



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Немања Фимић

**Једно решење прилагођења
програмске подршке за управљање
дигиталним ТВ пријемником
заснованом на Андроид платформи**

ДИПЛОМСКИ РАД
- Основне академске студије -

Нови Сад, 2013.



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: | Немања Фимић | | |
| Ментор, МН: | др Јелена Ковачевић | | |
| Наслов рада, НР: | Једно решење прилагођења програмске подршке за управљање дигиталним ТВ пријемником заснованом на Андроид платформи | | |
| Језик публикације, ЈП: | Српски / латиница | | |
| Језик извода, ЈИ: | Српски | | |
| Земља публиковања, ЗП: | Република Србија | | |
| Уже географско подручје, УГП: | Војводина | | |
| Година, ГО: | 2013. | | |
| Издавач, ИЗ: | Ауторски репринг | | |
| Место и адреса, МА: | Нови Сад; трг Доситеја Обрадовића 6 | | |
| Физички опис рада, ФО: (поглавља/страна/цитата/табела/ слика/графика/прилога) | 7/33/0/1/14/0/0 | | |
| Научна област, НО: | Електротехника и рачунарство | | |
| Научна дисциплина, НД: | Рачунарска техника | | |
| Предметна одредница/Кључне речи, ПО: | Прилагођење, ДТВ, Андроид, Искоришћење процесора | | |
| УДК | | | |
| Чува се, ЧУ: | У библиотеци Факултета техничких наука, Нови Сад | | |
| Важна напомена, ВН: | | | |
| Извод, ИЗ: | У раду су приказани резултати прилагођења једног решења дигиталног ТВ пријемника на андроид платформи физичкој архитектури. Циљ прилагођења је максимално искоришћење могућности циљне платформе и смањење искоришћења процесора при основним операцијама везану за обраду ДТВ сигнала. Забележили смо одличне резултате, смањивши искоришћење централног процесора на свега 4% при репродукцији једног телевизијског сервиса. | | |
| Датум прихватања теме, ДП: | | | |
| Датум одбране, ДО: | | | |
| Чланови комисије, КО: | Председник: | проф. др Никола Теслић | |
| | Члан: | мр Милан Савић | Потпис ментора |
| | Члан, ментор: | др Јелна Ковачевић | |



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: | Nemanja Fimić | |
| Mentor, MN: | Jelena Kovačević, PhD | |
| Title, TI: | One Solution of DTV Software Based on Android Platform Adjustment | |
| 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: | 2013. | |
| Publisher, PB: | Author's reprint | |
| Publication place, PP: | Novi Sad, Dositeja Obradovica sq. 6 | |
| Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices) | 7/33/0/1/14/0/0 | |
| Scientific field, SF: | Electrical Engineering | |
| Scientific discipline, SD: | Computer Engineering, Engineering of Computer Based Systems | |
| Subject/Key words, S/KW: | Adjustment, DTV, Android, CPU usage | |
| UC | | |
| Holding data, HD: | The Library of Faculty of Technical Sciences, Novi Sad, Serbia | |
| Note, N: | | |
| Abstract, AB: | This paper presents the results of adjustment of DTV software, based on Android platform, to hardware architecture. Goal of the adjustment is to use platform capabilities to full extent and to minimize CPU usage in basic DTV functionality. We have achieved excellent results, bringing the CPU usage down to 4% while reproducing one DTV service. | |
| Accepted by the Scientific Board on, ASB: | | |
| Defended on, DE: | | |
| Defended Board, DB: | President: | Nikola Teslić, PhD |
| | Member: | Milan Savić, MSc |
| | Member, Mentor: | Jelena Kovačević, PhD |
| | | Menthor's sign |

Zahvalnost

Zahvaljujem se institutu RT-RK na pruženoj mogućnosti za realizaciju ovog rada.

Takođe se zahvaljujem mentoru dr Jeleni Kovačević, stručnim saradnicima Nikoli Kuzmanoviću i Milanu Saviću, kao i kolegama iz AMUSE tima na savetima i stručnoj pomoći prilikom izrade ovog rada.

Na kraju, zahvaljujem se svojoj porodici na pruženoj podršci tokom mog školovanja.

SADRŽAJ

| | | |
|-------|--|----|
| 1. | Uvod..... | 1 |
| 2. | Teorijske osnove | 3 |
| 2.1 | Android..... | 3 |
| 2.2 | Digitalna televizija | 5 |
| 2.3 | Digitalni TV prijemnik | 6 |
| 2.4 | DTV prijemnik kao potrošač električne energije | 8 |
| 2.5 | Opis ciljne platforme..... | 9 |
| 3. | Koncept rešenja..... | 10 |
| 3.1 | Srednji sloj - Comedia..... | 10 |
| 3.1.1 | Prilagodni sloj – CHAL | 11 |
| 3.2 | Mogućnosti platforme sa više frekvencijskih birača..... | 12 |
| 3.3 | Ciljevi prilagođenja u pogledu smanjenja zahteva fizičke arhitekture | 12 |
| 3.4 | Analiza polaznog rešenja i sprovedene promene | 13 |
| 4. | Programsko rešenje..... | 15 |
| 4.1 | Realizacija TDAL_TS modula na personalnom računaru | 15 |
| 4.1.1 | Reprodukcijska DTV servisa korišćenjem TDAL_TS modula..... | 18 |
| 4.1.2 | Ostali načini korišćenja TDAL_TS modula | 18 |
| 4.2 | Prilagođenje TDAL_TS modula fizičkoj arhitekturi | 20 |
| 5. | Rezultati | 22 |
| 6. | Zaključak | 24 |
| 7. | Literatura..... | 25 |

SPISAK SLIKA

| | |
|--|----|
| Slika 2.1 Arhitektura Android steka | 3 |
| Slika 2.2 Arhitektura Android steka sa DTV proširenjima..... | 5 |
| Slika 2.3 Arhitektura DTV prijemnika..... | 6 |
| Slika 2.4 Programska podrška DTV prijemnika | 7 |
| Slika 2.5 Najznačajniji srednji slojevi..... | 8 |
| Slika 2.6 Količina električne energije potrebna za napajanje uređaja u domaćinstvu | 8 |
| Slika 2.7 BCM97435 VMS | 9 |
| Slika 3.1 Arhitektura srednjeg sloja..... | 11 |
| Slika 3.2 Jedan scenario iskorišćenja više frekvencijskih birača | 12 |
| Slika 3.3 Prikaz istovremene reprodukcije i snimanja DTV servisa u polaznom rešenju ... | 14 |
| Slika 3.4 Prikaz istovremene reprodukcije i snimanja DTV servisa nakon prilagođenja | 14 |
| Slika 4.1 Reprodukcija uživo dtv servisa korišćenjem TDAL_TS modula | 18 |
| Slika 4.2 Primer snimanja jednog i odložene reprodukcije drugog dtv servisa | 19 |
| Slika 4.3 Dijagram sekvenca snimanja DTV servisa korišćenjem TDAL_TS modula | 20 |

SPISAK TABELA

Tabela 5.1 Iskorišćenje centralnog procesora u nekoliko različitih scenarija.....23

SKRAĆENICE

| | |
|------------|--|
| DTV | - <i>Digital Television</i> , Digitalna televizija |
| CPU | - <i>Central Processor Unit</i> , Centralni procesor |
| OS | - Operativni sistem |
| EPG | - <i>Electronic Program Guide</i> , Elektronski programski vodič |
| PVR | - <i>Personal Video Recorder</i> , Lični video snimač |
| PID | - <i>Packet Identifier</i> , Identifikator paketa |

1. Uvod

U radu je prikazano jedno rešenje prilagođenja programske podrške za kontrolu DTV prijemnika zasnovanog na Android platformi. Rešenje je prilagođeno za BCM97435 VMS platformu. Kako je digitalna televizija u velikom razvoju poslednjih godina, mogućnosti i zahtevi su u porastu. Pored reprodukcije TV servisa, što predstavlja osnovnu funkcionalnost DTV prijemnika, potrebno je realizovati i naprednije i zahtevnije DTV funkcionalnosti. Pomenute naprednije funkcionalnosti su:

- elektronski programski vodič (engl. Electronic Program Guide – EPG) sa zakazivanjem snimanja (engl. Scheduler and Reminder)
- lični video snimač (engl. Personal Video Recorder – PVR)
- deljenje multimedijalnih sadržaja u lokalnoj mreži
- prosleđivanje televizijskog signala na druge uređaje (tablet računare, mobilne telefone i td.)

Kako je programska podrška našeg DTV prijamnika zasnovana na Android platformi, dodatna mogućnost DTV prijemnika je izvršavanje mnogobrojnih Android aplikacija dostupnih za preuzimanje (engl. Google Play).

Kako bi sve pomenute mogućnosti DTV prijemnika funkcionišale pouzdano, pri realizaciji je vođeno računa o optimalnom korišćenju sistemskih resursa, koji su ograničeni na namenskom sistemu kao što je DTV prijemnik. Za polaznu tačku je uzeto već postojeće rešenje programske podrške zasnovane na Android platformi [1]. Navedeno rešenje je prilagođeno ciljnoj platformi iskorišćenjem ubrzanja fizičke arhitekture gde je to bilo moguće. Rasterećivanje centralnog procesora, koji predstavlja jedan od najznačajnijih resursa sistema, je omogućilo realizaciju naprednijih DTV funkcionalnosti poput mozaik režima rada, ličnog video snimača, elektronskog programskog vodiča sa zakazivanjem snimanja, prosleđivanja DTV signala na

prenosive uređaje. Takođe, manje zauzeće centralnog procesora sa sobom povlači i manju potrošnju električne energije, a samim tim i manje zagrevanje uređaja, što garantuje duži životni vek uređaja, kao i pouzdanu i kvalitetnu uslugu.

Rad je sačinjen od 7 poglavlja. U prvom poglavlju je predstavljen cilj rada, dok drugo sadrži potrebne informacije da bi se razumela suština problema i da bi se stekla generalna slika o sistemu. Treće poglavlje opisuje koncept rešenja, a četvrto detaljnije predstavlja određene delove rešenja. Peto poglavlje sadrži pregled postignutih rezultata, a u šestom poglavlju je predstavljen značaj postignutih rezultata i mogućnosti daljeg unapređenja. Poslednje, 7 poglavlje, daje pregled korišćene literature.

2. Teorijske osnove

U ovom poglavlju su date teorijske osnove potrebne za bolje razumevanje problema kao i celog sistema.

2.1 Android

Android je platforma pre svega namenjen mobilnim telefonima i tablet računarima. Za razliku od ostalih, Android je otvorenog koda što omogućava korisniku određeni stepen slobode u prilagođenju svojim potrebama. Android aplikacije su nezavisne od fizičke arhitekture, odnosno mogu se izvršavati na raznim platformama. Sve ovo čini Android jednim od najpopularnijih platformi za mobilne uređaje pre svega, ali u poslednje vreme i za druge uređaje kao na primer DTV prijemnike.



Slika 2.1 Arhitektura Android steka

Na slici iznad je prikazana arhitektura Android platforme. Kao što vidimo Android stek možemo podeliti u različite slojeve i to su:

- operativni sistem
- srednji sloj (engl. middleware)
- aplikativni sloj

Ceo Android operativni sistem je zasnovan na linux 2.6 jezgru. Linux jezgro je zaduženo za komunikaciju sa fizičkom arhitekturom i u ovom sloju se nalaze svi potrebni rukovaoci elementima fizičke arhitekture. Ovaj sloj u isto vreme deluje i kao apstrakcioni sloj između fizičke arhitekture i ostalih slojeva programske podrške. Linux jezgro je zaduženo i za upravljanje memorijom, upravljanje procesima, mrežne operacije, sigurnosna podešavanja i td.

Sledeći sloj predstavljaju nativne biblioteke. Neke od osnovnih biblioteka su:

- Sistemska C biblioteka – BSD izvedena implementacija standardne C sistemske biblioteke, podešene za namenske Linux bazirane uređaje.
- Medija biblioteka – podrška za reprodukciju i snimanje mnogih popularnih audio i video formata, kao i statičnih slika, uključujući i MPEG4, X.264, MP3, AAC, AMP, JPG, PNG.
- Upravljač površine – upravlja pristupu podsistemu prikaza.
- Biblioteke Web pretraživača – moderan web pretraživač.
- SGL – osnovna 2D grafička podrška.
- 3D biblioteke – zasnovane na OpenGL EC 1.0 API-ju. Biblioteke koriste 3D ubrzanje fizičke arhitekture gde je dostupno ili veoma optimizovanu 3D programsku podršku.
- Slobodan tip – bitmape i vektore iscrtavanja (eng. Free types - bitmap and vector drawing).
- SQLite – moćan i lak alat za relacione baze podataka dostupne za sve aplikacije.

Pored nativnih biblioteka Android obuhvata i skup osnovnih biblioteka koje sadrže najveći deo funkcionalnosti sadržan u bibliotekama Java programskog jezika.

Android aplikacije se izvršavaju na Dalvik virtualnoj mašini. Dalvik virtualna mašina je tip Java virtualne mašine optimizovana kako bi trošila manje procesorskog vremena i memorije. Razlika je i što se na Dalvik virtualnoj mašini izvršavaju fajlovi (.dex) formata, a ne (.class).

Aplikativni sloj se sastoji od aplikacijskog okvira (engl. Application Framework) i samih aplikacija. Aplikacijski okvir sadrži programe koji upravljuju osnovnim resursima, dok aplikacije direktno koriste ove programe.

Za realizaciju DTV prijemnika zasnovanog na Android platformi potrebno je dodatno proširiti Android stek [2] sa proširenjima namenjenim upravljanju DTV resursima kao što je to prikazano na slici 2.2.



Slika 2.2 Arhitektura Android steka sa DTV proširenjima

2.2 Digitalna televizija

Digitalna televizija predstavlja prenos audio i video zapisa, kao i dodatnih informacija u digitalnom format. Prednosti digitalne televizije u odnosu na analognu su pre svega kvalitetniji zvuk i slika. Takođe, digitalna televizija omogućuje prenos više TV servisa na jednoj frekvenciji i uopšte veću količinu informacija. Digitalni TV prijemnici uglavnom poseduju i pristup internet, kao i mogućnost izvršavanja mnogobrojnih aplikacija pisanih u Javi, HTML-u i drugim programskim jezicima, što dovodi do postojanja interaktivne televizije. Takođe, kao posledica veće količine informacija koje se prenose tokom podataka u digitalnoj televiziji gledalac je u

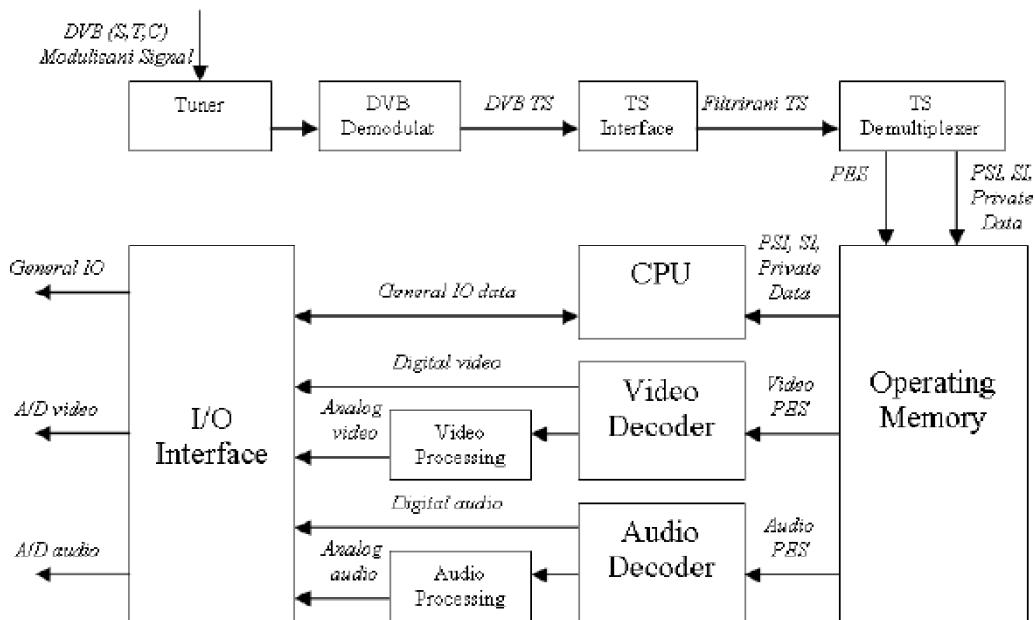
mogućnosti da bira između različitih video ili audio zapisa, jezika prevoda na jednom TV servisu, kao i da koristi TV aplikacije poput programskog vodiča ili ličnog video snimača.

Najznačajniji standardi digitalne televizije su:

- DVB – grupa standarda koja se koristi u Evropi ali i u većem delu sveta
- ATSC – standardi koji se primenjuju u digitalnom TV prenosu zemaljskim vezama u SAD, Kanadi, Meksiku, Južnoj Koreji
- OCAP – standard koji se koristi u SAD u kablovskoj digitalnoj televiziji
- ISDB – grupa DTV standarda nastala u Japanu

2.3 Digitalni TV prijemnik

Uobičajena arhitektura DTV prijemnika je prikazana na slici 2.3.

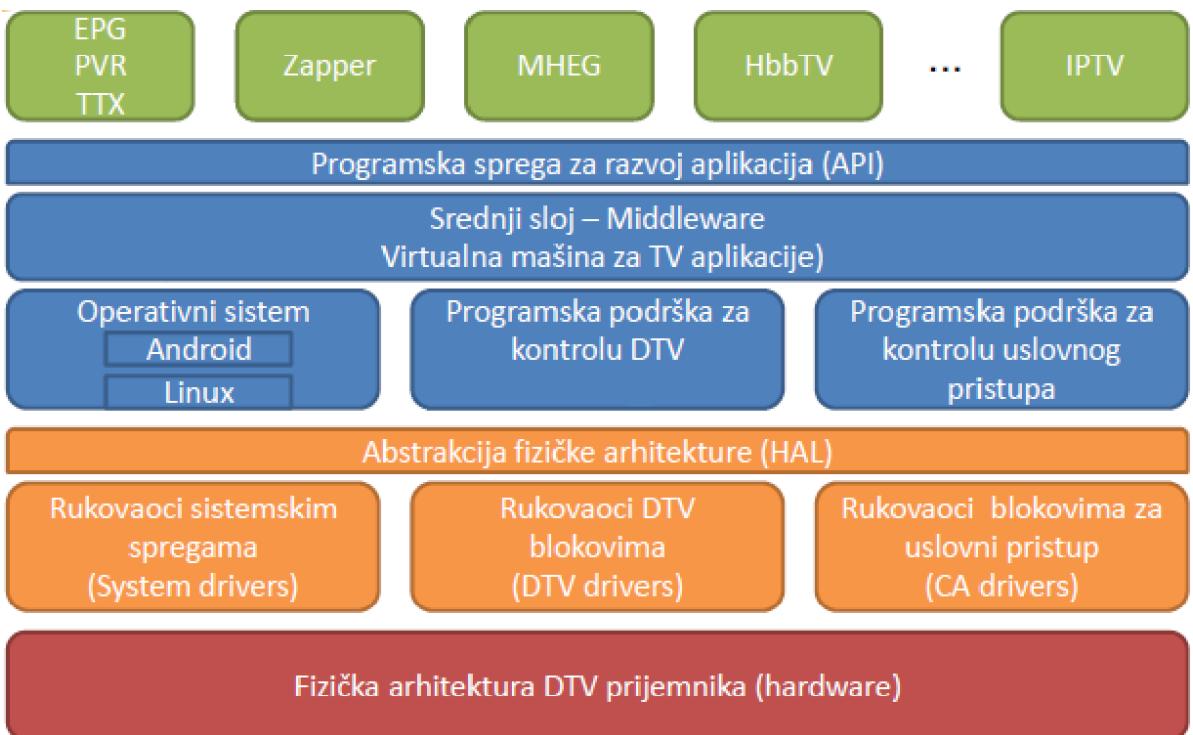


Slika 2.3 Arhitektura DTV prijemnika

Rad svih blokova je kontrolisan i koordinisan od strane centralnog procesora. Modulisani signal stiže u frekvencijski birač čiji je zadatak da izdvoji signal emitovan na frekvenciji zadatoj od strane centralnog procesora. Signal se zatim prosleđuje u odgovarajući demodulator, na osnovu tipa signala (satelitski, kablovski, zemaljski), na čijem izlazu se dobija digitalni signal. Frekvencijski birač i demodulator zajedno čine mrežni sprežni modul (engl. Network interface module – NIM) i često se nalaze u zajedničkom kućištu. Zatim se vrši demultipleksiranje transportnog toka podataka gde se prvo izdvajaju paketi koji sadrže PID-ove zadate od strane

centralnog procesora, a zatim se organizuju u elementarne tokove podataka i prosleđuju audio ili video dekoderu ukoliko je reč o audio ili video zapisima, ili centralnom procesoru ukoliko je reč o tabelama.

Što se tiče programske podrške za DTV prijemnike, uobičajni izgled programskog steka je dat na slici 2.4.



Slika 2.4 Programska podrška DTV prijemnika

Najvažniji deo programske podrške, koji sadrži gotovo kompletну DTV funkcionalnost i omogućava izvršavanje aplikacija, je srednji sloj (engl. Middleware). Uloga srednjeg sloja je da realizuje najvažnije operacije kao što su raščlanjavanje DVB podataka, kontrola pristupa i organizacija servisa, prikupljanje podataka o programima (engl. Event Information Table – EIT), podrška za snimanje, kontrola podsetnika, dekodovanje multimedijalnih sadržaja i drugi, kao i da kontroliše fizičku arhitekturu kroz komunikaciju sa nižim slojevima i da obezbedi potrebne programske sprege ka višim programskim slojevima. Važna osobina srednjeg sloja je da on apstrahuje funkcionalnosti DTV uređaja, fizičke arhitekture i operativnog sistema i na taj način obezbeđuje da proizvođač aplikativne DTV programske podrške ne mora da bude upoznat sa specifičnostima fizičke arhitekture. Drugim rečima srednji sloj formira virtualnu mašinu na kojoj se izvršava aplikativni deo programske podrške. Ovo takođe garantuje identično izvršavanje iste aplikacije na različitim fizičkim arhitekturama ukoliko se koristi isti srednji sloj.

Neki od najznačajnijih srednjih slojeva su:

- DVB-MHP
- Open TV

- MediaHighway
- Cabot
- IWEDIA

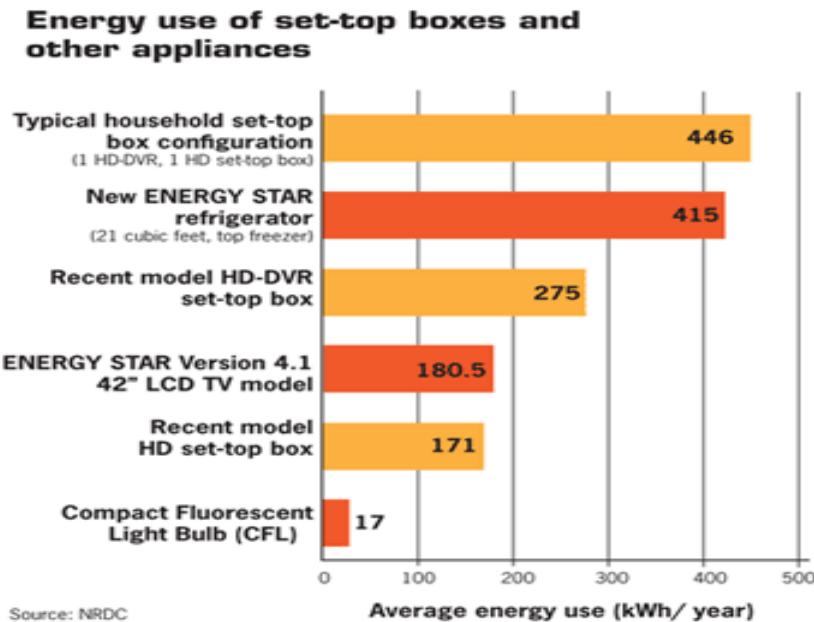


Slika 2.5 Najznačajniji srednji slojevi

U ovom radu korišćen je srednji sloj Comedia 3.0 kompanije IWEDIA.

2.4 DTV prijemnik kao potrošač električne energije

DTV prijemnik predstavlja jedan od glavnih potrošača električne energije u domaćinstvu što se može videti na slici 2.6.



Slika 2.6 Količina električne energije potrebna za napajanje uređaja u domaćinstvu

Ovolika količina električne energije je potrebna za napajanje DTV prijemnika prvenstveno zbog činjenice da DTV prijemnik predstavlja uređaj koji je stalno uključen, čak i kada ga niko ne koristi.

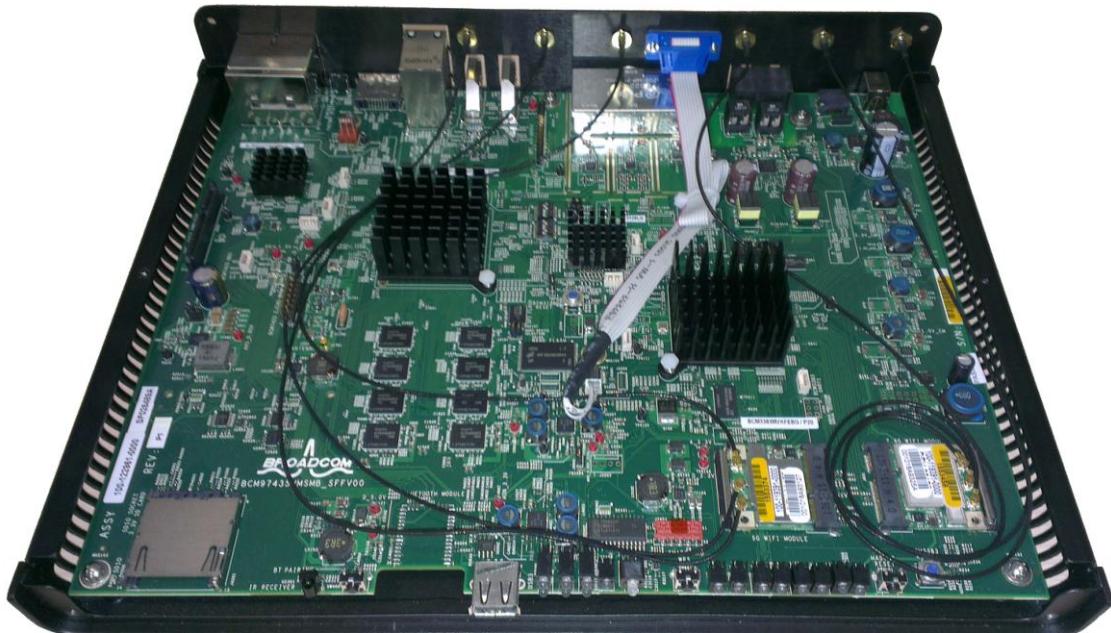
U cilju smanjenja potrebne električne energije za napajanje DTV prijemnika, uvedeni su različiti režimi rada uređaja, pa tako uvodimo stanje pripravnosti. U suštini DTV prijemnik je u

stanju pripravnosti kada je zauzeće centralnog procesora ispod 5% od ukupnog vremena. Takva situacija je najčešće kada niko ne koristi DTV prijemnik i tada je dovoljno da procesor obavlja minimum operacija. U ovom režima rada potrošnja električne energije je za oko 25% manja nego u normalnom režimu rada, što je na globalnom nivou količina električne energije proizvedena u 7 elektrana [3].

Takođe, veća količina električne energije konzumirana od strane DTV prijemnika izaziva i veće zagrevanje integrisanih kola u uređaju što negativno utiče na životni vek uređaja, kao i na pouzdanost sistema [4].

2.5 Opis ciljne platform

Za realizaciju je korišćena platforma BCM97435 VMS. Platforma poseduje 8 frekvencijskih birača, dvojezgarni procesor sa taktom 1,5 GHz, 2 GB radne memorije i 320 GB stalne memorije, 4 video i 6 audio dekodera, 3 kartice za bežični prenos podataka po 802.11 standardu.



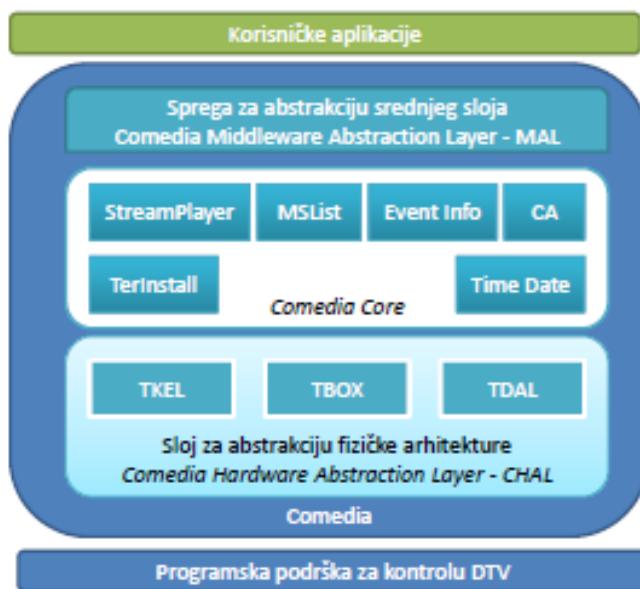
Slika 2.7 BCM97435 VMS

3. Koncept rešenja

Cilj prilagođenja programske podrške je da se smanje zahtevi fizičke arhitekture, da se poboljšaju performanse sistema i da se programska podrška prilagodi ciljnoj platformi, odnosno da se iskoriste sve mogućnosti fizičke arhitekture.

3.1 Srednji sloj - Comedia

Za bolje razumevanje koncepta rešenja potrebno je prvo upoznati se sa arhitekturom srednjeg sloja koja je prikazana na slici 3.1. Uočavamo tri celine, a to su prilagodni deo (engl. Comedia Hardware Abstraction Layer – CHAL), jezgro (engl. Comedi Core) i sprežni deo prema višim slojevima, odnosno aplikacijama (Middleware Abstraction Layer – MAL). Prilagodni sloj sadrži funkcionalnost koja je direktno vezana za fizičku arhitekturu, dok jezgro sadrži većinu logike i kontrolne funkcije. Ipak, određeni deo logike je moguće izmestiti iznad srednjeg sloja zahvaljujući sprežnom delu srednjeg sloja koji omogućava potpuno kontrolisanje srednjeg sloja.



Slika 3.1 Arhitektura srednjeg sloja

3.1.1 Prilagodni sloj – CHAL

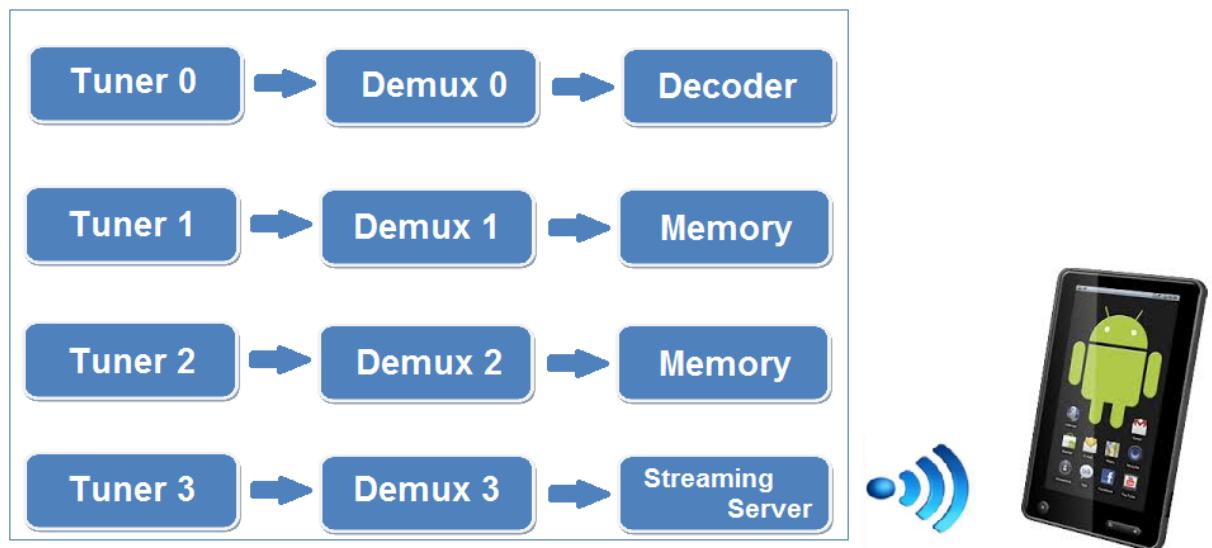
Prilagodni sloj se sastoji od tri modula:

- TKEL – apstrahuje funkcionalnost operativnog sistema
- TBOX - deklariše pomoćne funkcije i makroe i služi u svrhu otklanjanja nepravilnosti u radu programske podrške
- TDAL - sloj za apstrakciju rukovalaca (engl. drivers) i sastoji se od mnoštva modula. Svaki blok fizičke arhitekture DTV prijemnika poseduje odgovarajući TDAL rukovalac. Najvažniji moduli su:
 - TDAL DMD – upravlja fizičkim mrežnim sprežnim modulom - NIM (birač kanala (engl. tuner) i demodulator)
 - TDAL TS - upravlja distribucijom prenosnog toka od izvođačnog bloka (engl. producer) do potrošačkog bloka (engl. consumer)
 - TDAL DMX - upravlja demultiplexerom za razdvajanje audio/video komponenti, PES paketa i filtriranje SI sekcija
 - TDAL AV - kontroliše postupak dekodovanja video i audio PES u okviru DTV prijemnika (sa kontrolom fizičkih video i audio dekoder blokova)
 - TDAL DISP - upravlja prikazom slike na video izlazu. Jedna od osnovnih uloga DISP modula je upravljanje grafičkim slojevima (engl. layers) i funkcijama za povezivanje (miksovanje) slojeva (engl. blenders)
 - TDAL OUTPUT - kontroliše izlazni podsistem za sliku i zvuk u okviru DTV prijemnika

3.2 Mogućnosti platforme sa više frekvencijskih birača

Kako je polazno rešenje bilo namenjeno platformi sa jednim frekvencijskim biračem, a ciljna platforma sadrži 8 frekvencijskih birača, dodata je podrška za veći broj frekvencijskih birača. Kao posledica većeg broja frekvencijskih birača javljaju se i nove mogućnosti platforme, pa tako možemo istovremeno uživo da reprodukujemo jedan TV servis na DTV prijemniku, da snimamo nekoliko drugih i da prosleđujemo elementarne tokove podataka na prenosive uređaje (engl. streaming). Takodje moguće je i uživo reprodukovati više različitih TV servisa na DTV prijemniku u mozaik režimu rada. Neke od ovih mogućnosti su već bile realizovane u polaznom rešenju u određenoj meri, poput reprodukcije ili snimanja DTV servisa, dok je za ostale, kao na primer prosleđivanje elementarnih tokova na prenosive uređaje ili mozaik režim rada, dodata kompletна programska podrška.

Pored toga što otvara nove mogućnosti, postojanje više frekvencijskih birača zahteva i dodatnu logiku radi optimalnog iskorišćenja svih sistemskih resursa. Naime, u takvoj situaciji nije dovoljna jedna ruta koja povezuje frekvencijski birač i demultiplexer, kao što je slučaj kod platforme sa jednim frekvencijskim biračem. U cilju optimalnog korišćenja rute se prave dinamički kad za tim ima potrebe. Kontrola ruta je zadatak TDAL_TS modula o kojem će biti više reči u narednom poglavlju.



Slika 3.2 Jedan scenario iskorišćenja više frekvencijskih birača

3.3 Ciljevi prilagođenja u pogledu smanjenja zahteva fizičke arhitekture

U pogledu smanjenja zahteva fizičke arhitekture najviše pažnje je posvećeno smanjenju zauzeća centralnog procesora i optimalnom iskorišćenju raspoložive radne memorije. S obzirom na to da je DTV prijemnik baziran na Android operativnom sistemu, koji u situacijama kada nema

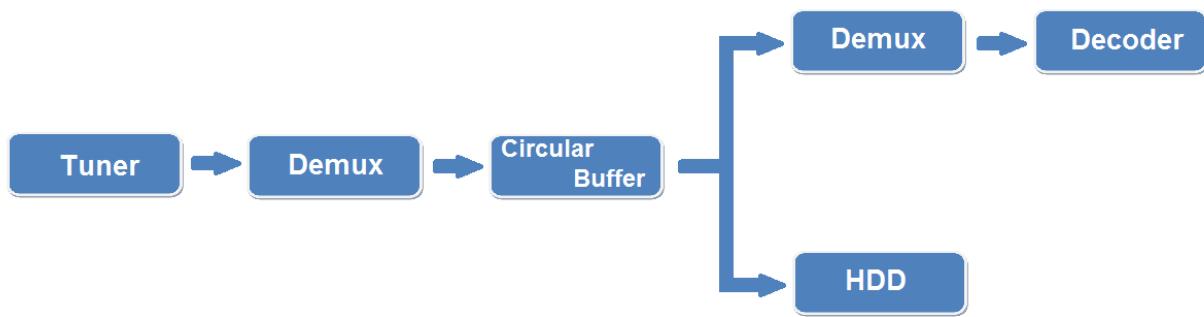
dovoljno radne memorije za svoje potrebe ubija druge procese koji zauzimaju najviše memorije, jasno je da je važno uštedeti radnu memoriju gde god je to moguće. Promene i dodaci srednjem sloju koji imaju najviše uticaja na uštedu procesorskog vremena i radne memorije su sprovedeni u prilagodnom sloju, pa će u daljem tekstu te promene biti detaljnije razmotrene.

3.4 Analiza polaznog rešenja i sprovedene promene

Polazno rešenje programske podrške je usklađeno sa platformom koja ne podržava višestruki pristup toku podataka sa frekvencijskog birača. To znači da ukoliko pristupamo toku podataka radi dobijanja elementarnih tokova jednog DTV servisa koje hocemo da reprodukujemo uživo ceo tok podataka je zaključan za ostale pristupe, odnosno ne možemo iste elementarne tokove da zapisujemo u stalnu memoriju. Ovaj problem je prevaziđen uvođenjem cirkularnih bafera u koje se smeštaju izdvojeni podaci iz toka podataka pre nego što se pošalju na dekodovanje. Pošto je cirkularnom baferu omogućen višestruki pristup, na ovaj način možemo istovremeno podatke slati na dekodovanje u cilju reprodukcije uživo i zapisivati u stalnu memoriju.

Dakle, problem je rešen, ali smo dodatno opteretili procesor i radnu memoriju. Iako veličina memorije potrebna za cirkularni bafer iznosi do 10 Mb, što i ne deluje puno, kada se prevede na našu ciljnu platformu gde ima 8 frekvencijskih birača, odnosno potrebno je 8 cirkularnih bafera, dolazi se do značajne količine zauzete memorije. S obzirom na to da se memorija potrebna za cirkularne bafera zauzima na inicijalizaciji sistema, odnosno da je zauzeta bez obzira na to da li se baferi zaista koriste ili ne, vidimo da u startu gubimo određenu količinu radne memorije.

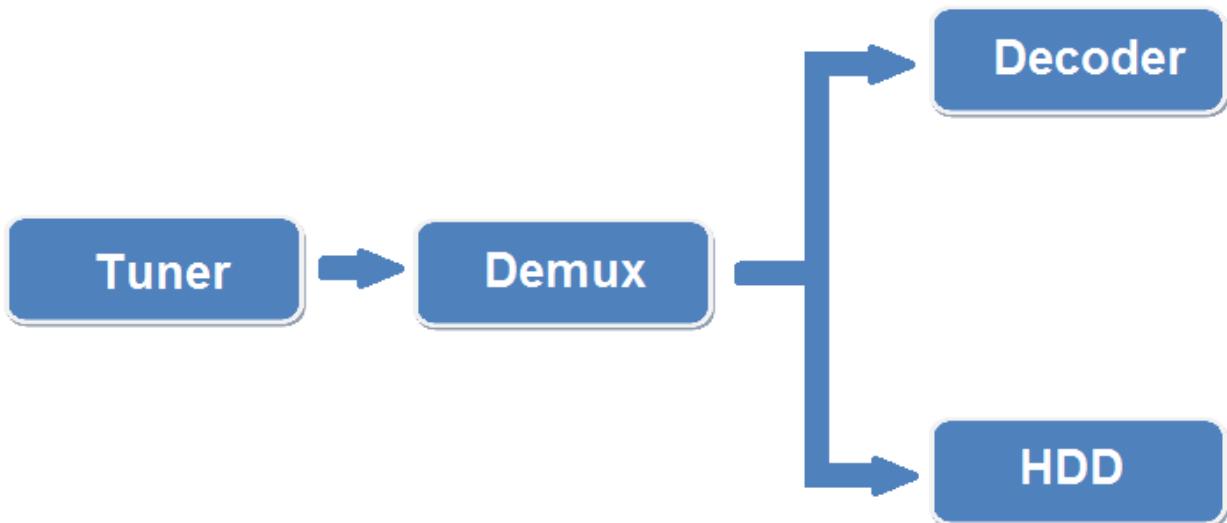
U zavisnosti od toga da li postoji podrška u fizičkoj arhitekturi za filtriranje podataka uloga centralnog procesora u ovom scenariju može biti manja ili veća. Ukoliko postoji pomenuta podrška, tada je uloga centralnog procesora postavljanje odgovarajućih filtera i sinhronizacija upisa i čitanja u i iz cirkularnog bafera. Takođe treba napomenuti da je u ovakvoj situaciji potrebno dva puta filtrirati iste podatke, jednom pri izdvajanju iz toka podataka i opet posle čitanja iz cirkularnog bafera i slanja na dekodere. S druge strane, ukoliko podrška za filtriranje ne postoji u fizičkoj arhitekturi, tada je kompletno filtriranje posao centralnog procesora i to predstavlja ozbiljno opterećenje za sistem.



Slika 3.3 Prikaz istovremene reprodukcije i snimanja DTV servisa u polaznom rešenju

Pošto je na ciljnoj platformi omogućen višestruki pristup toku podataka, nema potrebe za postojanjem cirkularnih bafera. Dakle, moguće je podatke izdvojene iz toka podataka direktno sa demultipleksera, koji vrši filtriranje, slati istovremeno na dekodovanje radi reprodukcije uživo i zapisivati u stalnu memoriju, bez prethodnog zapisivanja u privremenu memoriju. Posledica je veća količina radne memorije na raspolažanju ostatku sistema.

Što se tiče centralnog procesora, s obzirom na to da postoji podrška u fizičkoj arhitekturi za filtriranje, njegov posao je samo da postavi odgovarajuće parametre filtera, što ga značajno rastereće. Veća količina sistemskih resursa na raspolažanju nam omogućava lakšu realizaciju ostalih funkcionalnosti platforme koje su prethodno pomenute. Sve pomenute izmene su izvedene na takav način da sprega ka ostalim slojevima ostane ista kao i u rešenju sa cirkularnim baferima, jer se jezgro nije promenilo, odnosno redosled poziva funkcija prilagodnog sloja je ostao isti.



Slika 3.4 Prikaz istovremene reprodukcije i snimanja DTV servisa nakon prilagođenja

4. Programsко rešenje

Na osnovu prethodno izloženog koncepta sprovedene su promene u srednjem sloju. Promena koja je najviše doprinela uštedi sistemskih resursa je sprovedena u TDAL_TS modulu prilagodnog sloja.

4.1 Realizacija TDAL_TS modula na personalnom računaru

Ovaj modul sadrži podmodule TDAL_TSROUTE, TDAL_TSSOURCE i TDAL_TSSINK.

TDAL_TSROUTE kontroliše rute. Ruta se sastoji od izvora i odredišta. I izvor i odredište su istog tipa tTDAL_TS_Object. Ovaj tip je zapravo 16-to bitni broj koji jedinstveno određuje svaki izvor i odredište u sistemu. Prva četiri bita ovog broja nam govore da li se radi o izvoru (vrednost 0) ili o odredištu (vrednost 1). Sledećih osam bita određuju da li je u pitanju tok podataka, memorija, ili demultiplexer, dok poslednja četiri bita predstavljaju index TDAL_TSROUTE_Object-a tog tipa. Na osnovu tipa izvora i odredišta razlikujemo nekoliko tipova ruta:

- Ruta za reprodukciju uživo (spajaju se tok podataka (izvor) sa demultiplikserom (odredište))
- Ruta za snimanje DTV servisa (tok podataka i memorija)
- Ruta za reprodukciju prethodno snimljenog sadržaja (memorija i demultiplexer)

Ruta koja spaja memoriju sa memorijom se za sada ne koristi, ali u budućnosti će imati primenu prilikom transkodovanja DTV servisa.

Uloga TDAL_TSSOURCE modula je da obezbedi podatke koji ulaze u sistem. Na osnovu izvora tih podataka razlikujemo dva režima rada:

- demultiplexerski (podaci dolaze tokom podataka)

- memorijski (podaci se čitaju iz stalne memorije, odnosno reproducuje se prethodno snimljeni sadržaj)

Sve informacije ovog modula su predstavljene strukturu Extraction_s.

```

struct Extraction_s
{
    tTDAL_TSSOURCE_Type           type;
    enum TDAL_TSSOURCE_ExtractionState state;
    bool                           configured;
    enum TDAL_TSSOURCE_OpenStep_e   open_step;
    enum TDAL_TSSOURCE_ConfigStep_e config_step;
    enum TDAL_TSSOURCE_AddPidStep_e addpid_step;
    enum TDAL_TSSOURCE_StartStep_e  start_step;
    tTDAL_TSSOURCE_CircularBuffer buffer;
    uint32_t                        packet_length;
    struct TaskControl_s           taskctrl;
    union {
        struct dmx
        {
            tTDAL_TS_Object          demux;
            uint32_t                  hSource;
            struct Stream_s          streams[kTDAL_TSSOURCE_MAX_NB_STREAMS];
            uint8_t                   streams_count;
            tTDAL_TSSOURCE_TimestampingCallbackFct timestamping_callback;
            void*                     timestamping_client_tag;
            bool                      outOfTheLoop;
        } dmx;
        struct {
            TDAL_sema_id             sema_client;
            uint32_t                  client_request;
            tTDAL_TSSOURCE_ProducerCallbackFct producer_callback;
            tTDAL_TSSOURCE_RequestJumpCallbackFct request_jump_callback;
            void*                     private_data;
            TDAL_sema_id             sema_pause;
            bool                      paused;
        } mem;
    } u;
};

```

Najvažnija polja u ovoj strukturi su:

- type – predstavlja režim rada
- buffer – cirkularni bafer u koji se upisuju podaci i koji predstavlja vezu sa TDAL_TSSINK-om
- hSource – veza sa fizičkim demultiplexerom
- streams – kanali za izdvajanje podataka sa zadatim PID-om iz toka podataka

TDAL_TSSINK ima zadatak da usmeri podatke koje obezbeđuje TDAL_TSSOURCE u ostatak sistema. I ovde razlikujemo dva režima rada:

- demultiplekserski (podaci se prosleđuju na dekodovanje)
- memorijski (podaci se zapisuju u stalnu memoriju)

Kao i kod TDAL_TSSOURCE-a i TDAL_TSSINK je predstavljen jednom strukturom.

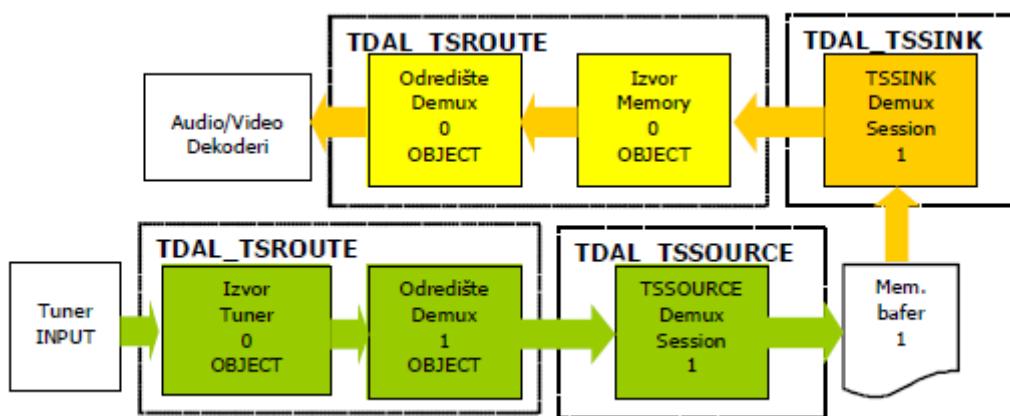
```
struct tTDAL_TSSINK_Injection_s
{
    tTDAL_TSSINK_Type           type;
    enum TDAL_TSSINK_InjectionState_e state;
    enum TDAL_TSSINK_OpenStep_e   open_step;
    enum TDAL_TSSINK_StartStep_e start_step;
    tTDAL_TSSOURCE_Handle      tssource_handle;
    tTDAL_TSSINK_CircularBuffer * p_buffer;
    uint8_t                      consumer_index;
    uint8_t *                    stopped_consumer_address;
    struct TaskControl_s        taskctrl;
    uint32_t                     inject_counter;
    union
    {
        struct
        {
            uint32_t          hIB;
            uint32_t          hSource;
            bool              paused;
            uint32_t          BufferConsumerSize;
            bool              TS_FirstPacketReceived;
            bool              Discontinuity;
            bool              TransfertRelatedToPrevious;
            uint32_t          TS_PacketTimeInMs;
            uint32_t          TS_PacketPosition;
            int32_t           TS_PacketPreviousPosition;
            int32_t           TS_FirstPacketPosition;
            int32_t           TimeToJumpInMs;
            int32_t           NbPacketsTojump;
            bool              JumpRequested;
            bool              JumpRelative;
            bool              RadioInjection;
        } dmx;
        struct
        {
            TDAL_sema_id      sema_client;
            uint32_t           client_request;
            tTDAL_TSSINK_ConsumerCallbackFct consumer_callback;
            void *             client_tag;
        } mem;
    } u;
}
```

Najvažnija polja ove strukture su:

- type - predstavlja režim rada
- p_buffer – pokazivač na cirkularni bafer iz kog se čitaju podaci
- hSource – veza sa fizičkim demultiplerom
- consumer_callback – funkcija kojom se javlja o potrošnji podataka iz bafera

4.1.1 Reprodukcija DTV servisa korišćenjem TDAL_TS modula

U situaciji kada želimo uživo da reprodukujemo jedan DTV servis potrebno je prvo povezati frekvencijski birač i demultiplexer korišćenjem funkcija TDAL_TSROUTE modula. Zatim treba pozvati funkciju TDAL_TSSOURCE_Open i otvoriti TSSOURCE u demultiplikatorskom režimu rada čiji je zadatak da zapiše podatke u cirkularni bafer. Zatim se podese željeni parametri kao što su povratna funkcija, koja služi da obavesti sistem kada su podaci spremni za preuzimanje, i vrednosti PID-ova koje je potrebno izdvojiti. Posle toga pozivom funkcije TDAL_TSSOURCE_Start podaci počinju da se zapisuju u kružni bafer. Istovremeno treba i otvoriti TSSINK pozivom funkcije TDAL_TSSINK_Open kako bi se podaci i čitali iz kružnog bafera i slali na dekodovanje. Veza između TSSOURCE-a i TSSINK-a se ostvaruje pozivom funkcije TDAL_TSSINK_SetTSSource kojom se TSSINK-u prosledi pokazivač na TSSOURCE. Na taj način je osigurano da TSSOURCE i TSSINK pristupaju istom kružnom baferu. Pozivom funkcije TDAL_TSSINK_Start podaci počinju da se čitaju iz kružnog bafera. Da bi podaci stigli do dekodera potrebna nam je još jedna ruta koja će kao izvor imati memoriju, odnosno kružni bafer, a kao odredište demultiplexer preko kojeg se ostvaruje veza sa dekoderima.



Slika 4.1 Reprodukcija uživo dtv servisa korišćenjem TDAL_TS modula

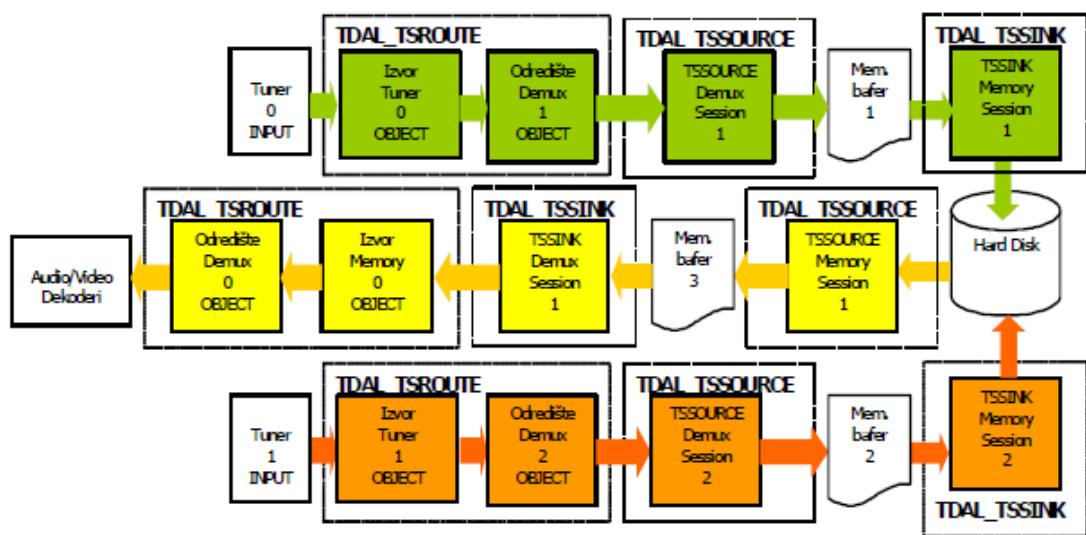
4.1.2 Ostali načini korišćenja TDAL_TS modula

Ukoliko želimo da snimimo DTV servis tada prvi deo rute, do kružnog bafera ostaje isti kao i u prethodnom primeru, dok se TDAL_TSSINK otvara u memorijskom režimu. U ovom

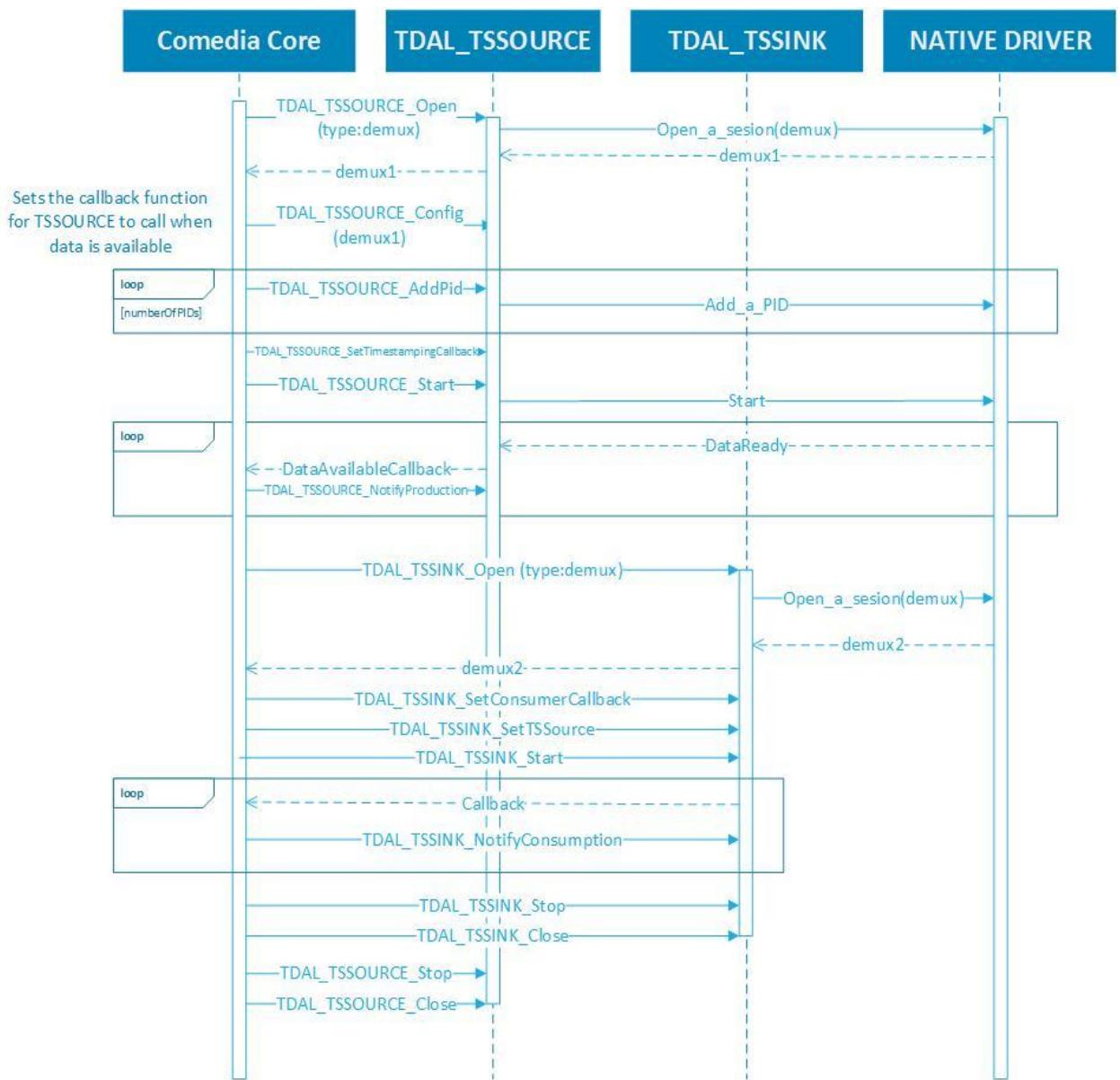
slučaju pre startovanja TDAL_TSSINK-a potrebno je pozvati funkciju TDAL_TSSINK_SetConsumerCallback da bi se postavila funkcija kojom se podaci prosleđuju do modula koji vrši zapisivanje u trajnu memoriju, TDAL_FS modula.

Kada želimo da reprodukujemo prethodno snimljeni sadržaj, tada dolazi do promene u prvom delu rute, a deo iza kružnog bafera ostaje isti kao u slučaju reprodukcije DTV servisa uživo. Naime, u ovom slučaju nam je potreban TDAL_TSSOURCE u memorijskom režimu na početku, da bi se podaci prebacili iz trajne memorije u kružni bafer.

Kombinacijom prethodno navedenih scenarija možemo da ostvarimo razne funkcionalnosti poput reprodukcije, snimanja iii odložene reprodukcije (engl. Timeshift) što se vidi na slici 4.2.



Slika 4.2 Primer snimanja jednog i odložene reprodukcije drugog dtv servisa



Slika 4.3 Dijagram sekvence snimanja DTV servisa korišćenjem TDAL_TS modula

4.2 Prilagođenje TDAL_TS modula fizičkoj arhitekturi

Opisano rešenje je izmenjeno tako što je izbačen kružni bafer između TDAL_TSSOURCE-a i TDAL_TSSINK-a, ali redosled poziva funkcija TDAL_TS modula nije promenjen. Iz tog razloga promenjena je i funkcionalnost određenih funkcija gde je to bilo potrebno. Pa tako funkcija TDAL_TSSOURCE_AddPid se više ne obraća nativnom rukavaocu u cilju dodavanja PID-ova, već se samo zapamte vrednosti željenih PID-ova, koji se kasnije pozivom funkcije TDAL_TSSINK_SetTSSource proslede do TDAL_TSSINK-a gde se zapravo koriste. Dakle u prilagođenom rešenju uloga TDAL_TSSOURCE-a je samo da prosledi potrebne podatke do TDAL_TSSINK-a u kojem se nalazi sva funkcionalnost. Ako je u pitanju reprodukcija DTV servisa uživo, tada se prosleđuje demultiplexer koji je povezan sa frekvenčnim biračem kako bi se iskoristio za filtriranje podataka koje treba poslati na

dekodere. U suprotnom, ako želimo da snimimo određeni DTV servis, tada se do TDAL_TSSINK-a prosleđuju potrebni PID-ovi, kao i funkcija za vremensko pečaćenje koja je potrebna radi pravilnog merenja vremena snimljenog sadržaja.

Pored izmena u funkcijama promenjene su i strukture koje sadrže informacije o ovim modulima. U strukturi Extraction_s izbačen je cirkularni bafer, a dodati su pokazivači na sve TDAL_TSSINK-ove sa kojima je taj TDAL_TSSOURCE povezan. Takođe su izbačeni kanali za izdvajanje podataka, a umesto njih se čuvaju samo informacije potrebne za otvaranje PID-ova. Izbačena su i polja koja su se koristila za sinhronizaciju pri otvaranju TDAL_TSSOURCE-a, jer je većina te funkcionalnosti premeštena u TDAL_TSSINK.

Premeštanjem funkcionalnosti iz TDAL_TSSOURCE-a u TDAL_TSSINK gubi se potreba za zadatkom (engl. Task) koji je postojao u TDAL_TSSOURCE-u i koji se sastojao iz funkcije kojom se javlja da su podaci spremni i potvrde da su podaci iskorišćeni, kao i za kružnim baferom koji je već ranije pomenut. Ipak, kružni bafer nije potpuno uklonjen iz rešenja. Naime, u slučaju snimanja dtv servisa i dalje je potrebno imati kružni bafer, sada u TDAL_TSSINK-u, koji služi kao veza sa TDAL_FS modulom, da bi se osiguralo da se svi podaci uspešno zapišu u trajnu memoriju. Ovakvo rešenje je mnogo bolje u pogledu uštede radne memorije, jer se kružni baferi alociraju samo kada su zaista potrebni i odmah po završetku zadatka za koju su se koristili memorija se oslobođa.

5. Rezultati

Rezultati izvedenog prilagođenja se najbolje ogledaju u smanjenju zahteva fizičke arhitekture, zahvaljujući čemu su realizovane i naprednije funkcionalnosti DTV prijemnika pored onih osnovnih. Kao glavni pokazatelji smanjenja zahteva fizičke arhitekture se izdvajaju količina potrebne radne memorije, kao i zauzeće centralnog procesora.

Što se tiče radne memorije, izvedenim prilagođenjem je količina potrebne radne memorije za ispravno funkcionisanje DTV prijemnika smanjena za 80 Mb što rešava problem nasilnog završetka naše aplikacije od strane Android operativnog sistema.

Kada posmatramo centralni procesor, najzanimljivija nam je situacija kada se vrši reprodukcija jednog DTV servisa uživo, jer je to i najčešći način korišćenja DTV prijemnika. Pre izvršenog prilagođenja, zauzeće centralnog procesora u takvoj situaciji je iznosilo 40%. Posle prilagođenja za istu situaciju se koristi samo 4% ukupnog procesorskog vremena. Takođe i za mozaik režim rada, kada je potrebno reprodukovati više DTV servisa istovremeno, opterećenje procesora je gotovo isto kao i kada se reprodukuje samo jedan DTV servis i iznosi 5%.

Ovakve vrednosti zauzeća centralnog procesora su dodatno značajne jer nam omogućavaju da naš DTV prijemnik radi u režimu pripravnosti većinu vremena, što smanjuje količinu električne energije potrebne za napajanje do 25%. Manja količina električne energije konzumirana od strane DTV prijemnika ima pozitivan uticaj na ekonomski aspekt domaćinstva, kao i na životnu sredinu ako se posmatra na globalnom nivou. Naime, količina električne energije koja bi se uštedela koristeći DTV prijemnike u režimu pripravnosti na nivou cele države jednak je količini električne energije koja se proizvede u nekoliko elektrana. Takođe, sam DTV prijemnik bolje funkcioniše na nižim temperaturama, a zagrevanje integrisanih kola je srazmerno količini električne energije koja se koristi za napajanje. Pored toga što nam obezbeđuje

funkcionosanje bez grešaka, koje su za DTV prijemnik nedopustive, manje zagrevanje integrisanih kola produžava i životni vek DTV prijemnika.

Treba napomenuti da je u prethodno opisanom scenariju korišćena podrška fizičke arhitekture za filtriranje, dok je u situaciji kada je filtriranje zadatak programske podrške, odnosno centralnog procesora, opterećenje procesora iznosilo 60% za rešenje pre izvršenog prilagođenja i 14% po izvršenju prilagođenja.

Pored reprodukcije jednog dtv servisa uživo posmatran je i uticaj snimanja i distribucije DTV servisa na opterećenost centralnog procesora. Kada je reč o snimanju DTV servisa situacija je sledeća. Pre prilagođenja za reprodukciju i snimanje jednog DTV servisa je bilo potrebno 60% procesorskog vremena, dok se ta brojka povećava za oko 15% sa svakim dodatim servisom koji se snima. Posle prilagođenja potrebno je oko 15% procesorskog vremena za reprodukciju i snimanje jednog DTV servisa, dok svaki novi servis koji se snima doprinosi opterećenju procesora sa oko 5%. Ako govorimo o distribuciji DTV servisa u lokalnoj mreži, tada imamo sledeću situaciju. Pre prilagođenja je za reprodukciju i distribuciju jednog DTV servisa bilo potrebno 53% procesorskog vremena. Kako je za reprodukciju potrebno 40% procesorskog vremena, vidimo da je udeo distribucije u opterećenju centralnog procesora oko 13%. Posle prilagođenja situacija je značajno promenjena. Na opterećenje procesora od 4% koliko je potrebno za reprodukciju jednog dtv servisa, distribucija svakog sledećeg servisa doprinosi sa svega 1%.

Ako pogledamo rezultate zabeležene analiziranjem situacije kada se vrše snimanje ili distribucija DTV servisa, vidimo da distribucija ne opterećuje značajno centralni procesor i moguć je rad centralnog procesora u režimu pripravnosti, dok kod snimanja to nije slučaj. Ipak, značajno smanjenje zauzeća centralnog procesora pri snimanju dtv servisa donosi nam mogućnost snimanja više DTV servisa istovremeno nego što je to bio slučaj pre prilagođenja, a tu mogućnost nam pruža i postojanje više frekvencijskih birača.

Vrednosti iskorišćenja centralnog procesora prilikom nekih od mogućih scenarija date su u tabeli 5.1.

| Iskorišćenje centralnog procesora | Reprodukcijski dokument uživo | Reprodukcijski i snimanje dokument servisa | Reprodukcijski i snimanje dokument servisa | Reprodukcijski i distribucijski dokument servisa |
|---|-------------------------------------|---|---|---|
| Pre prilagodenja | 40% | 60% | 75% | 53% |
| Posle prilagodenja | 4% | 15% | 20% | 5% |

Tabela 5.1 Iskorišćenje centralnog procesora u nekoliko različitih scenarija

6. Zaključak

U ovom radu je prikazano jedno rešenje prilagođenja programske podrške za upravljanje DTV prijemnikom BCM97435 VMS zasnovanom na Android platformi. Cilj prilagođenja je bio optimizacija programske podrške, za konkretnu platformu, pri izvršavanju osnovnih DTV funkcionalnosti radi realizacije naprednijih DTV funkcionalnosti.

Prilagođenje je sprovedeno nad srednjim slojem programske podrške, a promene koje imaju najveći uticaj na smanjenje zauzeća sistemskih resursa su napravljene u prilagodnom sloju srednjeg sloja programske podrške. Takođe dodata su određene funkcionalnosti poput prikaza mozaika i distribucije DTV servisa u lokalnoj mreži u realnom vremenu, dok su druge funkcionalnosti poput ličnog video snimača proširene kako bi se dostigao pun potencijal fizičke arhitekture za koju je prilagođenje rađeno.

Rezultati prilagođenja se najbolje vide kroz posmatranje opterećenosti centralnog procesora i količini potrebne radne memorije pre i posle prilagođenja. Radna memorija je svedena na nivo koji omogućava pouzdan rad na ciljnoj platformi i ostavlja prostora za dalja proširenja i unapređenja postojećih funkcionalnosti. Što se tiče centralnog procesora, značajno je smanjeno opterećenje pri osnovnim, ali i naprednjim DTV funkcionalnostima. Time je omogućeno da DTV prijemnik većinu vremena radi u režimu pripravnosti odakle proizilaze i dodatni rezultati prilagođenja, a to su manja potrošnja električne energije, koja sa sobom povlači i manje zagrevanje te pouzdaniji rad i duži životni vek uređaja.

Šta se tiče daljih proširenja, u cilju unapređenja distribucije sadržaja u lokalnoj mreži, potrebno je uvesti transkodovanje. U kombinaciji sa prilagodljivim protokolom, transkodovanje bi omogućilo sigurnu distribuciju sadržaja u lokalnoj mreži i pri uslovima koji nisu idealni, što je česta situacija u realnom svetu. Transkodovanje se vrlo lako može uključiti u postojeći sistem rutiranja korišćenjem ruta koje već postoje, a ne koriste se.

7. Literatura

- [1] M. Vidakovic N. Teslic, T. Maruna and V. Mihic, “Android4TV: a Proposition for Integration of DTV in Android Devices”, IEEE 30th International Conference on Consumer Electronics (ICCE), Las Vegas, January 2012, pp. 441-442.
- [2] N. Kuzmanovic, T. Maruna, M. Savic, G. Miljkovic, D. Isailovic, “Google’s android as an application environment for DTV decoder system”, IEEE 14th International Symposium on Consumer Electronics (ISCE), Braunschweig, June 2010.
- [3] N. Horowitz, C. Calwell, S. Foster, “ Opportunities and recommendations for reducing the energy consumption of consumer electronics products”, IEEE International Symposium on Electronics and the Environment, May 2005.
- [4] P. Lall, M. G. Pecht, E. B. Hakim, “Influence of Temperature on Microelectronics and System Reliability”, CRC PressINC, 1997.
- [5] Sajt Android podrške za razvoj, www.developer.android.com
- [6] Predavanja iz predmeta Programska podrška u televiziji i obradi slike 1, <http://www.rtrk.uns.ac.rs/studijski-program-2009/vii-2009/pputvios1/750-pputvios-1-literatura>
- [7] Predavanja iz predmeta Programska podrška u televiziji i obradi slike 2, <http://www.rtrk.uns.ac.rs/studijski-program-2009/ix-2009/pputvios2/761-pputvios-2-predavanja>
- [8] N. Fimić, D. Rapić, N. Kuzmanović, R. Pavlović, D. Dejanović, “Jedno rešenje prilagođenja programske podrške za upravljanje DTV prijemnikom zasnovanom na Android OS-u”, 57. konferencija etran, Zlatibor, jun 2013.