

# DOPPLER RADAR SENZOR ZA DINAMIČKU DETEKCIJU POKRETA S PRILAGODLJIVIM ZVUČNIM ODGOVOROM

---

Kalajžić, Pjero

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:228:156534>

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

Download date / Datum preuzimanja: **2025-01-06**



Repository / Repozitorij:

[Repository of University Department of Professional Studies](#)



UNIVERSITY OF SPLIT



**SVEUČILIŠTE U SPLITU**  
**SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE**

Preddiplomski stručni studij Elektronika

**PJERO KALAJŽIĆ**

**ZAVRŠNI RAD**

**DOPPLER RADAR SENZOR ZA DINAMIČKU  
DETEKCIJU POKRETA S PRILAGODLJIVIM  
ZVUČNIM ODGOVOROM**

Split, Rujan 2024

**SVEUČILIŠTE U SPLITU**  
**SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE**

Preddiplomski stručni studij Elektronika

**Predmet:** Ugradbenivračunalni sustavi

**ZAVRŠNI RAD**

**Kandidat:** Pjero Kalajžić

**Naslov rada:** Doppler radar senzor za dinamičku detekciju pokreta s  
prilagodljivim zvučnim odgovorom

**Mentor:** dr. sc. Marko Vukšić, prof.s.s. u t.i.

Split, Rujan 2024

# SADRŽAJ

Sažetak .....	1
Summary .....	2
1. UVOD .....	3
2. OPIS PROJEKTOG ZADATKA .....	4
3. ARHITEKTURA PROJEKTOVIH KOMPONENTI .....	6
3.1. Arduino Uno .....	6
3.2. HLK-LD1115H .....	8
3.3. Niskopropusni (RC) filter .....	10
3.4. Pulsno-širinska modulacija (PWM) .....	12
3.5. PAM8610 .....	14
3.6. Piezo zvučnik .....	17
4. SCHEMA SPAJANJA .....	18
5. UPUTE ZA KONFIGURACIJU SENZORA .....	20
5.1. Raspored registara za svaki parametar .....	22
6. KONFIGURACIJA KODA .....	25
7. ZAKLJUČAK .....	32
8. LITERATURA .....	33
9. POPIS SLIKA .....	34

## Sažetak

### DOPPLER RADAR SENZOR ZA DINAMIČKU DETEKCIJU POKRETA S PRILAGODLJIVIM ZVUČNIM ODGOVOROM

U ovom radu detaljno se analizira rad HLK-LD1115H senzora akceleracije koji je primjenjen za detekciju pokreta te na temelju udaljenosti i jačine pokreta se reproducira melodija s prilagodljivim parametrima. Ovaj senzor koristi Dopplerov efekt za detekciju ljudske prisutnosti i pokreta, te s njim nije moguće direktno mjeriti udaljenost kao nekim drugim ultrazvučnim senzorima. Udaljenost na kojoj senzor prepoznaje pokret bit će regulirana promjenom parametara namjenjenih za namještanje osjetljivosti senzora koji su upisani u njegove registre. Nadalje, detaljno se opisuje svaki registar senzora te funkcija u kodu kojom se pristupa svakom od njih. Osim senzora za pokret kao mikrokontroler koristi se Arduino Uno za obradu primljenog signala iz senzora koji se iz digitalnog oblika pretvara u analogni koristeći niskopropusni RC filter i dalje se pojačava pomoću pojačala D klase. Na jednom od kanala pojačala priključen je zvučnik za reprodukciju zvuka. Razvojni alat za pisanje koda u kojem je napisan kod je Arduino IDE, a odabrani programski jezik je C++. Nakon izvedbe cjelokupnog projekta, zaključeno je kako je ovo vrlo praktičan rad sa širokom primjenom u stvarnom životu. Osim kontrole zvuka, sličan pristup može se upotrijebiti i za kontrolu rasvjete, upravljanje brzinom motora i sl. Najveći izazov je kalibracija parametara senzora prema vlastitim potrebama. Treba napraviti velik broj mjerenja za što preciznije podatke. Također, uloženo je dosta truda i vremena za izračun vrijednosti pojedinih komponenti korištenih u radu. Za realizaciju projekta, osiguran je stabilan napon napajanja pojačala, prilagođen je RC filter, odabrani su prikladan zvučnik i potenciometar za kontrolu jačine zvuka.

**Ključne riječi:** Dopplerov efekt, RC filter, C++

## Summary

### DOPPLER RADAR SENSOR FOR DYNAMIC MOTION DETECTION WITH ADJUSTABLE SOUND RESPONSE

In this undergraduate dissertation, the operation of the HLK-LD1115H acceleration sensor, which is used for motion detection, is analyzed in detail, and based on the distance and strength of the movement, a melody with adjustable parameters is reproduced. This sensor uses the Doppler effect to detect human presence and movement, and it is not possible to directly measure the distance like some other ultrasonic sensors. The distance at which the sensor detects movement will be regulated by changing the parameters intended for adjusting the sensitivity of the sensor, which are written in its registers. Furthermore, each sensor register and the function in the code that accesses each of them is described in detail. In addition to the motion sensor as a microcontroller, an Arduino Uno is used to process the received signal from the sensor, which is converted from digital to analog using a low-pass RC filter and further amplified using a class D amplifier. A speaker for sound reproduction is connected to one of the channels of the amplifier. The development tool for writing the code in which the code is written is Arduino IDE, and the chosen programming language is C++. After the implementation of the entire project, it was concluded that this is a very practical work with wide application in real life. In addition to sound control, a similar approach can be used for lighting control, engine speed control, etc. The biggest challenge is to calibrate the sensor parameters according to your own needs. A large number of measurements should be made for as precise data as possible. Also, a lot of effort and time was invested in calculating the value of individual components used in the work. For the realization of the project, a stable supply voltage of the amplifier was provided, an RC filter was adjusted, a suitable speaker and a potentiometer were selected for volume control.

**Key words:** Doppler's effect, Rc filter, C++

## 1. UVOD

U ovom radu ćemo analizirati rad HLK-LD1115H FMCW senzora pokreta i ljudske prisutnosti koji je izveden pomoću Arduino Uno programabilne pločice, RC filtra, pojačala i jednog zvučnika. HLK-LD1115H je modul visoke osjetljivosti koji radi na frekvenciji od 24GHz. Koristi za otkrivanje ljudske prisutnosti ili pokreta. Ovaj modul koristi Dopplerov efekt kako bi otkrio promjene u frekvenciji odbijenog mikrovalnog signala. Kada se objekt, kao što je osoba kreće u području pokrivanja modula, dolazi do promjene u frekvenciji reflektiranog signala. Modul osjeti tu promjenu i detektira prisutnost pokreta. Često se koristi u sigurnosnim sustavima, automatizaciji zgrada, rasvjetnim sustavima, alarmnim uređajima i sl. Zahvaljujući visokoj osjetljivosti i sposobnosti rada na 24GHz, modul pruža precizno i pouzdano otkrivanje prisutnosti. Modul ima pet pinova od kojih se četiri povezuju sa Arduino Uno pločicom. Potrebno je osigurati odgovarajuće napajanje, povezati izlazni signal modula s odgovarajućim ulaznim pinovima na Arduino. Nadalje, na izlazni PWM pin bit će potrebno spojiti RC filter za pretvorbu signala i na izlazu filtera dodati pojačalo za pojačanje signala i zvučnik za reprodukciju signala. Slijedeći korak je, napisati program koji čita ulazni signal iz modula i izvršava željene radnje kada se detektira pokret. Važno je napomenuti da modul podržava serijsku komunikaciju.

## 2. OPIS PROJEKTOG ZADATKA

Cilj je projektirati uređaj koji će detektirati ljudski pokret na određenoj udaljenosti i u ovisnosti u tome kreće li se osoba, ili nepomično stoji pokrenuti određenu melodiju te istu zaustaviti ako nije detektiran pokret. Za mikrokontroler koristit će se Arduino Uno platforma na koju će biti spojen HLK-LD1115H senzor pokreta koji radi na principu Dopplerovog efekta. Nakon detektiranja pokreta digitalni signal šalje se na PWM izlaz Arduina i koristeći jednostavni RC filter pretvara se u analogni. Nakon pretvorbe signal se šalje na pojačalo D klase. Na izlazu pojačala priključen je zvučnik za reprodukciju zvuka. Prvo treba osigurati da su svi pinovi senzora ispravno spojeni na Arduino pločicu i to tako da je jedan pin spojen na napajanje od 5V, jedan pin je spojen na uzemljenje GND, zatim su RX i TX pinovi spojeni na TX i RX pinove Arduina. TX pin prenosi podatke prema mikrokontroleru, dok RX prima podatke od mikrokontrolera. V0 izlazni pin u ovom projektu neće se koristiti. Opcionalno se spaja na analogne ulaze mikrokontrolera i podržava analogni prijenos podataka. To znači da ovaj pin daje napon koji je proporcionalan intezitetu detekcije pokreta. Vrijednost koja se čita sa ovog pina u rasponu je od 0-1023 (ako se koristi 10-bitni ADC na Arduinu).

Za prijenos podataka koristi se serijska komunikacija koja se između Arduina i ovog senzora postiže putem UART komunikacije, koja koristi TX (Transmit) i RX (Receive) pinove. HLK senzor ima mogućnost izlaznog formata podataka u obliku ASCII znakova preko serijskog porta. Kako bi ovo ispravno funkcioniralo trebat će pravilno konfigurirati serijsku komunikaciju na Arduino ploči da bi se čitali podaci koje šalje HLK senzor. TX pin senzora se spaja na RX pin Arduina, a RX senzora na TX Arduina da bi se omogućila dvosmjerna serijska komunikacija. Na PWM izlazu Arduina nalazi se RC filter koji je izveden pomoću tri serijski spojena otpornika koja ukupno daju otpor od  $1.5k\Omega$  i njima paralelno spojenog folijskog blok kondenzatora od  $0,1\mu F$ . Nakon što je digitalni signal pretvoren u zvuk, odnosno u analogni signal potrebno ga je pojačati. Pojačanje signala bit će izvedeno pomoću pojačala klase D PAM8610. Pojačalo se napaja izvorom od 12V i pruža pojačanje od 10W po kanalu uz opterećenje od 8 Ohma. U ovom projektu koristit se samo jedan kanal pojačala. Na izlazu pojačala nalazi se otpornik od  $4,7\Omega$ , snage 5W zbog korištenja neadekvatnog zvučnika od  $4\Omega$ . Kako nije izvršeno prilagođenje



potrebno je koristiti otpornik kao alternativu da bi se izbjegle smetnje i šumovi. Također, bit će potrebno osigurati stabilan izvor napajanja sa ugrađenim filterom i minimalno jednaku ili veću struju od one predviđene za rad ovog pojačala za napajanje istog. Nadalje, prije ulaza u pojačalo potrebno je koristiti koaksijalni audio kabel koji ima izolaciju da bi se spriječile dodatne smetnje signala. Da bi u potpunosti bili sigurni da neće doći do zagrijavanja komponenti, smetnji ili šumova bit će potrebno osigurati i dobro i stabilno uzemljenje. Zvuk će se reproducirati preko zvučnika od 4 Ohma. Software koji će se koristiti za pisanje i izvršavanje koda je Arduino IDE. Treba napisati odgovarajući kod koji će detektirati pokret i u ovisnosti o pokretima osobe pokrenuti određenu melodiju. Za postavljanje serijske komunikacije koristimo funkciju „Serial.begin()“ i „Serial.read()“ za čitanje podataka sa TX i RX pinova. Nadalje, prema zahtjevu, u kodu će biti dodane i dodatno pojašnjene funkcije koje su odgovorne za čitanje podataka i pojedinih registara senzora. Na serial monitoru istovremeno će se pozivati i ispisivati vrijednosti pojedinih registara.

### **3. ARHITEKTURA PROJEKTNIH KOMPONENTI**

Za ovaj projekt koristit će se mikrokontroler Arduino Uno, HLK-LD1115H senzor prisutnosti i pokreta, jedan niskopropusni filter, PAM8610 pojačalo D klase, piezo zvučnik za reprodukciju zvuka, baterija od 12V kao izvor napajanja, jedan elektrolitski kondenzator, te jedan njemu paralelno spojen keramički kondenzator za filtraciju napona. Predstavljena je arhitektura te princip rada pojedinih komponenti te njihova uloga u realizaciji ovog završnog rada. Za akumulator od 12V te dva navedena kondenzatora nisu potrebna dodatna pojašnjenja jer se radi o osnovnim elektroničkim pasivnim komponentama, te DC izvoru napajanja.

#### **3.1. Arduino Uno**

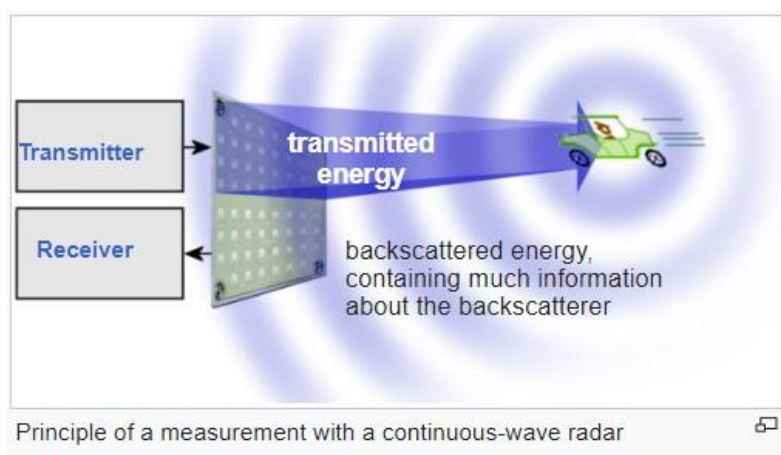
Arduino Uno je popularna mikrokontrolerska ploča koja se koristi za izradu različitih elektroničkih projekata. Koristi niskonaponski CMOS Atmega328P mikrokontroler visoke izvedbe temeljen na RISC arhitekturi. Arduino Uno ima ukupno 14 digitalnih ulazno/izlaznih pinova i 6 analognih ulaznih pinova. Ovi pinovi omogućuju spajanje različitih senzora, motora i drugih komponenti. Arduino ima ukupno 32 KB flash memorije za pohranu programa. Radni napon Arduina je 5V, što ga čini kompaktabilnim sa širokim rasponom elektroničkih komponenata. Ima ugrađeno USB sučelje koje omogućuje programiranje i komunikaciju s računalom. Arduino je open-source platforma što znači da je hardver i softver otvorenog koda. To omogućava inženjerima da razvijaju vlastite projekte i dijele ih s drugima. Arduino Uno platforma podržava različite dodatne module koji se mogu priključiti na ploču kako bi se proširile njene mogućnosti, kao što su Ethernet, bežične komunikacije i drugo. Arduino Uno je odličan izbor za početnike i napredne korisnike koji žele eksperimentirati s elektronikom i programiranjem. Na slici 3.1. prikazan je Arduino Uno.



*Slika 3.1. Arduino Uno*

### 3.2. RADARSKI MODUL HLK-LD1115H

HLK-LD1115H je radarski modul visoke osjetljivosti koji radi na frekvenciji od 24GHz i služi otkrivanju ljudske prisutnosti. Prikazan je na slici 3.3. Glavna značajka ovog modula je detekcija ljudskog tijela prikupljanjem sitnih pokreta, a temelji se na Dopplerovu efektu. Bazira se na FMCW (frequency modulated continuous wave radar). Mikrovalovi se šalju do objekta i odbijaju se natrag do senzora, pri čemu senzor detektira promjenu frekvencije primljenog vala. HLK senzor ima pet pinova. Jedan pin za napajanje (3.3-5V), jedan pin za uzemljenje, dva pina za odašiljanje i primanje valova (RX i TX), te analogni pin koji nosi informacije o detektiranom pokretu V0. Može detektirati pokret od 0,01m pa do 16m udaljenosti. Osjetljivost detekcije prisutnosti i detekcije pokreta mogu se regulirati po potrebi i željama putem serijske komunikacije. Senzor koristi UART protokol za serijsku komunikaciju. Ovaj protokol koristi asinhroni prijenos podataka, gdje se podaci šalju kao niz bitova, s prvim i završnim bitom za sinhronizaciju. Tipična brzina komunikacije je 115200 bps, a podaci se šalju u formatu sa 8-bitnim bajtovima, sa jednim startnim i jednim stop bitom. Korištenje serijske komunikacije omogućava preciznije očitavanje i konfiguraciju senzora. Serijska komunikacija prenosi se putem serijskog porta. To je hardversko ili softversko sučelje koje omogućava međusobni prijenos podataka između dva uređaja, jedan po jedan bit preko jedne ili više linija. Na slici 3.2. prikazan je princip rada CW radara.



Slika 3.2. Princip rada CW radara

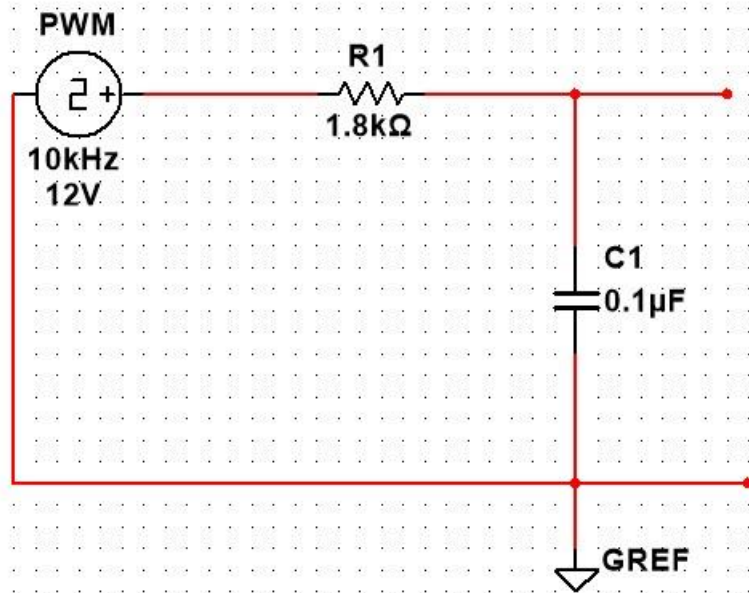


*Slika 3.3. Radarski modul HLK-LD1115H*

### 3.3. Niskopropusni (RC) filter

Zvuk je analogni signal, sukladno tome javlja se potreba za korištenjem niskopropusnog filtra da bi mogli simulirati analogni signal koristeći izlazni digitalni signal iz Arduina. Signal moduliran širinom impulsa može se prilično dobro izgladiti u stabilan napon. Dakle, digitalno-analognu pretvorbu moguće je postići korištenjem hardvera za mijenjanje radnog ciklusa PWM (engl. Pulse Width Modulation) signala (željeni DAC napon =  $A \cdot \text{radni ciklus}$ , gdje  $A$  predstavlja logički visoki napon, odnosno amplitudu). Korišteni RC krug sastoji se od otpornika od  $1.8 \text{ k}\Omega$  i jednog njemu paralelno spojenog kondenzatora od  $0,1 \text{ }\mu\text{F}$ . Otpornik se postavlja ispred kondenzatora kako bi se kontrolirao protok struje kroz kondenzator. Odabir pravilnih vrijednosti otpornika i kondenzatora u RC filtru ovisi o željenoj graničnoj frekvenciji filtra, koja određuje koje će frekvencije signala proći kroz filter, a koje će biti prigušene. Najčešće, pa i u ovom slučaju koristit će se niskopropusni filter koji omogućuje prolazak niskih frekvencija, točnije svih frekvencija do frekvencije rezanja, dok prigušuje visoke frekvencije. Ovakav niskopropusni tip filtra koristan je kod PWM signala jer uklanja visokofrekventne komponente, odnosno šumove PWM signala, ostavljajući samo željeni analogni signal. Frekvencija rezanja određuje na kojoj se frekvenciji signal počinje prigušivati. Tipično, frekvencija rezanja uglavnom se bira tako da je ispod frekvencije PWM signala, kako bi se učinkovito filtrirao šum, ali dovoljno visoko da ne prigušuje željeni audio signal. Ako znamo da je frekvencija PWM signala na Arduinu konstantna i iznosi oko  $490 \text{ Hz}$  ili  $980 \text{ Hz}$ , ovisno o korištenom pinu, onda je jednostavno odabrati ispravne vrijednosti korištenih komponenti uz danu relaciju ispod teksta gdje  $f_c$  označava frekvenciju rezanja. Za audio primjenu, bitno je da se audio signal ne priguši ( $20 \text{ Hz} - 20 \text{ kHz}$ ) i da se sačuva kvaliteta zvuka. Korišteni PWM izlaz Arduina u ovom slučaju bit će na pinu 6 koji generira frekvenciju od  $980 \text{ Hz}$ . Na slici 3.4. prikazan je RC filter koji će se koristiti u radu.

$$f_c = \frac{1}{2\pi RC}$$

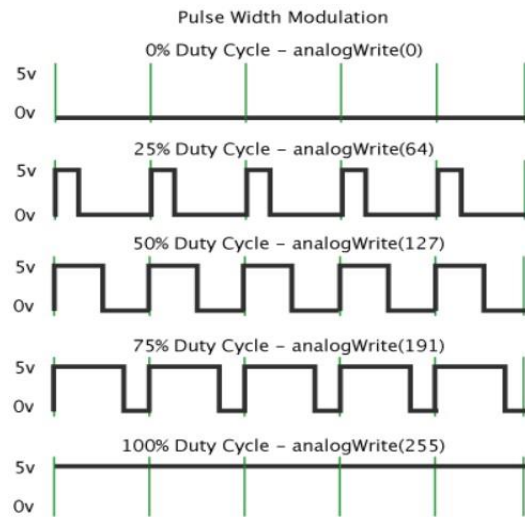


*Slika 3.4. Korišteni RC filter*

### 3.4. Pulsno-širinska modulacija (PWM)

PWM signal je digitalni signal koji se sastoji od periodičkih impulsa visokog i niskog stanja. Izgled PWM signala prikazan je na slici 3.5. Promjenom omjera trajanja visokog i niskog stanja PWM može simulirati analogni signal s različitim razinama napona ili struje. PWM signal koji je generiran sa Arduina određuje vrijednost napona na izlazu. Arduino nam nudi mogućnost upravljanja jačinom ili brzinom priključenih trošila. Na Arduino pločici pored nekih od digitalnih izvoda, odnosno pokraj njihovih oznaka postoji i znak ~ (tilda) što znači da se na njima može generirati PWM signal. Pulsno-širinska modulacija je tehnika dobijanja analognih vrijednosti pomoću digitalnih impulsa konstantne amplitude. Iako znamo da digitalni izlazi mogu biti samo u visokom (1) ili niskom (0) stanju pomoću navedenih PWM izlaza možemo simulirati analogni izlaz. Mjerenjem napona na tim izvodima možemo dobiti vrijednosti između 0V i 5V. PWM modulacija mijenja širinu impulsa signala. U generiranom signalu amplituda i frekvencija su konstantne, a mijenja se period signala. PWM signal koji se šalje na RC filter sadrži informaciju o periodu signala. Period signala predstavlja omjer vremena tijekom kojeg je PWM signal u visokom stanju u odnosu na ukupan period signala. Promjenom perioda signala mijenja se prosječna snaga signala, a prema tome i brzina ventilatora. U praksi to bi značilo da kontrolom trajanja visokog i niskog stanja PWM signala postignemo različite razine napona koji se primjenjuje na jačinu zvuka. Vrijednosti PWM signala kreću se od 0 do 255. Funkcija koja je odgovorna za rad PWM pinova zove se „analogWrite“ i koristimo ju u petlji loop dajući vrijednost postotka uključenosti pina kao drugi argument. U ovom radu nije korištena ta, već samo funkcija „tone“ koja je sama po sebi dovoljna iz razloga što je po zadanim postavkama Arduina period signala PWM signala podešen na 50%, što znači da je signal pola vremena u visokom stanju (5V), a pola vremena u niskom (0V). To znači da je ukupan izlazni napon 2.5V što je za ovaj rad dovoljno. Promjene između visokog i niskog napona RC filter pretvara u analogni signal s određenom amplitudom i frekvencijom što se percipira kao zvuk. Povećanje perioda signala utjecalo bi isključivo na jačinu zvuka, ali bi se i dodatno opteretilo pojačalo. Period signala PWM-a u ovom slučaju ne smije biti 100% (5V) jer u tom slučaju Arduino ne modulira signal već daje konstantan DC napon na izlazu. Nadalje, u tom slučaju RC filter propuštao bi konstantan DC signal jer nema PWM za filtriranje što pojačalo može interpretirati kao DC „offset“ i izazvati distorziju signala. Na slici 3.5. prikazana su stanja prema različitim vrijednostima perioda signala.





*Slika 3.5. PWM modulacija*

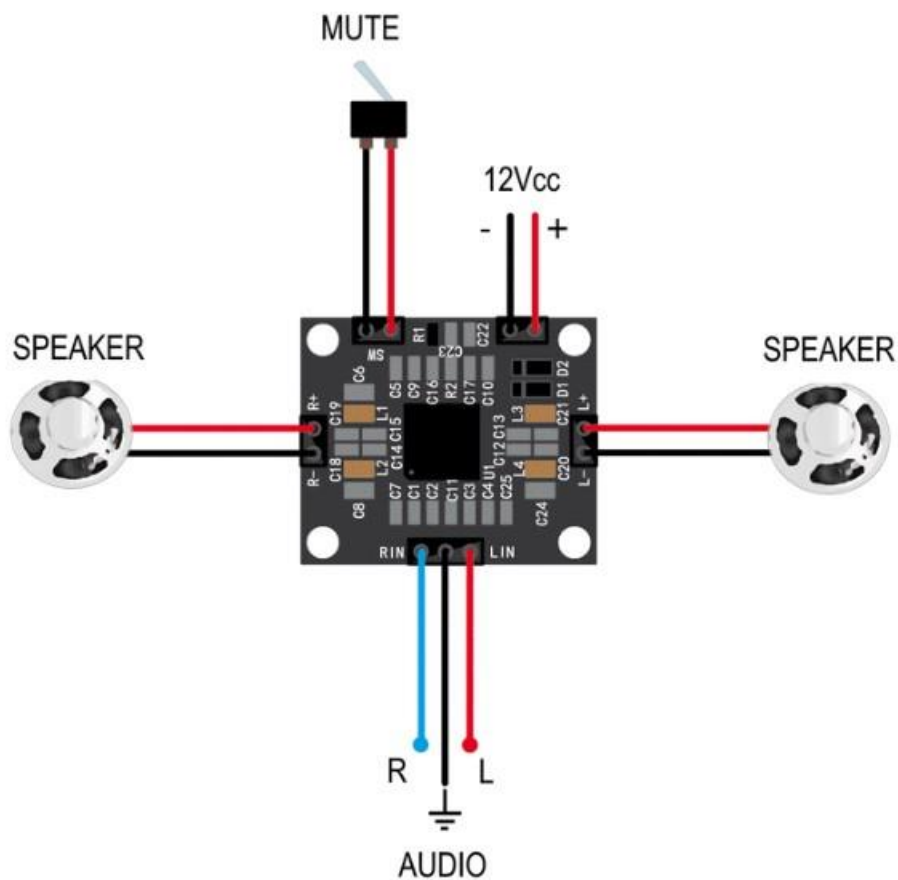
### 3.5. POJAČALO PAM8610

PAM8610 je stereo audio pojačalo razreda D, maksimalne snage do 2x10W po kanalu pri napajanju od preporučenih 12V i 2A s DC kontrolom glasnoće koje pruža nisku THD+N (Total harmonic distortion) (0,1%), nisku EMI i dobar PSRR, čime osigurava reprodukciju zvuka visoke kvalitete. Kontrola glasnoće uključuje 32 koraka i ima raspon od +32dB do -75dB. PAM8610 radi na napajanju od 7V do 15V s puno većom učinkovitošću u usporedbi s IC-ima konkurenata. Budući da je D-klase, učinkovitost mu je iznad 90%. PAM8610 zahtijeva vrlo malo vanjskih komponenti, što značajno štedi troškove i prostor na ploči. Također, dva izlazna kanala omogućavaju reprodukciju stereo zvuka. PAM8610 ima ugrađene zaštitne funkcije kao što je zaštita od KS, pregrijavanja i prenaponska zaštita. D-klase pojačala rade na visokim frekvencijama, obično između 250 kHz i 1 MHz što omogućava filtriranje visoke frekvencije kako bi se proizveo čisti audio signal. U ovom radu koristit će se pojačalo klase D jer je kompaktabilno sa Arduino PWM izlazom. Naime, D-klase pojačala koriste PWM signal za modulaciju izlaznog napona, stoga pravilno filtriranje može pomoći u minimiziranju visokofrekventnog šuma. Ovaj model temelji se na konverziji analognog zvuka u PWM impulse čija širina varira u skladu s amplitudom signala, gdje se zatim signal pojačava pomoću tranzistora (gotovo uvijek MOSFET) koje rade kao sklopke i imaju vrlo visoku učinkovitost. Napajanje je stabilno i čisto, što je izuzetno bitno da bi se izbjegle distorzije i šumovi u zvuku. Što se tiče povezivanja zvučnika, pojačala D-klase su osjetljiva na uparivanje sa zvučnicima, stoga je važno osigurati prilagođenje impedancije.

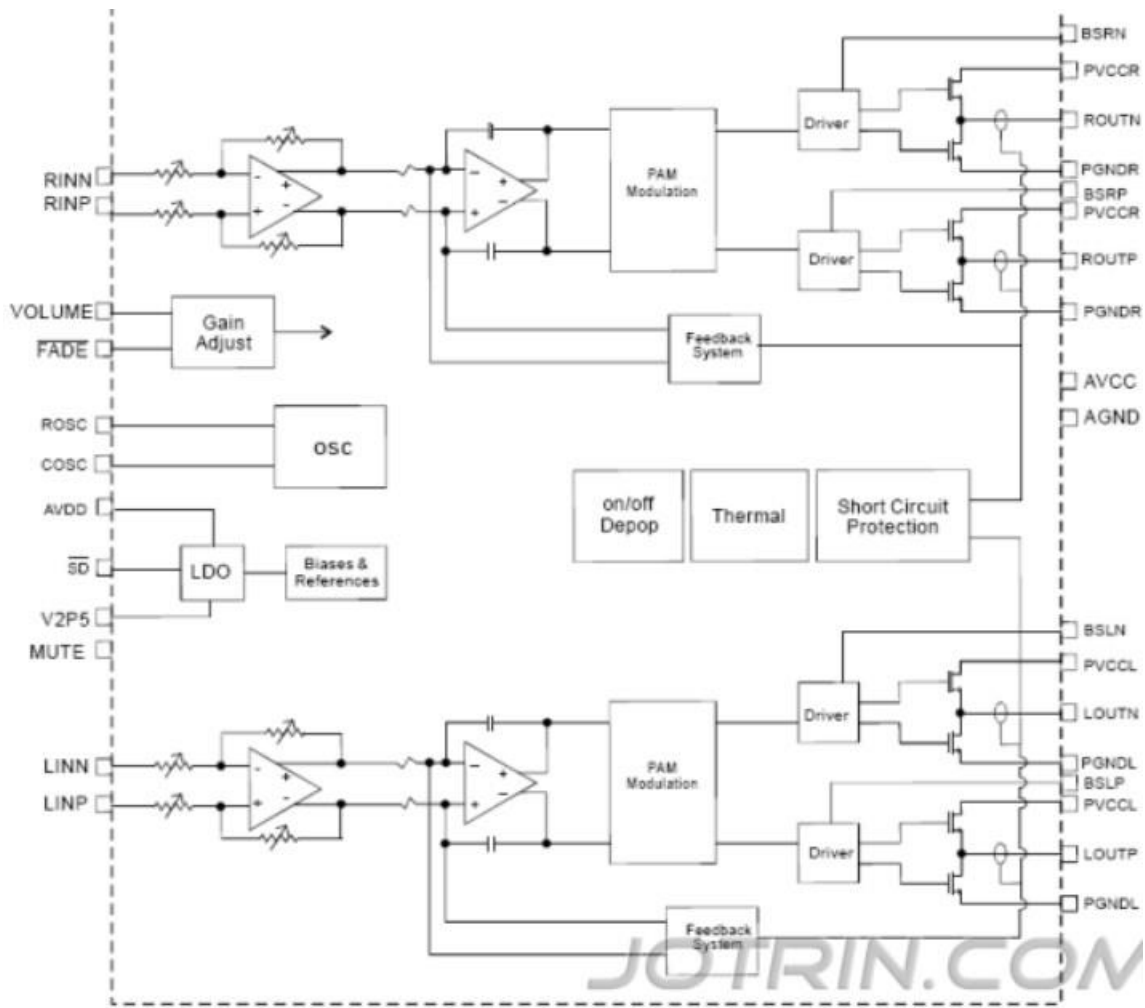
Što se tiče same električne sheme ovog pojačala, ima dvostruki LC filter koji je smješten na izlazu mosta i koristi se u pojačalima D-klase za linearizaciju izlaznog signala koji se sastoji od niza pravokutnih impulsa i stoga je isprekidan. Umetanjem LC filtera i pravilnim podešavanjem frekvencije, signal se linearizira, ali problem je što tada dolazi do zamjene faze, a da bi se to spriječilo filter se postavlja izvan propusnog pojasa. Integrirano pojačalo sadrži stereo pretpojačalo i dvostruki PWM modulator. Svaki odjeljak sastoji se od komparatora. Svaki komparator ima diferencijalni izlaz koji pokreće MOSFET koji je konfiguriran kao most. Ulazno pretpojačalo osigurava kontrolu pojačanja putem digitalnog bloka za podešavanje glasnoće u 32 koraka, dopuštajući pojačanje između -75dB (virtualno utišani izlazi) i 32dB (maksimalna glasnoća) .

Shematski prikaz PAM8610 prikazan je na slici 3.7. U ovom slučaju, digitalna kontrola se ne koristi, a stereo pojačalo radi na fiksnom pojačanju, određenom potencijalom koji AVDD

izlaz (pin 3) unutarnjeg LDO regulatora primjenjuje na 5 (VOL) i na VREF (pin 4 ). Preciznije, pojačalo, budući da je AVDD oko 3V, radi s pojačanjem od 32dB, stoga mora osigurati maksimalnu snagu na 8Ω i RMS napon od oko 8,8 V i uzimajući u obzir da se pojačalo sastoji od dva mosta potrebno je osigurati polovicu tog napona. Pojačanje od 32dB znači da se maksimalna snaga postiže s oko s oko 200mV na svakom ulazu. S obzirom na malu snagu, integrirani krug ne zahtijeva nikakav hladnjak. Čak i ako se pregrrije, PAM8610 ima unutarnju toplinsku zaštitu koja sprječava prekoračenje opasnih temperatura. Izgled ovog pojačala i način spajanja prikazan je na slici 3.6.



Slika 3.6. PAM8610 pojačalo



Slika 3.7. Blok shema PAM8610

### 3.6. Piezo zvučnik

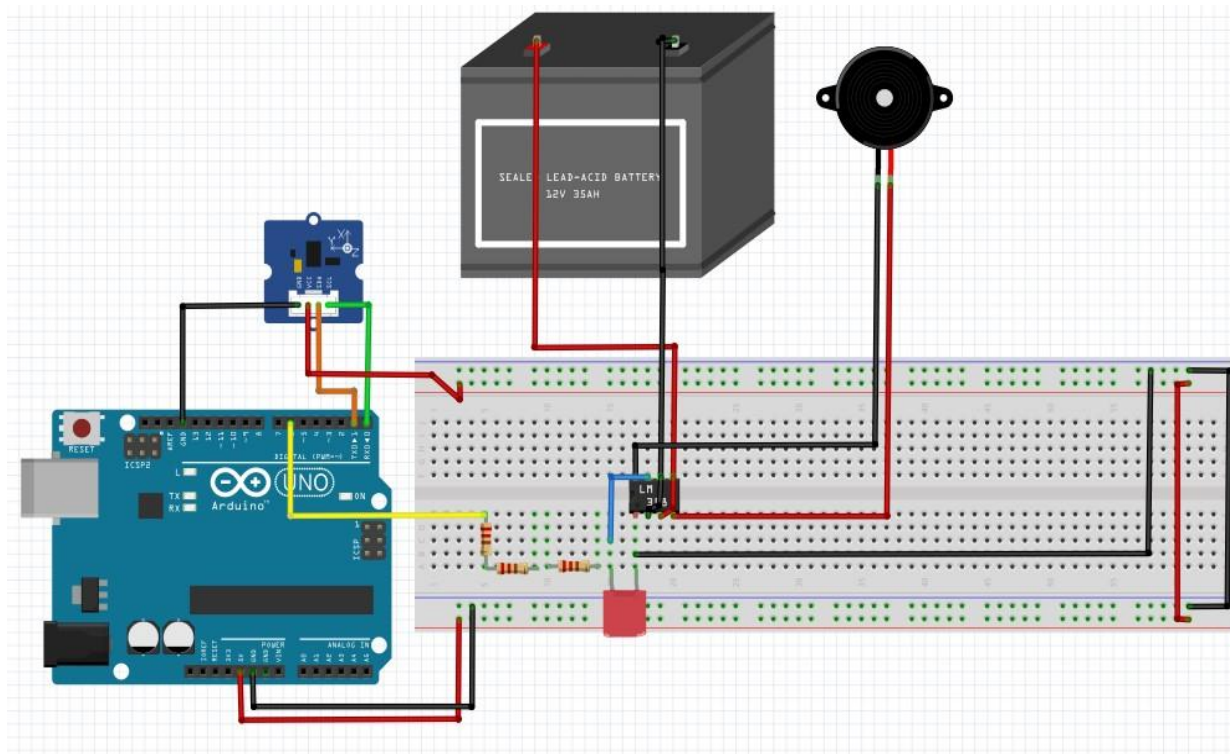
Piezo zvučnik, poznat i kao piezoelektrični zvučnik ili zvučni signalizator, vrsta je audio pretvarača koji generira zvučne valove koristeći piezoelektrični učinak. Piezoelektrični učinak je pojava u kojoj određeni materijali proizvode električni napon kao odgovor na mehanički stres ili pritisak, te obratno, generiraju mehanički stres ili pomak kao odgovor na primijenjeno električno polje. Osnovna komponenta piezo zvučnika je piezoelektrični element, obično napravljen od materijala poput kvarca ili keramike. Kada se na piezoelektrični kristal primijeni električni napon, on se mehanički deformira, te proizvodi zvuk. Povezana s piezoelektričnim elementom, postoji tanka dijafragma ili membrana koja vibrira kao odgovor na mehaničko pomicanje generirano piezoelektričnim svojstvom materijala. Piezo zvučnici sposobni su proizvoditi zvuk visoke frekvencije i često se koriste u aplikacijama gdje je potreban kompaktan i učinkovit zvučni izlaz, poput alarma i elektroničkih uređaja. U ovom radu koristi se zvučnik od 4 Ohma koji je prikazan na slici 3.8.



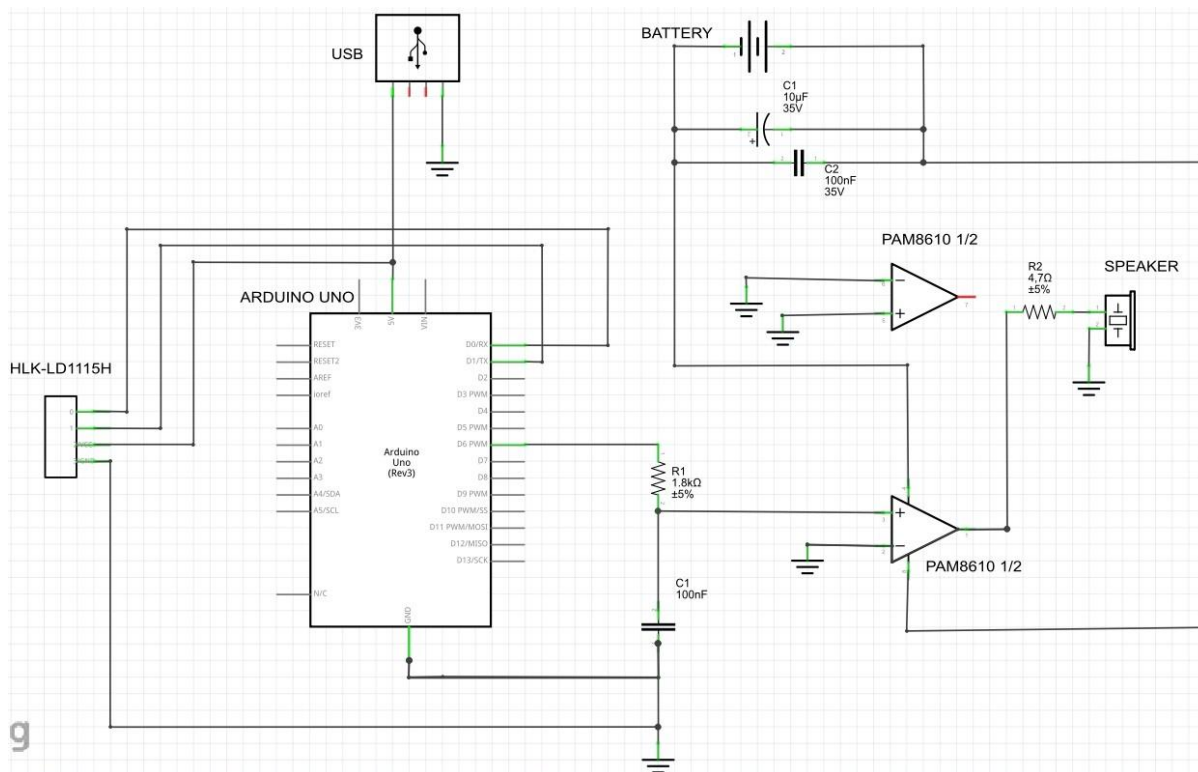
*Slika 3.8. Piezo zvučnik*

## 4. SHEMA SPAJANJA

Shema spajanja prikazana je na slici 4.1. Prilikom postavljanja senzora poželjno je izbjeći metalne i druge predmete koji blokiraju prijenos elektromagnetskih valova ispred modula. Ovisno o okruženju parametre je potrebno fino podesiti prema stvarnim uvjetima. Osjetljivost je podesiva prema korisničkim scenarijima. Senzor šalje podatke o pokretu na Arduino, na izlaz 6 Arduina spojen je RC filter u kojeg dolazi generirani PWM signal. Zatim se signal šalje na pojačalo i dalje na zvučnik kako bi se reproducirao zvuk. Pojačalo se napaja sa vanjskim izvorom od 12V koji mora biti stabilan. U ovom projektu koristit će se samo jedan kanal pojačala. Kako zvučnik ima drugačiji otpor od pojačala, prilagođenje će se izvršiti dodavanjem jednog otpornika od  $4,7\Omega$ , snage 5W. Ispod vizualnog prikaza sheme spajanja prikazana je i električna shema na slici 4.2.



Slika 4.1. Shema spajanja



Slika 4.2. Električna shema

## 5. UPUTE ZA KONFIGURACIJU SENZORA

Senzor se konfigurira pomoću parametara koji služe za postavljanje osjetljivosti detekcije prisutnosti, detekcije pokreta i vremenom čitanja promjene stanja na ulazu. Nakon namještanja tih parametara potrebno ih je spremirati da se isključenjem senzora ne izgube.

Parametar `th1=**`, služi za postavljanje osjetljivosti detekcije pokreta. Integer je cijeli broj. Zadana konfiguracija prema datasheetu je 120. Što je veća vrijednost to je modul manje osjetljiv.

Parametar `th2=**`, služi za postavljanje osjetljivosti detekcije prisutnosti. Integer je cijeli broj. Zadana konfiguracija je 250. Što je vrijednost veća, modul je manje osjetljiv na otkrivanje prisutnosti.

Parametar `save`, služi za spremanje namještenih parametara. Ako ne spremimo konfiguraciju, pokretanje će biti bezuspješno.

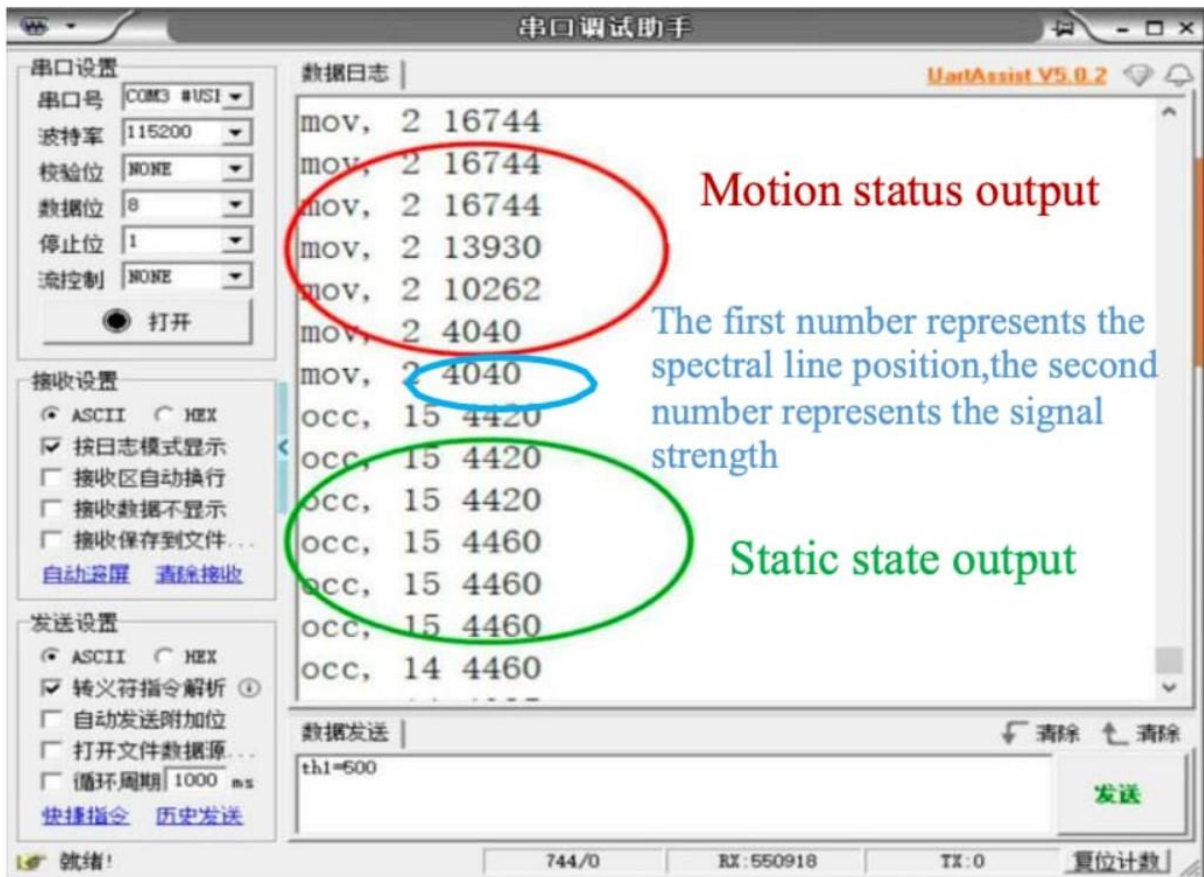
Parametar `get_all`, služi za ispisivanje trenutnih postavki namještenih parametara.

Parametar `dtime=**`, služi za postavljanje detekcije GPIO (general-purpose input/output) kašnjenja. Po zadanim postavkama to je 5s.

Parametar `mov_sn=**`, služi za detektiranje informacija visokog stanja, odnosno veći/jači pokret GPIO se postavlja u visoko stanje. Što je veća vrijednost, osjetljivost je manja. Koristi se kada želimo smanjiti osjetljivost senzora na pokrete, kako ne bi detektirao svaku manju prisutnost. Po zadanim postavkama je 3.

Parametar `occ_sn=**`, služi detektiranju informacija malih amplituda, odnosno malih pa i najmanjih pokreta ili same prisutnosti čovjeka u blizini senzora. Nije preporučljivo povećavati vrijednost jer će osjetljivost biti manja. Po zadanim postavkama je 5. Znači, kada se detektira veća aktivnost na serial monitoru će se ispisati „`mov, *****`“. Druga znamenka u ispisu označava snagu signala. Kada senzor detektira samu prisutnost čovjeka na serial monitoru ispisat će se „`occ, *****`“, gdje druga znamenka također označava snagu signala. Kada senzor ne može detektirati nikakvu prisutnost prestat će davati podatke na izlazu, odnosno na serial monitoru se neće ništa ispisati. Prva znamenka označava položaj spektralne linije (korisnik ju može ignorirati). Na slici 5.1. prikazan je ispis stanja HLK-LD1115H senzora.





Slika 5.1. Prikaz parametara mov\_sn i occ\_sn na Serial Monitoru

## 5.1. Raspored registara za svaki parametar

Prema datasheetu senzor ima četiri registra. Registar REG\_STATUS koji zauzima jedan bajt memorije, REG\_DISTANCE veličine 2 bajta, REG\_OCCUPANCY veličine jednog bajta i REG\_MOTION koji zauzima jedan bajt memorije. Raspored registara prikazan je na slici 5.2.

Registar	Opis	Veličina
REG_STATUS	Status senzora	1 bajt
REG_DISTANCE	Udaljenost do objekta	2 bajta
REG_OCCUPANCY	Prisutnost objekta	1 bajt
REG_MOTION	Pokret objekta	1 bajt

*Slika 5.2. Raspored registara HLK-LD1115H*

REG\_STATUS registar sadrži sljedeće podatke:

Bit 0: Ovaj bit označava da li je senzor uključen ili isključen. Vrijednost "1" označava da je senzor uključen, a vrijednost "0" označava da je senzor isključen.

Bit 1: Ovaj bit označava da li je senzor detektirao objekt. Vrijednost "1" označava da je senzor detektirao objekt, a vrijednost "0" označava da nije.

Bit 2: Ovaj bit označava da li je senzor detektirao pokret. Vrijednost "1" označava da je senzor detektirao pokret, a vrijednost "0" označava da nije.

REG\_DISTANCE registar sadrži udaljenost do objekta. Vrijednost je pohranjena u dvojnomo obliku, s 11 bitova za mantisu i 5 bitova za eksponent. 11 bitova za mantisu: Ovi bitovi predstavljaju mantisu broja udaljenosti. Mantisa je broj koji se množi s eksponentom kako bi se dobila stvarna udaljenost. 5 bitova za eksponent: Ovi bitovi predstavljaju eksponent broja udaljenosti. U stvarnosti u ovom registru pohranjeni su podaci za namještanje parametara osjetljivosti senzora. Senzor ima šest parametara koja se mogu regulirati, a to su th1\_motion,

th2\_occupancy, th3\_range, mov\_sn, occ\_sn i dtime. Što je vrijednost navedenih parametara veća, osjetljivost senzora na detekciju pokreta i prisutnosti je manja. Th1, th2 i th3 parametrima možemo smanjiti ili povećati doseg detekcije senzora, što je broj veći doseg mu je manji.

REG\_OCCUPANCY registar sadrži podatke o prisutnosti objekta. Vrijednost je pohranjena u 1 bajtu. 0: Objekt nije prisutan. 1: Objekt je prisutan.

REG\_MOTION registar sadrži podatke o pokretu objekta. Vrijednost je pohranjena u 1 bajtu. 0: Objekt se ne kreće. 1: Objekt se kreće.

Dakle, REG\_STATUS registar zauzima prvi bajt memorije. Bitovi 0, 1 i 2 su rezervirani za ovaj registar. Prikazan je na slici 5.3.

Bit	Vrijednost
0	Bit uključenja/isključenja
1	Bit detekcije objekta
2	Bit detekcije pokreta

Izvezi u Tablice

Slika 5.3. REG\_STATUS registar

REG\_DISTANCE registar zauzima sljedeća dva bajta memorije (2 i 3). Bitovi 0 do 11 prvog bajta su rezervirani za mantisu broja udaljenosti, a bitovi 0 do 4 drugog bajta su rezervirani za eksponent broja udaljenosti. Prikazan je na slici 5.4.

Bitovi	Vrijednost
0-11	Mantisa broja udaljenosti
0-4	Eksponent broja udaljenosti

Izvezi u Tablice

Slika 5.4. REG\_DISTANCE registar

REG\_OCCUPANCY registar zauzima sljedeći bajt memorije (4). Prikazan je na slici 5.5.

Bitovi	Vrijednost
0	Prisutnost objekta

Izvezi u Tablice

Slika 5.5. REG\_OCCUPANCY registar

REG\_MOTION registar zauzima sljedeći bajt memorije (5). Prikazan je na slici 5.6.

Bitovi	Vrijednost
0	Pokret objekta

Izvezi u Tablice

Slika 5.6. REG\_MOTION registar

## 6. KONFIGURACIJA KODA

Parametri za serijsku komunikaciju za ovaj senzor nevedeni su u njegovom data sheetu. Brzina prijenosa (baud rate) je 115200, 8 podatkovnih bita, 1 stop bit, bez provjere parnosti (check bit) i bez kontrole toka. Ovaj Arduino kod upravljat će povezanim senzorom i zvučnikom koji reproducira melodiju kada se otkrije pokret. Također, kod podešava pragove za detekciju pokreta, prisutnosti i raspona udaljenosti senzora. Na Serial monitoru se ispisuju „occ“ i „mov“ vrijednosti. Sada će se objasniti svaka linija koda koji je prikazan ispod.

```
#include <SoftwareSerial.h> uključuje biblioteku SoftwareSerial koja omogućava korištenje softverske serijske komunikacije na bilo kojim digitalnim pinovima.
```

```
#define naredbe definiraju frekvencije za sve note koje će se koristiti u melodiji.
```

Frekvencija svake note dana je u Hz.

```
melody[] je niz koji sadrži sekvencu nota koje čine melodiju „Für Elise“ od Beethovena.
```

```
noteDurations[] je niz koji definira trajanje svake note iz melody[]. Trajanje je specificirano u relativnim vrijednostima što znači da 8 predstavlja notu koja traje osminu vremena, a 4 četvrtinu definiranog vremena.
```

Parametri za podešavanje osjetljivosti `th1`, `th2`, `th3`, `mov_sn`, `occ_sn`, `dtime` definiraju pragove i parametre za detekciju pokreta, prisutnosti i raspona detekcije senzora. Također, postavljaju se i osjetljivosti za različite vrste detekcija i vrijeme kašnjenja između detekcija.

```
lastMovTime je pomjenjiva funkcija za praćenje vremena posljednje detekcije pokreta.  
melodyPlaying provjerava svira li melodija trenutno, ili ne.
```

U `setup()` funkciji `Serial.begin` započinje komunikaciju sa serijskim monitorom, `sensorSerial.begin` započinje komunikaciju sa senzorom te se nakon toga ispisuju početne vrijednosti postavki za osjetljivost koristeći `Serial.println()` funkciju.

U `loop()` funkciji se provjerava jesu li dostupni podaci sa senzora. Ako su podaci dostupni čitaju se do kraja linije `sensorSerial.readStringUntil` i ispisuju na serijski monitor. Ako primljeni podaci počinju sa „mov“, melodija se pokreće (ako već nije pokrenuta). Ako primljeni podaci ne počinju sa „mov“ melodija se zaustavlja (ako trenutno svira). Radi stabilnosti dodat će se malo kašnjenje `delay`.

U funkciji `playMelody()` se izvršava sviranje melodije i to na način da funkcija prolazi kroz sve note u `melody[]`. Za svaku notu određuje trajanje i koristi funkciju `tone()` za reprodukciju te note. Nakon svake note pravi se pauza. Ako se tokom sviranja pojavi novi signal koji ne počinje sa „mov“ melodija se prekida.

Funkcija koja ispisuje trenutne vrijednosti i stanja određenih registara senzora na serijski monitor je `Serial.println()`. Ova funkcija je korištena u `setup()` funkciji za ispisivanje početnih postavki senzora. Ova linija uzima vrijednost parametra (npr. `th1`), konvertira ga u string pomoću `nString()` funkcije, a zatim ga ispisuje na serijski monitor pomoću `Serial.println()`. Ovaj dio koda odgovoran je za ispis početnih vrijednosti parametara senzora na serijski monitor kada se program pokrene. `Serial.println()` se može koristiti u bilo kojem dijelu koda da bi se ispisale trenutne vrijednosti varijabli.

Na početku koda potrebno je definirati sve korištene note koje se sastoje od imena i frekvencije prikazane u hercima (Hz). Note su poredane prema frekvenciji, počevši od najnižih prema višim.

```
#include <SoftwareSerial.h>

// Definicija svih frekvencija za sve note
#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78
#define NOTE_E2  82
#define NOTE_F2  87
#define NOTE_FS2 93
#define NOTE_G2  98
#define NOTE_GS2 104
#define NOTE_A2  110
#define NOTE_AS2 117
#define NOTE_B2  123
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
```

```
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
```



```

#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978

// Deklaracija melodije i trajanja nota (Für Elise - Beethoven)
int melody[] = {
NOTE_E5, NOTE_DS5, NOTE_E5, NOTE_DS5, NOTE_E5, NOTE_B4, NOTE_D5,
NOTE_C5,
NOTE_A4, NOTE_C4, NOTE_E4, NOTE_A4, NOTE_B4, NOTE_E4, NOTE_GS4,
NOTE_B4,
NOTE_C5, NOTE_E4, NOTE_E5, NOTE_DS5, NOTE_E5, NOTE_DS5, NOTE_E5,
NOTE_B4,
NOTE_D5, NOTE_C5, NOTE_A4
};

int noteDurations[] = {
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 4

```

```

};
// Parametri za podešavanje osjetljivosti
float th1_motion = 1200; // Prag za detekciju pokreta
float th2_occupancy = 2500; // Prag za detekciju prisutnosti
float th3_range = 50000; // Prag za raspon detekcije
int mov_sn = 4; // Osjetljivost za detekciju pokreta
int occ_sn = 120; // Osjetljivost za detekciju prisutnosti
int dtime = 5; // Vrijeme kašnjenja detekcije

// Definicija pinova i varijabli
SoftwareSerial sensorSerial(0, 1); // RX i TX pinovi
const unsigned long BAUD_RATE = 115200;
const int speaker = 6; // Pin za zvučnik

// Varijable za praćenje vremena
unsigned long lastMovTime = 0;
bool melodyPlaying = false;

void setup() {
  Serial.begin(BAUD_RATE); // Inicijalizacija serijskog monitora
  sensorSerial.begin(BAUD_RATE); // Inicijalizacija serijskog porta za
  senzor

  // Ispis parametara na serijski monitor
  Serial.println("Pokretanje programa...");
  Serial.println("Postavke osjetljivosti:");
  Serial.println("th1_motion: " + String(th1_motion));
  Serial.println("th2_occupancy: " + String(th2_occupancy));
  Serial.println("th3_range: " + String(th3_range));
  Serial.println("mov_sn: " + String(mov_sn));
  Serial.println("occ_sn: " + String(occ_sn));
  Serial.println("dtime: " + String(dtime));
}

void loop() {
  if (sensorSerial.available()) {
    String input = sensorSerial.readStringUntil('\n'); // Pročitaj
    podatke do novog reda
  }
}

```

```

Serial.println("Primljeni podaci: " + input);

// Ako primimo "mov" signal
if (input.startsWith("mov")) {
  if (!melodyPlaying) {
    playMelody(); // Pokreni melodiju
    melodyPlaying = true;
  }
}
// Ako primimo "occ" signal ili bilo koji drugi signal
else {
  if (melodyPlaying) {
    noTone(speaker); // Zaustavi melodiju
    melodyPlaying = false;
  }
}
delay(10); // Malo kašnjenje za stabilnost
}

void playMelody() {
  for (int thisNote = 0; thisNote < sizeof(melody) /
    sizeof(melody[0]); thisNote++) {
    int noteDuration = 1500 / noteDurations[thisNote];
    tone(speaker, melody[thisNote], noteDuration);

    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);

    // Ako se pojavi drugi signal, prekini melodiju
    if (sensorSerial.available() &&
      !sensorSerial.readStringUntil('\n').startsWith("mov")) {
      noTone(speaker);
      break;
    }
  }
}
}

```

## 7. ZAKLJUČAK

U ovom projektu realiziran je pouzdan i osjetljiv uređaj za dinamičku detekciju pokreta i ljudske prisutnosti sa promjenjivim zvučnim odgovorom koristeći Arduino Uno pločicu kao mikrokontroler, HLK-LD1115H senzor, jednostavan RC filter, integrirano pojačalo klase D, te zvučnik za reprodukciju zvuka uz dodatno napajanje za pojačalo. Glavi cilj projekta je ispunjen, senzor uspješno detektira prisutnost pokreta ili osobe i daje daljnje naredbe mikrokontroleru koji na temelju podataka reproducira zvuk na zvučniku. Najveći izazov što se tiče softvera je prilagoditi kod vlastitim potrebama, tako da se na izlazu mogu čitati samo relevantni podaci prilagođeni trenutnom okruženju. HLK senzor na svom izlazu ispisuje dosta različitih parametara koji se mogu dalje koristiti za izradu još naprednijeg uređaja sa više mogućnosti. Vezano za hardverski dio, potrebno je poznavati izvedbu, svrhu i način funkcioniranja RC filtera te rada u cjelini. Nadalje, kod audio tehnike svaki parametar je bitan i može ovisiti o kvaliteti reproduciranog zvuka. Tako je bilo potrebno osigurati stabilno napajanje sa ispravljačem, sa strujom većom od struje potrebne za napajanje pojačala. Od velikog značaja je i uzemljenje koje mora biti adekvatno i stabilno. Važno je i korištenje koaksijalnog audio kabela, a sve to u svrhu čistog zvuka bez distorzije i šumova. Što se tiče senzora pokreta, maksimalna udaljenost pokreta koju može detektirati je 16 metara, a samu prisutnost čovjeka detektira na maksimalno 8 metara udaljenosti. Koačno, HLK-LD1115 je prilično jeftin i prema napravljenim testovima u dužem vremenskom periodu jako pouzdan i precizan senzor koji koristi Dopplerov efekt kako bi detektirao pokret. Ovaj projekt ima široku primjenu i mogućnosti.

## 8. LITERATURA

<https://community.home-assistant.io/t/how-to-work-with-hlk-ld1115h-and-wemos-d1-mini-for-human-presence-detection/434427>

[3.1.] <https://www.aliexpress.com/item/1005004630565330.html#nav-specification>

[3.2.] <https://community.home-assistant.io/t/how-to-work-with-hlk-ld1115h-and-wemos-d1-mini-for-human-presence-detection/434427/54>

[3.3.] <https://revspace.nl/FMCWRadar>

[3.4.] <https://forum.arduino.cc/t/how-to-use-pwm-with-low-pass-rc-filter-to-control-power-supply-voltage/649604>

[3.5.] <https://docs.arduino.cc/learn/microcontrollers/analog-output/>

[3.6.] <https://www.open-electronics.org/efficient-and-compact-stereo-power-amplifier-unleash-high-quality-audio-with-the-pam8610/>

[https://www.reddit.com/r/diysound/comments/10ovjec/pam8610\\_red\\_board\\_kit\\_schematics/?rdt=52998](https://www.reddit.com/r/diysound/comments/10ovjec/pam8610_red_board_kit_schematics/?rdt=52998)

## 9. POPIS SLIKA

1. Slika 3.1. Arduino Uno ..... **Pogreška! Knjižna oznaka nije definirana.**
2. Slika 3.2. Princip rada CW radara ..... **Pogreška! Knjižna oznaka nije definirana.**
3. Slika 3.3. HLK-LD1115H senzor ..... **Pogreška! Knjižna oznaka nije definirana.**
4. Slika 3.4. Korišteni RC filter ..... **Pogreška! Knjižna oznaka nije definirana.**
5. Slika 3.5. PWM modulacija ..... **Pogreška! Knjižna oznaka nije definirana.**
6. Slika 3.6. PAM8610 pojačalo ..... **Pogreška! Knjižna oznaka nije definirana.**
7. Slika 3.7. Blok shema PAM8610..... **Pogreška! Knjižna oznaka nije definirana.**
8. Slika 3.8. Piezo zvučnik..... **Pogreška! Knjižna oznaka nije definirana.**
9. Slika 4.1. Shema spajanja ..... **Pogreška! Knjižna oznaka nije definirana.**
10. Slika 4.2. Električna shema..... **Pogreška! Knjižna oznaka nije definirana.**
11. Slika 5.1. Opis izlaznih stanja na Serial Monitoru..... **Pogreška! Knjižna oznaka nije definirana.**
12. Slika 5.2. Raspored registara HLK-LD1115H..... **Pogreška! Knjižna oznaka nije definirana.**
13. Slika 5.3. REG\_STATUS registar ..... **Pogreška! Knjižna oznaka nije definirana.**
14. Slika 5.4. REG\_DISTANCE registar..... **Pogreška! Knjižna oznaka nije definirana.**
15. Slika 5.5. REG\_OCCUPANCY registar..... **Pogreška! Knjižna oznaka nije definirana.**
16. Slika 5.6. REG\_MOTION registar ..... **Pogreška! Knjižna oznaka nije definirana.**