

# Koncept i realizacija upravljanja domaćinstvom korišćenjem postojećih uređaja kućne elektronike

Milan Z. Bjelica, Nikola Teslić, Zoran Jovanović, Zoran Marčeta, *Fakultet tehničkih nauka, Novi Sad*

**Sadržaj** — U okviru rada je izložen koncept kontrolera za automatizaciju domaćinstva koji može biti ugrađen kao proširenje nekog od postojećih uređaja potrošačke elektronike u domaćinstvu, kao što je TV prijemnik ili uređaj za reprodukciju multimedijalnog sadržaja. Koncept je proveren kroz realizaciju kontrolera kao proširenja digitalnom TV prijemniku, sa dodatkom adaptera za kontrolu svetla i uređaja putem električne instalacije. Osnovni cilj je bio da se predloži uopšteni model programske podrške koja bi bila u stanju da podrži većinu rasprostranjenih protokola za automatizaciju i rasvetu. Dodatni aspekti od značaja se odnose na modeliranje ponašanja domaćinstva, prijemčivu grafičku korisničku spregu (GUI) i ekonomičnu upotrebu sistemskih resursa.

**Ključne reči** — automatizacija domaćinstva, pametna kuća.

## I. UVOD

LAKO i brzo obavljanje aktivnosti koje donosi svakodnevni život i uvođenje računarskih tehnologija kao podrške automatizaciji domaćinstva predstavljaju ciljeve koji nisu novina. Ipak, opšti utisak je da su postojeća rešenja podeljena na veliki broj različitih, specifičnih primena u domaćinstvu [1]. Da bi se domaćinstvo automatizovalo, potrebno je nabaviti, ugraditi i podesiti raznolike komponente fizičke arhitekture, koje neretko zahtevaju postavljanje dodatnih kablova široke palete standarda. Korišćenje postojećih uređaja i instalacija u domaćinstvu za njegovu kontrolu nameće se kao zanimljiva alternativa.

U svakom modernom domaćinstvu nalazi se veliki broj uređaja koji koriste procesore, iako njihova prisutnost nije tako očigledna. Osim PC i laptop računara, uređaji koji svoje funkcije obavljaju zahvaljujući ugrađenim procesorima su i TV prijemnici, multimedijalni plejeri i brojni uređaji bele tehnike. Idealnu osnovu za preuzimanje funkcija upravljanja domaćinstvom može da predstavlja TV prijemnik. S obzirom da je TV prijemnik ugrađeni uređaj (*embedded system*), odlikuje se povećanom pouzdanošću rada i robustnošću. Mogućnost reprodukcije

multimedijalnih formata svedoči o moćnim procesorima za obradu koje ovaj uređaj poseduje (često 32-bitna arhitektura sa 32 MB radne memorije ili više). Pripravni režim rada (*stand-by*) omogućio bi neprekidnu posvećenost jednostavnim zadacima kontrole domaćinstva. Kontroler za automatizaciju, koji je opisan u nastavku rada, sastoji se od prenosive programske podrške koja je optimizovana za rad na platformama skromnih resursa. Za podršku vezi sa svetlima, uređajima i senzorima neophodni su dodatni fizički adapteri.

Dalje, sistem za automatizaciju domaćinstva zavisi od infrastrukture za povezivanje. U cilju lakšeg uvođenja sistema, kontrola se može obavljati preko postojeće električne mreže. Prenosjenje multimedijalnog sadržaja se može obaviti i preko Ethernet mreže, ako postoji.

U ovom radu predlaže se ugrađeno rešenje za kontrolu domaćinstva koje je zasnovano na prenosivoj POSIX/Linux programskoj podršci sa serijskom vezom do komunikacionog adaptera ka električnoj mreži. Korišćenje adaptera za druge vrste medijuma je takođe moguće. U nastavku je dat pregled koncepta koji se odnosi na programsku podršku, modelovanje ponašanja i prenosivost GUI. Na kraju je predstavljena realizacija kontrolera i izvedeni su zaključci o njegovoj upotrebnoj vrednosti.

## II. KONCEPTI UGRAĐENOG KONTROLERA

Da bi se kontroler zaista mogao realizovati na različitim platformama, programska podrška mora da bude prenosiva, a korišćene sprege rasprostranjene u industriji potrošačke elektronike. Osnova razvoja se zasniva na sledećim smernicama:

- *Programski jezik za ugrađene sisteme*, s obzirom na zahteve rada na raznolikim fizičkim arhitekturama. Najbolji izbor u ovom slučaju je programski jezik C.
- *Nezavisna programska podrška*, što znači da programski kod ne sme da se oslanja na biblioteke koje su specifične za razvojno okruženje. Zauzimanje memorije treba obavljati statički. Treba izbegavati operacije koje se oslanjaju na koncept sistemskog steka, kao što su rekurzivni pozivi funkcija. Pošto je konkurentno programiranje neizbežno, potrebno je koristiti standardnu konkurentnu biblioteku. Operacije nad datotekama i sistemskim spregama takođe moraju biti standardne. Poslednja dva zahteva mogu se ispuniti korišćenjem POSIX biblioteka.
- *Standardna sprega ka adapterima*. Preporučuje se povezivanje adaptera sa razvojnom platformom korišćenjem serijske sprege UART (*Universal*

Ovaj rad je delimično finansiran od Ministarstva za nauku Republike Srbije, projekat 11005, od 2008. god.

Milan Z. Bjelica, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija (telefon: 381-21-4801139; faks: 381-21-450721; e-mail: milan.bjelica@rt-rk.com).

Nikola Teslić, e-mail: nikola.teslic@rt-rk.com

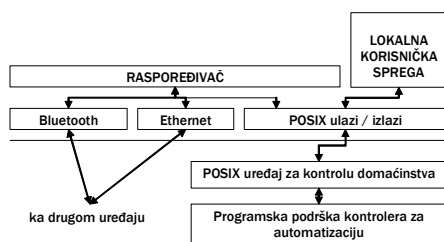
Zoran Jovanović, e-mail: zoran.jovanovic@rt-rk.com

Zoran Marčeta, e-mail: zoran.marčeta@rt-rk.com

*Asynchronous Receiver/Transmitter*), pošto je ona prisutna u mnogim ugrađenim uređajima. Potrebno je obezbediti modularnost programske podrške u cilju omogućavanja rada sa različitim spregama.

- *Spoljni sistemski časovnik*, koji obezbeđuje podatke o datumu i vremenu. Časovnik treba povezati sa ciljnom platformom korišćenjem standardne sprege.
- *Prilagodljivi srednji sloj programske podrške korisničke sprege*, pošto nije moguće obezbediti jedinstven mehanizam korisničke sprege za različite uređaje u funkciji kontrolera za automatizaciju domaćinstva. Umesto toga, potrebno je omogućiti jednostavno prilagođavanje programske podrške različitim grafičkim okruženjima i fizičkim ulaznim uređajima. U cilju povećavanja verovatnoće upotrebe programske podrške bez izmena, preporučuje se korišćenje *DirectFB* grafičkog podsistema zasnovanog na operativnom sistemu (OS) *Linux*, uz dodatak okruženja za kreiranje GUI kao što su *GTK* ili *QT*.
- *Jednostavna navigacija korisničke sprege*. Mogućnost primene u većini uređaja potrošačke elektronike biće moguća ako se navigacija oslanja na koncept „strelica“ (↑ ↓ ← →) i tastera OK za potvrđivanje.

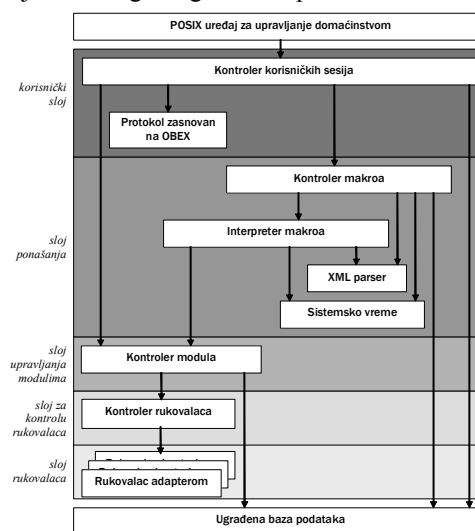
Odabrani standardi u okviru predloženog koncepta su C/POSIX/UART. Da bi se olakšala integracija korisničke sprege, celokupna programska podrška kontrolera je „skrivena“ unutar standardnog POSIX uređaja. Komunikacija sa ovim uređajem se obavlja standardnim *read/write* pozivima, uz korišćenje protokola zasnovanog na OBEX [2], [3]. Na taj način kontroler pruža usluge korisničkoj sprezi, uz jasno razdvajanje funkcionalnosti (automatizacija domaćinstva) od prezentacije (korisnička sprega). Ove usluge može koristiti korisnička aplikacija na samoj platformi, ili se ulazi/izlazi POSIX uređaja mogu povezati sa ostalim POSIX spregama (Ethernet, WiFi, Bluetooth itd). Ovo omogućava izmeštanje korisničke sprege na drugi uređaj. Npr. TV prijemnik može biti korišćen kao kontroler za automatizaciju domaćinstva, dok jednostavna Java aplikacija može omogućiti kontrolu sa mobilnog telefona! Ovaj koncept je prikazan na **Slici 1**.



Slika 1. Razdvajanje funkcionalnosti od prezentacije

Programska podrška kontrolera za automatizaciju domaćinstva se sastoji od nekoliko slojeva zaduženih za različite zadatke (Slika 2). Najniži sloj u hijerarhiji (*sloj rukovalaca*) zadužen je za interakciju sa adapterima za fizičko povezivanje (npr. električna mreža, lokalni *ethernet*) posredstvom UART sprege. Jedinstveni prikaz svih rukovalaca, njihovu evidenciju i inicijalizaciju obezbeđuje *sloj za kontrolu rukovalaca*. Kako se domaćinstvo sastoji od različitih modula (svetla, uređaji, senzori) te kako modul predstavlja objekat upravljanja,

uvodi se *sloj upravljanja modulima*. Operacije ovog sloja obuhvataju zadavanje komandi modulima (npr. za zatamnivanje svetla), kao i mogućnost registrovanja za prijem različitih događaja koji moduli objavljuju (npr. otkrivanje prisustva, promena intenziteta dnevne svetlosti i sl). Upravljanje modulima se takođe može obavljati korišćenjem *modela ponašanja*. Ponašanje se definiše kao makro zasnovan na XML-u u kome je naveden redosled upravljačkih operacija koje se izdaju modulima nakon prijema određenog događaja (vreme, datum, događaj koji objavljuje neki od modula). Interpretaciju makroa, operacije upravljanja makroima kao i njihovo skladištenje obezbeđuje *sloj ponašanja*. Konačno, sloj najvišeg nivoa hijerarhije, pod nazivom *korisnički sloj*, zadužen je za komunikaciju sa GUI korišćenjem sesionog protokola zasnovanog na OBEX. Primitive POSIX uređaja se neposredno oslanjaju na odgovarajuće API (*Application Programming Interface*) pozive korisničkog sloja. Dodatno, za omogućavanje skladištenja podataka obezbeđena je baza podataka, kao podrška svim slojevima. Baza je zasnovana na mehanizmu dvostruko spregnutih listi sa unapred zauzetom memorijom, te je pogodna za realizaciju na mnogim ugrađenim platformama.

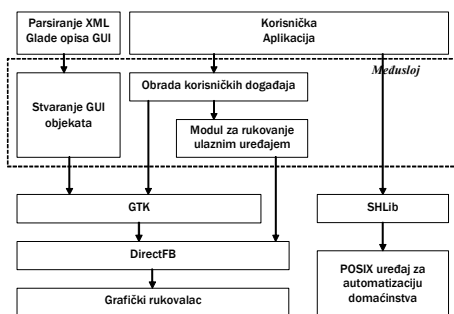


Slika 2. Slojevita programska podrška

Da bi programska podrška korisničke sprege jednostavno koristila funkcije kontrolera, obezbeđena je posebna C biblioteka za različite ciljne platforme (*Windows, Linux, Java*), pod imenom *SHLib*. Ova biblioteka sadrži API sa svim potrebnim funkcijama, uz skrivanje detalja komunikacije od programera. Npr. moguće je pregledati inventar modula, kontrolisati svetla, odabirati željeni scenario ponašanja domaćinstva i sl. jednostavnim pozivanjem odgovarajuće *SHLib* funkcije.

Iako je moguće razdvojiti korisničku spregu od mehanizama za kontrolu domaćinstva, korisnička sprega može biti razvijena i za lokalnu platformu. Predloženi koncept se oslanja na grafičko okruženje otvorenog koda *GTK* [4] koje koristi usluge *DirectFB* [5] grafičkog rukovaoca pod OS *Linux*. Pokazano je da je ovakav izbor osnovne grafičke programske podrške efikasan za aplikacije na ugrađenim uređajima. [6]. Zahvaljujući tome, GUI se lako može kreirati korišćenjem alata *Glade* [7], uz

određena pravila projektovanja. Ova pravila se odnose na pozicioniranje i imenovanje grafičkih oblika. Međusloj programske podrške, koji se oslanja na *GTK* i *DirectFB*, tumači XML opis projekta sačuvan u okviru *Glade* alata i kreira grafičke oblike *GTK* okruženja. Pravilima projektovanja se određuje koji su grafički oblici (npr. ekranski tasteri) aktivni, kako izgledaju kada ih korisnik odabere ili aktivira, kao i koja funkcija u okviru programske podrške se poziva nakon aktiviranja. Međusloj takođe upravlja prikazom trenutnog stanja grafičkih oblika, tako što reaguje na različite korisničke događaje (npr. pritisak na tastere ulaznog uređaja). Način generisanja ovih događaja zavisi od ciljne platforme, pa je potrebno dodati programski modul za rukovanje ulaznim uređajima na toj platformi. Celokupan koncept je prikazan na Slici 3.



Slika 3. Koncept programske podrške GUI

Predloženi model ponašanja koristi XML makroe koji to ponašanje opisuju. Pokretanje određenog slede akcija može biti uslovljeno prijemom objave događaja od strane nekog od modula, ili datumom i vremenom. Kada se uslov ispuni, pokreće se sekvenca programskog koda ugnježdjena u telo makroa. Ovaj kod se dalje prevodi u naredbe koje se upućuju određenim modulima. Npr. moguće je uključiti TV prijemnik i zatamniti svetla u spavaćoj sobi svakog utorka i četvrtka u 22h, pošto se tada emituje željeni TV program. Makroi mogu da sadrže programske konstrukte uobičajene za programske jezike višeg nivoa (petlje *for* i *while*, definisanje promenljivih, uslovno izvršavanje i *sl*). Mehanizam interpretacije makroa zasnovan je na *XML-like* skript interpreteru [8]. Primer jednog makroa za opis ponašanja domaćinstva prikazan je na Slici 4.

```
<driver type=signal modulename=daysensor status=on
intensity=10>
<print> Driver for morning wakeup </print>
<print> Description: When curtain opens second
time at daylight, turn lights off.
</print>
<print> </print>

<for>
<var name="counter" val="1"/>
<poll modulename="Curtain" status="on"/>
<if>
<case name="counter" val="2" rel="="= ">
<var name="counter" val="1"/>
<for loop=3>
<instruct moduleid="1"
function="oFF" .../>
<instruct modulename="Lamp"
function="oFF" .../>
<instruct moduleid="5"
function="oFF" .../>
</for>
</case>
<default>
<var name="counter" val="++"/>
</default>
</if>
</for>
</driver>
```

Slika 4. Primer makroa za opis ponašanja domaćinstva

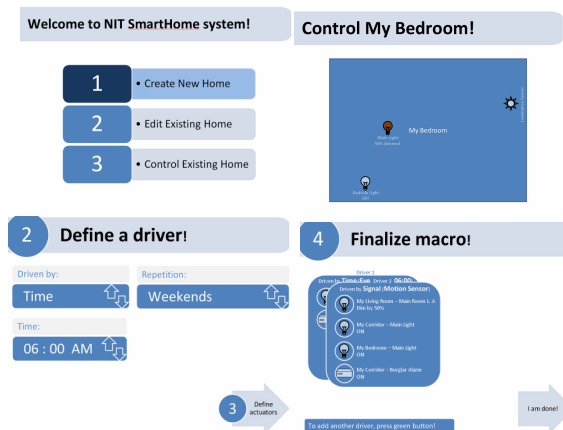
### III. STUDIJA SLUČAJA

Za proširenje funkcijama za kontrolu domaćinstva odabran je uređaj za reprodukciju multimedijalnog sadržaja i TV programa (*set-top box*), zasnovan na integrisanim kolima za digitalnu televiziju kompanije *Trident Microsystems* [9]. Osnovno integrisano kolo sadrži 32-bitni MIPS procesor sa OS *Linux* za realno vreme. Opredelilo se za razvoj GUI na istom uređaju, s obzirom na to da je opremljen fizičkom arhitekturom za rad sa grafikom, kao i spregom na ekran (HDMI).

Programska podrška kontrolera za automatizaciju domaćinstva napisana je kao proširenje postojećem *Linux* jezgri (kao *.ko* objekat), i predstavljena *Linux* korisničkom prostoru kao uobičajeni POSIX uređaj u sistemu datoteka. *ShLib* biblioteka prevedena je kao standardna *Linux* deljena biblioteka (*.so*). GUI aplikacija je razvijena u okviru korisničkog prostora *Linux*-a. Razvoju je prethodilo prevođenje *GTK* okruženja za MIPS procesor, što je obavljeno na posebnom PC računaru. Dobijene *GTK* biblioteke su zatim prenesene na ciljnu MIPS platformu. GUI međusloj je povezan sa krajnjom GUI aplikacijom. Napisan je poseban programski modul za prijem signala sa daljinskog upravljača kao jedinog ulaznog uređaja i objavljivanje odgovarajućih događaja međusloju.

Da bi se omogućila stvarna kontrola svetla i uređaja u domaćinstvu, korišćen je tradicionalni protokol za komunikaciju putem postojeće električne mreže *X10* [10]. Da bi se podržao *X10*, korišćen je *Marmitek CM11 X10* primopredajnik [11]. UART sprega je korišćena za povezivanje *CM11* sa ciljnom platformom. Jednostavan serijski komunikacioni protokol je razvijen u obliku rukovaoca adapterom kao dela programske podrške kontrolera za automatizaciju. Na strani svetla korišćen je *LW12* [12] aktuatorski modul, koji se ugrađuje iza uobičajenih prekidača u domaćinstvu, dok je *AM12* [13] sa impulsnim relejem korišćen za uključivanje uređaja.

Veliki izazov je predstavljao razvoj GUI tako da ona bude razumljiva i jednostavna za korišćenje. Cilj je bio da se omogući jednostavna kontrola svetla i uređaja (osnovni režim) kao i opisivanje ponašanja domaćinstva (poboljšani režim), bez potrebe za unošenjem koda makroa. Korišćen je GUI sa više stranica, uz koncept Dalje>Natrag za navigaciju. Neke GUI stranice su prikazane na Slici 5. Detaljan opis realizacije sistema autori daju i u [14].



Slika 5. Delovi GUI za upravljanje domaćinstvom

Nakon korisničkog ispitivanja, pokazano je da je sistem efikasan i ugodan za korišćenje. Primedbe su se odnosile na prirodu odredišnog uređaja kao deljenog resursa: dešavalo se da je potrebno koristiti daljinski upravljač i ekran za gledanje TV programa i kontrolu domaćinstva u isto vreme. S druge strane, koncept opisivanja ponašanja i jednostavan način izmene aktivnog scenarija usloveli su manje čestu upotrebu same aplikacije. Te mogućnosti su učinile kontrolu domaćinstva vrlo zanimljivom. Studije slučaja vezane za upotrebu uređaja kao deljenog resursa i motivaciju korisnika date su u [15] i [16].

#### IV. RANIJA ISTRAŽIVANJA

Koncept uopštavanja kontrole domaćinstva, sa idejama sličnim onim u ovom radu, naveden je u okviru ranijih istraživanja u ovoj oblasti. Coyle et al. u [17] predstavili su međusloj za upravljanje domaćinstvom, pokušavajući da omoguće različitu primenu uređaja i senzora u domaćinstvu. Na sličan način, Casimiro, Kaiser i Verissimo prikazuju okruženje za povezivanje inteligentnih komponenti [18]. Pellegrino, Bonino i Corno u [19] uvode koncept slojevite programske podrške za primenu u pametnim kućama (*domotics*). Gauger et al. u [20] pokazuju mogućnosti primene mreža senzora i aktuatora u upravljanju domaćinstvom, sa posebnim osvrtom na probleme rasvete. Korisničke sprege za upravljanje domaćinstvom takođe su bile tema nekih radova, iako su predložena rešenja posedovala samo delove funkcionalnosti. Koskela i Vaananen uporedili su PC, TV i mobilni telefon kao uređaje za spregu sa korisnikom u cilju upravljanja domaćinstvom [21]. Nichols i Myers predlažu upotrebu inteligentnih telefona (*smart phones*) za kontrolu uređaja u domaćinstvu, i uvode termin *Lični univerzalni upravljač (Personal Universal Controller)* [22]. Ovakav koncept se takođe realizuje kao programska podrška otvorenog koda, u okviru projekta *Open Remote* [23].

Sudeći po našem poznavanju oblasti, do sada nije bilo radova na temu proširenja postojećih uređaja u domaćinstvu kako bi se omogućilo upravljanje domaćinstvom. Perumal, Ramli i Leong predlažu koncept upravljanja domaćinstvom slanjem poruka sa mobilnog telefona [24], ali se stvarno upravljanje takođe obavlja posebnim uređajem. U okviru ovog rada, pokušalo se da se skicira sistem i razvije programska podrška koja se sa lakoćom može primeniti u postojećim kućnim uređajima. Dodatno, mogućnost B2B modela sa proizvođačima kućne elektronike čini ovaj predlog komercijalno atraktivnim.

#### V. ZAKLJUČAK

U ovom radu dat je koncept koji može biti osnova za dalja istraživanja u oblasti sistema za automatizaciju domaćinstva, sa naglaskom na lakoći ugradnje i efikasnoj upotrebi postojećih resursa u domaćinstvu. Predstavljena je realizacija kontrolera domaćinstva sa ugrađenom GUI, kao proširenja standardnom uređaju za reprodukciju multimedijalnog sadržaja, koji se pokazao efikasnim i zanimljivim za krajnje korisnike.

#### LITERATURA

- [1] *Home Automation Index*, <http://www.homeautomationindex.com>
- [2] Megowan, P, Suvak, D, Kogan, D, *IrDA Object Exchange Protocol OBEX™*, Infrared Data Association, Version 1.2, 1999
- [3] Bjelica, M. Z, Savic, M, Aleksic, T, *Jedno rešenje pregledača datoteka sa povezanog Bluetooth uređaja na TV prijemniku*, zbornik radova konferencije ETRAN, 2008
- [4] Krause, A, *Foundations of GTK+ development*, Apress, 2007
- [5] Hundt, A, *DirectFB Overview*, PDF, 2004
- [6] Bjelica, M. Z, Teslic, N, Savic, M, *Upotreba Open Source okruženja za razvoj grafičke korisničke sprege za TV prijemnike*, zbornik radova konferencije ETRAN, 2009
- [7] *Glade – A User Interface Designer*, <http://glade.gnome.org/>
- [8] Bjelica, M. Z, Savic, M, Vujanovic, V, Temerinac, M, *Izrada PC aplikacije za emulaciju komunikacionog podsistema integrisanog kola*, zbornik radova konferencije TELFOR 2008
- [9] *Trident Microsystems*, <http://www.tridentmicro.com>
- [10] Gerhart, J, *Home Automation and Wiring*, McGraw-Hill/TAB Electronics, 1<sup>st</sup> edition, 1999
- [11] *CM11 Manual*, <http://www.marmitek.com/en/manual/9647.pdf>
- [12] *LW12 Manual*, <http://www.marmitek.com/en/manual/9611.pdf>
- [13] *AM12 Manual*, <http://www.marmitek.com/en/manual/8915.pdf>
- [14] Bjelica, M. Z, *Realizacija sistema za automatizaciju domaćinstva sa grafičkom korisničkom spregom na TV prijemniku*, master rad, biblioteka Fakulteta Tehničkih Nauka, Novi Sad, 2008
- [15] Bjelica, M. Z, Teslic, N, *A Concept of System Usability Assessment: System Attention as the Measure of Quality*, in proceedings of IEEE Eastern European Regional Conference on the Engineering of Computer Based Systems, 2009
- [16] Bjelica, M. Z, Teslic, N, *A Concept of Usability Assessment for User-Centered Multimedia Applications*, in proceedings of IMCSIT MMAP 2009, Poland (to be published)
- [17] Coyle, L, Neely, S, Stevenson, G, Sullivan, M, Dobson, S, Nixon, P, *Sensor Fusion-Based Middleware for Smart Homes*, International Journal of ARM, Vol. 8, No. 2, June 2007
- [18] Casimiro, A, Kaiser, J, Verissimo, P, *An Architectural Framework and a Middleware for Cooperating Smart Components*, in proceedings of CF'04 Italy, ACM, 2004
- [19] Pellegrino, P, Bonino, D, Corno, F, *Domotic House Gateway*, in proceedings of SAC'06 France, ACM, 2006
- [20] Gauger, M, Minder, D, Marron, P. J, Wacker, A, Lachenmann, A, *Prototyping Sensor-Actuator Networks for Home Automation*, in proceedings of REALWSN'08 Glasgow, ACM, 2008
- [21] Koskela, T, Vaananen-Vainio-Matilla, K, *Evolution towards smart home environments: empirical evaluation of three user interfaces*, Personal Ubiquitous Computing, Vol. 8, 2004
- [22] Nichols, J, Myers, B. A, *Controlling Home and Office Appliances with Smart Phones*, IEEE Pervasive Computing, 2006
- [23] *Open Remote: The Digital Home Operating System* <http://www.openremote.org/display/HOME/OpenRemote>
- [24] Perumal, T, Ramli, A. R, Leong, C. Y, *Remote triggering based embedded server for smart home environment*, in proceedings of ISITS'07, Malaysia, 2007

#### ABSTRACT

This paper proposes a concept of a home controller for home automation as an add-on to existing consumer electronic devices in the household, such as TV sets and set-top boxes. The concept is investigated through implementation in a digital set-top box extended with a unit for power line communication to control lights and appliances. Main goal was to propose a generic software model that would be able to support the majority of field-proven protocols for automation and lighting. Additional important aspects, such as home behavior modeling, comprehensive graphical user interface (GUI) and economic use of resources, were covered.

#### UTILIZING CONSUMER ELECTRONIC DEVICES IN HOME FOR HOME AUTOMATION TASKS

M. Z. Bjelica, N. Teslić, Z. Jovanović, Z. Marčeta