

IMPLEMENTACIJA I KONFIGURACIJA ETHERCAT SISTEMA IMPLEMENTATION AND CONFIGURATION OF THE ETHERCAT SYSTEM

Mladen Knežić, Branko Dokić, Željko Ivanović, *Elektrotehnički fakultet, Univerzitet u Banjoj Luci*

Sadržaj - *Industrijske fieldbus mreže zasnovane na Ethernet tehnologiji sve češće se koriste za komunikaciju između uređaja na nivou polja (senzori i aktuatori) u sistemima automatizacije. Jedno od takvih rješenja je EtherCAT. U ovom radu opisuje se postupak implementacije i konfiguracije jednog sistema za brzi razvoj prototipa zasnovanog na EtherCAT protokolu. Posebna pažnja posvećena je realizaciji slejv uređaja, te instalaciji i konfiguraciji EtherLab softvera na master strani EtherCAT mreže.*

Abstract - *Industrial fieldbus networks based on Ethernet technology are most commonly used for communication between field devices (sensors and actuators) of the automation systems. One of these solutions is EtherCAT. This paper describes procedure for implementation and configuration of a rapid-prototyping system based on EtherCAT protocol. Special attention is given to the realization of a slave device, and installation and configuration of the EtherLab software on the master side of the EtherCAT network.*

1. UVOD

Industrijske komunikacione mreže zasnovane na *Ethernet* tehnologiji sve više su prisutne u današnjim sistemima automatizacije. Posljednja istraživanja pokazuju da je trenutno u svijetu instalirano preko 2 miliona čvorova koji kao komunikacioni mehanizam koriste neki od industrijskih protokola koji se bazira na *Ethernet*-u [1]. U studiji [2] je pokazano da i u privredi Republike Srpske postoji veliko interesovanje da se u bliskoj budućnosti instalira veći broj komunikacionih čvorova za povezivanje uređaja u polju (senzori i aktuatori) sa nadzornim i upravljačkim sistemom direktno preko *Ethernet* infrastrukture.

Neke od dobrih osobina *Ethernet*-a su njegova veoma široka zastupljenost u klasičnim računarskim mrežama što veoma utiče na cijenu infrastrukturne opreme, veoma velika baza znanja među inženjerima o ovoj tehnologiji, mogućnost lake integracije računarske mreže sa industrijskom komunikacionom infrastrukturom što omogućava povezivanje sektora menadžmenta sa sektorom proizvodnje, kao i neprekidan i veoma brz razvoj ove tehnologije (*Ethernet* sa protokom 10Gb/s je već komercijalno dostupan, a ubrzano se radi na razvoju terabitskog *Ethernet*-a).

Jedan od osnovnih zahtjeva svake nove *fieldbus* tehnologije je mogućnost lake konfiguracije i instalacije u cilju što ranijeg puštanja u rad sistema. Pored toga potrebno je obezbijediti bolji ili isti nivo pouzdanosti, cijene održavanja i sigurnosti uz povećanje brzine rada i smanjenje cijene instalacije opreme.

Fieldbus protokoli zasnovani na *Ethernet* tehnologiji su unijeli svojevrstan preokret u ovoj oblasti. Osnovna prednost ovih protokola je znatno veći protok podržan od strane *Ethernet*-a koji se može iskoristiti za prenos kako sinhronih procesnih podataka (PDO – *Process Data Object*), tako i asinhronih servisnih podataka (SDO – *Service Data Object*)

koji služe za definisanje i praćenje parametara u sistemu, kao i generisanje alarmnih poruka. Najznačajniji predstavnici protokola visokih performansi (vrijeme komunikacionog ciklusa manje od 1ms) koji se zasnivaju na *Ethernet*-u su: *EtherCAT*, *Profinet IRT*, *Powerlink* i *Sercos III*. Ovi protokoli se još nazivaju i industrijskim protokolima treće kategorije [3]-[5].

U ovom radu opisan je jedan sistem za brzo razvijanje prototipa zasnovan na *EtherCAT* protokolu. Kao softverska podrška koristi se *EtherLab* [6] koji omogućava generisanje kôda u okviru *Simulink/RTW* okruženja za izvršavanje u realnom vremenu na x86 hardverskoj platformi korišćenjem RTAI (*Real-Time Application Interface*) ekstenzije *Linux* operativnog sistema [7].

U okviru ovog softvera implementiran je *EtherCAT* master uređaj koji se sa slejv uređajima u sistemu povezuje preko standardne mrežne kartice u skladu sa pravilima *EtherCAT* protokola. Na taj način moguće je realizovati akvizicioni sistem visokih performansi koji u realnom vremenu izvršava kôd kojim se mogu realizovati različite funkcionalnosti (akvizicija podataka u realnom vremenu, *Hardware-In-the-Loop* sistem ili softverski realizovan PLC na industrijskoj PC platformi).

S obzirom da se za generisanje kôda koristi *Simulink/RTW*, realizacija sistema se vrši na visokom nivou kreiranjem odgovarajućeg modela, a zatim se prevodenjem njegove funkcionalnosti u odgovarajući programski kôd kompatibilan sa RTAI ekstenzijom dobijaju softverski moduli koji mogu da se izvršavaju u realnom vremenu.

U drugom poglavlju dat je pregled *EtherCAT* protokola i njegovih osnovnih karakteristika. Treće poglavlje opisuje postupak realizacije slejv uređaja u *EtherCAT* sistemu dok je u četvrtom poglavlju opisana instalacija i konfiguracija softvera na master strani koji je neophodan za funkcionisanje

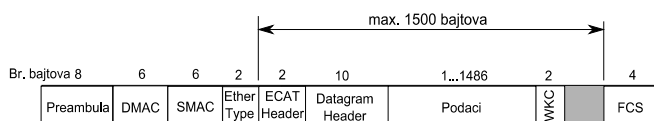
sistema. U petom poglavlju su sumirani rezultati i izvedeni osnovni zaključci.

2. ETHERCAT PROTOKOL

Princip komunikacije u *EtherCAT* sistemu zasniva se na prenosu procesnih i servisnih podataka unutar tzv. sumirajućeg okvira koji se enkapsulira u standardne *Ethernet* okvire. Veličina podataka u industrijskim mrežama uglavnom je malena (svega nekoliko bajtova), tako da komunikacija korišćenjem sumirajućeg okvira koji prenosi podatke za više uređaja, za razliku od prenosa korišćenjem individualnih okvira (svakom uređaju podaci se šalju u zasebnom okviru), daje veoma dobre performanse sistema i kratka vremena komunikacionog ciklusa u mreži.

EtherCAT mreža uvijek ima strukturu logičkog prstena koja ne zavisi od fizičke topologije. Na taj način postižu se dobre karakteristike u sistemu bez obzira na korišćenu topologiju mreže. Svu komunikaciju u sistemu inicira softverski realizovan master uređaj. Ovaj uređaj se povezuje sa slejv uređajima korišćenjem standardnog mrežnog interfejsa koji podržava *Ethernet* protokol. Okvir generisan u master uređaju prolazi kroz sve slejv uređaje u mreži, a zatim se od posljednjeg uređaja u logičkom prstenu vraća nazad do mastera. Procesiranje okvira se obavlja hardverski „u letu“, za vrijeme prolaska kroz slejv uređaj, i to samo u direktnom smjeru, dok se u povratnom smjeru okvir ne obrađuje. Na taj način su omogućena malena vremena propagacije kroz mrežu što znatno utiče na ukupne performanse sistema.

Struktura *EtherCAT* okvira prikazana je na Sl.1. Sa slike možemo vidjeti da se u standardne *Ethernet* okvire enkapsuliraju *EtherCAT* okviri koji se sastoje iz zaglavlja okvira (*ECAT Header*) veličine 2 bajta, zaglavlja datagrama (*Datagram Header*) veličine 10 bajtova, polja podataka koje može da bude veličine do 1486 bajtova (za jedan *Ethernet* okvir) i radnog brojača (*WKC – Working Counter*) veličine 2 bajta koji služi za dijagnostičke svrhe. U okviru 10 bajtova zaglavlja datagrama nalazi se informacija o adresi uređaja sa kojim se obavlja proces razmjene podataka.



Sl.1. Struktura *EtherCAT* okvira.

U *EtherCAT* protokolu postoje tri načina adresiranja:

- Poziciono adresiranje – uređaj se adresira na osnovu njegove pozicije u logičkom prstenu. Adresa zavisi od fizičke topologije. Ovo adresiranje najčešće se koristi pri inicijalizaciji ili rekonfiguraciji sistema;
- Čvorno adresiranje – uređaj se adresira na osnovu predefinisane adrese. Koristi se pri slanju komandnih riječi za konfiguraciju sistema i prenos statusnih informacija, kao i prenos servisnih podataka;
- Logičko adresiranje – ne vrši se fizičko adresiranje uređaja, već se pristupa jedinstvenoj logičkoj slici procesnog sistema. Uglavnom se koristi za sinhroni prenos procesnih podataka.

Pored adrese, u zaglavlju datagrama se nalazi i komanda koja definiše tok podataka (ulaz ili izlaz), kao i način adresiranja. U tipičnoj *EtherCAT* mreži za svaki uređaj potrebno je rezervirati jedan datagram za prenos podatka.

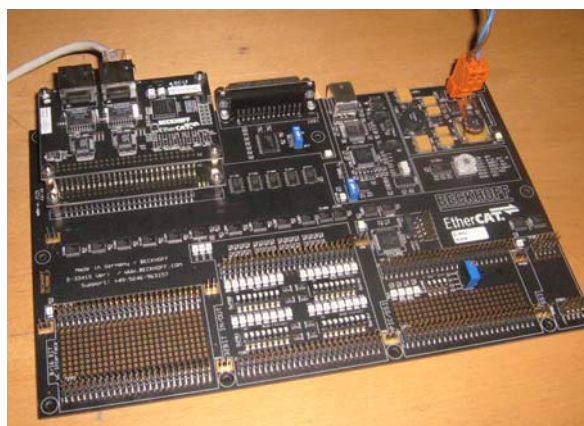
Ovakav pristup veoma brzo dovodi do značajnog *overhead*-a u slučaju velikog broja slejv uređaja u mreži. Sa aspekta razmjene procesnih podataka, najpogodniji način adresiranja predstavlja logičko adresiranje koje daje brojne mogućnosti za optimizaciju veličine okvira, a samim tim i vremena potrebnog za razmjenu podataka u mreži. Logičkim adresiranjem moguće je vršiti prenos procesnih podataka za sve uređaje u mreži u okviru samo jednog datagrama koji predstavlja logičku procesnu sliku sistema.

Pored opisanih karakteristika, *EtherCAT* protokol podržava znatan broj opcija za očuvanje konzistentnosti podataka, automatsku konfiguraciju i rekonfiguraciju sistema u slučaju promjene topologije mreže, visok nivo sinhronizacije u mreži (ispod 1 μ s) korišćenjem IEEE 1588 protokola [8], te funkcije sigurnosti i pouzdanosti. Vremena komunikacionog ciklusa koja mogu da se ostvare u *EtherCAT* mreži srednje veličine (do 100 slejv uređaja) mogu da budu ispod 100 μ s [5], [9].

3. REALIZACIJA SLEJV UREĐAJA

Svaki slejv uređaj u mreži se sastoji iz komunikacione jedinice (*ESC – EtherCAT Slave Controller*) koja je zadužena za implementaciju funkcija protokola na fizičkom i sloju toka podataka. Ova jedinica je realizovana hardverski u ASIC ili FPGA tehnologiji. Aplikacioni sloj (kao što je npr. *canOpen*) realizuje se u mikrokontroleru ili mikroprocesoru koji se sa *ESC* povezuje korišćenjem tzv. *PDI (Process Data Interface)* interfejsa. Softver u mikrokontroleru zadužen je za realizaciju svih funkcionalnosti slejv uređaja koje se odnose na sloj aplikacije u ISO/OSI modelu komunikacije.

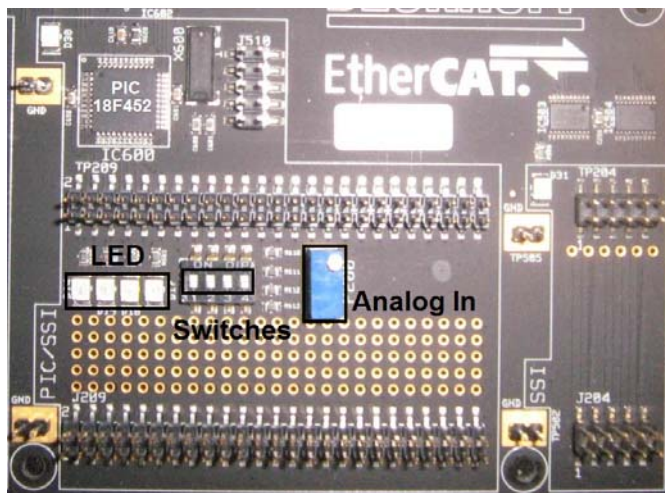
Za realizaciju slejv uređaja korišćena je razvojna ploča *EL9800* njemačkog proizvođača *Beckhoff* sa odgovarajućom softverskom podrškom za razvoj aplikacionog sloja prikazana na Sl.2. Kao mikrokontroler je iskorišćen *PIC18F452* koji je integrisan na razvojnoj ploči i koji se povezuje sa *ESC* kontrolerom preko *SPI* interfejsa.



Sl.2. Razvojna ploča *EL9800*.

Na razvojnoj ploči implementirana je aplikacija koja vrši nadzor i upravljanje nad 4 digitalna izlaza (LED diode na

razvojnoj ploči), 4 digitalna ulaza (prekidači na razvojnoj ploči) i jedan analogni ulaz koji se A/D konverzijom u mikrokontroleru pretvara u odgovarajuću 16-bitnu digitalnu riječ (promjenu analognog signala moguće je vršiti potenciometrom koji se nalazi na ploči). Dio ploče na kojem se nalaze pomenuti uređaji prikazan je na Sl.3.



Sl.3. Dio razvojne ploče EL9800 sa naznačenim ulaznim i izlaznim uređajima.

Aplikacioni softver u mikrokontroleru realizovan je u programskom jeziku C i podržava *canOpen* protokol za parametrizaciju i konfiguraciju sistema, kao i prenos procesnih podataka.

Za realizaciju prototipskog sistema u ovom radu je korišćen jedan bit digitalnog izlaza (*Relay*) i analogni ulaz (*Analog In*).

4. ETHERLAB SOFTVER

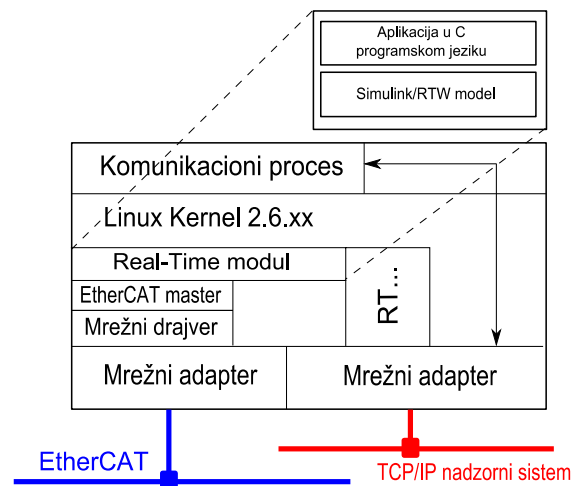
U *EtherCAT* mreži svu komunikaciju i upravljanje inicira i nadzire master uređaj. Master uređaj se realizuje softverski na *Embedded* ili PC platformi koji se sa mrežom se povezuje korišćenjem standardne mrežne kartice.

EtherLab softverski paket obuhvata skup aplikacija kojim se realizuje sistem za brzi razvoj prototipa baziran na *EtherCAT* protokolu. Ovaj skup aplikacija se sastoji iz sljedećih komponenata:

- *EtherCAT* master realizovan u skladu sa specifikacijama protokola;
- Mrežni drajver prilagođen za cikličnu razmjenu procesnih podataka;
- RTAI ekstenzija za *Linux* operativni sistem koja omogućava izvršavanje programskih modula u realnom vremenu;
- *Simulink/RTW* biblioteka blokova za *EtherCAT* slejv uređaje;
- *Testmanager* – *Windows* aplikacija za nadzor parametara u sistemu preko TCP/IP protokola.

Pojednostavljena arhitektura *EtherLab* sistema prikazana je na Sl.4. Sa slike možemo da vidimo da se aplikacija koja se izvršava u realnom vremenu kao programski modul može realizovati ili u programskom jeziku C ili na visokom nivou korišćenjem *Simulink/RTW* okruženja u kojem se nalaze

EtherCAT blokovi iz *EtherLab* biblioteke. Aplikacija koristi funkcije *EtherCAT* master softvera za komunikaciju sa slejv uređajima koji su povezani na mrežnu karticu.



Sl.4. Pojednostavljena arhitektura *EtherLab* sistema.

Mrežni drajver je posebno prilagođen za razmjenu cikličnih podataka tako što radi u režimu u kojem su onemogućeni prekidi u sistemu mrežne kartice a sva kontrola nad komunikacijom povjerena *EtherCAT* master softveru čime je zaobiden standardni mrežni stek protokola u *Linux*-u.

S druge strane, preko standardnog steka protokola (preciznije rečeno korišćenjem TCP/IP protokola), moguće je vršiti nadzor svih promjenljivih i parametara u sistemu.

4.1. Instalacija i konfiguracija sistema

S obzirom da je sistem baziran na *Real-Time* ekstenziji *Linux* operativnog sistema, najprije je potrebno izvršiti konfiguraciju i instalaciju posebno prilagođenog *Linux* kernela za rad u realnom vremenu. Kao aplikacioni interfejs za rad u realnom vremenu korišćen je RTAI, iako je moguće koristiti bilo koju drugu ekstenziju kao što je *Xenomai*, *RTLinux* itd. Sistem čak može da funkcioniše i bez bilo kakve *Real-Time* ekstenzije ukoliko je to u skladu sa zahtjevima aplikacije. Detaljna uputstva oko konfiguracije u instalacije kernela mogu se naći u odgovarajućoj dokumentaciji [6], [7].

Nakon instalacije modifikovanog kernela, slijedi instalacija *EtherCAT* master softvera. *EtherCAT* master se instalira kompajliranjem izvornog kôda i instalacijom odgovarajućih modula sa nekoliko standardnih *Linux* komandi:

```
cd $TARGET/src/ethercat/
./configure --enable-r8169
make all modules
make install, modules_install
depmod
```

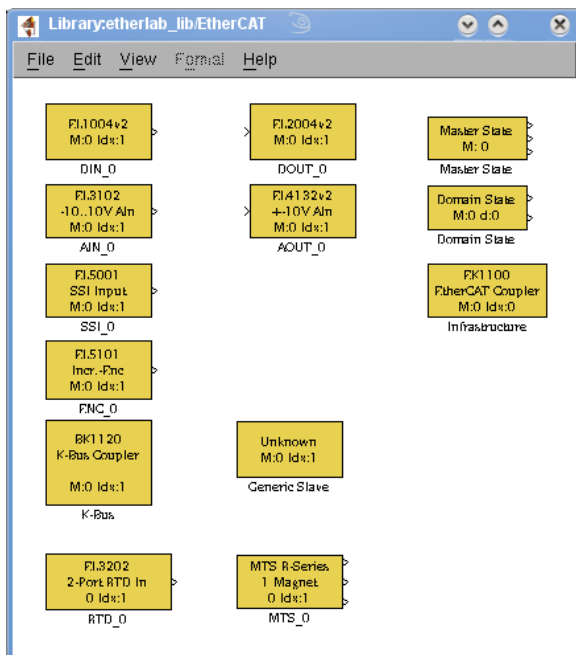
Prilikom konfiguracije potrebno je definisati parametre kao što su naziv drajvera mrežne kartice koja se koristi u sistemu (u konkretnom slučaju *r8169*), putanju do izvornog koda za *Linux* kernel koji se koristi i putanju do direktorijuma u kojem je instaliran RTAI ukoliko sistem treba da radi u

realnom vremenu. Opciono, dodatno se može uključiti i virtuelni mrežni interfejs za debugovanje master komunikacije. Nakon toga, potrebno je editovati konfiguracioni fajl tako da se podesi MAC adresa mrežne kartice koja će se koristiti za *EtherCAT* komunikaciju, kao i tip uređaja, tj. naziv drajvera odgovarajuće mrežne kartice. U konkretnom slučaju korišćena je *Realtek RTL8101* mrežna kartica sa drajverom `r8169`. Softver se pokreće i zaustavlja pomoću posebnog skript-fajla pod nazivom `ethercat` koji se nalazi u direktorijumu u kojem je softver instaliran.

EtherLab softver za generisanje programskih modula koji se izvršavaju u realnom vremenu u *Simulink/RTW* instalira se na sličan način kao i *EtherCAT* master softver. Jedina razlika je u tome što je nakon instalacije potrebno pokrenuti u *Matlab*-u skript-fajl pod nazivom `setup_etherlab` koji instalira biblioteku *EtherCAT* blokova neophodnu za realizaciju modela. Da bi programski moduli mogli da se izvršavaju potrebno je da bude pokrenut kernel modul `rt_appcore` koji je sastavni dio *EtherLab* softvera. Ovaj modul se u *Linux*-u pokreće i zaustavlja pomoću skript-fajla pod nazivom `etherlab`. Nakon toga, aplikacija koju smo dobili kompajliranjem u *Simulink/RTW* može se pokrenuti pomoću `insmod` komande.

4.2. Kreiranje i kompajliranje Simulink/RTW modela

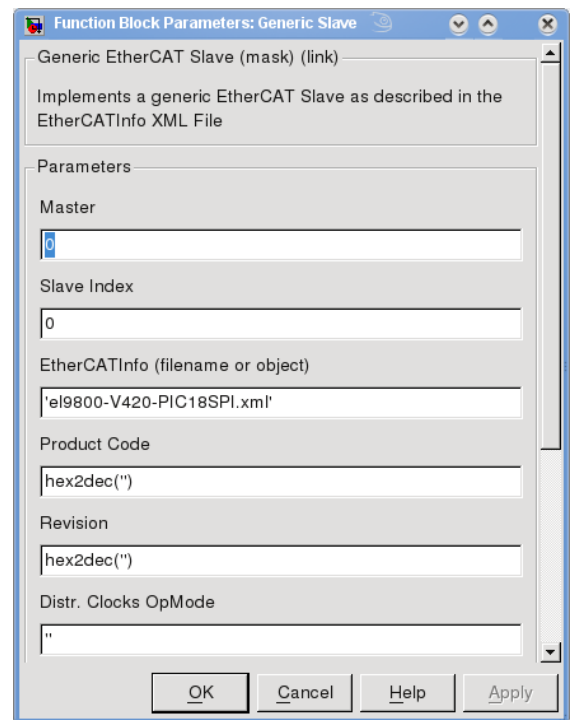
Prilikom kreiranja modela mogu se koristiti standardni blokovi u *Simulink*-u koji se povezuju sa blokovima iz *EtherCAT* biblioteke u cilju realizacije odgovarajućeg algoritma aplikacije. Sadržaj biblioteke *EtherCAT* blokova prikazan je na Sl.5. Ova biblioteka se poziva u *Matlab*-u pomoću komande `etherlab_lib` i podržava sve slejv uređaje koji se danas najčešće koriste na tržištu.



Sl.5. Biblioteka *EtherCAT* blokova u *Simulink*-u.

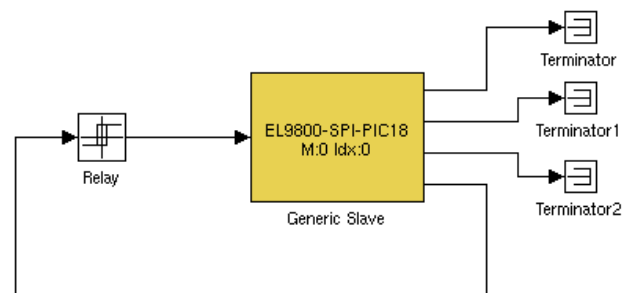
Pored predefinisanih blokova slejv uređaja, u okviru ove biblioteke se nalazi i blok kojim se definiše generički slejv uređaj čija je funkcionalnost opisana odgovarajućim XML fajlom koji se dobija od proizvođača, a koji se specificira u

okviru prozora za podešavanje parametara generičkog slejv uređaja (polje *EtherCATInfo* na Sl.6).



Sl.6. Prozor za podešavanje parametara generičkog slejva.

U slučaju razvojne ploče EL9800 korišćen je XML fajl koji opisuje strukturu aplikacije slejv uređaja povezanog sa ESC preko SPI interfejsa. Kao primjer aplikacije, kreiran je jednostavan model prikazan na Sl.7. U ovom modelu je realizovana relejna funkcionalnost izlaza *Relay* (četvrt LED dioda sa Sl.3), pri čemu se stanje (dioda uključena ili isključena) mijenja na osnovu vrijednosti analognog ulaza *Analogue In* sa gornjim pragom 800 i donjim pragom 600.



Sl.7. Izgled modela realizovanog u *Simulink*-u.

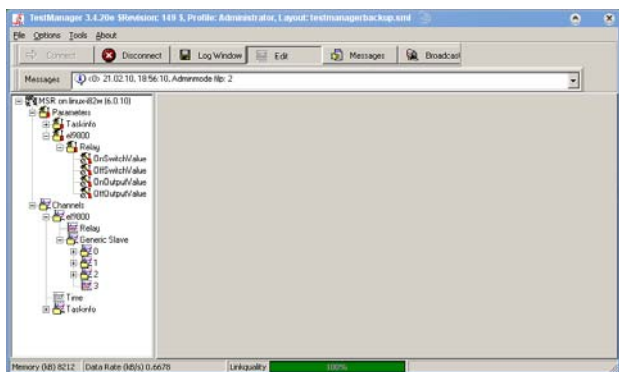
Prije nego što se pristupi postupku generisanja i kompajliranja kôda potrebno je podesiti radno okruženje *Simulink/RTW* za kreirani model. Najprije je potrebno u meniju *Simulation/Configuration Parameters...* podesiti opciju *Realtime Workshop/System target file* tako da bude odabran `etherlab.tlc`. Nakon toga, potrebno je definisati osnovni period vremena odmjerenja u okviru opcije *Solver/Fixed-step size*. U konkretnoj aplikaciji odabran je period od 0.001s ili 1000Hz. Ostaje još da se pokrene kompajliranje modela u *Real-Time Workshop* okruženju (npr. kombinacijom tastera `Ctrl+B`). Kao rezultat dobija se istoimeni kernel modul u istom direktorijumu u kojem se nalazi i model. Pokretanje programskog modula se inicira izvršavanjem komande `insmod` kojoj se kao parametar

prosljeđuje naziv modula. Da bi ovaj modul mogao da se izvršava, prije njegovog pokretanja moraju da budu pokrenuti *EtherCAT* master i *rt_appcore* moduli.

4.3. Aplikacija za nadzor sistema

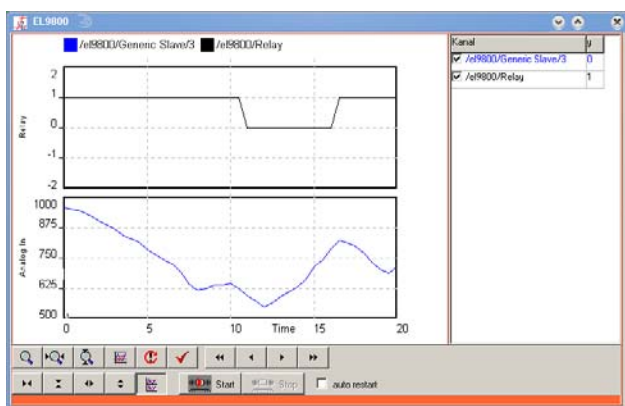
U okviru *EtherLab* softvera moguće je vršiti nadzor *EtherCAT* sistema na istom računaru na kojem se izvršava upravljački algoritam ili na udaljenoj radnoj stanici za nadzor koja se sa upravljačkim sistemom povezuje preko TCP/IP protokola. U tu svrhu služi *Windows* aplikacija pod nazivom *Testmanager*. Da bi nadzor bio omogućen, na upravljačkoj radnoj stanici najprije treba pokrenuti modul pod nazivom *etherlab_buddy* koji služi kao interfejs za komunikaciju između upravljačkog i nadzornog sistema.

Testmanager aplikacija omogućava praćenje i grafički prikaz svih promjenljivih u sistemu. Komunikacija se obavlja preko TCP porta 2345 pomoću posebno razvijenog XML baziranog MSR protokola [6]. Glavni prozor *Testmanager* aplikacije za prethodno opisani sistem prikazan je na Sl.8.



Sl.8. Izgled glavnog prozora *Testmanager* aplikacije.

Ova aplikacija posjeduje niz grafičkih kontrola za praćenje i definisanje ulaznih i izlaznih promjenljivih i prametara sistema. Na Sl.9 prikazan je izgled *Scope* kontrole na kojoj se prati promjena analognog ulaza *Analog In* i relejnog izlaza *Relay* u realnom vremenu. U okviru kontrole može se definisati način grafičkog prikaza, broj aktivnih kanala, vrijeme praćenja stanja promjenljivih u sistemu itd.



Sl.9. Izgled prozora u kojem se prati stanje promjenljivih.

Pored ove, postoji još niz drugih kontrola koje se standardno koriste u modernim SCADA sistemima. U okviru aplikacije osim nadzora, moguće je vršiti i kontrolu nad upravljačkom radnom stanicom, definisati načine pristupa i nivo privilegija za svakog korisnika, te pokretati i zaustavljati izvršavanje upravljačkog algoritma.

5. ZAKLJUČAK

U radu je prezentovan sistem za brzi razvoj prototipa zasnovan na *EtherCAT* protokolu i *Simulink/RTW* okruženju za modelovanje, razvoj i testiranje upravljačkih sistema. Na jednostavnom primjeru pokazan je postupak instalacije i konfiguracije *EtherCAT* sistema sa podrškom *EtherLab* softvera, a zatim su opisane njegove mogućnosti i oblasti primjene.

LITERATURA

[1] ARC Advisory Group, „Ethernet-Based Device Networks Worldwide Outlook: Five Year Market Analysis and Technology Forecast Through 2012“, preuzeto sa <http://www.arcweb.com>, Januar 2010.

[2] B. Dokić, M. Knežić, Ž. Ivanović, T. Pešić-Brđanin, i B. Blanuša, „Elaborat: Implementacija i testiranje EtherCAT sistema“, završni izvještaj projekta sufinansiranog od strane Ministarstva nauke i tehnologije RS, Novembar 2009.

[3] J. Jasperneite, J. Imtiaz, M. Schumacher, K. Weber, “A Proposal for a Generic Real-Time Ethernet System”, *IEEE Trans. on Industrial Informatics*, vol. 5, no. 2, pp. 75-85, May 2009.

[4] M. Felser, T. Sauter, “Standardization of industrial Ethernet – The next battlefield”, *Proc. of the WFCSS2004, IEEE Int. Workshop on Factory Communication Systems*, pp. 413-420, September 2004.

[5] J. Jasperneite, M. Schumacher, K. Weber, “Limits of increasing the performance of industrial Ethernet protocols”, *IEEE Conf. on Emerging Technologies and Factory Automation, Patras, Greece*, pp. 17-24, September 2007.

[6] EtherLab. An Open Source Toolkit for rapid real time code generation under Linux® using Simulink®/RTW® and EtherCAT® Technology. www.etherlab.org.

[7] RTAI. Real-Time Application Interface. www.rtai.org.

[8] IEEE. IEEE 1588, „Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems“, 2008.

[9] G. Prytz, “A performance analysis of EtherCAT and PROFINET IRT”, *IEEE Conf. on Emerging Technologies and Factory Automation, Hamburg, Germany*, pp. 408-415, September 2008.