



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

НОВИ САД

Департман за рачунарство и аутоматику

Одсек за рачунарску технику и рачунарске комуникације

ЗАВРШНИ (BACHELOR) РАД

Кандидат: Митар Танкосић

Број индекса: 10745

Тема рада: Једно решење програмске подршке за акустичке сензоре на бази XMOS вишејезгarnог процесора

Ментор рада: др Мирослав Поповић

Нови Сад, фебруар, 2015



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Митар Танкосић		
Ментор, МН:	Др Мирослав Поповић		
Наслов рада, НР:	Једно решење програмске подршке за акустичке сензоре на бази XMOS вишејезгarnог процесора, XMOS SliceKit евалуационе плоче и SmartMP3 додатне плоче		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2015		
Издавач, ИЗ:	Ауторски репримт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилога)	7/32/0/1/12/0/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	XMOS, xCORE, WiFi, RTSP, RTP, аудио сигнал, кодирање / декодирање		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	У раду је приказано једно решење аудио система заснованог на xCORE архитектури, компаније XMOS. Микроконтролер је повезан са VS1053 аудио кодер/декодер чипом, за који је реализована управљачка програмска подршка преко SPI магистрале. Помоћу VS1053 се снима и репродукује звук. Комуникација са интернетом је реализована уз помоћ XA-SK-WIFI-TIWISL модула. Микроконтролер преко локалне WIFI-мреже шаље аудио сигнал снимљен VS1053 чипом.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:	19.02.2015.		
Чланови комисије, КО:	Председник:	Др Јелена Ковачевић	
	Члан:	Др Драган Самарџија	Потпис ментора
	Члан, ментор:	Др Мирослав Поповић	



KEY WORDS DOCUMENTATION

Accession number, ANO:			
Identification number, INO:			
Document type, DT:	Monographic publication		
Type of record, TR:	Textual printed material		
Contents code, CC:	Bachelor Thesis		
Author, AU:	Mitar Tankosic		
Mentor, MN:	Miroslav Popovic, PhD		
Title, TI:	One solution software for acoustic sensors based on XMOS multicore processor on XMOS StartKit development board and SmartMP3 additional board		
Language of text, LT:	Serbian		
Language of abstract, LA:	Serbian		
Country of publication, CP:	Republic of Serbia		
Locality of publication, LP:	Vojvodina		
Publication year, PY:	2015		
Publisher, PB:	Author's reprint		
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6		
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)	7/32/0/1/12/0/0		
Scientific field, SF:	Electrical Engineering		
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems		
Subject/Key words, S/KW:	XMOS, xCORE, WiFi, RTSP, RTP, audio signal, coding / decoding		
UC			
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, N:			
Abstract, AB:	This paper describes one solution to the audio system based on xCORE architecture of XMOS company. Microcontroller is connected to VS1053 audio CODEC chip, which is implemented control software via the SPI bus. Recording and reproduction is done with VS1053. Realisation of communication over the internet is done with XA-SK-WIFI-TIWISL module. The audio signal recorded with VS1053 chip, microcontroller sends via local WiFi-network.		
Accepted by the Scientific Board on, ASB:			
Defended on, DE:	19.02.2015.		
Defended Board, DB:	President:	Jelena Kovacevic, PhD	
	Member:	Dragan Samardzija, PhD	Menthor's sign
	Member, Mentor:	Miroslav Popovic, PhD	

Zahvalnost

Zahvaljujem se Institutu RT-RK na pruženoj podršci za realizaciju rada kao i stručnom saradniku Nenadu Četiću na savetima, stručnoj pomoći, razumevanju i motivaciji tokom izrade ovog rada.

Takođe, neizmerno sam zahvalan svojoj porodici i svim ljudima oko mene koji su me podržavali tokom čitavih akademskih studija.

SADRŽAJ

1.	Uvod	7
2.	Teorijske osnove.....	9
2.1	RTSP protokol	9
2.2	RTP protokol.....	12
2.3	xCORE arhitektura.....	14
2.4	Kodovanje i dekodovanje audio signala	15
3.	Koncept rešenja	17
3.1	Elementi rešenja.....	17
4.	Programsko rešenje.....	21
4.1	Modul app_wmsn_node.....	22
4.2	Modul module_smart_mp3	23
4.3	Modul module_spi_master.....	26
4.4	Modul module_wifi_tiwisl.....	27
5.	Ispitivanja i rezultati	28
6.	Zaključak	31
7.	Literatura	32

SPISAK SLIKA

Slika 1.1 RTSR, RTP i ostali protokoli	8
Slika 2.1 RTSP veza	10
Slika 2.2 RTP protokol	13
Slika 2.3 Format RTP zaglavlja	13
Slika 2.4 xCORE arhitektura	14
Slika 2.5 ADPCM kompresija	15
Slika 3.1 StartKit ploča	18
Slika 3.2 WiFi Slice	18
Slika 3.3 SmartMP3	19
Slika 3.4 Šematski prikaz SmartMP3	19
Slika 5.1 VLC player	30
Slika 5.2 Wireshark.....	30

SPISAK TABELA

Tabela 4.1 SPI modes 26

SKRAĆENICE

MCU	- <i>Microcontroller</i> , mikrokontroler
RISC	- <i>Reduced Instruction Set Computing</i> , procesor sa smanjenim skupom instrukcija
DSP	- <i>Digital Signal Processing</i> , digitalna obrada signala ili <i>Digital Signal Processor</i> , digitalni signal procesor
MIPS	- <i>Milion Instructions Per Second</i> , milion instrukcija u sekundi
RTP	- <i>Real-Time Transport Protocol</i> , protokol prenosa u realnom vremenu
RTSP	- <i>Real-Time Streaming Protocol</i> , protokol kontrole toka podataka u realnom vremenu
IP address	- IP adresa, 32-bitna adresa dodeljena svakom računaru koji učestvuje u TCP/IP internetu
HTTP	- <i>HyperText Transfer Protocol</i> , protokol za prenos HTML stranica
TCP	- <i>Transmission Control Protocol</i> , protokol za kontrolu prenosa
UDP	- <i>User Datagram Protocol</i> , protokol korisničkih datagrama
FTP	- <i>File Transfer Protocol</i> , protokol za prenos datoteka
URL	- <i>Uniform resource Locator</i> , uniformni lokator resursa
SPI	- <i>Serial Peripheral Interface</i> , serijski sprežni sistem za periferije
IoT	- <i>Internet of Things</i> , internet stvari
PCM	- <i>Pulse Code Modulation</i> , modulacija impulsnog koda
IPv4	- <i>Internet Protocol version 4</i> , internet protokol verzija 4
WiFi	- <i>Wireless Fidelity (wireless local area network)</i> , bežična lokalna mreža

ADPCM	- <i>Adaptive Differential Pulse Code Modulation</i> , adaptivna diferencijalna PCM
MEMS	- <i>Microelectromechanical systems</i> , mikro-elektronika-mehanički sistemi
API	- <i>Application programming interface</i> , programski interfejs
SDP	- <i>Session Description protocol</i> , format za opisivanje parametara inicijalizacije striming medija

1. Uvod

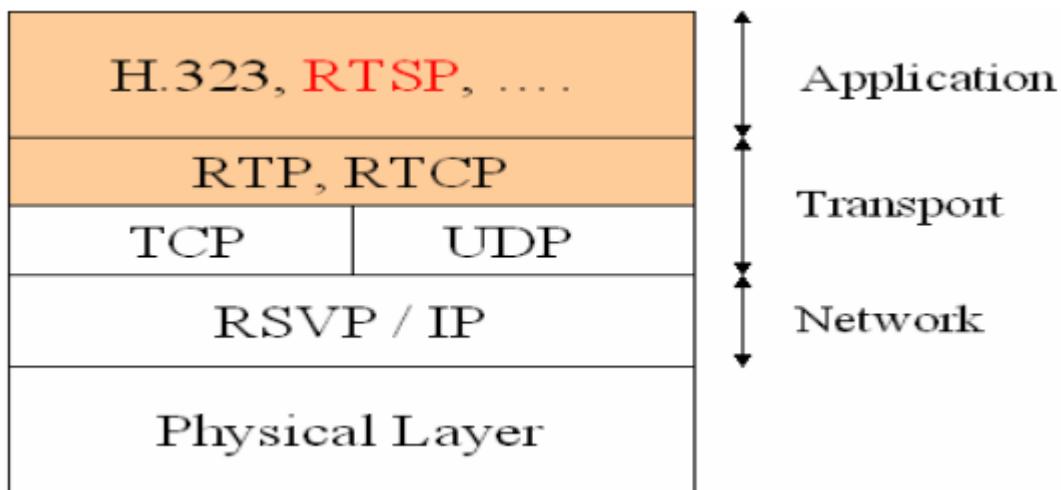
Tema ovog rada je realizacija jednog rešenja programske podrške za akustičke senzore na bazi xCORE višejezgarnog procesora kompanije XMOS. Cilj zadatka je da se ispitaju mogućnosti fizičke arhitekture i da se pokažu ubrzanja višejezgarnog xCORE mikrokontrolera pri paralelnom izvršavanju operacija.

Potreбно је realizovati programsku podršку за snimanje и пренос звука уз помоћ беžичних акустичких сензора. Архитектуру решења засновати на послуžиоцу који се налази на интернету (Cloud server). У оквиру овог рада, потребно је реализовати систем за пренос звука до удаљеног рачунара. Пренос се одвија преко послуžиоца посредника који је задушен за комуникацију са сензорним чворовима, додавање звучних токова путем RTP протокола, кодовање звучних сигнала, стварање компримованог битског тока и његово даље слање на удаљени послуžиоц. Сензорни чворови су базирани на раније развијеној IoT (*eng. Internet of Things*) платформи [1].

У данашње време већина људи за појам рачунара везује велике могућности и огромне моći обраде података. Рачунари који се користе у научним и војним истраживањима, прве се тако да буду што моћнији без обзира на njihovu цену, док комерцијални, рачунари који служе за пословне и кућне употребе акценат имају на нижој цени али зато имају и slabije моći обраде. Међутим, ово је само један део велике породице рачунара, док је други део, уграден у mnoge uređaje које svakodnevno koristima, ljudskoj svesti manje видljiv. Управљање већ машином, мобилни телефони, телевизори, аутомобилски мотори, навигација, praćenje saobraćaja, savremena medicinska dijagnostika, свемирски програми, модерни војни uređaji i robotizovani proizvodni програми данас су nezamislivi bez računarskih jedinica i programske podrške. Податак да се

95% svih proizvedenih procesora današnjice ne nalazi u klasičnim računarima nego baš skriven duboko u drugim uređajima [2].

Niz godina unazad internet tehnologija je iznenađujuće rasprostranjena, tako da je postala neophodna i neizbežna u našem životu. Poseban razvoj je izražen u multimedijalnim uslugama na internetu i mobilnim uređajima. Mrežne tehnologije servisa za prenos zvučnih i video signala kao što su RTSP, RTP, RTCP, TCP I UDP protokoli, upotrebljavaju se za poboljšanje kvaliteta usluga prenosa podataka i veoma su efikasni u tome [3].



Slika 1.1 RTSR, RTP i ostali protokoli

Realizacija uređaja je izvedena upotrebom xCORE višejezgarnog mikrokontrolera kompanije XMOS i pomoćnih podsistema koji zajedno čine jedan takozvani ugrađeni sistem (*eng. embedded system*). Cilj zadatka je da se ispitaju neke od mogućnosti primene IoT koncepta i da se pokažu prednosti višejezgarnog xCORE mikrokontrolera pri paralelnom izvršavanju zadataka.

Rad je podeljen u sedam poglavlja. U drugom poglavlju su opisane teorijske osnove: RTSP protocol, RTP protocol, xCORE arhitekture i teorijske osnove kodovanja i dekodovanja audio signala. U trećem poglavlju je predstavljen koncept rešavanja problema navedenih u zahtevima rada. U četvrtom poglavlju su izneti detalji programske realizacije na ciljnoj platformi. U petom poglavlju opisan je postupak ispitivanja i verifikacije datog rešenja. U šestom poglavlju je ukratko opisano šta je urađeno i šta se može dodatno unaprediti. Sedmo, poslednje poglavlje, sadrži spisak korišćene literature.

2. Teorijske osnove

U ovom poglavlju, izložene su teorijske osnove rada.

Najbolji način puštanja zvuka preko interneta, je strimovanje (RTP/RTSP isporuka). Za tokove uživo koji se moraju preneti u realnom vremenu, strimovanje je i jedini odgovor. Svaki način isporuke ima dobre i loše strane.

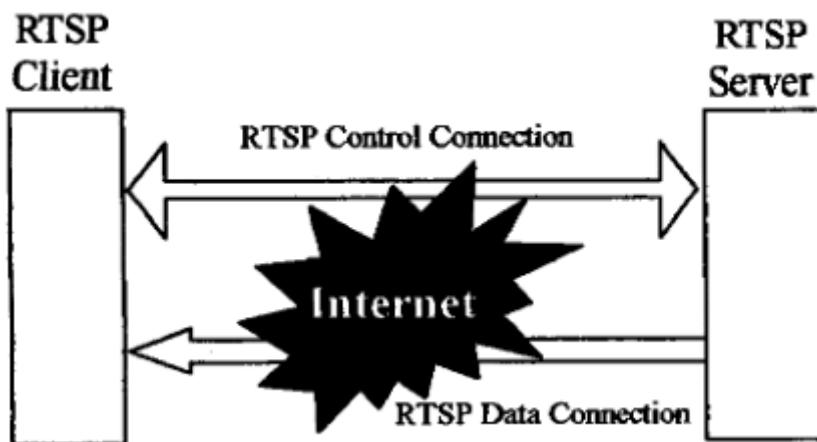
Kombinovanje tokova (strimovanje) u realnom vremenu pomoću RTP/RTSP protokola je komercijalizovano putem interneta i mobilnih servisa, jer su ti protokoli stabilniji od ostalih protokola za multimediju. Na internetu, to je uobičajena usluga strimovanja gde klijent sa web servera, obično dobija URL informaciju o serveru koji vrši strimovanje i povezuje se sa njim. Tada se podaci strimuju preko RTP/RTSP sprege. Apple-ov Quick Time, MPEG4IP i VLC su poznate aplikacije koje koriste RTP/RTSP. Iako je RTP/RTSP dobro dizajniran za striming, mreža može biti centralizovana na serveru. Tipičan sistem za strimovanje u realnom vremenu se sastoji od više klijenata i jednog servera. Klijenti zahtevaju vezu sa serverom. Nakon povezivanja server zahteva sadržaj od klijenta, onda šalje tražene striming podatke klijentu, koristeći RTP.

2.1 RTSP protokol

Real-Time Streaming Protocol – Je protokol za kontrolu nad dostavom podataka u realnom vremenu, na aplikacionom nivou. RTSP omogućava proširivi okvir kako bi omogućio kontrolu dostave podataka na zahtev u realnom vremenu, kao što su audio i video. Izvori podataka mogu biti i živi izvori i pohranjeni isečci podataka. Ovaj protokol je namenjen za kontrolu isporuke sesija sa više podataka, osigurava zahteve za odabir kanala

isporuke, kao što su UDP, TCP i UDP većem broju, i osigurava zahteve mehanizmu za odabir isporuke baze na RTP.

Tok podataka pod kontrolom RTSP može koristiti RTP, ali rad RTSP ne zavisi od transportnog mehanizma koji se inače koristi za prenos kontinualnih sadržaja. Protokol je namerno sličan u sintaksi i radu s HTTP/1.1 tako da se proširenje mehanizama za HTTP može u većini slučajeva takođe dodati u RTSP. Međutim, RTSP se razlikuje u nekoliko važnih aspekata od HTTP [4].



Slika 2.1 RTSP veza

Ovde su predstavljeni osnovni RTSP zahtevi. Podrazumevani transportni sloj je port broj 8554 i za UDP i za TCP.

OPTIONS zahtev – vraća tipove zahteva koje će server prihvati (DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN).

```
C->S: OPTIONS rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 1
      Require: implicit-play
      Proxy-Require: gzipped-messages

S->C: RTSP/1.0 200 OK
      CSeq: 1
      Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE
```

DESCRIBE zahtev – sadrži RTSP URL (rtsp://...) i tip odgovora podataka kojim se može rukovati. Ovaj odgovor uključuje opis prezentacije, obično u SDP (Session Description protocol) formatu. Osim toga opis prezentacije navodi medije kontrolisane združenim URL-om. Tipično, postoji jedan tok podataka za svaki audio i video.

```
C->S: DESCRIBE rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 2

S->C: RTSP/1.0 200 OK
      CSeq: 2
      Content-Base: rtsp://example.com/media.mp4
      Content-Type: application/sdp
      Content-Length: 460

      m=video 0 RTP/AVP 96
      a=control:streamid=0
      a=range:npt=0-7.741000
      a=length:npt=7.741000
      a=rtpmap:96 MP4V-ES/5544
      a=mimetype:string;"video/MP4V-ES"
      a=AvgBitRate:integer;304018
      a=StreamName:string;"hinted video track"
      m=audio 0 RTP/AVP 97
      a=control:streamid=1
      a=range:npt=0-7.712000
      a=length:npt=7.712000
      a=rtpmap:97 mpeg4-generic/32000/2
      a=mimetype:string;"audio/mpeg4-generic"
      a=AvgBitRate:integer;65790
      a=StreamName:string;"hinted audio track"
```

SETUP zahtev - navodi kako jedan tok podataka mora da se prenosi. Ovo mora da se uradi pre nego što se pošalje PLAY zahtev. Zahtev sadrži URL toka podataka i transportni specifikator. Ovaj specifikator obično uključuje lokalni port za prijem RTP podataka (zvuka ili filmova), a drugi RTCP podatke. Odgovor servera obično potvrđuje odabrane parametre i popunjava delove koji nedostaju, kao što su portovi servera. Svaki tok podataka mora biti konfiguriran SETUP zahtevom pre nego što se skup zahteva za puštanje može poslati.

```
C->S: SETUP rtsp://example.com/media.mp4/streamid=0 RTSP/1.0 CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8000-8001
S->C: RTSP/1.0 200 OK CSeq: 3 Transport:
      RTP/AVP;unicast;client_port=8000-8001;server_port=9000-9001 Session:
      12345678
```

PLAY zahtev – prouzrokuje da se pusti jedan ili svi tokovi podataka. PLAY zahtevi se mogu složiti slanjem više PLAY zahteva. URL može biti skup URL-ova (ako puštamo sve tokove podataka) ili jedan URL tok podataka (za puštanje samo jednog toka podataka). Opseg

može biti naveden. Ako opseg nije naveden, tok podataka se pušta od početka i ide do kraja ili ako je strim stopiran, nastavlja se sa mesta gde je zaustavljen.

```
C->S: PLAY rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 4
      Range: npt=5-20
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 4
      Session: 12345678
      RTP-Info: url=rtsp://example.com/media.mp4/streamid=0;seq=9810092;rtptime=3450012
```

PAUSE zahtev - Privremeno zaustavlja jedan ili sve tokove podataka, tako da se kasnije oni mogu nastaviti pomoću zahteva PLAY. Zahtev sadrži ukupan ili jedan URL toka podataka.

```
C->S: PAUSE rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 5
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 5
      Session: 12345678
```

TEARDOWN zahtev – se koristi da prekine sesiju. On zaustavlja sve tokove podataka I oslobođa sve podatke vezane za sesiju na serveru.

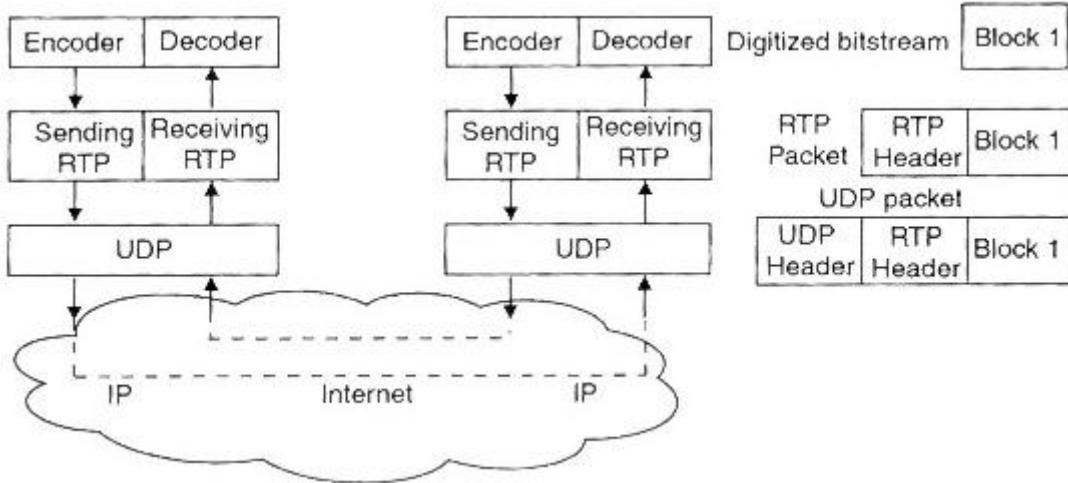
```
C->S: TEARDOWN rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 8
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 8
```

2.2 RTP protokol

Protokol koji se koristi za prenos digitalizovanih zvučnih i video signala preko IP interneta poznat je kao *protokol prenosa u realnom vremenu (Real-Time Transport*

Protocol)[5]. RTP ne sadrži mehanizme koji obezbeđuju pravovremenu isporuku; tu garanciju mora da obezbedi prenosni sistem koji se nalazi u osnovi. Umesto toga RTP nudi dve ključne mogućnosti: redni broj svakog paketa koji omogućava primaocu da otkrije isporuku van redosleda ili gubitak i vremenski žig koji omogućava primaocu da kontroliše reprodukciju. Na slici 2.2 imamo šematski prikaz koji ilustruje korišćenje RTP protokola. Audio signal je digitalizovan pomoću posebnog kodera. Tako dobijenim blokovima se najpre dodaje RTP zaglavje i pravi se RTP blok, da bi se onda formirao UDP blok i na kraju IP blok [6].



Slika 2.2 RTP protokol

Kako je RTP projektovan za prenos veoma različitih podataka u realnom vremenu, uključujući zvuk i video, RTP ne nameće jednoliku interpretaciju semantike, već svaki paket počinje fiksnim zaglavljem. Polja u tom zaglavju određuju kako se tumače ostala polja zaglavja i kako se tumače noseći podaci.

Bit offset ^[b]	0-1	2	3	4-7	8	9-15	16-31
0	Version	P	X	CC	M	PT	Sequence Number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifiers ...						
96+32×CC	Profile-specific extension header ID				Extension header length		
128+32×CC	Extension header ...						

Slika 2.3 Format RTP zaglavja

2.3 xCORE arhitektura

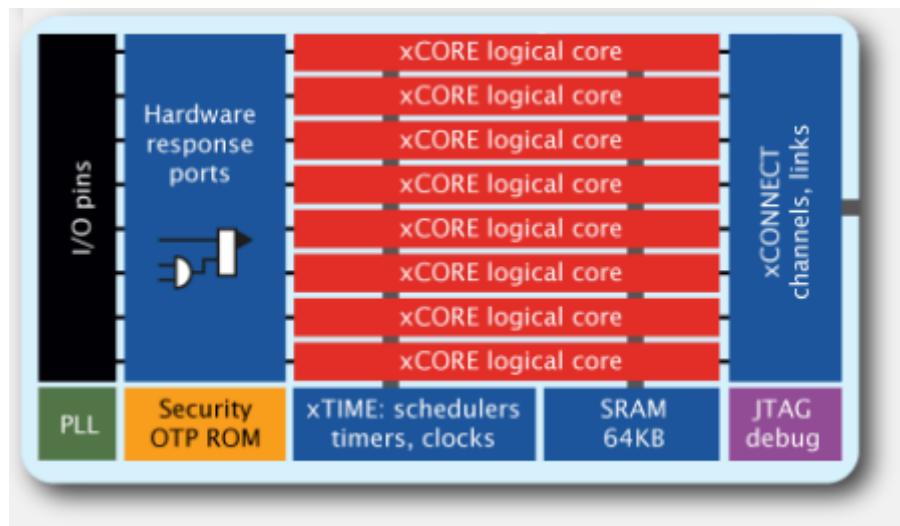
xCORE predstavlja novu klasu mikrokontrolera koja ima više procesorskih jezgara (*eng. logical processor core*), na zadatom broju pločica (*eng. tile*), izuzetno prilagodljiv ulaz / izlaz i jedinstven deterministički odziv, što ga čini izuzetno lakis za korišćenje. [6]

Za razliku od konvencionalnih mikrokontrolera xCORE višejezgarni mikrokontroler je u stanju da istovremeno izvršava višestruke zadatke u realnom vremenu. xCORE podržava programski jezik visokog nivoa XC kao i standardnu C/C++ implementaciju na 32-bitnom procesorskom jezgru koristeći RISC instrukcije.

U ovom trenutku su nam dostupni xCORE višejezgarni mikrokontroleri sa: 4, 6, 8, 10, 12, 16 i 32 jezgra na 1, 2 and 4 pločice u jedinstvenom kućištu. Svaka pločica obezbeđuje 500MIPS na 500MHz, dok se za svako jezgro garantuje 125MIPS na 500MHz procesorske snage, što čini ovaj mikrokontroler znatno moćnijim od konvencionalnih mikrokontrolera.

Jezgra koja se nalaze na istoj pločici međusobno dele memoriju i druge resurse te pločice, ali svako jezgro poseduje i svoje memorijske registre.

Svako pojedinačno jezgro izvršava jedan zadatak, dok skup ovih jezgara predstavlja procesor koji je u stanju da izvršava više zadataka istovremeno. Jezgra komuniciraju međusobno preko kanala (*eng. channel*), na čemu se zasniva xTIME tehnologija koja omogućava xCORE uređajima izuzetnu vremensku predvidljivost i odziv u realnom vremenu. Slika 2.4 prikazuje hardversku organizaciju čipa izrađenog u xCORE arhitekturi.



Slika 2.4 xCORE arhitektura

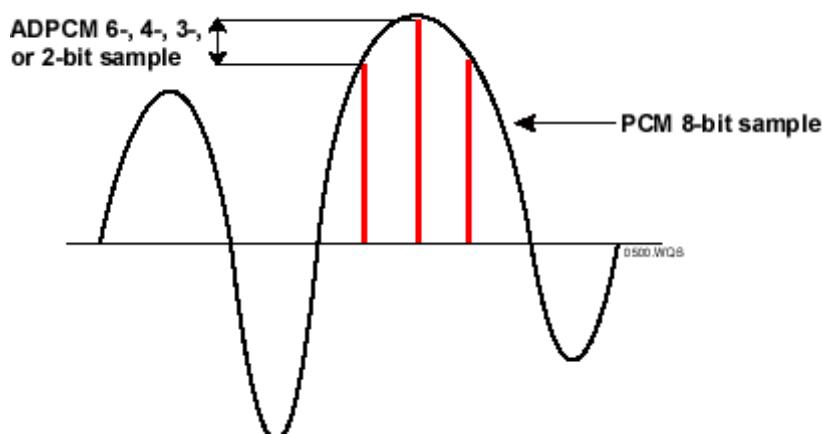
xCORE više jezgarni mikrokontroleri se mogu programirati kako u asembleru tako i u C/C++ programskom jeziku. Da bi iskoristio sve prednosti xCORE arhitekture, XMOS je

napravio jedno proširenje C programskog jezika i nazvao ga XC, koje podržava paralelne zadatke i kontrolu odziva.

2.4 Kodovanje i dekodovanje audio signala

Da bismo audio signal prenosili digitalnim sistemima prenosa, moramo ga prethodno digitalizovati. Teorema o odabiranju glasi: ako kontinualna funkcija vremena $f(t)$ ima spektar koji se nalazi u intervalu učestanosti od 0 do f_m , onda je ta funkcija u potpunosti definisana svojim trenutnim vrednostima, uzetim u ekvidistantnim tačkama na apscisi koje obrazuju niz čiji je interval $\Delta t = 1/(2f_m)$.

Telefonski govorni kanal zauzima opseg do 4kHz pa je prema teoremi odabiranja učestanost odabiranja 8kHz. Svaki odbirak (*eng. sample*) se koduje sa 8 bita što nam daje protok po telefonskom kanalu od 64kb/s. PCM (*eng. Pulse Code Modulation*) je definisan u okviru ITU-T G.711 (*eng. International Telecommunication Union*) preporuke koja predviđa dva načina kodovanja signala: A-law (EU) i μ -law (US). A-law i μ -law zakoni kompresije su nastali u trenutku kada tehnologija nije omogućavala korišćenje naprednijih algoritama. Oni obezbeđuju efektivnih 12-13 bita gde se u suštini prenosi samo osam bita po odbirku. Dalji razvoj tehničke omogućio je da se umesto PCM obrade koriste napredniji algoritmi kompresije govora. Drugi algoritam je baziran na ADPCM - adaptivna diferencijalna PCM, (*eng. Adaptive Differential Pulse Code Modulation*) i definisan je u okviru preporuke ITU-T G.726. Princip rada se zasniva na tome da se koduje samo razlika između dva odbirka i u zavisnosti od broja bita koji se koriste imamo sisteme sa 16kb/s, 24kb/s, 32kb/s i 40kb/s. Na slici 2.6 je prikazan način uzimanja odbiraka prilikom ADPCM kompresije. [8]



Slika 2.5 ADPCM kompresija

PCM radi uzimajući odbirke u diskretnim razmacima (nazivaju se brzina odabiranja). Uobičajene frekvencije su 11 kHz, 22 kHz i 44kHz. Što je veća frekvencija odabiranja, to je bolje predstavljen originalni signal i bolji je kvalitet zvuka. Svaki odbirak je realan broj sa beskonačnom rezolucijom u intervalu od +1.0 do -1.0. Zato se oni moraju čuvati kao konačno-precizni digitalni brojevi, tako da su podaci predstavljeni kao 16-bitni ili 8-bitni PCM, a obično ih zovemo 8-bitni ili 16-bitni odbirci. Kako 16-bitni podaci imaju bolju rezoluciju, digitalni talasni oblik zvuči bolje. Dok 8-bitni podaci imaju manju rezoluciju, daju lošiji zvuk ali zato zahtevaju manje prostora na disku [9].

ADPCM kao oblik kompresije, je efikasniji način čuvanja talasnog signala nego 16-bitni i 8-bitni PCM. Koristi samo 4 bita po odbirku, tako da zauzima četvrtinu memoriskog prostora na disku u odnosu na 16-bitni PCM. Međutim, kvalitet zvuka je niži. Kako Windows Sound System koristi 8/16-bitni PCM, računar mora kodovati i dekodovati ADPCM u/iz PCM, što zahteva vreme CPU-a.

ADPCM čuva razlike između vrednosti dva susedna PCM odbirka i daje neke pretpostavke koje omogućavaju smanjenje podataka. Zbog ovih pretpostavki niže frekvencije se pravilno reprodukuju, ali visoke frekvencije imaju tendenciju iskrivljenja. Distorzija se lako čuje u 11 kHz fajlovima, ali postaje sve teže uočljiva sa višim stopama odabiranja, a gotovo ju je nemoguće prepoznati u fajlovima od 44 kHz.

3. Koncept rešenja

Rešenje je realizovano tako što smo prvo fizički povezali XMOS-ovu StartKit evolucionu ploču sa WiFi Slice modulom (standardnim dodatkom XMOS ploče) i SmartMP3 dodatnom pločom, koje se povezuju pomoću SPI serijske spreve.

Zatim sledi realizacija programske podrške za kodovanje i dekodovanje audio signala (čip VS1053) kao drajvera koji će snimati i reprodukovati audio signal.

Kako je i rečeno, komunikaciju preko interneta ostvarujemo pomoću WiFi Slice modula. Pri ostvarivanju konekcije, proveravamo da li je to RTSP zahtev, koji u našem slučaju šalje VLC player sa nekog udaljenog računara, osluškivajući port 8554. Poruke se nakon toga parsiraju i šalju se odgovarajući odgovori, da bi se po konfigurisanju komunikacije i dobijanju PLAY zahteva pristupilo pokretanju RTP rutine i slanja audio signala putem RTP protokola.

Pre samog slanja audio signal se koduju ADPCM algoritmom.

3.1 Elementi rešenja

Rešenje je realizovano sa StartKit low-cost razvojnom pločom, WiFi Slice dodatkom za bežičnu komunikaciju i SmartMP3 dodatkom za obradu audio signala. StartKit ploča sadrži jedan XS1-A8DEV-64-FB217, 8-jezgarni, 32-bitni mikrokontroler brzine 500MIPS-a sa neophodnim priključcima. Akcenat je stavljen na xCORE višejezgarnu arhitekturu

mikrokontrolera proizvođača XMOS. StartKit ploča sa mikrokontrolerom prikazana je na slici 3.1.



Slika 3.1 StartKit ploča

WiFi Slice ploča za bežičnu komunikaciju, takođe razvijena od strane kompanije XMOS, prikazana je na slici 3.2. Koristi 2.4GHz WLAN modul visokih performansi, ima 802.11b/g bežičnu konekciju sa protokom do 7Mbps.



Slika 3.2 WiFi Slice

Ova ploča je bazirana na LS Research's TiWi-SL Wi-Fi modulu.

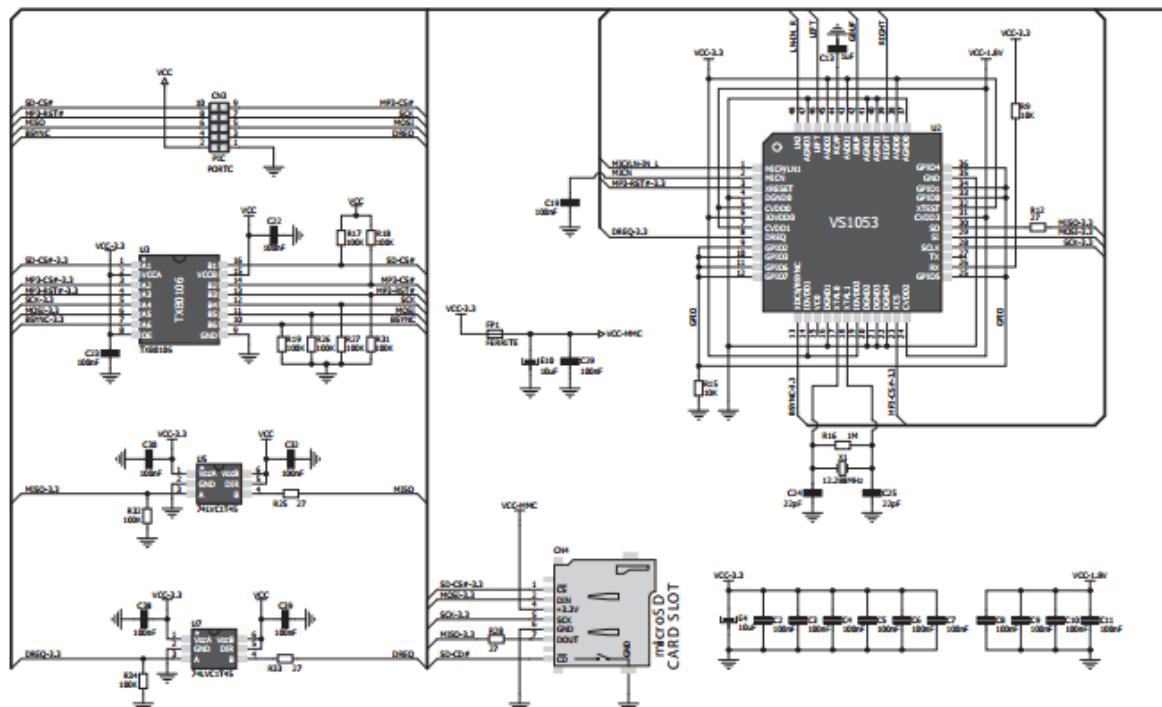
TiWi-SL je prvi 802.11b/g Wi-Fi modul sa industrijskim temperaturnim opsegom. Dizajniran je za jednostavno povezivanje najrasprostranjenijih mikrokontrolerskih platformi u industriji.

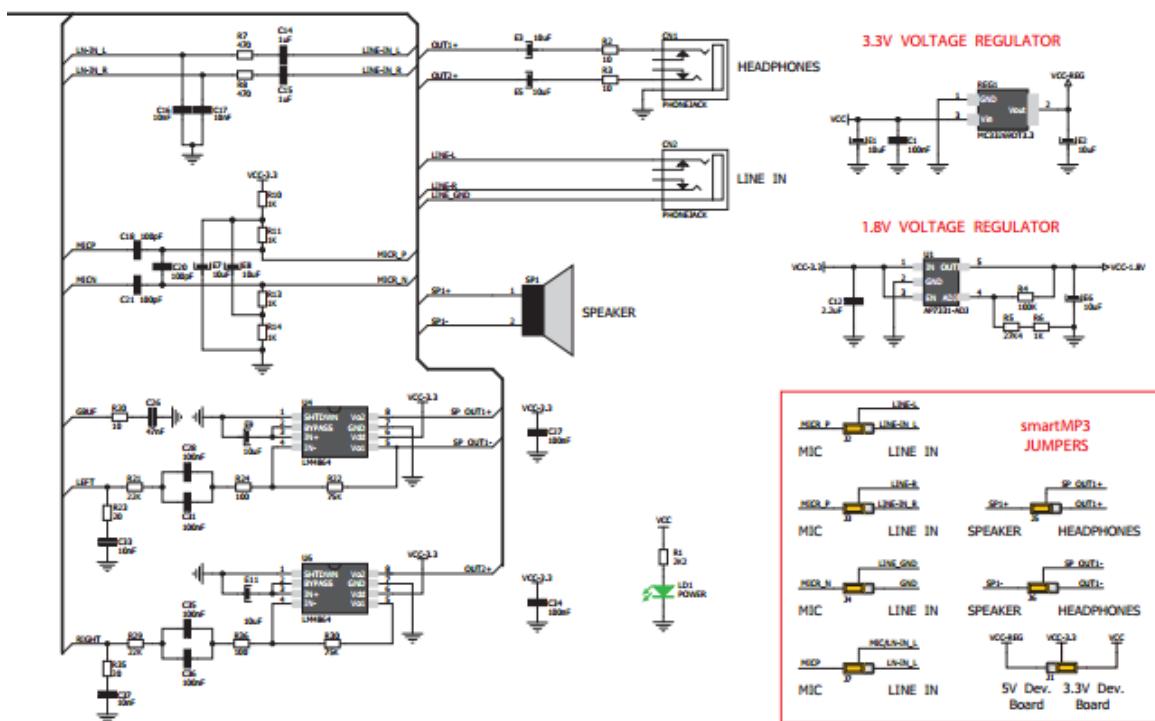
Na slici 3.3 je prikazana SmartMP3 ploča proizvođača MikroElektronika, koja se koristi za reprodukciju audio datoteka u više formata.



Slika 3.3 SmartMP3

Kako različiti proizvođači ploča koriste različite snage napona za napajanje, potrebno je i SmartMP3 pločicu prilagoditi nivou napona preko odgovarajućeg prekidača (*eng. jumper*) na ploči. Ovim prekidačem biramo da li želimo nivo napona od 3.3V ili napon od 5v. Na slici 3.4 je prikazano šta sve SmartMP3 ploča sadrži i kako to zapravo radi.





Slika 3.4 Šematski prikaz SmartMP3

VS1053 je koder / dekoder čip visokih performansi. Podržava audio streaming, kontrole visokih i niskih (bass and treble), režim rada sa niskom potrošnjom, mikrofonski ulaz, linijski izlaz, izlaz za slušalice, zvučnik i slot za Micro SD karticu. Podržava i rad sa različitim spregama kao: I2S sprega za rad sa spoljnim D/A konvertorom, SPI sprega za upravljanje čipom kao i UART za debagovanje. Pored toga sadrži i 8 GPIO pinova za najrazličitije primene.

Ploča SmartMP3 sadrži integrisano kolo VS1053 koje predstavlja koder / dekoder za audio streaming aplikacije. [9]

- Dekoder za formate
 - Ogg Vorbis / MP3 / AAC / WMA / FLAC / MIDI
- Koder za formate
 - Ogg Vorbis / IMA ADPCM / 16-bit PCM

4. Programsко rešenje

Programsko rešenje rada je podeljeno na 4 funkcionalna modula:

- app_wmsn_node
- module_smart_mp3
- module_spi_master
- module_wifi_tiwisl

i realizovano je u programskom jeziku XC i C/C++.

U nastavku su dati delovi programskog koda koji ilustruju upotrebu funkcija (*eng. tasks*) u paralelnom radu. Main funkcija je posebna i ona sadrži blokove programskog koda u okviru kojeg su date funkcije i koji se mogu izvršavati paralelno.

```
par {
    task1(5);
    task2();
}
```

Često funkcije nemaju izlaz u klasičnom smislu već sadrže beskonačnu petlju.

```
#include <platform.h>

...
int main() {
```

```

par {

    on tile[0]: task1();

    on tile[1].core[0]: task2();

    on tile[1].core[0]: task3();

}

}

```

4.1 Modul app_wmsn_node

Ovaj modul predstavlja funkcionalnu osnovu projekta. Čine ga sledeće osnovne funkcionalne jedinice:

app_wmsn_node.xc

U ovoj datoteci se nalazi inicijalizacija SPI sprege sa pripadajućim prekidnim (*eng. interrupt*) i selektabilnim (*eng. chip select*) pinovima, kao i main funkcija sa nitima (*eng. thread*).

xrtsp.xc

Predstavlja datoteku sa funkcijama vezanim za rad sa WiFi modulom, kao što su:

- void xrtsp(chanend c_wifi, chanend c_audio, chanend c_response);
 - uključuje WiFi modul
 - vrši konekciju na udaljeni server (u našem slučaju VLC player)
 - procesira RTSP događaje i poziva funkciju za slanje RTP paketa

rtsp.c

Datoteka koja sadrži sve vitalne funkcije za rad sa mrežom, kao i automat stanja (*eng. Finite-state machine*). Najbitnije su:

- void rtsp_init(chanend tcp_svr, REFERENCE_PARAM(wifi_ap_config_t, ap_config));
- void rtsp_handle_event(chanend tcp_svr, REFERENCE_PARAM(xtcp_connection_t, conn), chanend c_audio, chanend c_response);

Tu su još:

- void rtsp_recv (chanend c_tcp, chanend c_addr, xtcp_connection_t *conn){ }
- void rtsp_send(chanend c_tcp, chanend c_audio, xtcp_connection_t *conn){ }
- void rtsp_init_state(chanend c_tcp, xtcp_connection_t *conn){ }
- int parse_rtsp_request (rtsp_state_t *hs, char *data, int len){ }

wifi_tiwisl_conf.h

Datoteka sa definisanim parametrima mreže: SSID, PASSWORD, WPA2, itd.

rtp.c

Sadrži funkciju koja šalje RTP pakete.

- void send_rtp_packet(chanend tcp_svr, xtcp_connection_t *conn);

Ova funkcija se poziva nakon dobijanja RTSP komande PLAY.

4.2 Modul module_smart_mp3

Modul koji predstavlja drajver tj. upravljačku programsку podršku za čip VS1053.

vs10xx.xc

Glavna funkcionalna datoteka sa svim potrebnim funkcijama za rad sa čipom VS1053. Sadrži i inicijalizaciju SPI interfejsa za ploču sa VS1053 čipom kao i inicijalizacije kontrolnih pinova (CS, RESET, DREQ i DCS) i funkcije za uključivanje firmware-patch elemenata. Lista pripadajućih funkcija:

- void selectSCI();
- void deSelectSCI();
- void selectSDI();
- void deSelectSDI();
 - selekcja / deselekacija registara VS1053.
- void checkDREQ();
 - monitoring aktivnosti VS1053.
- void holdDeSelectSDI();
 - uslovna selekcija регистра SDI
- void hwResetVS1053();
 - hardverski reset VS1053.

- void swResetVS1053();
 - softverski reset VS1053.
- int vsInitMcuHardware();
 - inicijalizacija SPI interfejsa između VS1053 i mikrokontrolera.
- int vsInitSoftware();
- int vsTestInitSoftware();
 - inicijalno stanje pojedinih registara i jednostavan sanity check.
- void writeSci(u_int8 addr, u_int16 data);
 - upis jednog bajta u SCI registar.
- void writeVS10xxMem(u_int16 addr, u_int16 data);
- void writeVS10xxMem32(u_int16 addr, u_int32 data);
 - upis 2 ili 4 bajta podatka na datu memoriju lokaciju u VS1053.
- void writeSdiByte(u_int8 data);
 - upis jednog bajta u SDI registar.
- int writeSdiLong(u_int32 data);
 - upis 4 bajta u SDI registar.
- int writeSdi32(u_int8* data);
 - upis niza od 32 bajta u SDI registar.
- u_int16 readSci(u_int8 addr);
 - čitanje 2 bajta iz SCI registra.
- u_int16 readVS10xxMem(u_int16 addr);
 - čitanje 2-bajtne vrednosti sa zadate adrese iz memorije.
- u_int32 readVS10xxMem32(u_int16 addr);
 - čitanje 4-bajtne vrednosti sa zadate adrese iz memorije.

- `u_int32 readVS10xxMem32Counter(u_int16 addr);`
 - čitanje 2-bajtne vrednosti sa zadate adrese iz memorije na određeni način.

- `void set16(u_int8* d, u_int16 n);`
- `void set32(u_int8* d, u_int32 n);`
 - dodavanje IMA ADPCM RIFF Header-a.

- `void loadPlugin(const u_int16 *d, u_int16 len);`
 - učitavanje plug-in-a.

- `void startPcmRecording();`
 - inicijalizacija VS1053 za početak snimanja.

- `void doRecording(chanend c_audio, int blocks);`
 - snimanje upisivanjem podataka u niz.

- `void startPlayback();`
 - inicijalizacija VS1053 za početak reprodukcije.

- `void doPlayback(chanend c_audio, int blocks);`
 - čitanje podataka iz niza i prosleđivanje VS1053 u cilju reprodukcije.

vs1053_test.xc

Datoteka sa funkcijama i definicijama upotrebljavana za vreme razvoja i testiranja funkcionalnosti VS1053 čipa. Pripadajuće funkcije:

- `int buttonPressed();`
 - start / stop snimanje.

- `void sineTest();`
 - test reprodukcije sinusnog signala određene frekvencije.

- `void loop();`

- petlja u kojoj se na pritisak tastera izvrši reprodukcija audio snimka u formi niza podataka.

4.3 Modul module_spi_master

Ovaj modul obezbeđuje povezivanje StartKit razvojne ploče sa dodatnim pločama (WiFiSlice i SmartMP3) putem SPI serijske sprege. U njemu se nalazi struktura koja predstavlja SPI interfejs sa pripadajućim podešavanjima:

// SPI master					
//					
// SPI modes:					
// +-----+-----+-----+					
// Mode CPOL CPHA Supported					
// +-----+-----+-----+					
// 0 0 0 Yes					
// 1 0 1 Yes					
// 2 1 0 Yes					
// 3 1 1 Yes					
// +-----+-----+-----+					

Tabela 4.1 SPI modes

```
#define DEFAULT_SPI_CLOCK_DIV 8
#define SPI_MASTER_MODE 3
```

kao i funkcijama za rad sa SPI interfejsom:

- void spi_master_init(spi_master_interface &spi_if, int spi_clock_div);
- void spi_master_shutdown(spi_master_interface &spi_if);
- unsigned char spi_master_in_byte(spi_master_interface &spi_if);
- unsigned short spi_master_in_short(spi_master_interface &spi_if);
- unsigned int spi_master_in_word(spi_master_interface &spi_if);
- void spi_master_in_buffer(spi_master_interface &spi_if, unsigned char buffer[], int num_bytes);
- void spi_master_out_byte(spi_master_interface &spi_if, unsigned char data);
- void spi_master_out_short(spi_master_interface &spi_if, unsigned short data);
- void spi_master_out_word(spi_master_interface &spi_if, unsigned int data);

-
- void spi_master_out_buffer(spi_master_interface &spi_if, const unsigned char buffer[], int num_bytes);

4.4 Modul module_wifi_tiwisl

Modul koji obezbeđuje funkcionalnost bežične komunikacije sa internetom, tj. povezuje objekat sa internetom čineći tako koncept IoT.

Osnovna funkcionalna jedinica je:

- wifi_tiwisl_server.xc - programska sprega sa WiFi dodatkom

Sadrži funkcije za rad sa priključcima (*eng. socket*) i događajima (*eng. events*).

Najbitnija funkcija:

- void wifi_tiwisl_server(chanend c_xtcp,
REFERENCE_PARAM(spi_master_interface, tiwisl_spi),
REFERENCE_PARAM(wifi_tiwisl_ctrl_ports_t, tiwisl_ctrl));

Unutar ove funkcije se nalazi automat konačnog broja stanja koji radi u sprezi sa ostalim delovima programa.

5. Ispitivanja i rezultati

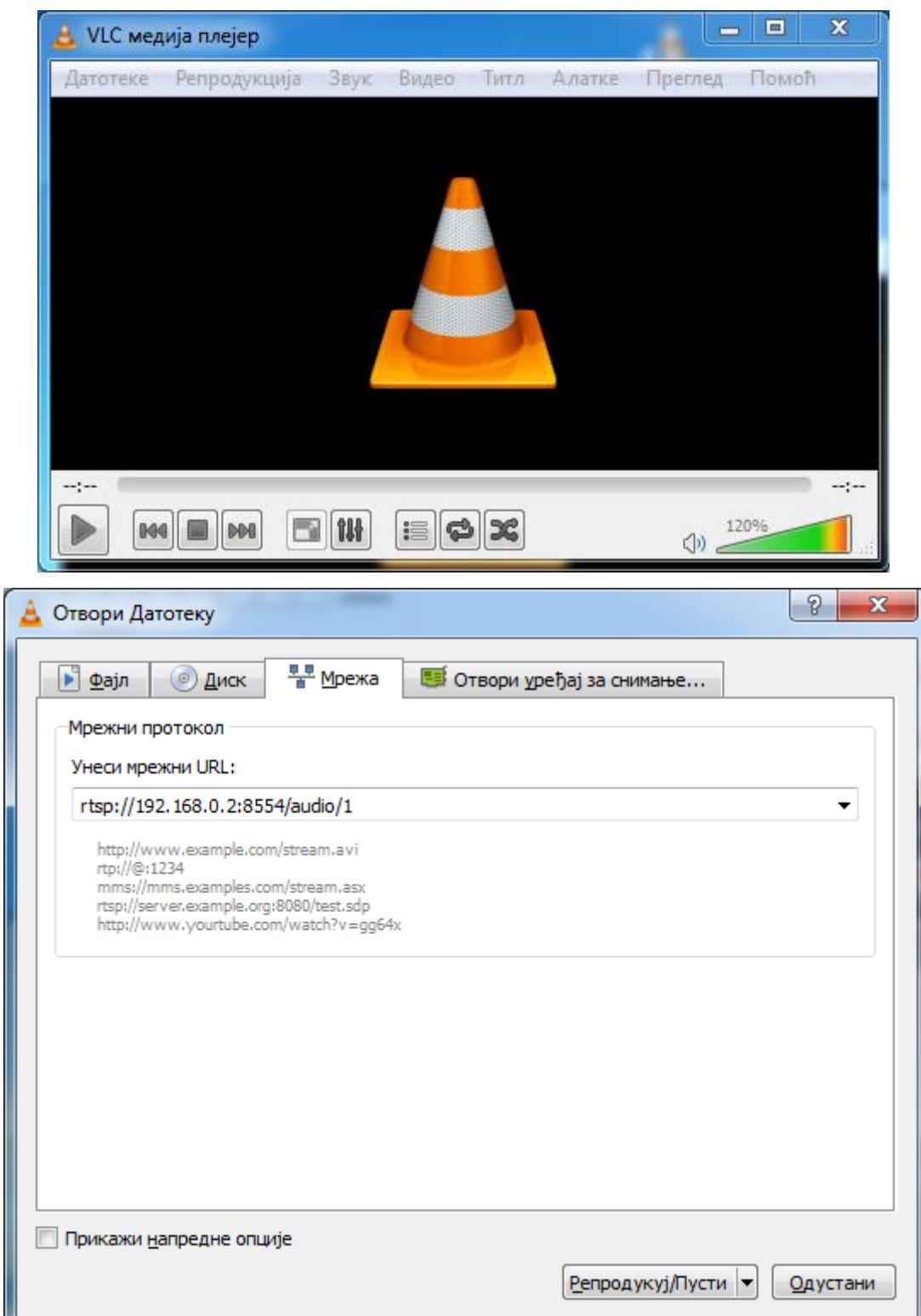
Ispitivanja i verifikacije realizovanog rešenja izvršene su na dva načina:

- ✓ ispitivanje audio izlaza i
- ✓ ispitivanje RTSP komunikacije (slanja, primanja i parsiranja traženih poruka).

Prvi deo se sastoji u tome da proizvođač audio signala realizujemo u vidu C koda, pisanog u *Microsoft Visual Studio* razvojnom okruženju. Pri tome se koristila aritmetika pokretnog zareza i reč dužine 32 bita za predstavljanje audio odbiraka. Kako xCORE arhitektura ne podržava aritmetiku pokretnog zareza i nema mogućnost skladištenja velike količine podataka, dati kod je prepravljen da bi bio pogodan za korišćenje celobrojne aritmetike i imao odbirke dužine 16 bita. Dobijeni izlazi skladišteni su u datoteke. Nakon toga, program je prilagođen xCORE arhitekturi i prebačen na ciljnu platformu. Dalje je implementiran drajver za kodovanje i dekodovanje audio signala u vidu čipa VS1053. Dodatno se vodilo računa o toma da SPI sprega između VS1053 čipa, WiFi modula i XMOS mikrokontrolera, bude realizovana tako da kasnije oba elementa mogu raditi istovremeno u različitim programskim nitima.

Pri ispitivanju drugog dela sistema korišćen je program Wireshark. Započinjanje komunikacije nastupa otvaranjem mrežnog toka sa VLC pleyera, koji je prikazan na Slici 5.1. Wireshark nam omogućuje da vizuelno pratimo komunikaciju vezanu određenu IP adresu koju želimo da pratimo. Na Slici 5.2 se redosledno vidi komunikacija između našeg poslužioca i udaljenog poslužioca sa kojeg nam stižu RTSP zahtevi. Kao što se na slici 5.1 vidi sa IP adrese 192.168.0.1 na IP adresu 192.168.0.2 stiže RTSP zahtev “OPTIONS”, nakon parsiranja naš server utvrđuje da je zahtev korektan i šalje odgovor udaljenom server. Taj

odgovor pored potvrde da je zahtev uspešno prihvaćen sadrži i opcije koje server može da prihvati (DESCRIBE, SETUP, TEARDOWN, PLAY I PAUSE).

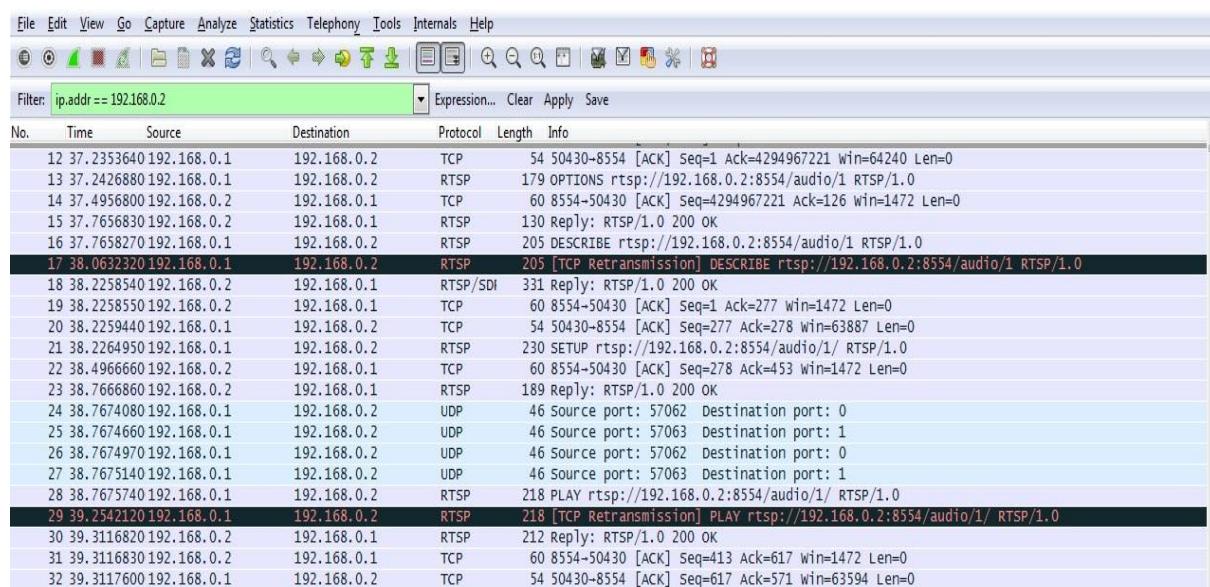


Slika 5.1 VLC player

Nakon čega dobijamo novi RTSP zahtev "DESCRIBE" koji se takođe parsira i verifikuje. Describe odgovor sadrži URL (rtsp://192.168.0.2:8554/audio) i tip podatka kojim se može rukovati (u našem slučaju audio signal). Ovaj odgovor uključuje i opis prezentacije, obično u

SDP formatu. Kao što se vidi na slici, nakon DESCRIBE odgovora, od strane udaljenog servera stiže SETUP zahtev. Nakon uspešne obrade implementirani poslužioc šalje odgovor gde navodi kako tok podataka mora da se prenosi i on mora da se pošalje pre nego što usledi PLAY zahtev. Pored URL-a toka podataka, ovaj odgovor (DESCRIBE) šalje i transportni specifikator, koji sadrži lokalni port za prijem RTP podataka.

Dobijanje zahteva PLAY i njegovo uspešno parsiranje označava da prenos toka podataka (ili više tokova podataka) koji je prethodno konfigurisan, može da počne.



The screenshot shows a Wireshark interface with the following details:

- Filter:** ip.addr == 192.168.0.2
- Expression...:** Expression...
- Apply:** Apply
- Save:** Save
- Columns:** No., Time, Source, Destination, Protocol, Length, Info
- Data:** The table lists 32 network frames. Key entries include:
 - Frame 12: TCP ACK from 192.168.0.2 to 192.168.0.1 (seq=1, Ack=4294967221, Len=0)
 - Frame 13: RTSP OPTIONS from 192.168.0.1 to 192.168.0.2 (rtsp://192.168.0.2:8554/audio/1 RTSP/1.0)
 - Frame 14: TCP ACK from 192.168.0.1 to 192.168.0.2 (seq=4294967221, Ack=126, Len=0)
 - Frame 15: RTSP 130 Reply from 192.168.0.1 to 192.168.0.2 (RTSP/1.0 200 OK)
 - Frame 16: RTSP 205 DESCRIBE from 192.168.0.1 to 192.168.0.2 (rtsp://192.168.0.2:8554/audio/1 RTSP/1.0)
 - Frame 17: RTSP 205 [TCP Retransmission] DESCRIBE from 192.168.0.1 to 192.168.0.2 (rtsp://192.168.0.2:8554/audio/1 RTSP/1.0)
 - Frame 18: RTSP/SDI 331 Reply from 192.168.0.2 to 192.168.0.1 (RTSP/1.0 200 OK)
 - Frame 19: TCP 60 8554-50430 [ACK] from 192.168.0.1 to 192.168.0.2 (seq=1, Ack=277, Len=0)
 - Frame 20: TCP 54 50430-8554 [ACK] from 192.168.0.2 to 192.168.0.1 (seq=277, Ack=278, Len=0)
 - Frame 21: RTSP 230 SETUP from 192.168.0.1 to 192.168.0.2 (rtsp://192.168.0.2:8554/audio/1/ RTSP/1.0)
 - Frame 22: TCP 60 8554-50430 [ACK] from 192.168.0.2 to 192.168.0.1 (seq=278, Ack=453, Len=0)
 - Frame 23: RTSP 189 Reply from 192.168.0.1 to 192.168.0.2 (RTSP/1.0 200 OK)
 - Frame 24: UDP 46 from 192.168.0.1 to 192.168.0.2 (Source port: 57062, Destination port: 0)
 - Frame 25: UDP 46 from 192.168.0.2 to 192.168.0.1 (Source port: 57063, Destination port: 1)
 - Frame 26: UDP 46 from 192.168.0.1 to 192.168.0.2 (Source port: 57062, Destination port: 0)
 - Frame 27: UDP 46 from 192.168.0.2 to 192.168.0.1 (Source port: 57063, Destination port: 1)
 - Frame 28: RTSP 218 PLAY from 192.168.0.1 to 192.168.0.2 (rtsp://192.168.0.2:8554/audio/1/ RTSP/1.0)
 - Frame 29: RTSP 218 [TCP Retransmission] PLAY from 192.168.0.1 to 192.168.0.2 (rtsp://192.168.0.2:8554/audio/1/ RTSP/1.0)
 - Frame 30: RTSP 212 Reply from 192.168.0.2 to 192.168.0.1 (RTSP/1.0 200 OK)
 - Frame 31: TCP 60 8554-50430 [ACK] from 192.168.0.1 to 192.168.0.2 (seq=413, Ack=617, Len=0)
 - Frame 32: TCP 54 50430-8554 [ACK] from 192.168.0.2 to 192.168.0.1 (seq=617, Ack=571, Len=63594, Len=0)

Slika 5.2 Wireshark

Na isti način, kao i prethodno praćenje RTSP komunikacije, vrši se i kontrola prenosa RTP i RTCP paketa, za odabranu IP adresu.

6. Zaključak

U ovom radu je prikazano jedno rešenje traženog audio sistema zasnovanog na xCORE višejezgarnom mikrokontroleru kompanije XMOS. Cilj zadatka je bio da se ispita mogućnosti fizičke arhitekture i da se pokaže ubrzanje višejezgarnog xCORE mikrokontrolera pri paralelnom izvršavanju, u ovom slučaju slanja audio odbiraka.

Ovakav uređaj koji se sastoji od WiFi sliceCARD-a, Smart MP3 ploče i StartKIT razvojne ploče, može se već u ovom obliku koristiti za prenos audio signala na velike udaljenosti, korišćenjem kućnih ili poslovnih WiFi sistema, što je u ovom radu i verifikovano.

Dalji razvoji i mogućnosti istraživanja:

- ✓ Ispitivanje ovog uređaja u pravim okruženjima pri lošim komunikacionim uslovima na ivici dometa ili sa nekim smetnjama
- ✓ Ispitivanje tačnog kašnjenja prouzrokovanih na uređaju i transportovanju

Ostaje neopravdano da li je dobro slati nekomprimovane podatke ili ih komprimovati pa onda slati.

7. Literatura

- [1] “Jedno rešenje IoT platforme na bazi XMOS višjezgarnog procesora”, Srđan Furman, 2014. BSc rad, FTN
- [2] “Arhitekture i algoritmi DSP-a 1”, V. Kovačević, M. Temerinac, M. Popović, N. Teslić 2004
- [3] “Enhanced distributed streaming system based on RTP/RTSP in resurgent ability”, Jooyoen Lee, Junghak Kim, Shinho Kim , Changku Lim ,Jaeil Jung; Division of Electrical and Computer Engineering Hanyang University, Hangdang-dong 17, Sungdong-ku, Seoul, 133-791 Korea
- [4] “A DAVIC Video-on-Demand System Based on the RTSP”, Changwoo Jee; School of Computer Sci., Kang G. Shin; The University of Michigan
- [5] “Povezivanje mreža; TCP/IP; principi, protokoli I arhitekture”, Douglas E. Comer, Odsek računarskih nauka, Univerzitet Purdue, West Lafayette, Indiana,
- [6] „RTP, RTCP, and RTSP – Internet protocols for Real-Time Multimedia Communication“, Arjan Durresi; Louisiana State University, Raj Jain; Nayna Networks
- [7] “XMOS Multicore Microcontrollers”, <https://www.xmos.com>, 2015.
- [8] Mr Nenad Krajnović. ETF, 2015. [Online]. Available:
http://telekomunikacije.etf.bg.ac.rs/predmeti/ot4ipt/Kompresija_govora.pdf
- [9] “Microsoft”, <http://support.microsoft.com/>, 2015