



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Леон Серенчеш

ТВ апликација

ИСПИТНИ РАД

- Програмска подршка у телевизији и обради слике -

Ментор: Милан Бјелица

Нови Сад, 2013

SADRŽAJ

1. Zadatak projekta	3
2. Teorijski uvod	6
2.1 Digitalna Televizija	6
2.2 MPEG i DVB standardi.....	6
3. Koncept rešenja.....	11
3.1 Blok dijagram programske podrške	11
3.2 Opis modula	12
4. Implementacija rešenja	14
4.1 Glavna logika aplikacije (zapper.c).....	14
4.2 Parsiranje konfiguracione datoteke (config_parser.c).....	16
4.3 Parsiranje SI tabela (tables_parser.c)	17
4.4 Obrada ulaza sa daljinskog upravljača (input.c)	18
4.5 Crtanje grafičkih elemenata (gui.c)	18
5. Zaključak	20

1. Zadatak projekta

Korišćenjem razvojnog okruženja (STB, HD monitor), razvojnih alata (gcc, PuTTY, tekst editor) i dostupnih DTV biblioteka, potrebno je razviti aplikaciju za TV prijemnik, ispitati ispravnost njenog rada i dokumentovati urađeno. Dat je sledeći spisak zahteva:

a) Nakon pokretanja, aplikacija učitava konfiguracionu datoteku čija je lokacija određena jedinim parametrom komandne linije: `tv_app /home/my_config/config.ini`

Izgled konfiguracione datoteke je sledeći:

```
initial_config {  
    transponder {  
        frequency=470;  
        module=DVB_T;  
        bandwidth=8 MHz;  
    }  
    initial_channel {  
        audio_PID=101;  
        audio_type=AC3;  
        video_PID=102;  
        video_type=MPEG2;  
    }  
}
```

b) Na osnovu parametara konfiguracione datoteke, pri pokretanju aplikacija podešava ulazni stepen STB prijemnika i otpočinje prijem signala.

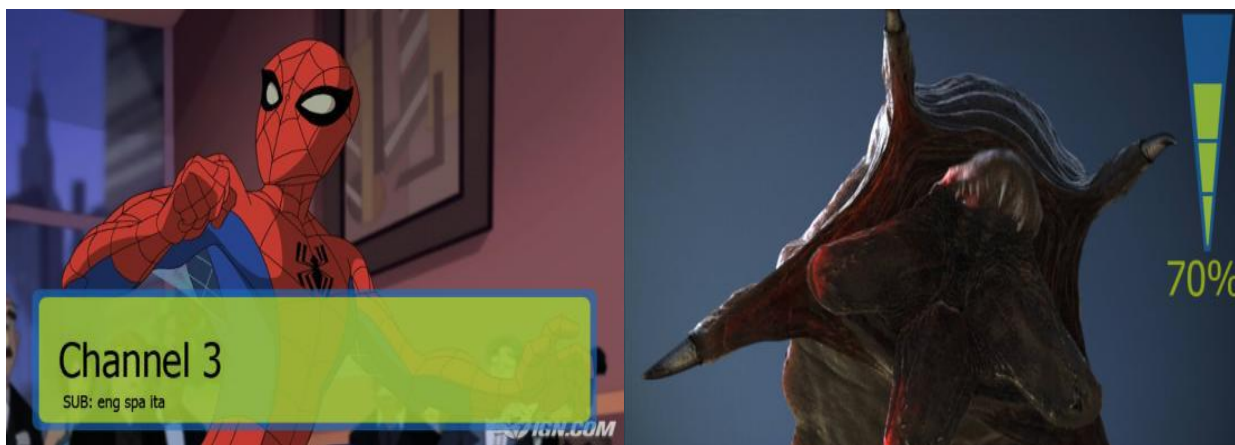
c) Pri pokretanju aplikacija pokreće audio i video dekodovanje korišćenjem PID-ova iz konfiguracione datoteke. Slika je vidljiva na ekranu.

d) Na osnovu parsiranja PAT i PMT sekcija, aplikacija dobavlja i memoriše spisak servisa sa svim parametrima potrebnim za dekodovanje.

e) Omogućiti korišćenje daljinskog upravljača. Aplikacija omogućava izmenu kanala korišćenjem strelica gore/dole, kao i izmenu jačine zvuka korišćenjem za to predviđenih tastera. Nakon pritiska na taster strelica, potrebno je dobiti prethodno memorisane parametre iz spiska servisa i pokrenuti audio/video dekodovanje i reprodukciju na ekran.

f) Omogućiti ručni unos broja kanala i promenu reprodukcije na uneti kanal.

g) Pri svakoj izmeni kanala, aplikacija prikazuje grafički info dijalog sa informacijom o broju kanala iz liste, kao i sa informacijom o tome da li na aktivnom kanalu ima subtitle. U koliko subtitle postoji, treba izlistati jezik svake trake (Slika 1a). Info dijalog je potrebno automatski ukloniti nakon 4 sekunde.



Slika 1.1 – (a) Izgled info dijaloga, (b) Izgled dijaloga za prikaz jačine zvuka

h) Pri svakoj promeni jačine zvuka, aplikacija prikazuje grafički dijalog sa skalom trenutne jačine zvuka (Slika 1b).

i) U info dijalog dodati informaciju o trenutno gledanoj emisiji (naziv) - slika 2. Informacija o ovome se dobija parsiranjem EIT tabele. Info dijalog uvek treba da prikazuje tačne i aktuelne informacije o trenutno gledanoj emisiji.



Slika 1.2 – Izgled info dijaloga

j) Dodati novi dijalog koji se prikazuje kada korisnik pritisne taster OK dok je info dijalog aktivan. Novi dijalog treba da prikazuje više informacija o trenutno gledanoj emisiji (vreme početka i završetka, kao žanr i opis same emisije) – slika 3. Dijalog se zatvara pritiskom na taster BACK.



Slika 1.3 – Izgled dijaloga sa dodatnim informacijama

2. Teorijski uvod

2.1 Digitalna Televizija

Jedan od najvažnijih pravaca razvoja televizijske tehnologije je pojava digitalne televizije, što predstavlja prenos televizijskog signala u digitalnom, tj. binarnom formatu.

Binaran format omogućuje komprimovanje televizijskih podataka što dozvoljava da se istim propusnim opsegom kojom se emitovao jedan analogni kanal sada emituje približno deset digitalnih kanala. Druge prednosti digitalne televizije su:

- Mnogo bolji kvalitet prenosa slike i zvuka, jer je digitalni signal mnogo otporniji od interferencije i kao rezultat slika i zvuk koje digitalni signal nosi su isti kao i na izvoru emitovanja, sve dok signal ne postane toliko slab da prijem više nije moguć.
- Mogućnost izvršavanja aplikacija kao što je interaktivna televizija koja omogućava izvršavanje računarskih aplikacija pisanih u Java, html ili ActionScript jeziku.
- Mogućnost izbora formata slike (4:3 ili 16:9) kao i zvuka (mono, stereo ili "surround").
- Uvođenje novih usluga kao što su izbor jezika za titlovanje i audio kanala, interaktivni i multimedijalni sadržaji i pristup Internetu.

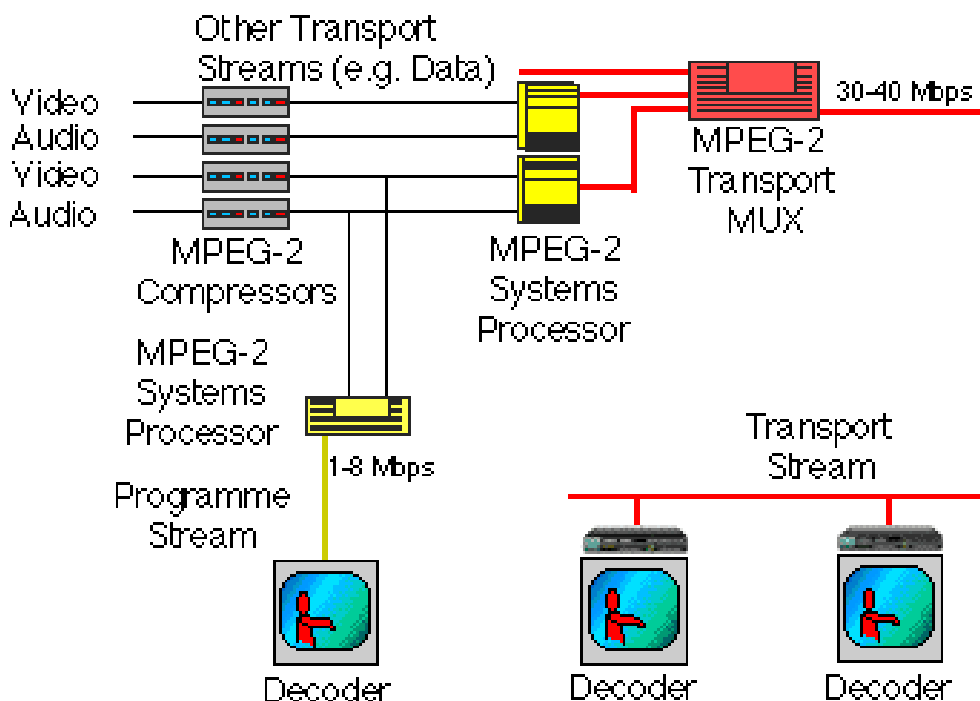
2.2 MPEG i DVB standardi

U kompresiji video toka trenutno je dominantan MPEG2 standard, a pojavom HDTV se polako uvodi H.264 AVC. Tokovi MPEG2 audio i video podataka koji se emituju nazivaju se *elementary streams* (ES). Koduju se i prenose kao zasebni tokovi podataka. Kodovani podaci se prenose korišćenjem paketski orijentisanih protokola u kombinaciji sa odgovarajućim mehanizmima za detekciju i obradu grešaka. Da bi se na prijemnoj strani ES mogli povezati u

logičke celine, kao i da bi se obezbedila odgovarajuća sinhronizacija audio i video podataka, pored ES moraju se prenositi i dodatni podaci. Ovako kompleksna struktura prenosa podataka zahteva postojanje posebnog transportnog sloja/protokola koji se takođe definiše MPEG2 standardom, ali i odgovarajućim Digital TV standardom.

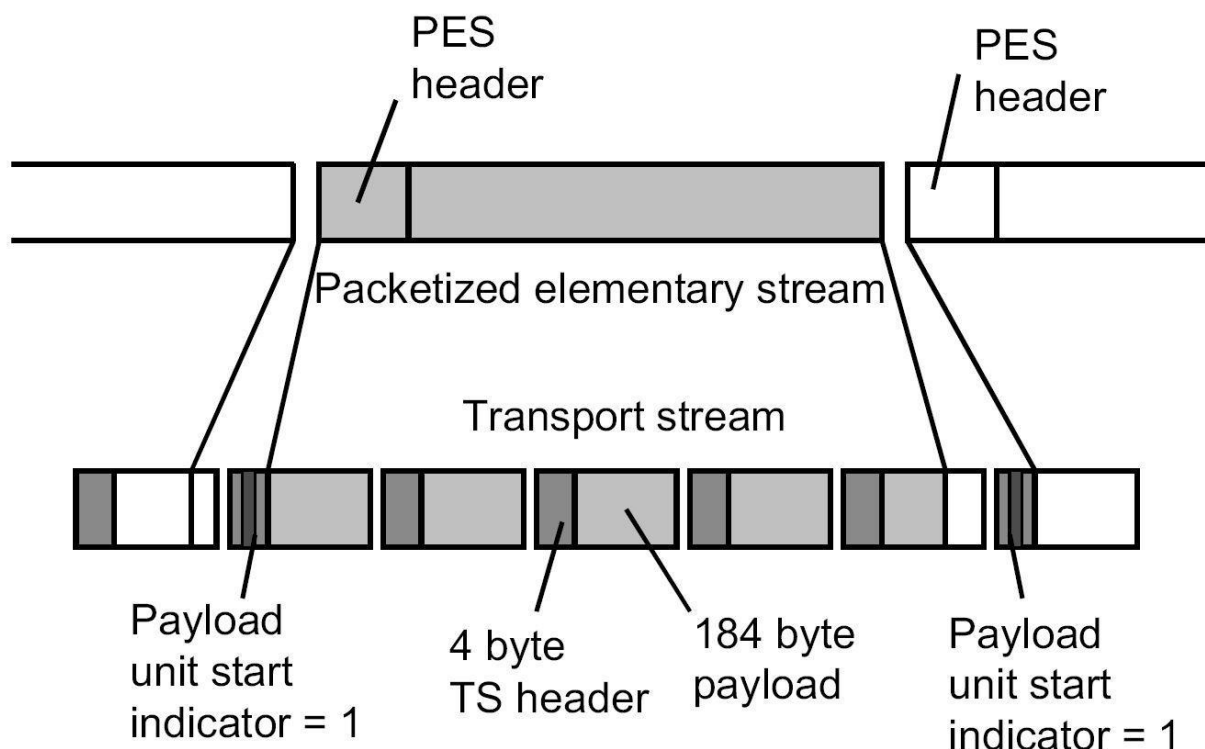
Jedan od najrasprostranjenijih standarda digitalne televizije je DVB (Digital Video Broadcast) standard koji se koristi u Evropi, Africi, Australiji i delu Azije.

U zavisnosti od prenosnog medijuma MPEG2 standard definiše različite formate za kombinovanje MPEG2 elementarnih tokova u jedinstven informacioni tok podataka. Ukoliko prenosni medijum nije podložan pojavi grešaka u toku prenosa, MPEG2 elementarni tokovi se kombinuju u tzv. *Program Stream* (PS). Ovakav format omogućava jednostavnije programsko rukovanje podacima i koristi se za audio/video reprodukciju (npr. reprodukcija sa CD-a ili DVD-a) kao i u nekim mrežnim aplikacijama. Ukoliko je, međutim, prenosni put podložan pojavi grešaka (npr. broadcasting), elementarni tokovi se kombinuju i prenose kao *Transport Stream* (TS). Sem toga TS format je pogodan za kombinovanje više TV programa u jedinstven informacioni tok. Ovaj proces kombinovanja MPEG2 elementarnih tokova u jedinstveni informacioni tok u skladu sa definisanim formatom se naziva se *Multipleksiranje* MPEG2 strima. U digitalnoj televiziji se za prenos podataka koristi MPEG2 Transport Stream format koji se dodatno proširuje u okviru različitih DTV standarda (npr. DVB Transport Stream).



Slika 2.1 – Multipleksiranje elementarnog toka

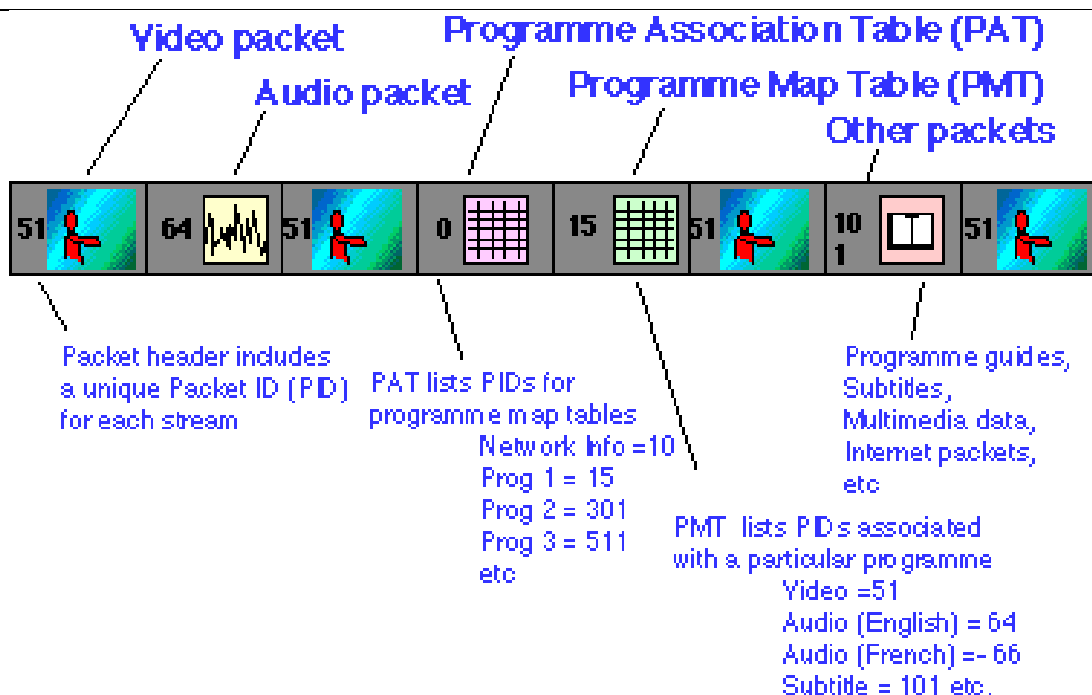
MPEG elementarni tokovi se paketizuju u standardan format dobijajući *paketizovan elementaran tok* (Packetized Elementary Stream - PES). Da bi se PES prenosila kao TS mora dalje da se paketizuje u TS Pakete (Slika 2.2)



Slika 2.2 Paketizacija TS-a

Svaki TS paket sadrži zaglavlje od 4 bajta (okteta). Jedno polje od 13 bita unutar zaglavlja je nazvan *Packet Identifier* (PID) koji igra ključnu ulogu u rukovanju TS. Svaki TS paket sadrži podatke koji pripadaju jednom PES što se razlikuje odgovarajućim PID vrednostima.

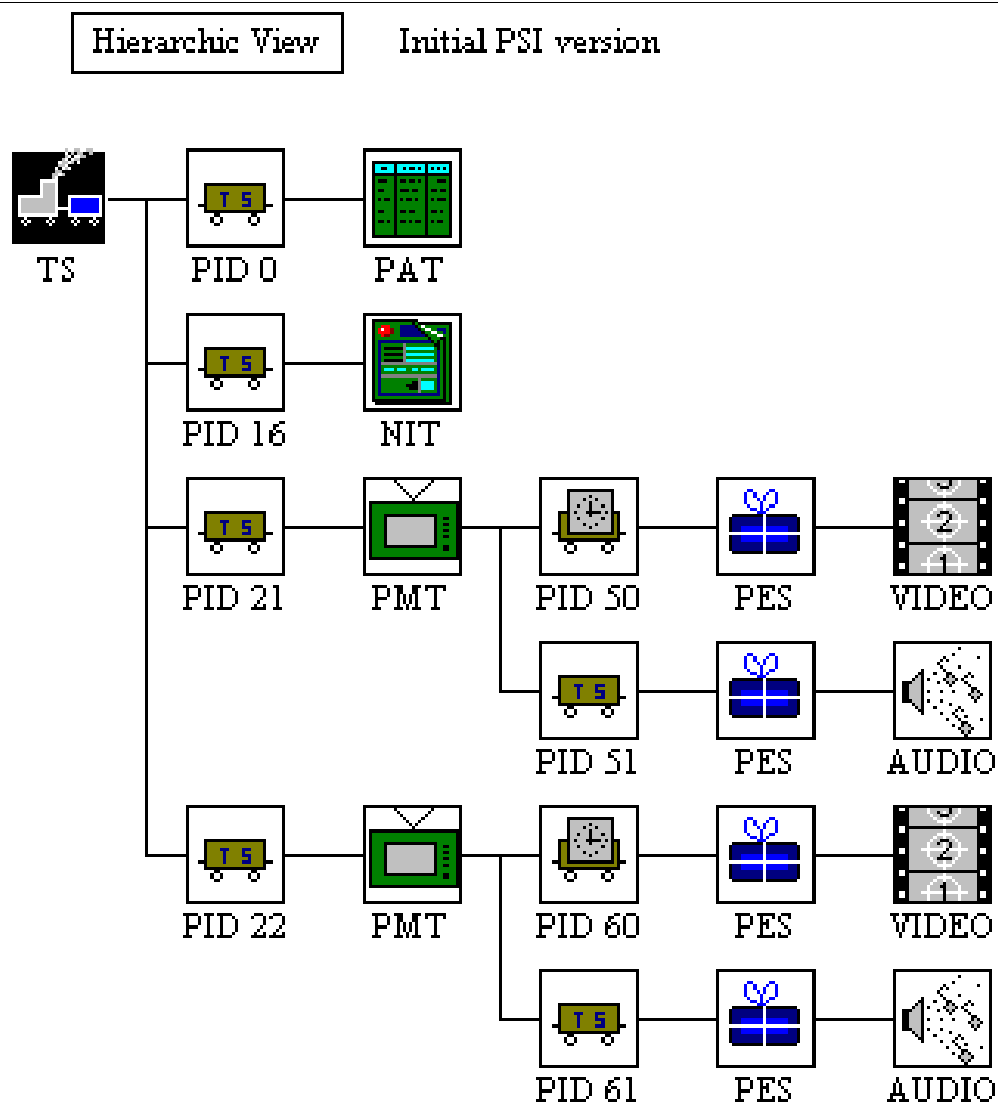
Da bi se prijemnoj strani omogućilo da poveže PID vrednosti sa odgovarajućim DTV servisima sadržanim u TS, zajedno sa PES paketima u istom TS se prenose i specijalni kontrolni strimovi koji sadrže tzv. *signalne tabele* (*Signalling Tables*). Ove tabele nose podatke o svakom od DTV servisa koji se prenosi unutar TS. Signalne tabele se prenose kao zasebni tokovi podataka unutar TS, multipleksirani zajedno sa ostalim TS paketima (Slika 2.3). Tabele (nazvane *Program Specific Information* (PSI) u MPEG-2 standardu) se sastoje od opisa ES koje treba kombinovati da bi se dobio određeni DTV servis, kao i od opisa samih DTV servisa. Svaka PSI tabela je sadržana u jednoj ili više *PSI sekcija* (*PSI Sections*).



Slika 2.3 – Primer TS multipleksa

Najvažnije signalne tabele su PAT (*Program Association Table*) i PMT (*Program Map Table*). PAT sadrži listu PID vrednosti TS paketa sa PSI tabelama (PMT tabele) koje u sebi nose podatke o DTV servisima koji se emituju na tom transponderu (frekvenciji). PAT se uvek šalje u TS paketima sa predefinisanim PID-om koji ima vrednost 0x0000. PMT definiše listu PID vrednosti TS paketa koji sadrže PES (npr. video, audio, teletext) pridružene određenom programu.

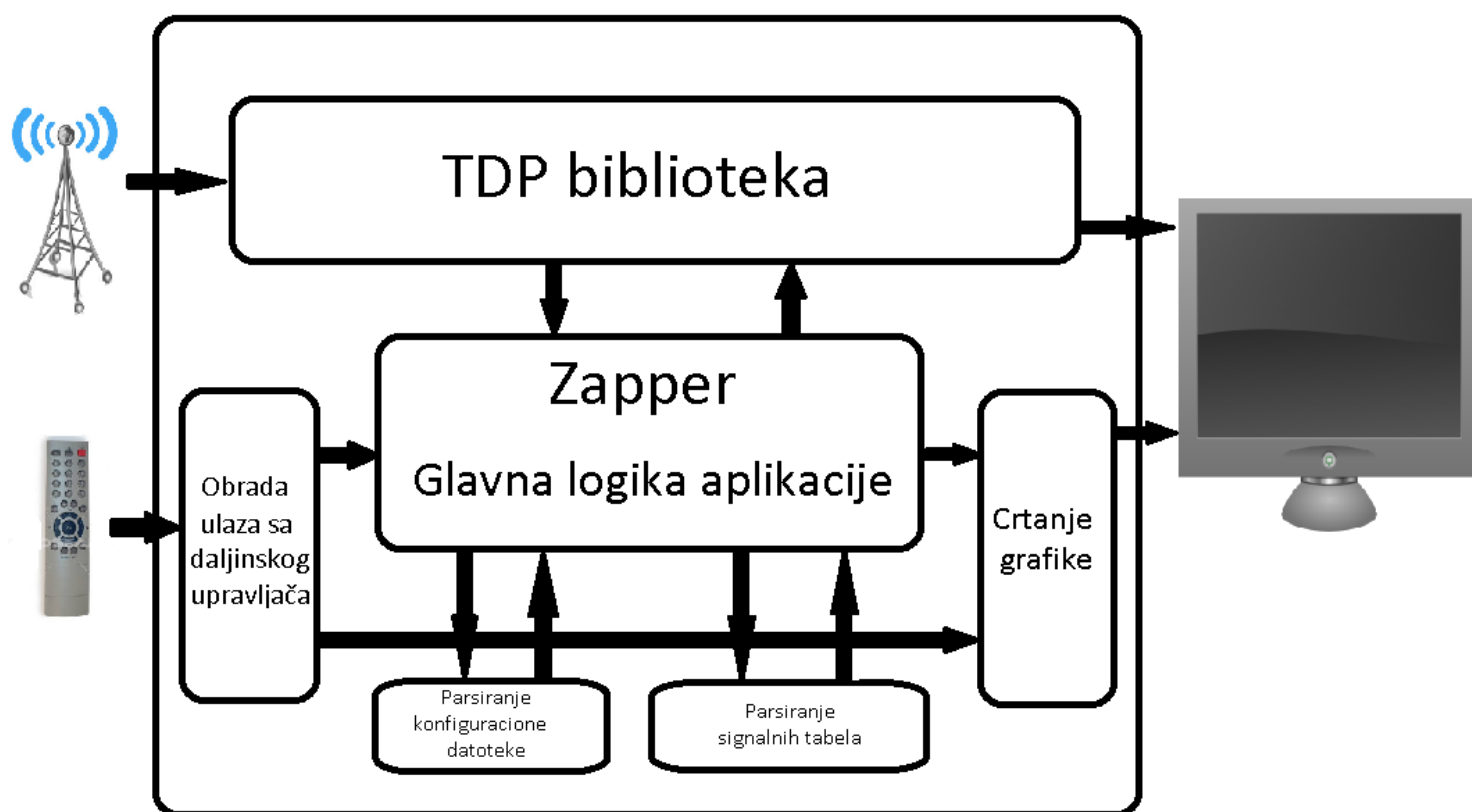
DVB standard proširuje skup signalnih tabela uvodeći tzv. *tabele sa servisnim informacijama* (*Service Information Tables – SI tables*). Ove tabele nose informacije o DTV servisima sadržanim u TS i posmatranim kao logičke celine, a ne kao skup PES. Jedan od SI tabela od interesa za ovaj projekat je Event Information Table (EIT) koja sadrži detalje o rasporedu emitovanja programa, i deskriptora sa dodatnim informacijama o emisijama kao što su imena, tekstualni opisi, i žanrovi emisija. EIT tabele imaju predefinisani PID koja ima vrednost 0x0012.



Slika 2.4 - Ilustracija hijerarhije PSI/SI tabela

3. Koncept rešenja

3.1 Blok dijagram programske podrške



Slika 3.1 – Blok dijagram aplikacije

3.2 Opis modula

TDP biblioteka predstavlja skup funkcija za rukovanje sa tunerom, demultiplekserom i player-om (Uređaj koji pušta audio i video sadržaj) TV uređaja i omogućava programu da otpočinje prijem DVB signala na određenoj frekvenciji, da namesti demultiplekser radi primanja TS sekcije sa određenim PID-ovima, i da pokreće reprodukciju audio i video tokova na izlazni televizijski uređaj.

Zapper sadrži glavnu logiku aplikacije. Izvršavanje aplikacije započinje sa parsiranjem konfiguracione datoteke koja sadrži parametre za tuner i inicijalni kanal, inicijalizacijom TDP biblioteke, puštanje inicijalnog kanala, registracijom callback funkcija koje reaguju na zaključavanje tunera i primanje sekcije iz demultipleksera i podešavanje filtera demultipleksera za prijem PAT tabele. Program zatim pokreće nit koja obrađuje korisničke komande dobijene preko daljinskog upravljača. Kada program primi korisničku komandu za završetak programa program izvrši potrebne deinicijalizacije TDB biblioteke pre završetka.

Zapper takođe sadrži funkcije kojima se pokreće process menjanja kanala i menjanje jačine zvuka reprodukcije. Ove funkcije su pozvane od strane niti koja obrađuje korisničke komande.

Modul za parsiranje konfiguracione datoteke sadrži funkciju koja učitava ulaznu konfiguracionu datoteku, proverava njenu ispravnost, i pravilno parsira potrebne parametre za početak rada aplikacije.

Modul za parsiranje SI tabele sadrži funkcije neophodne za parsiranje PAT, PMT i EIT tabele. Ove funkcije su pozvane kada demultiplekser izdvoji TS sekciju čiji PID odgovara onoj zabeleženoj u filtru demultipleksera, zatim poziva callback funkciju za obradu sekcije koja na osnovu polja table_id zaključuje koju tabelu ta sekcija predstavlja i poziva odgovarajuću funkciju iz modul za parsiranje SI tabele koja će da formira strukturu podataka koja sadrži važne informacije iz tabele.

Modul za obradu ulaza sa daljinskog upravljača sadrži funkciju koja će biti pokrenuta kao zasebna nit programa čiji zadatak je da reaguje na korisničke komande tako što će da poziva funkcije za menjanje kanala i jačinu zvuka iz zapper modula, i funkciju za iscertavanje dijaloga za prikaz broja kanala iz modula za crtaje grafike u slučaju prijema komande za prikaz tog dijaloga.

Modul za crtanje grafike koristi DirectFB biblioteku za crtanje grafičkih elemenata za prikazivane korisniku informacije vezane za tekuću TV emisiju. Modul sadrži funkcije koje spremaju informaciju koja mora biti prikazana korisniku u zavisnosti od trenutnog stanja programa, pa zatim poziva funkciju koja izvršava samo crtanje elemenata i prikazuje rezultat na izlaznom monitoru.

4. Implementacija rešenja

4.1 Glavna logika aplikacije (zapper.c)

int main(int32_t argc, char argv)**

main funkcija počinje sa radom parsiranjem konfiguracione datoteke čija putanje se dobija kao argument komandne linije pri pokretanju programa. Nakon uspešnog parsiranja konfiguracione datoteke program inicijalizuje TDP i DirectFB biblioteke, registruje callback funkcije, pokreće inicijalni kanal čiji PID-ovi za audio i video stream-ova su dobijeni iz konfiguracione datoteke, i pokreće nit za obradu komande dobijene preko daljinskog upravljača, zatim čeka završetak te niti, što se dogodi kada korisnik pritisne taster za završetak programa. Nakon toga program deinicijalizuje TDP i DirectFB biblioteke i završava se program.

int32_t tunerStatusCallback(t_LockStatus status)

tunerStatusCallback je callback funkcija koja TDP API poziva nakon uspeha ili neuspeha zaključavanja tunera na određenu frekvenciju. Preko uslovne promenljive signalizira glavnu nit da je zaključavanje tunera uspešno, ili u slučaju neuspeha ne radi ništa, što posle određenog vremena izaziva glavnu nit da prekine rad aplikacije.

int32_t sectionReceivedCallback(uint8_t* buffer)

Ova funkcija je pozvana od strane TDP API kada se primaju TS sekcije čiji PID odgovara onoj u filtru demultipleksa. Funkcija pročita prvi bajt (table_id) iz memorijske lokacije na koju pokazuje pokazivač buffer, i na osnovu njene vrednosti zaključuje da li sekcija predstavlja PAT (table_id = 0x00), PMT (table_id = 0x02) ili EIT tabelu za tekući događaj (table_id = 0x4E), i poziva odgovarajuću funkciju za parsiranje tabele.

Ako je primljena PAT tabela, program je parsira i sačuva njene informacije i na osnovu broja servisa prisutnih u PAT tabeli program alokira memoriju na koju će pokazivati pokazivač `eitTables`, za toliko EIT tabela koliko postoji servisa u PAT tabeli. Za rezultujući niz EIT tabela, *i*-ta tabela odgovara *i*-tom servisu unutar PAT tabele. Takođe se alokira memorija za niz logičkih vrednosti `eitTableReceived` koja govori o prisutnost EIT informacija, tj. *i*-ti element niza govori da li je primljena EIT tabela za *i*-ti servis iz spiska servisa u PAT tabeli, zatim se podesi filter demultipleksera za primanje EIT sekcija.

Ako je primljena EIT tabela, funkcija pročitava bajt koji predstavlja `service_id`, tj. identifikacioni broj koji povezuje EIT tabelu sa određenim servisom, i proverava da li u PAT tabeli postoji servis sa istim `service_id`. Ako postoji, EIT tabela se parsira i sačuva se kao element niza `eitTables`, postavlja se odgovarajući element niza `eitTableReceived`.

Ako je primljena PMT, ona se parsira i ako je logička promenljiva `changingChannels` istinita (korisnik je tražio promenu kanala), započinje promena reprodukcije audio i video sadržaja na novi kanal pozivom funkcije `changeStreams()`. Zatim podešava filter demultipleksera za primanje EIT sekcija.

`void changeStreams(void)`

Kada je ova funkcija pozvana, otpočinje zamena tekuće audio i video reprodukcije na one čije TS PID-ovi se nalaze u promenljivoj `pmtTable`. Program iterira kroz listu stream-ova u PMT tabeli tražeći onaj čija vrednost `streamType` je jednaka 2 (za video tok) ili 3 (za audio tok), i pušta stream-ove koje uspešno pronađe.

`void changeChannelUp(void),`

`void changeChannelDown(void),`

`void changeChannelToNumber(uint8_t channelInputNumber)`

Ove funkcije počinju postupak menjanja TV kanala, tako što promene vrednost promenljive `currentChannel`, pri tome pazeći da ostane u opsegu od 0 do `patTable.serviceInfoCount - 1` (broj servisa prisutnih u PAT tabeli), postavlja logičku promenljivu `changingChannels` na istinitu vrednost, i podešava filter demultipleksera za primanje PMT tabele čiji pid se nalazi u nizu servisa PAT tabele sa indeksom koji je jednak `currentChannel`. Nakon određenog vremena `sectionReceivedCallback` će biti pozvan da obradi PMT sekciju i da dovrši postupak menjanje kanala

void increaseVolume(void),

void decreaseVolume(void)

Ove funkcije menjaju jačinu izlaznog zvuka promenom promenljive `currentVolume` koja je celobrojna vrednost od 0 do 100, i na osnovu te skale funkcija `convertFromVolumeScale` će da izračuna vrednost koja se mora proslediti TDP funkciji `Player_Volume_Set` da bi se pravilno podesila jačinu zvuka

uint32_t convertFromVolumeScale(uint8_t input)

ulazni parametar je broj od 0 do 100 na osnovu kojeg se izračunava vrednost koja se mora proslediti TDP funkciji `Player_Volume_Set`.

Formula za jačina zvuka u decibelima u zavisnosti od vrednosti prosleđenoj `Player_Volume_Set` funkciji je:

$$Db = 20 * \log_{10}(TDPparam/0x80000000)$$

Gde jačina od 0 Db predstavlja maksimalnu jačinu zvuka

Formula za računanje *TDPparam* je:

$$TDPparam = 10^{\frac{(input*0.6-60)}{20}} * 0x80000000$$

Vrednost $input * 0.6 - 60$ predstavlja jačinu zvuka u decibelima, koja se kreće od -60 Db (kada je programska skala 0) do 0 Db (kada je programska skala 100).

void createFullDialogue(void),

void createSimpleDialogue(void)

Ove funkcije pozivaju `drawFullDialogue` i `drawSimpleDialogue` respektivno, radi iscrtaavanja info dijaloga nakon promene kanala, ili po zahtevu korisnika. Ako ne postoji EIT tabela za trenutno pušten kanal, funkcija `drawFullDialogue` se poziva parametrima (NULL, NULL, 0, 0, 0)

4.2 Parsiranje konfiguracione datoteke (config_parser.c)

`config_parser.c` sadrži jednu funkciju **parseConfigFile** koja učitava konfiguracionu datoteku i izdvaja parametre potrebne za početak rada programa. Funkcija čita datoteku liniju po liniju i formira niz tekstualnih linija datoteke, zatim za svaki parametar koji se mora pročitati iterira kroz sve linije datoteke i pokušava da izdvoji parametre upotrebom `sscanf` funkcije. Ako parametar

koji traži nije pronađen posle prolaska kroz tekst datoteke, funkcija prijavi grešku i program prestaje sa radom.

4.3 Parsiranje SI tabela (**tables_parser.c**)

U ovom modulu su definisane strukture **PatTable**, **PmtTable** i **EitTable** (i odgovarajuće podstrukture) koje definišu signalne tabele koje sadrže informaciju potrebnu za reprodukciju audio i video sadržaja i dodatne informacije o emisijama.

ParseErrorCode parsePatTable(const uint8_t* patSectionBuffer, PatTable* patTable)

Ova funkcija, uz pomoć funkcija **parsePatHeader** i **parsePatServiceInfo** realizuje parsiranje PAT tabele. Parametar **patSectionBuffer** pokazuje na memorijsku lokaciju gde se nalazi pristigla sekcija, a parametar **patTable** je pokazivač na **PatTable** strukturu gde će biti smeštena PAT informacija nakon parsiranja. Funkcija **printPatTable** može da štampa sadržaj PAT tabele na standardnom izlazu radi provere sadržaja.

ParseErrorCode parsePmtTable(const uint8_t* pmtSectionBuffer, PmtTable* pmtTable)

Ova funkcija, uz pomoć funkcija **parsePmtHeader** i **parsePmtStreamInfo** realizuje parsiranje PMT tabele. Parametar **pmtSectionBuffer** pokazuje na memorijsku lokaciju gde se nalazi pristigla sekcija, a parametar **pmtTable** je pokazivač na **PmtTable** strukturu gde će biti smeštena PAT informacija nakon parsiranja. Pored informacija o stream-ovima vezanim za određen servis, PMT može da sadrži subtitle deskriptore koji govore o prisutnost titlova i na kom su jeziku. Funkcija **printPmtTable** može da štampa sadržaj PMT tabele na standardnom izlazu radi provere sadržaja.

ParseErrorCode parseEitTable(const uint8_t* eitSectionBuffer, EitTable* eitTable)

Ova funkcija realizuje parsiranje EIT tabele. Parametar **eitSectionBuffer** pokazuje na memorijsku lokaciju gde se nalazi pristigla sekcija, a parametar **eitTable** je pokazivač na **PmtTable** strukturu gde će biti smeštena EIT informacija nakon parsiranja. EIT tabela sadrži informaciju o rasporedu emitovanja emisija, *short event* deskriptore koji sadrže imena i tekstualni opisi emisija, i *content* deskriptore koji govore o žanrovima emisija.

4.4 Obrada ulaza sa daljinskog upravljača (input.c)

void* input_thread_function(void* arg)

Ova funkcija će da bude pokrenuta kao nit koja će da reaguje na korisničke komande preko daljinskog upravljača. Podržane komande od strane programa su:

- promena kanala naviše i naniže
- direktan unos dvocifrenog broja kanala
- pojačavanje i slabljenje jačine zvuka
- prikazivanje info dijaloga sa broj kanala, informacija o prisutnim titlovima i ime emisije, kada se pritisne taster 'i'
- prikazivanje proširenog dijaloga sa ime i opis emisije, vreme i dužina emitovanja, i žanr emisije, kada se pritisne taster 'OK'
- uklanjanje proširenog dijaloga, kada se pritisne taster 'back'
- izlazak iz aplikacije, kada se pritisne taster 'exit'

4.5 Crtanje grafičkih elemenata (gui.c)

int32_t initDirectfb(void)

Funkcija vrši inicijalizaciju DirectFB biblioteke, učitava datoteke koje sadrži pozadine oba dijaloga, i priprema timer događaje za signaliziranje automatskog uklanjanja manjeg dijaloga i indikatora promena jačina zvuka posle određenog vremena. Ako je nesušešno kreiranje timer događaja funkcija vraća nenulti rezultat, prijavi grešku i program prekida sa izvršavanjem.

void deinitDirectfb(void)

Funkcija vrši deinicijalizaciju DirectFB biblioteke i briše timer-e za uklanjanje grafičkih elemenata

void drawSimpleDialogue(int channelNumber, char* subtitleInfo, char* eventName)

Funkcija priprema informaciju (u vidu tekstualnih stringova) potrebnu za prikazivanje info dijaloga koji sadrži broj tekućeg kanala, ime tekuće emisije, i informacije o titlovima, i postavlja logičku promenljivu simpleDialogueFlag na 1, da bi renderGui funkcija znala da iscrta dijalog, zatim poziva renderGui funkciju i pokreće timer koji će nakon 6 sekundi pozvati

removeSimpleDialogue funkciju koja uklanja dijalog, tako što postavlja simpleDialogueFlag na 0 i poziva renderGui

void drawFullDialogue(char* eventDescription, char* eventName, uint8_t genre, uint32_t startTime, uint32_t duration)

Funkcija priprema informaciju (u vidu tekstualnih stringova) potrebnu za prikazivanje proširenog info dijaloga koji sadrži ime i opis tekuće emisije, vreme i dužinu emitovanja, i žanr emisije. Vreme i dužina trajanja se dobija u BCD (Binary Coded Decimal) formatu. Prošireni info dijalog se samo prikazuje ako je trenutno aktivan manji dijalog, tj. ako je simpleDialogueFlag = 1, zatim postavlja simpleDialogueFlag na 0 i fullDialogueFlag na 1, da bi renderGui funkcija znala da iscrta prošireni dijalog. Na kraju se poziva renderDijalog. Kada korisnik pritisne taster 'back' na daljinskom upravljaču, poziva se funkcija **removeFullDialogue** koja uklanja prošireni dijalog, tako što postavlja fullDialogueFlag na 0 i poziva renderGui

void drawVolumeIndicator(uint8_t volume)

Funkcija priprema informaciju za iscertavanje indikatora za promenu jačine zvuka, postavlja logičku promenljivu volumeIndicatorFlag na 1 da bi renderGui funkcija znala da iscrta indikator za zvuk, i funkcija pozove renderGui. Zatim se pokreće timer koji će nakon 3 sekunde pozvati funkciju **removeVolumeIndicator** koja će ukloniti indikator za zvuk, tako što postavlja volumeIndicatorFlag na 0 i poziva renderGui.

void drawChannelChangeIndicator(uint8_t channelDigit)

Funkcija se poziva kada se unese prva cifra kanala pri direktnom unosu broja kanala od strane korisnika radi prikazivanja prve unete cifre na ekranu. Postavlja channelChangeIndicatorFlag na 1, pa poziva renderGui. Kada se unese druga cifra ili menja kanal na drugi način, uklanja se pozivom **removeChannelChangeIndicator** koji postavlja channelChangeIndicatorFlag na 0 i poziva renderGui

void renderGui(void)

Funkcija se poziva na kraju svake funkcije koja menja neka stanja vezana za grafičke elemente. Na osnovu flag promenljive i sadržaja tekstualnih stringova iscertava grafičke elemente koji treba da budu prikazani u grafičkom baferu, i prikazuje sadržaj na izlaznom ekranu. Funkcija je zaštićena objektima za međusobno isključivanje, sprečavajući više nita da izvrše crtanje istovremeno, time izazivajući grafički kvar.

5. Zaključak

Realizovana je aplikacija za TV prijemnik koja omogućava primanje komandi sa daljinskog upravljača. Realizovan skup komandi dozvoljavaju menjanje kanala, menjanje jačine zvuka, i prikazivanje informacionih dijaloga koji obavješavaju korisnika o broju tekućeg kanala, imenu, opisu, vremenu emitovanja, dužini trajanja, opisu žanra i informaciji o prisutnost titlova tekuće emisije.

U ranoj fazi testiranja su uviđeni problemi vezani za robusnost aplikacije, tj. aplikacija je pokazivala probleme pri brzom menjanju kanala kao što su grafičke greške i ponekad prestanak rada. Postoji i problem kod prikazivanja nestandardnih slova kao što su španska slova. Problemi vezani za robusnost su rešeni uvođenjem objekata za međusobno isključivanje niti kod osetljivih delova aplikacije kao što je iscrtavanje grafike i podešavanje filtra demultipleksera.