

**UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
NOVI SAD**

**Odsek/Smer/Usmerenje :Računarstvo i
automatika/Računarska tehnika i
računarske komunikacije**

DIPLOMSKI – MASTER RAD

Kandidat : Zloh Jan

Broj indeksa : E9353

**Tema rada : Jedno rešenje programske podrške za automatsko
testiranje TV prijemnika**

Mentor rada : Dr Teslić Nikola, vanr. prof.

Novi Sad, Novembar 2006.



KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj, RBR:	
Identifikacioni broj, IBR:	
Tip dokumentacije, TD:	Monografska publikacija
Tip zapisa, TZ:	Tekstualni štampani materijal
Vrsta rada, VR:	Master rad
Autor, AU:	Zloh Jan
Mentor, MN:	prof. Dr Nikola Teslić
Naslov rada, NR:	Jedno rešenje programske podrške za automatsko testiranje TV prijemnika
Jezik publikacije, JP:	Srpski
Jezik izvoda, Jl:	Srpski
Zemlja publikovanja, ZP:	Srbija
Uže geografsko područje, UGP:	Srbija
Godina, GO:	2006
Izdavač, IZ:	Autorski reprint
Mesto i adresa, MA:	Kovačica, Šturova 116
Fizički opis rada, FO: (poglavlja/strana/citata/tabela/slika/grafika/priloga)	8/50/0/2/26/0/0
Naučna oblast, NO:	Elektrotehnika i računarska tehnika
Naučna disciplina, ND:	Računarska tehnika
Predmetna odrednica/Ključne reči, PO:	
UDK	
Čuva se, ČU:	U biblioteci FTN, Novi Sad, Trg Dositeja Obradovića 6
Važna napomena, VN:	
Izvod, IZ:	
Datum prihvatanja teme, DP:	
Datum odbrane, DO:	02.11.2006
Članovi komisije, KO:	Predsednik: Prof. dr Vojin Šenk
	Član: Prof. dr Vladimir Kovačević
	Član, mentor: Prof. dr Nikola Teslić
	Potpis mentora



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



KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monographic's publication
Type of record, TR :	Word printed record
Contents code, CC :	Master work
Author, AU :	Zloh Jan
Mentor, MN :	Ph. D.E.E. Nikola Teslić
Title, TI :	One solution of software for automatic testing TV receiver
Language of text, LT :	Serbian
Language of abstract, LA :	Serbian
Country of publication, CP :	Serbia
Locality of publication, LP :	Serbia
Publication year, PY :	2006
Publisher, PB :	Author's reprint
Publication place, PP :	
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendices)	8/50/0/2/26/0/0
Scientific field, SF :	Electrical engineering and computer science
Scientific discipline, SD :	Electrical engineering
Subject/Key words, S/KW :	
UC	
Holding data, HD :	Library of FTN, Novi Sad, Trg Dositeja Obradovića 6
Note, N :	
Abstract, AB :	
Accepted by the Scientific Board on, ASB :	
Defended on, DE :	02.11.2006

Defended Board, DB :	President:	Prof. dr Vojin Šenk	
	Member:	Prof. dr Vladimir Kovačević	Mentor's sign
	Member, Mentor:	Prof. dr Nikola Teslić	

SKRAĆENICE

AVIL	- <i>Automated Verification Item List</i> , Programska podrška za automatsku proveru liste testova
CRT	- <i>Cathode Ray Tube</i> , Ekran zasnovan na katodnoj cevi
DBASE	- <i>Data Base</i> , Baza podataka
DDR	- <i>Double Data Rate</i> , Prenos dvostrukom brzinom
DHCP	- <i>Dynamic Host Configuration Protocol</i> , Protokol za dinamičku dodelu adresa
DRAM	- <i>Dynamic RAM</i> , Dinamčka RAM memorija
DTV	- <i>Digital Television</i> , Digitalna televizija
DVD	- <i>Digital Video Disc</i> , Digitalni video disk
DVI	- <i>Digital Video Interface</i> , Sprega za digitalni video
EDTV	- <i>Enhanced Definition TV</i> , Televizija povećane rezolucije
FFT	- <i>Fast Fourier Transformation</i> , Brza Furijeova transformacija
FPGA	- <i>Field Programming Gate Array</i> , Lokalno programabilna sekvencijalna mreža
HDTV	- <i>High Definition TV</i> , Televizija visoke rezolucije
HTML	- <i>HyperText Markup Language</i> , Programski jezik za kreiranje Internet stranica
IR	- <i>Infra Red</i> , Infra crvena svetlost
JPEG	- <i>Joint Photographic Experts Group</i> , Grupa koja je razvila JPEG format za čuvanje slika visokog kvaliteta
LCD	- <i>Liquid Crystal Display</i> , Ekran zasnovan na tečnim kristalima
LVDS	- <i>Low voltage differential swing</i> , Niskonaponska diferencijalna signalizacija
MFC	- <i>Microsoft Foundation Class Library</i> , Osnovne Microsoft-ove klase
MSDN	- <i>Microsoft Developer Network</i> , Pomoć za Microsoft Visual Studio razvojno okruženje
NFS	- <i>Network File System</i> , Protokol za mrežni pristup datotekama
NTSC	- <i>National Television System Committee</i> , Nacionalna Komisija za Televizijske Sisteme
OSD	- <i>On Screen Display</i> , Dodatna slika na ekranu TV-a
PC	- <i>Personal Computer</i> , Personalni računar
PAP	- <i>Picture And Picture</i> , Slika pored slike
PAL	- <i>Phase Alternating Line</i> , Linija izmenjive faze
PAT	- <i>Picture And Text</i> , Slika i tekst
PIP	- <i>Picture In Picture</i> , Slika u slici
PSNR	- <i>Peak Signal to Noise Ratio</i> , Odnos signal šum
RAM	- <i>Random Access Memory</i> , Memorija sa slučajnim pristupom
RGB	- <i>Red-Green-Blue output</i> , Crveni-Zeleni-Plavi signal na izlazu

ROI -*Region Of Interest*, Oblast od značaja

RTOS -*Real Time Operating System*, Operativnim sistem za rad u realnom vremenu

SDRAM - *Synchronous DRAM*, Sinhrona dinamička RAM memorija

SDTV -*Standard Definition TV*, Televizija standardne rezolucije

USB -*Universal Serial Bus*, Univerzalna serijska magistrala

VGC -*Video Graphics Controller*, Grafički video kontroler

SADRŽAJ

SKRAĆENICE	5
SADRŽAJ	7
1. UVOD.....	9
2. OPIS SISTEMA ZA AUTOMATSKO TESTIRANJE.....	11
3. OKRUŽENJE ZA RAZVOJ I TESTIRANJE PROGRAMSKE PODRŠKE ZA TESTIRANJE MULTIMEDIJALNIH UREĐAJA	15
3.1 OPIS MICRONAS VGCA EVALUTION BOARD RAZVOJNOG OKRUŽENJA.....	15
3.2 OPIS BBT ACQUISITION BOARD PLATFORME ZA SNIMANJE DIGITALNOG VIDEO ZAPISA.....	17
3.2.1 OPIS PLOČE.....	17
3.2.2 OSTVARIVANJE MREŽNE KONEKCIJE IZMEĐU BBT ACQUISITION BOARD PLATFORME I PC RAČUNARA	18
3.2.3 PRENOS SLIKE SA BBT ACQUISITION BOARD PLATFORME NA RAČUNAR	19
3.2.4 RUKOVAOC BBT ACQUISITION BOARD PLATFORME I PROGRAMSKA PODRŠKA ZA SNIMANJE I PRENOS SLIKE	19
3.3 OPIS QUANTUMDATA701 GENERATORA SLIKA ZA TESTIRANJE	20
3.3.1 UPRAVLJANJE QUATUMDATA701 GENERATOROM PUTEM SERIJSKE KOMUNIKACIJE	22
3.4 OPIS EMULATORA DALJINSKOG UPRAVLJAČ	22
3.4.1 UPRAVLJANJE EMULATOROM DALJINSKOG UPRAVLJAČA PUTEM SERIJSKE KOMUNIKACIJE	24
4. OPIS PROGRAMSKE PODRŠKE ZA AUTOMATSKO TESTIRANJE TV PRIJEMNIKA	25
4.1 FUNKCIONALNI OPIS PROGRAMSKE PODRŠKE.....	25
4.2 STRUKTURA PROGRAMSKE PODRŠKE ZA AUTOMATSKO TESTIRANJE.....	26
4.3 ALGORITAM ZA POREĐENJE SLIKA	27
4.3.1 OBJEKTIVNO OCENJIVANJE KVALITETA SLIKE.....	27
4.3.2 REALIZACIJA ALGORITMA ZA POREĐENJE SLIKA.....	28
4.4 ALGORITAM ZA PREPOZNAVANJE ZVUKA	29
4.4.1 REALIZACIJA ALGORITMA ZA POREĐENJE ZVUKA.....	31
4.5 RUKOVANJE PROGRAMSKOM PODRŠKOM ZA AUTOMATSKO TESTIRANJE	31
4.5.1 IZGLED PROGRAMSKE PODRŠKE ZA AUTOMATSKO TESTIRANJE	31
4.5.2 OPIS POSTUPKA ZA KREIRANJA TESTOVA.....	38
4.5.3 OPIS POSTUPKA ZA AŽURIRANJE TESTOVA	39
4.5.4 OPIS POSTUPKA ZA KREIRANJE PROJEKTA	39
4.5.5 IZVRŠAVANJA TESTOVA I PROJEKATA U OKVIRU PROGRAMSKE PODRŠKE ZA AUTOMATSKO TESTIRANJE.....	40
4.5.6 PREGLED REZULTATA TESTOVA PROGRAMSKE PODRŠKE ZA AUTOMATSKO TESTIRANJE.....	42
5. TESTIRANJE PROGRAMSKE PODRŠKE ZA AUTOMATSKO TESTIRANJE.....	44

6. POREĐENJE SA DRUGIM REALIZACIJAMA SISTEMA ZA AUTOMATSKO TESTIRANJE TV PRIJEMNIKA	45
7. ZAKLJUČAK	48
8. LITERATURA.....	50

1. Uvod

Neprestanim razvojem novih tehnologija, javlja se potreba za testiranjem TV prijemnika namenjenih za kreiranje i prikaz audio video sadržaja. TV prijemnici koji nalaze primenu u svakodnevnom životu postaju sve „inteligentniji“ i pružaju sve više različitih usluga, koje su pre samo nekoliko godina bile nezamislive. Kao jedna od bitnih karakteristika svakog multimedijalnog uređaja ističe se i njegova pouzdanost u radu. Iz tog razloga došlo se na ideju o realizaciji programske podrške za automatsko testiranje TV prijemnika.

Cilj diplomskog rada je realizacija programske podrške, kojom bi se uz odgovarajuću fizičku arhitekturu, automatizovala procedura testiranja TV prijemnika. Akcenat kod testiranja TV prijemnika se stavlja na njegovu funkcionalnost. Svaki test ispituje ispravnost odziva testiranog uređaja na komandu daljinskog upravljača. Kao uređaj za testiranje u okviru diplomskog – master rada, koristi se *Micronas VGCA EVA Evaluation Board* sa *VGC 5969B* čipom namenjen za rad u oblasti digitalne televizije tj. za obradu digitalnih video signala i njihov prikaz na ekranu. U daljem tekstu umesto *Micronas VGCA EVA Evaluation Board* razvojno okruženje, koristiće se samo oznaka VGCA EVA.

Potreba za testiranjem uređaja nikako nije mogla da zaobiđe i jednu od najprofitabilnijih industrija – televizijsku industriju, koja se nalazi u neprestanom razvoju. Sredinom devedesetih godina prošlog veka razvijeni su standardi za prenos televizijskog signala u digitalnom obliku. Digitalna televizija obuhvata prenos audio-vizuelnog sadržaja u digitalnom obliku kao i njihovu sinhronizaciju. Stvaranje i primena standarda koji se koriste u digitalnoj televiziji se vezuje za poslednju dekadu 20. veka, pri čemu je rad digitalne televizije prvi put demonstriran 1995 godine. Postoje brojne prednosti koje digitalna televizija poseduje u odnosu na analognu. Sadržaj koji se prenosi je binarno kodiran, što znači da je moguće izvesti njegovu kompresiju i na taj način smanjiti količinu podataka koji se prenose. U istom frekventnom opsegu, moguć je prenos više digitalnih kanala umesto jednog analognog. 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 objedinjavanja, digitalna televizija omogućava nove programske podrške kao što je interaktivna televizija koja omogućava izvršavanje računarskih programskih podrški pisanih u Java-i ili HTML-u. Digitalni prenos obezbeđuje bolji kvalitet slike i zvuka, koji više ne mogu biti oštećeni u prenosu interferencijom sa drugim signalima. Slika i zvuk koju digitalni signal nosi su isti kao i na izvoru emitovanja. Digitalna televizija je više od proste zamene za postojeću analognu televiziju. Osim što obezbeđuje mnogo bolji kvalitet slike i zvuka i njihovu različitu prezentaciju (mogućnost izbora formata slike 4:3 ili 16:9, kao i zvuka: mono, stereo ili višekanalni), digitalna televizija omogućava uvođenje novih usluga kao što su izbor jezika za titlovanje, izbor audio kanala, interaktivni i multimedijalni sadržaji, pristup Internetu itd.

Diplomski - master rad se sastoji iz 8 poglavlja. Poglavlje 1 sadrži u kratkim crtama opis cilja diplomskog rada kao i oblast moguće primene. U poglavlju 2 je opisan sistem za automatsko testiranje – lista uređaja potrebnih za njegovu realizaciju, njihova uloga u sistemu kao i međusobna povezanost uređaja. Poglavlje 3 sadrži opis načina upravljanja i rada svakog dela sistema za testiranje navedenog u prethodnom poglavlju. Opis načina rada programske podrške za testiranje, implementacija algoritama za poređenje slika i zvuka kao i način korišćenja programske podrške za automatsko testiranje opisan je u poglavlju 4. Poglavlje 5 sadrži opis oblika testiranja razvijene programske podrške, dok u poglavlju 6 je dat primer automatskog sistema za testiranje realizovanog od strane *Rohde&Schwarz* kompanije. U poglavlju 7 je u kratkim crtama opisana prednost razvijenog testiranja i mogućnosti daljeg razvoja programske podrške. Spisak korišćene literature i drugih izvora informacija su navedeni u poglavlju 8.

2. Opis sistema za automatsko testiranje

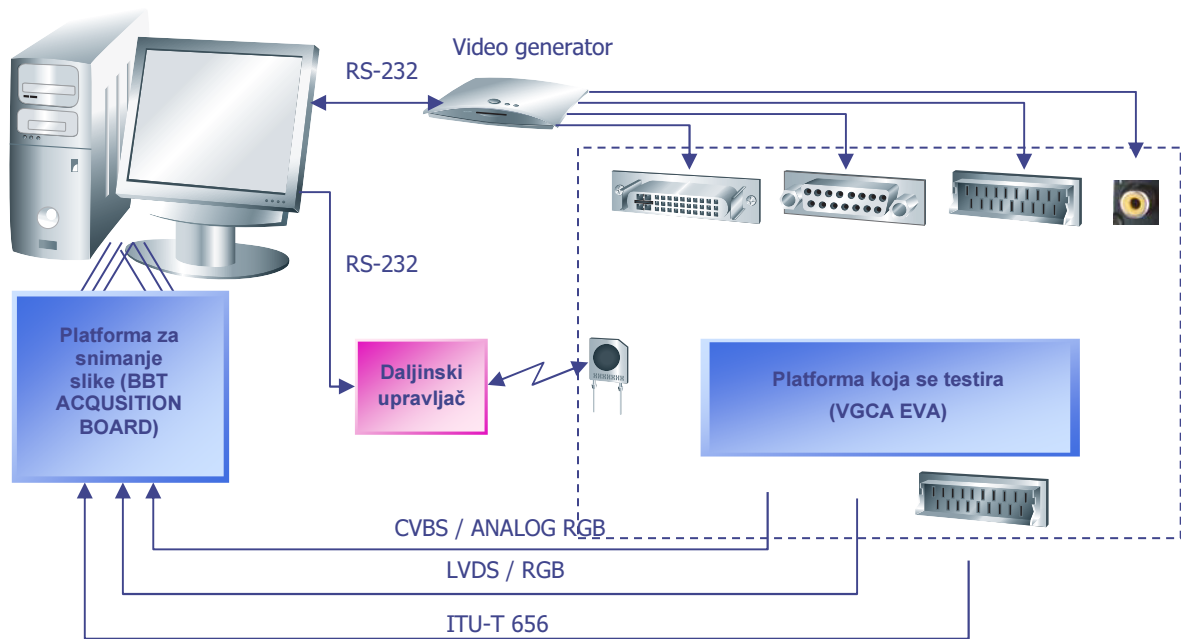
U okviru diplomskog – master rada, razvijeni su delovi sistema za testiranje, prikazanog na slici 1. U centru sistema se nalazi uređaj koji se testira (u ovom diplomskom - master radu se korsiti VGCA EVA okruženje). Pomoću generatora slike i zvuka (QuantumData), na ulaz testiranog uređaja se dovode odgovarajući signali slike i zvuka a korišćenjem emulatora daljinskog upravljača, testiranom uređaju se izdaju komande. Testirani uređaj na izlazu daje sliku, koja se snima uz pomoć platforme za snimanje (BBT ACQUISITION BOARD) i smešta se na masovnu memoriju PC računara. Programska podrška za testiranje upoređuje dobijenu sliku sa referentnom, i na osnovu toga zaključuje da li se testirani uređaj ponaša u skladu sa očekivanjima ili ne.



Slika 1: Sistem za automatsko testiranje

Centralnu ulogu u toku pravljenja i izvršavanja testova ima PC računar sa Windows XP operativnim sistemom, na kome je pokrenuta programska podrška za automatsko testiranje, AutomatedVIL. Njegova uloga je upravljanje sledećim delovima sistema, kako bi se postiglo željeno ponašanje celog sistema za testiranje:

- Upravljanje emulatorom daljinskog upravljača – za zadavanje komandi testiranom uređaju putem infracrvene veze
- Upravljanje QuantumData701 video generatorom – za generisanje video signala koji se dovodi na ulaz testiranog uređaja
- Upravljanje BBT ACQUISITION BOARD platformom – za snimanje slike sa video izlaza testiranog uređaja



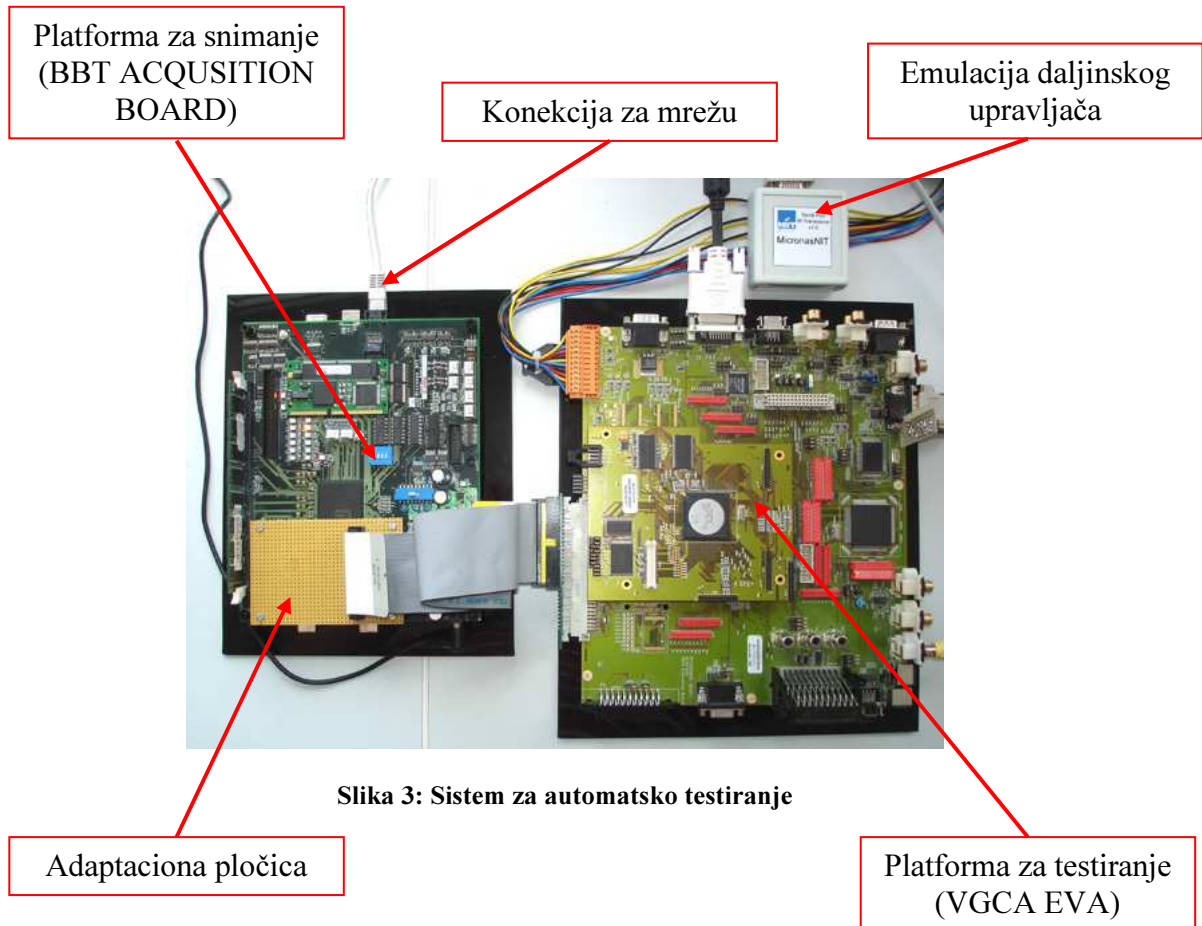
Slika 2: Šematski prikaz sistema za automatsko testiranje

Za komunikaciju računara sa pojedinim delovima sistema za testiranje koriste se različite veze i protokoli. Komunikacija sa QuantumData701 generatorom i daljinskim upravljačem ostvaruje se preko serijske veze po RS-232 protokolu, dok se za komunikaciju sa BBT ACQUISITION BOARD platformom koristi Ethernet veza sa TCP/IP protokolom.

Treba naglasiti da je veza sa daljinskim upravljačem jednosmerna, i zbog jednostavnosti njegovog logičkog kola ne postoji nikakav odziv sa tog elementa sistema. Putem serijske veze, daljinski upravljač preuzima reči dužine 14 bita formirane na način opisan u daljem tekstu, i pretvara ih u infracrvenu svetlost koju šalje ka VGCA EVA platformi.

Komunikacija sa QuantumData701 generatorom je dvosmerna, i zbog toga je moguće i slati podatke ka njemu, i preuzimati sa njega. Jedan od slučajeva kada se čitaju podaci sa generatora, je slučaj kada je potrebno proveriti da li je generator u toku pokretanja programske podrške AutomatedVIL povezan sa računarom, jer u suprotnom određeni testovi u kojima on ima ključnu ulogu ne bi se mogli pravilno izvršavati. Podaci koji se šalju ka njemu, odnosno parametri, vezani su za signal koji treba da generiše sam generator tj. vrsta signala (npr. DVI, S-VIDEO ...), rezolucija slike (npr. 640x480, 1024x768 ...), frekvencija osvežavanja slike (npr. 60Hz, 75Hz ...), test slika (npr. FlatRed, ColorBar, Grid ...) itd.

Komunikacija sa BBT ACQUISITION BOARD platformom je takođe dvosmerna. Programska podrška za testiranje šalje platformi zahteve za snimanje slike, dok od platforme dobija podatke koji predstavljaju sliku.



Slika 3: Sistem za automatsko testiranje

Na slici 2 i 3 se vidi način kako su komponente sistema za testiranje međusobno povezane i kako sistem izgleda. QuantumData701 generator za povezivanje sa VGCA EVA platformom koristi RCA i/ili DVI konektor, u zavisnosti od signala koji se koristi (kompozitni ili DVI signal). Povezivanje između BBT ACQUISITION BOARD i VGCA EVA platforme je realizovano namenskom spregom. To je izvedeno pomoću namenskog 80-pinskog kabla. VGCA EVA platforma je konfigurisana na taj način da na video izlazu preko ABC konektora daje digitalne R, G, i B komponente slike, koji se putem 80-pinskog kabla i adaptacione pločice prenose na BBT ACQUISITION BOARD platformu. U tabeli 1. data je lista signala koji se prenose putem namenskog kabla.

Tabela 1. Lista signala na ABC konektoru VGCA EVA platforme

Naziv signala	Vrednost	Opis
R[1..8]	Logička 0 +0.0 V Logička 1 +3.3 V	Signal za crvenu boju (8 bita)
G[1..8]	Logička 0 +0.0 V Logička 1 +3.3 V	Signal za zelenu boju (8 bita)
B[1..8]	Logička 0 +0.0 V Logička 1 +3.3 V	Signal za plavu boju (8 bita)
CLK	65 MHz Logička 0 0.0 V Logička 1 +3.3 V	Signal takta
DE	40 kHz Logička 0 0.0 V Logička 1 +3.3 V	Signal raspoloživosti podataka (eng. Data Enable)
VSYNC	40 Hz Logička 0 0 V Logička 1 +3.3 V	Signal za vertikalnu sinhronizaciju
HSYNC	40 kHz Logička 0 0 V Logička 1 +3.3 V	Signal za horizontalnu sinhronizaciju

3. Okruženje za razvoj i testiranje programske podrške za testiranje multimedijalnih uređaja

Za razvoj i testiranje programske podrške u okviru ovog diplomskog – master rada, korišćen je programski paket *Microsoft Visual C++ 6.0*. Programska podrška je napisana u C++ jeziku, uz korišćenje standardnih MFC klasa. Sve korišćene DLL biblioteke su statički povezane sa programskom podrškom, odnosno nalaze se u sklopu izvršne datoteke.

3.1 Opis Micronas VGCA Evalution Board razvojnog okruženja

VGCA EVA predstavlja razvojno okruženje proizvođača *Micronas*, namenjeno za rad sa LCD ili CRT ekranima. VGCA EVA sadrži sve glavne funkcije koje treba da poseduje napredni grafički kontroler, smešten u jednom monolitnom integrisanom kolu. U ovom diplomskom - master radu se koristi kao multimedijalni uređaj koji se testira.



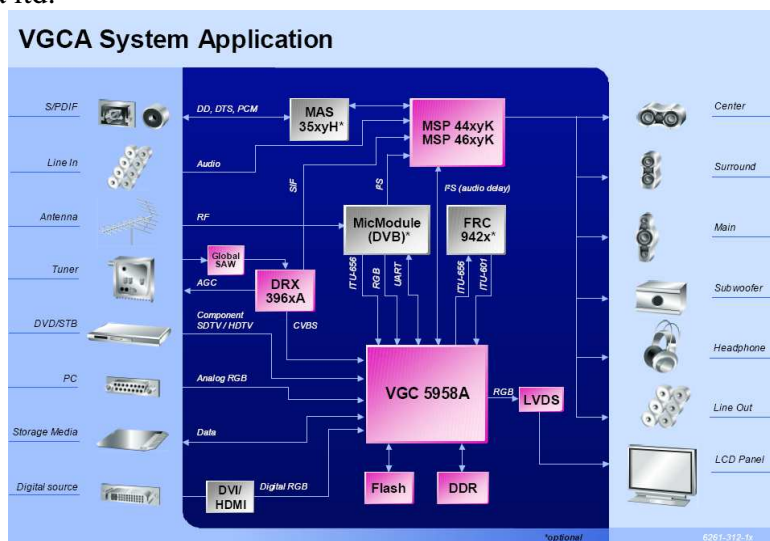
Slika 4: Izgled VGCA EVA razvojnog okruženja

Za prikaz slike na ekranu pomoću VGCA EVA razvojnog okruženja, mogu se birati različiti izvori signala. Svi tipovi signala (HDTV, EDTV, SDTV, PC) mogu se kombinovati na različite načine (čak i dvostruki HDTV).

U PIP (slika u slici) režimu prikaza slike, obrađuju se dva nezavisna asinhrona ulaza (Master i Slave), pri čemu se jedna slika nalazi unutar druge. U PAP (slika i slika) odnosno PAT (slika i tekst) režimu, dve slike se prikazuju jedna pored druge. Master (glavni) kanal je ograničen na rezoluciju 1920x1080, a Slave (sporedni) kanal na 1366x768. Master i Slave kanal se mogu slobodno podešavati po rezoluciji, čak i ako se kombinuju različiti tipovi signala. Platforma obrađuje digitalne i analogne video signale (nabrojane u daljem tekstu), a za konekciju za ekranom zahteva samo LVDS dekodir.

Programski kod, video podaci i teletext stranice smešteni su na spoljašnjem DDR SDRAM-u. Pri priključenju napajanja programski kod se kopira iz FLASH memorije u SDRAM memoriju. Od komponenti prisutnih na VGCA EVA platformi tu su: MSP 44xyK zadužen za obradu analognih audio signala koji dolaze sa konektora ili sa birača kanala, MAS 352x koji se može upotrebiti za višekanalno dekodovanje zvuka, MDE 95xxD na opcionom Mic modulu koji dekodira digitalne audio i video servise. Digitalni video sa prevodom se prenosi do VGCA EVA-a putem ITU656 sprege. VGCA EVA kombinuje analognu i digitalnu sliku i OSD.

VGCA EVA platforma je podržana širokim spektrom razvojnih alata kao što je GNU C++ prevodilac, JTAG bazirana fizička arhitektura za otklanjanje grešaka (eng. debugger), operativni sistem u realnom vremenu (RTOS), OSD builder, teletext dekodir, JPEG pregled slika itd.



Slika 5: Međusobna povezanost komponenti VGCA EVA ploče

Podržani su sledeći ulazni video signali:

- Analogni signali:
 - CVBS (svi TV standardi), Y/C (S-video)
 - YUV, YPbPr sa sinhronizacijom na Y (DVD, set-top box)
 - YUV+H/V, YPbPr+H/V
 - RGB sa sinhronizacijom na G (posebni VGA)
 - RGB + CVBS + FBL (SCART sa odvojenim OSD)
 - RGB + Y/C + FBL
 - RGB + H/V (PC signal)

- Digitalni signali:
 - ITU656, DS656, ITU601
 - 24 bit RGB or YUV

Podržani su sledeći izlazni video signali:

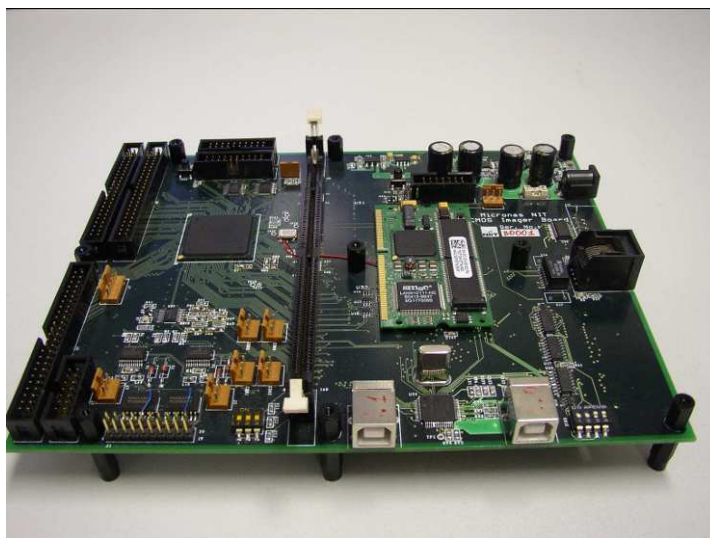
- Analogni signali:
 - CVBS, Y/C
 - Y+C interno dodat da se dobije CVBS iz Y/C
- Digitalni signali:
 - ITU656, DS656, ITU601

3.2 Opis BBT ACQUISITION BOARD platforme za snimanje digitalnog video zapisa

3.2.1 Opis ploče

Za preuzimanje slike sa VGCA EVA platforme, zadužena je *MicronasNIIT* BBT ACQUISITION BOARD platforma. Osnovne karakteristike platforme su:

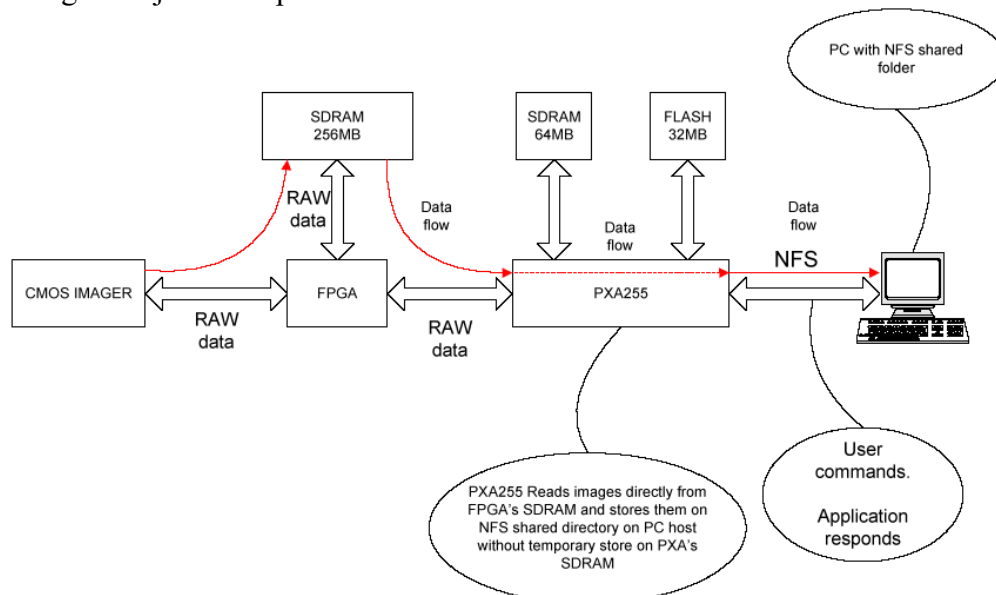
- Xilinx Spartan FPGA čip
- Do 2 GB SDRAM memorije, PC100 ili PC133
- Xilinx FLASH PROM
- Podrška za USB 2.0 konekciju
- Podrška za 100 Mb mrežnu konekciju za prenos sadržaja ka PC računaru
- Podrška za RS-232 serijsku konekciju



Slika 6: BBT ACQUISITION BOARD platforma

Xilinx Spartan familija FPGA-a čipa podržava pet različitih načina konfiguracije, pri čemu su na *BBT ACQUISITION BOARD* ploči podržana dva načina:

- Ploča poseduje *Xilinx* FLASH PROM memoriju, pri čemu se ova memorija može programirati korišćenjem *Xilinx* iMPACT alata i *Xilinx* JTAG kabla. Pri priključivanju napajanja na ploču, sadržaj FLASH memorije će konfigurirati FPGA ploču koristeći Master Serial konfiguracioni način. Ovaj način konfigurisanja se koristi u slučaju kada je programski kod stabilan i ne menja se često.
- FPGA ploča može biti konfigurisana iz PXA255 procesora, korišćenjem konfiguracione datoteke koja se nalazi na NFS uređaju. U ovom slučaju koristi se Slave Serial konfiguracioni način. Ploča se konfigurira tokom faze učitavanja Linux jezgra koji se izvršava na PXA255. Ovakav način konfigurisanja se koristi tokom razvoja ili u slučaju kada korisnik ne poseduje JTAG kabl. U toku razvoja i testiranja ovog diplomskog - master rada, upravo ovaj način je korišćen prilikom konfigurisanja FPGA platforme.



Slika 7: Tok podataka na BBT ACQUISITION BOARD platformi

3.2.2 Ostvarivanje mrežne konekcije između BBT ACQUISITION BOARD platforme i PC računara

Za prenos sadržaja snimljene slike sa platforme na PC računar koristi se mrežna (eng. Ethernet) konekcija (NFS). Da bi se ostvarila mrežna konekcija, mrežni kabal mora biti povezan sa platformom tokom učitavanja Linux okruženja na platformi. Na taj način se omogućava kreiranje mrežne sprege. Nakon povezivanja platforme sa mrežom putem DHCP protokola, platforma dobija odgovarajuću IP adresu. Dalja komunikacija sa platformom putem mrežne konekcije, vrši se posredstvom `telnet` komande iz Linux okruženja PC računara ili korišćenjem `putty` programske podrške iz Windows okruženja. Za mrežni pristup BBT ACQUISITION BOARD platformi posredstvom `putty` programske podrške, neophodno je znati IP adresu platforme. Taj podatak je moguće saznati izdavanjem komande `ifconfig` sa PC računara putem serijske veze sa platformom, koristeći alat kao što *Hyper Terminal* ili slično. U slučaju potrebe za promenom IP adrese ploče koristi se sledeća komanda:

```
ifconfig eth0 <desired_IP_addr>
```

3.2.3 Prenos slike sa BBT ACQUISITION BOARD platforme na računar

NFS (Network File System) predstavlja protokol koji omogućava računaru da pristupa datotekama preko lokalne mreže kao da su one smeštene na masovnoj memoriji računara. Za prenos slike sa BBT ACQUISITION BOARD platforme na računar, neophodno je pokrenuti NFS server na PC računaru, pomoću kojeg se deli sistem za razmenu datoteka. Takođe na računaru moraju postojati UNIX servisi za Windows. Za povezivanje (eng. mount) NFS deljenog direktorijuma na PC računaru i lokalnog direktorijuma na BBT ACQUISITION BOARD platformi primenom **putty** programske podrške koristi se sledeća komanda:

```
mount -o rsize=8192,wsiz=8192 <host_ip_addr>:<NFS_dir_name> <mount_dir>
```

3.2.4 Rukovaoc BBT ACQUISITION BOARD platforme i programska podrška za snimanje i prenos slike

Na BBT ACQUISITION BOARD platformi je smešteno nekoliko datoteka neophodnih za njeno konfiguiranje i rad:

- **fpga.ko** - rukovalac za FPGA. Učitava se prilikom startovanja Linux-a
- **bbt_app** - glavna programska podrška čiji rad omogućava svu neophodnu funkcionalnost za snimanje i prenos slike. Pokreće je korisnik putem **putty** programske podrške
- **bbt_init.sh** - tekst komande koje izvršavaju sve neophodne korake da bi se obezbedilo pravilno funkcionisanje BBT ACQUISITION BOARD platforme. Izvršava se tokom startovanja Linux-a. Na ovaj način se podešava vreme čekanja SRAM memorije, učitava se FPGA i I2C rukovaoc, konfiguriše se FPGA itd.

fpga.ko – dužnost rukovaoca na ovoj platformi je da obezbedi komunikaciju između korisnikove programske podrške (**bbt_app**) i FPGA čipa. Omogućava mapiranje SDRAM memorije platforme prema korisniku, i na taj način obezbeđuje mogućnost čitanja sadržaja memorije direktno iz korisnikove programske podrške.

Pre pokretanja korisničke programske podrške (**bbt_app**), neophodno je učitavanje rukovaoca te platforme. Programska podrška putem rukovaoca postavlja sve neophodne parametre, kako bi platforma radila u skladu sa zahtevima za njen rad. To su sledeći parametri: broj slika za snimanje, vremensko kašnjenje prilikom pokretanja, I2C Imager resetovanje, način rada platforme itd.

Nakon snimanja slike, programska podrška čita sadržaj sa SDRAM memorije platforme, i kopira ga u lokalni direktorijum na koji je montiran NFS deljeni direktorijum na PC računaru. Svaka slika se snima kao posebna datoteka u RAW (sirovom) obliku bez ikakvog zaglavlja, sa rezolucijom 1366 x 768. Svaki tačka slike (eng. piksel) je veličine 24 bita, po 8 bita za svaku R, G i B komponentu.

bbt_app - obezbeđuje sledeće funkcionalnosti:

1. Podešavanje broja slika koje se snimaju
2. Okidanje (TRIGGERING) FPGA za prenos slike
3. Obaveštavanje korisnika o završetku snimanja slike
4. Prenos slika prema NFS deljenom direktorijumu na PC računaru

BBT ACQUISITION BOARD platforma tokom rada prolazi kroz nekoliko stanja. Pokretanjem **bbt_app** programske podrške otvara se TCP utičnica (eng. socket) preko koje platforma komunicira sa PC računarom posredstvom mreže, i platforma se dovodi u stanje **CONNECTIN_IN_PROGRESS_STATE**. U tom stanju formira se programska nit, koja neprestano očekuje povezivanje sa PC računarom putem TCP utičnice. Po ostvarivanju veze nit prekida sa radom i platforma prelazi u **CONNECTED_STATE** stanje. U novom stanju se kreira nova nit, koja preko ostvarene veze očekuje komande sa PC računara. Primita komanda se obrađuje i u zavisnosti od izdate komande, platforma prelazi u odgovarajuće stanje. U slučaju prijema **MSG_SET_CAPTURE_PICTURE** poruke platforma prelazi u **CAPTURE_PICTURE_STATE** stanje, u kojem se rukovaocu platforme izdaje zahtev za snimanje slike u SDRAM memoriju platforme. Po završetku snimanja generiše se prekid (eng. interrupt), kojim platforma prelazi u **COPY_PICTURE_STATE** stanje. U tom stanju vrši se kopiranje sadržaja SDRAM memorije sa slikom u datoteku koja se nalazi u direktorijumu koji dele platforma i PC računar posredstvom NFS protokola.

3.3 Opis QuantumData701 generatora slika za testiranje

Za potrebe testiranja analognih i digitalnih video ulaza VGCA EVA platforme koristi se *QuantumData* model 701 video test generator.



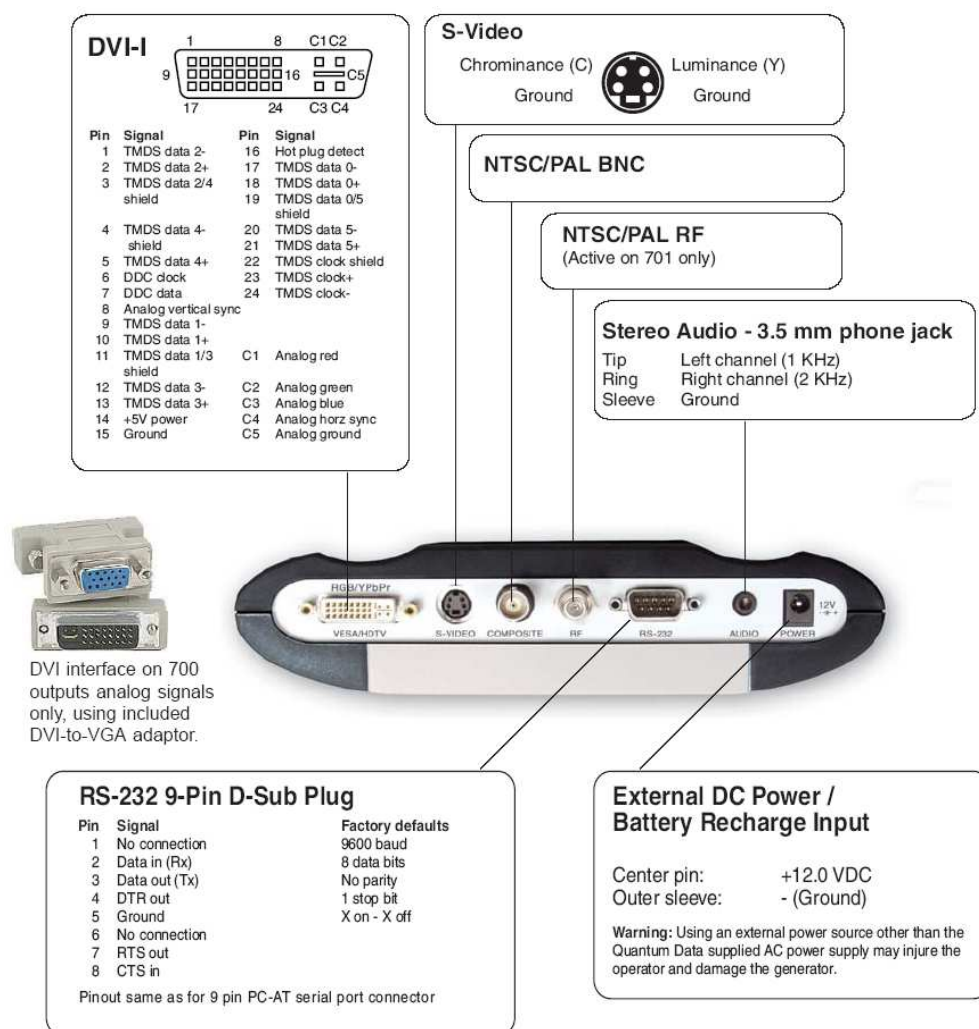
Slika 8: QuantumData 701 video test generator

Na izlazu test generatora zastupljeni su sledeći izlazni konektori:

- S-Video izlaz
- Kompozitni izlaz
- DVI izlaz

S-Video i kompozitni izlaz se koriste za analogne signale. Formati slike koji su zastupljeni na ovim konektorima su standardni NTSC (525 linija/60 Hz) i PAL (625/50 Hz) format kao i njihove varijacije.

DVI izlaz koristi digitalni signal. Podržan je veliki broj formata ovog tipa, različitih rezolucija od 640x480 pa sve do 1920x1080, sa različitim frekvencijama osvežavanja. Generator poseduje 20 test slika, pomoću kojih je moguće proveriti širok spektar kriterijuma koje treba da zadovolji uređaj koji se testira. Na taj način je moguće istestirati boje, veličinu, linearnost, konvergenciju, fokus, perzistentnost i druge osobine vezane za sliku.



Slika 9: Izlazni konektori za QuantumData701

3.3.1 Upravljanje QuatumData701 generatorom putem serijske komunikacije

Generator poseduje mogućnost promene vrste video signala, slike i raznih drugih detalja vezanih za signal koji generiše, putem serijske komunikacije preko PC računara. Na taj način se isključuje potreba za rukovanjem generatorom putem tastera na njemu, već se to radi slanjem odgovarajućih komandi preko serijske veze. Kao protokol za serijsku komunikaciju, koristi se standardni RS-232 protokol. Generator zahteva da serijska komunikacija poseduje sledeće osobine:

- Broj bita u sekundi: 9600
- Dužina reči: 8 bita
- Paritet: Bez pariteta
- Broj stop bita: 1 bit
- Kontrola protoka: Xon/Xoff

Programska podrška za upravljanje generatorom je realizovana u obliku klase *CQuantumData*, koja sadrži sve neophodne attribute i metode potrebne za upravljanje generatorom. Za rad sa generatorom koriste se sledeće metode:

```
int PortOpen(char* PortName, DWORD Baud, BYTE Length, BYTE Parity, BYTE Stop)
int SendQDCommand(char *apcBuf)
int ReadQDResponse(char *apcBuf, unsigned int anLen)
int PortClose()
```

Metodom *PortOpen* otvara se serijska veza između generatora i PC računara, sa parametrima koji su prosleđeni kao argumenti te metode. Metoda se poziva samo tokom inicijalizacije programske podrške za testiranje tj. prilikom njenog pokretanja. Metoda *SendQDCommand* služi za slanje komandi prema generatoru. Komande su u obliku predefinisanih ASCII znakova koje prepoznaje generator i na odgovarajući način reaguje na njih. Za potrebe čitanja odgovora generatora na neku komandu, koristi se metoda *ReadQDResponse*. U trenutnoj verziji programske podrške za testiranje, koristi se samo u slučaju provere povezanosti generatora sa PC računarom. Prilikom zatvaranja programske podrške za testiranje, pozivom *PortClose* metode zatvara se serijska komunikacija sa generatorom i oslobađa se za dalji rad. Klasa *CQuantumData* je instancirana u okviru klase glavnog prozora programske podrške za testiranje.

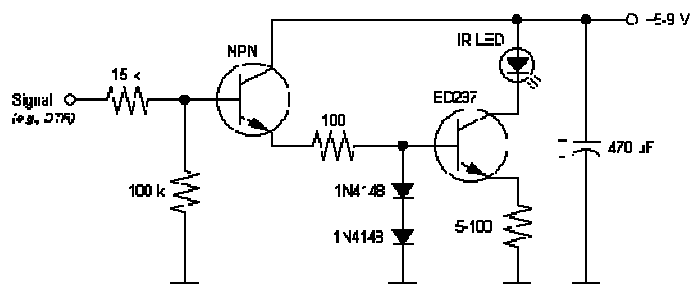
3.4 Opis emulatora daljinskog upravljača

Da bi se odredila reakcija testiranog uređaja (VGCA EVA) na neku komandu, potrebno je uređaju izdati komandu. To je postignuto emulacijom daljinskog upravljača tj. slanjem komande putem infracrvene veze, i u tu svrhu koristi se programska podrška za kontrolu daljinskog upravljača. Pomoću nje je moguće, uz odgovarajuću fizičku

arhitekturu koja se povezuje sa serijskim priključkom na PC računaru, dekodovati i poslati infracrvene signale ka odgovarajućem uređaju.



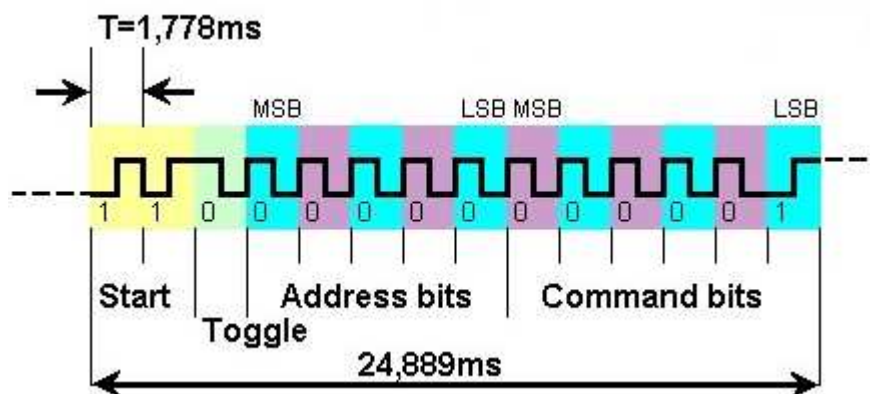
Slika 10: Infracrveni predajnik



Slika 11: Šema infracrvenog predajnika

Jedan od najrasprostranjenijih protokola za infracrvenu komunikaciju je RC5 protokol, koji je primenjen i u ovom slučaju. RC5 obezbeđuje 2048 različitih komandi koje su podeljene u 32 različite adrese od kojih svaka poseduje 64 različite komande. Svaki uređaj koji prima komandu ovim protokolom prepoznaje svoju adresu u njoj, i samo on reaguje na nju. Prenesena reč sastoji se od 14 bita, koji su podeljeni na sledeći način:

- 2 bita za automatsku kontrolu na prijemniku signala
- 1 bit za detekciju da li se konstantno šalje jedna te ista komanda
- 5 bita za adresu prijemnika signala
- 6 bita za komandu koja se šalje



Slika 12: RC5 protokol

RC5 protokol koristi bifaznu modulaciju, što znači da se svaki bit u signalu sastoji iz dva dela koja nikad nisu ista, odnosno, vrednost signala u jednom trenutku se detektuje na osnovu prelaska signala sa niskog na visoki nivo ili obrnuto. Po RC5 protokolu 0 se predstavlja prelaskom sa visokog nivoa na niski a 1 prelaskom sa niskog nivoa na visoki nivo. Da bi se obezbedila druga funkcija svakog tastera upotrebom SHIFT tastera, potrebno je vrednost adrese u svakoj reči uvećati za 1. Za potrebe testiranja VGCA EVA platforme, izvršena je emulacija *Micronas* daljinskog upravljača sa ukupno 88 komandi.

3.4.1 Upravljanje emulatorom daljinskog upravljača putem serijske komunikacije

Daljinskim upravljačem se upravlja na isti način kao i sa generatorom video signala opisanim u prethodnom delu tj. koristi se serijska veza sa RS-232 protokolom. Programska podrška za upravljanje simulatorom daljinskog upravljača je realizovana u obliku niza funkcija, pri čemu su najznačajnije:

```
int RC_Init(CString sConfigFile)
int RC_SendRemote(CString command)
```

Funkcijom *RC_Init* inicijalizuju se sve strukture neophodne za rad simulatora daljinskog upravljača. Vrednosti polja struktura su zapisane u datoteci, koja se prosleđuje kao argument funkcije. Datoteka sadrži i listu komandi koje podržava simulator daljinskog upravljača. Svaka komanda je opisana kodom u obliku 14 bita opisanih na prethodni način. Komande se šalju *RC_SendRemote* funkcijom, čiji argument je niz karaktera u obliku “komanda x”, pri čemu *komanda* predstavlja ime komande čija definicija postoji u inicijalizacionoj datoteci simulatora daljinskog upravljača, a *x* broj ponavljanja te komande.

4. Opis programske podrške za automatsko testiranje TV prijemnika

4.1 Funkcionalni opis programske podrške

Programska podrška za automatsko testiranje – AutomatedVIL (Automated Verification Item List) predstavlja programski paket, koji omogućava testiranje funkcionalnosti multimedijalnih uređaja. Zasniva se na poređenju slike i zvuka preuzetih u toku pravljenja testa sa slikom i zvukom koji se prikazuju u fazi testiranja. Na osnovu poređenja može se zaključiti da li je odziv uređaja na komandu bio očekivajući ili ne.

Programska podrška omogućava izvršavanje pojedinačnih testova kao i izvršavanje čitave kolekcije tj. liste testova. Glavne karakteristike aplikacije su:

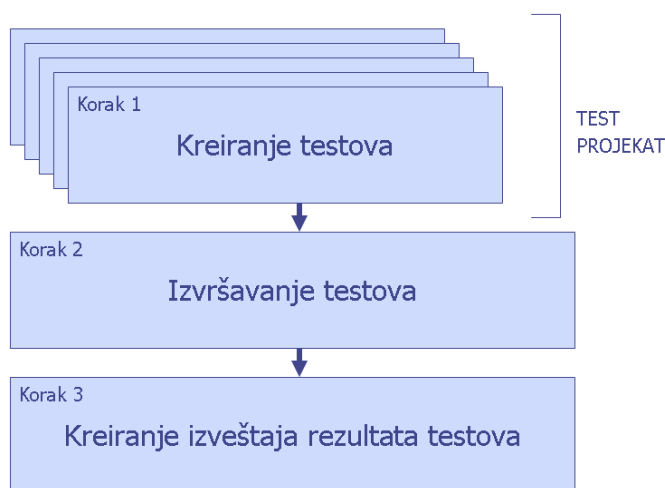
- Kreiranje novih testova
- Ažuriranje postojećih testova
- Brisanje testova
- Izvršavanje testova
- Kreiranje kolekcije testova
- Izvršavanje kolekcije testova
- Generisanje rezultata testova (HTML i dBASE oblik)
- Vizuelni prikaz toka testiranja
- Upravljanje QuantumData generatorom video signala
- Upravljanje emulatorom daljinskog upravljača za slanje komandi

Prilikom izbora testa, korisniku se prikazuju podaci o izabranom testu: kratak opis testa, kao i kriterijumi koji zadovoljavaju taj test, pojedinačni koraci koji se izvršavaju u toku testa i očekivanje promene koje su posledice takvih komandi.

Pokretanjem testa najpre se vrši provera da li su zadovoljeni očekivani početni uslovi, i ukoliko je to zadovoljeno, prelazi se na zadavanje niza komandi pri čemu se posmatra reakcija na komande. Komande se šalju putem infra crvene veze odnosno IR (Infra Red) poruka, koje u stvari predstavljaju komande sa daljinskog upravljača. Moguće su 3 vrste rezultata testa:

- Pass - test je zadovoljio kriterijume testa
- Fail - test nije zadovoljio kriterijume testa
- Inccclusive - nije moguće odrediti rezultat testa iz nekog razloga

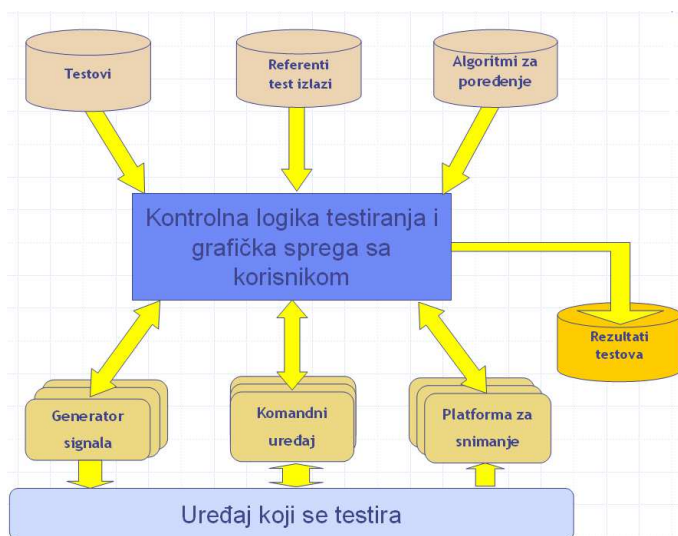
Krajnji rezultati testiranja, odnosno, da li je test prošao ili ne, kao i drugi relevantni podaci (datum i vreme izvršavanja testa, ime lica koje je testiralo, komentar itd.) smeštaju se u bazu podataka, pri čemu je moguće napraviti izveštaj rezultata testova u HTML obliku. Međurezultati svakog testa, kao što je stepen poklapanja slika i zvuka nakon svake pojedinačne komande koja se izdaje putem daljinskog upravljača, smeštaju se u tekstualnu datoteku, kako bi korisnik u slučaju neuspešnog izvršavanja testa stekao uvid u kom trenutku tj. nakon koje komande je test bio neuspešan.



Slika 13: Koraci u korišćenju AutomatedVIL aplikacije

4.2 Struktura programske podrške za automatsko testiranje

Na slici 14. je prikazana struktura realizovane programske podrške za automatsko testiranje.



Slika 14: Struktura programske podrške za automatsko testiranje

Opis testa i referentni test izlazi predstavljaju ulazne parametre programske podrške u procesu izvršavanja testova. Kontrolna logika predstavlja jezgro programske podrške. Na osnovu definicije testa i skupa test vektora koji su definisane za taj test, kontrolna logika izvršava test tj. upravlja uređajima za generisanje signala, uređajem za zadavanje komandi i platformom za snimanje, i procenjuje rezultate testova. Izbor pri korišćenju tipa generatora signala zavisi o funkcionalnosti testiranog uređaja koja se želi podvrgnuti testu (slika, zvuk, teletekst itd.). Algoritmi za poređenje predstavljaju deo programske podrške namenjene za poređenje izlaza testiranog uređaja sa odgovarajućim referentnim uzorkom. Jedini izlazni parametar testiranja predstavljaju rezultati testova, koji se smeštaju u odgovarajuću bazu podataka. Baza podataka se koristi kao osnova u kreiranju drugih oblika prezentovanja rezultata (HTML, EXCEL itd.). Reprerentacija rezultata testiranja predstavlja važan aspekt programske podrške. Pravilna organizacija rezultata i pravovremeno izveštavanje o pronađenim nepravilnostima u radu testiranog sistema, kao i njihovo otklanjanje, jedan je od ključnih elemenata potrebnih za unapređenje kvaliteta realizovane testirane programske podrške.

Treba nagovestiti da se ovim rešenjem pored TV prijemnika mogu testirati i drugi multimedijalni uređaji, jer programska podrška za automatsko testiranje ne zavisi od načina realizacije konkretnog testiranog uređaja koji se posmatra kao crna kutija.

4.3 Algoritam za poređenje slika

4.3.1 Objektivno ocenjivanje kvaliteta slike

Subjektivno ocenjivanje kvaliteta slike predstavlja procenu gde posmatrač vizuelno procenjuje kvalitet po određenom kriterijumu koji je definisan nekim standardom (npr. ITU-R BT.500-10). Takav način rada ima nekoliko nedostataka. Na prvom mestu je razlika u rezultatima ocenjivanja kod svakog posmatrača koji učestvuje u testiranju. Kao jedan od nedostataka je i to što se ovakav način testiranja primenjuje nad kratkim video testovima. Zbog svega toga, ljudi koji rade na razvoju video sistema sve više primenjuju objektivno ocenjivanje kvaliteta slike, ali pored svega objektivno ocenjivanje još uvek nije u potpunosti zamenilo subjektivno testiranje.

Verovatno jedan od najkorišćenijih načina za objektivno merenje kvaliteta slike je PSNR algoritam (eng. Peak Signal to Noise Ratio). PSNR koristi logaritamsku skalu i bazira se na srednjoj kvadratnoj grešci između referentne (originalne) slike I i testirane slike M , rezolucije $m \times n$. MAX_I predstavlja maksimalnu vrednost koju može imati svaka tačka (eng. piksel) u slici.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2$$

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$

PSNR se računa na relativno jednostavan način, i zbog toga predstavlja vrlo popularno merilo kvaliteta. Najčešće se koristi kao metoda za poređenje kvaliteta kompresovane i dekompresovane slike. Međutim i pored svega toga, PSNR poseduje nekoliko ograničenja. Ovakav način procene kvaliteta slike zahteva postojanje 'netaknute' originalne slike za poređenje, što u nekim slučajevima može predstavljati otežavajuću okolnost, jer se postavlja pitanje kako proveriti perfektnu oštrinu ('čistoću') referentne slike. Još značajnije ograničenje predstavlja i činjenica da rezultati PSNR metode nisu usaglašeni sa subjektivnom ocenom kvaliteta slike kao što je npr. ITU-R 500. Za neku datu sliku, visoka vrednost PSNR-a predstavlja procenu visokog kvaliteta slike u odnosu na referentnu dok niska vrednost PSNR-a predstavlja nizak kvalitet slike. Međutim, u nekim slučajevima pojedinačne vrednosti PSNR-a nisu uvek jednake subjektivnom merilu kvaliteta.

Zbog nabrojanih problema, radi se na razvoju sofisticiranijih objektivnih testova koji daju rezultate bliske rezultatima subjektivnih testova. Predložen je izvestan broj različitih predloga, ali trenutno nijedan se nije pokazao kao alternativa subjektivnom testu.

4.3.2 Realizacija algoritma za poređenje slika

Kao algoritam za poređenje slika, primenjen je prethodno opisan algoritam baziran na oduzimanju slika (PSNR eng. Peak Signal to Noise Ratio). Razlika između dve slike se određuje pronalaženjem razlike između odgovarajućih piksela u slikama. Da bi ovaj algoritam bio primenljiv, slike moraju biti poravnate tako da se određene tačke poklapaju i osvetljaj piksela mora biti isti, što predstavlja nedostatak. Kao rezultat, dobija se slika koja je potpuno crna u slučaju potpunog poklapanja tih slika, odnosno dolazi do pojave svetlih tačaka (eng. piksela) u rezultujućoj slici na onim mestima gde se slike razlikuju. U daljem tekstu dat je opis realizacije algoritma za poređenje slika.

```
snr_r = r/n;
snr_r = snr_r / 255*255;
snr_r = -10*log10(snr_r);

snr_g = g/n;
snr_g = snr_g / 255*255;
snr_g = -10*log10(snr_g);

snr_b = b/n;
snr_b = snr_b / 255*255;
snr_b = -10*log10(snr_b);

snr = snr_r + snr_g + snr_b;
```

Vrednosti r , g , b predstavljaju kvadrat razlike crvene, zelene, odnosno plave boje između dveju slika. Svaka od tih vrednosti se deli sa ukupnim brojem tačaka u slici kako bi se odredila srednja vrednost odstupanja po svakoj komponenti u svakoj tački. Da bi se ta vrednost normalizovala tj. svela na vrednost između 0 i 1 ona se deli sa kvadratom broja 255, jer upravo je to najveća vrednost koju može imati svaka komponenta boje opisana sa 8 bita. Nakon toga, izračunava se stepen poklapanja svake komponente slike primenom logaritamske funkcije. Krajnji rezultat tj. stepen poklapanja dve slike predstavlja zbir vrednosti odnosa signal šum sve tri komponente. Algoritam za poređenje slika realizovan je obliku klase *CAudioVideoProcessing*. Iz imena klase da se zaključiti da je pomoću iste klase realizovano i poređenje zvuka. Time se želelo postići da se na jednom mestu nalaze

sve metode neophodne za prepoznavanje slike i audio zapisa. U oblasti prepoznavanja slike, ova klasa obezbeđuje:

- Podešavanje nivoa prolaznosti poklapanja slike
- Određivanje vrednosti poklapanja dve slike

```
void SetSNR(int iParam)
char VideoProcessingShell(VideoDataType tData, float *brunt, float *psnr)
```

Metodom *SetSNR* se zadaje prag, na osnovu kojeg metoda za poređenje slika procenjuje, da li se dve slike poklapaju ili ne. Prag je brojčana vrednost, ne manja od 0, a empirijskim putem je određeno da se vrednost praga nalazi u intervalu od 100 do 130, što zavisi od testirane platforme i od rezolucija slika koje se porede. Metoda *VideoProcessingShell* vrši poređenje slika. Pre poziva neophodno je obezbediti da struktura *tData* (tabela 2), koja se prosleđuje kao argument metode, popuni odgovarajućim podacima. Struktura pored informacija o rezoluciji slika koje se porede, sadrži i adrese memorijskih zona na kojima su smeštene slike koje se porede, kao i adresu memorijske zone koja predstavlja razliku tih slika. Pozivom metode, iterativno se oduzimaju odgovarajući bajti u memorijskim zonama koji predstavljaju komponente boja (RGB) slika. Tako dobijena razlika se kvadrira i akumulira u odgovarajućoj promenljivoj za svaku komponentu slike. Prolaskom kroz celu memorijsku zonu, te tri promenljive sadrže ukupnu vrednost razlike tačaka date komponente na celoj slici. U daljem postupku sledi primena jednačina datih u prethodnom delu. Ukoliko je krajnja vrednost jednačina veća od vrednosti zadate kao prag poklapanja, metoda *VideoProcessingShell* procenjuje da se date slike poklapaju.

Slike su predstavljene u BMP formatu, što znači da je svaki piksel predstavljen sa tri komponente – R (crvena), G (zeleni) i B (plava). Svaka komponenta ima širinu od 8 bita, tako da veličina datoteke sa slikom rezolucije 1366 x 768 koja se koristi u algoritmu ima veličinu 3 073,5 MB.

```
IMAGE_SIZE = NUMBER_OF_BYTES_PER_PIXEL * IMAGE_WIDTH * IMAGE_HEIGHT
IMAGE_SIZE = 3 * 1366 * 768 = 3 147 264 [B] = 3 073,5 [MB]
```

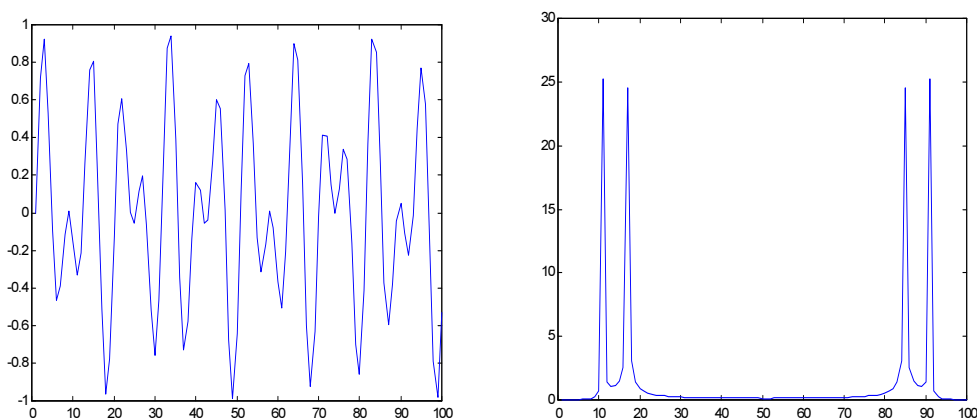
Tabela 2. Polja strukture *tData*

Naziv	Tip	Opis
width	int	Broj tačaka u liniji slike
height	int	Broj linija u slici
origData	unsigned char*	Adresa memorije referentne slike
testData	unsigned char*	Adresa memorije testirane slike
diffData	unsigned char*	Adresa memorije namenjena za međurezultate rada

4.4 Algoritam za prepoznavanje zvuka

Prepoznavanje zvuka se bazira na matematičkoj analizi audio signala korišćenjem Furijeove transformacije. Pomoću transformacija određeni signal se prevodi iz jednog

domena u neki drugi, u kom je lakše analizirati dati signal. U ovom slučaju pomoću Furijeove transformacije vrši se pretvaranje signala iz vremenskog domena gde je vreme nezavisna promenljiva u frekventni domen (slika 15). Furijeovom transformacijom periodična funkcija se razlaže na svoje spektralne komponente radi jednostavnije analize. Na taj način, data funkcija je razdvojena na sumu sinusoidnih funkcija, pomnoženih koeficijentima koji predstavljaju njenu amplitudu. Upravo ti koeficijenti određuju u kolikoj meri je prisutan neki harmonik u signalu.



Slika 15: Signal zvuka sa dve sinusne funkcije u vremenskom domenu (levo) i frekventnom domenu (desno)

Brza Furijeova transformacija (često se označava i kao FFT - Fast Fourier Transformation) jeste algoritam za “brzo” izračunavanje vrednosti diskretne Furijeove transformacije. Broj tačaka nad kojim se vrši transformacija predstavlja stepen broja 2, što govori da se radi o tzv. *radex2* algoritmu. Što je veći broj tačaka nad kojim se primenjuje algoritam, to je veća preciznost, odnosno razlaganje na veći broj osnovnih sinusnih funkcija.

Programska podrška za automatsko testiranje poseduje dva načina testiranja zvuka:

1. Prepoznavanje oblika audio zapisa na osnovu prethodno opisanog algoritma za *brzu Furijeovu transformaciju*. Dva audio zapisa se porede na taj način, što se ispituje da li oni sadrže sinusne signale istih frekvencija tj. porede se vrednosti koeficijenata dobijenih pomoću *brze Furijeove transformacije*. Za potrebe testiranja, koristio se audio zapis koji predstavlja zbir dva sinusna signala različitih frekvencija. Radi provere poklapanja dva audio zapisa, posmatraju se dve maksimalne vrednosti koeficijenata kod oba signala. Audio zapisi su se podudarali ukoliko su redni brojevi maksimalnih koeficijenata bili isti.
2. Poređenje jačine dva audio zapisa izračunavanjem energije signala korišćenjem koeficijenata dobijenih pomoću *brze Furijeove transformacije*. Energija signala predstavlja zbir kvadrata vrednosti koeficijenata *brze Furijeove transformacije*. Ukoliko je razlika energija dva signala manja od zadatog praga, smatra se da su audio zapisi iste jačine.

4.4.1 Realizacija algoritma za poređenje zvuka

Kao što je već izneto, algoritam za poređenje zvuka je realizovan u okviru klase *CAudioVideoProcessing*, i omogućava poređenje oblika dva audio zapisa kao i poređenje njihovih jačina. Za te potrebe, razvijene su sledeće metode:

AudioShapeMatching (short* oBuff, short* tBuff) AudioLoudnessMatching(short* oBuff, short* tBuff)
--

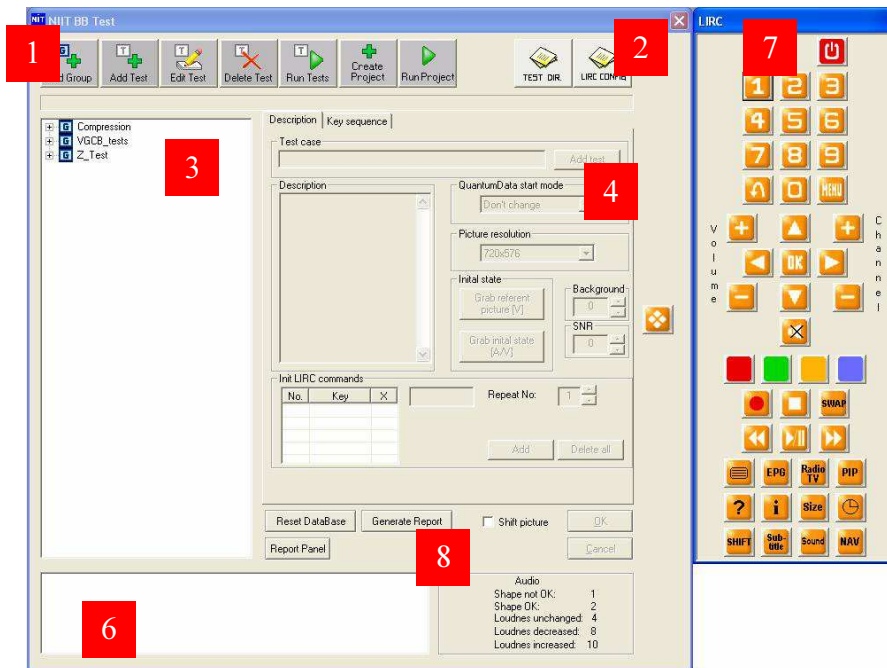
Obe metode sadrže kao ulazne argumente adrese memorijskih zona, na kojima su smeštene vrednosti audio zapisa koji se porede. Audio zapis u memorijskim zonama može biti u stereo obliku (2 kanala), s tim da pre primene *brze Furijeove transformacije* on mora svesti na mono signal (1 kanal), jer se transformacija primenjena u ovom slučaju koristi samo nad takvim signalom. To se postiže izbacivanjem svakog drugog odbirka audio zapisa iz memorijskih zona. Takođe, svaki odbirak audio zapisa mora biti dužine 16 bita, kako bi algoritam funkcionisao na pravilan način. Trenutna verzija programske podrške primenjuje *brzu Furijeovu transformaciju* koristeći 1024 odbirka audio zapisa kao ulazni niz podataka, a kao rezultat dobija se 512 vrednosti Furijeovih koeficijenata, koji se analiziraju na prethodno opisan način.

4.5 Rukovanje programskom podrškom za automatsko testiranje

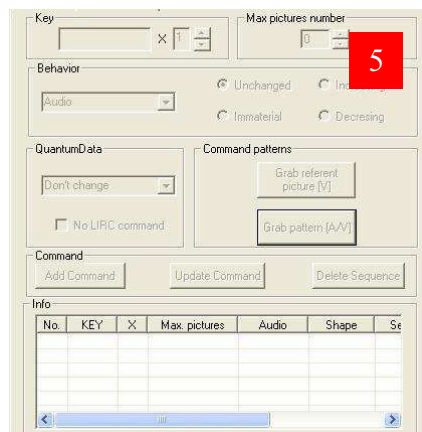
4.5.1 Izgled programske podrške za automatsko testiranje

Programska podrška za automatsko testiranje (slika 16 i 17) sadrži sledeće delove:

1. Komandne tastere
2. Konfiguracione tastere
3. Listu postojećih testova
4. Opis selektovanog testa
5. Sekvencu komandi selektovanog testa
6. Izveštaj o toku izvršavanja testa
7. Daljinski upravljač
8. Generisanje izveštaja testova



Slika 16: Izgled AutomatedVIL aplikacije sa aktiviranim *Description* prozorom



Slika 17: *Key Sequence* prozor

1. Komandni tasteri - namenjeni za manipulaciju sa testovima. Preko tih tastera omogućeno je korišćenje svih glavnih funkcija koje pruža programska podrška AutomatedVIL. Korišćenjem tastera implementirane su sledeće radnje koje se mogu primenjivati nad testovima:
 - Kreiranje grupe testova (Add group) – namenjeno za kreiranje grupe kojoj će pripadati testovi vezani za testiranje jedne srodne oblasti testova (testiranje menija, teleteksta, NTSC video formata, DVI video formata, zvuka itd.). Pritiskom na taster prikazuje se dijalog u kome je potrebno uneti naziv grupe. Potvrdom izbora imena, kreira se grupa i prikazuje se u listi testova sa odgovarajućom ikonicom pored sebe kao indentifikacijom grupe. Treba naglasiti da je grupu moguće kreirati samo u okviru neke druge grupe (podgrupe) koja u sebi ne sadrži testove.

- Kreiranje testa (Add test) – namenjeno za kreiranje pojedinačnih testova. Da bi se testovi mogli izvršavati neophodno ih je prvo kreirati. Pritiskom na taster omogućava se unošenje imena testa (Test Case) u odgovarajuće polje *Description* prozora. Sve do potvrde imena testa sva ostala polja u vezi sa kreiranjem testa u *Description* i *Key Sequence* prozoru su onesposobljena kako bi se sprečilo unošenje detalja testa a da pre toga nije navedeno ime testa koje predstavlja njegov jedinstveni identifikator. Test se kreira u okviru grupe koja je selektovana u glavnom dijalogu aplikacije. Način kreiranja testa, kao i značenje pojedinačnih polja u *Description* i *Key Sequence* prozoru biće kasnije detaljnije objašnjeni.
 - Uređivanje testa (Edit test) – namenjeno je uređivanju (promeni) već postojećih testova u slučaju da korisnik želi promeniti neki detalj u sadržaju odgovarajućeg testa. Jedina osobina testa koja se ne može menjati je ime testa, jer kako je naglašeno, ono predstavlja njegov identifikator.
 - Brisanje testa (Delete test) – namenjeno je brisanju postojećeg testa. Sa ciljem smanjivanja broja komandnih tastera, putem ovog tastera omogućeno je i brisanje grupe testova. U slučaju da selektovana grupa poseduje neke podgrupe sa testovima, programska podrška će obrisati i njihov sadržaj. Pre samog brisanja korisnik se upozorava o brisanju testa vizuelno i zvučno.
 - Izvršavanje testa (Run test) – namenjeno je izvršavanju selektovanog testa u glavnom prozoru aplikacije. Pored kreiranja testa ono predstavlja glavnu radnju koju pruža programska podrška. Na ovaj način moguće je pokrenuti samo jedan test iz liste postojećih testova. Pritiskom na ovaj taster pokreće se izvršavanje testa i pri tome se pojavljuje prozor na kome se prikazuje tok izvršavanja testa što će kasnije biti detaljnije objašnjeno. Na kraju izvršavanja testa, korisniku se saopštava rezultat o uspešnosti testa.
 - Kreiranje projekta (Create project) – namenjeno je mogućnosti izvršavanja kolekcije (grupe) testova, pošto je tasterom za izvršavanje testa (Run test) moguće izvršiti samo jedan pojedinačni test. Pritiskom na taster prikazuje se prozor sa tabelom koju korisnik popunjava testovima koji će pripadati datom projektu. Slično kao što i test ima svoje ime koje ga identifikuje, tako i svaki projekat poseduje odgovarajuće ime.
 - Izvršavanje projekta (Run project) – namenjeno je izvršavanju prethodno kreiranog projekta (grupe testova). Pritiskom na taster prvo se prikazuje prozor, gde se očekuje od korisnika da unese svoje ime i verziju programske podrške koja je prisutna na testiranom uređaju (Build file), i nakon toga se pristupa izvršavanju grupe testova, redosledom koji je napravljen prilikom kreiranja projekta. Slično kao i prilikom izvršavanja pojedinačnog testa, na ekranu se prikazuje tok izvršavanja projekta, što će kasnije biti detaljnije objašnjeno.
2. Konfiguracioni tasteri – koriste se za podešavanje putanja direktorijuma i datoteka značajnih za rad aplikacije. Da ne bi došlo do gubitka informacija prilikom zatvaranja aplikacije, kao rešenje koristi se snimanje tih podataka u *Registry* bazu računara.
- Putanja direktorijuma koji sadrži testove (Test dir) – namenjeno za smeštanje putanje direktorijuma u koji će se smeštati kreirani testovi. Pritiskom na taster, očekuje se od korisnika da odabere direktorijum i nakon potvrde ta putanja se smešta u *Registry* bazu računara i u glavnom dijalogu

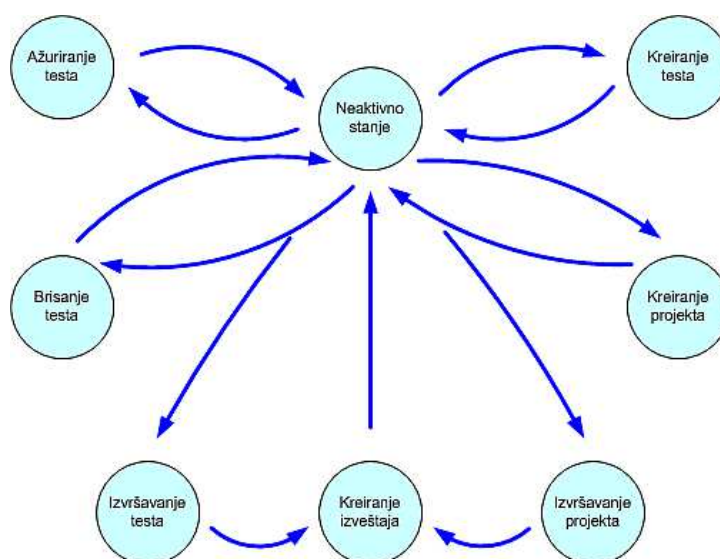
- aplikacije prikazuju se testovi koji se nalaze u prethodno zadatom direktorijumu.
- Putanja konfiguracione datoteke daljinskog upravljača (RC config) – namenjeno za smeštanje putanje konfiguracione datoteke daljinskog upravljača. Datoteka poseduje .cfg ekstenziju i u sebi sadrži kodove komandi koje podržava ta verzija daljinskog upravljača. Na sličan način kao i kod direktorijuma za smeštanje testova, pritiskom na taster prikazuje se prozor u kome se očekuje od korisnika da odredi putanju do te datoteke, koja se takođe smešta u *Registry* bazu računara.
3. Lista postojećih testova – polje u kom su prikazani svi postojeći testovi koji se nalaze u trenutno selektovanom direktorijumu. Testovi su razvrstani po grupama, pri čemu testovi imaju oznaku **T**, a grupe testova oznaku **G**. Selekcijom jednog od testa iz liste, *Description* i *Key Sequence* prozori se popunjavaju odgovarajućim podacima o tom selektovanom testu.
4. Opis selektovanog testa – prozor (*Description*) sadrži sve generalne karakteristike koje opisuju jedan test. Ovaj prozor ima dvostruku ulogu, u zavisnosti od toga u kom režimu se koristi. U slučaju selekcije testova iz glavnog prozora aplikacije, polja u tom prozoru se popunjavaju odgovarajućim podacima selektovanog testa, što znači da se on tada koristi kao prozor za prikaz informacija. Kada je programska podrška u režimu kreiranja testa, odgovarajuća polja tog prozora popunjava korisnik sa željenim podacima, što znači da se tada taj prozor koristi za prihvatanje informacija. Prozor sadrži sledeće celine:
- Polje za ime testa (*Test case*)
 - Polje tekstualnog opisa testa (*Description*)
 - Lista QuantumData video formata (*QuantumData start mode*) – predstavlja listu video formata koju podržava QuantumData generator slike. Na samom početku izvršavanja testa, odabrani video format (signal) se postavi na izlazu generatora.
 - Tasteri za snimanje slike i zvuka početnog stanja (*Initial state*) – na ovaj način se preuzima slika i zvuk sa BBT ACQUISITION BOARD platforme u fazi kreiranja testa. Ta slika i zvuk predstavljaju referentne početne uslove koje dati test mora da zadovolji, kako bi mogao da nastavi sa daljim testiranjem. Slika i zvuk se smeštaju u direktorijum testa sa imenom datoteke INIT_V_PAT.raw i INIT_A_PAT.pcm respektivno.
 - Polje za vrednost odnosa signal/šum za prepoznavanje slike (*SNR*) – predstavlja polje koje se popunjava prilikom kreiranja testa, i služi za određivanje granice koju treba da zadovolji stepen poklapanja slike u algoritmu za prepoznavanje slike, kako bi se moglo smatrati da se dve slike poklapaju tj. da su istovetne. Empirijskim putem je određeno da se za rezoluciju slike 1366x768 ta vrednost kreće oko 100.
 - Lista komandi daljinskog upravljača (*Init RC commands*) – predstavlja niz komandi daljinskog upravljača koji se izdaju na samom početku izvršavanja testa. To je neophodno uraditi iz razloga, što se testirana platforma (VGCA EVA) prethodnim testom dovela u neko stanje koje se razlikuje od početnih uslova trenutno aktivnog testa. Da bi se ona dovela u neko zamišljeno početno stanje, neophodno je izdati niz komandi daljinskog upravljača, koje će upravo to uraditi. Moguće je višestruko izdavanje jedne te iste komande.

5. Sekvenca komandi selektovanog testa – prozor (*Key Sequence*) sadrži listu pojedinačnih koraka (komandi) koji se izdaju tokom izvršavanja testa. Isto kao i *Description* prozor, ovaj prozor ima dvostruku ulogu, u zavisnosti od toga u kom režimu se koristi. U slučaju selekcije testova iz glavnog prozora aplikacije, polja u tom prozoru se popunjavaju odgovarajućim podacima selektovanog testa, što znači da se on tada koristi kao prozor za prikaz informacija. Kada je programska podrška u režimu kreiranja testa, u tom slučaju, odgovarajuća polja tog prozora popunjava korisnik sa željenim podacima, što znači da se tada taj prozor koristi za prihvatanje informacija. Prozor sadrži sledeće celine:
- Polje za komandu daljinskog upravljača (*Key*) – predstavlja komandu koja se šalje. Komanda se bira tako što se na prozoru daljinskog upravljača bira željena komanda. U slučaju da je potrebno ponoviti istu komandu više puta, na odgovarajućem polju pored naziva komande potrebno je uneti broj ponavljanja.
 - Polje za maksimalan broj slika (*Max pictures number*) – predstavlja polje koje sadrži maksimalan broj slika koje će snimiti platforma za snimanje sve dok se ne proglasi neuspešnost testa. Zbog relativno spore mrežne komunikacije, za prenos slike potrebno je 2 do 3 sekunde u zavisnosti od mreže. U tom broju slika očekuje se slika koja zadovoljava kriterijume prolaska pojedinačnog koraka. Treba naglasiti da se ne snimaju uvek sve slike, već čim se snimi slika koja zadovoljava kriterijume trenutnog koraka, prelazi se na sledeći korak u testu. U slučaju da se nije pojavila slika koja zadovoljava kriterijume koraka, a snimljen je maksimalan broj slika, test je neuspešan. U slučaju prelaska prenosa slike za mrežne komunikacije na USB 2.0 komunikaciju, koja je mnogo brža, postoji mogućnost promene značenja ovog polja na vremenski interval u kom se mora pojaviti slika koja zadovoljava postavljene kriterijume. Taj vremenski interval predstavlja vreme odziva testiranog uređaja na zadatu komandu.
 - Lista očekivanih promena slike i zvuka (*Behavior*) – predstavlja niz kriterijuma koji moraju biti zadovoljeni, kako bi trenutni korak u testu ispunio očekivani odziv na komandu daljinskog upravljača ili promenu slike usled promene video formata na QuantumData generatoru. Ponuđena su tri kriterijuma koja se ispituju:
 - Video – ispituju se promene na celoj slici tj. upoređuju se sve tačke na slici rezolucije 1366x768. Po ovom kriterijumu, mogući su sledeći rezultati koji zadovoljavaju trenutni korak u testu:
 - Detektovana je promena u slici (*Changed*)
 - Nije detektovana promena u slici (*Unchanged*)
 - Kriterijum se ne uzima u obzir (*Immaterial*)
 - Service – ispituju se promene slike samo u njenim uglovima, što znači, da se upoređuju samo delovi slike. To su 4 kvadrata svaki u jednom uglu veličine 100x100 piksela. Koristi se u slučajevima kada je potrebno detektovati promene u pozadini slike, tj. testira se promena servisa. Na taj način se ne uzima u obzir trenutno prikazani meni (OSD) u sredini slike, već samo prikazana slika. Po ovom kriterijumu, mogući su sledeći rezultati koji zadovoljavaju trenutni korak u testu:

- Detektovana je promena (*Changed*)
- Nije detektovana promena (*Unchanged*)
- Kriterijum se ne uzima u obzir (*Immaterial*)
- Audio – ispituju se promene na zvuku. Po ovom kriterijumu, mogući su sledeći rezultati koji zadovoljavaju trenutni korak u testu:
 - Detektovana je promena u zvuku, pri čemu je oblik signala nepromenjen a jačina zvuka se povećala (*Increased*)
 - Detektovana je promena u zvuku, pri čemu je oblik signala nepromenjen a jačina zvuka se smanjila (*Decreased*)
 - Nije detektovana promena ni u obliku zvuka, ni u njegovoj jačini (*Unchanged*)
 - Kriterijum se ne uzima u obzir (*Immaterial*)
- Lista video formata na QuantumData generatoru (*QuantumData*) – predstavlja video format koji se nalazi na izlazu generatora tokom trenutno testiranog koraka, što znači da je moguće menjati video signal u svakom koraku jednog testa. Postoji opcija zadržavanja video formata koji je bio prisutan za vreme prethodnog koraka, ukoliko postoji potreba za time.
- Tasteri za snimanje referentne slike i referentnog zvuka trenutnog koraka (*Command patterns*) – na ovaj način se preuzima slika i zvuk sa BBT ACQUISITION BOARD platforme u fazi kreiranja testa. Ta slika i zvuk predstavljaju referentne uslove koje trenutni korak mora da zadovolji, kako bi prešao na testiranje sledećeg koraka u testu. Slika i zvuk se smeštaju u direktorijum testa sa imenom datoteke CMD_V_PATxx.raw i CMD_A_PATxx.pcm respektivno, pri čemu xx predstavlja redni broj koraka u testu.
- Tasteri za rad sa korakom testa (*Command*) – korišćenjem ovih tastera vrši se ažuriranje liste koraka trenutno kreiranog testa. Postoje tri tastera:
 - Dodaj korak (*Add command*) – dodaje se konfigurisan korak u listu koraka testa
 - Ažuriraj korak (*Update command*) – vrši ažuriranje (promenu) već postojećeg koraka u listi koraka
 - Obriši korake (*Delete commands*) - brišu se svi koraci u listi koraka trenutno kreiranog testa
- Lista koraka u testu (*Info*) – predstavlja listu koraka (komandi) koji sadrži jedan test. Prilikom selektovanja testa iz liste postojećih testova, lista koraka se popunjava sa detaljima o svakom koraku testa koji je trenutno selektovan. U slučaju kreiranja testa, listu popunjava korisnik konfigurisanim koracima, korišćenjem tastera za rad sa koracima testa (*Command*) opisanim u prethodnom delu. Lista poseduje nekoliko kolona koje sadrže različite podatke o svakom koraku kao što su:
 - Redni broj koraka u testu (*No*)
 - Komanda daljinskog upravljača (*Key*)
 - Broj komandi koji se šalje (*X*)
 - Maksimalan broj snimljenih slika (*Max pictures*)
 - Kriterijum promena na zvuku (*Audio*)
 - Kriterijum promena na celoj slici (*Shape*)
 - Kriterijum promena na delovima slike (*Service*)

6. Izveštaj o toku izvršavanja testa (*Report panel*) – predstavlja kratak izveštaj o toku izvršavanja trenutno aktivnog testa. Sadrži niz informacija kao što ime testa, vreme početka izvršavanja testa, vrednosti stepena poklapanja slike u svakom koraku i zvuka itd. Pritiskom na taster *Report panel* dati prozor se pojavljuje/uklanja sa ekrana.
7. Daljinski upravljač (*RC*) – predstavlja prozor sa 45 tastera, pri čemu svaki taster reprezentuje jednu komandu. Pritiskom na svaki taster prilikom kreiranja testa, testiranom uređaju (VGCA EVA) se šalje komanda, i u polju za komandu daljinskog upravljača (*Key*) prikazuje se ime pritisnutog tastera. Kako bi se obezbedila druga funkcija svakog tastera (2nd function), koristi se taster *Shift* koji promeni boju kada je aktivan. Iz razloga što je daljinski upravljač realizovan u obliku posebnog prozora, bilo je neophodno rešiti način kako glavnom prozoru saopštiti koja je komanda izdata od strane prozora daljinskog upravljača. To je rešeno korišćenjem poruka (eng. messages). Prozor daljinskog upravljača šalje poruku tipa *UM_UPDATEEDITKEY* ka glavnom prozoru sa argumentom koji predstavlja ime komande. Glavni prozor po prijemu te poruke, uzima njen argument i prepoznaje pritisnut taster.
8. Generisanje izveštaja testova – predstavlja deo programske podrške koji služi za rad sa izveštajima rezultata testova. Prilikom izvršavanja svakog testa, rezultat testa zajedno sa njegovim opisom smešta se u zajedničku bazu podataka. Ukoliko je potrebno napraviti izveštaj testiranja u HTML obliku tj. u obliku Internet stranice, to se može uraditi korišćenjem *Generate Report* tastera, pri čemu je samo potrebno navesti putanju do datoteke *Result.dbf* koja sadrži bazu podataka. Za brisanje svih podataka u bazi rezultata testova koristi se *Reset Database* taster.

Programska podrška za automatsko testiranje u toku rada se može nalaziti u jednom od nekoliko nabrojanih stanja prikazanih na slici 18:

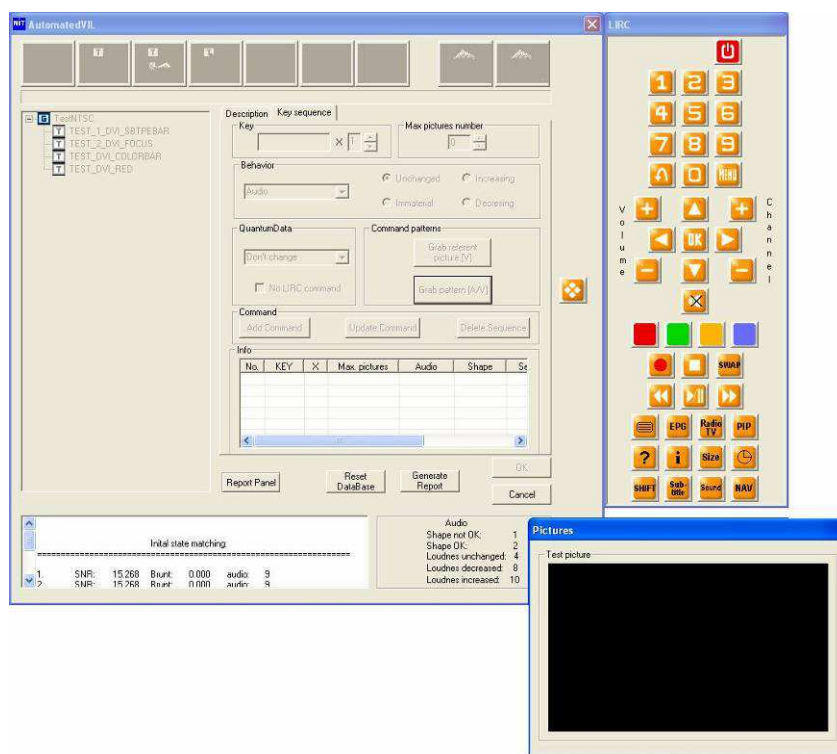


Slika 18: Stanja programske podrške za automatsko testiranje

4.5.2 Opis postupka za kreiranja testova

Procedura dodavanja testova se sastoji iz nekoliko koraka, koje redom treba preći da bi se postiglo očekivano ponašanje testa tokom izvršavanja:

1. Selektovati grupu u glavnom prozoru aplikacije u koju se želi dodati test. U slučaju da grupa nije kreirana, tasterom *Add Group* iz komandnih tastera kreirati grupu.
2. Pritisnuti *Add test* taster za dodavanje testa. Prikazuje se prozor na kome se vidi trenutna slika koja je prisutna na video izlazu testiranog uređaja (VGCA EVA), i polje za unos imena testa (*Test case*) postaje dostupno.
3. Upisati ime testa u odgovarajuće polje (*Test case*) i pritisnuti taster *Add test*.
4. Uneti opis testa u polje *Description* i snimiti referentni zvuk i slike početnog stanja korišćenjem tastera *Grab referent picture* i *Grab referent sound*.
5. Dodati niz komandi daljinskog upravljača u *Init RC commands* sekciji, kako bi se obezbedilo početno stanje.
6. U *Key Sequence* prozoru dodati niz koraka testa, koji se sastoje od izdavanja komandi daljinskog upravljača i promene video formata (signala) na QuantumData generatoru.
7. Pritisnuti taster OK radi potvrde dodavanja testa.



Slika 19: Izgled programske podrške za testiranje tokom kreiranja testa

U koraku 2 postupka za kreiranje testa, istaknuto je da se tokom faze kreiranja testa konstantno prikazuje poseban prozor sa trenutnom slikom koju prikazuje testirani uređaj. Prozor za prikaz slike je isti kao i prozor koji se koristi u toku izvršavanja testova. U fazi izvršavanja testova prozor sadrži referentnu sliku i sliku koju daje testirani uređaj na izlazu

dok u fazi kreiranja testova, prozor sadrži samo sliku koja je prisutna na video izlazu testiranog uređaja. Prikaz slike je realizovan u okviru niti, koja se pokreće prilikom pokretanja faze kreiranja testa. Nit sadrži petlju, koja izdaje zahteve za preuzimanje slike sa platforme za snimanje i tako preuzet sadržaj prikazuje u prozoru za prikaz slike. Slika se po preuzimanju sa platforme, smešta u odgovarajuću memorijsku zonu rezervisanu za tu namenu, a prilikom prikaza preslikava u prozor namenjen za prikaz. Zbog relativno spore komunikacije između PC računara i platforme za snimanje (za prenos jedne slike potrebno je od 2 do 3 sekundi), između dva uzastopna zahteva za slikom, nepostoji nikakvo uspaljivanje niti, jer bi to još dodatno usporilo osvežavanje prikaza slike koje prikazuje testirani uređaj. Uslov za izlazak iz petlje, jeste kraj faze kreiranja testa, što se postiže pritiskom na *Ok* ili *Cancel* tastere na glavnom prozoru.

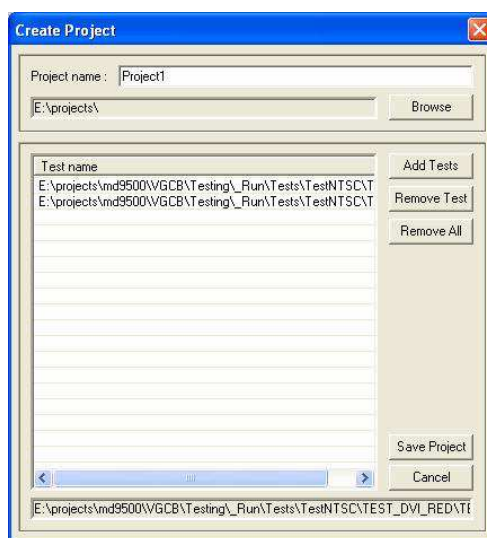
Zbog zahteva koji nameće funkcija za prikaz slike u prozoru, neophodno je uraditi konverziju slike u memoriji pre njenog prikaza. Ta konverzija se sastoji u promeni redosleda smeštanja komponenti (RGB) slike. Po prijemu slika u memoriji je u RGB redosledu dok zbog pravilnog prikaza u prozoru ona mora biti BGR redosledu, što se jednostavno postiže zamenom mesta R i B komponenti u celoj slici.

4.5.3 Opis postupka za ažuriranje testova

Ažuriranje testova predstavlja postupak koji je vrlo sličan postupku kreiranja testa. Jedina razlika je u tome što je potrebno promeniti samo one korake testa koje se žele promeniti, dok ostali koraci u testu će zadržati svoj prvobitne referentne slike, komande i uslove ispunjavanja odziva na komandu napravljene u fazi kreiranja testa.

4.5.4 Opis postupka za kreiranje projekta

Kao što je već rečeno, programska podrška za automatsko testiranje podržava rad sa projektima, što podrazumeva njihovo kreiranje i izvršavanje. Pritiskom na taster *Create Project* iz grupe komandnih tastera, prikazuje se prozor za kreiranje projekta (slika 20).



Slika 20: Prozor za kreiranje projekta

U polje *Project name* navodi se ime projekta i pomoću tastera *Browse* određuje se mesto gde će biti uskladišten na masovnoj memoriji radi kasnijeg korišćenja. Tasterom *Add Tests* dodaju se testovi u projekat, na taj način, što se prilikom selekcije testa, navodi glavna datoteka testa sa ekstenzijom *.tst* u kojoj su smešteni svi podaci o testu. Za brisanje pojedinačnih ili svih testova iz liste koriste se tasteri *Remove Test* i *Remove All* respektivno. Pritiskom na taster *Save Project*, projekat se snima u datoteku sa ekstenzijom *.bbp* i postupak kreiranja projekta je završen.

4.5.5 Izvršavanja testova i projekata u okviru programske podrške za automatsko testiranje

Postoje dva oblika izvršavanja testova, s tim da su ona međusobno vrlo slična. Razlika je u tome, da se kod jednog od njih zahteva veće učešće čoveka zbog mogućnosti da se na proizvoljan način biraju testovi, dok kod drugog načina korisnik nema toliku slobodu. Prvi način je pojedinačno izvršavanje testova, pri čemu korisnik iz liste ponuđenih testova bira test koji će izvršavati. Po završetku testa, korisnik ukoliko želi, prelazi na sledeći ponovnim biranjem iz liste testova. Ovaj način je pogodan u slučaju korišćenja samo pojedinačnih testova. Za potebe testiranja pomoću većeg broja testova, koristi se drugi način izvršavanja testova tzv. projekti. Kod projekata testovi se izvršavaju sekvencijalno jedan za drugim, bez potrebe za ikakvom intervencijom čoveka. Sve što korisnik treba da uradi je, samo da pokrene projekat, i kad se njegovo izvršavanje završi, korisnik treba da pogleda rezultate testova. Testovi se izvršavaju redosledom koji je naveden prilikom kreiranja projekta. Ukoliko se želi promeniti niz testova koje sadrži projekat ili redosled izvršavanja tih testova, potrebno je ažurirati sadržaj datoteke projekta. Za potrebe testiranja putem projekata, implementirana je dodatna opcija. Postoji mogućnost kružnog (cirkularnog) izvršavanja testova. Ukoliko je ova funkcija aktivna, nakon izvršenja zadnjeg testa iz projekta, ponovo se prelazi na izvršenje prvog, sve dok korisnik ne prekine izvršavanje projekta. Smisao uvođenja projekta je, kao što je već rečeno, smanjenje učešća čoveka prilikom izvršavanja testova, ali i mogućnost organizovanja testova u odgovarajuće celine. Na ovaj način testovi se grupišu u grupe, od kojih svaka testira samo jedan deo funkcija testiranog uređaja (zvuk, PIP (slika u slici), PAT (slika i tekst), meni, promena video signala itd.). Izvršavanje testova se odvija sledećim redosledom:

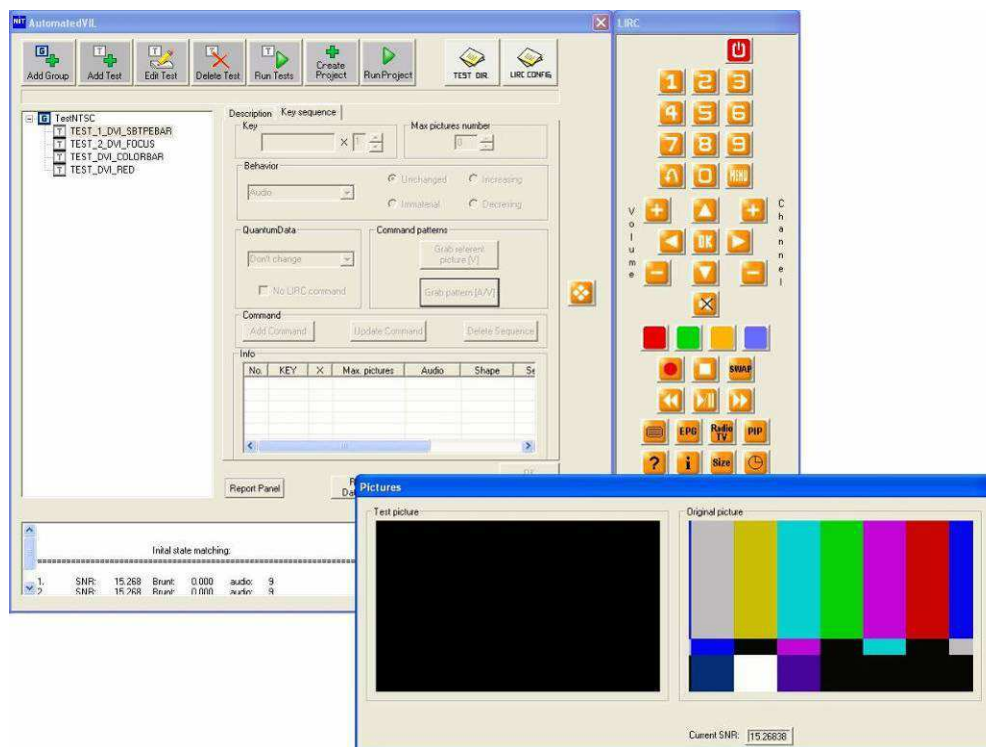
1. Programska podrška šalje komande početnog stanja daljinskom upravljaču i QuantumData generatoru kako bi obezbedila početne uslove za izvršavanje testa.
2. Pomoću platforme za snimanje slike (BBT ACQUISITION BOARD), snima se trenutni video izlaz na testiranom uređaju i prenosi se na PC računar
3. Na računaru se korišćenjem algoritma za poređenje slika, snimljena slika upoređuje sa slikom početnog stanja koja je snimljena u toku kreiranja testa. U slučaju poklapanja, prelazi se na izvršavanje pojedinačnih koraka u testu (tačka 4), a u slučaju nepoklapanja, test se završava kao neuspešan.
4. Izdaje se komanda daljinskog upravljača i ako postoji potreba, menja se signal na video generatoru. Preuzima se slika sa izlaza testiranog uređaja i upoređuje sa referentnom slikom. Ako je došlo do poklapanja slika, prelazi se na sledeći korak.

5. Prethodna tačka se ponavlja sve dok se ne izvrše svi koraci u testu.
6. U prvom koraku kod kog se javi nepoklapanje slike sa referentnom, test prestaje sa daljim izvršavanjem i proglašava se neuspešnim. U slučaju poklapanja slike kod svih koraka, test se smatra uspešnim.

Tokom izvršavanja testa, na ekranu u okviru posebnog prozora, može se pratiti tok izvršavanja (slika 21). Pokretanjem testa na ekranu se pojavljuje prozor, na kome se prikazuje slika koja je trenutno prisutna na video izlazu testiranog uređaja, i slika koja se očekuje – referentna slika (napravljena u toku kreiranja testa).

Prikaz slike koja je trenutno prisutna na testiranom uređaju, je realizovan na isti način kao i tokom faze kreiranja testa koji je opisan u prethodnom delu. Za razliku od slike koja predstavlja trenutni prikaz testiranog uređaja a čiji sadržaj je smešten u memorijskoj zoni PC računara, sadržaj referentne slike je prvobitno smešten u datoteci, pa zbog toga je neophodno prethodno njen sadržaj preneti u memorijsku zonu.

U toku razvoja programske podrške, uočila se pojava da testirani uređaj prilikom korišćenja istog video formata (signala) na ulazu, pomera sliku na izlazu za najviše 4 tačke po horizontali, što je u granicama tolerancije za testirani uređaj. Do pomeranja dolazi ako se u međuvremenu kao izvor, koriste različiti signali, pa pri povratku na prvobitni slika je pomerena. Pošto je primenjeni algoritam za prepoznavanje slike osetljiv na pomeraj, iako su slike istovetne, testovi će se neuspešno izvršavati. Kao rešenje, implementirana je opcija, da se slika pre poređenja sa referentnom, pomera po horizontali za odgovarajući broj tačaka. To se pokazalo kao jedno rešenje, sa nedostatkom da se usporava rad programske podrške za testiranje, jer pomeranje slike opterećuje rad procesora PC računara u velikoj meri.



Slika 21: Izgled programske podrške za testiranje tokom izvršavanja testa

4.5.6 Pregled rezultata testova programske podrške za automatsko testiranje

Pored procedure testiranja uređaja, veoma važan deo procesa testiranja predstavlja način i oblik prikazivanja, odnosno saopštavanja rezultata testa. To je bitno, kako radi lakše analize rezultata testova, tako i radi arhiviranja tj. praćenja istorije rezultata testova. Programska podrška AutomatedVIL generiše izveštaj rezultata testova u 3 oblika:

1. izveštaj u obliku tekstualne datoteke (txt format)
 2. izveštaj u obliku baze podatka (dbf format)
 3. izveštaj u obliku Internet stranice (html format)
- Izveštaj u obliku tekstualne datoteke – predstavlja običnu tekstualnu datoteku (slika 22), koja se može pregledati bilo koji alatom za čitanje ASCII tekstualnih datoteka. Ovaj oblik izveštaja se vezuje samo za jedan test tj. datoteka sa izveštajem sadrži rezultat samo jednog testa. U slučaju ponovnog izvršavanja testa, sadržaj datoteke sa izveštajem se briše i formira se novi izveštaj. Datoteka se smešta u direktorijum gde se nalazi test a njeno ime je isto kao i ime testa, s time da je ekstenzija *.log*. Izveštaj sadrži informacije kao što je vreme i datum početka testa, vrednosti rezultata poklapanja slike kako u početnom stanju tako i u svakom koraku testa, vreme i datum završetka testa sa njegovim rezultatom.

```

Listner - [E:\projects\md9500\WGCBVTesting_Run\Project1.txt]
File Edit Options Help
*****
Test MENUQUITRACE started at: Sat Jun 17 19:12:09 2006
*****
TESTING INITIAL STATE:
SNR: 150.000000
INITIAL VIDEO MATCHING PASSED at Sat Jun 17 19:12:13 2006
TESTING STATE AFTER SENDING COMMAND:
SNR: 40.097340
SNR: 150.000000
TESTING STATE AFTER SENDING COMMAND:
SNR: 150.000000
TESTING STATE AFTER SENDING COMMAND:
SNR: 150.000000
TESTING STATE AFTER SENDING COMMAND:
SNR: 150.000000
TESTING STATE AFTER SENDING COMMAND:
SNR: 150.000000
TESTING STATE AFTER SENDING COMMAND:
SNR: 150.000000
TEST PASSED at Sat Jun 17 19:12:40 2006
*****
Test SOURCESMENUBROWSE started at: Sat Jun 17 19:12:40 2006
*****
TESTING INITIAL STATE:
SNR: 150.000000
INITIAL VIDEO MATCHING PASSED at Sat Jun 17 19:12:44 2006

```

Slika 22: Rezultat testa u tekstualnom obliku

- Izveštaj u obliku baze podataka – predstavlja kumulativni oblik izveštaja, u koji se stalno dodaju rezultati različitih testova. Korisnik može eksplicitno da obriše sadržaj baze podataka korišćenjem *Reset database* tastera iz glavnog prozora aplikacije. Za razliku od tekstualnog oblika izveštaja, baza podataka sadrži samo konačan rezultat testa, dok se međurezultati testa zbog kumulativne prirode izveštaja ne stavljaju u bazu podataka. Pored rezultata testa, izveštaj sadrži još i ime korisnika koji je pokrenuo testiranje, verziju programske podrške prisutne na testiranom uređaju, ime testa

zajedno sa njegovim kratkim opisom, vreme i datum pokretanja testa. Datoteka baze podataka se smešta u direktorijum gde se nalazi AutomatedVIL programska podrška.

- Izveštaj u obliku Internet stranice – predstavlja oblik izveštaja koji je po sadržaju vrlo sličan prethodnom obliku s tom razlikom da je zapisan u obliku Internet stranice (slika 23). Glavni razlog uvođenja ovog oblika je bolji pregled rezultata testova, kao i mogućnost prezentovanja izveštaja putem Interneta. Izveštaj poseduje i jednostavan oblik statistike rezultata prolaznosti testova.

NIIT BB Testing Results							
No	Date and time	Tester	Test case	Build file	SW check-out time	Result	Comment
1	17.06.2006 19:12	Vukob	MENUGUITRACE	build_123	-	PASS	Description: Control the operation of the GUI on the display. Starting conditions: Any channel is selected, none of menu functions is active. Testing procedure: Press the Menu key to start the Menu window. Run over the menu's pages with the right D.
2	17.06.2006 19:13	Vukob	SOURCEMENUBROWSE	build_123	-	FAIL	Description: Open and close sources menu. Starting conditions: Any channel is selected and no menu is active. Testing procedure: Press 0 to start sources menu. Then press 12 x down. Pass-Fail criteria: Source menu is started and all signal source menu.
3	17.06.2006 19:13	Vukob	SOURCEMENUONOFF	build_123	-	INCO	Description: Open and close sources menu. Starting conditions: Any channel is selected and no menu is active. Testing procedure: Press 0 to start sources menu. Then press 0. Press 0 once again and then back. Pass-Fail criteria: Source menu can be open.

Tested cases:	3	
Passed:	1	33%
Failed:	1	33%
Inconclusive:	1	33%

Slika 23 : Rezultat testa u obliku Internet stranice

5. Testiranje programske podrške za automatsko testiranje

Na polju alata za testiranje, postavlja se pitanje na koji način testirati opremu koja je upravo namenjena za tu svrhu. Problem je u tome, što je vrlo teško odrediti skup ispitivanja, koja bi se primenila nad sistemom za testiranje, i na osnovu njih sa sigurnošću odredila pouzdanost takvog sistema u radu. Važnost pouzdanosti sistema za testiranje utoliko je veća, zbog činjenice da se ne sme sumnjati u pouzdanost rada takvog sistema. U suprotnom, prilikom neuspeha testiranja nekog uređaja moglo bi se postaviti pitanje šta u tom testiranju nije pouzdano, uređaj koji se testira, ili sam sistem za testiranje. Stvar je još gora u slučaju da zbog nepouzdanosti rada sistema za testiranje, sistem oceni da je uređaj pouzdan a on to u stvari nije.

Kao način ispitivanja ovakvog sistema, na osnovu kog bi se odredila njegova pouzdanost u radu, u ovom diplomskom - master radu korišćena su sledeća dva oblika testiranja programske podrške za automatsko testiranje:

- Programska podrška je data na korišćenje korisnicima, koji uopšte nisu imali nikakvog učešća u njenom razvoju i zbog toga nisu bili upoznati sa njenim radom. Iz tog razloga oni su koristili aplikaciju na način koji se razlikuje od načina koji se koristio prilikom razvoja, a koji podrazumeva predviđeni redosled izvršavanja koraka pri korišćenju programske podrške. Time su izazvali neke neočekivane radnje na koje se nije toliko računalo. Na te neočekivane radnje, programska podrška treba pravilno da odreaguje i upozori korisnika o nepravilnom korišćenju. Na ovaj način otkriven je izvestan broj nedostataka ali istovremeno i niz sugestija za poboljšanjem rada aplikacije.
- U cilju određivanja pouzdanosti rada sistema za testiranje, ceo sistem je više puta korišćen u toku dužih vremenskih perioda, ne kraćih od 10 sati. Na taj način se otkrivaju greške koje se ne javljaju prilikom testiranja u nekom relativno kratkom periodu. U tim slučajevima korišćeni su jednostavni testovi napravljeni za tu namenu, koji su se u tom periodu konstantno kružno izvršavali, što znači da se po završetku poslednjeg testa ponavljao prvi. Proces se izvršavao sve dok sistem od strane korisnika pritiskom na odgovarajući taster nije prekinut sa radom.

Takođe, jedan od oblika testiranja je i testiranje tokom razvoja aplikacije, odnosno posmatranje različitih delova sistema da bi se utvrdilo da li se oni ponašaju u skladu sa očekivanim zahtevima.

6. Poređenje sa drugim realizacijama sistema za automatsko testiranje TV prijemnika

Na tržištu i u eksploataciji postoji relativno širok spektar sistema za testiranje. Svaki od njih testira uređaj na svojstven način, u zavisnosti od aspekta testiranja. Sa ciljem da se karakteristike realizovanog sistema što bolje iskažu, izvršeno je poređenje, koje se ogleda u upoređivanju karakteristika realizovanog sistema za automatsko testiranje u okviru ovog diplomskog – master rada, sa sličnim sistemima koji postoje na tržištu.

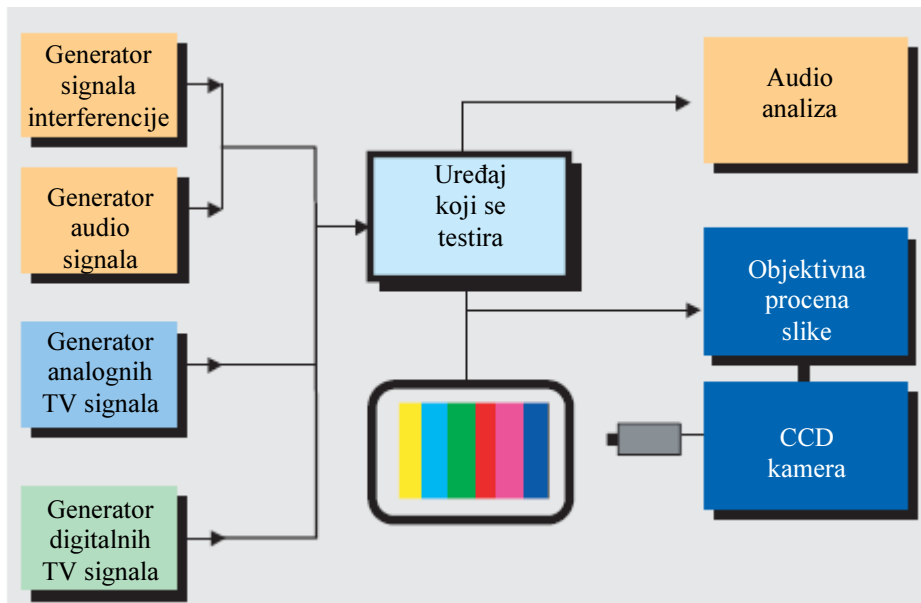
Iz skupa sistema za testiranje, kao sistem za poređenje uzet je *EMS Test System Audio and Video R&S TS9980 + TV-Mon* kompanije *Rohde&Schwarz*. Iako namena testiranja ovakvog sistema se razlikuje od namene sistema realizovanog u ovom radu, glavni razlog za upoređivanje tih sistema je što i jedan i drugi predstavljaju automatske sisteme za testiranje, čiji se rad tj. procena uspešnosti testa, bazira na poređenju referentne i testirane slike.



Slika 24: Rohde&Schwarz TS9980 test sistem

Test sistem *R&S TS9980* (slika 24) je namenjen za automatsko merenje elektromagnetske osetljivosti multimedijalnih prijemnika prema EN 55020 i CISPR 20 standardu, koji propisuju nivo elektromagnetske otpornosti za različitu opremu namenjenu prijemu signala kao što su TV prijemnici i njihove prateće opreme. Ovim sistemom pokrivena su sledeća merenja:

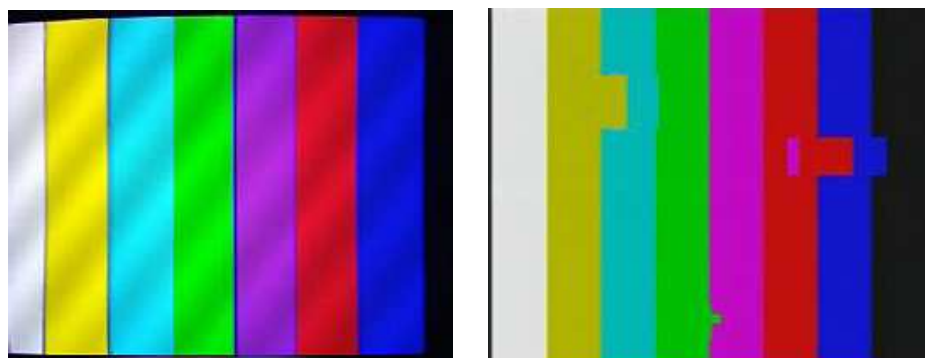
- Otpornost na interferenciju ulaznog signala
- Otpornost na RFI napone i struje
- Otpornost na emitovanu interferenciju
- Efikasnost zaštite testirane opreme



Slika 25: Blok dijagram *Rohde&Schwarz TS9980* test sistema

Princip rada sistema se zasniva na proceni kvaliteta slike, koju testirani uređaj emituje tokom rada pod uticajem interferentnih signala. Procena kvaliteta slike se zasniva na poređenju sa referentnom slikom. Pre početka testiranja, snimi se više referentnih slika (ne manje od 5), na osnovu kojih algoritam za poređenje određuje granicu maksimalnog izobličenja slike bez uticaja interferentnih signala. Nakon određivanja granice, može se pristupiti testiranju. Testirani uređaj se izlaže različitim signalima interferencije, koji se razlikuju kako po frekvenciji tako i po jačini. U takvim okolnostima snimaju se test slike, na osnovu kojih se određuje izobličenje u slici. U slučaju da je vrednost izobličenja u toku testiranja veća od vrednosti određene kao granice, na slici se primećuju vidne promene.

Algoritam za poređenje detektuje različite vrste izobličenja na slici koju emituje testirani uređaj na čiji ulaz se dovodi analogni ili digitalni video signala (slika 26).



Slika 26: Slika sa uticajem interferencije na analogni (levo) i digitalni video signal (desno)

U slučaju uticaja signala interferencije na analogni signal, dolazi do promena u slici kao što je valovitost slike, gubitak osvetljaja i boja, gubitak sinhronizacije, dok kod digitalnih signala prateći efekti su efekat blokova, zamrznuta – nepokretna slika, crn ekran itd. Za slučaj da testirani uređaj poseduje ekran (TV prijemnik), slika se preuzima pomoću visoko kvalitetne CCD kamere. RGB signal sa kamere se snima pomoću posebne fizičke

arhitekture za tu namenu (eng. RGB frame grabber), i prosleđuje se jedinici za poređenje slika. Ukoliko testirani uređaj ne poseduje ekran (DVD, VCR, STB), video signal se direktno dovodi na jedinicu za snimanje slike.

Postavlja se pitanje koja je oblast primene ovakvog sistema? Porastom komunikacija putem satelitskih, kablovskih i zemaljskih veza, i sve većim rastom broja korišćenih frekvencija unutar istog frekventnog opsega, može doći do smanjenja kvaliteta prijema signala. Upravo iz tog razloga neophodno je svaki uređaj pre početka serijske proizvodnje podvrgnuti testiranju na imunost prema spoljašnjim smetnjama. Pomoću obimnih tzv. EMS testova, potvrđuje se sposobnost kvalitetnog prijema signala čak i u najnepovoljnijim uslovima. Pošto izvođenje takvih testova je krajnje kompleksno i zahteva veliki broj pojedinačnih merenja, sve je to objedinjeno u jedan automatski sistem za testiranje – *Rohde&Schwarz TS9980*. Za razliku od *TS9980* sistema, oblast primene automatskog sistema za testiranje realizovanog u okviru ovog diplomskog – master rada je funkcionalno testiranje. Znači testira se ispravnost funkcije koja je implementirana na testiranom uređaju. Karakteristika oba sistema je da se mogu direktno integrisati u razvoj konkretnog uređaja, skraćujući vreme potrebno za njegovo iznošenje na tržište.

7. Zaključak

Automatsko testiranje predstavlja brz i efikasan način testiranja, koji isključuje učešće ljudi tokom izvršavanja testova. Prednost testiranja nekog uređaja kada se on posmatra kao crna kutija je u tome što nije potrebno poznavati njegovu implementaciju. Sama programska podrška za testiranje zna samo ispravnu reakciju testiranog uređaja na odgovarajuću komandu. Osoba koja upravlja testovima ne mora posedovati detaljno znanje o sistemu koji se testira, pošto se testovi za ciljnu platformu izvršavaju sa stanovišta krajnjeg korisnika. Osnovna ideja sastoji se u tome što TV prijemnik posmatramo kao crnu kutiju sa unapred poznatim ulazima, sa mogućnošću kontrolisanja kao i sa skupom referentnih audio/video izlaza. Za tako posmatrani skup uređaja razvijena je programska podrška koja omogućava njihovo kontrolisanje i funkcionisanje kao jedinstvene platforme.

Razvojem programske podrške i testiranjem konkretnog uređaja pronađen je određeni broj grešaka u radu testirane platforme. Pronađene greške su takve prirode da ih nije bilo moguće detektovati manualnim testiranjem (pri promeni ulaznog video signala testirana platforma pomera prikaz slike za jednu ili više tačaka (eng. piksela)). Realizacijom algoritma za poklapanje blokova bilo je moguće detektovati nastalu grešku. Takođe koristeći razvijenu programsku podršku za testiranje, moguće je detektovati razlike u slici pri korišćenju 3D filtra sa različitim vrednostima kompresije na testiranoj platformi (VGCA). Testovi se uspešno izvršavaju samo za vrednosti kompresije 3D filtra za koji su pravljani.

Sav dalji rad zasniva se na željama i zahtevima krajnjih korisnika programske podrške. Postoji širok spektar mogućnosti unapređenja funkcionalnosti programske podrške:

- Trenutno opis svakog testa sa referentnim datotekama (slika i audio zapis) se smešta u zaseban direktorijum. Kako bi se postigla centralizacija podataka, moguće je detalje testa smestiti u jedinstvenu bazu podataka.
- Projekti se formiraju na taj način što se kreira grupa testova koju izabere korisnik. Na taj način, datoteka projekta sadrži listu imena testova. Ukoliko se želi dodati test u već formiran projekat, mora se naknadno ažurirati datoteka projekta. Jedno od proširenja postojeće programske podrške je ako se svakom testu dodele odgovarajući atributi, koji opisuju funkcije koje testira taj test (testovi funkcionalnosti, PIP, PAT, meni, performanse, zvuk itd.). Tako se omogućuje da datoteke projekata ne sadrži imena testova, već samo niz atributa testova koje koristi taj projekat. Pokretanjem projekta, programska podrška iz grupe svih testova izdvoji testove sa datim atributima i započne njihovo izvršavanje. Tako je moguće kombinovati testove na različite načine.
- Testovi u aplikaciji predstavljaju celinu za sebe tj. rezultat jednog testa ne utiče na rezultat sledećeg testa. Kako bi se postigao neki oblik ulančavanja testova, potrebno je obezbediti na nivou projekta da rezultat jednog testa bude ulaz u sledeći test. Na taj način je moguće uslovno izvršavanje testova, što znači da u zavisnosti od rezultata testa stvaraju se različite grane toka izvršavanja.

- Za prepoznavanje slika koristi se algoritam baziran na računanju razlike između dve slike (PSNR). To je relativno jednostavan način za prepoznavanje, sa nedostacima kao što je osetljivost na osvetljaj u slikama koje se porede, osetljivost na pomeraj u slikama po vertikali ili horizontali. Rešenje predstavlja implementacija algoritma zasnovanog na Furijevoj transformaciji.

8. Literatura

- [1] Literatura sa predavanja iz predmeta Programska podrška u televiziji i obradi slike, FTN Novi Sad 2005
- [2] Teach yourself Visual C++ in 21 days, literatura u HTML obliku
- [3] Wikipedia, Internet enciklopedija (www.wikipedia.org)
- [4] BBT ACQUISITION BOARD user manual, PDF dokument za korišćenje BBT ACQUISITION BOARD platforme
- [5] VGCA EVA 5969 user manual, PDF dokument za korišćenje VGCA EVA platforme
- [6] MSDN library, Pomoć za Microsoft Visual Studio 6.0 razvojno okruženje
- [7] Video Codec Design, Iain E.G. Richardson, Razvoj sistema za video kompresiju
- [8] Rohde&Schwarz, Internet prezentacija (www.rohde-schwarz.com)