**Tehnička škola Čakovec**

**ELABORAT ZAVRŠNOG RADA**

**BABY MONITOR**

|  |  |
| --- | --- |
| Mentor: | Učenik: |
| Krešimir Kočiš, prof. | **Martin Molnar, 4.RT** |

**Čakovec, svibanj 2025.**

**Tehnička škola Čakovec**

**Prosudbeni odbor za završni rad**

**Učenik: Martin Molnar**

**Razred: 4. RT**

**Školska godina: 2025.**

**Obrazovno područje: računalstvo**

**Zanimanje: Tehničar za računalstvo**

**Naziv zadatka: Baby monitor**

**Opis zadatka:** Izraditi sustav za nadzor djeteta. Sustav se sastoji od kamere spojene s mikrokontolerom (ESP32, Arduino, Raspberry Pi…) i mobilnog uređaja s aplikacijom za upravljanje kamerom. Preko aplikacije moguće je zumirati, rotiranje kamere lijevo/desno i gore/dolje, audio u oba smjera, puštanje umirujeće melodije. Dijete (plač) se treba čuti i kad se aplikacija ne koristi (nije u fokusu). Sustav treba raditi u LAN (WiFi) mreži i preko Interneta.

|  |  |  |
| --- | --- | --- |
| Zadatak zadan:  31.listopada 2024. | Rok predaje pisanog rada:  16. svibnja 2025. | Predviđeni datum obrane:  9.lipnja 2025. |
|  |  |  |

Mentor:

[1. UVOD 7](#_Toc197629364)

[1.1 OPIS ZADATKA 7](#_Toc197629365)

[1.2 PLAN RADA 7](#_Toc197629366)

[2. VIDEO NADZOR I UPRAVLJANJE KAMEROM 8](#_Toc197629367)

[2.1. KORIŠTENI ALATI 8](#_Toc197629368)

[2.1.1. SKLOPOVLJE 8](#_Toc197629369)

[2.1.2. SOFTVER 9](#_Toc197629370)

[2.2. OPIS RADA 9](#_Toc197629371)

[2.3 REALIZACIJA RADA 9](#_Toc197629372)

[2.3.1. Postavljanje ESP32-CAM modula 9](#_Toc197629373)

[2.3.2. KONFIGURACIJA VIDEO STREAMINGA 10](#_Toc197629374)

[2.3.3. IMPLEMENTACIJA UPRAVLJANJA SERVO MOTORA 10](#_Toc197629375)

[2.3.4. PRIKAZ VIDEA I KONTROLA U ANDROID APLIKACIJI 11](#_Toc197629376)

[3. DVOSMJERNA AUDIO KOMUNIKACIJA I REPRODUKCIJA MELODIJE 11](#_Toc197629377)

[3.1. KORIŠTENI ALATI 11](#_Toc197629378)

[3.1.1. SKLOPOVLJE 11](#_Toc197629379)

[3.1.2. SOFTVER 13](#_Toc197629380)

[3.2. OPIS RADA 13](#_Toc197629381)

[3.3. REALIZACIJA RADA 13](#_Toc197629382)

[3.3.1. PRIKUPLJANJE ZVUKA 13](#_Toc197629383)

[3.3.2. SLANJE ZVUKA S ESP1 NA ANDROID APLIKACIJU 13](#_Toc197629384)

[3.3.3. PRIMANJE ZVUKA I REPRODUKCIJA NA ANDROID APLIKACIJI 14](#_Toc197629385)

[3.3.4. SLANJE ZVUKA S ANDROID APLIKACIJE NA ESP32 14](#_Toc197629386)

[3.3.5. REPRODUKCIJA ZVUKA NA ESP2 (MAX98357A I ZVUČNIK) 14](#_Toc197629387)

[3.3.6. IMPLEMENTACIJA REPRODUKCIJE UMIRUJUĆE MELODIJE 15](#_Toc197629388)

[4. ANDROID APLIKACIJA – POZADINSKI RAD I MREŽNA KOMUNIKACIJA 15](#_Toc197629389)

[4.1. KORIŠTENI ALATI 15](#_Toc197629390)

[4.1.1. SOFTVER 15](#_Toc197629391)

[4.2. OPIS RADA 16](#_Toc197629392)

[4.3. REALIZACIJA RADA 16](#_Toc197629393)

[4.3.1. STRUKTURA ANDROID APLIKACIJE 16](#_Toc197629394)

[4.3.2. UPRAVLJANJE TCP KONEKCIJAMA I PRIJENOS PODATAKA 17](#_Toc197629395)

[4.3.4. POZADINSKA OBAVIJEST I RAD APLIKACIJE 18](#_Toc197629396)

[5. ZAKLJUČAK 19](#_Toc197629397)

[LITERATURA 20](#_Toc197629398)

[POPIS SLIKA 21](#_Toc197629399)

# UVOD

## OPIS ZADATKA

Zadatak ovog završnog rada je izraditi sustav za nadzor djeteta, poznatiji kao "Baby monitor". Sustav treba omogućiti roditeljima daljinski nadzor djeteta koristeći video i audio prijenos. Ključne komponente sustava su kamera povezana s mikrokontrolerom (u ovom slučaju ESP32-CAM) te mobilni uređaj s Android aplikacijom za upravljanje i interakciju. Aplikacija mora podržavati funkcije poput zumiranja video prikaza, rotiranja kamere u horizontalnom i vertikalnom smjeru, dvosmjernu audio komunikaciju te mogućnost reprodukcije umirujuće melodije na jedinici kod djeteta. Važan zahtjev je da se zvuk djeteta (npr. plač) može čuti na mobilnom uređaju čak i kada aplikacija nije aktivno korištena (nije u fokusu). Sustav mora biti funkcionalan unutar lokalne Wi-Fi (LAN) mreže, s ciljem omogućavanja rada i preko Interneta. Ovaj projekt integrira znanja iz područja ugradbenih sustava, mrežnih komunikacija i razvoja mobilnih aplikacija.

## PLAN RADA

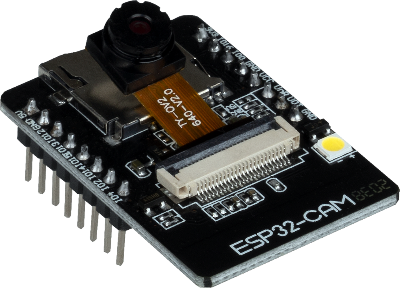
Ovaj rad je strukturiran u nekoliko glavnih cjelina kako bi se sustavno prikazao proces razvoja Baby monitora. Prva cjelina nakon uvoda bavi se implementacijom video nadzora i daljinskog upravljanja kamerom. Objasnit će se korištene hardverske i softverske komponente, način postavljanja ESP32-CAM modula, konfiguracija video streaminga te integracija s Android aplikacijom za prikaz i kontrolu servo motora. Druga cjelina detaljno opisuje razvoj dvosmjerne audio komunikacije i funkcije reprodukcije umirujuće melodije. Uključuje opis prikupljanja zvuka pomoću INMP441 mikrofona, slanje i primanje audio podataka između ESP32 modula i Android aplikacije, te reprodukciju zvuka putem MAX98357A pojačala i zvučnika. Treća cjelina fokusira se na arhitekturu Android aplikacije, s posebnim naglaskom na pozadinski rad servisa za mrežnu komunikaciju (StreamingService), upravljanje TCP konekcijama i osiguravanje da audio notifikacije rade i kada aplikacija nije u prvom planu. Četvrta cjelina razmatra aspekte sigurnosti i pristupa sustavu, s osvrtom na rad u LAN okruženju i potrebne korake za omogućavanje pristupa preko Interneta. Na kraju, u zaključku se sumiraju postignuti rezultati, analiziraju moguća ograničenja te daju prijedlozi za daljnji razvoj i poboljšanja sustava.

# VIDEO NADZOR I UPRAVLJANJE KAMEROM

### 2.1. KORIŠTENI ALATI

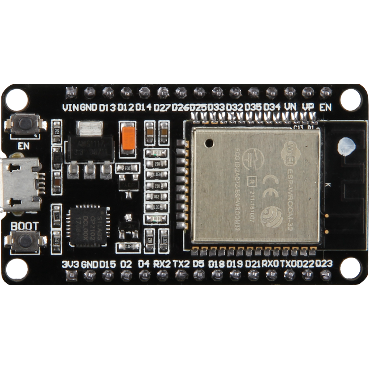
#### 2.1.1. SKLOPOVLJE

ESP32-CAM modul (Slika 1.) : Mikrokontrolerski modul s integriranom kamerom (OV2640) i Wi-Fi funkcionalnošću, korišten za snimanje i streaming videa.



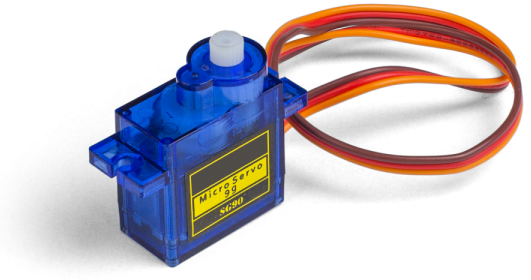
Slika 1. ESP32-CAM modul

ESP32 razvojna pločica (Slika 2. ili Slika 9.): Standardna ESP32 pločica koja upravlja servo motorima.



Slika 2. ESP32 razvojna pločica

Servo motori SG90 (Slika 3.) : Mali servo motori za horizontalno (pan) i vertikalno (tilt) pomicanje ESP32-CAM modula.



Slika 3. Servo motor SG90

ESP32 Motherboard (Slika 4.) : Pločica koja olakšava programiranje i spajanje ESP32-CAM modula.

Slika na kojoj se prikazuje elektronika, elektronička komponenta, elektroničko inženjerstvo, komponenta strujnog kruga

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 4. ESP32 Motherboard

#### 2.1.2. SOFTVER

Arduino IDE: Razvojno okruženje za programiranje ESP32 i ESP32-CAM modula (C++).

Android Studio: Razvojno okruženje za izradu Android aplikacije (Java).

## 2.2. OPIS RADA

Cilj ove cjeline je implementirati sustav za video nadzor koji omogućuje korisniku pregled slike s kamere u stvarnom vremenu putem Android aplikacije. Dodatno, korisnik mora imati mogućnost daljinskog pomicanja kamere (gore-dolje, lijevo-desno) kako bi prilagodio vidno polje. ESP32-CAM modul zadužen je za snimanje i slanje video streama, dok ESP32 pločica (ESP2) upravlja servo motorima na koje je kamera montirana. Android aplikacija prima video stream i šalje kontrolne naredbe za pomicanje kamere.

## 2.3 REALIZACIJA RADA

#### 2.3.1. Postavljanje ESP32-CAM modula

ESP32-CAM modul programiran je odabirom CAMERA\_MODEL\_AI\_THINKER i korištenjem odgovarajućih GPIO pinova definiranih u camera\_pins.h. Konfiguracija kamere uključuje postavljanje frekvencije XCLK na 20MHz, inicijalnu veličinu slike, format piksela PIXFORMAT\_JPEG i kvalitetu JPEG kompresije. Modul se spaja na Wi-Fi mrežu i pokreće web server. U nastavku se vidi programski isječak ESP32-CAM - setup() funkcija, dio za inicijalizaciju kamere (Slika 5.) :

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 5. ESP32-CAM – inicijalizacija kamere

#### 2.3.2. KONFIGURACIJA VIDEO STREAMINGA

Nakon inicijalizacije, ESP32-CAM pokreće HTTP web server. Video stream je dostupan putem URL-a http://<IP\_ADRESA\_KAMERE>:81/stream. Android aplikacija koristi WebView komponentu za prikaz ovog streama. U nastavku se vidi programski isječak ESP32-CAM - setup() funkcija, dio za pokretanje servera (Slika 6.) :

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 6. ESP32-CAM - pokretanje servera

#### 2.3.3. IMPLEMENTACIJA UPRAVLJANJA SERVO MOTORA

Na drugu ESP32 pločicu spojena su dva servo motora. Ova pločica pokreće TCP server na portu 8082 koji sluša naredbe za pomicanje servo motora. Naredbe su u formatu S1=kut za servo 1 i S2=kut za servo 2 gdje kut predstavlja željeni kut u rasponu 0-180 stupnjeva. Koristi se ESP32Servo biblioteka (Slika 7.) :

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 7. Programski isječak handleControlClient() funkcije

#### 2.3.4. PRIKAZ VIDEA I KONTROLA U ANDROID APLIKACIJI

Android aplikacija koristi WebView za prikaz video streama s ESP32-CAM. URL za stream. Digitalni zoom implementiran je skaliranjem WebView komponente pomoću SeekBar-a. Za pomicanje kamere, SeekBar elementi u aplikaciji šalju odgovarajuće naredbe prema StreamingService-u, koji ih prosljeđuje na ESP2 modul preko TCP konekcije. Ta dva SeekBar-a možemo vidjeti u nastavku (Slika 8.):

Slika na kojoj se prikazuje tekst, račun, snimka zaslona, Font

Sadržaj generiran umjetnom inteligencijom može biti netočan.

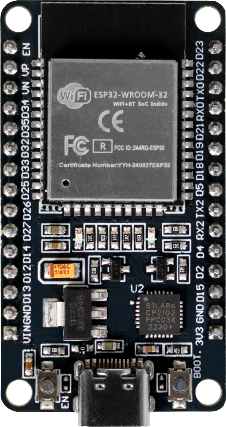
Slika 8. Prikaz SeekBar-a za kontroliranje servo motora

# 3. DVOSMJERNA AUDIO KOMUNIKACIJA I REPRODUKCIJA MELODIJE

## 3.1. KORIŠTENI ALATI

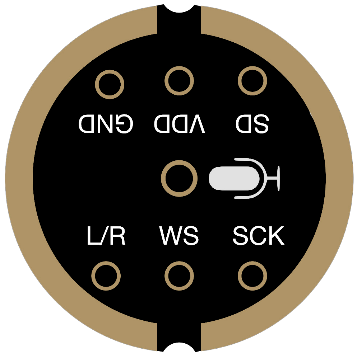
#### 3.1.1. SKLOPOVLJE

ESP32 (ESP1) razvojna pločica za mikrofon (Slika 9. ili Slika 2.): Zadužena za prikupljanje zvuka.



Slika 9. ESP32 razvojna pločica za mikrofon

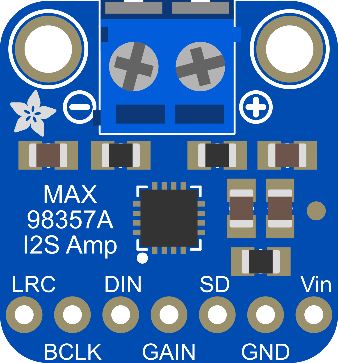
INMP441 digitalni I2S mikrofon (Slika 10.) : Spojen na ESP1, snima zvuk.



Slika 10. INMP441 digitalni I2S mikrofon

ESP32 (ESP2) razvojna pločica za zvučnik (Slika 1.) : Ista pločica koja upravlja i servo motorima, zadužena i za reprodukciju zvuka.

MAX98357A I2S mono pojačalo (Slika 11.) : Spojeno na ESP2, pogoni zvučnik.



Slika 11. MAX98357A I2S mono pojačalo

Mali 5V zvučnik (Slika 12.) : Spojen na MAX98357A, reproducira zvuk.

Slika na kojoj se prikazuje elektronika, elektroničko inženjerstvo, kabel, elektronička komponenta

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 12. Mali 5V zvučnik

#### 3.1.2. SOFTVER

Arduino IDE: Za programiranje ESP32 modula (C++), korištenje I2S drivera.

Android Studio: Za programiranje Android aplikacije (Java).

Android AudioRecord klasa: Za snimanje zvuka s mikrofona telefona.

Android AudioTrack klasa: Za reprodukciju primljenog zvuka na telefonu.

## 3.2. OPIS RADA

Ova cjelina opisuje implementaciju dvosmjerne audio komunikacije između Android aplikacije i ESP32 sustava, te mogućnost reprodukcije umirujuće melodije na jedinici kod djeteta. Zvuk djeteta snima se INMP441 mikrofonom i šalje aplikaciji. Korisnik aplikacije može slušati taj zvuk, ali i poslati svoj glas koji se reproducira na zvučniku spojenom na ESP2. Također, korisnik može inicirati reprodukciju predefinirane melodije na ESP2.

## 3.3. REALIZACIJA RADA

#### 3.3.1. PRIKUPLJANJE ZVUKA

ESP1 modul konfigurira I2S sučelje za rad s INMP441 mikrofonom. Postavljaju se I2S pinovi (I2S\_WS\_PIN 15, I2S\_SCK\_PIN 14, I2S\_SD\_PIN 32), brzina uzorkovanja (I2S\_SAMPLE\_RATE 16000 Hz) i 32-bitni format uzoraka (Slika 13.):

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 13. Programski isječak ESP1 - dio setup() funkcije za I2S

#### 3.3.2. SLANJE ZVUKA S ESP1 NA ANDROID APLIKACIJU

ESP1 pokreće TCP server na portu 8080. Nakon što se Android aplikacija spoji kao klijent, ESP1 kontinuirano čita podatke iz I2S međuspremnika (i2s\_read()) i šalje ih (client.write()) spojenom klijentu (Slika 14.):

Slika na kojoj se prikazuje tekst, Font, snimka zaslona

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 14. Programski isječak ESP1 - dio loop() funkcije:

#### 3.3.3. PRIMANJE ZVUKA I REPRODUKCIJA NA ANDROID APLIKACIJI

U Android aplikaciji, StreamingService (unutar ReceiveAndPlayTask) spaja se na TCP server ESP1 modula. Primljeni audio podaci (32-bitni PCM) se zatim reproduciraju pomoću AudioTrack objekta, konfiguriranog za odgovarajuću brzinu uzorkovanja i format (Slika 15.):

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 15. Programski isječak Android - StreamingService.java, dio RecordAndSendTask

#### 3.3.4. SLANJE ZVUKA S ANDROID APLIKACIJE NA ESP32

Kada korisnik želi poslati glas, StreamingService (unutar RecordAndSendTask) koristi AudioRecord za snimanje zvuka s mikrofona telefona (16-bitni PCM, 16000 Hz). Ti podaci se šalju preko TCP konekcije na ESP2 modul, koji sluša na portu 8081(Slika 16.):

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 16. Programski isječak Android - StreamingService.java, dio RecordAndSendTask

#### 3.3.5. REPRODUKCIJA ZVUKA NA ESP2 (MAX98357A I ZVUČNIK)

ESP2 modul prima audio podatke od Android aplikacije na TCP portu 8081 (unutar handleAudioClient() funkcije). Primljeni 16-bitni PCM podaci prosljeđuju se na I2S izlaz (i2s\_write()) koji je spojen na MAX98357A pojačalo i zvučnik (Slika 17.):

Slika na kojoj se prikazuje tekst, Font, snimka zaslona

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 17. Programski isječak ESP2 - dio handleAudioClient() funkcije

#### 3.3.6. IMPLEMENTACIJA REPRODUKCIJE UMIRUJUĆE MELODIJE

Kada korisnik u Android aplikaciji pritisne tipku "Play Soothing Melody", MainActivity šalje ACTION\_PLAY\_MELODY Intent prema StreamingService-u. Servis zatim pokreće SendMelodyTask. Ovaj zadatak čita audio datoteku (R.raw.soothing\_melody) iz resursa aplikacije i šalje je u dijelovima preko iste TCP konekcije (port 8081) prema ESP2 modulu, koji je zatim reproducira na zvučniku. Tijekom reprodukcije melodije, slanje zvuka s mikrofona telefona je pauzirano (Slika 18.):

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 18. Programski isječak Android - StreamingService.java, dio SendMelodyTask

# 4. ANDROID APLIKACIJA – POZADINSKI RAD I MREŽNA KOMUNIKACIJA

## 4.1. KORIŠTENI ALATI

#### 4.1.1. SOFTVER

Android Studio: Kao glavno razvojno okruženje.

Java: Kao primarni programski jezik za razvoj Android aplikacije.

Android SDK: Uključuje klase za:

Korisničko sučelje: Activity, WebView, Button, SeekBar, EditText, TextView.

Mrežnu komunikaciju: java.net.Socket, java.io.InputStream, java.io.OutputStream, java.io.PrintWriter.

Pozadinske operacije: android.app.Service, android.app.Notification, android.app.NotificationChannel, android.os.Handler, android.os.Looper.

Audio: android.media.AudioRecord, android.media.AudioTrack, android.media.AudioFormat, android.media.AudioManager, android.media.MediaRecorder.AudioSource.

Upravljanje dozvolama: androidx.core.app.ActivityCompat, androidx.core.content.ContextCompat

java.util.concurrent.ExecutorService: Za upravljanje paralelnim zadacima (mrežne operacije, reprodukcija melodije) unutar StreamingService-a kako bi se izbjeglo blokiranje glavne (UI) niti.

androidx.localbroadcastmanager.content.LocalBroadcastManager: Za internu komunikaciju (slanje statusnih poruka) između StreamingService-a i MainActivity.

XML (Extensible Markup Language): Za definiranje korisničkog sučelja (activity\_main.xml) i konfiguraciju aplikacije (AndroidManifest.xml).

## 4.2. OPIS RADA

Android aplikacija predstavlja centralnu točku interakcije korisnika sa sustavom Baby monitor. Njezina je zadaća osigurati intuitivno korisničko sučelje za sve funkcionalnosti sustava, uključujući prikaz video streama, dvosmjernu audio komunikaciju, upravljanje kamerom i reprodukciju melodije. Ključan aspekt aplikacije je njezina sposobnost da stabilno upravlja mrežnim konekcijama s tri različita ESP32 entiteta (kamera, mikrofon, zvučnik/servo). Poseban izazov i zahtjev zadatka je osigurati da korisnik može čuti zvuk djeteta (plač) čak i kada aplikacija nije aktivno prikazana na zaslonu. Ovo se postiže implementacijom ForegroundService-a koji održava mrežne veze i audio reprodukciju aktivnima u pozadini te informira korisnika o svom radu putem trajne obavijesti. Aplikacija također mora korektno rukovati dozvolama za pristup mikrofonu i prikaz obavijesti.

## 4.3. REALIZACIJA RADA

#### 4.3.1. STRUKTURA ANDROID APLIKACIJE

Aplikacija je strukturirana oko dvije glavne komponente:

MainActivity.java: Predstavlja korisničko sučelje (UI) definirano u activity\_main.xml. Odgovorna je za inicijalizaciju svih UI elemenata, obradu korisničkih unosa (npr. IP adrese ESP32 modula, interakcija s tipkama i klizačima), te pokretanje i zaustavljanje StreamingService-a. MainActivity komunicira sa StreamingService-om slanjem Intenta (npr. za pokretanje, zaustavljanje, slanje naredbi, reprodukciju melodije) i primanjem statusnih ažuriranja putem LocalBroadcastManager-a kako bi dinamički osvježila prikaz na sučelju. Također upravlja životnim ciklusom WebView-a za prikaz kamere.

StreamingService.java: Ovaj servis je ključan za održavanje mrežne komunikacije i audio obrade neovisno o tome je li MainActivity vidljiva. Pokreće se kao ForegroundService s trajnom obavijesti, što mu daje veći prioritet i smanjuje vjerojatnost da ga sustav ugasi. Servis koristi ExecutorService za pokretanje i upravljanje s više paralelnih zadataka (niti):

ReceiveAndPlayTask: Za primanje zvuka od ESP1 (mikrofon) i reprodukciju.

RecordAndSendTask: Za snimanje zvuka s mikrofona telefona i slanje na ESP2 (zvučnik).

ControlConnectTask: Za uspostavljanje i održavanje TCP veze s ESP2 za slanje kontrolnih naredbi (servo).

SendMelodyTask: Za slanje audio datoteke melodije na ESP2. Servis inicijalizira i upravlja AudioRecord i AudioTrack objektima, TCP socketima, te šalje statusne poruke MainActivity.

AndroidManifest.xml: Manifest datoteka definira osnovne karakteristike aplikacije, uključujući:

Deklaraciju MainActivity kao glavne i pokretačke aktivnosti.

Deklaraciju StreamingService-a, specificirajući da je to ForegroundService s tipovima microphone i mediaPlayback (za Android 10 i novije).

Popis potrebnih korisničkih dozvola: INTERNET (za mrežu), RECORD\_AUDIO (za korištenje mikrofona), POST\_NOTIFICATIONS (za prikaz obavijesti na Androidu 13+), FOREGROUND\_SERVICE te specifične dozvole za foreground servise.

Postavku android:usesCleartextTraffic="true" koja omogućava HTTP komunikaciju (potrebno za video stream s ESP32-CAM koji ne koristi HTTPS)

#### 4.3.2. UPRAVLJANJE TCP KONEKCIJAMA I PRIJENOS PODATAKA

Svaki mrežni zadatak (ReceiveAndPlayTask, RecordAndSendTask, ControlConnectTask) odgovoran je za uspostavljanje TCP socketa s odgovarajućim ESP32 modulom na definiranom portu i IP adresi. Koristi se java.net.Socket klasa.

ESP1 (mikrofon) šalje 32-bitne PCM audio podatke na Android. ReceiveAndPlayTask čita te podatke iz InputStream-a socketa i piše ih u AudioTrack objekt konfiguriran za AudioFormat.ENCODING\_PCM\_32BIT.

Android šalje 16-bitne PCM audio podatke (snimljene AudioRecord-om ili iz melodijske datoteke) na ESP2 (zvučnik). RecordAndSendTask i SendMelodyTask pišu te podatke u OutputStream socketa.

Android šalje tekstualne kontrolne naredbe (npr. "S1=90") na ESP2 (kontrola). ControlConnectTask uspostavlja vezu, a metoda sendControlCommand() u servisu koristi PrintWriter za slanje naredbi preko OutputStream-a socketa.

Svi zadaci implementiraju rukovanje greškama (npr. IOException, SocketTimeoutException) i šalju odgovarajuće statusne poruke MainActivity. Pri zaustavljanju servisa (stopStreamingInternal()), sve aktivne konekcije i zadaci se prekidaju, a resursi (socketi, AudioRecord, AudioTrack) oslobađaju.

#### 4.3.4. POZADINSKA OBAVIJEST I RAD APLIKACIJE

Da bi se osiguralo da se zvuk djeteta čuje i kada aplikacija nije u fokusu, StreamingService se pokreće kao ForegroundService. U metodi onCreate() ili prije poziva startForeground(), kreira se NotificationChannel (za Android 8.0+). Zatim se kreira Notification koja se prikazuje korisniku i koja obavještava da je Baby monitor aktivan. Ova obavijest također može sadržavati akciju za brzo zaustavljanje servisa.

#### 4.3.5 STRUKTURA ANDROID APLIKACIJE

Korisničko sučelje aplikacije, definirano u datoteci activity\_main.xml, osmišljeno je da bude intuitivno i pruži lak pristup svim funkcionalnostima sustava. Glavni zaslon aplikacije (Slika 19.) i (Slika 20.) sastoji se od nekoliko ključnih dijelova:

Prikaz video streama: Centralni dio zauzima WebView komponenta koja prikazuje video prijenos s ESP32-CAM modula.

Kontrole za zumiranje: Ispod video prikaza nalazi se klizač (SeekBar) za digitalno zumiranje slike.

Unos IP adresa: Dva polja za unos (EditText) omogućuju korisniku da specificira IP adrese ESP32 modula zaduženih za mikrofon (ESP1) i za zvučnik/kontrolu (ESP2).

Glavne kontrole: Tipka "Start/Stop Audio/Control Streams" (Button) služi za pokretanje i zaustavljanje svih mrežnih veza i audio prijenosa. Tipka "Play Soothing Melody" (Button) aktivira reprodukciju umirujuće melodije.

Statusni prikazi: Nekoliko tekstualnih polja (TextView) informiraju korisnika o statusu primanja zvuka s ESP32 mikrofona, statusu slanja zvuka s telefonskog mikrofona, statusu reprodukcije melodije te statusu kontrolne veze s ESP32 modulom za servo motore.

Kontrole za servo motore: Dva klizača (SeekBar), već prikazana na Slici 8, omogućuju precizno upravljanje horizontalnim i vertikalnim pomicanjem kamere. Ovako strukturirano sučelje omogućuje korisniku brz pregled stanja sustava i jednostavno upravljanje svim dostupnim opcijama."

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 19. Glavni zaslon android aplikacije-1. dio

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, broj

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika 20. Glavni zaslon android aplikacije-2. dio

# 5. ZAKLJUČAK

Ovaj završni rad uspješno je realizirao sustav "Baby Monitor" s video i audio nadzorom te upravljanjem putem Android aplikacije. Ključne funkcionalnosti, uključujući dvosmjerni audio i pomicanje kamere, implementirane su korištenjem ESP32 tehnologije. Projekt je pružio praktično iskustvo u izradi IoT rješenja i temelj je za moguća buduća unapređenja.

# LITERATURA

Espressif Systems. ESP32 Tehnička dokumentacija.

<https://www.espressif.com/en/products/socs/esp32/resources>

(pristupljeno 2. svibnja 2025)

Espressif Systems. ESP32-CAM Vodič za proizvod

<https://docs.platformio.org/en/latest/boards/espressif32/esp32cam.html>

(pristupljeno 2. svibnja 2025)

TDK InvenSense. INMP441 Datasheet.

<https://product.tdk.com/en/search/sw_piezo/mic/mems-mic/info?part_no=INMP441>

(pristupljeno 14. ožujka 2025)

Analog Devices (Maxim Integrated). MAX98357A Datasheet.

<https://www.analog.com/en/products/max98357a.html>

(pristupljeno 15. ožujka 2025)

Arduino. ESP32Servo Library Documentation

<https://github.com/jkb-git/ESP32Servo>

(pristupljeno 27. ožujka 2025)

# POPIS SLIKA

Slika 1. ESP32-CAM modul

Slika 2. ESP32 razvojna pločica

Slika 3. Servo motor SG90

Slika 4. ESP32 Motherboard

Slika 5. ESP32-CAM – inicijalizacija kamere

Slika 6. ESP32-CAM - pokretanje servera

Slika 7. Programski isječak handleControlClient() funkcije

Slika 8. Prikaz SeekBar-a za kontroliranje servo motora

Slika 9. ESP32 razvojna pločica za mikrofon

Slika 10. INMP441 digitalni I2S mikrofon

Slika 11. MAX98357A I2S mono pojačalo

Slika 12. Mali 5V zvučnik

Slika 13. Programski isječak ESP1 - dio setup() funkcije za I2S

Slika 14. Programski isječak ESP1 - dio loop() funkcije:

Slika 15. Programski isječak Android - StreamingService.java, dio RecordAndSendTask

Slika 16. Programski isječak Android - StreamingService.java, dio RecordAndSendTask

Slika 17. Programski isječak ESP2 - dio handleAudioClient() funkcije

Slika 18. Programski isječak Android - StreamingService.java, dio SendMelodyTask

Slika 19. Glavni zaslon android aplikacije-1. dio

Slika 20. Glavni zaslon android aplikacije-2. dio

**KONZULTACIJSKI LIST IZRADE ZAVRŠNOG RADA**

Ime i prezime učenika: Martin Molnar

Razred: 4.RT

Program - zanimanje: Tehničar za računalstvo

Mentor: Krešimir Kočiš

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **R. br.** |  | **DATUM KONZULTACIJE** | **SADŽAJ RADA** | **POTPIS MENTORA** |
| 1. |  | 31. listopada 2024. | Konzultacija s mentorom i definiranje teme i ciljeva rada |  |
| 2. |  | 13. prosinca 2024. | Proveden uvid u dosadašnji rad |  |
| 3. |  | 14. ožujka 2025. | Proveden uvid u dosadašnji rad |  |
| 4. |  | 8. svibnja 2025. | Završna provjera rada i upute za doradu |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Završni rad ocijenjen pozitivno

DA / NE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_