Pametno ogledalo

Kosić, Jerko

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:356536

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-24**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

PAMETNO OGLEDALO

Završni rad

Jerko Kosić

Osijek, 2019.

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. SKLOPOVLJE 2

2.1. Korišteno sklopovlje 2

2.1.1 Raspberry Pi 3B 2

2.1.2 LCD zaslon 7

2.1.3 DHT22 8

2.1.4 Eksperimentalna pločica 10

2.2. Kućište pametnog ogledala 12

3. PROGRAMSKO RJEŠENJE 14

3.1. Korištene tehnologije 14

3.1.1 Python 14

3.1.2 HTML 16

3.1.3 CSS i JavaScript 19

3.1.4 PHP i Apache 22

3.2. Dijelovi programa 23

3.2.1 Instalacija operacijskog sustava 23

3.2.2 Python i DHT22 24

3.2.3 PHP skripta 26

4. TESTIRANJE 28

4.1. Prikaz rada 28

4.2. Testiranje rada 28

5. ZAKLJUČAK 30

6. LITERATURA 31

7. SAŽETAK 32

8. ABSTRACT 33

ŽIVOTOPIS 34

PRILOZI 35

1. UVOD

U završnom radu objašnjen je jedan od načina kako napraviti pametno ogledalo. Pametno ogledalo na sebi ima ispisane podatke o datumu i vremenu. Osim toga pokazuje i podatke o trenutnoj temperaturi zraka i relativnoj vlažnosti zraka. Ogledalo koristi minijaturno računalo pomoću kojega sa senzora prikuplja podatke te ih prikazuje na LCD zaslonu. Svi podatci su zajedno prikazani na internet stranici. Zaslon se nalazi iza jednosmjernog ogledala koje propušta svjetlost samo u jednom smjeru. Kada se ogledalo uključi u struju, računalo se pali, te automatski pokreće internet pretraživač na kojemu se pokreće internet stranica te na taj način korisnik vidi podatke.

U radu je opisano korišteno sklopovlje, kućište koje štiti korištenu opremu, napisani programski kodovi, te tehnologije korištene za izradu završnog rada. Na kraju je dan prikaz rada pametnog ogledala, te je napravljeno testiranje rada. Za programsko rješenje korišten je Python koji služi za komunikaciju sa senzorom, te skriptni jezik PHP koji služi za generiranje i prikaz internet stranice. Zbog sigurnosnih razloga nije korišten internet za pokretanje poslužitelja, nego je napravljen lokalni Apache poslužitelj na računalu. Uz PHP korišteni su i HTML, stilski jezik CSS, te JavaScript za prikaz vremena.

1.1. Zadatak završnog rada

Zadatak završnog rada je napraviti pametno ogledalo, odnosno ogledalo koje osim obične funkcije ogledala prikazuje i neke podatke na sebi. To je napravljeno postavljanjem zaslona iza jednosmjernog ogledala. Na ogledalu su prikazani datum, vrijeme, te trenutna temperatura i vlažnost zraka.

2. SKLOPOVLJE

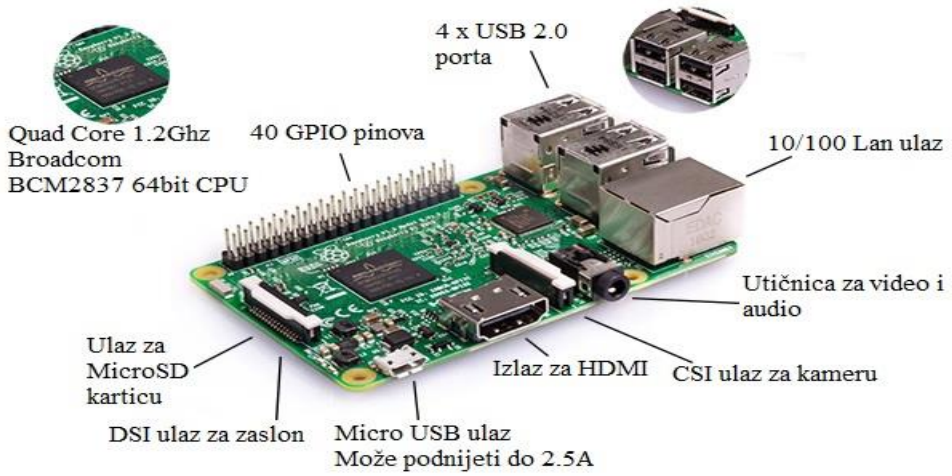
U ovom dijelu završnog rada bit će opisano korišteno sklopovlje, odnosno svi dijelovi korišteni za izradu pametnog ogledala. Svi dijelovi uklopljeni su u ručno izrađeno kućište i nalaze se iza jednosmjernog ogledala.

2.1. Korišteno sklopovlje

Postoji veliki broj načina za izradu pametnog ogledala. Završni rad je realiziran korištenjem minijaturnog računala Raspberry Pi 3B. Informacije na ogledalu su prikazane preko LCD zaslona, a podatci o vlazi i temperaturi dobiveni su sa senzora za vlagu i temperaturu DHT22. Senzor je spojen na Raspberry Pi korištenjem eksperimentalne pločice. Raspberry Pi ne dolazi sa ugrađenim RTC (*Real Time Clock*) satom. U završnom radu su prikazani datum i vrijeme te je bilo potrebno ugraditi RTC sat.

2.1.1 Raspberry Pi 3B

Raspberry Pi 3B je računalo veličine kreditne kartice razvijeno u Velikoj Britaniji od strane Raspberry Pi Foundation. Namjena Rasperry Pi Foundation-a je povećanje osnovnog znanja o računalima. Cijena najnovije verzije iznosi 260Kn, a cilj Raspberry Pi Foundation-a je razvijanje sve jačih računala uz održavanje iste cijene. U trenutku pisanja ovog teksta najnovija verzija je Raspberry Pi 4, dok je u završnom radu korišten Raspberry Pi 3 model B (u nastavku teksta Raspberry Pi), prikazan na slici 2.1.



Slika 2.1. Raspberry Pi 3B.

Raspberry Pi koristi Broadcom BCM2837 sustav na čipu. Sustav na čipu je elektronički integrirani krug koji ima razne elektroničke komponente koje su napravljene da zajedno rade kako bi postigle neki cilj. Sustav znači da je riječ o složenom elektroničkom sklopu, dok čip znači da se sve komponente tog sustava nalaze na jednom integriranom krugu. Česte komponente sustava na čipu, koje se nalaze i na Raspberry Pi-u, su središnja jedinica za obradu, grafički procesor, radna memorija itd. Središnja jedinica za obradu je glavni dio računala koji izvodi osnovne radnje nad podatcima. Raspberry Pi koristi ARM Cortex A53 koji sadrži četiri jezgre te ima frekvenciju radnog takta od 1.2 GHz. Grafički processor služi za prikaz slike na zaslonu. U Raspberry Pi je ugrađen Broadcom VideoCore IV. Radna memorija se koristi za čuvanje podataka tijekom rada programa, a jedna od glavnih karakteristika je da se pristup svim memorijskim lokacijama izvodi u jednakom vremenskom intervalu. Raspberry Pi sadrži 1 GB LPDDR2 radne memorije koja ima frekvenciju radnog takta od 900 MHz.

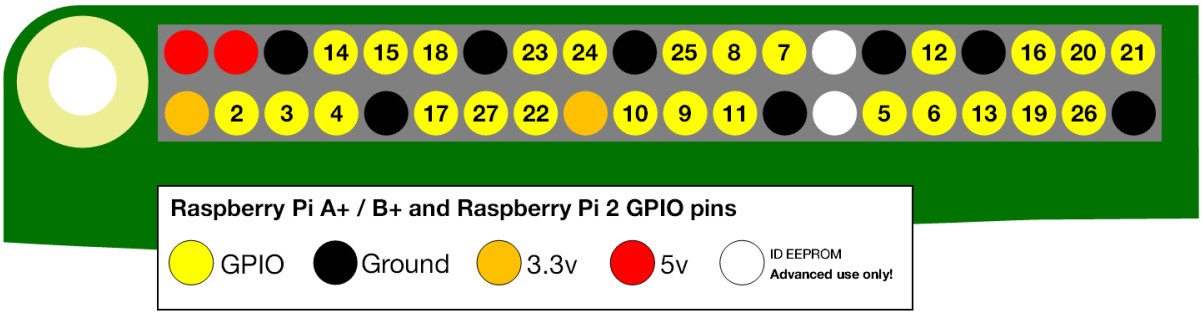
Za umrežavanje se koristi 10/100 Ethernet, te 2.4GHz 802.11n za bežično umrežavanje. Ethernet je zasnovan na rješenju koje se zove CSMA/CD (eng. *Carrier Sense Multiple Access / Collision Detection*) i najkorištenija je tehnologija za lokalne mreže. Ethernet 10/100 je mreža s brzinom od 10 i 100 Mbps. Jako stara računala su ograničena na 10 Mbps, ali većina novijih računala podržava 100 Mbps. Standard za bežično umrežavanje 802.11n koristi više antena kako bi povećao brzinu prijenosa podataka.

Raspberry Pi koristi Bluetooth 4.1 Classic, Bluetooth Low Energy. Blutooth se koristi za bežičnu razmjenu podataka između dva ili više uređaja. Većina računala, mobitela, i audio uređaja ima mogućnost slanja podataka preko bluetootha. Veza se uspostavlja preko radio valova u frekvencijskom području od 2,4 do 2,48 GHz. Bluetooth koristi radio vezu pa zbog toga uređaji ne moraju biti u optičkoj vidljivosti, a veza se može ostvariti na udaljenosti do 10m oko uređaja. Raspberry Pi za pohranu podataka koristi microSD memorijsku karticu. Memorijske kartice se često koriste u raznim elektroničkim uređajima, kao što su mobiteli, kamere, fotoaparati, laptopi itd. Glavne karakteristike takvih kartica su mala veličina, mogućnost pisanja i brisanja sadržaja te zadržavanje podataka bez korištenja električne energije.

Raspberry Pi ima 40 GPIO (*General Purpose input/output*) pinova. Pinovi se nalaze na gornjem dijelu ploče i trenutno ih imaju sve Raspberry Pi ploče osim Pi Zero (slika 2.2.).

Slika 2.2. Raspberry Pi GPIO pinovi.

Bilo koji od GPIO pinova se može postaviti kao ulazni ili izlazni pin i može biti korišten za veliki broj raznih potreba.



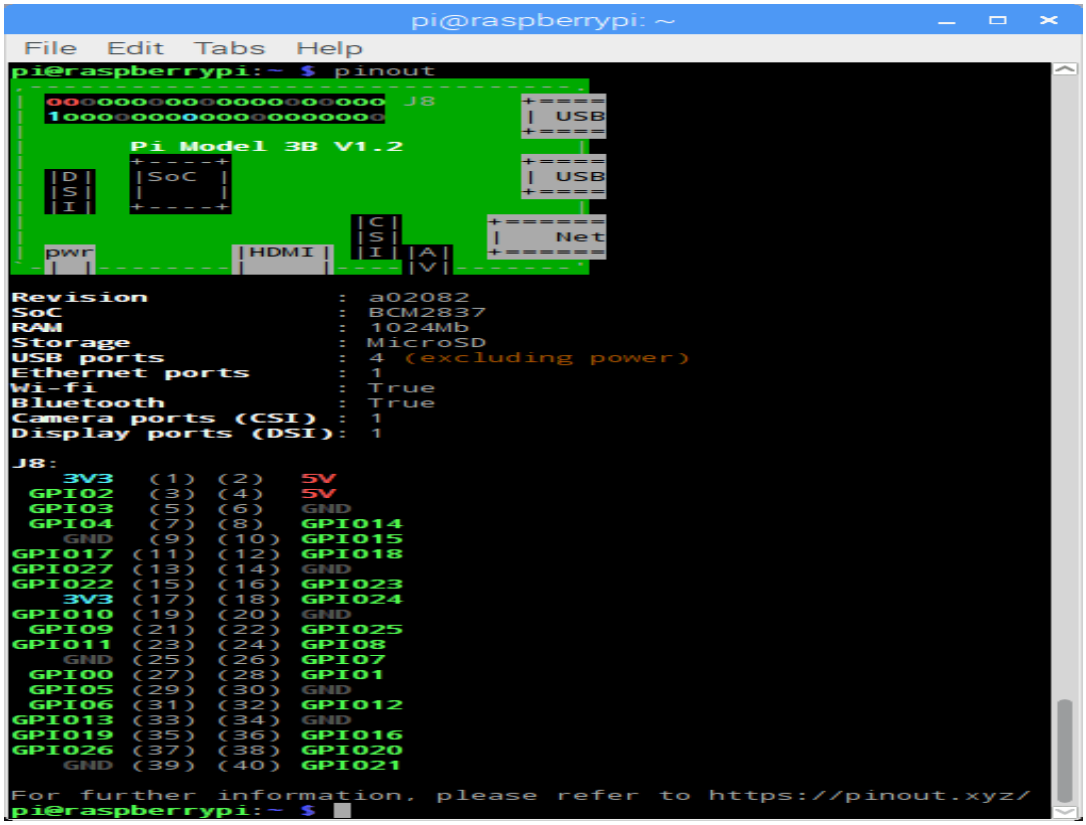
Slika 2.3. Raspberry Pi GPIO pinovi shema.

Ploča sadrži 2 pina od 5V, te dva pina od 3V3 volta, kao i nekoliko pinova za uzemljenje (GND) koji su nepodesivi (slika 2.3.). Pinovi definirani kao izlaz mogu biti postavljeni na visoki (3V3) ili niski (GND) napon. Pinovi definirani kao ulaz mogu biti čitani kao visoki (3V3) ili niski (GND) napon. Ovo je olakšano korištenjem unutarnjih *pull-up* ili *pull-down* otpornika. Pinovi GPIO2 i GPIO3 imaju fiksirane *pull-up* otpornike dok za druge ovo može biti programski podešeno.

Važno je biti svjestan koji je pin koji kako ne bi došlo do neželjenih kvarova. Jedan od rješenja je korištenje etiketa za pinove kao što se vidi na slici 2.4.

Slika 2.4. Etiketa za GPIO pinove.

Referenci za pinove se može pristupiti i u Raspberry Pi-u otvaranjem terminalnog prozora i pokretanjem komande *pinout* kao što se vidi na slici 2.5.

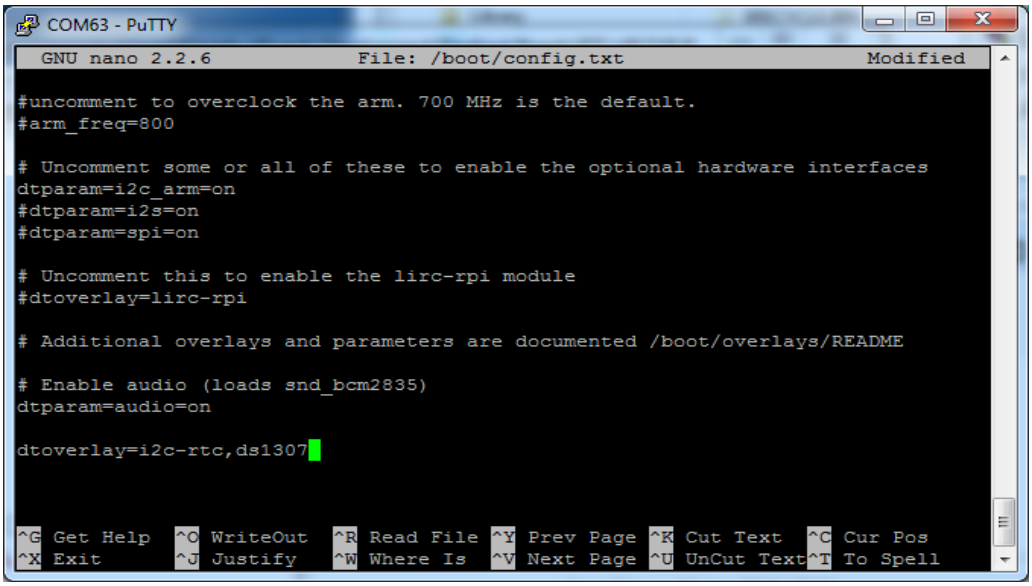


Slika 2.5. GPIO pinovi u terminalnom prozoru.

Moguće je kontrolirati GPIO pinove koristeći nekoliko programskih jezika i alata kao Python ili Scratch.

Uz sve navedeno Raspberry Pi ima veliki broj priključaka, kao što su HDMI priključak, 3,5mm analogni audio-video priključak, 4 USB 2.0 priključka te serijsko sučelje za kameru i zaslon.

Raspberry Pi nema ugrađen RTC (*Real Time Clock*) sat, pa je potrebno ugraditi ga i instalirati. Bez RTC sata, pri pokretanju Raspberry Pi pokazuje datum i vrijeme koje je bilo u trenutku kada je zadnji put ugašen. Podrška za RTC se može dodati uređivanjem *config.txt* datoteke. Jedan od načina za to napraviti jest pokretanjem naredbe *sudo nano /boot/config.txt* u terminalnom prozoru. U datoteku *config.txt* je potrebno dodati *dtoverlay=i2c-rtc,ds1307*. *Nano* je naziv za uređivač teksta u terminalnom prozoru. Na slici 2.6. je prikazan *nano* uređivač sa otvorenom *config.txt* datotekom.



Slika 2.6. *Nano* uređivač teksta.

Nakon što je datoteka uređena, potrebno je resetirati Raspberry Pi. To se može napraviti pokretanjem naredbe *sudo reboot*. Sljedeći korak je onesposobiti *fake hwclock* (*hardware clock*) jer se mješa sa radom RTC sata. Za onesposobljavanje *fake hwclock* potrebno je pokrenuti naredbe prikazane u programskom kodu 2.1.

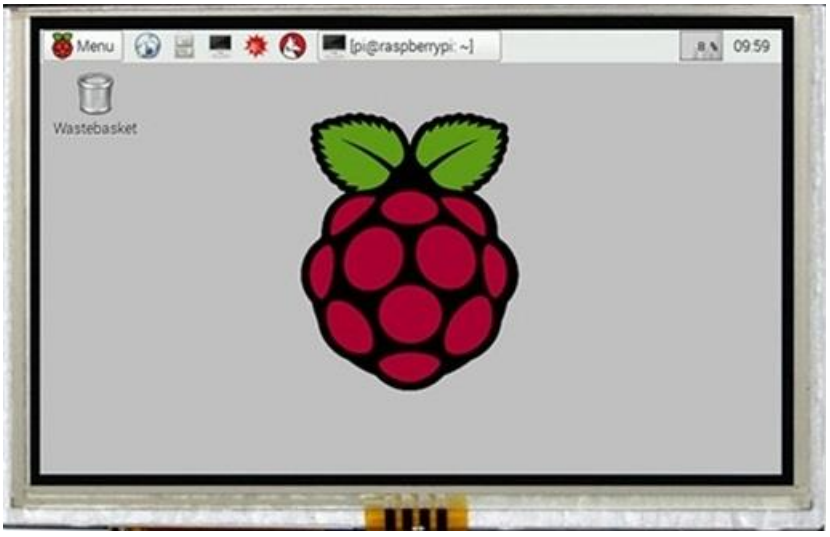
```
sudo apt-get –y remove fake-hwclock
sudo update-rc.d –f fake-hwclock remove
sudo systemctl disable fake-hwclock
```

Programski kod 2.1. Naredbe za onesposobljavanje *fake hwclock*.

Kada je RTC sat spreman za upotrebu prvo je potrebno postaviti točan datum i vrijeme. Povezivanjem Raspberry Pi-a na internet datum i vrijeme će se sami podesiti. Pokretanjem naredbe *sudo hwclock –w* datum i vrijeme sa Raspberry Pi-a će se spremiti u RTC sat. Naredbom *sudo hwclock –r* se može pročitati vrijeme sa RTC sata.

2.1.2 LCD zaslon

Korišteni LCD zaslon ima rezoluciju veličine 800x480 piksela, zaslon osjetljiv na dodir i može se direktno priključiti na bilo koju verziju Raspberry Pi-a (osim prve verzije Raspberry Pi-a modela B koja zahtjeva HDMI kabel). Ima mogućnost upravljanja stražnjim svjetlom radi manje potrošnje energije. Osim uz Raspberry Pi može se koristiti i sa drugim minijaturnim računalima kao Banana Pi, Pcduino, Cubieboard, Mashboard. LCD zaslon je prikazan na slici 2.7.



Slika 2.7. LCD zaslon.

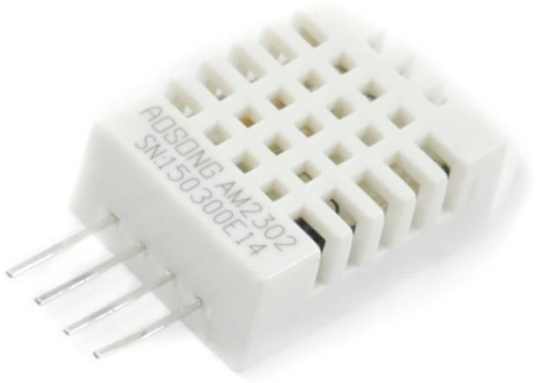
LCD (engl. *Liquid Crystal Display*) je ekran temeljen na tehnologiji tekućih kristala. Najčešće se koriste u LCD monitorima u obliku aktivnih TFT LCD ekrana. TFT LCD ekran se sastoji od definiranog broja piksela i oni su poredani ispred svjetlosnog izvora (najčešće LED-ice, a nekada su to bile CCFL cijevi). Količina potrošene električne energije je jako mala i zauzima malo

prostora. Tekući kristal je otkriven 1888. godine od strane austrijskog botaničara F. Reintzera, kada je proučavao *cholesteryl benzoate*. Taljenjem te tvari dobio je mutnu tekućinu koja se bistrila dok se hladi te se na kraju kristalizira. Tek je 1968. godine pronađena tvar koja je imala ovakva svojstva na sobnoj temperaturi.

Kod LCD-ova u boji, svaka točka (piksel) sadrži tri manje točke (podpiksela). Te manje točke su dodatnim filterima obojene u crveno, zeleno i plavo. Svakom od pod-točaka se upravlja zasebno i njihova krajnja kombinacija je ton jedne boje. Tekući kristal u sredini zakreće svjetlost o čemu ovisi svjetlina pod-točke.

2.1.3 DHT22

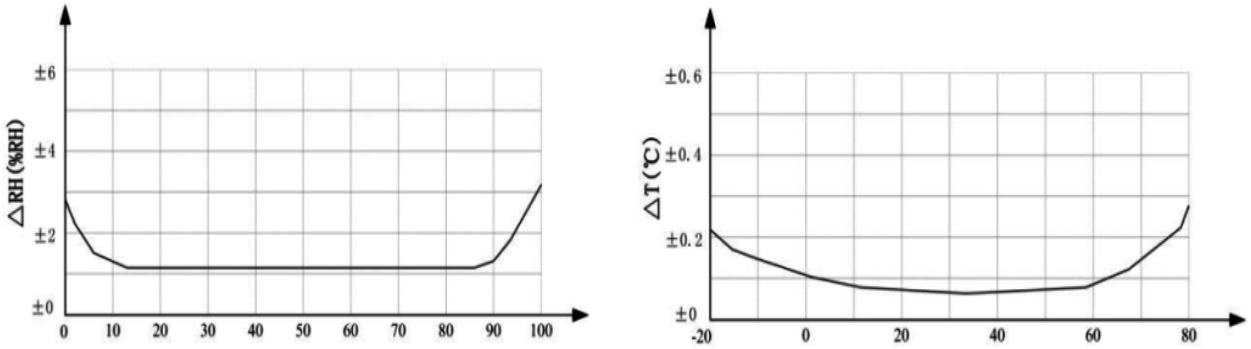
DHT22 ili AM2303 je kapacitativni senzor koji omogućava mjerenje relativne vlažnosti zraka i temperature zraka (slika 2.8.). Senzor je vrlo precizan, ali je nešto skuplji te sporiji (2 sekunde). od drugih senzora koji obavljaju istu zadaću. Osim uređaja za mjerenje, modul se sastoji i od 8-bitnog mikrokontrolera i OTP memorije preko kojih se izvodi kalibriranje svakog očitanja. OTP (engl. *One Time Programmable*) kratica je za memoriju koja se može programirati samo jednom. Naziva se još i PROM (engl. *Programmable Read Only Memory*) i služi da se konačni razvijeni program upiše u memoriju i trajno ostaje zapisan. DHT22 ima male dimenzije, malu potrošnju i mogućnost prijenosa podataka na velike udaljenosti (>20m) i zbog toga je pogodan za razne primjene.



Slika 2.8. DHT22.

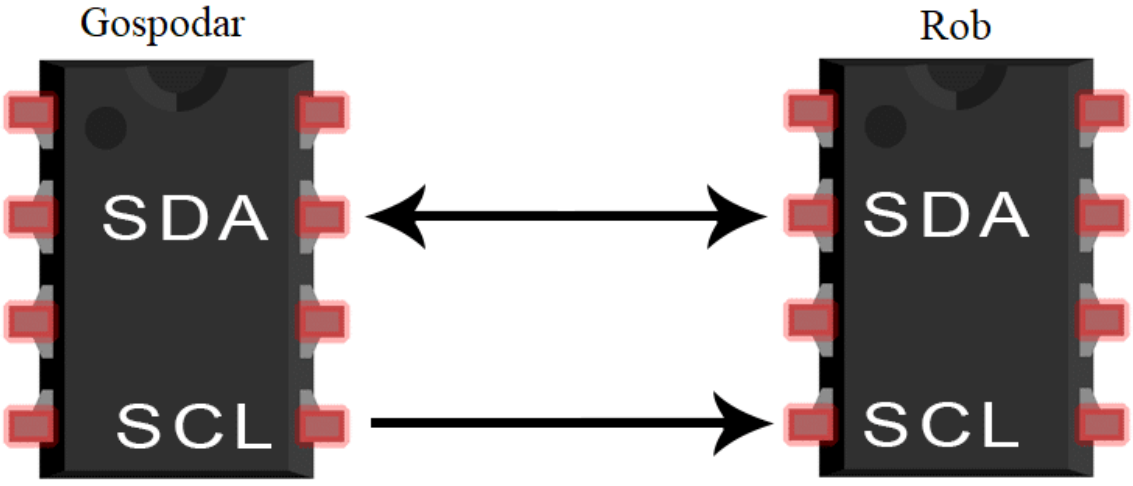
DHT22 radi na naponu od 3.3V do 5.5V, a preporučeno je korištenje na 5V. Raspon mjerenja za relativnu vlažnost iznosi od 0 do 100%, dok raspon mjerenja za temperaturu iznosi od -40°C do +80°C. Točnost mjerenja za relativnu vlažnost je ±2% dok je točnost mjerenja za temperaturu

±0,5°C. Vrijeme potrebno za izvršenje jednog mjerenja iznosi 2 sekunde. Dijagrami na slici 2.8. prikazuju točnost očitanja u ovisnosti o mjerenjima.



Slika 2.9. Točnost očitanja DHT22.

Raspberry Pi koristi I2C (*Inter-Integrated Circuit*) protokol. I2C se koristi za komunikaciju elektroničkih uređaja. Elektronički uređaji međusobno komuniciraju tako što šalju bitove podataka preko žica. I2C koristi gospodar/rob (*master/slave*) model komunikacije. U takvoj komunikaciji gospodar ima kontrolu nad uređajem koji mu je rob. Preko I2C protokola se može uspostaviti komunikacija u kojoj jedan gospodar ima više robova ili komunikacija u kojoj više gospodara ima jednog ili više robova. I2C koristi samo dvije žice za prijenos podataka (prikazano na slici 2.10.). SDA (*serial data*) je linija preko koje gospodar i rob međusobno šalju podatke. SCL (*serial clock*) je linija koja nosi taktni impuls.

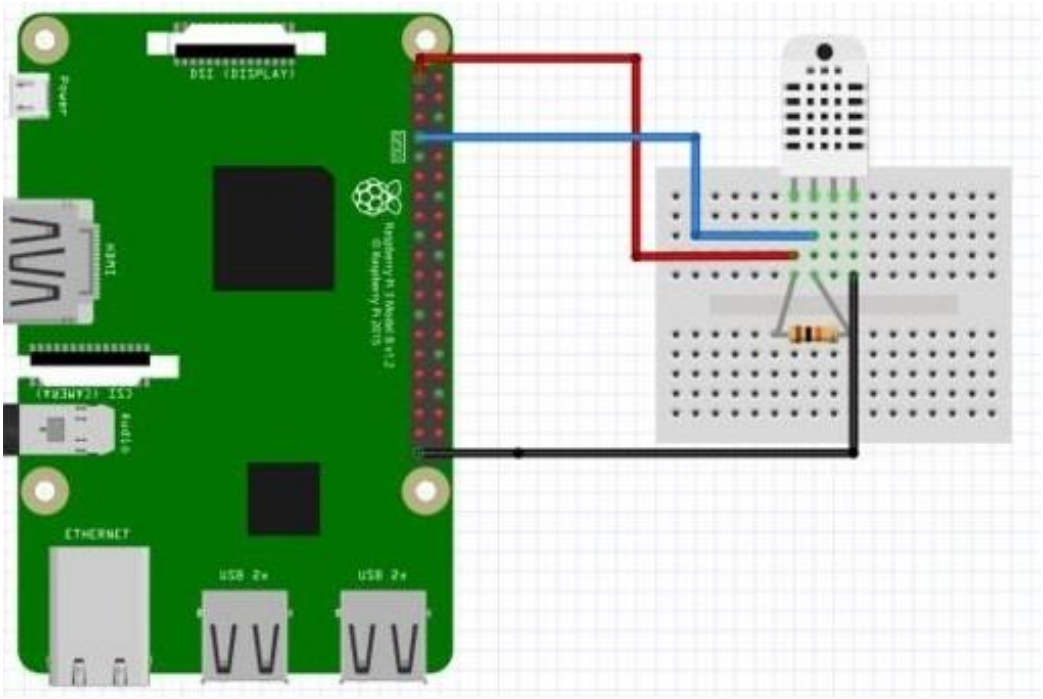


Slika 2.10. Prijenos podataka s I2C protokolom.

Raspberry Pi i DHT22 komuniciraju preko I2C protokola. Zbog svoje *master/slave* structure, SDA pinu je potreban vanjski *pull-up* otpornik. Vrijednost otpornika ovisi o duljini kabela

između mikrokontrolera i senzora. Preporuka je koristiti otpornike od 5100Ω za 5V napon, dok za duljine kabela veće od 20m nije potreban otpornik. Ukoliko se koristi 3,3V napon duljina kabela može biti do 1m maksimalno.

Na slici 2.11. se vidi shema spajanja DHT22 senzora na Raspberry Pi korištenjem eksperimentalne pločice, žice i jednog otpornika od 10.000Ω. Prvi s lijeva izlaz iz DHT-a se spaja na napon od 3,3V od Raspberry Pi-a, dok se prvi s desna izlaz spaja na bilo koji pin za uzemljenje. Drugi s lijeva pin DHT-a se spaja na pin za ulaz podataka na Raspberry Pi-u. Otpornik se spaja između pina za napon i pina za slanje podataka.



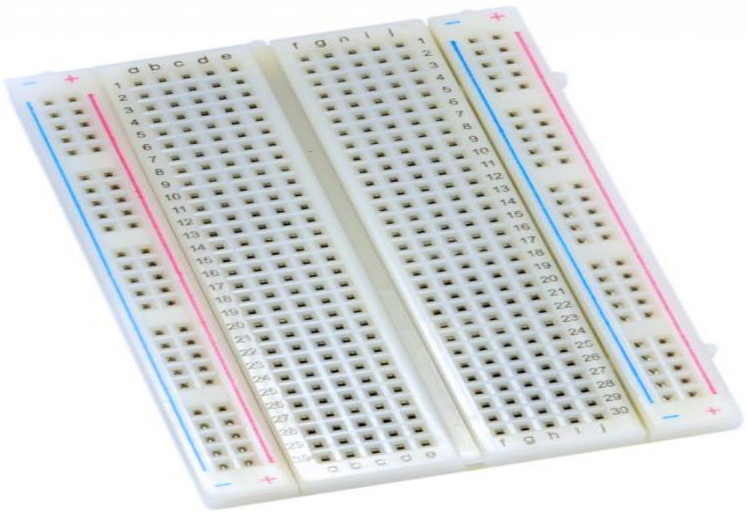
Slika 2.11. Shema spajanja DHT22 senzora.

2.1.4 Eksperimentalna pločica

Eksperimentalna pločica (engl. *Solderless breadboard*) je pomagalo korišteno pri spajanju privremenih električkih strujnih krugova. Može se pronaći u različitim dimenzijama i oblicima, ali su u osnovi sve jednake.

Kao što se vidi na slici 2.12., sastoji se od kućišta na čijoj se gornjoj strani nalazi veliki broj rupica koje služe za umetanje nožica različitih komponenti. Međusobni razmaci između rupica

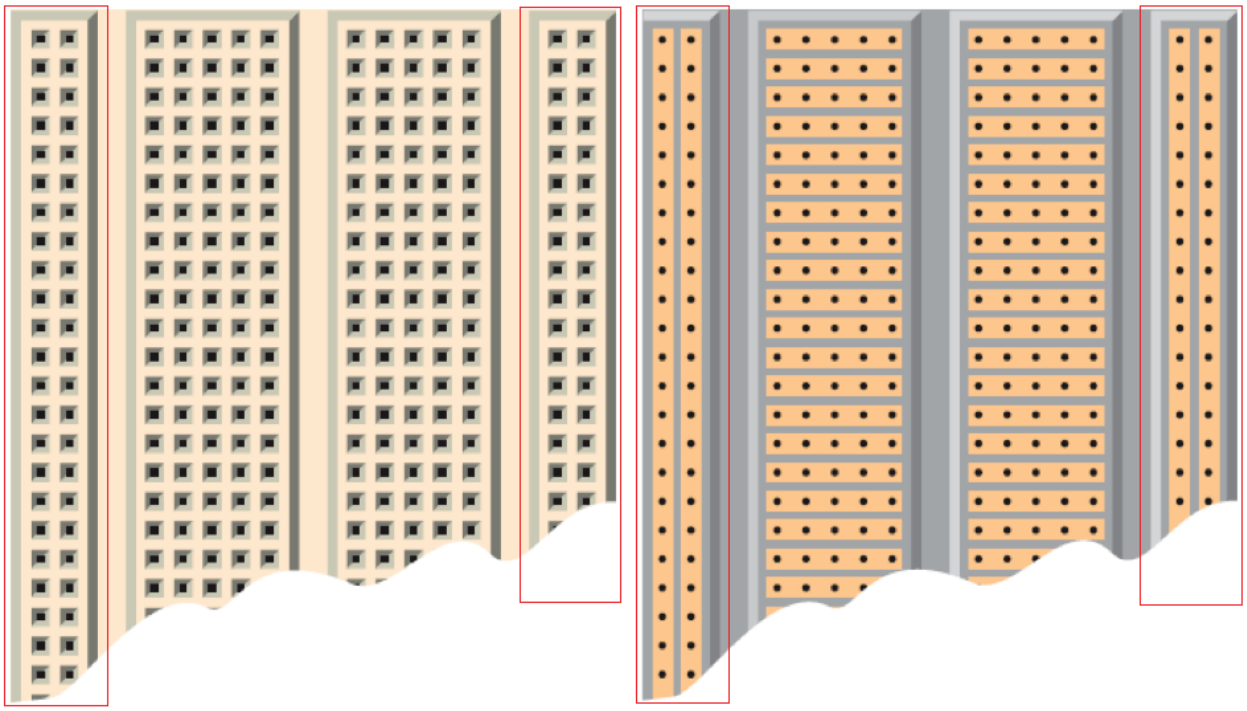
su standardizirani i iznose 2,5mm. Osim toga razmak je usklađen s razmakom među nožicama većine integriranih krugova i komponenti.



Slika 2.12. Eksperimentalna pločica.

Rupice u unutrašnjosti pločice su međusobno povezane prema određenom pravilu. Kao što je prikazano na slikama 2.12. i 2.13. razlikujemo 3 osnovna dijela eksperimentalne pločice. Sa svake strane pločice postoji područje (na slici 2.13. zaokruženo crvenom bojom) koje se sastoji od dva niza rupica međusobno spojenih na način da tvore dvije okomite "trake" koje se koriste za napajanje. Najčešće se uz svaku traku nalazi linija, crvena ili plava (crna) zajedno sa simbolima (+) ili (-). Na većini eksperimentalnih pločica rupice unutar jedne linije nisu spojene po cijeloj dužini, nego su prekinute na sredini (ponekad i na više mjesta) što treba provjeriti prije korištenja. Ovo je obično slučaj s većim eksperimentalnim pločicama, dok kod manjih nema prekida.

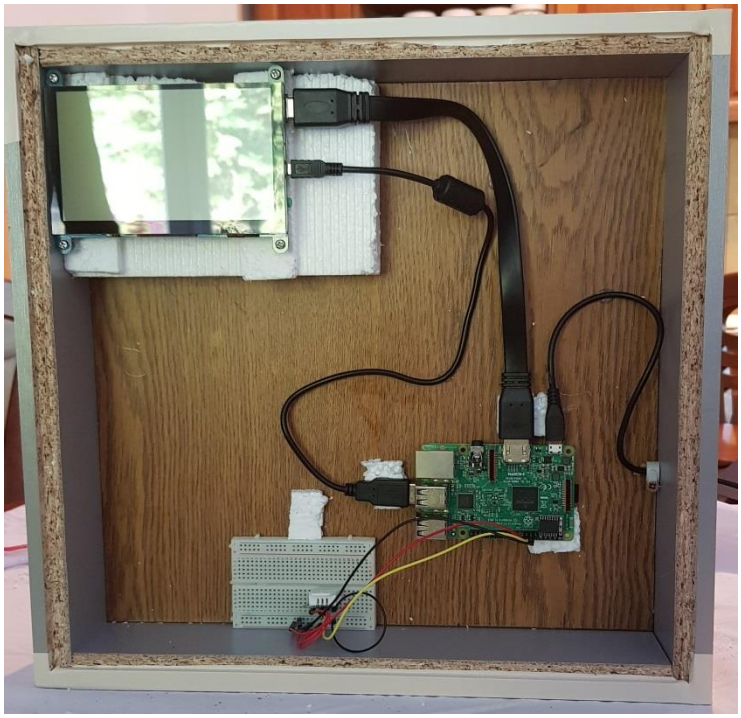
U samoj sredini eksperimentalne pločice nalazi se vertikalni rascjep koji dijeli dva područja. Njegova veličina je standardizirana i omogućava umetanje integriranih krugova. Rupice unutar svakog sivog područja spojene su vodoravno čineći u istom redu dva vodoravna niza od po pet rupica. U većini slučajeva redovi na pločici su označeni brojem (npr. 1-30), a stupci slovima (a-j).



Slika 2.13. Unutrašnjost eksperimentalne pločice.

2.2. Kućište pametnog ogledala

Za završni rad se koristi jednosmjerno ogledalo, odnosno ogledalo koje propušta svjetlost samo u jednom smjeru. Jedna strana ogledala odbija svjetlost, odnosno radi kao i obično ogledalo, dok druga strana propušta svjetlost, odnosno vidi se kroz nju kao kroz obično staklo. Korišteno ogledalo je veličine 30 x 30cm. LCD zaslon je mnogo manji, te je zbog toga korišten crni papir koji je zalijepljen na stranu ogledala koja propušta svjetlost. Na papiru je škarama isječen kvadrat veličine zaslona. Na zaslonu je prikazana crna pozadina s bijelim slovima, te se zbog toga na ogledalu vide samo slova. Oko ogledala je napravljeno drveno kućište koje štiti komponente koje se nalaze iza ogledala. Kako bi ogledalo izgledalo ljepše, na drveno kućište su zalijepljene tapete za namještaj (slika 2.15.). Iza ogledala, odnosno unutar drvenog kućišta zalijepljene su elektroničke komponente (slika 2.14.).



Slika 2.14. Elektroničke komponente unutar drvenog kućišta.

Slika 2.15. Drveno kućište pametnog ogledala.

3. PROGRAMSKO RJEŠENJE

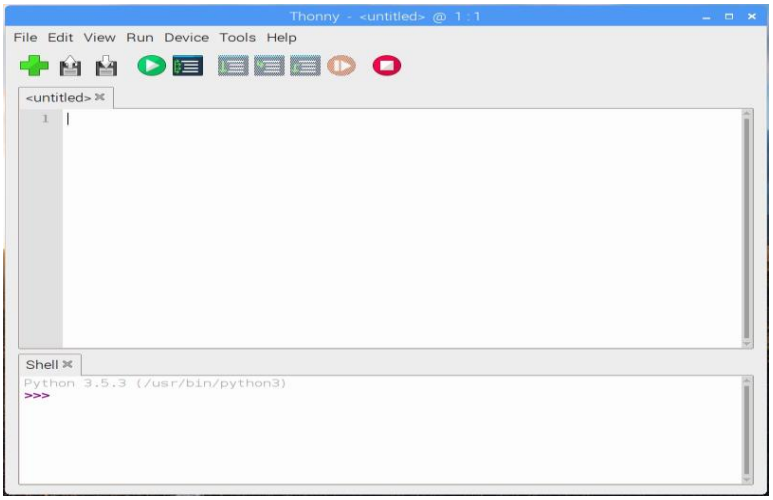
U ovom poglavlju će biti opisane tehnologije korištene za izradu završnog rada, te svi napisani programski kodovi.

3.1. Korištene tehnologije

Tehnologije korištene u završnom radu su programski jezik Python koji se koristi za komunikaciju sa senzorom, skriptni jezik PHP, te HTML, JavaScript i stilski jezik CSS. Za rad PHP-a potreban je poslužitelj, te se zbog toga koristi Apache.

3.1.1 Python

Python je programski jezik opće namjene, interpretiran i svrstava se u programske jezike visoke razine. Napravio ga je Guido van Rossum 1990. godine, iako je prva verzija objavljena tek u veljači 1991. Godine. Ime je dobio po televizijskoj seriji *Monthy Python's Flying Circus*. Ima automatsku memorijsku alokaciju pa je zbog toga sličan jezicima kao što su Perl, Ruby, Smalltalk itd. Python dopušta programerima nekoliko stilova pisanja programa. Dopušteni stilovi za programirčanje su objektno orijentirano, strukturno i aspektno orijentirano programiranje i ta fleksibilnost čini programski jezik Python sve popularnijim. Python se najviše koristi na Linux operacijskom sustavu, ali postoje i inačice za druge operacijske sustave. Za pisanje Python skripti na Raspberry Pi računalu, najčešće se koristi Thonny integrirano razvojno okruženje (slika 3.1.).



Slika 3.1. Thonny razvojno okruženje.

Postoje česte kritike na račun Pythona zbog njegove sporosti. Python je interpreterski jezik i zbog toga programi napisani u njemu izvode sporije za razliku od programa napisanih u programskim jezicima kao što su C, C++ i slični. Iako je sporiji, Python se poprilično često koristi (ponajviše kao *backend* programski jezik). Python se često uspoređuje s Javom. Java je također interpreterski jezik. Java i Python nemaju podršku za višejezgreno izvođenje programa jer oba koriste samo jednu procesorsku jezgru. Java kao jezik se mnogo više koristi u izradi mobilnih aplikacija i interaktivnog web sadržaja. Java i Python su poprilično jednaki što se tiče brzine izvođenja programa.

Python koristi uvlačenje za razlikovanje programskih blokova, tj. ne koristi vitičaste zagrade ili ključne riječi kao većina programskih jezika. Povećanje uvlačenja znači da dolazi novi, ugniježđeni blok, dok smanjenje označava kraj trenutnog bloka.

U Pythonove ključne riječi spadaju:

- *If* izraz, izvodi programski blok ako je istinit neki uvjet, zajedno s *else* i *elif* (else-if varijanta)
- *For* izraz, iterira kroz iterabilan objekt i svaki element upisuje u lokalnu varijablu koja se koristi u pridruženom bloku
- *While* izraz, izvodi programski blok sve dok je neki uvjet istinit
- *Try* izraz, omogućava da iznimke, koje se nalaze u bloku koda kojeg *try* obuhvaća, budu uhvaćene i obrađene u *except* bloku
- *Class* izraz, deklarira klasu (u objektno orijentiranom programiranju)
- *Def* izraz, definira neku funkciju ili metodu
- *Assert* izraz, koristi se pri debugiranju kako bi se provjerilo vrijede li određeni uvjeti
- *Import* izraz, koristi se pri uključivanju dodatnih modula

U programskim kodovima 3.1. i 3.2. prikazani su neki primjeri korištenja Pythonovih ključnih riječi.

```
#komentar u Pythonu
#na ekran se ispiše "Pametno ogledalo"
print ("Pametno ogledalo")

#primjer liste (polja):
temperature = [5, 18, 25, 40, 55]

#ispis prvog elementa u listi brojevi, u ovom slučaju "5"
print (temperature[0])

#primjer ispitivanja uvjeta:
a = 5
b = 3
if a > b:
    print ("a je veće od b")
else:
    print ("b je veće od a")

#primjer iteriranja kroz listu i ispis njenih elemenata:
for temp in temperature:
    print(temp)

#primjer funkcije koja ispisuje "Pametno ogledalo"
def smartMirror():
    print ("Pametno ogledalo")
```

Programski kod 3.1. Primjeri koda u Pythonu.

```
class PametnoOgledalo():
    #konstruktor bilo koje klase se definira kao def
__init__(self):
    def __init__(self):
        pass
    #definicija metode ispis()
    def ispis(self):
        print ("Pametno ogledalo!")
```

Programski kod 3.2. Primjer klase u Pythonu.

3.1.2 HTML

HTML je kratica za *HyperText Markup Language*, što znači prezentacijski jezik za izradu internet stranica. Pomoću HTML jezika stvara se hipertekst dokument. HTML je jednostavan za korištenje i veoma lako se uči, što je jedan od razloga njegove popularnosti i opće prihvaćenosti. Vrlo je raširen zbog toga jer je jednostavan i jer je od početka bio zamišljen kao besplatan i tako

dostupan svima. Internet preglednik omogućava prikaz hipertekst dokumenta. Osnovna zadaća HTML jezika jest pokazati internet pregledniku kako prikazati hipertekst dokument. Nastoji se da taj dokument izgleda jednako bez obzira o kojem je internet pregledniku, računalu i operacijskom sustavu riječ. HTML nije programski jezik i ljudi koji ga koriste nisu programeri. On ne može izvesti nikakvu zadaću, čak ni najjednostavniju operaciju zbrajanja ili oduzimanja dvaju cijelih brojeva. On služi samo za opis hipertekstualnih dokumenata. HTML datoteke su zapravo samo obične tekstualne datoteke, ekstenzija im je .html ili .htm. Osnovni element svake stranice su znakovi (*tags*) i oni opisuju kako će se nešto prikazati u internet pregledniku. Poveznice unutar HTML dokumenata povezuju dokumente u uređenu hijerarhijsku strukturu i time određuju način na koji posjetitelj doživljava sadržaj stranica.

Hipertekst je tekstualna struktura koja se sastoji od jedinica informacije koje su međusobno povezane i prikazane na nekom elektroničkom uređaju. Za razliku od jednostavnog tradicionalnog teksta, hipertekst nema jedinstven redoslijed čitanja, nego ga čitatelj određuje tijekom čitanja. Zato se smatra da je tradicionalni tekst sekvencijalan, a hipertekst nesekvencijalan. Postoje dvije vrste hiperteksta. Statički hipertekst je predodređen i unaprijed zadan te se ne mijenja tijekom rada. Dinamički hipertekst se kontinuirano mijenja u interakciji s korisnikom.

Prvi javno dostupan opis HTML-a je dokument zvan HTML tags (oznake) i prvi put je spomenut od strane Tim Berners-Leeja krajem 1991. Taj dokument se sastoji od 20 elemenata početnog, relativno jednostavnog, dizajna HTML-a. Trinaest od tih 20 elemenata još uvijek postoji u HTML4. Prva verzija HTML-a objavljena je 1993. godine. U to je vrijeme HTML bio jako ograničen i nije još bilo moguće niti dodati slike u dokument. Razvoj je nastavljen prvom „imenovanom“ verzijom – 2.0, ali ni ona nije postala standardna. HTML4 predstavljen je u prosincu 1997. i tada je morao prihvatiti oznake nametnute od strane raznih proizvođača različitih internet preglednika, ali je u isto vrijeme i pokrenuto čišćenje standarda, čime su neke od oznaka proglašene suvišnim. HTML5 je prva nova verzija nastala nakon verzije HTML 4.01, a objavljena je 1999. godine. HTML5 donosi brojne nove mogućnosti koje HTML 4.01 nije imao, kao što je reprodukcija videa na stranicama bez korištenja Adobe flasha, mogućnost upravljanja pomoću tipkovnice, *drag and drop, canvas* i mnoge druge. Na internet adresi *html5test.com* se može testirati u kojoj mjeri browser podržava html5.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pametno ogledalo</title>
  </head>
  <body>
    <p>Internet stranica za pametno ogledalo</p>
  </body>
</html>
```

Programski kod 3.3. Primjer HTML dokumenta.

Svaki HTML dokument se sastoji od osnovnih blokova – HTML elemenata. Neki od HTML elemenata su prikazani u programskom kodu 3.3. Svaki HTML element se sastoji od para HTML oznaki (engl. *tags*). Svaki element također može imati atribute kojim mu se određuju svojstva. Na samom početku dokumenta preporučljivo je staviti *<!DOCTYPE>* element, kojim se označava DTD (engl. *Document Type Declaration*) i kojim se definira točna verzija standarda koji se koristi za izradu HTML dokumenta. Elementom *<html>* označava se početak HTML dokumenta. Unutar njega se nalaze elementi *<head>* i *<body>*. *<head>* element je zaglavlje dokumenta i u njemu se najčešće specificiraju jezične značajke HTML dokumenta kao i sam naslov (engl. *title*) stranice. Pomoću određenih HTML elemenata unutar zaglavlja se dodaju i stilska obilježja stranice i ona mogu biti direktno ugrađena ili dodana kao referenca na vanjsku CSS datoteku. U zaglavlju se često definiraju i skripte napisane u JavaScript jeziku. U elementu *<body>* kreira se sadržaj HTML dokumenta.

Svaka HTML oznaka počinje znakom < (manje od), a završava znakom > (veće od). Zatvarajuća HTML oznaka kreira se na isti način kao i otvarajuća, samo se prije početnog znaka < dodaje kosa crta / (engl. *slash*).

Postoje i samozatvarajuće oznake, kao na primjer referenca na vanjski CSS dokument *<link rel="stylesheet" type="text/css" href="stil.css" />*. Prema preporuci W3C-a, kod samozatvarajućih elemenata poželjno je ostaviti razmak između definiranih atributa i njihovih vrijednosti i zatvarajućih znakova (/>).

HTML dokument se može napraviti u bilo kojem uređivaču teksta, ali za neke naprednije funkcije, olakšanje repetitivnih aktivnosti te vizualni pregled web stranice preporučeno je koristiti neki od naprednijih alata, kao Adobe Dreamweaver.

Komentari se mogu unositi bilo gdje unutar HTML dokumenta i tekst unutar njih se neće prikazati na stranici, tj. može se vidjeti samo ako se dokument otvori u nekom uređivaču teksta. Na taj način korisnik može sebi ostavljati poruke i skrenuti pažnju na određeni dio skripte. Komentar počinje sa oznakom <*!*-- te se zatvara sa oznakom -->. Primjer komentara: <*!*-- *ovo je komentar* -->. U boljim HTML editorima komentar će biti ispisan sivim slovima kako bi se razlikovao od ostatka teksta.

3.1.3 CSS i JavaScript

CSS (engl. *Cascading Style Sheets*) je stilski jezik koji se rabi za opis prezentacije dokumenta napisanog pomoću *markup* (HTML) jezika. Kako se internet razvijao, prvo su u HTML ubačeni elementi za definiciju prezentacije, ali je brzo uočena potreba za stilskim jezikom koji će osloboditi HTML od potrebe prikazivanja sadržaja i njegovog oblikovanja (čemu služi CSS). Stil definira kako prikazati HTML elemente. CSS-om se uređuje sam izgled i raspored stranice.

Style sheet u CSS-u se sastoji od nekoliko pravila i svako se pravilo sastoji od selektora i deklaracijskog bloka.

Selektor označava dio HTML-a na koji se primjenjuje stil. Selektor mogu biti svi elementi istog tipa, elementi određenog *id* ili *class* atributa. *Id* je jedinstven element, dok *class* može obuhvaćati više od jednog elementa.

Deklaracijski blok su vitičaste zagrade zajedno sa deklaracijama koje se nalaze unutar njih. Svaka se deklaracija sastoji od svojstva, dvotočke (:) i vrijednosti. Između svake dvije uzastopne deklaracije mora se nalaziti točka zarez (;). Vrijednosti mogu biti ključne riječi kao *center* (sredina) i *inherit* (naslijedi), brojčane vrijednosti kao 100 (debljina fonta), 200px (200 piksela), 50vw (50% širine viewporta) ili 80% (80% širine prozora). Vrijednosti boja mogu biti ključne riječi (npr. *red* za crveno), heksadecimalne vrijednosti (npr. #FF0000 ili #F00) ili RGB vrijednost od 0 do 255(npr. *rgb(255, 0 ,0)*). Primjer CSS-a prikazan je u programskom kodu 3.4.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: red;
  text-align: center;
}
</style>
</head>
<body>
<p>Pametno ogledalo!</p>
</body>
</html>
```

Programski kod 3.4. Primjer CSS-a u zaglavlju HTML dokumenta.

JavaScript je skriptni jezik kojim se u statičke HTML stranice mogu uvesti interaktivni elementi. JavaScript je jednostavan, interpretiran programski jezik namjenjen ponajprije razvoju interaktivnih HTML stranica. Jezgra JavaScripta uključena je u većinu današnjih preglednika (*Internet Explorer*, *Google Chrome*, *Mozilla Firefox*). JavaScript omogućuje izvođenje određenih radnji u inače statičnim HTML-dokumentima, npr. interakciju s korisnikom. JavaScript nije pojednostavljena verzija programskog jezika Java. Jedino što ih povezuje je slična sintaksa i to što se koriste za izvođenje određenih radnji unutar preglednika. JavaScript se trebao zvati *LiveScript*, ali da bi se potakla upotreba novog skriptnog jezika nazvan je slično jeziku Java.

JavaScript se razvija od 1995. kad je *Netscape* objavio nekoliko prvih inačica jezika. Nedugo nakon toga *Microsoft* je objavio jezik sličan JavaScript-u koji je nazvao *JScript*.

Program koji obrađuje i izvodi skripte naziva se *interpreter*. *Interpreter* za JavaScript ugrađen je u većinu današnjih preglednika. Skriptni se jezici koriste jer je razvoj programa znatno jednostavniji. Za razliku od programa pisanih u programskim jezicima, kod skriptnog jezika ne treba prevoditi skripte u strojni jezik. Kod programskih jezika potrebno je napisati ili popraviti program, zatim prevesti program u strojni jezik, a nakon toga pokrenuti prevedeni program i ako su potrebne popravke ponoviti postupak. Kod skriptnih jezika potrebno je napisati ili popraviti skriptu, zatim pokrenuti *interpreter* te ako su potrebne popravke ponoviti postupak. Skriptni jezici imaju jedan korak manje u odnosu na programske jezike.

JavaScript se može pisati prema standardu *Unicode*, no preporuča se pisanje prema standardu *ASCII*, osim u komentarima i nizovima znakova. Prilikom pisanja sintakse važno je zapamtiti da JavaScript razlikuje velika i mala slova. Primjer JavaScript-a je prikazan u programskom kodu 3.5.

```
<!DOCTYPE html>
<html>
<head>
<script>
    function mojaFunkcija() {
      document.getElementById("text").innerHTML = "Pametno
ogledalo!";
    }
</script>
</head>
<body>
<p id="text">Smart mirror!</p>
<button type="button" onclick="mojaFunkcija()">Pritisni
dugme!</button>
</body>
</html>
```

Programski kod 3.5. Primjer JavaScript skripte unutar zaglavlja HTML dokumenta.

U završnom radu JavaScript je iskorištena za prikaz sati i minuta. Način na koji je to učinjeno prikazan je u programskom kodu 3.6.

```
<script>
     function startTime() {
          var today = new Date();
          var h = today.getHours();
          var m = today.getMinutes();
          m = checkTime(m);
          document.getElementById('dateTime').innerHTML = h +
":" + m;  }
     function checkTime(i) {
          if(i < 10) {i = "0" + i}; // dodaje nulu ispred
brojeva manjih od < 10
     return i;
     }
</script>
```

Programski kod 3.6. Dohvaćanje sati i minuta s JavaScript-om.

3.1.4 PHP i Apache

PHP je *open-source server-side* skriptni programski jezik za dinamičko generiranje HTML koda. Drugim riječima, PHP je skriptni jezik kojim se može kreirati HTML stranica, popunjena dinamičkim sadržajem, na serveru prije nego je poslana klijentu. Primjer dinamičkog generiranja HTML koda prikazan je u programskom kodu 3.7. Ovim načinom generiranja sadržaja klijent ne može vidjeti kod (skriptu) koji je generirao sadržaj koji gleda, već ima samo pristup čistom HTML kodu. *Open-source* znači da svatko može pristupiti izvornim PHP kodovima pisanim u C-u i ukoliko ih razumije može ih mijenjati te dodavati nove PHP funkcije. Izvorni kodovi i instalacijske datoteke se mogu skinuti sa službene internet stranice PHP-a.

PHP je trenutno jedna od najnaprednijih i najkorištenijih server-side skriptnih tehnologija. PHP je svojom sintaksom sličan mnogim drugim jezicima, čak ima i istoznačne funkcije kao i neki drugi jezici kao što su C ili Perl. To znači da se jedna radnja može izvesti korištenjem više različitih funkcija.

PHP sadrži brojne funkcije za manipuliranje različitim tipovima sadržaja, od manipuliranja grafikom do učitavanja .NET modula i rada sa XML-om. PHP ima podršku za upravljanje raznim bazama podataka. Podržava sve popularnije baze podataka kao MySQL, dBase, Oracle... Njegova neovisnost o operacijskom sustavu i pristupačna cijena (besplatan je) čini ga prvim izborom mnogih korisnika.

```
<?php
echo "<tr>
      <td class='days'>PON</td>
      <td class='days'>UTO</td>
      <td class='days'>SRI</td>
      <td class='days'>CET</td>
      <td class='days'>PET</td>
      <td class='days'>SUB</td>
      <td class='days'>NED</td>
       </tr>";
?>
```

Programski kod 3.7. Primjer ubacivanja retka tablice koristeći PHP.

Apache HTTP poslužitelj, također zvan Apache je besplatna *open-source* programska podrška za servere. Apache je internet poslužitelj koji može raditi na raznim operacijskim sustavima kao što su Unix sustavi, Microsoft Windows sustavi i drugi. Najkorišteniji je poslužitelj na internetu s

udjelom većim od 60%. Apache razvija i održava otvorena zajednica programera pod vodstvom *Apache Software Foundation*. Apache se koristi za otprilike 65% svih hrvatskih domena.

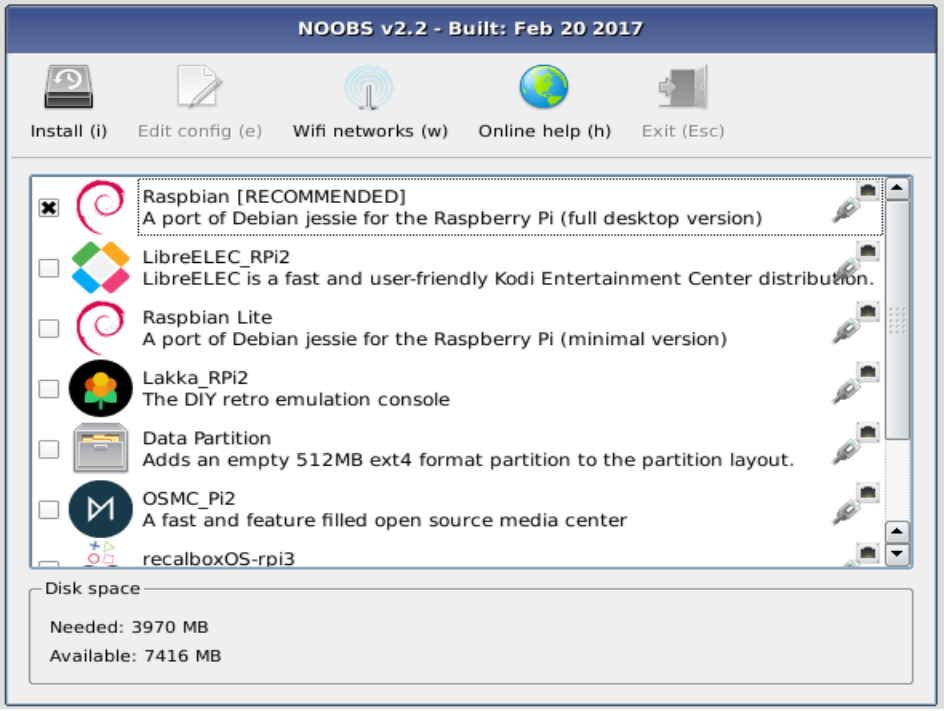
3.2. Dijelovi programa

U ovom poglavlju završnog rada bit će opisani svi napisani dijelovi programa. Korišten je programski jezik Python kako bi se dobila informacija sa senzora o temperaturi i relativnoj vlažnosti zraka. Za prikaz datuma, vremena te podataka sa senzora korišteni je HTML, JavaScript i PHP, te za pogodan izgled iskorišten je stilski jezik CSS. Uz sve te dijelove programa bilo je potrebno instalirati operacijski sustav, te napraviti nekoliko izmjena na samom Raspberry Pi-u kao što su uklanjanje *boot up* zaslona pri pokretanju, te izmjena pozadinske slike.

3.2.1 Instalacija operacijskog sustava

Kako bi bilo moguće raditi s Raspberry Pi-om potrebno je prvo instalirati operacijski sustav. NOOBS (*New Out Of Box Software*) je jednostavan upravitelj za instalaciju operacijskog sustava na Raspberry Pi-u. Raspberry Pi koristi *microSD* kartice, te je zbog toga prvo potrebno staviti NOOBS na SD karticu. Moguće je kupiti SD karticu sa unaprijed instaliranom NOOBS programskom podrškom. Preporučeno je koristiti SD karticu sa minimalnim kapacitetom od 8 GB. Ukoliko nije kupljena kartica s unaprijed instaliranim NOOBS-om, prvo što treba napraviti je staviti karticu u računalo koje ima čitač kartica, te zatim skinuti NOOBS programsku podršku sa službene Raspberry Pi stranice i raspakirati ju na karticu. Preporučeno je formatirati SD karticu prije kopiranja datoteka na nju. Postoji više načina za formatirati karticu. Najlakši način je jednostavno pronaći učitanu SD karticu na računalo, te desnim klikom na nju otvoriti izbornik koji ima opciju *Format*. Osim toga moguće je i skinuti sa interneta programske podrške namijenjene za formatiranje SD kartica.

Nakon uspješnog prenošenja NOOBS programske podrške na SD karticu, kartica je spremna za korištenje u Raspberry Pi-u. Nakon što se ubaci kartica i upali računalo, pokrenut će se NOOBS prozor sa listom operacijskih sustava koji se mogu instalirati (slika 3.2.).



Slika 3.2. NOOBS prozor.

Preporučeno je instalirati *Raspbian* operacijski sustav. Sljedeći korak je označiti operacijski sustav i kliknuti na *Install*. Za upravljanje je moguće uključiti miš i tipkovnicu u neki od 4 USB porta. Potrebno je određeno vrijeme da se instalacija završi, te se nakon nje pojavljuje konfiguracijski izbornik u kojemu je moguće postaviti datum, vrijeme i druge postavke. Zadano korisničko ime za *Raspbian* jest *pi*, dok je zadana zaporka *raspberry*. Potrebno je paziti jer pri upisivanju lozinke na zaslonu se neće pojaviti nikakvi znakovi. To je jedno od sigurnosnih svojstava *Linuxa*.

3.2.2 Python i DHT22

Za podatke o temperaturi i relativnoj vlažnosti zraka korišten je senzor DHT22. Da bi se dobili podatci sa senzora, nakon što se spoji sa računalom, potrebno je pokrenuti skriptu u Pythonu (Programski kod 3.8.).

```
import os
import Adafruit_DHT as dht

os.environ['PYTHON_EGG_CACHE'] =
'/tmp'
DHT = 23

h,t = dht.read_retry(dht.DHT22, DHT)

print(t)
print(h)
```

Programski kod 3.8. Python skripta za senzor DHT22.

PHP pokreće Python skriptu. Budući da se PHP ne može pokrenuti bez poslužitelja, zbog toga je potrebno napraviti lokalni Apache poslužitelj. Python nekada ima problema s pokretanjem na Apache poslužitelju te je zbog toga ubačena biblioteka *os*. Za zaobilazak problema s pokretanjem skripte služi linija koda *os.environ['PYTHON_EGG_CACHE'] = '/tmp'*. U riječniku *environ* varijabli *'PYTHON_EGG_CACHE'* postavlja vrijednost *'/tmp'*.

Python ima biblioteku za rad sa senzorom koja se zove Adafruit_DHT. Potrebno je skinuti biblioteku s interneta, te instalirati kako bi se mogla koristiti. Prvo što je potrebno napraviti je osigurati da sustav može skinuti Python biblioteke koristeći *pip*. *Pip* je sistem za upravljanje paketima koji se koristi za instalaciju i upravljanje paketima napisanim u Pythonu. *Pip* se može instalirati otvaranjem terminalnog prozora te upisivanjem komandi prikazanih u programskom kodu 3.9.

```
sudo apt-get update
sudo apt-get install python-pip
sudo python -m install --upgrade pip setuptools wheel
```

Programski kod 3.9. Komande za instalaciju *pip* sistema za upravljanje paketima.

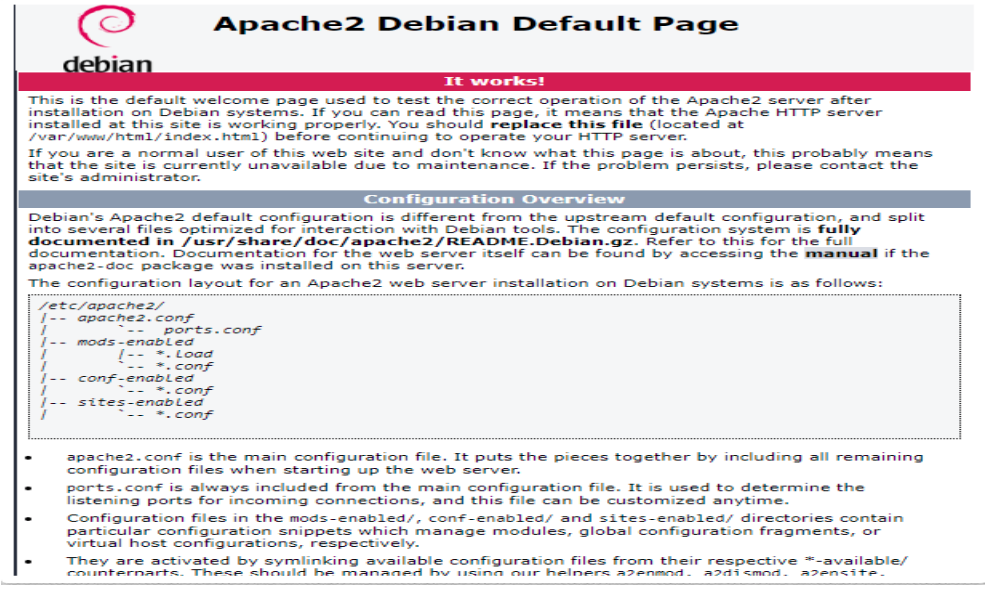
Sudo je skraćenica za „*super user do"* i omogućava programu da se pokrene kao glavni korisnik. *Apt-get update* naredba skida listu paketa iz repozitorija i nadopunjuje ju najnovijim verzijama paketa. *Apt-get install python-pip* instalira sustav za upravljanje paketima u Pythonu. Zadnja naredba nadopunjuje *pip* i njegove alate najnovijim nadogradnjama. Nakon instalacije sustava za upravljanje paketima ostaje još instalacija biblioteke *Adafruit_DHT*. Koristeći *pip* potrebna je

samo jedna naredba za instalaciju, *sudo pip install Adafruit_DHT*. Kada je biblioteka instalirana potrebno je resetirati računalo nakon čega se može koristiti u Python skripti.

U varijablu *DHT* se sprema broj pina Raspberry Pi-a na koji će se spojiti DHT senzor. *Dht.read_retry (dht.DHT22, DHT)* je funkcija iz *Adafruit_DHT* biblioteke koja uzima podatke iz senzora te ih sprema u dvije varijable *h* i *t*. U varijablu *h* se spremaju podatci o relativnoj vlazi zraka, dok se u varijablu *t* spremaju podatci o temperaturi. Nakon dohvaćanja podataka, temperatura i relativna vlažnost zraka će biti ispisani te će se koristiti u PHP skripti.

3.2.3 PHP skripta

PHP se koristi za prikaz kalendara te za dohvaćanje i prikaz podataka o temperaturi i relativnoj vlažnosti zraka. PHP je *server-side* skriptni programski jezik i zbog toga je potrebno instalirati Apache lokalni poslužitelj. Prvo što je potrebno napraviti jest instalirati Apache poslužitelj na Raspberry Pi-u. To se može napraviti pokretanjem naredbe *sudo apt-get install apache2 –y* u terminalnom prozoru. Naredba *–y* kaže programu da na sva pitanja prilikom instalacije automatski odgovori sa da (*yes*), kako ne bi čekao da korisnik odgovori da ili ne. Sljedeći korak jest instalacija PHP-a. To se može napraviti pokretanjem naredbe *sudo apt-get install php5* u terminalnom prozoru. Kada se pokrene internet preglednik te upiše adresa *http://localhost/* dobit će se zadana internet stranica koja pokazuje da je poslužitelj ispravno instaliran (slika 3.3.). Na njoj pišu instrukcije za nastavak rada s poslužiteljem.



Slika 3.3. Zadana stranica Apache poslužitelja.

Svakih 30 sekundi internet stranica se osvježava, te osvježava podatke. To je napravljeno HTML linijom *<meta http-equiv="refresh" content="30"/>*. Za realizaciju priakaza temperature i vlažnosti zraka, potrebno je u PHP-u pamtiti podatke o njihovim vrijednostima. Zbog toga se koristi *session* koji pamti varijable internet stranice sve dok se pretraživač ne ugasi. Na početku skripte linija *session_start();* pokreće *session*.

Kalendar je napravljen koristeći PHP. S obzirom da je svaki mjesec drugačiji (nemaju svi isto dana, ne počinje svaki mjesec na isti dan u tjednu) PHP popunjava tablicu redak po redak. Prvo provjerava koji je dan, mjesec, koliko mjesec ima dana, te koji je prvi dan u mjesecu. Kada se zna prvi dan u mjesecu može se napraviti prvi redak u tablici. Nakon prvog retka, ostali retci su jednostavni za popuniti. Način na koji je kalendar napravljen prikazan je u programskim kodovima 8.3., 8.4.

U programskom kodu 8.5. prikazana je PHP skripta koja pokreće Python skriptu, dohvaća podatke sa senzora i prikazuje ih na internet stranici. Linija PHP skripte koja pokreće Python skriptu je *exec(„python /var/www/html/dht22.py", $sensorData);*. Podatci iz Python skripte se spremaju u varijablu *$sensorData*. Podatci sa senzora se prikupljaju svakih 30 sekundi. Kako ne bi došlo do nagle promjene u temperaturi ili relativnoj vlažnosti zraka uzrokovane vanjskim elementima (iznenadan hladan ili topal zrak), koristi se polje s 10 elemenata u koje se spremaju podatci. Na zaslonu je prikazana srednja vrijednost tih 10 elemenata, odnosno srednja vrijednost zadnjih 10 mjerenja senzora. *Session* se koristi iz razloga što moramo pamtiti podatke u polju i kad se internet stranica osvježi. Nakon izračuna srednje vrijednosti, podatci se prikažu na zaslonu.

4. TESTIRANJE

U ovom poglavlju će biti prikazan rad pametnog ogledala, te će biti testirana točnost rada.

4.1. Prikaz rada

Na zaslonu je prikazan kalendar napravljen u skriptnom jeziku PHP, te vrijeme koje je prikazano pomoću JavaScripta. Temperatura i relativna vlažnost zraka su prikazani pomoću PHP-a koji pokreće Python skriptu. Svi podatci su prikazani na slici 4.1.

Slika 4.1. Izgled zaslona pametnog ogledala.

4.2. Testiranje rada

Zbog nemogućnosti mjerenja trenutne relativne vlažnosti zraka, senzor je testiran samo na preciznost mjerenja temperature zraka. U trenutku testiranja temperatura zraka iznosi oko 27°C kao što je prikazano na slici 4.2.

Senzor je testiran korištenjem toplog zraka iz fena. Senzor i termometar su izloženi toplom zraku jednak broj sekundi te su očitanja temperature zapisana u tablici 4.1. Temperatura senzora i termometra prije početka testiranja iznosi 27.5°C.

Vrijeme izloženo toplom zraku	Temperatura senzora DHT22[°C]	Temperatura termometra[°C]
3 sekundi	31.5	31
5 sekundi	33	34
8 sekundi	36	37
10 sekundi	39.5	40

Tablica 4.1. Testiranje senzora DHT22

Slika 4.2. Testiranje preciznosti mjerenja temperature zraka.

5. ZAKLJUČAK

U završnom radu je korišten Raspberry Pi 3 model B, ali osim njega za izradu pametnog ogledala se može iskoristiti i Arduino. Za prikaz podataka korišten je LCD zaslon veličine 5 incha, odnosno dimenzija 121x78mm. Raspberry Pi 3B nema mogućnost pamćenja vremena, te se zbog toga koristi RTC sat koji pamti datum i vrijeme. Za mjerenje temperature i relativne vlažnosti zraka koristi se senzor DHT22. Na zaslonu su prikazani kalendar, trenutno vrijeme i podatci sa senzora.

Za izradu završnog rada korišten je skriptni jezik PHP. Programski jezik Python dohvaća podatke sa senzora. PHP pokreće Python skriptu, te prikazuje podatke na zaslonu. Također, PHP prikazuje kalendar koji je uređen pomoću stilskog jezika CSS. Trenutno vrijeme je prikazano koristeći JavaScript.

Izrađeno ogledalo ugrađeno je u kompaktno kućište koje štiti elektroničke komponente i omogućava postavljenje pametnog ogledala na stol ili zid.

Nakon izrade svih sklopovskih i programskih dijelova, pametno ogledalo je testirano više puta da se utvrdi ispravno funkcioniranje. Također, provjerena je točnost rada temperaturnog senzora na način da je mjerena vrijednost uspoređivanja s vrijednošću temperature izmjerene dodatnim termometrom.

Prilikom izrade završnog rada, uočen je problem s veličinom zaslona, jer je količina podataka za prikaz ograničena na trenutno korištenom LCD zaslonu. Kako bi se omogućio prikaz veće količine podataka i napravilo više funkcionalnosti, u budućem radu bi se moglo koristiti veći LCD zaslon (oko 20 incha, odnosno dimenzija 410 x 300mm) koji bi prikazivao podatke preko cijelog ogledala.

Za prikaz podataka korištena je internet stranica koja se nalazi na lokalnom poslužitelju. Umjesto internet stranice moguće je napraviti aplikaciju koja se pokreće na Linux operacijskom sustavu i prikazuje željene podatke. Jedan od načina za napraviti aplikaciju je korištenjem *Qt* programske podrške.

6. LITERATURA

[1] Raspberry Pi Foundation, https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started [7.9.2019.]

[2] Encyclopedia, https://www.encyclopedia.com/science-and-technology/computers-and-electrical-engineering/computers-and-computing/liquid-crystal-display [4.9.2019.]

[3] The Engineering Projects, https://www.theengineeringprojects.com/2019/02/introduction-to-dht22.html [8.9.2019.]

[4] P., Barry, Head First Python: A Brain-Friendly Guide, O'Reilly Media, 2010

[5] J., Ducket, HTML&CSS design and build wesites, John Wiley & Sons, Inc, Indianapolis, Indiana, 2011

[6] J., Ducket, JavaScript&jQuery interactive front-end web development, John Wiley & Sons, Inc, Indianapolis, Indiana, 2014

[7] L., Beighley, M., Morrison, Head First PHP & MySQL: A Brain-Friendly Guide, O'Reilly Media, 2008

7. SAŽETAK

U završnom radu objašnjen je jedan od načina kako napraviti pametno ogledalo. Pametno ogledalo na sebi ima ispisane podatke o datumu i vremenu. Osim toga pokazuje i podatke o trenutnoj temperaturi zraka i relativnoj vlažnosti zraka. Postoji veliki broj načina za izradu pametnog ogledala. Završni rad je realiziran korištenjem minijaturnog računala Raspberry Pi 3B. Informacije na ogledalu su prikazane preko LCD zaslona, a podatci o vlazi i temperaturi dobiveni su sa senzora za vlagu i temperaturu DHT22. Senzor je spojen na Raspberry Pi 3B korištenjem eksperimentalne pločice. Tehnologije korištene u završnom radu su programski jezik Python koji se koristi za komunikaciju sa senzorom, skriptni jezik PHP, te HTML, JavaScript i stilski jezik CSS. Za rad PHP-a potreban je poslužitelj, te se zbog toga koristi Apache. Zbog male veličine LCD zaslona količina podataka za prikaz na ogledalu je ograničena. Kućište pametnog ogledala je napravljeno od drveta, te je zbog ljepšeg izgleda ulijepljeno tapetama za namještaj.

Ključne riječi: pametno ogledalo, raspberry pi, DHT22.

8. ABSTRACT

Smart mirror

This final paper is explaining one of the ways of making a smart mirror. Smart mirror is showing information about date and time. Except for date and time, it is also showing current air temperature and relative humidity. There are many ways to make a smart mirror. This one is made using miniature computer Raspberry Pi 3B. Information on the mirror are shown over LCD display. The information about temperature and relative humidity of the air come from DHT22 sensor. Sensor is connected to Raspberry Pi 3B with a breadboard. Technology used for this smart mirror is Python programming language which is used for communication with sensor, scripting language PHP, HTML, JavaScript and CSS. Apache is used to make a local server for PHP scripts. Because of the small LCD display size, the amount of information shown on the display is limited. Smart mirror case is made out of wood and surrounded with white wallpaper for nicer appearance.

Key words: smart mirror, raspberry pi, DHT22.

ŽIVOTOPIS

Jerko Kosić rođen je 20.1.1998 u Žepču, Bosna i Hercegovina. Prebivalište mu je u Žepču, gdje i upisuje osnovnu školu OŠ ŽEPČE 2003. godine, te nakon odličnog uspjeha, upisuje srednju školu KŠC Don Bosco Žepče. Sudjelovao je na državnom natjecanju iz Tehničke Mehanike u Osijeku, 2015. godine. Nakon završene srednje škole, 2016. godine upisuje preddiplomski sveučilišni studij Računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Zanima ga Android, a od programskih jezika poznaje C, C++, Javu,Swift, te HTML, CSS, JavaScript i PHP. Izvrsno poznaje engleski jezik.

PRILOZI

```
import os
import Adafruit_DHT as dht

os.environ['PYTHON_EGG_CACHE'] = '/tmp'
DHT = 23

h,t = dht.read_retry(dht.DHT22, DHT)

print(t)
print(h)
```

Programski kod 8.1. Python skripta za dohvaćanje podataka sa senzora.

```
<?php session_start(); ?>
<!DOCTYPE html>
<html lang="en">
<head>
      <meta http-equiv="refresh" content="30"/>
      <meta charset="utf-8">
      <title>Pametno ogledalo</title>
      <style>
            #dateTime{
                  font-size:70px;
                  font-family:Gentium-R;
                  color:white;
                  float: left;
                  margin-top:5px;
                  margin-left:50px;}
            #tempHum{
                  font-size:40px;
                  font-family:Gentium-R;
                  color:white;}
            .days {
                  padding: 0px 10px 0px 10px;
                  border-bottom: 1px solid white;
                  border-top: 1px solid white;}
            #calendar {
                  font-size:40px;
                  font-family:Gentium-R;
                  color:white;
                  float: left;
                  margin-left: 40px;
                  border: 1px solid white;}
            th, td{ text-align:center; }
            body{ background: black; }
      </style>
      <script>
            function startTime() {
                  var today = new Date();
                  var h = today.getHours();
                  var m = today.getMinutes();
                  m = checkTime(m);
                  document.getElementById('dateTime').innerHTML = h
+ ":" + m;
            }
            function checkTime(i) {
                  if(i < 10) {i = "0" + i}; // dodaje nulu ispred
brojeva manjih od < 10
                  return i;  }
      </script> </head>
```

Programski kod 8.2. PHP skripta (1. dio).

```
<body onload="startTime()">
<div id="calendar">
     <?php
          $date = time();
$day = date('d', $date);
          $month = date('m', $date);
          $month_int = date('n')-1;
          $year = date('y', $date);
          $month_text = array("Siječanj", "Veljača", "Ožujak",
"Travanj", "Svibanj",
          "Lipanj" , "Srpanj" ,"Kolovoz" ,"Rujan" ,"Listopad"
,"Studeni" ,"Prosinac");
          $first_day_of_month = date('D',mktime(0, 0, 0,
date('m'), 1));
switch($first_day_of_month)
          {
               case "Mon": $blank = 0; break;
               case "Tue": $blank = 1; break;
               case "Wed": $blank = 2; break;
               case "Thu": $blank = 3; break;
               case "Fri": $blank = 4; break;
               case "Sat": $blank = 5; break;
               case "Sun": $blank = 6; break;
          }
          $days_in_month = cal_days_in_month(0, $month, $year);
          echo "<table>";
          echo "<tr>
               <th colspan='4'>$month_text[$month_int]</th>
               <th colspan='3'>20$year</th>
                    </tr>";
          echo "<tr>
               <td class='days'>PON</td>
               <td class='days'>UTO</td>
               <td class='days'>SRI</td>
               <td class='days'>CET</td>
               <td class='days'>PET</td>
               <td class='days'>SUB</td>
               <td class='days'>NED</td>
                    </tr>";
```

Programski kod 8.3. PHP skripta (2. dio).

```
$day_count = 1;
            echo "<tr>";
            while($blank > 0) {
                  echo"<td></td>";
                  $blank -=1;
                  $day_count++;
            }
            $day_num = 1;
            while($day_count < 8) {
                  if($day_num == $day){
                        echo "<td style='border:2px solid
white;font-weight:bold'>$day_num</td>";
                  } else {
                  echo "<td>$day_num</td>";
                  }
                  $day_num++;
                  $day_count++;
            }
echo "</tr>";
            $day_count = 1;
            while($day_num < $days_in_month) {
                  echo "<tr>";
                  while($day_count < 8) {
                        if($day_num > $days_in_month) {break;}
                        if($day_num == $day){
                              echo "<td style='border:2px solid
white;font-weight:bold'>$day_num</td>";
                        } else {
                              echo "<td>$day_num</td>";
                        }
                        $day_count++;
                        $day_num++;
                  }
                  $day_count = 1;
                  echo "</tr>";
            }
            echo "</table>";
      ?>
</div>
<div id="dateTime"></div>
```

Programski kod 8.4. PHP skripta (3. dio).

```
<div id="tempHum">
     <?php
          $temperature = 0;
          $humidity = 0;
          exec("python /var/www/html/dht22.py", $sensorData);
if(!isset($_SESSION["counter"]) or $_SESSION["counter"] == 10) {
               $_SESSION["counter"] = 0;
          }
          if(!isset($_SESSION["temp"])) {
               $_SESSION["temp"] = array(x,x,x,x,x,x,x,x,x,x);
          }
          if(!isset($_SESSION["hum"])) {
               $_SESSION["hum"] = array(x,x,x,x,x,x,x,x,x,x);
          }
          $_SESSION["temp"][$_SESSION["counter"]] = $sensorData[0];
          $_SESSION["hum"][$_SESSION["counter"]] = $sensorData[1];
          for($i = 0; $i < 10; $i++) {
               if($_SESSION["temp"][$i] == x) {break;}
               $temperature += $_SESSION["temp"][$i];
               $humidity += $_SESSION["hum"][$i];
          }
          echo "<img src='temp_icon.jpg'
style='width:50px;height:50px;margin-left:85px;margin-top:15px;'/>";
          echo number_format((float)$temperature/$i,1, '.',
'')."°C";

          echo "<img src='hum_icon.png'
style='width:50px;height:50px;margin-left:85px;margin-top:15px;'/>";
          echo number_format((float)$humidity/$i,2, '.', '')."%";
          $_SESSION["counter"]++;
     ?>
</div>
</body>
</html>
```

Programski kod 8.5. PHP skripta (4. dio).