

Program za komunikaciju računara sa proizvoljnom razvojnom platformom po I²C magistrali

Ivan Letvenčuk, Vladimir Maksović, Tomislav Maruna, Nikola Teslić

Sadržaj — Predstavljen je program za komunikaciju računara sa proizvoljnim razvojnim platformama po I²C magistrali. U uvodnom delu je pored namene programa opisana struktura i način rada Eclipse platforme – platforme za koju je program razvijen. Takođe je opisana i I²C magistrala, njen značaj i njene osnovne karakteristike. U drugom delu je prikazan sam model programa sa detaljnim opisom funkcionalnosti svakog njegovog dela.

Ključne reči — dll, Eclipse, I²C, Java, JNI, vi2c.

I. UVOD

esto se komunikacija između računara i razvojne platforme obavlja po I²C magistrali. Jedan od primera korišćenja ove magistrale je promena vrednosti registara kontrolera uređaja na razvojnoj platformi radi konfigurisanja uređaja, provere ispravnosti rada i nalaženje grešaka u radu samog uređaja, ukoliko postoje.

U ovom slučaju se putem aplikacije na PC strani u registre kontrolera uređaja čita ili piše, a sama aplikacija je realizovana kao proširenje za Eclipse razvojno okruženje. Većina kôda aplikacije je napisana u programskom jeziku Java, izuzev dela koji služi za komunikaciju sa uređajem, a koji je napisan u programskom jeziku C.

A. Eclipse platforma

Eclipse platforma [1] predstavlja razvojno okruženje koje obezbeđuje osnovu i usluge pomoću kojih se kreiraju proširenja za Eclipse. Istovremeno, predstavlja i okruženje u kome se proširenja učitavaju, integrišu i izvršavaju. Naime, sama platforma je realizovana kao skup proširenja koji međusobno komuniciraju u cilju izvršenja nekog zadatka. Ovakav način funkcionisanja dozvoljava proširivanje funkcionalnosti Eclipse okruženja. Takođe, Eclipse platforma služi i za razvoj raznovrsnih samostalnih aplikacija kao i kreiranje potpuno novih aplikacija od već postojećih delova – RCP (Rich Client Platform) koncept

Rad je delimično podržan u okviru projekta TR-6136B Ministarstva za nauku i zaštitu životne sredine Republike Srbije.

Ivan Letvenčuk, autor, Fakultet Tehničkih Nauka, Novi Sad, Srbija (e-mail: ivan.letvencuk@micronasnit.com, telefon: 064/23-11-487).

Vladimir Maksović, Koautor, MicronasNIT, Institut za informacione tehnologije, Novi Sad, Srbija (e-mail: vladimir.maksovic@micronas.com).

Tomislav Maruna, Koautor, MicronasNIT, Institut za informacione tehnologije, Novi Sad, Srbija (e-mail: tomislav.maruna@micronas.com).

Nikola Teslić, Koautor, Fakultet Tehničkih Nauka, Novi Sad, Srbija (e-mail: nikola.teslic@micronas.com).

[2].

Platforma je razvijena tako da radi na većini operativnih sistema uključujući Windows®, Linux™, Mac OS X, Solaris AIX, itd. To je postignuto time što je jezgro Eclipse platforme implementirano u programskom jeziku Java. Naime, portabilni programi se mogu implementirati i u drugim programskim jezicima koji obezbeđuju portabilnost kao na primer C, C++, itd. Problem sa ovim jezicima je u tome što se pri prevođenju programa u binarni kôd, prevodioci moraju usmeriti na konkretan procesor. Sa druge strane, prevodilac programskog jezika Java ne generiše izvorni kôd, već tzv. bajt kôd. Bajt kôd predstavlja visookoptimizovan skup instrukcija koji u trenutku pokretanja programa tumači Javin izvršni sistem, poznat kao Java Virtualna Mašina (Java Virtual Machine – JVM). JVM je, u stvari, program koji se ponaša kao virtualni procesor koji interpretira, a potom i izvršava instrukcije bajt kôda. Zbog toga i Eclipse platforma u svom radu koristi JVM.

Namena same Eclipse platforme je da omogući projektantima što lakši i brži razvoj programskih alata i aplikacija. S obzirom da platforma, između ostalog, već sadrži razvijen podsistem za komunikaciju sa korisnikom (ulazno-izlazni podsistem) kao i to da poseduje proširenje specijalizovano za komunikaciju sa razvojnim platformama – RSE (Remote System Explorer), stoga je i VI2C aplikacija realizovana kao proširenje za Eclipse.

B. I²C magistrala

Elektronske komponente se međusobno često povezuju mnoštvom adresnih provodnika i provodnicima za podatke. Takav način povezivanja uslovljava i postojanje adresnih dekodera kao i postojanje posebnih logičkih komponenti koje omogućavaju da čitav sistem ispravno funkcioniše. Iz tih razloga sistem postaje složen, težak za izradu, podložan greškama pri projektovanju i na kraju nepristupačan krajnjim korisnicima zbog visoke cene proizvodnje.

Da bi prevazišla ove probleme, kompanija Philips je osamdesetih godina prošlog veka razvila I²C magistralu [3]. Njene osnovne karakteristike su jednostavnost i fleksibilnost, što je postignuto sledeći način:

- svaki uređaj na magistrali poseduje jedinstvenu adresu,
- sama magistrala je realizovana pomoću samo dva provodnika – jedan služi za sinhronizaciju između

- uređaja, a drugi za prenos adresa i podataka,
- uređaj na magistrali može biti vodeći (master) ili prateći (slave). Uređaj je vodeći ako on inicira prenos podataka a u tom trenutku su svi ostali uređaji prateći. Na magistrali može postojati više vodećih uređaja
- signal takta generiše vodeći uređaj,
- svaki uređaj može biti predajnik (transmitter) ili prijemnik (receiver) tj. može i da šalje i da prima podatke,
- specifikacijom podržane brzine prenosa su: 100kb/s, 400kb/s i 3,4Mb/s, itd.

Princip rada magistrale je sledeći:

1. vodeći uređaj prvo pošalje na magistralu START bit, zatim sledi sedmobitna adresa pratećeg uređaja iza koje sledi još jedan bit koji označava smer prenosa podataka,
2. prateći uređaj koji prepozna svoju adresu šalje bit potvrde i prenos podataka počinje,
3. predajnik šalje podatke na magistralu u osmобitnim paketima,
4. po prijemu svakog paketa prijemnik šalje bit potvrde,
5. ukoliko je vodeći uređaj istovremeno i prijemnik, bit potvrde izostaje po prijemu poslednjeg paketa,
6. prenos podataka se završava kada vodeći uređaj pošalje STOP bit.

Prvobitna namena I²C magistrale je bila da međusobno povezuje komponente koje se nalaze na istoj razvojnoj platformi, mada se sada koristi i za komponente koje se ne nalaze na istoj platformi.

II. MODEL VI2C PROŠIRENJA ZA ECLIPSE

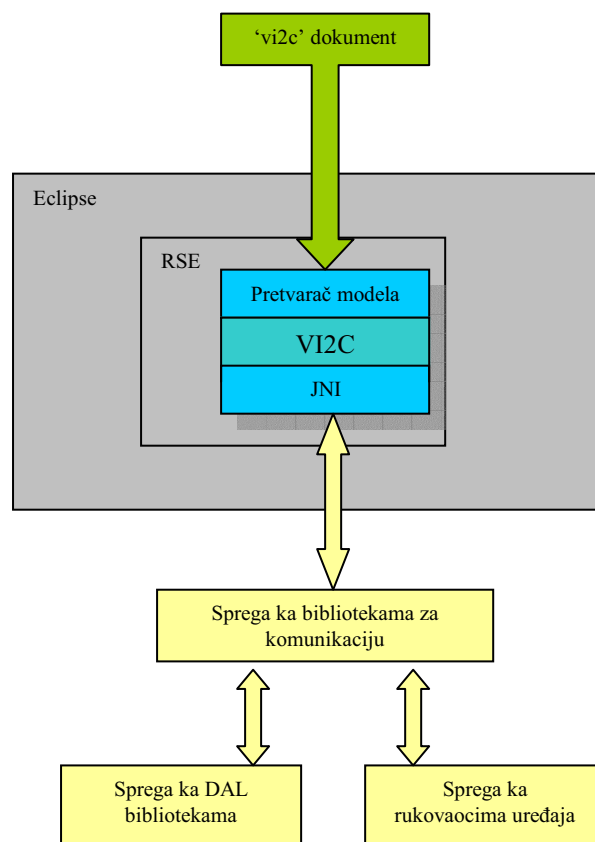
VI2C proširenje za Eclipse je program koji putem neke od podržanih sprega ka I²C magistrali (serijski port, paralelni port, USB, itd.) šalje i prima podatke sa kontrolera uređaja, a potom ih i prikazuje. Blok dijagram aplikacije je prikazan na slici Sl. 1.

A. Učitavanje VI2C dokumenta

Kontroler je modelovan skupom od jednog ili više XML (Extensible Markup Language) [4] dokumenata sa ekstenzijom 'vi2c'. Glavni razlog za izbor XML jezika kao jezika za modelovanje kontrolera je taj što je struktura XML dokumenta precizno definisana specifikacijom XML jezika. To znači da će svaki alat koji služi za rad sa XML dokumentima, sadržaj dokumenta interpretirati na potpuno isti način. Sa druge strane, ova činjenica otvara mogućnost komunikacije između više aplikacija putem XML dokumenata. Pomoću XML jezika se može opisati bilo koja struktura podataka, jer elemente XML dokumenta definiše sam korisnik. Takođe, XML dokument je tekstualnog formata, a ne binarnog, što ga čini čitljivim (razumljivim) za čoveka.

Dokument, u ovom slučaju, interpretira deo VI2C proširenja za Eclipse zadužen za parsiranje i prevodi ga u

interni model. Takav prevedeni model nadalje služi kao osnova za komunikaciju sa razvojnom platformom i grafički prikaz. Sadržaj dokumenta je hijerarhijski uređen skup elemenata gde svaki element sadrži: a) podatke koji se prenose po I²C magistrali i b) dodatne podatke koji služe aplikaciji za grafički prikaz.



Sl. 1. Blok dijagram aplikacije

Dodatni podaci koje element sadrži mogu biti i reference na druge 'vi2c' dokumente kao i skriptove pisane u Tcl skript jeziku [5]. Svaki element 'vi2c' dokumenta može da sadrži svoj Tcl skript. Ovi skriptovi dodatno proširuju funkcionalnost programa. Određene akcije tokom rada programa određuju momenat kada će se skript izvršiti. Te akcije mogu biti, na primer, operacije čitanja ili pisanja na I²C magistralu, učitavanje 'vi2c' dokumenta, itd.

Tipičan primer korišćenja Tcl skripta je promena formata vrednosti – vrednost audio pojačanja se u registru kontrolera zapisuje kao celobrojna vrednost, dok korisnik ovu vrednost posmatra izraženu u količini decibela. Idealno bi bilo napisati skript koji preuzima celobrojnu vrednost koju program pročita, preračuna njegovu vrednost u iznosu decibela i kao takvu je prikaže.

B. Komunikacija sa uređajem

Program je projektovan tako da bude lako proširiv podrškom za nove uređaje i komunikacione sprege. Radi ovoga, on poseduje spregu ka dva skupa dinamički uvezivih biblioteka (Dynamic Linking Library – dll).

Prvi skup su programski rukovaoci I²C magistralom i oni poseduju funkcije za upis na i čitanje sa same I²C

magistrale.

Drugi skup je namenjen komunikaciji sa konkretnim uređajima i te biblioteke se nazivaju DAL (Dynamic Access Library) bibliotekama. DAL biblioteka je vrsta dinamički uvezivih biblioteka koja poseduje implementaciju određenih funkcija kojima program komunicira sa uređajem i ima ekstenziju 'dal'.

Biblioteke za komunikaciju sa platformom su pisane u programskom jeziku C. Pristup ovakvim bibliotekama se iz Java okruženja izvodi putem JNI (Java Native Interface – JNI) mehanizma [6], [7]. Za oba skupa dinamički uvezivih biblioteka postoji poseban modul zadužen za upravljanje istim. Dodatno, postoji i treći modul koji predstavlja spregu ka samom VI2C proširenju za Eclipse, a realizovan je kao "omotač" oko prva dva.

C. JNI mehanizam

Dinamički uvezive biblioteke se mogu koristiti samo u okviru operativnog sistema za koji su namenski pisane. Da bi mogla da pristupi funkcijama ovih biblioteka, Java Virtuelna Mašina podržava JNI mehanizam.

Da bi se iskoristio ovakav mehanizam prvo je neophodno definisati koje metode će sadržati Java klasa koja će pristupati funkcijama biblioteka. Dalje se, korišćenjem *javah* alata, generiše zaglavlje C modula koji treba implementirati. Pomenuti alat se nalazi u standardnom skupu alata koji se isporučuju uz razvojno okruženje za Java jezik (Java Development Kit – JDK). C modul se prevodi u biblioteku karakterističnu za operativni sistem na kome će se aplikacija izvršavati. S obzirom da su komunikacione biblioteke već prevedene i spremne za korišćenje, bilo je neophodno napraviti "omotač" biblioteku koja sadrži funkcije za upravljanje komunikacionim bibliotekama. Ova biblioteka podržava učitavanje i rad sa više DAL i rukovalac biblioteka. "Omotač" biblioteka sadrži tri modula:

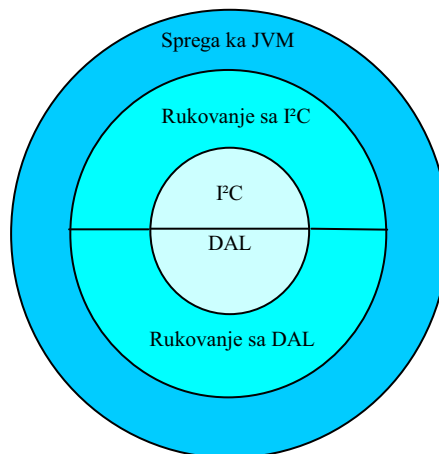
- sprega ka JVM, odnosno ka VI2C proširenju za Eclipse,
- rukovanje I²C rukovalac bibliotekama,
- rukovanje DAL bibliotekama.

Imajući u vidu relativnu sporost JNI mehanizma, "omotač" biblioteka je projektovana tako da se što manje podataka prenosi između Java i C kôda. Da bi se to postiglo, moduli za rukovanje sadrže listu struktura kojima je opisana biblioteka i funkcije za učitavanje biblioteka i podataka vezanim za njih. Sprega ka JVM sa jedne strane upravlja modulima za rukovanje komunikacionim i rukovalac bibliotekama, dok sa druge obezbeđuje funkcionalnosti potrebne aplikaciji. Na slici Sl. 2 je prikazana šema "obmotavanja" biblioteka za komunikaciju sa uređajima.

D. DAL biblioteke

Proširivost aplikacije za rad sa novim uređajima je realizovana veoma jednostavno. Da bi aplikacija mogla da radi sa novim uređajem, potrebno je samo učitati DAL biblioteku specifičnu za taj uređaj. Koja biblioteka će se učitati odlučuje korisnik kada se definišu parametri veze sa

uređajem.



Sl. 2. Šema "obmotavanja" komunikacionih biblioteka

Za izvestan broj uređaja DAL biblioteke već postoje i koisnik može da izabere neku od njih. Međutim, ukoliko nijedna od već postojećih biblioteka ne odgovara novom uređaju, ista se mora implementirati. DAL biblioteka je vrsta dinamički uvezivih biblioteka (dll). Uslov koji se postavlja pred projektante DAL biblioteka je taj da ona mora da sadrži implementaciju osam određenih funkcija. Dve najbitnije od pomenutih osam funkcija su funkcije za čitanje i upis podataka u registre kontrolera samog uređaja. Ostalih šest služe za ispravnu integraciju DAL biblioteka u VI2C proširenje za Eclipse.

E. RSE proširenje za Eclipse

RSE proširenje za Eclipse obezbeđuje osnovu i sadrži skup alata koji služe za konfigurisanje i upravljanje udaljenim sistemima (remote systems), ostvarenim vezama i njihovim uslugama. Pod udaljenim sistemima se podrazumevaju razvojne platforme sa kojima se, na bilo koji način, može ostvariti veza.

Veoma bitna karakteristika RSE proširenja za Eclipse je transparentnost. To, u ovom slučaju, znači da se svi udaljeni sistemi, koliko god da se oni međusobno razlikuju, korisniku programa prikazuju na jedinstven način. Korisniku je time omogućeno da sa udaljenim sistemima rukuje kao da se nalaze na istom računaru na kome radi. Da bi se ovo postiglo, RSE proširenje poseduje prikaz u Eclipse radnom okruženju koji implementira spregu sa korisnikom. RSE prikaz sadrži listu uspostavljenih veza. Nad sistemima sa kojima je veza uspostavljena mogu se pregledati usluge i resursi, izvršavati komande, upravljati njihovim procesima, itd.

Koncept Eclipse platforme, na jednostavan način, omogućava međusobnu saradnju svojih proširenja. Zahvaljujući tome, i VI2C proširenje za Eclipse koristi sve usluge koje obezbeđuje RSE proširenje bez potrebe da se ispočetka razvija podsistem za komunikaciju sa razvojnim platformama.

III. ZAKLJUČAK

Ovaj rad predstavlja program koji rešava problem komunikacije između računara i razvojne platforme po I²C magistrali. Program je implementiran u programskom jeziku Java i realizovan kao proširenje za Eclipse platformu – VI2C proširenje.

VI2C proširenje se putem I²C magistrale povezuje sa razvojnom platformom učitavanjem DAL biblioteka. Sa druge strane, interpretira ‘vi2c’ dokumente koji opisuju kontrolere razvojne platforme i akcije vezane za njih, a koje su definisane Tcl skriptovima. Takođe sadrži podsistem za spregu sa korisnikom putem kojeg se sadržaj registara kontrolera može čitati i/ili menjati i zatim prikazuje rezultate.

Prednost ovog u odnosu na slična rešenja je upravo činjenica da je program realizovan kao proširenje za Eclipse platformu. Svakom proširenju, uključujući i VI2C, se može dodati proizvoljna funkcionalnost. Takođe, može se implementirati drugo proširenje ili čak sasvim nova aplikacija koja koristi usluge VI2C proširenja. Sve ove mogućnosti su na raspolaganju bilo kojem projektantu.

A. Dalji razvoj

Prikazan program se može dalje proširivati novim uslugama. Za sada se sadržaj ‘vi2c’ datoteka i Tcl skriptova može menjati jedino pomoću tekst editora. Ovakav način modifikacije je podložan greškama i samim tim bi se ispravan rad programa doveo u pitanje. Stoga je jedan od primera budućeg razvoja implementacija editora ‘vi2c’ datoteka i Tcl skriptova, koji bi dozvoljavao samo određene akcije nad sadržajem datoteke i time smanjio mogućnost pojave greške.

Postojeći program omogućava čitanje i pisanje podataka u registre kontrolera razvojne platforme u realnom vremenu, ali samo dok je platforma povezana sa računarom. Da bi program mogao da se koristi za testiranje i obuku novih korisnika bilo bi dobro implementirati i simulator I²C saobraćaja.

Takođe, sadržaj registara kontrolera je dostupan samo dok je program aktivan. Po prestanku rada programa, sadržaj se gubi. Još jedna korisna funkcija koju bi program u budućnosti mogao da obezbedi je i omogućavanje trajnog čuvanja sadržaja registara kontrolera.

LITERATURA

- [1] *Eclipse Platform Technical Overview*, 2006 International Business Machines Corp.
- [2] *Rich Client Platform*, (http://en.wikipedia.org/wiki/Rich_Client_Platform), From Wikipedia – the free encyclopedia
- [3] *I2C*, (<http://en.wikipedia.org/wiki/I2C>), From Wikipedia – the free encyclopedia
- [4] E. R. Harold, “Processing XML with Java”, Elliotte Rusty Harold, 2002.
- [5] J. A. Zimmer, “Tcl/Tk for Programmers: With Solved Exercises that Work with Unix and Windows”, IEEE Computer Society, 2000.
- [6] S. Liang, “The Java™ Native Interface – Programmer’s Guide and Specification”, ADDISON – WESLEY, June 1999.
- [7] C. Batty, *Using The Java Native Interface*, Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada, October 2003.

ABSTRACT

This paper presents a software tool for communication between computer and arbitrary integrated circuit via I²C bus. The tool consists of a few related parts and there is a detailed functional description of every part. This tool is implemented as an Eclipse IDE plugin. Therefore, a short description of what Eclipse Platform is and how it works has been included. I²C bus is described as well. At the end, there is a short conclusion about tool usability and possible further development.

SOFTWARE TOOL FOR COMMUNICATION BETWEEN COMPUTER AND ARBITRARY INTEGRATED CIRCUIT VIA I²C BUS

Ivan Letvenčuk, Vladimir Maksović, Tomislav Maruna,
Nikola Teslić