



Šolski center Celje

Srednja šola za kemijo, elektrotehniko in računalništvo

PAMETNI DOM Z ARDUINO MEGA 2560 MIKROKONTROLERJEM

RAZISKOVALNA NALOGA

Avtorja:

Gašper Gril, E-2.b

Kristjan Šoln, E-2.b

Mentor:

Matej Kališek, inž. el.

Mestna občina Celje, Mladi za Celje

Celje, 2017

KAZALO VSEBINE

KAZALO SLIK	3
1. POVZETEK	4
2. UVOD	5
2.1 PREDSTAVITEV PROBLEMA	5
2.2 HIPOTEZE	5
2.3 PREDSTAVITEV RAZISKOVALNIH METOD	5
3. OPIS DELOVANJA	6
4. OPIS KOMPONENT	7
4.1 ARDUINO MEGA 2560 MIKROKONTROLER	7
4.2 REGULATOR NAPETOSTI LM2596	8
4.3 MOSFET STIKALNO VEZJE	9
4.4 ZASLON NA DOTIK (TOUCHSCREEN MODUL)	11
4.5 SVETLOBNI SENZOR	12
4.6 TEMPERATURNI SENZOR DS18B20	13
4.7 LED TRAK	14
4.8 VENTILATOR (HLADILNO TELO)	14
4.9 ŽARNICA (GRELNO TELO)	15
5. IZDELAVA	16
5.1 MODEL HIŠE	16
5.2 NAPAJANJE	17
5.3 SVETLOBNI SENZOR	17
5.4 TEMPERATURNI SENZOR	18
5.5 MOSFET VEZJE	18
5.6 VENTILATOR, ŽARNICA IN LED TRAKOVI	19
6. PROGRAM	20
6.1 GLAVNI MIKROKONTROLER	20
6.1.1 BRANJE SENZRJEV DS180	20
6.1.2 NADZOR TEMPERATURE	21
6.1.3 BRANJE SVETLOBNIH SENZORJEV	21
6.1.4 NADZOR SVETLOSTI	21
6.1.5 SERIJSKA KOMUNIKACIJA	22
6.1.6 EEPROM	24
6.2 MIKROKONTROLER Z ZASLONOM NA DOTIK	24
7. UGOTOVITVE IN REZULTATI	28

8.	ZAKLJUČEK	29
9.	VIRI	30
10.	ZAHVALA	31
11.	IZJAVA	32

KAZALO SLIK

Slika 1:	Razporeditev komponent.....	6
Slika 2:	Arduino Mega 2560.....	8
Slika 3:	Modul LM2596	8
Slika 4:	Shematika tipične vezave regulatorja napetosti LM2596. Vir: https://www.onsemi.com/pub/Collateral/LM2596-D.PDF	9
Slika 5:	Shema vezave posameznega mosfeta BS170.	10
Slika 6:	Shema vezave celotnega mosfet vezja	11
Slika 7:	Arduino Mega 2560 in priključki na adapterju za modul z zaslonom na dotik	12
Slika 8:	Modul z zaslonom na dotik	12
Slika 9:	Shema vezave fotoupora	13
Slika 10:	Shema vezave več senzorjev DS18B20 z parazitnim napajanjem.....	14
Slika 11:	LED trak, štirje segmenti	14
Slika 12:	Ventilator	15
Slika 13:	Žarnica z podnožjem	16
Slika 14:	Postavitev posameznih elementov po sobi	17
Slika 15:	Napajanje modela	17
Slika 16:	Fotoupor.....	18
Slika 17:	Priključek za DS18B20	18
Slika 18:	Mosfet stikalno vezje	19
Slika 19:	Razporeditev žarnic, ventilatorjev in LED trakov na maketi	20
Slika 20:	Koda za branje senzorjev DS18B20.....	20
Slika 21:	Koda za nadzor temperature	21
Slika 22:	Koda za branje in nadzor svetlosti	22
Slika 23:	Grafični vmesnik.....	24
Slika 24:	Primer kode za grafični vmesnik	25
Slika 25:	Funkcija za branje podatkov serijske povezave na krmilniku z zaslonom na dotik	26
Slika 26:	Del funkcije za obdelovanje podatkov serijske povezave na glavnem krmilniku	27

1. POVZETEK

V tej raziskovalni nalogi sva želela izdelati napravo, ki bi lahko uravnavala temperaturo in količino svetlobe v prostoru ter hkrati uporabniku omogočila preprosto nastavljanje željenih vrednosti in vpogled v meritve. Izdelala sva maketo hiše s štirimi sobami, v katerih lahko nadzorujemo ta dva dejavnika. Glavni del te makete sta dva mikrokontrolerja Arduino Mega 2560, od katerih en nadzira zaslon na dotik, drug pa toploto in svetlost v prostoru ter bere podatke iz senzorjev. Med sabo komunicirata preko serijske povezave, pri tem pa uporabljata vnaprej določene kode. Za uporabo take naprave v praksi potrebujemo le močnejša grelna, hladilna in svetlobna telesa.

In this research project, we wanted to make a device, which would be able to regulate temperature and light levels inside a room. At the same time, it would enable the user to set the desired temperature and brightness levels and see current values displayed on a screen. We have made a model of a house with four rooms to see how the device works in practice. The main part of it are two Arduino Mega 2560 microcontrollers. One of them drives a touchscreen, while the other one regulates temperature and takes readings from the sensors. They use pre-defined codes and a serial connection to communicate with each other. To use this kind of device in real life, the only thing you would have to do is choose appropriate heating, cooling and lighting solutions for each room.

2. UVOD

2.1 PREDSTAVITEV PROBLEMA

Na ugodno počutje in zmožnost koncentracije v sobi vpliva mnogo dejavnikov, kot recimo hrup, temperatura in vlažnost zraka, količina svetlobe, koncentracija CO₂ v prostoru ipd. Tako sva se s sošolcem odločila, da bova naredila napravo, ki bo čim bolj avtomatizirala regulacijo čim več fizikalnih dejavnikov, ki bi lahko vplivali na počutje.

Z napravo sva želela avtomatizirati nadzor temperature v prostoru. Lahko jo seveda nadzorujemo z ročnim odpiranjem oken, a najina naprava bi lahko bolje rešila problem in bi pripomogla k vzdrževanju temperature.

Z napravo sva hotela ustvariti tudi enakomerno količino svetlobe v prostoru. Ko bi se količina sončne svetlobe, ki prihaja skozi okno, v nekem časovnem intervalu spremenila, bi se prilagodila svetlost luči in tako dosegla bolj konstantno količino svetlobe v prostoru.

Želela sva tudi omogočiti ogled vrednosti, dobljenih iz senzorjev, in nastavitve željenih vrednosti narediti uporabniku čim bolj prijazno. Tako sva se odločila za zaslon na dotik, ki obenem reši oba problema.

2.2 HIPOTEZE

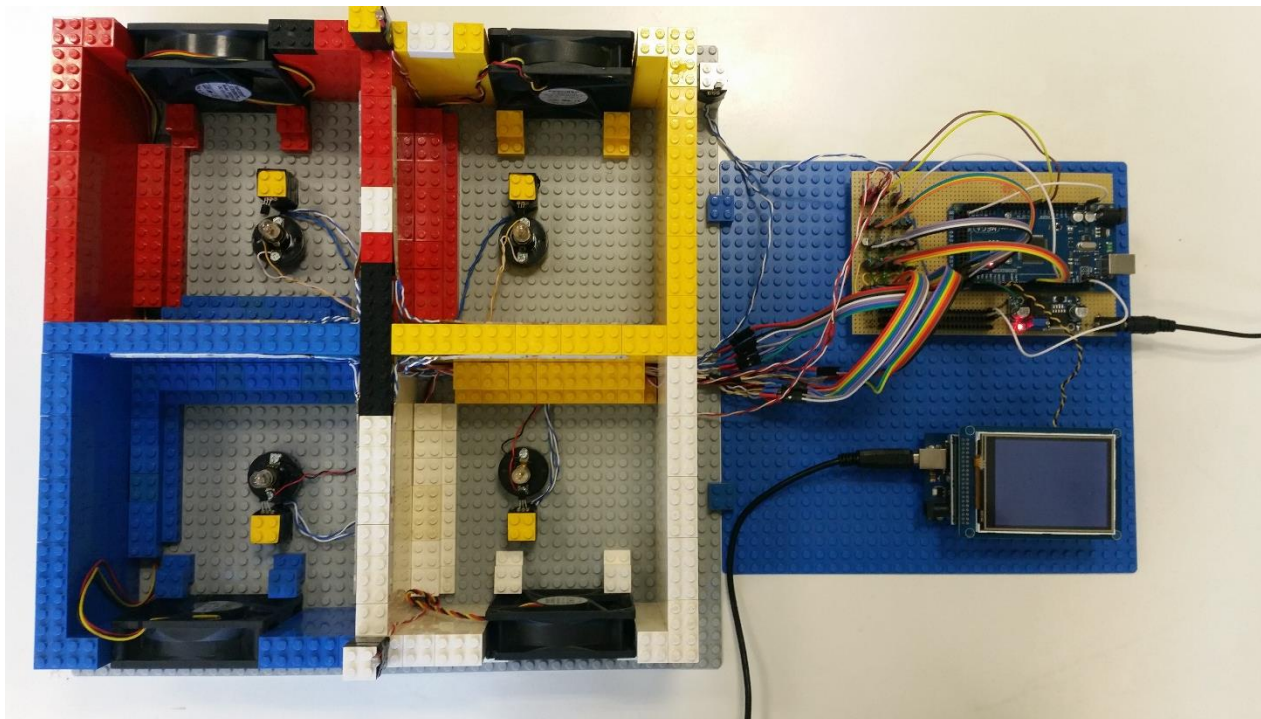
- Avtomatizirana regulacija temperature v prostorih
- Avtomatizirana regulacija količine svetlobe v prostorih
- Preprost vpogled v meritve vsakega sensorja z zaslonom na dotik
- Nastavljanje željene temperature in jakosti svetlobe v posamezni sobi

2.3 PREDSTAVITEV RAZISKOVALNIH METOD

Za raziskovanje sva najprej zbrala čim več informacij in idej o merjenju, nadzoru in potem še načinu prikaza željenih količin. Potrebno je bilo preučiti, katere komponente, materiale ipd. imava na razpolago in kako jih povezati v celoto. Program za Arduino ploščici sva napisala v računalniškem programu Arduino IDE (Arduino Integrated Development Environment). Potem sva sestavila model, ki bi omogočil preizkus delovanja programa in pokazal probleme, ki jih prej nisva predvidela. Za lažje timsko delo sva si pomagala z portalom GitHub, ki omogoča preprosto deljenje in razvoj kode.

3. OPIS DELOVANJA

Model je sestavljen iz štirih sob ter vezja, ki vključuje 2 ploščici Arduino Mega 2560 (eden je glavni, drug pa skrbi za delovanje zaslona na dotik), modul z zaslonom na dotik (ang. touchscreen module), pretvornik napetosti iz 12V na 5V ter mosfet stikalno vezje. Vsaka soba je opremljena s senzorjem za temperaturo DS18B20, ventilatorjem za hlajenje, žarnico za gretje in LED trakom, ki osvetljuje sobo. Tudi zunaj hiše je postavljen en temperaturni senzor, ki meri zunanjo temperaturo. Zunaj hiše (na nasprotnih straneh) sta postavljena dva svetlobna senzorja, ki merita jakost sončne svetlobe.



Slika 1: Razporeditev komponent

Glavni mikrokontroler odčitava podatke iz temperaturnih senzorjev DS18B20 in iz fotouporov, prav tako pa z mosfeti nadzoruje LED trakove, ventilatorje in žarnice.

Temperaturni senzorji so vezani vzporedno (pravimo jim OneWire), na Arduino jih priključimo s samo eno povezavo. Mikrokontroler jih potem kliče po serijskih številkah in tako dobi podatke od vsakega senzorja.

Senzor za merjenje jakosti svetlobe je pravzaprav delilnik napetosti, sestavljen iz fotoupora ter običajnega upora. Fotoupora sta postavljena na nasprotni strani modela hiše, iz katerih Arduino dobi informacije o količini svetlobe zunaj.

Poleg mikrokontrolerjev je zraven mosfet stikalno vezje, s katerim krmilimo porabnike z večjimi nazivnimi tokovi ali napetostmi kot jih lahko mikrokontroler prenese. Maksimalni tok na posameznem izhodu mikrokontrolerja je 40 mA, skupni tok pa ne sme presegati 200 mA, s tem krmilnim vezjem pa lahko upravljamo porabnike, ki potrebujejo do 500 mA toka in do 60 V. Primer porabe posameznega ventilatorja je približno 150 mA pri napetosti 12 V.

Glavni mikrokontroler je preko serijske povezave povezan z drugim, enakim mikrokontrolerjem, katerega naloga je, da nadzoruje zaslon na dotik. Preko tega zaslona lahko vidimo meritve senzorjev, nastavimo željeno temperaturo v sobi, željeno območje svetlosti v sobi ter vklapljammo in izklapljammo luči.

Za dodatni mikrokontroler sva se odločila, ker je modul z zaslonom na dotik narejen tako, da se priklopi na Arduino ploščico neposredno in hkrati zakrije večino izhodov na njej. Na sliki 7 desno so prikazani priključki modula.

Celotno maketo napaja adapter, ki napetost 220 V/50 Hz pretvori na 12 V DC. S to napetostjo napajava vse 12 V komponente. Dodala sva še regulator napetosti, ki to napetost pretvori na 5 V.

Ventilator simulira hladilno telo. V praksi bi ga lahko nadomestili z prezračevalnimi sistemi na električno upravljanje, ki bi odpirali okna in tako hladili prostor. Poleti, oziroma ko je zunanja temperatura višja od notranje, bi se okna odprla samo ponoči. Nadomestek temu bi lahko bila tudi klimatska naprava, ki bi jo nadzirali z mikrokrmilnikom.

Poleg hlajenja želimo imeti tudi gretje. Na maketi za to poskrbijo 5 V žarnice, ki so postavljene blizu senzorja. V praksi bi jih nadomestili z relejem, ki bi vklapljal in izklapljal centralno ogrevanje.

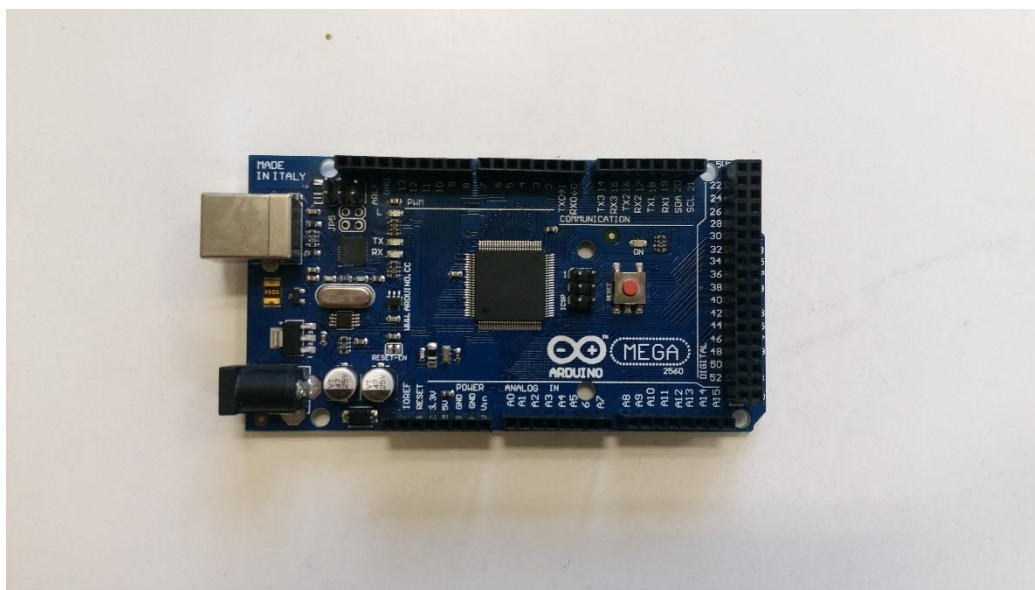
LED trakovi na modelu simulirajo luči. Uporabili bi jih lahko tudi v praksi, kajti z visoko-svetlečimi LED trakovi se da tudi osvetliti prostor, poleg tega pa je možno z njimi privarčevati na porabi energije.

4. OPIS KOMPONENT

4.1 ARDUINO MEGA 2560 MIKROKONTROLER

To je mikrokontroler, ki vsebuje 8-bitni Atmelov mikrokontroler ATmega 2560 z 256KB flash, 4 KB EEPROM in 8 KB SRAM pomnilnika. Ima 54 digitalnih vhodno-izhodnih (I/O) priključkov, od teh jih 15 podpira PWM, 16 analognih vhodov ter 16 MHz oscilator. Deluje na 5 V, kot vir napetosti pa poleg teh 5 V nudi še napetost 3.3 V. Maksimalni izhodni tok na posameznem priključku je 40 mA, celotni izhodni tok pa praviloma ne sme preseči 200 mA. Vgrajena je tudi varovalka, ki pri toku 500 mA izklopi ploščico. Ko se vrednost zmanjša pod to mejo, ponovno vklopi ploščico.

Mikrokontroler lahko napajamo na več načinov. Lahko ga napajamo s 5 V preko vhoda USB. Tu je maksimalni vhodni tok omejen na 250-500 mA, odvisno od računalnika. Naslednji način napajanja je preko 2.1 mm vtičnice. Tu je zahtevana napetost med 6 V in 20 V, priporočeno pa je območje med 7 V in 12V. Med ta vir napetosti in regulator je postavljena še dioda za zaščito pred nepravilno polariteto. Zadnji način je preko 5 V priključka na ploščici. Običajno ta priključek uporabimo za napajanje vezja, a ga lahko uporabimo za napajanje same ploščice. Tu je potrebna previdnost, saj tu obidemo diodo in varovalko. Pri izdelavi sva uporabila ravno ta način napajanja, zato sva potrebovala še pretvornik napetosti, ki napetost 12 V pretvori v 5 V.

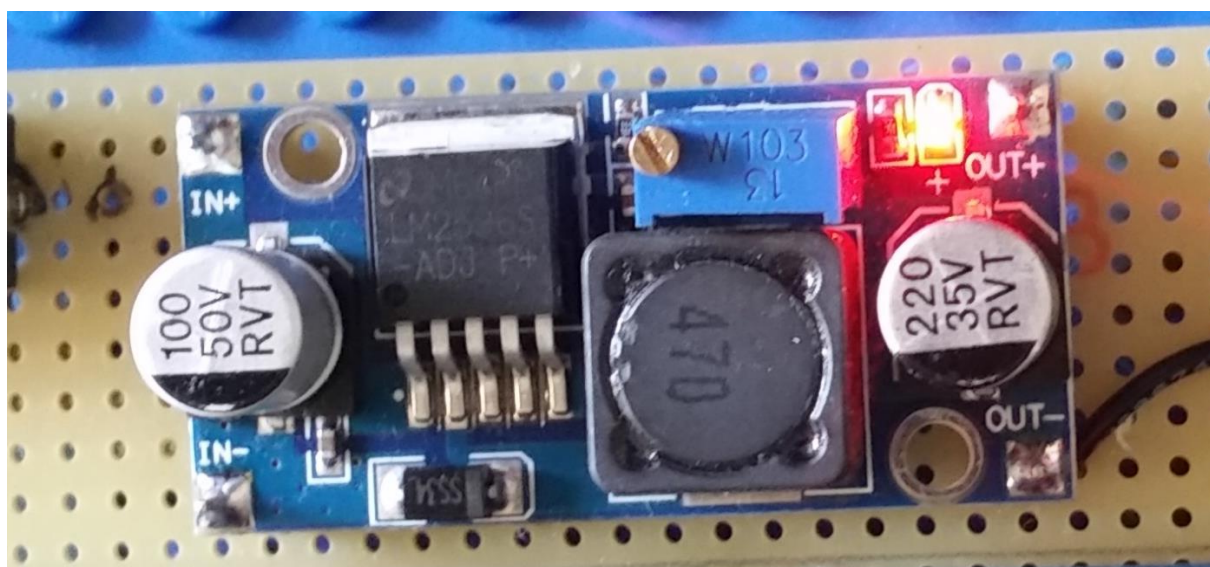


Slika 2: Arduino Mega 2560

4.2 REGULATOR NAPETOSTI LM2596

Model potrebuje dve napetosti: 5 V in 12 V. 12 V nam priskrbi adapter, ki pretvori 230 V izmenične napetosti v 12 V enosmerne napetosti. Maksimalni izhodni tok tega adapterja je 5 A, vhodni pa 1.6 A.

To napetost pripeljemo do vezja, kjer z njo poleg 12 V komponent napajamo tudi modul LM2596. To je modul, ki lahko regulira izhodno napetost na od 12V do 1.5V. Napetost stabilizira, da se ne spreminja ob spremembi izhodnega toka. Reguliramo jo s pomočjo potenciometra na samem modulu. Vse dodatne komponente, ki jih potrebuje čip, so že vključene v modul.



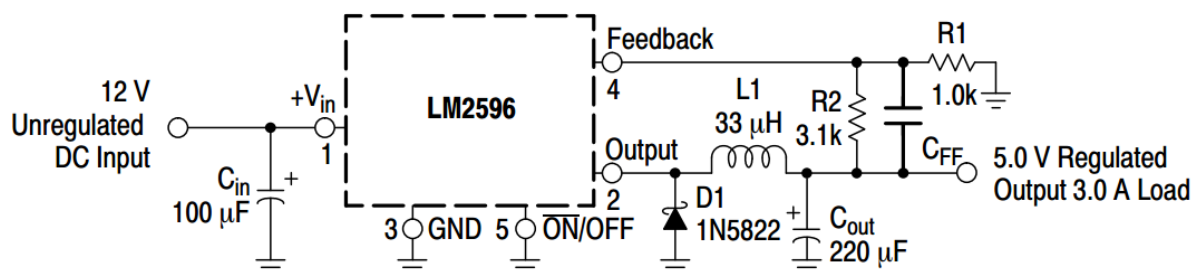
Slika 3: Modul LM2596

Glavni del modula je LM2596 stikalni regulator napetosti. Je integrirano vezje, narejeno za vhodne napetosti do 45 V, izhodne napetosti od 1.23 do 37 V in izhodni tok do 5 A. Ima 150 kHz notranji oscilator in zaščito pred prevelikim izhodnim tokom in preveliko temperaturo vezja. Ima tudi možnost stanja pripravljenosti. Lahko ga vklopimo tako, da na kontakt ON/OFF pripeljemo napetost, ki je višja od 1.6 V in manjša od vhodne napetosti.

Je stikalni regulator napetosti. To pomeni, da elektronsko stikalo s frekvenco 150 kHz vhodno napetost »razseka« in jo transformira na sekundarno stran, kjer se zgladi in usmeri. Dodatno regulacijsko vezje je potrebno za reguliranje časa odprtja stikala. Razmerje med časom, ko je stikalo odprto in časom periode s tujko pravimo »duty cycle«. S tem razmerjem določimo izhodno napetost.

LM2596

Typical Application (Adjustable Output Voltage Version)



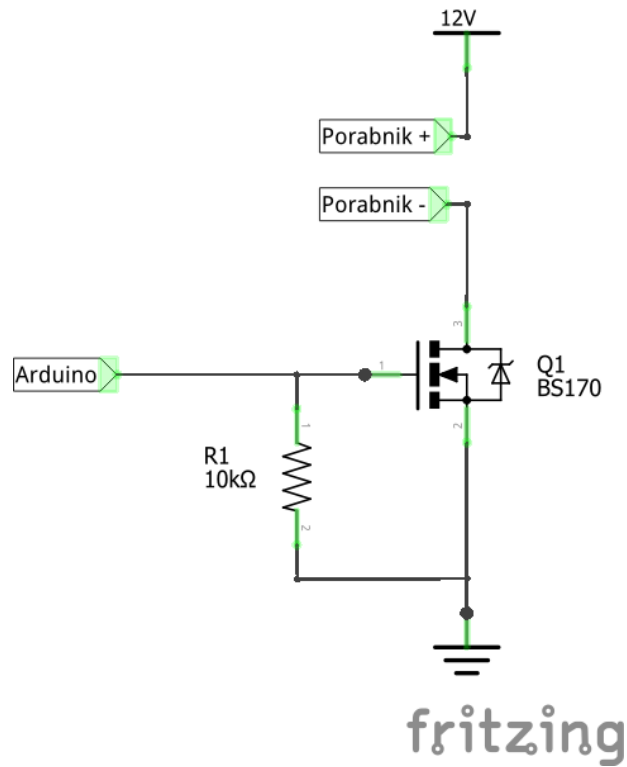
Slika 4: Shematika tipične vezave regulatorja napetosti LM2596. Vir: <https://www.onsemi.com/pub/Collateral/LM2596-D.PDF>

Zgornja slika prikazuje tipično vezavo. Na prvi kontakt čipa je pripeljana vhodna napetost, na drugem je regulirana napetost, tretji se priključi na 0V, četrti je priključek »Feedback«, peti pa je že prej omenjen ON/OFF priključek. Upor R2 je na modulu zamenjan z 10 k Ohm potenciometrom. Z njim nastavljamo željeno izhodno napetost.

4.3 MOSFET STIKALNO VEZJE

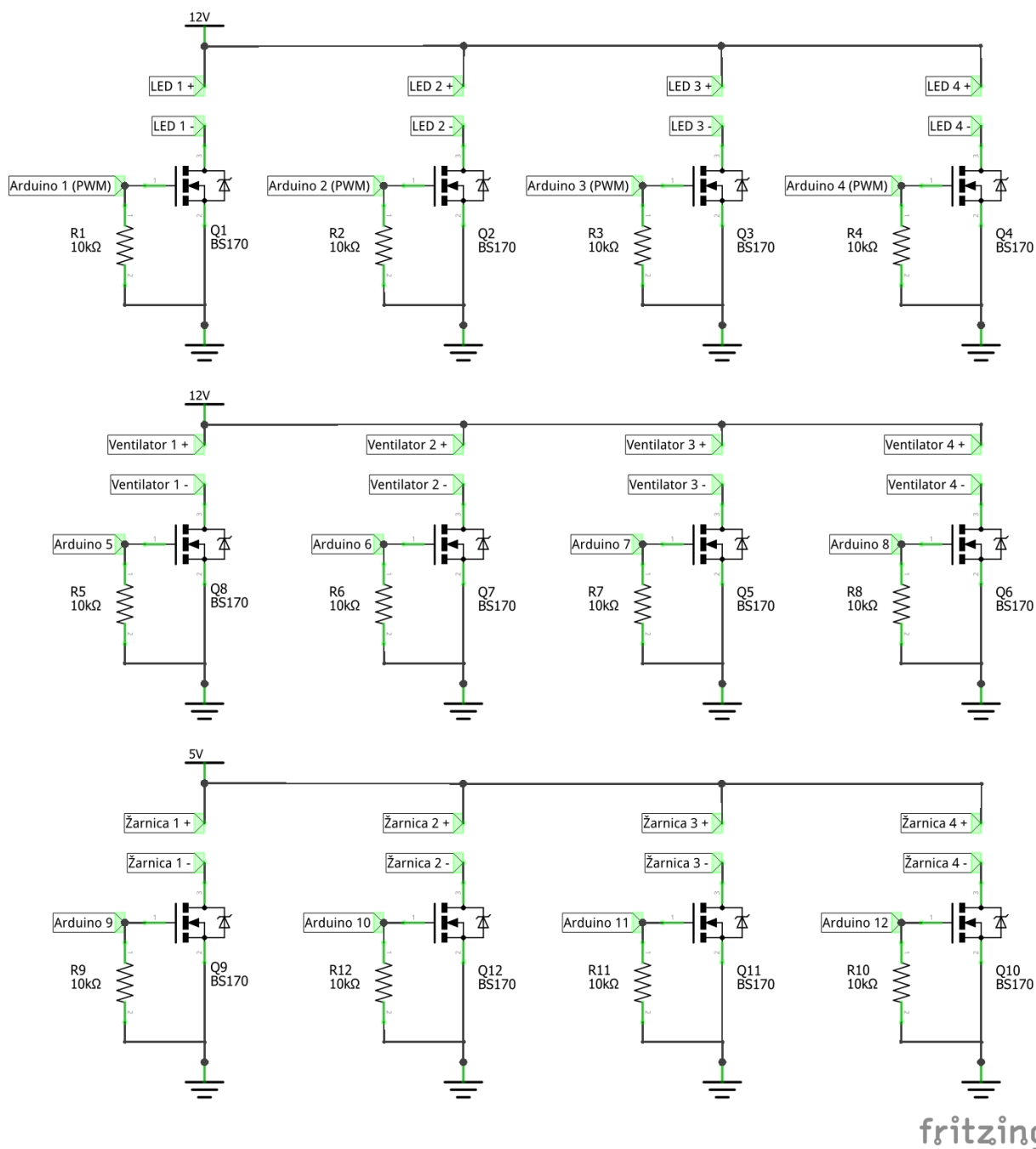
Stikalno vezje mosfet vključuje trikrat po štiri vrste mosfetov BS170, ki s ploščico Arduino nadzirajo napetost na treh prej omenjenih porabnikih. Shemi vezja sta prikazani v nadaljevanju. Če na »gate« mosfeta priključimo izhod PWM (izhod z pulzno-širinsko modulacijo), lahko napetost U_{gs} spreminjamo od 0 V do 5 V. To pomeni, da lahko nadziramo, kolikšen bo I_{ds} in posledično napetost na porabniku. Tako lahko zatemnimo trakove LED. Ti mosfeti imajo nazivni tok 500m A in 60 V, v praksi pa bi jih nadomestili z mosfeti, ki prenesejo večje tokove in napetosti. Deloma bi jih lahko nadomestili tudi z releji, ki pa delujejo kot stikalo, ne pa kot regulator, ki ga potrebujemo za trakove LED.

Spodnja slika prikazuje shemo vezave posameznega mosfeta. Porabnik je priključen med »Porabnik +« in »Porabnik -«. »Porabnik +« je na potencialu +12 V, »Porabnik -« pa je priključen na »drain«. V primeru žarnice kot porabnika je »Porabnik +« na potencialu 5 V.



Slika 5: Shema vezave posameznega mosfeta BS170.

Z vezjem na zgornji sliki lahko le krmilimo eno komponento, zato je končno vezje dvanajst kopij zgornjega primera, in sicer štirje mosfeti v treh vrstah. Vsaka vrsta je namenjena krmiljenju enega tipa porabnika. Shema končnega vezja je sledeča:

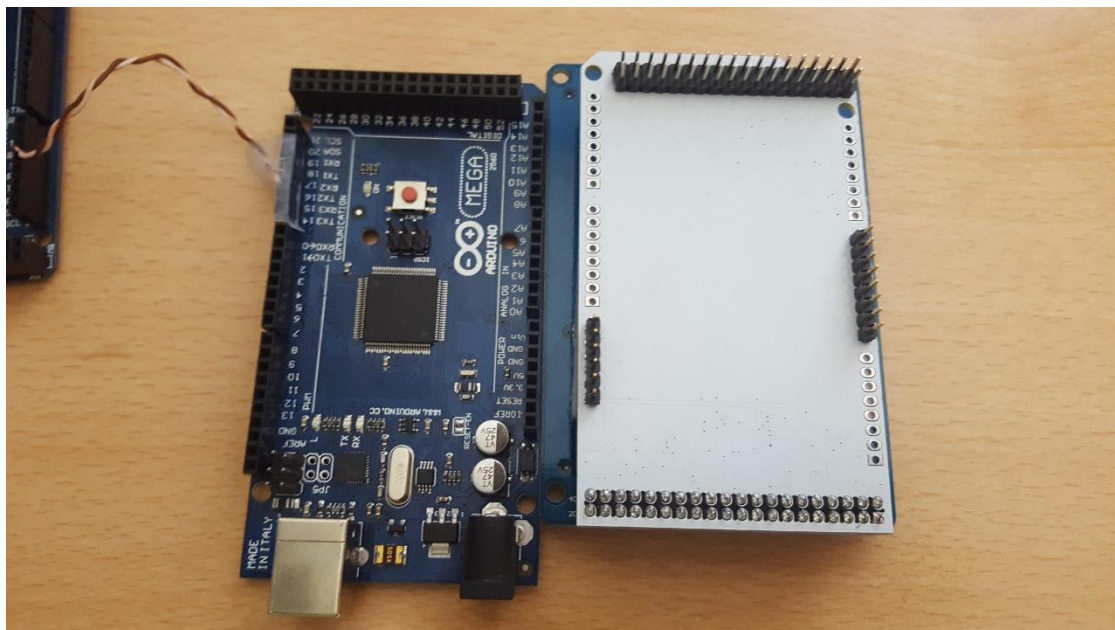


Slika 6: Shema vezave celotnega mosfet vezja

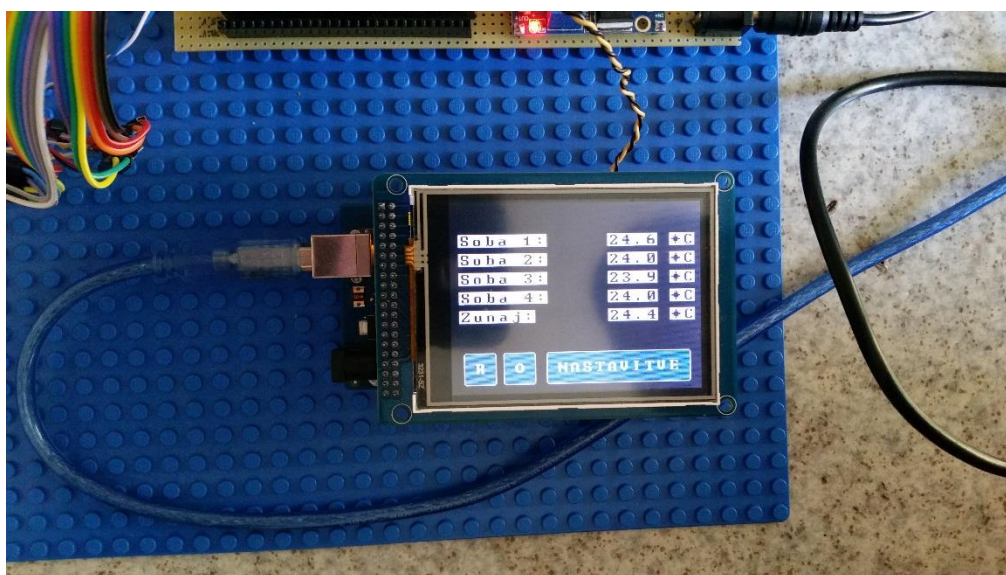
4.4 ZASLON NA DOTIK (TOUCHSCREEN MODUL)

Na dodatnem mikrokontrolerju Arduino je zaslon na dotik premera 8.1 cm z ločljivostjo 320x240 pikselov. Za dva mikrokontrolerja sva se odločila zato, ker je modul z zaslonom na dotik pritrjen na adapter, da ga lahko uporabljamo kot »shield«, kar pomeni, da smo priključke modula, ki so razporejeni vzporedno v dve vrsti, z adapterjem razporedili tako, da lahko modul z adapterjem priključimo neposredno, brez kablov na ploščico Arduino Mega. Modul sicer ne potrebuje toliko vhodov iz Arduina, a adapter vseeno fizično zakrije skoraj vse. To pomeni, da če hočeva na Arduino imeti prej omenjeni modul, ne moreva priključiti nanj ostalih senzorjev, krmiliti LED trakov, ventilatorjev ipd. Težavo sva rešila z dodatnim mikrokontrolerjem za zaslon na dotik. Med sabo se

krmilnika »pogovarjata« preko serijske povezave. Arduino Mega ima namreč 3 dodatne (TX1 & RX1, TX2 & RX2, TX3 & RX3) poleg TX0 & RX0, ki sta rezervirana za komunikacijo z računalnikom.



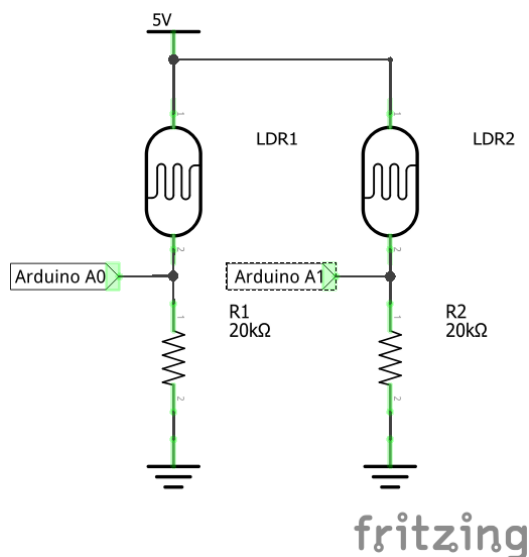
Slika 7: Arduino Mega 2560 in priključki na adapterju za modul z zaslonom na dotik



Slika 8: Modul z zaslonom na dotik

4.5 SVETLOBNI SENZOR

Fotoupor je pritrjen na model in je povezan na +5 V, z druge strani pa na analogni vhod na mikrokontrolerju. Iz tega vhoda je narejena povezava na upor z vrednostjo 20 k Ohm, ki je povezan z potencialom 0 V. Ko se spremeni razmerje upornosti, se spremeni tudi razmerje napetosti. Arduino tako bere napetost na pull-down uporu in nam poda informacijo o količini svetlobe v prostoru.



Slika 9: Shema vezave fotoupora

4.6 TEMPERATURNI SENZOR DS18B20

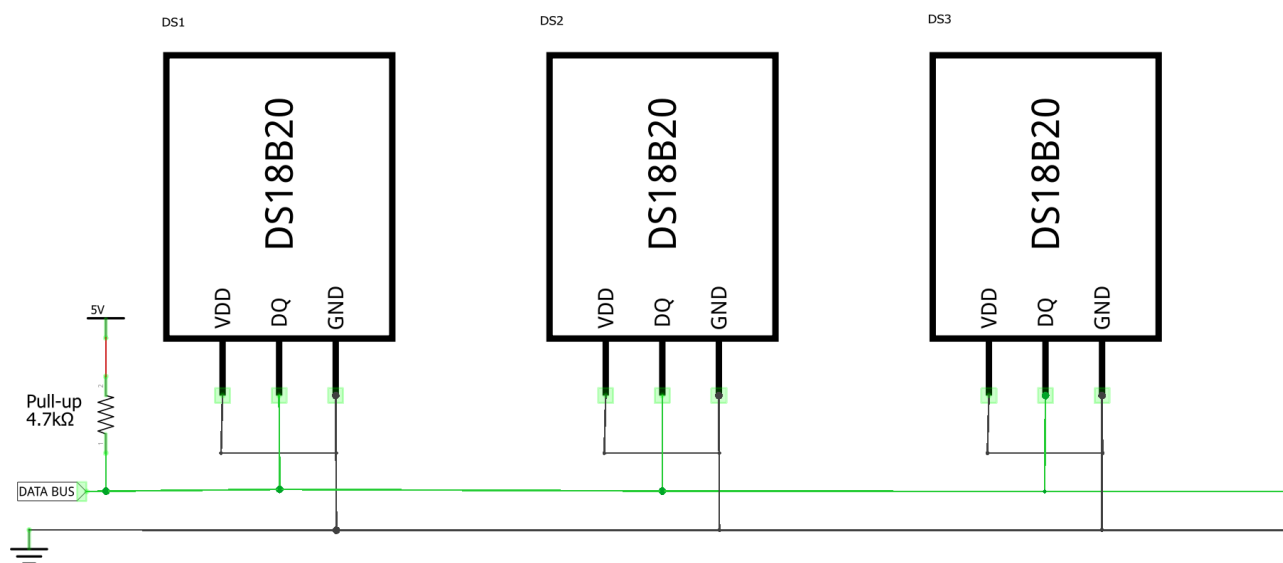
DS18B20 je digitalni senzor za merjenje temperature. Omogoča merjenje med -55 in $+125^{\circ}\text{C}$ in omogoča od 9 do 12-bitno merjenje temperature preko enožilne povezave, ki ji s tujko pravimo OneWire. Ker ima vsak senzor unikatno serijsko številko, jih lahko vežemo več na isto povezavo (vzporedno). Program jih potem razvrsti po tej številki in jim določi indekse. Natančnost sensorja pri 12-bitni resoluciji je $\pm 0,06^{\circ}\text{C}$.

Deluje lahko v sistemih z napetostjo od 3 do 5.5 V. Običajno ga napajamo tako, da Vdd kontakt priključimo na napetost 5 V, GND kontakt pa na 0 V. Med Vdd in Data kontaktom potrebuje »pull-up« upor z upornostjo 4,7 k Ohma.

Ti senzorji imajo poleg običajnega načina napajanja še možnost parazitnega napajanja. Napetost za branje, pisanje in pretvorbo vrednosti lahko priskrbimo kar preko priključka Data, brez dodatnega vira napetosti. Uporabila sva parazitno napajanje, da sta zunanja priključka (Vdd in GND) na potencialu 0 V, srednja pa na digitalni vhod na Arduino. Preko tega se tako čip hkrati napaja in pošilja podatke mikrokontrolerju.

Tako ni nevarnosti, da bi prišlo do zamenjave Vcc in GND priključkov, saj sta obe na 0V, srednji (Data) kontakt pa je priključen na Arduino. Ta sprememba naredi vezavo bolj preprosto, saj tako potrebujemo le dve žici in ne tri kot pri običajnem napajanju.

Na sliki spodaj je prikazana shema vezave teh senzorjev.



Slika 10: Shema vezave več senzorjev DS18B20 z parazitnim napajanjem.

4.7 LED TRAK

To je običajen LED trak. Na voljo sva imela tudi visoko svetlečega, a je že običajen dovolj svetel. Poleg tega tudi porabi manj toka.

Sestavljen je iz več 5 cm dolgih segmentov. Mesta, kjer se stikajo, so označena z navpično črto, zraven pa sta po dva para kontaktov za napajanje. Na teh mestih lahko trak prerežemo in tako prilagodimo dolžino. Na kontakte se potem prispajkajo žice, preko katerih napajamo trak. Napajamo ga z 12 V, posamezen segment pa potrebuje 20 mA toka. V vsaki sobi so štiri segmenti, torej je njihov skupni tok 80 mA. Na spodnji strani traka je lepilo, da ga z lahkoto pritrdimo na površino. To je skrajšalo čas izdelovanja makete.

Na spodnji sliki je prikazan trak z štirimi segmenti, vidna pa so tudi stičišča teh.



Slika 11: LED trak, štiri segmenti

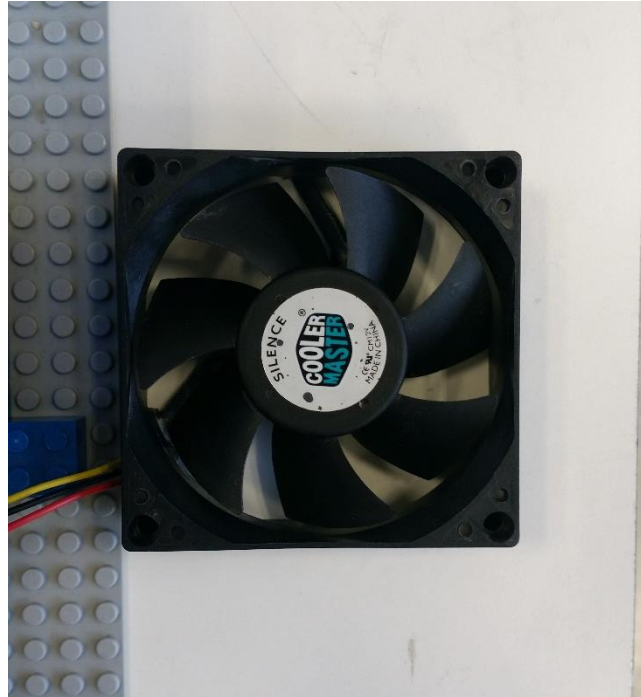
4.8 VENTILATOR (HLADILNO TELO)

Če želimo regulirati temperaturo v prostoru, potrebujemo nekaj, kar lahko ohladi zrak v prostoru. V praksi bi za to lahko uporabili prezračevalne sisteme na električno upravljanje, ki bi ob prejemu signala iz mikrokontrolerja odprli okna in tako temperaturo sobe znižali. Poleti bi problem višje

temperature zunanjega zraka rešili tako, da bi okna odprli čez noč. Alternativa takim sistemom je klimatska naprava, ki bi jo krmilili z mikrokontrolerjem.

Na maketi za hlajenje senzorjev uporabljamo računalniški ventilator. Ta jih seveda ne more ohladiti na temperaturo, nižjo od temperature prostora, v katerem smo, ker zraka ne ohladi, ampak ga samo premeša. To ni problem, če minimalno sprejemljivo temperaturo nastavimo tako, da je višja od temperature prostora.

Sam ventilator je dimenzij 8 cm x 8 cm, deluje pa na napetosti 12 V. Porabi od 100 do 180 mA toka. V vsaki sobi je eden, kar pomeni, da so na maketi štirje ventilatorji.



Slika 12: Ventilator

4.9 ŽARNICA (GRELNO TELO)

Če imamo v prostoru hladilno telo, potrebujemo tudi grelno telo. V praksi je to lahko električni radiator, kalorifer, stenski grelnik, IR paneli ali pa radiatorje opremimo z brezžičnimi termostatskimi ventili, ki jih krmilimo z mikrokontrolerjem.

Na maketi delo grelnega telesa opravlja 7 V žarnica z žarilno nitko. Narejena je sicer za 7 V in 300 mA toka, a je na maketi priključena kar na 5 V, ker napetosti 7 V ni na voljo. Ko je priključena na 5 V, je to dovolj, da senzor zagreje za približno 3°C.

Moč ki se troši na žarnici pri napetosti 7 V je $P = 7 \text{ V} \times 0,3 \text{ A} = 2,1 \text{ W}$. Če je napetost enaka 5 V, skozi njo teče tok 220 mA, torej je dejanska moč, ki se troši na posamezni žarnici $P = 5 \text{ V} \times 0,22 \text{ A} = 1,1 \text{ W}$.

Vsaka žarnica potrebuje tudi podnožje, in sicer velikosti E10. Je izolirano z plastiko. Žarnica z podnožjem je prikazana na spodnji sliki.



Slika 13: Žarnica z podnožjem

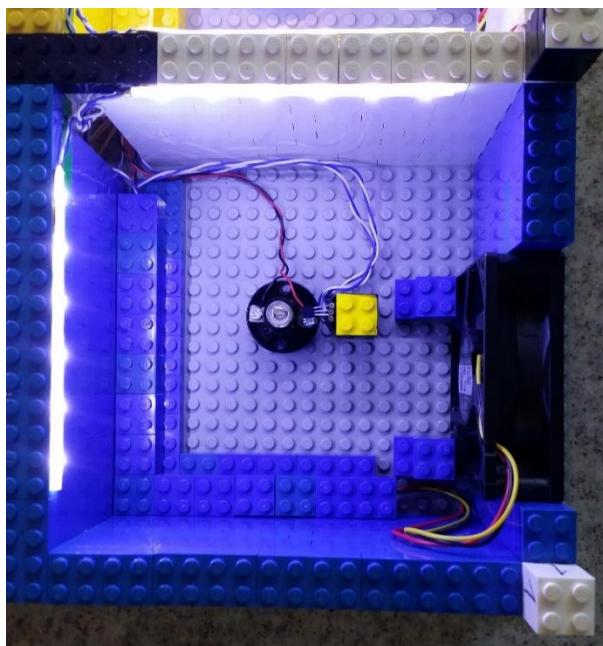
5. IZDELAVA

5.1 MODEL HIŠE

Za izdelavo modela sva najprej pomislila na les. Ker nihče od naju nima preveč izkušenj z obdelovanjem lesa, je prišla v poštev alternativa, Lego kocke. Glavni faktor za to odločitev je bila hitra in preprosta izdelava modela. Za izdelavo modela z kockami sva porabila 2 do 3 ure, če bi uporabila les, bi izdelava trajala občutno dlje. Dodatna prednost Lego kock je preprosto spreminjanje konstrukcije, kar ne velja za les.

Dimenzije celotne hiše so 33,5 cm x 33,5 cm, posamezne sobe pa 16 cm x 16 cm. Model hiše ima štiri sobe in v vsaki je v steni prostor za ventilator.

Na spodnji sliki je prikazana postavitve posameznih elementov po sobi.

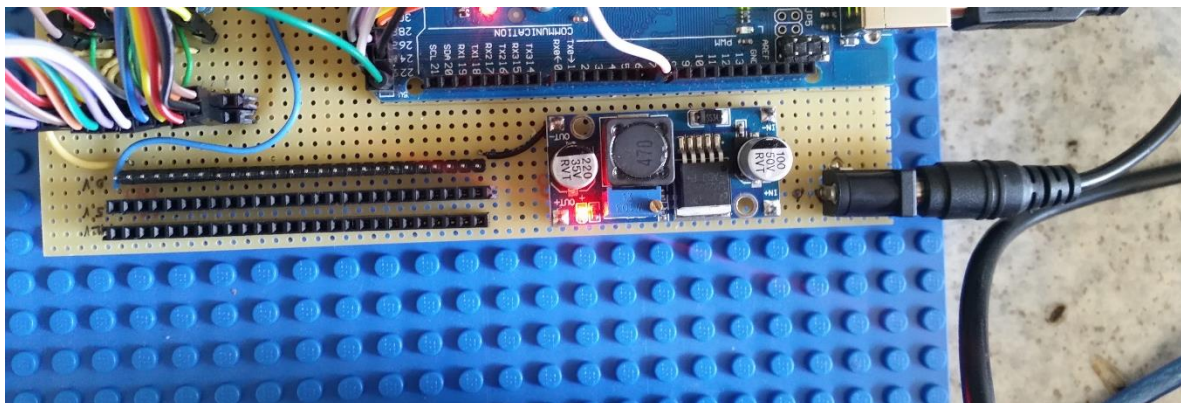


Slika 14: Postavitev posameznih elementov po sobi

5.2 NAPAJANJE

Model napaja adapter, ki pretvori 230 V izmenične napetosti v 12 V enosmerne napetosti. Priklučen je na 2.1 mm vtičnico v desnem spodnjem kotu vezja. Povezana je z letvicama za napajanje 12 V komponent ter z regulatorjem napetosti LM2596, katerega izhod je povezan še na eno letvico, na kateri je napetost 5 V. Preko nje napajamo Arduino mikrokrmilnika, vseh senzorje ter žarnice.

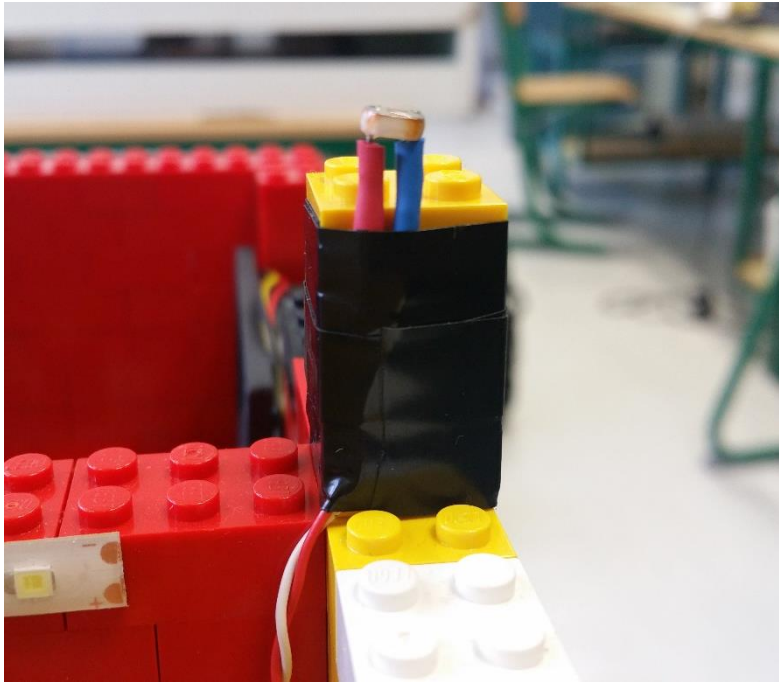
Adapter, pretvornik in kontakti so prispajkani na ploščo z bakrenimi očesi (angl. perfboard). Levo na spodnji sliki so vidne letvice (od zgoraj navzdol: 0 V, 5 V 12 V), na sredini je regulator napetosti, na desni pa je 2.1 mm priključek za napajalnik.



Slika 15: Napajanje modela

5.3 SVETLOBNI SENZOR

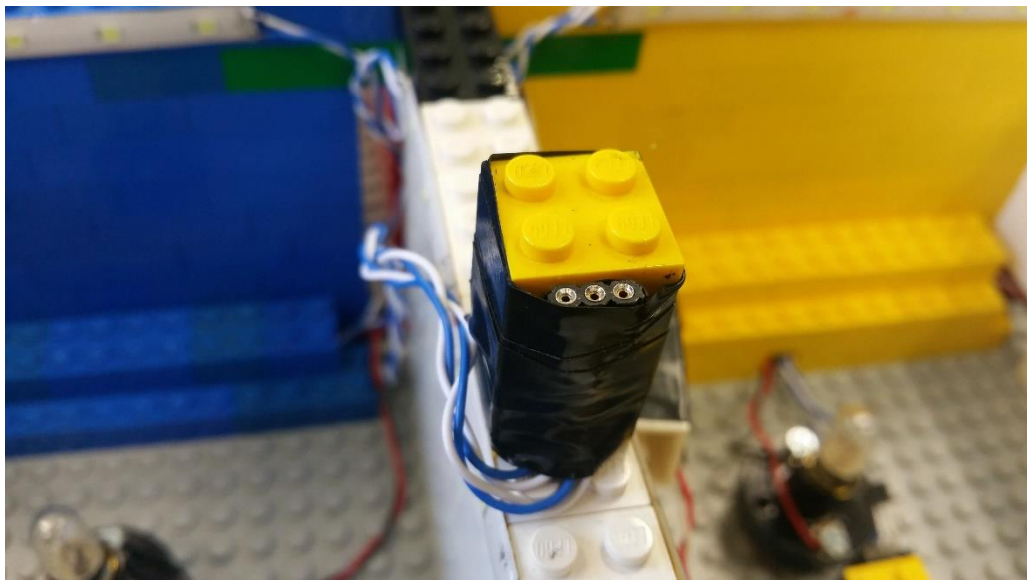
Na modelu sta dva svetlobna senzorja. Sestavljena sta iz fotouporov, ki se nahajata na modelu hiše, in »pull-down« uporov skupaj s kontakti. Ti se nahajajo na vezju. Posamezen fotoupor je pritrjen na stolpec iz treh Lego kock za preprosto premikanje. To je prikazano na naslednji sliki.



Slika 16: Fotoupor

5.4 TEMPERATURNI SENZOR

Senzor DS18B20 napajamo na parazitni način. Zunanja priključka (Vdd, GND) sta na potencialu 0V. Srednji priključek je priključen na vezje, na katerem je tudi potreben »pull-up« upor in ženski kontakt za priklop na Arduino. Žice ne smejo biti prispajkane neposredno na senzor, ker je bilo potrebno menjavati vrstni red. Program jim namreč samodejno določi sobo glede na serijsko številko. Žice so prispajkane na ženske konektorje, po tri za vsak senzor. Kot že omenjeno, sta zunanja priključka povezana.

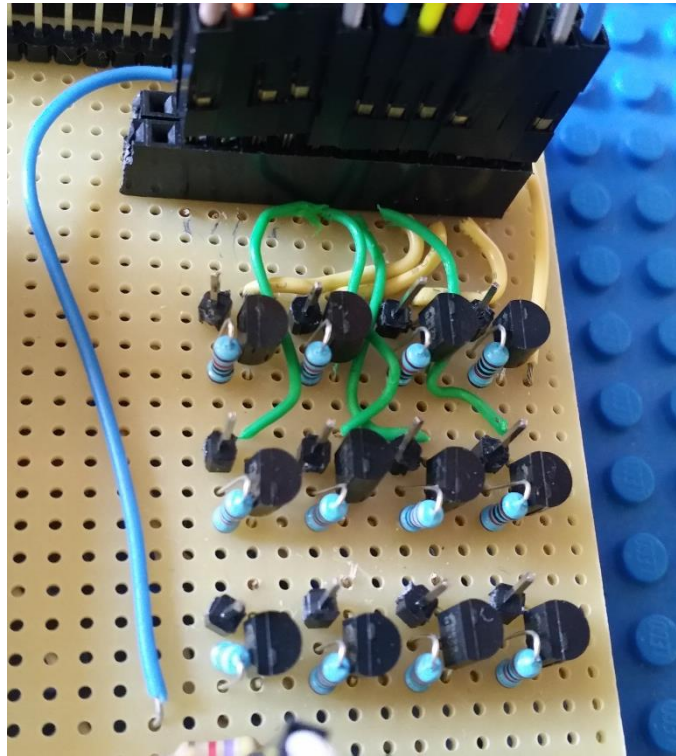


Slika 17: Priključek za DS18B20

5.5 MOSFET VEZJE

Shematika vezja je prikazana na sliki 6. Oznake na shematiki, na katere se priključi porabnik, so na vezju letvice, ki so na sliki 18 prikazane zgoraj. Povezane so s 3 vrsticami mosfetov. Prva vrstica krmili

LED trakove, druga vrsta ventilatorje, tretja pa žarnice. Zraven vsakega mosfeta je moški priključek, na katerega priključimo izhode na Arduino ploščici.

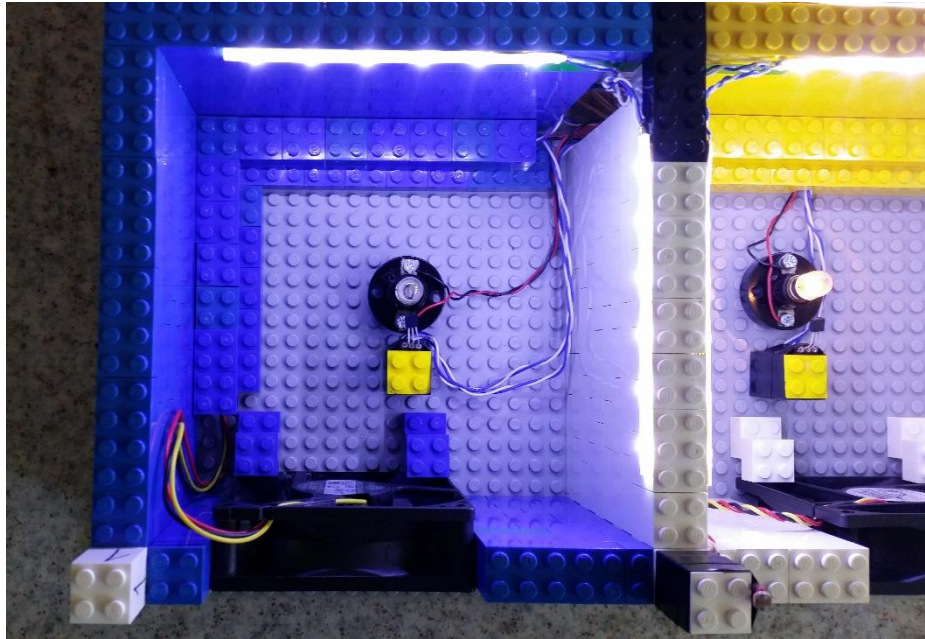


Slika 18: Mosfet stikalno vezje

5.6 VENTILATOR, ŽARNICA IN LED TRAKOVI

Računalniški ventilator s kabli povežemo s priključki Vcc in GND na mosfet vezje.

Priključke Vcc in GND na ventilatorju s kabli povežemo na letvice na mosfet vezju, ki so jim namenjene. Enako storimo z LED trakovi. Pri tem je potrebno paziti na polariteto, ker imajo ventilatorji diodo, ki preprečuje vrtenje v nasprotni smeri, prav tako LED diode ne delujejo z obrnjeno polariteto. Tudi žarnice priključimo na letvice, pri njih pa ni potrebno pazit na polariteto. Kable speljemo po kanalih iz Lego kock, kar je vidno na spodnji sliki.



Slika 19: Razporeditev žarnic, ventilatorjev in LED trakov na maketi

6. PROGRAM

6.1 GLAVNI MIKROKONTROLER

Program na glavnem mikrokontrolerju lahko v glavnem razdelimo na sledeče dele: nadzor in branje temperature, nadzor in branje svetlosti, EEPROM in serijsko komunikacijo.

6.1.1 BRANJE SENZRJEV DS180

Branje senzorjev je relativno preprosto. Zasluga za to gre knjižnicama OneWire in DallasTemperature, ki opravita vse pretvorbe. Tako za branje temperature potrebujemo le nekaj funkcij, kot so recimo:

- OneWire ourWire(*pin*) - vzpostavi sejo za komunikacijo z OneWire napravami;
- DallasTemperature sensors(&ourWire) - pove knjižnici DallasTemperature, naj uporabi OneWire knjižnico.

Na spodnji sliki je prikazana koda, ki prebere temperaturo in jo zapiše v spremenljivko. Zapiše jih v spremenljivke tipa »float«, pri tem pa uporablja for zanko v kombinaciji z arrayi. Prebrano temperaturo potem izpiše na serijski vmesnik, da si jo lahko ogledamo na računalniku.

```
//Temperatura////////////////////////////////////  
Serial.println("BRANJE TEMPERATURE");  
sensors.requestTemperatures();  
  
for (int i = 0; i <= 4; i++) {  
temp[i] = sensors.getTempCByIndex(i);  
}
```

Slika 20: Koda za branje senzorjev DS18B20

6.1.2 NADZOR TEMPERATURE

Nadzor temperature je podobno kot branje narejen z uporabo for zanke in arrayi, ker je tako potrebno veliko manj kode. Narejen je z if zankami, ki preverjajo, če je temperatura višja od maksimalne oziroma nižja od minimalne in tako vklopi ali izklopi ventilator ali žarnico. Če senzor ob zahtevi vrne vrednost 85°C (ko senzor ni pravilno napajan), se ne zgodi nič.

```
289 //vklop/izklop gretja
290 for (int i = 0; i <= 3; i++) {
291     if (temp[i] < minTemp[i] && (int)temp[i] != 85) {
292         Serial.print("soba ");
293         Serial.print(i);
294         Serial.print(" mrzlo: ");
295         Serial.println(temp[i]);
296         digitalWrite(ventilator[i], LOW);
297         digitalWrite(grelnaLuc[i], HIGH);
298     }
299     else {
300         digitalWrite(grelnaLuc[i], LOW);
301     }
302
303     if (temp[i] > maxTemp[i] && (int)temp[i] != 85) { //vklop/izklop hlajenja
304         Serial.print("soba ");
305         Serial.print(i);
306         Serial.print(" vroce: ");
307         Serial.println(temp[i]);
308         digitalWrite(grelnaLuc[i], LOW);
309         digitalWrite(ventilator[i], HIGH);
310     }
311     else {
312         digitalWrite(ventilator[i], LOW); //izklopi ventilator
313     }
314 }
```

Slika 21: Koda za nadzor temperature

6.1.3 BRANJE SVETLOBNIH SENZORJEV

Branje poteka s funkcijo `analogRead(pin)`. Zapiše vrednost od 0 do 1024 v integer spremenljivko. To številko je potrebno pretvoriti na območje od 0 do 255 za funkcijo `analogWrite(pin)`. Pretvarjamo jo z funkcijo `map()`. Območje pretvorbe (spremenljivko namesto števil 0 in 255) lahko spreminjamo preko zaslona na dotik in tako določimo, kako močno vrednost foto senzorja vpliva na svetlost sobe.

6.1.4 NADZOR SVETLOSTI

Svetlost sob se nadzira s funkcijo `analogWrite()`. Uporablja vrednosti od 0 do 255. Spet je za optimizacijo kode uporabljena for zanka z arrayi.

Na spodnji sliki je do 336. vrstice prikazana koda za branje svetlobnih senzorjev, za tem pa je koda za nadzor svetlosti.

```

321 //svetloba////////////////////////////////////
322 Serial.println();
323 Serial.println("SVETLOBA");
324
325 for (int i = 0; i <= 3; i++) {
326
327     Serial.print("Analogna vrednost fotoupera ");
328     Serial.print(i);
329     Serial.print(" : ");
330     Serial.println(analogRead(fotoupor[i]));
331     fotoVrednost[i] = map(analogRead(fotoupor[i]), 0, 1024, highvalue, lowvalue);
332     Serial.print("Osvetlitev sobe " + i);
333     Serial.println(" : " + fotoVrednost[i]);
334 }
335
336 //LED
337 for (int i = 0; i <= 3; i++) {
338     if (stikalo[i] == true) {
339         analogWrite(led[i], fotoVrednost[i]);
340     }
341     else {
342         digitalWrite(led[i], LOW);
343     }
344 }
345 Serial.println();
346 }

```

Slika 22: Koda za branje in nadzor svetlosti

6.1.5 SERIJSKA KOMUNIKACIJA

Glavni mikrokontroler (v nadaljevanju Arduino 1) ter mikrokontroler z zaslonom na dotik (Arduino 2) komunicirata prek serijske povezave, ki omogoča prenos informacij med več napravami. Za prenos sta potrebni le dve žici, priključeni na TX2 in RX2 priključka za oddajanje in sprejemanje signala. Ko Arduino 2 potrebuje podatke za prikaz na zaslon, pošlje vnaprej določeno kodo, ki jo Arduino 1 analizira in vrne zahtevane informacije.

Te kode so štirimestne številke, vsaka številka pa ima določen pomen: če je **prva številka 1** pomeni da je to koda za temperaturo, če pa je 2, pa pomeni, da gre za razsvetljavo. **Druga številka** pove vrsto informacije npr. 1 za trenutno temp., 2 za minimalno itd.; **3. številka** pove številko sobe (od 0 do 3); če je **4. številka 0** pomeni da gre za komando za branje informacij, če pa je 1, pa za pisanje. Enki tako sledijo decimalke, ki jih želimo nastaviti kot določeno vrednost.

Spodaj je seznam kod, ki jih uporabljata mikrokontrolerja . Če je komanda za branje (read), Arduino 2 pošlje to komando Arduino 1, ki mu vrne željen podatek. Če je komanda za pisanje (write), Arduino 2 pošlje kodo Arduino 1 in tako nastavi vrednost neke nastavitve.

100 – uparjanje mikrokontrolerjev

- 101 Arduino 1
- 102 Arduino 2

1000 – temperatura

- 1100 - Trenutne temp. po sobah
 - 1100
 - 1110
 - 1120
 - 1130
 - 1140 (zunanja)
- 1200 – branje minimalne temperature po sobah
 - 1200

1210

1220

1230

- 1201 – pisanje minimalne temperature po sobah

1201, *vrednost*

1211, *vrednost*

1221, *vrednost*

1231, *vrednost*

- 1300 – Branje maksimalne temperature po sobah

1300

1310

1320

1330

- 1301 – pisanje maksimalne temperature po sobah

1301, *vrednost*

1311, *vrednost*

1321, *vrednost*

1331, *vrednost*

2000 razsvetljava

- 2100 Trenutna vrednost

2100

2110

2120

2130

- 2200 – Branje minimalne svetlosti po sobah

2200

2210

2220

2230

- 2201 – Pisanje minimalne svetlosti po sobah

2201, *vrednost*

2211, *vrednost*

2221, *vrednost*

2231, *vrednost*

- 2300 – Branje maksimalne svetlosti po sobah

2300, *vrednost*

2310, *vrednost*

2320, *vrednost*

2330, *vrednost*

- 2301 – Pisanje maksimalne svetlosti po sobah

2301, *vrednost*

2311, *vrednost*

2321, *vrednost*

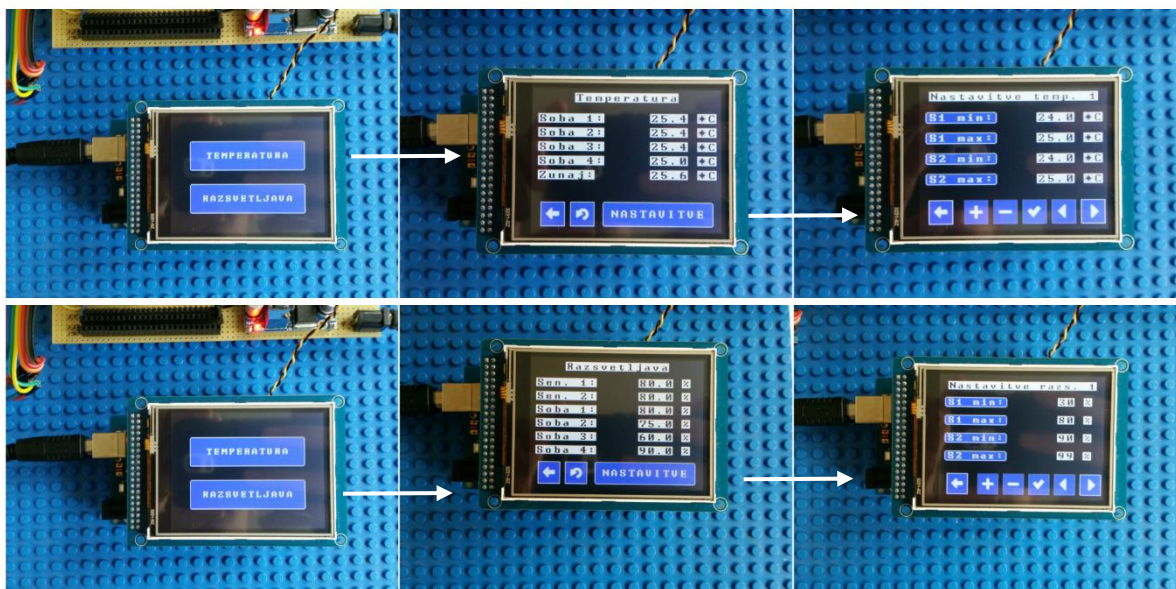
2331, *vrednost*

6.1.6 EEPROM

Electrically Erasable Programmable Read-Only Memory ali Električno izbrisljiv programirljiv bralni pomnilnik je tip spomina, v katerega lahko trajno shranjujemo podatke tudi po izklopu napajanja na mikrokontrolerju za razliko od RAM-a ki se izbriše. Zaradi tega je popoln za shranjevanje nastavitvev, kot so minimalne in maksimalne temperature po sobah. Njegova slabost je, da ima omejeno število pisanj in branj na le 100 000, kar bi bilo za praktično uporabo relativno malo.

6.2 MIKROKONTROLER Z ZASLONOM NA DOTIK

Vmesnik na zaslonu na dotik sva razdelila sistematično: na domačem meniju imamo možnosti Temperatura ter Razsvetljava. Pritisk na posamezen gumb prikaže trenutne vrednosti sensorjev za temperaturo oz. svetlobnih sensorjev in trenutno svetlost luči. Na obeh imamo tudi gumb Nastavitve, na katerem lahko urejamo minimalne in maksimalne temperature posameznih sob ter maksimalno svetlost luči in vklop ter izklop luči po posameznih sobah. Za razliko od drugih programskih okolij (recimo Visual Studio), kjer vmesnik sestavimo grafično in se vse avtomatsko pozicionira, je treba pri Arduinu vse gumbе, besedilo ter vse ostalo ročno pozicionirati z vpisom koordinat. Gumbе lahko preprosto ustvarimo s funkcijo `addButton()`, ki ji v parametre vnesemo x in y koordinati, širino in višino gumba in besedilo, ki ga želimo prikazati kot napis v gumbu, ki jo nastavimo v spremenljivko. Po tem ko ustvarimo vse željene gumbе, pa je potrebno uporabiti funkcijo `drawButtons()`, ki gumbе tudi prikaže na zaslonu. Če želimo izvedeti, ali je bil gumb pritisnjen uporabimo funkcijo `checkButtons()`, ki nam pove kateri (če sploh kateri) gumb je bil pritisnjen, na osnovi česar potem izvedemo željeno kodo za gumb. Gumbе lahko tudi vključimo oziroma izključimo, kar pomeni, da se gumb ne bo odzval na pritisk; to je na primer uporabno pri gumbih za prestavljanje strani naprej in nazaj, ko eden od teh dveh ni na voljo. To naredimo z funkcijama `enableButton()` in `disableButton()`, ki imata za parametre spremenljivko gumba ter boolean ali želimo osvežiti gumb takoj ali ne.



Slika 23: Grafični vmesnik

```
158 void temp()
159 {
160     myGLCD.clrScr();
161     myButtons.deleteAllButtons();
162     myGLCD.print("Temperatura", CENTER, 5);
163     btnR = myButtons.addButton( 15, 190, 40, 40, RETURN);
164     btnRef = myButtons.addButton( 65, 190, 40, 40, REFRESH);
165     btn1 = myButtons.addButton( 120, 190, 185, 40, "NASTAVITVE");
166     myButtons.drawButtons();
167
168     myGLCD.print("Soba 1:", 10, 40);
169     myGLCD.printNumF(serialSender(1100), 1, 200, 40);
170     myGLCD.print("*C", 280, 40);
171     myGLCD.print("Soba 2:", 10, 64);
172     myGLCD.printNumF(serialSender(1110), 1, 200, 64);
173     myGLCD.print("*C", 280, 64);
174     myGLCD.print("Soba 3:", 10, 88);
175     myGLCD.printNumF(serialSender(1120), 1, 200, 88);
176     myGLCD.print("*C", 280, 88);
177     myGLCD.print("Soba 4:", 10, 112);
178     myGLCD.printNumF(serialSender(1130), 1, 200, 112);
179     myGLCD.print("*C", 280, 112);
180     myGLCD.print("Zunaj:", 10, 136);
181     myGLCD.printNumF(serialSender(1140), 1, 200, 136);
182     myGLCD.print("*C", 280, 136);
183     while (true)
184     {
185         if (myTouch.dataAvailable() == true)
186         {
187             btnPressed = myButtons.checkButtons();
188             if (btnPressed == btnR)
189             {
190                 mainMenu();
191             }
192             else if (btnPressed == btnRef)
193             {
194                 temp();
195             }
196             else if (btnPressed == btn1)
197             {
198                 tempSettings();
199             }
200         }
201     }
202 }
```

Slika 24: Primer kode za grafični vmesnik

```
77 float serialSender(float req, bool decimals = false)
78 {
79     float received;
80     Serial2.flush();
81     while (true)
82     {
83         if (decimals)
84         {
85             Serial2.println(req, 4);
86         }
87         else
88         {
89             Serial2.println(req);
90         }
91         delay(1500);
92         if (Serial2.available() > 0)
93         {
94             Serial.println("RECEIVED");
95             received = Serial2.parseFloat();
96             Serial.println("Received request: " + String(received));
97             if (received == 0)
98             {
99                 Serial.println("interfereance");
100             }
101             else
102             {
103                 return received;
104             }
105         }
106     }
107 }
108 }
```

Slika 25: Funkcija za branje podatkov serijske povezave na krmilniku z zaslonom na dotik

```
80 float serialReceive()
81 {
82     float received;
83     float change;
84     if (Serial2.available() > 0)
85     {
86         received = Serial2.parseFloat();
87         Serial.println("//////////SERIAL//////////");
88         Serial.print("Received request: ");
89         Serial.println(received, 3);
90         Serial.println("");
91         if (received == 0)
92         {
93             Serial.println("Interfereance");
94         }
95         if (received > 1000.00)
96         {
97             if ((int)received == 1100.00)
98             {
99                 Serial2.println(temp[0]);
100            }
101            else if ((int)received == 1110)
102            {
103                Serial2.println(temp[1]);
104            }
105            else if ((int)received == 1120)
106            {
107                Serial2.println(temp[2]);
108            }
109            else if ((int)received == 1130)
110            {
111                Serial2.println(temp[3]);
112            }
113            else if ((int)received == 1140)
114            {
115                Serial2.println(temp[4]);
116            }
117            else if ((int)received == 1200)
118            {
119                Serial2.println(minTemp[0]);
120            }
121            else if (received > 1201.00 && received < 1202)
122            {
123                Serial2.println(1001);
124                change = (received - 1201)*100;
```

Slika 26: Del funkcije za obdelovanje podatkov serijske povezave na glavnem krmilniku

7. UGOTOVITVE IN REZULTATI

Na začetku sva si postavila štiri hipoteze. Med izdelovanjem raziskovalne naloge sva te potrdila.

- Avtomatizirana regulacija temperature v prostorih
- Avtomatizirana regulacija količine svetlobe v prostorih
- Preprost vpogled v meritve vsakega sensorja z zaslonom na dotik
- Nastavljanje željene temperature in jakosti svetlobe v posamezni sobi

Hipoteze sva potrdila, toda obstaja veliko možnosti za optimizacijo in izboljšave. Primer je hitrost pošiljanja podatkov med mikrokontrolerjema preko serijske povezave. Ta ni preveč zadovoljiva, tudi povezava je včasih nezanesljiva. Včasih problem povzročajo sensorji za temperaturo, ki ob slabem stiku (kadar se maketa zatrese, premika ipd.) vrnejo vrednost 85°C. Težavne so tudi žarnice, ker senzor zagrejejo za največ 3°C zaradi premajhne moči. Vsake toliko časa pa povzročajo probleme tudi zaslon na dotik, ki se iz neznanega razloga preneha odzivati. V tem primeru ga je potrebno resetirati. To najlažje storimo tako, da za trenutek prekinemo napajanje mikrokontrolerja, najboljša rešitev pa bi bil zunanja tipka, ki bi povezala reset in GND priključek.

8. ZAKLJUČEK

Za to raziskovalno nalogo je bila potrebna velika količina znanja, najzahtevnejše pa je bilo vse stvari povezati v enoten sistem. Poglobiti se je bilo potrebno v okolje Arduino, predvsem kar se tiče zaslona na dotik. To je bil najin prvi program, ki je skupaj presešel 1000 vrstic. Glede na to, da nihče od naju še nikoli ni sodeloval pri izdelavi raziskovalne naloge, sva z rezultatom zadovoljna.

Maketo bi lahko nadgradila v veliko pogledih. Lahko bi izdelala ohišje za mikrokontrolerja, maketo po vrhu pokrila z pleksi steklom in uporabila močnejše žarnice, da bi se zagrel tudi zrak v sobi, ne pa samo senzor. Dodala bi lahko tudi varnostni sistem, merilec koncentracije CO₂, izpisovanje podatkov na grafu, zaslon na dotik nadomestila z aplikacijo na telefonu, programe gretja, ki bi se prilagajali času dneva ter podobno. Najin dolgoročni cilj je, da bi koncept aplicirala na pravo sobo.

9. VIRI

Načrtovanje in izvedba pametne hiše z mikrokrmilnikom Arduino. Dostopno na:
<https://dk.um.si/Dokument.php?id=82065> [10. 12. 2016].

LM2596 regulator napetosti. Dostopno na: <http://www.hotmcu.com/lm2596-buck-dcdc-module-p-50.html> [10. 12. 2016].

Stikalni regulatorji napetosti. Dostopno na:
ftp://ftp.scv.si/vss/franc_stravs/Elektronika%20v%20mehatroniki%20dodatki,%20%20tem.%20vpra%9A./Gradivo%20za%20EME%20okt.09.pdf [10. 12. 2016].

DS18B20 temperaturni senzor. Dostopno na: <http://pdf1.alldatasheet.com/datasheet-pdf/view/108957/MAXIM/DS18B20-PAR.html> [10. 12. 2016].

Parazitno napajanje DS18B20. Dostopno na:
<https://developer.mbed.org/users/snatch59/notebook/1-wire-parasitic-power/> [10. 12. 2016].

BS170 mosfet. Dostopno na: http://www.onsemi.com/pub_link/Collateral/BS170-D.PDF [10. 12. 2016].

PROGRAMSKE KNJIŽNICE :

UTFT. Dostopno na: <http://www.rinkydinkelectronics.com/library.php?id=51> [25. 1. 2016].

UTFT_Buttons. Dostopno na: <http://www.rinkydinkelectronics.com/library.php?id=61> [25. 1. 2016].

UTouch (ki se je 19. 6. 2016 preimenovala v URTouch) . Dostopno na:
<http://www.rinkydinkelectronics.com/library.php?id=92> [25. 1. 2016].

OneWire. Dostopno na: <https://github.com/PaulStoffregen/OneWire> [10. 12. 2016].

DallasTemperature. Dostopno na: <https://github.com/milesburton/Arduino-Temperature-Control-Library> [10. 12. 2016].

10. ZAHVALA

Velika zahvala gre mentorju, Mateju Kališku, inž. el., ki nama je pomagal pri delu in nama priskrbel veliko komponent, ki jih nisva imela.

Zahvaljujeva se tudi lektorici, dr. Tanji Jelenko, ki je delo jezikovno pregledala.

Zahvala gre tudi Gregorju Kramerju, univ. dipl. inž. el. in Andreju Grilcu, univ. dipl. inž. el. za svetovanje in pomoč pri reševanju problemov.

Pri delu nama je pomagal tudi sošolec Domen Romih, za kar se mu zahvaljujeva.

11. IZJAVA

IZJAVA

Mentor, Matej Halušek, v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi naslovom PAMETNI DOM Z ARDUINO MEGA 2560 MIKROKONTROLERJEM, katere avtorja sta Gasper Gyul, in Kristijan Šaln:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene 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 nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo 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, 13.3.2017

Podpis mentorja

Podpis odgovorne osebe