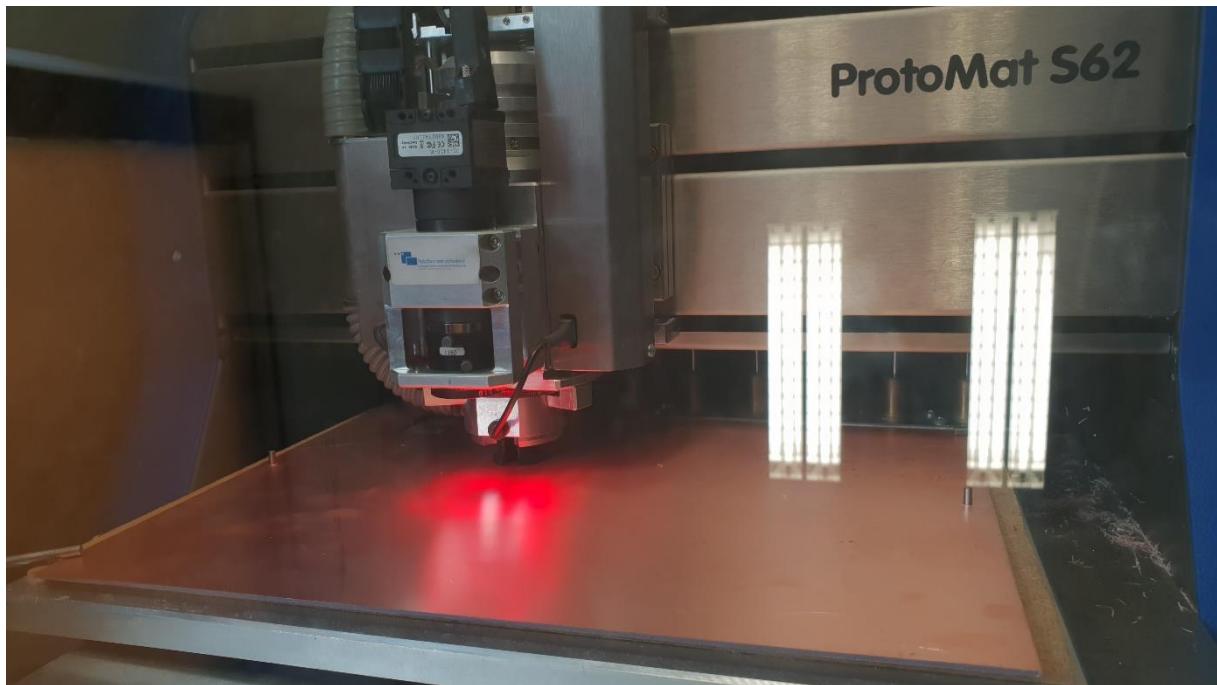


Slika 15: Shema vezja za USB konektor



Slika 16: Tiskano vezje za USBkonektor

Andraž Špan, Rok Lešnik, Nadgradnja nadzora mikroklima v prostoru s krmilnikom Arduino
Šolski center Celje, Srednja šola za kemijo elektrotehniko in računalništvo, 2018/2019



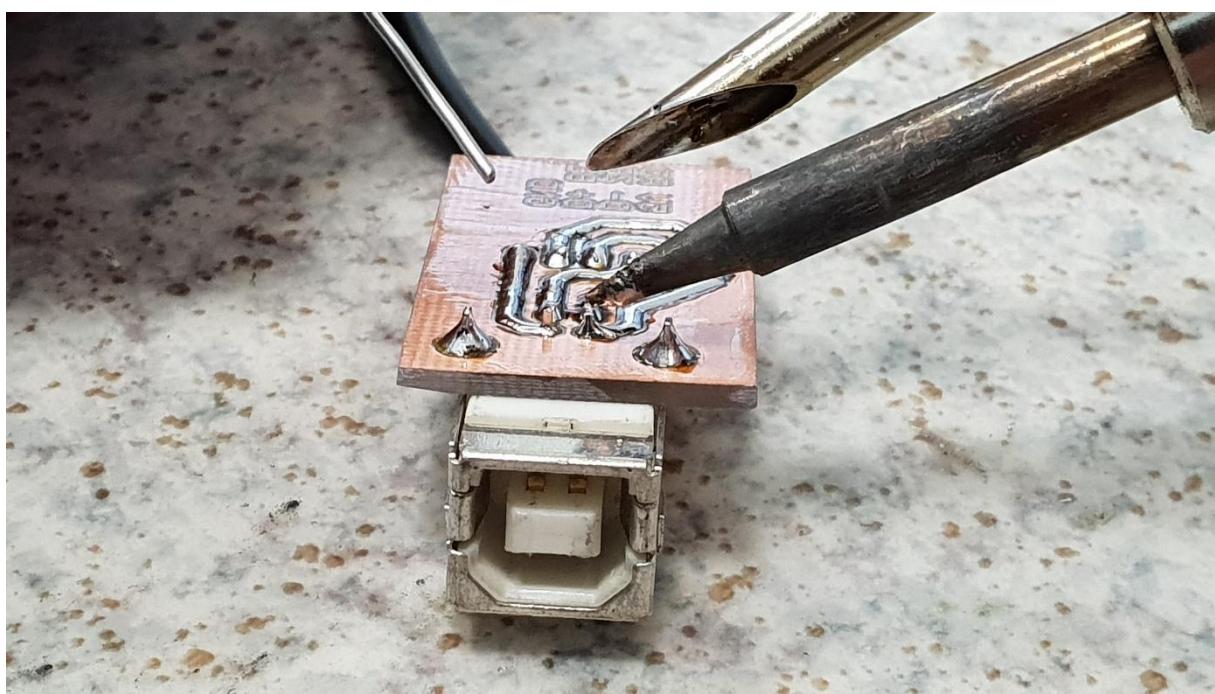
Slika 17:Rrezkanje tiskanega vezja



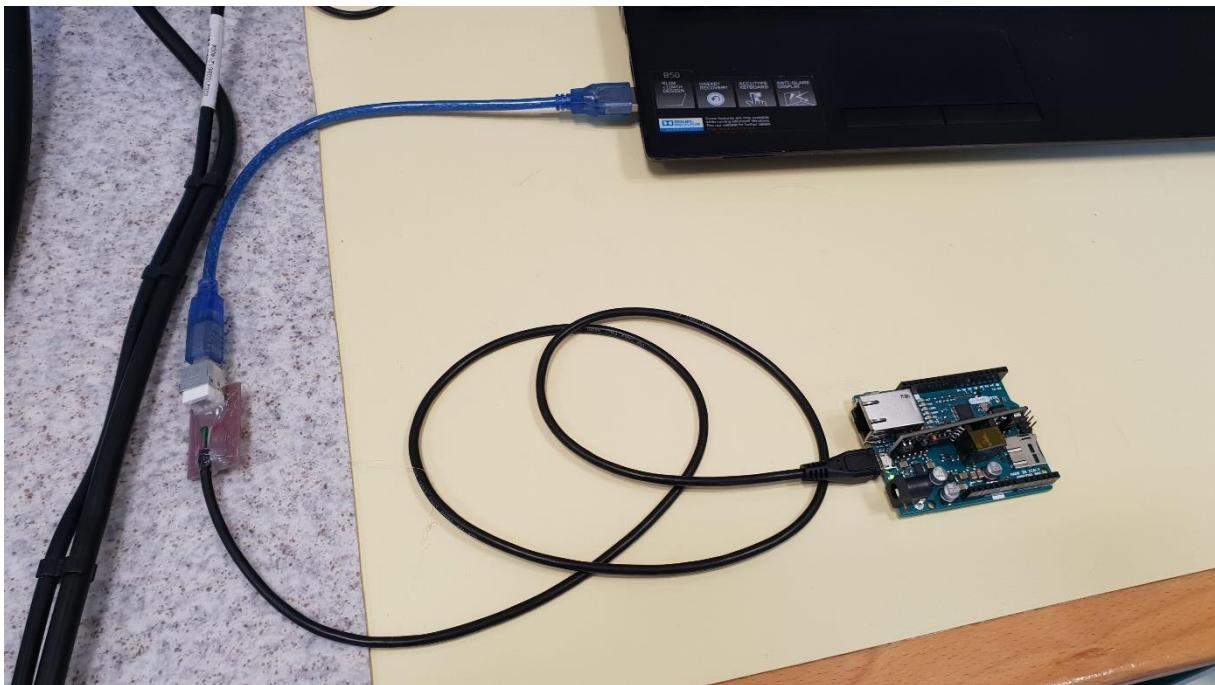
Slika 18: Izrezkano tiskano vezje



Slika 19: Priprava na premaz z kolofonijo



Slika 20: Spajkanje USB konektorja



Slika 21: Testiranje delovanja USBkonektorja

5.3 NAPAJALNIK ZA VREMENSKO POSTAO

Napajanje je bilo prvotno izvedeno kar z usmernikom, saj nisva pravočasno dobila napajalnika, ki bi ustrezal najini vremenski postaji, ki deluje na 24 V in porabi približno 0.1 A toka. Usmernik je težek in seveda ni primeren za uporabo na dolgi rok, saj se mu delovna temperatura dvigne itd. tako da sva ga zamenjala z ustreznim napajalnikom, ki deluje od 100 do 240 V AC, njegov izhod pa je 24 V DC in 1.75 A. Vremenska postaja je priključena z dvema vodnikoma, + in – in je brez zaščitnega vodnika. Napajalnik sva priključila kar preko power kabla, ki sva mu odščipnila konektor in namestila votlice, enka sva storila na izhodu za vremensko postajo.



Slika 22: Napajalnik

5.4 VREMENSKA POSTAJA

Na vremenski postaji sva opravila več posodobitev. Ugotovila sva, da se v dozi, ki je nameščena poleg vremenske postaje, nabira kondenz, tako da je bilo v njej nekaj vode. Na dnu sva zvrtala majhno luknjico, da lahko skozi njo izteka voda.

Do sedaj UTP kabel ter kabel za napajanje nista bila položena v nobeno inštalacijsko cev. Ker zaradi vremenskih vplivov (sneg, dež, spremembe temperature ...) to ni dobro, sva se odločila položiti cev, ki sva jo namestila ob steni prizidka pa vse do stojala z vremensko postajo.

Ugotovila sva tudi, da je stojalo z vremensko postajo nameščeno preblizu odvodnemu in hkrati lovilnemu sistemu strelovoda, za to sva se odločila, da jo prestaviva na bolj primerno mesto na strehi, ki bo še vedno dovolj sonca. Pomislila sva tudi na morebiten udar strele v kovinsko stojalo, saj je to sedaj najvišji del strehe. Po posvetu s profesorji sva spoznala, da je zaradi upornosti, ki jo ustvari podlaga med streho in stojalom možnost za udar minimalna.



Slika 23: Končni izgled vremenske postaje

5.5 UPS

Sistem je pripravljen za namestitev UPS-a. Povezan je na eno točko in posledično na en inštalacijski odklopnik, kar pomeni, da je preklop z direktnega napajanja na napajanje preko UPS-a enostaven. Program pa je prav tako pripravljen na nadgradnjo. UPS-a zaradi težav s finančnimi sredstvi še ni, sva pa podala potrebne podatke za nakup. Izbrali smo UPS, ki je na fotografiji.



Slika 24: Izbran UPS. Vir: <https://i.cdn.nrholding.net/41063934/2000/2000>

6 PROGRAM

V sistemu skupaj delujeta dva programa. Program na vsakem Arduino in program na centralni enoti Raspebrry pi.

6.1 ARDUINO PROGRAM

Glede na želje uporabnikov in zahteve ravnatelja sva iz programa odstranila del avtomatizacije. Sedaj se kupole avtomatsko zaprejo le, če temperatura pade pod 18°C, če dežuje ali piha. Kupola se odpre le ob pritisku na tipko ali preko spletne strani. Program temelji na starem programu, iz katerega so bile odstranjene nepotrebne funkcije.

Kot osnova nama je služil prototip programa, v katerem je že deloval nadzor strešnih kupol preko tipk, ter avtomatizacije, ki sva jo uvedla lani. Po pogovoru z ravnateljem smo ustvarili seznam želenih funkcij in začeli s posodabljanjem programa in implementacijo vseh želenih funkcij. Dodala sva možnost le delnega odpiranja oziroma zapiranja kupol ob dežju, saj se je pojavila težava, da je bila klima v prostoru, kjer ni drugega vira zračenja neprimerna. V primeru da uporabnik tipko drži manj kot štiri sekunde, se kupola le delno odpre oziroma zapre, če uporabnik tipko drži dlje kot eno sekundo, se kupola popolnoma zapre oziroma odpre. Pri temperaturnem odpiranju in zapiranju sva glede na vrednost temperature odčitane s senzorja in sobnega termometra določila relativno napako meritve in v programu na novo ustreznno prilagodila mejne vrednosti za odpiranje in zapiranje kupol.

S pomočjo centralne nadzorne enote, za katero sva uporabila računalnik Raspberry Pi z operacijskim sistemom Raspbian in namensko, kodo napisano v Python programskem jeziku, lahko avtomatiziramo odpiranje in zapiranje kupol ob določenih časovnih točkah. Tako sva v prejšnji različici programa nastavila odpiranje oz. zapiranje kupol ob odmorih, kar se je izkazalo za nesmiselno, tako da sva ta del programa odstranila. S pomočjo tega programa zjutraj omogočimo odpiranje in zvečer, ko je objekt prazen, onemogočimo odpiranje kupol, spremenila sva le časovne vrednosti tako, da se kupole samodejno zaprejo ob 6. uri zvečer, ob 20. uri pa se sistem onemogoči in je odpiranje nemogoče, razen preko spletne strani v načinu admin. Vremenska postaja, ki je sposobna meriti hitrost vetra in zaznavati padavine, pošilja podatke v centralno nadzorno enoto. V primeru padavin ali prevelike hitrosti vetra (sedaj nad 50km/h), ki ne sme biti le kratkotrajni sunek, se sproži zapiranje kupol in onemogoči odpiranje za naslednjih deset minut. Nato sledi ponovno preverjanje vremenskih razmer.

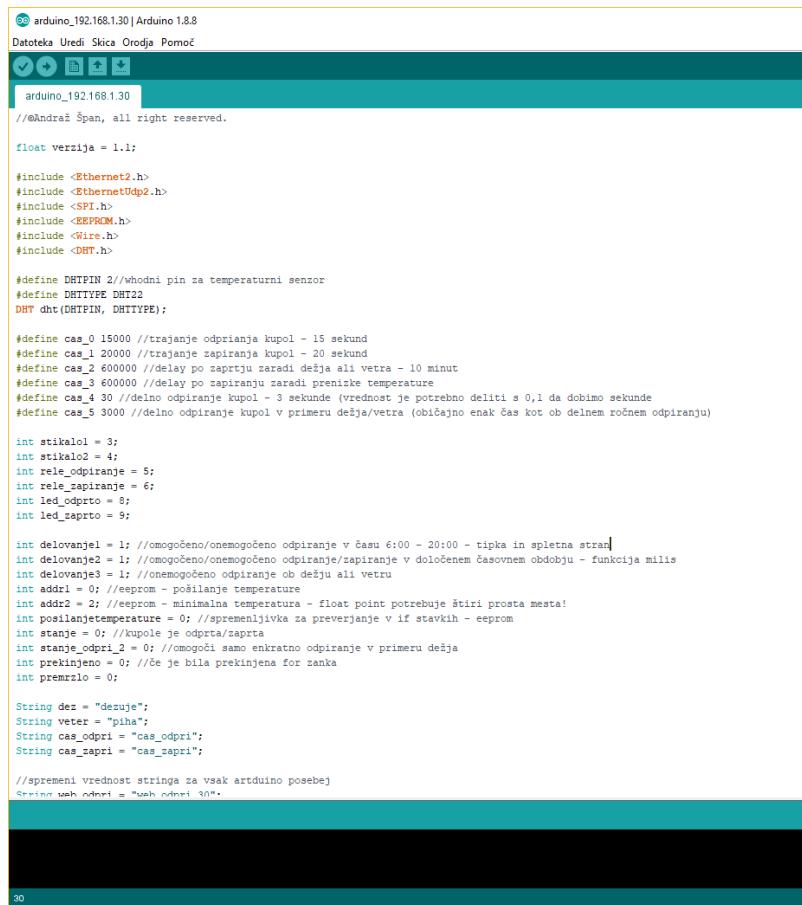
Program sva postopoma nadgrajevala, mu dodajala ter odstranjevala razne funkcije, da sva delovanje čim bolj optimizirala.

Vsi dogodki, ki se zgodijo v sistemu, kot so odpiranje ali zapiranje kupol, razlogi zanju, ki so lahko pritisk tipk, oddaljen nadzor, presežena temperaturna meja, časovni nadzor ali

vremenska postaja, so zabeleženi na centralni nadzorni enoti. Ta zabeleži še točen čas in datum dogodka in v primeru ukaza preko spletja še IP naslov. Ti podatki se enkrat dnevno nalagajo v oblačno storitev, kjer je možen pregled in analiza dogajanja. Vsi podatki so iz varnostnih razlogov shranjeni poleg oblačne storitve še lokalno na centralni nadzorni enoti, dostop pa je mogoč le lokalno.

Program za nadzor kupol, ki se izvaja na mikrokrmlniku Arduino Leonardo ETH je karseda optimiziran za zanesljivo in hitro delovanje. To vključuje tudi skrb, da velikost programa ni prevelika, saj bi to lahko povzročilo nestabilnost. V programu so vključeni že vsi elementi, za katere menimo da bi lahko bili potrebni pri nadgradnjah v prigodnosti, kar pomeni da bo večina razvoja v prihodnosti potekala na centralni nadzorni enoti, razen v primeru zahtevka za popolnoma nepričakovano funkcijo.

Komunikacija med centralno nadzorno enoto in mikrokrmlnikom Arduino poteka po UDP standardu preko lokalnega omrežja. Vse naprave so med sabo povezane preko LAN omrežja.



The screenshot shows the Arduino IDE interface with the following details:

- Top menu bar: File, Uredi, Skica, Orodja, Pomoč.
- Toolbar icons: Save, Open, New, Copy, Paste, Find, Select All, Undo, Redo, Run, Stop, Reload, Help.
- Code editor area:

```
//@Andraž Špan, all right reserved.

float verzija = 1.1;

#include <Ethernet2.h>
#include <EthernetUDP.h>
#include <SPI.h>
#include <EEPROM.h>
#include <Wire.h>
#include <DHT.h>

#define DHTPIN 2 //vhodni pin za temperturni senzor
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

#define cas_0 15000 //trajanje odpiranja kupol - 15 sekund
#define cas_1 20000 //trajanje zapiranja kupol - 20 sekund
#define cas_2 600000 //delay po zaprtju zaradi dežja ali vetera - 10 minut
#define cas_3 600000 //delay po zapiranju zaradi prenizke temperature
#define cas_4 30 //delno odpiranje kupol - 3 sekunde (vrednost je potrebno deliti s 0,1 da dobimo sekunde
#define cas_5 3000 //delno odpiranje kupol v primeru dežja/vetra (običajno enak čas kot ob delnem ročnem odpiranju)

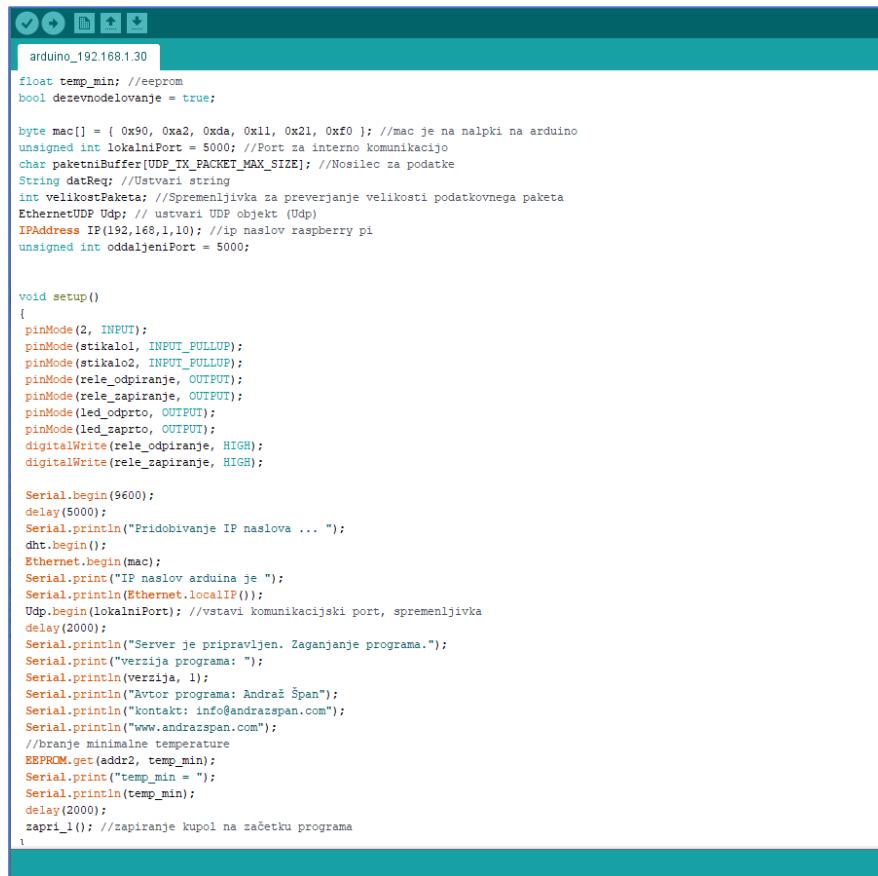
int stikalol = 3;
int stikaloz = 4;
int rele_odpiranje = 5;
int rele_zapiranje = 6;
int led_odprto = 8;
int led_zaprt = 9;

int delovanjel = 1; //omogočeno/onemogočeno odpiranje v času 6:00 - 20:00 - tipka in spletna stran
int delovanje2 = 1; //omogočeno/onemogočeno odpiranje/zapiranje v določenem časovnem obdobju - funkcija millis
int delovanje3 = 1; //onemogočeno odpiranje ob dežju ali veteru
int addrl = 0; //eeprom - pošiljanje temperature
int addr2 = 2; //eeprom - minimalna temperatura - flosa point potrebuje štiri prosta mesta!
int poslanjetemperature = 0; //spremenljivka za preverjanje v if stavkih - eeprom
int stanje = 0; //kupole je odprta/zaprt
int stanje_odpri_2 = 0; //omogoči samo enkratno odpiranje v primeru dežja
int prekinjeno = 0; //če je bila prekinjena for zanka
int premrzlo = 0;

String dez = "dezuje";
String veter = "piha";
String cas_odpri = "cas_odpri";
String cas_zapri = "cas_zapri";

//spremeni vrednost stringa za vsak arduino posebej
String web_adresi = "web_adresi.in".
```
- Bottom status bar: 30

Slika 25: Definiranje spremenljivk za program



```

arduino_192.168.1.30

float temp_min; //eeprom
bool dezenodelovanje = true;

byte mac[] = { 0x90, 0xa2, 0xd0, 0x11, 0x21, 0xf0 }; //mac je na nalogki na arduino
unsigned int lokalniPort = 5000; //Port za interno komunikacijo
char paketniBuffer[UDP_TX_PACKET_MAX_SIZE]; //Nosilec za podatke
String datReq; //Ustvari string
int velikostPaketa; //Spremenljivka za preverjanje velikosti podatkovnega paketa
EthernetUDP Udp; // ustvari UDP objekt (Udp)
IPAddress IP(192,168,1,10); //ip naslov raspberry pi
unsigned int oddaljeniPort = 5000;

void setup()
{
    pinMode(2, INPUT);
    pinMode(stikal01, INPUT_PULLUP);
    pinMode(stikal02, INPUT_PULLUP);
    pinMode(rele_odpiranje, OUTPUT);
    pinMode(rele_zapiranje, OUTPUT);
    pinMode(led_odprto, OUTPUT);
    pinMode(led_zaprt, OUTPUT);
    digitalWrite(rele_odpiranje, HIGH);
    digitalWrite(rele_zapiranje, HIGH);

    Serial.begin(9600);
    delay(5000);
    Serial.println("Pridobivanje IP naslova ... ");
    dht.begin();
    Ethernet.begin(mac);
    Serial.print("IP naslov arduina je ");
    Serial.println(Ethernet.localIP());
    Udp.begin(lokalniPort); //vstavi komunikacijski port, spremenljivka
    delay(2000);
    Serial.println("Server je pripravljen. Zaganjanje programa.");
    Serial.print("verzija programa: ");
    Serial.println(verzija, 1);
    Serial.println("Avtor programa: Andraž Špan");
    Serial.println("kontakt: info@andrazspan.com");
    Serial.println("www.andrazspan.com");
    //branje minimalne temperature
    EEPROM.get(addr2, temp_min);
    Serial.print("temp_min = ");
    Serial.println(temp_min);
    delay(2000);
    zapri_1(); //zapiranje kupol na začetku programa
}

```

Slika 27: Zagon programa.



```

arduino_192.168.1.30

void loop()
{
    //branje vhodnih vrednosti stikala v učilnici
    if(digitalRead(stikal01) == LOW && delovanje3 == 1 && premrzlo == 0) //če je pritisnjena gumb za odpiranje, če ne dežuje ali piha in če ni bilo zaprto zaradi prenizke temperature
    {
        //dodatana log funkcija - pošiljanje razloga dejanja
        Udp.beginPacket(IP, oddaljeniPort); //Pripravi paket za pošiljanje
        Udp.print("tipka odpri"); //Pošlje podatke
        Udp.endPacket(); //Konča paket

        odpri_0();
        delay(1000);
        for(int a=0; a<cas_2; a++) //preverja če uporabnik drži tipko dlje kot dve sekundi
        {
            if(digitalRead(stikal01) == LOW) //če uporabnik drži tipko dlje kot dve sekundi
            {
                odpri_1();
                prekinjeno = 1;
                break; //konča for zanka
            }
            delay(100);
        }
        digitalWrite(rele_odpiranje, HIGH); //rele ugasne v vsakem primeru (failsafe), podatek pošlje serverju le če uporabnik drži tipko manj kot dve sekundi
        if(prekinjeno == 0) //če uporabnik drži tipko mnaj kot dve sekundi (for zanka ni bila prekinjena)
        {
            Udp.beginPacket(IP, oddaljeniPort); //Pripravi paket za pošiljanje
            Udp.print(odprt); //Pošlje podatke
            Udp.endPacket(); //Konča paket
            stanje = 1;
        }
    }

    if(digitalRead(stikal01) == LOW && delovanje3 == 0 && dezenodelovanje == true) //če je pritisnjena gumb za odpiranje, če dežuje ali piha in če je omogočeno odpiranje ob dežu
    {
        //dodatana log funkcija - pošiljanje razloga dejanja
        Udp.beginPacket(IP, oddaljeniPort); //Pripravi paket za pošiljanje
        Udp.print("tipka odpri dež"); //Pošlje podatke
        Udp.endPacket(); //Konča paket
    }

    odpri_2();
}

if(digitalRead(stikal01) == LOW && delovanje3 == 1 && premrzlo == 1) //če je pritisnjena gumb za odpiranje, če ne dežuje ali piha in če je bilo zaprto zaradi prenizke temperature
{
    //dodatana log funkcija - pošiljanje razloga dejanja
    Udp.beginPacket(IP, oddaljeniPort); //Pripravi paket za pošiljanje
}
```

Slika 26: Zaznavanje pritiska tipke in pošiljanje UDP sporočila

```
arduino_192.168.1.20
IPAddress IP(192,168,1,10); //ip naslov raspberry pi
unsigned int oddaljeniPort = 5000;

EthernetServer server(80); //spletna stran
String readString; //spletna stran

void setup()
{
pinMode(senzor_veter, INPUT);
pinMode(senzor_dez, INPUT_PULLUP); //internal pullup v testnem načinu, preveri kako je na vremenski postaji in ustrezno prilagodi!
pinMode(led_dez, OUTPUT);
pinMode(led_veter, OUTPUT);

Serial.begin(9600);
Ethernet.begin(mac);
Serial.println("Pridobivanje IP naslova ... ");
delay(5000);
Serial.print("IP naslov serverja je ");
Serial.println(Ethernet.localIP());
Udp.begin(lokalniPort); //vstavi komunikacijski port, spremenljivka
delay(2000);
Serial.println("Server je pripravljen. Zaganjanje programa.");
Serial.print("verzija programa: ");
Serial.println(verzija, 1);
Serial.println("Avtor programa: Andraž Špan");
Serial.println("kontakt: info@andrazspan.com");
Serial.println("www.andrazspan.com");
delay(2000);
}

void loop()
{
//zaznavanje dežja
trenuten_dez = digitalRead(senzor_dez);
Serial.print("dez: ");
Serial.print(trenuten_dez);
if(trenuten_dez == meja_dez && delovanje == 1) //če je zaznan dež
}
```

Slika 28: Zagonski del programa vremenske postaje

```
arduino_192.168.1.20
void loop()
{
//zaznavanje dežja
trenuten_dez = digitalRead(senzor_dez);
Serial.print("dez: ");
Serial.print(trenuten_dez);
if(trenuten_dez == meja_dez && delovanje == 1) // če je zaznan dež
{
Serial.println(dez);
digitalWrite(led_dez, HIGH);
Udp.beginPacket(IP, oddaljeniPort); //Pripravi paket za pošiljanje
Udp.print(dez); //Pošlje podatke
Udp.endPacket(); //Konča paket
digitalWrite(led_dez, LOW);
zadenMillis = millis();
delovanje = 0;
Serial.println("posiljanje obvestil o vremenu je onemogoceno");
}

//zaznavanje vetra
trenuten_veter = analogRead(senzor_veter);
Serial.print("  ");
Serial.print("veter: ");
Serial.println(trenuten_veter);
if(trenuten_veter > meja_veter && delovanje == 1) //če zazna sunek vetra
{
for (int a=0; a<10; a++) //zakasnitev da preverimo da ni le sunek
{
trenuten_veter = analogRead(senzor_veter);
Serial.print("veter: ");
Serial.println(trenuten_veter);
delay(100);
}
trenuten_veter = analogRead(senzor_veter);
if(trenuten_veter > meja_veter && delovanje == 1) //če še vedno piha
{
Serial.println(veter);
}
```

Slika 29: Zaznavanje dežja in vetra.

Program uporablja naslednje knjižnice:

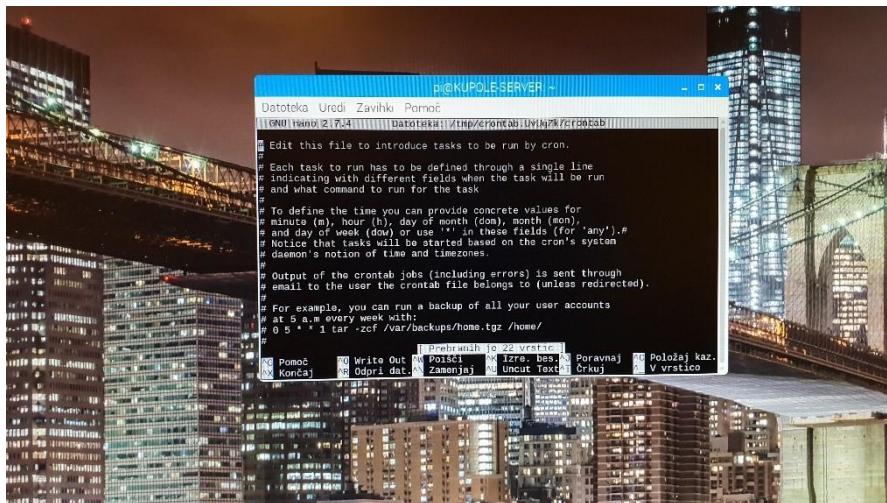
- Ethernet2 – knjižnica omogoča povezavo Arduina s spletom
- EthernetUdp2 – knjižnica omogoča pošiljanje in prejemanje UDP sporočil preko spletja.
- SPI – knjižnica omogoča uporabo SPI naprav.
- EEPROM – knjižnica omogoča uporabo EEPROM spomina
- Wire – knjižnica omogoča uporabo I2C naprav.
- DHT – knjižnica omogoča uporabo senzorja za temperaturo DHT22.

6.1 RASPBERRY PI PROGRAM

Raspberry pi program trenutno služi le za komunikacijo med enotami in za beleženje dogodkov. Vremenska postaja v primeru vetra ali dežja pošlje sporočilo da trenutno dežuje ali piha. Program to sporočilo obdela in pošlje vsem ostalim enotam v sistemu. Ostale enote se odzovejo tako da zaprejo kupole in nazaj pošljejo sporočilo s katerim potrjujejo zapiranje kupol. Vsaka posamezna enota kupole pošilja sporočila o dogodkih, kot so odpiranje oziroma zapiranje kupole in razlog odpiranja oziroma zapiranja. Razlog odpiranja je lahko pritisk na tipko ali ukaz s spletne strani. Razlog za zapiranje je lahko pritisk na tipko, ukaz s spletne strani ali prenizka temperatura v prostoru. Spletna stran je del Raspberry pi programa.

Vsaka enota kupol prav tako omogoča pošiljanje temperature iz prostora, kar je namenjeno prikazu temperature na spletni strani, ki bo predvidoma javno dostopna in je trenutno še v postopku izdelave. Pripravljen je tudi program za UPS, ki bo zaznal izpad električne energije in v tem primeru zaprl vse kupole in onemogočil odpiranje. Na Raspberry teče tudi časovni program, ki vsak dan ob 6:00 omogoči odpiranje in ob 20:00 onemogoči odpiranje kupol. Prav tako ob 18:00 zapre kupole, ki so morebiti ostale odprte.

V planu je nadgradnja, ki bi upoštevala še poseben režim delovanja med prazniki in šolskimi počitnicami.



Slika 30: Priprava programa za odpiranje in zapiranje ob določeni uri

```

7.11.2017.py - C:\Users\andra\OneDrive\strešne kupole\raspberry pi\končan program\7.11.2017.py (3.6.5)
File Edit Format Run Options Window Help
#program izpisuje prejete stringe in posreduje podatke o vremenu
import socket
import time
import logging

RASPBERRY_IP = "192.168.1.10"
ARDUINO1_IP = "192.168.1.30"
ARDUINO2_IP = "192.168.1.31"
ARDUINO3_IP = "192.168.1.32"
ARDUINO4_IP = "192.168.1.33"
ARDUINO5_IP = "192.168.1.34"
PORT = 5000
sporocilol = "dezuje"
sporocilo2 = "piha"

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

rpi = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
rpi.bind((RASPBERRY_IP, PORT))

logging.basicConfig(filename='/home/pi/Desktop/log/odpiranje_in_zapiranje.log', level=logging.INFO)

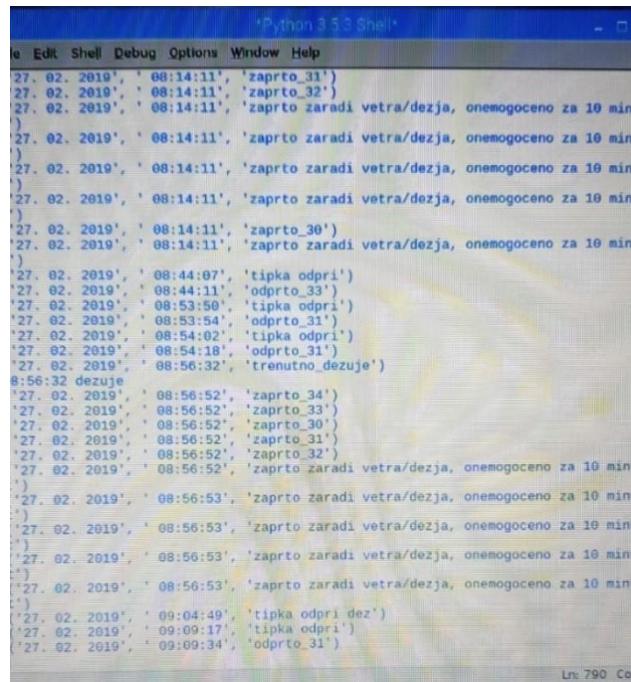
while True:
    podatek, addr = rpi.recvfrom(2048) #branje prejetih podatkov
    podatek2 = podatek.decode('utf_8') #dekodiranje iz bit v string (python 3)
    vrednost = time.strftime("%d. %m. %Y"), time.strftime(" %H:%M:%S"), podatek2 #ustvari string
    print (vrednost) #izpis v shell
    logging.info(vrednost) #datalog

    if podatek2 == "trenutno_dezuje": #če dežuje
        print(time.strftime("%H:%M:%S"), "dezuje") #izpis trenutneg časa ob prejetju ukaza
        sock.sendto(bytes(sporocilol, "utf-8"), (ARDUINO1_IP, PORT)) #python 3 zahteva kodiranje iz string v bit
        sock.sendto(bytes(sporocilol, "utf-8"), (ARDUINO2_IP, PORT))
        sock.sendto(bytes(sporocilol, "utf-8"), (ARDUINO3_IP, PORT))
        sock.sendto(bytes(sporocilol, "utf-8"), (ARDUINO4_IP, PORT))
        sock.sendto(bytes(sporocilol, "utf-8"), (ARDUINO5_IP, PORT))

    if podatek2 == "trenutno_piha": #če piha
        print(time.strftime("%H:%M:%S"), "piha") #izpis trenutneg časa ob prejetju ukaza
        sock.sendto(bytes(sporocilo2, "utf-8"), (ARDUINO1_IP, PORT))
        sock.sendto(bytes(sporocilo2, "utf-8"), (ARDUINO2_IP, PORT))
        sock.sendto(bytes(sporocilo2, "utf-8"), (ARDUINO3_IP, PORT))
        sock.sendto(bytes(sporocilo2, "utf-8"), (ARDUINO4_IP, PORT))
        sock.sendto(bytes(sporocilo2, "utf-8"), (ARDUINO5_IP, PORT))

```

Slika 31: Trenutno napisan program za zapiranje kupol v primeru dežja ali vetra.



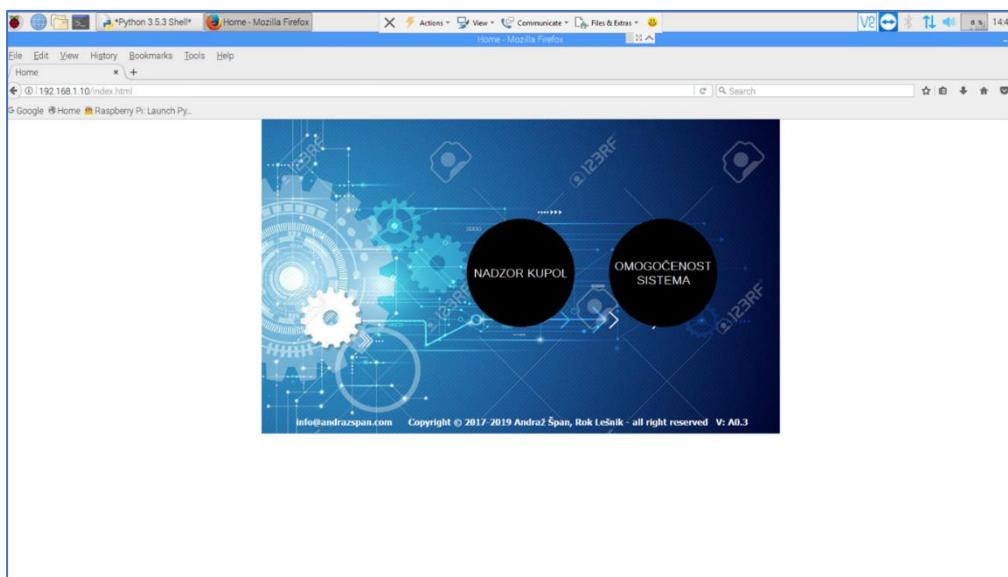
The screenshot shows a terminal window titled "Python 3.5.3 Shell". The window contains a log of sensor readings from February 27, 2019. The log includes timestamped entries for various sensors, such as 'zaprto_31', 'zaprto_32', and 'tipka odpri'. Some entries include descriptive text like 'onemogoceno za 10 min' (disabled for 10 minutes). The log ends with a timestamp of '09:09:32' and the message 'trenutno_dezuje' (currently active).

```
27. 02. 2019', '08:14:11', 'zaprto_31')
27. 02. 2019', '08:14:11', 'zaprto_32')
27. 02. 2019', '08:14:11', 'zaprto zaradi vetra/dezja, onemogoceno za 10 min')
27. 02. 2019', '08:14:11', 'zaprto zaradi vetra/dezja, onemogoceno za 10 min')
27. 02. 2019', '08:14:11', 'zaprto zaradi vetra/dezja, onemogoceno za 10 min')
27. 02. 2019', '08:14:11', 'zaprto zaradi vetra/dezja, onemogoceno za 10 min')
27. 02. 2019', '08:14:11', 'zaprto_30')
27. 02. 2019', '08:14:11', 'zaprto zaradi vetra/dezja, onemogoceno za 10 min')
27. 02. 2019', '08:44:07', 'tipka odpri')
27. 02. 2019', '08:44:11', 'odprto_33')
27. 02. 2019', '08:53:50', 'tipka odpri')
27. 02. 2019', '08:53:54', 'odprto_31')
27. 02. 2019', '08:54:02', 'tipka odpri')
27. 02. 2019', '08:54:18', 'odprto_31')
27. 02. 2019', '08:56:32', 'trenutno_dezuje')
08:56:32 dezuje
27. 02. 2019', '08:56:52', 'zaprto_34')
27. 02. 2019', '08:56:52', 'zaprto_33')
27. 02. 2019', '08:56:52', 'zaprto_30')
27. 02. 2019', '08:56:52', 'zaprto_31')
27. 02. 2019', '08:56:52', 'zaprto_32')
27. 02. 2019', '08:56:52', 'zaprto zaradi vetra/dezja, onemogoceno za 10 min')
27. 02. 2019', '08:56:53', 'zaprto zaradi vetra/dezja, onemogoceno za 10 min')
27. 02. 2019', '08:56:53', 'zaprto zaradi vetra/dezja, onemogoceno za 10 min')
27. 02. 2019', '08:56:53', 'zaprto zaradi vetra/dezja, onemogoceno za 10 min')
27. 02. 2019', '08:56:53', 'zaprto zaradi vetra/dezja, onemogoceno za 10 min')
27. 02. 2019', '09:04:49', 'tipka odpri dez')
27. 02. 2019', '09:09:17', 'tipka odpri')
27. 02. 2019', '09:09:34', 'odprto_31')
```

Slika 32: Primer izpisa dogajanja v programu.

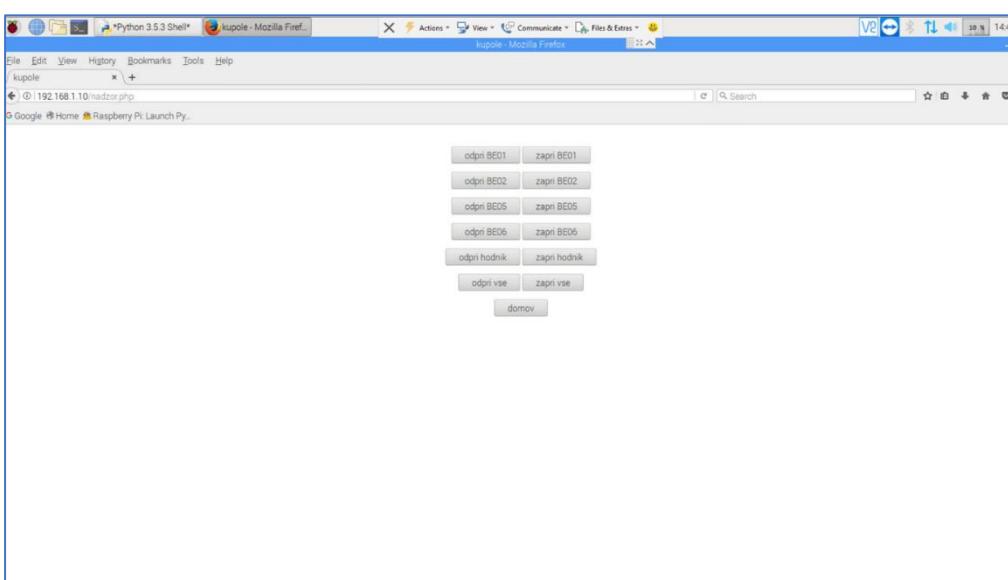
6 SPLETNA STRAN

Spletna stran omogoča odpiranje in zapiranje kupol in spremjanje mejne minimalne temperature, pri kateri se kupole zaprejo. Dostop do spletne strani je varovan z geslom, stran pa zabeleži IP in MAC naslov uporabnika. Spletna stran je trenutno funkcionalna, potrebno pa je spremeniti grafično podobo, ki je za enkrat zelo osnovna.



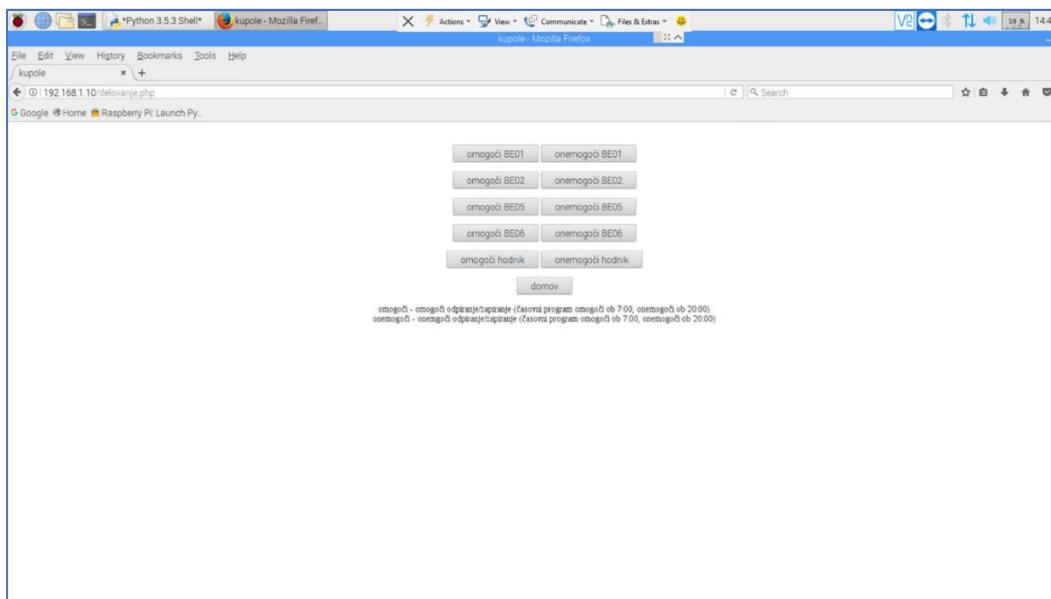
Slika 33: Domača stran spletne strani

Spletna stran je napisana v php programskem jeziku. Ko uporabnik pritisne gumb na spletni strani se sproži javascript, ki sproži python program. Ta python program pa nato pošlje sporočilo za odpiranje oziroma zapiranje odvisno od tega, kateri gumb je bil pritisnjen.



Slika 34: Zavihek nadzor kupol

Spletna stran je trenutno v beta stanju, kar pomeni da še ni dokončana. Potrebno je spremeniti grafično podobo ter dodati nekaj stvari. Izdelati je potrebno še stran, ki bo javno dostopna in prikazovala podatke o temperaturah v prostorih ter stran, kjer bo mogoče spremirnjati minimalno temperaturo za zapiranje kupol. Dodelati je potrebno še prijavo z uporabniškimi imeni in gesli ter beleženje podatkov obiskovalcev, kot so IP naslov in MAC naslov.



Slika 35: Zavihek omogočenost sistema

8 MOŽNOSTI NADGRADNJE V PRIHODNOSTI

V sistemu sva dopustila nekaj možnosti za nadgradnjo.

Na tem mestu bova samo omenila, da je možnost za nagradno z UPS-om, vendar je to natančneje opisano v poglavju 3.5.

Poleg sedanjega poslopja šole, se bo dogradil prizidek, ki naj bi bil z obstoječim objektom povezan preko hodnika v zraku. Načrtovano je, saj se je sistem pokazal kot zelo uporaben, da ga namestimo tudi v novem prizidku z uporabo obstoječe vremenske postaje. V osnovi bo potrebno položiti potrebne kable in obstoječe premakniti, saj zaradi morebitnega zamakanja nisva speljala kablov neposredno s strehe, ampak sva preboj naredila v steno, kjer je sedaj po načrtu mišljen hodnik.



Slika 36: Predlogi

8.1 NEREALIZIRANI OZ. OVRŽENI PREDLOGI UPORABNIKOV

Dobila sva predlog, da bi uporabila senzor gibanja, ki bo, ko bo zaznal da se v prostoru nekaj časa ni nič premaknilo, sprožil avtomatiko, ko pa bi zaznal premikanje, bi sistem deloval v režimu ročno. Predlog sva po posvetu z ravnateljem opustila, saj smo ugotovili da bi bil primeren le za kabinet, bilo pa bi veliko dela s pisanjem programa in dodatno nadgradnjo s senzorjem, za to se nam ideja ni zdela racionalna.

Velikokrat se je pojavila tudi ideja, da bi sistemu dodala še zaveso, ki bi bila nameščena pod kupolo saj je pouk v sončnih dneh zelo otežen, ker se projekcija na platno ne vidi, sonce v prostoru dodatno dvigne temperaturo in tako naprej. Predlagano je bilo, da namestimo zavese, ki se bodo odpirale na pritisk tipke v prostoru. Predlog se nama je zdel zanimiv, saj bi to vplivalo tudi na klimo v prostoru, preučila sva možnosti, vendar sva ugotovila, da za realizacijo tega, tudi če bi jih namestila le v učilnicah, nimamo dovolj sredstev.



Slika 37: Sestanek z uporabniki

9 UGOTOVITVE

Na začetku sva si postavila šest hipotez, v nadaljevanju so opisane ugotovitve v zvezi z njimi:

- Po izpolnitvi uporabniških želj in zahtev bo uporabniška izkušnja boljša, hkrati se bo izboljšalo mnenje o sistemu; hipotezo sva potrdila, saj so sedaj uporabniki bolj zadovoljni s sistemom kot pred nadgradnjo.
- Premik senzorja temperature bo omogočil zaznavanje realnejše vrednosti temperature; hipotezo sva potrdila, saj se je zaznavanje temperature močno izboljšalo, kar pomeni da je sistem sedaj bolj zanesljiv in učinkovit.
- Poenostavitev programa bo sprostila prostor na mikrokrmlniku; hipotezo sva potrdila, saj se je zasedenost spomina zmanjšala za 10 %.
- UPS bo izboljšal varnost sistema oz. prostora; hipoteze ne morava ne potrditi in ne ovreči, saj zaradi finančnih sredstev UPS-a še ni v sistemu, lahko pa sklepava, da se bo varnost izboljšala.
- Vgradnja USB priključkov bo olajšala morebitne posodobitve programa v prihodnosti in lokalni nadzor delovanja; hipotezo sva potrdila, saj lahko sedaj program posodobiva veliko hitreje, prav tako pa lahko zelo hitro odčitava vrednosti, ki jih izpisuje serijski vmesnik.
- Zapiranje ob 20. uri bo preprečilo morebitne nevšečnosti zaradi vremena. Hipoteze ne morava ne potrditi ni ovreči, saj ni nikoli prišlo do nobene večje nevšečnosti, lahko pa sklepava, da bi to preprečilo morebitne neljube dogodke v prihodnosti.

10 ZAKLJUČEK

Naloga nama je pokazala, da kljub trudu v preteklem šolskem letu v projektih, kot je ta, vedno strmimo k nadgradnji in posodobitvam, da sistem vedno znova izboljšujemo.

Med pripravo naloge sva pridobila veliko potrebnih izkušenj za nadaljnjo poklicno pot v smeri elektrotehnika. Pridobila sva znanje na področju programiranja krmilnikov v jeziku C+ in phyton ter nekaj osnov s HTML kodo, prav tako sva se naučila uporabljati analizator moči Fluke, dodatne izkušnje pa sva pridobila tudi pri montaži stikal, nadometnih vtičnic, polaganja kablov, montaže serverske omarice, izvajanju raznih meritev in še kaj bi se našlo.

V raziskavo in izvedbo projekta sva vložila veliko dela, porabila sva veliko prostega časa, hkrati pa pridobila veliko izkušenj in praktičnega znanja na področju elektrotehnike, računalništva, zdravstva in strojništva.



Slika 38: Upravljanje serverja

VIRI IN LITERATURA

- Elektrotehniški priročnik, Ljubljana: Tehniška založba Slovenije, d.d., 2013
- A. Špan, R. Lešnik, Nadzor mikroklima v prostoru z mikrokrmlnikom Arduino, Celje: Šolski center Celje, 2018
- Podatki o temperaturnem senzorju: <https://www.electroschematics.com/wp-content/uploads/2015/02/DHT22-datasheet.pdf>, pridobljeno 12. 9. 2018
- Arduino Leonardo ETH 2 with POE. Dostopno na: <https://store.arduino.cc/arduino-ethernet-2-with-poe>, pridobljeno 14. 9. 2018
- Arduino Leonardo ETH. Dostopno na: <https://store.arduino.cc/arduino-leonardo-eth>
- Arduino IDE. Dostopno na: <https://www.arduino.cc/en/Main/Software>
- AutodeskEagle. Dostopno na: <https://www.autodesk.com/products/eagle/overview>
- Arduino Ethernet2 Library. Dostopno na: <https://www.arduinolibraries.info/libraries/ethernet2>
- Arduino Wire Library. Dostopno na: <https://www.arduino.cc/en/reference/wire>

Večina slik je iz lastnega arhiva. Slike pridobljene s spleta imajo poleg imena še povezavo na kateri so dostopne.

PRILOGE

IZJAVA

IZJAVA*

Mentor (-ica) , Andrej Grilc , v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi naslovom Nadgradnja nadzora mikroklima v prostoru z mikrokrmilnikom Arduino kateré avtorji (-ice) so Poh Lešnik , Andraž Špan , : :

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljeni literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo (-ičino) dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno naložo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna nalogasta nastala v okviru projekta Mladi za Celje,
- da je raziskovalno naložo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje

Celje, 6.2.2019



Podpis mentorja(-ice)

Podpis odgovorne osebe

*

POJASNILO

V skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja(-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja(-ice) fotografskega gradiva, katerega ni avtor(-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

NAVODILA ZA UPORABO

NAVODILA ZA UPORABNIKE

Sistem omogoča odpiranje in zapiranje kupol s tipkami.

Delovanje nadziramo na naslednji način:

1. Kratek pritisk na tipko za odpiranje, delno odpre kupolo.
2. Kratek pritisk na tipko za zapiranje, delno zapre kupolo.
3. Dolg pritisk na tipko za odpiranje, popolno odpre kupolo.
4. Dolg pritisk na tipko za zapiranje, popolno zapre kupolo.

Delovanje sistema v primeru dežja ali močnejšega vetra:

V tem primeru se celoten sistem samodejno zapre, kar pomeni da je odpiranje kupol onemogočeno. To ne drži za prostore, kjer ni drugih virov zračenja (BE-01 in BE-02), kjer je omogočeno delno odpiranje kupol s kratkim pritiskom na tipko odpiranje.

Delovanje sistema v odvisnosti od temperature:

Ko se temperatura v prostoru spusti pod 18°C, sistem zapre kupolo v tistem prostoru.

Časovna omejitev sistema:

Vse zgoraj naštete funkcije ne delujejo od 20. ure zvečer do 6. ure zjutraj. Takrat je odpiranje možno le preko spletnega nadzora. Ob 18. uri se sistem samodejno zapre.