Model kućišta za automatizirano prikupljanje uzoraka s digitalnog refraktometra

Osmakčić, Dorijan

Undergraduate thesis / Završni rad

2022

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:200:239560

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-16**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

MODEL KUČIŠTA ZA AUTOMATIZIRANO PRIKUPLJANJE UZORAKA S DIGITALNOG REFRAKTOMETRA

Završni rad

Dorijan Osmakčić

Osijek,2022.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 02.09.2022.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Dorijan Osmakčić
Studij, smjer:	Preddiplomski sveučilišni studij Elektrotehnika i informacijska
Mat. br. Pristupnika, godina upisa:	4716, 22.07.2019.
OIB Pristupnika:	98998490348
Mentor:	Izv. prof. dr. sc. Ivan Aleksi
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Model kućišta za automatizirano prikupljanje uzoraka s digitalnog refraktometra
Znanstvena grana rada:	**Automatika (zn. polje temeljne tehničke znanosti)**
Zadatak završnog rad:	Temu rezervirao: Dorijan Osmakčić. U ovom radu je potrebno osmisliti model za automatizirano prikupljanje uzoraka vina za digitalni refraktometar Hanna Instruments 96801. Potrebno je osposobiti sustav kamere i osvjetljenja za očitanje stanja na digitalnom pokazniku. Potrebno je napraviti sustav za doziranje vina, odnosno vode za ispiranje pomoću peristaltičkih pumpi.
Prijedlog ocjene završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	02.09.2022.
Datum potvrde ocjene od strane Odbora:	07.09.2022.

Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 07.09.2022.

Ime i prezime studenta:	Dorijan Osmakčić
Studij:	Preddiplomski sveučilišni studij Elektrotehnika i informacijska tehnologija
Mat. br. studenta, godina upisa:	4716, 22.07.2019.
Turnitin podudaranje [%]:	6

Ovom izjavom izjavljujem da je rad pod nazivom**: Model kućišta za automatizirano prikupljanje uzoraka s digitalnog refraktometra**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Ivan Aleksi

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. PRIJENOSNI REFRAKTOMETAR HI96801 2

2.1. Uloga i opis uređaja 2

2.2. Postojeći sustavi za mjerenje šećera 3

2.2.1. Analogni refraktometar KERN ORA 3AA 5

2.2.2. OenoFoss FTIR uređaj 7

3. DIJELOVI SUSTAVA ZA MJERENJE RAZINE ŠEĆERA 8

3.1. ESP32-CAM IoT modul 8

3.2. Firebase baza podataka 11

3.2.1. Povezivanje kamere s Arduinom 11

3.2.2. Izrada Firebase projekta 16

3.2.3. Povezivanje Arduina s Firebase-om 20

3.3. Model uređaja za prikupljanje slike s refraktometra 33

4. IMPLEMENTACIJA I EKSPERIMENTALNI REZULTATI 35

4.1. IoT uređaj za prikupljanje slika s refraktometra 35

4.2. Firebase baza prikupljenih slika 36

5. ZAKLJUČAK 38

LITERATURA 39

SAŽETAK 42

ABSTRACT 42

ŽIVOTOPIS 43

1. UVOD

U današnje vrijeme postoje razni uređaji koji se koriste za mjerenje količine šećera u voću, vinu, sokovima i raznim drugim tekućim uzorcima. Uređaji koji se koriste u takve svrhe su refraktometri. U ovom završnom radu koristit se model HI96801 koji pretvara indeks loma otopine u % brix [1].

Brix je mjera kroz koju se odlučuje o količini šećera iz uzorka. Ovakav uređaj uglavnom koriste vinari za provedbu detaljne analize vina.

Osim što je potrebno mjeriti razinu šećera iz uzoraka, potrebno je i digitalizirati rezultate mjerenja s LCD-a na refraktometru odnosno spremiti rezultate na računalo zbog dohvaćanja slika za daljnju obradu. Dohvaćanje slika s LCD pokaznika obavlja se pomoću modula AI Thinker ESP32-CAM. Modul je baziran na ESP32 mikroupravljaču s OV2640 kamerom. Slike koje se dohvaćaju spremaju se na Firebase platformu u njeno skladište radi boljeg upravljanja podataka. Firebase je povezan s kamerom preko Arduina, programskim kodom koji inicijalizira Firebase, kameru i WiFi na koji se povezuje gdje se pokretanjem kôda spremaju slike u skladište Firebase-a .

U sustavu za prikupljanje slike s kamere koristi se i LED traka za osvjetljenje LCD-a na refraktometru radi dohvaćanja jasnije slike s kamere i razlučivanja znamenki sa slike.

1.1. Zadatak završnog rada

Potrebno je napraviti stalak odnosno model koji povezuje kameru, LED traku i refraktometar u jednu funkcionalnu cjelinu te osposobiti sustav kamere koji dohvaća slike mjerenja s LCD-a. Potrebno je pozicionirati LED traku tako da bude zadovoljavajuće osvjetljenje na LCD-u refraktometra.

2. PRIJENOSNI REFRAKTOMETAR HI96801

U ovom dijelu detaljnije se opisuje već spomenuti model refraktometra, njegova uloga i tehnički podaci. Osim toga spominju se i metode mjerenja razine šećera kao i uređaji koji se koriste za mjerenje. Uspoređuje se i HI96801 s nekim poznatijim refraktometrima.

2.1. Uloga i opis uređaja

Uloga HI96801 refraktometra, prikazanog na slici 2.1 jako je bitna, jer na osnovu njegovih dimenzija, načina rada, oblika i drugih jednako bitnih stvari ovise svi koraci izrade modela.

HI96801 idealan je za analizu voća, energetskih napitaka, pudinga, sojinog mlijeka, sokova, džema, marmelade, meda, juha, želea, tofua i začina. Spomenuti refraktometar je jednostavan uređaj za mjerenje šećera u uzorcima hrane na terenu ili u laboratoriju [1].

Iz navedene slike se može vidjeti s lijeve strane LCD koji pokazuje rezultate mjerenja, dok se s desne strane može vidjeti krug na koji se stavlja uzorak. Nakon što se destiliranom vodom očisti dio gdje ide uzorak, stavlja se opet destilirana voda te se zatim koristi tipka ZERO zbog uspoređivanja uzorka bez šećera sa sljedećim koji u sebi ima šećera.

Sl.2.1.Prijenosni refraktometar HI96801 [1].

Kod ovog uređaja bitno je naglasiti određene značajke koje su prikazane u tablici 2.1.

Tablica 2.1.Tehničke specifikacije digitalnog refraktometra HANNA HI96801 [1][2].

Raspon sadržaja šećera	**0 do 85% brix**
Razlučivost sadržaja šećera	0.1% brix
Točnost sadržaja šećera	±0.2% brix
Raspon temperature	0 do 80°C
Temperaturna kompenzacija	automatski između 10 i 40°C
Automatsko isključivanje	nakon 3 minute neaktivnosti
Vrsta baterije/životni vijek	9V/približno 5000 očitanja
Izvor svijetlosti refraktometra	žuti LED
Kalibracija	destiliranom vodom
Veličina uzorka	100µl
LCD	Dvorazinski
Kučište	Vodootporno

2.2. Postojeći sustavi za mjerenje šećera

Kvaliteta voća,povrća i nekih sokova jako je bitna stavka za ljudsko zdravlje. Jedna od bitnih stvari koje određuju kvalitetu hrane i pića je sadržaj šećera koji se nalazi u njima. Sadržaj šećera se može

odrediti raznim analitičkim metodama. Neke od njih uključuju korištenje refraktometra, hidrometra i dr... [3].

Mjerenje šećera u otopini se vrši u više skala. Za vinogradare su najbitniji stupnjevi Brixa i Oeshleovi stupnjevi. Stupnjevi Brixa (simbol °Bx) je sadržaj šećera u vodenoj otopini. Jedan stupanj Brixa je 1 gram saharoze u 100 grama otopine i predstavlja jačinu otopine kao postotak mase. Ako otopina sadrži otopljene krute tvari osim čiste saharoze, tada °Bx samo približno odgovara sadržaju otopljenih krutih tvari [4].

Oeshleova skala (simbol °Oe) je skala koja se koristi za mjerenje gustoće mošta u grožđu. Postoje dvije definicije Oechslea. To su "stara" i "nova". Stara se temelji na specifičnoj težini, dok se nova definicija temelji se na indeksu loma [5].

Postoji jako puno uređaja koji mjere šećer. Tu su razne vrste digitalnih refraktometra, kao i analognih. Za primjer usporedbe, u tablici 2.1 su prikazane neke bitne stavke određenog tipa analognog i digitalnog refraktometra , te FTIR uređaja.

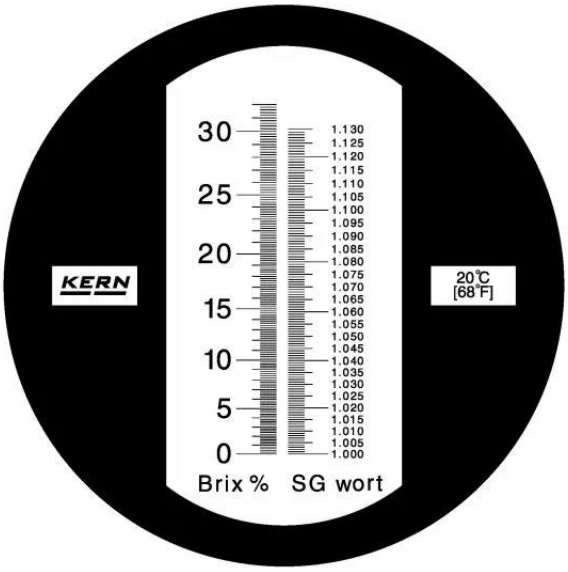
Tablica 2.2. Usporedba uređaja za mjerenje šećera u otopini [2][6-8].

	Analogni refraktometar KERN ORA 3AA	Digitalni HI96801 refraktometar	OenoFoss FTIR uređaj
CIJENA UREĐAJA	800-900 kn	1.400,00-1.500,00 kn	150.000,00-250.000,00kn
BRZINA OČITANJA	30 sekundi	1,5 sekundi	2 min
DIMENZIJE UREĐAJA	178 x 40 x 40 mm	192 x 102 x 67 mm	189 × 154 × 321 mm
TEŽINA UREĐAJA	0,145kg	0,420kg	Vinska jedinica:6,3 kg. Jedinica boje: 3,8 kg
RASPON MJERENJA (BRIX)	0-32%	0-85%	12-27%
NAČIN PRIKAZA IZMJERENE VRIJEDNOSTI	Brix skala	LCD pokaznik	Zaslon računala

2.2.1. Analogni refraktometar KERN ORA 3AA

KERN ORA 3AA, prikazan na slici 2.3 je analogni prijenosni refraktometar manjih dimenzija koji je jednostavan za upotrebu. Uređaj je ručni tip (engl.*handheld)*. Koristi se za promatranje ponašanja svjetlosti dok prolazi od prizme poznatih svojstava do objekta koji se ispituje.

Uzorak se stavlja na jedan kraj uređaja, nakon čega se zatvara prozirnim poklopcem. Rezultati se nakon par sekundi mogu vidjeti na skali prikazanoj na slici 2.2 koja služi za očitavanje brixa pri mjerenju.



Sl. 2.2. Skala KERN ORA 3AA refraktometra [7].

Sl. 2.3. Prikaz analognog refraktometra [6].

2.2.2. OenoFoss FTIR uređaj

Ova vrsta uređaja prikazanog na slici 2.4, daje rezultate za čak više od 10 parametara uz korištenje računala, zbog čega je naravno i cijena uređaja velika. Vrste uzoraka koje se mogu koristiti za analizu su mošt, mošt u vrenju, slatko vino, gotovo vino itd.. Uzorak se stavlja u bušotinu, nakon čega se zatvara poklopcem. Pritiskom gumba narančaste boje, nakon par minuta rezultati se prikazuju na računalu na koje je uređaj spojen [8].

Sl. 2.4. OenoFoss FTIR uređaj [9].

3. DIJELOVI SUSTAVA ZA MJERENJE RAZINE ŠEĆERA

Osim refraktometra koji mjeri rezultate, za prikupljanje rezultata koriste se još dvije komponente. Te dvije komponente su ESP32-CAM IoT modul, prikazan na slici 3.1, koji dohvaća slike s refraktometra i model koji objedinjuje refraktometar i modul. Slike se dohvaćaju i spremaju na Firebase bazu podataka koja je detaljnije opisana u nastavku.

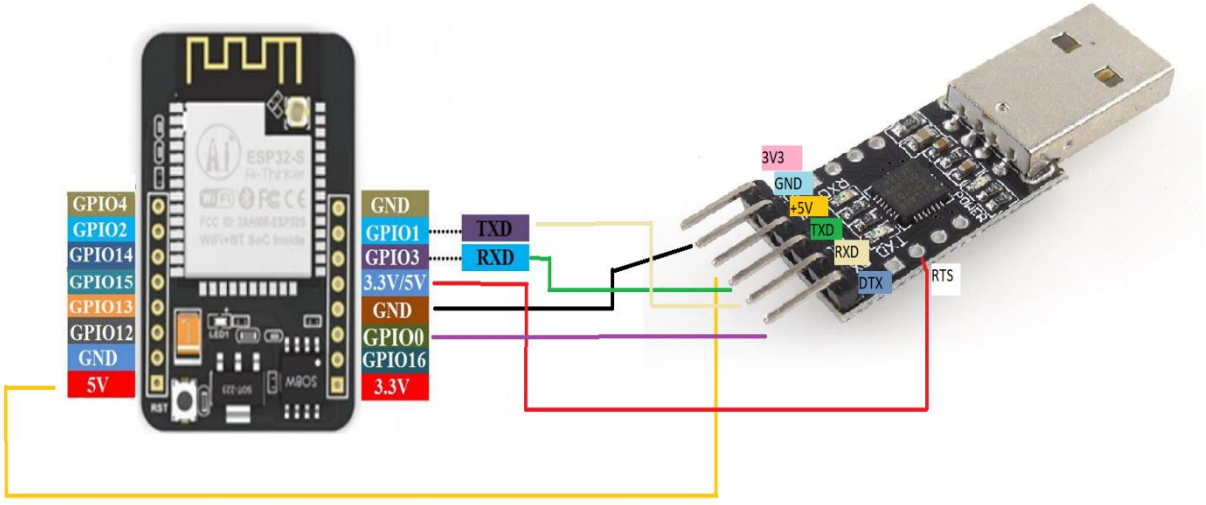
3.1. ESP32-CAM IoT modul

Postoje razni moduli koji sadrže WiFi odnosno kojima se mogu spremati i razmjenjivati podatke putem interneta. Ti moduli uglavnom pripadaju ESP32 seriji. ESP32 je serija jeftinih sustava male potrošnje na mikro-kontrolerima s čipom s integriranim Wi-Fi i dual-mode Bluetoothom [10].

ESP32-CAM je Wi-Fi modul s ugrađenom video kamerom. Kamera pruža razne funkcije, poput nadzora ili špijunaže na daljinu, te hvatanja odnosno snimanje svega što se događa na posjedu vlasnika kada je nije prisutan. Važno je naglasiti da je to modul koji se može koristiti s mnoštvom projekata koji uključuju Arduino [11].

Sl.3.1 ESP32-CAM [11].

Sl. 3.2. Raspored pinova ESP32-CAM [11].



Sl. 3.3. Raspored spajanja žica

U tablici 3.1 se mogu vidjeti tehničke specifikacije modula.

Tablica 3.1.Tehničke specifikacije ESP32-CAM modula [11][12].

Model modula	**ESP32-CAM**
Veličina	27 x 40.5 x 4.5(±0.2)*mm*
Radna memorija	520KB SRAM + 4MB PSRAM + utor za SD Karticu
Bluetooth	Bluetooth 4.2BR/EDR i BLE standarde
Wi-Fi	802.11 b/g/n/e/i
Sučelje	UART, SPI, I2C, PWM
IO port	9
Stopa serijskog porta	115200 bauda
Format izlaza slike	JPEG (podržava samo OV2640), BMP, SIVO
Raspon spektra	2412 ~ 2484 MHz
Oblik antene	ugrađena PCB antena, pojačanje 2dBi
Sigurnost	WPA/WPA2/WPA2-Enterprise/WPS
Raspon napajanja	5V
Radna temperatura	-20℃ ~ 85℃
Kamera	Podržava OV2640 kamere koji se mogu kupiti odvojeno ili u paketu. Kamera sadrži: 2 MP na senzoru Veličina polja UXGA od 1622 × 1200 px Izlazni format YUV422, YUV420, RGB565, RGB555 i 8-bitna kompresija podataka. Može prenijeti sliku između 15 i 60 FPS.

3.2. Firebase baza podataka

Zbog realiziracije spremanja slike u bazu podataka, prvo se povezuje kamera preko računala s Arduinom, zatim se izrađuje Firebase projekt, te se na kraju povezuje Firebase i Arduino.

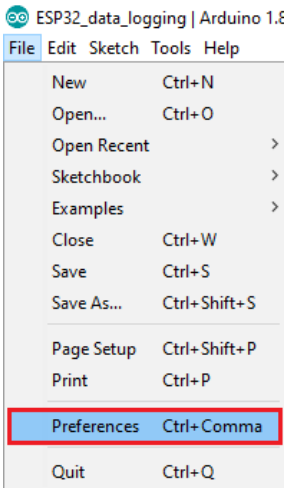
Prije nego što se opiše postupak, važno je znati što je to zapravo Firebase.

Firebase je Googleova platforma koja se koristi za razvoj mobilnih aplikacija. Ona pomaže korisniku izgraditi, poboljšati i razviti svoju aplikaciju. Ima mnogo usluga koje se koriste za upravljanje podacima iz bilo koje Android, IOS ili web aplikacije kao što su autentifikacija, baza podataka u stvarnom vremenu, hosting, pohrana itd.. [13].

3.2.1. Povezivanje kamere s Arduinom

Pri povezivanju kamere odnosno modula ESP32-CAM potrebno je prvo instalirati najnoviju verziju Arduino IDE na računalu. Nakon toga je potrebno instalirati ESP32 ploču u vlastiti Arduino IDE u par koraka:

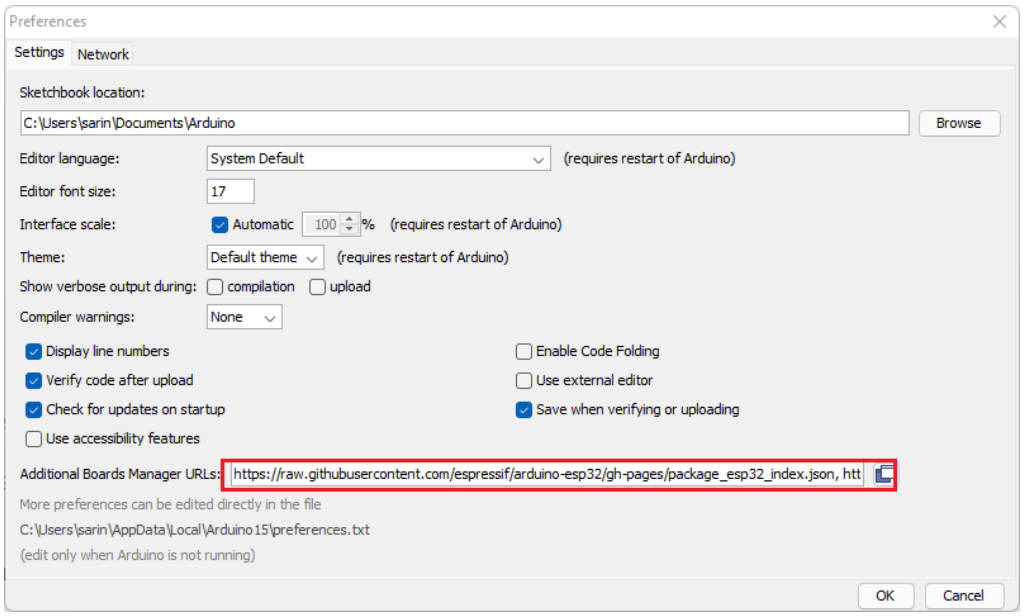
1.U Arduino IDE-u potrebno je otići na Datoteka (engl. *File*) > Preporučene vrijednosti (engl. *Preferences*) [14].



Sl. 3.4. Postavke u Arduinu [14].

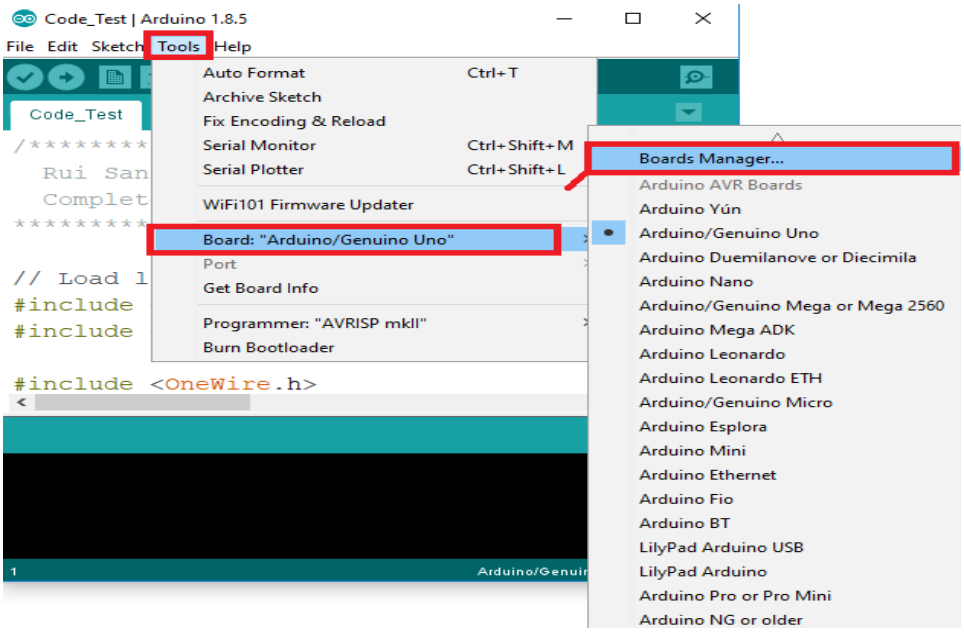
2. Unijeti sljedeće u polje "*Additional Board Manager URLs":*

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json [14].



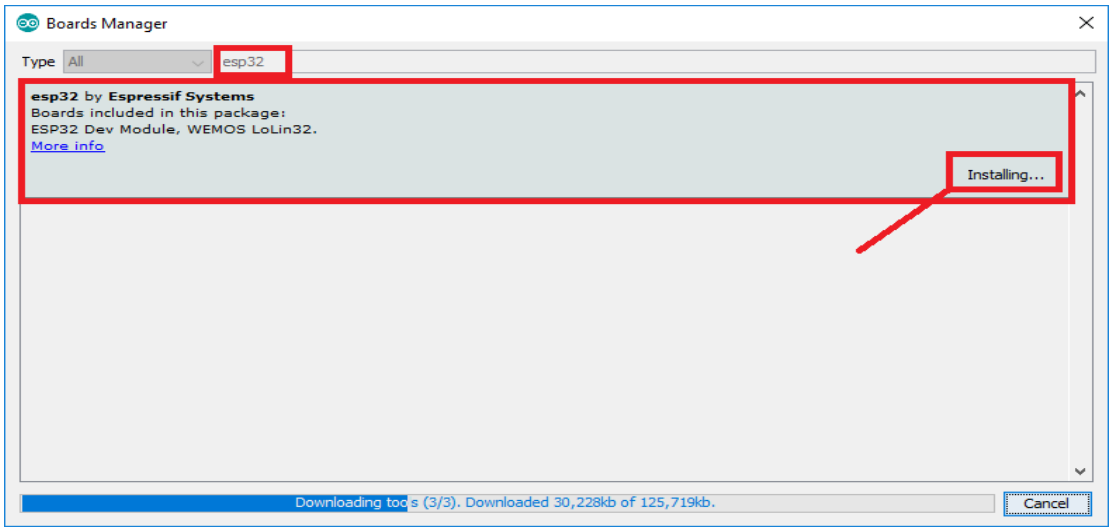
Sl. 3.5. Postavke u Arduinu [14].

3. Otvoriti upravitelj pločica. Otići na Alati (engl. *Tools*) > Pločica (engl. *Board*) > Upravitelj pločica...(engl. *Boards Manager*) [14].



Sl. 3.6. Postavke u Arduinu [14].

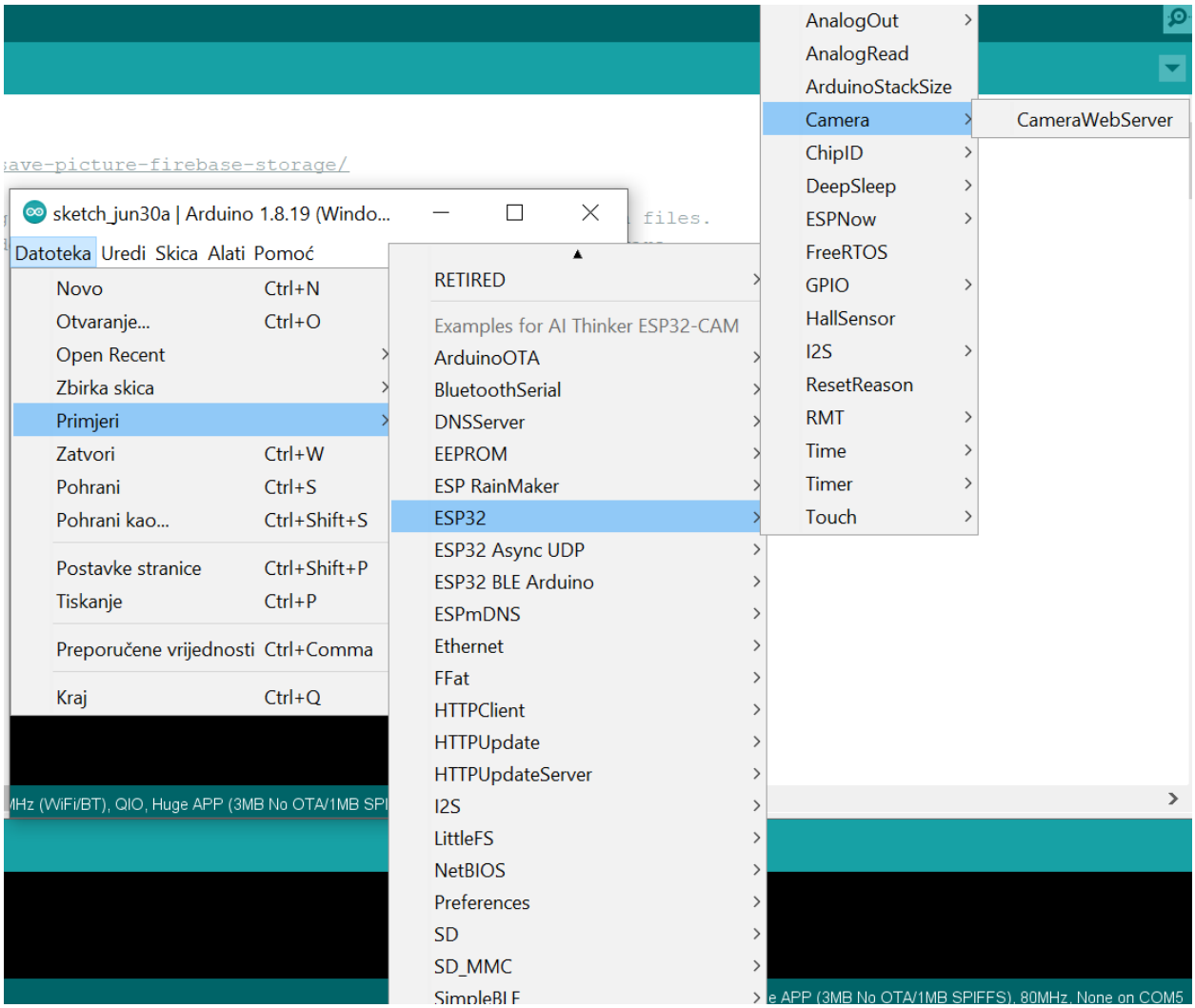
4. Potražiti ESP32 i pritisnuti gumb za instalaciju za " *ESP32 by Espressif Systems* " [14].



Sl. 3.7. Postavke u Arduinu [14].

5.Nakon par sekundi paket je instaliran [14].

Nakon tih koraka testira se instalacija tako što se ESP32 ploča spoji na računalo i u izborniku Alati (engl. *Tools*) >Ploča (engl. Board) se izabere odgovarajuća ploča. Budući da se u ovom radu koristi model *AI Thinker* ESP32-CAM, u tom slučaju je potrebno tu ploču odabrati. Potrebno je i odabrati odgovarajući port u izborniku Alati (engl. *Tools*) > Port. Arduino IDE u postavkama ima mogućnost da se pristupi tzv. streamingu kamere na webu tako što se pokretanjem serijskog monitora dobije IP adresa gdje se mogu testirati razne opcije slike(rezolucija, kvaliteta, osvjetljenje itd.). Koraci koji dovode do tog kôda su prikazani na slici 3.8.



Sl. 3.8. Postavke u Arduinu.

Poslije otvaranja kôda unose se podaci Wi-Fi-ja na koji se kamera povezuje.

SSID i šifra se unose u ovom dijelu kôda prikazanom na slici između navodnika umjesto zvjezdica.

```
const char* ssid = "**********";
const char* password = "**********";
```

Sl. 3.9. Dio kôda iz Arduino IDE.

Poslije upisa, zadnji korak povezivanja kamere s Arduinom je učitavanje kôda i otvaranje serijskog monitora s brzinom prijenosa od 115200.

Osim toga funkcionalnost kamere se može testirati raznim projektima. U ovom slučaju je to korištenjem skladišta Firebase-a.

3.2.2. Izrada Firebase projekta

Kako bi se slike mogle slati na Firebase skladište, kreira se račun na koji se Arduino IDE povezuje. To se može napraviti u par koraka:

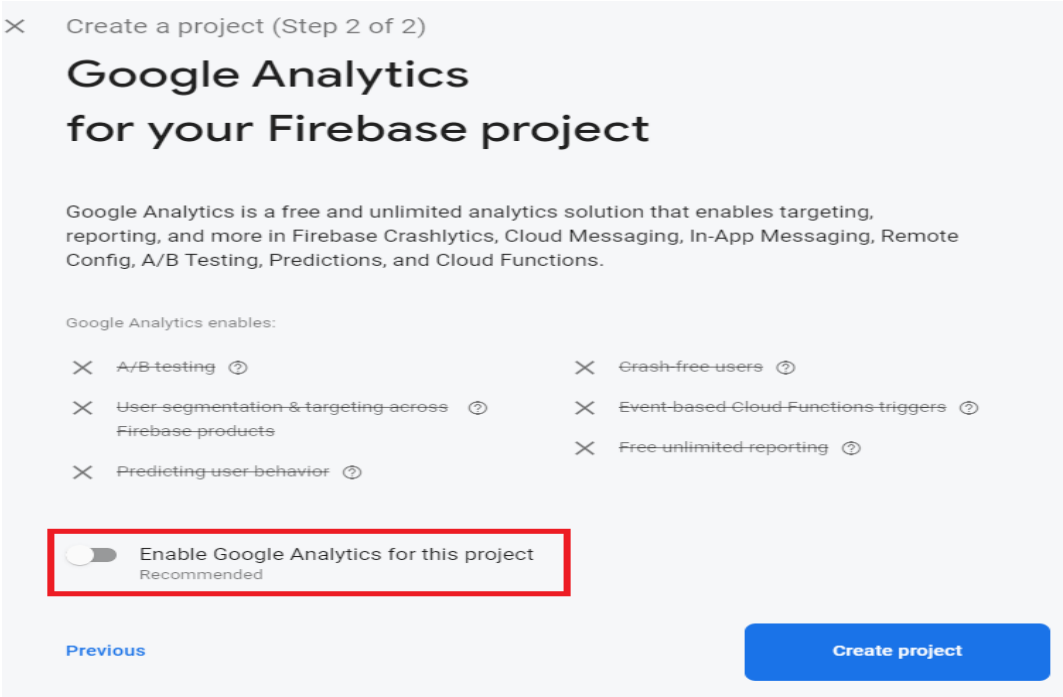
1. Na Firebase-u se prijavljuje pomoću Google računa.

2.Nakon toga se pritišće gumb Započni (engl. *Get Started*) zatim Dodaj projekt (engl. *Add project*) kako bi se kreirao novi projekt.

3.Zatim se dodjeljuje ime projektu(u ovom primjeru je *ESP Firebase Demo*) [15].

Sl. 3.10. Postavke u Firebase-u [15].

4. Isključuje se opcija Omogući Google Analitiku za ovaj projekt jer nije potrebna i klikće se Kreiraj projekt (engl. *Create project*) [15].



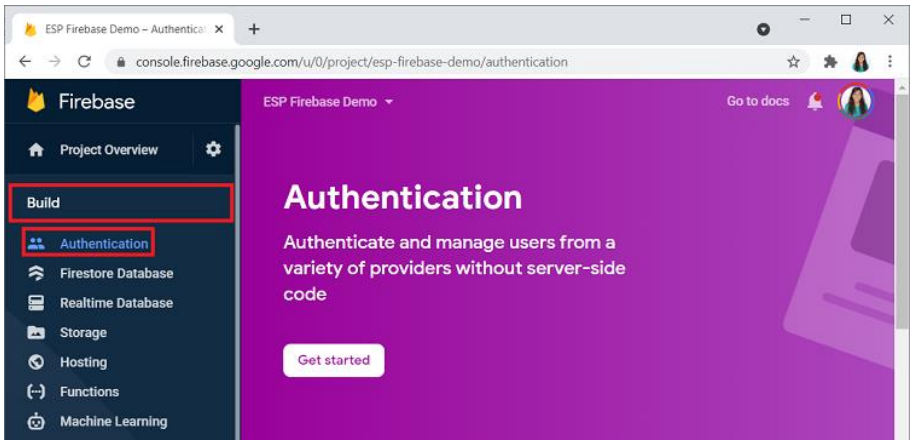
Sl. 3.11. Postavke u Firebase-u [15].

5.Nakon što se postavi projekt, pritišće se Nastavi (engl. *Next*).

6.Zatim se korisnik preusmjerava na stranicu konzole Projekt [15].

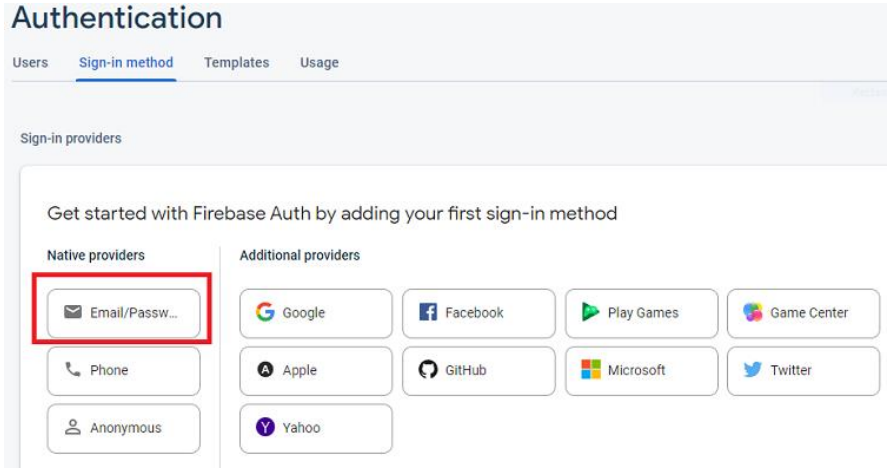
Nakon svih tih koraka bitna je autentifikacija. Da bi se omogućila autentifikacija putem e-pošte i lozinke, prvo se postavljaju metode autentifikacije za aplikaciju. Većina aplikacija mora znati identitet korisnika. Drugim riječima, brine se za prijavu i identifikaciju korisnika (u ovom slučaju, ESP32-CAM). Poznavanje identiteta korisnika omogućuje aplikaciji sigurno spremanje korisničkih podataka u *cloud* i pružanje istog personaliziranog iskustva na svim uređajima korisnika. To se ostvaruje u sljedećim koracima:

1.Odlazi se u Firebase-u na ovjeru (engl. *Authentication*) [16].



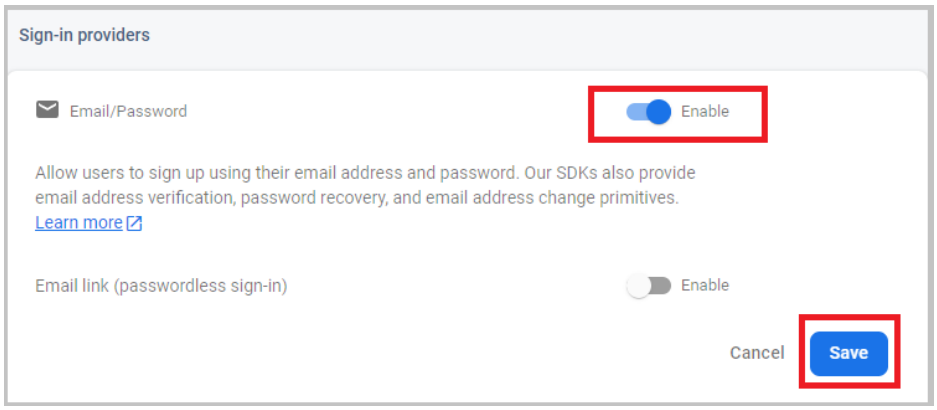
Sl. 3.12.Postavke u Firebase-u [16].

2.Odabire se metoda *Email/Password* [16].



Sl. 3.13. Postavke u Firebase-u [16].

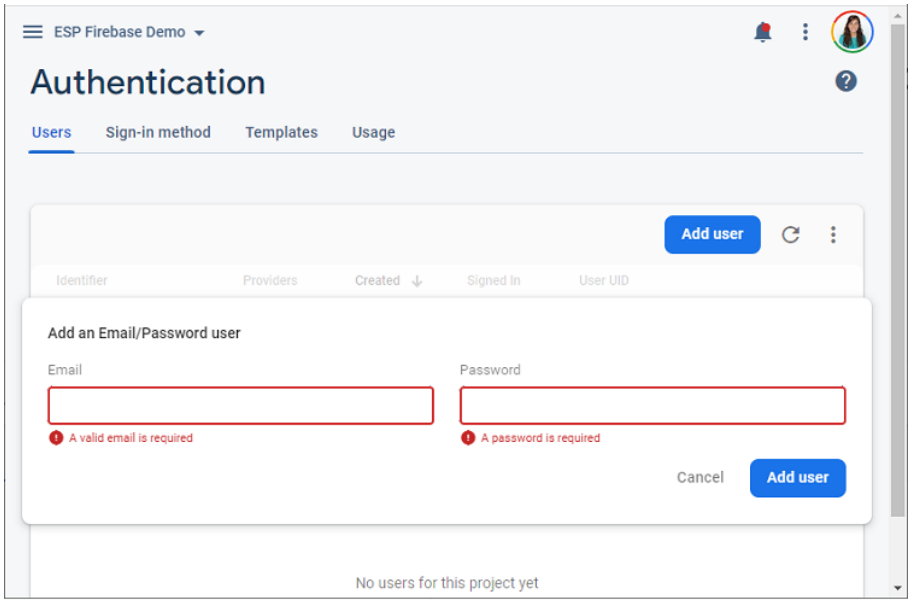
3.Omogućuje se metoda provjere autentičnosti klikom na Spremi (engl. *Save*) [16].



Sl. 3.14. Postavke u Firebase-u [16].

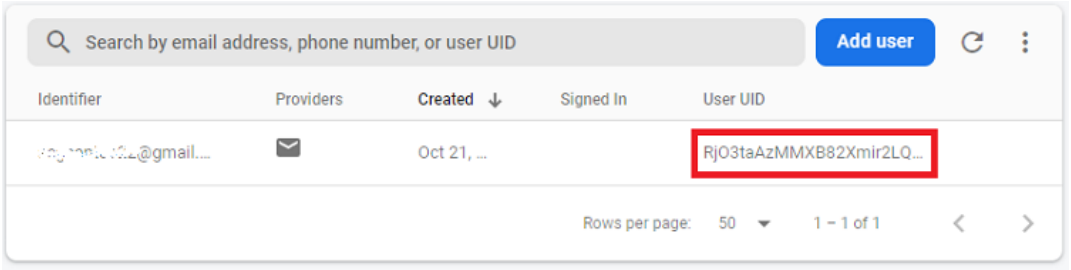
6.Nakon toga, autentifikacija putem e-pošte i lozinke je omogućena.

7.Sljedeći korak je dodavanje korisnika [16].



Sl. 3.15. Postavke u Firebase-u [16].

8.Nakon tih koraka korisnik je uspješno kreiran [16].



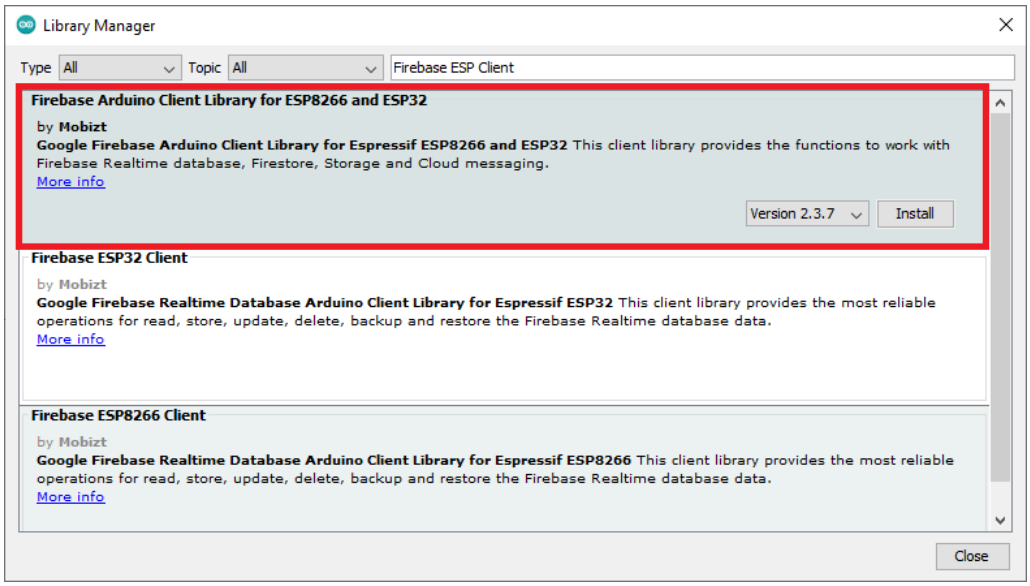
Sl. 3.16. Postavke u Firebase-u [16].

Firebase stvara jedinstveni UID za svakog registriranog korisnika. Korisnički UID omogućuje identifikaciju korisnika i praćenje korisnika da bi se omogućilo ili zabranio pristup projektu ili bazi podataka.

I zadnji veliki korak u Firebase-u je stvaranje spremnika za pohranu. Nakon klika na skladište na lijevoj strani Firebase-a, klikće se na gumb Započni (engl. *Get started*), zatim se prihvaćaju zadana sigurnosna pravila klikom na gumb Sljedeće (engl. *Next*). Nakon toka odabire se regija (najbližu zemlji korisnika). Nakon par sekundi sve na Firestore-u je spremno za rad s Arduinom [17].

3.2.3. Povezivanje Arduina s Firebase-om

Prvi korak pri otvaranju Arduina je instaliranje biblioteke *ESP Firebase Client Library*. Biblioteka *Firebase-ESP-Client* nudi nekoliko primjera sučelja s Firebase uslugama. Pruža primjer koji pokazuje kako poslati datoteke u Firebase skladište. Postupak instaliranja biblioteke je već objašnjen kod povezivanja kamere a Arduinom [18].



Sl. 3.17. Postavke u Arduinu [18].

Sljedeće što je potrebno je pisanje samog kôda u novoj datoteci Arduina.

Kako bi sve funkcioniralo kod kompiliranja, na početku se uključuju potrebne biblioteke.

```
#include "WiFi.h"
#include "esp_camera.h"
#include "Arduino.h"
#include "soc/soc.h"          // Onemogućuje problem zatamnjenja
#include "soc/rtc_cntl_reg.h"//Onemogućuje problem zatamnjenja
#include "driver/rtc_io.h"
#include <SPIFFS.h>
#include <FS.h>
#include <Firebase_ESP_Client.h>
//Pruža informacije o procesu generiranja tokena
#include <addons/TokenHelper.h>
```

Sl. 3.18. Dio kôda iz Arduino IDE [18].

Sljedeći dio kôda je već objašnjen ranije kod spajanja kamere s internetom korisnika.

```
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

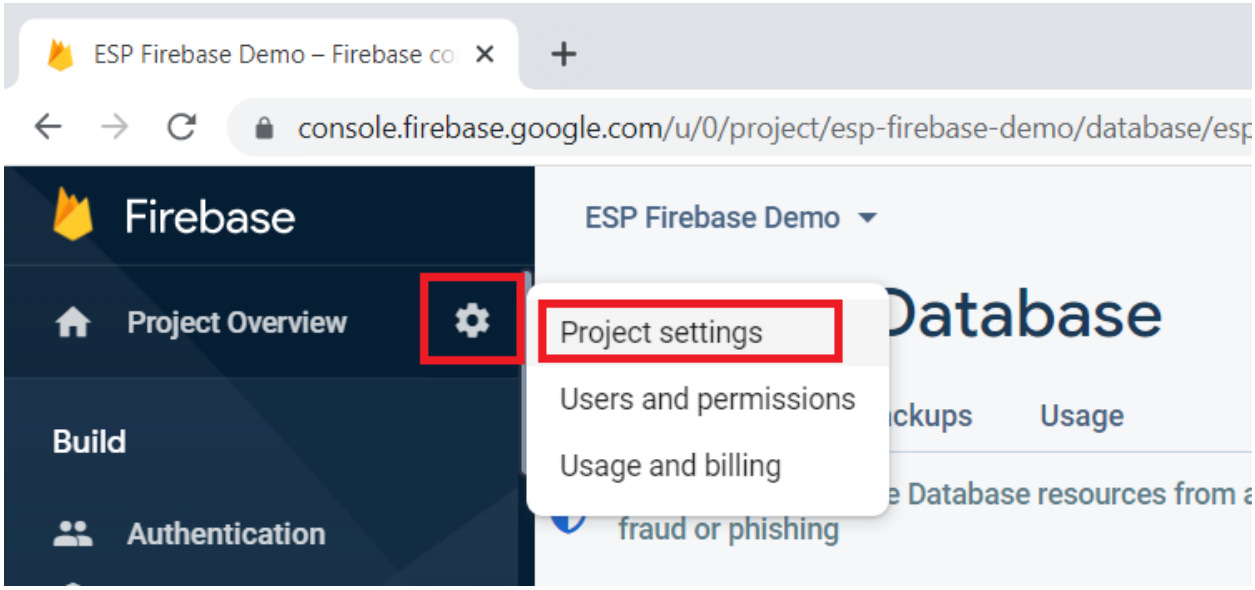
Sl. 3.19. Dio kôda iz Arduino IDE [18].

U ovom dijelu se stavlja API ključ koji se može izvući s Firebase-a.

```
#define API_KEY "REPLACE_WITH_YOUR_FIREBASE_PROJECT_API_KEY."
```

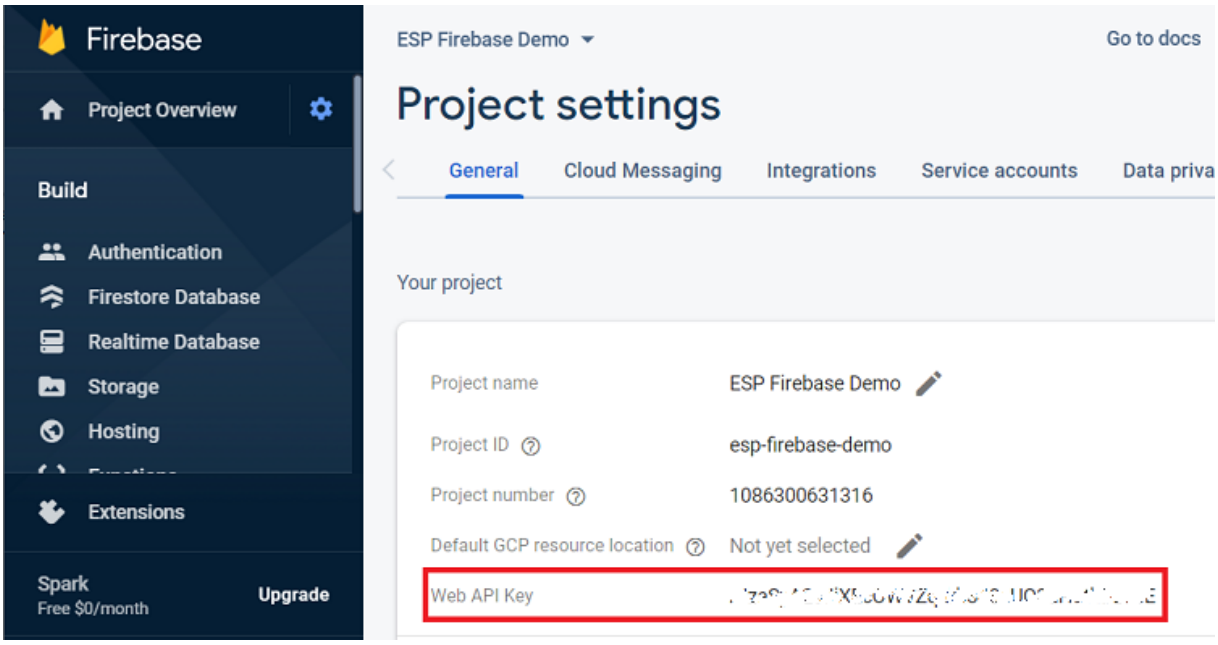
Sl. 3.20. Dio kôda iz Arduino IDE [18].

API ključ se dohvaća tako što se na Firebase-u klikne s lijeve strane u izborniku na postavke projekta.



Sl. 3.21. Postavke u Firebase-u [19].

Iz slike 3.22 se može vidjeti gdje se nalazi API ključ.



Sl. 3.22. Postavke projekta u Firebase-u [19].

```
#define USER_EMAIL "REPLACE_WITH_THE_AUTHORIZED_USER_EMAIL"
#define USER_PASSWORD "REPLACE_WITH_THE_AUTHORIZED_USER_PASSWORD"
```

Sl. 3.23. Dio kôda iz Arduino IDE [18].

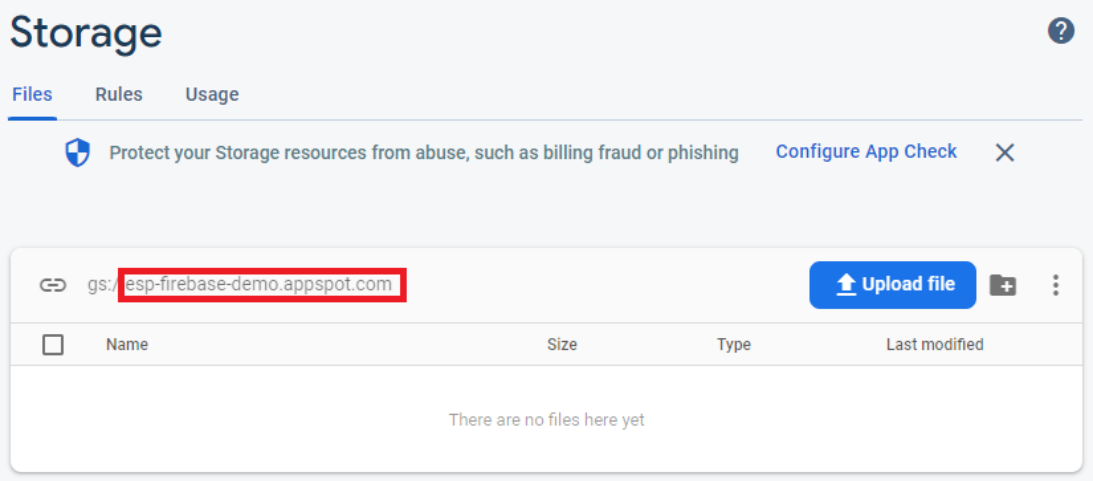
U nastavku kôda se upisuju e-mail i šifra odnosno podaci koji su dodani pri autentifikaciji.

```
#define STORAGE_BUCKET_ID "REPLACE_WITH_YOUR_STORAGE_BUCKET_ID"
```

Sl. 3.24. Dio kôda iz Arduino IDE [18].

Definira se ID Firebase spremnika koji se nalazi u izborniku Skladište (engl. *Storage*).

Prema slici 3.25. crveno označeni dio je potrebno zalijepiti u kôd između navodnika.



Sl. 3.25. Skladište Firebase-a [17].

U nastavku se definira put odnosno staza gdje se slika sprema i pinovi za modul ESP32-CAM AI Thinker koji se koristi u ovom radu.

```
#define FILE_PHOTO "/data/photo.jpg"
// pinovi OV2640 modula kamere (CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22
```

Sl. 3.26. Dio kôda iz Arduino IDE [18].

Sljedeće što slijedi su funkcije i varijable koje povezuju Arduino i ESP32-CAM. Na vrhu se nalaze boolean varijable i funkcije koje provjeravaju uspješnosti određenih radnji poput dohvaćanja slike pri pokretanju ploče, povezivanja na Firebase, spremanje slike i slično. Funkcija capturePhotoSpiffs prvo provjerava uspješnost dohvaćanja slike. Nakon provjere, ako je uspješno slika dohvaćena, funkcija ju sprema u datoteku slike u određenom formatu. Tu su i funkcije initWiFi, initSPIFFS i initCamera koje se koriste u funkciji setup i koje služe za inicijalizaciju kamere, SPIFFS datotečnog sustava i WiFi-a. Dohvaćanje slike i slanje na Firebase se obavlja u loop funkciji.

```
boolean takeNewPhoto = true;//takeNewPhoto provjerava je li vrijeme za
snimanje nove fotografije. Postavit ćemo ga na true, tako da slika
kad se ploča prvi put pokrene.

// Zatim definiramo Firebase konfiguracijske podatkovne objekte.
FirebaseAuth auth;
FirebaseConfig configF;

bool taskCompleted = false;// taskCompleted je boolean varijabla
koja provjerava jesmo li se uspješno povezali na Firebase.

// checkPhoto funkcija provjerava je li slika uspješno snimljena i
spremljena u SPIFFS.
bool checkPhoto( fs::FS &fs ) {
  File f_pic = fs.open( FILE_PHOTO );
  unsigned int pic_sz = f_pic.size();
  return ( pic_sz > 100 );
}

// capturePhotoSaveSpiffs funkcija snima fotografiju i sprema je u
datotečni sustav ESP32.
void capturePhotoSaveSpiffs( void ) {
  camera_fb_t * fb = NULL; // pointer
  bool ok = 0; // Provjera je li slika dohvaćena točno
  do {
    // Dohvaćanje slike
    Serial.println("Taking a photo...");

    fb = esp_camera_fb_get();
    if (!fb) {
      Serial.println("Camera capture failed");
      return;
    }
    // Naziv datoteke fotografije
    Serial.printf("Picture file name: %s\n", FILE_PHOTO);
    File file = SPIFFS.open(FILE_PHOTO, FILE_WRITE);
    // Ubacuje podatke u datoteku slike
    if (!file) {
      Serial.println("Failed to open file in writing mode");
    }
    else {
      file.write(fb->buf, fb->len); // veličina (slika), veličina
dužina
      Serial.print("The picture has been saved in ");
      Serial.print(FILE_PHOTO);
      Serial.print(" - Size: ");
      Serial.print(file.size());
      Serial.println(" bytes");
    }
```

Sl. 3.27. Dio kôda iz Arduino IDE [18].

```
    // Zatvaranje datoteke
    file.close();
    esp_camera_fb_return(fb);

    // provjera je li datoteka uspješno spremljena u SPIFFS
    ok = checkPhoto(SPIFFS);
  } while ( !ok );
}

void initWiFi(){ //funkcija initWIfi inicijalizira Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
}

void initSPIFFS(){ //funkcija initSPIFFS inicijalizira datotečni sustav
SPIFFS

  if (!SPIFFS.begin(true)) {
    Serial.println("An Error has occurred while mounting SPIFFS");
    ESP.restart();
  }
  else {
    delay(500);
    Serial.println("SPIFFS mounted successfully");
  }
}
```

Sl. 3.28. Dio kôda iz Arduino IDE [18].

```
void initCamera(){ //funkcija initCamera inicijalizira ESP-32 CAM
 // OV2640 Kamera modul
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

if (psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }
// Inicijalizacija kamere
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    ESP.restart();
  }
}
```

Sl. 3.29. Dio kôda iz Arduino IDE [18].

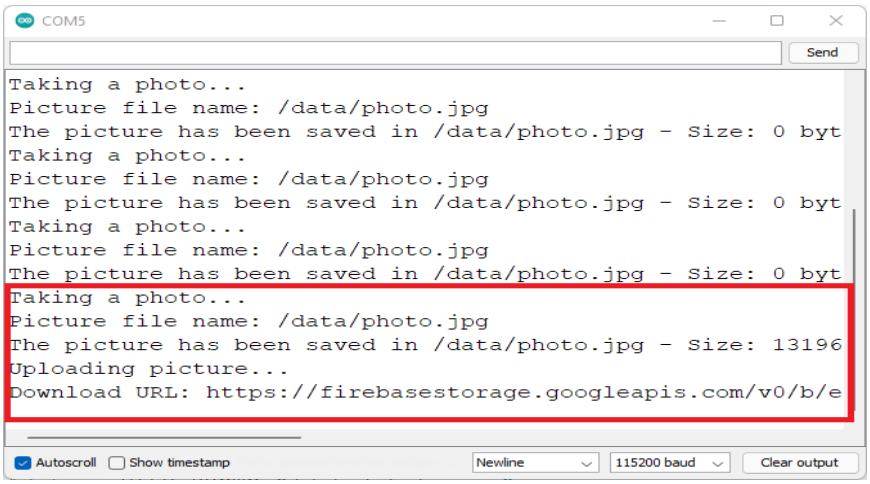
```
void setup() {
  // Serijski port za potrebe debuggiranja
  Serial.begin(115200);
  initWiFi();
  initSPIFFS();
  // Isključuje se „detektor prekida“
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  initCamera();

  //dodijeljuju se sljedeće postavke Firebase konfiguracijskim objektima
  // Dodjeljuje se api ključ
  configF.api_key = API_KEY;
  //dodjeljuju se korisnički podaci za prijavu
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
  //Dodjeljivanje povratne funkcije za dugotrajni zadatak generiranja
tokena
  configF.token_status_callback = tokenStatusCallback; //pogledati
addons/TokenHelper.h


  Firebase.begin(&configF, &auth); //inicijalizacija firebase-a
  Firebase.reconnectWiFi(true);

}
void loop() {
  if (takeNewPhoto) { //petlja za dohvaćanje i spremanje nove slike u
datotečni sustav
    capturePhotoSaveSpiffs();
    takeNewPhoto = false;
  }
  delay(1);
          //dio koda za slanje slike u  Firebase
  if (Firebase.ready() && !taskCompleted){
    taskCompleted = true;
    Serial.print("Uploading picture... ");
```

Sl. 3.30. Dio kôda iz Arduino IDE [18].

```
   //naredba koja zapravo šalje sliku je Firebase.Storage.upload(…)
 if (Firebase.Storage.upload(&fbdo, STORAGE_BUCKET_ID , FILE_PHOTO ,
mem_storage_type_flash ,FILE_PHOTO, "image/jpeg" )){
      Serial.printf("\nDownload URL: %s\n", fbdo.downloadURL().c_str());
    }
    else{
      Serial.println(fbdo.errorReason());
    }
  }
}
```

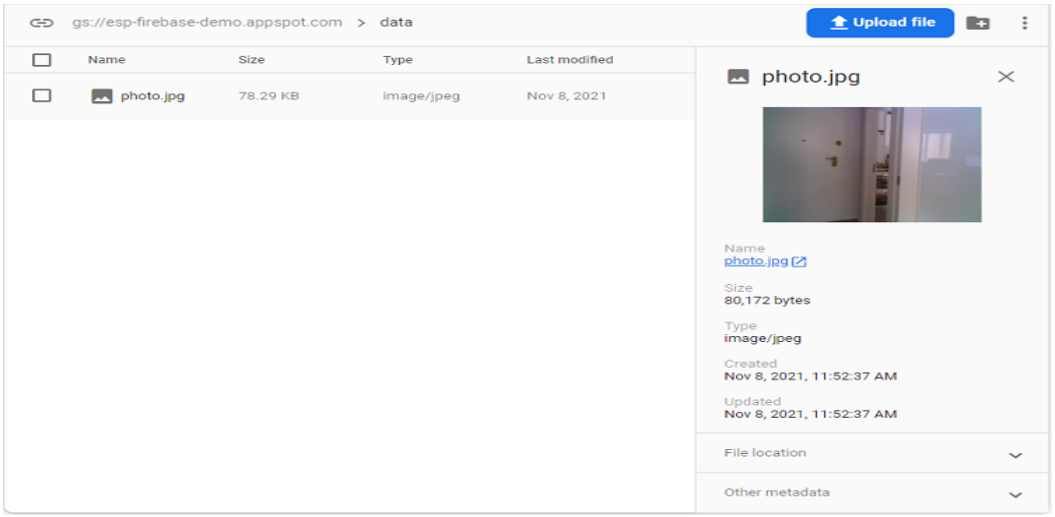
Sl. 3.31. Dio kôda iz Arduino IDE [18].

Nakon učitavanja kôda, otvara se serijski monitor, prikazan na slici 3.32, pri brzini prijenosa podataka od 115200.



Sl. 3.32. Serijski monitor iz Arduina [18].

Nakon toga kreira se tzv. mapa data, prikazana na slici 3.33, koja sadrži sliku i preko koje se na Firebase-u pristupa slici. Također se slici može pristupiti preko URL-a za preuzimanje ispisanom na serijskom monitoru [18].



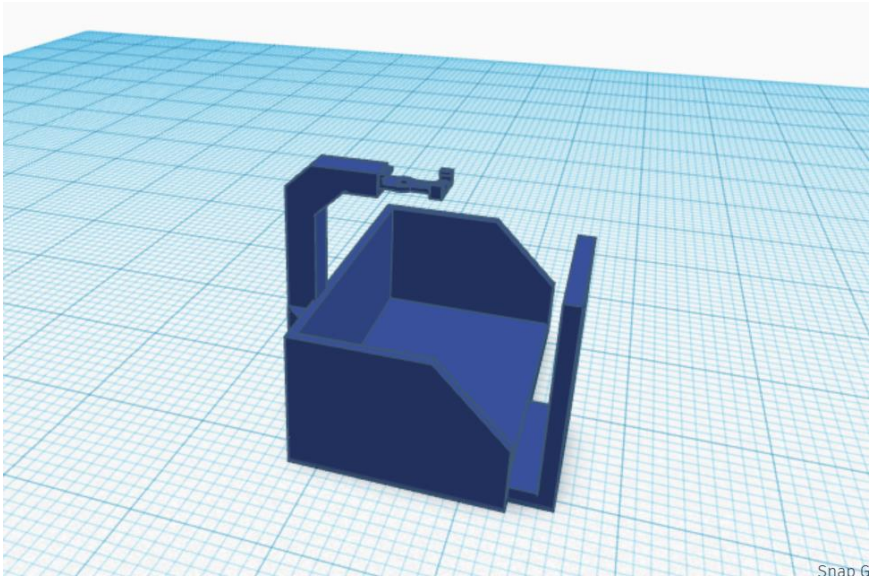
Sl. 3.33. Skladište slika u Firebase-u [18].

Na taj način se slike mogu spremati u datoteku koja se nalazi u skladištu Firebase-a. Bitno je naglasiti da se ovakav projekt može upotrijebiti s web aplikacijom gdje se dohvaćene slike mogu prikazati i iskoristiti.

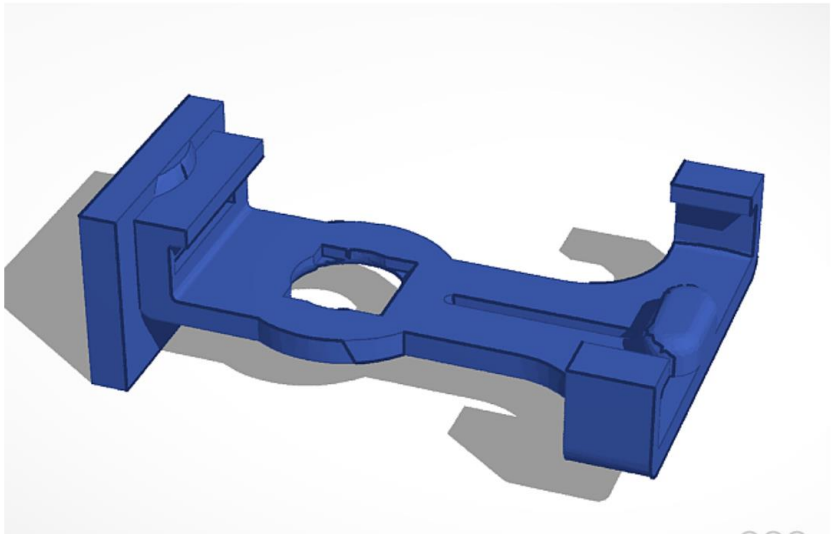
3.3. Model uređaja za prikupljanje slike s refraktometra

Nakon osposobljavanja sustava kamere, zadnji korak je projektiranje odnosno crtanje modela u programu za 3D dizajn, te nakon toga printanje tog istog dizajna. Funkcija samog modela je već predstavljena u samom uvodu. U ovom dijelu se prikazuje model u programu za 3D dizajn.

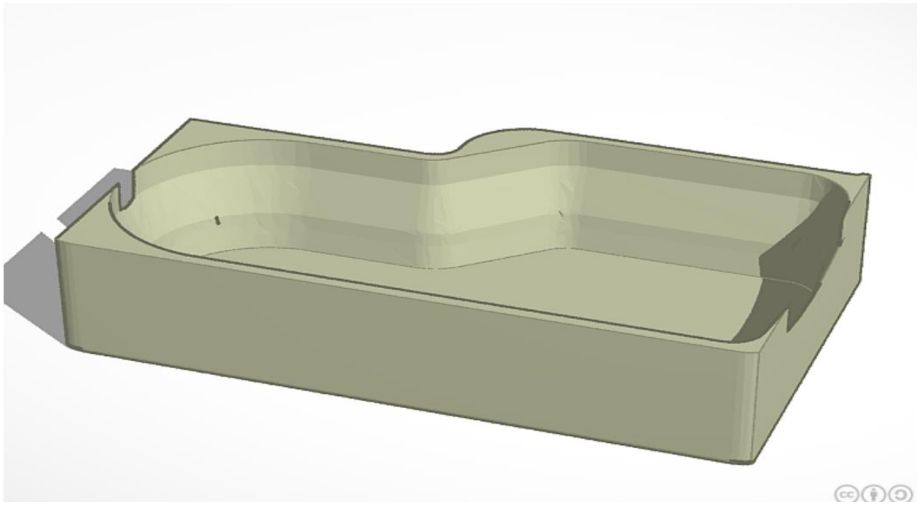
Model, prikazan na slici 3.34, se sastoji od kutije u kojoj je smješten refraktometar. Ta kutija, vidljiva na slici 3.36, je smještena u jednu veću kutiju. Veća kutija sadržava držač za kameru, koji je prikazan na slici 3.35. Držač je pozicioniran tako da gleda u LCD refraktometra s visine od 20 cm. Na suprotnoj strani od stupa koji drži kameru je 15cm visoka pregrada na kojoj je smještena LED traka. Traka je pozicionirana tako da daje zadovoljavajuće osvjetljenje LCD-a. Sljedeće slike prikazuju dizajn modela, odnosno njegovih komponenti u Tinkercadu.



Sl. 3.34. Prikaz 3D modela za prikupljanje uzoraka u Tinkercadu.



Sl. 3.35. Prikaz 3D držača za kameru u Tinkercadu.



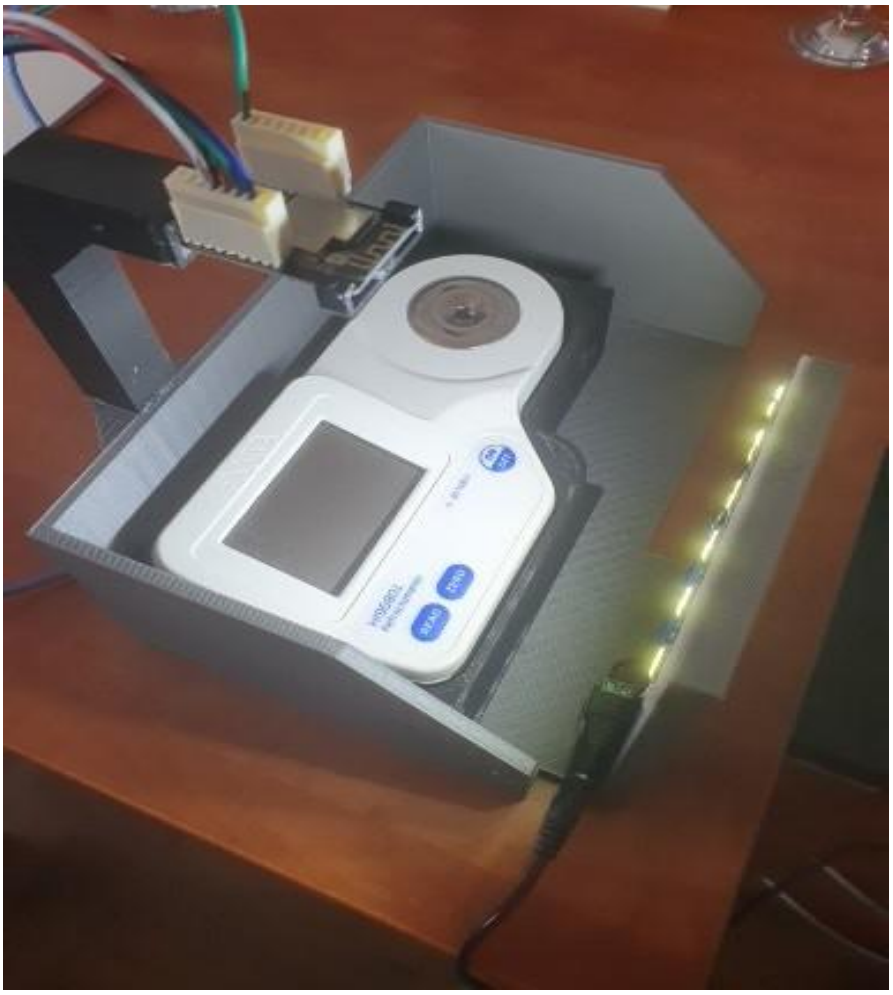
Sl. 3.36. Prikaz 3D držača za refraktometar u Tinkercadu.

4. IMPLEMENTACIJA I EKSPERIMENTALNI REZULTATI

Nakon 3D dizajna i printanja, isprintani model se objedinjuje s kamerom i refraktometrom uz osvjetljenje LED trake. IoT uređaj, prikazan na slici 4.1, i uspješnost prikupljanja slika su opisani u nastavku.

4.1. IoT uređaj za prikupljanje slika s refraktometra

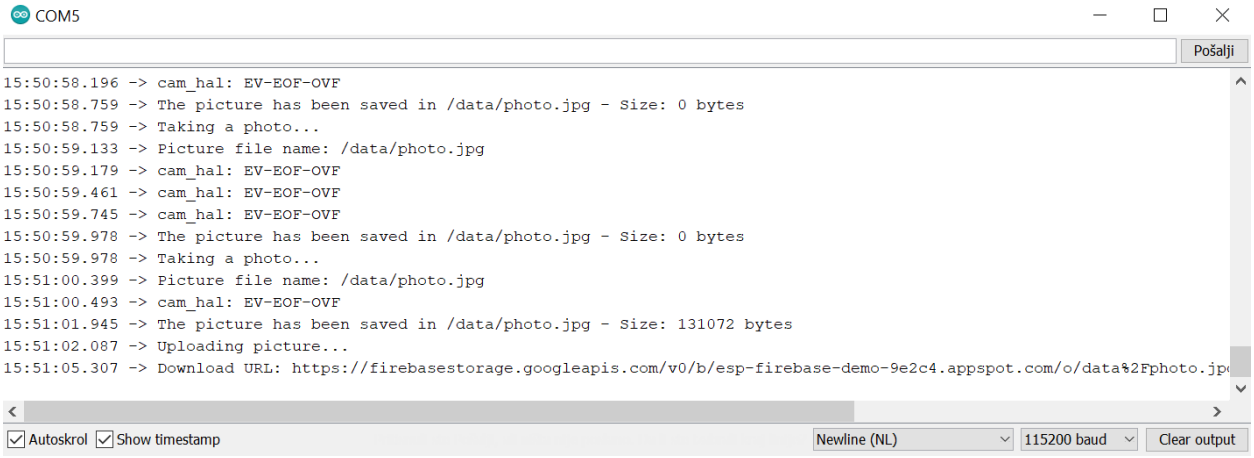
Smještanjem refraktometra i kamere na određene pozicije isprintanog modela može se testirati funkcionalnost Iot uređaja. Iz navedene slike 4.1 može se vidjeti prikaz uređaja koji sadrži i LED traku koja pomaže kameri za dobivanje kvalitetnije slike.



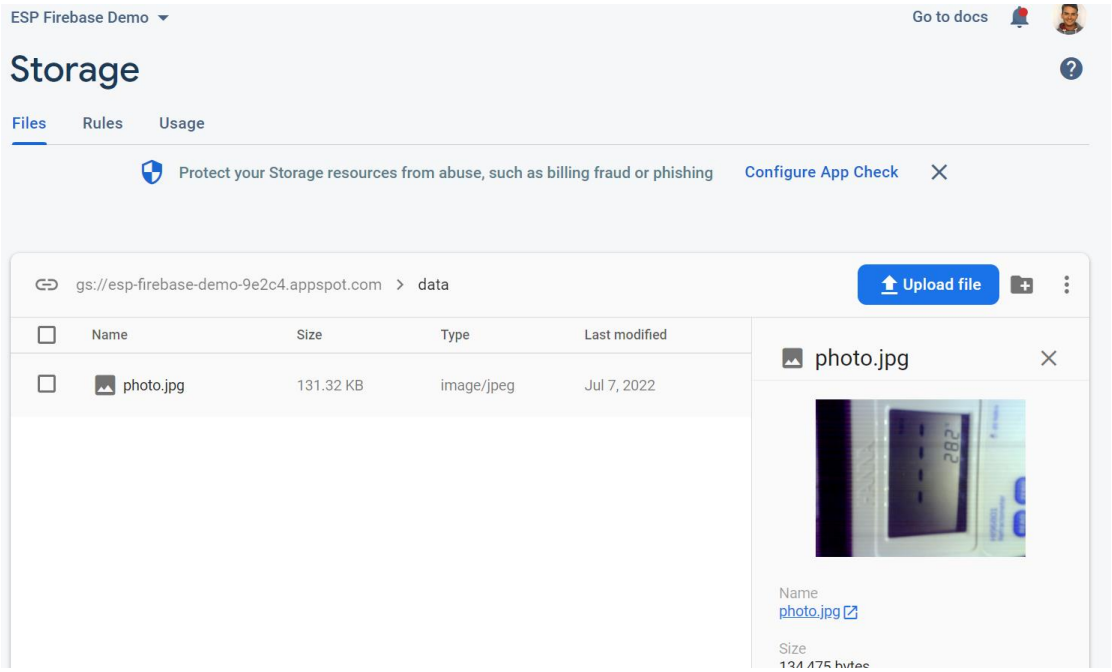
Sl. 4.1. Prikaz realiziranog modela za prikupljanje podataka s refraktometra.

4.2. Firebase baza prikupljenih slika

Nakon postavljanja kamere i osvjetljenja na potrebno mjesto, pokreće se kod na Arduinu kako bi se testiralo dohvaćanje i spremanje slike na Firebase. Na slikama 4.6 i 4.7 i 4.8 se mogu vidjeti rezultati i korištena LED traka.

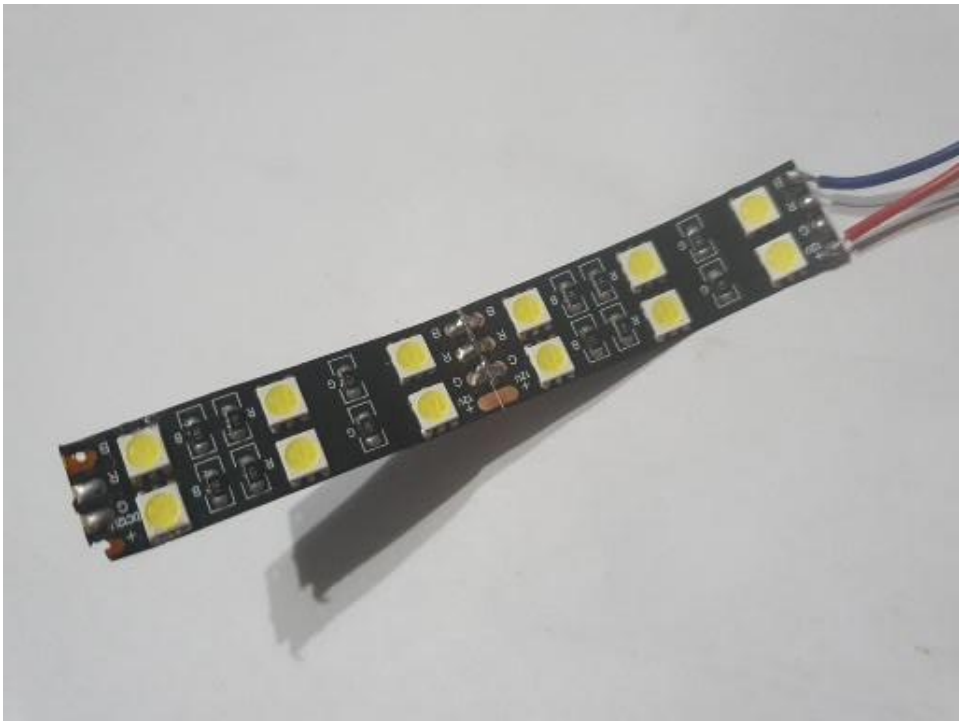


Sl. 4.2. Rezultati sa serijskog monitora u Arduinu.



Sl.4.3. Prikaz skladišta Firebase-a u kojem je spremljena slika s kamere.

Sl. 4.4. Rezultat mjerenja dohvaćen s kamere.



Sl. 4.5. LED traka.

5. ZAKLJUČAK

Kroz ovaj je rad opisan refraktometar, postupak mjerenja s njim, sustav kamere tj. rukovanje s njom, model koji je dizajniran u 3D programu te prikaz uređaja s kamerom, refraktometrom i LED trakom. Opisan je i način spremanja slike dohvaćene s kamere na Firebase uz korištenje Arduina. Cilj je bio prikupljati podatke mjerenja na što lakši način, što je i na neki način ostvareno. Već je spomenuto u drugoj cjelini glavnog dijela da se brojnim projektima mogu prikupljati slike s kamere. Firebase je bio samo jedan od tih projekata koji zapravo za razliku od nekih ima dosta potencijala tj. može se iskoristiti kroz web aplikaciju koja se može izraditi i nuditi razne opcije dijeljenja slika korisnicima. ESP32-CAM modul zna nekad imati poteškoće s povezivanjem na Internet, ali u većini slučajeva je zadatak dohvaćanja slike uspješno bio obavljen. Osim kamere sve ostalo je samo bilo potrebno staviti na dobar položaj. Kada se već govori o položaju, poziciju LED trake je dosta nezgodno odrediti, jer u većini pozicija daje osvjetljenje prejako, odnosno zna stvorit tzv. „blijesak“ na LCD-u refraktometra. Što se tiče izrade modela, njegova izrada u 3D programu dosta skrati posao i poboljša ostvarivanje funkcionalnosti, jer to je uglavnom kreiranje nečega što se kasnije realizira printanjem, bez ikakvih traženja dijelova i rezanja koje mogu dodatno odužiti posao i može se reći da je upravo to dobra strana dizajniranja u 3D programima.

U konačnici ovaj završni rad je predstavio samo male dijelove koji se mogu iskoristiti za puno toga. Iz ovoga svega se može dovoljno naučiti da se napravi dosta toga s 3D dizajnom, ili čak ESP32-CAM.

LITERATURA

[1] *More informations, Hanna instruments*, dostupno na: https://hannainst.in/hi96801-digital-refractometer-for-brix-analysis(zadnji put pristupljeno:7.7.2022.)

[2] *Specifications, Hanna instruments*, dostupno na: https://hannainst.in/hi96801-digital-refractometer-for-brix-analysis (zadnji put pristupljeno:7.7.2022.)

[3] Jaywant, S.A.; Singh, H.; Arif, K.M., *Sensors and Instruments for Brix Measurement:* A Review. Sensors 2022, str.1, ožujak 2022.

[4] Brix, Wikipedia, dostupno na: https://en.wikipedia.org/wiki/Brix

(zadnji put pristupljeno:7.7.2022)

[5] *Brix measurement: Oechsle,* Metler Toledo, dostupno na:

https://www.mt.com/int/en/home/perm-lp/product-organizations/ana/brix-meters.html?cmp=sea_04010123&SE=GOOGLE&Campaign=MT_ANA-DERE_EN_ROW&Adgroup=DERE_Brix-Meter_Phrase&bookedkeyword=%2Bmeter%20%2Bbrix&matchtype=b&adtext=396530510117&placement=&network=g&kclid=_k_Cj0KCQjwlK-WBhDjARIsAO2sErTfxxSFJoMpFfyLzUxGfHkKCsAQrXzSKll6URFxPwvLikErt682vdIaAlj4EALw_wcB_k_&cq_src=google_ads&cq_cmp=260833706&cq_con=60227225162&cq_term=%2Bmeter%20%2Bbrix&cq_med=&cq_plac=&cq_net=g&cq_pos=&cq_plt=gp&gclid=Cj0KCQjwlK-WBhDjARIsAO2sErTfxxSFJoMpFfyLzUxGfHkKCsAQrXzSKll6URFxPwvLikErt682vdIaAlj4EALw_wcB

(zadnji put pristupljeno:24.8.2022.)

[6] *Kern Analogni refraktometar KERN ORA 3AA, Brix skale,* Conrad Electronic, dostupno na:

https://www.conrad.hr/p/kern-analogni-refraktometar-kern-ora-3aa-brix-skale-2522534

(zadnji put pristupljeno:24.8.2022.)

[7] *Priručnik s uputama za refraktometar KERN ORA 3AA-AB Bee Analogs*, 2022., dostupno na:

https://manuals.plus/sn/kern/ora-3aa-ab-bee-analoges-refractometer-manual#technical_data

(zadnji put pristupljeno:24.8.2022.)

[8] *OenoFoss; Tehničke karakteristike*, FOSS, dostupno na:

https://www.fossanalytics.com/en/products/oenofoss?utm_source=Google_Adwords&utm_medium=Adwords&utm_campaign=Brand-Instruments&utm_keyword=oenofoss&gclid=Cj0KCQjw0oyYBhDGARIsAMZEuMsaTEjqpcCZ2gn4tQlInjxEjFSD4aKq9EfnIa2JCIgwOFozDRFLRhQaAkaAEALw_wcB

(zadnji put pristupljeno:24.8.2022.)

[9] *Kontrola slatkih vina FTIR-om*, FOSS, dostupno na:

https://www.fossanalytics.com/es-ar/news-articles/wine/dr-loosen

(zadnji put pristupljeno:24.8.2022.)

[10] ESP32, *Wikipedija*, dostupno na: https://en.wikipedia.org/wiki/ESP32

(zadnji put pristupljeno:7.7.2022)

[11] Isaac, *ESP32-CAM: što biste trebali znati o ovom modulu, HWlibre*, ažurirano dana 22.02.2021., dostupno na:

https://www.hwlibre.com/hr/esp32-cam/?utm_source=dlvr.it&utm_medium=twitter

(zadnji put pristupljeno:7.7.2022)

[12] *ESP32-CAM WIFI + BLUETOOTH CAMERA MODULE WITH CAMERA MODULE OV2640:Product specifications*, SAR, dostupno na: https://www.botnroll.com/en/esp/3259-esp32-cam-wifi-bluetooth-camera-module-with-camera-module-ov2640.html

(zadnji put pristupljeno:7.7.2022)

[13] *ESP32-CAM Save Picture in Firebase Storage:What is Firebase?, Random Nerd Tutorials* , dostupno na: https://randomnerdtutorials.com/esp32-cam-save-picture-firebase-storage/

(zadnji put pristupljeno:7.7.2022)

[14] *Installing the ESP32 Board in Arduino IDE (Windows, Mac OS X, Linux),Random Nerd Tutorials*, dostupno na: https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/ (zadnji put pristupljeno:7.7.2022)

[15] *ESP32-CAM Save Picture in Firebase Storage:Create a Firebase project, Random Nerd Tutorials* , dostupno na:

https://randomnerdtutorials.com/esp32-cam-save-picture-firebase-storage/

(zadnji put pristupljeno:7.7.2022)

[16] *ESP32-CAM Save Picture in Firebase Storage:Set Authentication Metods, Random Nerd Tutorials* , dostupno na:

https://randomnerdtutorials.com/esp32-cam-save-picture-firebase-storage/

(zadnji put pristupljeno:7.7.2022)

[17] *ESP32-CAM Save Picture in Firebase Storage:Create Storage Bucket, Random Nerd Tutorials*, dostupno na:

https://randomnerdtutorials.com/esp32-cam-save-picture-firebase-storage/

(zadnji put pristupljeno:7.7.2022)

[18] *ESP32-CAM Save Picture in Firebase Storage: ESP32-CAM – Send Pictures to Firebase Storage, Random Nerd Tutorials*, dostupno na:

https://randomnerdtutorials.com/esp32-cam-save-picture-firebase-storage/

(zadnji put pristupljeno:7.7.2022)

[19] *ESP32-CAM Save Picture in Firebase Storage:Get Project API Key, Random Nerd Tutorials*, dostupno na:

https://randomnerdtutorials.com/esp32-cam-save-picture-firebase-storage/#get-API-Key

(zadnji put pristupljeno:7.7.2022)

SAŽETAK

Zadatak je bio napraviti odnosno realizirati model kućišta za automatizirano prikupljanje uzoraka s digitalnog refraktometra. Bilo je potrebno povezati kameru, refraktometar i računalo na način da se svakim mjerenjem rezultata na refraktometru slike dohvate i spreme na računalo odnosno skladište Firebase-a uz korištenje Arduino programa. Prvi dio uvodi u opis samog refraktometra, radi boljeg razumijevanja njegove funkcija i načina mjerenja s njim. Nakon opisa refraktometra, uvodi se odnosno opisuje modul kamere ESP32-CAM, njegove funkcije te njezina glavna uloga koja je zapravo dohvaćanje i spremanje slike na Firebase. Objašnjen je postupak povezivanja kamere s Arduinom koji povezuje kameru i Firebase. Spomenut je razlog odabira korištenja Firebase-a,te je objašnjen detaljno kod s Arduina uz komentare kako bi se bolje moglo razumjeti spajanje kamere s Firebase-om. I zadnji korak je izrada modela preko programa za 3D dizajniranje te realizacija istog u koji se smješta kamera i refraktometar kako bi se sve to moglo povezati u jedno. Nakon realizacije, u konačnici je izrađen model na koji se smješta refraktometar i kamera, te se realizira glavni zadatak ovog završnog rada.

Ključne riječi:refraktometar, ESP32-CAM, model, 3D dizajniranje, Arduino

ABSTRACT

The task was to make or realize a model of the housing for the automated collection of samples from a digital refractometer. It was necessary to connect the camera, refractometer and computer in such a way that with each measurement of the results on the refractometer, the images are retrieved and saved on the computer or the Firebase storage by using the Arduino program. The first part of the introduction is the description of the refractometer itself, in order to better understand its function and the way of measuring with it. After the description of the refractometer, the ESP32-CAM camera module is introduced or described, its functions and its main role, which is actually retrieving and saving the image on Firebase. The procedure for connecting the camera to the Arduino, which will connect the camera and Firebase, is explained. The reason for choosing to use Firebase is mentioned, and is explained in detail in the Arduino comments to better understand the connection of the camera to Firebase. The last step is the creation of a model using a 3D design program and the realization of the same in which the camera and refractometer will be placed so that it can all be connected into one. After the realization, a model was finally created on which the refractometer and camera are placed, and the main task of this final work is realized.

Keywords: refractometer, ESP32-CAM, model, 3D design, Arduino

ŽIVOTOPIS

Dorijan Osmakčić rođen je 13. travnja 1999. godine u Osijeku. Pohađao je osnovnu školu Stari Jankovci. Nakon završetka osnovne škole upisao je 2014. Tehničku školu Ruđera Boškovića u Vinkovcima, smjer: arhitektonski tehničar. Godine 2018. završava srednju školu i polaže državnu maturu. Nakon polaganja državne mature odlazi u Njemačku raditi jednu godinu. Godine 2019. na osnovu mature iz 2018. upisuje preddiplomski studij elektrotehnike i informacijske tehnologije na Fakultetu elektrotehnike, računarstva i informacijskih tehnologije u Osijeku.