



# УНИВЕРЗИТЕТ У НОВОМ САДУ ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА  
НОВИ САД  
Департман за рачунарство и аутоматику  
Одсек за рачунарску технику и рачунарске комуникације

## ЗАВРШНИ (BACHELOR) РАД

Кандидат: Марко Ковачевић  
Број индекса: Е12669

Тема рада: Једно решење приказа садржаја интерактивних сервиса дигиталне телевизије у андроид графичкој равни

Ментор рада: Проф. др Никола Теслић

Нови Сад, Јун, 2012.



## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:	
Идентификациони број, ИБР:	
Тип документације, ТД:	Монографска документација
Тип записа, ТЗ:	Текстуални штампани материјал
Врста рада, ВР:	Завршни (Bachelor) рад
Аутор, АУ:	Марко Ковачевић
Ментор, МН:	Проф. др Никола Теслић
Наслов рада, НР:	Једно решење приказа садржаја интерактивних сервиса дигиталне телевизије у андроид графичкој равни
Језик публикације, ЈП:	Српски / латиница
Језик извода, ЈИ:	Српски
Земља публиковања, ЗП:	Република Србија
Уже географско подручје, УГП:	Војводина
Година, ГО:	2012
Издавач, ИЗ:	Ауторски репримт
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/слика/графика/прилога)	7/35/0/0/21/0/0
Научна област, НО:	Електротехника и рачунарство
Научна дисциплина, НД:	Рачунарска техника
Предметна одредница/Кључне речи, ПО:	Дигитална телевизија, телетекст, превод, андроид, Јава сервис, програмска подршка телевизијског пријемника
УДК	
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, ВН:	
Извод, ИЗ:	У овом раду је приказано једно решење приказа садржаја интерактивних сервиса дигиталне телевизије у андроид графичкој равни. Решење је реализовано на Marvell BG2 SOC платформи на којој је инсталiran Андроид оперативни систем.
Датум прихватања теме, ДП:	
Датум одбране, ДО:	
Чланови комисије, КО:	Председник: Члан: Члан, ментор:
	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:	Bachelor Thesis	
Author, AU:	Marko Kovačević	
Mentor, MN:	PhD Nikola Teslić	
Title, TI:	One visualization solution of interactive digital television services content in android graphics plane	
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:	2012	
Publisher, PB:	Author's reprint	
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)	7/35/0/0/21/0/0	
Scientific field, SF:	Electrical Engineering	
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, S/KW:	Digital television, Teletext, Subtitles, Android, Java service, DTV Middleware	
UC		
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, N:		
Abstract, AB:	This paper presents visualization solution of interactive digital television services content in Android graphic plane. Solution is implemented on Marvell BG2 SOC platform on which Android operating system is installed.	
Accepted by the Scientific Board on, ASB:		
Defended on, DE:		
Defended Board, DB:	President:  Member:  Member, Mentor:	Menthor's sign

## SADRŽAJ

1.	Uvod.....	1
2.	Teorijske osnove .....	3
2.1	Android operativni sistem.....	3
2.2	Digitalna televizija .....	3
2.3	Teletekst.....	4
2.4	Prevod (eng. <i>Subtitles</i> ).....	6
2.5	MHEG (eng. <i>Multimedia and Hypermedia Experts Group</i> ) .....	7
2.6	Opis ciljne platforme .....	7
3.	Koncept rešenja .....	9
4.	Programsko rešenje.....	13
4.1	Realizacija iscrtavanja Teleteksta .....	13
4.2	Bafer deljene memorije (Anonymous SHared MEMory subsystem) .....	19
4.3	Realizacija iscrtavanja prevoda (eng. <i>Subtitles</i> ).....	20
4.4	Realizacija iscrtavanja MHEG-a.....	23
5.	Ispitivanje i verifikacija .....	24
5.1	Ispitivanje teletekst realizacije .....	24
5.2	Ispitivanje realizacije prevoda .....	26
5.3	Ispitivanje MHEG realizacije.....	27
6.	Zaključak.....	28
7.	Literatura .....	29

## SPISAK SLIKA

Slika 2.1 Najvažniji standardi digitalne televizije .....	4
Slika 2.2 MPEG i DVB komponente.....	5
Slika 2.3 Teletekst podaci u PES paketu.....	5
Slika 2.4 PID teletekst servisa u PMT .....	6
Slika 2.5 Hjerarhijska struktura DTV prevoda (eng. <i>Subtitles</i> ) .....	6
Slika 2.6 Marvell BG2 SOC platforma.....	8
Slika 3.1 Programski slojevi za integraciju DTV u Android OS .....	10
Slika 3.2 Programski slojevi namenjeni iscrtavanju teleteksta, prevoda i MHEG-a.....	11
Slika 3.3 Interakcija korisnika i interaktivnih servisa digitalne televizije .....	12
Slika 4.1 Teletekst modul u Comedia programskoj podršci .....	13
Slika 4.2 Primer korišćenja CLUT .....	14
Slika 4.3 Primer korišćenja True Color Bitmap-e .....	15
Slika 4.4 Dijagram razmene poruka prilikom iscrtavanja teleteksta .....	16
Slika 4.5 Programski koraci prilikom iscrtavanja teleteksta .....	18
Slika 4.6 Način korišćenja android deljene memorije .....	20
Slika 4.7 Dijagram razmene poruka prilikom iscrtavanja DTV prevoda .....	21
Slika 5.1 Teletekst .....	25
Slika 5.2 Transparentan teletekst.....	25
Slika 5.3 Skaliranje teleteksta .....	26
Slika 5.4 Prevod (eng. <i>Subtitle</i> ).....	26
Slika 5.5 MHEG aplikacija .....	27

## SKRAĆENICE

<b>API</b>	-	<i>Application programming interface</i> , Programska sprega
<b>CLUT</b>	-	<i>Colour Look-Up Table</i> , Tabela boja
<b>DTV</b>	-	<i>Digital Television</i> , Digitalna televizija
<b>EPG</b>	-	<i>Electronic program guide</i> , Elektronski programski vodič
<b>GUI</b>	-	<i>Graphical user interface</i> , Grafička korisnička sprega
<b>JNI</b>	-	<i>Java Native Interface</i> , Sprega Java programskog jezika i izvornog C koda
<b>PAT</b>	-	<i>Program Association Table</i> , Lista PID vrednosti TS paketa sa PMT tabelama
<b>PES</b>	-	<i>Packetized elementary stream</i> , Paket programskog toka
<b>PID</b>	-	<i>Packet Identifier</i> , Identifikator paketa
<b>PMT</b>	-	<i>Program Map Table</i> , Lista PID vrednosti TS paketa koji sadrže PES pridružene određenom programu
<b>PTS</b>	-	<i>Presentation time stamp</i> , Informacija o vremenskom trenutku kada je potrebno prezentovati sadržaj na koji se PTS odnosi (obično audio ili prevod)
<b>PVR</b>	-	<i>Personal video recorder</i> , Digitalni snimač
<b>STB</b>	-	<i>Set-top box</i> , Digitalni televizijski prijemnik
<b>TS</b>	-	<i>Transport stream</i> , Prenosni tok podataka
<b>TV</b>	-	<i>Television</i> , Televizija

## 1. Uvod

U ovom radu je opisano jedno rešenje prikaza sadržaja interaktivnih servisa digitalne televizije (prevod (eng. *Subtitles*), multimedijalni i teletekst sadržaji) u android grafičkoj ravnini korišćenjem izvornog C/C++ koda [1]. Za prikupljanje i obradu DTV (eng. *Digital television*) podataka korišćena je Comedia programska podrška za televizijske prijemnike (eng. *Middleware*) napisana u C programskom jeziku [2]. Rešenje je realizovano na Marvell BG2 SOC platformi sa ugrađenim biračem kanala (eng. *Tuner*).

Većina današnjih televizora i digitalnih TV prijemnika je programirana u nekom izvornom programskom jeziku (C/C++). Međutim, trenutni ugrađeni procesori imaju vise jezgara i veću radnu frekvenciju što omogućava korišćenje programskih jezika kao što je Java. Trenutni DTV uređaji su zasnovani na Linux operativnom sistemu i na njih se može instalirati Android operativni sistem koji nudi veliki izbor aplikacija za preuzimanje i instalaciju. Ključno pitanje je da li je moguće uključiti DTV u Android operativni sistem?

Jedan korak u rešavanju tog problema koji se odnosi na prikaz interaktivnih servisa digitalne televizije prikazan je u ovom radu.

Rezultati su ispitani korišćenjem Android aplikacije koja obezbeđuje GUI (eng. *Graphical user interface*) koji prikazuje sadržaj navedenih DTV servisa i omogućava interakciju korisnika sa tim sadržajem. Android aplikacija putem Java servisa komunicira sa programskom podrškom televizijskog prijemnika. Komunikacija Java servisa i izvornog C koda programske podrške televizijskog prijemnika se odvija putem JNI (eng. *Java Native Interface*) sloja [3].

Rad je organizovan na sledeći način:

U drugom poglavlju izložene su osnovne informacije o Android operativnom sistemu, principi digitalne televizije, realizacija teleteksta, prevoda i MHEG-a u digitalnoj televiziji i opis ciljne platforme.

U trećem poglavlju ukratko je opisan koncept iscrtavanja teleteksta, prevoda i MHEG-a u android grafičkoj ravni.

Četvrto poglavlje sadrži opis realizovanih programskih modula.

U petom poglavlju opisani su načini ispitivanja i verifikovanja navedenog programskog rešenja.

Šesto poglavlje predstavlja kratak pregled onoga što je realizovano u ovom radu.

U sedmom poglavlju dat je spisak korišćene literature za izradu rada.

## 2. Teorijske osnove

U ovom poglavlju izložene su osnovne informacije o Android operativnom sistemu, osnovni principi digitalne televizije, realizacija teleteksta, prevoda i MHEG-a u digitalnoj televiziji i opis ciljne platforme.

### 2.1 Android operativni sistem

Android operativni sistem je Linux zasnovani operativni sistem namenjen mobilnim uređajima kao što su pametni telefoni i tablet računari. U današnje vreme mnogi proizvođači mobilnih telefona koriste Android kao platformu. U narednim godinama očekuje se da će se Android nalaziti u milionima mobilnih telefona i drugim mobilnim uređajima, čineći Android glavnom platformom za aplikativne programere.

Zašto je Android operativni sistem toliko popularan? Šta ga čini tako posebnim?

Za razliku od ostalih operativnih sistema, kao što je na primer iOS, koji pokreće iPhone i koji je pod potpunom kontrolom kompanije Apple, Android funkcioniše kao operativni sistem otvorenog koda, tako da proizvođači mobilnih uređaja koji se odluče za ovu platformu mogu do izvesne mere da ga prilagode svojim potrebama. On omogućuje da se aplikacije koje su programirane za Android izvršavaju podjednako dobro na svim mobilnim telefonima koji koriste Android platformu, nezavisno od proizvođača. U ovom radu Android operativni sistem predstavlja komponentu programske podrške DTV uređaja.

### 2.2 Digitalna televizija

Digitalna televizija predstavlja prenos audio/video zapisa i dodatnih informacija u digitalnom formatu [4]. Pojava i uspostavljanje standarda koji se koriste u digitalnoj televiziji vezuje se za poslednju dekadu prošlog veka. Prva demonstracija digitalnog TV (eng. *Television*) prenosa održana je 1995. godine. Računari i savremene telekomunikacione mreže zasnovani su

na digitalnoj tehnologiji. Sa televizijom koja koristi istu tehnologiju ove tri oblasti se mogu kombinovati mnogo jednostavnije. Kao rezultat ovog kombinovanja digitalna televizija podržava nove aplikacije kao što je interaktivna televizija koja omogućava izvršavanje aplikacija pisanih u Javi, HTML-u, pristup Internetu. Digitalna televizija obezbeđuje mnogo bolji kvalitet slike i zvuka u odnosu na analognu, mogućnost izbora formata slike ili zvuka, omogućava uvođenje novih usluga kao što je izbor jezika za prevod, izbor audio kanala, multimedijalni sadržaji, EPG (eng. *Electronic program guide*), PVR (eng. *Personal video recorder*).

Najvažniji DTV standardi su:

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

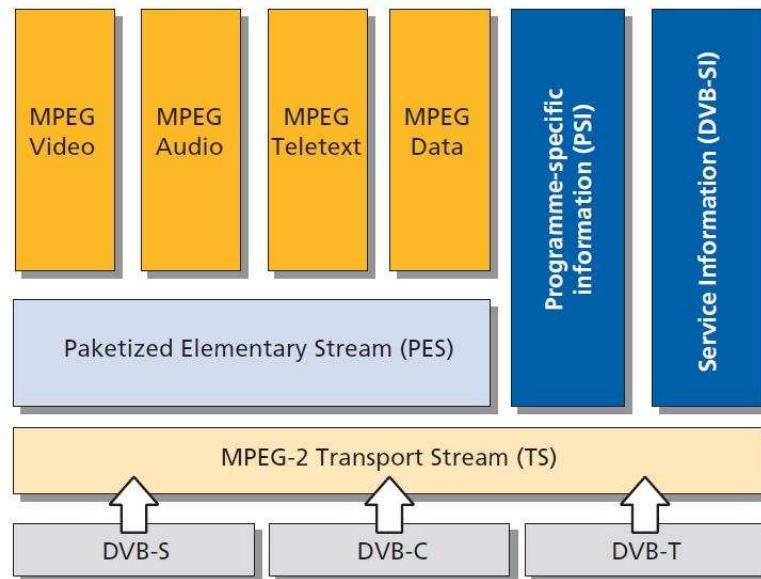


Slika 2.1 Najvažniji standardi digitalne televizije

## 2.3 Teletekst

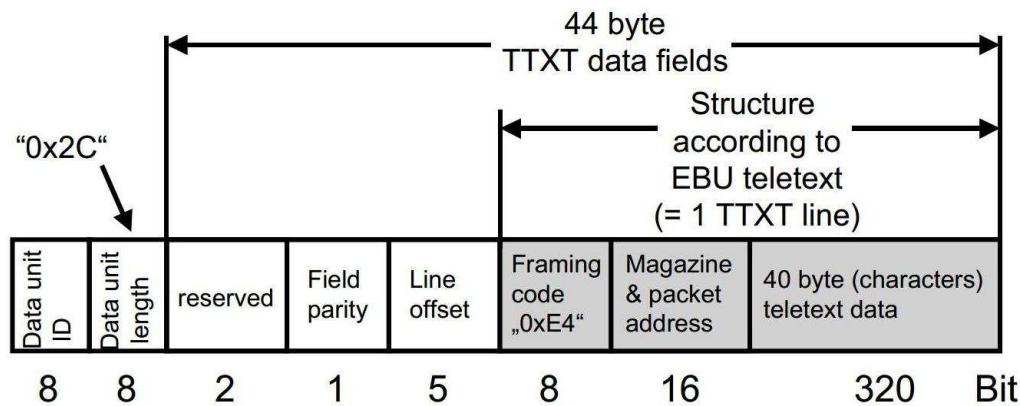
Teletekst je televizijski informacioni servis razvijen u Velikoj Britaniji 1970-tih godina. Nudi veliki izbor tekst-baziranih informacija, kao što su nacionalne, internacionalne i sportske vesti, vremenska prognoza i TV program. Osvrt na broj pregleda na dan – približno 9.5 miliona gledalaca za ARD i ZDF teletekst strane – ilustruje značajnost ovog servisa.

U analognoj televiziji teletekst je ubačen u VBI (eng. *Vertical blanking interval*). DVB standard koristi MPEG-2 (eng. *Moving Picture Experts Group*) standard koji definiše algoritme za kodovanje i dekodovanje audio/video podataka. Na slici 2.2 ilustrovana je lokacija teleteksta pored MPEG-2 - video, audio sadržaja i kanala podataka [5].



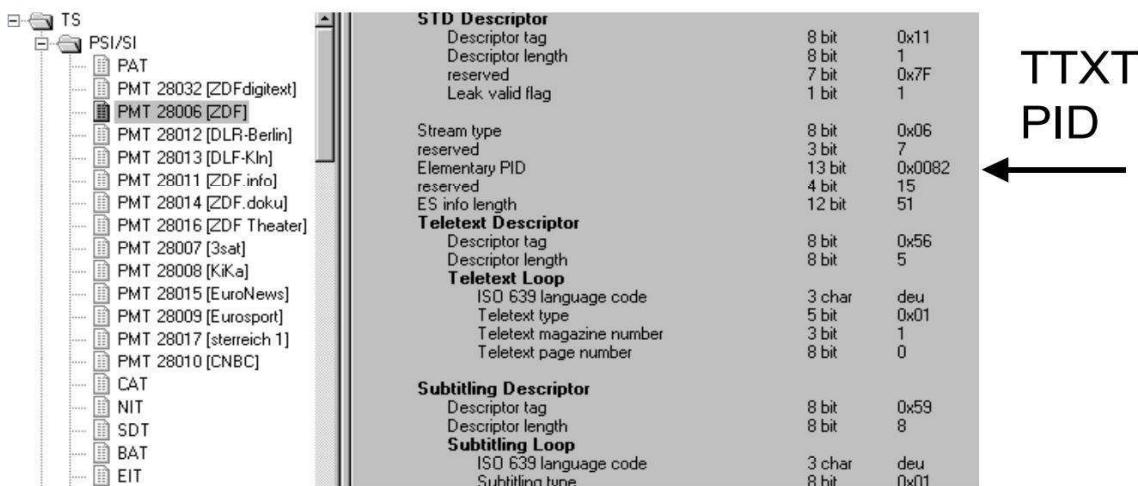
Slika 2.2 MPEG i DVB komponente

Teletekst je direktno ubačen u PES (eng. *Packetized elementary stream*) pakete. U slučaju teleteksta ovi paketi sadrže 45 bajtova tako da se cela linija teleteksta može preneti u jednom paketu (Slika 2.3). Jedan bajt predstavlja jedan karakter. Strana teleteksta se sastoji od 24 linije sa po 40 karaktera. Preostalih 5 bajtova su potrebni za teletekst kontrolne informacije, kao što su jezik i set karaktera.



Slika 2.3 Teletekst podaci u PES paketu

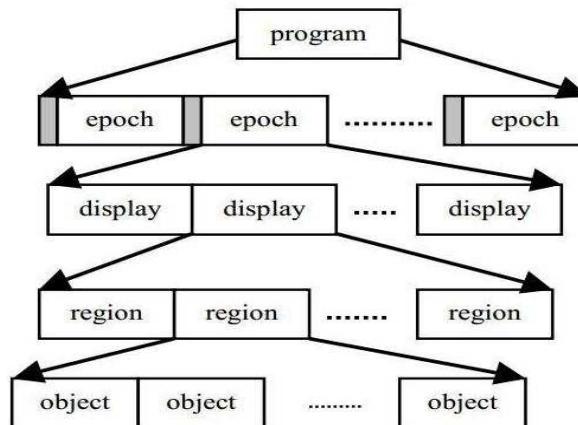
PID (eng. *Packet Identifier*) paketa programskega toka koji sadrže teletekst se nalazi u PMT (eng. *Program Map Table*) tabeli za odgovarajući servis (Slika 2.4).



Slika 2.4 PID teletekst servisa u PMT

## 2.4 Prevod (eng. *Subtitles*)

Digitalna televizija obezbeđuje gledaocima novi prevod (eng. *Subtitles*) servis sa više interaktivnosti nego analogni televizijski sistem. Gledaoci mogu korišćenjem daljinskih upravljača prikazivati ili sklanjati prevod, birati jezik prevoda. Prevod je kodiran i multipleksiran tako da formira prevod tok podataka, koji se prenosi u MPEG-2 prenosnom toku podataka u PES paketima do STB gde se tokovi podataka demultipleksiraju i dekoduju. Dekodovani prevod (u formi Bitmap-e) se zatim prikazuje na ekranu. DVB sistem za prevod koristi na regionima-zasnovanu, bit-mapiranu i indeksiranu kolor grafiku [6]. Za definisanje boja grafičkih elemenata koriste se palete boja (eng. *Colour Look-Up Table* (CLUT)). Slika 2.5 prikazuje logičku strukturu DVB prevoda. Svaki TV program se sastoji iz jedne ili više epoha. Prevod podaci jedne epohe moraju biti u memoriji prevod dekodera dok se ne signalizira dekodiranje sledeće epohe. Epoха uključuje sekvencu od jednog ili više prikaza. Prikaz može ostati na ekranu nekoliko sekundi. Svaki prikaz se sastoji od regiona sa specifičnom pozicijom.

Slika 2.5 Hierarchyjska struktura DTV prevoda (eng. *Subtitles*)

Region je pravougaona oblast ekrana sa horizontalnim i vertikalnim dimenzijama i dubinom piksela. Svaki region je povezan sa CLUT, koja definiše boju i transparentnost svakog piksela. Dubine piksela od 2, 4 i 8- bita su podržane dozvoljavajući da 4, 16, ili 256 različitih piksela bude korišćeno u svakom regionu. Svaki region se sastoji od grafičkih objekata kao što su prevod, logo kanala, mapa...

DVB prevodi se prenose kao PES paketi u MPEG prenosnom toku podataka [4]. Kompletna lista svih programa dostupnih u prenosnom toku podataka se nalazi u PAT tabeli (eng. *Program Association Table*). Svaki program sadrži broj programa i PID paketa programskega toka koji sadrži PMT. PMT sadrži PID-ove PES paketa koji sadrže video, audio ili podatke (npr. prevod (eng. Subtitles)) tog programa. Segmenti prevoda se nalaze u PES paketima. U zagлављу PES paketa se nalazi polje PTS (eng. *Presentation time stamp*) koje označava vreme pojavljivanje prevoda sadržanog u tom PES paketu na ekranu.

## 2.5 MHEG (eng. *Multimedia and Hypermedia Experts Group*)

MHEG je međunarodni standard za prezentaciju multimedijalnih informacija nastao 1995. godine [7]. Prisutan je na vise tržišta: UK, Novi Zeland, Malezija, Irska, Australija, Severna Afrika i Hong Kong. MHEG ima svoj objektno bazirani programski jezik koji omogućava:

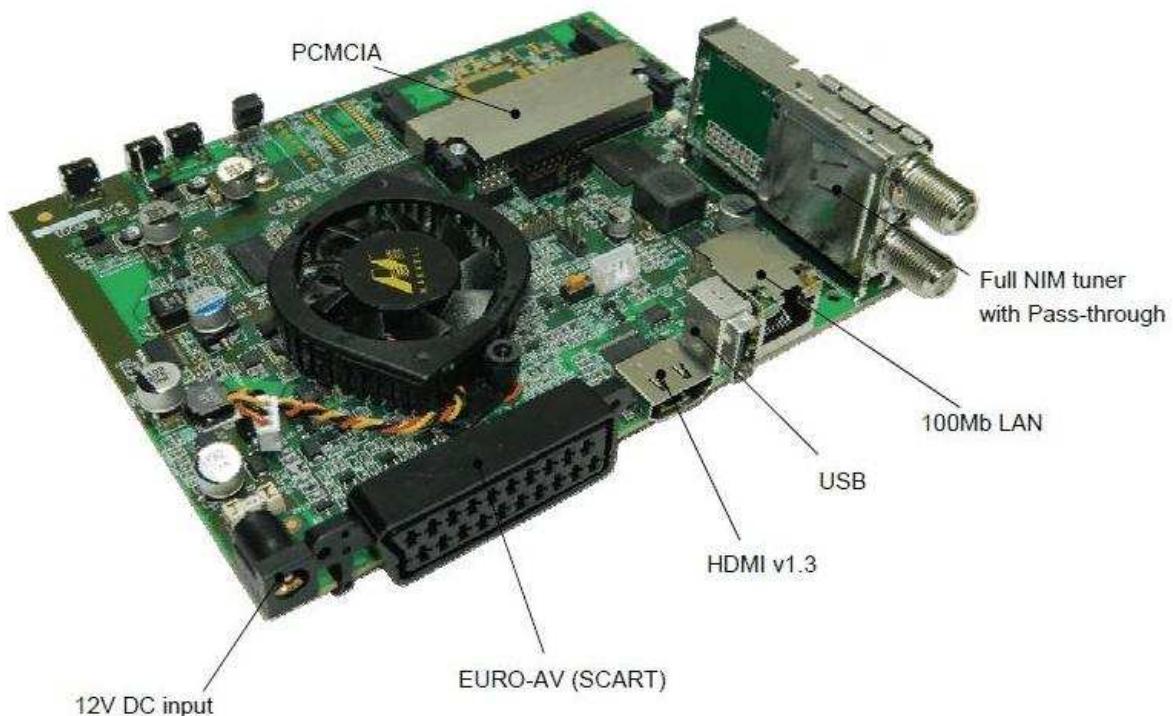
- Kontrolu nad prezentacijom sadržaja kao što su: audio, video, tekst i grafički objekti
- Interakciju korisnika sa MHEG aplikacijom
- Podršku za video i audio prezentaciju u realnom vremenu

MHEG aplikacije omogućavaju prikaz tekstualnih informacija kao što su: vesti, vreme, finansijski izveštaji i sportski rezultati. Kupovina od kuće (eng. *Home shopping*) takođe može biti podržana. MHEG može podržati i veliki opseg jednostavnih igrica, uključujući one koje zahtevaju od gledaoca aktivno korišćenje daljinskog upravljača. Mnoge igrice popularne na mobilnim telefonima mogu biti realizovane.

Kada je upaljen, MHEG preuzima kontrolu nad skaliranjem i prezentacijom video i audio sadržaja. To omogućava više od jednog pogleda na isti događaj – na primer posmatranje različitih mečeva na teniskom turniru, različitih igrača na golf turniru ili posmatranje fudbalske utakmice iz različitih uglova.

## 2.6 Opis ciljne platforme

Rešenje iscrtavanja interaktivnih servisa digitalne televizije predstavljeno u ovom radu realizovano je na Marvell BG2 SOC platformi na kojoj je instaliran Android operativni sistem.(Slika 2.6)



Slika 2.6 Marvell BG2 SOC platforma

Ključne komponente platforme su:

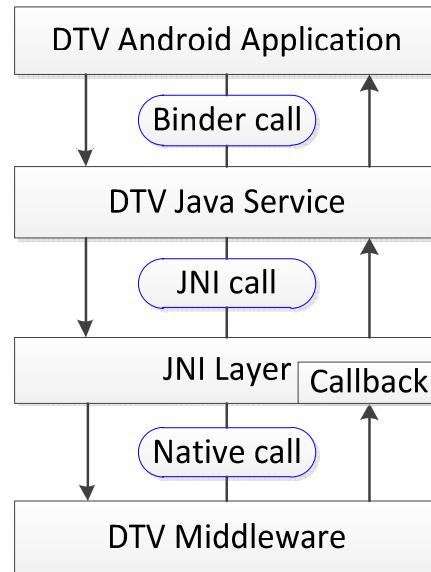
- Satelitski birač kanala Serit SP2230 – koristi se za preuzimanje signala sa satelita
- CIMax čip proizvođača SmartDTV – koristi se za usmeravanje prenosnog toka od birača kanala do dekodera
- Dva modula uslovnog pristupa – koriste se za dešifrovanje zaštićenog prenosnog toka
- Procesor ARM familije, Marvell ARMADA 1500
- DDR3 operativna memorija Samsung K4B2G1646C-HCK0 – memorija koja se stavlja na raspolaganje sistemu
- NAND memorija Samsung K9GBG08U0A – memorija koja se koristi za smeštanje aplikacije na ciljnoj platformi

### 3. Koncept rešenja

Rešenje prikaza interaktivnih servisa digitalne televizije (teletekst, prevod, MHEG) u android grafičkoj ravni počinje sa Android aplikacijom (Java aplikacija) koja se ponaša kao DTV aplikacija. Aplikacija koristi regularne Android komponente kao što su VideoView komponenta za prikaz video sadržaja i Dialog komponenta za prikaz teleteksta, prevoda i MHEG-a [8]. Takođe koristi DTV Java API (eng. *Application programming interface*) za rad sa DTV strukturama podataka kao što su Service (informacije o servisu: ime servisa, indeks servisa, tip servisa...), Service List (Lista servisa), EpgEvent itd.

DTV Android aplikacija je povezana sa DTV Java servisom, koji takođe koristi isti DTV Java API. Ovaj servis ima u sebe ugrađen JNI sloj kao android deljenu biblioteku koji služi za komunikaciju Java servisa sa Comedia programskom podrškom. Comedia programska podrška takođe predstavlja deljenu biblioteku ugrađenu u DTV Java servis. Ovo znači da se metode i klase Java API-ja prevode u izvorne (C/C++) funkcije i strukture.

DTV Java servis je izložen DTV Android aplikaciji kroz Binder (mehanizam za komunikaciju između procesa specifičan za Android) kao što je uobičajeno u Android OS [9]. Kada se prozove DTV metoda Android aplikacije ona proziva odgovarajuću metodu Java servisa koja se zatim propagira ka JNI sloju koji će prozvati odgovarajuću funkciju DTV programske podrške (eng. *Middleware*). U suprotnom smeru, kada se desi događaj u DTV programskoj podršci, DTV programska podrška poziva povratnu metodu (eng. *Callback*) u JNI sloju, koji poziva odgovarajuću Java metodu u Java servisu (Slika 3.1). Ako je neophodno ovaj sloj zatim izaziva odgovarajuću metodu u Java aplikaciji.



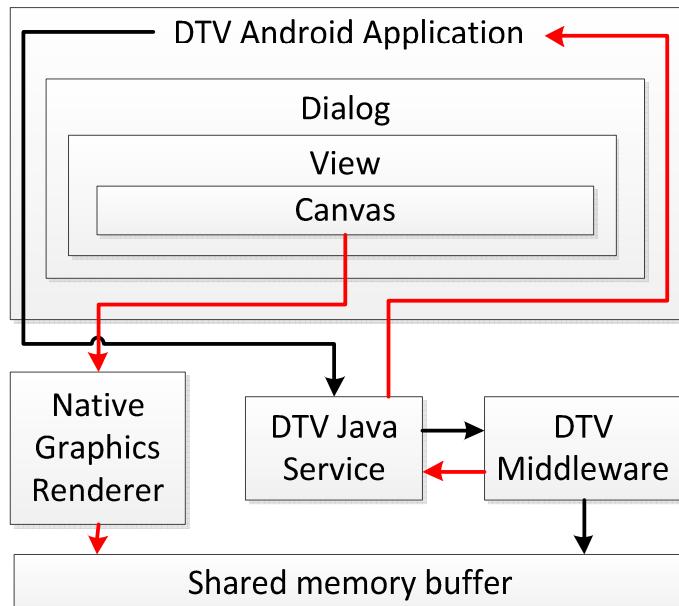
Slika 3.1 Programski slojevi za integraciju DTV u Android OS

Teletekst, prevod i MHEG su implementirani u DTV programskoj podršci. DTV programska podrška sakuplja teletekst, prevod i MHEG podatke iz TS (eng. *Transport stream*, prenosni tok podataka) i stvara odgovarajuće Bitmap-e. Postavlja se pitanje kako prikazati ove Bitmap-e u android grafičkoj ravni? Najjednostavniji način da se to postigne je slanje odgovarajućih Bitmap-a iz DTV programske podrške (C programski kod) kroz JNI sloj do Java servisa koji će zatim proslediti Bitmap-e Android aplikaciji kako bi se u Javi izvršilo iscrtavanje. Ovaj način iscrtavanja je veoma spor i nepouzdan zbog slanja velikih količina podataka od DTV programske podrške, preko JNI sloja, zatim kroz Java servis do Java aplikacije. Zbog toga se iscrtavanje mora izvršiti u izvornom C/C++ kodu koji obezbeđuje dovoljnu brzinu izvršavanja programa i zadovoljava visoke zahteve za performansama koje Java ne može da ispunи.

Rešenje prikazano u ovom radu sastoji se u konverziji odgovarajućih Bitmap-a u Andoid podržane formate i crtanjem tih Bitmap-a na Canvas odgovarajućeg Android Dialog-a (npr. TeletextDialog) u izvornom C/C++ kodu. Canvas je Java klasa koja sadrži sve metode za crtanje. Da bi se nešto nacrtalo u Android OS, potrebne su 4 osnovne komponente: Bitmap-a da drži piksele, Canvas da prihvata pozive crtanja, primitiva za crtanje (pravougaonik, tekst, Bitmap) i Paint (da opiše boje i stilove za crtanje). Android API obezbeđuje set 2D funkcija (drawBitmap(...), drawRect(...), drawText(...)) za crtanje koje omogućavaju iscrtavanje grafike na Canvas-u [10].

Slika 3.2 prikazuje programske slojeve koji su uključeni u iscrtavanje teleteksta, prevoda i MHEG-a u DTV Android aplikaciji. DTV Android aplikacija kreira Dialog koji ima svoj View koji sadrži Canvas (Dialog, View i Canvas su bazični gradivni blokovi za korisnički interfejs u Androidu. Dialog je obično mali prozor koji se pojavljuje ispred trenutno aktivne aplikacije.

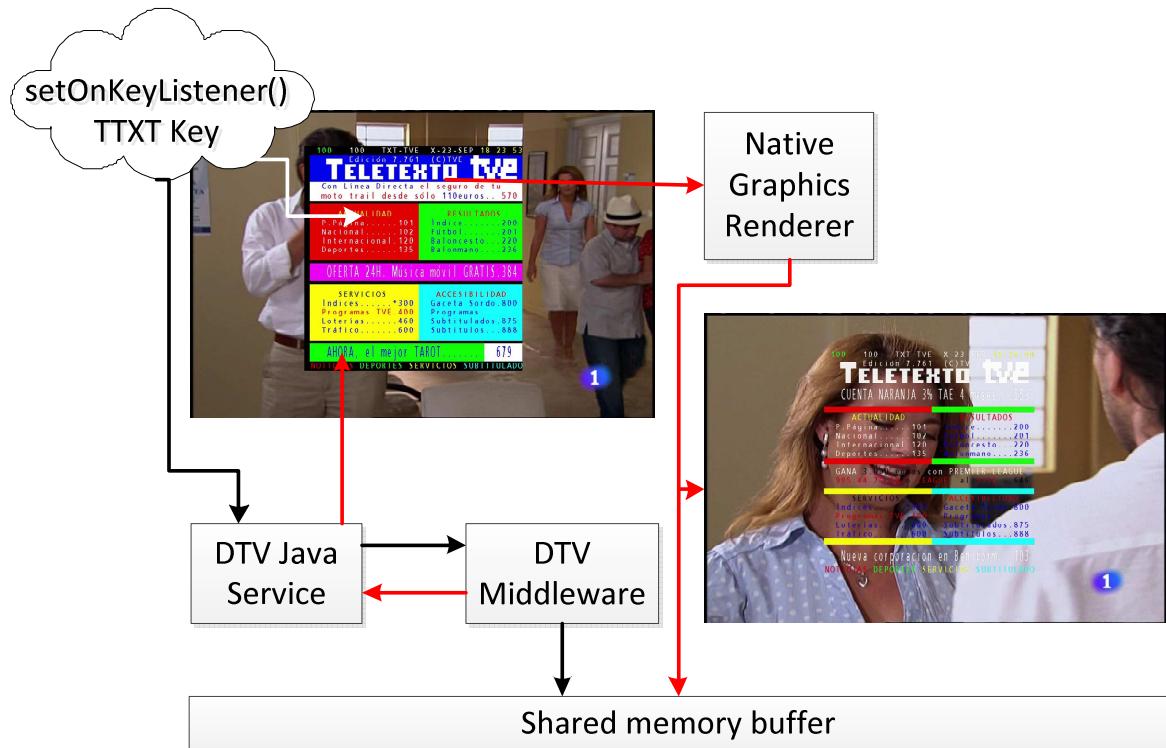
Kada je aktivan Dialog, aplikacija gubi fokus i Dialog prihvata sve korisničke interakcije. View zauzima pravougaonu oblast na ekranu i odgovoran je za crtanje i rukovanje događajima [8]). Canvas objekat stvorenog Android Dialog-a se spušta kroz JNI do Native Graphics Renderer modula. Ovaj programski modul napisan u C++ programskom jeziku predstavlja deo DTV Android aplikacije koji je zadužen za iscrtavanje teletekst, prevod i MHEG Bitmap-a. Bitmap se stvara u DTV programskoj podršci i upisuje u Android deljenu memoriju. Kada je Bitmap-a stvorena i upisana u deljenu memoriju, DTV programska podrška prouzrokuje povratnu metodu (eng. *callback*) u Java Servisu kroz JNI sloj (Slika 3.2). Ova metoda proziva povratnu metodu u DTV Android aplikaciji koja će naterati Native Graphics Renderer modul da iscrtava Bitmap-u na Canvas Android Dialog-a (Bitmap se nalazi u android deljenoj memoriji). Time je iscrtavanje završeno i teletekst, prevod ili MHEG su prikazani u DTV Android aplikaciji.



Slika 3.2 Programski slojevi namenjeni iscrtavanju teleteksta, prevoda i MHEG-a

Interakcija korisnika i interaktivnih servisa digitalne televizije se odvija zahvaljujući KeyListener-u (KeyListener je sprega (eng. *Interface*) za prihvatanje događaja sa tastature ili daljinskog upravljača) koji se kači na odgovarajući dijalog (Teletext, Subtitle, MHEG Dialog) [8]. Korisnik putem daljinskog upravljača salje komande Android aplikaciji koje hvata KeyListener zakačen na dijalog i inicira poziv odgovarajuće metode DTV Java servisa koji dalje kroz JNI sloj proziva odgovarajuću metodu DTV programske podrške koja će izvršiti zahtevanu komandu (Slika 3.3). U slučaju teleteksta interakcija obuhvata: prikaz teleteksta, promenu strane teleteksta, menjanje transparentcije teleteksta, skaliranje teleteksta, pomeranje teleteksta, sakrivanje teleteksta. Kod prevoda moguće opcije su: prikaz prevoda, promena jezika, sklanjanje prevoda sa ekrana. Navigacija kroz MHEG aplikacije se takođe odvija pritiskom dugmeta na

daljinskom upravljaču uključujući: Up, Down, Left, Right i alfa-numeričke dugmiće. Upotreba dugmića u boji na daljinskom upravljaču (Red, Green, Yellow, Blue) olakšava korišćenje teleteksta i MHEG aplikacija.



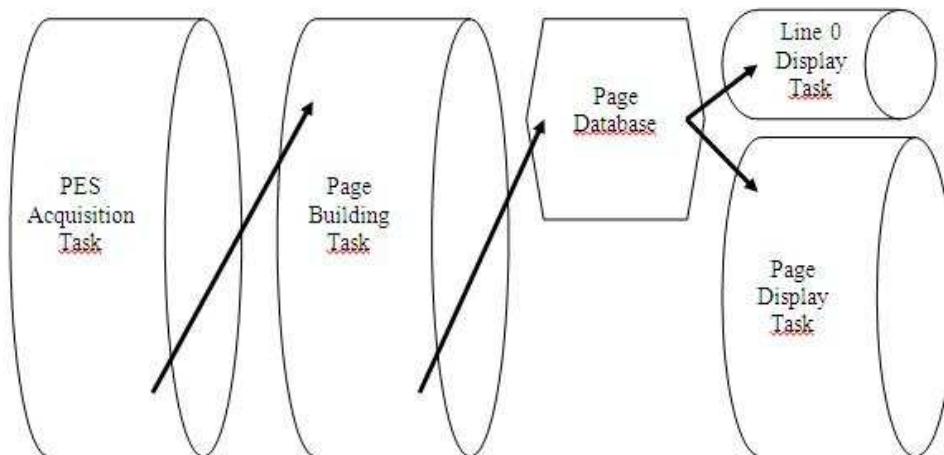
Slika 3.3 Interakcija korisnika i interaktivnih servisa digitalne televizije

## 4. Programsко rešenje

### 4.1 Realizacija iscrtavanja Teleteksta

Teletekst modul u Comedia programskoj podršci (eng. *DTV Middleware*) podeljen je u 4 dela kao što je prikazano na slici 4.1:

- Prikupljanje PES paketa
- Formiranje stranica teleteksta
- Prikazivanje stranica
- Baza podataka teletekst stranica



Slika 4.1 Teletekst modul u Comedia programskoj podršci

PES paketi koji sadrže teletekst podatke se upisuju u kružni bafer.

Formiranje stranica teleteksta započinje pretraživanjem kružnog bafera. Svi paketi koji pripadaju istoj stranici se dekoduju i stavljuju zajedno da bi formirali stranicu. Kada se formira, stranica se smešta u bazu podataka.

Kada korisnik zatraži odgovarajuću stranicu teleteksta, ona se preuzima iz baze stranica i biva prikazana od strane modula za prikazivanje stranica. Ovaj modul rukuje i osvežavanjem tako što prikazuje samo promenjene linije teletekst stranice. Prva linija teletekst stranice se neprestano osvežava (očitavanje stranice i sat).

Kada je potrebno iscrtati pripremljenu stranicu teleteksta koja je u formi Bitmap-e Comedia programska podrška inicira poziv funkcije TDAL\_GFX\_BmpDraw(...) u TDAL\_GFX modulu. Parametri ove funkcije su:

```
tTDAL_GFX_Point offset – Pozicija na kojoj se Bitmap-a nalazi na regionu za iscrtavanje
tTDAL_GFX_Bitmap *pBitmap – Pokazivač na strukturu koja sadrži Bitmap-u za crtanje

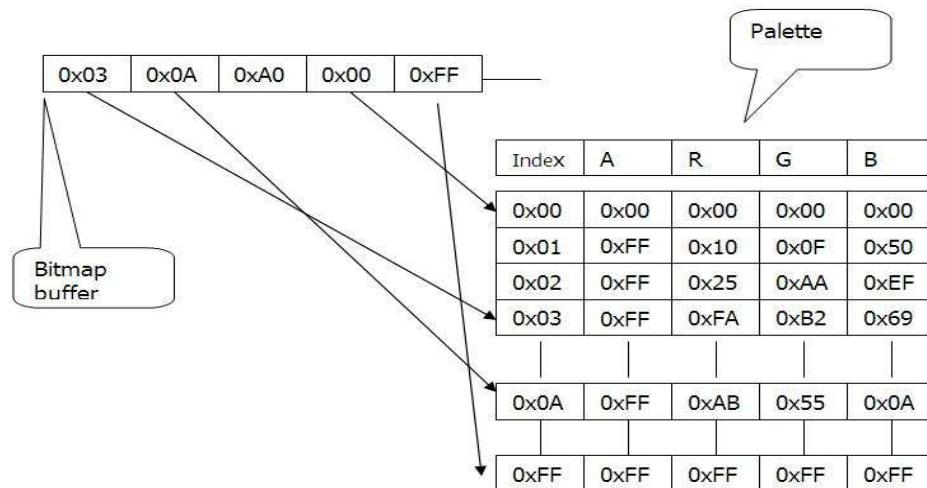
typedef struct
{
    int32_t      x;
    int32_t      y;
}tTDAL_GFX_Point;

typedef struct
{
    tTDAL_GFX_ColorType    colorType;
    tTDAL_GFX_Size         size;
    void*     pData;
}tTDAL_GFX_Bitmap;
```

Polje colorType pokazuje u kom formatu stižu podaci (pData) Bitmap-e (neki od formata piksela su: CLUT\_AYCRCB8888, CLUT\_ARGB8888, TRUE\_COLOR\_AYCRCB8888, TRUE\_COLOR\_ARGB8888 ...).

Kada se koristi CLUT Mod (model palete) Bitmap-a za prikaz daje indeks boje u paleti za svaki piksel (Jedan bajt po pikselu).

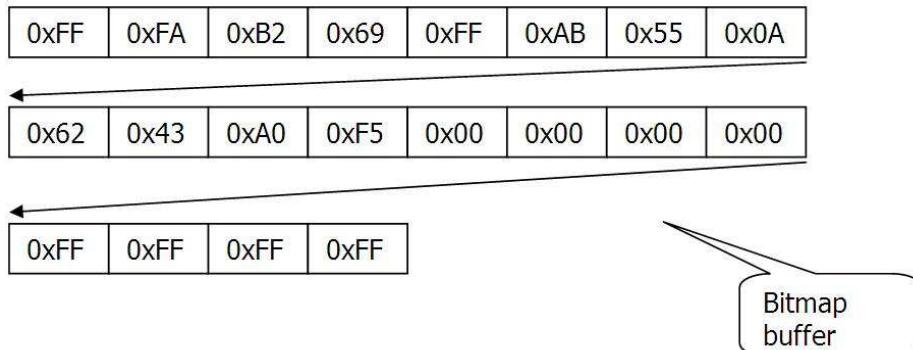
Primer:



Slika 4.2 Primer korišćenja CLUT

Kada se koristi True Color Mod, Bitmap-a se koduje sa „pravom“ bojom piksela, tri bajta za svaku komponentu + jedan bajt za transparentnu vrednost (Četiri bajta po pikselu).

Primer:



Slika 4.3 Primer korišćenja True Color Bitmap-e

U slučaju teleteksta colorType = CLUT\_ARGB8888 tako da pData predstavlja niz uint8\_t[size.width\*size.height] gde svaki element niza predstavlja jedan piksel (odnosno indeks u odgovarajućoj CLUT tabeli). Alpha, R, G i B komponenta svakog piksela se dobijaju čitanjem vrednosti CLUT tabele na odgovarajućoj poziciji. Npr. R, G, B i Alpha komponente prvog piksela Bitmap-e se dobijaju na sledeći način:

```
R = TTXT_palette.pColor[*((uint8_t *)pData)].ARGB8888.R;
G = TTXT_palette.pColor[*((uint8_t *)pData)].ARGB8888.G;
B = TTXT_palette.pColor[*((uint8_t *)pData)].ARGB8888.B;
Alpha = TTXT_palette.pColor[*((uint8_t *)pData)].ARGB8888.alpha;
```

Originalni format teletekst Bitmap-e koji je indeks baziran se konvertuje izborom odgovarajuće boje u tabeli boja na osnovu prosleđenog indeksa i smeštanjem sve četiri komponente (R, G, B i Alpha) kao četiri bajta u zauzeti bafer (pDataRGBA) koji je 4 puta veći od veličine pData (svaki piksel se sastoji od 4 bajta).

Jedna strana teleteksta se sastoji od više ovakvih Bitmap-a visine pBitmap->size.height (21 ili 42 piksela). To znači da se funkcija TDAL\_GFX\_BmpDraw(...) poziva više puta prilikom iscrtavanja jedne stranice teleteksta. Informacija o položaju prosleđene Bitmap-e u regionu je sadržana u argumentu offset, funkcije TDAL\_GFX\_BmpDraw(...).

Po završenoj konverziji dobijeni bafer (pDataRGBA) koji sadrži piksele Bitmap-e (u formi TRUE\_COLOR\_RGBA8888) se prosleđuje u funkciju:

```
drawTeletext(offset.x,offset.y,pBitmap->size.width, pBitmap->size.height, pDataRGBA);
```

U ovoj funkciji se vrši kopiranje teletekst Bitmap-e na odgovarajuću poziciju u bafer deljene memorije (alociran iz ASHMEM modula), tako da mu DTV Android aplikacija može pristupiti radi iscrtavanja.

---

```
memcpy(&(base[y*width*4]), pDataRGBA, height*width*4); //base – pokazivač na bafer deljene memorije zauzet za teletekst Bitmap-u
```

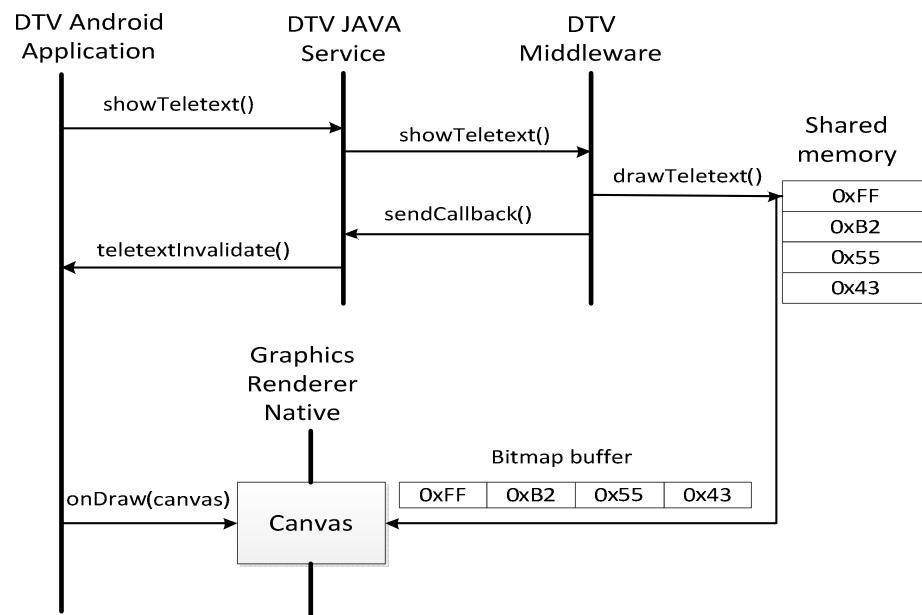
Nakon kopiranja piksela Bitmap-e u bafer deljene memorije poziva se funkcija sendCallback(...) koja obaveštava DTV Android aplikaciju da je teletekst Bitmap-a pripremljen (Slika 4.4). U funkciji sendCallback() trenutna nit se mora zakačiti za Java Virtuelnu mašinu kako bi se dobio pokazivač na JNI interfejs [3]:

```
(*cached_jvm)->AttachCurrentThread(cached_jvm,(void**)&env, NULL);
```

i zatim pozvati odgovarajuća funkcija DTV Java servisa: (\*env)->CallStaticVoidMethod(...); (poziva se Java funkcija callbackCall(...)).

Sve do sada opisano predstavlja izvorni C kod koji je ugrađen u Comedia programsku podršku. Ovaj kod se nalazi u okviru DTV Java servisa u formi deljene biblioteke (Android shared library).

Funkcija callbackCall(...) u DTV Java servisu će inicirati poziv funkcije teletextInvalidate() DTV Android aplikacije. teletextInvalidate() inicira osvežavanje View-a teletekst dijaloga pozivom funkcije postInvalidate() koja poziva funkciju TeletextDialogView.onDraw(Canvas canvas) i Canvas argument se prosleđuje u nativni sloj (u modul Graphics Renderer Native kao što je prikazano na slici 4.4). Ovaj nativni sloj će izvršiti crtanje teletekst Bitmap-e iz bafera deljene memorije na Canvas.



Slika 4.4 Dijagram razmene poruka prilikom iscrtavanja teleteksta

---

Ključni elemenat prilikom iscrtavanja je Canvas objekat. Ovaj objekat se prosleđuje iz Java u nativni sloj kroz funkciju:

```
Java_android_dtv_graphics_GraphicsRendererNative_draw (... jobject canvas,...);
```

Funkcija GraphicsRendererNative\_draw() predstavlja deo Android deljene biblioteke koja je uključena u DTV Android aplikaciju. Namena funkcije je da od piksela Bitmap-e upisanih u bafer deljene memorije formira objekat Bitmap koji se iscrtava na Canvas.

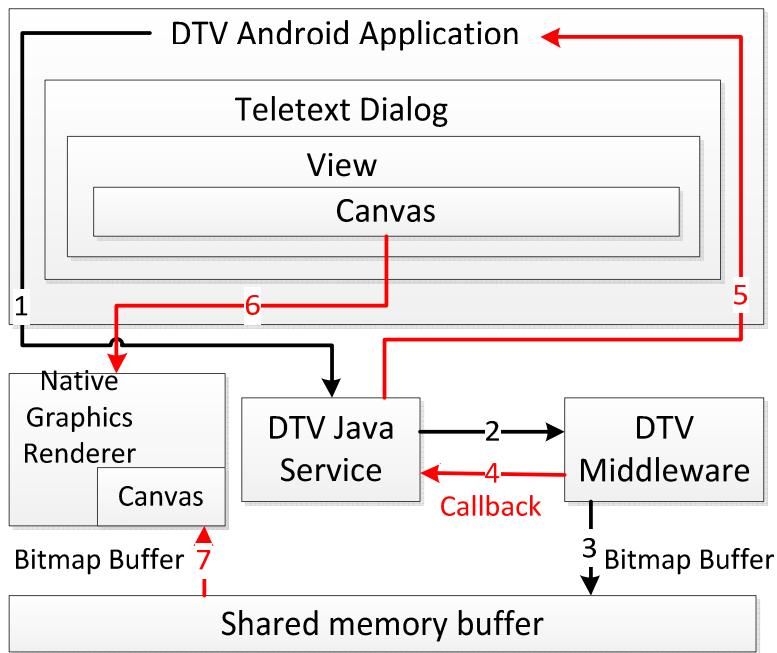
```
SkBitmap bmTeletext;
```

```
bmTeletext.setConfig(SkBitmap::kARGB_8888_Config, 600, 600, 0);  
bmTeletext.setPixels(&(base[0]), NULL); //base – pokazivač na deljenu memoriju  
canvas->drawBitmap(bmTeletext,0,0, paint);
```

Funkcije za formiranje Bitmap-e od piksela i nativno iscrtavanje Bitmap-e na Canvas (drawBitmap(), setPixels()), kao i ostale funkcije za rad sa objektima SkCanvas i SkBitmap su implementirane u Androidovoj deljenoj biblioteci libskia.so koja je uključena u modul za iscrtavanje teleteksta. (SKIA predstavlja grafički podsistem koji Android koristi. To je grafička biblioteka niskog nivoa implementirana u nativnom kodu koja rukuje iscrtavanjima u Android grafičkoj ravni [11]).

Procedura za iscrtavanje teleteksta u android grafičkoj ravni je prikazana na slici 4.5.

- Korisnik inicira prikaz teleteksta u DTVAndroid aplikaciji
- DTV Android aplikacija formira TeletextDialog
- DTV programska podrška kopira piksele Bitmap-e u bafer deljene memorije i obaveštava DTV Java servis
- DTV Java servis poziva povratnu metodu DTV Android aplikacije
- DTV Android aplikacija osvežava TeletextDialog (prosleđuje Canvas dijaloga u izvorni C++ kod)
- Izvorni C++ kod formira Bitmap-u na osnovu bafera deljene memorije koju zatim iscrtava na Canvas-u



Slika 4.5 Programski koraci prilikom iscrtavanja teleteksta

Interakcija korisnika sa teletekstom se odvija na sledeći način:

U DTV android aplikaciji na teletekst Dialog se kači Key Listener koji prihvata komande sa daljinskog upravljača. Kada korisnik pritisne neki broj na daljinskom Key Listener će prozvati metodu `sendInputControl(key)` DTV Java servisa koja će inicirati poziv funkcije `cmSPPlayer_SelectTeletextPage (key)` DTV programske podrške. Kada DTV programska podrška prihvati sve tri cifre (tri cifre označavaju jednu stranicu teleteksta) ona započinje crtanje nove stranice teleteksta tako što upisuje novu Bitmap-u teleteksta u deljenu memoriju i javlja DTV Android aplikaciji da je nova Bitmap-a spremna i da se može izvršiti njeno iscrtavanje na Canvas teletekst dijaloga.

Kada korisnik zatraži promenu dimenzija teleteksta DTV Android aplikacija proziva funkciju `scale(width, height)` koja inicira poziv nativne funkcije u biblioteci Graphic Renderer Native koja dovodi do poziva metode `scale` (SKIA metoda) nad Canvas-om teletekst dijaloga:

```
canvas->scale(scaleFactorX, scaleFactorY);
```

Prikaz i transparencija teleteksta se postižu pozivom Java funkcije `showTeletext(int alpha)`. Ako se kao argument funkcije prosledi 255 teletekst će biti potpuno transparentan. 0 označava da teletekst neće biti transparentan.

## 4.2 Bafer deljene memorije (Anonymous SHared MEMory subsystem)

DTV programska podrška koja je deo DTV Java servisa upisuje piksele teletekst Bitmap-e u memoriju odakle ih mora čitati DTV Android aplikacija koja predstavlja sasvim različit proces. Postavlja se pitanje kako ostvariti navedeno deljenje memorije između različitih procesa u Android OS? Kao rešenje navedenog problema u android okruženju nameće se korišćenje Ashmem-a (eng. *Anonymous SHared MEMory subsystem*).

Ashmemp je android deljena memorija dodata na Linux kernel. Ashmem podržava brojanje referenci tako da ako mnogo procesa koristi isto područje memorije, područje neće biti uklonjeno sve dok ga svi procesi ne oslobole. Ashmem se zauzima funkcijom `ashmem_create_region()` koja vraća File Descriptor. Ključno pitanje koje se postavlja je kako razmenjivati ovaj File Descriptor između procesa? File Descriptor je namenjen jednom procesu i ne može se deliti između procesa samo tako. Rešenje je deljenje File Descriptor-a uz pomoć Binder-a pošto Binder ima specijalne funkcije koje podržavaju prenos File Descriptor-a. Binder je mehanizam za komunikaciju između procesa specifičan za Android OS. Android proces može pozvati rutinu drugog Android procesa, korišćenjem Binder-a koji identificuje metodu koja će se prozvati i prenosi argumente između procesa. Kao pomoć pri korišćenju Ashmem-a i deljenju File Descriptor-a koristi se klasa MemoryHeapBase. Deljena memorija se zauzima na sledeći način:

```
sp<MemoryHeapBase> mMemHeap;
mMemHeap = new MemoryHeapBase(MEMORY_SIZE);
```

Da bi se preneo objekat MemoryHeapBase sa serverske strane koristi se metoda `asBinder()` nad IMemoryHeap spregom (eng. *Interface*):

```
sp<IMemoryHeap> memHeap = ...
```

`reply->writeStrongBinder(memHeap->asBinder()); //asBinder() – vraća Binder objekat povezan sa spregom (eng. Interface)`

Na klijentskoj strani objekat MemoryHeapBase se dobija na sledeći način:

```
sp<IMemoryHeap> memHeap;
memHeap = interface_cast<IMemoryHeap>(reply.readStrongBinder());
```

Bafer deljene memorije za teletekst, prevod i MHEG Bitmap se zauzima u nativnom Android servisu (Ashmem Service Server modul) napisanom u C++ programskom jeziku. Ovaj nativni servis se startuje iz DTV Java servisa pri inicijalizaciji DTV programske podrške. On predstavlja izvršni (eng. *Executable*) servis koji radi u pozadini. Namena mu je da zauzme

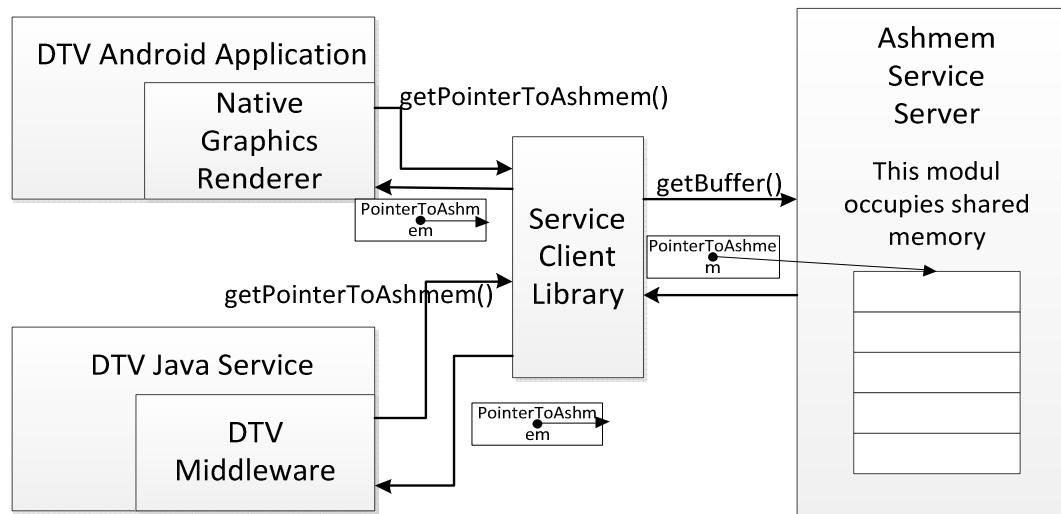
deljenu memoriju i da izloži funkciju za dobavljanje pokazivača na tu memoriju klijentskim aplikacijama preko Binder-a na sledeći način:

```
defaultServiceManager()->addService(String16("vendor.droid.Buffer"),new  
DroidBufferService());
```

Ovim je servis zakačen na Binder tako da druge aplikacije mogu da ga pozivaju preko identifikatora: "vendor.droid.Buffer".

Servis preko Binder-a izlaže funkciju `getBuffer()` koja vraća pokazivač na zauzetu deljenu memoriju klijentskim aplikacijama.

Kako bi klijentske aplikacije (DTV programska podrška (deo DTV Java servisa) koja upisuje Bitmap u deljenu memoriju i Native Graphics Renderer (deo DTV Android aplikacije) koji čita Bitmap-u iz deljene memorije) pristupile deljenoj memoriji svaka od njih treba da uključi klijentsku biblioteku (Service Client Library). Ova biblioteka se kači na nativni servis (Ashmem Service Server) koji je zauzeo deljenu memoriju (`defaultServiceManager()->getService(String16("vendor.droid.Buffer"))`) i poziva njegovu funkciju `getBuffer()` kako bi dobila pokazivač na zauzetu memoriju koji će dalje koristiti klijentske aplikacije (DTV programska podrška i Native Graphics Renderer) što je prikazano na slici 4.6.

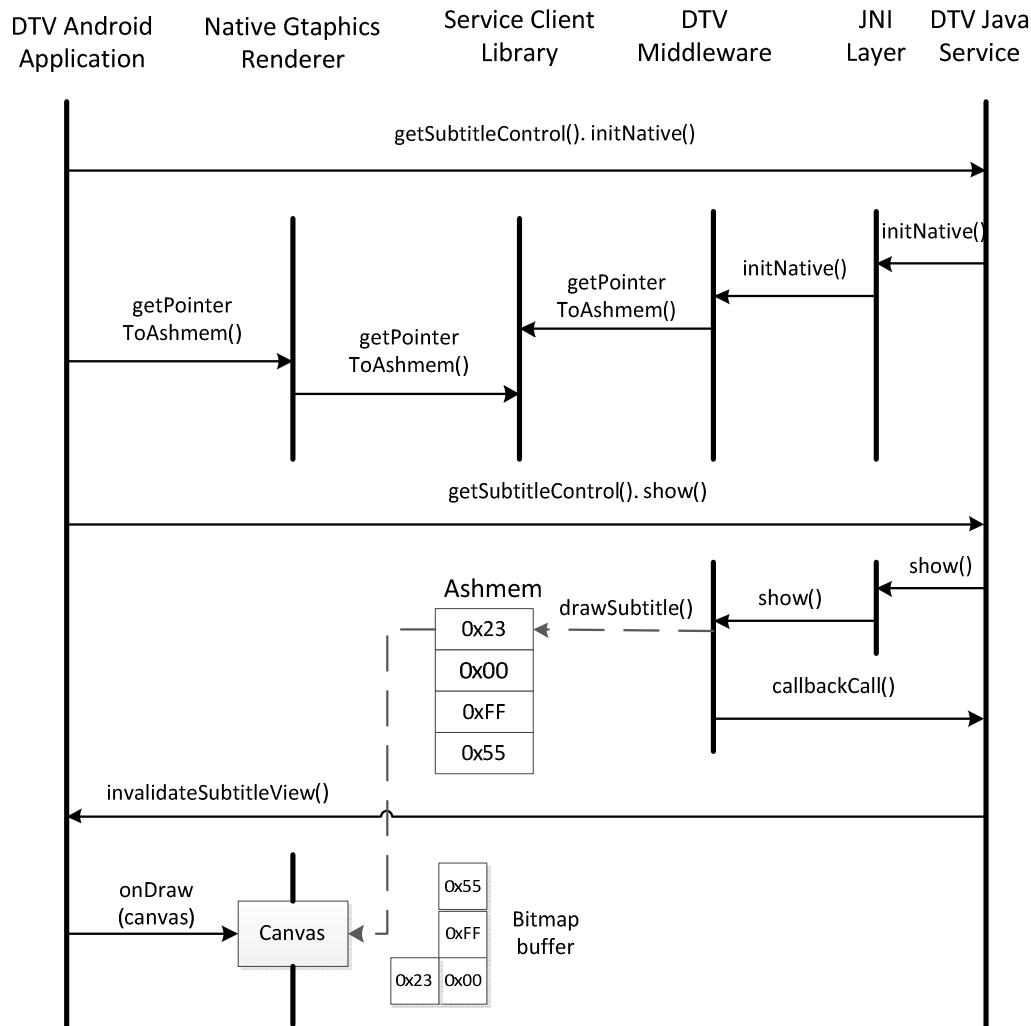


Slika 4.6 Način korišćenja android deljene memorije

### 4.3 Realizacija iscrtavanja prevoda (eng. *Subtitles*)

Kada korisnik zatraži prikaz prevoda, DTV Android aplikacija stvara dijalog određenih dimenzija i određenog položaja na ekranu (SubtitleDialog). Potom poziva funkciju

getSubtitleControl().initNative() DTV Java servisa koja će putem JNI sloja inicirati odgovarajuću nativnu funkciju u DTV programskoj podršci koja će preuzeti pokazivač na deljenu memoriju u koju DTV programska podrška upisuje Bitmap piksele prevoda (Pokazivač se obezbeđuje pozivom funkcije getPointerToAshmem() koja je implementirana u deljenoj biblioteci Service Client Library. U funkciji getPointerToAshmem() klijent se kači na nativni C++ servis koji radi u pozadini i koji je zauzeo deljenu memoriju (objašnjeno u poglavlju 4.2). Kada se klijent zakači na servis poziva se metoda servisa getBuffer() koja obezbeđuje pokazivač na deljenu memoriju). Nakon toga iz DTV Android aplikacije sledi poziv nativne funkcije GraphicsRendererNative.getPointerToAshmem() koja obezbeđuje pokazivač na deljenu memoriju na strani DTV Aplikacije koja će koristiti tu memoriju da pročita upisanu Bitmap-u prevoda i da je iscrta na Canvas-u prevod dijaloga. Na kraju će iz DTV Android aplikacije uslediti poziv funkcije DTV Java servisa (service.getSubtitleControl().show()) koja će putem JNI sloja obavestiti Comedia programsku podršku da počne sa iscrtavanjem prevoda (Slika 4.7).



Slika 4.7 Dijagram razmene poruka prilikom iscrtavanja DTV prevoda

DTV programska podrška vodi računa o sinhronizaciji prevoda na osnovu PTS koji se nalazi u zaglavlju PES paketa koji nosi prevod podatke. PTS kazuje u kom trenutku u vremenu objekat (Subtitle Bitmap) treba da se prikaže na ekranu.

Prikaz prevoda može ostati na ekranu nekoliko sekundi. Svaki prikaz se sastoji od više regiona sa specifičnom pozicijom. Region je pravougaona oblast ekrana sa horizontalnim i vertikalnim dimenzijama i dubinom piksela. Svaki region je povezan sa CLUT, koja definiše boju i transparentnost svakog piksela [6]. Za prevod postoji vise različitih CLUT (veličina 4, 16 ili 256 piksela). Pošto svaki region može imati svoju tabelu boja nakon stvaranja regiona DTV programska podrška poziva funkciju TDAL\_GFX\_RegionPaletteSet(...) kako bi setovala odgovarajuću tabelu boja odgovarajućem regionu. Isrtavanje prevod Bitmap-e vrši funkcija TDAL\_GFX\_BmpDraw(...). Najvažniji parametar ove funkcije je:

`tTDAL_GFX_Bitmap *pBitmap` – Pokazivač na strukturu koja sadrži Bitmap za prikaz  
U slučaju prevoda podaci `pBitmap->pData` predstavljaju indekse u odgovarajućoj CLUT\_AYCRBCB8888 tabeli za region koji se trenutno iscrtava. Alpha, R, G i B komponente svakog piksela se dobijaju čitanjem vrednosti CLUT tabele na odgovarajućoj poziciji i njihovom konverzijom iz AYCRBCB8888 u ARGB8888. Npr. R, G, B, Alpha komponente nekog piksela Bitmap-e se dobijaju čitanjem A, Y, Cr, Cb komponente iz odgovarajuće CLUT tabele i konverzijom:

$$\begin{aligned} R &= Y + 1.402*(Cr-128) \\ G &= Y - 0.34414*(Cb-128) - 0.71414*(Cr-128) \\ B &= Y + 1.772*(Cb-128) \\ A &= A \end{aligned}$$

Originalni format prevod Bitmap-e koji je indeks baziran se konvertuje traženjem odgovarajuće boje u tabeli boja na osnovu prosleđenog indeksa, konverzujom boja u R, G, B, Alpha format i smeštanjem sve četiri komponente (R, G, B i Alpha) kao četiri bajta u zauzeti bafer (`pDataRGBA`) koji je 4 puta veći od veličine `pData` (svaki piksel se sastoji od 4 bajta).

Po završenoj konverziji dobijeni bafer (`pDataRGBA`) koji sadrži piksele Bitmap-e se prosleđuje funkciji:

```
drawSubtitle(offset.x, offset.y, pBitmap->size.width, pBitmap->size.height, pDataRGBA,
subtitleRegion);
```

U ovoj funkciji se vrši kopiranje prevod Bitmap-e u bafer deljene memorije (alociran iz Ashmem modula), tako da mu Java aplikacija može pristupiti radi iscrtavanja.

```
memcpy(&(base[SUBTITLE_OFFSET_ASHMEM+subtitleRegion*width*4]),
pDataRGBA, height*width*4); //base – pokazivač na bafer deljene memorije
```

SUBTITLE\_OFFSET\_ASHMEM je pozicija prevod Bitmap-e u deljenom baferu. Isti bafer se koristi i za teleteksBitmap-u koja se smešta na početak bafera, dok se prevod smešta od pozicije SUBTITLE\_OFFSET\_ASHMEM u baferu.

Nakon kopiranja Bitmap piksela u bafer deljene memorije poziva se funkcija sendCallback() koja poziva funkciju callbackCall() DTV Java servisa.

Sve do sada opisano predstavlja izvorni C kod koji je ugrađen u Comedia programsku podršku (DTV Middleware). Ovaj kod se nalazi u okviru DTV Java servisa u formi deljene biblioteke (Android shared library).

Funkcija callbackCall(...) u DTV Java servisu inicira poziv funkcije DTV Java aplikacije za osvežavanje SubtitleDialogView-a. Osvežavanje se sastoji u pozivu nativne funkcije GraphicsRendererNative.draw(canvas). Ova funkcija crta prevod Bitmap-u iz bafera deljene memorije na Canvas (Slika 4.7). U nativnom sloju Canvas stiže u funkciju:

```
Java_android_dtv_graphics_GraphicsRendererNative_draw (... jobject canvas,...);
```

Ova funkcija je deo Android deljene biblioteke (Graphics Renderer Native) koja je uključena u DTV Android aplikaciju. U ovoj funkciji se od bafera deljene memorije (pikseli prevod Bitmap-e) formira objekat Bitmap koji se iscrtava na Canvas.

```
SkCanvas* canvasSubtitle;
SkBitmap bmSubtitle;
canvasSubtitle=GraphicsJNI::getNativeCanvas(env,canvas);
bmSubtitle.setConfig(SkBitmap::kARGB_8888_Config, 720, 600, 0);
bmSubtitle.setPixels(&(base[SUBTITLE_OFFSET_ASHMEM]), NULL);
canvasSubtitle->drawBitmap(bmSubtitle,0,0, paint);
```

Ovim je krug za iscrtavanje prevod Bitmap-e zatvoren. On se ponavlja kada PTS callback javi DTV programskoj podršci da treba prikazati sledeću Bitmap-u prevoda.

## 4.4 Realizacija iscrtavanja MHEG-a

MHEG se u android grafičkoj ravni iscrtava analogno teletekstu. Za MHEG, DTV programska podrška obezbeđuje Bitmap-e različitih dimenzija koje se kombinuju u jednu Bitmap-u veličine 720 x 576 piksela na osnovu položaja koji stiže kao argument u funkciju TDAL\_GFX\_BmpDraw(...). Vrednosti piksela Bitmap-e se upisuju u bafer deljene memorije na odgovarajuću poziciju, odakle će ih pročitati i iscrtati DTV Android aplikacija zahvaljujući svojoj nativnoj biblioteci Graphics Renderer Native. MHEG ima svoju paletu boja koja se sastoji od 256 različitih boja u ARGB8888 formatu.

## 5. Ispitivanje i verifikacija

Uspešnost iscrtavanja teleteksta, prevoda i MHEG-a korišćenjem DTV Android aplikacije i DTV Java servisa u koji je ugrađena Comedia programska podrška ispitana je na MARVEL BG2 SOC platformi na kojoj je instaliran Android operativni sistem. Rešenje je ispitano na dve verzije Android OS (4.0.x Ice Cream Sandwich i 2.3.x Gingerbread).

Narednim ispitivanjima potvrđena je uspešnost realizacije iscrtavanja interaktivnih servisa digitalne televizije u Android OS koja je opisana u ovom radu. Ispitivanjima je potvrđena ispravnost sledećih programskih modula:

- DTV Android aplikacija (Java kod)
- DTV Java servis (Java kod)
- Comedia programska podrška (C kod)
- Modul za zauzimanje deljene memorije - Ashmem Service Server (C++ kod)
- Modul za preuzimanje pokazivača na deljenu memoriju – Service Client Library (C++ kod)
- Modul za iscrtavanje Bitmap-e na Canvas - Native Graphics Renderer (JNI C++ kod)
- Modul za spregu DTV Java servisa i Comedia programske podrške (JNI C++ kod)

### 5.1 Ispitivanje teletekst realizacije

Pritiskom na dugme TTXT u slučaju postojanja teleteksta na tom servisu doći će do iscrtavanja teleteksta (Slika 5.1). Pritiskom na neko od alfanumeričkih dugmića doći će do promene stranice teleteksta. Pritiskom na plavo dugme prikazaće se Subtitle koji se nalazi na teletekst stranici 888.



Slika 5.1 Teletekst

Ponovnim pritiskom na dugme TTXT teletekst prelazi u transparentni mod (Slika 5.2). Još jedan pritisak na dugme TTXT dovodi do sklanjanja teleteksta.



Slika 5.2 Transparentan teletekst

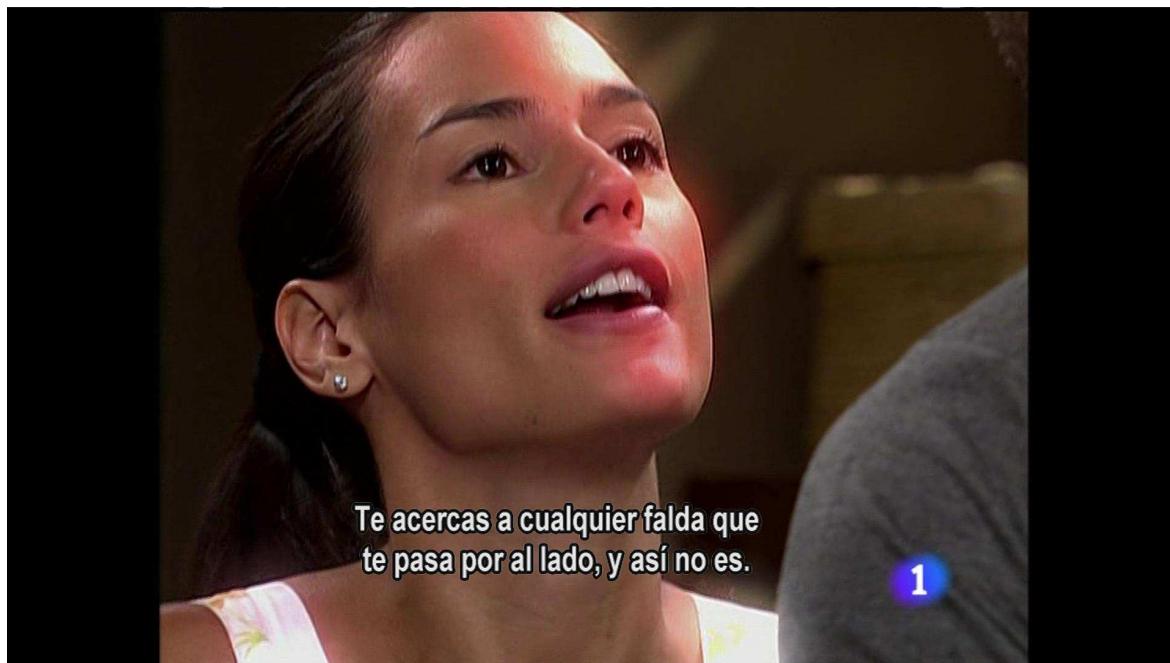
Izborom opcije “resize” može se izvršiti skaliranje teleteksta (Slika 5.3).



Slika 5.3 Skaliranje teleteksta

## 5.2 Ispitivanje realizacije prevoda

Pritiskom na dugme Subtitle pojavljuje se prevod (Slika 5.4). Ponovnim pritiskom na dugme Subtitle prevod se sklanja sa ekrana.

Slika 5.4 Prevod (eng. *Subtitle*)

### 5.3 Ispitivanje MHEG realizacije

Pritiskom na TEXT dugme MHEG aplikacija se prikazuje na ekranu (Slika 5.5). Sledećim pritiskom na isto dugme MHEG nestaje sa ekrana. Navigacija kroz MHEG aplikaciju se odvija korišćenjem Up, Down, Left, Right i alfa-numeričkih dugmića. Upotreba dugmića u boji na daljinskom upravljaču (Red, Green, Yellow, Blue) olakšava korišćenje MHEG aplikacije.



Slika 5.5 MHEG aplikacija

## 6. Zaključak

Ovaj rad predstavlja jedno rešenje za integraciju servisa digitalne televizije u Java programski jezik. Ciljni operativni sistem je Android operativni sistem. Rad je fokusiran na prikaz sadržaja interaktivnih servisa digitalne televizije (teletekst, prevod, MHEG) u android grafičkoj ravni. Pošto DTV servisi zahtevaju značajne resurse uobičajena procedura u Android operativnom sistemu je implementacija najkritičnijih detalja kao izvornih biblioteka. Iz tog razloga iscrtavanje interaktivnih servisa izvršeno je u izvornom C/C++ kodu korišćenjem grafičke biblioteke niskog nivoa implementirane u nativnom kodu koja rukuje iscrtavanjima u Android grafičkoj ravni (SKIA grafička biblioteka). Na taj način postignute su zadovoljavajuće performanse. Za prikupljanje i obradu DTV podataka korišćena je Comedia programska podrška za televizijske prijemnike (eng. *Middleware*) napisana u C programskom jeziku.

Da bi se realizovalo iscrtavanje interaktivnih servisa digitalne televizije u android grafičkoj ravni realizovani su sledeći programski moduli:

- DTV Android aplikacija – obezbeđuje GUI koji prikazuje sadržaj navedenih DTV servisa i omogućava interakciju korisnika sa tim sadržajem (Java kod)
- DTV Java servis – sprega između Android aplikacije i Comedia programske podrške (Java kod)
- Comedia programska podrška – rukuje prikupljanjem, obradom DTV podataka i upisivanjem odgovarajućih Bitmap-a u android deljenu memoriju (C kod)
  - Ashmem Service Server – zauzima deljenu memoriju za Bitmap-e (C++ kod)
  - Service Client Library - preuzima pokazivač na deljenu memoriju (C++ kod)
  - Native Graphics Renderer - modul za iscrtavanje Bitmap-e na Canvas (JNI C++ kod)
  - JNI sloj - sprega DTV Java servisa i Comedia programske podrške (JNI C++ kod)

Rešenje je ispitano na Marvell BG2 SOC platformi na kojoj je instaliran Android OS.

## 7. Literatura

- [1] M. Vidaković, N. Teslić, T. Maruna, and V. Mihić, “Android4TV: a proposition for integration of DTV in Android devices”, IEEE 30th International Conference on Consumer Electronics (ICCE), Las Vegas, January 2012, pp. 441-442.
- [2] iWedia sajt, [www.iwedia.com](http://www.iwedia.com), učitano 14.06.2012.
- [3] JNI Oracle sajt, [www.oracle.com/javase/1.4.2/docs/guide/jni/spec/functions.html](http://www.oracle.com/javase/1.4.2/docs/guide/jni/spec/functions.html), učitano 14.06.2012.
- [4] W. Fischer, *Digital Video and Audio Broadcasting Technology*, A Practical Engineering Guide, Third Edition
- [5] ETS 300 472: *Digital Video Broadcasting (DVB); Specification for conveying ITU-R System B Teletext in DVB bitstreams*
- [6] ETSI EN 300 743: *Digital Video Broadcasting (DVB); Subtitling systems*; v.1.3.1 (2006-11).
- [7] ETSI ES 202 184: *MHEG-5 Broadcast Profile*, v2.1.1 (2010-01)
- [8] Sajt [www.developer.android.com](http://www.developer.android.com), učitano 14.06.2012.
- [9] Frank Ableson, Charlie Collins, Robi Sen, *Unlocking Android*, A Developer’s Guide
- [10] Reto Meier, *Professional Android 2 Application Development*
- [11] Dokumentacija za Android grafičku biblioteku (SKIA - 2D Graphics Library),  
<http://skia-autogen.googlecode.com/svn/docs/html/index.html>