

# Izrada alarmnog sustava za pametnu kuću uz primjenu IoT-a

---

Hohnjec, Domagoj

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:045793>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

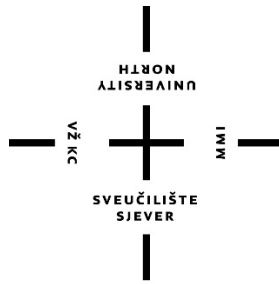
Download date / Datum preuzimanja: **2024-11-23**



Repository / Repozitorij:

[University North Digital Repository](#)





# Sveučilište Sjever

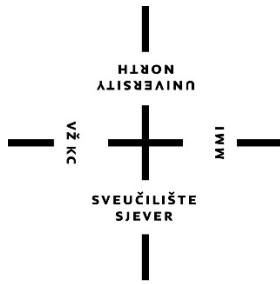
**Završni rad br. 537/EL/2024**

## **Izrada alarmnog sustava za pametnu kuću uz primjenu IoT-a**

**Domagoj Hohnjec, 0336039964**

Varaždin, srpanj 2024. godine





# Sveučilište Sjever

Odjel za Ime odjela

Završni rad br. 537/EL/2024

## Izrada alarmnog sustava za pametnu kuću uz primjenu IoT-a

**Student**

Domagoj Hohnjec, 0336039964

**Mentor**

Josip Srpak, dipl. ing.

Varaždin, srpanj 2024. godine

# Prijava završnog rada

## Definiranje teme završnog rada i povjerenstva

ODJEL Odjel za elektrotehniku

STUDIJ preddiplomski stručni studij Elektrotehnike

PRISTUPNIK Domagoj Hohnjec

MATIČNI BROJ 0336039964

DATUM 17.06.2024.

KOLEGIJ PLC sustavi upravljanja

NASLOV RADA Izrada alarmnog sustava za pametnu kuću uz primjenu IoT-a

NASLOV RADA NA ENGL. JEZIKU Creating an alarm system for a smart home with the application of IoT

MENTOR Josip Srpak

ZVANJE viši predavač

ČLANOVI POVJERENSTVA

1. mr.sc. Ivan Šumiga, viši predavač
2. Josip Srpak, viši predavač
3. doc.dr.sc. Dunja Srpak
4. Miroslav Horvatić, viši predavač
5. \_\_\_\_\_

VŽ KC

MMI

## Zadatak završnog rada

BROJ 537/ELJ2024

OPIS

U završnom radu treba objasniti pojam i primjenu Interneta stvari (IoT), te navesti prednosti i nedostatke. Zatim opisati način rada alarmnog sustava koji će biti izrađen i njegove osnovne komponente (procesor, ekran, senzore, napajanje). Nakon toga je potrebno izraditi i testirati rad sustava.

Pri tome rad treba sadržavati:

- opis koncepta IoT-a sustava
- opis i tehničke podatke korištene opreme
- sheme spajanja elemenata sustava
- prikaz postupka izrade i spajanja pojedinih dijelova u sustav
- opis izrađenih softverskih aplikacija
- rezultate testiranja realiziranog alarmnog sustava.

ZADATAK URUČEN

01.07. 2024.



POTPIS MENTORA

Josip Srpak

## **Predgovor**

Zahvaljujem se mentoru i predavaču, dipl. ing. Josipu Srpaku, na pomoći i savjetima prilikom izrade završnog rada. Zahvaljujem se i svim profesorima Sveučilišta Sjever na prenijetom znanju koje je pridonijelo izradi ovog završnog rada. Na kraju, zahvaljujem se svojoj obitelji koja me podržavala tijekom izrade ovog rada kao i tijekom cijelog akademskog obrazovanja.

## Sažetak

U ovom završnom radu opisan je postupak dizajniranja, izrade i implementacije alarmnog sustava temeljenog na ESP32 mikroupravljaču. Sustav je opremljen raznim sensorima i komponentama, uključujući zaslon, tipkovnicu, buzzer, temperaturni senzor, RFID modul, napajanje i reed relej. Cilj projekta bio je stvoriti funkcionalan i pouzdan alarmni sustav koji se može daljinski upravljati putem mobilne aplikacije Blynk. Rad započinje opisom IoT tehnologija te teoretskim pregledom komponenti sustava. Praktični dio rada fokusira se na proces povezivanja i programiranja ESP32 mikroupravljača. Detaljno su opisani koraci povezivanja svih komponenti. Poseban dio rada posvećen je izradi i sklapanju kućišta. Kućište je dizajnirano u programu SketchUp i izrađeno od MDF materijala, čime je osigurana zaštita komponenata i kompaktan sustav. U završnom dijelu rada opisana je implementacija Blynk aplikacije, koja omogućuje daljinsko upravljanje i praćenje sustava putem mobilnog uređaja.

Ključne riječi: alarmni sustav, ESP32 mikroupravljač, IoT

## **Summary**

This thesis describes the process of designing, creating and implementing an alarm system based on the ESP32 microcontroller. The system is equipped with various sensors and components, including a display, keypad, buzzer, temperature sensor, RFID module, power supply, and reed relay. The goal of the project was to create a functional and reliable alarm system that can be remotely controlled via the Blynk mobile application. The thesis begins with a description of IoT technologies and a theoretical overview of the system components. The practical part of the thesis focuses on the process of connecting and programming the ESP32 microcontroller. The steps for connecting all components are described in detail. A special part of the thesis is dedicated to the design and assembly of the enclosure. The enclosure was designed using SketchUp and made from MDF material, ensuring the protection of components and a compact system. The final part of the thesis describes the implementation of the Blynk application, which allows remote control and monitoring of the system via a mobile device.

Key words: alarm system, ESP32 microcontroller, IoT



## Popis korištenih kratica

<b>ADC</b>	Analog - Digital Converter - Analogno digitalni pretvornik
<b>CPU</b>	Central Processing Unit - Centralna procesorska jedinica
<b>Flash</b>	Flash memory – Vrsta elektroničke memorije
<b>GND</b>	Ground - Naponska točka od 0 V
<b>HTTP</b>	Hyper Text Transfer Protocol – Protokol za komunikaciju klijent-server
<b>I2C</b>	Inter-Integrated Circuit – serijsko komunikacijski protokol
<b>IOT</b>	Internet of Things - Internet stvari
<b>IP</b>	Internet protokol
<b>MQTT</b>	Message Queue Telemetry Transport - Komunikacijski protokol
<b>OLED</b>	Organic Light-Emitting Diode - Organske svjetleće diode
<b>PWM</b>	Pulse Width Modulation - Pulsno širinska modulacija
<b>RFID</b>	Radio-frequency identification -Radio frekvencijska identifikacija
<b>SPI</b>	Serial Peripheral Interface - Sučelje za serijsku komunikaciju podataka
<b>UID</b>	Unique Identifier – Jedinstveni identifikacijski broj
<b>USB</b>	Universal Serial Bus - Univerzalna serijska sabirnica
<b>Wi-Fi</b>	Wireless Fidelity - Bežični mrežni protokol

# Sadržaj

1.	Uvod .....	1
2.	IoT (Internet of Things) – „Internet stvari“ .....	2
2.1.	Definicija .....	2
2.2.	Ključne komponente IoT Sustava.....	2
2.3.	Primjena.....	3
2.4.	Prednosti .....	4
2.5.	Nedostatci .....	4
3.	Opis rada sustava .....	6
4.	Elementi alarmnog sustava .....	8
4.1.	ESP32.....	8
4.2.	RFID RC522 modul .....	10
4.3.	Reed relej.....	12
4.4.	OLED zaslon .....	13
4.5.	Matrična tipkovnica .....	15
4.6.	Buzzer.....	17
4.7.	DHT11 senzor temperature i vlage.....	19
4.8.	Modul za napajanje .....	20
4.9.	LED diode .....	23
5.	Povezivanje i programiranje sustava .....	25
5.1.	Shema spajanja .....	25
5.2.	Razvojno okruženje i alati .....	28
5.3.	Implementacija funkcionalnosti.....	29
5.3.1.	Detekcija otvaranja vrata.....	29
5.3.2.	Aktivacija alarma.....	30
5.3.3.	Deaktivacija alarma.....	30
5.3.4.	Očitavanje kartica pomoću RFID.....	32
5.3.5.	Postavljanje i promjena lozinke.....	34
5.3.6.	Prikaz potrošnje baterije.....	37
5.3.7.	Web server.....	38
5.3.8.	Praćenje promjene temperature i vlage .....	39
6.	Integracija s mobilnom aplikacijom Blynk Iot Cloud .....	41
6.1.	Uvod u Blynk.....	41
6.2.	Konfiguracija aplikacije .....	41
7.	Izrada i sklapanje kućišta .....	47
8.	Zaključak.....	49
9.	Literatura.....	50
10.	Popis slika .....	52
11.	Popis programskih blokova i tablica .....	53

# 1. Uvod

U današnje vrijeme sve veći naglasak stavlja se na sigurnost i zaštitu imovine, kako u domaćinstvima tako i u poslovnim prostorima. U skladu s tim, razvoj naprednih tehnoloških rješenja postaje neizbježan. U ovom radu istražuje se implementacija, odnosno primjena alarmnog sustava s mikroupravljač ESP32, koji kombinira pouzdanost, fleksibilnost i jednostavnost upotrebe.

Cilj ovog završnog rada je izrada i evaluacija alarmnog sustava temeljenog na mikroupravljaču ESP32. ESP32 je moćan i svestran mikroupravljač s integriranom Wi-Fi i Bluetooth podrškom, što ga čini idealnim za razvoj IoT (Internet of Things) aplikacija. Zbog svojih značajki, ESP32 omogućava jednostavnu integraciju s raznim senzorima i aktuatorima, čime se povećava funkcionalnost i pouzdanost alarmnog sustava.

Razvitkom tehnologije susrećemo se s pojmom „pametne kuće”. Radi se o kući u kojoj korisnici imaju mogućnost nadzora, prilagođavanja i podešavanja funkcionalnosti prostora kao što je kontrola rasvjete, upravljanje klimatizacijom i grijanjem. Sustav koji bi nam sve to omogućio treba se sastojati od podsustava od kojih je jedan alarmni sustav. Namjena alarmnog podsustava je zaštita korisnika objekta od neželjenog ulaska ili provale. Takvim načinom korisniku se pruža sigurnost tijekom boravka u određenom objektu ili nadzor stanja sustava tijekom odsutnosti.

U ovom završnom radu korištene su sljedeće komponente: reed relej za detekciju otvaranja vrata, OLED zaslon za prikaz informacija, tipkovnica za unos podataka, zvučnik za zvučno upozorenje, DHT11 senzor za praćenje temperature i vlage prostorije, RFID čitač za deaktivaciju, LED diode za vizualne indikatore te modul napajanja s Li-ion 18650 baterijom kapaciteta 3300mAh za autonomno napajanje sustava. Sustav je dizajniran tako da se može upravljati i nadzirati putem Blynk IoT aplikacije koja omogućava daljinsko aktiviranje i deaktiviranje alarma te praćenje temperature i vlage.

## 2. IoT (Internet of Things) – „Internet stvari“

### 2.1. Definicija

IoT (eng. Internet of Things, Iot) predstavlja mrežu fizičkih uređaja koji s ugrađenom elektronikom, softverom, sensorima i aktuatorima koja im omogućuje povezivanje i razmjenu podataka. IoT omogućuje prikupljanje i razmjenu podataka između različitih uređaja pomoću interneta, omogućujući im da djeluju autonomno ili uz minimalnu ljudsku intervenciju. IoT predstavlja revoluciju u načinu na koji komuniciramo s našim uređajima i upravljamo podacima te omogućuje pametnija i učinkovitija rješenja u gotovo svim aspektima života ljudi [2].

### 2.2. Ključne komponente IoT Sustava

#### **Senzori i aktuatori:**

- Senzori prikupljaju podatke iz okoline (npr. temperatura, vlaga, svjetlost) i šalju ih na obradu
- Aktuatori izvode fizičke akcije na temelju dobivenih uputa (npr. uključivanje svjetla, podešavanje termostata)

#### **Uređaji i mikroupravljači:**

- Uređaji poput ESP32 mikroupravljača prikupljaju podatke sa senzora, obrađuju ih i komuniciraju s drugim uređajima ili centralnim sustavima

#### **Komunikacijski protokoli:**

- IoT uređaji koriste različite protokole za komunikaciju, kao što su Wi-Fi, Bluetooth, Zigbee, MQTT i HTTP

#### **Oblak i poslužitelj:**

- Podaci prikupljeni od uređaja često se šalju u oblak gdje se dalje pohranjuju, analiziraju i koriste za donošenje odluka

#### **Korisničko sučelje:**

- Korisnici mogu upravljati IoT uređajima i pregledavati podatke putem aplikacija na pametnim telefonima, računalima ili tabletima

## 2.3. Primjena

### Pametni domovi:

- Uređaji poput pametnih termostata, sigurnosnih kamera i pametnih žarulja omogućuju automatizaciju i daljinsko upravljanje kućanskim aparatima

### Industrijska automatizacija:

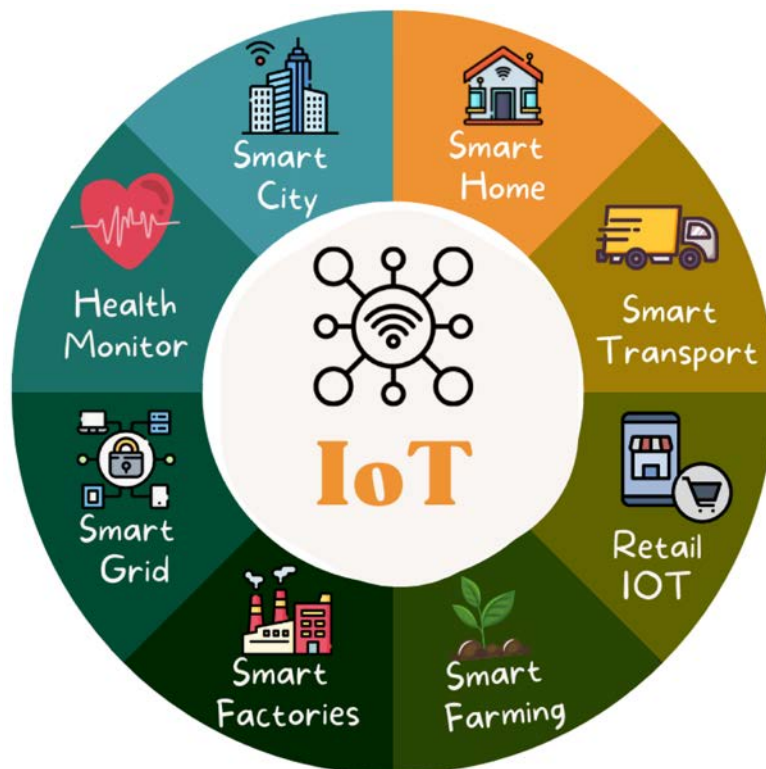
- IoT se koristi za praćenje i optimizaciju industrijskih procesa, smanjenje troškova održavanja i poboljšanje učinkovitosti proizvodnje

### Zdravstvo:

- Nosivi uređaji i medicinski senzori prate vitalne znakove pacijenata i šalju podatke liječnicima u stvarnom vremenu

### Pametni gradovi:

- IoT tehnologije omogućuju optimizaciju resursa poput vode i energije, poboljšanje sigurnosti i upravljanje prometom u urbanim sredinama



Slika 2.1 Primjena IoT [2]

## 2.4. Prednosti

### **Povećana učinkovitost i produktivnost:**

- IoT sustavi omogućuju automatizaciju zadatka i optimizaciju procesa, što rezultira povećanom produktivnošću i učinkovitosti

### **Bolje donošenje odluka:**

- Prikupljanje i analiza podataka u stvarnom vremenu omogućuju bolje i brže donošenje odluka

### **Povećana sigurnost:**

- IoT uređaji poput pametnih sigurnosnih kamera i senzora mogu poboljšati sigurnost domova i poslovnih objekata

### **Ušteda energije:**

- Pametni uređaji mogu automatski prilagoditi potrošnju energije na temelju trenutnih uvjeta, čime se smanjuje nepotrebna potrošnja energije

### **Personalizacija:**

- IoT omogućuje prilagođavanje uređaja i usluga prema potrebama i preferencijama korisnika

## 2.5. Nedostatci

### **Sigurnosni rizici:**

- IoT uređaji su često meta napada zbog njihove povezanosti s internetom, što može dovesti do krađe podataka i drugih sigurnosnih incidenata

### **Privatnost:**

- Prikupljanje velikih količina osobnih podataka može ugroziti privatnost korisnika

**Kompleksnost:**

- Postavljanje i održavanje IoT sustava može biti složeno zbog različitih protokola, uređaja i standarda

**Visoki troškovi:**

- Početni troškovi implementacije IoT rješenja mogu biti visoki, što može predstavljati prepreku za manja poduzeća i korisnike

**Ovisnost o internetu:**

- IoT uređaji ovise o stabilnoj internetskoj vezi, a gubitak veze može uzrokovati prekid rada sustava

### 3. Opis rada sustava

U ovom poglavlju opisan je rad izrađenog alarmnog sustava. Sustav se uključi tako da se pritisne tipka na modulu za napajanje koji se nalazi unutar kućišta sustava. Nakon toga se ispisuje na zaslonu „Alarmni sustav“ te IP adresa web servera na koji se korisnik može povezati putem preglednika za dodatni uvid u temperaturu, vlagu prostorije te postotak i napon baterije.



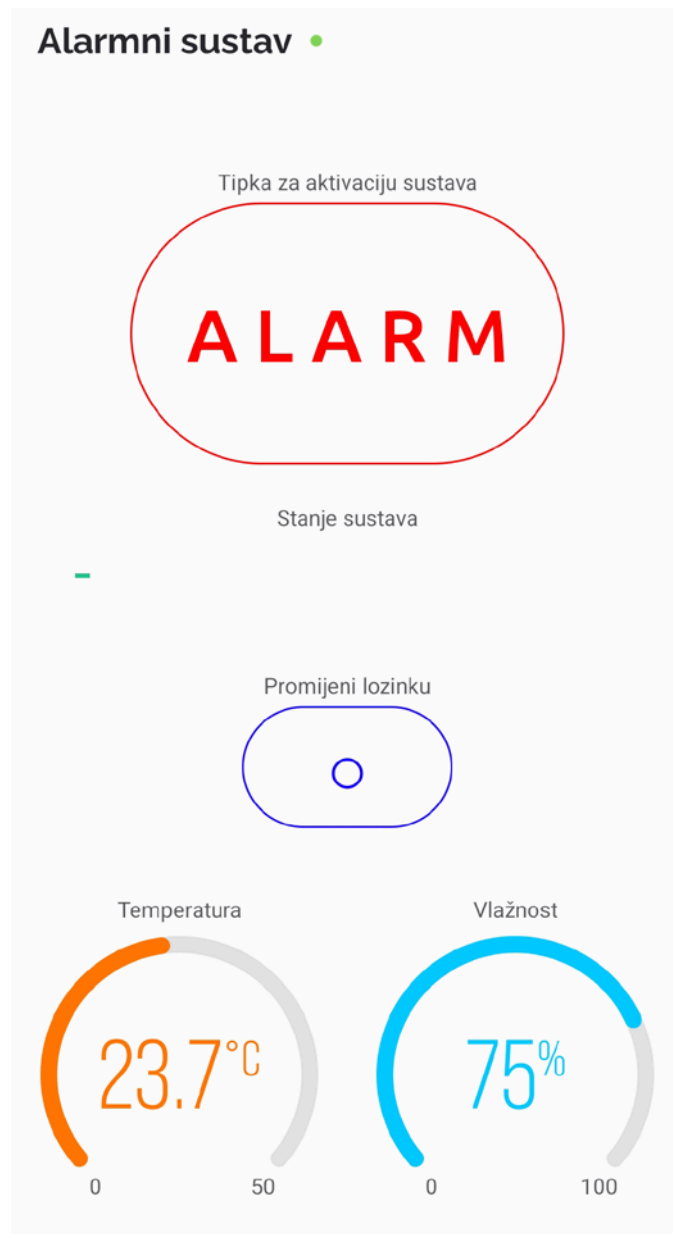
*Slika 3.1 Prikaz početnog stanja sustava*

Korisnik može aktivirati sustav pomoću Blynk IoT aplikacije s bilo kojeg mjesta gdje ima pristup internetu ili aktivacija je moguća pritiskom tipke „A“ na tipkovnici sustava prilikom odlaska od kuće. Pritiskom tipke „A“ alarm će se aktivirati za 10 sekundi tako da korisnik ima vremena otvoriti i zatvoriti ulazna vrata. Kada je sustav aktivan, pomoću reed releja prate se ulazna vrata.



*Slika 3.2 Prikaz aktivnog sustava*





*Slika 3.3 Izgled mobilne aplikacije*

Ukoliko netko otvori vrata ima 8 sekundi za skeniranje kartice ili unos točnog pina. Ukoliko se ne očita točna kartica ili ne unese točan pin, sustav pokreće zvučnu signalizaciju i šalje obavijest na mobilnu aplikaciju koja također uključuje alarmni zvuk na mobilnom uređaju. Alarm se može ugasiti unosom pina ili skeniranjem kartice. Mobilna aplikacija ima izrađeno sučelje gdje se može pomoću tipke aktivirati alarm, pratiti temperatura i vlaga prostorije u zadnjih sedam dana te aktivirati promjena lozinke.

## 4. Elementi alarmnog sustava

### 4.1. ESP32

#### Općenito

ESP32 (slika 4.1) je svestrani mikroupravljač razvijen od strane Espressif Systems, koji je posebno dizajniran za IoT aplikacije. Ovaj mikroupravljač kombinira visoke performanse, nisku potrošnju energije i široku podršku za različite komunikacijske protokole, što ga čini idealnim za razne projekte u području elektronike i automatizacije.



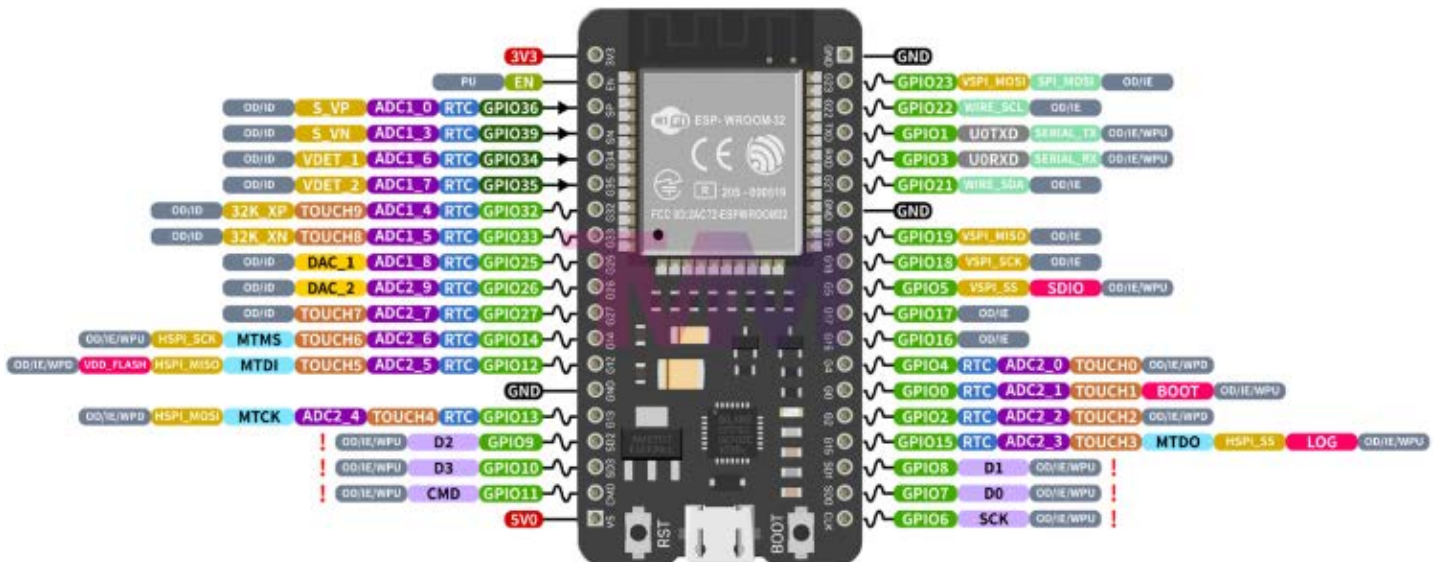
*Slika 4.1 ESP32 WROOM-32*

#### Specifikacije [1]:

- Procesor: Dual-core Tensilica LX6 mikroprocesor s radnim taktovima do 240 MHz
- Memorija: 520 KB SRAM, 448 KB ROM
- Pohrana: Integrirani 4 MB Flash
- Komunikacija: Wi-Fi 802.11 b/g/n, Bluetooth 4.2 (BLE)
- Ulazno-izlazni pinovi: 34 GPIO pinova
- ADC/DAC: 18 ADC kanala (12-bit), 2 DAC kanala (8-bit)
- PWM: 16 PWM kanala
- Komunikacijski protokoli: SPI, I2C, UART, I2S
- Napajanje: 2.2V do 3.6V

## ESP32 raspored pinova

ESP32 pinovi su kategorizirani u digitalne pinove, analogne pinove i pinove za napajanje. Na slici 4.2 prikazane su funkcije pinova, također su prikazane sekundarne funkcije koje često služe u komunikacijske svrhe kao što su I2C i SPI ili kao ADC kanali.



Slika 4.2 Raspored pinova ESP32 WROOM modela [1]

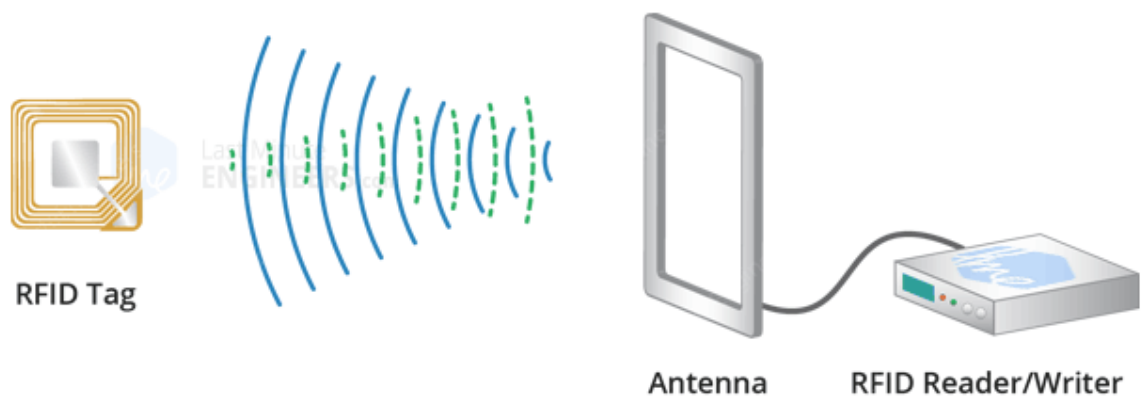
## Prednosti ESP32

- **Integracija Wi-Fi i Bluetooth modula:** ESP32 ima ugrađene Wi-Fi i Bluetooth module, što omogućuje jednostavno povezivanje s mrežama i drugim uređajima, čime se smanjuje potreba za dodatnim modulima
- **Visoke performanse:** Dvojezgreni (eng. Dual core) procesor omogućuje brzu obradu podataka i izvršavanje kompleksnih zadataka što je ključno za zahtjevne IoT aplikacije
- **Niska potrošnja energije:** ESP32 podržava različite režime niske potrošnje energije što je bitno za baterijsko napajanje sustava
- **Velik broj ulazno-izlaznih pinova:** s 34 GPIO pinova, ESP32 omogućuje povezivanje velikog broja senzora i aktuatora

- **Širok raspon radne temperature:** Pogodan je za rad u teškim uvjetima čime se povećava njegova primjenjivost u različitim okruženjima
- **Podrška za različite komunikacijske protokole:** SPI, I2C, UART i I2S omogućuju fleksibilnost u povezivanju različitih perifernih uređaja

## 4.2. RFID RC522 modul

RFID (Radio Frequency Identification) tehnologija koristi elektromagnetska polja za automatsku identifikaciju i praćenje oznaka (tagova) pričvršćenih na objekte. RFID se sastoji od dvije glavne komponente (slika 4.3), oznake pričvršćene na objekt koji treba identificirati i čitača koji čita oznaku. Čitač se sastoji od radiofrekvencijskog modula i antene koja stvara visokofrekventno elektromagnetsko polje. Oznaka je obično pasivni uređaj (nema bateriju). Sastoji se od mikročipa koji pohranjuje i obrađuje informacije te antene za primanje i odašiljanje signala. Kada se oznaka približi čitaču, čitač stvara elektromagnetsko polje. To uzrokuje kretanje elektrona kroz antenu oznake i potom napaja čip [4].



Slika 4.3 Prikaz rada RFID modula [4]

## **Specifikacije RFID RC522 modula [4]**

RFID RC522 je čitač i pisac RFID kartica koji radi na frekvenciji od 13,56 MHz. Na slici 4.4 prikazan čitač. Njegove ključne specifikacije su:

- **Radna frekvencija:** 13,56 MHz
- **Komunikacijski protokol:** SPI (Serial Peripheral Interface)
- **Domet čitanja:** Do 5 cm (ovisno o veličini i tipu RFID oznake)
- **Podržane oznake:** ISO/IEC 14443 Type A
- **Napajanje:** 2,5V do 3,3V
- **Potrošnja energije:** Maksimalno 13-26 mA pri aktivnom radu
- **Dimenzije:** 40 mm x 60 mm

## **Integracija RFID RC522 modula s ESP32**

Integracija RFID RC522 modula s ESP32 mikroupravljačem omogućuje jednostavno čitanje i pisanje RFID oznaka. SPI sučelje omogućava brzu komunikaciju između modula i mikroupravljača. U ovom projektu, RFID RC522 modul je korišten za autentifikaciju korisnika pomoću RFID kartica ili oznaka.



*Slika 4.4 RFID RC522 modul [4]*

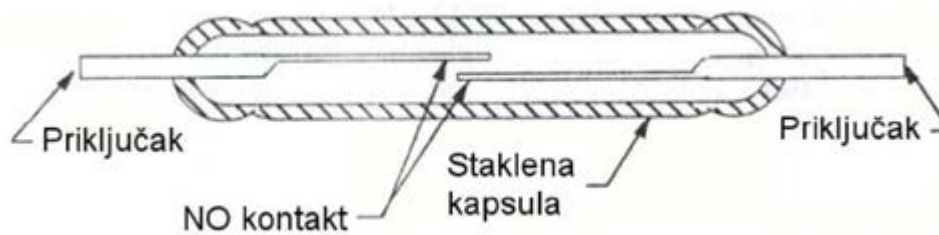
### 4.3. Reed relej

Reed relej je elektronička komponenta koja se koristi za detekciju prisutnosti magnetskog polja. U ovom projektu reed relej se koristi za detekciju otvorenosti vrata što je ključno za funkcionalnost alarmnog sustava. Reed releji su jednostavni, pouzdani i često korišteni u sigurnosnim sustavima [11].

#### Princip rada reed releja

Reed relej se sastoji od dva feromagnetska kontakta smještena unutar staklene kapsule što se vidi na slici 4.5. Kada se u blizini releja nađe magnet, magnetsko polje uzrokuje spajanje ili razdvajanje kontakata, ovisno o konfiguraciji releja (normalno otvoren ili normalno zatvoren) [5].

- **Normalno otvoren (NO):** Kontakti su razdvojeni u odsutnosti magnetskog polja. Kada je magnet prisutan, kontakti se spajaju.
- **Normalno zatvoren (NC):** Kontakti su spojeni u odsutnosti magnetskog polja. Kada je magnet prisutan, kontakti se razdvajaju.



Slika 4.5 Princip rada reed releja [5]

### Upotreba reed releja u projektu

U ovom projektu koristi se normalno otvoreni reed relej za detekciju otvorenosti vrata. Releji se postavljaju tako da magnet bude pričvršćen na vrata, a reed relej na okvir vrata. Kada su vrata zatvorena, magnet drži kontakte reed releja spojene. Kada se vrata otvore, kontakti se razdvajaju, što detektira ESP32 mikroupravljač

## 4.4. OLED zaslon

OLED (Organic Light Emitting Diode) zaslone su poznati po svojim svijetlim bojama, visokom kontrastu i širokim kutovima gledanja. Za razliku od LCD zaslona, OLED zaslone ne zahtijevaju pozadinsko osvjetljenje jer svaki piksel emitira vlastitu svjetlost. Ova tehnologija omogućuje tanje i energetski učinkovitije zaslone.

### Specifikacije 2.4-inčnog OLED zaslona [6]

2.4-inčni OLED zaslon (slika 4.6) koristi I2C sučelje za komunikaciju s mikroupravljačem, što znači da su mu potrebne dvije žice za komunikaciju uz napajanje. Slijede ključne specifikacije ovog zaslona:

- **Veličina zaslona:** 2.4 inča
- **Rezolucija:** 128x64 piksela
- **Sučelje:** I2C (Inter-Integrated Circuit)
- **Radni napon:** 3.3V
- **Potrošnja energije:** Niska, zahvaljujući OLED tehnologiji
- **Kontrast:** Visok, s mogućnošću prikazivanja pravih crnih boja

- **Kut gledanja:** Širok, tipično oko 160 stupnjeva
- **Vrijeme odziva:** Brzo, omogućujući fluidne animacije i brzo osvježavanje

### Prednosti OLED zaslona

1. **Visok kontrast i svjetlina:**
  - OLED zasloni mogu prikazati vrlo svijetle boje i savršene crne boje, što rezultira visokim kontrastom
2. **Širok kut gledanja:**
  - Kvaliteta slike ostaje dosljedna čak i kada se zaslon gleda iz oštrog kuta
3. **Energetska učinkovitost:**
  - OLED pikseli sami emitiraju svjetlost, eliminirajući potrebu za pozadinskim osvjetljenjem i smanjujući potrošnju energije
4. **Brzo vrijeme odziva:**
  - OLED zasloni imaju brzo vrijeme odziva, što omogućuje glatke prijelaze i animacije

### I2C komunikacija

OLED zaslon koji je korišten u onom završnom radu sadrži I2C (Inter-Integrated Circuit) komunikaciju. I2C (Inter-Integrated Circuit) je serijski komunikacijski protokol koji omogućuje povezivanje više uređaja s jednim ili više glavnih uređaja pomoću samo dvije žice: SCL (clock) i SDA (data). Ovaj protokol je razvijen od strane Philipsa (danas NXP Semiconductors) i koristi se za komunikaciju između mikroupravljača i perifernih uređaja kao što su senzori, zasloni, EEPROM-ovi, itd.



Slika 4.6 OLED I2C zaslon 2.4“ [6]



## 4.5. Matrična tipkovnica

Matrična tipkovnica, posebno ona s konfiguracijom 4x4, je široko korištena komponenta za unos podataka u različitim elektroničkim projektima. Ova tipkovnica omogućuje korisnicima unos podataka poput šifri, kontrolnih naredbi i drugih interaktivnih unosa.

### Specifikacije 4x4 matrične tipkovnice

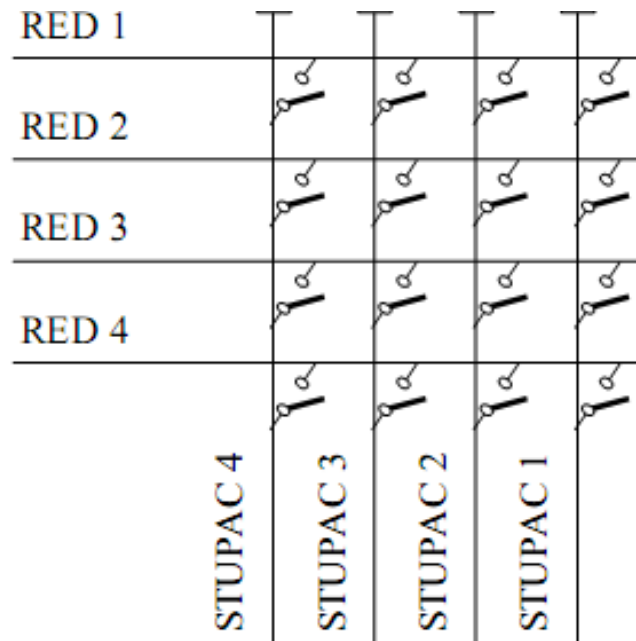
4x4 matrična tipkovnica (slika 4.8) se sastoji od 16 tipki raspoređenih u četiri reda i četiri stupca. Tipkovnica koristi matricu sklopki za povezivanje tipki sa mikroupravljačem.

- **Broj tipki:** 16
- **Konfiguracija:** 4 reda x 4 stupca
- **Tipke:** Numeričke (0-9) i posebne znakove (A-D, \*, #)
- **Radni napon:** 3.3V do 5V
- **Dimenzije:** 69 mm x 69 mm
- **Povezivanje:** 8 pinova (4 reda + 4 stupca)

### Princip rada 4x4 matrične tipkovnice

Za spajanje tipki na ulaz nekog računalnog sustava potrebna nam je jedna ulazna varijabla za svaku tipku. Ako imamo veći broj tipki kao što je to na današnjim tipkovnicama, tipke su spojene u matricu. Pomoću matrice koristi se manji broj ulaza. Primjer spajanja 16 tipaka u matričnu tipkovnicu prikazan je na slici 4.7. U ovom radu korištena je tipkovnica sa 16 tipaka koje se spajaju na način da se redovi spoje na izlaz, a stupci na ulaz. Očitavanje pritisnute tipke vrši se na sljedeći način:

- na prvi red tipkovnice dovede se logička jedinica, a na ostale redove nula
- čita se jedan po jedan stupac
- ako je pritisnuta neka od tipki iz prvog reda na odgovarajućem stupcu pročitat će se 1, a u ostalim stupcima 0
- ako nije pritisnuta nijedna tipka u ovom redu, pročitat će se sve 0
- nakon pregleda prvog reda, dovodi se 1 na drugi red i nula na sve ostale redove te se ponavlja čitanje stupaca, sve dok se ne pročitaju svi redovi



*Slika 4.7 Princip rada matrične tipkovnice*

### **Prednosti 4x4 matrične tipkovnice**

#### **1. Jednostavnost korištenja:**

- Tipkovnica je jednostavna za integraciju s mikroupravljačima i omogućuje intuitivan unos podataka

#### **2. Mali broj pinova:**

- Iako ima 16 tipki, tipkovnica koristi samo 8 pinova za povezivanje, što štedi GPIO pinove na mikroupravljaču

#### **3. Izdržljivost:**

- Matrične tipkovnice su dizajnirane da izdrže dugotrajno korištenje i česte pritiske tipki

#### **4. Raznovrsnost primjene:**

- Tipkovnica se može koristiti za unos šifri, kontrolu uređaja, navigaciju kroz izbornike i mnoge druge interaktivne aplikacije



*Slika 4.8 Matrična tipkovnica 4x4 [7]*

## 4.6. Buzzer

Buzzer je elektronička komponenta koja proizvodi zvuk kada se napaja električnom strujom. Koristi se u raznim aplikacijama za pružanje zvučnih upozorenja ili povratnih informacija korisnicima. U ovom projektu, buzzer se koristi za zvučnu signalizaciju alarma. Izgled korištenog buzzera prikazan je na slici 4.9.

### Specifikacije buzzera [8]

Buzzer koji u ovom sustavu ima promjer od 23 mm i može raditi u širokom rasponu napona. Njegove ključne specifikacije su:

- **Promjer:** 23 mm
- **Radni napon:** 3V do 24V
- **Vrsta:** Aktivni buzzer (ugrađeni oscilator)
- **Frekvencija:** 2 kHz
- **Struja:** Mala potrošnja, obično nekoliko miliampera
- **Montaža:** Može se lako integrirati u kućišta ili PCB pločice

## Prednosti buzzera

### 1. Širok raspon napona:

- Mogućnost rada na naponu od 3V do 24V čini buzzer fleksibilnim za korištenje u različitim aplikacijama

### 2. Jednostavna integracija:

- Buzzer se lako integrira s mikroupravljačima i drugim elektroničkim komponentama, zahtijevajući samo jednostavno napajanje

### 3. Mala potrošnja energije:

- Buzzer troši vrlo malo struje, što je pogodno za baterijski napajane sustave

### 4. Snažan zvuk:

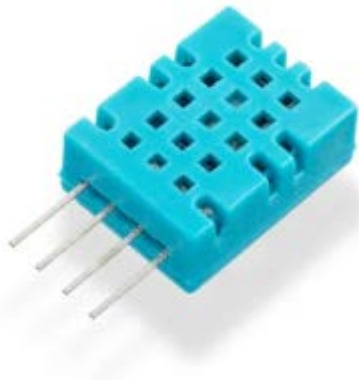
- Unatoč maloj veličini, buzzer može proizvesti dovoljno glasan zvuk za upozoravanje korisnika



*Slika 4.9 Buzzer 3-24V [8]*

## 4.7. DHT11 senzor temperature i vlage

DHT11 je popularan senzor koji može mjeriti temperaturu i relativnu vlažnost zraka. Prikazan je na slici 4.10. Koristi se u raznim aplikacijama, uključujući sustave za praćenje okoliša, pametne domove i uređaje za automatizaciju. U ovom sustavu, DHT11 senzor omogućuje praćenje temperaturnih uvjeta, što može biti korisno za dodatne sigurnosne značajke. DHT11 prati temperaturu i vlažnost te ukoliko se desi nagla promjena korisnik će primiti obavijest o promjeni temperature ili vlage u prostoriji [9].



*Slika 4.10 DHT11 [9]*

### Specifikacije DHT11 senzora [9]

DHT11 senzor je poznat po svojoj jednostavnosti korištenja i niskoj cijeni. Ključne specifikacije su:

- **Raspon mjerenja temperature:** 0°C do 50°C
- **Raspon mjerenja vlažnosti:** 20% do 90% RH
- **Radni napon:** 3.3V do 5V
- **Vrijeme odziva:** 5-10 sekundi
- **Izlazni signal:** Digitalni
- **Dimenzije:** 15.5mm x 12mm x 5.5mm

## **Prednosti DHT11 senzora**

### **1. Jednostavna implementacija:**

- Senzor je lako integrirati u projekte pomoću popularnih mikroupravljača kao što su ESP32

### **2. Niska cijena:**

- DHT11 je jedan od najjeftinijih senzora za mjerenje temperature i vlage

### **3. Digitalni izlaz:**

- Omogućuje jednostavno očitavanje podataka preko jednog digitalnog pina

### **4. Niska potrošnja energije:**

- Senzor troši vrlo malo energije što je pogodno za baterijski napajane sustave

## **Integracija DHT11 senzora s ESP32**

Senzor koristi jedan digitalni pin za komunikaciju, a dostupne su i razne biblioteke (eng. library) koje pojednostavljuju proces očitavanja podataka. U Arduino IDE-u se koristi DHT biblioteka za upravljanje DHT11 senzorom. Instalacija ove knjižnice može se obaviti putem Library Managera.

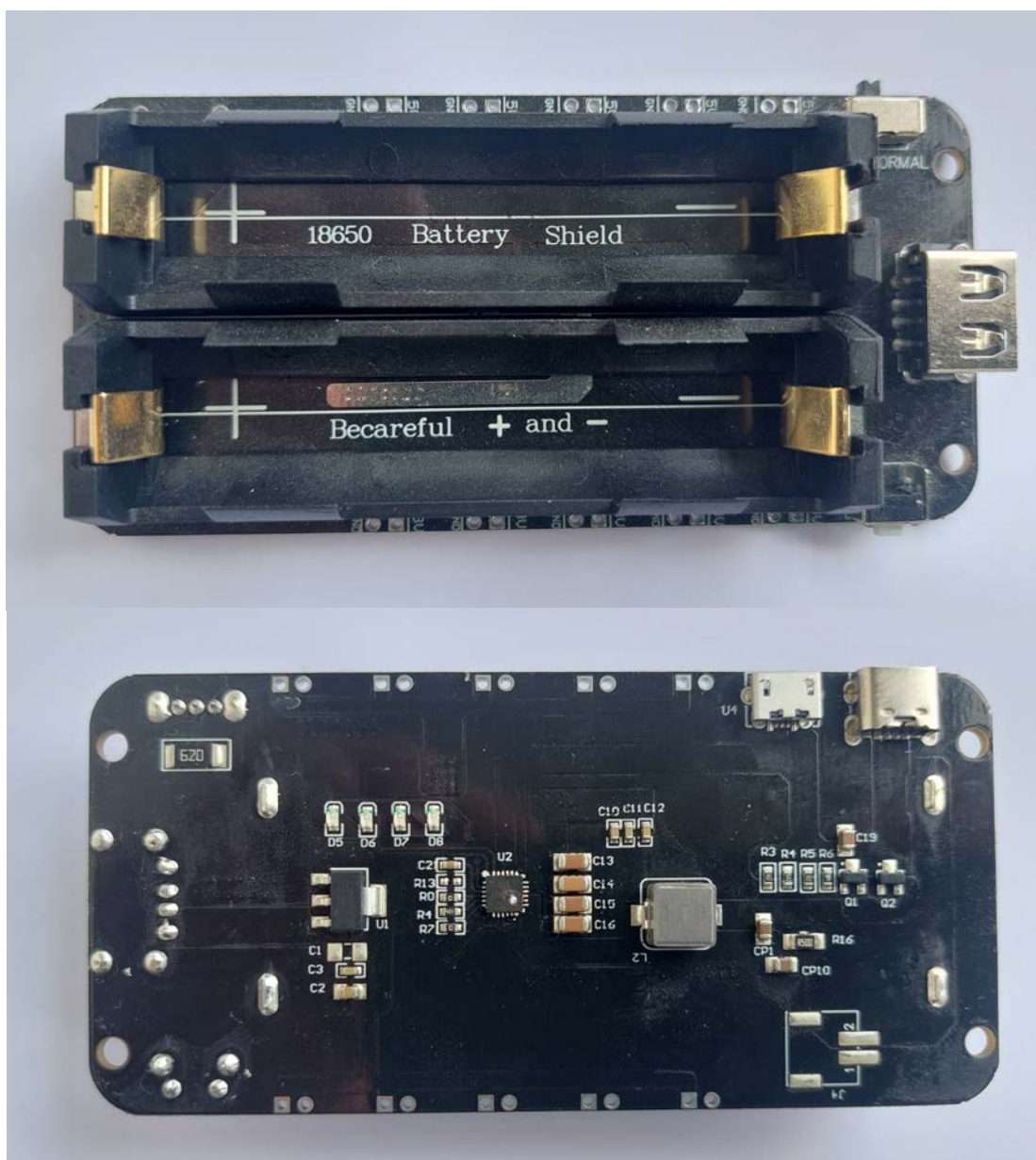
## **4.8. Modul za napajanje**

Modul za napajanje (slika 4.11) dizajniran je za punjenje i upravljanje litij-ionskim baterijama tipa 18650. Ovaj modul omogućuje napajanje mikroupravljačkog sustava, pružajući stabilno napajanje i upravljanje punjenjem baterija.

### **Specifikacije**

- **Podrška za baterije:** 2x 18650 litij-ionske baterije
- **Ulazni napon:** 5V (micro USB)
- **Izlazni napon:** 5V

- **Izlazna struja:** Maksimalno 1A (ovisno o stanju baterije)
- **Zaštita od prepunjenja:** Da
- **Zaštita od prekomjernog pražnjenja:** Da
- **Zaštita od kratkog spoja:** Da
- **Dimenzije:** 85mm x 35mm



*Slika 4.11 Modul za napajanje*

## **Prednosti i nedostaci modula**

### **Prednosti:**

#### **1. Sigurnost:**

- Ugrađene zaštite osiguravaju sigurno punjenje i korištenje baterija smanjujući rizik od oštećenja

#### **2. Praktičnost:**

- Punjenje putem USB-a omogućuje jednostavno punjenje bez potrebe za dodatnim punjačima

#### **3. Dugotrajno napajanje:**

- Koristeći dvije 18650 baterije, ploča može osigurati dugotrajno napajanje za vaš projekt što je posebno korisno za baterijski napajane uređaje

### **Nedostaci:**

#### **1. Veličina:**

- Ploča zajedno s dvije 18650 baterije može zauzimati više prostora što može biti izazov u kompaktnim projektima

#### **2. Ograničena izlazna struja:**

- Izlazna struja je ograničena na 1A što može biti nedovoljno za projekte s većom potrošnjom

#### **3. Ovisnost o kvaliteti baterija:**

- Performanse ploče značajno ovise o kvaliteti i stanju 18650 baterija koje se koriste



## 4.9. LED diode

LED diode (Light Emitting Diodes) su poluvodičke komponente koje emitiraju svjetlost kada kroz njih prolazi električna struja. U ovom alarmnom sustavu korištene su crvena i zelena LED dioda za vizualnu indikaciju različitih stanja sustava, kao što su aktivacija i deaktivacija alarma [10].

### Prednosti LED dioda

1. **Niska potrošnja energije:**

- LED diode troše vrlo malo energije što ih čini idealnim za baterijski napajane uređaje

2. **Dug životni vijek:**

- LED diode imaju dug životni vijek često iznad 50.000 sati

3. **Brzo uključivanje/isključivanje:**

- LED diode se uključuju i isključuju vrlo brzo što omogućava korištenje u aplikacijama koje zahtijevaju brze promjene stanja

4. **Visoka svjetlosna efikasnost:**

- LED diode emitiraju svjetlost visoke efikasnosti uz minimalno zagrijavanje

### Integracija LED dioda s alarmnim sustavom

LED diode su integrirane u kućište alarmnog sustava tako da su jasno vidljive korisnicima. Na slici 4.12 prikazana je crvena LED dioda kakva je korištena za ovaj sustav. Njihovo svijetljenje ili treptanje može se programirati za različite indikacije, kao što su:

- **Kontinuirano svijetljenje crvene LED diode:** Alarm je aktiviran.
- **Kontinuirano svijetljenje zelene LED diode:** Alarm je deaktiviran.
- **Treptanje zelene LED diode:** Uspješna autentifikacija ili unos šifre.

Ova vizualna indikacija poboljšava korisničko iskustvo i pruža brz i jasan pregled statusa sustava.



*Slika 4.12 LED dioda [10]*

### **Integracija LED dioda s ESP32**

Povezivanje LED dioda s ESP32 mikroupravljačem je jednostavno. LED diode se mogu kontrolirati pomoću digitalnih pinova na ESP32. Preporuča se korištenje otpornika za ograničavanje struje kroz LED diode kako bi se spriječilo njihovo oštećenje.

#### **Povezivanje:**

- **Katoda:** Povezati na GND (uzemljenje)
- **Anoda:** Povezati na digitalni pin na ESP32 preko otpornika (npr.  $220\Omega$ )

## **5. Povezivanje i programiranje sustava**

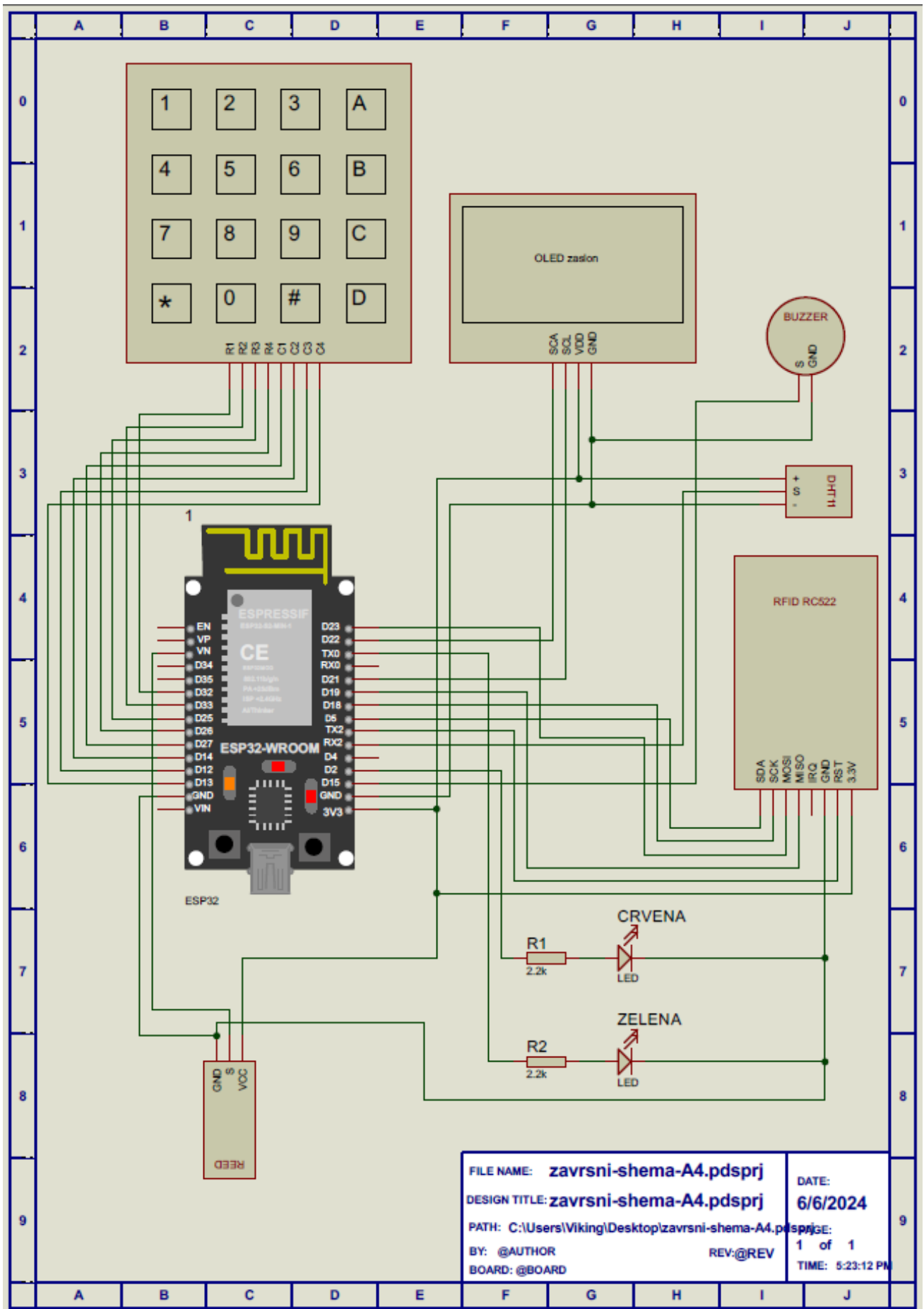
U ovom poglavlju detaljno ćemo opisati proces povezivanja svih komponenti s ESP32 mikroupravljačem te objasniti kako ih programirati za uspješno upravljanje alarmnim sustavom. ESP32 je centralni dio sustava, zadužen za prikupljanje podataka iz senzora, obradu tih podataka i komunikaciju s korisnikom putem Blynk aplikacije.

### **5.1. Shema spajanja**

U tablici 1. prikazano je detaljno spajanje svih pinova komponenti s pinovima mikroupravljača ESP32. Nakon tablice prikazana je shema spajanja koja je izrađena u programu Proteus Professional 8. Pridržavanje ove sheme osigurava pravilno funkcioniranje sustava i omogućuje lako praćenje spojeva tijekom izrade.

Tablica 5.1 Spajanje pinova komponenti na pinove ESP32

OLED I2C zaslon	VCC (Napajanje)	3.3V
	GND (Uzemljenje)	GND
	SCL (Serial Clock Line)	GPIO22
	SDA (Serial Data Line)	GPIO21
Reed relej	VCC	3.3V
	SIGNAL	GPIO36
	GND (uzemljenje)	GND
Tipkovnica	R1	GPIO32
	R2	GPIO33
	R3	GPIO25
	R4	GPIO26
	C1	GPIO27
	C2	GPIO14
	C3	GPIO12
	C4	GPIO13
Buzzer	GND	GND
	SIGNAL	GPIO15
DHT11	VCC	3.3V
	GND	GND
	DATA	GPIO16
RFID RC522	SDA	GPIO5
	SCK	GPIO18
	MOSI	GPIO23
	MISO	GPIO19
	IRQ	-
	GND	GND
	RST	GPIO0
	3.3V	3.3V
LED CRVENA	KATODA	GND
	ANODA	GPIO2
LED ZELENA	KATODA	GND
	ANODA	GPIO1



Slika 5.1 Shema spajanja

## 5.2. Razvojno okruženje i alati

Razvoj softverskog dijela alarmnog sustava zahtijeva korištenje specifičnih razvojnih alata i okruženja koja omogućuju programiranje, testiranje i implementaciju koda na ESP32 mikroupravljaču. U ovom poglavlju predstavljeno je razvojno okruženje i opisani su alati koji su korišteni u izradi ovog projekta.

### Arduino IDE

Arduino IDE (Integrated Development Environment) je glavno razvojno okruženje korišteno za programiranje ESP32 mikroupravljača. Arduino IDE je popularno među hobistima i profesionalcima zbog svoje jednostavnosti i pristupačnosti.

#### Ključne značajke Arduino IDE-a:

- **Podrška za više mikroupravljača:** Arduino IDE podržava širok spektar mikroupravljača, uključujući ESP32
- **Jednostavno korisničko sučelje:** Intuitivno sučelje omogućava brzo učenje i rad
- **Veliki broj biblioteka:** Postoji veliki broj ugrađenih i dodatnih biblioteka koje olakšavaju razvoj različitih projekata
- **Serijski monitor:** Alat za praćenje serijskih podataka koje mikroupravljača šalje

#### Instalacija i postavljanje:

1. **Preuzimanje i instalacija:** Arduino IDE se može preuzeti s službene Arduino stranice
2. **Postavljanje ESP32 podrške:**
  - Otvorite Arduino IDE i idite na File > Preferences.
  - U polje "Additional Boards Manager URLs" dodajte sljedeću URL adresu: [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json).
  - Idite na Tools > Board > Boards Manager, potražite "ESP32" i instalirajte "esp32 by Espressif Systems".

### 5.3. Implementacija funkcionalnosti

U ovom poglavlju detaljno će biti opisana implementacija glavnih funkcionalnosti alarmnog sustava s ESP32 mikroupravljačem. Implementacija je podijeljena na nekoliko ključnih dijelova, uključujući detekciju otvaranja vrata, aktivaciju alarma, očitavanje senzora temperature i vlage, korištenje RFID čitača, te upravljanje LED diodama i OLED zaslonom.

#### 5.3.1. Detekcija otvaranja vrata

Detekcija otvaranja vrata oslanja se na reed relej koji reagira na magnetsko polje. Kada su vrata zatvorena, magnet drži reed relej u određenom stanju (otvoreno ili zatvoreno). Kada se vrata otvore, stanje reed releja se mijenja što detektira ESP32. Kod programiranja prate se vrata tek kada je sustav aktivan. U programskom bloku 5.1 prikazan je kod koji se odnosi na detekciju otvaranja vrata kad je sustav aktivan.

```
if(sustavAktivan && !alarmAktivan)
{
  Serial.println("sustav aktivan");
  Blynk.virtualWrite(V5, "Sustav aktivan");
  digitalWrite(LEDc, HIGH);
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(25,10);
  display.println("Sustav");
  display.setCursor(20,32);
  display.println("Aktivan");
  display.drawRect(0, 0, 128, 64, WHITE);
  display.display();
  delay(50);
  if (vrata == 1 && sustavAktivan)
  {
    tone(buzz,1000);
    delay(200);
    noTone(buzz);
    delay(1000);
    provjeriRFID();

    if (vrata == 1 && sustavAktivan)
    {
      ukljuciAlarm();
    }
  }
}
```

*Programski blok 5.1 Arduino kod za detekciju otvaranja vrata*

### 5.3.2. Aktivacija alarma

Aktivacija alarma vrši se funkcijom „uključAlarm“ koja se aktivira ukoliko je sustav bio aktivan i netko je otvorio vrata. U programskom bloku 5.2 može se primijetiti da funkcija „uključAlarm“ aktivira buzzer, šalje obavijest aplikaciji i ispisuje ikonu na ekran.

```
void uključAlarm()  
{  
  alarmAktivan = true;  
  tone(buzz,1000);  
  Serial.println("Alarm!");  
  Blynk.virtualWrite(V5, "Alarm! Vrata otvorena!");  
  display.clearDisplay();  
  display.drawBitmap( 0, 0, alarmalarm_icon, 128, 64, WHITE);  
  display.display();  
  Blynk.logEvent("alarm", "Alarm! Vrata otvorena!");  
}
```

*Programski blok 5.2 Aktivacija alarma*

### 5.3.3. Deaktivacija alarma

Alarm se može deaktivirati pomoću skeniranja autorizirane kartice ili upisivanjem točnog pina. Skeniranjem autorizirane kartice izvršava se funkcija „StopAlarmRFID“ koja postavlja aktivnost sustava na 0, isključuje buzzer i led diodu te ispisuje na ekranu da je kartica prihvaćena. Upisivanjem točnog pina aktivnost sustava se postavlja na nulu, ukoliko je aktiviran alarm se isključuje i ispisuje se na zaslonu da je točan pin. U programskom bloku 5.3 prikazane su funkcije za deaktivaciju alarma.



```

void stopAlarmRFID()
{
    alarmAktivan = false;

    sustavAktivan = false;
    noTone(buzz);
    digitalWrite(LEDc, LOW);
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(20,10);
    display.println("kartica");
    display.setCursor(3,30);
    display.println("prihvacena");
    display.drawRect(0, 0, 128, 64, WHITE);
    display.display();
    digitalWrite(LEDz, HIGH);
    tone(buzz,1500);
    delay(100);
    noTone(buzz);
    delay(3000);
    digitalWrite(LEDz, LOW);
    display.clearDisplay();
    display.display();
}

void stopAlarmPassword() {
    sustavAktivan = false;
    alarmAktivan = false;
    noTone(buzz);
    digitalWrite(LEDc, LOW);
    digitalWrite(LEDz, HIGH);
    tone(buzz,1500);
    delay(100);
    noTone(buzz);
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(42,10);
    display.println("PIN");
    display.setCursor(13,30);
    display.println("prihvacen");
    display.drawRect(0, 0, 128, 64, WHITE);
    display.display();
    delay(3000);
    digitalWrite(LEDz, LOW);
    display.clearDisplay();
    display.display();
}

```

*Programski blok 5.3 Kod za deaktivaciju alarma pomoću tipkovnice i RFID modula*

### 5.3.4. Očitavanje kartica pomoću RFID

Za čitanje kartica korištena je biblioteka (*eng. library*) MFRC522.h. U ovom projektu, RFID čitač RC522 koristi se za očitavanje UID (Unique Identifier) brojeva RFID kartica ili tagova koji se koriste za autorizaciju i upravljanje alarmnim sustavom. Da bi se očitao UID broj kartice pomoću RFID čitača RC522, potrebno je koristiti biblioteku za RFID čitač koja omogućava komunikaciju između ESP32 i RFID čitača.

Kod za očitavanje UID broja kartice koristi MFRC522 biblioteku i omogućava ispis UID broja na serijski monitor. Kada se RFID kartica približi čitaču, njen UID broj će biti prikazan.

```
void loop() {  
  if (!rfid.PICC_IsNewCardPresent()) {  
    return;  
  }  
  if (!rfid.PICC_ReadCardSerial()) {  
    return;  
  }  
  
  Serial.print("UID kartice: ");  
  for (byte i = 0; i < rfid.uid.size; i++) {  
    Serial.print(rfid.uid.uidByte[i] < 0x10 ? " 0" : " ");  
    Serial.print(rfid.uid.uidByte[i], HEX);  
  }  
  Serial.println();  
  rfid.PICC_HaltA();  
}
```

*Programski blok 5.4 Kod za ispis UID broja kartice*

Kada su poznati određeni UID brojevi kartica i oznaka u „loop“ funkciji sustava, aktivira se skeniranje kartica, ukoliko je skenirana jedna od autoriziranih kartica pokreće se funkcija „stopAlarmRFID“ koja deaktivira sustav.

```
#include <SPI.h>
#include <MFRC522.h>
MFRC522 mfrc522(SS_PIN, RST_PIN);
byte authorizedCard1[] = {0xF3, 0x8E, 0x29, 0xFB}; //UID prve
kartice
byte authorizedCard2[] = {0x23, 0xAD, 0x46, 0xC5}; //UID druge
kartice

void loop()
{
  if (mfrc522.PICC_IsNewCardPresent() &&
mfrc522.PICC_ReadCardSerial())
  {
    if (isAuthorizedCard())
    {
      stopAlarmRFID();
    } else
    {
      karticaOdbijena();
      Serial.println("Pristup odbijen!");
    }
  }
  mfrc522.PICC_HaltA();
}
```

*Programski blok 5.5 Kod za stalno skeniranje kartica*

### 5.3.5. Postavljanje i promjena lozinke

Prije postavljanja lozinke potrebno je deklarirati konstante za rad tipkovnice. U programskom bloku 5.6 prikazano je postavljanje varijabli koje su potrebne za ispravan rad tipkovnice.

```
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {32, 33, 25, 26}; //pinovi za red
byte colPins[COLS] = {27, 14, 12, 13}; //pinovi za stupac
Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins,
ROWS, COLS);
```

*Programski blok 5.6 Postavljanje varijabli za rad tipkovnice*

Zatim je potrebno deklarirati početnu točnu lozinku i lozinku koja aktivira promjenu lozinke (programski blok 5.7). Nakon toga u „loop“ funkciji treba uvijek pratiti upis znakova tipkovnice te se pomoću znaka „#“ potvrdi unos. Tipkom „\*“ se poništava i briše unos znakova. Znakovi su šifrirani na zaslonu sa znakom „\*“. U programskom bloku 5.8 prikazane su funkcije za provjeru i promjenu lozinke koje se aktiviraju unosom početno definiranih lozinka.

```

char correctPassword[PASSWORD_LENGTH] = {'1', '1', '1', '1'}; // točna
lozinka
char changePasswordCode[PASSWORD_LENGTH] = {'D', 'D', 'D', 'D'}; //za
aktivaciju promjene lozinke

void loop() {
char key = customKeypad.getKey();
  if (key != NO_KEY)
{
    if (!passwordChangeMode)
    {
      provjeriPassword(key);
    }
  }
  if (key != NO_KEY && !sustavAktivan && !alarmAktivan)
  {
    if(passwordChangeMode)
    {
      promjeniPassword(key);
    }
  }
}

```

*Programski blok 5.7 Kod za postavljanje početne lozinke i za aktivaciju promjne lozinke*

```

void provjeriPassword(char key) {
    static char password[PASSWORD_LENGTH + 1] = "";
    static int passwordIndex = 0;
    if (key != NO_KEY) {
        if (key == '*') {
            passwordIndex = 0;
            memset(password, 0, sizeof(password));
        } else if (key == '#') {
            // provjeri upisanu lozinku
            if (strncmp(password, correctPassword, PASSWORD_LENGTH) == 0) {
                // lozinka točna, deaktiviraj sustav
                stopAlarmPassword();
                Serial.println("sustav deaktiviran");
            } else if (strncmp(password, changePasswordCode,
PASSWORD_LENGTH) == 0){

                passwordChangeMode = true;
                Serial.println("promjena lozinke.");
                display.clearDisplay();
                display.setTextSize(1);
                display.setTextColor(WHITE);
                display.setCursor(20,28);
                display.println("Upisi novi PIN");
                display.drawRect(0, 0, 128, 64, WHITE);
                display.display();
                delay(3000);

            }
            else {
                netocnaLozinka();
                Serial.println("netocna lozinka");
            }
            passwordIndex = 0;
            memset(password, 0, sizeof(password));
        } else if (passwordIndex < PASSWORD_LENGTH) {
            password[passwordIndex++] = key;
        }
        display.clearDisplay();
        display.setTextSize(2);
        display.setTextColor(SSD1306_WHITE);
        display.setCursor(40, 25);
        for (int i = 0; i < passwordIndex; i++) {
            display.print('*'); // prikazi lozinku sa
****
            display.drawRect(35, 23, 55, 20, WHITE);
        }
        display.display();
    }
}

```

*Programski blok 5.8 Funkcije za promjenu i postavljanje lozinke*

### 5.3.6. Prikaz potrošnje baterije

Prikaz postotka i napona baterije postignut je spajanjem baterije na analogni ulaz ESP32, korištenjem otpornika za dijeljenje napona kako bi se smanjio napon na prihvatljiv nivo za ESP32 koji iznosi 3.3V. Ukoliko baterija ima manji napon i postotak napunjenosti korisnik će biti obavješten o tome pomoću mobilne aplikacije. Znajući da je maksimalni napon baterije 4.2V, koriste se dva otpornika, jedan od 100k $\Omega$  i jedan od 330k $\Omega$ . Koristeći ova dva otpornika, maksimalni napon iznosit će 1.02V što je unutar sigurnosnog raspona za ESP32. U kodu se koristi ADC (Analog to Digital Converter) za očitavanje napona na bateriji. ADC je komponenta unutar mikroupavljača koja omogućava mjerenje napona na analognim ulazima i pretvaranje tih napona u digitalne vrijednosti koje se mogu obrađivati u programu. ESP32 koristi 12-bitni ADC, što znači da može proizvesti digitalne vrijednosti u rasponu od 0 do 4095. ADC (Analog-to-Digital Converter) rezolucija odnosi se na broj bitova koje ADC koristi za pretvaranje analognog signala u digitalnu vrijednost. U programskom bloku 5.9 naveden je kod pomoću kojeg se izračunao napon baterije.

```
#define BATTERY_PIN 39 //baterija
const float R1 = 330000.0; // Vrijednost otpornika R1 (u ohmima)
const float R2 = 100000.0; // Vrijednost otpornika R2 (u ohmima)

float readBatteryVoltage() {
    int adcValue = analogRead(BATTERY_PIN);
    float stv_napon = adcValue * (3.3/4095);
    float napon = stv_napon * (R1 + R2) / R2;
    return napon;
}

int batteryPercentage() {
    float napon = readBatteryVoltage();
    float minVoltage = 3; // Minimalni napon baterije (potpuno prazna)
    float maxVoltage = 4.2; // Maksimalni napon baterije (potpuno
puna)
    int percentage = (int)((((napon - minVoltage) / (maxVoltage -
minVoltage)) * 100);
    if (percentage > 100) percentage = 100;
    if (percentage < 0) percentage = 0;
    delay(1000);
    return percentage;
}
```

*Programski blok 5.9 Kod za prikaz postotka napunjenosti i napona baterije*

### 5.3.7. Web server

U ovom radu funkcionalnost web servera implementirana je kodom u programskom bloku 5.10 koji omogućuje prikaz trenutne temperature, vlage, napona baterije i postotka napunjenosti baterije putem web sučelja. Ovaj web server omogućuje korisniku nadzor nad uvjetima u prostoriji i stanjem baterije. Za postavljanje web servera na ESP32, potrebno je instalirati odgovarajuće biblioteke i konfigurirati ESP32 kao klijenta. U ovom projektu se koristi Wi-Fi mrežu za povezivanje na lokalnu mrežu. U programskom bloku ispod prikazane su biblioteke koje su korištene i web stranica koja je izrađena pomoću html jezika.

```
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
char ssid[] = "SamsungGalaxyA52s"; //wifi ssid
char pass[] = "10203040"; //wifi lozinka
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Povezivanje na WiFi..."); }

Serial.print("IP adresa: ");
Serial.println(WiFi.localIP());
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    String html = "<!DOCTYPE html><html>";
    html += "<head><title>ESP32 Alarm Server</title>";
    html += "<style>";
    html += "body { font-family: Arial, sans-serif; text-align: center; padding: 50px; }";
    html += "h1 { color: #333; }";
    html += "p { font-size: 1.5em; }";
    html += "</style></head>";
    html += "<body>";
    html += "<h1>Alarmni sustav</h1>";
    html += "<p>Trenutna temperatura: " +
String(dht.readTemperature()) + " &#8451;</p>";
    html += "<p>Trenutna vlaznost: " + String(dht.readHumidity()) +
" %</p>";
    html += "<p>Trenutni napon baterije: " +
String(readBatteryVoltage()) + " V</p>";
    html += "<p>Postotak baterije: " + String(batteryPercentage())
+ " %</p>";
    html += "</body></html>";
    request->send(200, "text/html", html);
});
server.begin();
```

*Programski blok 5.10 Kod za implementaciju lokalnog servera*



Na slici 5.2 prikazan je izgled web sučelja na lokalnoj adresi otvoren pomoću web preglednika.



*Slika 5.2 Prikaz lokalnog web sučelja*

### **5.3.8. Praćenje promjene temperature i vlage**

Temperatura i vlaga prate se pomoću DHT11 senzora koji ima vlastitu biblioteku. U programskom bloku 5.11 prikazana je funkcija „provjeri\_promjenu\_temperature“ koja prati vrijednosti temperature u posljednjih deset minuta te ukoliko dođe do promjene veće od 5°C, obavijest se šalje putem mobilne aplikacije.

```

#include <DHT.h>
float temp = dht.readTemperature();
float vlaznost = dht.readHumidity();
void loop()
{
  if (provjeri_promjenu_temp()) {
    Blynk.logEvent("promjena_temperature", "Nagla promjena
temperature!");
  }
}
bool provjeri_promjenu_temp() {
  // provjeri ako se temperatura promjenila za 5°C zadnjih 10 minuta
float oldestTemperature = temperatureReadings[currentIndex];
  for (int i = 1; i < numReadings; i++) {
    int index = (currentIndex - i + numReadings) % numReadings;
    float deltaTemperature = abs(temperatureReadings[index] -
oldestTemperature);
    if (deltaTemperature >= 5.0) {
      return true;
    }
  }
  return false;
}

```

*Programski blok 5.11 Kod za prikaz i praćenje temperature i vlage*

## 6. Integracija s mobilnom aplikacijom Blynk Iot Cloud

Blynk je platforma za IoT koja omogućuje jednostavno povezivanje i upravljanje uređajima putem mobilne aplikacije. U ovom projektu, Blynk aplikacija se koristi za upravljanje alarmnim sustavom.

### 6.1. Uvod u Blynk

Blynk platforma sastoji se od tri glavne komponente:

- Blynk mobilna aplikacija koja se koristi za kreiranje sučelja u upravljanje sustavom
- Blynk server koji posreduje između aplikacije i uređaja
- Blynk biblioteka koja omogućuje spajanje mikroupravljača sa Blynk serverom

### 6.2. Konfiguracija aplikacije

Za početak je potrebna registraciji na Blynk web stranici. Nakon registracije potrebno je dodati uređaj, u ovom slučaju ESP32 te imenovati novi projekt. Na mikroupravljaču je potrebno instalirati određene biblioteke (*eng. library*), upisati ime projekta i dobiveni jedinstveni broj te ih definirati na početku koda (programski blok 6.1).

```
#include <BlynkSimpleEsp32.h>
#include <WiFi.h>
#define BLYNK_TEMPLATE_ID "TMPL4qCycmL_Q"
#define BLYNK_TEMPLATE_NAME "Alarmni sustav"
#define BLYNK_AUTH_TOKEN "iWr0lD99Kl-O6NVTNdFVyOD4ToQ6Z8W6"
```

*Programski blok 6.1 Definiranje Blynk biblioteka*

Nakon definiranja biblioteka dodaje virtualne ulaze tj. varijable. Na slici 6.1 prikazan je popis virtualnih podataka koje su uključene u ovaj sustav. Programski kod 6.2 prikazuje kako se virtualni ulazi aktiviraju u kodu.

**Alarmni sustav**

Home **Datastreams** Web Dashboard Automation Templates Metadata Connection Lifecycle Events & Notifications Mobile Dashboard

Q Search datastream

Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min	Max	Decimals	Default
1	aktivan	aktivan	<span style="color: red;">■</span>	V2	Integer		false	0	1	-	0
2	Temperatura	Temperatura	<span style="color: yellow;">■</span>	V3	Double	°C	false	0	50	###	
3	Vlažnost	Vlaznost	<span style="color: cyan;">■</span>	V4	Integer	%	false	0	100	-	
4	Status	Status	<span style="color: green;">■</span>	V5	String		false			-	
5	changepassword	changepassword	<span style="color: blue;">■</span>	V1	Integer		false	0	1	-	0

Slika 6.1 Blynk virtualni ulazi

```

BLYNK_WRITE(V2) { // virtualni pin za aktivnost sustava
  sustavAktivan = param.asInt();
  if (sustavAktivan) {
    Serial.println("Sustav aktivan.");
  } else {
    digitalWrite(LEDc, LOW);
    display.clearDisplay();
    display.display();
    Serial.println("Sustav Neaktivan.");
  }
}

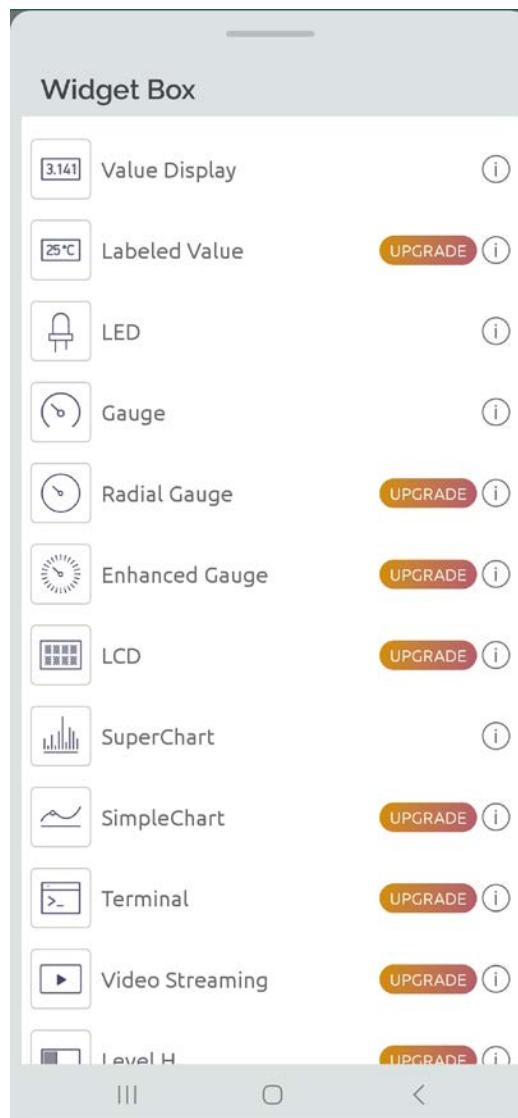
BLYNK_WRITE(V1) { // virtualni pin za promjenu lozinke
  if (param.asInt() == 1) {
    passwordChangeMode = true;
    Serial.println("promjena lozinke.");
    display.clearDisplay();
    display.setCursor(20,28);
    display.println("Unesi novi PIN");
    display.drawRect(0, 0, 128, 64, WHITE);
    display.display();
  }
}

Blynk.virtualWrite(V3, temp); // virtualni pin za temperaturu
Blynk.virtualWrite(V4, vlaznost); // virtualni pin zavlagu
Blynk.virtualWrite(V5, "stanje sustava"); // virtualni pin za stanje

```

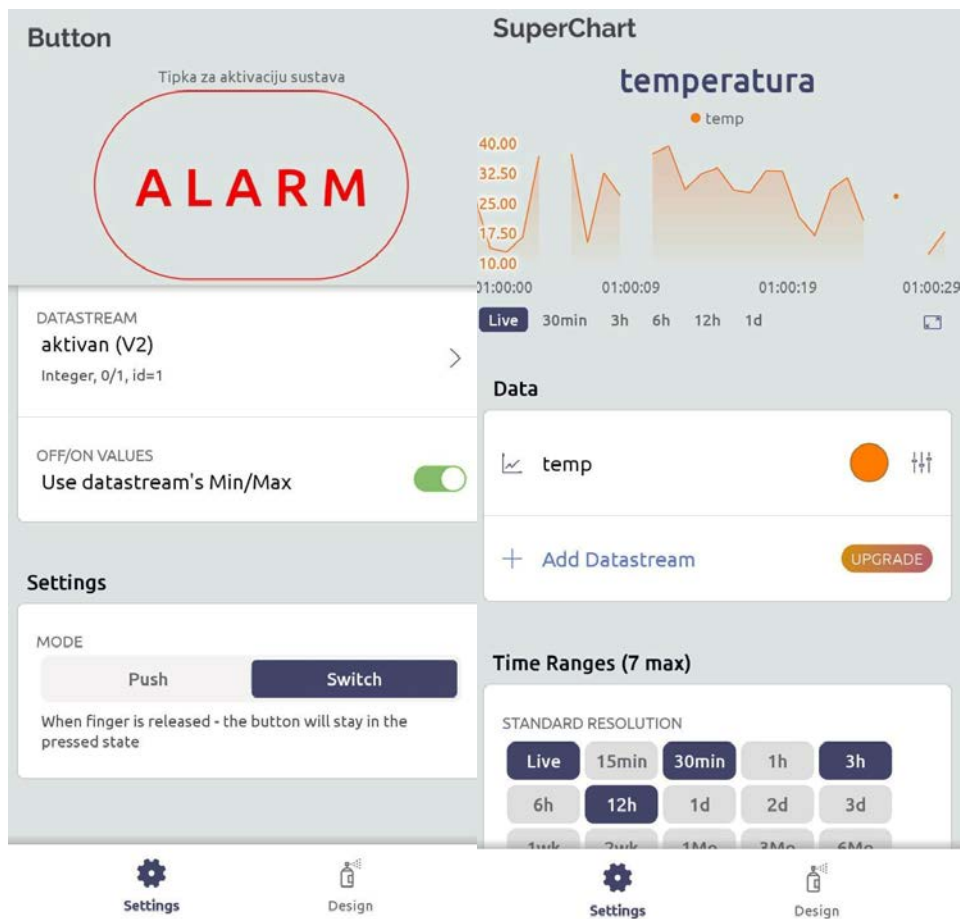
Programski blok 6.2 Aktiviranje virtualnih ulaza

Nakon dodavanja svih ulaza potrebno je izraditi sučelje za aplikaciju. Sučelje se izrađuje na mobilnom uređaju. Potrebno je instalirati aplikaciju Blynk IoT te se prijaviti svojim računom koji je prethodno izrađen. Zatim se odabere kreirani projekt i klikne na „Customize“, otvara se izbornik na kojem se mogu odabrati razne sklopke, grafovi itd. (Slika 6.2).

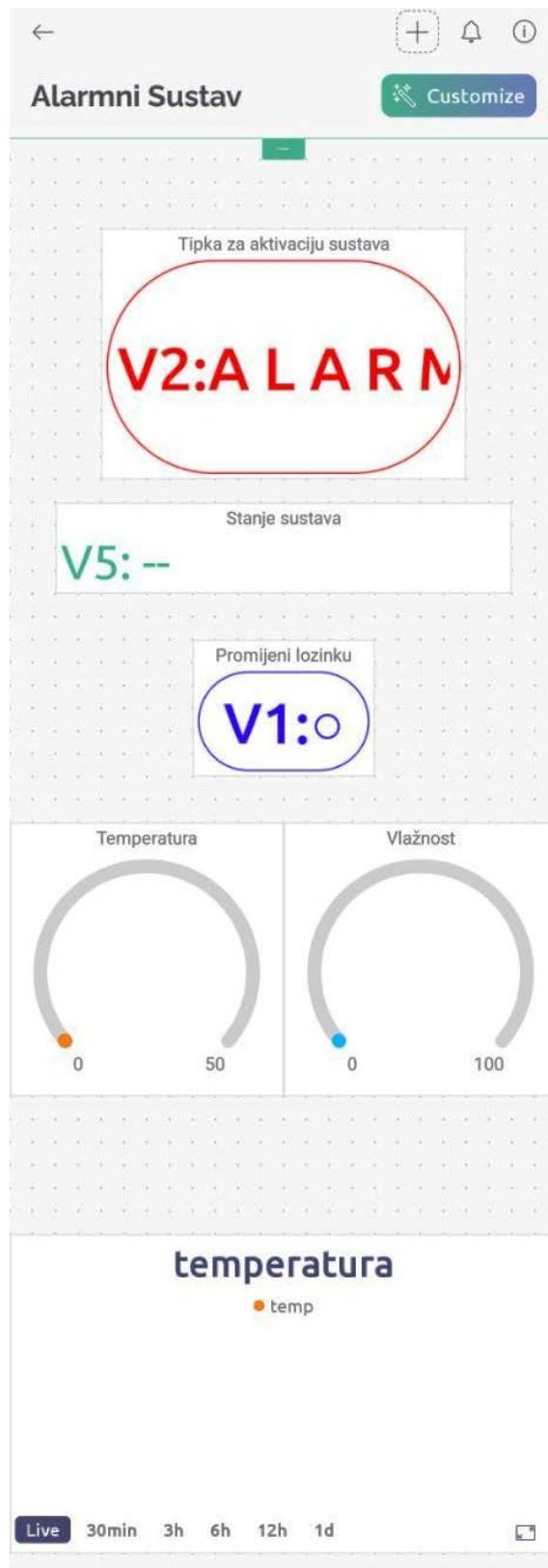


*Slika 6.2 Izbornik prikaza*

Nakon dodavanja svih željenih prikaznih elemenata potrebno ih je povezati sa varijablama (slika 6.3). Na slici 6.4 prikazan je završni izgled sučelja aplikacije za ovaj projekt.

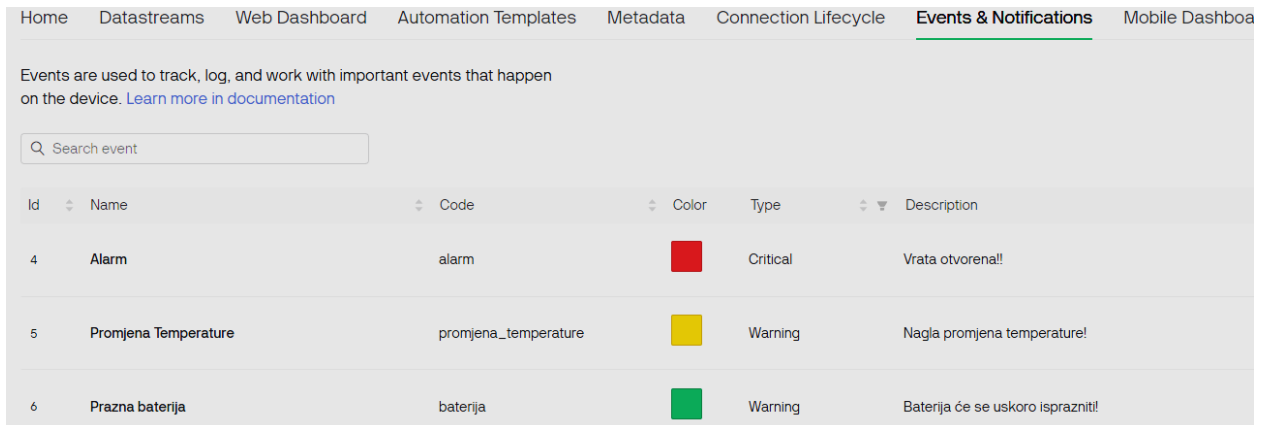


Slika 6.3 Dodavanje varijabli na sučelje



Slika 6.4 Završni izgled sučelja

Potrebo je još dodati obavijesti i zvučni alarm. Postupak je sličan kao i za ulazne varijable. U mobilnoj aplikaciji ili na web pregledniku nakon prijave na Blynk server i odabira vlastitog projekta odabere se „Events & Notifications“ (hrv. događaji i obavijesti). Zatim se kreiraju vrste obavijesti i kreiraju im imena. Na slici 6.5 prikazan je popis događaja korištenih u ovom radu. Nakon kreiranja svih željenih događaja potrebno ih je pozivati u kodu kao što je to prikazano u programskom bloku 6.3.



The screenshot shows the Blynk Events & Notifications dashboard. At the top, there are navigation tabs: Home, Datastreams, Web Dashboard, Automation Templates, Metadata, Connection Lifecycle, Events & Notifications (selected), and Mobile Dashboa. Below the tabs, there is a heading: "Events are used to track, log, and work with important events that happen on the device. [Learn more in documentation](#)". A search bar labeled "Search event" is present. Below the search bar is a table with the following columns: Id, Name, Code, Color, Type, and Description.

Id	Name	Code	Color	Type	Description
4	Alarm	alarm	Red	Critical	Vrata otvorena!!
5	Promjena Temperature	promjena_temperature	Yellow	Warning	Nagla promjena temperature!
6	Prazna baterija	baterija	Green	Warning	Baterija će se uskoro isprazniti!

Slika 6.5 Prikaz događaja i obavijesti

```
if(napon < 2.5)
{
  Blynk.logEvent("baterija","Baterija će se uskoro isprazniti!");
}

Blynk.logEvent("alarm","Alarm! Vrata otvorena!");

if (provjeri_promjenu_temp())
{
  Blynk.logEvent("promjena_temperature","Nagla promjena
temperature!");
}
```

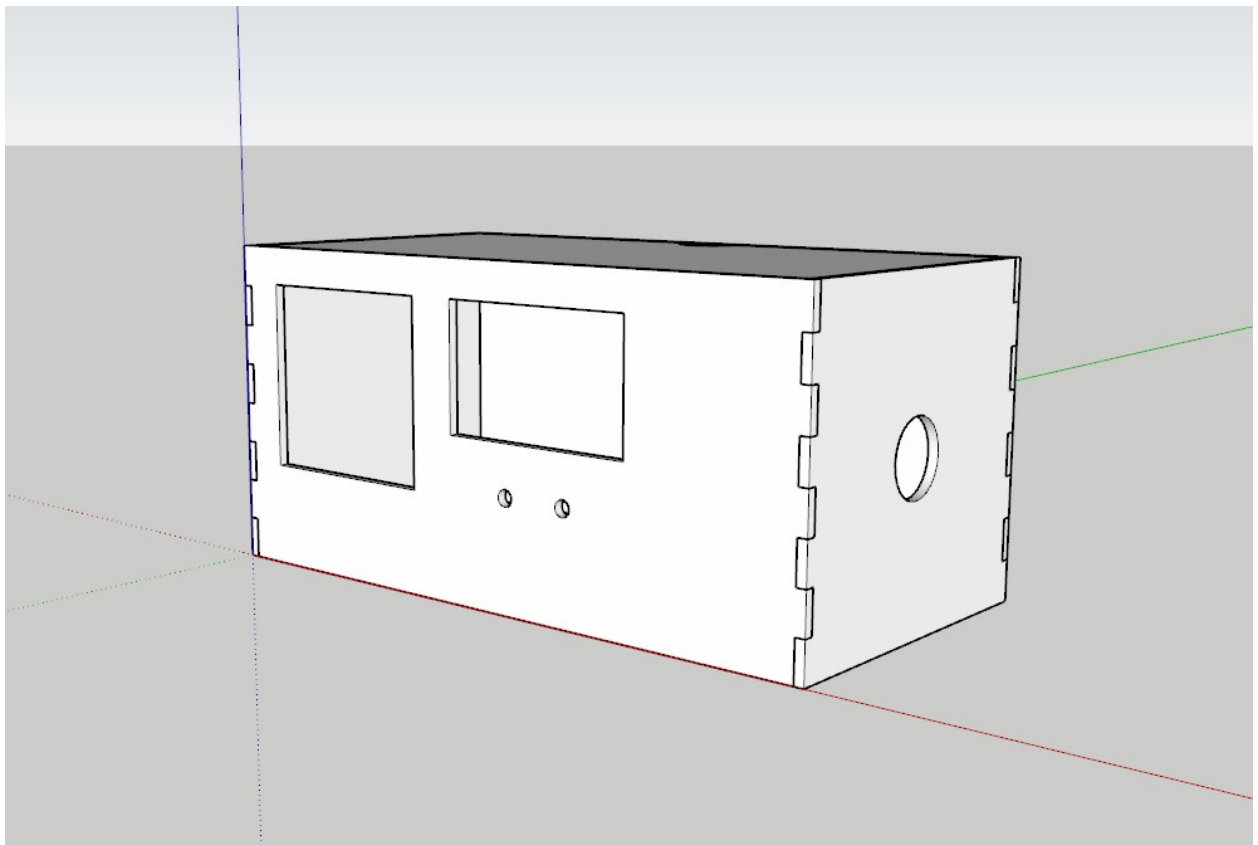
Programski blok 6.3 Aktiviranje događaja i obavijesti



## 7. Izrada i sklapanje kućišta

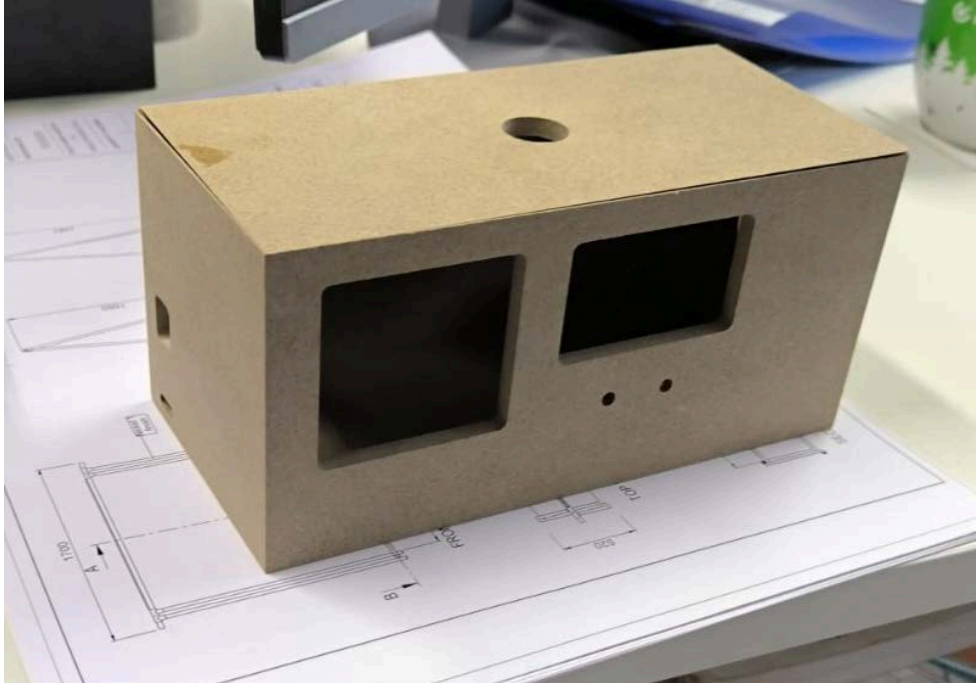
Kućište je ključan element svakog uređaja jer pruža zaštitu komponentama i osigurava estetski izgled. U ovom projektu, kućište za alarmni sustav dizajnirano je u programu SketchUp i izrađeno pomoću MDF (*eng. Medium-Density Fiberboard*) ploča. MDF ili medijapan ploče izrađene od drvene prašine i smole te komprimirane pod visokim tlakom. Imaju veliku gustoću te su ujednačajnije i kompaktnije u usporedbi s pločom iverice. Ovo poglavlje detaljno opisuje proces dizajniranja, izrade i sklapanja kućišta [12].

SketchUp je popularan alat za 3D modeliranje koji omogućuje jednostavno dizajniranje i vizualizaciju projekata. Prije modeliranja kućišta potrebno je izmjeriti komponente koje će se sklapati u kućište te prema njima izraditi model. Slika 7.1 prikazuju završen model kućišta u programu.



*Slika 7.1 Prikaz modela kućišta*

Nakon crtanja, medijapan ploče izrezane su po mjeri te zalijepljene ljepljivom za drvo. Na slici 7.2 prikazano je zalijepljeno kućište.



*Slika 7.2 Kućište nakon lijepljenja*

Nakon što se ljepljivo posušilo slijedilo je bojanje. Bojanje se vršilo sa sprejom. Zatim se na dno kućišta postavila eksperimentalna pločica (*eng. breadboard*) sa mikroupravljačem. Komponente su pričvršćene pomoću ljepljive trake i vrućeg ljepljiva. Završni izgled kućišta prikazan je na slici 7.3.



*Slika 7.3 Završni izgled kućišta*

## 8. Zaključak

U ovom završnom radu detaljno je opisan proces dizajniranja, izrade i implementacije alarmnog sustava temeljenog na ESP32 mikroupravljaču. Sustav je opremljen raznim senzorima i komponentama. Funkcionalnost alarmnog sustava proširena je upotrebom Blynk aplikacije i implementacijom web servera. Projekt je započeo teoretskim pregledom ključnih komponenti. Ovaj teoretski okvir omogućio je bolje razumijevanje funkcionalnosti i prednosti pojedinih komponenti te njihovih uloga u sustavu.

U praktičnom dijelu rada detaljno su opisani procesi povezivanja i programiranja ESP32 mikroupravljača. Implementacija funkcionalnosti sustava obuhvatila je detekciju otvaranja vrata pomoću reed releja, aktivaciju alarma, očitavanje temperature i vlage, upravljanje RFID čitačem te uspostavu web servera za daljinsko praćenje podataka.

Posebna pažnja posvećena je izradi i sklapanju kućišta. Kućište je dizajnirano u programu SketchUp i izrađeno od MDF materijala, čime je osigurana zaštita komponenata i kompaktni sustav. Precizno rezanje, brušenje i bojanje kućišta osigurali su kvalitetnu završnu obradu.

Implementacija Blynk aplikacije omogućila je daljinsko upravljanje i praćenje sustava putem mobilnog uređaja, čime je povećana funkcionalnost i pristupačnost sustava. Blynk aplikacija pružila je sučelje za upravljanje alarmom, praćenje temperature i vlage.

Zaključno, ovaj projekt uspješno je demonstrirao mogućnosti ESP32 mikroupravljača u izradi naprednog alarmnog sustava. Integracija različitih senzora i komponenti te korištenje IoT tehnologija omogućila je stvaranje funkcionalnog, pouzdanog i korisnički orijentiranog sustava. Ovaj rad pokazuje kako se suvremene tehnologije mogu koristiti za poboljšanje sigurnosnih sustava i pruža solidnu osnovu za daljnji razvoj i nadogradnju.

## 9. Literatura

[1] Mikroupravljač ESP32

<https://www.teachmemicro.com/esp32-pinout-diagram-wroom-32/> (Pristupljeno 20.5.2024.)

[2] Definicija IoT

<https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>

(Pristupljeno 20.5.2024.)

[3] Primjena „Internet stvari“

<https://www.mdpi.com/1424-8220/22/23/9271> (Pristupljeno 20.5.2024.)

[4] RFID modul

<https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>

(Pristupljeno 23.5.2024.)

[5] dipl. ing. Srpak Josip: Automatizacija strojeva i uređaja, Sveučilište Sjever u Varaždinu,

Skripte za učenje

[6] OLED zaslon

<https://www.otronic.nl/de/24-zoll-oled-display-128x64-gelb-i2c-ss-149803019.html>

(Pristupljeno 23.5.2024.)

[7] Matrična tipkovnica 4x4

<https://www.az-delivery.de/en/products/4x4-matrix-numpad> (Pristupljeno 23.5.2024.)

[8] Buzzer

<https://www.botshop.co.za/shop/ec-084-buzzer-continuous-3-24v-632>

(Pristupljeno 23.5.2024.)

[9] DHT11 senzor za temperaturu i vlagu

<https://soldered.com/hr/proizvod/dht11-senzor-temperature-i-vlage/>

(Pristupljeno 25.5.2024.)

[10] LED dioda

<https://kronos.hr/dioda-led-i-kucista/7510-led-8-crvena.html> (Pristupljeno 25.5.2024.)

[11] Reed relej

<https://www.arrow.com/en/research-and-events/articles/the-reed-switch-ingeniously-simple-sensing> (Pristupljeno 6.6.5.2024.)

[12] Medijapan ploče

[https://www.emajstor.hr/clanak/577/mdf\\_ploce\\_ploca\\_od\\_iverice\\_ili\\_nesto\\_drugo](https://www.emajstor.hr/clanak/577/mdf_ploce_ploca_od_iverice_ili_nesto_drugo)

(Pristupljeno 6.6.2024.)

## 10. Popis slika

Slika 2.1 Primjena IoT [2].....	3
Slika 3.1 Prikaz početnog stanja sustava .....	6
Slika 3.2 Prikaz aktivnog sustava.....	6
Slika 3.3 Izgled mobilne aplikacije .....	7
Slika 4.1 ESP32 WROOM-32.....	8
Slika 4.2 Raspored pinova ESP32 WROOM modela [1] .....	9
Slika 4.3 Prikaz rada RFID modula [4] .....	10
Slika 4.4 RFID RC522 modul [4].....	12
Slika 4.5 Princip rada reed releja [5] .....	13
Slika 4.6 OLED I2C zaslon 2.4“ [6].....	14
Slika 4.7 Princip rada matrične tipkovnice .....	16
Slika 4.8 Matrična tipkovnica 4x4 [7] .....	17
Slika 4.9 Buzzer 3-24V [8] .....	18
Slika 4.10 DHT11 [9] .....	19
Slika 4.11 Modul za napajanje .....	21
Slika 4.12 LED dioda [10] .....	24
Slika 5.1 Shema spajanja .....	27
Slika 5.2 Prikaz lokalnog web sučelja .....	39
Slika 6.1 Blynk virtualni ulazi.....	42
Slika 6.2 Izbornik prikaza .....	43
Slika 6.3 Dodavanje varijabli na sučelje.....	44
Slika 6.4 Završni izgled sučelja.....	45
Slika 6.5 Prikaz doadaja i obavijesti.....	46
Slika 7.1 Prikaz modela kućišta .....	47
Slika 7.2 Kućište nakon lijepljenja .....	48
Slika 7.3 Završni izgled kućišta .....	48

## 11. Popis programskih blokova i tablica

Programski blok 5.1 Arduino kod za detekciju otvaranja vrata.....	29
Programski blok 5.2 Aktivacija alarma .....	30
Programski blok 5.3 Kod za deaktivaciju alarma pomoću tipkovnice i RFID modula.....	31
Programski blok 5.4 Kod za ispis UID broja kartice.....	32
Programski blok 5.5 Kod za stalno skeniranje kartica.....	33
Programski blok 5.6 Postavljanje varijabli za rad tipkovnice.....	34
Programski blok 5.7 Kod za postavljanje početne lozinke i za aktivaciju promjene lozinke .....	35
Programski blok 5.8 Funkcije za promjenu i postavljanje lozinke .....	36
Programski blok 5.9 Kod za prikaz postotka napunjenosti i napona baterije.....	37
Programski blok 5.10 Kod za implementaciju lokalnog servera.....	38
Programski blok 5.11 Kod za prikaz i praćenje temperature i vlage.....	40
Programski blok 6.1 Definiranje Blynk biblioteka.....	41
Programski blok 6.2 Aktiviranje virtualnih ulaza .....	42
Programski blok 6.3 Aktiviranje događaja i obavijesti.....	46
Tablica 5.1 Spajanje pinova komponenti na pinove ESP32 .....	26



### IZJAVA O AUTORSTVU

Završni/diplomski/specijalistički rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Domagoj Hohnjec (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog/specijalističkog (obrisati nepotrebno) rada pod naslovom Izrada alarmnog sustava za pamentnu kuću uz primjenu IoT-a (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:  
(upisati ime i prezime)

Domagoj Hohnjec  
(vlastoručni potpis)

Sukladno članku 58., 59. i 61. Zakona o visokom obrazovanju i znanstvenoj djelatnosti završne/diplomske/specijalističke radove sveučilišta su dužna objaviti u roku od 30 dana od dana obrane na nacionalnom repozitoriju odnosno repozitoriju visokog učilišta.

Sukladno članku 111. Zakona o autorskom pravu i srodnim pravima student se ne može protiviti da se njegov završni rad stvoren na bilo kojem studiju na visokom učilištu učini dostupnim javnosti na odgovarajućoj javnoj mrežnoj bazi sveučilišne knjižnice, knjižnice sastavnice sveučilišta, knjižnice veleučilišta ili visoke škole i/ili na javnoj mrežnoj bazi završnih radova Nacionalne i sveučilišne knjižnice, sukladno zakonu kojim se uređuje umjetnička djelatnost i visoko obrazovanje.