



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Небојша Кошутић

**Једно решење интеграције и приказа
сервиса дигиталне телевизије на
пријемницима телевизијског сигнала
заснованим на Андроид платформи**

МАСТЕР РАД

Нови Сад, 2014

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Број:
	ЗАДАТАК ЗА МАСТЕР РАД	Датум:

(Податке уноси предметни наставник - ментор)

СТУДИЈСКИ ПРОГРАМ:	Рачунарство и аутоматика
РУКОВОДИЛАЦ СТУДИЈСКОГ ПРОГРАМА:	Проф. др Никола Јорговановић

Студент:	Небојша Кошутић	Број индекса:	E2 61/2011
Област:	Програмска подршка у телевизији и обради слике 2		
Ментор:	Проф. др Никола Теслић		
НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА МАСТЕР РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА: - проблем – тема рада; - начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна;			

НАСЛОВ МАСТЕР РАДА:

Једно решење интеграције и приказа сервиса дигиталне телевизије на пријемницима телевизијског сигнала заснованим на Андроид платформи

ТЕКСТ ЗАДАТКА:

Реализовати проширење Андроид оперативног система интеграцијом постојеће програмске подршке дигиталне телевизије. Проширење треба да омогући претрагу и приказ сервиса дигиталне телевизије, подешавање квалитета слике и тона, приказ електронског програмског водича, приказ телетекст садржаја и снимање телевизијског програма у сврху одложеног гледања. Интеграцију треба извести на нивоу Андроид радног окружења, тако да могућности програмске подршке дигиталне телевизије буду доступне за имплементацију корисничких апликација за приказ сервиса дигиталне телевизије.

Руководилац студијског програма:	Ментор рада:

Примерак за: ☐ - Студента; ☐ - Ментора
--



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР :	
Идентификациони број, ИБР :	
Тип документације, ТД :	Монографска документација
Тип записа, ТЗ :	Текстуални штампани материјал
Врста рада, ВР :	Дипломски – мастер рад
Аутор, АУ :	Небојша Кошутин
Ментор, МН :	Никола Теслић проф. др
Наслов рада, НР :	Једно решење интеграције и приказа сервиса дигиталне телевизије на пријемницима телевизијског сигнала заснованим на Андроид платформи
Језик публикације, ЈП :	Српски / латиница
Језик извода, ЈИ :	Српски
Земља публикавања, ЗП :	Република Србија
Уже географско подручје, УГП :	Војводина
Година, ГО :	2014.
Издавач, ИЗ :	Ауторски репринт
Место и адреса, МА :	Нови Сад; трг Доситеја Обрадовића 6
Физички опис рада, ФО : (поглавља/страница/ цитата/табела/слика/графика/прилога)	9/58/21/5/30/8/0
Научна област, НО :	Електротехника и рачунарство
Научна дисциплина, НД :	Рачунарска техника
Предметна одредница/Кључне речи, ПО :	Пријемник дигиталне телевизије, Андроид, Интеграција
УДК	
Чува се, ЧУ :	У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, ВН :	
Извод, ИЗ :	Овим радом описан је један приступ интеграције програмске подршке дигиталне телевизије у Андроид оперативни систем. Дат је кратак преглед стања у области и теоријске основе потребне за разумевање датог проблема. Реализовано решење описано је до нивоа детаља могућности поновне имплементације и извршена је анализа експерименталних резултата рада реализованог система.
Датум прихватања теме, ДП :	
Датум одбране, ДО :	
Чланови комисије, КО :	Председник: Проф. др Иштван Папп
	Члан: Проф. др Никола Челановић
	Члан, ментор: Проф. др Никола Теслић
	Потпис ментора



KEY WORDS DOCUMENTATION

Accession number, ANO :			
Identification number, INO :			
Document type, DT :	Monographic publication		
Type of record, TR :	Textual printed material		
Contents code, CC :	Master Thesis		
Author, AU :	Nebojsa Kosutic		
Mentor, MN :	PhD Nikola Teslic		
Title, TI :	One solution for integration and presentation of digital television services an Android based DTV platform		
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 :	2014.		
Publisher, PB :	Author's reprint		
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6		
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	9/58/21/5/30/8/0		
Scientific field, SF :	Electrical Engineering		
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems		
Subject/Key words, S/KW :	Digital television receiver, Android, integration		
UC			
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, N :			
Abstract, AB :	This paper describes an approach to the integration of digital television software in the Android operating system. Gives a brief review of state in area and the theoretical foundations needed for problem understanding. Actual solution is described to the level of details that allows re-implementation and there is a given analysis of the experimental results of implemented system operations.		
Accepted by the Scientific Board on, ASB :			
Defended on, DE :			
Defended Board, DB :	President:	PhD Istvan Papp	Menthor's sign
	Member:	PhD Nikola Celanovic	
	Member, Mentor:	PhD Nikola Teslic	



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

SADRŽAJ

1. Uvod	1
2. Stanje oblasti istraživanje.....	5
2.1 Postojeća indistrijska rešenja	5
2.2 Naučna dostignuća	6
2.3 Android kao DTV platforma.....	9
3. Teorijske osnove	13
3.1 Digitalna televizija	13
3.2 Teletext i prevodi	16
3.3 Android	16
3.4 Binder.....	18
3.5 Upravljanje resursima.....	19
4. Analiza problema i postavke ciljeva	21
5. Opis rešenja.....	23
5.1 Media Player (upravljanje resursima).....	23
5.2 Modul za upravljanje DTV uređajem	25
5.3 Pretraga dostupnih DTV servisa.....	26
5.4 Izbor prikaza DTV servisa	28
5.5 Podešavanje kvaliteta slike i tona	29
5.6 Prikaz elektronskog programskog vodiča	30
5.7 Prikaz teletext sadržaja	32
5.8 Prikaz prevoda	33
5.9 PVR podrška.....	33
5.10 DTV Middleware Service.....	34

5.10.1	Programska sprega za pretragu dostupnih televizijskih servisa	36
5.10.2	Programska sprega za manipulaciju listom pronađenih televizijskih servisa .	37
5.10.3	Programska sprega za podešavanje kvaliteta slike	39
5.10.4	Programska sprega za podešavanje kvaliteta zvuka.....	40
5.10.5	Programska sprega za upravljanje teletext podrškom	42
5.10.6	Programska sprega za upravljanje EPG podrškom.....	43
5.10.7	Programska sprega za upravljanje prikazom podnapisa.....	44
5.10.8	Programska sprega za upravljanje PVR podrškom.....	45
5.10.9	Programska sprega za upravljanje HbbTV podrškom.....	47
6.	Eksperimentalni rezultati	48
6.1	Analiza performansi rešenja.....	48
6.1.1	Zauzeće centralne procesorske jedinice.....	49
6.1.2	Zauzeće radne memorije.....	51
6.2	Analiza odziva sistema	52
7.	Pravci daljeg razvoja	55
8.	Zaključak	56
9.	Literatura	57

SPISAK SLIKA

Slika 1.1: Zastupljenost televizijskih prijemnika u domaćinstvima SAD [1]	1
Slika 1.2: Broj domaćinstava sa pristupom DTV sadržaju visoke rezolucije u SAD [1].....	2
Slika 1.3: Broj dostupnih Android aplikacija [2].....	3
Slika 2.1: LG 55G2 Android DTV prijemnik kompanije LG [4].....	5
Slika 2.2: SFR Android DTV prijemnik [5].....	6
Slika 2.3: Posrednički DTV Android uređaji Asus Cube [6] levo i Sony NSZ-GS8 desno [7].....	6
Slika 2.4: Primer integracije DTV MDW na nivou Java programskog jezika	7
Slika 2.5: Primer integracije platformski zavisne DTV MDW podrške.....	8
Slika 2.6: Organizacija DTV proširenja Android radnog okruženja	9
Slika 2.7: Tok korisničkog unosa	10
Slika 2.8: Media Player signalizacija.....	11
Slika 2.9: Android Media Player programska organizacija	12
Slika 3.1: Organizacija DVB prenosnog toka podataka	14
Slika 3.2: MPEG-2 paketizovani elementarni tokovi podataka	15
Slika 3.3: MPEG-2 transportni tok podataka.....	15
Slika 3.4: DTV prijemnik	15
Slika 3.5: Programska organizacija Android operativnog sistema.....	17
Slika 3.6: Tok Binder komunikacije	18
Slika 3.7: Tok upravljanja resursima	19
Slika 4.1: Dijagram slučaja korišćenja televizijske podrške.....	21
Slika 5.1: Blok dijagram algoritma pokretanja reprodukcije DTV servisa	24
Slika 5.2: UML dijagram klasa sprežavanja DTV podrške.....	26

Slika 5.3: Blok dijagram algoritma pretrage DTV servisa	27
Slika 5.4: Tok pretrage DTV servisa	28
Slika 5.5: Tok izbora prikaza DTV servisa	29
Slika 5.6: Organizacija korisničke sprege za podešavanje slike i tona.....	29
Slika 5.7: Tok promene i čitanja vrednosti parametara slike i tona.....	30
Slika 5.8: Blok dijagram algoritma akvizicije EPG sadržaja	31
Slika 5.9: Tok akvizicije EPG sadržaja.....	31
Slika 5.10: Tok prikaza teletext sadržaja	32
Slika 5.11: Tok prikaza prevoda.....	33
Slika 5.12: Tok pokretanja snimanja trenutno aktivnog DTV sadržaja.....	34
Slika 5.13: Programska organizacija DTV podrške	35
Slika 6.1: Graf dijagram utroška procesorskog vremena.....	49
Slika 6.2: Graf dijagram <i>load average</i> vrednosti	50
Slika 6.3: Graf dijagram zauzeća radne memorije izraženo u KB	51
Slika 6.4: Vreme uključanja referentnog sistema	52
Slika 6.5: Uporedni prikaz vremena uključanja televizijskih uređaja	53

SPISAK TABELA

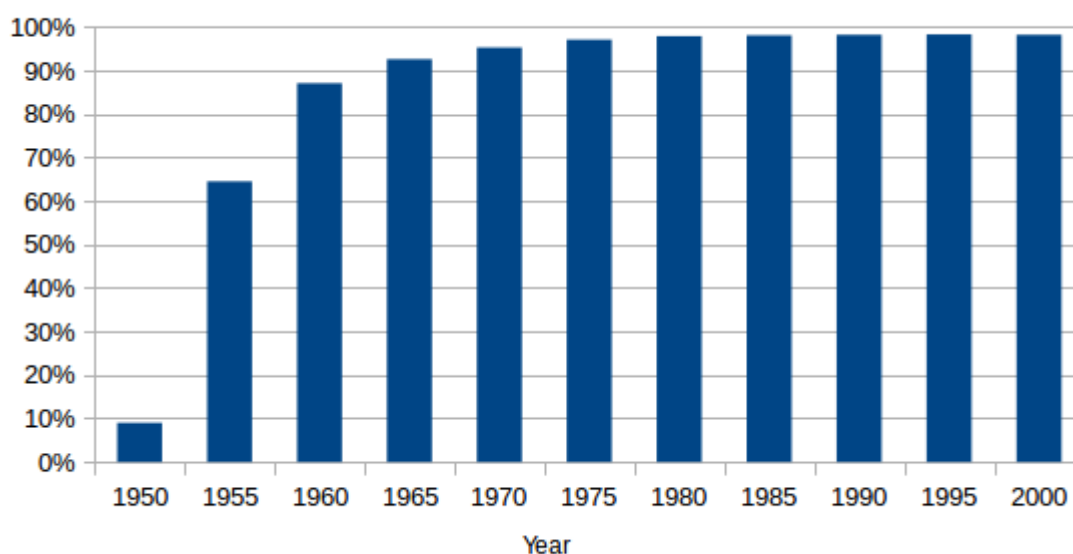
Tabela 3.1: Minimalne učestanosti servisnih informacija [17]	14
Tabela 4.1: Osnovne funkcionalnosti DTV prijemnika.....	22
Tabela 4.2: Napredne funkcionalnosti DTV prijemnika	22
Tabela 6.1: Vreme uključenja sistema izraženo u sekundama.....	53
Tabela 6.2: Vreme promene televizijskog servisa izraženo u sekundama.....	54

SKRAĆENICE

DTV	- <i>Digital television</i> , Digitalna televizija
PAT	- <i>Program Association Table</i> , Tabela asocijacije servisa
CAT	- <i>Conditional Access Table</i> , Tabela uslovnog pristupa
TSDT	- <i>Transport Stream Description Table</i> , Tabela opisa transportnog toka
PMT	- <i>Program Map Table</i> , Tabela mapiranja servisa
NIT	- <i>Network Information Table</i> , Tabela opisa mreže
SDT	- <i>Service Description Table</i> , Tabela opisa servisa
EIT	- <i>Event Information Table</i> , Tabela opisa događaja
TDT	- <i>Time and Date Table</i> , Tabela vremena i datuma
PES	- <i>Packetized Elementary Stream</i> , Paketizovani elementarni tok
AIT	- <i>Application Information Table</i> , Tabela informacija o aplikacijama
STB	- <i>Set-Top Box</i>
EPG	- <i>Electronic Program Guide</i> , Elektronski programski vodič
SOC	- <i>System On a Chip</i> , Sistem u okviru integrisanog kola
CPU	- <i>Central Processing Unit</i> , Centralna procesorka jedinica
IPC	- <i>Inter Process Communication</i> , Međuprocesna komunikacija

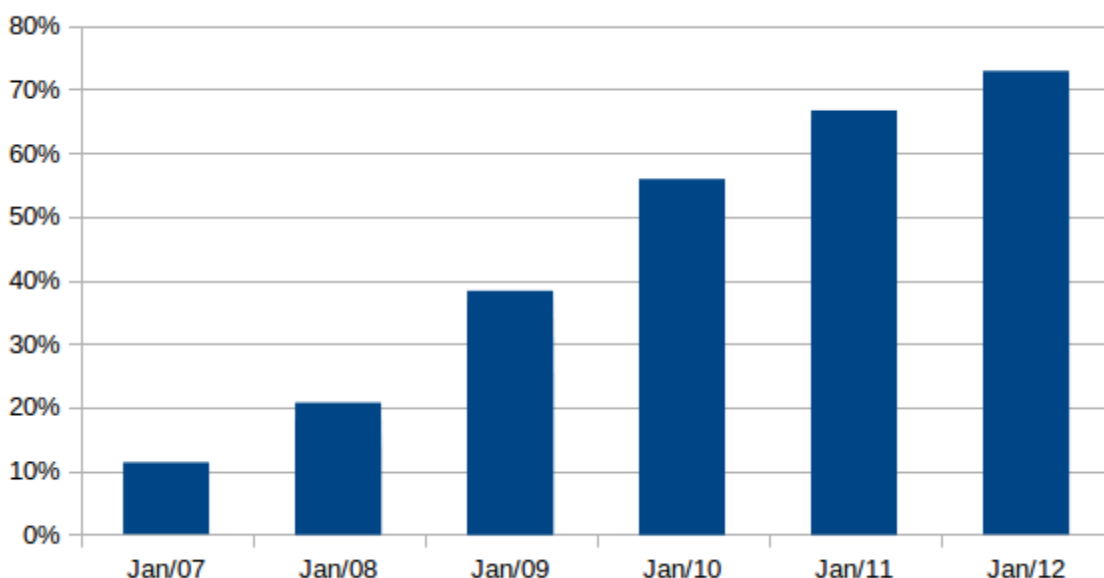
1. Uvod

Od svog nastanka, televizija zauzima značajno mesto u ljudskom društvu. Televizija u odnosu na druge klasične medije, pruža mogućnost najbržeg i najpotpunijeg informisanja. Svi značajni događaji emituju se posredstvom televizije. Vrlo često emitovani sadržaj dostupan je u celom svetu putem satelita. Pored informativne uloge, televizija ima značajnu ulogu u domenu zabave. Postoji velik broj televizijskih servisa koji emituju samo zabavni sadržaj određenog tipa, na primer igrani filmovi, dokumentarni filmovi, sport itd. Slika 1.1 prikazuje broj televizijskih prijemnika u domaćinstvima Sjedinjenih Američkih Država već od 1965. godine prevazilazi 90%.



Slika 1.1: Zastupljenost televizijskih prijemnika u domaćinstvima SAD [1]

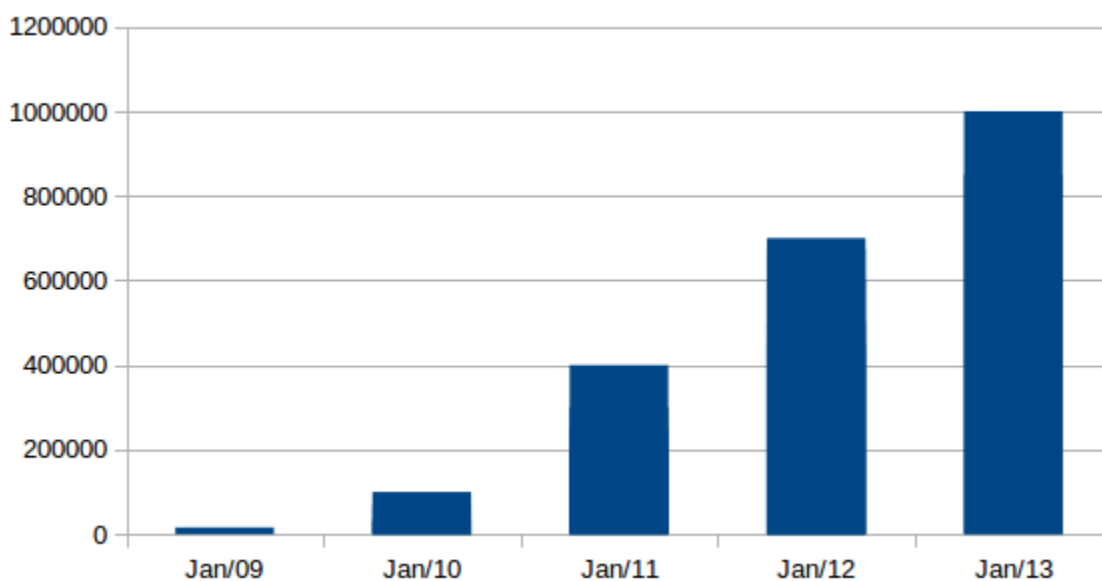
Zbog značaja i velike rasprostranjenosti, postoji stalna težnja za razvojem televizije. Dosadašnji razvoj ogleda se uglavnom u tehničkom napretku. Tako su televizijski prijemnici prešli put od analognih monohromatskih uređaja sa elektronskim cevima do mikro računarskih sistema, koji su u stanju da prikažu video sadržaj visoke rezolucije i imaju pristup internetu. Najveća promena u domenu televizije nastala je uvođenjem standarda digitalne televizije, kao što su DVB i ATSC. Digitalna televizija (engl. Digital Television, DTV) u poređenju sa analognom televizijom mnogo bolje koristi frekventni spektar, što omogućava emitovanje video sadržaja boljeg kvaliteta. Uz dodatne informacije kao što su višezječni audio sadržaj, teletext (engl. Teletext, TTX), prevodi (engl. Subtitle, SUB) i elektronski raspored programa (engl. Electronic Program Guide, EPG). Slika 1.2 sadrži pregled dostupnosti televizijskog sadržaja visoke rezolucije u Sjedinjenim Američkim Državama, na osnovu čega se može izvesti zaključak o stabilnom trendu rasta.



Slika 1.2: Broj domaćinstava sa pristupom DTV sadržaju visoke rezolucije u SAD [1]

U poslednje vreme, televizijski prijemnici prestaju da budu samo izvor najnovijih informacija koje se prezentuju gledaocu i postaju centar domaćinstva sa stanovišta upotrebe koja prevazilazi ustaljen oblik. Televizijski prijemnici postaju uređaji koji se koriste za igranje, pregled multimedijalnog sadržaja, pregled internet sadržaja i pristup brojnim internet servisima. Pomenute tendencije za posledicu imaju sve moćnija rešenja televizijskih prijemnika, sa stanovišta fizičke arhitekture. Višezvezgarne centralne procesorske jedinice naprednih mogućnosti, kao i velika količina dostupne radne memorije predstavljaju pogodnu platformu za upotrebu operativnih sistema opšte namene pri implementaciji televizijskih prijemnika.

Operativni sistem Android predstavlja prihvatljivo rešenje za upotrebu u domenu televizijskih prijemnika. Osnovna prednost Android operativnog sistema je otvoreni programski kod čime je obezbeđena prilagodljivost različitim platformama fizičke arhitekture i mogućnost proširenja postojećih funkcionalnosti datog programskog rešenja. Sa stanovišta arhitekture centralne procesorske jedinice operativni sistem Android podržava ARM, MIPS i x86 arhitekture. Prednost u funkcionalnom smislu ogleda se u lakoj proširivosti skupa funkcionalnosti na visokom nivou apstrakcije u programskom jeziku Java i već javno dostupna velika baza računarskih aplikacija sa trendom rasta.



Slika 1.3: Broj dostupnih Android aplikacija [2]

Cilj rada je implementacija televizijskog prijemnika baziranog na Android operativnom sistemu, integracijom televizijske podrške u postojeću arhitekturu operativnog sistema. Televizijski prijemnik podrazumeva uređaj sa mogućnošću prijema i reprodukcije televizijskog signala. Data je analiza već postojećih komercijalnih rešenja u datoj oblasti, kao i kratak pregled naučnih publikacija na zadatu temu. Navedeni sadržaj je proizvod istraživanja stanja u oblasti, da bi zadovoljavajuća rešenja mogli preuzeti u celini ili delovima, čemu možemo dodati i opis podsistema televizijske podrške Android radnog okruženja. Za razumevanje detalja opisa predloženog rešenja postoji potreba za odgovarajućim skupom teorijskih znanja iz domena digitalne televizije, kao i arhitekture i određenih elemenata programske podrške Android operativnog sistema. Zarad jasnog definisanja ciljeva dato je poglavlje koje pored pregleda osnovnih i naprednih funkcionalnosti DTV prijemnika sadrži i analizu stepena implementacije neophodnih komponenti podsistema DTV programske podrške. Opis rešenja predstavlja tehničku dokumentaciju implementiranog rešenja, čime su prikazani detalji implementacije

upotrebom formalnih i neformalnih dijagrama, do nivoa pojedinosti koji omogućava ponovnu realizaciju datog rešenja. Analiza stabilnosti, performansi, odziva i zauzeća resursa fizičke arhitekture realizovanog sistema, ima za cilj utvrđivanje upotrebne mogućnosti u realnim radnim uslovima. Takođe, dat je pregled ostvarene televizijske programske podrške, kao i mogućnosti unapređenja predloženog rešenja.

2. Stanje oblasti istraživanje

2.1 Postojeća indistrijska rešenja

Postoje mnogobrojna komercijalno dostupna industrijska rešenja bazirana na Android operativnom sistemu u domenu DTV prijemnika [3]. Međutim, retka su rešenja sa potpuno integrisanom televizijskom podrškom, odnosno uređaji sa potpunim skupom funkcionalnosti za samostalni prijem i reprodukciju televizijskog signala. U najvećoj meri zastupljeni su posrednički uređaji, gde posredničkim nazivamo uređaje koji televizijski sadržaj dobijaju od već postojećih DTV prijemnika posredstvom HDMI sprege. Uobičajeni slučaj korišćenja posredničkih uređaja ogleda se u njegovom povezivanju sa izvorom video signala, nakon čega posrednički uređaj dodaje sadržaj Android operativnog sistema preko izvorišnog video signala.

Kao primer integrisanog DTV prijemnika zasnovanog na Android operativnom sistemu možemo navesti LG 55G2 DTV prijemnik kompanije LG [4]. Reč je o televizijskom prijemniku analogne i digitalne televizije (ATSC standard) sa pristupom internetu (Slika 2.1). Pored uobičajenog televizijskog sadržaja korisnik ima mogućnost pristupa multimedijalnim servisima za iznajmljivanje video i audio sadržaja. Takođe, korisnik na raspolaganju ima velik broj računarskih aplikacija iz širokog spektra oblasti.



Slika 2.1: LG 55G2 Android DTV prijemnik kompanije LG [4]

Rešenje francuskog telekomunikacionog distributera SFR [5], podržava DVB-T standard digitalne televizije i posredstvom interneta ima pristup multimedijalnom sadržaju distributera, kao što je video na zahtev, radio servisi itd. Pored sadržaja obezbeđenog od strane samog distributera, korisnicima je takođe dostupan multimedijalni sadržaj brojnih internet servisa.



Slika 2.2: SFR Android DTV prijemnik [5]

Kod posredničkih uređaja korisnici na raspolaganju imaju multimedijalne internet servise kao i druge mogućnosti Android operativnog sistema u vidu mnogobrojnih računarskih aplikacija. Kao posrednička rešenja izdvajaju se Asus Cube [6] (Slika 2.3), Sony NSZ-GS8 [7] (Slika 2.3), Logitech Revue [8], NETGEAR NeoTV Prime (GTV100) [9], Hisense Pulse [10], Vizio Co-Star Stream Player (VAP430) [11].

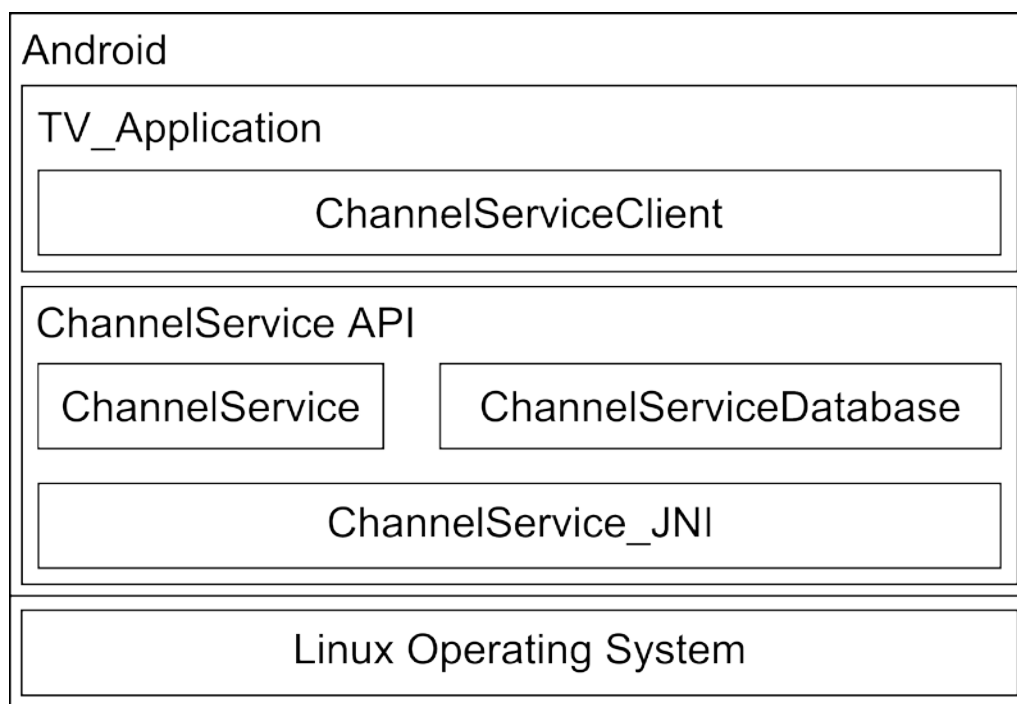
Slika 2.3: Posrednički DTV Android uređaji Asus Cube [6] levo i Sony NSZ-GS8 desno [7]

2.2 Naučna dostignuća

Moderni multimedijalni uređaji zasnovani su na vešejezgarnim računarskim integrisanim kolima, koji poseduju i namenska jezgra, takozvana SOC rešenja (eng. System on Chip). Namenska jezgra su optimizovana za velik broj operacija množenja i akumulacije, što je pogodno za digitalnu obradu signala. Kao takva, namenska jezgra su deo grafičkog podsistema ili podsistema za video i audio prikaz. Opšte namenska jezgra ovakvih rešenja izvršavaju operativni sistem, dok se namenska jezgra aktiviraju po potrebi. S obzirom da već postojeći uređaji bazirani na Android operativnom sistemu poseduju opisanu fizičku arhitekturu, Android

kao operativni sistem je izuzetno pogodan za implementaciju DTV prijemnika. Za mogućnost prijema i reprodukcije televizijskog signala u postojeće programsko rešenje operativnog sistema potrebno je dodati podsistem programske podrške za digitalnu televiziju – DTV MDW (eng. DTV Middleware) [12].

Korisnička programska podrška u Android operativnom sistemu realizovana je u programskom jeziku Java, da bi zahvaljujući apstrakciji virtuelne mašine ista bez dodatnih izmena bila dostupna na uređajima različitih proizvođača. Iz tog razloga i DTV programska podrška mora biti dostupna na nivou Java programskog jezika. Jedan pristup podrazumeva DTV MDW implementaciju u programskom jeziku Java [13]. Kod ovog pristupa DTV MDW je realizovan u vidu Android servisa (Slika 2.4). Uloga DTV MDW servisa je pružanje usluga DTV aplikacijama posredstvom definisane programske sprege. Osnovne usluge DTV MDW servisa su mogućnost pronalaženja i prikaz servisa digitalne televizije. Kako bi DTV MDW servis mogao upravljati komponentama fizičke arhitekture neophodno je obezbediti pristup programskoj podršci niskog nivoa apstrakcije posredstvom JNI programske sprege.

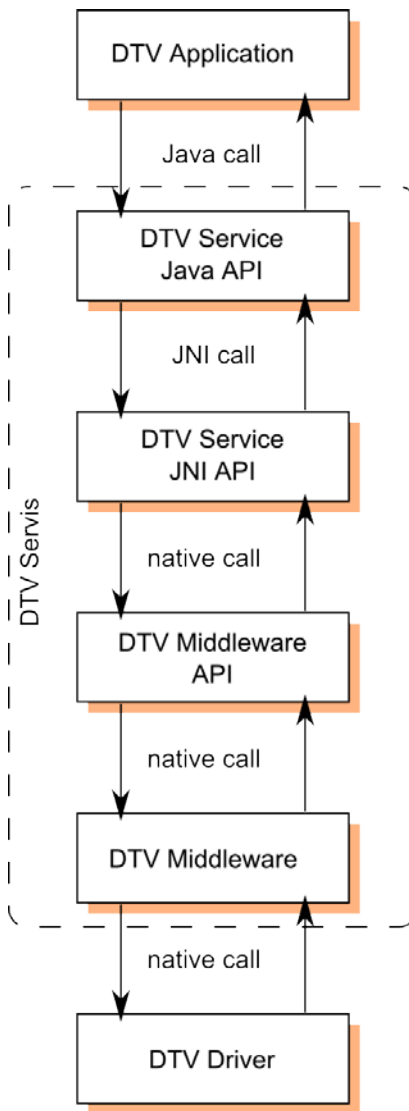


Slika 2.4: Primer integracije DTV MDW na nivou Java programskog jezika

DTV korisnička aplikacija, kao deo programske podrške koji pruža spregu ka korisniku, DTV servisu pristupa uz pomoć mehanizma međuprocene komunikacije. Korisnička sprega ogleda se u prikazu DTV sadržaja i mogućnosti korisničke interakcije sa DTV prijemnikom.

Uzimajući u obzir činjenicu da se sva Java programska rešenja izvršavaju posredstvom virtuelne mašine, takva rešenja imaju loše performanse, koje su od suštinskog značaja za televizijsku podršku. Iz tog razloga, DTV MDW je potrebno implementirati na nivou apstrakcije

koji obezbeđuje obradu u realnom vremenu [15]. Implementacijom DTV MDW podsistema uz pomoć programskog jezika C/C++ ostvarena je potreba za vremenski kritičnom programskom podrškom, međutim u ovom slučaju pristup televizijskoj programskoj podršci sa nivoa apstrakcije Java programskog jezika ostvaruje se posredstvom JNI programske sprege (Slika 2.5).

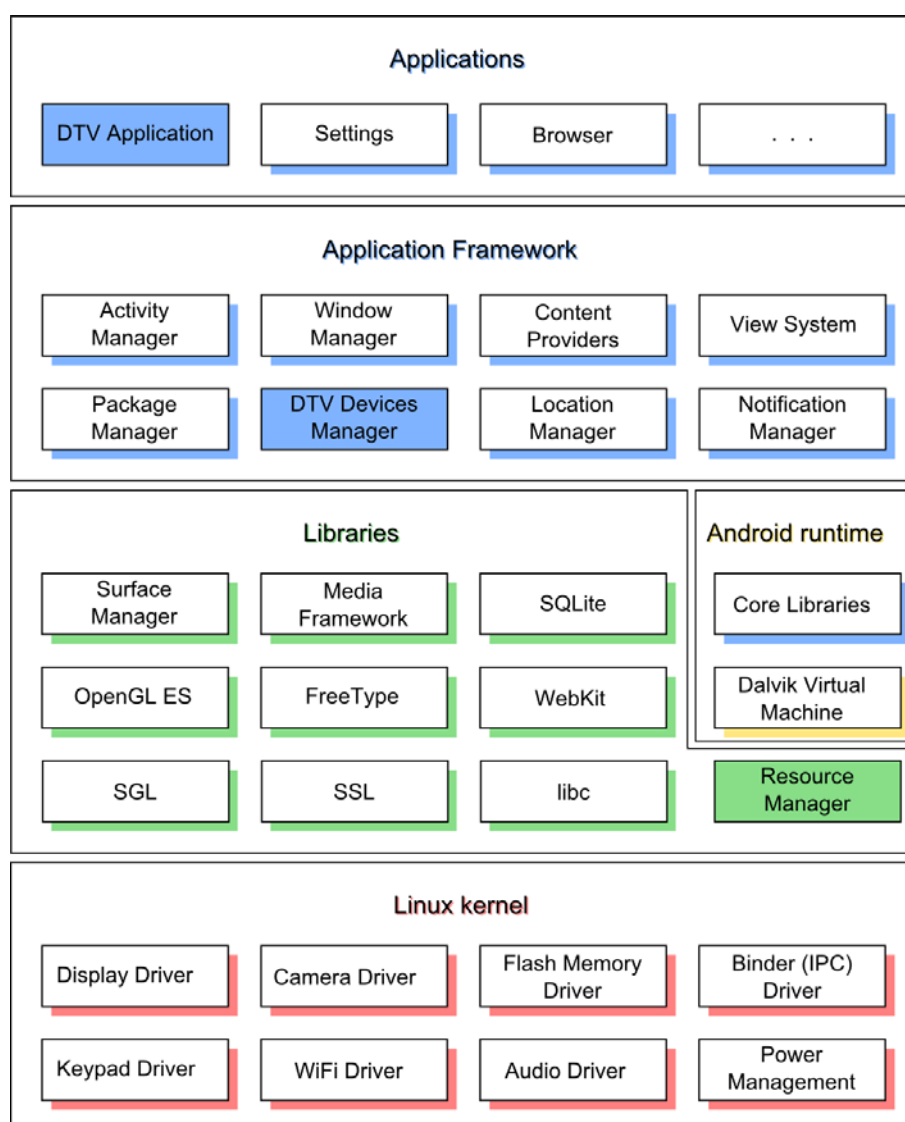


Slika 2.5: Primer integracije platformski zavisne DTV MDW podrške

I ovim rešenjem DTV programska podrška realizovana je Android servisom, međutim u ovom slučaju programsko rešenje na nivou apstrakcije programskog jezika Java samo pokreće i pruža programsku spregu ka DTV MDW implementaciji čija funkcionalnost je u potpunosti realizovana na nižem nivou apstrakcije. Možemo videti da je ovakvo rešenje slojevitije, odnosno sadrži veći broj nivoa apstrakcije, što ga čini komplikovanijim u odnosu na Java implementaciju, ali performanse ovakvog rešenja su dokazano dovoljne za obradu u realnom vremenu.

2.3 Android kao DTV platforma

Iako operativni sistem Android ne poseduje DTV podršku, nudi mehanizme za integraciju iste. U okviru Android Java radnog okruženja postoji odgovarajući podsistem za integraciju DTV podrške, tako da je moguće ostvariti primarnu televizijsku funkcionalnost zahvaljujući upravljanju resursima. Iako standardno rešenje Android operativnog sistema namenjeno prenosnim uređajima ne poseduje mogućnost upravljanja resursima, rešenje namenjeno televizijskim prijemnicima poseduje pomenutu funkcionalnost. Slika 2.6 sadrži prikaz organizacije Android operativnog sistema namenjenog televizijskim prijemnicima.

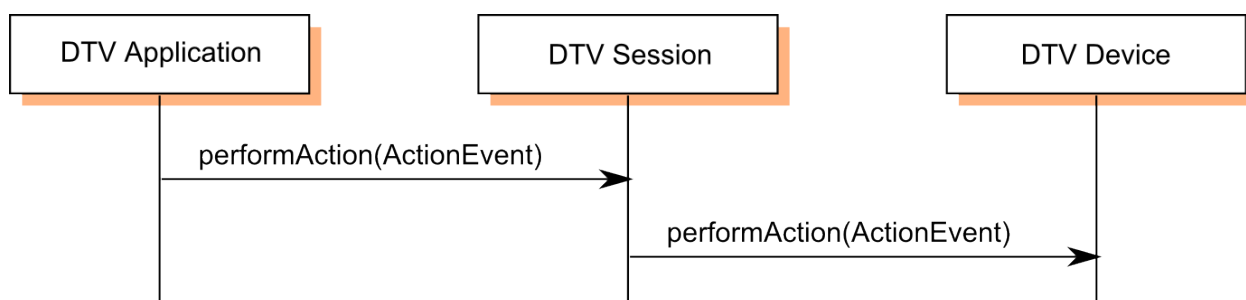


Slika 2.6: Organizacija DTV proširenja Android radnog okruženja

U okviru Android radnog okruženja osnovno sredstvo apstrakcije DTV podrške je DTV uređaj. DTV uređaj predstavlja programsku reprezentaciju uređaja televizijske podrške i sadrži neophodne informacije o mogućnostima i stanju datog uređaja. Pomenuti uređaj može biti podsistem u okviru samog Android uređaja, kod integrisanih DTV prijemnika, ako DTV uređaj

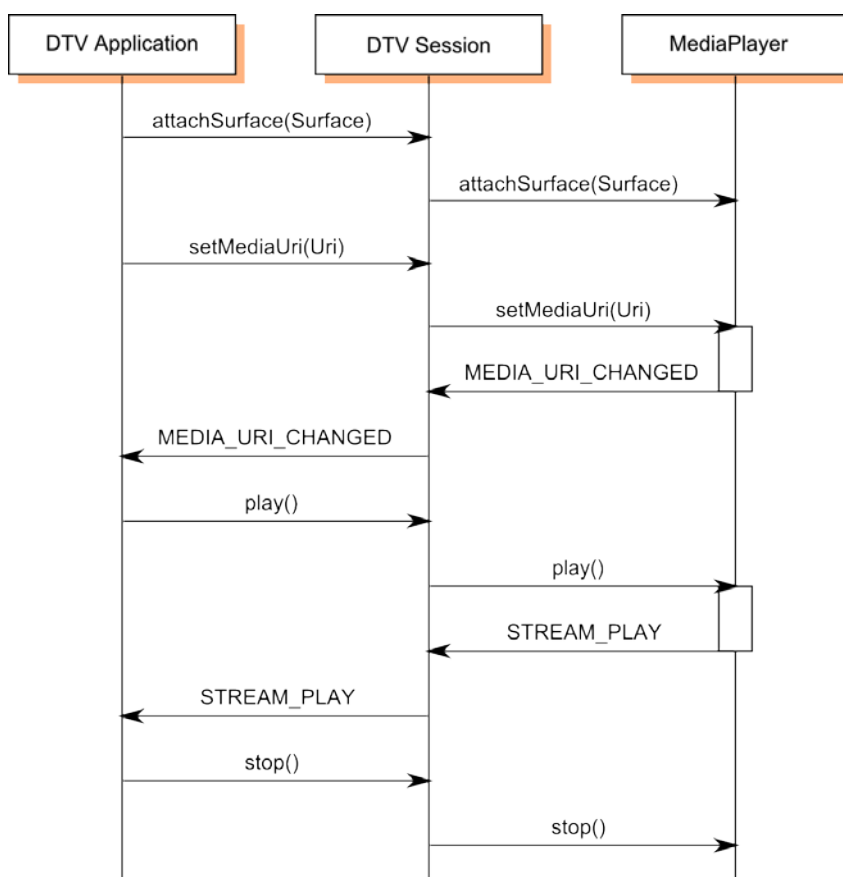
posедуje odgovarajuće fizičke komponente za prijem i reprodukciju televizijskog signala ili pak spoljni uređaj kao što je slučaj kod implementacije posredničkih rešenja. Spoljni uređaj se koristi kao izvor video signala, a upravljanje njime je omogućeno putem ulazne (npr. *Infrared/Bluetooth*) sprege. U okviru sistema može postojati više DTV uređaja sa televizijskom podrškom, međutim u datom trenutku reprodukcija može biti aktivna samo sa jednog uređaja.

Arhitektura podsistema za integraciju DTV podrške Android radnog okruženja sastoji se od DTV aplikacije, DTV servisa i modula za upravljanje DTV uređajem. Za svaki DTV uređaj mora postojati odgovarajući modul za upravljanje datim DTV uređajem. Pri integraciji DTV podrške neophodno je implementirati modul za upravljanje DTV uređajem i Android servis, koji ostatku sistema obezbeđuje pristup realizovanoj upravljačkoj komponenti.



Slika 2.7: Tok korisničkog unosa

DTV aplikacija je komponenta zadužena za ostvarivanje korisničke grafičke sprege, prikaz DTV sadržaja i korisnička interakcija. Za interakciju sa uređajem baziranim na Android platformi korisnik na raspolaganju ima razne ulazne računarske uređaje, poput tastature, miša i ekrana osjetljivog na dodir. Korisničke komande ka modulu za upravljanje DTV uređajem prosleđuju se u vidu instance Action Event klase. Action Event na uniforman način objedinjuje značenje korisničkih unosa sa svih dostupnih ulaznih uređaja. Slika 2.7 prikazuje tok prosleđivanja korisničke komande ka modulu za upravljanje DTV uređajem.

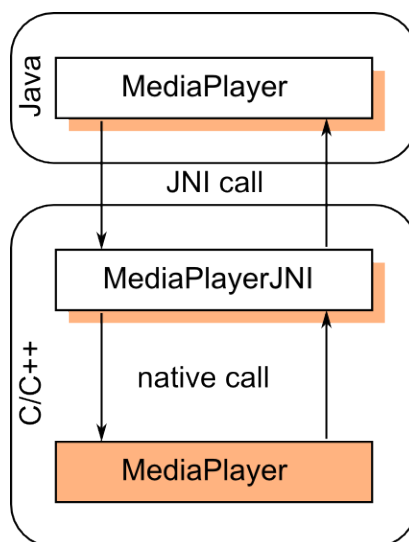


Slika 2.8: Media Player signalizacija

Kao posrednik između DTV aplikacije i modula za upravljanje DTV uređajem postoji DTV sesija. DTV sesija povezuje DTV aplikaciju sa odgovarajućim modulom za upravljanje DTV uređajem. Takođe, DTV sesija sadrži instancu Media Player programske klase. Osnovna uloga Media Player programske klase je upravljanje reprodukcijom audio i video sadržaj, za čiju realizaciju je upotrebljen asinhron pristup (Slika 2.8). Media Player takođe ima izuzetno bitnu ulogu u procesu upravljanja resursima, reč je o komponentama video i audio toka fizičke arhitekture sistema. Upravljanje resursima je od velike važnosti u slučaju televizijskih prijemnika, jer za razliku od prenosnih uređaja postoji potreba za stalnom reprodukcijom televizijskog sadržaja. U toku reprodukcije televizijskog programa svaka druga reprodukcija video i audio sadržaja (npr. PIP, PAP, reprodukcija video sadržaja sa interneta i reprodukcija lokalnog video sadržaja) ima nepredvidljiv rezultat bez mogućnosti upravljanja resursima. Media Player sadrži klijentski deo mehanizma upravljanja resursima, čime se garantuje pravilna razmena resursa pri pokretanju, odnosno zaustavljanju reprodukcije video sadržaja. Opisanom organizacijom Media Player se vezuje za DTV uređaj, što je razlog da DTV aplikacija koristi Surface View programsku klasu, čija je uloga da napravi prozor u grafičkoj ravni kroz koji se vidi sadržaj video ravni. Video View kao standardna komponenta programske podrške za prikaz

video sadržaja nije upotrebljena u okviru podsistema televizijske podrške jer ima direktnu vezu sa Media Player instancom.

Slika 2.9 prikazuje organizaciju Android Media Player rešenja. Celokupna funkcionalnost Media Player modula realizovana je platformski zavisnom programskom podrškom, programski jezik C/C++. Za potrebe Android Java radnog okruženja realizovana je Media Player programska klasa, koja uz pomoć JNI programske sprege pristupa odgovarajućim funkcionalnostima za upravljanje reprodukcijom video i audio sadržaja.



Slika 2.9: Android Media Player programska organizacija

Centralna komponenta podsistema za integraciju televizijske podrške je DTV servis. DTV servis u toku rada vrši pronalaženje dostupnih DTV uređaja i vodi računa o njihovim mogućnostima. Kada DTV aplikacija inicira prikaz DTV sadržaja DTV servis instancira kreira novu sesiju i pridružuje joj odgovarajući modul za upravljanje DTV uređajem u skladu sa potrebama reprodukcije DTV sadržaja.

3. Teorijske osnove

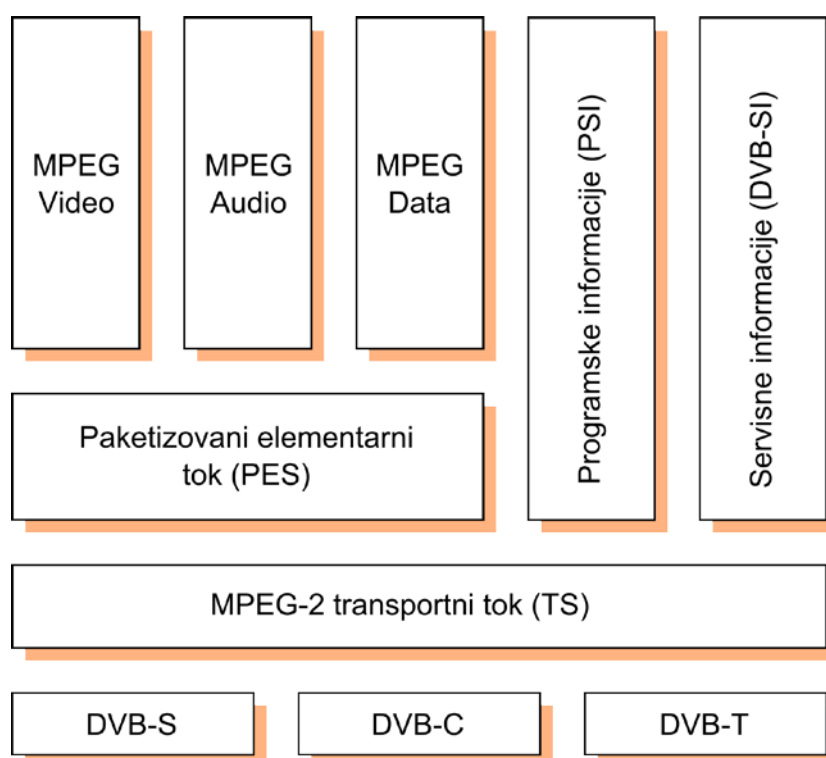
3.1 Digitalna televizija

Prva ozbiljna promena u televiziji, u tehničkom smislu, nastaje uvođenjem digitalne televizije. DTV unosi izmene u organizaciji i prenosu podataka, što je glavni razlog za sporu primenu u praksi. DVB je internacionalno prihvaćen DTV standard kojim su definisani različiti načini prenosa podataka, u zavisnosti od tipa prenosnog puta: DVB-S za satelitsku, DVB-C za kablovsku i DVB-T za zemaljsku televiziju. Takođe, postoji ATSC kao standard koji se koristi u Severnoj Americi i ISDB kao japanski DTV standard. Svi navedeni standardi se oslanjaju na MPEG-2 standard kojim je definisan način kodovanja i prenosa video sadržaja i njemu pridruženog audio sadržaja.

MPEG-2 dozvoljava da se video signal standardne rezolucije, za koji je potrebna brzina prenosa podataka 270 Mb/s, kompresuje na nivo od 7 Mb/s. Zahvaljujući kompresiji moguće je na jednoj frekvenciji vršiti emitovanje nekoliko različitih televizijskih servisa, moguće je i postojanje različitih video i audio signala za isti televizijski servis. Na primer, DVB-S ostvaruje brzinu prenosa podataka od oko 38 Mb/s po frekvenciji, gde je moguće vršiti emitovanje 6 televizijskih servisa brzine prenosa 6 Mb/s [16].

DVB prenos podataka je realizovan uz pomoć MPEG-2 prenosnog toka podataka (Slika 3.1). Iako je MPEG-2 pre svega kreiran za prenos video i audio podataka, može se upotrebiti za prenos druge vrste podataka, na primer prevodi, teletext itd. U sastav MPEG-2 transportnog toka ulaze i programske informacije: tabela asocijacije programa (PAT), tabela uslovnog pristupa (CAT), tabela opisa transportnog toka (TSDDT) i tabela mapiranja programa (PMT). DVB standardom uvedene su servisne informacije: tabela informacija o mreži (NIT), tabela

informacija o servisu (SDT), tabela rasporeda televizijskog programa (EIT) i tabela informacija o vremenu i datumu (TDT).



Slika 3.1: Organizacija DVB prenosnog toka podataka

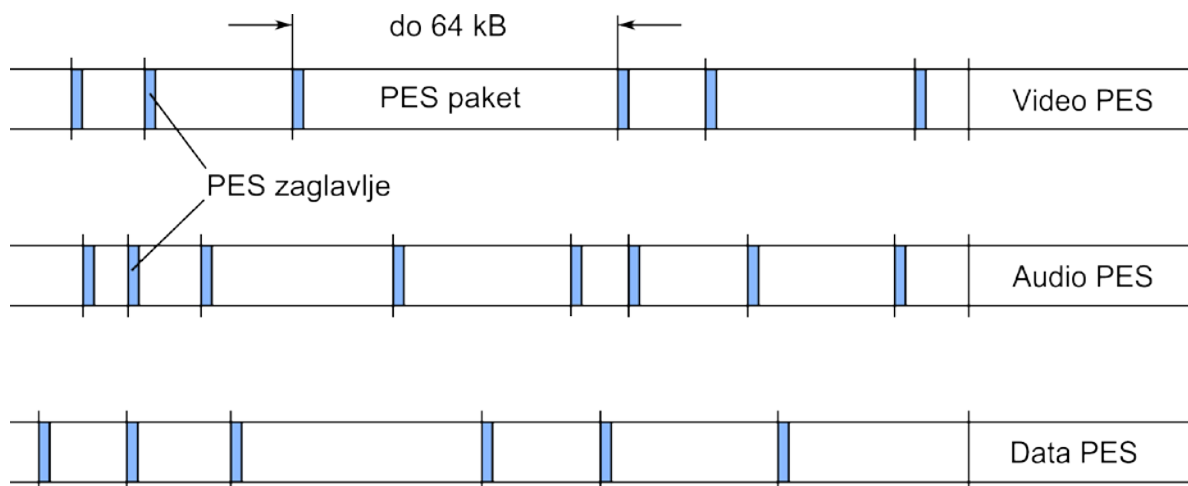
S obzirom da je učestanost pristizanja servisnih informacija reda veličine 100 ms (Tabela 3.1) postoji potreba za obradom podataka u realnom vremenu. Uslov za obradu podataka u realnom vremenu je da implementacija televizijske podrške bude realizovana na niskom nivou apstrakcije.

Tabela	Učestanost
PAT	100 ms
PMT	100 ms
EIT	2 s
SDT	2 s
TSDT	10 s
NIT	10 s
TDT	30 s
CAT	nedefinisano

Tabela 3.1: Minimalne učestanosti servisnih informacija [17]

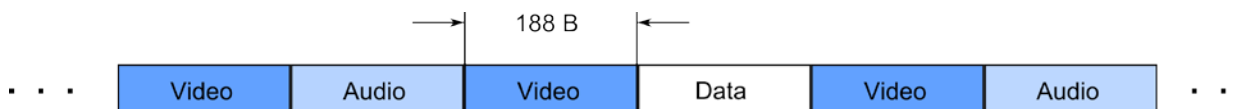
Kompresovani video i audio signali, kao i drugi podaci u MPEG-u nazivanju se elementarnim tokovima podataka. Elementarni tokovi podeljeni su na pakete promenljive

veličine do 64 kB, koje nazivamo paketizovanim elementarnim tokovima (PES). Informacija o veličini paketa dostupna je u zaglavlju paketa veličine 6 bajtova (Slika 3.2).



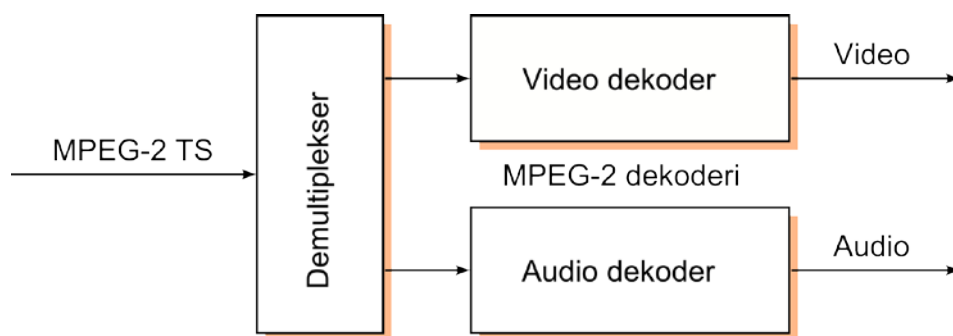
Slika 3.2: MPEG-2 paketizovani elementarni tokovi podataka

PES paketi zbog svoje relativno velike veličine nisu pogodni za prenos na daljinu. Iz pomenutog razloga od PES paketa formiraju se manji paketi veličine 188 bajtova, takozvani transportni paketi. Pri slanju transportnih paketa vrši se njihov vremenski multipleks (Slika 3.3). U slučaju kada se na istoj frekvenciji emituje nekoliko različitih televizijskih servisa svi elementarni tokovi multipleksiraju se na opisan način.



Slika 3.3: MPEG-2 transportni tok podataka

Za potrebe reprodukcije DTV prijemnik mora posedovati demultiplekser za izdvajanje video i audio paketa iz transportnog toka. Takođe, neophodne komponente DTV prijemnika su video i audio dekođer za dekompresiju video i audio sadržaja (Slika 3.4).



Slika 3.4: DTV prijemnik

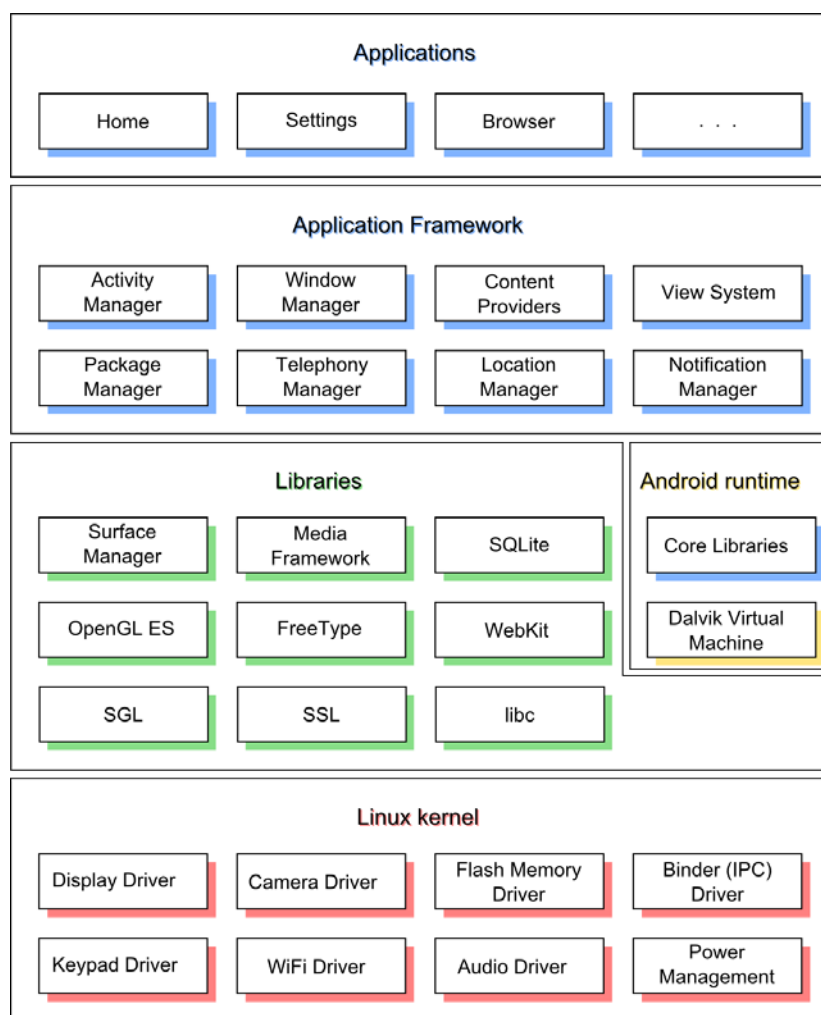
3.2 Teletext i prevodi

Teletext je televizijski servis stvoren u Velikoj Britaniji 70-tih godina dvadesetog veka. Osmišljen je u vreme analogne televizije, tako da se informacije šalju u vreme praznog vertikalnog intervala. Teletext sadržaj je podeljen u stranice, a svaka stranica sastoji se od 24 linije sa po 40 karaktera. Pruža niz informacija baziranih na tekstu, kao što su vesti, vremenska prognoza i raspored televizijskog programa. Takođe, posredstvom teletexta, moguće je obezbediti prikaz prevoda.

U digitalnoj televiziji, DVB standard, teletext i prevodi emituju se u okviru toka podataka [18]. Analognom televizijom brzina protoka podataka za teletext bila je limitirana na 100 – 150 kb/s. Radi usklađenosti sa već postojećim televizijskim prijemnicima zadržane su pomenute brzine prenosa podataka, iako ih je digitalnom televizijom moguće prevazići.

3.3 Android

Android je operativni sistem otvorenog programskog koda, namenjen prenosnim uređajima kao što su mobilni telefoni. Kod prenosnih uređaja najbitnija je autonomija, odnosno mala potrošnja energije, iz tog razloga Android ima agresivnu politiku upravljanja potrošnjom energije. Zbog potrebe uštede energije u prenosne uređaje ugrađuju se ekonomičnije centralne procesorske jedinice sa slabijim performansama u odnosu na rešenja koja se upotrebljavaju u personalnim računarima. I količina radne i masovne memorije može se smatrati ograničenim resursom kod prenosnih uređaja. Android je optimizovan za rad u opisanim uslovima, tako da se nedostatak resursa ne primeti pri normalnom radu. Ustaljeno pravilo za operativne sisteme namenjene prenosnim uređajima, tako i za Android, je da se korisnička aplikacija potpuno isključuje tek pri potrebi za dodatnom memorijom. Ovo znači da pri izlasku iz aplikacije ista ostaje u memoriji, ali u opštem slučaju ne troši procesorsko vreme. Čuvanjem aplikacije u memoriji postiže se brže pokretanje aplikacije, čime korisnik stiče utisak boljeg odziva sistema.



Slika 3.5: Programska organizacija Android operativnog sistema

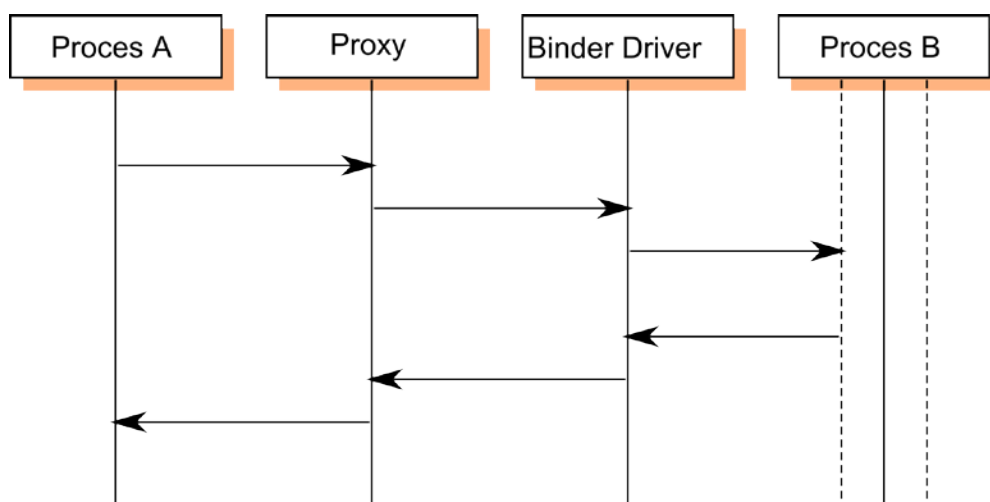
Zahvaljujući apstrakciji virtualne mašine i programskom jeziku Java, korisničke aplikacije napravljene za Android operativni sistem funkcionišu na širokom spektru uređaja, različitih proizvođača, performansi i mogućnosti. Televizijski uređaji imaju mnoge razlike u odnosu na prenosne uređaje, što za posledicu ima nemogućnost upotrebe svih korisničkih aplikacija namenjenih Android operativnom sistemu. S obzirom da se kod prenosnih uređaja ulazno izlazna sprega najčešće ostvaruje posredstvom ekrana osjetljivog na dodir, korisničke aplikacije zahtevaju određena prilagođenja za televizijske prijemnike gde je osnovni ulazni uređaj daljinski upravljač baziran na infracrvenoj sprezi. Televizijski uređaji najčešće nemaju mogućnost bežičnog umrežavanja, u smislu lokalnih računarskih mreža i telekomunikacionih sistema. Takođe, prenosni uređaji poseduju širok spektar senzorskih uređaja, koji nisu na raspolaganju televizijskim prijemnicima, kao što su akcelerometar, giroskop, termometar, GPS senzor itd. Za implementaciju korisničkih aplikacija na raspolaganju je Java radno okruženje, koje pored skupa programskih klasa sadrži sistemske servise čije usluge su na raspolaganju pojedinačnim aplikacijama. Android poseduje posebnu Java virtualnu mašinu registarske arhitekture Dalvik, za

razliku od drugih Java virtuelnih mašina magazinske arhitekture. Dalvik takođe koristi poseban format datoteka koji je optimizovan za upotrebu na prenosnim uređajima.

Ispod Java radnog okruženja i virtuelne mašine, u smislu apstrakcije, nalaze se programske biblioteke i servisi implementirani programskim jezikom C/C++. Sve programske komponente u ovom apstraktnom nivou moraju biti prevedene u odnosu na Bionic. Bionic je C biblioteka realizovana za potrebe Android operativnog sistema i karakterišu je sledeće osobine: BSD licenca, mala memorijska veličina, brzina (osmišljena je za centralne procesorske jedinice relativno niskog radnog takta). Osnovu Androida čini Linux kernel, koji je proširen odgovarajućim komponentama za rad u kontekstu prenosnih uređaja.

3.4 Binder

Binder radno okruženje je mehanizam za komunikaciju među procesima operativnog sistema. Android operativni sistem integriše Binder u vidu proširenja Linux kernela, jer je jedino u kernel memorijskom prostoru moguć pristup deljenoj memoriji. Binder je prvobitno razvijen pod imenom OpenBinder, kao sredstvo povezivanja dva procesa operativnog sistema međusobnim pozivom funkcija i razmenom podataka. Sa stanovišta programera najzanimljivija osobina Bindera je što se metode objekta iz drugog procesa mogu pozivati kao metode lokalnog objekta. Pomenuta mogućnost je realizovana sinhronim pozivima, odnosno pozivajući proces je blokiran dok ne dobije odgovor. Za pozive iz Java programskog jezika realizovan je poseban programski jezik (eng. *Android Interface Definition Language – AIDL*), kojim se definiše programska sprega koja je dostupna posredstvom Bindera.



Slika 3.6: Tok Binder komunikacije

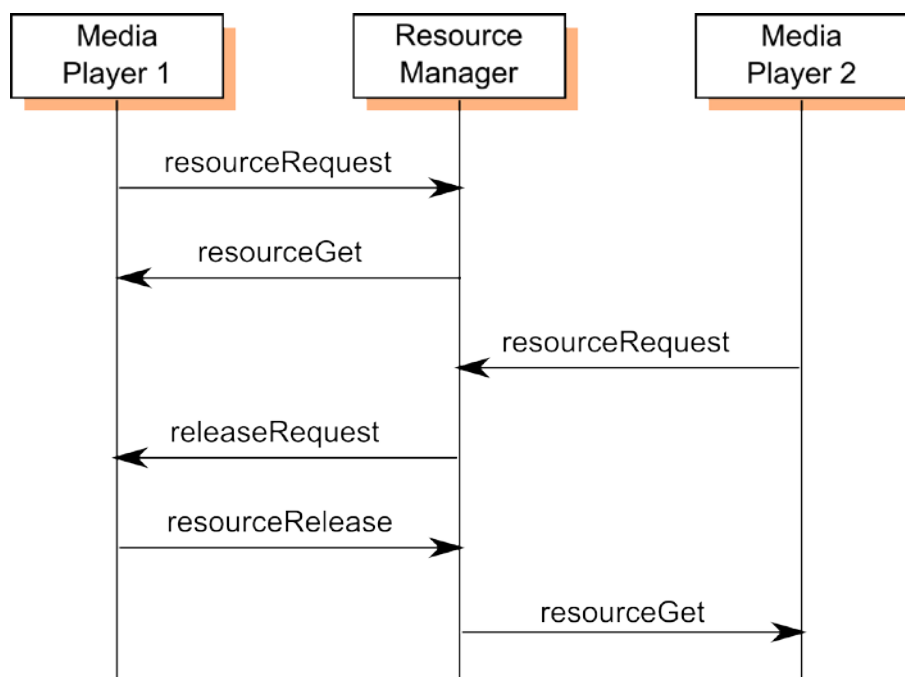
Binder komunikacija realizovana je po modelu klijent poslužilac [14]. Slika 3.6 prikazuje tok Binder komunikacije. Binder klijent inicira komunikaciju i čeka odgovor Binder poslužioca. Proces A je klijent i sadrži objekat posrednik, koji implementira komunikaciju sa Binder kernel

drajverom. Proces B je poslužilac koji sadrži posebnu programsku nit za svakog klijenta koji inicira komunikaciju.

3.5 Upravljanje resursima

Neizostavan deo multimedijalnih sistema predstavlja mogućnost upravljanja resursima. Pod resurse podrazumevamo fizičke komponente sistema zadužene za reprodukciju video i audio signala. Iako moderni televizijski prijemnici imaju mogućnost reprodukcije više video signala istovremeno, broj tokova za reprodukciju je ograničen i rezultat je nepredvidljiv u slučaju pokušaja zauzimanja već zauzetih resursa. Na primeru sistema sa mogućnošću reprodukcije jednog video signala, ako je u toku reprodukcija televizijskog programa, a korisnik želi da pokrene reprodukciju lokalnog video sadržaja, neophodno je zaustaviti ranije započetu reprodukciju i osloboditi potrebne resurse. Takođe, pri zaustavljanju reprodukcije lokalnog video sadržaja, resurse treba vratiti reprodukciji televizijskog programa.

PiP i PaP su specifični slučajevi prethodno opisanog, sa tom razlikom da platforma mora posedovati bar dva video toka. S obzirom na postojanje više video tokova, pri zahtevu za resursima dodeljuje se prvi slobodan tok. Za situacije zauzetosti svih raspoloživih video tokova može postojati stepen prioriteta, gde je najprioritetnija reprodukcija televizijskog sadržaja.



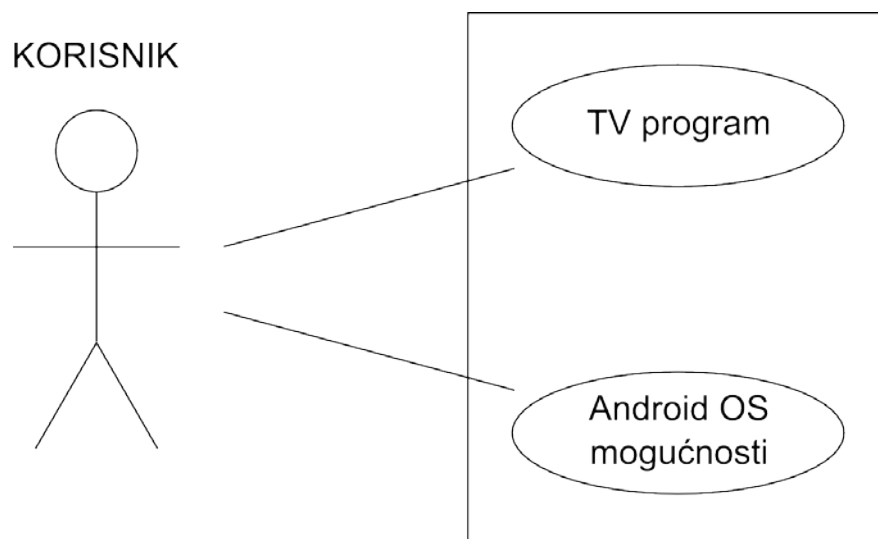
Slika 3.7: Tok upravljanja resursima

Za uspešno upravljanje resursima, komponenta zadužena za upravljanje mora biti realizovana na nivou operativnog sistema, u tipičnom slučaju kao servis kojem bi drugi procesi pristupali putem međuprocenke komunikacije. Slika 3.7 sadži prikaz dijagram toka međuprocenke komunikacije upravljanja resursima. Sa stanovišta Android operativnog sistema

prenosni uređaji ne poseduju mogućnost upravljanja resursima, jer ne postoji potreba za stalnom reprodukcijom video sadržaja. Za razliku od prenosnih uređaja televizijski prijemnici neposredno nakon uključivanja pokreću prikaz televizijskog programa, koji se prekida jedino u slučaju reprodukcije video sadržaja sa drugog izvora. Da bi se obezbedila stalna reprodukcija Android operativni sistem mora biti proširen mogućnošću upravljanja resursima.

4. Analiza problema i postavke ciljeva

Za potrebe ovog rada treba izvršiti integraciju DTV podrške u Android operativni sistem, kako bi se isti mogao upotrebiti za implementaciju DTV prijemnika. DTV prijemnicima smatramo uređaje namenjene za prijem i reprodukciju televizijskog sadržaja digitalne televizije. Očekivani rezultat integracije televizijske podrške je dobijanje operativnog sistema čija je primarna funkcionalnost prikaz televizijskog sadržaja. Potrebno je da postoji stalna reprodukcija televizijskog programa, a ostalim mogućnostima Android operativnog sistema pristupa se posredstvom korisničke sprege.



Slika 4.1: Dijagram slučaja korišćenja televizijske podrške

Tabela 4.1 sadrži pregled osnovnih funkcionalnosti koje bi svaki DTV prijemnik trebao da poseduje. To su funkcionalnosti koje pružaju mogućnost upravljanja raspoloživim servisima i podešavanje kvaliteta slike i tona.

Funkcionalnost	Opis
Pretraga televizijskih servisa	Pretraga frekventnog spektra za raspoloživim televizijskim servisima.
Manipulacija listom televizijskih servisa	Izbor televizijskog servisa za reprodukciju i uređivanje liste servisa.
Podešavanje kvaliteta slike	Mogućnost promene parametara koji utiču na kvalitet prikaza video sadržaja, kao što je osvetljenje, kontrast itd.
Podešavanje kvaliteta tona	Mogućnost promene parametara koji utiču na kvalitet audio sadržaja, kao što je jačina tona.

Tabela 4.1: Osnovne funkcionalnosti DTV prijemnika

Tabelom Tabela 4.2 dat je pregled naprednih funkcionalnosti DTV prijemnika. Napredne funkcionalnosti korisniku pružaju mogućnost pristupa dodatnim informacijama i povoljno utiču na kvalitet servisa ponuđenih korisniku.

Funkcionalnost	Opis
Teletext	Prikaz teletext sadržaja (ETSI ETS 300 706)
EPG	Prikaz elektornskog programskog vodiča (ETSI ETS 300 707)
Prevod	Prikaz raspoloživog tekstualnog prevoda (ETSI ETS 300 743)
PVR	Mogućnost snimanja televizijskog sadržaja

Tabela 4.2: Napredne funkcionalnosti DTV prijemnika

Teletext pruža razne tekstualne informacije, kao što su vesti, vremenska prognoza, ili raspored emitovanja televizijskog programa. EPG sadrži detaljan pregled rasporeda televizijskog programa za određeni period, sa mnoštvom dodatnih informacija kao što su tip i opis emitovanog sadržaja. Standardi digitalne televizije pružaju mogućnost emitovanja EPG informacija od strane televizijskih servisa, posredstvom takozvanih EIT sekcija [16]. PVR je napredna funkcionalnost DTV prijemnika koja korisniku pruža mogućnost snimanja televizijskog programa u svrhu odloženog gledanja.

Važan deo Android operativnog sistema predstavlja Android Java radno okruženje. Android Java radno okruženje sadrži mehanizme i elemente za upravljanje uređajem i realizaciju korisničke sprege nezavisne od tipa ili modela samog uređaja. Pošto integracija televizijske podrške zahteva postojanje korisničke sprege, integraciju je neophodno sprovesti na nivou Android Java radnog okruženja. Takođe, postojanje televizijske podrške u okviru Android Java radnog okruženja pruža mogućnost trećim licima razvoj sopstvenih računarskih aplikacija koje koriste mogućnosti televizijske podrške.

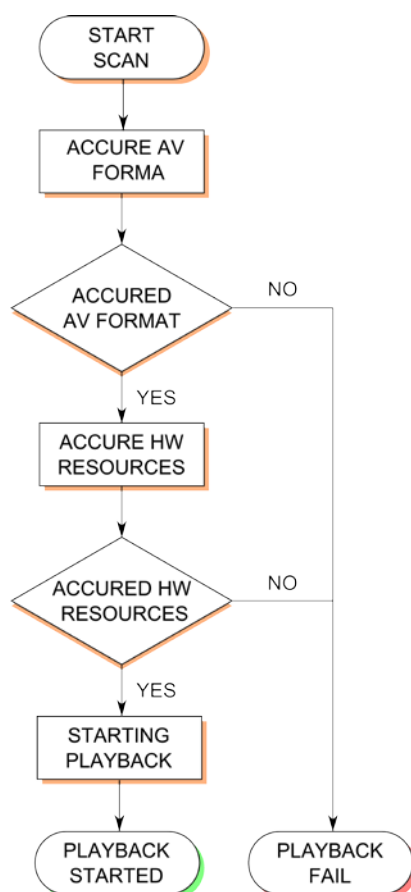
5. Opis rešenja

Android radno okruženje predstavlja dobru polaznu tačku za integraciju televizijske podrške, jer poseduje odgovarajuće mehanizme za upravljanje i prikaz televizijskog sadržaja. Osnovna komponenta podsistema televizijske podrške je DTV MDW, koja sadrži potrebnu DTV funkcionalnost, dok je ostatak podsistema neophodan za integraciju DTV podrške u Android radno okruženje. DTV MDW uzimamo kao već postojeće rešenje, realizovano u programskom jeziku C/C++ na najnižem nivou apstrakcije celokupnog podsistema DTV podrške i neće biti opisa detalje njegove implementacije.

5.1 Media Player (upravljanje resursima)

Media Player kao programski modul zadužen za upravljanje reprodukcijom video i audio sadržaja predstavlja kritičnu tačku za proces upravljanja resursima. Pre započinjanja reprodukcije Media Player mora izvršiti akviziciju odgovarajućih resursa fizičke arhitekture sistema, kao i njihovo oslobađanje po završetku reprodukcije. Preuzeto DTV MDW rešenje sadrži Media Player implementaciju, stoga Media Player u okviru Android radnog okruženja podsistema televizijske podrške mora biti zamenjen implementacijom koja će vršiti signalizaciju ka DTV MDW Media Player rešenju.

U slučaju prikaza DTV servisa Media Player mora pribaviti informacije o formatu video i audio sadržaja na osnovu analize PMT SI tabele. Format reprodukovanog sadržaja je neophodan sa stanovišta reprodukcije i upravljanja resursima, kako bi Media Player mogao zahtevati resurse fizičke arhitekture sa odgovarajućim sposobnostima za ciljanu reprodukciju sadržaja. Slika 5.1 sadrži prikaz blok dijagram algoritma pokretanja reprodukcije video i audio sadržaja DTV servisa.



Slika 5.1: Blok dijagram algoritma pokretanja reprodukcije DTV servisa

Iniciranju prikaza DTV servisa prethode dva koraka, prikupljanje informacija o formatu video i audio sadržaja i zauzeće odgovarajućih resursa fizičke arhitekture sistema, potrebnih za reprodukciju datog sadržaja. U slučaju greške u jednom od dva pomenuta koraka, pokušaj reprodukcije se smatra neuspehim i nastavak procedure se prekida.

U nastavku teksta dat je primer programske podrške kojim se vrši zahtev za odgovarajućim resursima potrebnim pri reprodukciji video sadržaja. U opštem slučaju zahtev sadrži skup potrebnih komponenti video toka fizičke arhitekture i skup odgovarajućih mogućnosti nevedenih komponenti. Iz datog primera možemo videti da je upućivanje zahteva asinhrona operacija koja zahteva obradu dobijenog odgovora.

```

void requestResources(sp<MetaData> videoMeta){
    sp<RMRequest> request = new RMRequest();
    request->addComponent(kComponentVideoDecoder);
    request->addCapability(kRMCapabilityVideoCoding,
        videoMeta->getCodingType());
    ....
    mRMClient->requestResources(request);
}
  
```

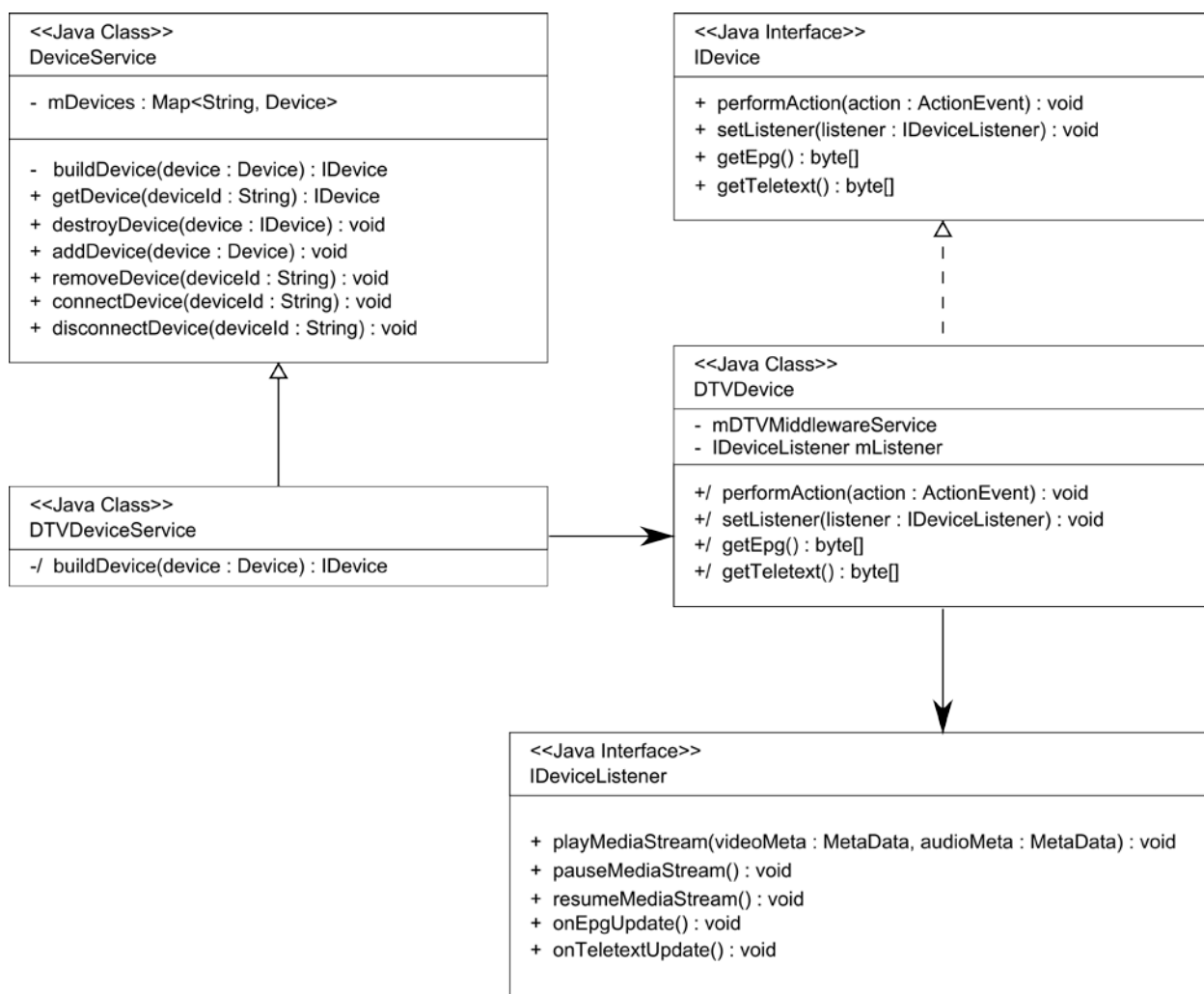
Obrada odgovora na prethodno upućeni zahtev sastoji se od provere da li su svi zahtevani resursi dobijeni. Odgovor je dobijen u vidu `RMResponse` instance kontejnerske programske klase, koja sadrži sve dobijene komponente u skladu sa zahtevima. Ukoliko neka od zahtevanih komponenti nije prisutna u odgovoru, smatra se da tražena komponenta nije na raspolaganju što za posledicu ima nemogućnost pokretanja reprodukcije. Svi resursi pobrojani objektom programske klase `RMResponse` smatraju se zauzetim dok dati Media Player ne uputi zahtev za oslobađanje resursa.

```
void handleRequestResponse(const sp<RMResponse>
                           &response){
    Resource res;
    if(response->getResource
        (kComponentVideoDecoder, &res)){
        ...
    }
}
```

Oslobađanje resursa može biti inicirano i zbog potreba druge reprodukcije, u tom slučaju Media Player dobija zahtev za oslobađanje resursa. S obzirom da je Media Player i mehanizam upravljanja resursima realizovan na sistemskom nivou, nema bojazni da može doći do ne oslobađanja resursa od strane aplikacija korisničke programske sprege. Ukoliko dati Media Player u predviđenom vremenskom roku ne odgovori na zahtev, resursi se smatraju slobodnim za dalju upotrebu.

5.2 Modul za upravljanje DTV uređajem

Integracija DTV programske podrške izvršena je u vidu Android servisa, komponenta programske aplikacije Android Java radnog okruženja. Nasleđivanjem `Service` programske klase njena funkcionalnost je proširena instanciranjem modula za upravljanje DTV uređajem. Modul za upravljanje DTV uređajem implementira programsku spregu DTV podrške ka ostatku podsistema integracije DTV podrške Android Java radnog okruženja. Funkcionalnost DTV podrške ostvaruje se korišćenjem usluga Android servisa DTV podrške (`DTV Middleware Service`) posredstvom mehanizma međuprocene komunikacije. Slika 5.2 predstavlja prikaz međusobnog odnosa programskih klasa podsistema DTV podrške na nivou Java programskog jezika.

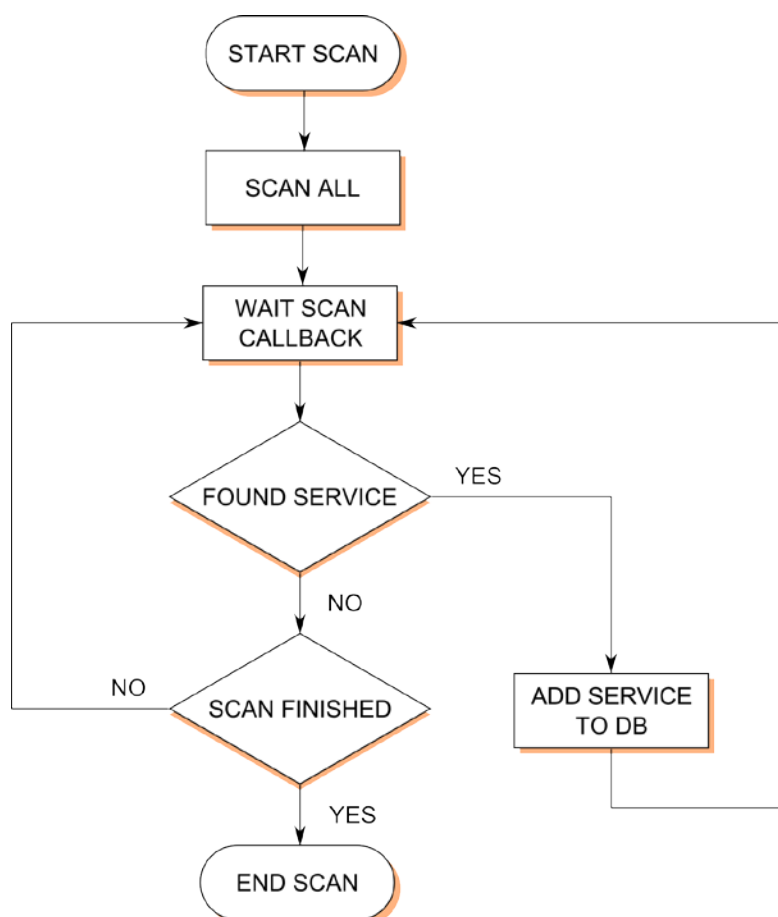


Slika 5.2: UML dijagram klasa sprezanja DTV podrške

5.3 Pretraga dostupnih DTV servisa

Tipičan slučaj pretrage dostupnih DTV servisa podrazumeva osvežavanje celokupne liste servisa pronalaženjem svih postojećih servisa, prolaskom kroz ceo frekventni spektar u slučaju DVB-T standarda. Ovakav pristup je vremenski zahtevniji, ali mnogo jednostavniji za korisnika nego pojedinačna pretraga servisa. Kod pojedinačne pretrage od korisnika se zahteva odgovarajući skup parametara na osnovu kojih se vrši pretraga, kao što je na primer frekvencija.

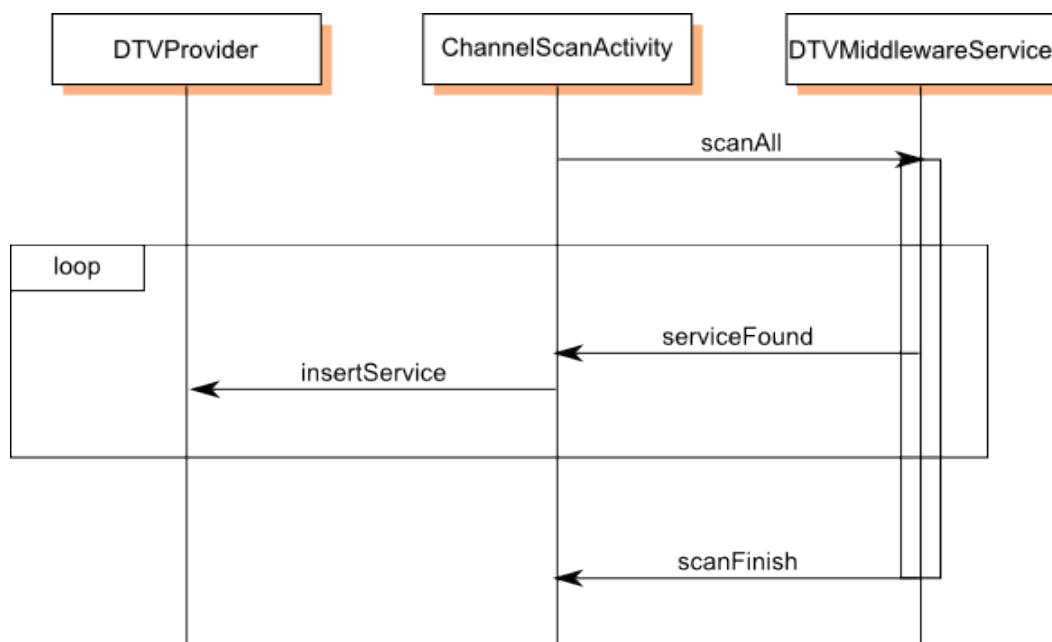
Preuzeto DTV MDW rešenje podržava pojedinačnu i celokupnu pretragu DTV servisa. Zbog jednostavnije primene i implementacije izabrana je celokupna pretraga servisa (Slika 5.3). Pretraga celokupnog skupa servisa sastoji se od poziva instrukcije za pretragu servisa DTV podrške i upisa pronađenih servisa u bazu podataka DTV podrške. Iako je procedura pretrage servisa digitalne televizije daleko komplikovanija od opisanog rešenja, celokupna procedura nije opisana jer je realizovana u okviru DTV MDW podsistemu programske podrške, koja je preuzeta kao gotovo rešenje.



Slika 5.3: Blok dijagram algoritma pretrage DTV servisa

Za pretragu DTV servisa implementirana je posebna komponenta korisničke grafičke sprege (Channel Scan Activity). Realizovana komponenta grafičke korisničke sprege za pretragu DTV servisa pokreće se iz samog Android radnog okruženja pri potrebi za pretragom servisa. Android radno okruženje poseduje mogućnost pronalaženja odgovarajućih komponenti programske podrške u zavisnosti od željene akcije. Da bi data komponenta programske podrške mogla biti pronađena, za svaku komponentu prethodno moraju biti definisane sve podržane akcije od strane date komponente.

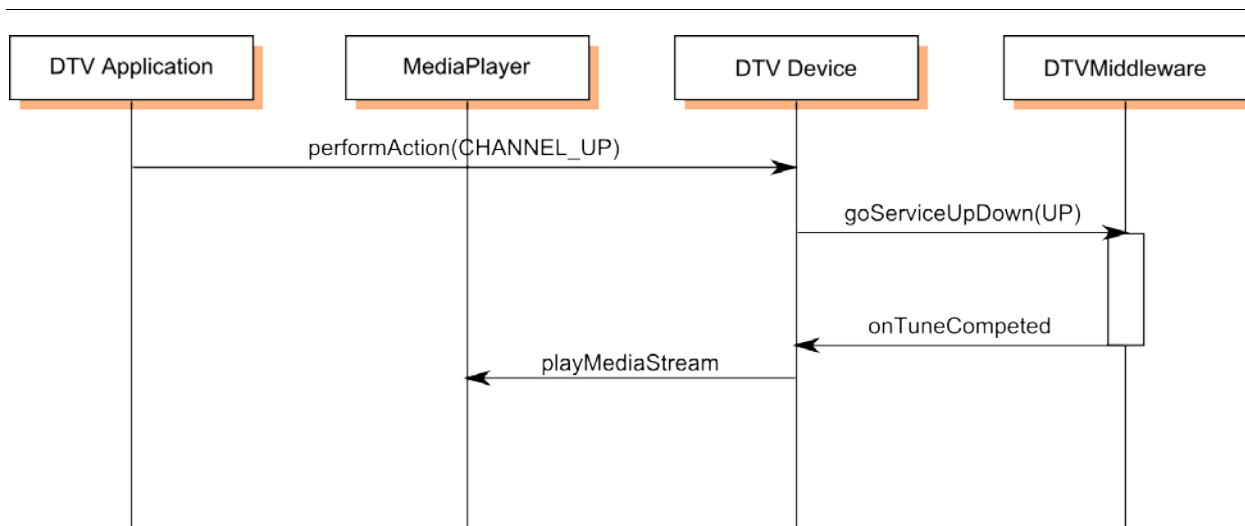
Pretraga servisa digitalne televizije oslanja se na mehanizam asinhronne programske podrške, jer je u pitanju vremenski zahtevna operacija. Posredstvom mehanizma asinhronne programske sprege komponenta korisničke sprege za pretragu servisa biva obaveštena o pronalaženju DTV servisa i završetku pretrage (Slika 5.4). Pored informacije o pronađenom servisu, DTV MDW dostavlja i sve postojeće podatke o datom servisu. Dostavljeni podaci koriste se za upis pronađenog servisa u bazu podataka DTV podrške.



Slika 5.4: Tok pretrage DTV servisa

5.4 Izbor prikaza DTV servisa

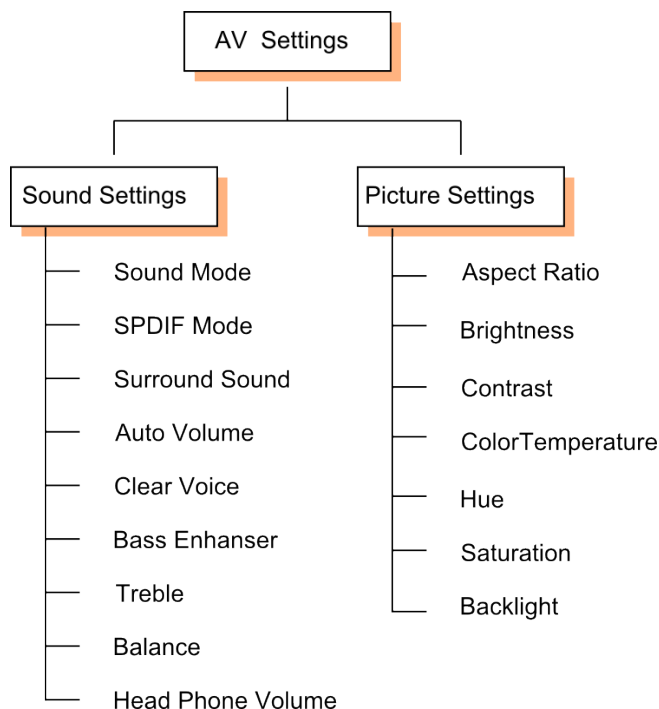
Kao i pretraga dostupnih DTV servisa izbor prikaza DTV servisa je asinhrona operacija, odnosno zbog relativno dugog vremenskog izvršenja implementirana je signalizacija o završetku operacije od strane DTV MDW rešenja. Promena ili izbor DTV servisa inicira se od strane korisnika preko DTV korisničke aplikacije. Korisnička aplikacija preko DTV aplikacije prosleđuje korisnički unos do instance DTV sesije koja je instancirana od strane DTV servisa. Svi pristigli korisnički unosi do DTV sesije bivaju prosleđeni uparenoj instanci modula za upravljanje DTV uređajem. Novo pristigli korisnički zahtev za izbor DTV servisa rezultira pozivom odgovarajuće metode za izbor DTV servisa DTV MDW implementacije. Pored sekvencijalne mogućnosti izbora servisa, prvi sledeći ili prvi prethodni servis iz liste servisa, postojeće DTV MDW rešenje podržava i mogućnost izbora servisa na osnovu njegovog rednog broja u listi servisa.



Slika 5.5: Tok izbora prikaza DTV servisa

5.5 Podešavanje kvaliteta slike i tona

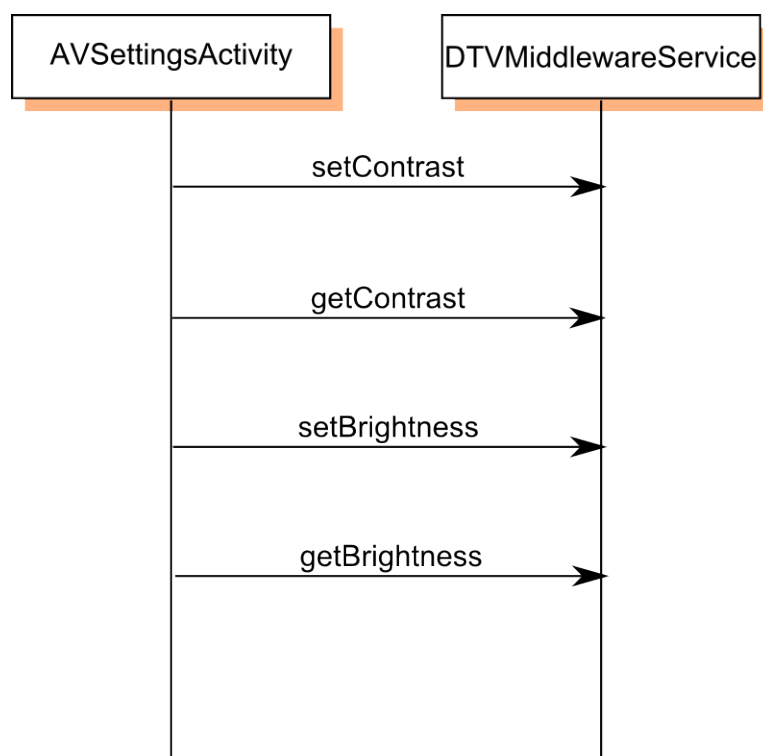
Za potrebe podešavanja kvaliteta slike i tona realizovana je posebna komponenta korisničke grafičke sprege. Korisnička grafička sprega podešavanja slike i tona integrisana je u okviru Android radnog odruženja na mesto predviđeno za ovakav vid podešavanja. Opcije korisničke programske sprege podeljene su u dve sekcije podešavanja slike i podešavanja zvuka (Slika 5.6).



Slika 5.6: Organizacija korisničke sprege za podešavanje slike i tona

Sve opcije korisničke grafičke sprege vrše jednostavno preslikavanje mogućnosti DTV MDW programske podrške, odnosno sva funkcionalnost vezana za podešavanje slike i tona DTV

MDW rešenja data je korisniku na upravljnje posredstvom korisničke grafičke sprege. Reč je sinhronoj realizaciji poziva programske sprege, gde je rezultat poziva odmah poznat (Slika 5.7).

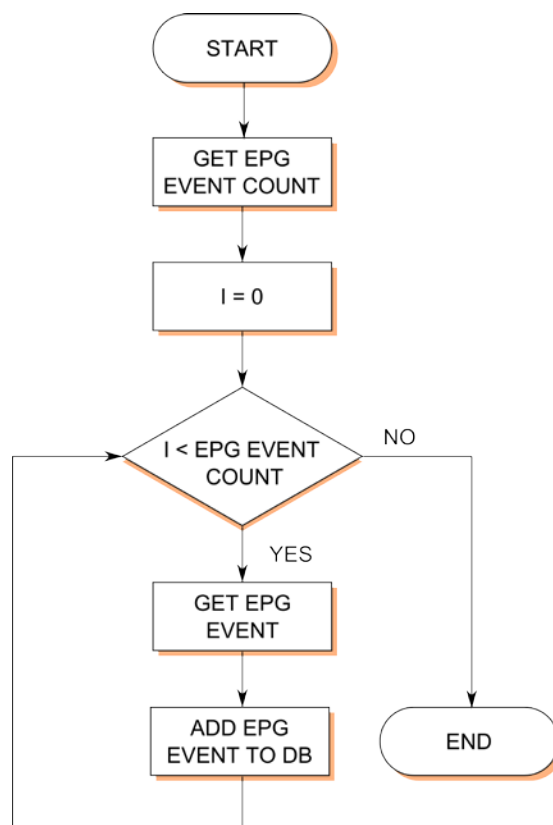


Slika 5.7: Tok promene i čitanja vrednosti parametara slike i tona

5.6 Prikaz elektronskog programskog vodiča

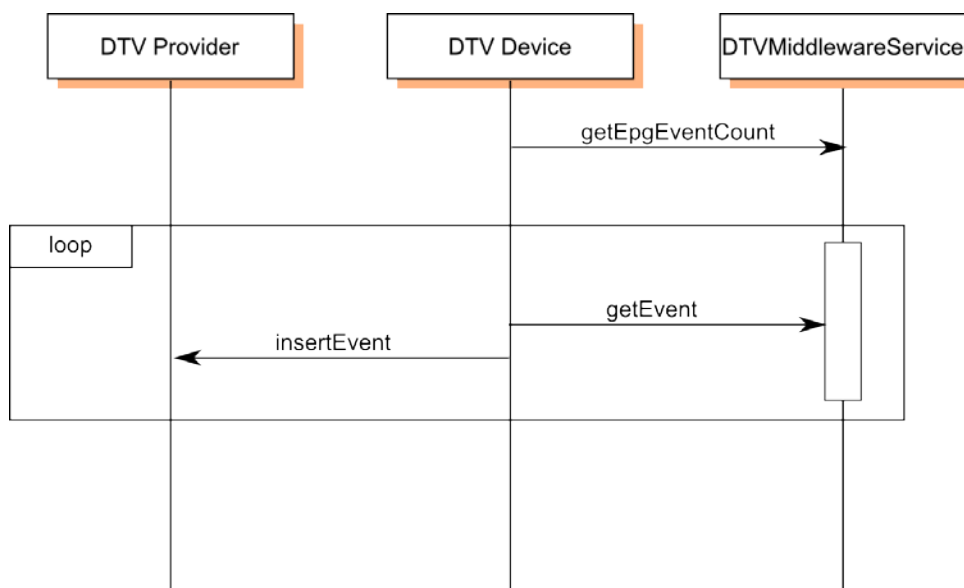
Prikazu i pretrazi elektronskog programskog vodiča prethodi akvizicija relevantnih EPG podataka. Postojeće DTV MDW rešenje poseduje implementiran mehanizam prikupljanja EPG podataka. Da bi omogućili pretragu EPG sadržaja od strane Android radnog okruženja već prikupljene EPG podatke treba upisati u bazu podataka DTV podrške, jer se pretraga zasniva na pristupu pomenutoj bazi podataka.

Programska sprega preuzete DTV MDW implementacije omogućava sekvencijalni pristup EPG podacima, koji su organizovani u vidu televizijskih događaja. Pristup pojedinačnom televizijskom događaju moguć je jedino po rednom broju događaja u odnosu na DTV servis o čijem događaju je reč. Realizacija osvežavanja Android DTV baze podataka, zbog opisane DTV MDW programske sprege, podrazumeva implementaciju prebrojive programske petlje sa brojem iteracija jednakim broju dostupnih televizijskih događaja po jednom DTV servisu (Slika 5.8). Takođe, DTV bazu podataka treba neprekidno osvežavati tokom rada DTV programske podrške kako bi raspolagali validnim EPG podacima.



Slika 5.8: Blok dijagram algoritma akvizicije EPG sadržaja

Implementacija upisa EPG sadržaja u DTV bazu podataka realizovana je u okviru DTV servisa. Međutim, EPG podatke mora obezbediti modul za upravljanje DTV uređajem implementacijom odgovarajućih metoda definisane programske sprege. Modul za upravljanje DTV uređajem određuje postojanje novih televizijskih događaja o čemu obaveštava DTV service, koji preuzima nove događaje i upisuje ih u bazu podataka (Slika 5.9).



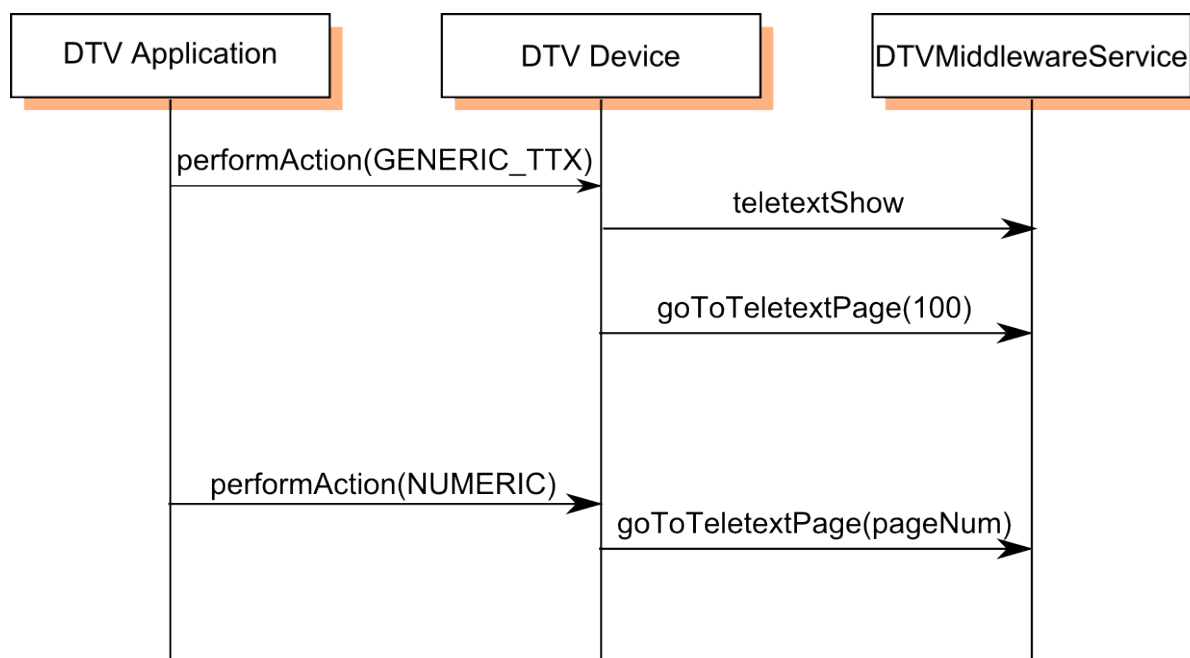
Slika 5.9: Tok akvizicije EPG sadržaja

Za potrebe prikaza EPG sadržaja realizovana je posebna komponenta grafičke korisničke sprege koja se prikazuje kad modul za upravljanje DTV uređajem prihvati zahtev za izvršenje akcije sa vrednošću GUIDE. Prikaz EPG sadržaja zasniva se na pristupu DTV bazi podataka i sistematičnom prikazu postojećih televizijskih događaja.

5.7 Prikaz teletext sadržaja

Android radno okruženje ne predviđa klasičan prikaz teletext sadržaja, već samo pretragu na osnovu sadržaja DTV baze podataka. Upis teletext sadržaja u DTV bazu podataka u velikoj meri liči na proceduru upisa EPG sadržaja, iz tog razloga detalji implementacije neće biti razmatrani.

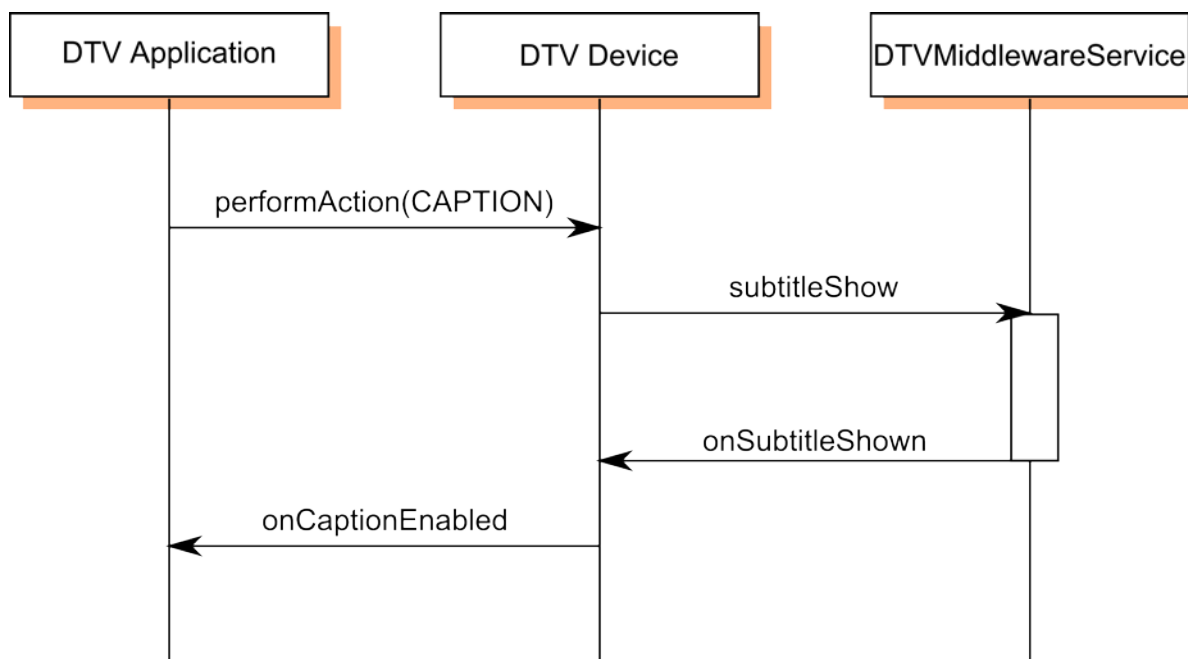
S obzirom da postojeće DTV MDW rešenje podržava mogućnost iscertavanja teletext sadržaja čime je omogućena integracija klasičnog prikaza teletext sadržaja. U smislu prikaza DTV MDW programska sprega pruža podršku za prikaz, odnosno uklanjanje i izbor željene stranice teletext sadržaja. Pošto od strane Android radnog okruženja ne postoji definisana akcija za prikaz teletext sadržaja, kao zamena izabrana je jedna od generičkih akcija. Bitno je naznačiti da uvođenjem podrške za prikaz teletext sadržaja moramo voditi računa o kontekstu u kom se sistem nalazi. Naime, ako je teletext sadržaj prikazan numerički korisnički unos se odnosi na broj teletext stranice, a ne izbor DTV servisa kao što je ranije bio slučaj (Slika 5.10).



Slika 5.10: Tok prikaza teletext sadržaja

5.8 Prikaz prevoda

Android radno okruženje podržava integraciju prevoda posedovanjem odgovarajuće programske sprege. DTV aplikacija kao element korisničke sprege sadrži metodu za uključivanje odnosno isključivanje prikaza prevoda. Programska sprega za uključivanje ili isključivanje prevoda realizovana je asinhrono (Slika 5.11). S obzirom da postojeća DTV MDW implementacija sadrži mogućnost prikaza prevoda, prikaz ili uklanjanje istih se svodi na poziv odgovarajuće metode iz modula za upravljanje DTV uređajem.

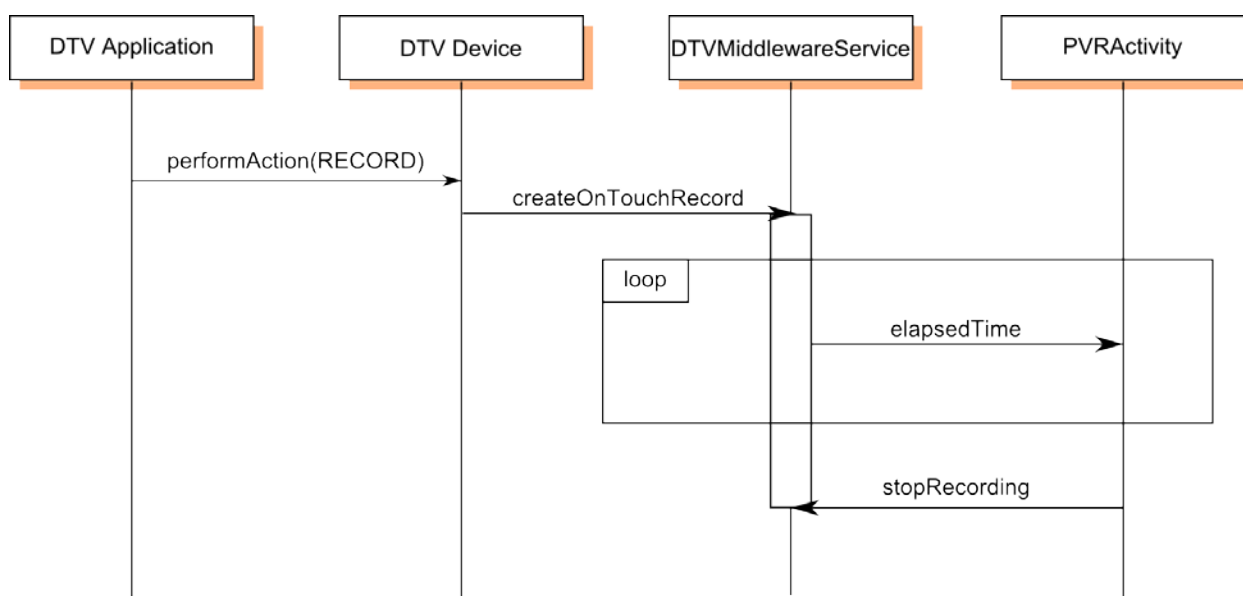


Slika 5.11: Tok prikaza prevoda

5.9 PVR podrška

Celokupna funkcionalnost PVR programske podrške realizovana je preuzetim DTV MDW rešenjem. DTV MDW implementacija pruža mogućnost snimanja trenutno gledanog DTV sadržaja, kao i zakazivanje snimanja televizijskog događaja u budućnosti na osnovu elektronskog programskog vodiča.

U slučaju trenutno prikazanog televizijskog sadržaja snimanje je započeto neposredno nakon poziva odgovarajuće funkcije programske sprege DTV MDW rešenja. Snimanje se započinje na osnovu RECORD akcije pristigle preko DTV aplikacije (Slika 5.12). Korisnici imaju uvid u trenutno snimani sadržaj na osnovu komponente korisničke grafičke sprege realizovane specijalno za tu namenu (PVR Activity). Na osnovu grafičke korisničke sprege korisnici imaju mogućnost uvida u količinu snimljenog sadržaja, kao i zaustavljanje ranije započetog snimanja. Pored aktivnosti snimanja sadržaja korisnička sprega omogućava reprodukciju i brisanje prethodno snimljenog sadržaja.

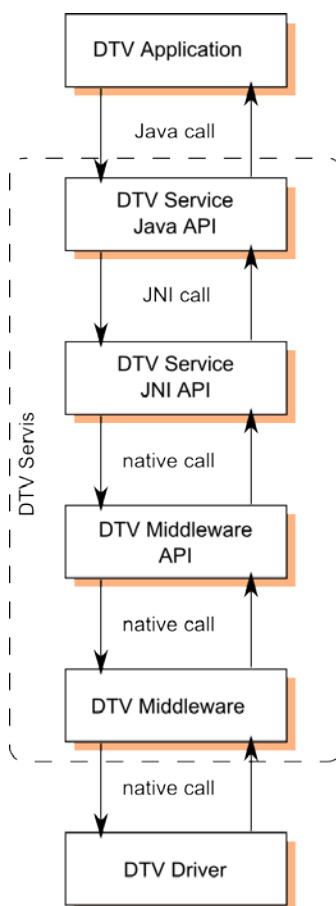


Slika 5.12: Tok pokretanja snimanja trenutno aktivnog DTV sadržaja

Snimanje DTV sadržaja na osnovu televizijskog događaja pokreće se iz grafičke korisničke sprege za prikaz elektronskog programskog vodiča (EpgActivity). Na osnovu vremena prikazivanja i DTV servisa datog televizijskog događaja zakazuje se snimanje televizijskog sadržaja. Listi zakazanih televizijskih događaja za snimanje korisnici mogu pristupiti iz već pomenute grafičke korisničke sprege vezane za snimanje televizijskog sadržaja (PVR Activity).

5.10 DTV Middleware Service

Modul za upravljanje DTV uređajem ostvaruje pristup televizijskoj programskoj podršci posredstvom usluga DTV Middleware Service Android servisa. DTV Middleware Service realizuje programsku spregu ka DTV MDW rešenju na visokom nivou apstrakcije programskog jezika java. S obzirom da je DTV MDW implementacija realizovana na nižem nivou apstrakcije, uz pomoć platformski zavisnih programskih jezika (C/C++), neophodno je postojanje prilagodnog sloja programske podrške u vidu JNI programske sprege. Slika 5.13 sadrži prikaz programske organizacije međusobnog sprezanja modula za upravljanje DTV uređajem sa funkcionalnom realizacijom DTV programske podrške.



Slika 5.13: Programska organizacija DTV podrške

DTV JNI programski sloj realizovan je u vidu dinamičke programske biblioteke, koja se učitava u vreme izvršavanja podsistema DTV programske podrške. Učitavanje pomenute programske biblioteke realizovano je u statičkom bloku, pri inicijalizaciji DTV Middleware Native Java programske klase.

```

static {
    System.loadLibrary("DTVMiddlewareJNI");
}

```

Realizacija JNI sloja DTV podrške podrazumeva poštovanje određenih pravila definisanih od strane JNI podsistema. Java virtualna mašina funkcije platformski zavisnog programskog rešenja traži na osnovu imena koje se sastoji od Java paketa, imena programske klase i imena same metode čiji poziv se izvršava. Takođe vrlo je bitan broj i tip formalnih argumenata funkcije. Svaka programska funkcija u ovom sloju programske podrške mora posedovati najmanje dva parametra, kontekst virtualne mašine i objekat Java programske klase čija metoda je realizovana datom funkcijom.

```
com_rtrk_dtvmiddleware_DTVMiddlewareNative_GoServiceUpDown(
jnienv *env, jobject thiz, jint upDown) {
    ...
}
```

U daljem tekstu dat je pregled realizovane Java programske sprege Android servisa DTV programske podrške. Opis programske sprege grupisan na osnovu funkcionalnosti koja je datom programskom spregom realizovana.

5.10.1 Programska sprega za pretragu dostupnih televizijskih servisa

Prototip	int ScanAll(int signalType, int updateList)
Opis	Metoda pokretaće automatsku pretragu dostupnih televizijskih servisa.
Parametri	signalType – tip emitovanog signala po kom se vrši pretraga (DVB-S, DVB-T, DVB-C, IP, PAL). updateList – indikator kojim se određuje da li se postojeća lista pronađenih servisa osvežava ili se zanemaruje.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int AbortScan()
Opis	Metoda prekida prethodno započetu pretragu dostupnih televizijskih servisa.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int ManualScan(int signalType, int updateList)
Opis	Metoda pokreće pojedinačnu pretragu dostupnih televizijskih servisa na osnovu prethodno postavljenih parametara.
Parametri	signalType – tip emitovanog signala po kom se vrši pretraga (DVB-S, DVB-T, DVB-C, IP, PAL). updateList – indikator kojim se određuje da li se postojeća lista pronađenih servisa osvežava ili se zanemaruje.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int SetFrequency(int frequency)
Opis	Metodom se postavlja vrednost frekvencije za pojedinačnu pretragu televizijskih servisa.
Parametri	frequency – vrednost frekvencije
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int GetFrequency()
Opis	Metodom se čita vrednost trenutno postavljene frekvencije za pojedinačnu pretragu televizijskih servisa.
Povratna vrednost	Vrednost frekvencije.

Prototip	int SetSymbolRate(int symbolRate)
Opis	Metodom se postavlja vrednost brzine prenosa elementarnih jedinica podataka za pojedinačnu pretragu televizijskih servisa.
Parametri	symbolRate – vrednost brzine prenosa elementarnih jedinica podataka
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int GetSymbolRate()
Opis	Metodom se čita vrednost trenutno postavljene brzine prenosa elementarnih jedinica podataka za pojedinačnu pretragu televizijskih servisa.
Povratna vrednost	Vrednost brzine prenosa elementarnih jedinica podataka.

Prototip	int SetPolarization(int polarization)
Opis	Metodom se postavlja vrednost polarizacije televizijskog signala za pojediničnu pretragu televizijskih servisa.
Parametri	polarization – vrednost polarizacije (HORIZONTAL, VERTICAL, CIRCULAR_LEFT, CIRCULAR_RIGHT, ALL).
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int GetPolarization()
Opis	Metodom se čita trenutna vrednost polarizacije televizijskog signala za pojedinačnu pretragu televizijskih servisa.
Povratna vrednost	Vrednost polarizacije (HORIZONTAL, VERTICAL, CIRCULAR_LEFT, CIRCULAR_RIGHT, ALL).

Prototip	int GetSignalInfo(SignalInfo signalInfo)
Opis	Metodom se čita informacija o kvalitetu signala za ranije postavljene parametre pojedinačne pretrage televizijskih servisa, kao što su snaga i kvalitet signala.
Parametri	signalInfo – pročitane informacije o kvalitetu signala
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

5.10.2 Programska sprega za manipulaciju listom pronađenih televizijskih servisa

Prototip	int SetServiceName(int serviceListIndex, String name)
Opis	Metodom se postavlja ime liste pronađenih televizijskih servisa, specificirane datim indeksom.
Parametri	serviceListIndex – indeks liste televizijskih servisa name – ime liste televizijskih servisa
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int GetServiceName(int serviceListIndex, String name)
Opis	Metodom se čita ime liste pronađenih televizijskih servisa, specificirane datim indeksom.
Parametri	serviceListIndex – indeks liste televizijskih servisa name – pročitano ime liste televizijskih servisa
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int GetServicesNumber(int serviceListIndex, int signalType, int serviceType)
Opis	Metodom se čita broj pronađenih televizijskih servisa za dati tip signala i servisa, u listi specificiranoj datim indeksom.
Parametri	serviceListIndex – index liste pronađenih servisa signalType – tip emitovanog signala po kom se vrši pretraga (DVB-S, DVB-T, DVB-C, IP, PAL).

	serviceType – tip pronađenih servisa (DIGITAL, ANALOG, RADIO, STREAMED, DATA,)
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int GoServiceUpDown(int serviceListIndex, int upDown)
Opis	Metodom se pokeraće reprodukcija televizijskog servisa koji se nalazi iza ili ispred servisa koji je trenutno u toku, iz liste pronađenih servisa specificiranoj datim indeksom liste servisa.
Parametri	serviceListIndex – index liste pronađenih servisa upDown – smer promene servisa (UP, DOWN)
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int GoServiceIndex(int serviceListIndex, int serviceIndex)
Opis	Metodom se pokeraće reprodukcija televizijskog servisa određenog indeksom, iz liste pronađenih servisa specificiranoj datim indeksom liste servisa.
Parametri	serviceListIndex – index liste pronađenih servisa serviceIndex – index izabranog servisa
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int StopActiveService(int serviceListIndex)
Opis	Metodom se zaustavlja prethodno započeta reprodukcija televizijskog servisa.
Parametri	serviceListIndex – index liste pronađenih servisa
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int GetActiveServiceIndex(int serviceListIndex)
Opis	Metodom se čita index televizijskog servisa čija reprodukcija je u toku.
Parametri	serviceListIndex – index liste pronađenih servisa
Povratna vrednost	Index televizijskog servisa čije reprodukcija je u toku.

Prototip	int GetService(int serviceListIndex, int serviceIndex, Service serviceData)
Opis	Metodom se čitaju informacije o televizijskom servisu na osnovu indeksa, iz liste pronađenih servisa specificirane indeksom liste servisa.
Parametri	serviceListIndex – index liste pronađenih servisa serviceIndex – index servisa čije informacije treba pročitati serviceData – pročitane informacije o servisu
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int SetLastWatchedChannel(int serviceListIndex, int serviceIndex)
Opis	Metodom se postavlja televizijski servis čija je reprodukcija prethodno bila u toku, na osnovu indeksa servisa.
Parametri	serviceListIndex – index liste pronađenih servisa serviceIndex – index servisa koji se postavlja
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int GetLastWatchedChannel(int serviceListIndex)
Opis	Metodom se čita indeks televizijskog servisa čija je reprodukcija prethodno bila u toku.
Parametri	serviceListIndex – index liste pronađenih servisa
Povratna vrednost	Indeks prethodno reprodukovanog televizijskog servisa.

5.10.3 Programska sprega za podešavanje kvaliteta slike

Prototip	int SetAspectRatio(int width, int height)
Opis	Metodom se postavlja odnosa stranica slike.
Parametri	width – širina slike. height – visina slike.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetAspectRatio()
Opis	Metodom se čita vrednost odnosa stranica slike.
Povratna vrednost	Odnos stranica slike (4_3, 16_9).

Prototip	int SetBrightness(int brightness)
Opis	Metodom se postavlja vrednost intenziteta osvetljenja slike.
Parametri	brightness – vrednost intenziteta osvetljenja slike.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetBrightness()
Opis	Metodom se čita trenutna vrednost intenziteta osvetljenja slike.
Povratna vrednost	Vrednost intenziteta osvetljenja slike.

Prototip	int SetContrast(int contrast)
Opis	Metodom se postavlja vrednost intenziteta kontrasta slike.
Parametri	contrast – vrednost intenziteta kontrasta slike.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetContrast()
Opis	Metodom se čita trenutna vrednost intenziteta kontrasta slike.
Povratna vrednost	Vrednost intenziteta kontrasta slike.

Prototip	int SetColorTemperature(int color_temperature)
Opis	Metodom se postavlja vrednost intenziteta temperature boje slike.
Parametri	colorTemperature – vrednost intenziteta temperature boje slike.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetColorTemperature()
Opis	Metodom se čita trenutna vrednost intenziteta temperature boje slike.
Povratna vrednost	Vrednost intenziteta temperature boje slike.

Prototip	int SetHue(int hue)
Opis	Metodom se postavlja vrednost nijanse boje slike.
Parametri	hue – vrednost nijanse boje slike.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetHue()
Opis	Metodom se čita trenutna vrednost nijanse boje slike.
Povratna vrednost	Vrednost nijanse boje slike.

Prototip	int SetSaturation(int saturation)
----------	-----------------------------------

Opis	Metodom se postavlja vrednost zasićenja boje slike.
Parametri	saturation – vrednost zasićenja boje slike.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetSaturation()
Opis	Metodom se čita trenutna vrednost zasićenja boje slike.
Povratna vrednost	Vrednost zasićenja boje slike.

Prototip	int SetBacklight(int backlight)
Opis	Metodom se postavlja vrednost intenzitet pozadinskog osvetljenja ekrana.
Parametri	backlight – vrednost intenziteta pozadinskog osvetljenja ekrana.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetBacklight()
Opis	Metodom se čita trenutna vrednost intenziteta pozadinskog osvetljenja ekrana.
Povratna vrednost	Vrednost intenziteta pozadinskog osvetljenja ekrana.

5.10.4 Programska sprega za podešavanje kvaliteta zvuka

Prototip	int SetVolume(int volume)
Opis	Metodom se postavlja vrednost intenziteta jačine zvuka.
Parametri	volume – vrednost intenziteta jačine zvuka.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetVolume()
Opis	Metodom se čita trenutna vrednost intenziteta jačine zvuka.
Povratna vrednost	Vrednost intenziteta jačine zvuka.

Prototip	int SetAudioMute(int audioMute)
Opis	Metodom se uključuje i isključuje zvuk.
Parametri	audioMute – indikator za uključivanje i isključivanje zvuka (ON, OFF).
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetAudioMute()
Opis	Metodom se čita trenutno stanje tona.
Povratna vrednost	Stanje tona (ON, OFF).

Prototip	int SetActiveSoundMode(int soundMode)
Opis	Metodom se aktivira željeni predefinisani režim zvuka.
Parametri	soundMode – željeni režim zvuka.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	int GetActiveSoundMode()
Opis	Metodom se čita trenutno aktivni režim zvuka.
Povratna vrednost	Režim zvuka.

Prototip	int SetActiveSpdifMode(int spdifMode)
Opis	Metodom se postavlja željeni predefinisani režim zvuka na spdif izlazu uređaja.

Parametri	spdifMode – željeni režim zvuka na spdif izlazu uređaja.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetActiveSpdifMode()
Opis	Metodom se čita trenutno aktivnog režima zvuka na spdif izlazu uređaja.
Povratna vrednost	Režim zvuka na spdif izlazu uređaja.

Prototip	int SetSurroundSound(int surround)
Opis	Metodom se uključuje i isključuje Surround režim zvuka.
Parametri	surround – indikator za uključivanje ili isključivanje surround režima zvuka (ON, OFF).
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetSurroundSound()
Opis	Metodom se čita stanje Surround režima zvuka.
Povratna vrednost	Stanje surround režima zvuka (ON, OFF).

Prototip	int SetAutoVolume(int autoVolume)
Opis	Metodom se uključuje i isključuje automatsko podešavanje vrednosti intenziteta zvuka.
Parametri	autoVolume – indikator za uključivanje ili isključivanje automatskog podešavanja vrednosti intenziteta zvuka (ON, OFF).
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetAutoVolume()
Opis	Metodom se čita trenutno stanje automatskog podešavanja vrednosti intenziteta zvuka.
Povratna vrednost	Stanje automatskog podešavanja intenziteta zvuka (ON, OFF).

Prototip	int SetClearVoice(int clearVoice)
Opis	Metodom se uključuje ClearVoice režim zvuka.
Parametri	clearVoice – indikator za uključivanje ili isključivanje ClearVoice režima zvuka (ON, OFF).
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetClearVoice()
Opis	Metodom se čita trenutno stanje ClearVoice režima zvuka.
Povratna vrednost	Stanje ClearVoice režima zvuka (ON, OFF).

Prototip	int SetBassEnhancer(int bassEnhancer)
Opis	Metodom se uključuje i isključuje Bass Enhancer režim zvuka.
Parametri	bassEnhancer – indikator za uključivanje ili isključivanje BassEnhancer režima zvuka (ON, OFF).
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetBassEnhancer()
Opis	Metodom se čita trenutno stanje BassEnhancer režima zvuka.
Povratna vrednost	Stanje BassEnhancer režima zvuka (ON, OFF).

Prototip	int SetTreble(int treble)
Opis	Metodom se podešava vrednost intenziteta jačine zvuka visokih tonova.

Parametri	treble – vrednost intenziteta jačine zvuka visokih tonova.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetTreble()
Opis	Metodom se čita trenutna vrednost intenziteta jačine zvuka visokih tonova.
Povratna vrednost	Vrednost intenziteta jačine zvuka visokih tonova.

Prototip	int SetBalance(int balance)
Opis	Metodom se podešava željena vrednost ravnoteže zvuka.
Parametri	balance – vrednost ravnoteže zvuka.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetBalance()
Opis	Metodom se čita trenutna vrednost ravnoteže zvuka.
Povratna vrednost	Vrednost ravnoteže zvuka.

Prototip	int SetBass(int bass)
Opis	Metodom se podešava vrednost intenziteta jačine zvuka niskih tonova.
Parametri	bass – vrednost intenziteta jačine zvuka niskih tonova.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetBalance()
Opis	Metodom se čita trenutna vrednost intenziteta jačine zvuka niskih tonova.
Povratna vrednost	Vrednost intenziteta jačine zvuka niskih tonova.

Prototip	int SetHeadPhoneVolume(int hpVolume)
Opis	Metodom se podešava vrednost intenziteta jačine zvuka na izlazu za slušalice.
Parametri	hpVolume – vrednost intenziteta jačine zvuka na izlazu za slušalice.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetHeadPhoneVolume()
Opis	Metodom se čita trenutna vrednost intenziteta jačine zvuka na izlazu za slušalice.
Povratna vrednost	Vrednost intenziteta jačine zvuka na izlazu za slušalice.

5.10.5 Programska sprega za upravljanje teletext podrškom

Prototip	int ShowTeletext(int alpha);
Opis	Metodom se uključuje prikaz teletext sadržaja.
Parametri	alpha - vrednost intenziteta transparentije prikaza teletext sadržaja.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int HideTeletext()
Opis	Metodom se isključuje prikaz teletext sadržaja.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetTeletextState()
Opis	Metodom se čita trenutno stanje prikaza teletext sadržaja.
Povratna vrednost	Stanje prikaza teletext sadržaja (ON, OFF).

Prototip	int SetTeletextMode(int teletextMode)
Opis	Metodom se podešava željeni režim prikaza teletext sadržaja.
Parametri	teletextMode – režim prikaza teletext sadržaja (FULL, HALF, MIX).
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetTeletextMode()
Opis	Metodom se čita trenutno aktivni režim prikaza teletext sadržaja.
Povratna vrednost	Režim prikaza teletext sadržaja (FULL, HALF, MIX).

Prototip	int GetTeletextTrackCount()
Opis	Metodom se čita broj dostupnih jezika za prikaz teletext sadržaja.
Povratna vrednost	Broj dostupnih jezika za prikaz teletext sadržaja.

Prototip	int GetTeletextTrack(int trackIndex, TeletextTrack teletextTrack)
Opis	Metodom se čitaju informacije o jeziku za prikaz teletext sadržaja po datom indeksu.
Parametri	trackIndex – index jezika za prikaz teletext sadržaja. teletextTrack – pročitani jezik za prikaz teletext sadržaja.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int SetCurrentTeletextTrack(int trackIndex)
Opis	Metodom se vrši izbor željenog jezika za prikaz teletext sadržaja po datom indeksu.
Parametri	trackIndex– index teletext stranice.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetCurrentTeletextTrackIndex()
Opis	Metodom se čita index trenutno aktivnog jezika za prikaz teletext sadržaja.
Povratna vrednost	Index trenutno aktivnog jezika za prikaz teletext sadržaja.

Prototip	int SendInputControl(int ctrl, int param)
Opis	Metodom se vrši prosleđivanje korisničkog unosa.
Parametri	ctrl – kod unosa. param – dodatni parametar.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetText(out TeletextText plainTeletext)
Opis	Metodom se čita tekst aktivne teletext stranice.
Parametri	plainText – pročitani tekst teletext stranice.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GoToPage(int pagenum)
Opis	Metodom se vrši izbor teletext stranice.
Parametri	pagenum – broj teletext stranice.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

5.10.6 Programska sprega za upravljanje EPG podrškom

Prototip	int GetEpgEvent(int serviceIndex, int eventIndex, EpgEvent epgEvent)
----------	---

Opis	Metodom se čitaju informacije o televizijskom događaju na osnovu indeksa televizijskog servisa i događaja.
Parametri	serviceIndex – indeks televizijskog servisa. eventIndex – indeks televizijskog događaja. epgEvent – pročitane informacije o televizijskom događaju.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetEpgEventListCount(int serviceIndex)
Opis	Metodom se čita broj dostupnih televizijskih događaja za dati televizijski servis.
Parametri	serviceIndex – indeks televizijskog servisa.
Povratna vrednost	Broj dostupnih televizijskih događaja.

5.10.7 Programska sprega za upravljanje prikazom podnapisa

Prototip	int ShowSubtitle(int alpha)
Opis	Metodom se uključuje prikaz podnapisa.
Parametri	alpha – vrednost intenziteta transparentije prikaza podnapisa.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).
Prototip	int HideSubtitle()
Opis	Metodom se isključuje prikaz podnapisa.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetSubtitleState()
Opis	Metodom se čita trenutno stanje prikaza podnapisa.
Povratna vrednost	Stanje prikaza podnapisa (ON, OFF).

Prototip	int GetSubtitleAsBitmap(SubtitleBitmap subtitleBitmap)
Opis	Metodom se čita sadržaj podnapisa u vidu bitmape.
Parametri	subtitleBitmap – pročitani podnapisi u vidu bitmape.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	String GetSubtitleAsText()
Opis	Metodom se čita sadržaj podnapisa u vidu teksta.
Povratna vrednost	Tekst podnapisa.

Prototip	int SetSubtitleLanguage(int subtitleLanguageCode)
Opis	Metodom se vrši izbor željenog jezika za prikaz podnapisa.
Parametri	subtitleLanguageCode – kod željenog jezika za prikaz podnapisa.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetSubtitleLanguage()
Opis	Metodom se čita trenutno aktivni jezik za prikaz podnapisa.
Povratna vrednost	Kod jezika za prikaz podnapisa.

Prototip	int SetSubtitleMode(int subtitleMode)
Opis	Metodom se podešava željeni režim prikaza podnapisa.
Parametri	subtitleMode – režim prikaza podnapisa (MODE1, MODE2).
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetSubtitleMode()
Opis	Metodom se čita trenutno aktivni režim prikaza podnapisa.
Povratna vrednost	Režim prikaza podnapisa (MODE1, MODE2).

5.10.8 Programska sprega za upravljanje PVR podrškom

Prototip	int CreateOnTouchRecord(int serviceID)
Opis	Metodom se pokreće snimanje televizijskog programa.
Parametri	serviceID – identifikator televizijskog servisa.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int StopRecord()
Opis	Metodom se zaustavlja prethodno pokrenuto snimanje televizijskog programa.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).
Prototip	int StartTimeshiftRecord()
Opis	Metodom se pokreće snimanje televizijskog programa za odloženo gledanje.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int StopTimeshiftRecord()
Opis	Metodom se zaustavlja prethodno pokrenuto snimanje televizijskog programa za odloženo gledanje.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int PauseResumeTimeshiftPlayback(int pauseResume)
Opis	Metodom se pokreće i zaustavlja reprodukcija odloženog gledanja televizijskog programa.
Parametri	pauseResume – indikator za pokretanje i zaustavljanje odloženog gledanja televizijskog sadržaja.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int SetTimeShiftPlaybackSpeed(int playbackSpeed)
Opis	Metodom se podešava vrednost brzine reprodukcije odloženog gledanja televizijskog programa.
Parametri	playbackSpeed – vrednost brzine reprodukcije.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int StartPlayback(int itemID)
Opis	Metodom se pokreće reprodukcija prethodno snimljenog televizijskog programa.
Parametri	itemID – identifikator ranije snimljenog televizijskog programa.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int PauseResumePlayback(int pauseResume)
Opis	Metodom se pauzira i ponovo pokreće ranije započeta reprodukcija snimljenog programa.
Parametri	pauseResume – indikator za pauziranje i pokretanje reprodukcije snimljenog programa.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	<code>int StopPlayback(int startLive)</code>
Opis	Metodom se zaustavlja prethodno pokrenuta reprodukcija snimljenog televizijskog programa.
Parametri	<code>startLive</code> - indikator za započinjanje reprodukcije trenutno emitovanog televizijskog programa (TRUE, FALSE).
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	<code>int FastForwardPlayback()</code>
Opis	Metodom se reprodukcija snimljenog televizijskog programa premotava u napred.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	<code>int RewindPlayback()</code>
Opis	Metodom se reprodukcija snimljenog televizijskog programa premotava u nazad.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	<code>int UpdateRecordPlaylist()</code>
Opis	Metodom se osveža lista snimljenog televizijskog programa.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja)

Prototip	<code>String GetItemTitle(int itemID)</code>
Opis	Metodom se čita naziv snimljenog televizijskog programa.
Parametri	<code>itemID</code> – identifikator snimljenog televizijskog programa.
Povratna vrednost	Naziv snimljenog televizijskog programa.

Prototip	<code>String GetItemDescription(int itemID)</code>
Opis	Metodom se čita opis snimljenog televizijskog programa.
Parametri	<code>itemID</code> – identifikator snimljenog televizijskog programa.
Povratna vrednost	Opis snimljenog televizijskog programa.

Prototip	<code>int GetItemDuration(int itemID)</code>
Opis	Metodom se čita dužina trajanja snimljenog televizijskog programa.
Parametri	<code>itemID</code> – identifikator snimljenog televizijskog programa.
Povratna vrednost	Vreme trajanja snimljenog televizijskog programa.

Prototip	<code>String GetItemDateTime(int itemID)</code>
Opis	Metodom se čita tačno vreme i datum snimanja televizijskog programa.
Parametri	<code>itemID</code> – identifikator snimljenog televizijskog programa.
Povratna vrednost	Vreme i datum snimanja televizijskog programa.

Prototip	<code>int CreateSmartRecord(EpgEvent scheduledEvent)</code>
Opis	Metodom se vrši zakazivanje snimanja televizijskog događaja.
Parametri	<code>scheduledEvent</code> – televizijski događaj.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	<code>int GetScheduledEventsListCount()</code>
Opis	Metodom se čita broj zakazanih televizijskih događaja za snimanje.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	<code>int GetScheduledEvent(int position, EpgEvent scheduledEvent)</code>
----------	---

Opis	Metodom se čita zakazani televizijski događaj za snimanje.
Parametri	position – pozicija u listi zakazanih televizijskih događaja za snimanje. scheduledEvent – pročitani televizijski događaj.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int RemoveScheduledEvent(int position)
Opis	Metodom se uklanja zakazani televizijski događaj za snimanje.
Parametri	position – pozicija u listi zakazanih televizijskih događaja za snimanje.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int RemoveItemInRecordedList(int position)
Opis	Metodom se uklanja snimljeni televizijski program.
Parametri	position – pozicija u listi snimljenog televizijskog programa.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

5.10.9 Programska sprega za upravljanje HbbTV podrškom

Prototip	int SetHbbTvEngineState(int newState)
Opis	Metodom se uključuje i isključuje prikaz HbbTV sadržaja.
Parametri	newState – indikator uključivanja i isključivanja prikaz HbbTV sadržaja (ON, OFF).
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int GetHbbTvEngineState()
Opis	Metodom se čita trenutno aktivno stanje prikaza HbbTV sadržaja.
Povratna vrednost	Stanje prikaza HbbTV sadržaja (ON, OFF).

Prototip	int SendHbbTvInputControl(int ctrl, int param)
Opis	Metodom se vrši prosleđivanje korisničkog unosa.
Parametri	ctrl – kod unosa. param – dodatni parametar.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

Prototip	int FailedToLoadHbbTvUrl(int reason)
Opis	Metodom se vrši indikacija o neuspešnom učitavanju HbbTV sadržaja.
Parametri	reason – kod greške.
Povratna vrednost	Kod greške (0 u slučaju uspešnog izvršenja).

6. Eksperimentalni rezultati

U okviru ovog poglavlja dat je pregled i analiza prikupljenih eksperimentalnih rezultata rada predloženog rešenja integracije DTV podrške u Android operativni sistem. Rad sistema razmatran je iz dva aspekta, aspekt performansi sistema i aspekt odziva sistema. Sa stanovišta performansi sistema analizirana je zauzetost radne memorije i centralne procesorske jedinice. Analizom odziva sistema utvrđena su vremena reagovanja pri korisničkoj interakciji sa realizovanim računarskim sistemom.

Razvojna platforma korišćena za prikupljanje eksperimentalnih rezultata sadrži 512 MB radne memorije i takozvani sistem u okviru integrisanog kola (SOC – *System on Chip*). Pomenuti SOC poseduje mogućnost dekodovanje audio izvora, paralelno dekodovanje dva video izvora i CPU sa dva jezgra ARMv7 arhitekture, nominalne radne frekvencije 1,2 GHz. Opisana platforma poseduje prosečne mogućnosti u domenu digitalne televizije sa stanovišta fizičke arhitekture.

6.1 Analiza performansi rešenja

Performanse sistema utvrđene su na osnovu rezultata merenja zauzeća radne memorije i centralne procesorske jedinice, pri reprodukciji televizijskog sadržaja u toku 13 časova i 23 minuta. Merenje je izvršeno čitanjem odgovarajućih vrednosti iz *proc* fajl sistema, Linux operativnog sistema, dok je vremenski interval prikupljanja odbiraka jednak 5 sekundi. Za cilj merenja performansi imamo određivanje količine potrebnih resursa za nesmetan rad, kao i određivanje količine raspoloživih resursa za izvršavanje dodatnih aktivnosti. Takođe, na osnovu rezultata merenja utvrđena je stabilnost sistema u toku dugotrajnog rada sistema.

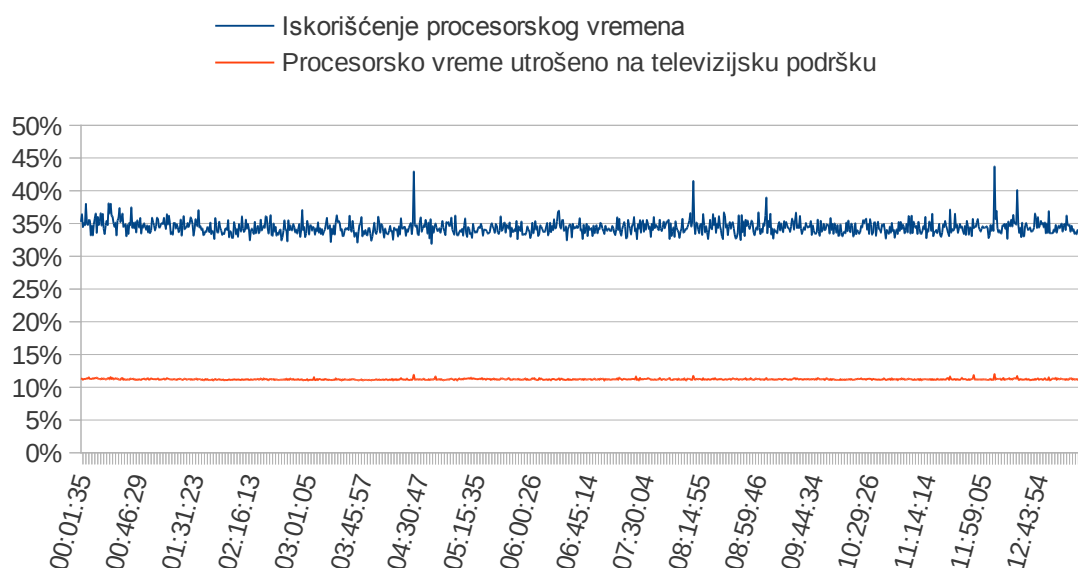
6.1.1 Zauzeće centralne procesorske jedinice

Merenje zauzeća centralne procesorske jedinice je komplikovan proces, jer se uopšteno gledano radi o diskretnoj vrednosti, CPU je uključen ili isključen. Dok je uključen CPU mora izvršavati instrukcije, iz tog razloga Unix operativni sistem poseduje *idle* proces. *Idle* proces je proces operativnog sistema niskog prioriteta koji se izvršava kada ne postoji potreba za izvršenjem nekog drugog procesa. Možemo definisati dva vida određivanja zauzeća centralne procesorske jedinice: zauzeće kao funkcija procesorskog vremena i zauzeće kao funkcija broja aktivnih procesa.

Utrošak procesorskog vremena određuje se u odnosu na vremenski interval i predstavlja odnos ukupnog procesorskog vremena i procesorskog vremena provedenog u izvršavanju procesa različitih od *idle* procesa. Određivanje procentualne vrednosti utroška procesorskog vremena definisano je jednačinom (6.1).

$$cpu_load = \frac{total_cpu_time - idle_cpu_time}{total_cpu_time} \cdot 100 \quad (6.1)$$

Slika 6.1 prikazuje je graf dijagram utroška procesorskog vremena nad celokupnim intervalom obavljenog merenja. Pored ukupnog iskorišćenja procesorskog vremena graf dijagram sadrži i prikaz utrošenog procesorskog vremena od strane televizijske podrške. Kako prikazani graf dijagram ne sadrži značajne oscilacije, možemo zaključiti da je sistem radio stabilno tokom merenja. Takođe, vidimo da je prosečna iskorišćenost 35% ukupnog procesorskog vremena, čime je potvrđena mogućnost izvršavanja dodatnih aktivnosti.



Slika 6.1: Graf dijagram utroška procesorskog vremena

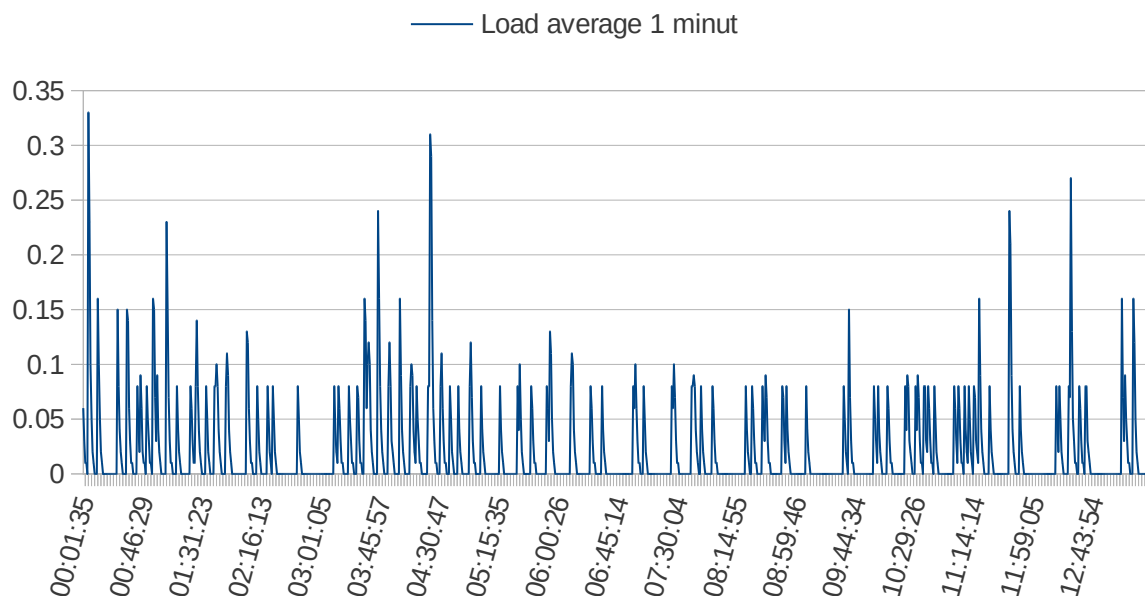
Unix operativni sistem definiše *load average* kao meru zauzetosti centralne procesorske jedinice. *Load average* vrednost predstavlja broj aktivnih procesa u prethodno definisanom vremenskom intervalu. Na primer, za zadati interval od jednog minuta *load average* će imati vrednost jedan, ako je u toku prethodnog minuta neprekidno bio aktivan samo jedan proces. Kada je *load average* vrednost jednaka broju jezgara centralne procesorske jedinice, reč je o optimalnom iskorišćenju iste. Dok u slučaju *load average* vrednosti manje od jedan i veće od jedan, radi se o neiskorišćenju odnosno preopterećenju respektivno. Operativni sistem Unix pruža mogućnost uvida u *load average* za tri vremenska intervala: 1 minut, 5 minuta i 15 minuta.

Linux kernel koji je realizovan po ugledu na Unix, takođe vrši određivanje *load average* vrednosti. *Load average* u okviru Linux kernela, izračunava se na sledeći način:

$$load_average(t) = load_average(t-1) \cdot e^{\frac{-u}{T}} + n \cdot (1 - e^{\frac{-u}{T}}) \quad (6.2)$$

gde je n broj aktivnih procesa u trenutku uzimanja odbirka, u period uzimanja odbirka (5 sekundi), a T vremenski interval za koji se određuje *load average* vrednost. Iz jednačine (6.2) vidi se da *load average* vrednost teži nuli, kada je broj aktivnih procesa jednak nuli.

Slika 6.2 prikazuje *load average* vrednost izračunata za vremenski interval od jednog minuta, u toku celokupnog trajanja merenja. Možemo videti da *load average* ne prelazi 0,35 čime je potvrđena rasterećenost sistema u odnosu na broj aktivnih procesa.



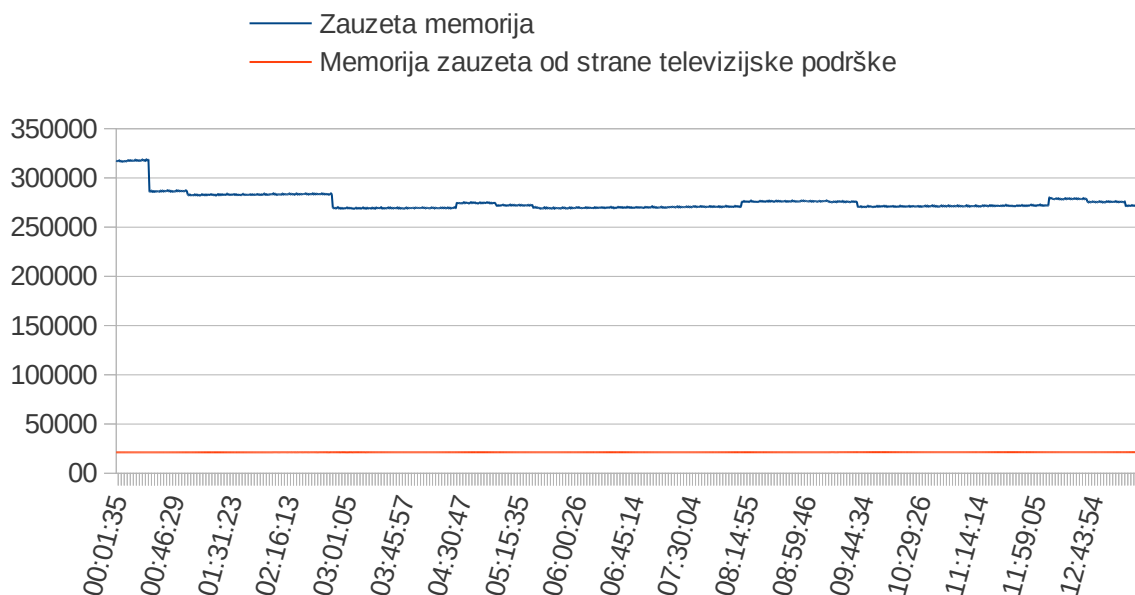
Slika 6.2: Graf dijagram *load average* vrednosti

6.1.2 Zauzeće radne memorije

Analizom zauzeća radne memorije želimo utvrditi količinu radne memorije potrebne za nesmetan rad sistema, pri reprodukciji televizijskog programa. Linux kernel pruža mogućnost uvida u stanje radne memorije preko četiri vrednosti *MemTotal*, *MemFree*, *Buffers* i *Cached*. Količina zauzete memorije utvrđena je jednačinom (6.3), gde je *MemTotal* ukupna raspoloživa memorija, *MemFree* slobodna memorija, a *Buffers* i *Cached* privremeno zauzeta memorija od strane operativnog sistema koja se po potrebi oslobađa. Za merenje zauzeća radne memorije odabran je vremenski period za uzimanje odbiraka od 5 sekundi.

$$UsedMemory = MemTotal - (MemFree + Buffers + Cached) \quad (6.3)$$

Slika 6.3 prikazuje ukupno zauzeće radne memorije u toku celokupnog vremenskog intervala izvršenog merenja. Pored ukupnog zauzeća prikazano je i zauzeće od strane integrisane televizijske podrške. Sa prikazanog dijagrama možemo uočiti da nema velikih oscilacija, kao i da ne postoji trend rasta zauzete memorije, iz čega sledi da je sistem radio stabilno tokom merenja sa aspekta potrošnje radne memorije. Takođe može se zaključiti da je 512 MB radne memorije, koliko poseduje razvojna platforma dovoljno za nesmetanu reprodukciju televizijskog programa i izvršavanje dodatnih aktivnosti.

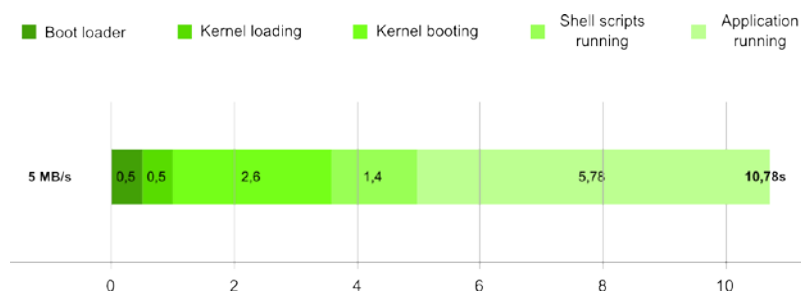


Slika 6.3: Graf dijagram zauzeća radne memorije izraženo u KB

6.2 Analiza odziva sistema

Sa stanovišta udobnosti, u sferi uređaja potrošačke elektornike, važan faktor predstavlja odziv računarskog sistema pri korisničkoj interakciji. U smislu odziva, potrebno je obraditi korisnički zahtev što je brže moguće ili u slučaju dugotrajne obrade neophodno je obezbediti povratnu informaciju o prijemu zahteva. Za potrebe određivanja odziva sistema, mereno je vreme uključenja sistema (vreme potrebno za prelazak iz isključenog u aktivno stanje sistema) i vreme promene reprodukcije televizijskog servisa. Merenje je izvršeno analizom video snimka koji sadrži aktivnosti koje želimo izmeriti. S obzirom da je za merenje upotrebljen standardni video snimak, 40 milisekundi je minimalan vremenski interval koji je moguće izmeriti.

Rezultate merenja poredićemo sa vremenom uključenja komercijalno dostupnog televizijskog prijemnika [19]. Referentni sistem sastoji se od centralne procesorske jedinice MIPS arhitekture, 128 MB radne memorije, NAND fleš masovne memorije brzine čitanja 5 MB/s, tjunera i dekodera. Slika 6.4 prikazuje pregled vremena potrebnog za uključenje referentnog sistema, po fazama sekvence uključenja. Ukupno vreme za uključenje referentnog sistema je 10,78 sekundi, gde se sistem smatra uključenim kad se uključi ekran televizijskog uređaja, jer korisnici poistovećuju uključenje ekrana sa uključenjem uređaja.



Slika 6.4: Vreme uključenja referentnog sistema

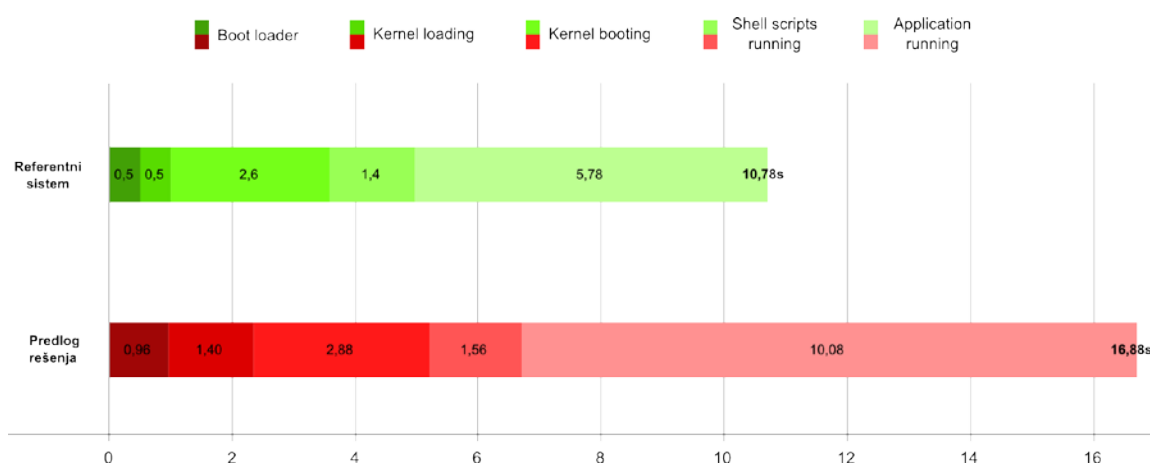
Tabelom Tabela 6.1 dat je pregled vremena uključenja realizovanog sistema, na osnovu izvršenih deset merenja. Radi boljeg uvida u vreme potrebno za uključenje sistema, dat je pregled po fazama procesa uključenja. Prosečno vreme potrebno za uključenje realizovanog sistema je 23,09 sekundi.

Broj merenja	Boot loader	Kernel loading	Kernel booting	Shell scripts	Applications loading	Playblack
1	0,96	1,40	2,88	1,60	10,08	23,28
2	0,96	1,40	2,88	1,56	10,08	22,72
3	0,96	1,40	2,88	1,56	10,04	23,40
4	0,96	1,40	2,88	1,56	10,12	22,60
5	0,96	1,40	2,88	1,56	10,04	22,80

6	1,00	1,40	2,88	1,56	10,12	23,32
7	0,96	1,40	2,88	1,56	10,08	23,68
8	0,96	1,40	2,88	1,56	10,12	22,76
9	0,96	1,40	2,88	1,56	10,08	22,88
10	0,96	1,40	2,88	1,56	10,08	23,48
Srednja vrednost	0,96	1,40	2,88	1,56	10,08	23,09

Tabela 6.1: Vreme uključenja sistema izraženo u sekundama

Slika 6.5 sadrži je uporedni prikaz vremena potrebnog za uključenje referentnog sistema i predloženog rešenja. Možemo videti da je vreme utrošeno za *kernel booting* i *shell scripts running* uporedivo sa referentim sistemom, dok postoji razlika u fazama *boot loader* i *kernel loading* kao posledica različite veličine kernela i *boot procedure*. Može se primetiti značajna razlika u fazi *application running*, gde referentni sistem uključuje samo jednu aplikaciju, DTV aplikacija, dok se u slučaju Android operativnog sistema uključuje čitav set aplikacija. Dužina poslednje faza pokretanja uređaja nije uzrokovana predloženim rešenjem, već samim operativnim sistemom, odnosno sva rešenja baziranja na Android operativnom sistemu imaju isti problem.



Slika 6.5: Uporedni prikaz vremena uključenja televizijskih uređaja

Kvalitet korisničkog iskustva upotrebe digitalnih televizijskih prijemnika u velikoj meri zavisi od brzine promene televizijskog servisa. Ako promenu reprodukcije televizijskog servisa posmatramo kao slučaj korisničke interakcije sa računarskim sistemom, kvalitet odziva možemo odrediti u odnosu na opštu skalu odziva [20], do 0,1 sekund pruža osećaj momentalnog odziva, 1 sekund je dovoljan da se ne prekine tok misli korisnika i 10 sekundi je maksimalno vreme za očuvanje korisničke pažnje. Zadovoljavajuće vreme promene televizijskog servisa teško je utvrditi, jer se pre svega radi o subjektivnom osećaju, zato postoje različiti kriterijumi ispod 1 s [21] ili ispod 0,43 s [22].

Broj merenja	Vreme promene kanala
1	3,76
2	3,84
3	3,84
4	3,76
5	3,56
6	3,80
7	3,76
8	3,60
9	3,68
10	3,72
Srednja vrednost :	3,73

Tabela 6.2: Vreme promene televizijskog servisa izraženo u sekundama

Prosečno vreme promene televizijskog servisa je 3.73 sekundi, određeno na osnovu deset izmerenih vrednosti (Tabela 6.2). Izmerene vrednosti su vremena promene televizijskog servisa od slike do slike. Postignuti rezultat prevazilazi ranije definisane kriterijume, međutim nalazi se u okviru 10 sekundi kao maksimum dozvoljenog vremena interakcije sa računarom. Za dobijanje boljih rezultata potrebno je izvršiti optimizaciju u okviru DTV MDW implementacije, jer je vreme propagacije zahteva za promenom servisa kroz Android radno okruženje reda veličine 10 ms.

7. Pravci daljeg razvoja

Moguća unapređenja postojećeg rešenja odnose se na izmene programske organizacije predloženog rešenja. Predloženim rešenjem DTV MDW implementacija je u Android operativni sistem uvedena u vidu servisa čije usluge su ostatku sistema dostupne posredstvom mehanizma međuprocenke komunikacije. Uzimajući u obzir da Android radno okruženje poseduje podsistem za pristup DTV programskoj podršci, usluge dodatnog servisa nisu potrebne. Integracijom DTV MDW rešenja na nivou modula za upravljanje DTV uređajem ostvaruje se bolja organizacija sa stanovišta implementacije DTV podrške.

Preuzeta DTV MDW implementacija sadrži Media Player, zbog čega je bilo neophodno izmeniti postojeći Media Player Android radnog okruženja. Odgovarajuće DTV MDW rešenje sadrži programsku spregu neophodnu za upotrebu eksterne Media Player realizacije, čime se postiže međusobna nezavisnost DTV podrške od Media Player implementacije.

U pogledu skladištenja podataka relevantnih za DTV podršku predloženo rešenje podrazumeva dve baze podataka. Sama DTV MDW implementacija poseduje svoju bazu podataka, dok podsistem DTV podrške Android radnog okruženja formira drugu bazu podataka. Ovakav pristup vrlo često ima problem sa sinhronizacijom podataka, gde su rezultati nepredvidljivi u slučaju nepodudaranja podataka u obe baze. Takođe, zbog postojanja dve baze podataka troši se dodatno procesorsko vreme na sinhronizaciju. Svrishodnije je rešenje koje podrazumeva jednu bazu podataka i to na višem nivou apstrakcije, kao što je podsistem televizijske podrške Android Java radnog okruženja.

8. Zaključak

Zadatak ovog rada je proširenje postojećih mogućnosti Android operativnog sistema sa funkcionalnostima televizijske podrške zarad implementacije televizijskog prijemnika. Na osnovu opisa predloženog rešenja može se zaključiti da su ciljevi zadatka u potpunosti ostvareni.

U pogledu osnovnih funkcionalnosti televizijskih prijemnika postojeća programska podrška Android operativnog sistema je na zadovoljavajućem nivou, u smislu odgovarajuće programske sprege i elemenata korisničke grafičke sprege.

Sa stanovišta naprednih mogućnosti televizijskih prijemnika Android radno okruženje u potpunosti ne predviđa mogućnost njihove implementacije. Za prikaz elektronskog programskog vodiča i teletexta neophodno je implementirati sopstvenu korisničku grafičku spregu, jer postojeće rešenje Android operativnog sistema podržava samo pretragu sadržaja na osnovu ključnih reči [3]. U smislu integracije PVR podrške takođe postoji potreba za realizacijom sopstvene korisničke grafičke sprege, jer u smislu PVR podrške Android radno okruženje sadrži jedino kontrole za upravljanje reprodukcijom.

Način integracije i performanse predloženog rešenja u velikoj meri zavise od preuzete DTV MDW implementacije, jer se televizijska podrška ostvaruje posredstvom datog podsistema programske podrške. Moguća unapređenja predloženog rešenja ogledaju se u izmenama integracije i implementacije DTV MDW, kao što je opisano prethodnim poglavljem.

Na osnovu eksperimentalnih rezultata može se zaključiti da realizovani sistem pokazuje stabilnost u radu na testnoj platformi. Sa stanovišta iskorišćenja centralne procesorske jedinice sistem pokazuje rasterećen rad, što znači da se neiskorišćeno procesorsko vreme može upotrebiti za izvršavanje korisničkih aplikacija. U pogledu radne memorije DTV programska podrška ima mali udeo u ukupno zauzetoj memoriji, takođe količina zauzete memorije kreće oko stalne vrednosti, što predstavlja još jedan pokazatelj stabilnosti sistema.

9. Literatura

- [1] TV Basics, Television Bureau of Advertising, June 2012
- [2] www.appbrain.com, Jul 2014.
- [3] AndroidTV, https://developers.google.com/tv/android/docs/gtv_android, Septembar 2014.
- [4] LG 55G2, <http://www.lg.com/us/tvs/lg-55G2-led-tv>, Septembar 2014.
- [5] TV SFR decodeur, <http://adsl.sfr.fr/television/decodeur-tv-sfr-avec-google-play/>, Septembar 2014.
- [6] ASUS CUBE, http://www.asus.com/us/Multimedia/ASUS_CUBE_with_Google_TV/, Septembar 2014.
- [7] Sony NSZGS8, <http://store.sony.com/internet-player-with-google-tv-zid27-NSZGS8/cat-27-catid-All-Internet-Players>, Septembar 2014.
- [8] Logitech Revue, <http://www.logitech.com/en-us/support/smart-tv>, Septembar 2014.
- [9] Netgear GTV100, <http://www.netgear.com/home/products/connected-entertainment/streaming-players/GTV100.aspx>, Septembar 2014.
- [10] Hisense Pulse, <http://www.hisense-usa.com/pulse/>, Septembar 2014.
- [11] Vizio Co-Star, <http://www.vizio.com/costar.html>, Septembar 2014.
- [12] Y. S. Lu, C. H. Lee, H. Y. Weng, Y. M. Huang, "Design and implementation of digital TV widget for Android on multi-core platform", International Computer Symposium (ICS), December 2010, pp. 576-580.
- [13] N. Kuzmanović, T. Maruna, M. Savić, G. Miljković, D. Isailović, "Google's Android as an application environment for DTV decoder system", 14th IEEE International Symposium on Consumer Electronics (ISCE), June 2010, pp. 1-5.
- [14] Thorsten Schreiber, "Android Binder - Android Interprocess Communication", 2011.
- [15] M. Vidaković, N. Teslić, T. Maruna, V. Mihić, "Android4TV: A proposition for integration of DTV in Android devices", IEEE International Conference on Consumer Electronic (ICCE), Januar 2012.
- [16] W. Fischer, "Digital Video and Audio Broadcasting Technology", 2010.
- [17] Steven Morris, Anthony Smith-Chaigneau, "Interactive TV Standards A Guide to MHP, OCAP and JavaTV", 2005, pp. 538.
- [18] W. Brückner, "Teletext in digital television", EBU Technical Review, Mart 2001.
- [19] Heeseung Jo, Hwanju Kim, Hyun-Gul Roh, Joonwon Lee, "Improving the startup time of digital TV", IEEE Transactions on Consumer Electronics, 2009, pp. 721-727.

-
- [20] Robert B. Miller, "Response time in man-computer conversational transactions", Fall Joint Computer Conference, 1968.
 - [21] F. Kozamernik, L. Vermaele, "Will Broadband TV shape the future of broadcasting?", EBU Technical Review, April 2005.
 - [22] R. Kooij, K. Ahmed, K. Brunnstrom, "Perceived Quality of Channel Zapping", Preceedings of the Fifth IASTED International Conference on Communication Systems and Networks, 2006.