

# Primer arhitekture ATV SW-a zasnovane na 8-bitnoj CPU arhitekturi

Dalibor Ličina, Đorđe Simić, Ivan Velikić, Branislav Atlagić

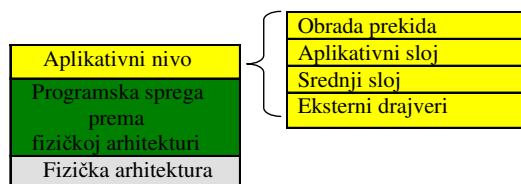
**Sadržaj** — U radu je naveden jedan primer arhitekture ATV SW-a zasnovanoj na 8-bitnoj arhitekturi. Naveden je opis strukture aplikativnog nivoa kao i način njenog funkcionisanja.

**Ključne reči** — aplikativni nivo, CC, EPG, TTX

## I. UVOD

U radu je predstavljeno jedno rešenje realizacije arhitekture ATV (*Analog Television*) programske podrške bazirane na 8-bitnoj CPU arhitekturi. Kako su resursi u ovom slučaju glavni ograničavajući faktor, neophodno je raspoložive resurse arhitekture koristiti na optimalan način. Kako se radi o sistemu sa velikim stepenom složenosti, postoji veliki broj registara koje on sadrži. U zavisnosti od režima u kojem se ATV trenutno nalazi, potrebno je određene registre ili delove registara podesiti u skladu sa tim.

Na sl. 1. je predstavljen blok-dijagram sistema. Osnova predstavljenog sistema jeste fizička arhitektura na koju se nadovezuje programska sprega prema fizičkoj arhitekturi. Programska sprega prema fizičkoj arhitekturi direktno koristi svu funkcionalnost fizičke arhitekture sistema i u funkciji je sprege između fizičke arhitekture sa jedne strane i aplikativnog nivoa sa druge. Krajnji korisnik ovakvog sistema ima neposrednu komunikaciju sa aplikativnim nivoom, pa je stoga ovaj nivo razvijan sa ciljem da bude što jednostavniji za korišćenje.



Sl. 1. Pregled programske arhitekture

Dalibor Ličina, MicronasNIT u Novom Sadu, Srbija (telefon: 381 21 4801 140); faks: 381 21 450721; e-mail: [dalibor.licina@micronas.com](mailto:dalibor.licina@micronas.com)

Đorđe Simić, MicronasNIT u Novom Sadu, Srbija e-mail: [djordje.simic@micronas.com](mailto:djordje.simic@micronas.com)

Ivan Velikić, MicronasNIT u Novom Sadu, Srbija e-mail: [ivan.velikic@micronas.com](mailto:ivan.velikic@micronas.com)

Branislav Atlagić, MicronasNIT u Novom Sadu, Srbija e-mail: [branislav.atlagic@micronas.com](mailto:branislav.atlagic@micronas.com)

## II. INTERNA SW STRUKTURA APLIKATIVNOG NIVOA

Struktura programskog koda na aplikativnom nivou je podeljena na četiri dela (sl. 1):

- Obrada prekida – poruka
- Aplikativni sloj
- Srednji sloj
- Spoljašnji programi za rukovanje fizičkom arhitekturom (*driver*)

### A. Obrada prekida - poruka

U ovom sloju realizovano je rukovanje događajima koji dolaze sa daljinskog upravljača, obrada sistemskih programskih prekida i vremenskih događaja, rukovanje promenom modula rada audia, obrada događaja vezanih za video sinhronizaciju, kao i obrada prekida vezana za CC (*Close Caption*), TTX (*Teletext*) i EPG (*Electronic Programable Guide*).

Osnovni delovi ovog sloja su obrađivači poruka i raspoređivači poruka (*dispatcher*). Rad celokupnog aplikativnog nivoa se zasniva na događajima. Sve dok se ne desi neki događaj, glavna petlja u programskom kodu (*while(1)-loop*) ne radi ništa. Svaka poruka ima svoj prioritet i način raspodele, odnosno obrada poruka se obavlja strogo po prioritetima. Obrada jedne poruke se ne može prekidati, tako da ostale obrade moraju da čekaju da se prethodna završi.



Sl.2. Sekvenca proveravanja novopristiglih poruka na nivou obrade poruka

S obzirom da se programski kod izvršava u jednom procesu, prednosti korišćenja mehanizma poruka, umesto običnih poziva funkcija, su:

- Na primer, običan poziv funkcije `Wait()` bi blokirao sistem suviše dugo. Umesto toga, program može da generiše poruku nakon nekog vremena, a vreme između tih događaja može da se iskoristi za obradu nečeg drugog.
- Obrade koje zahtevaju više vremena mogu da se podele u više sekvencijalnih delova. U tom slučaju završetak jednog dela će inicirati slanje poruke sa zahtevom za obradu narednog dela, omogućavajući da se između obrade tih delova, obradi i neki prioritarniji posao.

Nedostaci korišćenja mehanizma poruka su:

- Komplikovaniji su za pronalaženje grešaka u kodu
- Svaka nova poruka zahteva dodatni prostor u RAM-u
- Obrada je vremenski odložena (maksimalno dok se ne završi prethodna obrada) i zbog toga je problematična za one obrade koje moraju da se izvrše hitno.

Za rad sistema, navedeni nedostaci nisu od kritičnog značaja i kao takvi, ne ometaju njegov normalan rad.

#### B. Aplikativni sloj

Navigator je osnovna aplikacija koja se pokreće nakon uključivanja uređaja ili buđenja iz stanja pripravnosti (engl. *standby*). On omogućava navigaciju po raznim video ulazima i tjuner kanalima, a takođe omogućava i da se pozovu druge aplikacije. Neke od mogućih aplikacija su:

- TTX (*Teletext*)
- CC (*Close Caption*)
- EPG (*Electronic Program Guide*)
- Aplikacije za interakciju sa korisnikom

Aplikacije imaju pristup funkcijama i "znanjima" iz srednjeg sloja, delova jezgra sistema, kao što su funkcije vezane za TTX, CC, EPG i nekim sistemskim funkcijama, izuzev funkcijama za audio i video kojima se pristupa preko srednjeg sloja. Aplikativni sloj takođe ima omogućen pristup i vrednostima u EEPROM memoriji koji su od značaja za njega.

Sve reči koje se ispisuju na ekran, na svim raspoloživim jezicima, se nalaze u ovom sloju. Reči se posebnim alatima prevode u programski kod i tako omogućavaju jednostavno održavanje. Sve funkcije koje se koriste za rukovanje stringovima se nalaze takođe u ovom sloju.

Pored navedenog, u ovom sloju se nalaze i aplikacije koje služe za prikazivanjem mogućnosti same fizičke arhitekture.

#### C. Srednji sloj (Middleware)

Srednji sloj predstavlja jedan sloj apstrakcije iznad programske sprege prema fizičkoj arhitekturi. Ovaj sloj

sadrži skoro sve osnovne funkcionalnosti i sva znanja potrebna za kontrolisanje audia i videa u sistemu. Koristi funkcije programske sprege prema fizičkoj arhitekturi i prosleđuje skup funkcija aplikativnom sloju. Na ovaj način, obezbeđuje sve potrebne i osnovne audio-video funkcionalnosti.

Srednji sloj je sposoban da obavi autodetekciju platforme koju kontroliše. Takođe poseduje znanja o platformi na koju se oslanja i ima pristup funkcijama programskog jezgra. Za razliku od navedene kontrole nižih slojeva, srednji sloj nema pristup funkcijama i znanjima koja se nalaze u aplikativnom sloju.

Još jedan bitan zadatak srednjeg sloja jeste kontrola režima rada sistema (*power up/power down*) kao i podešavanje tabele prekida u zavisnosti od platforme na kojoj se programska podrška izvršava.

#### D. Eksterni drajveri

Ovaj sloj sadrži programe na niskom nivou koji kontrolišu rad integrisanih kola kao što su:

- EEPROM (Electrically Erasable Programmable Read-Only Memory)
- Tuner
- Spoljašnji DSP (*Digital Signal Processor*)
- HDMI (*High-Definition Multimedia Interface*) ulaz

### III. RAZMENA PORUKA UNUTAR SISTEMA

Tokovi poruka unutar predstavljenog rešenja su:

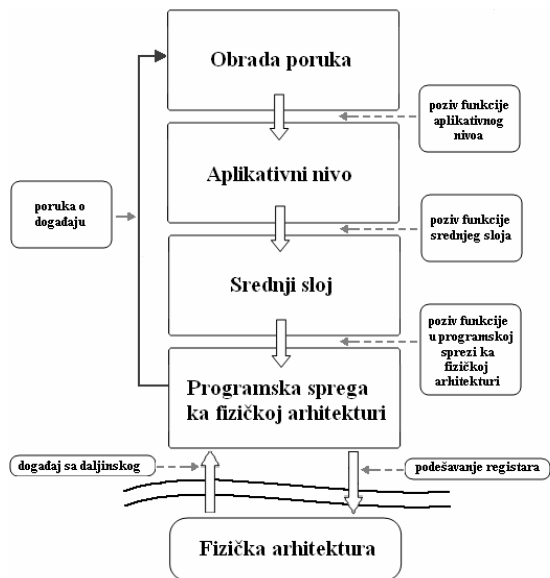
- "na dole" ako se radi o API (*Application Programming Interface*) zasnovanim pozivima funkcija i
- "na gore" ukoliko se radi o sistemskom porukama zasnovanim na događajima

Kako je rad sistema zasnovan na obradama događaja, gde su neki od njih potpuno asinhroni prirode, može se očekivati da se neki od događaja pojave u istom trenutku. Svi događaji unutar sistema koji se obrađuju, a dolaze do nivoa obrade poruka, se smeštaju u određeni niz. Pozicija u nizu na kojoj se događaj smešta direktno zavisi od prioriteta samog događaja. Prilikom zahvatanja događaja koji treba da se obradi, pristupa se tom nizu i na taj način određuje sam redosled obrade.

Najviši nivo u predstavljenom rešenju, nivo obrade poruka, dobija poruke od slojeva "ispod" sebe. Primer jednog od mogućih scenarija rada sistema u kojem je opisan tok i samo izvršavanje programa je dat u nastavku.

Kada krajnji korisnik pomoću IR (*Infrared*) kontrolera izabere opciju za menjanje trenutno prikazivanog kanala, na nivou obrade poruka otkriva se taj događaj i pristupa se njegovoj obradi. Nakon toga, poziva se određena funkcija na aplikativnom nivou koja ima za zadatak da rukuje sa događajem ovog tipa. Zatim, srednji sloj, koji predstavlja sprege između aplikativnog nivoa i programske sprege sa fizičkom arhitekturom, bi dalje poslao komandu nižem nivou da određenu grupu registara podesi na unapred poznate vrednosti kako bi se sistem doveo u traženi režim

rada. Na sl. 3. je prikazan blok-dijagram toka obrade poruke za opisan primer.



Slika 3. Primer sekvence obrade događaja u sistemu

Na sličan način se obrađuju i ostali događaji koji dolaze do nivoa obrade poruka.

#### IV. ZAKLJUČAK

Predstavljeno rešenje ATV SW-a sa značajnim ograničenjem sa stanovišta resursa, predstavlja veoma dobro rešenje što se u praksi i pokazalo. Pojednostavljena realizacija celokupne programske podrške je jednostavna za korišćenje i za eventualno nadograđivanje u nekim sledećim iteracijama.

#### LITERATURA

- [1] Jedno rešenje programske podrške za automatsko testiranje TV prijemnika
- [2] Wikipedia, free encyclopedia ([www.wikipedia.org](http://www.wikipedia.org))

#### ABSTRACT

This paper presents one solution of architecture ATV SW based on 8 bit CPU architecture. There is also described structure of application level and short introduction how does it works.

#### EXAMPLE OF ARCHITECTURE ATV SW BASED ON 8-bit CPU ARCHITECTURE

Dalibor Ličina, Đordje Simić, Ivan Velikić, Branislav Atlagić