

Praktična realizacija analizatora “CAN” protokola

Filipović Marko
Telopo
Beograd, Srbija
marko.f@msn.com

Mijalković Milan
VISER
Beograd, Srbija
milan.mijalkovic@viser.edu.rs

Sadržaj—Ovaj dokument predstavlja kratak opis praktične realizacije analizatora CAN protokola. Sam protokol, njegova primena, korišćeni mikrokontroler i njegove periferije su takođe opisani. Analizator se može koristiti za razvoj novog sistema, ali i za praćenje rada nekog već postojećeg. Može da vrši neprimetno praćenje saobraćaja na magistrali ali i da slanjem poruka aktivno učestvuje u radu sistema. On predstavlja interfejs između CAN magistrale i standardnog RS-232 porta na računaru.

Ključne reči—CAN protokol; CAN-BUS; magistrala; analizator; monitor; interfejs;

I. UVOD

CAN (Controller Area Network) protokol ili CAN-BUS kako se još označava je standard za jako pouzdanu serijsku komunikaciju na magistrali, baziranu na porukama. Prvobitna namena mu je bila komunikacija između mikrokontrolera i drugih elektronskih uređaja u vozilima, ali je to danas bitno prošireno pa se CAN često nalazi i u industrijskoj automatiki, sistemima kućne automatike i medicinskoj opremi.

Zbog široke prisutnosti cena CAN kontrolera i transivera je prilično niska. U zadnjih par godina primetan je trend ugradnje CAN periferija i u jako jeftine proizvode. Moderni automobili imaju ugrađeno i po 70 CAN uređaja. Procenjuje se da je do danas u svetu proizvedeno više stotina miliona kontrolera. Iako je kao standard CAN-BUS star više od 20 godina, čini se da tek sad dobija zamah i da u raznim oblastima automatike potiskuje RS-485 i Modbus. Takođe, zbog mogućnosti da se korišćenjem CAN-a premoste velika rastojanja između senzora i kontrolera (na nižim brzinama i preko 1 km) pogodan je za daljinsko očitavanje raznih vrsta senzora, ali i za upravljanje udaljenim izvršnim uređajima - aktuatorima.

Kod praktične realizacije analizatora jako je bitno imati u vidu njegovu namenu. Kada je u pitanju dijagnostika na vozilima, tu već postoji veliki broj gotovih, specijalizovanih uređaja i specijalizovanih softverskih paketa koji ih prate. Pravljenje novog uređaja i pratećeg softvera bez adekvatne podrške nekog od proizvođača vozila bilo bi teško izvodljivo i ekonomski neisplativo, zbog toga je prvenstvena namena ovog analizatora u oblasti industrijske i kućne automatike. Sa ovim uređajem je moguće pratiti saobraćaj ali i slati poruke na svakoj standardnoj CAN magistrali, nezavisno od brzine, tipa povezanih uređaja ili protokola na višem nivou.

Kako je ovaj uređaj prvenstveno namenjen razvoju i praćenju rada industrijskih sistema odnosno sistema kućne

automatike, pretpostavka je da će se korišćenje odvijati u industrijskim halama ili stambenim zgradama. Imajući ovo u vidu korisnički interfejs nije postavljen na samom uređaju već je predviđeno da se za ovu namenu koristi PC računar (najčešće notebook tipa). Ovakvo rešenje za posledicu ima prostiji razvoj, manju cenu uređaja i konformniji rad zbog brzine izvršavanja aplikacija i veličine ekrana na PC računarima.

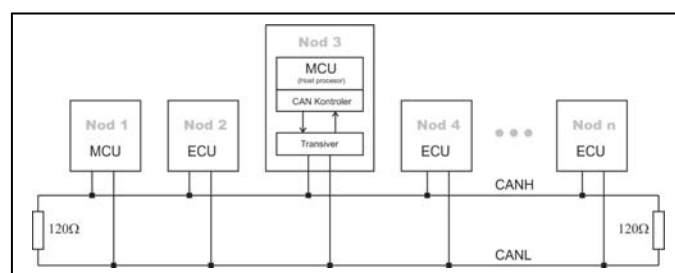
II. CAN PROTOKOL

A. Osnovno o protokolu

Kreiran je i patentiran sredinom osamdesetih godina (1983-1986) od strane nemačke firme Robert Bosch GmbH. Prve CAN kontroler čipove su 1987. godine po originalnoj specifikaciji proizveli Intel i Philips. Danas je aktuelna Bosch-ova CAN 2.0 specifikacija [1] koja datira još iz 1991. godine, ali se umesto nje često koristi međunarodni standard ISO 11898.

To je dobro definisan fieldbus protokol za real-time asinhroni serijski prenos podataka između entiteta koji se nazivaju nodovima i koji su povezani preko fizičkog transmisionog medijuma tj. magistrale (bus). Optimizovan za pouzdan prenos male količine podataka, a komunikacija je bazirana na porukama sa identifikatorima a ne na adresama.

Senzori i aktuatori se ne povezuju direktno na magistralu već preko ECU kontrolne jedinice. Za komunikaciju od jednog noda ka bilo kom drugom nodu, ili ka svim povezanim nodovima koristi se serijska magistrala (Sl. 1) sa svim ravnopravnim učesnicima (multi-master broadcast). Ovo znači da svaki nod preko magistrale može da šalje i prima serijske podatke ali da se slanje ne može obavljati simultano tj. veza je Polu-dupleks tipa.



Slika 1. Primer povezivanja uređaja na magistralu.

B. Fizički sloj

Kao i većina drugih mrežnih sistema, i CAN primenjuje slojeviti pristup. Na fizičkom nivou je moguće korišćenje različitih komunikacionih medijuma (upredene parice, jednožična linija, optički kabal, elektro instalacije).

Fizički medijum koji se najčešće koristi je balansirana upredena bakarna parica sa zajedničkim povratnim vodom. Moguće je komunicirati i preko samo jedne žice, što je vid komunikacije koji primenjuje firma General Motors u svojim automobilima. Eksperimentisano je i sa komunikacijom preko električnih vodova i preko radio talasa, ali su te varijante uglavnom napuštene. Optički kablovi su jedini medijum koji se pored bakarnih parica u praksi češće koristi, uglavnom za povezivanje dve ili više udaljenih žičanih CAN magistrala.

C. Arbitriranje

CAN protokol koristi nedestruktivni CSMA/BA (Carrier Sense Multiple Access/Bitwise Arbitration) metod pristupa što je modifikovana varijanta CSMA/CD tehnike pristupa koja se koristi kod Ethernet mreža. Svaki nod je projektovan tako da čuje sebe ali i druge nodove dok šalje poruku. Ovo je neophodno za nesmetan rad mehanizama koji određuju prioritet i obrađuju greške. Kada dođe do perioda bez aktivnosti na magistrali svaki od nodova ima podjednako pravo da šalje poruku. Ako dva noda počnu da šalju poruku u isto vreme jedan od njih će detektovati koliziju i prekinuti pa poruka višeg prioriteta ostaje nepromenjena.

Iz Tabele I. se vidi da da je stanje na magistrali određeno I (AND) logičkom funkcijom, i da bit koji je u dominantnom stanju uvek pobeđuje recisivni bit pri arbitriranju. Što je niža vrednost u polju ID viši je prioritet poruke, ovo obezbeđuje punu prohodnost porukama visokog prioriteta.

TABELA I. STANJA KADA DVA NODA VRŠE SLANJE ISTOVREMENO

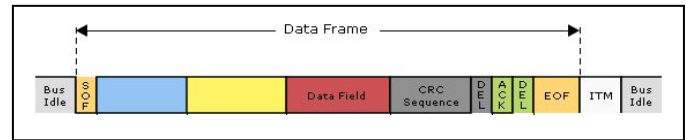
Logička stanja na magistrali		
	Dominantno (0)	Recesivno (1)
Dominantno (0)	Dominantno (0)	Dominantno (0)
Recesivno (1)	Dominantno (0)	Recesivno (1)

D. Poruke i ramovi

Poruke su kratke i pakuju se u jedan ili u više ramova. Ram može biti standardni ram sa podacima (data frame) prikazan na Sl. 2 ili ram sa zahtevom (remote frame). Sve poruke imaju dodeljeni prioritet tako da kritične poruke imaju prednost. Kratka dužina poruka stvara čestu priliku za slanje nove poruke. Maksimalno kašnjenje poruka visokog prioriteta pri brzini od 1 Mb/s je manje od 120 μs.

Svi nodovi u sistemu primaju svaku poruku koja se prenosi magistralom i koristeći ACK bit potvrđuju da je poruka primljena bez greške. Svakom nodu u sistemu je ostavljeno da samostalno odluči da li da primljenu poruku momentalno odbaci ili da je zadrži radi daljeg procesiranja. CAN poruku u osnovi čine prioritet (ID) i podaci. Količina podataka koju svaki ram nosi je određena DLC kontrolnim poljem i kreće se u rasponu od 0 do 8 bajtova.

Poruka može biti namenjena jednom pojedinačnom nodu (node-to-node) ili većem broju nodova (multicast) što zavisi samo od toga kako je sistem projektovan.



Slika 2. Format rama sa podacima (data frame).

III. RAD ANALIZATORA

Kompletan program na strani mikrokontrolera je napisan u standardnom C jeziku [2] koristeći okruženje ARM IAR Embedded Workbench [3] firme IAR Systems.

Svaki ram koji se pojavi na magistrali biva pročitan i analiziran. Čitanje se obavlja kroz bafer kapaciteta 8 poruka. Prilikom analiziranja se utvrđuje da li ram sadrži podatke ili je u pitanju ram sa zahtevom (RTR). Utvrđuje se identifikator (ID), broj bajtova podatka (DLC), koji su to bajtovi koje ram nosi, kao i tačno vreme prijema rama. Na osnovu tako razloženog CAN rama se formira jedan niz ASCII karaktera (string), koji se šalje preko RS-232 veze (UART 0 perifერიја) na PC računaru.

Analizator pored praćenja i prikaza saobraćaja na magistrali ima i mogućnost slanja poruka - dvosmerna komunikacija. Zbog toga je on u komunikaciji potpuno ravnopravan sa svim ostalim učesnicima (nodovima). Analizatoru se preko RS-232 veze šalju komande kao nizovi ASCII karaktera (stringovi). Te komande on tumači i u zavisnosti od vrste na njih adekvatno reaguje. Ukoliko primljena komanda nije validna, ili nije dozvoljena u tom trenutku, analizator šalje poruku o grešci - <BEL> karakter. Ukoliko je komanda validna, sledi izvršenje i izveštaj u uspešnom izvršenju - <CR> karakter.

Sva interakcija sa analizatorom se obavlja slanjem i primanjem ASCII stringova preko perifерије UART0. Da bi povezivanje bilo uspešno, potrebno je u komunikacionom programu (terminal emulatoru) podesiti parametre serijske veze. Podesava se komunikacioni (COM) port, brzina komunikacije, broj bitova za podatke, broj stop bitova kao i parnost.

COM port se bira na osnovu toga kako je i gde uređaj fizički priključen. To može biti neki od fizički postojećih COM portova, ali može biti i virtuelni COM port ako se koristi neki od USB-Serial adaptera (FTDI recimo). Broj bitova za podatke je uvek 8, broj stop bitova je 1, a bit parnosti se ne uzima u obzir (none).

Brzina komunikacije može da bude jedna od 4 ponuđene brzine:

1. 9600 bps (binarno 0)
2. 38400 bps (binarno 1)
3. 57600 bps (binarno 2)
4. 115200 bps (binarno 3)

Koja će se od ove 4 brzine koristiti određuje se na samom uređaju pritiskom na taster B1 (P0.15). Jedan kratak pritisak na ovaj taster ne menja brzinu komunikacije na UART portu već služi samo za davanje informacije o tome koja je brzina trenutno selektovana. Tek na sledeći pritisak istog tastera sistem reaguje promenom brzine. Brzina komunikacije se tada menja na sledeću brzinu u nizu. Informaciju o izabranoj brzini moguće je očitati na dva LED indikatora žute i zelene boje koji se nalaze na razvojnoj pločici. Ovi indikatori u normalnom radu imaju drugu funkciju, ali se po pritisku tastera B1 njihova funkcija menja i oni u trajanju od 4 sekunde pokazuju izabranu brzinu prenosa. Brzina je prikazana binarno tako što žuti LED pokazuje MSB, a zeleni LSB.

Posle uspešnog konektovanja analizatoru je moguće slati komande. Svaku poslatu komandu potrebno je završiti sa <CR> (Carriage Return - ASCII 13) karakterom. Na poslate komande analizator reaguje sa <CR> ako je komanda uspešno izvršena ili sa <BEL> (Bell - ASCII 7) ako izvršenje komande nije uspeo.

Analizator reaguje na sledeće komande:

O <CR>	- Otvaranje CAN kanala
L <CR>	- Otvaranje CAN kanala – samo slušanje
C <CR>	- Zatvaranje CAN kanala
V <CR>	- Verzija firmvera
N <CR>	- Serijski broj analizatora
I <CR>	- Trenutno stanje registara
Y <CR>	- Stanje registara u trenutku greške
Sn <CR>	- Brzina prenosa (n = 1-8)
Zn <CR>	- Prikaz vremena prijema (n = 0-1)

Formati ramova su:

iiiipppppppp	- Standardni ram
riii0	- Ram sa zahtevom (RTR)

Malo slovo t ili malo slovo r se šalju kao karakteri posle kojih idu parametri. Kod rama sa zahtevom d parametar mora uvek biti 0. Parametri imaju sledeće značenje:

iii	- Identifikator od tri heksadecimalne cifre
d	- DLC, Broj bajtova u delu za podatke
pppppppp	- Do 8 bajtova sa podacima

Kada analizator prikazuje ramove koji su stigli sa magistrale, koristi se isti format kao i kod slanja ramova. Znači, za ram sa podacima se koristi format: iiii0pppppppp, a za ram sa zahtevom format: riii0.

Opisani analizator je u velikoj meri kompatibilan sa analizatorom USB-CAN nemačke firme VS Com. Zajedničke su im upravljačke komande, kao i formati poruka. To omogućuje korišćenje ovog analizatora na svim mestima gde je moguće koristiti analizator firme VS Com. Svi programi za analizu napisani da rade sa VS Com analizatorom se mogu nesmetano koristiti.

Funkcionisanje analizatora se postiže stalnim pozivanjem dve funkcije čiji je rad opisan ispod. Obe ove funkcije se pozivaju prilikom svakog prolaza glavne petlje programa.

Funkcija `Prosled_i_Can_na_UART()` se izvršava neprekidno u glavnoj petlji. Njena uloga je da proverava da li je novi CAN ram stigao, i ako jeste, da ga prihvati, obradi i pošalje na UART kao string. Obrada se sastoji u tome da se svaki CAN ram razloži na sastavne delove i svaki deo predstavi ASCII karakterima.

Prilikom obrade prolaze se sledeći koraci:

- 1) Proverava se RTR bit i utvrđuje se da li se radi o standardnom ili ramu sa zahtevom. Ukoliko je u pitanju standardni ram prvi karakter će biti 't', a ukoliko je u pitanju ram sa zahtevom prvi karakter će biti 'r'.
- 2) Utvrđuje se identifikator (ID) i predstavlja se sa tri ASCII karaktera od kojih svaki označava jednu heksadecimalnu cifru.
- 3) Dodaje se broj bajtova sa podacima (DLC).
- 4) Dodaju se podaci koji se nalaze u delu za podatke pristiglog rama. Svaki bajt je predstavljen sa po dva ASCII karaktera od kojih svaki označava jednu heksadecimalnu cifru tj. 4 bita
- 5) Dodaje se na kraju tačno vreme prijema (time stamp) ako je ta opcija uključena komandom Zn

Kada se cela poruka primi tj. pošalje sa CAN-a na UART, na kraju se funkcijom `Posalji_Karakter_UART0(13)` šalje i jedan <CR> karakter koji označava kraj poruke. Nakon primljene poruke žuti LED indikator B1 (P0.15) trepne kako bi signalizirao aktivnost na magistrali.

Funkcija `Procitaj_i_Parsiraj_UART0()` se takođe neprekidno izvršava u glavnoj petlji. Njen zadatak je da čita komande koje korisnik zadaje analizatoru (iznad su opisane) i da na njih reaguje ako su validne ili da na UART pošalje karakter <BEL> ako nisu validne. Reagovanje na komande je u suštini pozivanje odgovarajućih funkcija i formiranje parametara za te funkcije.

Slanje nekog proizvoljnog rama na magistralu se obavlja tako što funkcija `Procitaj_i_Parsiraj_UART0()` na osnovu početnog slova (t ili r) utvrđi da li se radi o standardnom ramu ili ramu sa zahtevom i u zavisnosti od toga pozove neku od sledeće dve funkcije: `Formiraj_i_Posalji_STD()` ili `Formiraj_i_Posalji_RTR()`.

Obe ove funkcije prvo obavljaju proveru validnosti stringa koji im je prosleđen kao parametar i koji predstavlja buduću CAN poruku, zatim formiraju tu CAN poruku, i na kraju vrše slanje. Proverava se da li je identifikator u dozvoljenom opsegu (0 < ID < 2047), proverava se da li DLC polje sadrži samo dozvoljene vrednosti (0-8) i da li su podaci ispravni. Nakon ove provere CAN poruka se formira i šalje. Kada je poruka poslata, na UART se šalje karakter 'z' i iza njega karakter <CR>. Ova dva karaktera signaliziraju uspešno poslatu poruku. Nakon poslate poruke žuti LED indikator B1 (P0.15) trepne kako bi signalizirao aktivnost na magistrali.

IV. MIKROKONTROLER

Za hardversku realizaciju analizatora izabran je 32-bitni mikrokontroler LPC2129 [4], [5] firme NXP Semiconductors baziran na ARM7 TDMI-S sintetičkom jezgru. Serija LPC2000 mikrokontrolera je jedna od najraznovrsnijih i najdugotrajnijih serija ARM mikrokontrolera koja se u praksi jako dobro pokazala. Odlikuje se velikim izborom periferija na čipovima.

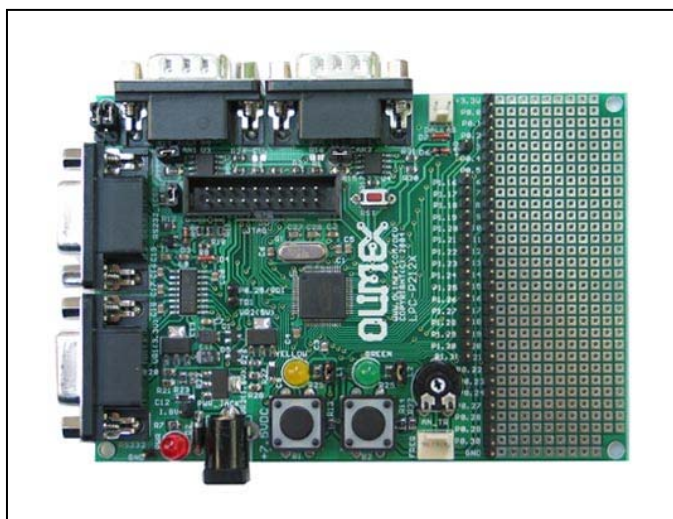
Konkretan model LPC2129 je izabran zato što poseduje dve CAN periferije, što je relativno jeftin, i zato što na tržištu postoji više gotovih razvojnih pločica sa ovim mikrokontrolerom (Sl. 3). Za realizaciju je korišćen proizvod LPC-P2129 [6] firme Olimex Ltd.

Neke od osnovnih karakteristika su kombinovani 16-bitni / 32-bitni način rada (ARM/Thumb), 16 kB statičkog RAM-a i 256k kB Flash memorije na samom čipu, maksimalna brzina sa PLL-om od 60 MHz, zaštita od neovlašćenog čitanja programa, 46 GPIO 5V pinova, nekoliko vrsta serijske veze (CAN, UART, I2C, SPI), 10-bitni A/D konverter, napredna kontrola prekida.

Sam ARM7 TDMI-S [7] je 32-bitni mikroprocesor opšte namene, koji se odlikuje visokim performansama i vrlo malom potrošnjom. ARM arhitektura se zasniva na RISC (Reduced Instruction Set Computer) principima, pa su zbog toga set instrukcija i mehanizam za dekodovanje mnogo prostiji u odnosu na CISC mikroprocesore. Ova jednostavnost za rezultat ima brz protok instrukcija kroz jezgro i impresivan odziv u realnom vremenu kod prekida.

Na samom čipu se nalazi 16kB statičke RAM memorije (SRAM) i 256kB Flash memorije. Obe ove vrste memorije se mogu koristiti za smeštanje koda i podataka. Kompletna Flash memorija je oslobođena za unos korisničkog koda i korisničkih podataka jer se Boot Loader nalazi na posebnoj lokaciji.

Programiranje se može obaviti preko JTAG interfejsa, preko UART periferije ili iz same aplikacije koristeći IAP funkcije.



Slika 3. Izgled razvojne pločice Olimex LPC-P2129

V. KORIŠĆENE PERIFERIJE

Opisani analizator u svom radu intenzivno koristi više različitih periferija i funkcija mikrokontrolera LPC2129. Pored standardnih VIC, PCB i PLL, koriste se još četiri periferije: GPIO, CAN, UART i TIMER.

A. CAN periferija

CAN periferija ovog mikrokontrolera [8] je napravljena tako da podrži komunikaciju na dve CAN magistrale istovremeno. Ovo dozvoljava uređaju da bude korišćen kao gejtvaj (gateway), svič (switch) ili ruter (router) između dve CAN magistrale.

Oba CAN kontrolera imaju strukturu registara jako sličnu strukturi 8-bitnog NXP SJA1000 nezavisnog CAN kontrolera. Osnovna razlika u odnosu na SJA1000 je to što je mehanizam filtriranja (acceptance filter) izmešten iz CAN periferije u posebnu periferiju.

Korišćeni pinovi:

CAN1: RD1 (P0.25) i TD1 (fiksni)
CAN2: RD2 (P0.23) i TD2 (P0.24)

Registri koji su od značaja:

CANMOD	- Kontroliše režim rada CAN kontrolera
CANBTR	- Određuje brzinu komunikacije
CANCMR	- Komandni bitovi
CANIER	- Uključivanje prekida
CANICR	- Status prekida
CANGSR	- Globalni status kontrolera
CANTFII	- Kontrolni bitovi poruke koja se šalje
CANTID1	- Identifikator (ID) poruke koja se šalje
CANTDA1	- Prvi deo podataka poruke koja se šalje
CANTDB1	- Drugi deo podataka poruke koja se šalje
CANRFS	- Kontrolni bitovi poruke koja je primljena
CANRID	- Identifikator (ID) poruke koja je primljena
CANRDA	- Prvi deo podataka poruke koja je primljena
CANRDB	- Drugi deo podataka poruke

B. UART periferija

Univerzalni Asinhroni Primopredajnik (Universal Asynchronous Receiver/Transmitter) je najčešće korišćena periferija za serijsku komunikaciju. U zavisnosti od korišćenog kola za prilagođavanje liniji, moguće je korišćenje sa nekoliko različitih tipova veza. Koristi se kod RS-232, RS-422, RS-485 komunikacije.

Moguće je komunicirati na različitim brzinama. Brzina, odnosno broj bitova u sekundi se određuje sadržajem u registrima DLL i DLM. Noviji kontroleri imaju i registar FDR (Fractional Divider Register) koji im omogućuje fino podešavanje brzine. Mikrokontroler LPC2129 nema ugrađen FDR registar.

Postoji i FIFO bafer za prijem i predaju koji je veličine 16 bajtova. Registri su raspoređeni i označeni po industrijskom standardu '550. Ugrađeni su i mehanizmi za softversku i hardversku kontrolu protoka (flow control).

Korišćeni pinovi:

PREDAJA: TxDO (P0.0)
PRIJEM: RxDO (P0.1)

Registri koji su od značaja:

UIER - Uključivanje prekida
UIIR - Određivanje vrste prekida
ULCR - Određivanje parametara prenosa
UDLL - Delilac od kojeg zavisi brzina prenosa
UTHR - Bajt za slanje na liniju
URBR - Registar u koji se prima bajt sa linije
ULSR - Registar koji pokazuje status linije

C. GPIO periferija

Ova periferija na mikrokontroleru LPC2129 ima dva porta (Port 0 i Port 1) sa ukupno 46 kontrolisanih pinova. Smer svakog pina je moguće nezavisno programski kontrolisati. Postoje posebni registri uz pomoć kojih je moguće jednom instrukcijom postaviti ili obrisati bilo koji broj bitova na jednom portu. Za razliku od nekih drugih mikrokontrolera iz ove serije LPC2129 ne poseduje brzi (fast) GPIO.

Na ovom analizatoru GPIO periferija se koristi za prihvatanje ulaza sa tastera, paljenje LED indikatora i zvučno signaliziranje.

Korišćeni pinovi:

TASTERI: P0.15, P0.16
LED: P0.11, P0.12, P0.13
BEEPER: P0.20
SNZORI: P1.16, P1.17, P1.18

Registri koji su od značaja:

IOPIN - Stanje na pinovima porta
IOSET - Setovanje postavlja HIGH na pinovima
IOCLR - Setovanje postavlja LOW na pinovima
IODIR - Kontrolisanje smera pinova

D. TIMER periferija

LPC2129 poseduje dve tajmer/brojač periferije, kod kojih se režim rada slobodno bira i koje su skoro identične. Jedina razlika je u adresama registara. Glavni deo svakog od dva ugrađena tajmer/brojača je jedan 32-bitni TC registar koji se uvećava na svakih PR+1 ciklusa periferijskog takta PCLK.

Tajmer broji i pokazuje broj ivica signala koje dolaze u konstantnim vremenskim intervalima. Kada znamo frekvenciju i broj ivica, lako računamo proteklo vreme. Kako je broj hardverskih tajmera konačan, moguće je koristiti jedan hardverski tajmer za implementaciju velikog broja softverskih tajmera. To je princip koji je korišćen i kod realizacije ovog analizatora, aktivan je samo jedan beskonačni hardverski tajmer koji se koristi za realizaciju većeg broja softverskih tajmera.

Sve funkcije koje kontrolišu rad softverskih tajmera se nalaze u fajlu SW_TAJMER.c. Korišćeno je ukupno 6 softverskih tajmera kojima se u tačno zadatim trenucima pozivaju odgovarajuće funkcije.

Da bi se obezbedilo nesmetano funkcionisanje softverskih tajmera i posle premotavanja hardverskog tajmera (posle vrednosti 4.294.967.295 sledeća vrednost tajmera je nula) koristi se prekid koji se aktivira posle svakog premotavanja i poziva prekidnu rutinu Handler_TIMER0_IRQ() koja odradi potrebne provere i prilagođavanja.

Registri koji su od značaja:

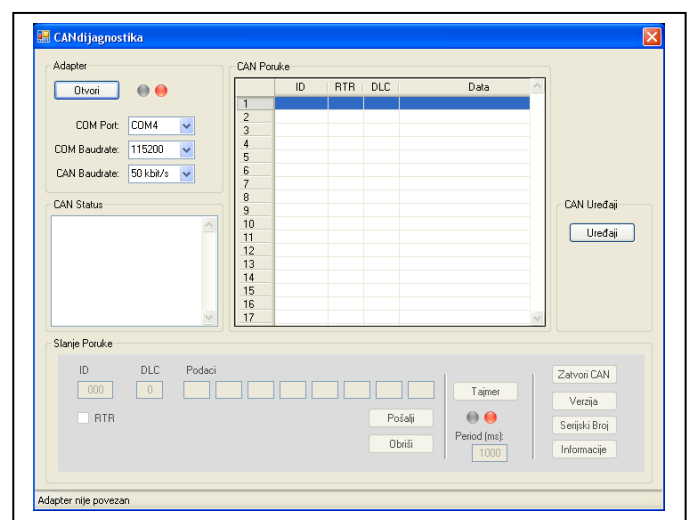
T0TC - Trenutna vrednost tajmera
T0IR - Registar prekida
T0TCR - Kontrolni registar tajmera
T0MCR - Akcija kod poklapanja (match)

VI. KORISNIČKI INTERFEJS

Analizator sve što pročita na magistrali šalje kao string na serijsku liniju preko UART periferije. Sa serijske linije poruke je moguće čitati bilo kojim terminal emulatorom (Hyper Terminal, TeraTerm...). Kako je svaki string završen karakterom <CR>, poruke će u terminalu biti prikazane jedna ispod druge. Ovakav prikaz može biti nedovoljno pregledan jer su delovi poruke prikazani svi zajedno u jednom redu bez bilo kakvog razdvajanja ili formatiranja. Teško je na prvi pogled videti koje heksadecimalne cifre predstavljaju identifikator, koje DLC polje a koje podatke, pošto su sve zapisane jedna do druge.

Rešenje može da bude korišćenje nekog od već postojećih CAN monitor programa, koji pored lepšeg i formatiranog ispisa nude još dosta drugih opcija. Opisani analizator je u potpunosti kompatibilan i testiran sa popularnim programom CANHacker (verzija 2.00).

U sklopu ovog rada je odrađeno i pisanje posebnog PC Windows monitor programa prikazanog na Sl. 4 koji služi za formatiran prikaz pristiglih poruka ali i za lakše slanje poruka na magistralu. Program se zove CANDijagnostika i napisan je korišćenjem Microsoft .NET tehnologije i C# programskog jezika [9].



Slika 4. Glavna forma programa CANDijagnostika

VII. KORIŠĆENJE PROGRAMA CANDIJAGNOSTIKA

U delu Adapter moguće je podesiti parametre serijske veze, COM port i bitsku brzinu kao i brzinu na CAN magistrali. Ostali parametri (broj bitova za podatke, broj stop bitova, parnost) su fiksirani i nije ih moguće menjati ni na analizatoru ni u programu. Posle podešavanja parametara, potrebno je kliknuti na dugme Otvori kako bi se COM port i CAN kanal otvorili. Da je CAN kanal otvoren može se videti po gašenju crvenog i paljenju zelenog indikatora pored dugmeta Otvori.

Ispod se nalazi CAN Status deo forme. To je višelinijsko polje za tekst (textbox) u kojem se prikazuju statusne i kontrolne poruke vezane za rad samog analizatora. U ovom polju se ispisuju poruke koje nam govore da je otvoren COM port, pronađen CAN adapter, otvoren ili zatvoren CAN kanal, poslata poruka na magistralu, došlo do greške...

Najveće polje sa nazivom CAN Poruke služi za prikaz prispelih CAN poruka tj. ramova. Poruke se prikazuju u tabeli (data grid) gde je svaka kolona odvojena i posebno prikazana. Moguće je jasno uočiti ID (identifikator), RTR, DLC ili podatke svake poruke. Kada je broj poruka veći od veličine tabele, koristi se skrol sa desne strane kako bi sve poruke bile vidljivije.

U donjem delu forme se nalazi polje Slanje Poruke. Ovo polje nam je od koristi kada želimo da neku našu, proizvoljnu poruku (ram) postavimo na CAN magistralu. Ručno u polje ID upisujemo identifikator kao tri heksadecimalne cifre, u polje DLC upisujemo broj bajtova podataka koje želimo da pošaljemo, i u 8 polja za podatke upisujemo do 8 bajtova podataka kao grupe po dve heksadecimalne cifre. Ako je ram koji želimo da pošaljemo ram sa zahtevom, potrebno je označiti (check) kvadratić pored RTR oznake. U tom slučaju unosi se samo identifikator, DLC je 0, a podaci se ne unose. Kada su svi potrebni elementi rama za slanje upisani potrebno je kliknuti na dugme Pošalji i tog trenutka se zadati ram pojavljuje na CAN magistrali.

Ako isti ram želimo poslati nekoliko puta, dovoljno je samo više puta kliknuti na dugme Pošalji. Ram je moguće delimično menjati, ili posle klika na dugme Obriši uneti sasvim novi ram. Omogućeno je i korišćenje tajmera čija je uloga da jedan isti uneti ram u pravilnim vremenskim intervalima šalje na

magistralu. Vremenski interval između dva rama se u polju Period (ms) zadaje u milisekundama. Klikom na dugme Tajmer, periodično slanje ramova započinje. Da je tajmer aktivan, govori nam zeleni indikator ispod dugmeta Tajmer i promena teksta dugmeta Tajmer u Zaustavi. Prestanak periodičnog slanja se postiže klikom na dugme Zaustavi čiji se tekst u tom trenutku ponovo menja u Tajmer. Po zaustavljanju se gasi zeleni indikator i pali se crveni.

U donjem desnom uglu se nalaze dugmići Zatvori CAN, Verzija i Serijski broj. Kao što se iz naziva samih dugmića može zaključiti, njihova uloga je da otvaraju/zatvaraju CAN kanal i daju informacije o verziji softvera (firmvera) i serijskom broju samog analizatora.

LITERATURA

- [1] Robert Bosch GmbH "CAN Specification", 1991.
- [2] Augie Hansen "Programiranje na jeziku C", Mirko knjiga, 1991.
- [3] IAR Systems "ARM IAR Embedded Workbench IDE User Guide", 2006.
- [4] NXP Semiconductors "UM10114 - LPC21xx and LPC22xx User manual", 2008.
- [5] NXP Semiconductors "LPC2109/2119/2129 Data Sheet", 2007.
- [6] Olimex Ltd "LPC-P2129 Get Started Guide", 2005.
- [7] ARM Holdings plc "ARM Architecture Reference Manual", 2005.
- [8] NXP Semiconductors "AN10438 - Philips LPC2000 CAN driver", 2006.
- [9] Andrew Troelsen "Pro C# 2008 and the .NET 3.5 Platform", Apress Media LLC, 2007.

ABSTRACT

The paper describes an implementation of CAN protocol analyzer. The CAN protocol itself, its main usage, and used microcontroller with peripherals are briefly described as well. The analyzer could be used for new system development or for monitoring of an existing system. Two operation modes are possible, normal and silent in which only listening is possible. Basically, this device is an interface between CAN bus and standard RS-232 port on the computer.

IMPLEMENTATION OF CAN PROTOCOL ANALYZER

Filipović Marko, Mijalković Milan