

Sustav za obradu audio signala u stvarnom vremenu temeljen na ARM Cortex M4 arhitekturi procesora

Šafarik, Antonio

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:168:976146>

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

Download date / Datum preuzimanja: **2025-03-14**



Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 144

**SUSTAV ZA OBRADU AUDIO SIGNALA U STVARNOM
VREMENU TEMELJEN NA ARM CORTEX M4 ARHITEKTURI
PROCESORA**

Antonio Šafarik

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 144

**SUSTAV ZA OBRADU AUDIO SIGNALA U STVARNOM
VREMENU TEMELJEN NA ARM CORTEX M4 ARHITEKTURI
PROCESORA**

Antonio Šafarik

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 144

Pristupnik: **Antonio Šafarik (0036518532)**

Studij: Elektrotehnika i informacijska tehnologija

Profil: Elektroničko i računalno inženjerstvo

Mentor: prof. dr. sc. Davor Petrinović

Zadatak: **Sustav za obradu audio signala u stvarnom vremenu temeljen na ARM Cortex M4 arhitekturi procesora**

Opis zadatka:

U sklopu diplomskog rada potrebno je koncipirati, projektirati, realizirati i verificirati rad ugradbenog računalnog sustava koji može digitalno obrađivati stereo audio signale korištenjem 32-bitnog mikrokontrolera arhitekture ARM Cortex M4. Odabrati odgovarajući codec (integrirani A/D i D/A pretvornik za audio primjene), te ga povezati s mikrokontrolerom koji će provoditi obradu signala u stvarnom vremenu. Osmisliti način inicijalizacije ovog codec-a u svrhu odabira i konfiguriranja svih parametara otipkavanja i rekonstrukcije audio signala te konfiguriranja I2S serijske veze codeca s mikrokontrolerom. Demonstrirati rad razvijenog i realiziranog sustava na primjeru digitalne filtracije ulaznog signala pomoću FIR i IIR filtara korištenjem funkcija iz CMSIS DSP biblioteke, te obrađeni signal proslijediti na D/A pretvornik codeca uz reprodukciju na izlaznom audio uređaju. Sustav temeljiti na razvojnoj pločici STM Discovery i programskom razvojnom alatu Keil Microvision. Za detaljnije informacije obratiti se mentoru.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

1. Uvod	4
2. Ideja i ciljevi rada	5
3. Hardverska izvedba	6
3.1. STM32F4 Discovery	6
3.2. Audio codec	8
3.3. Saleae logic analyzer	9
4. Softverska izvedba	12
4.1. STM32CubeMX	12
4.2. Keil μ Vision5.....	16
4.3. Saleae logic analyzer	19
5. Komunikacijski protokoli	21
5.1. I2C komunikacijski protokol	21
5.2. I2S komunikacijski protokol	23
5.3. Razlika između I2C i I2S protokola	24
6. Vlastita izvedba	25
6.1. Hardvedski dio izvedbe	25
6.2. Softverski dio izvedbe	28
7. Sljedeći koraci	34
8. Zaključak	35
Literatura	36

1. Uvod

Digitalna obrada govora postala je ključna tehnologija u mnogim modernim aplikacijama, uključujući pametne asistente, prepoznavanje govora, komunikacijske sustave, kao i u uređajima za pomoć osobama s oštećenjem sluha. Kvalitetna digitalna obrada govora omogućava precizno prikupljanje, analizu i manipulaciju audio signala, što je ključno za razvoj sofisticiranih algoritama i aplikacija. U ovom kontekstu, odabir odgovarajućeg hardvera je od velike važnosti.

STM32 mikrokontroleri su često korišten primjer u sustavima digitalne obrade signala zbog svoje fleksibilnosti, snage obrade i niske potrošnje energije. Ovi mikrokontroleri nude mogućnosti obrade u stvarnom vremenu i širok spektar periferija, čineći ih idealnim za mnoge primjene u obradi zvuka, uključujući govorne signale.

Međutim, iako STM32 mikrokontroleri imaju ugrađene ADC (analogno-digitalne pretvarače), za visokokvalitetnu obradu zvuka često je preporučljivo koristiti vanjske audio kodeke. Razlog tome je što eksterni audio kodeci nude specijalizirane ADC i DAC (digitalno-analogne pretvarače) koji su optimizirani za zvučne frekvencije, pružajući bolju preciznost, smanjenu buku i podršku za više kanala. Korištenjem eksternih audio kodeka omogućava se obrada zvuka na višoj kvalitetnoj razini, što je posebno bitno kod osjetljivih aplikacija kao što su prepoznavanje govora i audio komunikacija.

2. Ideja i ciljevi rada

Cilj ovog rada bio je implementirati sustav koji koristi STM32F4 Discovery pločicu u kombinaciji s vanjskim audio kodekom na protoboardu kako bi se generirao, prenosio i reproducirao digitalni sinusni signal putem vanjskih slušalica. Ključna ideja ovog sustava je generiranje sinusnog signala u digitalnom obliku unutar STM32 mikrokontrolera i njegova isporuka prema vanjskom audio kodeku putem I2C (*Inter-Integrated Circuit*) komunikacijskog protokola.

STM32F4 mikrokontroler se koristi za stvaranje sinusnog vala na osnovu zadanih parametara, uključujući frekvenciju signala, koja je konfigurabilna i može se dinamički mijenjati od strane korisnika. Nakon što mikrokontroler generira sinusni signal, on ga šalje audio kodeku koji služi kao specijalizirani vanjski uređaj za digitalno-analognu pretvorbu (DAC). Ova pretvorba je ključna za prijenos zvučnih signala u analogni oblik koji ljudsko uho može čuti putem slušalica.

Audio kodek na protoboardu preuzima digitalni signal putem I2C protokola, obavlja potrebnu konverziju i šalje signal u analognom obliku prema izlaznim uređajima, kao što su slušalice. Prednost korištenja vanjskog audio kodeka leži u njegovoj sposobnosti da pruži visoku kvalitetu zvuka uz minimalne šumove i distorziju, što je od presudne važnosti za točnost signala u audio aplikacijama.

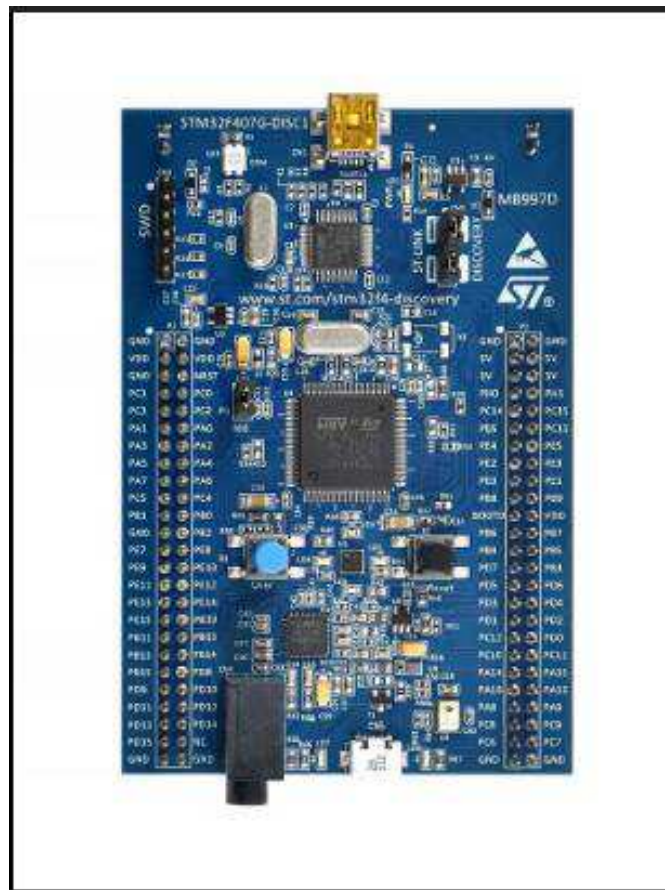
Fleksibilnost sustava očituje se u mogućnosti podešavanja frekvencije sinusnog signala, što korisniku omogućuje eksperimentiranje s različitim frekvencijama i analiziranje performansi uređaja pri različitim uvjetima rada. Ovo omogućava sustavu da bude koristan ne samo kao alat za testiranje, već i kao edukacijska platforma za razumijevanje digitalne obrade zvuka i primjenu mikrokontrolera u realnom vremenu.

Kombinacija STM32F4 mikrokontrolera, I2C protokola za prijenos podataka i vanjskog audio kodeka za digitalno-analognu pretvorbu omogućuje izgradnju visoko učinkovitog i fleksibilnog sustava za generiranje i reprodukciju zvučnih signala.

3. Hardverska izvedba

3.1. STM32F4 Discovery

STM32F4 Discovery je popularan izbor za projekte vezane uz digitalnu obradu signala (DSP), zahvaljujući svojoj moćnoj Cortex-M4 jezgri s ugrađenom FPU i podrškom za napredne algoritme obrade signala. Visoka brzina procesora, bogata memorija i podrška za razne periferije čine ga izvrsnim rješenjem za projekte kao što su audio aplikacije, prepoznavanje govora i aplikacije u stvarnom vremenu.



Slika 1: STM32F4 Discovery board

Ključne značajke STM32F4 Discovery pločice:

1. Mikrokontroler

STM32F4 *Discovery* pločica je opremljena STM32F407VGT6 mikrokontrolerom, koji sadrži 32-bitni ARM Cortex-M4 procesor s ugrađenom podrškom za DSP (*Digital Signal Processing*) i FPU (*Floating Point Unit*), uključujući audio aplikacije. Radni takt mikrokontrolera iznosi do 168 MHz, što pruža značajnu procesorsku snagu.

2. Memorija

Mikrokontroler dolazi s 1 MB Flash memorije i 192 KB RAM-a, što omogućava pohranu kompleksnih programa i podataka, uključujući audio signale i parametre za njihovu obradu.

3. Integrirani senzori i periferni uređaji

STM32F4 *Discovery* pločica dolazi s integriranim žiroskopom i mogućnošću proširenja sa širokim rasponom vanjskih perifernih uređaja, kao što su LCD zaslone, senzori i moduli za bežičnu komunikaciju. Također ima integrirane LED diode, korisnički gumb i ugrađeni USB OTG priključak.

4. Ulazi i izlazi (I/O)

Pločica nudi veliki broj I/O pinova koji podržavaju razne komunikacijske protokole poput UART, SPI, I2C, CAN, i GPIO za rad s vanjskim komponentama. Ovi pinovi omogućuju jednostavno povezivanje s vanjskim modulima kao što su senzori, aktuatori, ili, u slučaju ovog projekta, audio kodeci.

5. Komunikacijski protokoli

STM32F4 *Discovery* podržava razne komunikacijske protokole, uključujući I2C, SPI, UART, i CAN, što omogućuje fleksibilnu interakciju s različitim perifernim uređajima. U ovom projektu, I2C protokol se koristi za komunikaciju između mikrokontrolera i vanjskog audio kodeka.

6. Programiranje i *debugiranje*

STM32F4 *Discovery* pločica dolazi s ugrađenim ST-LINK/V2 programatorom i emulatorom koji omogućava jednostavno programiranje i *debugiranje* aplikacija putem USB veze s računalom. To olakšava razvoj i testiranje aplikacija na samoj pločici.

3.2. Audio codec

Korištenje eksternog audio kodeka u projektima digitalne obrade zvuka donosi brojne prednosti, uključujući bolju kvalitetu zvuka, veću fleksibilnost u dizajnu, smanjeno opterećenje mikrokontrolera i podršku za naprednije audio funkcije. Ove prednosti čine vanjske kodeke idealnim rješenjem za aplikacije koje zahtijevaju visokokvalitetnu reprodukciju zvuka, kao što su prepoznavanje govora, zvučni sustavi, komunikacijski uređaji i mnoge druge audio aplikacije.

1. Kvaliteta zvuka

Vanjski audio kodeci obično nude znatno višu kvalitetu konverzije audio signala. Specijalizirani su za rad s audio frekvencijama i optimizirani su za smanjenje šuma, distorzije i *jittera* (varijacija u vremenu uzorkovanja), što rezultira mnogo preciznijim i čistijim zvukom. U mikrokontrolerima, ugrađeni ADC/DAC često nisu dizajnirani s naglaskom na visoke performanse za audio obradu, što može rezultirati lošijom kvalitetom zvuka.

2. Širok dinamički raspon

Eksterni audio kodeci obično podržavaju širok dinamički raspon, što omogućava precizno hvatanje i reprodukciju signala u opsegu od tihih do vrlo glasnih zvukova, bez gubitka detalja ili pojave distorzije. U digitalnoj obradi govora ili glazbe, to je ključno za postizanje prirodnog zvuka.

3. Podrška za višekanalnu obradu

Eksterni audio kodeci često podržavaju višekanalni ulaz i izlaz, što je važno za složenije audio aplikacije kao što su stereo ili surround zvuk. Ovo omogućuje obradu i reprodukciju više audio kanala istovremeno, što može biti ograničeno na mikrokontrolerima s osnovnim ADC/DAC sustavima.

4. Veća frekvencija uzorkovanja

Kodeci dizajnirani za obradu zvuka često podržavaju visoke frekvencije uzorkovanja (44.1 kHz, 48 kHz, 96 kHz pa i više), što je nužno za kvalitetnu audio obradu, posebno u aplikacijama poput glazbene produkcije ili analize signala. Mikrokontroleri s integriranim

ADC-ima i DAC-ima obično rade s nižim frekvencijama uzorkovanja koje mogu biti ograničene za takve aplikacije.

5. Bolja izolacija od buke

Eksterni audio kodeci, budući da su odvojeni od glavnog mikrokontrolera, često nude bolju izolaciju od električne buke koja može nastati unutar mikrokontrolera, osobito pri radu s digitalnim signalima. Ova izolacija rezultira čistim audio signalima, bez smetnji koje mogu utjecati na kvalitetu zvuka.

6. Optimizirani za obradu audio signala

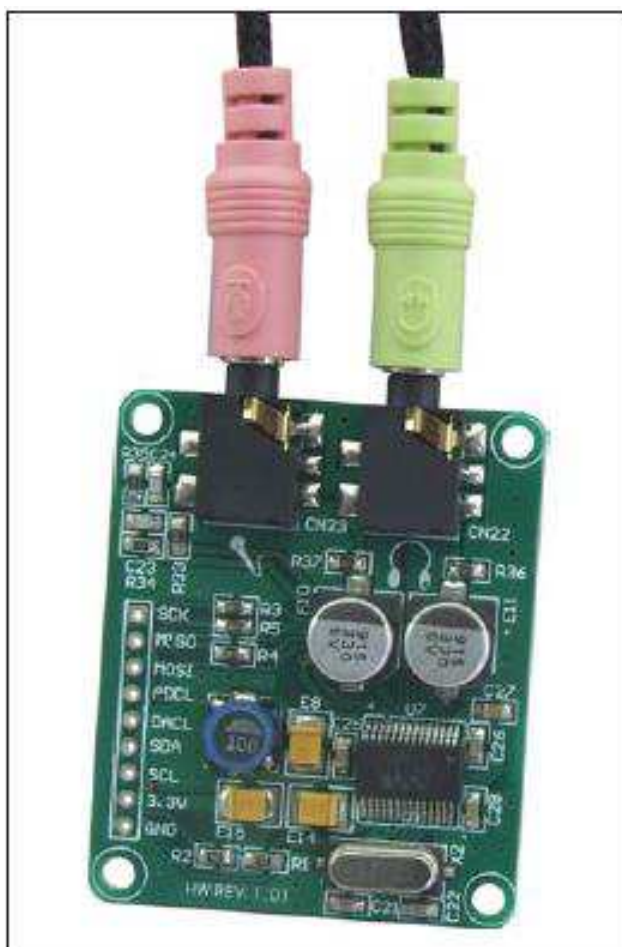
Specijalizirani audio kodeci uključuju optimizirane filtre i druge funkcionalnosti posebno prilagođene za obradu audio signala. Oni su dizajnirani za rukovanje različitim formatima digitalnih podataka (npr. PCM) i mogu prilagoditi kvalitetu audio signala prema potrebama aplikacije. Ovi kodeci često imaju integrirane značajke poput automatske kontrole glasnoće (AGC) i smanjenja šuma.

7. Smanjenje opterećenja mikrokontrolera

Korištenjem vanjskog audio kodeka, mikrokontroler je oslobođen zadataka konverzije signala i drugih audio-specifičnih operacija. Ovo smanjuje opterećenje procesora, omogućavajući mikrokontroleru da se usredotoči na druge važne zadatke, poput obrade podataka, upravljanja sustavima ili komunikacije, bez ugrožavanja performansi.

8. Veća fleksibilnost u dizajnu

Korištenjem vanjskog audio kodeka, dizajneri imaju veću fleksibilnost u odabiru komponenata prema specifičnim zahtjevima projekta. Mogu birati kodeke s različitim frekvencijama uzorkovanja, brojem kanala i raznim dodatnim funkcijama, što im omogućava prilagodbu sustava prema specifičnim audio potrebama.



Slika 2: Korišteni Audio codec proto

Audio Codec Board - PROTO je razvojni alat za WM8731 audio kodek, idealan za upotrebu u ugrađenim aplikacijama koje zahtijevaju reprodukciju zvuka. Na pločici se nalazi audio kodek WM8731 koji pruža stereo linijske i mono ulaze za razinu mikrofona. Ovaj modul također koristi stereo 24-bitne sigma-delta ADC i DAC pretvarače s višebitnim prebacivanjem, zajedno s digitalnim interpolacijskim i dekonvolucijskim filtrima. Stereo audio izlazi su pojačani kako bi omogućili upravljanje slušalicama putem programabilne kontrole glasnoće. Također su dostupni linijski izlazi, zajedno s krugovima za isključivanje (*anti-thump*) i upravljanje uključivanjem/isključivanjem. Uređaj se kontrolira putem dvožičnog (I2C) ili trožičnog (SPI) serijskog sučelja.

3.3. Saleae logic analyzer

Saleae Logic Analyzer hardver je kompaktan i svestran uređaj dizajniran za hvatanje digitalnih signala visoke preciznosti. Glavne karakteristike hardvera uključuju sljedeće stavke:

1. **Ulazni kanali:** Ovisno o modelu, *Saleae* logički analizatori dolaze s različitim brojem ulaznih kanala (od 8 do 16), što omogućuje istovremeno praćenje više signala. Svaki kanal može pratiti digitalne signale, a na višim modelima i analogne signale.
2. **Brzina uzorkovanja:** Ovi analizatori nude visoke stope uzorkovanja, u rasponu od nekoliko megaherca (MHz) pa sve do 500 MHz ili više, ovisno o modelu. To omogućava precizno hvatanje i analizu signala u brzom vremenskom okviru.
3. **Povezivanje:** *Saleae Logic Analyzer* se povezuje putem USB-a na računalo, koje služi kao sučelje za prikazivanje i analizu signala. USB 3.0 je podržan na novijim modelima, omogućujući brzi prijenos podataka s analizatora na računalo.

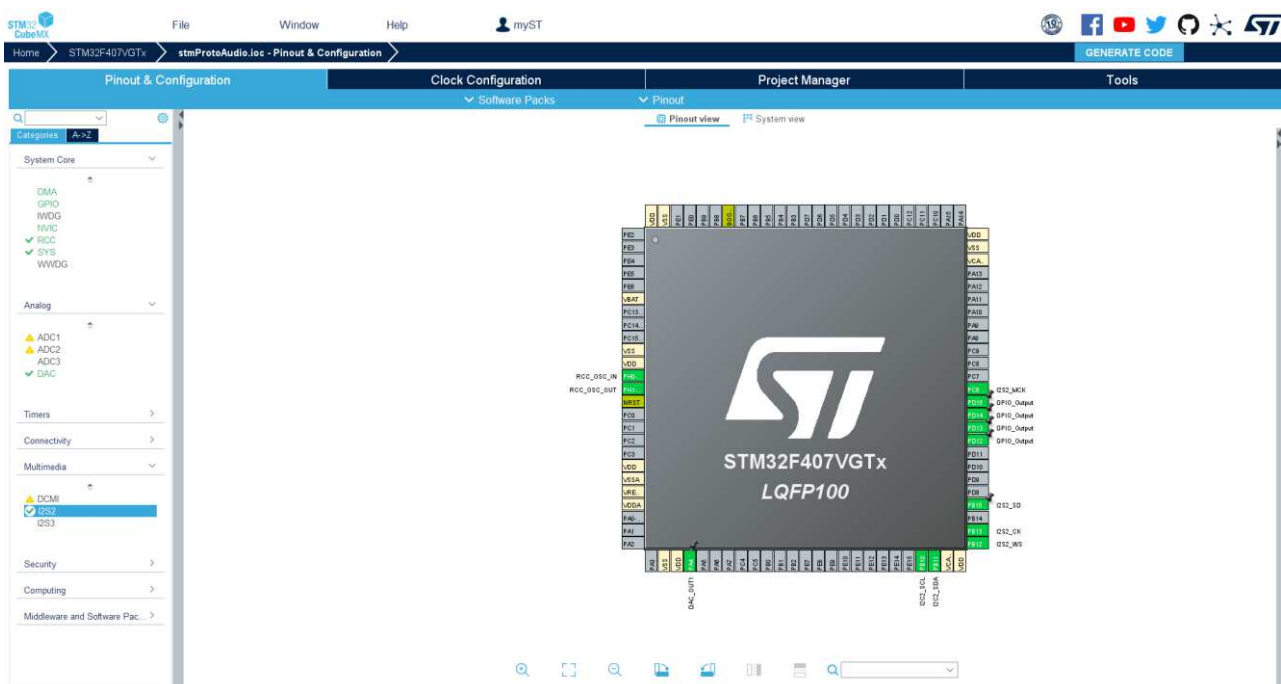


Slika 3: Saleae logic analyzer

4. Sofverska izvedba

4.1. STM32CubeMX

STM32CubeMX je korisnički orijentirani softverski alat koji omogućava jednostavno i brzo konfiguriranje STM32 mikrokontrolera. Korištenjem grafičkog sučelja, korisnici mogu vizualno odabrati periferije, postaviti pinove, te konfigurirati osnovne postavke poput sata, memorije i komunikacijskih sučelja. STM32CubeMX također automatski generira početni kod za HAL (*Hardware Abstraction Layer*), što ubrzava razvoj aplikacija. Osim toga, integrira se s raznim razvojnim okruženjima (IDE), olakšavajući prijelaz iz konfiguracije u kodiranje i testiranje. Ovaj alat je koristan za optimizaciju dizajna, osobito u složenim projektima sa STM32 mikrokontrolerima.



Slika 4: Primjer prozora za konfiguraciju STM32 mikrokontrolera

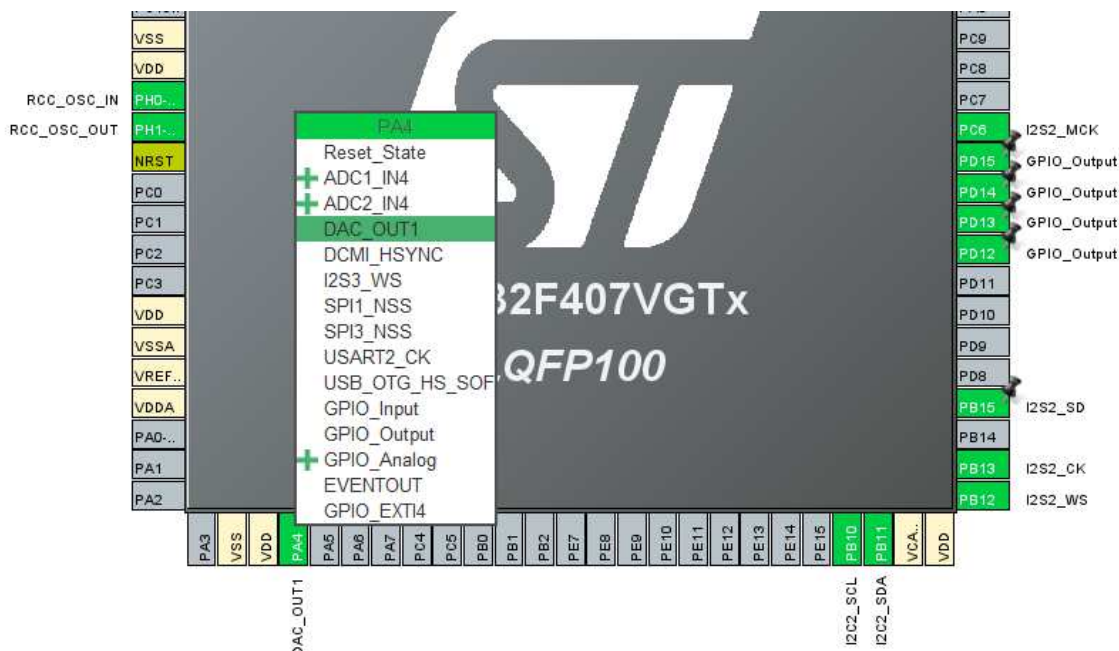
Prozor *Pinout & Configuration* u STM32CubeMX softveru ključan je dio procesa konfiguracije STM32 mikrokontrolera. Ovaj prozor nudi korisnicima intuitivan način za odabir i konfiguriranje funkcionalnosti pojedinih pinova mikrokontrolera, a pruža pregled svih raspoloživih periferija, ulaza/izlaza i njihovih poveznica s pinovima na mikrokontroleru.

Mogućnosti *Pinout & Configuration* prozora:

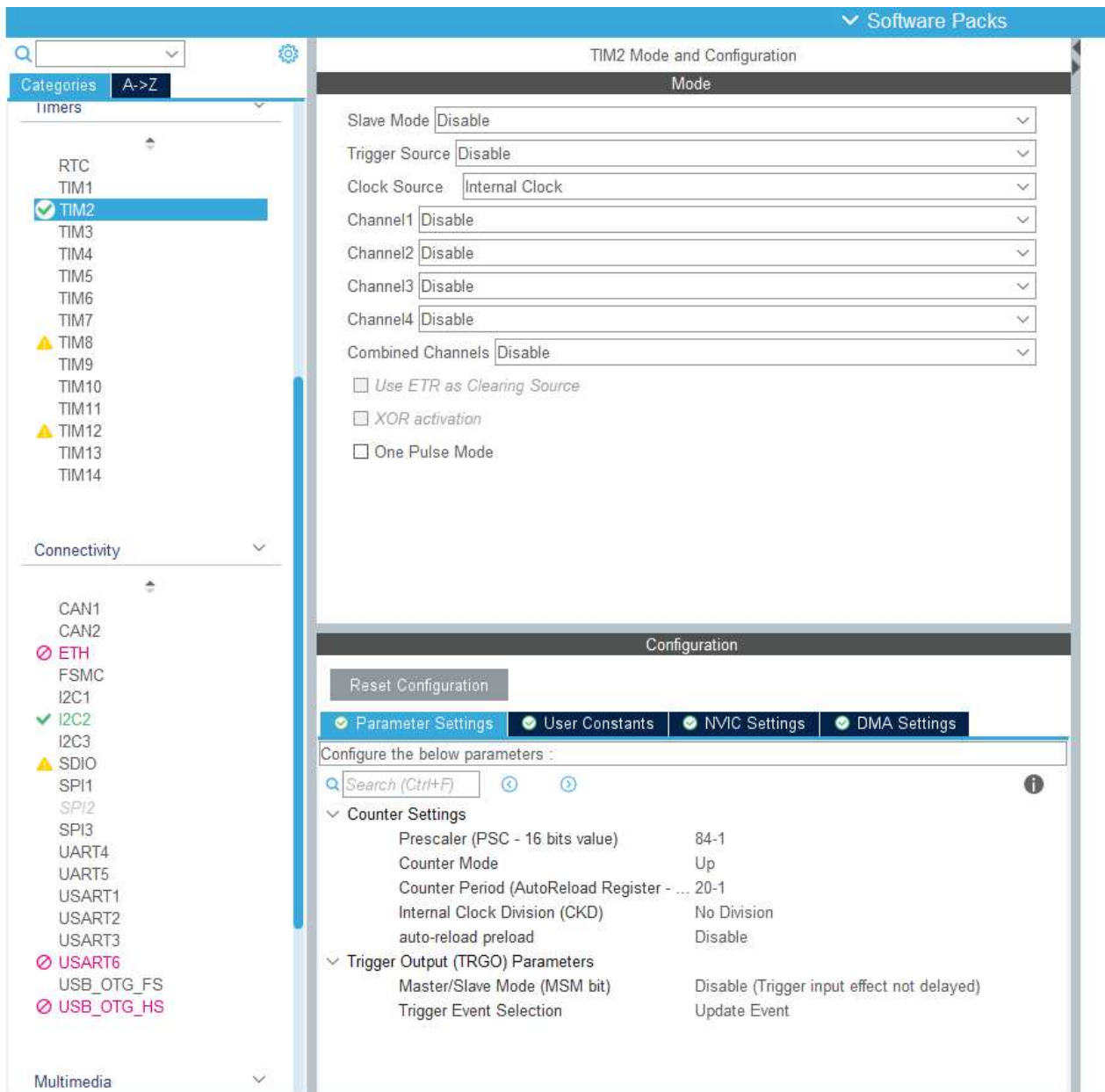
1. **Grafički prikaz pinova:** Prozor prikazuje sve pinove mikrokontrolera u obliku interaktivnog shematskog prikaza. Korisnik može vizualno odabrati i konfigurirati funkcije za svaki pin, poput digitalnih ulaza i izlaza, komunikacijskih sučelja (I2C, SPI, UART), PWM izlaza, ADC i DAC funkcija, te mnogih drugih.
2. **Automatski izbor periferija:** Kad korisnik odabere određenu periferiju, STM32CubeMX automatski dodjeljuje odgovarajuće pinove mikrokontrolera za tu funkciju, vodeći računa o njihovim kompatibilnostima i mogućim konfliktima. Na taj način se osigurava da su svi odabrani moduli ispravno konfigurirani bez sukoba između pinova.
3. **Konfiguracija i optimizacija periferija:** Prozor omogućava postavljanje specifičnih parametara za svaku periferiju. Na primjer, korisnik može postaviti brzinu komunikacijskih sučelja, način rada (master/slave), brzinu uzorkovanja ADC-a, te ostale važne parametre potrebne za rad odabranih modula.
4. **Prepoznavanje konflikata:** STM32CubeMX automatski detektira sukobe između pinova kada dvije periferije zahtijevaju korištenje istih pinova. Softver odmah upozorava korisnika na te konflikte, omogućujući brzo rješavanje kroz zamjenske pinove ili promjenu postavki.
5. **Filteri i sortiranje funkcija:** STM32CubeMX pruža mogućnost filtriranja funkcija po kategorijama, poput komunikacijskih sučelja, tajmera, analognih ulaza i izlaza, čime se korisniku olakšava pretraživanje i odabir željenih funkcionalnosti na mikrokontroleru.
6. **Generiranje izvještaja o pinovima:** Nakon što se završi konfiguracija, softver omogućava generiranje detaljnog izvještaja o raspodjeli pinova i konfiguraciji periferija. Ovaj izvještaj olakšava pregled funkcionalnosti i služi kao referenca tijekom faze razvoja.

Prednosti i važnost *Pinout & Configuration* prozora:

1. **Brzina i učinkovitost:** Umjesto ručnog pregledavanja dokumentacije za odabir pinova i njihovo podešavanje, ovaj prozor omogućava vizualni i automatizirani način rada, čime se značajno ubrzava proces konfiguracije mikrokontrolera.
2. **Smanjenje grešaka:** Korištenjem automatskog odabira pinova i prepoznavanja konflikata, smanjuje se mogućnost grešaka u dizajnu, posebno u kompleksnim projektima gdje je potrebno koristiti više periferija.
3. **Jednostavnost korištenja:** Intuitivno grafičko sučelje olakšava rad korisnicima svih razina iskustva. Početnici se lako mogu snaći u procesu konfiguracije, dok iskusni korisnici mogu brzo testirati različite opcije i kombinacije pinova.
4. **Fleksibilnost u dizajnu:** Prozor omogućava jednostavno isprobavanje različitih konfiguracija pinova i periferija kako bi se pronašlo optimalno rješenje za specifične potrebe projekta, omogućavajući prilagodljivost u razvoju aplikacija.



Slika 5; Višestruki odabir mogućnost namjene pojedinog pina (u ovom slučaju PA4 pina)



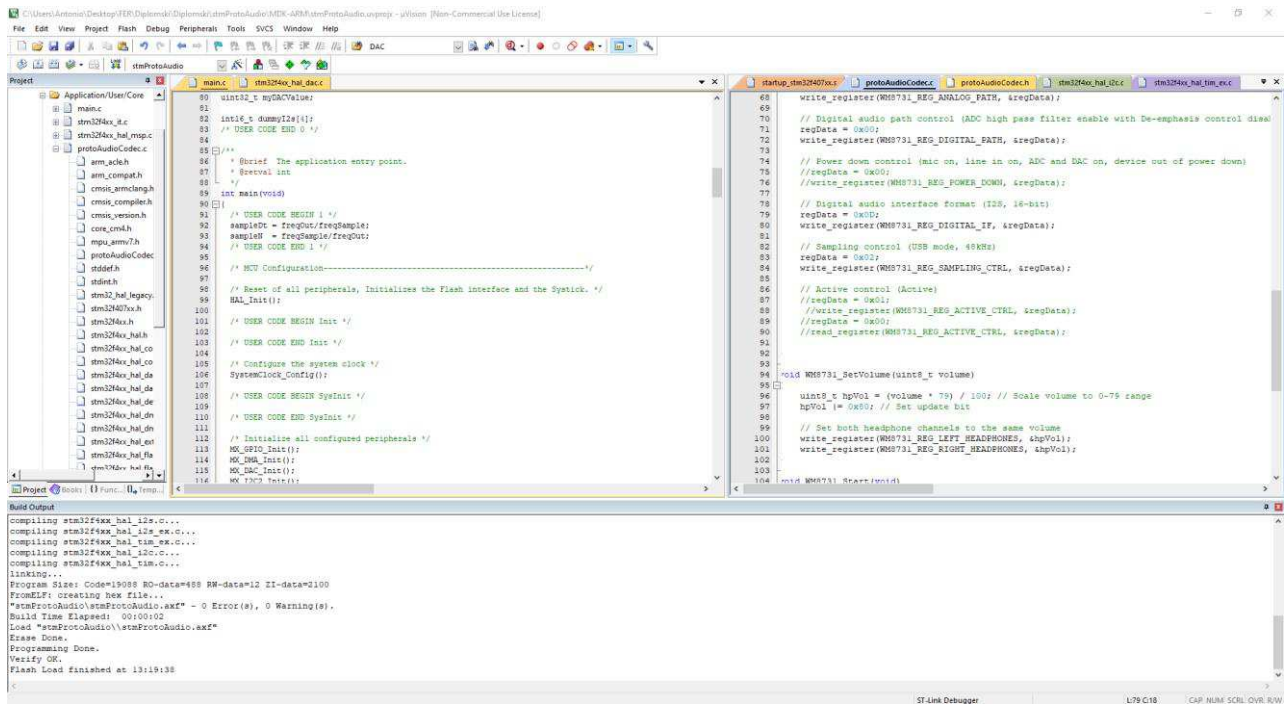
Slika 6; Prozor za konfiguraciju parametara neke periferije (u ovom slučaju Timer 2)

Zašto je *Pinout & Configuration* bitan?

Pinout & Configuration prozor je ključan alat za početnu fazu razvoja s STM32 mikrokontrolerima. On omogućuje brz i točan odabir perifernih funkcija, što je neophodno za pravilan rad sustava. Pomaže korisnicima da lako upravljaju kompleksnim sustavima s mnogim periferijama, čime se štedi vrijeme i smanjuje rizik od pogrešaka. Njegova funkcionalnost pojednostavljuje cijeli proces postavljanja mikrokontrolera, čineći ga važnim korakom u dizajniranju ugrađenih sustava.

4.2. Keil μ Vision5

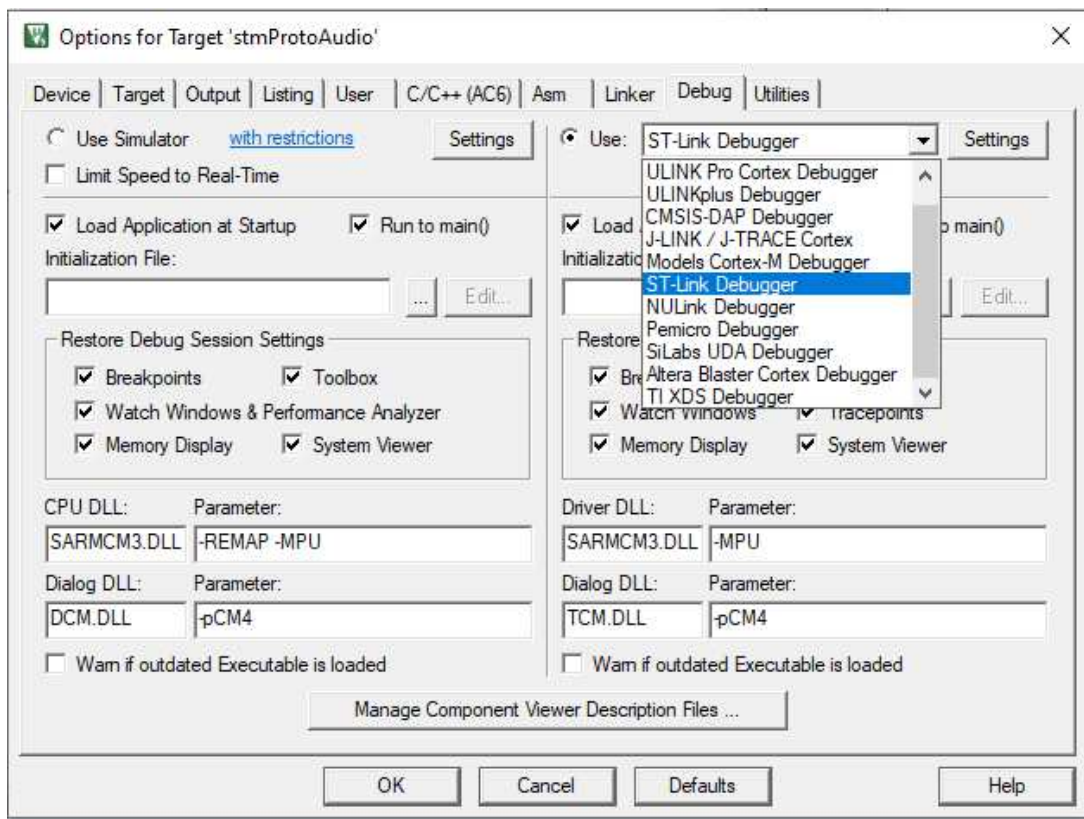
Keil MicroVision 5 (μ Vision5) je razvojno okruženje (IDE) za programiranje mikrokontrolera koje pruža sveobuhvatne alate za razvoj aplikacija baziranih na ARM Cortex-M procesorima, uključujući STM32 seriju. Omogućuje pisanje, debugiranje i optimizaciju koda koristeći C/C++ kompajler, s integriranim alatima za simulaciju i upravljanje projektima. IDE nudi podršku za ugradbene projekte, uključujući pregled hardverskih registara u stvarnom vremenu, što pomaže u detekciji i otklanjanju grešaka. MicroVision 5 također podržava CMSIS (Cortex Microcontroller Software Interface Standard) knjižnice, olakšavajući razvoj aplikacija na ARM arhitekturi.



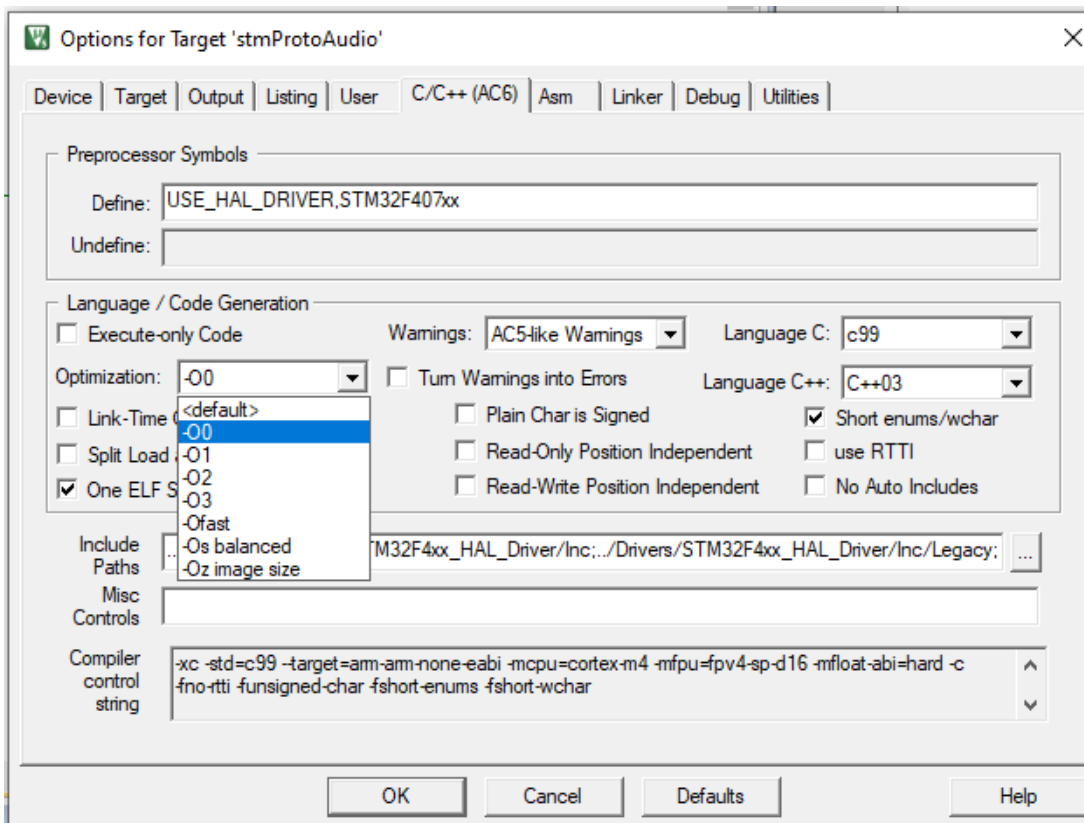
Slika 7: Keil Microvision 5

Keil MicroVision 5 je odličan alat za korištenje s STM32 mikrokontrolerima zbog nekoliko ključnih razloga:

1. **Široka podrška za STM32:** MicroVision 5 dolazi s potpunom podrškom za STM32 mikrokontrolere, uključujući CMSIS (*Cortex Microcontroller Software Interface Standard*) knjižnice i HAL (*Hardware Abstraction Layer*) podršku, što omogućava jednostavnije korištenje i konfiguraciju periferija STM32 čipova.
2. **Moćno okruženje za debugiranje:** IDE nudi napredne funkcije za debugiranje, uključujući mogućnost pregleda i manipulacije hardverskih registara u stvarnom vremenu, praćenje promjena varijabli, postavljanje breakpoints, te detaljan uvid u ponašanje aplikacije na niskoj razini. Ovo značajno olakšava otklanjanje grešaka u kodu.
3. **Optimizacija koda:** Keil kompajler je poznat po svojoj efikasnosti i mogućnostima optimizacije koda, što je važno za ugrađene sustave s ograničenim resursima, kao što su STM32 mikrokontroleri. Optimizacija omogućuje bolje performanse i manju potrošnju memorije.
4. **Integracija s STM32CubeMX:** MicroVision 5 se lako integrira s STM32CubeMX alatom, omogućujući korisnicima da brzo generiraju osnovni kod za konfiguraciju mikrokontrolera, uključujući pinove i periferije, te ga odmah uvezu u Keil IDE za daljnji razvoj.
5. **Opsežna podrška za simulaciju:** MicroVision 5 omogućava simulaciju rada mikrokontrolera bez fizičkog hardvera, što je korisno za razvoj i testiranje koda u ranijim fazama projekta.



Slika 8: Podešavanje Debug okruženja



Slika 9: Podešavanje okruženja za optimizaciju

4.3. *Saleae logic analyzer*

Saleae softver je intuitivna platforma koja se koristi zajedno s **Saleae Logic Analyzer** uređajem za snimanje, vizualizaciju i analizu digitalnih i analognih signala. Glavne značajke softvera uključuju:

1. **Vizualizacija signala:** Softver omogućava pregled signala u realnom vremenu, gdje korisnici mogu detaljno pratiti digitalne i analogne signale preko više kanala. Grafičko sučelje olakšava analizu vremenskih odnosa između signala, a povećavanje i pomicanje kroz vremensku skalu omogućava preciznu inspekciju podataka.
2. **Dekodiranje komunikacijskih protokola:** Saleae softver automatski dekodira mnoge uobičajene komunikacijske protokole kao što su I2C, SPI, UART, CAN i mnogi drugi, prikazujući podatke u lako razumljivom formatu. To korisnicima pomaže u brzom prepoznavanju i rješavanju problema u komunikacijama između uređaja.
3. **Istovremena analiza analognih i digitalnih signala:** Napredniji modeli Saleae analizatora podržavaju simultano snimanje analognih i digitalnih signala, a softver omogućava njihovu sinkroniziranu analizu. Ova funkcija je korisna za praćenje naponskih promjena u odnosu na digitalne operacije.
4. **Korisnički prilagodljivo sučelje:** Softver omogućava jednostavno podešavanje kanala, odabir uzorkovanja, postavljanje oznaka i mjerenje razmaka između događaja, čime se omogućava detaljna analiza vremenskih i frekvencijskih karakteristika signala.
5. **Snaga filtriranja i analiza podataka:** Saleae softver dolazi s opcijama za filtriranje i označavanje specifičnih dijelova signala, olakšavajući fokusiranje na ključne dijelove analize. Također omogućava izvoz podataka za daljnju analizu u drugim alatima.



Slika 10: Primjer Saleae softver prozora za čitanje poslanih/primljenih signala

5. Komunikacijski protokoli

5.1. I2C komunikacijski protokol

I2C (*Inter-Integrated Circuit*) je serijski (*2-wire protocol*) komunikacijski protokol koji omogućuje povezivanje više uređaja putem samo dvije linije: **SDA** (*Serial Data*) za prijenos podataka i **SCL** (*Serial Clock*) za sinkronizaciju. Ovaj protokol koristi *master-slave* arhitekturu, gdje jedan uređaj (*master*) upravlja komunikacijom, dok ostali uređaji (*slave*) odgovaraju na naredbe mastera.

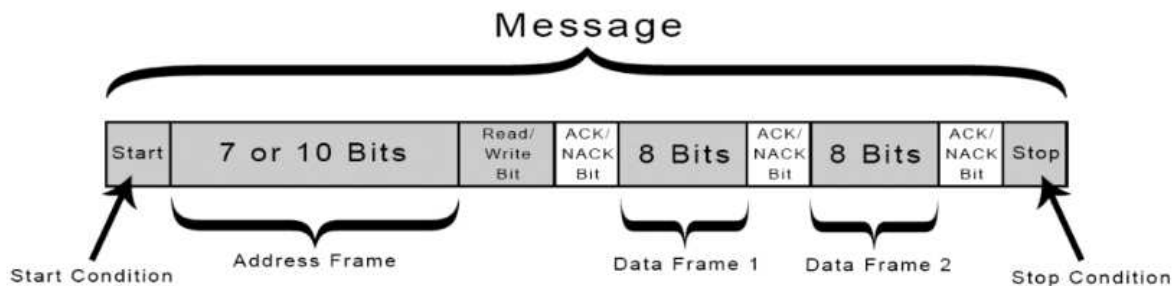
Kako radi:

- **Master** generira takt na SCL liniji i upravlja početkom i završetkom prijenosa podataka.
- **Slave uređaji** imaju jedinstvene adrese, a master im šalje podatke ili traži odgovore slanjem te adrese.
- Podaci se prenose bit po bit putem SDA linije, sinkronizirani SCL signalom.
- I2C podržava dvosmjernu komunikaciju i koristi **ACK/NACK** (potvrda/neuspjeh) bit za provjeru uspješnosti prijenosa.
-

I2C je pogodan za spajanje perifernih uređaja poput senzora, memorije i audio kodeka u sustavima s mikrokontrolerima.

Kod I2C protokola, podaci se prenose u porukama. Poruke se dijele na okvire podataka. Svaka poruka ima adresni okvir koji sadrži binarnu adresu *slave* uređaja, te jedan ili više okvira podataka koji sadrže podatke koji se prenose. Poruka također uključuje startna i zaustavna stanja, bitove za čitanje/pisanje, te ACK/NACK bitove između svakog okvira podataka.

Na slici 11 ćemo vidjeti kako izgleda sadržaj jedne I2C poruke:



Slika 11: Sadržaj I2C poruke

Startna Stanja: SDA linija prelazi s visoke razine napona na nisku razinu napona prije nego što SCL linija prijeđe s visoke na nisku razinu.

Zaustavna Stanja: SDA linija prelazi s niske razine napona na visoku razinu napona nakon što SCL linija prijeđe s niske na visoku razinu.

Adresni okvir: Sekvenca od 7 ili 10 bita jedinstvena za svakog slave uređaja koja identificira slave kada master želi komunicirati s njim.

Bit Čitanja/Pisanja: Jedan bit koji specificira šalje li master podatke slave uređaju (niska razina napona) ili traži podatke od njega (visoka razina napona).

ACK/NACK Bit: Svaki okvir u poruci prati bit potvrde/nepotvrde. Ako je adresni ili podatkovni okvir uspješno primljen, uređaj primatelj vraća ACK bit pošiljatelju.

5.2. I2S komunikacijski protokol

I2S (*Inter-Integrated Circuit Sound*) je serijski (*3-wire protocol*) komunikacijski protokol dizajniran za prijenos audio podataka između integriranih krugova. Koristi tri glavne linije: **SD** (*Serial Data*) za prijenos audio podataka, **SCK** (*Serial Clock*) za sinkronizaciju podataka, i **WS** (*Word Select*) za označavanje različitih audio kanala (lijevi/desni). I2S omogućuje visokokvalitetan prijenos audio signala s minimalnim kašnjenjem i bez kompresije, što ga čini idealnim za aplikacije kao što su digitalni audio procesori, DAC-ovi (digitalno-analogne pretvarače) i ADC-ovi (analogno-digitalne pretvarače).

Kako radi:

1. Osnovne linije:

- **SD (Serial Data)**: Linija za prijenos audio podataka. Podaci se šalju bit po bit.
- **SCK (Serial Clock)**: Linija za pružanje taktnih signala koji sinkroniziraju prijenos podataka.
- **WS (Word Select)**: Linija koja označava koji audio kanal (lijevi ili desni) se trenutno prenosi. Može se koristiti i za označavanje početka novog uzorka podataka.

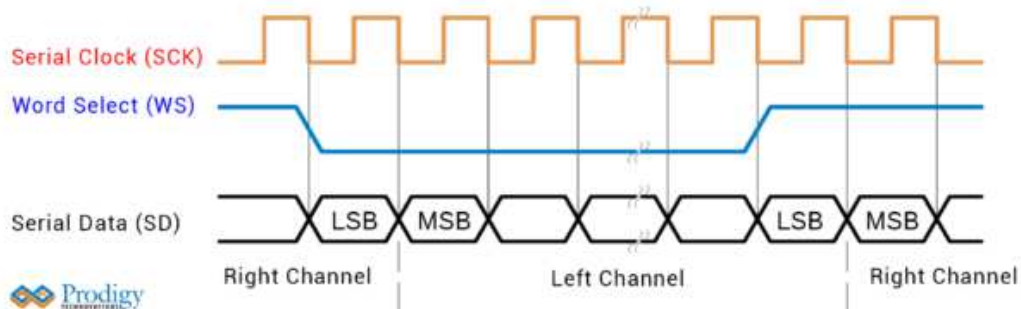
2. Radni ciklus:

- **Start**: Prijenos započinje kada master uređaj postavi taktni signal na SCK liniji.
- **Podaci**: Audio podaci se šalju serijski na SD liniji u skladu s taktovima na SCK liniji. Svaki bit podataka se prenosi u vrijeme kada SCK linija signalizira.
- **Word Select**: WS linija se koristi za označavanje da li se prenose lijevi ili desni kanal audio podataka. Ako je WS u visokom stanju, prenosi se jedan kanal (npr. lijevi); ako je WS u niskom stanju, prenosi se drugi kanal (npr. desni).

3. Format podataka:

- I2S obično koristi 16-bitni, 24-bitni ili 32-bitni format za audio podatke, ovisno o konfiguraciji.
- Svaki uzorak podataka (word) se prenosi u nekoliko taktova, ovisno o dubini bitova.

- Continuous Serial Clock (SCK);
- Word Select (WS); and
- Serial Data (SD)



The timing diagram for I2S

Slika 12: Vremenski dijagram slanja I2S poruke

5.3. Razlika između I2C i I2S protokola

- **Namjena:** I2C se koristi za povezivanje s općim *inter-IC* komunikacijama, poput EEPROM-a ili senzora, dok se I2S koristi isključivo za audio uređaje.
- **Arhitektura:** I2C podržava višestruke *master*e i *slave* uređaje, dok I2S podržava samo jednog *master*a.
- **Broj linija:** I2C je dvožična komunikacija, dok je I2S trožična.
- **Clock stretching:** I2C podržava produžavanje takta (*clock stretching*), dok I2S ne podržava ovu funkcionalnost.
- **Overhead:** I2C ima dodatni *overhead* s bitovima za start i stop, dok I2S ne koristi bitove za start i stop.
- **Potvrda:** I2C koristi komunikaciju potvrde (*acknowledgment*) nakon svakog bajta prijensa.
- **Otpornici:** I2C zahtijeva *pull-up* otpornike.

Prednosti I2C protokola:

- **Jednostavno dodavanje uređaja:** Lako je dodati novi slave uređaj u seriju sa ostalima. Samo dodajte novi uređaj bez potrebe za dodatnim odabirom, za razliku od SPI-a.

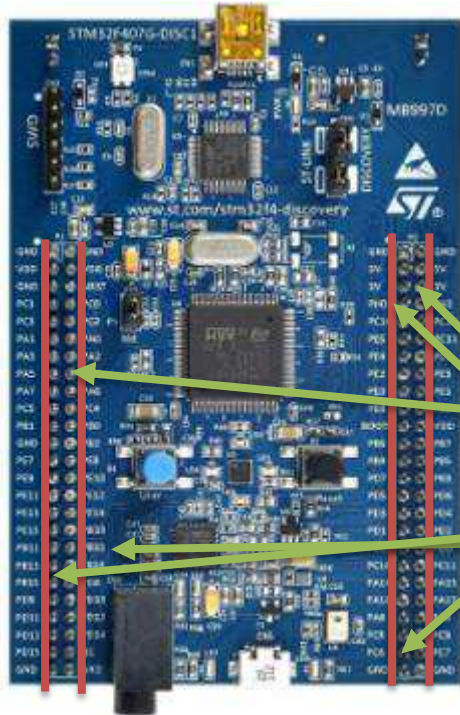
Prednost I2S protokola:

- **Audio:** Pruža cijeli audio lanac i eliminira potrebu za dodatnim predpojačalom te DAC-om i ADC-om.

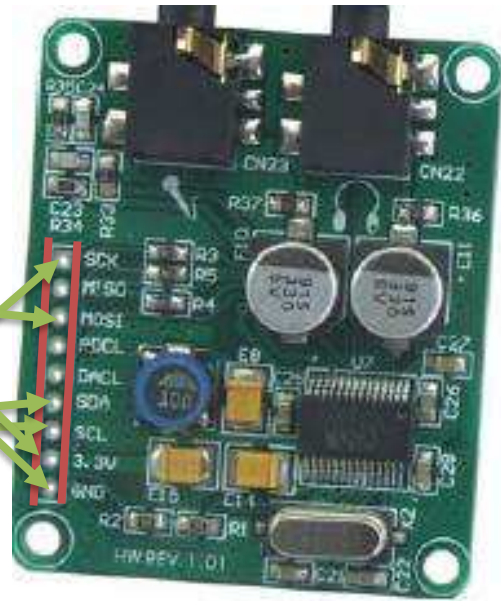
6. Vlastita izvedba

6.1. Hardverski dio izvedbe

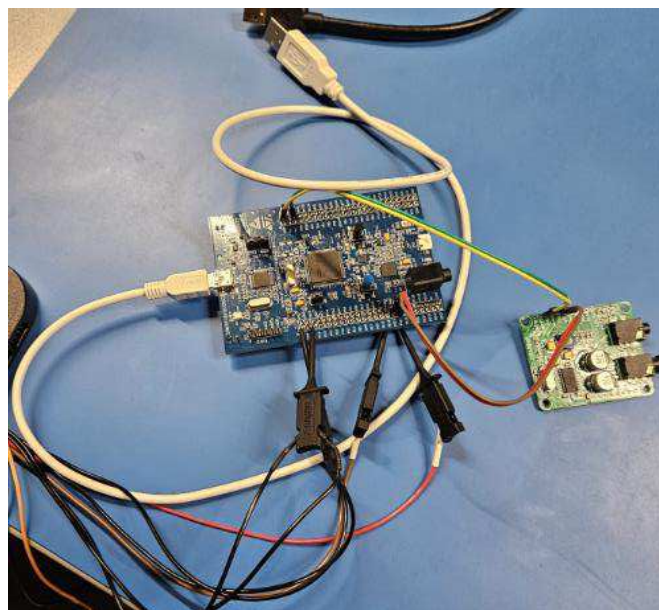
Od komponenti potrebnih za izvedbu ovog rada, potrebno nam je uspješno spojiti sljedeće komponente.



Slika 13: Proto audio codec



Slika 14: STM32F4 Discovery



Slika 15: Povezanost STM32F4 i kodeka

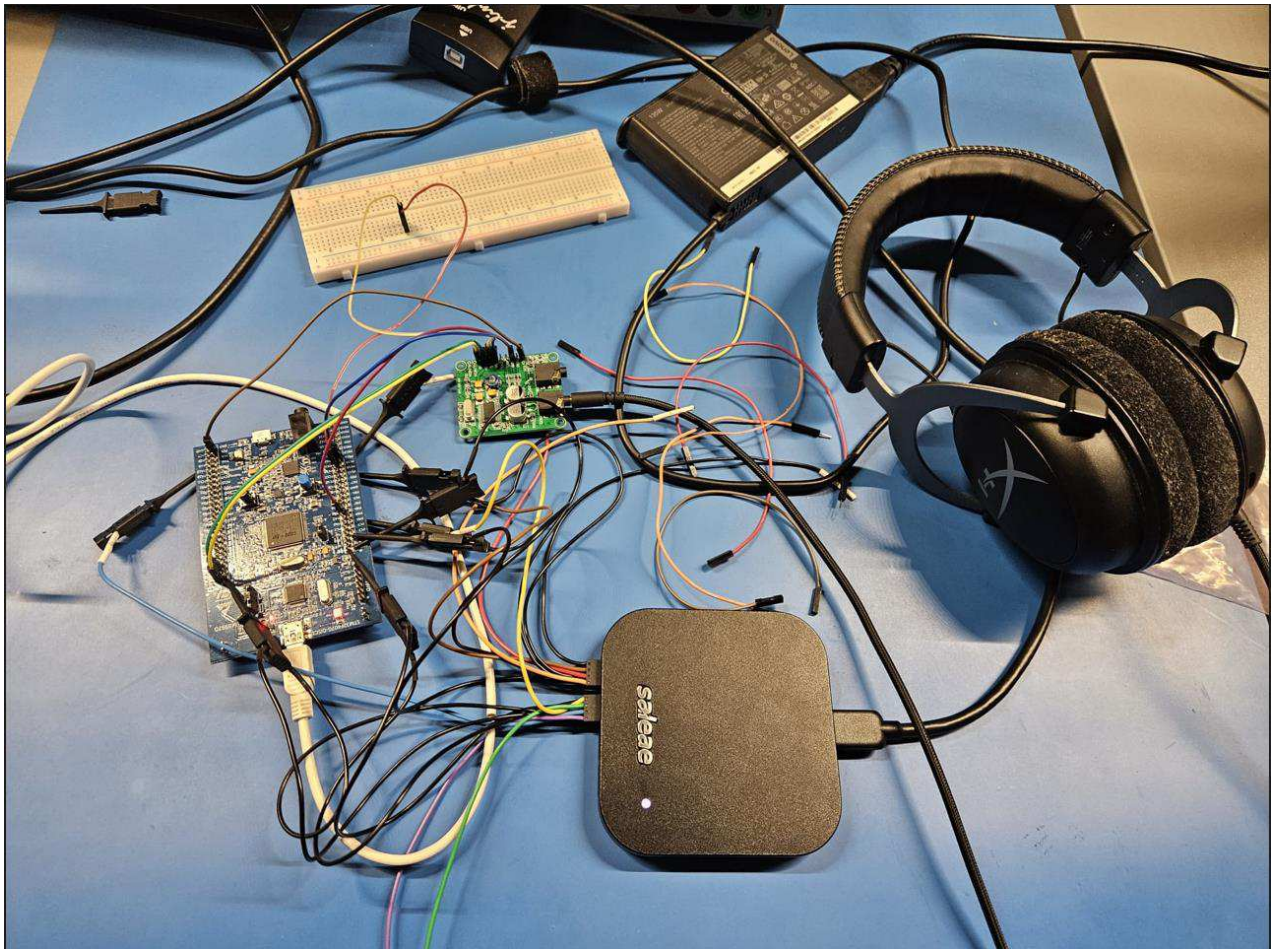
Od preostalih stavki su nam preostale analogne slušalice koje ćemo priključiti na izlazni priključak Proto audio kodeka, te nekoliko žica i konektora kojima ćemo povezati STM32F4 *Discovery* sa kodekom. Za provjeru signala koje šaljemo sa *Discovery-a* na kodek, koristiti ćemo Saleae logički analizator. Kako smo povezali sve hardverske komponente, moguće je vidjeti na slici 18.



Slika 16: Saleae logički analizator



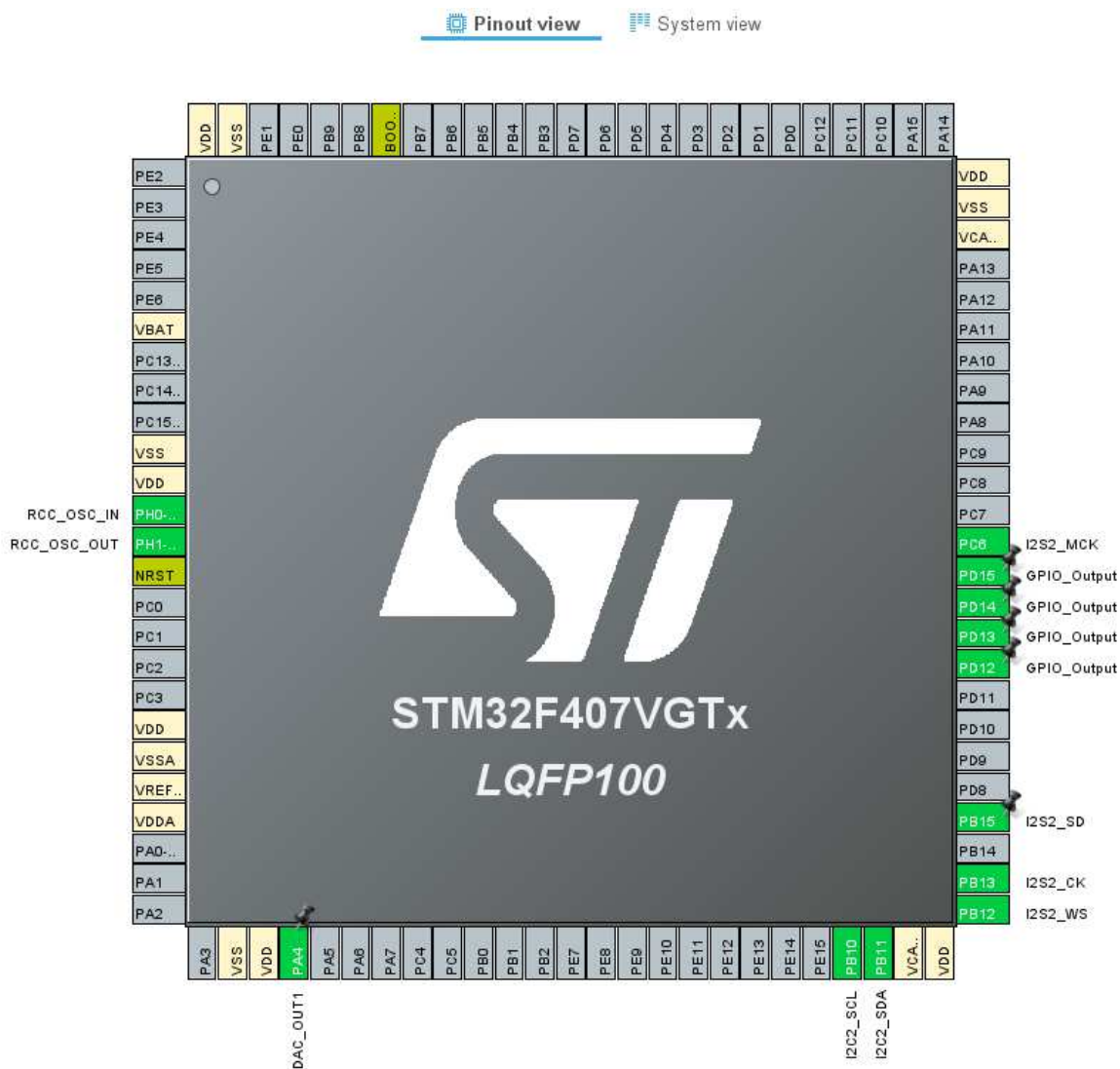
Slika 17: Slušalice



Slika 18: Ožičenje svih potrebnih komponenti

6.2. Softverski dio izvedbe

Kako bi pravilno konfigurirali željene periferije poput I2C-a, GPIO *input/outputa*, I2S-a, potrebno je pravilno konfigurirati željene *pinove* na STM32F4 *Discovery* pročići. To ćemo postići koristeći **STM32CubeMX** softver, a pinove ćemo konfigurirati na sljedeći način.



Slika 19: Konfiguracija pinova STM32F407VGT6 mikrokontrolera

Konfiguracija I2C sučelja:

- Kao što znamo, I2C komunikacijski protokol je dvožični pa stoga moramo podesiti 2 pina kako bi omogućili komunikaciju sa I2C protokolom
- Podešavanje pinova PB10 (kao SCL signal takta za I2C komunikaciju) i PB11 (kao SDA za upis i slanje podataka putem I2C komunikacijskog protokola)

Konfiguracija I2S sučelja

- Za razliku od I2C protokla, I2S je trožični te zbog toga moramo podesiti 3 pina kako bi uspješno mogli komunicirati sa kodekom
- Podešavanje pinova PB12 (*WS - Word Select* za odabir kanala), PB13 (signal takta za I2S komunikaciju) i PB15 (*SD - Serial Data* za prijenos podataka)

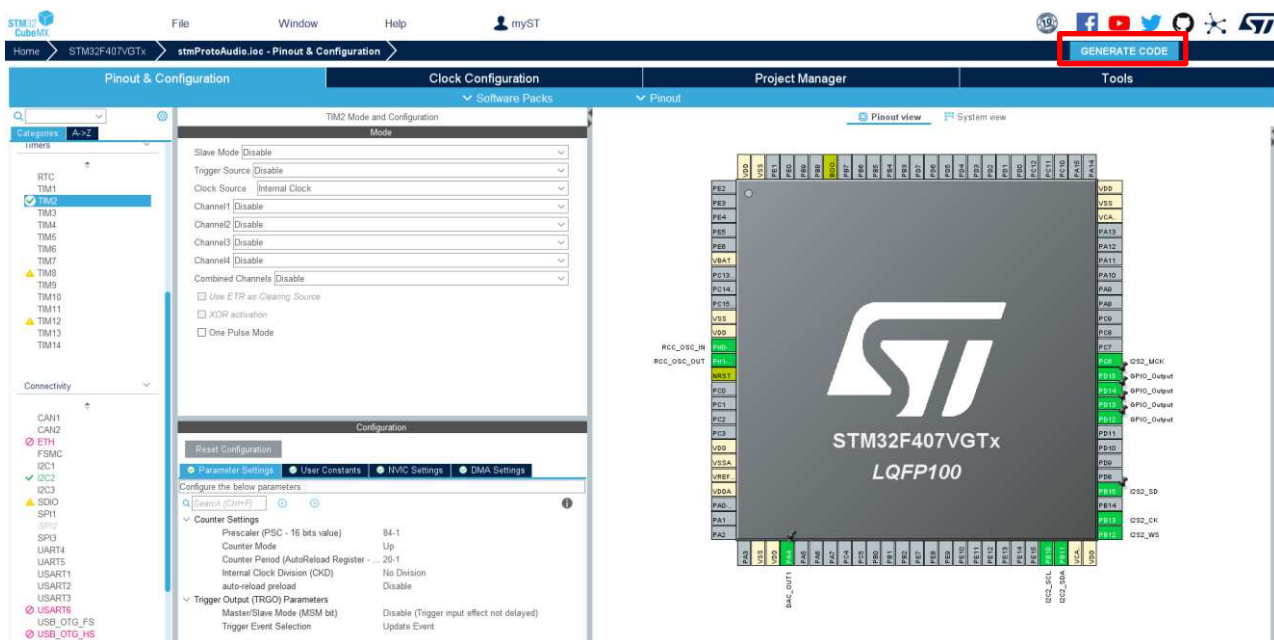
Konfiguracija DAC-a

- Omogućavanje digitalno-analognog izlaza kojeg ćemo serijski spojiti na kodek

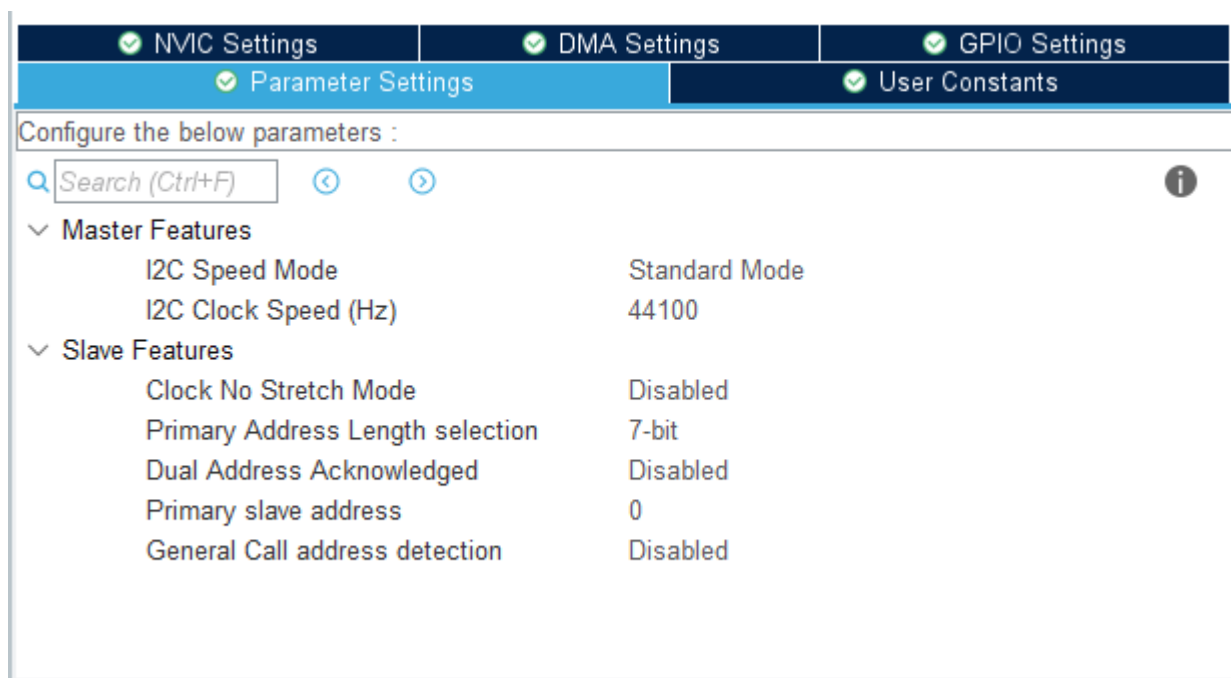
Dodatno:

- Kao dodatak smo omogućili i nekoliko GPIO-ova (ulaza/izlaza PD12-PD15) te STM-ov interni oscilator (PH0 i PH1)

Nakon što smo odabrali željene periferije i konfigurirali potrebne pinove, pritskom na opciju *Generate code* generiramo kod koji će sadržavati inicijalizacije naših komponenti sustava na vrijednosti koje smo postavili u STM32CubeMX-u.



Slika 20: Generiranje koda sa podešenim komponentama STM32F4 Discovery-a



Slika 21: Konfiguracija I2C-a u STM32CubeMX-u

```

244  /* USER CODE END I2C2_Init 1 */
245  hi2c2.Instance = I2C2;
246  hi2c2.Init.ClockSpeed = 44100;
247  hi2c2.Init.DutyCycle = I2C_DUTYCYCLE_2;
248  hi2c2.Init.OwnAddress1 = 0;
249  hi2c2.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
250  hi2c2.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
251  hi2c2.Init.OwnAddress2 = 0;
252  hi2c2.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
253  hi2c2.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
254  if (HAL_I2C_Init(&hi2c2) != HAL_OK)
255  {
256      Error_Handler();
257  }

```

Slika 22: Generirani kod u Keil Microvision5 za konfiguraciju postavljenu u STM32CubeMX-u

Kao testni uzorak smo koristili generirani digitalni sinusni signal frekvencije 1 kHz koji smo zatim morali postati na digitalno-analogni izlaz na *pinu* PA4 na sljedeći način.

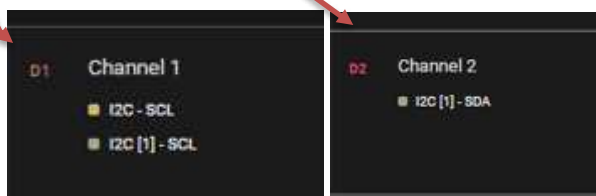
```
401 if(htim->Instance == TIM2){
402     mySinValue = sinf(i_t * 2 * PI * sampleDt);
403
404     //Convert float to decimal
405     myDACValue = (mySinValue +1)*127;
406
407     //Output the sample to DAC
408     HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1, DAC_ALIGN_8B_R, myDACValue);
409
410     i_t++;
411     if (i_t >= sampleN){
412         i_t = 0;
413     }
```

Slika 23: Generiranje sinusnog signala i postavljanje vrijednosti u memoriju odakle će se kasnije slati na DAC izlaz

Kako bi se uvjerali da uistinu šaljemo sinusni signal na slušalice, potrebno je najprije osposobiti komunikaciju sa kodekom koristeći I2C protokol. Da bi se uvjerali da zaista uspješno komuniciramo sa kodekom potrebno je spojiti napajanje od 3V, masu, te SCL i SDA signale. Na slici 24 će se jasno vidjeti kako je gornji signal signal takta za I2C protokol frekvencijom 44.1kHz dok je donji signal SDA signal koji označava podatke koje šaljemo prema kodeku. Za provjeru poslanih i primljenih signala koristiti će se *Saleae logic analyzer*.



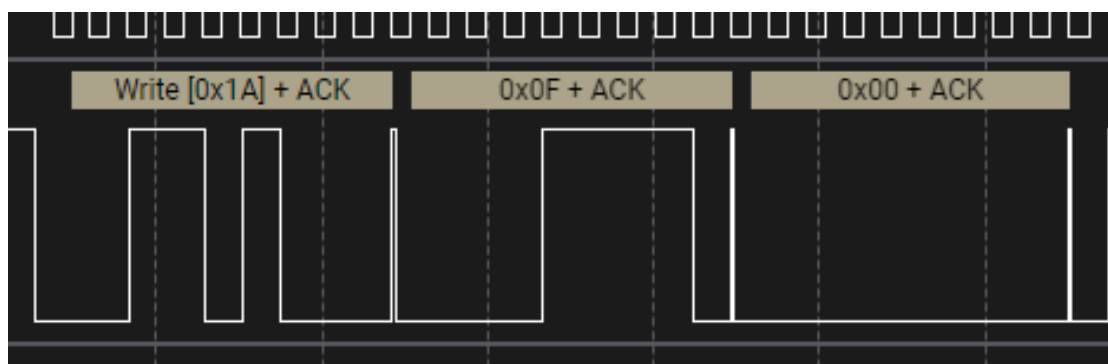
Slika 24: I2C komunikacija sa kodekom



Kako bi pobliže objasnili što šalje preko SDA, potrebno je također prisjetiti se kako izgleda poruka koja se šalje I2C protokolom.

Za primjer ćemo pogledati kako na početku možemo resetirati kodek kako bi se uvjerali da ćemo sve registre i postavke našeg kodeka vratiti na *default* postavke.

Kako bi uspješno resetirali kodek, najprije je potrebno saznati adresu mikrokontrolera Wolfson8731 koji se nalazi na našem kodeku i koji je zadužen za svu komunikaciju sa STM32F4 pločicom. Kako bi se informirali koja je njegova adresa te koji je registar unutar Wolfson-a zadužen za resetiranje mikrokontrolera, potrebno je pogledati shematski prikaz samog mikrokontrolera te njegov *datasheet* koji će nam dati sve potrebne informacije.



Slika 25: Resetiranje kodeka upisujući nule u registar za resetiranje

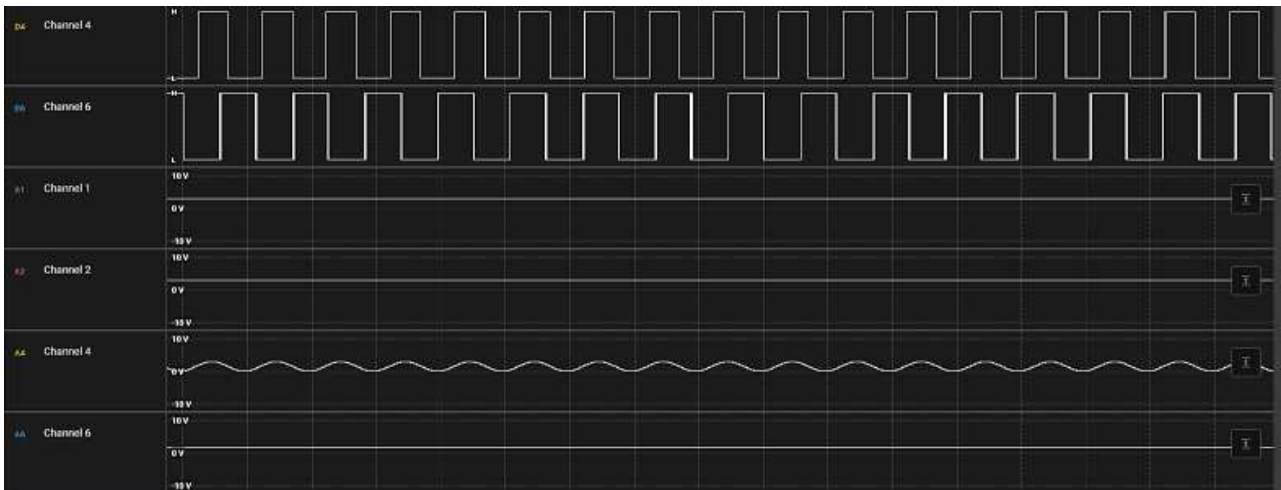
Da bismo uspješno resetirali kodek, potrebno je **upisati** tj. poslati podatke sa STM-a na Wolfson. To ćemo napraviti koristeći funkciju *Write*. Adresa Wolfson-a je 0x1A, *reset* registar je 0x0F, dok je vrijednost koju trebamo upisati u 0x0F registar, potrebna za resetiranje samog *chipa* jednaka 0x00. Kao potvrdu da smo zaista poslali te vrijednosti, te da ih je naš Wolfson primio nam govori “ACK” oznaka koja nam govori da je Wolfson potvrdio STM-u da je to upisano tamo gdje smo željeli.

Upisivanje vrijednosti u registre obaviti ćemo koristeći se sljedećom funkcijom.

```
11 // Static Functions Definitions
12 static void write_register(uint8_t reg, uint8_t *data)
13 {
14
15     iData[0] = reg;
16     iData[1] = data[0];
17     uint16_t addrShifted = (uint16_t) (WM8731_I2C_ADDR) << 1;
18     HAL_I2C_Master_Transmit(&i2cx, addrShifted, iData, 2, 100);
19 }
```

Slika 26: Funkcija za upisivanje vrijednosti u registre koristeći I2C protokol

Na sljedećoj slici ćemo moći vidjeti kako za generirani signal takta, periodički šaljemo sinusni signal.



Slika 27: Generirani signali takta i sinusnih uzoraka

Iz prvog signala (D4 *channel 4*) se jasno može vidjeti signal takta, dok su na sljedećem prozoru (D6 *channel 6*) lijepo prikazani periodički poslani podaci za digitalni sinusni signal. Na analognom prozoru (A4 *Channel 4*) se može vidjeti kako izgleda signal koji je prošao DA konverziju. Dobiveni izlazni analogni signal je pravilan sinusni signal, što smo željeni od početka.

Na analognom prozoru (A6 *Channel 6*) možemo vidjeti izlaz na slušalicama koji je ravan, što nam govori da još uvijek ne možemo čuti na slušalicama sinusni signal. Pretpostavka je da je problem u podešavanju signala takta između poslanih podataka i takta koji generira kodek.

7. Sljedeći koraci

Kao sljedeće korake bih naveo 2 bitna koraka za 100%-tni učinak ove metode.

1. korak: Sinkronizirati signal takta kako bi mogli uspješno čuti željeni generirani signal na slušalicama.
2. korak: Kako bi uklonili bilo kakve smetnje ili šumove, iskoristiti funkcije iz DSP CMSIS datoteke za FIR i IIR filtre nad ulaznim podacima (uzorcima sinusnog signala).

```
void arm_fir_f32 ( const arm_fir_instance_f32 * S,  
                 const float32_t *          pSrc,  
                 float32_t *                pDst,  
                 uint32_t                   blockSize  
                 )
```

Processing function for the floating-point FIR filter.

Parameters

- [in] **S** points to an instance of the floating-point FIR filter structure
- [in] **pSrc** points to the block of input data
- [out] **pDst** points to the block of output data
- [in] **blockSize** number of samples to process

Returns

none

Slika 28: Primjer funkcije iz CMSIS datoteke za korištenje FIR filtra

8. Zaključak

Zaključak korištenja I2S i I2C protokola u audio i općim komunikacijama između IC komponenti ističe njihovu specifičnu primjenu i prednosti u različitim scenarijima. **I2C** se pokazuje iznimno fleksibilnim za povezivanje velikog broja različitih uređaja, kao što su senzori, memorije i drugi periferni sklopovi, zahvaljujući jednostavnosti proširivanja i niskom broju potrebnih linija. **I2S**, s druge strane, specijaliziran je za prijenos audio signala s visokom kvalitetom i niskim kašnjenjem, što ga čini idealnim izborom u sustavima gdje je potreban čist i pouzdan prijenos zvuka, poput DAC-ova, ADC-ova i zvučnih sustava.

Kombinacija ovih protokola omogućuje razvoj učinkovitih sustava u ugrađenim aplikacijama gdje je potrebna jednostavna i pouzdana interkomunikacija između uređaja (I2C), kao i precizan prijenos audio signala (I2S). U konačnici, izbor protokola ovisi o specifičnim zahtjevima projekta, ali njihova upotreba omogućava visokokvalitetne performanse uz minimalnu složenost hardvera.

Literatura

<https://www.elprocus.com/i2s-protocol/>

<https://electronics.stackexchange.com/questions/36349/which-is-easier-to-solder-tssop-or-qfn>

<https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>

https://www.st.com/content/st_com/en/stm32cubemx.html

<https://microcontrollerslab.com/keil-uvision-first-program/>

https://download.mikroe.com/documents/add-on-boards/click/audio-codec-proto-board/Audio%20Codec%20Board%20-%20PROTO_v102_Schematic.PDF

<https://download.mikroe.com/documents/add-on-boards/other/audio-and-voice/audio-codec-proto/audio-codec-proto-manual-v100.pdf>

https://www.st.com/resource/en/user_manual/um1472-discovery-kit-with-stm32f407vg-mcu-stmicroelectronics.pdf

<http://www.cs.columbia.edu/~sedwards/classes/2008/4840/Wolfson-WM8731-audio-CODEC.pdf>

https://www.keil.com/pack/doc/CMSIS_Dev/DSP/html/group__FIR.html#ga0cf008f650a75f5e2cf82d10691b64d9

<https://support.saleae.com/user-guide>

Sustav za obradu audio signala u stvarnom vremenu temeljen na ARM Cortex M4 arhitekturi procesora

Sažetak

Zbog nedostatka AD/DA konvertera na STM32F4 *Discovery board-u* čiji se STM32F4 ugradbeni računalni sustav temelji na ARM Cortex-M4, za analizu audio signala korišten je eksterni audio kodek koji sadrži potrebne AD/DA konvertere. U radu je opisano povezivanje STM32F4 sa audio kodekom koristeći I2C i I2S komunikacijske protokole. Za konfiguriranje periferija STM32F4 mikrokontrolera korišten je STM32F4CubeMX softver dok je *debugiranje* i pisanje koda za programiranje STM32F4 i Wolfson8731 koji se nalazi na kodeku, obavljeno u Keil μ Vision5 programu. Za analizu komunikacije te slanja signala i podataka između pločica, korišten je softver *Saleae*.

Ključne riječi: Ugradbeni računalni sustav, STM32F4, Audio kodek, I2C komunikacijski protokol

Real-time Audio Signal Processing System Based on ARM Cortex M4 Processor Architecture

Abstract

Due to the lack of AD/DA converters on the STM32F4 Discovery board, whose embedded system is based on the ARM Cortex-M4, an external audio codec containing the necessary AD/DA converters was used for audio signal analysis. This paper describes the connection of the STM32F4 with the audio codec using I2C and I2S communication protocols. The STM32CubeMX software was used for configuring the peripherals of the STM32F4 microcontroller, while debugging and writing code for programming the STM32F4 and Wolfson8731 (located on the codec) were done in the Keil μ Vision5 program. The Saleae software was used for analyzing the communication and transmission of signals and data between the boards.

Keywords: Embedded system, STM32F4, Audio codec, I2C communication protocol