Maja Begić 0036497846

Emilija Čuljak 0036496144

SEMINARSKI RAD -

DKU-IRIM

zadaci za učenike

logoCCcrativeCC-logoSažetak

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

30. svibnja 2020

cc1cc3cc4cc2

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

Ovaj seminarski rad je izrađen u okviru predmeta „“ 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 3](#_Toc41676355)

[2. KORIŠTENE KOMPONENTE 3](#_Toc41676356)

[2.1 Shema sustava 3](#_Toc41676357)

[2.2 Arduino MKR1000 5](#_Toc41676358)

[2.3 Senzor za temperaturu 6](#_Toc41676359)

[2.4 RGB dioda 7](#_Toc41676360)

[2.5 Servo-motor 8](#_Toc41676361)

[2.6 Alfanumerički LCD zaslon 9](#_Toc41676362)

[2.7 Potenciometri (10 kOhm) 10](#_Toc41676371)

[3. DIZAJN I IZRADA SUSTAVA 11](#_Toc41676372)

[4. UPUTE ZA IZRADU BLYNK APLIKACIJE 16](#_Toc41676373)

[5. PROGRAMSKI KOD 20](#_Toc41676374)

[6. Zaključak 29](#_Toc41676375)

[7. Literatura 29](#_Toc41676376)

# 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.

# Koncept sustava

U ovom se projektu izrađuje kasa/sef za osobne potrebe koji ima posebno svojstvo da ….

Sef se izrađuje od ….

Sef se koristi tako da ….

Posebna svojstva sefa su …

Material koji se može naći besplatno je …

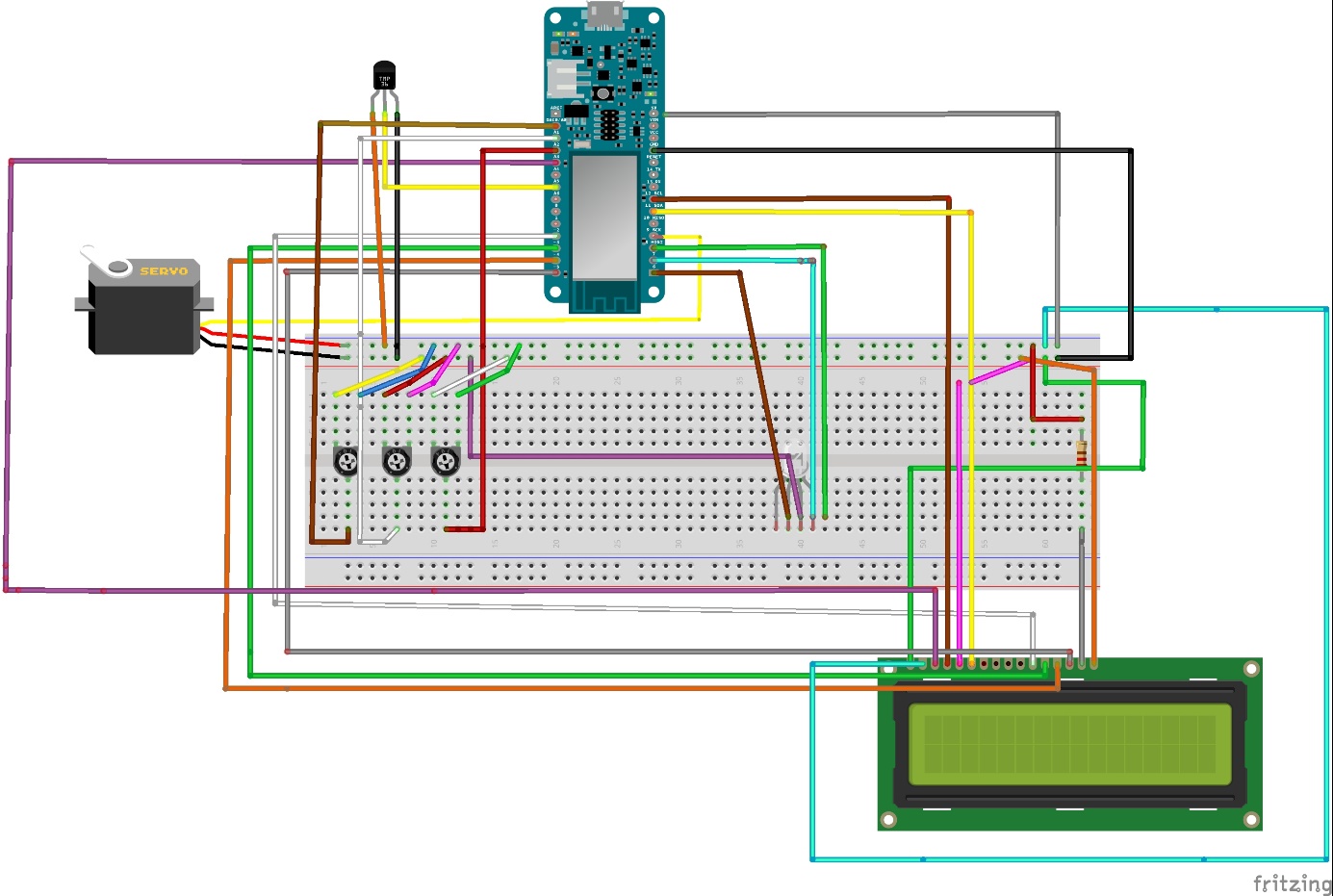
Materijal koji se mora nabaviti nabavlja se ---- gdje – i košta otprilike ….koliko …

# KORIŠTENE KOMPONENTE

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

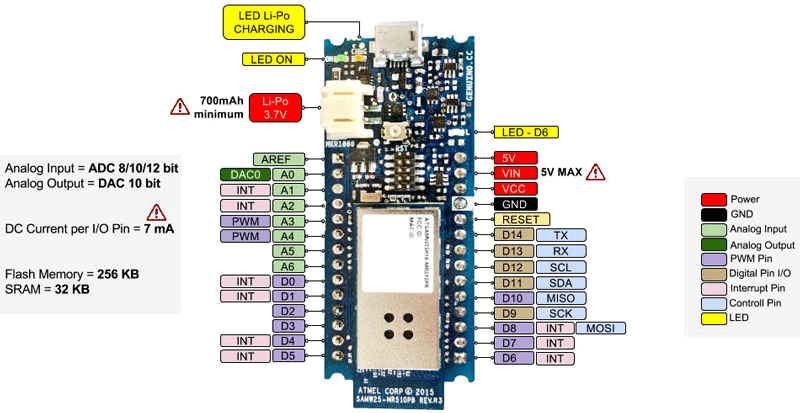
## 2.1 Shema sustava

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



Slika 1. Shema sustava

## 2.2 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.

## Senzor za temperaturu

Svrha senzora za temperaturu u projektu je mjerenje temperature korisnika. Ako je temperatura korisnika u rasponu normalne tjelesne temperature omogućuje korisniku otvaranje sefa postavljanjem dobitne kombinacije na potenciometrima.

U projektu se koristi temperaturni senzor XYZ. On pretvara temperaturu u …. Temepraturni raspon od… do.. pretvara se u … od – do s rezolucijom …

Slika 1**: Temperaturni senzor**

Senzor na mikrokotroler spajamo …, a u programu kontroliramo funkcijom ….

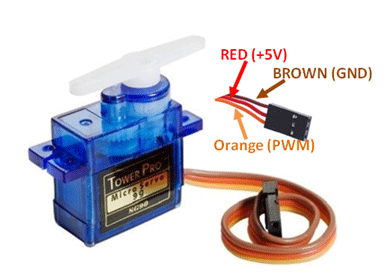
## RGB dioda

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.

Slika 2: RGB dioda

Boja diode se odabire postavljanjem u … stanje odgovarajućeg izvoda diode. Diodine priključke spajamo … a u programo kotroliramo funkcijom …

## Servo-motor

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

Slika 3: Servo-motor

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 slici 6 i slici 7.



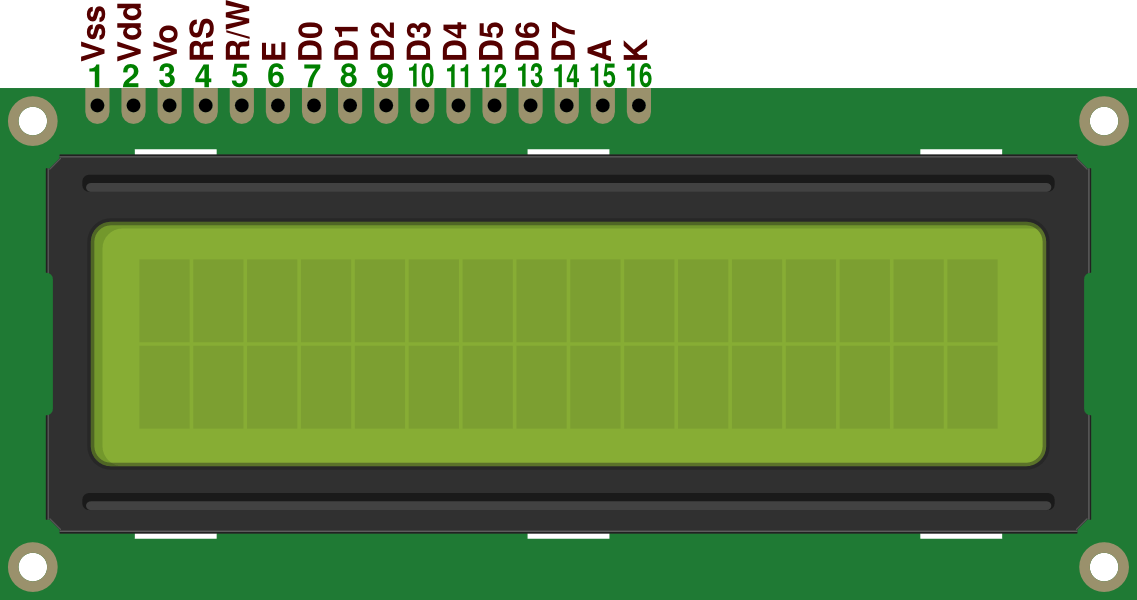
Slika 6. Širina PWM signala (1)



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.

## Alfanumerički LCD zaslon

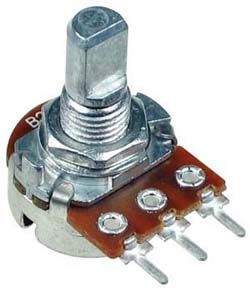
LCD zaslon vrlo je popularna komponenta i naširoko se koristi u raznim elektroničkim projektima. Dobri su za prikaz podataka s raznih senzora.

Slika 4: LCD zaslon

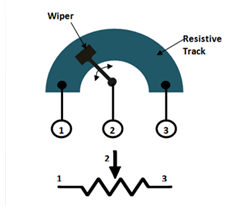
Rad s LCD zaslonom Arduinu omogućujemo učitavanjem LiquidCrystal biblioteke(#include<LiquidCrystal.h>).

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

## 2.7 Potenciometri (10 kOhm)



Slika 9. Potenciometar



Slika 10. Mehanizam rada potenciometra

U projektu potenciometre 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

Potenciometri 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 potenciometra 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 potenciometra. 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).

# 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:

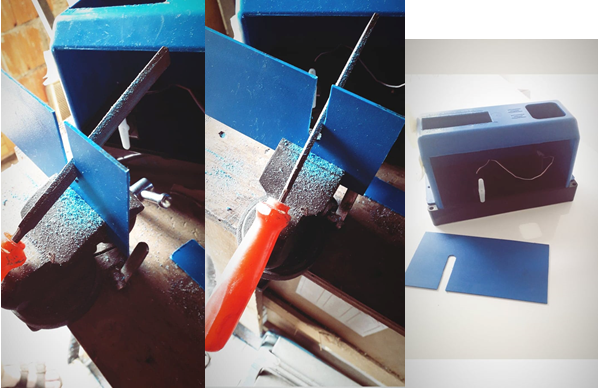
Pilu i skalpel koristili smo za rezanje vrata za sef i za rezanje otvora namijenjenog servo motoru. Pištolj za vruće lijepljenje koristili smo za učvršćivanje komponenti na sef dok smo ručnom bušilicom bušili rupe za RGB diodu, potenciometre i senzor za mjerenje temperature. Na kraju smo lemilicom zalemili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

Prvi korak: Koristimo plastičnu kutiju (Slika 14.) (u našem primjeru koristili smo plastičnu kutiju pumpe za vodu).



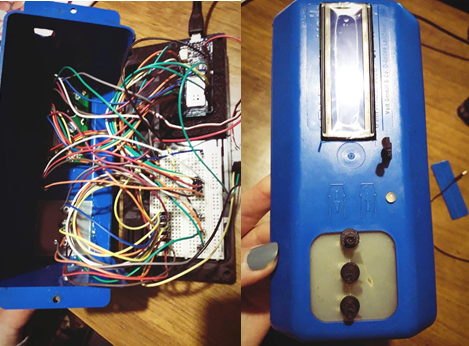
Slika 14.Plastična kutija korištena pri izradi sustava

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).



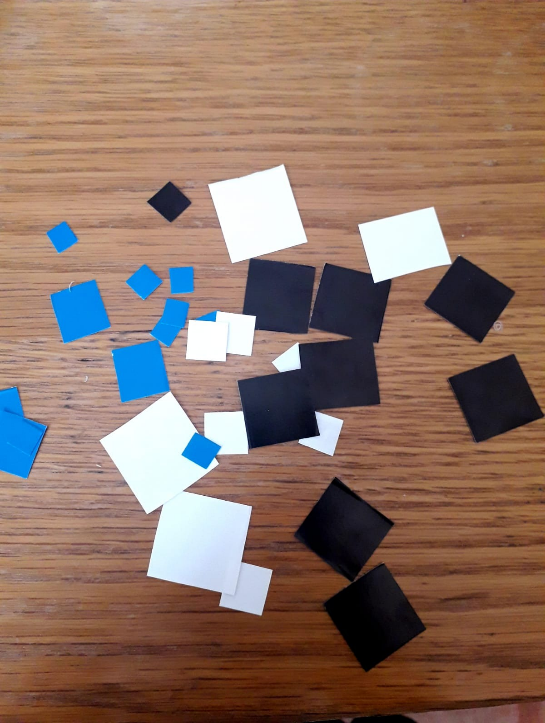
Slika 15. Drugi korak izrade kućišta za IoT sef

Treći korak: Pištoljem za lijepljenje pričvršćujemo komponente na vanjsku stranu sefa (Slika 18).



Slika 16.Treći korak izrade kućišta za IoT sef

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



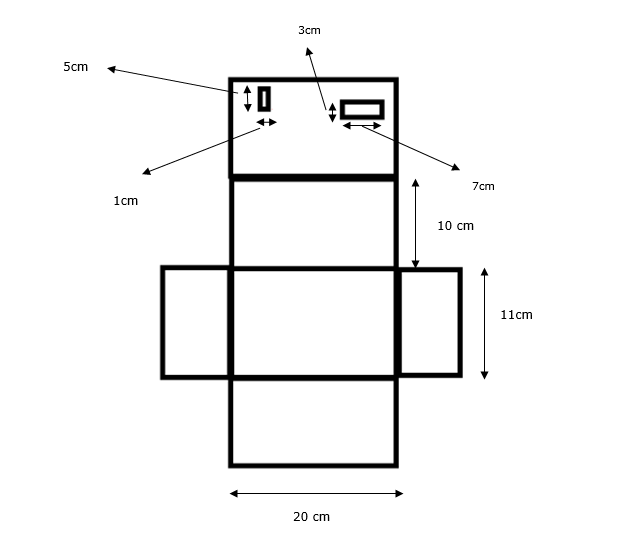
Slika 17.Kvadratići od samoljepljivog kolaža

Peti korak: Vanjski dizajn sustava i dovršavanje kućišta.



Slika 18.Vanjski dizajn sustava

Pri izradi vanjskog kućišta od kartona može se koristiti nacrt:

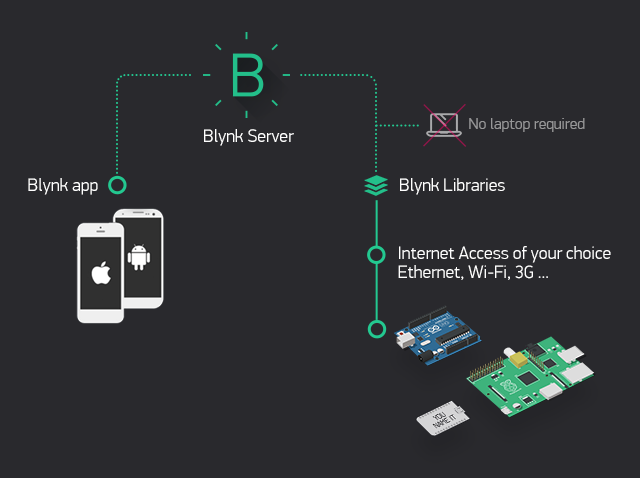


Slika 19. Nacrt za izradu vanjskog kućišta od kartona

# 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

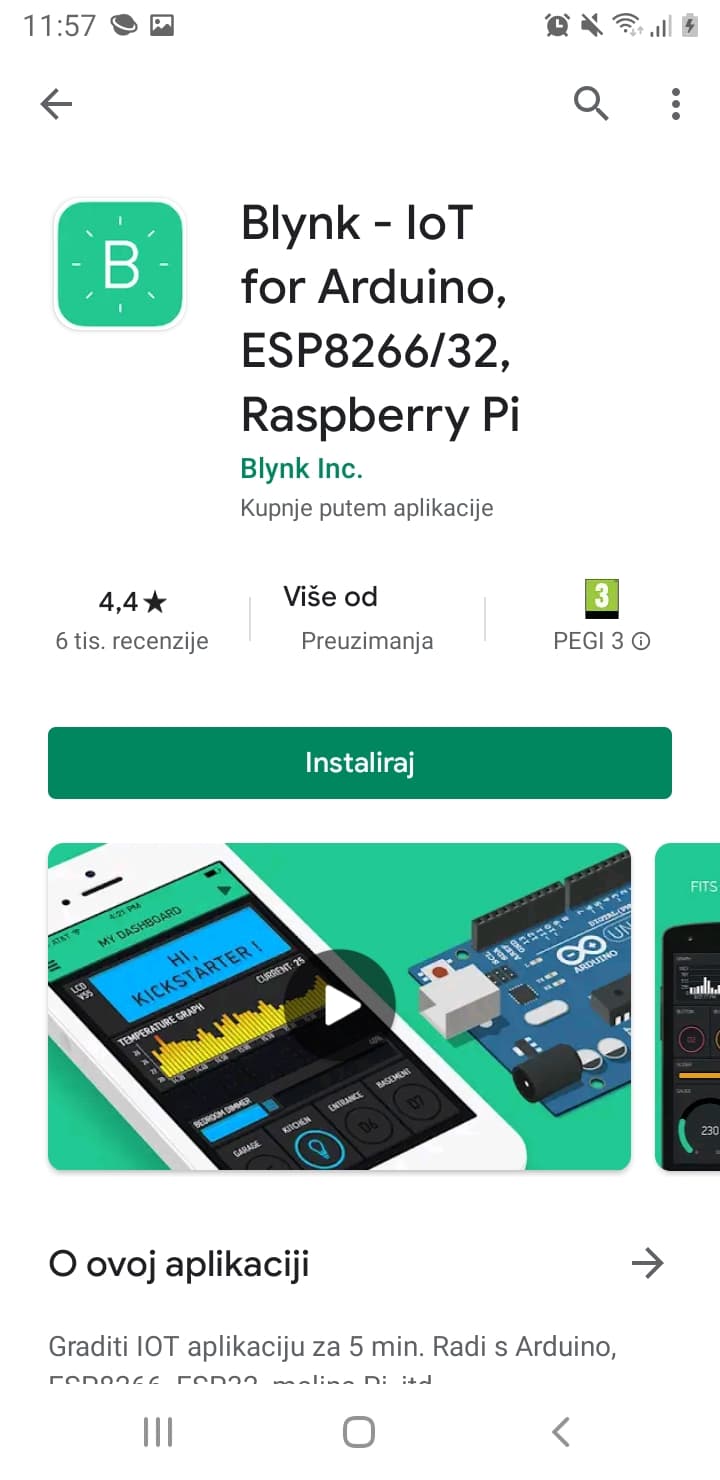
Izradom sučelja u Blynk aplikaciji putem Blynkovog cloud sustava omogućuje se izravna komunikacija aplikacije i hardvera (u ovom slučaju Arduina).

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.

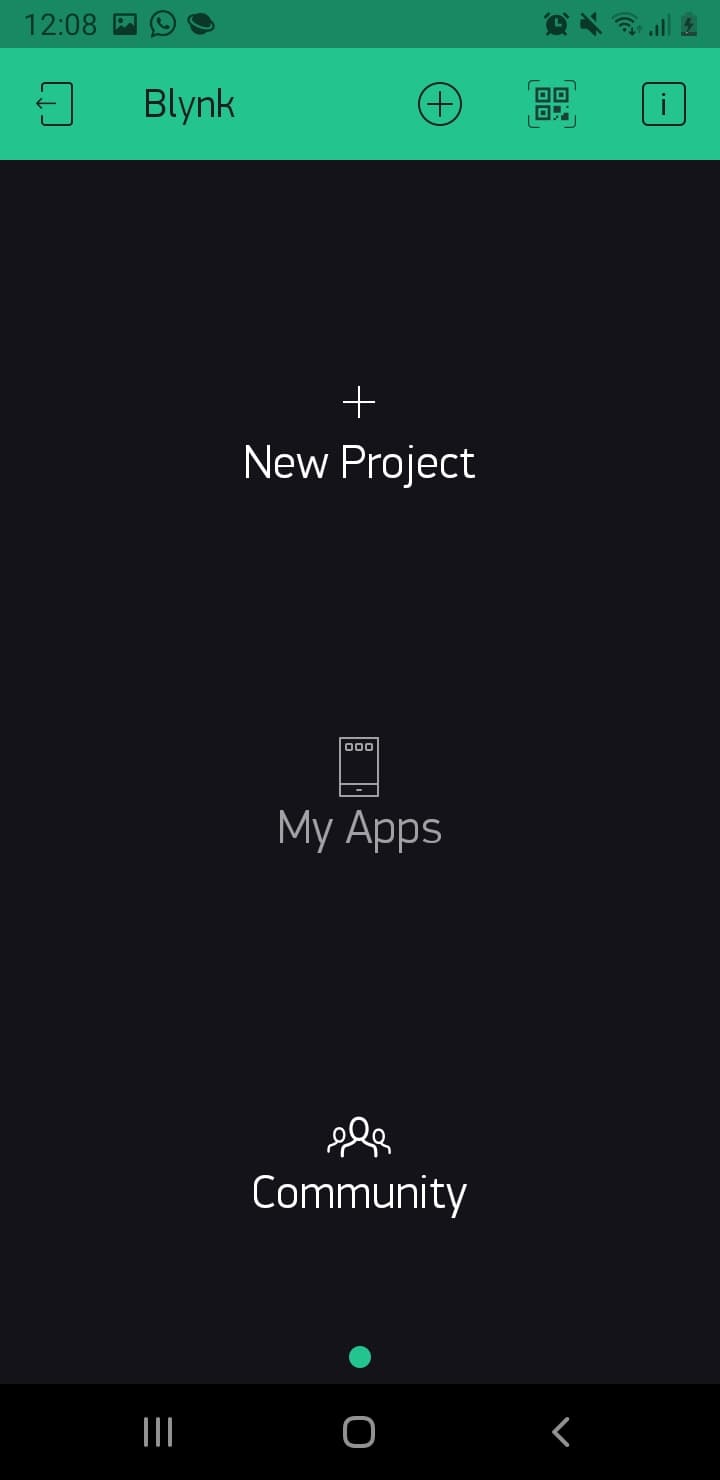
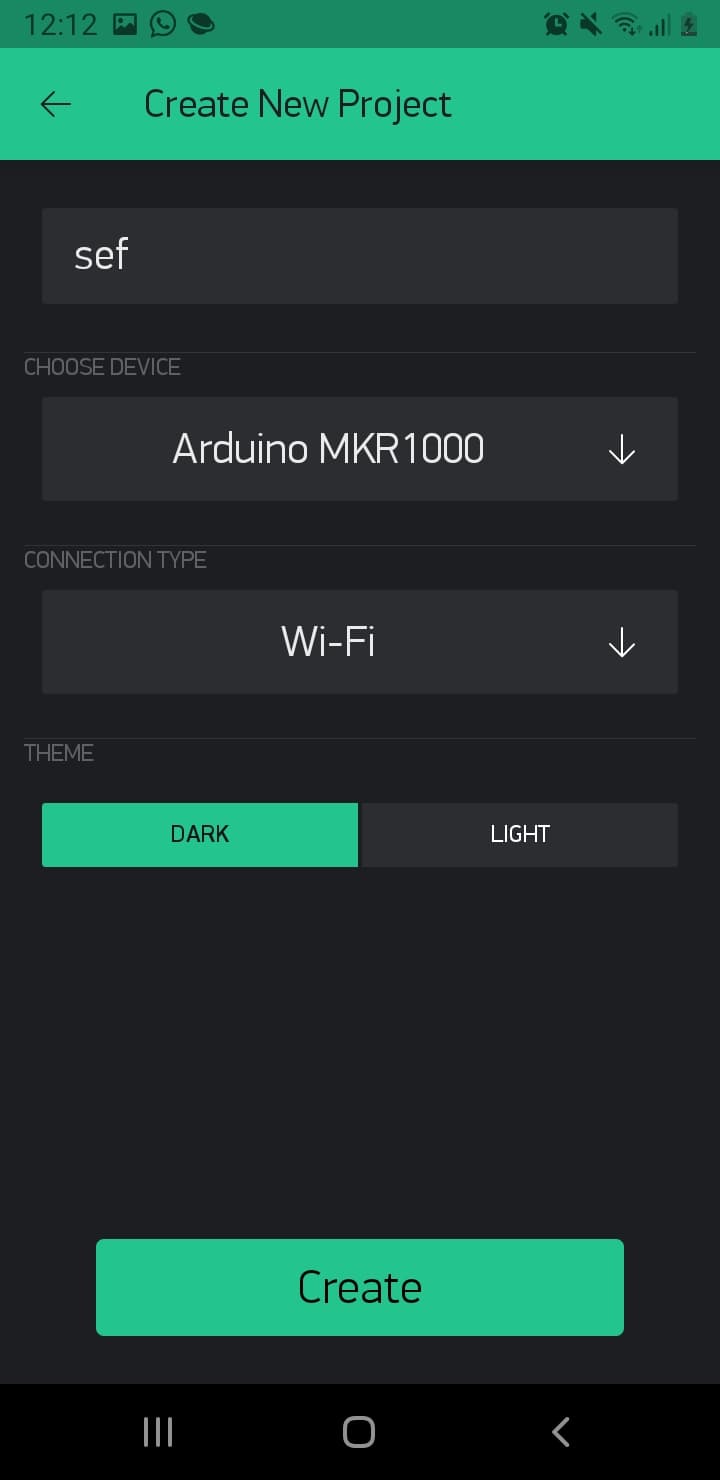
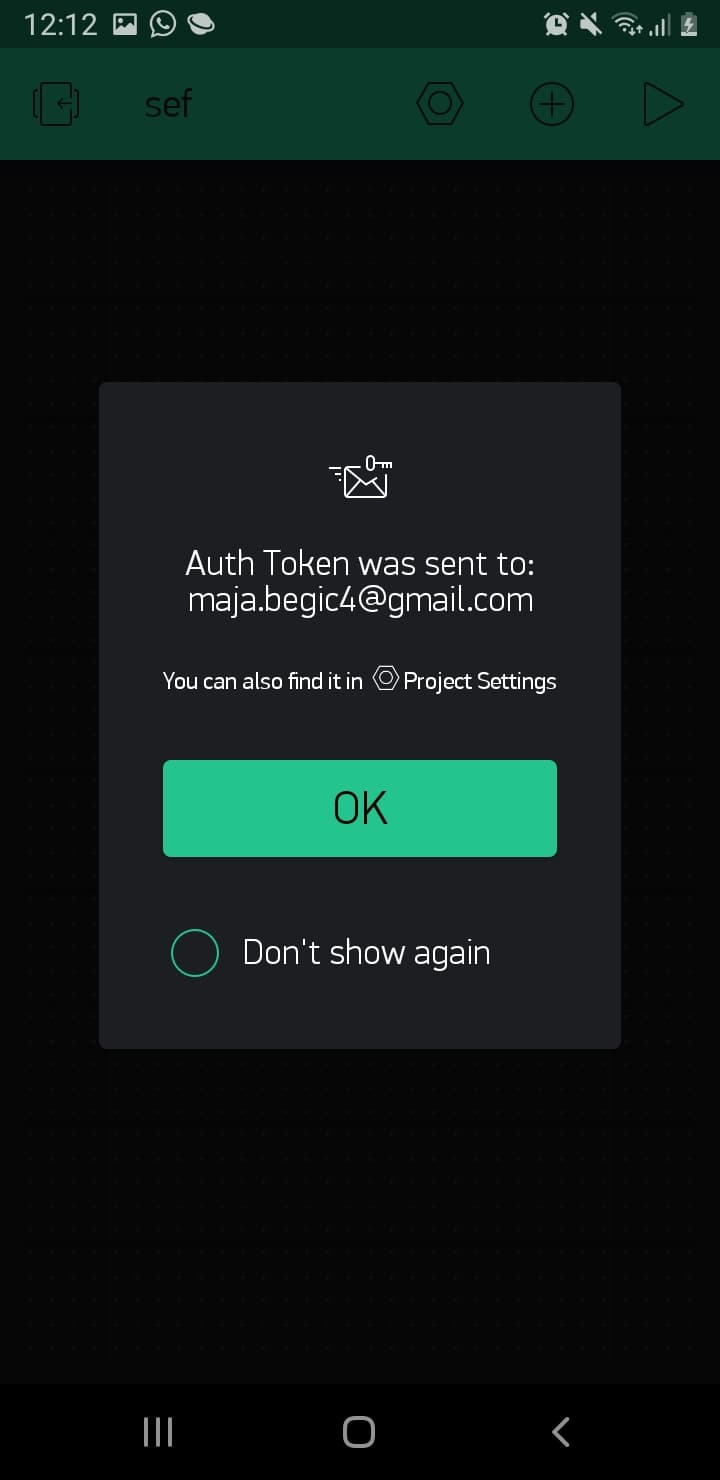
Koraci instalacije i kreiranja projekta prikazani su sljedećim slikama:

* Preuzimanje Blynk aplikacije :



Slika 21. Preuzimanje Blynk aplikacije

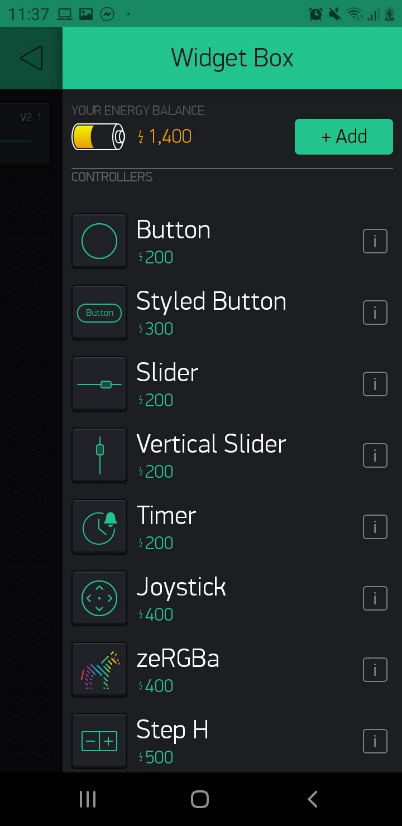
* Izrada korisničkog računa (potreban kako bi se projekti spremali i kako bismo mi imali pristup njima, jedna vrsta zaštite); kreiranje projekta:

Slika 22. Kreiranje projekta za projekt pod nazivom XYZ

Kad unesemo potrebne informacije, Bylnk će nam poslati e-mail poruku putem koje ćemo ….

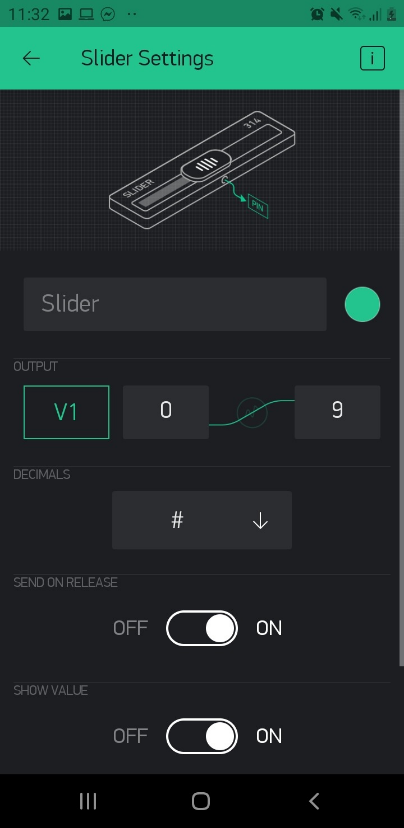
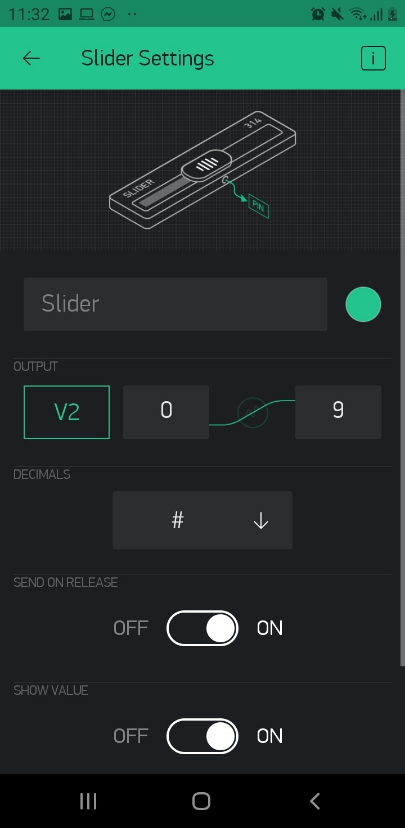
* Dodavanje „widgeta“ i izrada vlastitog projekta:



Slika 23. Dodavanje "widgeta"

Objasnite što i zašto radite

* Namještanje postavki na dodanim „widgetima“:

Slika 24. Postavljanje postavki na dodanim "widgetima"

Objasnite koje postavke postavljate, zašto i kako. Može li se drugačije i što bi se onda dogodilo

# PROGRAMSKI KOD

Prgamski kod je podijeljen u cjeline :

* xyz
* abc
* qwe

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[] = "FOIZw0NBLRCo2LqFLNiMoBsNANDYPyks"; //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 upravljat ce servo motorom

// ulazne vrijednosti s Blynk aplikacije

BLYNK\_WRITE(V1) {

Gumb\_prvi = param.asInt(); // ocitavamo vrijednost s virtualnog pina V1 na Blynk aplikaciji i spremamo u varijablu Gumb\_prvi

}

BLYNK\_WRITE(V2) {

Gumb\_drugi = param.asInt(); // ocitavamo vrijednost s virtualnog pina V2 na Blynk aplikaciji i spremamo u varijablu Gumb\_drugi

}

BLYNK\_WRITE(V3) {

Gumb\_treci = param.asInt(); // ocitavamo 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 najvecu 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 ocitavanje 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 nemoguce je vidjeti sto se prikazuje)

if (ind >= 20) //kad napunimo prvih 20 ocitaka 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 ocitaka 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 potenciometara:

int Prvi\_potenciometar = map(analogRead(A0), 0, 1023, 0, 9); //ocitavamo vrijednost s analognog izvoda A0 potenciometra 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 potenciometra 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 potenciometra 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 ocitana s prvog potenciometra

lcd.print(Prvi\_potenciometar);

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

lcd.print(Drugi\_potenciometar);

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

lcd.print(Treci\_potenciometar);

if (zastavica\_otvaranje) {

dioda\_svijetli(Prvi\_potenciometar, Drugi\_potenciometar, Treci\_potenciometar);

if ((Prvi\_potenciometar == Gumb\_prvi && Drugi\_potenciometar == Gumb\_drugi && Treci\_potenciometar == Gumb\_treci) && flag == true) { //sef se otvara ako smo potenciometrima 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 && Drugi\_potenciometar == Gumb\_drugi && Treci\_potenciometar == Gumb\_treci)) { //ako promijenimo bilo koji potenciometar 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\_potenciometar, int Drugi\_potenciometar, int Treci\_potenciometar) {

if (abs(Prvi\_potenciometar - Gumb\_prvi) <= 1 && abs(Drugi\_potenciometar - Gumb\_drugi) <= 1 && abs(Treci\_potenciometar - Gumb\_treci) <= 1 ) {

// Crvena

postaviBojudiode(255, 0, 0);

}

else if (abs(Prvi\_potenciometar - Gumb\_prvi) <= 2 && abs(Drugi\_potenciometar - Gumb\_drugi) <= 2 && abs(Treci\_potenciometar - Gumb\_treci) <= 2 ) {

// Zuta

postaviBojudiode(255, 255, 0);

}

else if (abs(Prvi\_potenciometar - Gumb\_prvi) <= 3 && abs(Drugi\_potenciometar - Gumb\_drugi) <= 3 && abs(Treci\_potenciometar - 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 odgovarajuce 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) { // krece od pozicije oznacene s 0 stupnjeva na poziciju oznacenu 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) { // krece od pozicije oznacene s 90 stupnjeva na poziciju oznacenu 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

}

}

# 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 zainteresiranima 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 osmisliti nove zadatke, kao i unaprijediti i nadograditi ovaj postojeći projekt.

# 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. Ispis temperature na serijski port. URL: <https://izradi.croatianmakers.hr/project/ispis-temperature-na-serijski-port/>