

Sustav za daljinsko otkrivanje sječe šuma pomoću zvuka

Filipović, Ivan

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:471234>

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

Download date / Datum preuzimanja: **2025-04-01**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 566

**SUSTAV ZA DALJINSKO OTKRIVANJE SJEČE ŠUMA
POMOĆU ZVUKA**

Ivan Filipović

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 566

**SUSTAV ZA DALJINSKO OTKRIVANJE SJEČE ŠUMA
POMOĆU ZVUKA**

Ivan Filipović

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 566

Pristupnik: **Ivan Filipović (0036527869)**
Studij: Računarstvo
Profil: Računalno inženjerstvo
Mentor: izv. prof. dr. sc. Vladimir Čeperić

Zadatak: **Sustav za daljinsko otkrivanje sječe šuma pomoću zvuka**

Opis zadatka:

U ovom diplomskom radu treba razviti sustav za otkrivanje sječe šuma na daljinu kroz analizu zvučnih signala, koristeći mikrokontroler STM32H7A3 i komunikacijski modul SX1272. Sustav treba obuhvaćati primjenu umjetnih neuronskih mreža za klasifikaciju zvučnih uzoraka dobivenih iz okoline, uz korištenje značajki kao što su mel-frekvencijski kepstalni koeficijenti (MFCC). Rad treba opisati konfiguraciju i implementirati sustav na mikrokontroleru. Također, treba istražiti i opisati uporabu protokola za komunikaciju s sensorima i modulima. U sklopu rada treba koristiti operacijski sustav Zephyr kao platformu za razvoj i testiranje sustava. Ako bude potrebno, treba razmotriti i potencijalnu lokalizaciju izvora zvuka koristeći konfiguraciju s četiri mikrofona.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

Uvod	1
1. Sklopovlje.....	2
1.1. Mikrokontroler STM32H7A3.....	4
1.2. Mikrofon IMP34DT05TR	5
1.3. LoRa modem SX1272	8
2. Programska podrška	9
2.1. ZephyrOS	10
2.2. Audio akvizicija.....	13
2.3. LoRa i LoRaWAN podrška	16
2.3.1. LoRa	16
2.3.2. LoRaWAN.....	17
2.3.3. Implementacija komunikacije u ZephyrOS-u	19
3. Umjetna neuronska mreža	20
3.1. Uvod u umjetne neuronske mreže	20
3.2. Implementacija umjetne neuronske mreže za detekciju zvuka motorne pile	23
3.2.1. Skup podataka	24
3.2.2. Pretprocesiranje ulaznih podataka	25
3.2.3. Definiranje modela neuronske mreže	26
3.2.4. Integracija modela u sustav	28
4. Aplikacijski sloj.....	29
4.1. Aplikacijska arhitektura.....	30
4.1.1. Sistem bodovanja u modulu scoring_system	32
4.2. Smanjivanje potrošnje	33
4.3. Opis rada sustava	34
5. Rezultati.....	36

5.1. Mjerenja potrošnje	37
5.2. Mjerenja točnosti klasifikacije	40
Zaključak	43
Literatura	44
Sažetak.....	46
Summary.....	47

Uvod

U trenutku kreiranja ovoga rada prema dostupnim informacijama Republika Hrvatska se nalazi pri vrhu ljestvice postotaka ilegalnih sječa šume u Europskoj Uniji. Procjenjuje se da je svaki dvadeseti kubni metar dobiven ilegalnim putem, odnosno neprijavljenom sječom ili krađom.

Postoji više načina za smanjenje i kontrolu nedozvoljene sječe. Neki od njih su povećavanje broja zaposlenih osoba zaduženih za vršenje kontrole, uvođenje dodatnih načina nadziranja poput kamera ili dronova. Glavni problem kod tih metoda su financije. Za realizaciju spomenutih rješenja potrebno je financirati obuku osoblja, nabavku opreme pri čemu ne treba zanemariti dugoročno održavanje opreme.

U ovome radu cilj je predstaviti još jedan način rješavanja problema, ali uz znatno manje resurse u vidu potrošnje energije i financija. Ideja je konstruirati sustav koji se bazira na edge AI¹ pristupu niske potrošnje. Sustav se sastoji od IoT² uređaja koji skuplja audio zapise, klasificira ih korištenjem umjetne neuronske mreže te po detekciji zvuka motorne pile šalje poruku na poslužitelj koristeći LoRaWAN³ tehnologiju. Kada osoba zadužena za nadziranje šuma dobije obavijest o detekciji, ona je zadužena za daljnju kontrolu, odnosno zaustavljanje sječe.

Fokus rada će biti na spomenutom IoT uređaju. Odabran je mikrokontroler STM32H7A3VIT6 na kojemu će biti korišten operacijski sustav za rad u stvarnom vremenu, ZephyrOS. Implementacija će biti opisana u teorijskom obliku, ali i u praktičnom, gdje se opisuje konkretna implementacija sustava. Fokus će biti na radu mikrofona, komunikaciji s pristupnikom i primjeni umjetne neuronske mreže.

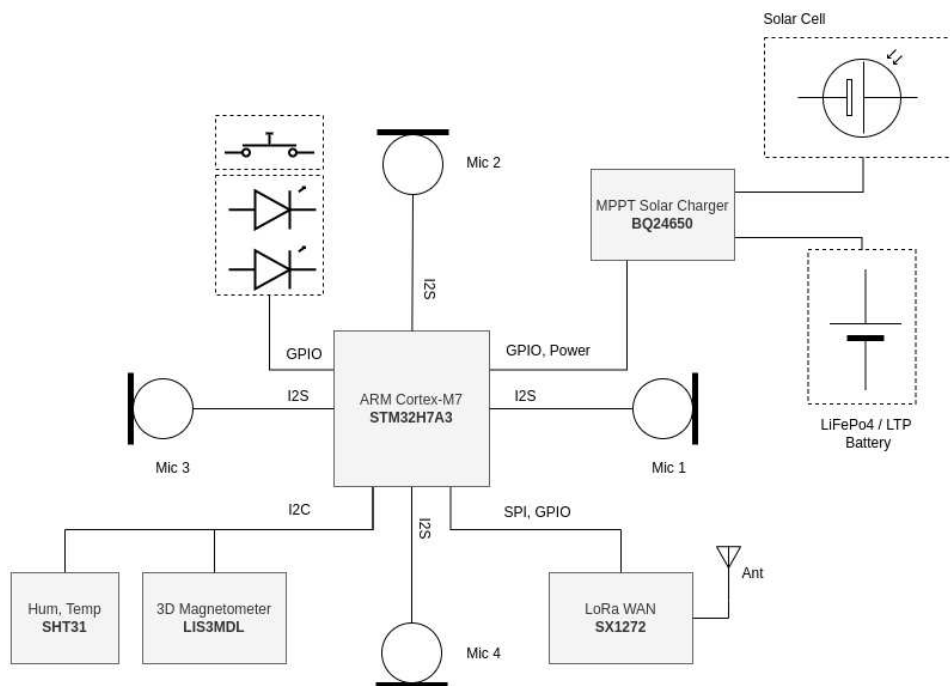
¹ Edge AI – Umjetna inteligencija na rubu mreže (obrada podataka se vrši na mikrokontroleru te se rezultat šalje na poslužitelj za daljnju obradu)

² IoT – Internet stvari (eng. *Internet of things*)

³ LoRaWAN – Bežična tehnologija za širokopojasne mreže na velikim udaljenostima (eng. *Long Range Wide Area Network*)

1. Sklopovlje

Kako bi se zadovoljili zahtjevi uređaja i poboljšao daljnji razvoj, dizajnirano je prilagođeno sklopovlje. Blok dijagram prikazan je niže (Slika 1.1), kao i prikaz finalnog izgleda sklopovlja (Slika 1.2). U ovome poglavlju proći će se kroz komponente i njihove osnovne zadaće, koje će u daljnjim poglavljima biti detaljnije opisane.



Slika 1.1 Blok dijagram sklopovlja

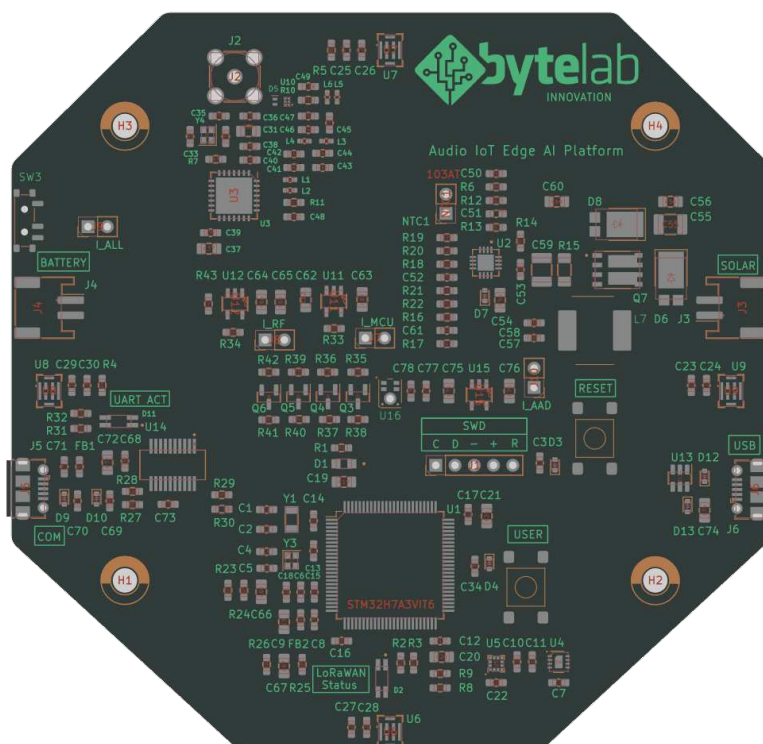
Središnja komponenta zadužena za kontrolu i povezivanje cijelog sustava je mikrokontroler STM32H7A3VIT6. Primarni razlog odabira tog mikrokontrolera je osiguranje dostatne količine resursa jer pruža visoku procesorsku snagu, te dovoljno RAM i flash memorije. U budućim iteracijama uređaja moguće je smanjivanje troškova korištenjem slabijeg mikrokontrolera.

Na mikrokontroler spojena su 4 mikrofona IMP32DT05TR koje je moguće kombinirati kako bi se ostvarila mogućnost lociranja izvora zvuka akustičkom triangulacijom. Nažalost, trenutno to prelazi opseg ovoga rada te će biti korišten samo jedan mikrofon. U sustavu postoji i peti mikrofon čija je zadaća generirati GPIO pobudu na detekciju razine zvuka definirane vrijednosti. Time bi se značajno smanjila potrošnja, no u trenutku kreiranja ovog rada to nije implementirano zbog tehničke neispravnosti dostupnih mikrofona i nemogućnosti nabavke ispravnih u potrebnom vremenskom roku.

Za svrhu lociranja izvora zvuka dodan je i magnetometar LIS3MDL koji bi se koristio kao kompas čime bi se uz poznatu početnu lokaciju uređaja i podataka mikrofona i magnetometra mogla odrediti točna geografska lokacija izvora zvuka.

Nakon prikupljanja podataka s mikrofona i obrade mikrokontrolerom, potrebno je rezultate poslati na oblak kako bi omogućili udaljeni nadzor. Za tu svrhu dodan je i RF primopredajnik (engl. *transceiver*) SX1272 koji podržava LoRa⁴ tehnologiju za komunikaciju na velike udaljenosti uz nisku potrošnju energije.

Kako bi cijeli sustav mogao raditi dugoročno bez mijenjanja baterija, uz programsku podršku za smanjenje potrošnje zadužen je i kontroler punjenja BQ24650 na kojeg su spojeni solarni panel i LTO⁵ baterija. Njegove glavne zadaće su optimiranje procesa punjenja baterije kako bi se produžio vijek trajanja iste te maksimizacija efikasnosti punjenja korištenjem MPPT⁶ tehnologije.



Slika 1.2 Prikaz projektiranog sklopovlja

⁴ LoRa – (engl. *long range*)

⁵ LTO – (engl. *Lithium-titanium-oxide*)

⁶ MPPT – praćenje maksimalne točke snage

1.1. Mikrokontroler STM32H7A3

Odabir prigodnog mikrokontrolera je jako bitan u ovakvim tipovima projekata. Potrebno je pronaći kompromis između performansi i potrošnje energije. S obzirom na potrebu korištenja neuronskih mreža koje su resursno zahtjevne pri dizajniranju sustava fokus je bio na performansama, te je ideja da se smanjenje potrošnje energije postigne programskom logikom kojom bi uređaj bio veći dio vremena u stanju mirovanja.

Kao što je u prethodnim poglavljima spomenuto, odabran je mikrokontroler STM32H7A3VIT6 koji će biti upravljan operacijskim sustavom ZephyrOS. Detaljniji opis oznaka može se iščitati u nastavku (Tablica 1.1).

Tablica 1.1 Oznake STM mikrokontrolera

Oznaka	Značenje oznake
STM32	Obitelj 32-bitnih mikrokontrolera tvrtke STMicroelectronics baziranih na ARM arhitekturi
H	Serijski mikrokontroler visokih performansi
7A3	STM32H7A3 bez kriptografskog akceleratora, optimiziran za nisku potrošnju (deklarirano 32 μ A u STOP modu)
V	100 pinova za spajanje periferija
I	Veličina flash memorije je 2 megabajta
T	Oznaka pakiranja, konkretno LQFP ECOPACK2
6	Industrijski temperaturni raspon od -40 do 85 °C

Iz gornjih podataka može se primijetiti da na raspolaganju imamo 2 megabajta flash memorije koja služi za pohranu programskog koda, kao i određenih podataka prilikom rada sustava. Uz to, bitno je naglasiti takt jezgre od 280Mhz i veličinu radne memorije od 1 megabajta.

Za integraciju sustava u cjelinu potrebno je na sučelja (pinove) mikrokontrolera spojiti komponente. Korišteni mikrokontroler podržava velik broj sučelja, od kojih su za ovu primjenu najbitniji i2s za spajanje mikrofona, SPI za spajanje LoRa modula, i2c za spajanje magnetometra i senzora temperature i vlažnosti, kao i gpio pinovi.

1.2. Mikrofon IMP34DT05TR

Sljedeća nužno potrebna komponenta ovoga sustava je mikrofon. Postoje različite tehnologije implementacije mikrofona poput dinamičkog, kondenzatorskog, elektretnog, MEMS⁷ mikrofona ili piezoelektričnog. Glavna zadaća kod svih je identična, na dolazak zvučnog vala potrebno je generirati signal na izlazu. Prema izlazu također razlikujemo analogne i digitalne mikrofone. Kod analognih nužno je programski obraditi signal kako bi iz njega dobili relevantne podatke. Digitalni su jednostavniji za upotrebu programeru jer daju standardizirani izlaz s kojim je lakše manipulirati. Na primjer, izlaz iz mikrofona može biti PDM⁸ ili PCM⁹ signal što značajno pojednostavljuje daljnju obradu.

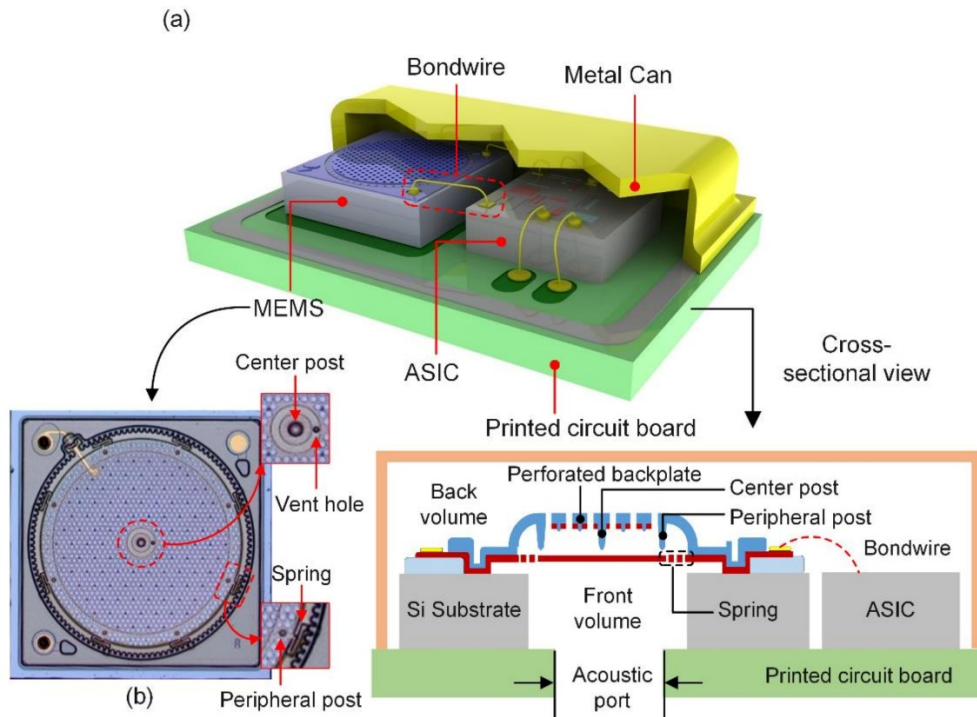
Izbor za ovaj uređaj je MEMS mikrofon IMP34DT05TR. Pogodan je zbog male potrošnje (cca. 650 μ A uz 1.8 V), relativno male cijene (trenutno između 2.00 € i 1.15 €, ovisno o veličini narudžbe) i jednostavnog korištenja (spaja se na i2s sučelje i daje PDM signal).

MEMS mikrofon (Slika 1.3) radi tako da kroz ulazni otvor ulazi zvučni val koji titra mehanički element. Taj element se sastoji od pokretne membrane i fiksirane ploče, te ujedno čini varijabilni kondenzator. Titranjem membrane zbog utjecaja zvučnih valova dolazi do promjene kapaciteta kondenzatora. S pomoću integriranog pojačala se promjene kapaciteta kondenzatora pretvaraju u varijacije napona. Uz to, integrirano pojačalo pojačava signal iz senzora i priprema ga za analogno-digitalnu konverziju gdje se, u ovom slučaju, analogni signal pretvara u PDM signal koji nadalje pretvaramo u PCM signal i dajemo kao ulaz umjetnoj neuronskoj mreži.

⁷ MEMS – engl. *Micro-Electro-Mechanical Systems*

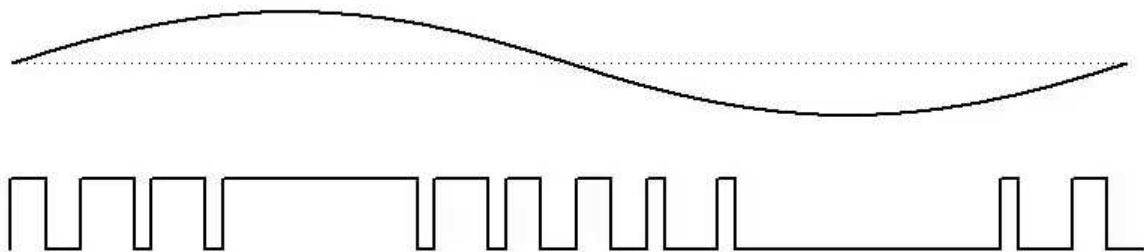
⁸ PDM – engl. *Pulse-Density Modulation*

⁹ PCM – engl. *Pulse-Code Modulation*



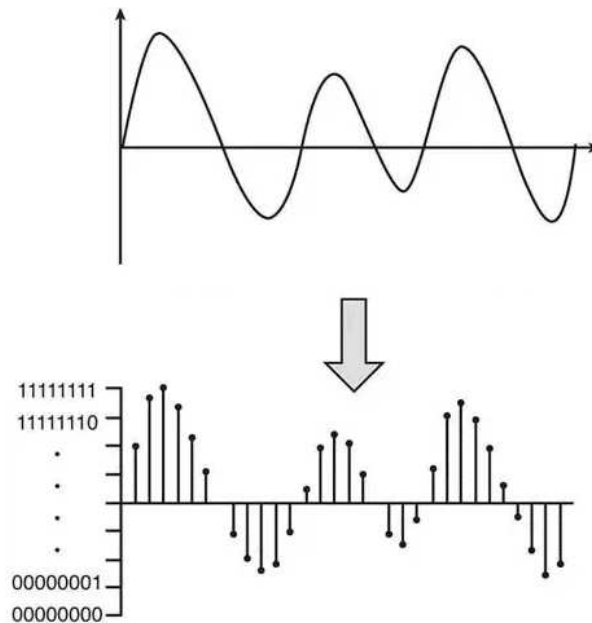
Slika 1.3 Građa MEMS mikrofona [9]

PDM signal predstavlja audio signal kao jedno bitni binarni signal, sastoji se od 0 i 1. Relativna gustoća jedinica ili nula u uzorku, odnosno visoke ili niske razine, predstavlja amplitudu analognog signala. Na donjoj skici (Slika 1.4) prikazan je odnos analognog sinusnog signala i digitalnog PDM signala. Može se uočiti odnos gustoće jedinica i nula, gdje je analogni signal „jači“, tu je digitalni signal duže u visokoj razini i obrnuto. Tipične frekvencije uzorkovanja PDM signala su u rasponu od 1 MHz do 3.072 MHz, što je dovoljno za audio frekvencijski raspon od 20 Hz do 20 kHz, odnosno raspon ljudskog sluha. U ovom sustavu koristi se frekvencija uzorkovanja od 2 MHz.



Slika 1.4 Odnos analognog sinusnog (gore) i digitalnog PDM signala (dolje) [16]

Ulaz u umjetnu neuronsku mrežu mora biti u PCM obliku. PCM signal (Slika 1.5) predstavlja analogni signal s n-bitnim binarnim vrijednostima. Tipična rezolucija signala je 8, 16 ili 32 bita uz frekvenciju uzorkovanja od 16 kHz za klasičnu upotrebu, pa do 44.1 kHz i 48 kHz za glazbenu i filmsku industriju. U ovoj implementaciji korišten je 16-bitni 16 kHz signal koji se pokazao kao optimum između kvalitete zvuka i potrošnje resursa.



Slika 1.5 Odnos analognog sinusnog signala (gore) i digitalnog PCM signala (dolje) [17]

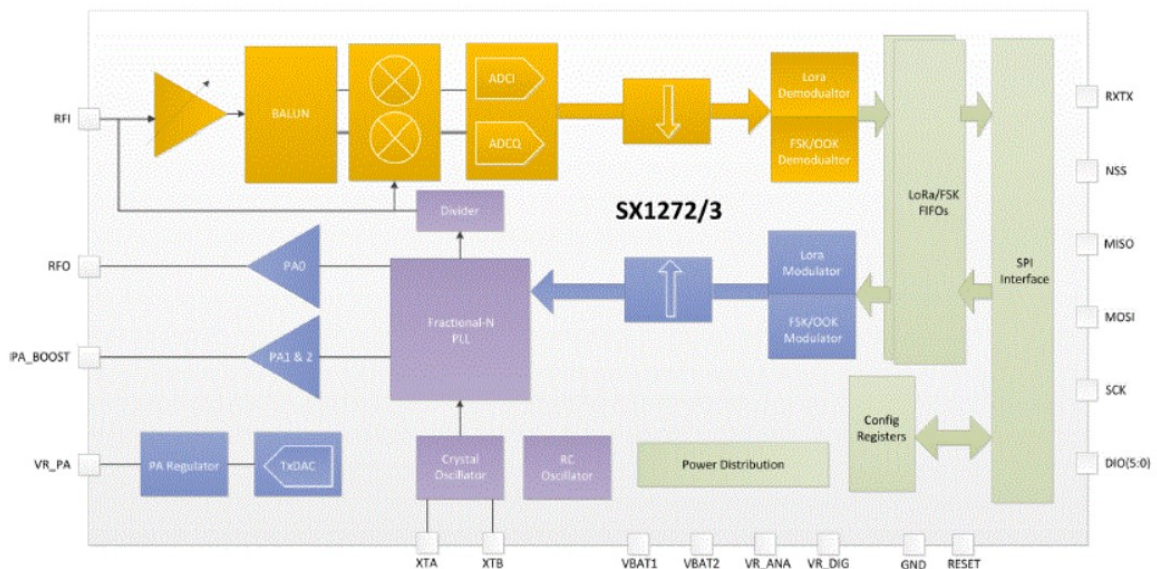
S obzirom na to da je na izlazu korištenog mikrofona PDM, a za neuronsku mrežu je potreban PCM signal, nužno je provesti postupak pretvorbe signala koji se sastoji od nekoliko koraka. Jedan od pristupa je taj da se započne provlačenjem PDM signala kroz niskopropusni filter. Time se uklanjaju šumovi visoke frekvencije, a radi tako da na temelju ulaznih n uzoraka generira prosječnu vrijednost na izlazu, pri čemu obično uzorci nemaju jednaku težinu. Nakon toga se provodi decimacija, gdje se uzimaju uzorci koji odgovaraju ciljnoj frekvenciji. Svaki n uzorak se uzima i tretira kao validan, u ovoj implementaciji je to svaki 125 uzorak ($2\text{MHz}/16\text{kHz} = 125$). Ovaj korak moguće je kombinirati s prethodnim kako bi se smanjio broj prolazaka kroz niskopropusni filter, jer bi se unaprijed moglo znati koji uzorci se mogu odbaciti. Nadalje, primjenjuje se visokopropusni filter za uklanjanje DC pomaka, cilj je da prosjek signala bude jednak nuli. Na kraju postupka moguće je korigirati pojačanje (engl. *gain*) signala tako da se svaki uzorak pomnoži s konstantnom vrijednošću.

1.3. LoRa modem SX1272

Za ključne funkcionalnosti sustava, potrebna je još jedna vrlo bitna komponenta. S obzirom na to da bi uređaj koji samo detektira zvukove i ne prosljeđuje ih dalje bio beskoristan u ovom sustavu, potrebno je dodati komponentu za komuniciranje na daljinu.

Postoje različiti pristupi rješenju toga problema i tehnologije koje bi se mogle koristiti, poput SigFox-a, NB-IoT¹⁰-a, Zigbee, Z-Wave i slično. Za ovu primjenu je potreban velik domet, niska potrošnja energije i niski troškovi implementacije. Stoga, kao idealna tehnologija odabrana je LoRaWAN mreža, o kojoj će više riječi biti u dijelu s programskom podrškom.

Za povezivanje na LoRaWAN mrežu, nužan je LoRa modem. U ovoj implementaciji to je LoRa modem SX1272 (Slika 1.6), proizvođača Semtech koji je spojen na SPI sučelje preko kojega komunicira s mikrokontrolerom. Spomenuti modem idealan je za ovu primjenu jer omogućava jednostavno korištenje (postoji programska podrška), te omogućava nisku potrošnju uz zadržavanje visoke osjetljivosti. Također je bitno napomenuti da podržava rad u frekvencijskom rasponu od 860 MHz do 1020 MHz, što uključuje nelicencirane pojaseve za Europu (868 MHz) i Sjevernu Ameriku (915 MHz).



Slika 1.6 Arhitektura LoRa modema SX1272/3 [5]

¹⁰ NB-IoT – engl. *Narrowband Internet of things*

2. Programska podrška

Kako bi prethodno opisano sklopovlje moglo funkcionirati ispravno i usklađeno, potrebno je implementirati programsku podršku za sustav i komponente. Baza većine naprednijih sustava na mikrokontrolerima obično je jedan od operacijskih sustava prilagođen za rad u stvarnom vremenu. Neki od njih su FreeRTOS, ZephyrOS, Contiki-OS, pa čak i prilagođene distribucije Linuxa. Naravno, moguće je kreirati sustav, tzv. baremetal, koji ne zahtijeva operacijski sustav, no to ne bi bilo zadovoljavajuće za ovu primjenu jer je potrebno duže vrijeme razvoja i mogućih komplikacija pri razvoju, najviše zbog potrebe za paralelnim izvođenjem više operacija. Na operacijski sustav kao osnovu dodaju se različiti upravljački programi koji i mogu, ali i ne moraju biti dio osnovnog operacijskog sustava.

Za ovu implementaciju odabran je operacijski sustav ZephyrOS u koji su ukomponirani razni dodatni upravljački programi, poput hal¹¹ funkcija za st mikrokontrolere, specifično stm32, kao i LoRaMac-Node koji je potreban za upravljanje LoRa modemom, CMSIS¹² za upravljanje periferijama i slično.

U daljnjim poglavljima pobliže će biti opisani najbitniji dijelovi programske podrške poput opisa korištenog operacijskog sustava, te podrške za upravljanje s mikrofonom i LoRa modemom. Također bit će objašnjeni postupci smanjivanja potrošnje koji su korišteni u sustavu.

¹¹ hal – engl. *Hardware abstraction layer*

¹² CMSIS – engl. *Common microcontroller software interface standard*

2.1. ZephyrOS

S obzirom na to da je ovo prva verzija projekta i na manjak osobnog iskustva za procjenu količine potrebnih resursa za ispravan rad sustava, kao idealan kandidat za operacijski sustav za rad u stvarnom vremenu nameće se ZephyrOS. Razlog tome je modularnost i mogućnost jednostavnijeg adaptiranja konfiguracije na neki novi mikrokontroler koji bi se po naučenome iskustvu mogao u budućnosti iskoristiti za smanjenje troškova i potrošnje energije.

U ZephyrOS-u već postoje “*device tree*”, skraćeno dts, datoteke čija je glavna zadaća opisati sklopovlje, njihove adrese, oznake za iznimke, kao i specifične vrijednosti registara po potrebi. Također, po uzoru na postojeće moguće je dodati i podršku za prilagođeno sklopovlje na način da se kreira “*overlay*” koji nadograđuje postojeću dts datoteku, ili da se napiše potpuno nova dts datoteka. Za ovaj projekt kreirana je nova dts datoteka naziva *lumber_guardian.dts* te je u njoj konfigurirana podrška za prilagođeno sklopovlje spomenuto u uvodu. Definirane su konfiguracije za LED diode, gumb, i2s, SPI i UART sučelja, adc komponente kao i frekvencije procesora i periferija. U nastavku slijedi isječak koji prikazuje opis iz device tree-a za uart komponentu na mikrokontroleru STM32H7A3VI.

```
/dts-v1/;
/ {
    model = "STM32H7A3VI";
    compatible = "st,stm32h7a3";
    cpus {
        cpu@0 {
            compatible = "arm,cortex-m7";
            reg = <0>;
        };
    };
    soc {
        #address-cells = <1>;
        #size-cells = <1>;
        uart1: uart@40011000 {
            compatible = "st,stm32-uart";
            reg = <0x40011000 0x400>;
            status = "okay"; current-speed = <115200>;
        };
    };
};
```

Objašnjenje oznaka u dts-u:

- ‘/’ - Koriijenski čvor
 - model : ime modela za koji je pisan
 - compatible : string koji opisuje kompatibilnost s komponentom (u .yaml datotekama su postavljena ograničenja za dts, te se po ovom parametru provjeravaju jesu li zadovoljena. Poput nužnih parametara, dozvoljenih i slično)
- ‘cpus’ - CPU čvor
 - cpu@0 : definicija CPU-a na adresi 0
 - compatible : opisano iznad
 - reg : adresa CPU-a
- ‘soc’ - SoC čvor
 - #address-cells : definira broj adresa koje će se koristiti u podčvorovima
 - #size-cells : definira veličine ćelija koje će se koristiti u podčvorovima (u jedinicama od 4B)
- ‘uart1’ - UART čvor
 - compatible : opisano iznad
 - reg : registarska adresa UART-a (0x40011000) i veličina registarskog prostora (0x400)
 - status : stanje komponente, *okay* znači da je aktivna, *disabled* da je onemogućena, *suspended* da je trenutno neaktivna i slično
 - current-speed : trenutna brzina prijenosa podataka (engl. *baud rate*)

Uz dts postoje dodatne opcije za brzu konfiguraciju poput *Kconfig* datoteke, koja omogućuje odabir specifičnih upravljačkih programa, modula i značajki, i *prj.conf*, koja sadrži korisnikove odabire opcija navedenih u *Kconfig* datotekama.

Bitno je naglasiti da ZephyrOS unutar svoje interne strukture sadrži i upravljačke programe za mnoge periferije, podsustave za mrežne protokole, upravljanje energijom i datotečne podsustave. Iznad je već spomenuto da sadrži konfiguracije za mnoge specifične razvojne ploče. Uz sve to postoje i primjeri korištenja različitih funkcionalnosti, te testovi za provjeru funkcionalnosti sustava. Cijela opisana struktura omogućava bržu prilagodbu i razvoj.

Jedan od glavnih razloga zašto koristimo operacijski sustav za rad u stvarnom vremenu u ovom projektu, umjesto pisanja "bare metal" aplikacije, je potreba za paralelnim izvršavanjem više operacija. To je najbolje vidljivo na primjeru kontinuiranog detektiranja u stvarnom vremenu, o čemu će biti više riječi kasnije.

Sustav koristi dva međuspremnik za audio podatke. Dok se jedan međuspremnik puni, drugi se koristi za klasifikaciju podataka korištenjem neuronske mreže. Bez podrške za višedretvenost i sinkronizaciju, ovakav proces bio bi vrlo teško izvediv.

Zato je operacijski sustav za rad u stvarnom vremenu ključan za efikasno upravljanje paralelnim zadacima, što omogućuje pouzdanu i točnu detekciju u stvarnom vremenu.

2.2. Audio akvizicija

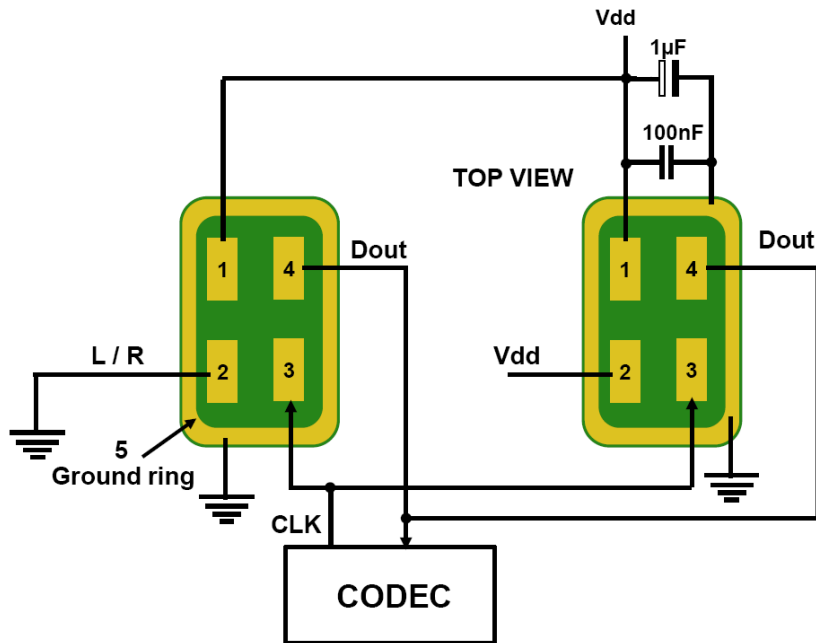
Mikrofonom opisanim u jednom od prethodnih poglavlja potrebno je upravljati i ispravno ga konfigurirati kako bi se mogle dobiti korisne informacije. Opisano je i da je izlaz iz mikrofona spojen na i2s sučelje, te da je u PDM formatu. Također je spomenut postupak pretvorbe u PCM format. U ovome poglavlju cilj je opisati proces konfiguriranja i dobivanja podataka s mikrofona.

Proces kreiranja podrške započinje pripremom varijabli za pohranu podataka i usklađivanjem s konfiguracijom mikrofona. Zbog potrebe za kontinuiranim radom u stvarnom vremenu glavni spremnik za pohranu je niz 16-bitnih cjelobrojnih varijabli veličine $audio_frekvencija \times duljina_prozora_u_sekundama \times broj_međuspreminka$. U trenutnoj konfiguraciji iznosi 32000 elemenata, odnosno 62.5 kilobajta ($16000 \times 1 \times 2$). Dakle, odabrana frekvencija zvuka je 16 kHz i rezolucija PCM signala je 16 bita.

ZephyrOS pruža podršku za digitalni mikrofonski (DMIC) koji je internim konfiguracijama povezan sa st-ovom nisko-razinskom podrškom za i2s (*i2s_ll_stm32.h*), te internim upravljačkim programom za generički mikrofonski (*mpxxdtyy.h*). Pri konfiguraciji potrebno je popuniti dmic konfiguracije za PCM tok (*pcm_stream_cfg*), te konfiguraciju samog mikrofona (*dmic_cfg*).

Konfiguracija *pcm_stream_cfg* zahtjeva definiranje PCM frekvencije, rezolucije, veličinu bloka (računa se ovisno o varijablama koje se koriste i frekvenciji) te pokazivač na blok memorije u koji se očekuje da će korisnik primiti PCM očitavanja s mikrofona. Nakon definiranja opisane konfiguracije, na red dolazi *dmic_cfg* koja se sastoji od parametara za postavljanje frekvencije (predaju se minimalne i maksimalne zadovoljavajuće vrijednosti pa se u nisko-razinskoj podršci odredi kompatibilna vrijednost), pokazivača na kreiranu konfiguraciju PCM toka, te parametri kanala.

Moguće je na jednom i2s sučelju imati 2 kanala, te ih mapirati kao lijevi ili desni. Sklopovska shema stereo mikrofona je također vrlo jednostavna, vidljivo na slici preuzetoj iz datasheeta mikrofona (Slika 2.1).



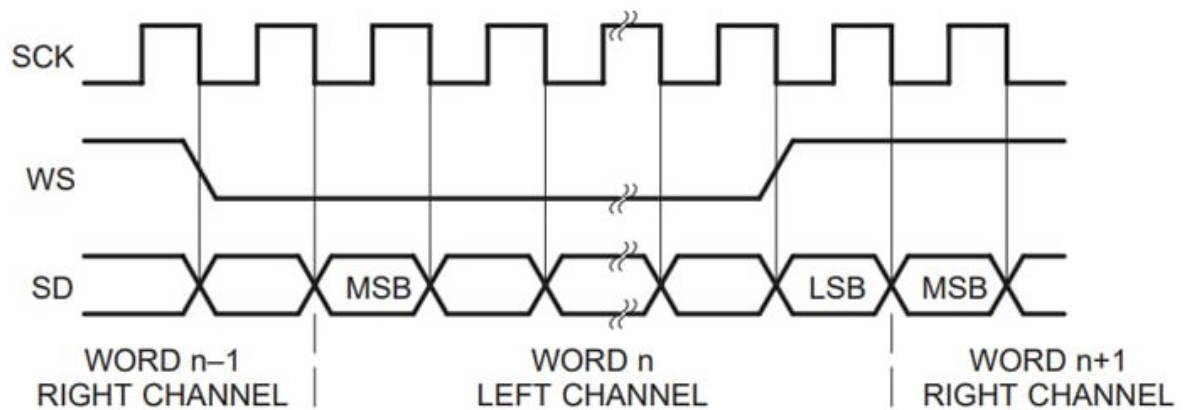
Slika 2.1 Shema spajanja mikrofona u stereo modu [2]

Nakon konfiguracije predane funkcijom *static inline int dmic_configure(const struct device *dev, struct dmic_cfg *cfg)* mikrofon je spreman za korištenje. Prije i nakon čitanja potrebno je poslati komandu dmic upravljačkom programu koristeći funkciju *static inline int dmic_trigger(const struct device *dev, enum dmic_trigger cmd)*, ta komanda može biti *DMIC_TRIGGER_START* ili *DMIC_TRIGGER_STOP*. Nakon pokretanja upravljačkog programa moguće je pozvati funkciju *static inline int dmic_read(const struct device *dev, uint8_t stream, void **buffer, size_t *size, int32_t timeout)*. Uzastopnim pozivanjem te funkcije sve dok se međuspremnik ne napuni provodimo postupak akvizicije zvuka.

Važno je napomenuti da treba biti oprezan s korištenjem, te onemogućiti blokiranje izvođenja dretve u kojoj je pokrenut dmic naredbom *DMIC_TRIGGER_START*, jer će u protivnom uređaj izgubiti sinkronizaciju s i2s-om i svako sljedeće pozivanje *dmic_read* funkcije će rezultirati pogreškom.

Za pretvorbu PDM signala u PCM zadužen je *mpxxdtyy* upravljački program, te se ona odvija automatski pri svakom čitanju i2s podatka pozivom funkcije *sw_filter_lib_run* u kojoj svaki PDM blok prolazi kroz *open_PDM_filter* funkciju.

Važan dio akvizicije zvuka je i2s protokol koji je jako dobro podržan u ZephyrOS-u kroz Zephyrove upravljačke programe i opciju uključivanja nisko-razinskih prilagođenih specifičnom sklopovlju, tj. STM32 mikrokontrolerima. I2s protokol je serijski protokol korišten za povezivanje digitalnih audio uređaja. Uz jednostavan dizajn od 3 osnovne linije (Slika 2.2) , SD (engl. *serial data* – prijenos podataka), SCK (engl. *serial clock* – sinkronizacija uređaja), WS (engl. *word select* – određivanje kanala) omogućuje stereo zvuk korištenjem jedne linije i preciznu sinkronizaciju, što rezultira visokokvalitetnim zvukom bez šuma i distorzije.



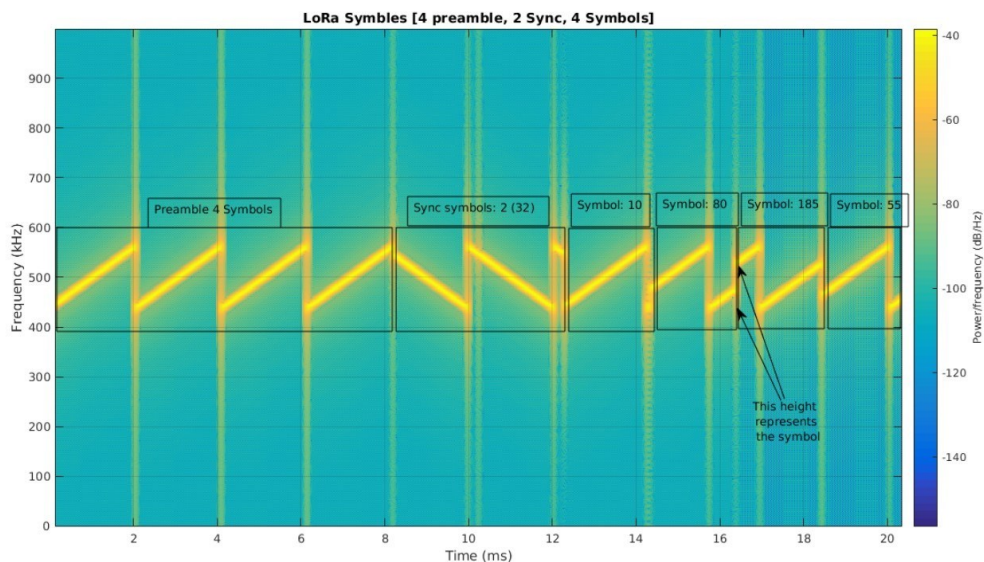
Slika 2.2 Opis i2s protokola [18]

2.3. LoRa i LoRaWAN podrška

Za povezivanje s uređajem na daljinu potrebno je prethodno opisani LoRa modem osposobiti. To uključuje konfiguraciju, te pravilno korištenje uz implementaciju LoRaWAN protokola. U ovom poglavlju ukratko će se proći kroz osnove LoRa komunikacijske tehnologije, protokol LoRaWAN, te će biti opisana implementacija u sustavu.

2.3.1. LoRa

LoRa je bežična modulacijska tehnologija izvedena iz *Chirp Spread Spectrum* (CSS) tehnologije. Enkodira podatke koristeći radiovalove u obliku tzv. chirp pulseva (Slika 2.3).



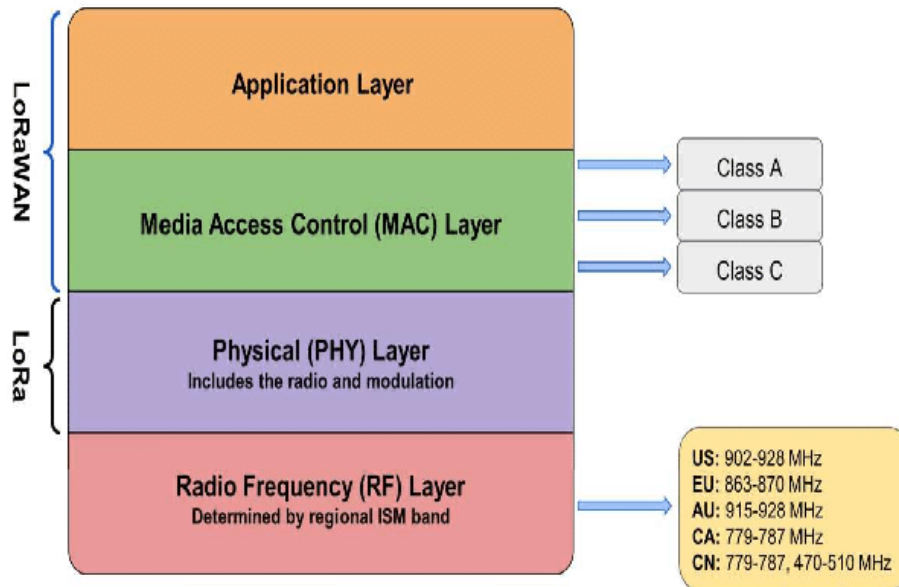
Slika 2.3 Primjer LoRa chirp pulseva [19]

Ovakav način enkodiranja omogućuje očuvanje energetske efikasnosti pri slanju informacija na velike udaljenosti. Moguće je prilagoditi brzinu prijenosa podataka (engl. *data rate*) mijenjanjem čimbenika širenja (engl. *spreading factor*). Odabir odgovarajućeg čimbenika širenja se provodi ovisno o namjeni uređaja, niži čimbenici širenja omogućuju veće brzine prijenosa, ali uz manji domet, dok je kod većih čimbenika širenja situacija obrnuta. Mijenjanje brzine prijenosa utječe i na potrošnju, što je prijenos podataka brži, to će odašiljač kraće morati biti aktivan, što rezultira manjom potrošnjom.

Korisno je napomenuti i da LoRa radi u nelicenciranom dijelu spektra, odnosno ispod jednog giga Hertza, što poboljšava domet i penetraciju signala uz mogućnost korištenja bez kupovanja licence, pa su troškovi licenciranja izbjegnuti.

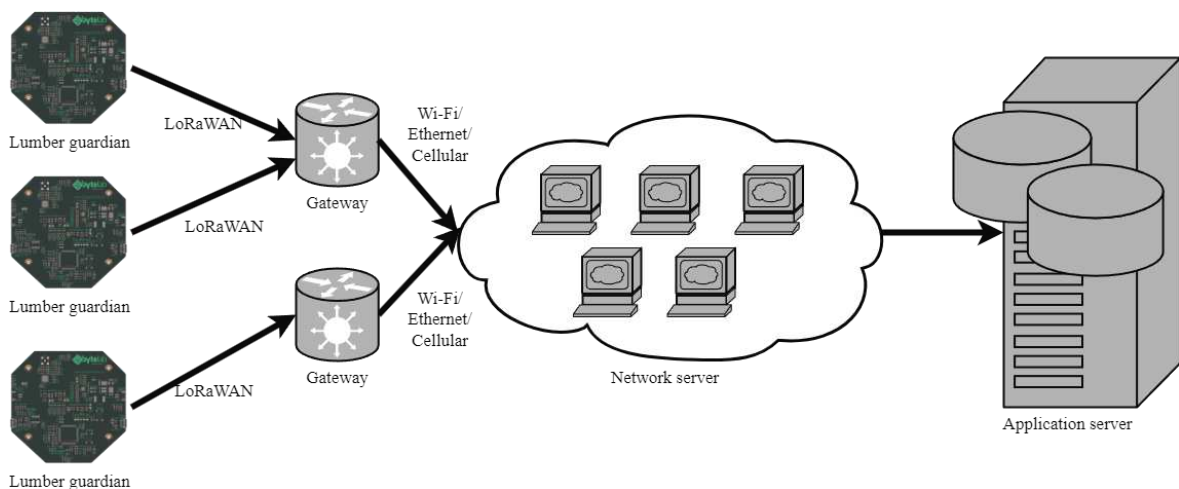
2.3.2. LoRaWAN

LoRa definira fizički sloj, dok je LoRaWAN zadužen za protokol i arhitekturu sustava (Slika 2.4).



Slika 2.4 LoRaWAN stog protokola [8]

Infrastruktura LoRaWAN mreže sastoji se od 4 glavna aktera (Slika 2.5). Krajnji uređaji, senzori ili aktuatori, šalju podatke radio signalom na pristupnik (engl. *gateway*). Zadaća pristupnika je te podatke prosljediti do centralnog poslužitelja (engl. *network server*) putem mreže, npr. 4G, 5G, Ethernet i slično. Zadnji u lancu, onaj kojem centralni poslužitelj šalje podatke, je poslužitelj aplikacija (engl. *application server*). Njegova zadaća je obraditi podatke i omogućiti usluge korisnicima na temelju tih podataka.



Slika 2.5 Infrastruktura LoRaWAN mreže

Krajnji uređaji mogu komunicirati i izravno jedan s drugim, te s jednim ili više pristupnika čime se smanjuje složenost i povećava pouzdanost komunikacije. Na mrežnom i aplikacijskom sloju koristi se AES-128, čime su zadovoljeni povjerljivost i integritet podataka.

Zbog potrebe za jednostavnijim kontroliranjem potrošnje, uvedene su 3 klase LoRa uređaja:

- Klasa A – Najmanja potrošnja, slanje podataka na uređaj moguće je samo nakon uspješnog slanja podataka s uređaja (ALOHA mehanizam)
- Klasa B – Primanje podataka na uređaju je moguće samo u raspoređenom vremenskom periodu, pristupnik šalje signal za sinkronizaciju
- Klasa C – Uređaj kontinuirano ima otvoren prozor za primanje podataka, jedino se zatvara u slučaju slanja s uređaja. Rezultira najvećom potrošnjom.

Postoje 2 glavna načina pristupa LoRaWAN mreži s rubnog uređaja, to su OTAA¹³ i ABP¹⁴. OTAA je dinamičan način aktivacije uređaja i sastoji se od 3 koraka. Uređaj šalje zahtjev za spajanjem u kojem se nalazi ID uređaja (*dev_eui*). Nakon provjere na mrežnom serveru, odgovara se s potvrdom zahtjeva u kojoj se nalaze dinamički generirane adresa uređaja na mreži i ključevi aplikacijske, te mrežne sesije. Time je osigurana veća sigurnost, promjena mreže bez pristupa uređajima i omogućena je reaktivacija uređaja. ABP je statičan način aktivacije uređaja, gdje se unaprijed konfiguriraju ključevi na uređajima i mrežnom serveru. Koriste se isti ključevi kao u već spomenutoj OTAA metodi. Ovaj pristup je dobar za aplikacije sa stabilnim uređajima, ali je veliki nedostatak manjak sigurnosti zbog mogućeg kompromitiranja ključeva, te je u slučaju potrebe za promjenom mreže ili servera potreban pristup i rekonfiguracija uređaja.

¹³ OTAA – (engl. *over-the-air activation*)

¹⁴ ABP – (engl. *activation by personalization*)

2.3.3. Implementacija komunikacije u ZephyrOS-u

U ovoj implementaciji komunikacije korištena su 2 eksterna modula. *Tinycrypt* modul omogućava jednostavno kreiranje pristupnih ključeva, a *LoRaMac-Node* implementaciju stoga rubnog LoRaWAN uređaja.

Kako bi se omogućila manja potrošnja uređaja, dodan je LDO¹⁵ regulator kojim, kada LoRa modem nije aktivan, isključuje se napajanje modema i smanjuje se potrošnju za cca 2 mA. Stoga je prvi i zadnji korak pri korištenju modema kontrola regulatora.

Prije slanja zahtjeva za spajanje na mrežu, generiraju se pristupni ključevi. Pri tome se koristi sha256 funkcija iz spomenutog *Tinycrypt* modula. Za svaki uređaj se iz njegove unutarnje strukture iščita njegov ID, enkodira se sha256 funkcijom i potom mu se na kraj dodaje Byte Labov dio MAC-a. Tako se generira *dev_eui* ključ.

Nakon što su ključevi generirani, pokreće se LoRaWAN stog na uređaju, te se postavlja klasu uređaja. Odlučeno je koristiti klasu A zbog minimizacije potrošnje uz zadovoljavanje svih zahtjeva za uređaj.

Nakon konfiguracije svih nužnih parametara, vrijeme je za pokušaj spajanja. Uređaj se iterativno pokušava spajati na mrežu, sve dok ne uspije. Nakon uspješnog spajanja, gasi se regulator i modem čeka ponovno buđenje koje će se dogoditi kada bude potrebno poslati obavijest serveru.

Slanje poruka će biti detaljnije opisano u aplikacijskom dijelu, a za sada se može napomenuti da se šalje struktura od 60 bajtova, te se koristi *data rate 3*.

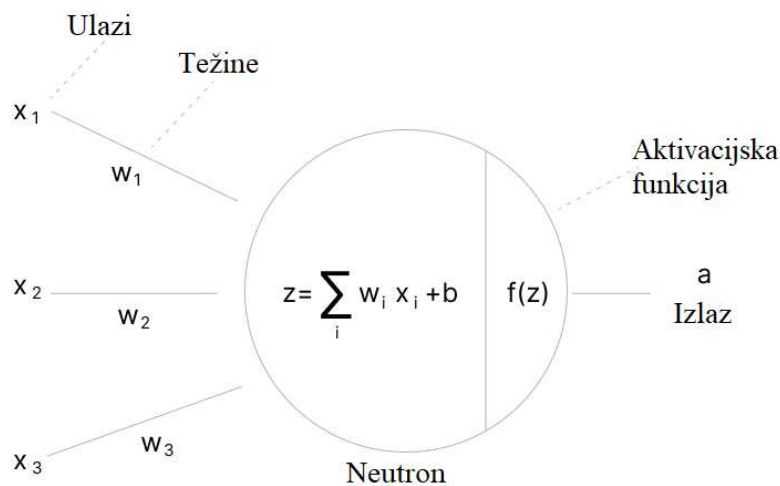
¹⁵ LDO – (engl. *low-dropout*)

3. Umjetna neuronska mreža

Za ostvarivanje ključne funkcionalnosti sustava, detekcije zvukova motorne pile, potrebno je koristiti umjetnu neuronsku mrežu. Ona omogućava da iz jednostavnog ulaza, u ovom slučaju PCM signala, generira jednostavan izlaz u obliku klasifikacije kao motorna pila ili prirodni zvuk.

3.1. Uvod u umjetne neuronske mreže

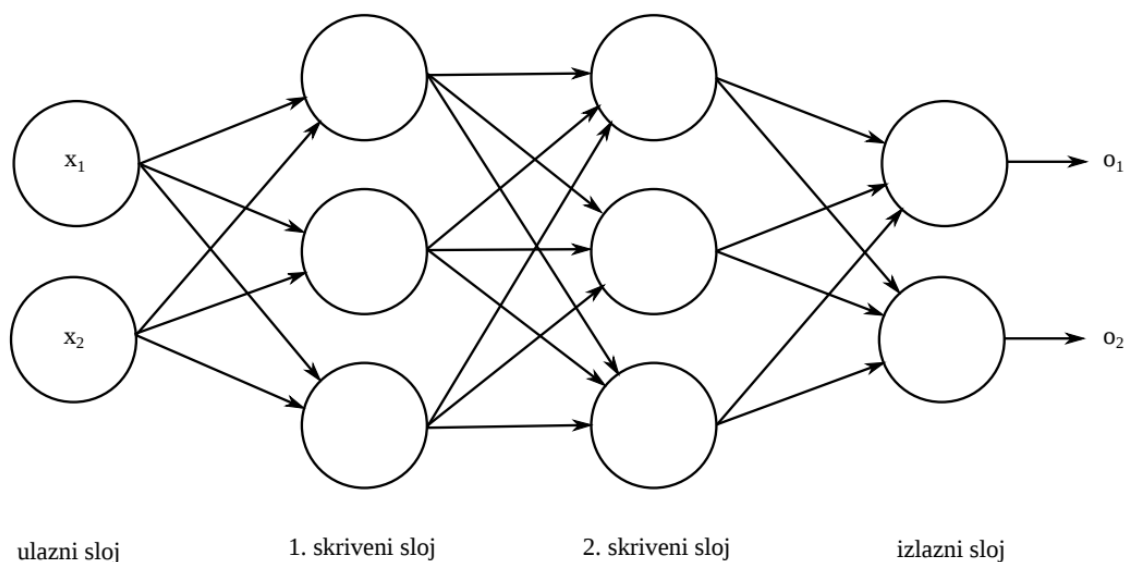
Umjetna neuronska mreža se sastoji od čvorova koji se zovu neuroni (Slika 3.1). Raspoređeni su u slojeve, te su međusobno povezani s neuronima u sljedećem sloju. Svaka poveznica ima pridruženu težinu koja se mijenja tijekom procesa učenja, odnosno treniranja mreže. Svaki neuron koristi neku aktivacijsku funkciju, poput sigmoidne, ReLU¹⁶, linearne, binarne i slično. Na ulaz neurona dovedeni su izlazi neurona iz prethodnog sloja i težine pridijeljene svakom ulazu. Izlaz svakog neurona prethodnog sloja se zasebno množi sa sebi pridijeljenom težinom, te se zbraja s ostatkom vrijednosti ulaza neurona. Ukupnoj sumi dodaje se pomak (engl. *bias*) te se ta vrijednost predaje aktivacijskoj funkciji. Rezultat te aktivacijske funkcije predstavlja izlaz neurona.



Slika 3.1 Građa neurona [20]

¹⁶ ReLU – (engl. *Rectified linear unit*) – aktivacijska funkcija oblika $ReLU(x) = \max(0, x)$

Primjer povezivanja neurona u jednostavnu slojevitu neuronsku mrežu $2 \times 3 \times 3 \times 2$ moguće je vidjeti u nastavku (Slika 3.2). Moguće je primijetiti način povezivanja težina, kao i međusobni utjecaj neurona, odnosno povezanost izlaza jednog neurona na ulaz drugog. Ulaz u ulazni sloj su podaci koje je potrebno klasificirati, a izlaz iz izlaznog sloja je rezultat klasifikacije.



Slika 3.2 Arhitektura neuronske mreže oblika $2 \times 3 \times 3 \times 2$

U koje će vrijednosti umjetna neuronska mreža preslikati ulazne podatke određeno je težinama i pragovima mreže. Postupak postavljanja i podešavanja težina i pragova naziva se treniranje neuronske mreže. Inicijalno se težinski koeficijenti postavljaju na slučajne vrijednosti. Zatim kreće unaprijedno širenje (engl. *feedforward*) gdje se ulazni podaci šire kroz neurone računajući izlaze svih neurona na temelju izlaza, kao što je prethodno objašnjeno. Razlika između stvarnog i željenog izlaza mreže kalkulira se uz pomoć funkcije gubitka, često se koristi srednja kvadratna pogreška (MSE). Potom kreće povratno širenje (engl. *backpropagation*) te se prilagođavaju težine kako bi se smanjila greška. Derivacija funkcije gubitka u odnosu na težine računa se koristeći lančano pravilo. Tako se računa gradijent koji govori u kojem smjeru treba promijeniti težine. Ažuriranje se provodi u suprotnom smjeru od gradijenta kako bi se greška smanjila. Postupak od unaprijednog širenja do povratnog širenja se ponavlja više puta za svaki ulazni uzorak u trening skupu. Jedna iteracija naziva se epoha. Broj epoha može biti unaprijed određen ili je moguće ponavljanje sve dok se ne dobije zadovoljavajuća točnost.

Skup podataka za trening se dodatno dijeli na skup za trening i validaciju. Treniranje se provodi na dijelu skupa za trening, te se po završetku treniranja koristi dio skupa za validaciju u svrhu provjere koliko je istrenirani model precizan. Rezultat treniranja moguće je prikazati matricom konfuzije koja prikazuje točnost klasifikacije (Tablica 3.1).

Tablica 3.1 Matrica konfuzije

		Stvarna vrijednost	
		1	0
Rezultat klasifikacije	1	Točno pozitivno	Lažno pozitivno
	0	Lažno negativno	Točno negativno

Po završetku treninga, nakon što su dobiveni zadovoljavajući rezultati konfuzijske matrice, poželjno je provesti testiranje kako bi se saznala stvarna preciznost kreiranog modela. Skup podataka za testiranje treba biti opširan i raznovrstan. Što je testiranje temeljitije, to će biti lakše prilagoditi model i uvidjeti greške nastale pri treniranju modela.

3.2. Implementacija umjetne neuronske mreže za detekciju zvuka motorne pile

Kao što je već istaknuto u prethodnim poglavljima, zahtjevi sustava nalažu da neuronska mreža za ulazne podatke u PCM formatu, na izlazu generira rezultat „*Chainsaw*“ ili „*Regular*“, odnosno motorna pila ili uobičajeni zvuk. Kako bi se što jednostavnije kreirao i generirao takav model, korištena je platforma Edge Impulse. Korištenjem te platforme olakšan je cijeli proces. Ova platforma značajno pojednostavljuje cijeli proces omogućujući pohranu skupa podataka, jednostavno podešavanje dizajna sustava, dodavanje blokova za pretprocesiranje, ali i dopušta detaljnu konfiguraciju modela korištenjem *TensorFlow* python biblioteke. Također, na samoj platformi je moguće vršiti treniranje i testiranje modela, te omogućava generiranje C++ biblioteke koja je korištena na mikrokontroleru.

3.2.1. Skup podataka

Kako bi neuronska mreža generirala što bolje rezultate, vrlo bitno je pripremiti dobar skup podataka nad kojima će se vršiti treniranje, validacija i testiranje. Potrebno je osigurati raznovrstan skup koji pokriva što više mogućih zvukova, primjerice u slučaju motorne pile to znači da skup mora sadržavati zvukove bez opterećenja, s opterećenjem, polovičnim opterećenjem, razne vrste motornih pila kao i razne vrste moguće okoline jer odbijanje zvuka također utječe na oblik zvuka. Iz prethodnog opisa, da se naslutiti koliko je to ustvari kompleksan zadatak. Dan je primjer samo za motornu pilu, a potrebno je pokriti i što više drugih prirodnih zvukova koji se mogu pojaviti čime se kompleksnost i količina potrebnih podataka uveliko povećava.

Prilikom implementacije skupljena su 8469 zvučna zapisa ukupnog trajanja 24 sata 32 minute i 47 sekundi. Od toga je 7376 (19 h 3 min 23 s) korišteno za trening i validaciju, a ostalih 2448 (5 h 29 min 24 s) korišteno za testiranje. Podjela prema „*Chainsaw*“ i „*Regular*“ oznakama je 6 sati 53 minute i 1 sekunda naprema 17 sati 39 minuta i 46 sekundi.

„*Regular*“ skup podataka treba sadržavati opsežan skup zvukova. Prikupljeni su zvukovi raznih ptica, vjetra, kiše, tišine, aviona, automobila, raznih životinja i slično. Problem kod ovog skupa podataka je taj što on može biti praktički beskonačan s obzirom na to da su u šumi nekontrolirani uvjeti te se može bilo što dogoditi da proizvede neki novi zvuk koji bi bio nepoznat ovakvom sustavu. Cilj je bio pokriti što više raznovrsnih zvukova koji bi omogućili što bolju generalizaciju.

Dio audio zapisa je prikupljen ručnim snimanjem uz pomoć razvojnog kompleta koji sadrži isti mikrofon kao korišteni uređaj, te se takav pristup pokazao kao vrlo koristan i njime se poboljšala preciznost sustava u praksi. Ostatak audio zapisa preuzet je sa stranica FreeSound, Kaggle i Googleovog AudioSeta.

3.2.2. Pretprocesiranje ulaznih podataka

Kao najbolja praksa primijećena kroz teorijske članke i testiranje pokazalo se da nije najbolje rješenje koristiti „čiste“, odnosno neobrađene podatke audio zapisa. Puno isplativije je koristiti dodatne blokove za pretprocesiranje ulaznih podataka. Tako se iz audio zapisa mogu izvlačiti karakteristike koje su prilagođene ljudskom uhu i efikasne su za obradu. U ovoj implementaciji korišteni su MFE¹⁷ i MFCC¹⁸ blokovi za digitalni proces signala, te su značajke generirane njima ulaz u umjetnu neuronsku mrežu.

Zadaća MFE bloka je ekstrakcija značajki iz audio signala gdje je fokus na raspodjelu energije unutar Mel-frekvencijske skale. Proces obrade MFE blokom sastoji se od nekoliko koraka. Prvo se ulazni „prozor“ dijeli na više preklapajućih okvira. Veličina i broj okvira određuju se parametrima *Frame length* i *Frame stride*. Nakon toga se za svaki okvir računa brza Fourierova transformacija (FFT). Broj FFT značajki za svaki okvir jednak je parametru $\frac{FFT\ size}{2} + 1$. Na taj način dobivamo spektrogram ulaznog audio signala. Potom se nad generiranim spektrogramom primjenjuju tzv. trokutasti filteri čime se izdvajaju frekvencijski pojasevi. Filteri su definirani parametrima *filter number*, *low frequency* i *high frequency*. Spektrogram se množi s Mel-filterbankom, gdje svaki filter izračunava ukupnu energiju u svojem frekvencijskom pojasu. Zatim se provodi logaritamska kompresija. Na taj način se oponaša ljudski sluh koji percipira razlike u intenzitetu zvuka logaritamski, a ne linearno. Također pospješuje se normalizacija varijacija u amplitudi.

MFCC blok je vrlo sličan MFE bloku, odnosno MFCC je nadogradnja na MFE blok. Prvo se rezultati dobiveni MFE procesiranjem podvrgavaju inverznoj diskretnoj kosinusnoj transformaciji (IDCT), čime se dobivaju cepstralni koeficijenti. Nakon toga je moguće komprimiranjem informacije u prvih nekoliko koeficijenata sažeti relevantne informacije o spektru. Obično se koristi prvih 12 do 13 MFCC koeficijenata, no moguće je koristiti i veći ili manji broj.

Kombiniranje MFE i MFCC blokova omogućava bolju reprezentaciju ključnih značajki čineći model robusnijim i preciznijim. Također povećava se skup ulaznih informacija što svakako poboljšava performanse modela.

¹⁷ MFE – Mel-filterbank energija

¹⁸ MFCC – Mel-frekvencijski cepstralni koeficijenti

3.2.3. Definiranje modela neuronske mreže

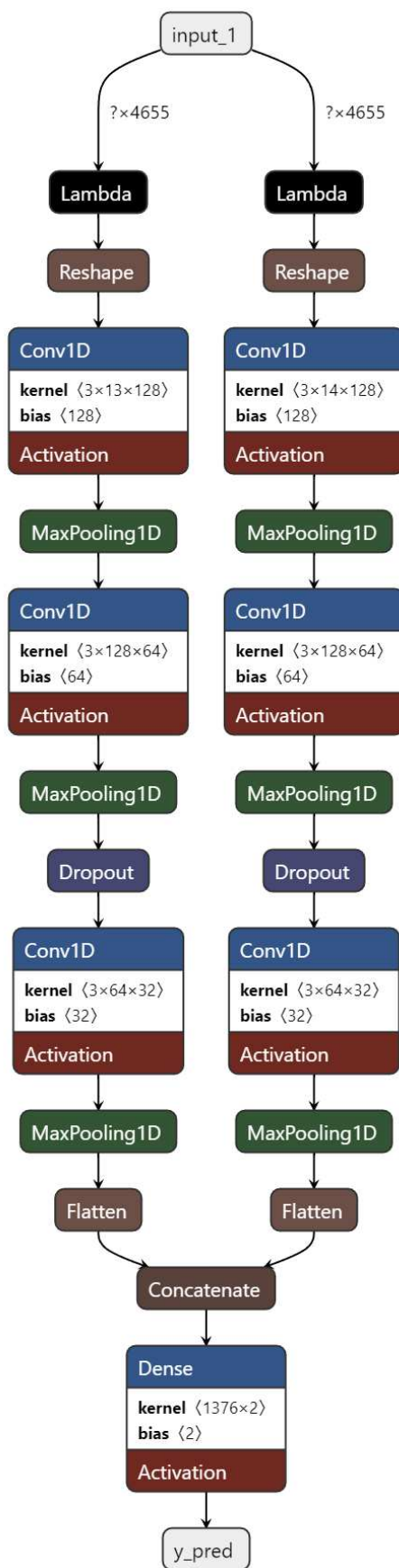
Definiranje prigodnog modela neuronske mreže koji zadovoljava zahtjeve ovog sustava je težak problem. Potrebno je kreirati model koji je dovoljno brz da osigurava rad u stvarnom vremenu pri čemu ne smije imati prevelike zahtjeve za memorijom kako bi ga bilo moguće pohraniti na uređaju. Zahtjev brzine je da osigurava rad u stvarnom vremenu, to znači da pretprocesiranje i klasifikacija ne smiju trajati duže od trajanja ulaznog zvučnog zapisa. U trenutnom modelu vrijeme trajanja ulaznog zvučnog zapisa je jedna sekunda.

Velik broj postojećih *state of the art* modela ne zadovoljava uvjete ovog sustava zbog potrebe za prevelikim resursima. Uglavnom su prihvatljivi za korištenje na računalima, ali su prezahtjevni za rad na mikrokontroleru.

Zbog navedenih razloga, odlučeno je da je najbolje rješenje kreirati vlastiti model. Edge Impulse platforma omogućuje detaljno konfiguriranje modela koristeći *TensorFlow* biblioteku u programskom jeziku Python. Prilikom postavljanja modela definiraju se i parametri treniranja, te će u nastavku ukratko biti opisan taj postupak.

Ulaz u model sastoji se od izlaza prethodno pojašnjениh blokova za digitalno procesiranje signala, MFE i MFCC (Slika 3.3). Kroz praksu je zaključeno da se bolji rezultati postižu ako se značajke tih blokova podijele na dvije grane i na kraju spoje. Obje grane se sastoje od identičnih slojeva. Na početku, ulazni podaci se preoblikuju iz formata (broj značajki ulaza, 1) u format (49,13) za MFCC značajke, odnosno (287,14) za MFE značajke. Nakon toga dolazimo do 1D konvolucijskog sloja veličine 128 konvolucijskih filtera, veličine jezgre 3×3 uz ReLU aktivacijsku funkciju. Potom se primjenjuje *pooling* sloj koji smanjuje dimenzije podataka uzimanjem maksimalne vrijednosti u svakom oknu podataka, gdje je veličina okna 2. Kombinacija prethodna 2 sloja se ponavlja, uz promjenu da se u konvolucijskom sloju koristi 64 umjesto 128 filtra. Po završetku primjenjuje se *dropout* sloj, gdje se nasumično odbacuje 25 % jedinica kako bi se spriječio problem pretreniranja. Zatim ponovno slijedi kombinacija 1D konvolucijskog i *pooling* sloja, uz smanjenje broja filtera na 32. Na kraju grane, podaci se „izravnavaju“ u jednodimenzionalni niz koji se spaja s drugom granom. Nakon spajanja grana podaci prolaze kroz još jedan sloj gusto povezanih neurona koji na izlazu daje podatak o rezultatu klasifikacije.

Treniranje se vrši na 300 epoha uz adaptivnu stopu učenja od 0.005, koja se po potrebi smanjuje sve do 10^{-10} . Ne koriste se determinističke funkcije.



Slika 3.3 Model sustava

3.2.4. Integracija modela u sustav

Korištena Edge impulse platforma pruža opciju generiranja podrške za korištenje generiranog modela u obliku C++ biblioteke. Na taj način je posao integracije uvelike olakšan.

Potrebno je u postojeći kod dodati podršku za poziv funkcije *run_classifier* kojoj predajemo neobrađene podatke s mikrofona, a koja vraća rezultat klasifikacije uz dodatne informacije koje mogu biti korisne, poput trajanja obrade pretprocesiranja, klasifikacije i slično.

Trenutačne procjene zauzeća prema Edge Impulse-ovim pretpostavkama su 85 kilobajta RAM memorije i 122.1 kilobajt flash memorije. S obzirom na prethodno spomenute performanse mikrokontrolera od 2 megabajta flash i 1 megabajta RAM memorije, možemo zaključiti da je model zadovoljavajući, te da osigurava dovoljnu količinu dodatnog prostora za korištenje pri kreiranju ostatka programske podrške i aplikacijske logike.

4. Aplikacijski sloj

Pri oblikovanju aplikacijske razine sustava fokus je bio na omogućavanju dodatne prilagodbe konkretnim zahtjevima i potrebama u praksi. Zbog toga se aplikacijska razina fokusira na što veću uporabu modula. Modul je samostalni dio aplikacijskog sustava koji obavlja specifičnu funkciju ili skup funkcija.

U ovom sustavu nalazi se 7 modula:

1. **user_interface** – omogućava kontrolu komponentama koje su izložene korisniku (gumbi, svjetleće diode i slično)
2. **audio_aquisition** – upravlja programskom potporom za mikrofonski, zadužen je za sinkronizaciju snimanja zvuka s ostatkom sustava, te održavanje dvostrukog spremnika podataka
3. **power_monitor** – kontrolira napajanje i punjenje baterije solarom, očitava parametre kontrolera punjenja i napone baterije i solarne ćelije
4. **sensors** – pruža komunikaciju sa senzorima, trenutno s magnetometrom i senzorom za temperaturu i vlažnost
5. **scoring_system** – prikuplja rezultate klasifikacije i odlučuje kada je potrebno slanje LoRaWAN obavijesti
6. **LoRaWAN** – osigurava povezivanje na LoRaWAN pristupnik i šalje LoRaWAN obavijesti pristupniku po potrebi
7. **Command shell** – koristeći UART ili RTT sučelje omogućava dodatnu kontrolu uređaja, izvršavanje prilagođenih naredbi i slično

Kroz ovo poglavlje bit će dan uvid u povezanost među modulima, slijed događaja u sustavu i pristup kojim se smanjuje potrošnja sustava.

biti u stanju mirovanja i imati vrlo malu potrošnju sve dok se ne primijeti razina glasnoće zvuka koju može proizvesti motorna pila. Sustav iz aktivnog stanja prelazi u stanje mirovanja po isteku mjerača vremena (engl. *timer*) ili po komandi pozvanoj iz *command_shell* modula. Mjerač vremena se postavlja na konfiguriranu vrijednost pri ulasku u aktivno stanje.

Prije ulaska u aktivno ili stanje mirovanja postoji još jedno implicitno stanje. To stanje je stanje inicijalizacije. Njegova zadaća je osigurati da su na pokretanju sustava sve komponente spremne za rad, te nakon što se uvjeri da je spreman, sustav prelazi u stanje mirovanja. Kada je sustav u stanju mirovanja njegova je zadaća smanjiti potrošnju što je više moguće, način na koji se to postiže bit će opisan kasnije, i čekati neku pobudu, odnosno potrebu da započne snimanje i klasifikacija zvuka. Po ulasku u aktivno stanje postavlja se već spomenuti mjerač vremena, te započinje kontinuirano snimanje i klasificiranje zvuka. Za te akcije zadužen je *audio_aquisition* modul.

audio_aquisition modul upravlja programskom podrškom za mikrofonski ulaz na način da naizmjenično puni 2 spremnika podataka, te njihove podatke prosljeđuje na ulaz u neuronsku mrežu (na slici prikazanoj gore oznaka *NN.cpp*) . Korištenjem 2 spremnika podataka omogućuje se paralelno vršenje akcija snimanja i klasifikacije. Dok se jedan spremnik puni, drugi se klasificira. Na taj način dobivamo rezultate klasifikacije u stvarnom vremenu uz odmak od trajanja jednog zvučnog zapisa, tj. jedne sekunde u trenutnoj konfiguraciji. Nakon postupka klasifikacije, pali se zelena ili crvena dioda putem *user_interface* modula, te se rezultat klasifikacije prosljeđuje modulu *scoring_system*. Obrada rezultata unutar modula *scoring_system* bit će objašnjena u sljedećem potpoglavlju. Ako se zadovolje uvjeti za slanje obavijesti o prisutnosti motorne pile *scoring_system* modul skuplja podatke od modula *power_monitor* i *sensors*, puni strukturu i prosljeđuje ju *LoRaWAN* modulu na slanje. *LoRaWAN* modul šalje poruku, ako je slanje bilo neuspješno, ponavlja se 5 puta s vremenskim odmacima. Cijeli postupak provodi se iterativno do isteka prethodno spomenutog mjerača vremena. Dok je sustav u aktivnom stanju, moguće je preko serijskog sučelja upravljati sa svim navedenim modulima. Ta opcija se pokazala vrlo korisnom prilikom razvoja i testiranja sustava.

4.1.1. Sistem bodovanja u modulu scoring_system

S obzirom na trajanje ulaznih audio zapisa neuronske mreže bilo bi besmisleno na svaku pozitivnu klasifikaciju²⁰ slati poruku obavijesti. Takav pristup rezultirao bi većom potrošnjom energije i ograničenog podatkovnog prometa koji pruža LoRaWAN standard. Također, uz korištenje neuronskih mreža uvijek postoji opcija lažne detekcije ili zavaravanje sustava zvukom drugog izvora zvuka čiji motor radi na istom principu poput motorne pile. Primjer toga je dvotaktni cross motocikl. Agregiranjem rezultata omogućava se ignoriranje takvih situacija i smanjenje nepotrebnih obavijesti. Korištene su 2 konfiguracije koje se pri kompiliranju koda mogu odabrati.

Prva konfiguracija je jednostavna. Temelji se na agregiranju pozitivnih klasifikacija unutar definiranoga vremenskog perioda. Deklariraju se 2 brojača. Svakom klasifikacijom povećava se brojač klasifikacija, a svakom pozitivnom klasifikacijom povećava se brojač pozitivnih klasifikacija. Nakon isteka vremenskog perioda, računa se postotak pozitivnih u odnosu na ukupan broj klasifikacija i ukoliko je veći od postavljene granice šalje se obavijest o detekciji ilegalne sječe i resetiraju se vrijednosti brojača.

Druga konfiguracija se sastoji od 2 sustava bodovanja. Prvi radi na principu nagrade i kazne. Broj bodova se inicijalizira na vrijednost 1. Pozitivna klasifikacija se nagrađuje na način da se broj bodova množi multiplikatorom, zadana vrijednost je 2 no moguće je konfigurirati dodatno. Negativna klasifikacija se kažnjava tako da se broj bodova umanjuje za konfigurirani postotak, zadana vrijednost je 20%, ali također postoji opcija dodatne konfiguracije. Obavijest o detekciji se šalje kada broj bodova prijeđe konfiguriranu vrijednost, zadana je 2048. Drugi sustav bodovanja se temelji na Bayesovom teoremu o vjerojatnosti. Korištenjem početnih podataka o točnosti detekcije motorne pile postavlja se vjerovanje o prisutnosti motorne pile. Svakom detekcijom, pozitivnom ili negativnom računa se utjecaj na vjerovanje o prisutnosti, pozitivnom se povećava, negativnom se smanjuje. Kada sustav ostvari 100-postotno vjerovanje da je motorna pila prisutna šalje se obavijest. Oba sustava se paralelno primjenjuju i čim prvi zadovolji svoj uvjet šalje se obavijest i reinicijaliziraju se vrijednosti.

²⁰ U ovome tekstu pozitivnim klasifikacijama smatraju se one koje znače da je detektiran zvuk motorne pile

4.2. Smanjivanje potrošnje

Da bi sustav bio održiv i dugotrajan potrebno je minimizirati potrošnju energije. Osnovne korištene ideje spomenute su u prethodnim poglavljima, no u ovome poglavlju je cilj objediniti ih i detaljnije objasniti.

Glavni pristup smanjenju potrošnje bila je podjela rada sustava na 2 stanja. Aktivno i stanje mirovanja. Cilj je što duže biti u stanju mirovanja, jer ono omogućava znatno manju potrošnju korištenjem raznih tehnika.

Ulaskom sustava u stanje mirovanja komponente se stavljaju u suspendirano stanje, u kojem ne zahtijevaju napajanje ili zahtijevaju minimalno, gasi se upravljanje putem *command_shell* modula. Također, procesor prelazi u *cdstop*²¹ stanje, gasi se glavni oscilator čime se prebacuje na sekundarni manje frekvencije. Tim pristupom omogućava se korištenje mjerača vremena i održavanje sinkronizacije sustava u stanju mirovanja.

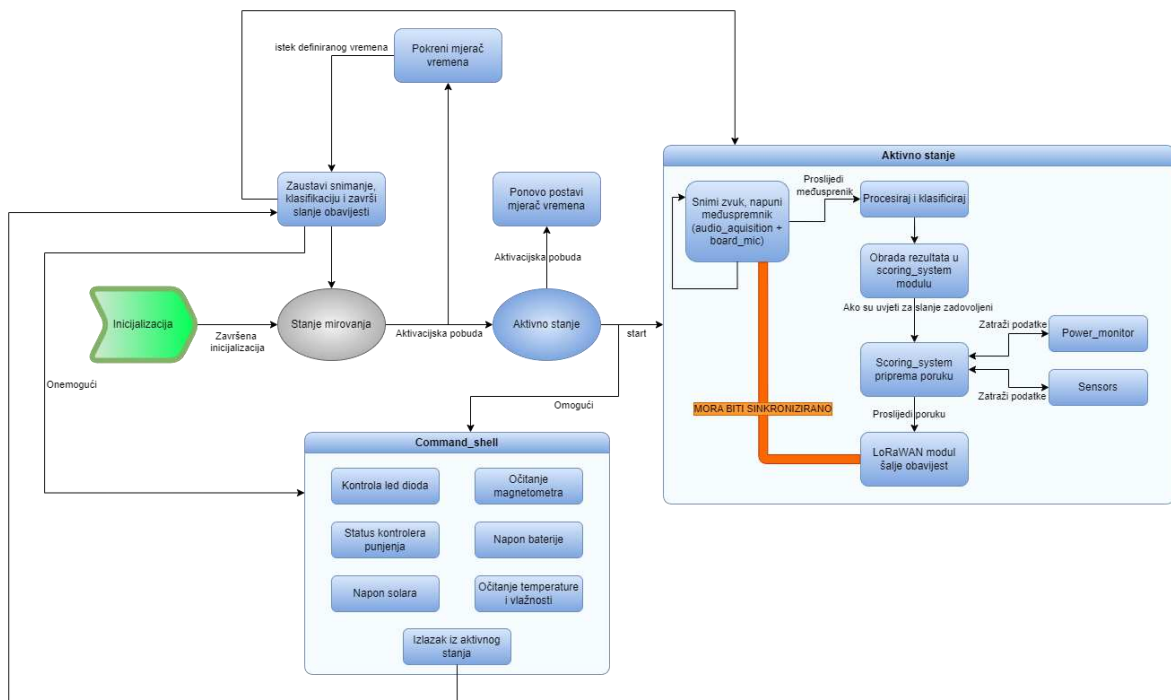
Prelaskom u aktivno stanje procesor se vraća na radnu frekvenciju i komponente postaju dostupne. Automatski se pokreće mjerač vremena i započinje prikupljanje podataka i klasifikacija.

Kroz testiranje sustava uočeno je da frekvencija kojom se postiže optimum potrošnje i pouzdanosti sustava iznosi 110 MHz. Povećanjem frekvencije ubrzava se klasifikacija i osigurava pouzdanost u smislu da će svaki međuspremnik biti obrađen prije nego li dođe na red za ponovno punjenje istog, ali se i prosječna potrošnja sustava povećava. Kada se smanjuje frekvencija procesora, klasifikacija se usporava i postoji mogućnost da se preklapa s punjenjem međuspremnik. U tom slučaju podaci neće biti validni kao ni rezultati klasifikacije. Ako izostavimo zahtjev za kontinuiranim uzorkovanjem podataka moguće je smanjiti frekvenciju i potrošnju, no onda dobivamo sustav koji za jednu sekundu audio zapisa procesuiru nekoliko sekundi čime je u konačnici moguće i povećanje potrošnje energije.

²¹ *cdstop* – stanje mikrokontrolera u kojem ima minimalnu potrošnju uz zadržavanje RAM memorije

4.3. Opis rada sustava

Kroz dosadašnje opise sustava moglo se naslutiti kojim redoslijedom se izvršavaju i na koji način su povezane operacije u sustavu. U ovom poglavlju predstaviti će se konkretne informacije o tome kako bi se poboljšalo razumijevanje. Vizualna reprezentacija rješenja vidljiva je na donjoj slici (Slika 4.2).



Slika 4.2 Dijagram toka sustava

Nakon dobivanja napajanja, uređaj započinje procesom inicijalizacije. Svaki modul zasebno inicijalizira sebi potrebne periferije i konfiguracije. Nakon što konačni automat stanja putem semafora zaprimi sve potvrde inicijalizacije, odlazi u stanje mirovanja gdje onemogućuje sve komponente i čeka GPIO pobudu. Detekcijom GPIO pobude prelazi se u aktivno stanje, ponovno se omogućuju komponente i pokreće se mjerač vremena. Po isteku vrijednosti postavljene u mjeraču vremena uređaj prelazi u stanje mirovanja. Ponovnom GPIO pobudom vrijednost mjerača vremena se resetira na početnu vrijednost.

Pri ulasku u aktivno stanje pokreće se kontinuirano snimanje i klasifikacija zvuka. Proces započinje punjenjem prvog međuspremnika. Po završetku punjenja, podaci se šalju na klasifikaciju, a započinje punjenje drugog međuspremnika. Nakon klasifikacije podatka rezultat se prosljeđuje do modula *scoring_system*. Tamo se prema jednoj od spomenutih konfiguracija obrađuje rezultat, te po potrebi šalje LoRaWAN obavijest. Kada je potrebno poslati obavijest, priprema se poruka za slanje te se pozivaju funkcije modula

power_monitor i *sensors* kako bi dobili potrebne podatke. Nakon oblikovanja poruke, ista se prosljeđuje *LoRaWAN* modulu gdje se šalje na pristupnik. Slanje LoRaWAN poruke i snimanje zvuka potrebno je sinkronizirati zbog toga što je Zephyr-ova implementacija *lorawan_send* funkcije blokirajuća, a snimanje zvuka mora biti zaustavljeno u slučaju blokiranja mikrokontrolera. Dok je sustav u aktivnom stanju omogućen je i *command_shell* modul te je moguće kontrolirati snimanje zvuka, svjetleće diode, očitavanje senzora i slično.

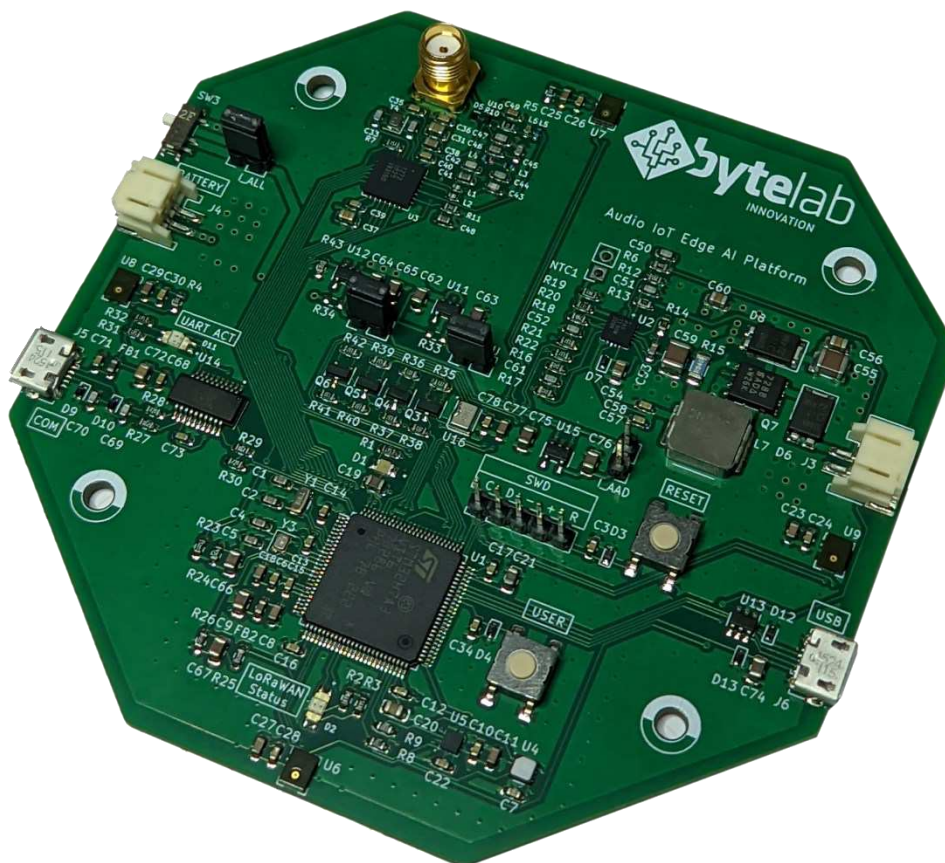
Po isteku mjerača vremena ili pozivanjem komande iz *command_shell* modula sustav prelazi u stanje mirovanja i čeka ponovnu pobudu pri kojoj se opisani proces aktivnog stanja ponavlja.

5. Rezultati

Kroz ovaj rad predstavljeno je rješenje za nadziranje ilegalne sječe šuma (Slika 5.1). Za objektivno promatranje kvalitete tog rješenja potrebno je provesti mjerenja. Stavke ovakvog sustava koje bi bilo bitno promatrati su točnost klasifikacije i potrošnja.

Potrošnja je bitna kako bi se omogućila procjena koliko dugo uređaj može raditi bez potrebe za fizičkim akcijama u vidu punjenja ili zamjene baterije. Točnost klasifikacije je važna zbog same korisnosti sustava, ako sustav ne generira točne klasifikacije gubi se njegova smisao.

U ovome poglavlju bit će opisani rezultati dobiveni početnim mjerenjima, implementirana poboljšanja i krajnji rezultati sustava.



Slika 5.1 Proizvedeno sklopovlje bez dodataka

5.1. Mjerenja potrošnje

Mjerenje potrošnje vršeno je nad cijelim uređajem pri ulaznom naponu od 3.8 V. Za osiguranje preciznosti i točnosti, te mogućnosti promatranja promjene u vremenu korišten je N6705C analizator istosmjerne struje. Vrlo korisna opcija bila je mogućnost spajanja na računalo, što je omogućilo pohranu izmjerenih vrijednosti i detaljnu analizu u vremenskoj domeni. Potrošnju će se promatrati kroz mijenjanje vrijednosti struje u sustavu.

Započnimo s promatranjem mjerenja struje sustava prije implementacije ikakvih optimizacija (Slika 5.2). Prvo što je moguće uočiti je problem prevelike potrošnje u stanju prije nego sustav uopće započne sa snimanjem i procesiranjem zvuka (crveni pravokutnik, slovo A). U tom stanju je kroz uređaj prolazila struja prosječne vrijednosti 58 mA, što je previše za stanje uređaja u kojem ne radi ništa korisno. Drugi segment mjerenja, u kojemu se koristi samo mikrofon za snimanje (plavi pravokutnik, slovo B) ima nešto manju potrošnju, vjerojatno zbog pozadinskih procesa, a to je 51 mA. Vidimo značajan porast struje prilikom klasifikacije i istovremenog punjenja drugog međuspremnika (žuti pravokutnik, slovo C). Taj period je kratak, no svejedno je problematičan zbog značajno visoke vrijednosti, u tim trenucima prolazi struja od 121.51 mA.

Moguće je uočiti da je prosječna struja tijekom iterativnog snimanja i klasifikacije korištenjem 2 međuspremnika podataka 60 mA, kao i da je za 7 punjenja i 6 klasifikacija iskorištena energija od 113.421 μ Ah.



Slika 5.2 Mjerenja struje prije optimizacije

Očito je da ako želimo održiv sustav koji će imati dugotrajnu autonomiju, odnosno da se može dovoljno napajati kombinacijom korištenog solara i baterije, moramo smanjiti trenutnu potrošnju. Tehnike koje su korištene za to su već spomenute, ali vrijedno ih je ponovno ponoviti u zbog jasnoće.

Ključan faktor bila je implementacija opcije da uređaj ide u stanje mirovanja po potrebi. U tom stanju mirovanja se periferije mikroprocesora stavljaju u suspendirano stanje, mikroprocesor ide u CDSTOP mod, smanjuje mu se napon i omogućava stanje dubokog sna. Dok je u stanju mirovanja, za interni vremenski takt se ne koristi brzi nego spori oscilator, odnosno koristi se LSE frekvencije 32768 kHz uz postavljeni *prescaler* 4. Taj oscilator se koristi za održavanje postavljenih mjerača vremena i čekanje na pobudu koja mijenja stanje uređaja. Kada se uređaj vraća u aktivno stanje, omogućuju se komponente, prebacuje se na glavni signal vremenskog takta i diže se napon procesora.

Dodatna pomoć u smanjenju potrošnje ostvarena je smanjivanjem frekvencije procesora u aktivnom stanju. Mjerenjima je ustanovljeno da se povećanjem frekvencije značajno povećava brzina obrade podataka u vidu procesiranja i klasifikacije, no problem je što se i potrošnja povećava. Bolji rezultati potrošnje su bili s nižim frekvencijama, ali je nastajao problem kada bi se one previše spustile. Prekomjernim spuštanjem frekvencija i usporavanjem obrade podataka gubi se rad u stvarnom vremenu. Razlog tome je duže vrijeme obrade podataka nego snimanja čime se javljaju kašnjenja ili nepravilnosti pri mijenjanju međuspremnika prilikom rada sustava. Optimum frekvencije koji osigurava ispravno izvođenje sustava i najnižu potrošnju je frekvencija od 110 MHz.

Na slici priloženoj u nastavku (Slika 5.3) možemo uočiti smanjenje potrošnje u odnosu na prijašnje rezultate, bez optimizacija (Slika 5.2). Unutar crvenog pravokutnika, označenog slovom A možemo primijetiti korisnost i rad sustava u stanju mirovanja. Prosječna vrijednost struje u stanju mirovanja iznosi 550 μ A, uz povremene šiljke u kojima se provjerava postoji li pobuda za izlazak iz stanja mirovanja. Ova potrošnja bi se mogla dodatno smanjiti korištenjem GPIO pobude na drugom pinu čime bi se omogućio rad korištenjem prekida umjesto trenutnog moda prozivanja u dretvi. Tada bi vrijednost struje iznosila oko 300 μ A. Prilikom uzorkovanja zvuka, odnosno samo punjenja međuspremnika prosječna vrijednost iznosa struje je smanjena na 20 mA (plavi pravokutnik, slovo B). Prosječna vrijednost iznosa struje tijekom klasifikacije i punjenja drugog međuspremnika je smanjena na 33 mA (žuti pravokutnik, slovo C). Također vrijedno je napomenuti da je prosječna vrijednost iznosa struje tijekom jednog perioda kontinuiranog rada uređaja

smanjena na 32 mA (narančasti pravokutnik, slovo D). Kroz 6 perioda aktivnog stanja sustava izmjerena je potrošnja energije u iznosu od 55.584 μAh , što je približno pola vrijednosti prije optimizacije.



Slika 5.3 Mjerenje struje uz optimizacije

Optimizacijom potrošnje značajno se produljuje trajanje baterije, smanjuje potrošnja i povećava efikasnost. Razlog tome je što se očekuje da će uređaj većinu vremena provesti u stanju mirovanja gdje je potrošnja u odnosu na stanje prije optimizacije smanjena gotovo 110 puta. Omjer struja je 60 mA prije optimizacije i 550 μA nakon optimizacije.

5.2. Mjerenja točnosti klasifikacije

Mjerenja točnosti klasifikacija unutar sustava moguće je opisati na 2 načina. Objektivni način je kroz rezultate točnosti dobivene pri treniranju i testiranju, a subjektivni način je osobni dojam u praksi. U ovome potpoglavlju biti će opisani rezultati objektivnog načina kao i tehnike testiranja korištene u subjektivnom načinu. Također bit će spomenuti utjecaj različitih pokušaja poboljšanja modela koji su korišteni.

Objektivni način mjerenja započinje treniranjem. Prilikom treniranja podešavaju se težine unutar neuronske mreže kako bi se uskladio dovedeni ulaz sa željenim izlazom iz sustava. Drugi dio treniranja sadrži i provjeru uspješnosti. U ovoj implementaciji ulazni skup se dijelimo u omjeru 80 naprema 20, te se 80 % skupa koristi za podešavanje težina, a preostalih 20 % se dovodi na ulaz mreže te se bilježi rezultat na izlazu, čime dobivamo točnost pri treniranju. Rezultat toga prikazuje se matricom konfuzije spomenutoj u prijašnjim poglavljima (Tablica 5.1).

Tablica 5.1 Rezultat treniranja modela u obliku matrice konfuzije

		Stvarna vrijednost	
		Chainsaw	Regular
Rezultat klasifikacije	Chainsaw	94.2 %	5.8 %
	Regular	3.0 %	97.0 %
F1 mjera		0.93	0.97

Ukupna preciznost sustava na treniranju iznosi 96.3 % uz vrijednost funkcije gubitka 0.16. Možemo uočiti da ova matrica konfuzije u odnosu na prethodno spomenutu sadrži red više, F1 mjeru. F1 mjera predstavlja harmonijsku sredinu između preciznosti i odziva, računa se formulom $F1 = 2 \times \frac{\text{preciznost} \times \text{odziv}}{\text{preciznost} + \text{odziv}}$. Preciznost je odnos između točno predviđenih pozitivnih uzoraka i svih uzoraka koji su predviđeni kao pozitivni, a odziv odnos između točno predviđenih pozitivnih uzoraka i svih stvarno pozitivnih uzoraka.

Drugi dio objektivnog mjerenja točnosti je testiranje istreniranog modela nad većim skupom podataka, odnosno nad skupom predviđenim za testiranje. Za ovaj slučaj to su 2448 zvučna zapisa ukupnog trajanja 5 sati 29 minuta i 24 sekunde. Ukupna točnost modela iznosi 93.91 % , a dobivena matrica konfuzije nalazi se u nastavku (Tablica 5.2). Možemo

primijetiti dodatni stupac u odnosu na prijašnje, taj stupac predstavlja nedefinirani izlaz modela. Nedefinirani izlaz se dobiva kada se modelom nije moguće odlučiti sa sigurnošću za izlaznu vrijednost. Na primjer to je situacija kada je 55 % siguran a je *chainsaw*, a 45 % siguran da je *regular*.

Tablica 5.2 Rezultat testiranja modela u obliku matrice konfuzije

		Stvarna vrijednost		
		Chainsaw	Regular	Nedefinirano
Rezultat klasifikacije	Chainsaw	89.3 %	8.5 %	2.3 %
	Regular	2.8 %	96.8 %	1.4 %
F1 mjera		0.91	0.96	

Subjektivni način testiranja provoden je kroz više testova. Prvi i najjednostavniji test bio je prilikom razvoja. Po završetku integriranja testnog modela preko zvučnika su puštani razni zvukovi uključujući i motornu pilu i nasumične ostale zvukove poput zvukova automobila, ptica, medvjeda, kiše i slično. U tu svrhu je odabrano 40 YouTube videa te se za svaku verziju modela bodovao rezultat prema kategorijama zvuka. Tim pristupom ubrzan je proces razvoja i uočeno je što nedostaje u skupovima podataka za treniranje. Drugi test bio je s aktivnim prepoznavanjem stvarne motorne pile. Tim testiranjem uočeno je da je vrlo bitno opterećenje motorne pile. Što je opterećenje veće, to će se pila glasnije čuti i domet detekcije bit će veći. Treći test bilo je promatranje zvukova bez prisustva motorne pile. Ostvaren je tako da se ukinuo mjerač vremena koji uređaj prebacuje u stanje mirovanja i da se dodao brojač ukupnih pozitivnih klasifikacija. U prosjeku su dobivani rezultati ispod 8 %, a na finalnom i korištenom modelu je bilo 5.35 % pri promatranju u trajanju od 40 sati. Veći fokus i cilj pri razvijanju nije bio na potpunom uklanjanju lažnih pozitivnih klasifikacija nego na sprječavanju sustava da pošalje lažnu obavijest o prisustvu motorne pile. Kombinacijom treniranja i implementacijom sustava bodovanja dobiveni su zadovoljavajući rezultati.

Prilikom razvoja modela najveći napredak se pokazao povećanjem i prilagođavanjem skupa za treniranje nakon što je uspostavljena stabilna konfiguracija modela. Povećanjem kompleksnosti modela javljao se problem resursa, procesiranje i klasifikacija traje predugo, a ne rezultira dovoljnim povećanjem preciznosti da bi se odbacio

zahtjev za radom u stvarnom vremenu. Prilikom stvaranja skupa podataka za treniranje uočeno je da prilikom procesiranja i klasifikacije značajnu ulogu ima i vrsta mikrofona kojim je sniman zvuk. Poboljšanje rezultata ostvarilo se presnimavanjem skupa podatka koristeći mikrofoni identičan korištenom na uređaju. Također, pri početku razvoja ustanovljeno je da objektivni način nije dovoljan za testiranje sustava i dobivanje stvarne slike kako će raditi u stvarnom svijetu. Razlog tome je što postoji mogućnost da korišteni skup podataka nema dovoljno informacija o mogućim zvukovima, te da se prilikom dolaska novog tipa zvuka ponaša nepredviđeno. Na primjer, kada se koristio manji skup podataka, recimo bez ptica, objektivni rezultati točnosti su bili iznad 98 %, no kada bismo na ulaz doveli njemu nepoznati zvuk ptica on bi ih klasificirao kao *chainsaw*.

Zaključak

Kroz ovaj rad predstavljeno je rješenje problema ilegalne sječe šuma. Kreiran je sustav koji se bazira na edge AI pristupu. Sastoji se od uređaja na rubu mreže koji je zadužen za snimanje zvuka i klasifikaciju, kao i slanje obavijesti do pristupnika koji ju prosljeđuje mrežnom poslužitelju gdje se obrađuje i s pomoću aplikacijskog poslužitelja pruža na korištenje korisniku.

Fokus pismenog dijela rada bio je na opisu krajnjeg uređaja kroz opis sklopovlja, programske podrške i pregleda integracije konvolucijske neuronske mreže na mikrokontroleru. Za najbitnije komponente objašnjena je teorijska pozadina i implementacija u praksi.

Rezultati rada koje je vrijedno istaknuti su točnost klasifikacije modela konvolucijske neuronske mreže od 93.91 % i potrošnja od približno 9.264 μAh po periodu koji se sastoji od klasifikacije i punjenja jednog međuspremnik. Bitna je i informacija da u stanju mirovanja uređaj koristi struju od 550 μA .

Sadašnji rezultati su zadovoljavajući, no postoji prostor za napredak i poboljšanja. Moguće je probati zamijeniti mikrokontroler s nekim slabijim kojim bi smanjili potrošnju, moguće je i povećati skup podataka za treniranje kako bi se dobili još točniji rezultati i slično. Zanimljiva poboljšanja moguća su uz implementaciju ostala 4 mikrofona što nažalost zbog vremenskog roka nije bilo moguće ostvariti. Time bi se omogućile razne opcije, od značajnog smanjenja potrošnje do opcije za točno lociranje izvora zvuka u odnosu na poziciju uređaja. Također, korištenjem kvalitetnijih mikrofona pretpostavlja se da bi se ostvario veći domet i bolja preciznost sustava.

Literatura

- [1] *Hrvatska pri vrhu po broju ilegalnih sječa šuma: U subotu potrubite i upozorite na problem.* Zagreb.info. (2019). Poveznica: <https://www.zagreb.info/vijesti/hrvatska-pri-vrhu-po-broju-ilegalnih-sjeca-suma-u-subotu-potrubite-i-upozorite-na-problem/231887/>; pristupljeno 12. travnja 2024.
- [2] IMP34DT05 MEMS Microphone. STMicroelectronics.
Poveznica: <https://www.st.com/en/mems-and-sensors/imp34dt05.html>; pristupljeno 15. travnja 2024.
- [3] Jurafsky, D., Martin, J. H. *Speech and Language Processing*. 3. izdanje. Stanford: Stanford University, 2024.
- [4] Rabiner, L., Schafer, R. W. *Introduction to Digital Speech Processing*. 1. izdanje. New York: Now Publishers Inc., 2007.
- [5] *SX1272 LoRa Transceiver.* Semtech.
Poveznica: <https://www.semtech.com/products/wireless-rf/lora-connect/sx1272>; pristupljeno 20. travnja 2024.
- [6] *What is LoRaWAN? The Things Network.*
Poveznica: <https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/>; pristupljeno 14. svibnja 2024.
- [7] *Expert Series: 5 Things You Need to Know About LoRaWAN-Based Gateways.* Semtech. (2024). Poveznica: <https://blog.semtech.com/expert-series-5-things-you-need-to-know-about-lorawan-based-gateways>; pristupljeno 18. svibnja 2024.
- [8] Saqib, N., Haque, K., Yelamarthi, K., Yanambaka, P., & Abdelgawad, A. *D2D-LoRa Latency Analysis: An Indoor Application Perspective.* WF-IoT. (202)
- [9] Shubham, S., Seo, Y., Naderyan, V., Song, X., Frank, A. J., Johnson, J. T. M. G., da Silva, M., & Pedersen, M. *A Novel MEMS Capacitive Microphone with Semiconstrained Diaphragm Supported with Center and Peripheral Backplate Protrusions,* Micromachines, 13,1 (2022)
- [10] Andreadis, A., Giambene, G., Zambon, R. *Monitoring Illegal Tree Cutting through Ultra-Low-Power Smart IoT Devices,* Sensors, 21,22 (2021)

- [11] Arevalo, J. D. C., Calica, P. C., Celestino, B. A. D. R., Dimapunong, K. A., Lopez, D. J. D., & Austria, Y. D. *Towards Real-Time Illegal Logging Monitoring: Gas-Powered Chainsaw Logging Detection System using K-Nearest Neighbors*. Proceedings of the 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), (2020)
- [12] Mporas, I., Perikos, I., Kelefouras, V., & Paraskevas, M. *Illegal Logging Detection Based on Acoustic Surveillance of Forest*. Applied Sciences, 10,20 (2020)
- [13] Andreadis, A., Giambene, G., & Zambon, R. *Monitoring Illegal Tree Cutting through Ultra-Low-Power Smart IoT Devices*. Sensors (Basel), 21,22 (2021)
- [14] Marin, F., & Marin, M. *Chainsaw Sound Detection Using DNN Algorithm*, The Annals of “Dunarea de Jos” University of Galati Fascicle IX Metallurgy and Materials Science, 46 (2024)
- [15] Kalhara, P. G., Jayasinghearachchi, V. D., Dias, A. H. A. T., Ratnayake, V. C., Jayawardena, C., & Kuruwitaarachchi, N. *TreeSpirit: Illegal logging detection and alerting system using audio identification over an IoT network*. Proceedings of the 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Malabe, Sri Lanka, (2017)
- [16] GreenPAK. *Class D Audio Amplifier*. Hackster.io, (2019). Poveznica: <https://www.hackster.io/greenpak/class-d-audio-amplifier-914d2a>; pristupljeno 01. lipnja 2024.
- [17] SWPU, Petroleum. *Optimizing sampling frequency of surface and downhole measurements for efficient stick-slip vibration detection*, Petroleum, 10 (2024)
- [18] *All About Circuits. Introduction to the I2S Interface*. (2021). Poveznica: <https://www.allaboutcircuits.com/technical-articles/introduction-to-the-i2s-interface/>; pristupljeno 03. lipnja 2024.
- [19] Gao, R., Zhao, Y., & Zhang, Y. *Federated Learning for Internet of Things: Concept, Applications, and Challenges*. Proceedings of the IEEE, 108,10 (2020)
- [20] Yang, K. *Evolving Minds: The Metamorphosis of AI Models from MLP to Transformer (Part 1)*. Medium. (2021). Poveznica: <https://medium.com/@yangkewenvigorous/evolving-minds-the-metamorphosis-of-ai-models-from-mlp-to-transformer-part-1-11db8ef3ac8d>; pristupljeno 25. svibnja 2024.

Sažetak

Ovaj rad opisuje sustav za otkrivanje ilegalne sječe šume. Detekcija ilegalne sječe temelji se na zvučnim zapisima koji se s pomoću mikrokontrolera prikupljaju i obrađuju. Ako su zadovoljeni uvjeti i smatra se da je prisutna motorna pila, šalje se obavijest. Za prikupljanje zvukova koristi se mikrofonski spojnik na I2S sučelje, a za obradu snimki koriste se digitalni procesirajući blokovi MFE i MFCC kao i umjetna konvolucijska neuronska mreža. Slanje obavijesti šalje se putem LoRaWAN mreže korištenjem LoRa modema spojenog na SPI sučelje mikrokontrolera STM32H7A3VIT6.

Uspješno je realiziran uređaj s točnošću predikcije od 93.91 % uz potrošnju ispod 10 μ Ah po periodu snimanja i klasifikacije.

Summary

This paper describes a system for detecting illegal logging. The detection of illegal logging is based on audio recordings that are collected and processed using a microcontroller. If the conditions are met and a chainsaw is detected, a notification is sent. A microphone connected to the I2S interface is used to collect the sounds, and digital processing blocks MFE and MFCC, as well as an artificial convolutional neural network, are used to process the recordings. Notifications are sent via the LoRaWAN network using a LoRa modem connected to the SPI interface of the STM32H7A3VIT6 microcontroller.

A device with a prediction accuracy of 93.91% was successfully implemented, with a consumption of less than 10 μ Ah per recording and classification period.