



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradbu informacija
Sveučilište u Zagrebu

DKU-IRIM- zadaci za učenike

- Δ Namijenjeno učenicima viših razreda osnovnih škola i srednjoškolcima
- Δ Softver Arduino IDE, mobilna aplikacija Blynk, MKR1000
- Δ Upravljanje Arduino pločicom pomoću mobilne aplikacije Blynk
- Δ Korištenje senzora za temperaturu, LCD zaslona, RGB diode, servo motora

7. lipnja 2020

Sažetak

Projekt „Zadaci za učenike-IRIM“ kao glavni cilj ima osmisliti i izraditi edukacijske materijale za mlade osnovnoškolske i srednjoškolske dobi. Korištenjem Arduina MKR1000 s ugrađenim WiFi modulom i odgovarajućih senzora i drugih komponenti realiziran je se sustav IoT sefa.

Napravljene su upute i dan programski kod za izradu ovog jednostavnog projekta kojem je glavna svrha upoznavanje djece i mladih s osnovnim principima rada s Arduinom i područjem Internet of Thingsa.

Sadržaj

1. UVOD.....	3
2. KONCEPT SUSTAVA.....	4
3. KORIŠTENE KOMPONENTE	5
3.1 Shema sustava	5
3.2 Arduino MKR1000.....	6
3.3 Senzor za temperaturu	7
3.4 RGB dioda	7
3.5 Servo-motor	8
3.6 Alfanički LCD zaslon	8
3.7 Potenciometri (10 kOhm)	9
4. DIZAJN I IZRADA SUSTAVA	11
5. UPUTE ZA IZRADU BLYNK APLIKACIJE.....	17
6. PROGRAMSKI KOD.....	22
7. ZAKLJUČAK.....	32
8. LITERATURA.....	33

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

1. UVOD

STEM je engleski akronim za „Science, Technology, Engineering and Mathematics“. Kroz STEM obrazovanje kod mladih se potiče kritičko razmišljanje, povećava znanstvena pismenost i razvijaju poželjne osobine koje će u budućnosti omogućiti razvoj mladih inovatora.

Institut za razvoj i inovativnost mladih (IRIM) pokrenuo je niz projekata kako bi približio STEM područje mladima [1]. Kroz različite edukativne radionice, djeca i mladi se kroz jednostavne zadatke upoznaju sa različitim tehnologijama i razvijaju digitalnu pismenost.

Materijali za izradu jednostavnih projektnih zadataka namijenjeni djeci i profesorima objavljeni su na digitalnom portalu Izradi!.

Cilj ovog projekta je osmisliti i izraditi edukacijske materijale koji će biti objavljeni na stranici Izradi!. Zadatak koji je osmišljen je izrada Internet of Things sefa. Osmišljeni zadatak je popraćen uputama za izradu hardverskog i softverskog dijela. Svrha projekta je upoznavanje mladih sa područjem Internet of Things kroz upotrebu Arduina i Blynk mobilne aplikacije.

2. Koncept sustava

U ovom se projektu izrađuje kasa/sef za osobne potrebe koji ima posebno svojstvo da mjeri temperaturu korisnika i onemogućuje otvaranje sefa ako je povišena.

Sef se izrađuje od plastične kutije koja je izrezana i spojena prema nacrtu. Također, sef može biti izrađen i od papira ili kartona.

Sef se koristi tako da korisnik postavlja kombinaciju koja otključava sef putem Blynk aplikacije. Pomoću potencijometara na kutiji, postavlja se kombinacija i u ovisnosti odgovara li ona ključnoj, pokreće se servo motor koji otvara vrata sefa. Tijekom postavljanja kombinacije, RGB dioda signalizira bojom koliko je kombinacija blizu onoj ključnoj. Toplija boja znači da je korisnik bliže pogotku).

Posebna svojstva sefa su mjerenje temperature korisnika, ispis vrijednosti u stupnjevima °C na LCD ekranu i svjetleća signalizacija o blizini ključne kombinacije.

Materijal koji se može naći besplatno je plastika, karton ili papir (može se iskoristiti ambalaža od proizvoda koje nalazimo u kućanstvu, npr. stara kartonska kutija).

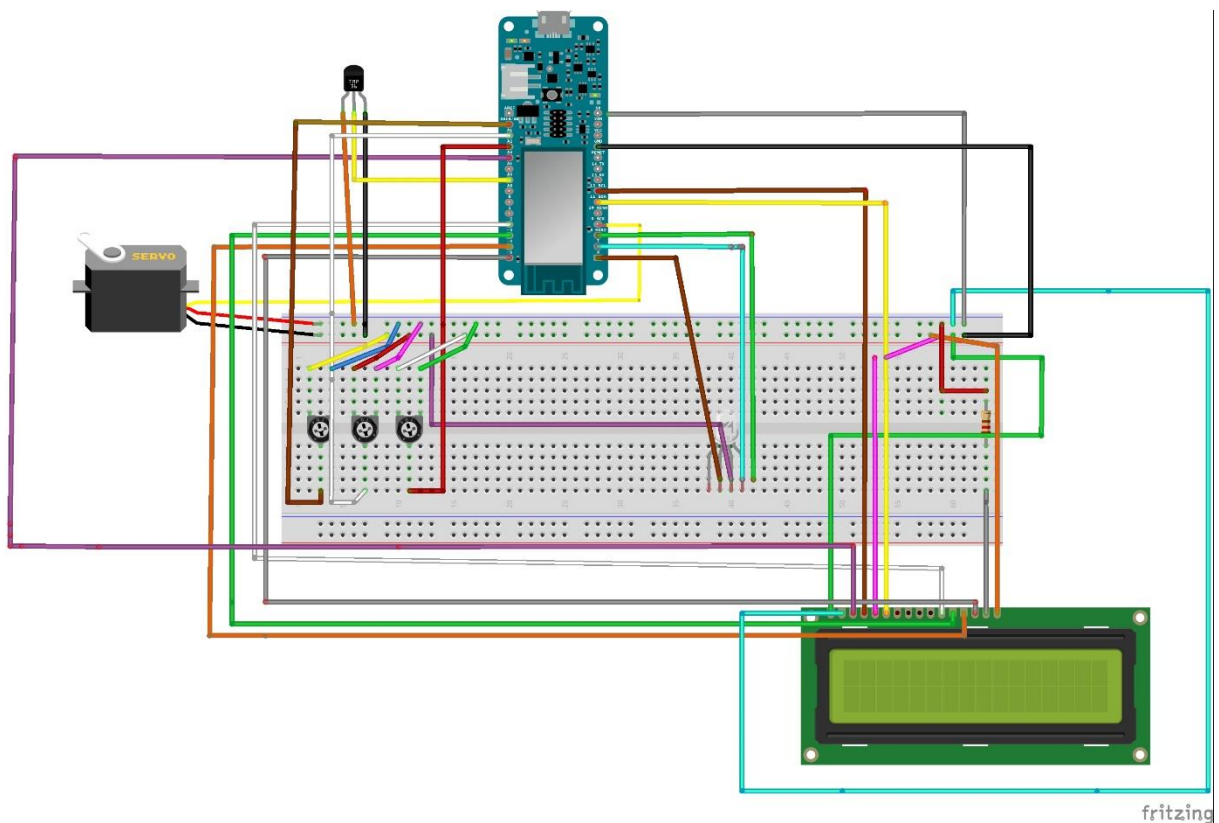
Materijal koji se mora nabaviti (Arduino MKR IoT bundle) nabavlja se putem interneta i košta otprilike 70 €. Dodatci za dizajn sustava i pribor za lijepljenje prilikom izrade nabavljeni su u trgovini mješovite robe i koštaju otprilike 30 kn.

3. KORIŠTENE KOMPONENTE

Za izradu projekta koristile su se sljedeće komponente: Arduino MKR1000, senzor za temperaturu, servo motor, LCD zaslون, mobilni telefon s Blynk aplikacijom, RGB dioda, potenciometri.

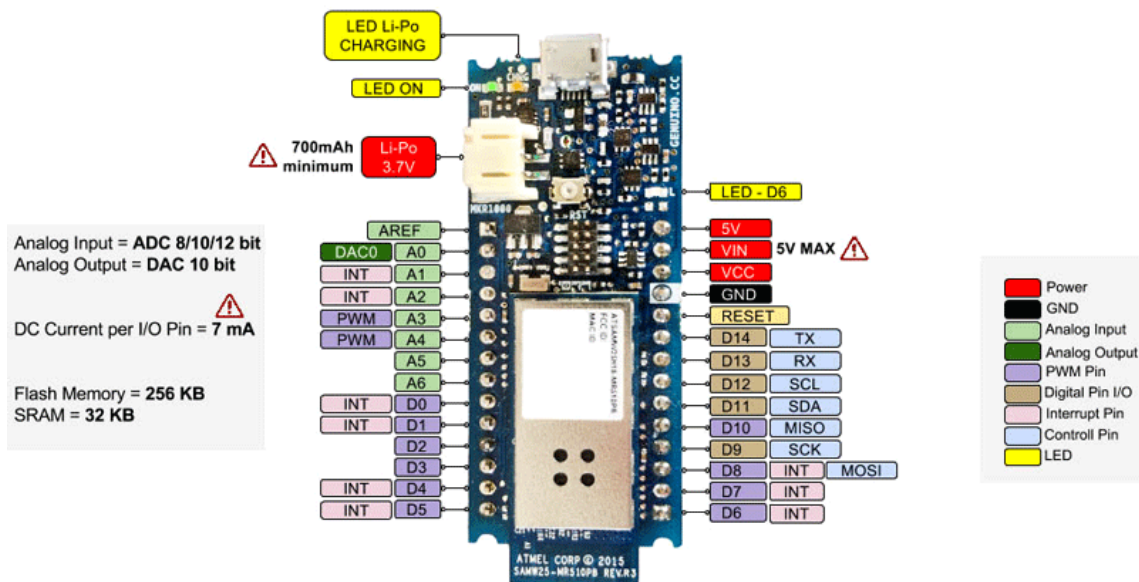
3.1 Shema sustava

Na Sliku 1 prikazana je shema spajanja sustava IoT sefa. Shema je izrađena korištenjem Fritzing programa (<https://fritzing.org/home/>).



Slika 1. Shema sustava

3.2 Arduino MKR1000



Slika 2. Arduino MKR1000

Arduino MKR1000 na sebi sadrži mikrokontroler, modul za povezivanje na WiFi, sklopovlje za napajanje putem LiPo baterije i njezino punjenje putem USB kabla te drugo sklopovlje potrebno za rad mikrokontrolera. Mikrokontroler možemo razmatrati kao malo računalo kojim možemo očitavati stanja različitih senzora i upravljati raznim uređajima putem programa koji pišemo u Arduino IDE razvojnom sučelju, a koji se onda izvršava na mikrokontroleru.

Izvodi mikrokontrolera služe kako bismo na njega mogli spajati elektroničke elemente, a zatim upravljati njima pomoću mikrokontrolera.

Da bi mikrokontroler znao što treba raditi i kako da upravlja s onime što na njega spojimo, moramo napisati program, a potom taj program učitati u mikrokontroler.

Postoje dvije vrste izvoda: digitalni i analogni izvodi, a svaki od njih može raditi u dva režima rada: ulazni i izlazni režim.

Digitalni izvodi obilježeni su brojem (0-14), ili slovom D i brojem, a mogu imati 2 stanja, 0 ili 1, odnosno isključeno ili uključeno

Analogni izvodi obilježeni su slovom A i brojem (0-6). Analogne izvode možemo koristiti kao digitalne, ali digitalne ne možemo koristiti kao analogne.

3.3 Senzor za temperaturu

Svrha senzora za temperaturu u projektu je mjerenje temperature korisnika. Ako je temperatura korisnika u rasponu normalne tjelesne temperature, korisnik je u mogućnosti otvoriti sef postavljanjem dobitne kombinacije na potenciometrima.

U projektu se koristi temperaturni senzor TMP36. Na izlazu daje napon linearno proporcionalan temperaturi u $^{\circ}\text{C}$. Temperaturni raspon od -40°C do 125°C pretvara se u napon od 100 mV do 1,75 V s rezolucijom od $0,1^{\circ}\text{C}$. Senzor na mikrokontroler spajamo tako da PIN 1 spojimo na napajanje, PIN 2 nam daje izlazni napon, a PIN 3 služi za uzemljenje. Programski, s Arduino pina 5 se očitava vrijednost temperature.



Slika 3. Temperaturni senzor

3.4 RGB dioda

Zadatak višebojne RGB (sadrži tri osnovne boje: crvena, zelena i plava boja) diode u projektu je generiranje boje ovisno o blizini pogotka naše postavljene kombinacije i dobitne kombinacije koja je postavljena preko Blynk aplikacije. Programski je izvedeno da za različite udaljenosti namještene preko potenciometara, određena boja zasvijetli. Četiri su moguće boje koje generiramo ovisno o blizini pogotka kombinacije. Što smo bliže dobitnoj kombinaciji, to RGB dioda generira topliju boju (crvena kao najtoplija), a što smo udaljeniji, to generira hladniju boju (plava kao najhladnija).

Primjerice, ako smo u Blynku postavili kao traženu kombinaciju 999, a potenciometrima korisnik je namjestio kombinaciju 888 dioda će zasvijetliti crveno i time mu signalizirati da je „blizu“ rješenja. S druge strane, ako je kombinacija zadana 999, a korisnik je unio 333 onda će dioda svijetliti plavom bojom i vizualno mu signalizirati da nije blizu kombinacije koja otvara kutiju.

Boja diode odabire se postavljanjem vrijednosti u rasponu od 0 do 255 za svaki od izvoda diode. Diodine priključke spajamo na PWM priključke Arduina, a u programu ih kontroliramo funkcijom `void postaviBojudiode(int red, int green, int blue)`.



Samostalna RGB dioda

Slika 4: RGB dioda

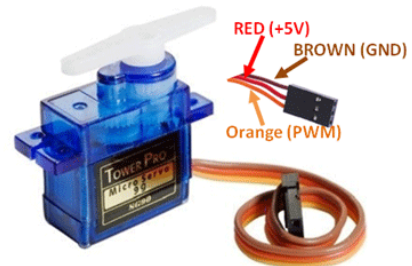
3.5 Servo-motor

Servo-motor je komponenta koja može okretati svoju ručicu (obično između 0 i 180 stupnjeva).

U projektu servo motor služi za otvaranje sefa ukoliko je naša temperatura u rasponu normalne tjelesne temperature i ukoliko smo namjestili dobitnu kombinaciju na potenciometrima.

Upravljanje Arduina servo motorom omogućujemo učitavanjem biblioteke `#include<servo.h>`

Kut zakreta određuje širina PWM signala kao na Slika i Slika 7. Širina PWM signala (2)



Slika 5: Servo-motor



Slika 6. Širina PWM signala (1)



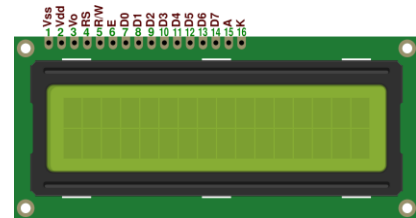
Slika 7. Širina PWM signala (2)

Ako je širina PWM signala `WIDTH_MIN`, servo motor pozicionira se u 0 stupnjeva, a ako je širina PWM signala `WIDTH_MAX`, servo motor rotira prema 180 stupnjeva. Servo motor na mikrokontroler spajamo tako da PIN1, koji nam predstavlja izlazni pin za Arduino, spojimo na PIN 9 od Arduina. PIN 2 spajamo na napajanje(5V), a PIN 3 na uzemljenje.

3.6 Alfnumerički LCD zaslon

LCD zaslon vrlo je popularna komponenta i naširoko se koristi u raznim elektroničkim projektima. Dobri su za prikaz podataka s raznih senzora.

Rad s LCD zaslonom Arduino omogućujemo učitavanjem LiquidCrystal biblioteke(`#include<LiquidCrystal.h>`).



Slika 8: LCD zaslon

U projektu LCD zaslon služi kako bi ispisao vrijednost temperature i kombinacije koje postavljamo preko potencijometara.

Definicija pinova s lijeva na desno:

PIN1(Vss): Osnovni pin LCD modula spaja se na GND

PIN2 (Vcc): Pin za napajanje (+5V)

PIN3 (VEE): Pin za podešavanje kontrasta (promjenom napona regulira se kontrast)

PIN4 (RS): Registar zapisa i podataka

PIN5 (R/W): Pin za definiciju pisanja / čitanja

PIN6 (E): Pin namijenjen za omogućavanje rada LCD modula

PIN7 – PIN14: Podatkovni pinovi

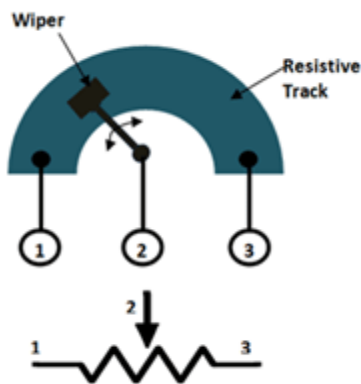
PIN15 (LED+): Anoda pozadinskog svjetla

PIN16 (LED-): Katoda pozadinskog svjetla

3.7 Potencijometri (10 kOhm)



Slika 9. Potencijometar



Slika 10. Mehanizam rada potencijometra

U projektu potencijometre koristimo kako bismo pokušali pogoditi ispravnu kombinaciju za otvaranje sefa. Pri tome ulazne cjelobrojne vrijednosti analognog pina(0-1023) skaliramo na vrijednosti brojeva 0-9 jer koristimo brojeve u tom rasponu

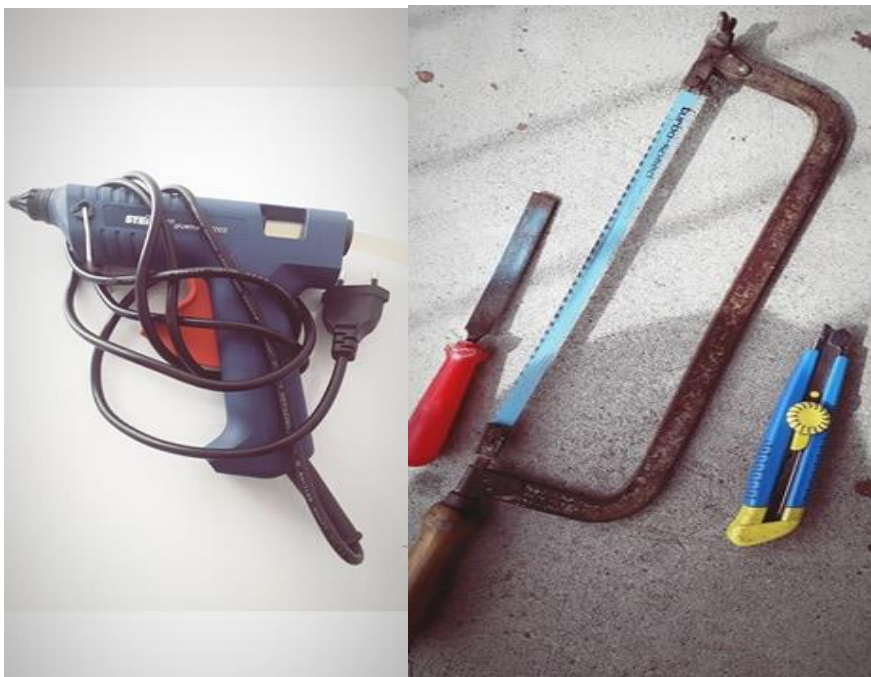
Potencijometri se obično sastoje od 3 pina(GND(pin koji spajamo na uzemljenje),VCC(pin koji spajamo na napajanje) i izlazni pin (prenosi napon na ulazni analogni pin od Arduina)). Osovina potencijometra može se zakretati od 0 stupnjeva(najbliže GND-u) do gornjeg graničnog kuta(najbliže VCC pinu) kojeg zovemo ANGLE_MAX. Izlazni napon proporcionalan je zakrenutim kutom osovine potencijometra. Ako je kut 0

stupnjeva, napon na izlaznom pinu je 0 V, također ako je kut maksimalan (ANGLE_MAX), na izlaznom pinu imamo VCC napon.

Ulazni analogni pin arduina pretvara vrijednost napona (između 0 V i VCC) u cijelobrojne vrijednosti (između 0 i 1023).

4. DIZAJN I IZRADA SUSTAVA

Prilikom izrade sustava korišteni su alati prikazani na slici: turpija, pila, skalpel, pištolj za vruće ljepilo, ručna bušilica, lemilica i samoljepljivi kolaž papir.



Slika 11. Pištolj za vruće ljepilo(slika lijevo) i turpija, pila i skalpel(slika desno)



Slika 12. Ručna bušilica (slika lijevo) i lemilica (slika desno)



Slika 13. Samoljepljivi kolaž papir

Postupak izrade:

Pilu i skalpel koristili smo za rezanje vrata za sef i za rezanje otvora namijenjenog servo motoru. Pištolj za vruće lijepljenje koristili smo za učvršćivanje komponenti na sef dok smo ručnom bušilicom bušili rupe za RGB diodu, potenciometre i senzor za mjerenje temperature. Na kraju smo lemilicom zalemili izvode potenciometara kako bi eliminirali oscilacije u

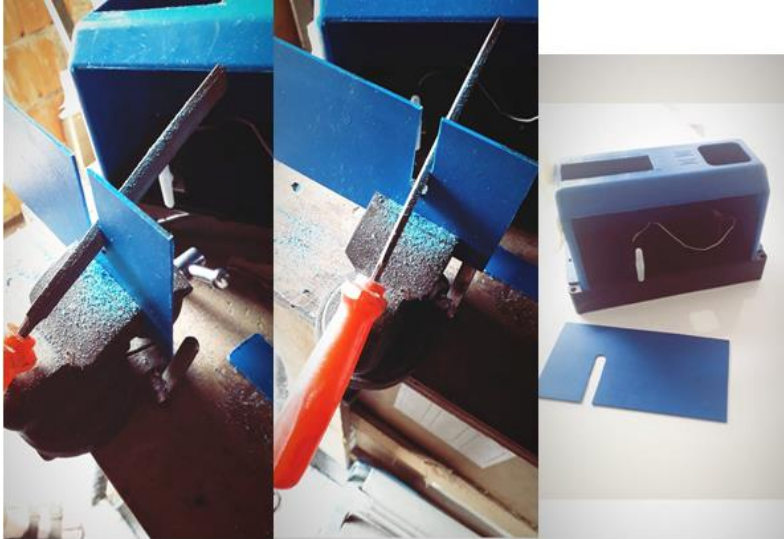
vrijednostima do kojih dolazi zbog mehaničkih nesavršenosti potencijometara.

Prvi korak: Koristimo plastičnu kutiju (Slika 14.) (u našem primjeru koristili smo plastičnu kutiju pumpe za vodu). Sustav je moguće izraditi i od drugih materijala, kao što su karton ili tvrđi papir. Može se iskoristiti kartonska ambalaža iz kućanstva koja se izreže po navedenim dimenzijama i nacrtu.



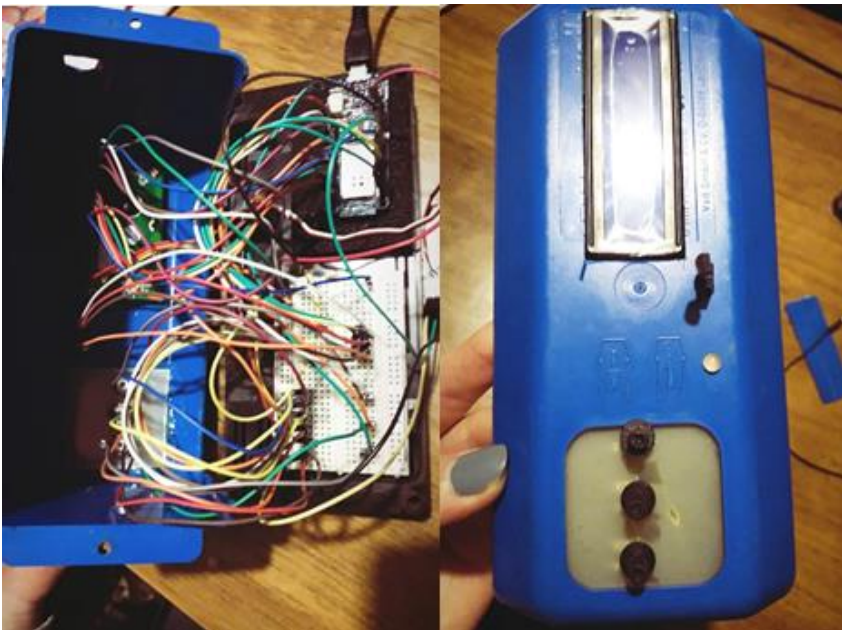
Slika 14. Plastična kutija korištena pri izradi sustava

Drugi korak: Pilom izrezujemo otvore za pojedine komponente koje će se nalaziti na vanjskoj strani sefa i za rezanje vrata koja ćemo koristiti za otvaranje sefa (Slika 15).



Slika 15. Drugi korak izrade kućišta za IoT sef

Treći korak: Pištoljem za lijepljenje pričvršćujemo komponente na vanjsku stranu sefa (Slika 16).



Slika 4. Treći korak izrade kućišta za IoT sef

Četvrti korak: Izrezujemo kvadratiće od samoljepljivog kolaža kako bismo pričvrstili vrata kojima otvaramo sef.



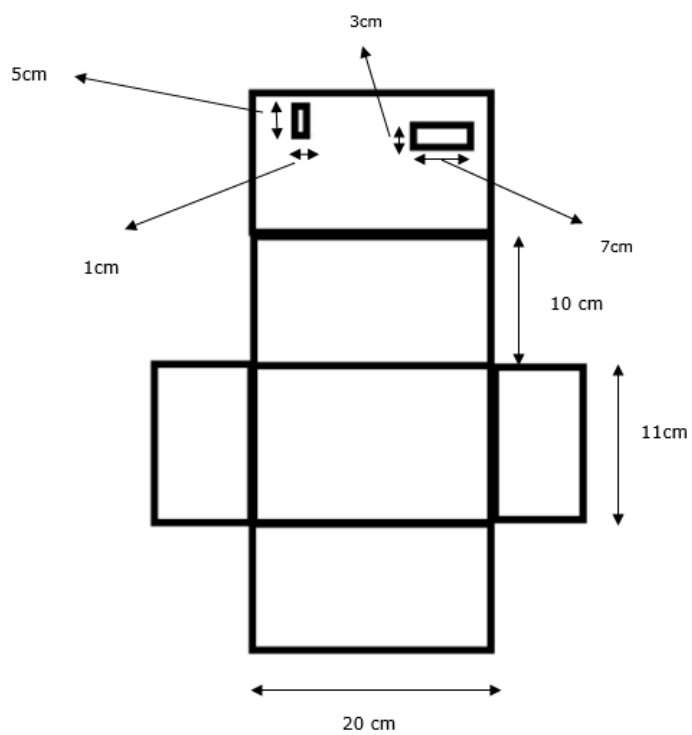
Slika 5. Kvadratići od samoljepljivog kolaža

Peti korak: Vanjski dizajn sustava i dovršavanje kućišta.



Slika 18. Vanjski dizajn sustava

Pri izradi vanjskog kućišta od kartona može se koristiti nacrt:

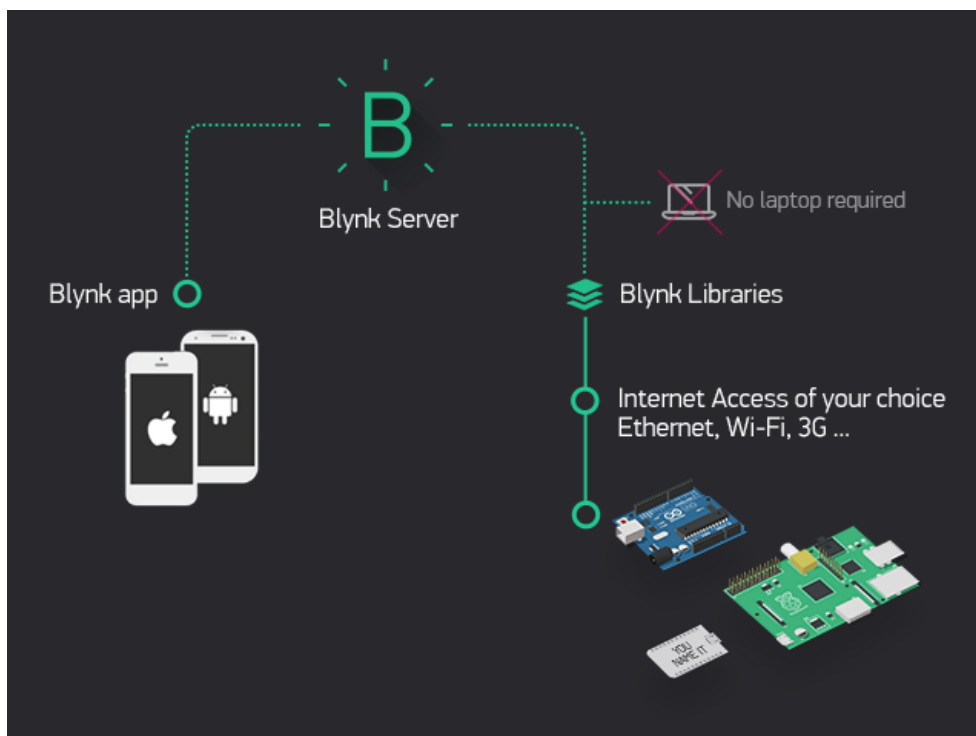


Slika 19. Nacrt za izradu vanjskog kućišta od kartona

5. UPUTE ZA IZRADU BLYNK APLIKACIJE

Blynk aplikacija (<http://www.blynk.cc/>) je aplikacija za pametne telefone i tablete koja omogućava brzu i jednostavnu izradu sučelja za upravljanje IoT projektima.

Princip rada Blynk platforme prikazan je sljedećom slikom:



Slika 20. Princip rada Blynk platforme

Izradom sučelja u Blynk aplikaciji putem Blynkovog cloud sustava omogućuje se izravna komunikacija aplikacije i hardvera (u ovom slučaju Arduina).

Postupak se sastoji od:

1. Prije izrade novog projekta, aplikaciju je potrebno preuzeti i instalirati na pametni telefon ili tablet.
2. Nakon preuzimanja aplikacije, potrebno je kreirati vlastiti korisnički račun.
3. Zatim je potrebno unijeti neke osnovne podatke o projektu.
4. Potom aplikacija kreira prazan projekt spreman za daljnji rad.

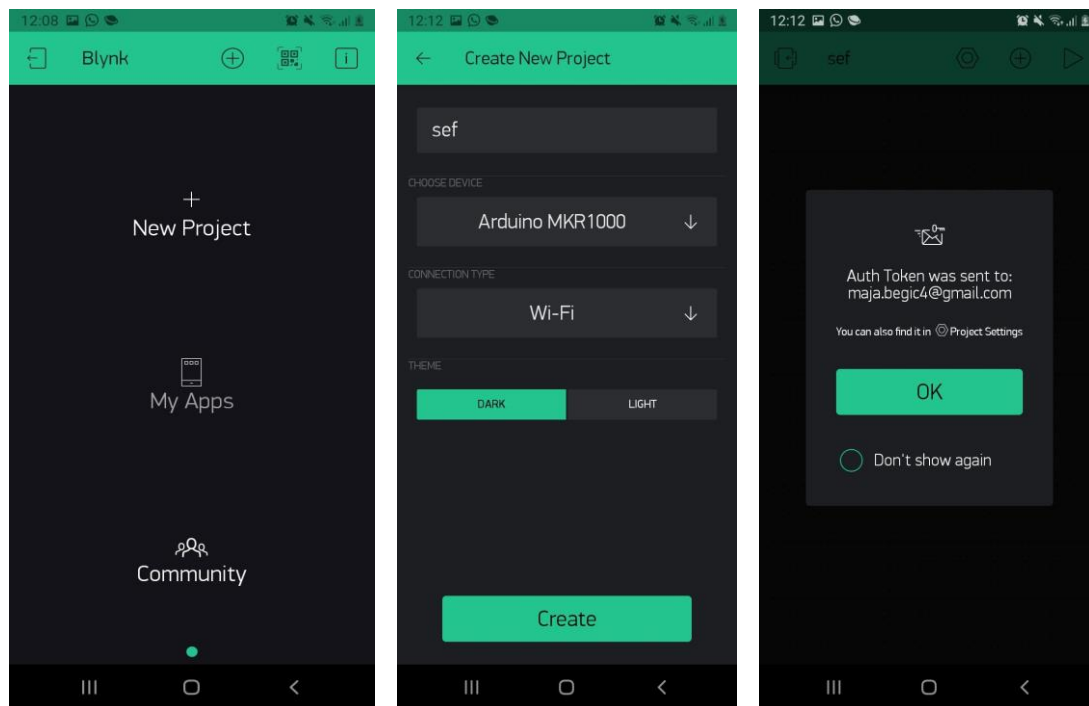
Koraci instalacije i kreiranja projekta prikazani su sljedećim slikama:

- Preuzimanje Blynk aplikacije :



Slika 21. Preuzimanje Blynk aplikacije

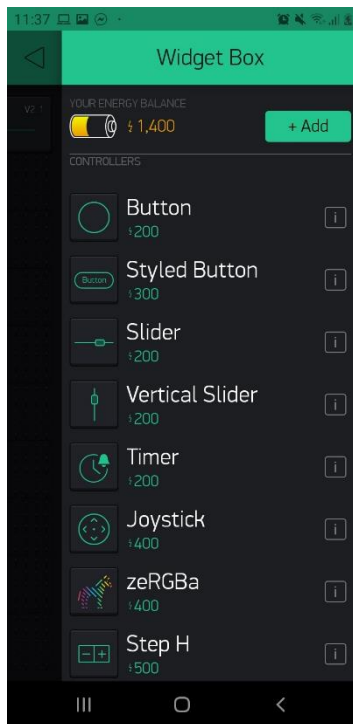
- Izrada korisničkog računa (potreban kako bi se projekti spremali i kako bismo mi imali pristup njima, jedna vrsta zaštite); kreiranje projekta:



Slika 22. Kreiranje projekta za projekt pod nazivom sef

Kad unesemo potrebne informacije, Blynk će nam poslati e-mail poruku s autentifikacijskim tokenom koji je jedinstven za svaki projekt i služi za povezivanje hardvera s Blynk aplikacijom. Kasnije ga iskoristavamo u programskom kodu za Arduino.

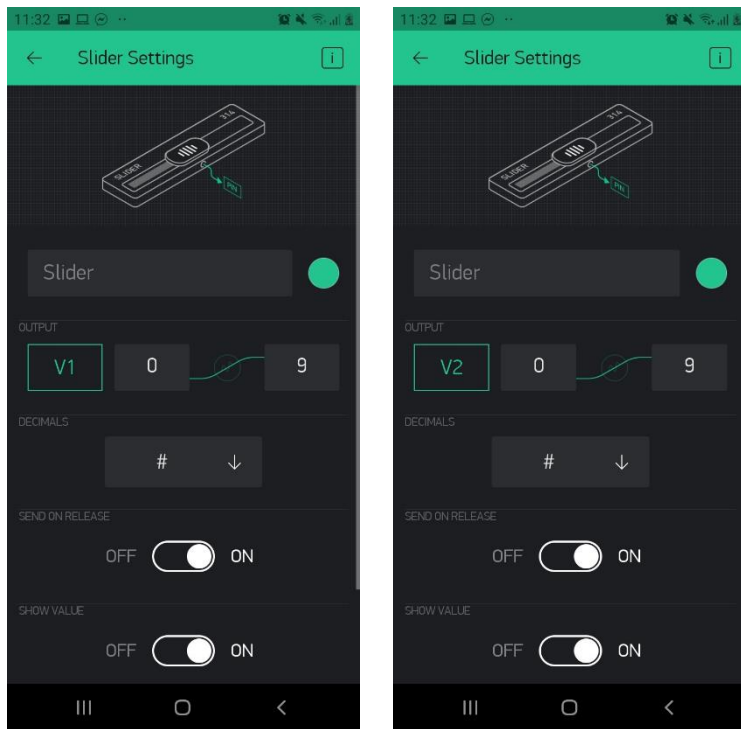
- Dodavanje „widgeta“ i izrada vlastitog projekta:



Slika 23. Dodavanje "widgeta"

S obzirom da ključne kombinacije postavljamo u rasponu brojeva od 0 do 9 odabiremo *slider* koji nam omogućuje postavljanje vrijednosti unutar tog raspona brojeva.

- Namještanje postavki na dodanim „widgetima“:



Slika 24. Postavljanje postavki na dodanim "widgetima"

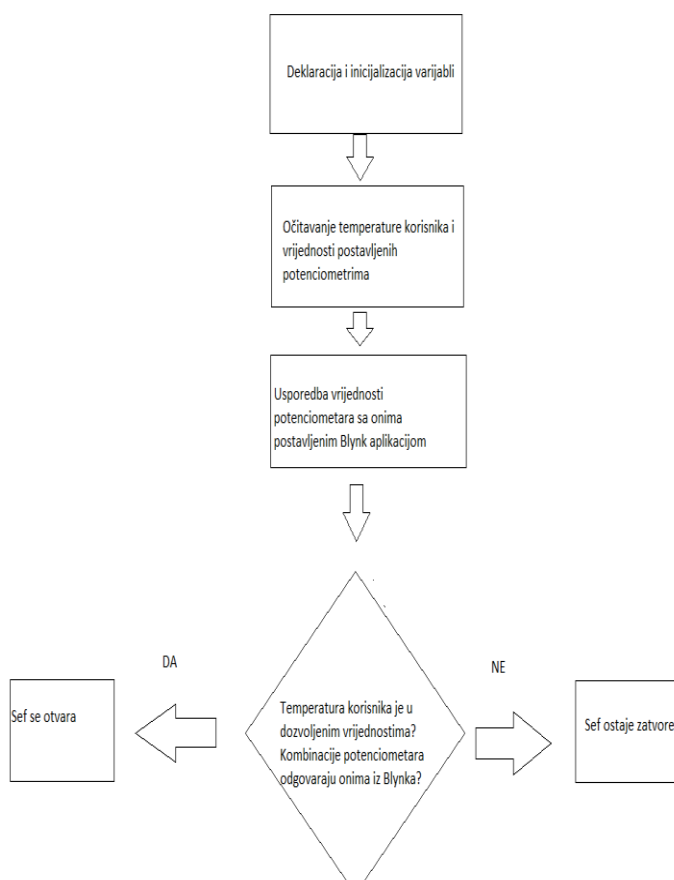
Za svaki od slidera postavljamo izlazne pinove kao virtualne. Odabrali smo virtualne pinove jer nam omogućavaju komunikaciju s bilo kojim senzorom i datotekama. Postavljamo mogući raspon brojeva 0-9.

6. PROGRAMSKI KOD

Programski kod je podijeljen u cjeline :

- Deklaracija i inicijalizacija korištenih varijabli, učitavanje ključnih vrijednosti za otvaranje sefa postavljenih na Blynk aplikaciji
- Očitavanje temperature korisnika, očitavanje i mapiranje vrijednosti kombinacija postavljenih potenciometrom
- Usporedba postavljenih vrijednosti potenciometara i onih učitanih sa Blynk aplikacije
- Ovisno o korisnikovoj temperaturi i postavljenim kombinacijama na potenciometrima sef se otvara

Ovaj dijagram toka objašnjava kako program radi:



```
#include <LiquidCrystal.h>
#include <SPI.h>
```

```
#include <WiFi101.h>
#include <BlynkSimpleWiFiShield101.h>
#include <Servo.h>

//omogucujemo rad s Blynk aplikacijom
const char* ssid = "P2qr4C"; //upis mreze
const char* password = ""; //upis lozinke
char auth[] = "FOIZw0NBLCrCo2LqFLNiMoBsNANDYPyks"; //token, dobijemo na mail
kad izradimo projekt

// deklariranje pinova RGB diode
int LedCrvena = 6; //LedCrvena spojen je na 6. pinu arduina
int LedZelena = 8; //LedZelena spojen je na 8. pinu arduina
int LedPlava = 7; //LedPlava spojen je na 7. pinu arduina

bool zastavica_otvaranje = true;

//deklariranje izvoda LCD zaslona
const int rs = 12,
        en = 11,
        d4 = 2,
        d5 = 3,
        d6 = 4,
        d7 = 5;

//inicijalno pozicija_servo_motora stavljamo da su vrijednosti koje pogađamo
na 9 9 9
int Gumb_prvi = 9;
int Gumb_drugi = 9;
int Gumb_treci = 9;

int pozicija_servo_motora = 0; //varijabla koja sprema poziciju na kojoj
se servo motor trenutno nalazi
Servo myservo; // varijabla myservo upravlja ce servo motorom

// ulazne vrijednosti s Blynk aplikacije
```

```
BLYNK_WRITE(V1) {
    Gumb_prvi = param.asInt(); // očitavamo vrijednost s virtualnog pina V1 na
    Blynk aplikaciji i spremamo u varijablu Gumb_prvi
}
BLYNK_WRITE(V2) {
    Gumb_drugi = param.asInt(); // očitavamo vrijednost s virtualnog pina V2
    na Blynk aplikaciji i spremamo u varijablu Gumb_drugi
}
BLYNK_WRITE(V3) {
    Gumb_treci = param.asInt(); // očitavamo vrijednost s virtualnog pina V3
    na Blynk aplikaciji i spremamo u varijablu Gumb_treci
}

const int temperaturePin = 5; //na pin 5 Arduina spojen je temperaturni senzor
//pomocne varijable za mjerenje temperature:
int staratemperatura = 0;
float temp_pomocni_niz[20];
float temp_pomocni_niz_srednja_vrijednost[20];
int ind = 0;
int avgInd = 0;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {

    pinMode(LedCrvena, OUTPUT); // izvod na koji je spojena LedCrvena
    postavljamo kao izlazni
    pinMode(LedZelena, OUTPUT); // izvod na koji je spojena LedZelena
    postavljamo kao izlazni
    pinMode(LedPlava, OUTPUT); // izvod na koji je spojena LedPlava postavljamo
    kao izlazni

    analogWrite(A3, 0); // stavljamo svjetlinu LCD zaslona na najveću
    vrijednost
    Serial.begin(9600); //zapocinjemo serijsku komunikaciju
    lcd.begin(16, 2); // Pokrecemo rad LCD zaslona(2 redax16stupaca)
    Blynk.begin(auth, ssid, password); // pokrecemo Blynk aplikaciju
```

```
myservo.attach(9); // my servo upravlja pinom 9
myservo.write(pozicija_servo_motora); //inicijalno stavljamo servo motor u
poziciju u kojoj poprima 0 stupnjeva
//Serial.println("...setup...");
//inicijalno postavljamo vrijednosti polja u vrijednost nula:
for (int i = 0; i < 20; i++)
{
    temp_pomocni_niz[i] = 0.00;
}
}
void loop()
{
    bool flag = true;
    //s obzirom da Blynk ne podrzava preveliki delay, a kako bi stabilizirali
    očitavanje temperature sa temperaturnog senzora i
    //kako bismo mogli prikazivati u realnom obliku tu vrijednost, koristimo
    dva niza temp_pomocni_niz i temp_pomocni_niz_srednja_vrijednost koje racunaju
    srednju vrijednost
    //izmjerene trenutne temperature i tek onda prikazuju vrijednost na LCD
    zaslon(inace se na LCD zaslonu prebrzo mijenja prikazivanje
    //trenutne temperature i nemoguće je vidjeti što se prikazuje)
    if (ind >= 20) //kad napunimo prvih 20 očitaka temperature tek onda se
    izvrsava ovaj dio
    {
        ind = 0; //postavljamo indeks ponovo u 0
        int sum = 0.00; //inicijalno postavljamo vrijednost sume na 0
        for (int i = 0; i < 20; i++)
        {
            sum = sum + temp_pomocni_niz[i];
        }
        temp_pomocni_niz_srednja_vrijednost[avgInd] = sum / 20.00; //racunamo
        srednju vrijednost prethodnih 20 očitaka temperature i punimo niz
        avgInd++; //povecavamo indeks
    }

    if(avgInd >= 20) //kad 20 puta napunimo prethodni niz
    temp_pomocni_niz_srednja_vrijednost
    {
```

```
    avgInd = 0;
    int avgSum = 0;
    for (int i = 0; i < 20; i ++ )
    {
        avgSum = avgSum + temp_pomocni_niz_srednja_vrijednost[i];
    }
    prikazitemperaturu(avgSum / 20.00); //prikazujemo srednju vrijednost
ocitanih temperatura
}
temp_pomocni_niz[ind] = izracunaj_temperaturu_u_stupnjevima(); //vraca nam
ocitanu temperaturu u celzijevim stupnjevima, punimo niz ocitanim
temperaturama
Serial.println("izracunaj_temperaturu_u_stupnjevima");
Serial.println(temp_pomocni_niz[ind]);

for (int i = 0; i < 20; i ++ )
{
    if (temp_pomocni_niz[i] < 22 || temp_pomocni_niz[i] > 37)
    {
        flag = false; //onemogucujemo otvaranje sefa ako je temperatura ispod
22 ili iznad 37 celzijevih stupnjeva
    }
}
// kombinacije s Blynka
Blynk.run(); // omogucujemo ucitavanje kljucne kombinacije postavljene na
Blynk aplikaciji
Serial.print("Kombinacija postavljena na Blynk aplikaciji: ");
Serial.print(Gumb_prvi);
Serial.print(" ");
Serial.print(Gumb_drugi);
Serial.print(" ");
Serial.println(Gumb_treci);
//vrijednosti ocitane sa potencijometara:
int Prvi_potencijometar = map(analogRead(A0), 0, 1023, 0, 9); //ocitavamo
vrijednost s analognog izvoda A0 potencijometra i iznos skaliramo na vrijednost
od 0 do 9
```

```
int Drugi_potenciometar = map(analogRead(A1), 0, 1023, 0, 9); //ocitavamo
vrijednost s analognog pina A1 potencijometra i iznos skaliramo na vrijednost
od 0 do 9

int Treci_potenciometar = map(analogRead(A2), 0, 1023, 0, 9); //ocitavamo
vrijednost s analognog pina A2 potencijometra i iznos skaliramo na vrijednost
od 0 do 9

lcd.setCursor(0, 0); //pozicioniramo kursor na LCD zaslonu na (0,0)
poziciju i na tom mjestu prikazuje se vrijednost očitana s prvog potencijometra

lcd.print(Prvi_potenciometar);

lcd.setCursor(2, 0); //pozicioniramo kursor na LCD zaslonu na (2,0) poziciju
i na tom mjestu prikazuje se vrijednost očitana s drugog potencijometra

lcd.print(Drugii_potenciometar);

lcd.setCursor(4, 0); //pozicioniramo kursor na LCD zaslonu na (4,0) poziciju
i na tom mjestu prikazuje se vrijednost očitana s treceg potencijometra

lcd.print(Treci_potenciometar);

if (zastavica_otvaranje) {
    dioda_svijetli(Prvi_potenciometar,                Drugi_potenciometar,
Treci_potenciometar);

    if ((Prvi_potenciometar == Gumb_prvi && Drugii_potenciometar == Gumb_drugi
&& Treci_potenciometar == Gumb_treci) && flag == true) { //sef se otvara ako
smo potencijometrima pogodili dobitnu kombinaciju i ako je temperatura u
rasponu normalne tjelesne temperature

        otvori_sef();
        zasvijetli_zeleno();
        zastavica_otvaranje = false;
        Serial.print("OPEN: ");
        Serial.println(zastavica_otvaranje);
    }
}

if (!zastavica_otvaranje) {
    if (!(Prvi_potenciometar == Gumb_prvi && Drugii_potenciometar ==
Gumb_drugi && Treci_potenciometar == Gumb_treci)) { //ako promijenimo bilo
koji potencijometar od pozicije prethodne dobitne kombinacije sef se zatvara

        zatvori_sef();
        zastavica_otvaranje = true;
    }
}
```

```
        Serial.print("CLOSE: ");
        Serial.println(zastavica_otvaranje);
    }
}

ind ++;
}

float izracunaj_temperaturu_u_stupnjevima() //funkcija za mjerenje napona i
njegovo pretvaranje u stupnjeve celzijuse
{
    float voltage, degreesC;
    voltage = analogRead(temperaturePin) * 0.004882814;
    degreesC = ((voltage - 0.5) * 100.0) * 0.3392857;
    return degreesC;
}

void prikazitemperaturu(float temp) //funkcija za prikazivanje temperature
{
    Serial.println("prikazitemperaturu");
    Serial.println(temp);
    int decimalPart = (temp - (int)temp) * 100;
    int decimalBig = decimalPart / 10;
    int decimalSmall = decimalPart % 10;
    int intBig = (int)temp / 10;
    int intSmall = (int)temp % 10;

    int n = 1;
    if (intBig != 0)
    {
        n = 0;
        lcd.setCursor(0, 1);
        lcd.print(intBig);
    }
    lcd.setCursor(1 - n, 1);
    lcd.print(intSmall);
}
```



```
lcd.setCursor(2 - n, 1);
lcd.print('.');
//decimal
lcd.setCursor(3 - n, 1);
lcd.print(decimalBig);
lcd.setCursor(4 - n, 1);
lcd.print(decimalSmall);
//celsius
lcd.setCursor(5 - n, 1);
//lcd.print('°');
lcd.setCursor(6 - n, 1);
lcd.print('C');
}
// ovisno koliko smo udaljeni od ispravnog rjesenja, tako ce nam svjetliti
RGB dioda,sto smo blize rjesenju
//boja ce biti toplija
void dioda_svijetli(int Prvi_potenciomtar, int Drugi_potenciomtar, int
Treci_potenciomtar) {

    if (abs(Prvi_potenciomtar - Gumb_prvi) <= 1 && abs(Drug_i_potenciomtar -
Gumb_drugi) <= 1 && abs(Treci_potenciomtar - Gumb_treci) <= 1 ) {
        // Crvena
        postaviBojudiode(255, 0, 0);
    }
    else if (abs(Prvi_potenciomtar - Gumb_prvi) <= 2 &&
abs(Drug_i_potenciomtar - Gumb_drugi) <= 2 && abs(Treci_potenciomtar -
Gumb_treci) <= 2 ) {
        // Zuta
        postaviBojudiode(255, 255, 0);
    }
    else if (abs(Prvi_potenciomtar - Gumb_prvi) <= 3 &&
abs(Drug_i_potenciomtar - Gumb_drugi) <= 3 && abs(Treci_potenciomtar -
Gumb_treci) <= 3 ) {
        // Boja maline
        postaviBojudiode(255, 255, 125);
    }
}
```

```
    else          if  (abs(Prvi_potenciometar  -  Gumb_prvi)  <=  4  &&
abs(Drugi_potenciometar  -  Gumb_drugi)  <=  4  &&  abs(Treci_potenciometar  -
Gumb_treci) <= 4 ) {
        //Boja aqua (svjetlo plava)
        postaviBojudiode(0, 255, 255);
    }
    else {
        //Plava boja
        postaviBojudiode(0, 0, 255);
    }
}

void zasvijetli_zeleno() {
    //kad pogodimo kombinaciju, RGB dioda zasvijetli zeleno:
        postaviBojudiode(0, 255, 0);
        postaviBojudiode(0, 0, 0);
        postaviBojudiode(0, 0, 0);
        postaviBojudiode(0, 0, 0);
        postaviBojudiode(0, 0, 0);
        postaviBojudiode(0, 255, 0);
        postaviBojudiode(0, 0, 0);
        postaviBojudiode(0, 0, 0);
        postaviBojudiode(0, 0, 0);
        postaviBojudiode(0, 0, 0);
        postaviBojudiode(0, 255, 0);
    }

//pozicija_servo_motoratavljamo odgovarajuće boje na RGB ledicu
void postaviBojudiode(int red, int green, int blue) {
    analogWrite(LedCrvena, red);
    analogWrite(LedZelena, green);
    analogWrite(LedPlava, blue);
}

void otvori_sef() {
```

```
Serial.println("OTVORI");

for (pozicija_servo_motora = 0; pozicija_servo_motora <= 90;
    pozicija_servo_motora += 1) { // kreće od pozicije označene s 0 stupnjeva na
    poziciju označenu s 90 stupnjeva
    myservo.write(pozicija_servo_motora); // svaki pomak moramo
    signalizirati
    delay(15); // pricekamo 15 ms da servo motor dosegne svaku od tih
    vrijednosti
    }
}

void zatvori_sef() {
    Serial.println("ZATVORI");
    for (pozicija_servo_motora = 90; pozicija_servo_motora >= 0;
        pozicija_servo_motora -= 1) { // kreće od pozicije označene s 90 stupnjeva
        na poziciju označenu s 0 stupnjeva
        myservo.write(pozicija_servo_motora); // svaki pomak moramo
        signalizirati
        delay(15); // pricekamo 15 ms da servo motor dosegne svaku od tih
        vrijednosti
        }
    }
```

7. Zaključak

Kroz ovaj projektni zadatak dane su i razvijene upute za izradu jednostavnog IoT sefa. Zadatak je osmišljen tako da demonstrira uporabu Arduina i mobilne aplikacije Blynk u Internet of Things području.

U ovom dokumentu navedene su korištene komponente i opisani načini spajanja. Objašnjena je uporaba Blynk mobilne aplikacije koja omogućava jednostavno upravljanje IoT projektom bez dodatnog hardverskog povezivanja. U ovom zadatku, putem Blynk aplikacije je postavljena odgovarajuća lozinka, a namještanjem 3 potenciometra na sefu na odgovarajuće vrijednosti (te iste zadane u Blynk aplikaciji), korisnik ispravnim unosom pokreće servo motor i vratašca sefa se otvaraju. Dodatak projektu je i mjerenje tjelesne temperature temperaturnim senzorom i uvjetno otvaranje vrata sefa, samo ako je temperatura u zadanim granicama normalne tjelesne temperature. Također, prilikom pogađanja kombinacija, iskorištena je vizualna signalizacija. Boje svjetleće diode su toplije ukoliko je korisnik bliže dobitnoj kombinaciji za otvaranje sefa.

Ovaj projekt izrađen je u svrhu popularizacije STEM područja među djecom i mladima. Iako namijenjene prvenstveno djeci i mladima, ove upute mogu biti poslužiti i profesorima i roditeljima zainteresiranim za rad s djecom u ovom zanimljivom i sveobuhvatnom području.

Projekti iz STEM područja potiču razmišljanje, kreativnost i inovativnost mladih i tako podižu svijest o važnosti ostvarivanja digitalne pismenosti.

U budućnosti je moguće izraditi i osmisлити nove zadatke, kao i unaprijediti i nadograditi ovaj postojeći projekt.

8. Literatura

- [1] Institut za razvoj i inovativnost mladih. URL: <https://croatianmakers.hr/hr/projektiirim-a/>
- [2] Digitalni portal Izradi!. URL: <https://izradi.croatianmakers.hr/>
- [3] Uvod u IoT. URL: <https://izradi.croatianmakers.hr/project/uvod-u-iot/>
- [4] Instalacija Blynk knjižnice za Arduino. URL: <https://izradi.croatianmakers.hr/project/instalacija-blynk-knjiznice-za-arduino/>
- [5] Spajanje LCD zaslona. URL: <https://izradi.croatianmakers.hr/project/ispis-stalnih-poruka/>
- [6] Spajanje RGB diode. URL: <https://izradi.croatianmakers.hr/project/promjena-boja/>
- [7] Spajanje servo-motora. URL: <https://izradi.croatianmakers.hr/project/spajanje-jednog-servo-motora/>
- [8] Puzzle Box. URL: <https://create.arduino.cc/projecthub/arduino/puzzlebox-c1f374>
- [9] Ispis temperature na serijski port. URL: <https://izradi.croatianmakers.hr/project/ispis-temperature-na-serijski-port/>