
CAN KOMUNIKACIJA

Projekti za ARM i Arduino



Prof Dr Dogan Ibrahim

Agencija Eho

www.infoelektronika.net

Sva prava zadržana. Nijedan deo ove knjige ne sme biti reprodukovan u bilo kom materijalnom obliku, uključujući fotokopiranje ili slučajno ili nenamerno smeštanje na bilo koji elektronski medijum sa ili uz pomoć bilo kog elektronskog sredstva, bez pismenog odobrenja nosioca autorskih prava osim u skladu sa odredbama zakona o autorskim pravima, dizajnu i patentima iz 1988. godine ili pod uslovima izdatim od Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1P 9HE. Prijave za pismene dozvole radi štampanja bilo kog dela ove publikacije upućuje se izdavaču ove knjige. Izdavač je uložio najveće napore da bi se obezbedila tačnost informacije sadržanih u ovoj knjizi. Izdavač ne može da pretpostavi neprijatnosti i ovom izjavom isključuje bilo kakvu odgovornost za bilo koju stranku koja bi imala gubitke ili štetu uzrokovanu greškama ili propustima u ovoj knjizi, bez obzira da li su greške ili propusti nastali usled nemara, nezgode ili bilo kog drugog razloga.

Posebno se zahvaljujemo kompanijama Cim-Team (www.cim-team.br) i Zuken (www.zuken.com) koje su nam dale odobrenje da upotrebimo sliku za naslovnu stranu.

ISBN 978-86-80134-10-9

Naslov originala: Controller Area Network

Autor: Dogan Ibrahim

Prevod: Biljana Tešić

Izdavač: Agencija Eho

Štampa: Agencija Eho

e-mail: redakcija@infoelektronika.net

Sadržaj

PREDGOVOR	13
POGLAVLJE 1 • AUTOMOBILSKI KOMUNIKACIONI SISTEMI	15
1.1 Mrežni sistemi vozila	17
1.2 CAN sistem	18
1.3 LIN sistem	19
1.4 FlexRay	20
1.5 MOST	22
1.6 Byteflight.....	23
1.7 Intellibus	24
1.8 Drugi sistemi.....	25
1.9 Upoređivanje automobilskih komunikacionih sistema	25
1.10 Kratka istorija CAN sistema	28
1.11 CAN u automobilsjoj industriji	31
1.12 Osnovne strukture CAN automobilskog sistema	32
1.13 Prednosti CAN sistema	38
1.14 Nedostaci CAN sistema	38
1.15 Svojstva CAN sistema	38
1.16 ISO/OSI referentni model i CAN	39
1.17 ISO/OSI model CAN sistema.....	41
1.18 CANopen	42
1.19 Rezime	44
POGLAVLJE 2 • STRUKTURA CAN FIZIČKOG SLOJA.....	45
2.1 Završetak na CAN liniji	45
2.2 Brzina prenosa podataka u CAN sistemu	48
2.3 Dužina spoja kabla.....	49
2.4 Čvor CAN linije	49
2.4 Nivoi signala CAN linije	50

2.4.1 CAN_H napon	51
2.4.2 CAN_L napon.....	51
2.4.3 CAN talasasti signal	51
2.4.4 Arbitraža na liniji.....	52
2.4.5 Primopredajnik linije.....	52
2.5 CAN konektori	53
2.6 CAN repetitori	56
2.7 CAN PC interfejs	57
2.8 Rezime	57
POGLAVLJE 3 • PORUKE U CAN KOMUNIKACIJI.....	59
3.1 Poruka sa podacima (Data Frame)	60
3.1.1 Početak poruke (SOF – Start of Frame)	62
3.1.2 Polje arbitraže (Arbitration Field).....	62
3.1.3 RTF polje	64
3.1.4 Kontrolno polje (Control Field)	64
3.1.5 Polje podataka (Data Field)	65
3.1.6 CRC polje.....	65
3.1.7 ACK polje.....	66
3.1.8 Polje kraja poruke (End of Frame Field).....	66
3.2 Poruka sa zahtevom za podacima (Remote Frame).....	67
3.3 Poruka o grešci (Error Frame).....	67
3.4 Poruka o zauzetosti (Overload Frame)	69
3.5 Proširene poruke CAN sistema	69
3.6 Rezime.....	71

POGLAVLJE 4 • STANJA GREŠKE NA CAN SISTEMU.....	73
4.1 Umetanje bitova (Bit Stuffing)	73
4.2 Detekcija greške na liniji	74
4.2.1 Greška u poslatom bitu	74
4.2.2 Greška pri umetanju bita.....	75
4.2.3 CRC greška	75
4.2.4 Greška u poruci	75
4.2.5 ACK greška	75
4.3 Upravljanje greškama na CAN sistemu	75
4.4 Rezime	77
POGLAVLJE 5 • RAZMENA PODATAKA NA CAN SISTEMU	79
5.1 Razmena podataka pomoću poruka sa podacima	79
5.2 Poruke sa zahtevom za podacima na liniji.....	82
5.3 Rezime	84
POGLAVLJE 6 • USKLAĐIVANJE VREMENA NA CAN SISTEMU.....	85
6.1 Vremensko usklađivanje bita	85
6.2 Odabir vremenskih segmenata bita	88
6.2.1 Prop_Seg	88
6.2.2 Tolerancija frekvencije oscilatora	89
6.3 Rezime	98
POGLAVLJE 7 • CAN KONTROLERI.....	99
7.1 Osnovna struktura CAN primopredajnika	100
7.2 Osnovna struktura CAN kontrolera.....	101
7.3 MCP2515 CAN kontroler (bez ugrađenog primopredajnika)	104
7.4 MCP2515 CAN kontroler (sa ugrađenim primopredajnikom)	105
7.5 Rezime	106

POGLAVLJE 8 • PROGRAMERSKE ALATKE CAN LINIJE	107
8.1 Hardverske programerske alatke	107
8.1.1 CAN MicroMOD razvojni komplet.....	107
8.1.2 mikroElektronika CAN komplet za komunikaciju	108
8.1.3 RCDK8C CAN razvojni komplet	108
8.1.4 mikroElektronika CAN SPI Click pločica	110
8.1.5 mikroElektronika CAN-1 pločica	110
8.1.6 CAN Bus Monitor Demo ploča	110
8.2 CAN analizatori	111
8.2.1 Microchip Inc CAN analizator.....	111
8.2.2 CANdo	112
8.2.3 PCAN Explorer	113
8.2.4 CAN-Bus-Tester (CBT2)	115
8.2.5 BitScope Logic	116
8.2.6. LAP-C logički analizator	117
8.3 CAN softverske programerske alatke	118
8.3.1 Keil Real-Time Library (RL-ARM)	118
8.3.2 mikroElektronika mikroC Pro for ARM	118
8.3.3 STM32F2xx standardna periferijska biblioteka	118
8.4 Rezime	119
POGLAVLJE 9 • ARHITEKTURA ARM MIKROKONTROLERA.....	121
9.1 STM32 porodica ARM mikrokontrolera	121
9.2 STM32F107VCT6 mikrokontroler	121
9.2.1 Osnovne funkcije mikrokontrolera STM32F407VCT6	122
9.2.2 Unutrašnji blok dijagram	123
9.2.3 Napajanje	125
9.2.4 Režimi niskog napajanja.....	125
9.2.5 Kolo radnog takta.....	126
9.2.6 Ulazi i izlazi opšte namene (GPIO-ovi).....	136
9.3 Rezime	140

POGLAVLJE 10 • PROGRAMIRANJE ARM MIKROKONTROLERA	141
10.1 Funkcije jezika mikroC za mikrokontroler STM32F407VCT6	141
10.2 Biblioteka ulazno-izlaznog pina opšte namene (GPIO).....	143
10.2.1 GPIO_Clk_Enable.....	143
10.2.2 GPIO_Clk_Disable	144
10.2.3 GPIO_Config	144
10.2.4 GPIO_Set_Pin_Mode	146
10.2.5 GPIO_Digital_Input	147
10.2.6 GPIO_Digital_Output.....	147
10.2.7 GPIO_Analog_Input	148
10.2.8 GPIO_Alternate_Function_Enable.....	148
10.3 Specifikatori tipa memorije	148
10.4 Ulaz-izlaz PORT-a	149
10.5 Pristupanje pojedinačnim bitovima.....	149
10.6 Tip podataka bit	149
10.7 Prekidi i izuzeci	149
10.7.1 Izuzeci	149
10.7.2 Rutina za obradu prekida	150
10.8 Kreiranje novog projekta	151
10.9 Simulacija	155
10.9.1 Podešavanje tačaka prekida	159
10.10 Debagiranje	160
10.11 Druge mikroC IDE alatke	161
10.11.1 ASCII Chart	161
10.11.2 GLCD Bitmap Editor	162
10.11.3 HID Terminal.....	162
10.11.4 Interrupt Assistant.....	163
10.11.5 LCD Custom Character	163
10.11.6 Seven Segment Editor	163
10.11.7 UDP Terminal	164

10.11.8 USART Terminal	164
10.11.9 USB HID Bootloader	165
10.11.10 Statistika	165
10.11.11 Library Manager	166
10.12 Rezime	167
POGLAVLJE 11 • RAZVOJNA PLOČA CLICKER 2 ZA STM32	169
11.1 LED-ovi	169
11.2 Tasterski prekidači	170
11.3 Napajanje	170
11.4 mikroBUS konektori	171
11.5 Programiranje mikrokontrolera na ploči Clicker 2 za STM32	172
11.6 Rezime	175
POGLAVLJE 12 • MIKROC PRO FOR ARM SA UGRAĐENIM FUNKCIJAMA CAN LINIJE ...	177
12.1 Mikrokontroler STM32F407VGT6 sa ugrađenim CAN modulom kontrolera 178	
12.1.1 Prenos poruke	179
12.1.2 Prijem poruke	179
12.2 Funkcije CAN linije za mikroC Pro for ARM	179
12.2.1 CANxSetOperationMode	180
12.2.2 CANxInitializeAdvanced	180
12.2.3 CANSetMask	181
12.2.4 CANSetFilter	181
12.2.5 CANSetFilterScale16	182
12.2.6 CANSetFilterScale32	182
12.2.7 CANxRead	182
12.2.8 CANxWrite	183
12.2.9 CANSlaveStartBank	183

POGLAVLJE 13 • PROJEKTI CAN LINIJE U KOJIMA SE KORISTE ARM MIKROKONTROLERI	185
13.1 Projekat – Daljinska kontrola LED-a	186
13.2 Projekat 2 – Daljinska kontrola temperaturnog alarma	202
13.3 Projekat 3 – Daljinska kontrola drajvera DC motora	214
13.4 Projekat 4 – CAN linija sa 3 čvora	225
POGLAVLJE 14 • UPOTREBA ANALIZATORA CAN LINIJE	245
14.1 Upotreba logičkog analizatora kao analizatora CAN linije	245
14.2 Upotreba Microchip Inc analizatora CAN linije (APGDT002)	249
14.2.1 Povezivanje analizatora CAN linije sa računarom i CAN linijom	250
14.2.2 Primer analizatora CAN linije	250
14.2.3 Prenos podataka na liniju	252
14.3 Rezime.....	253
POGLAVLJE 15 • CAN SPI.....	255
15.1 CAN SPI funkcije	255
15.1.2 CANSPISetOperationMode.....	258
15.1.3 CANSPIGETOperationMode.....	258
15.1.4 CANSPIInitialize	258
15.1.5 CANSPISetBaudRate	259
15.1.6 CANSPISetMask	259
15.1.7 CANSPISetFilter.....	260
15.1.8 CANSPIRead	260
15.1.9 CANSPIWrite.....	261
15.2 Primer projekta	261

POGLAVLJE 16 • UPOTREBA ARDUINA SA CAN LINIJOM	273
16.1 Arduino modul CAN linije	274
16.2 Funkcije CAN linije.....	275
16.2.1 CAN.begin(Baud)	275
16.2.2 CAN.checkReceive()	275
16.2.3 CAN.readMsgBuf(&msglen, msgbuf)	275
16.2.4 CAN.getCanId()	276
16.2.5 CAN.sendMsgBuf(msgID, ext, len, buf)	276
16.2.6 CAN.checkError().....	276
16.2.7 CAN.init_Mask(num, ext, uiData)	276
16.2.8 CAN.init_Filt(num, ext, uiData).....	276
16.2.9 CAN.isRemoteRequest()	276
16.2.10 CAN.isExtendedFrame()	277
16.3 Dodavanje datoteka.....	277
16.4 Primer CAN sistema	277

Predgovor

Korisnicima mikrokontrolera postaje važno da se brzo prilagode novim tehnologijama i da nauče arhitekturu i primenu 32-bitnih mikrokontrolera visokih performansi. Nekoliko proizvođača nude 32-bitne mikrokontrolere kao procesore opšte namene u ugrađenim aplikacijama. Na primer, pored veoma popularnih 8-bitnih i 24-bitnih mikrokontrolera, kompanija Microchip Inc nudi 32-bitne mikrokontrolere PIC porodice i programerske alatke.

ARM nudi 32-bitne i 64-bitne procesore uglavnom za primenu u ugrađenim sistemima. U današnje vreme, većina mobilnih uređaja, kao što su mobilni telefoni, tabletni računari i GPS prijemnici, zasnovani su na tehnologiji ARM procesora. Zbog njihove niske cene, male potrošnje energije i visokih performansi, oni su idealni kandidati za upotrebu u složenoj komunikaciji i aplikacijama sa mešovitim signalima.

Arduino je veoma popularna razvojna ploča porodice mikrokontrolera koju trenutno koriste studenti, inženjeri u praksi i hobisti.

CAN komunikacija protokola (Controller Area Network) je prvobitno napravljena za upotrebu u putničkim automobilima. CAN kontrolere sada proizvodi više od 20 proizvođača. CAN se primenjuje u mnogim industrijskim oblastima, uključujući medicinu, vazduhoplovstvo, upravljanje procesom, automatizaciju i tako dalje. Pojavom udruženja CAN u automatizaciji (CiA, engl. CAN in Automation) 1992. godine, proizvođači i korisnici su se udružili da bi razmenili ideje i razvili CAN standarde i specifikacije.

Tema ove knjige je upotreba ARM Cortex-M porodice mikrokontrolera i razvojne ploče Arduino Uno u praktičnim projektima koji se zasnivaju na CAN bežičnoj vezi. U ovoj knjizi je detaljno prikazan uvod u arhitekturu Cortex-M porodice. Ukratko su opisani primeri popularnih kompleta za razvoj hardvera i softvera. Pomoću ovih kompleta može da se pojednostavi ciklus dizajniranja ugrađenih sistema i olakša razvoj, otklanjanje grešaka i testiranje projekata zasnovanim na CAN sistemu. Arhitektura veoma popularnih ARM Cortex-M procesora STM32F407VGT6 je opisana na visokom nivou tako što su razmatrani različiti modeli. Pored toga, detaljno je opisana upotreba mikroPC-ija za ARM i Arduino UNO biblioteke za funkcije CAN sistema.

Ova knjiga je napisana za studente, inženjere u praksi, hobiste i za sve ostale koji žele da nauče više o CAN sistemu i njegovim aplikacijama. U ovoj knjizi se pretpostavlja da čitalac ima neko poznavanje osnovne elektronike. Poznavanje programskog jezika C će biti korisno za naredna poglavlja u knjizi, a poznavanje najmanje jednog mikrokontrolera će biti prednost, naročito ako čitalac namerava da programira mikrokontrolere zasnovane na projektima u kojima se koristi CAN sistem.

Knjiga bi trebalo da bude koristan izvor podataka svakome ko je zainteresovan za pronalaženje odgovora na jedno ili više sledećih pitanja:

- Koji komunikacioni sistemi su dostupni za automobilsku industriju?
- Koje su osnove rada CAN sistema?
- Koji tipovi paketa podataka su dostupni u CAN sistemu?
- Kako se mogu detektovati greške u CAN sistemima i koliko je on pouzdan?
- Koji tipovi kontrolera CAN sistema postoje?
- Koje su prednosti mikrokontrolera ARM Cortex-M?
- Kako se može kreirati CAN projekat korišćenjem ARM mikrokontrolera?
- Kako se može kreirati CAN projekat korišćenjem Arduino mikrokontrolera?
- Kako se mogu pratiti podaci na CAN sistemima?

Nadam se da će vam ova knjiga biti od koristi prilikom izrade vašeg sledećeg projekta CAN sistema pomoću mikrokontrolera ARM Cortex-M i Arduina.

Dogan Ibrahim,
London, 2016.

Poglavlje 1 • Automobilski komunikacioni sistemi

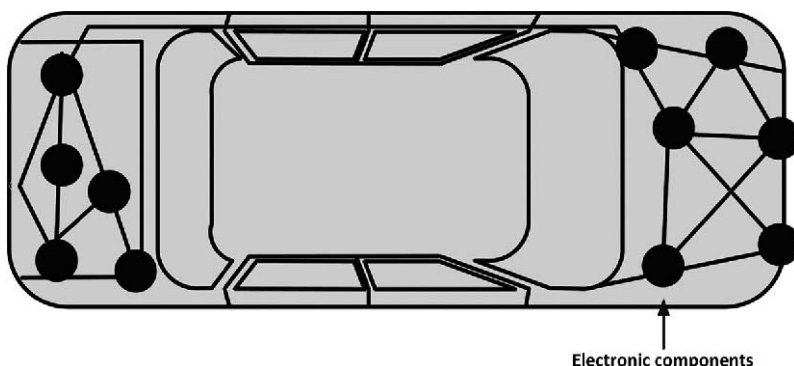
Današnja vozila su veoma kompleksne mašine koje sadrže mehaničke i elektronske delove. Poslednjih godina, broj elektronskih komponenata koje se koriste u vozilima rapidno raste. Povećanje zahteva za bezbednost, udobnost i performanse dovelo je do toga da se u moderna vozila dodaje mnogo više elektronskih komponenti. Došlo je do povećanja potražnje za povezivanjem ovih elektronskih komponenata na takav način da mogu komunicirati pouzdano, bezbedno i u realnom vremenu.

Današnji automobilski elektronski sistemi sadrže veliki broj senzora, aktuatora, jedinica za praćenje, sistema za zabavu i navigaciju koji se distribuiraju i ugrađuju u različite delove vozila. Procenjuje se da u tipičnom savremenom putničkom automobilu postoji više od 70 elektronskih upravljačkih jedinica koje razmenjuju preko 2500 signala i taj broj raste kako se povećava kompleksnost.

U prošlosti, elektronske jedinice su bile povezane na složen način pomoću stotina žica na različitim delovima vozila. Prema tome, bilo je veoma teško otkriti elektronski kvar. Nije postojala koordinacija između različitih delova elektronike jer se svakim elektronskim delom upravljalo nezavisno. Održavanje i popravka elektronskih delova vozila je bilo izuzetno teško, zato što u mnogim slučajevima nije bilo jednostavno locirati i promeniti neispravan deo.

Bezbednost putnika u savremenim vozilima je, takođe, postala jedna od najvažnijih tema. U protekloj deceniji bezbednosna oprema je evoluirala i prešla je sa fizičke na elektronsku bezbednost, počev od tehnologije kočnica i tipa, pa sve do zaštite od sudara i airbagova, a najskorije i na bezbednosne sisteme za pomoć vozačima. Najnovija savremena vozila su pametne mašine i opremljena su mnogim senzorima koji mogu da procene okruženje i da prikažu korisne i bezbednosne informacije vozačima. Ovi senzori formiraju pametne lokalne mreže zajedno sa aktuatorima, displejima i brzim digitalnim procesorima, kao što su ugrađeni mikrokontroleri za velike brzine.

Na slici 1.1 prikazan je tradicionalni elektronski sistem vozila sa senzorima i aktuatorima koji su međusobno povezani na složen način. Jedan od glavnih problema u ovom tipu dizajna je održavanje. Ožičenje je toliko složeno da je bilo gotovo nemoguće pratiti i popraviti kvar.

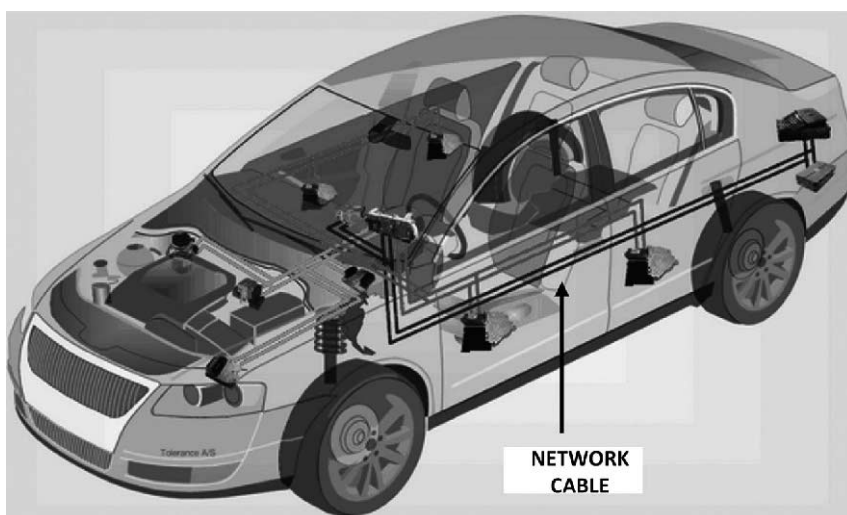


Slika 1.1 – Tradicionalna elektronika vozila

Pošto se složenost elektronike vozila povećavala, proizvođačima je postalo komplikovano da dizajniraju bezbedne i pouzdane elektronske sisteme zasnovane na starim tradicionalnim metodama. Trenutni zahtevi se ne mogu ispuniti pomoću jednostavnih elektronskih upravljačkih jedinica. Rešenje je da se različiti elektronski moduli povežu sa mrežom visokih performansi. Zato je postalo neophodno da se dizajnira elektronski sistem zasnovan na mreži u koju se elektronski moduli mogu lako dodati i zatim kontrolisati sa pametne centralne jedinice (tj. upravljačke jedinice motora). Rezultat toga je „pametna“ automobil u kome se koriste mnogi senzori i aktuatori za detektovanje okruženja i obavljanje više funkcija. Primer je automatsko uključivanje farova kada padne mrak ili kada automobil prolazi kroz tunel. Drugi primer je automatski rad brisača vetrobrana kada počne kiša i tako dalje.

Jedna od prednosti elektronskog sistema zasnovanog na umrežavanju je prilično lako praćenje i otkrivanje neispravnog modula. Pored toga, ožičenje je mnogo jednostavnije i lakše za održavanje. Na primer, komunikacijom sa centralnom pametnom upravljačkom jedinicom može da se otkrije da li je ceo elektronski sistem ispravan, a ako nije, neispravni moduli se mogu lako detektovati. Takođe, umreženi sistem omogućava različitim modulima na mrežnim vodovima da međusobno komuniciraju i razmenjuju informacije, ako je potrebno. Na primer, pametna upravljačka jedinica može da primi vrednost temperature motora sa modula senzora temperature. Ova temperatura može, zatim, da se prikaže na elektronskoj komandnoj tabli. Ukoliko je temperatura previsoka, odgovarajući signali se mogu poslati na određene komponente kako bi se preduzele zaštitne mere.

Na slici 1.2 prikazano je savremeno vozilo u kome se za povezivanje i kontrolu elektronskih modula koristi elektronski mrežni sistem.



Slika 1.2 – Savremeno vozilo sa elektronskom mrežom

Ovo poglavlje sadrži pregled najvažnijih mrežnih sistema vozila koje se trenutno koriste i tabelu u kojoj se upoređuju prednosti i nedostaci svakog sistema.

1.1 Mrežni sistemi vozila

Mrežne sisteme vozila (ili mreže) klasifikovalo je Udruženje automobilskih inženjera (SAE, engl. Society of Automotive Engineers) 1994. godine. Prema ovoj klasifikaciji, mrežni sistemi su klasifikovani na osnovu njihovog propusnog opsega (tj. brzine prenosa podataka) i funkcija na mrežnom sistemu. Klasifikacijom se komunikacione mreže dele na četiri klase: klase A, B, C i D.

Mreže A klase su jeftine mreže sa brzinom prenosa podataka manjom od 10 Kb/s. Ovi sistemi se uglavnom koriste u karoseriji automobila.

Mreže B klase rade između 10 i 125 Kb/s i koriste se za razmenu informacija, npr. instrument table, brzine vozila i tako dalje.

Mreže C klase rade između 125 i 1 Mb/s i imaju širok spektar primene, kao što je kontrola rada motora.

Mreže D klase rade iznad 1 Mb/s i najčešće se koriste u telematskim aplikacijama.

Postoji veliki broj automobilskih komunikacionih sistema, neke su razvili sami proizvođači automobila, a neke su razvili zajedničkim radom sa proizvođačima poluprovodnika. Popularni mrežni sistemi vozila sadrže:

- CAN protokol
- LIN protokol
- FlexRay
- MOST
- Byteflight

- DSI protokol
- Intellibus
- SAE J1850
- BST protokol
- NML protokol
- I druge...

U ovom odeljku ćemo razmatrati osnovna svojstva automobilskih mrežnih sistema koji se najčešće koriste, i to:

- CAN sistema
- LIN sistema
- FlexRay
- MOST
- Byteflight
- Intellibus

1.2 CAN sistem

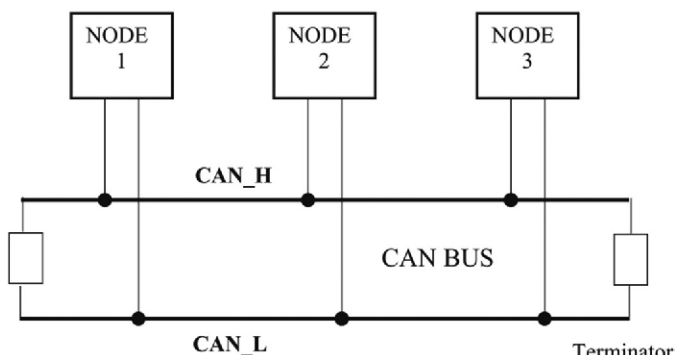
CAN komunikacioni sistem (Controller Area Network) je glavna tema ove knjige. U ovom odeljku ćemo razmatrati osnovna svojstva ovog automobilskeg komunikacionog sistema i drugih. Detaljni opisi CAN sistema i projekata zasnovanim na CAN sistemu dati su u ostalim poglavljima knjige.

CAN je serijski, dvožični multimaster sistem koju je napravila kompanija Robert Bosch GmbH 80-ih godina. To je danas jedna od najčešće korišćenih automobilskih komunikacionih sistema. Fizički sloj CAN sistema se sastoji od kablova sa upređenim paricama. CAN omogućava pouzdanu, robusnu i brzu komunikaciju do 1 Mb/s (uz dužinu mrežnog m). CAN 2.0A je originalni CAN sistem, koji se sastoji od polja: bit početka paketa (Start of Frame), 18-bitno zaglavlje (koje sadrži 11-bitni identifikator poruka), 0-8 bajtova podataka, 15-bitna ciklična provera redundanse (CRC, engl. Cyclic Redundancy Check), 3-bitni potvrdni otvor (ACK otvor) i 7-bitni kraj paketa (End of Frame).

CAN sistem se zasniva na mehanizmu CSMA/CR (Carrier Sense Multiple Access/Collision Resolution - Višestruki pristup sa osluškivanjem nosioca i otkrivanjem kolizija) za sprečavanje kolizije paketa tokom prenosa na komunikacionoj liniji. Svaki CAN čvor nadgleda sistem i kada detektuje da je u stanju mirovanja, može da započne prenos podataka. Ako ostali čvorovi na liniji pokušaju da pošalju podatke u isto vreme, dolazi do arbitraže i čvor sa najvišim prioritetom (identifikator poruke najnižeg prioriteta) će dobiti arbitražu i poslati podatke. CAN sistem ima jednostavan detektor greške i mehanizam za oporavak. Prijemni čvorovi proveravaju integritet poruka tako što pregledaju CRC polja. Ako se detektuje greška, ostali čvorovi na liniji se obaveštavaju porukama o grešci. Na slici 1.3 prikazana je tipična implementacija CAN sistema sa dva čvora, A i B. CAN je mreža tipa A i B klase.

Poglavlje 2 • Struktura CAN fizičkog sloja

Fizički sloj formira interfejs hardvera koji uspostavlja vezu između niza čvorova. Za komunikaciju na liniji potrebno je najmanje dva čvora. Na slici 2.1 prikazan je CAN sistem sa tri čvora. Komunikaciona linija se sastoji od kabla sa upredenim paricama i završava se otpornicima na oba kraja, tako da otpornik na liniji ima 120 oma. Jedan kabl se zove CAN_H, a drugi se zove CAN_I.



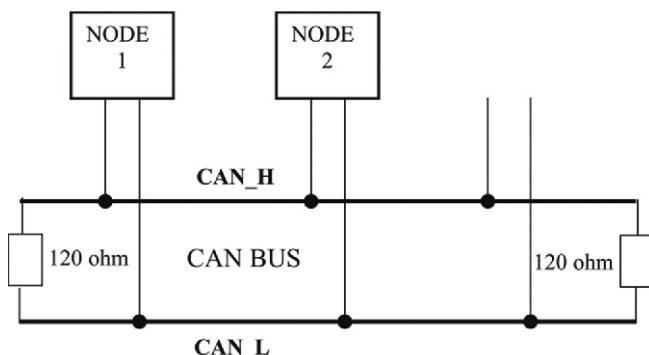
Slika 2.1 – CAN sistem sa tri čvora

2.1 Završetak na CAN liniji

Linija se prekida da bi se smanjile refleksije signala. Obično je na svakom kraju linije priključen jedan otpornik od 120 oma. Iako snaga izabranog otpornika nije naročito važna, treba uzeti u obzir moguće kratke spojeve u napajanju linije pri izboru snage otpornika. Najviše se koristi otpornik snage 1/4 W i tolerancije 5%, ali preporučuje se korišćenje veće snage otpornika (npr. 1 W) da bi se izbeglo oštećenje na liniji zbog mogućih kratkih spojeva primopredajnika.

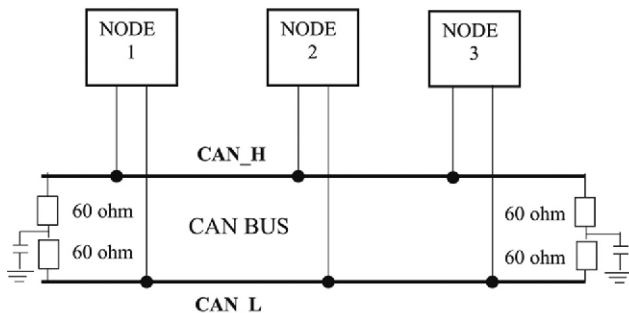
Iako se koristi jedan otpornik od 120 oma, uglavnom se mogu koristiti sledeće metode za završetak linije:

- Standardni završetak
- Podeljeni završetak
- Pristrasni podeljeni završetak
- Najčešće korišćena metoda završetka je standardni završetak za koji se koristi otpornik od 120 oma na svakom kraju linije, kao što je prikazano na slici 2.2.



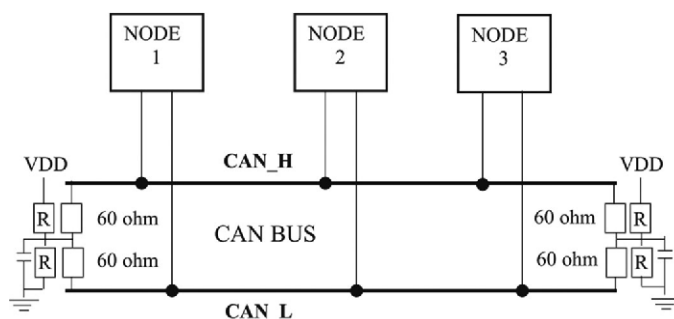
Slika 2.2 – Standardni završetak linije

Na slici 2.3 prikazan je podeljeni završetak koji postaje sve popularniji. Kod ove metode se koriste dva otpornika od 60 oma i kondenzator na svakom kraju linije. Prednost ove metode je to što se eliminiše šum visoke frekvencije iz komunikacionih linija. Treba voditi računa da se otpornici upare tako da se ne smanji efikasna imunost na liniji. Obično se bira 4,7 nF kondenzator koji generiše odgovor sa 3 dB tačkom na oko 1,1 Mb/s.



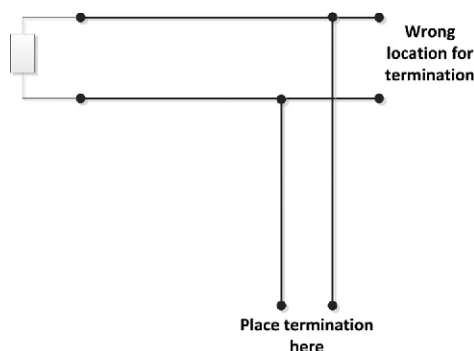
Slika 2.3 – Podeljeni završetak linije

Na slici 2.4 prikazan je pristrasni podeljeni završetak linije za koji se koriste razdelnik napona i kondenzator na svakom kraju linije. Kao i podeljeni prekid, pristrasni podeljeni prekid povećava elektromagnetnu kompatibilnost komunikacione linije.



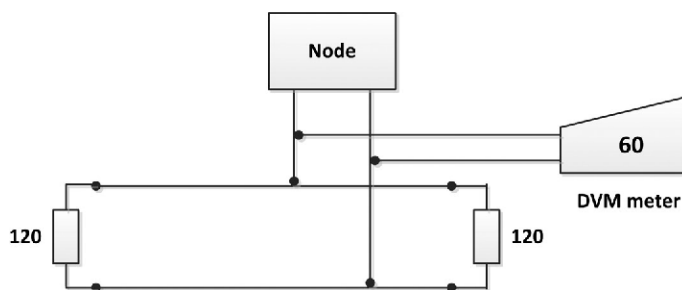
Slika 2.4 – Pristrasni podeljeni završetak linije

Važno je napomenuti da završni otpornici mora da budu postavljeni na krajevima najdužih linija. Na slici 2.5 jedan završni otpornik je postavljen na pogrešno mesto. Pogrešne vrednosti i pogrešan položaj završnog otpornika prouzrokuju greške na liniji.



Slika 2.5 – Pogrešan položaj završnog otpornika

Ispravan završetak linije možete da proverite pomoću digitalnog voltmetra za merenje otpora na liniji. Podesite voltmetar da biste izmerili otpor i povežite dva njegova kontakta sa krajevima čvora na liniji (pogledajte sliku 2.6). Voltmetar treba da izmeri 60 oma.



Slika 2.6 – Provera završetka linije

Poglavljje 3 • Poruke u CAN komunikaciji

Poruke u CAN komunikaciji se šalju i primaju u obliku paketa (Frames). Oni su slični paketima u TCP/IP mreži u kojoj se stvarni podaci kapsuliraju sa kontrolnim podacima. CAN komunikacija nije kao popularni tip komunikacije Client-Master (klijent-master). U CAN komunikaciji svi čvorovi imaju ista prava i mogu da obavljaju prenos, kao i da primaju podatke u odgovarajućem trenutku.

Neke važne funkcije CAN protokola su sledeće:

- CAN komunikacioni sistem je multimaster sistem. Kada je mreža slobodna, svaki uređaj koji je priključen na liniju može da započne slanje poruke. Kada više uređaja pokuša da pošalje podatke istovremeno, onda može doći do kolizija na liniji. Kolizije se detektuju i izbegavaju pomoću mehanizma za arbitražu.
- CAN protokol je fleksibilan. Uređaji povezani sa linijom nemaju adrese (ili ID-ove čvora), što znači da se poruke ne prenose sa jednog čvora na drugi na osnovu adrese. Umesto toga, svi čvorovi na liniji primaju svaku prenetu poruku, a svaki čvor odlučuje da li će se primljena poruka zadržati ili odbaciti. Jedna poruka može da bude usmerena na određeni uređaj na određenom čvoru ili na više čvorova, što zavisi od toga kako je komunikacioni sistem dizajniran. Poruke imaju identifikatore poruka (Message Identifiers) i adresne filtere (Acceptance Filters) na svakom čvoru koji odlučuju da li će prihvatiti poruku koja se prenosi na liniji. Još jedna prednost prenosa koji nije zasnovan na adresi je da kada se uređaj dodaje ili uklanja sa linije, nije potrebno menjati podatke o konfiguraciji (tj. linija se "priključuje tokom rada").
- Poruke koje su poslate na liniju imaju prioritete. Poruka sa nižom vrednošću identifikatora ima viši prioritet.
- Brzina CAN linije nije fiksna. Za uređaje koji su priključeni na liniju brzina komunikacije može da se poveća do dozvoljenog maksimuma.
- CAN sistem sadrži poruku sa zahtevom za podacima (RTR, engl. Remote Transmit Request), što znači da čvor na liniji može da zahteva podatke od drugih čvorova. Dakle, umesto da se čeka da čvor kontinuirano šalje podatke, zahtev za podacima može da se pošalje čvoru. Na primer, u vozilu u kome je temperatura motora važan parametar, sistem može da bude napravljen tako da se podaci o temperaturi periodično šalju na liniju. Međutim, elegantnije rešenje je da se podaci o temperaturi zahtevaju po potrebi. Ovim drugim pristupom smanjiće se saobraćaj u komunikaciji i preveliko opterećenje procesora, uz održavanje integriteta.
- Svi uređaji na liniji mogu da detektuju grešku. Uređaj koji je detektovao grešku odmah obaveštava sve druge uređaje o grešci. Čvorovi koji prenose neispravne podatke ili čvorovi koji neprestano primaju podatke sa greškom će se sami ukloniti sa magistrale.
- Prijemni čvorovi na liniji proveravaju validnost primljenog paketa i potvrđuju konzistentnost. Predajni čvor nadgleda liniju za vreme provere validnosti.
- Na liniji može da bude priključeno više uređaja istovremeno i ne postoje logična ograničenja za broj uređaja koji može da se priključi. U praksi, broj čvorova koji se može dodati liniji je ograničen vremenom kašnjenja linije i električnim opterećenjem na liniji.

U CAN komunikaciji postoje četiri tipa poruke:

- **Poruka sa podacima (Data Frame):** Definiše prenos podataka između čvora.
- **Poruka sa zahtevom za podacima (Remote Frame):** Ovu poruku koristi čvor da bi zahtevao prenos poruke (tj. podataka) od drugog čvora.
- **Poruka o grešci (Error Frame):** Bilo koji čvor na liniji može da pošalje poruku o grešci kako bi ukazao na grešku.
- **Poruka o zauzetosti (Overload Frame):** Ovu poruku koristi prijemni čvor da bi ukazao da još nije spreman za prijem paketa.

Sada ćemo detaljnije razmatrati svaku poruku.

U suštini postoje dva tipa CAN protokola: 2.0 A i 2.0 B. CAN 2.0A je stariji standard sa 11-bitnim identifikatorom (pogledajte sledeći odeljak), dok je CAN 2.0B noviji prošireni protokol sa 29-bitnim identifikatorom. Kontroleri 2.0B su potpuno kompatibilni sa 2.0A kontrolerima i mogu da primaju/prenose poruke u oba formata.

Prvo ćemo pogledati standardne CAN 2.0 poruke, a u narednim odeljcima ćemo istražiti CAN 2.0B poruke.

Postoje dva tipa 2.0A kontrolera. Prvi tip je u stanju da šalje i prima samo 2.0A poruke, a prijem 2.0B poruka će prouzrokovati grešku. Drugi tip 2.0A kontrolera (poznatiji i kao 2.0 B pasivni kontroler) šalje i prima 2.0A poruke, ali će potvrditi i prijem 2.0B poruka, a zatim ih zanemariti.

3.1 Poruka sa podacima (Data Frame)

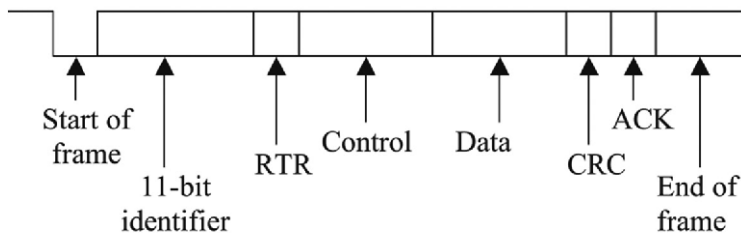
Poruku se podacima koristi predajni uređaj da bi poslao podatke prijemnim uređajima na liniji i ova poruka je najvažnija poruka kojom upravlja korisnik. Poruka sa podacima se može poslati kao odgovor na zahtev ili kad god je potrebno poslati vrednost nekog parametra drugim čvorovima na liniji (npr. vrednost temperature se može poslati u periodičnim intervalima).

Na slici 3.1 prikazana je struktura poruke sa podacima. Komunikaciona linija je u stanju mirovanja. Zatim, standardna poruka sa podacima počinje bitom početka poruke (SOF - Start of Frame), iza kojeg sledi 11-bitni identifikator i bit zahteva za podacima (RTR - Remote Transmission Request). Kontrolno polje je široko 6 bitova i označava koliko bajtova podataka se nalazi u polju podataka. Polje podataka može da bude široko od 0 do 8 bajtova i sadrži stvarne podatke koji se šalju. Iza polja podataka sledi 16-bitno polje kontrolnog zbira (CRC) koje proverava da li je sekvenca primljenog bita oštećena ili ne. ACK polje je široko 2 bita i koristi ga predajni čvor da bi primio potvrdu o validnom paketu sa svakog prijemnika. Kraj poruke je označen 7-bitnim poljem kraja poruke (EOF - End of Frame). Uzastopne poruke moraju biti odvojene razmakom od 3 bita, koji se naziva razmak između poruka (ITM - Interframe Space).

Ukupan broj bitova koje zahteva poruka sa podacima (pod pretpostavkom da se šalju uzastopne poruke):

SOF	1 bit
Identifikator	11 bitova
RTR	1 bit
Kontrola	6 bitova
Podaci	od 0 do 64 bita (0 do 8 bajtova)
CRC	16 bitova
ACK	2 bita
EOF	7 bitova
ITM	3 bita

Ukupno, od 47 bitova (bez podataka) do 111 bitova (8 bajtova podataka) je potrebno za poruku sa podacima.



Slika 3.1 – Standardna poruka sa podacima

Na slici 3.2 je detaljnije prikazana standardna poruka sa podacima.

Poglavlje 4 • Stanja greške na CAN sistemu

Pre nego što detaljno objasnimo tipove grešaka CAN sistema, važno je pogledati mehanizam umetanja bitova (Bit Stuffing) na liniji.

4.1 Umetanje bitova (Bit Stuffing)

CAN standard propisuje da svi bitovi istog polariteta (recesivni ili dominantni) na magistrali koji su duži od 5 bitova narušavaju standard. U stvari, ovaj standard se koristio za slanje poruka o grešci na liniji koja se sastoji od 6 dominantnih bitova u nizu.

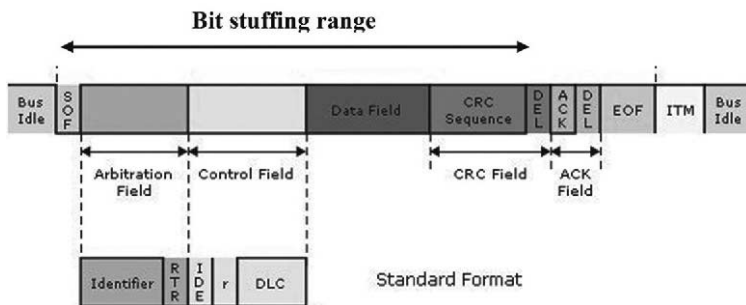
U nekim aplikacijama, možda je potrebno slanje više od 5 bitova istog polariteta (npr. bitovi podataka mogu da sadrže više od 5 bitova istog polariteta). Ovom vrstom situacije upravlja predajni čvor na liniji tako što umeće bit suprotnog polariteta posle petog bita. Prijemni čvor zatim uklanja ovaj bit. Ovaj mehanizam se zove umetanje bitova (Bit Stuffing) i omogućava sinhronizaciju prenosa i prijema operacija da bi se sprečile vremenske greške. Napominjemo da se poruke o grešci i zauzetosti prenose bez umetanja bitova. Takođe, ako je tokom prijema šesti bit isti kao i peti, onda se pojavljuje greška pri umetanju na liniji.

Umetanje bitova je dozvoljeno iz SOF polja u CRC polje (pogledajte sliku 4.1). Umetanje bitova nije dozvoljeno u statičkim poljima poruke, tj. nije dozvoljeno u sledećim poljima:

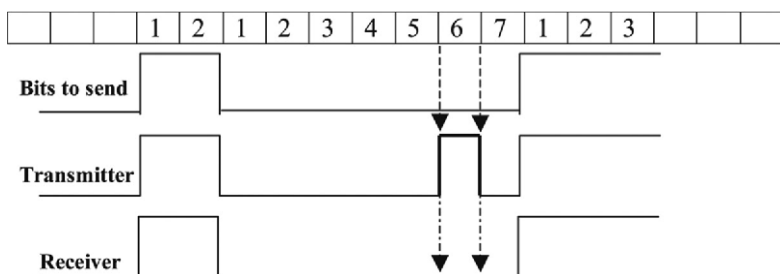
- CRC graničnik
- ACK polje
- EOF polje
- Razmak između poruka

Na slici 4.2 prikazan je primer umetanja bitova u kojem predajni čvor dodaje recisivni bit nakon petog dominantnog bita. Prijemni čvor uklanja ovaj bit, pa je više od 5 dominantnih bitova uspešno preneto na liniji.

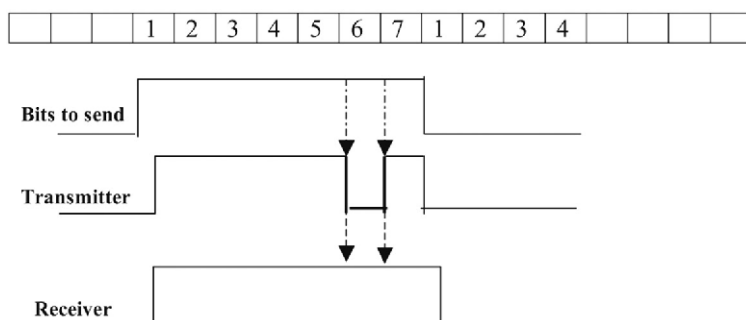
Slično tome, na slici 4.3 predajni čvor dodaje dominantni bit nakon petog recisivnog bita. Prijemni čvor ponovo uklanja umetnuti bit, pa je više od 5 recisivnih bitova uspešno preneto na liniji.



Slika 4.1 – Polja u kojoj se može koristiti umetanje bitova



Slika 4.2 – Primer umetanja bita (dodavanje recesivnog bita)



Slika 4.3 – Primer umetanja bita (dodavanje dominantnog bita)

4.2 Detekcija greške na liniji

Postoji pet tipova greške koje se mogu pojaviti na liniji. Kada se ove greške analiziraju, emituje se poruka o grešci, što dovodi do toga da se u okviru tekuće poruke na liniji označe nepravilnosti i svi čvorovi odbacuju tu poruku. Tipovi greške su sledeći:

- greška u poslatom bitu (Error Bit)
- greška pri umetanju bita (Bit Stuffing Error)
- CRC greška (CRC error)
- greška u poruci (Frame Error)
- ACK greška (ACK error)
- Sada ćemo detaljnije razmotriti ova stanja greške.

4.2.1 Greška u poslatom bitu

Kada čvor prenosi bit na liniji, on nadgleda liniju i upoređuje preneti bit sa stvarnim nivoom na liniji. Greška u poslatom bitu se dešava kada preneti bit nije isti kao nivo bita na liniji.

Napominjemo da se greška u poslatom bitu neće pojaviti tokom faze arbitriranja kada jedan čvor prenosi recesivni bit dok drugi čvor prenosi dominantni bit. Takođe, čvor prenosi poruku o grešci sa 6 uzastopnih recesivnih bitova i ako se na liniji detektuje dominantni bit, to neće dovesti do greške u poslatom bitu.

Poglavljje 5 • Razmena podataka na CAN sistemu

U ovom poglavlju ćemo razmatrati razmenu podataka na CAN sistemu. CAN sistem je multimaster sistem na kome svaki čvor može da obavi prenos ili prijem u bilo kom trenutku.

Kada čvor obavlja prenos na liniji, svi drugi čvorovi osluškuju i primaju prenetu poruku. Iako svi prijemni čvorovi primaju sve poruke, oni mogu da odluče da ne reaguju na sadržaj poruke, jer poruka možda nije od važnosti za čvor.

Čvorovi CAN sistema nemaju adrese. Stoga, poruke se ne prenose na adresirane čvorove, već umesto toga se emituju na liniji. Takođe, čvor ne mora da zna odakle dolazi poruka. Slično tome, predajni čvor ne mora da zna koji su čvorovi zaišta reagovali na poslatu poruku. Čvorovi primaju samo one poruke za koje su programirani.

Iako se unutrašnji rad različitih CAN kontrolera može razlikovati, u ovom poglavlju su kratko opisane osnove rada mehanizma za razmenu podataka.

5.1 Razmena podataka pomoću poruka sa podacima

Poruke sa podacima su najvažnije poruke u CAN sistemu, jer one omogućavaju slanje podataka iz predajnog čvora na druge čvorove magistrale. Poruka sa podacima sadrži dominantni RTR bit. Kada čvor hoće da pošalje podatke drugim čvorovima na liniji, on formira poruku sa podacima. Poruka sa podacima sadrži identifikator poruke, stvarne bajtove podataka i bitove za proveru greške.

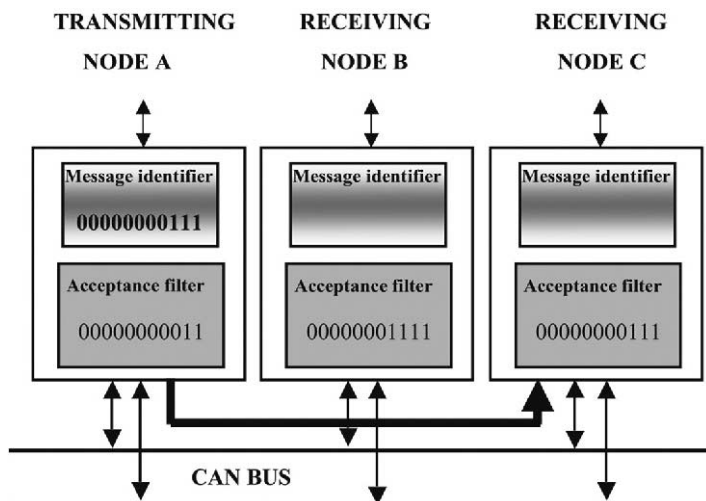
Identifikator poruke je veoma važan jer ga koriste prijemni čvorovi da bi odredili da li će prihvatiti poruku ili ne. CAN kontroleri u prijemnim čvorovima imaju ugrađene adresne filtere (Acceptance Filters) ili filtere prijema (Receive Filters). Programer ili korisnik sistema može da programira ove filtere ili da ih popuni vrednostima. Identifikatori poruka sa podacima na liniji se upoređuju sa vrednostima filtera, a zatim kontroler prihvata poruku ukoliko je vrednost filtera ista kao identifikator poruke. Ako identifikator poruke i vrednosti filtera nisu isti, onda kontroler ne prihvata poruku. Stoga, programiranjem adresnih filtera možemo da omogućimo da čvor prihvati ili odbije poruku (napominjemo da svi prijemni čvorovi primaju sve poruke, ali ne moraju da prihvate sve poruke).

U većini CAN kontrolera, takođe, su ugrađene *maske filtera* (Filter Masks). One se koriste da bi se utvrdilo koji bitovi u identifikatorima poruke se upoređuju sa vrednostima u adresnim filterima. Na primer, ako se sve vrednosti maske filtera podese na 1 s, svi bitovi identifikatora poruke se upoređuju sa svim bitovima adresnih filtera.

Na slici 5.1 prikazan je primer razmene podataka na liniji. U ovom primeru postoje tri čvora, A, B i C. Pretpostavimo da su adresnim filterima u svakom čvoru kontrolera tokom prenosa podataka zadate sledeće vrednosti:

Node A: Acceptance filter value = "00000000011" Node B: Acceptance filter value = "00000011111" Node C: Acceptance filter value = "00000000111"

Pretpostavimo da čvor A (Node A) ima visok prioritet i da prenosi poruku sa identifikatorom poruke podešenim na obrazac bita "00000000111". Druga dva čvora su u prijemnom režimu i upoređuju identifikator poruke na liniji sa vrednostima adresnog filtera. Čvor C (Node C) ima istu vrednost njegovog adresnog filtera kao identifikator poruke i zato njegov kontroler prihvata poruku sa podacima koju je poslao čvor A. Kontroler u čvoru B (Node B) upoređuje vrednost njegovog adresnog filtera sa identifikatorom poruke i zanemaruje poruku sa podacima, jer vrednosti nisu iste.



Slika 5.1 – Razmena poruke sa podacima na liniji

Napominjemo da kada je poruka poslata na liniji, kontroleri svih čvorova primaju ovu poruku, ali možda je neće prihvatiti. Takođe, iako kontroler može da prihvati poruku, ova poruka obično ostaje u prijemnom baferu kontrolera sve dok aplikacioni softver ne postavi zahtev za vraćanje poruke iz bafera i kopira je u njegove unutrašnje strukture podataka.

Na slici 5.2 prikazan je drugi primer u kome čvor A prenosi poruku sa podacima, a čvor B i čvor C prihvataju tu poruku. U ovom primeru, adresni filter čvora B je promenjen u "00000000111".

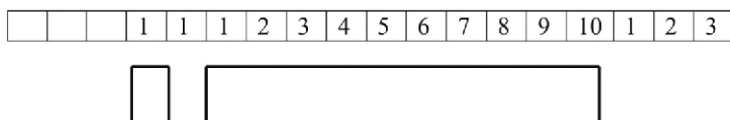
Poglavlje 6 • Usklađivanje vremena na CAN sistemu

Usklađivanje vremena i pravilna sinhronizacija različitih čvorova na CAN komunikacionom sistemu su veoma važni za pravilan rad komunikacione linije. U suštini, svi čvorovi na CAN sistemu su sinhronizovani opadajućom ivicom (od recesivnog do dominantnog prelaza) bita za početak poruke (SOF). Ovo je, u stvari, ograničen način na koji se osigurava pravilna sinhronizacija svih čvorova. U ovom poglavlju ćemo razmotriti usklađivanje vremena na CAN sistemu i metode sinhronizacije.

6.1 Vremensko usklađivanje bita

Signali na CAN sistemu se zasnivaju na NRZ (Non-Return-to-Zero – bez povratka na nulu) šemi signaliziranja u kojoj signal ostaje u istom stanju duže vreme bez ivica. Ovo veoma otežava sinhronizaciju podataka, jer nema ivica signala koje bi se mogle upotrebiti za sinhronizaciju. Na sreću, metoda umetanja bitova osigurava da poruka ne može da zadrži komunikacionu liniju na istom polaritetu više od 5 bitova, tako da je ova funkcija od pomoći pri sinhronizaciji podataka.

Na slici 6.1 prikazan je primer NRZ signala u kome se isti bit obrasca šalje duži vremenski period (u ovom primeru vreme trajanja od 10 bitova), što otežava prijemniku prepoznavanje početka ili kraja podataka, pa su zato potrebne dodatne metode sinhronizacije bita.



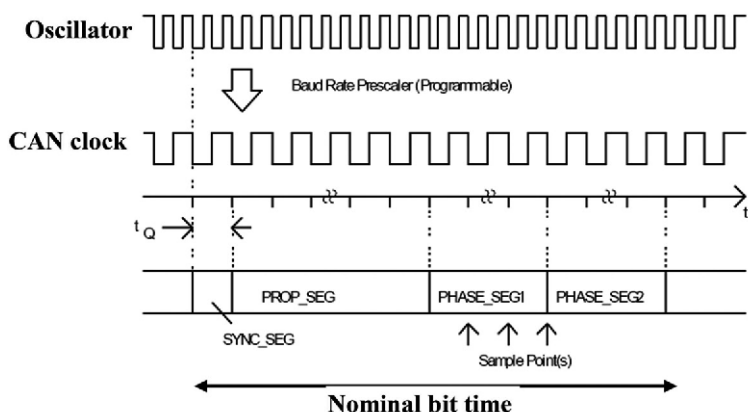
Slika 6.1 – Primer NRZ signala

Usklađivanje vremena na CAN sistemu zasniva se na mehanizmu sata koji je angažovan u CAN kontroleru. U suštini, kristalni sat koji radi bez prekida obezbeđuje impulse pri velikim brzinama prenosa. Unutar kontrolera ova brzina sata je podeljena preskalerom koji programira korisnik, BRP, (preskaler, obično, ima minimalnu brzinu deljenja od 2), a rezultujuća niža brzina sata se koristi kao brzina prenosa (Baud Rate) sata (ili CAN sat čvora) koja određuje vreme čvora.

Brzina jednog bita na CAN sistemu se definiše kao broj bitova koji se prenose svake sekunde bez sinhronizacije. Obrnuto od brzine jednog bita je trajanje jednog bita. Svi uređaji na CAN sistemu moraju da koriste istu brzinu u bitovima, čak i ako svaki čvor može imati unutrašnju frekvenciju radnog takta.

Kao što je prikazano na slici 6.2, nominalna brzina u bitovima na CAN sistemu se sastoji od četiri neprekidajuća vremenska segmenata pod nazivom:

- Segment za sinhronizaciju, *Sync_Seg*
- Segment za propagaciju, *Prop_Seg*
- Segment faze bafera 1, *Phase_Seg1*
- Segment faze bafera 2, *Phase_Seg2*



Slika 6.2 – Oscilator CAN kontrolera i vreme u bitovima

Segment *Sync_Seg* se koristi za sinhronizaciju različitih čvorova na komunikacionoj liniji, a očekuje se da je ivica smeštena u ovom segmentu. Čvorovi će detektovati svaku promenu u *Sync_Seg* i svaki će prilagoditi dužinu njegovog segmenta faze bafera u skladu sa resinhronizacijom čvora. U predajnom čvoru nova vrednost bita se prenosi od početka segmenta *Sync_Seg*. U prijemnom čvoru početak prijemnog bita se očekuje tokom segmenta *Sync_Seg*.

Segment *Prop_Seg* kompenzuje fizičko vreme kašnjenja na liniji, kao što je vreme propagacije od predajnika do prijemnika i obrnuto.

Segmenti *Phase_Seg1* i *Phase_Seg2* kompenzuju greške faze ivice. Ovi segmenti mogu da se produžuju i skraćuju radi sinhronizacije čvora.

Tačka uzorkovanja (Sample Point) (pogledajte sliku 6.2) je tačka u vremenu u kojoj se nalazi stvarna vrednost bita i pojavljuje se na kraju segmenta *Phase_Seg1*.

Svaki segment je podeljen na jedinice koju su poznate kao *kvant*, ili *T*, gde je kvant u osnovi jednak jednom vremenskom intervalu CAN sata (pogledajte sliku 6.2). Željeno trajanje jednog bita može da se postavi podešavanjem broja *T* koji se sastoji od jednog bita poruke i broja *T* koji se sastoji od segmenta u bitu.

Vremenski kvant svakog segmenta može da varira od 1 do 8. Dužine različitih vremenskih segmenata su sledeće:

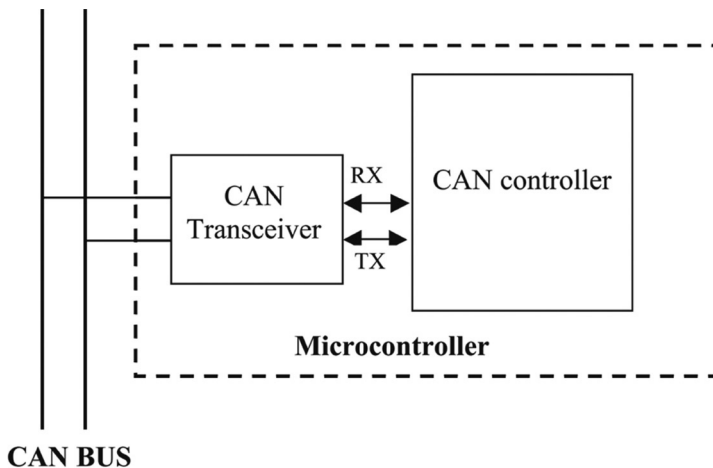
- *Sync_Seg* je uvek 1 *TQ*
- *Prop_Seg* is programabilan od 1 *TQ* do 8 *TQ*
- *Phase_Seg1* je programabilan od 1 *TQ* do 8 *TQ*
- *Phase_Seg2* je programabilan i jednak većem segmentu *Phase_Seg1* ili *IPT*-iju (Information Processing Time – Vreme za obradu informacija o novom bitu). *IPT* je inače jednak 2 *TQ*, ali može da bude jednak 3 *TQ* ako je brzina prenosa podataka preskalera podešena na 1 ili ako su izabrana 3 uzorka po bitu.

Poglavlje 7 • CAN kontroleri

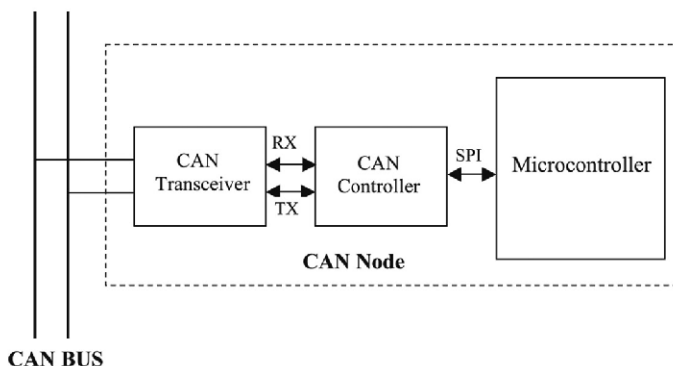
Uopšteno rečeno, CAN kontroler obezbeđuje interfejs između mikrokontrolera i CAN linije. Postoje dva tipa CAN kontrolera: sa ugrađenim primopredajnicima i bez ugrađenih primopredajnika.

CAN primopredajnik obezbeđuje fizički interfejs na CAN liniji. Sastoji se od predajnika signala za liniju i prijemnika. Kontroler je programabilan i sadrži logiku za detekciju greške, adresne filtere, maske i bafere. Kontroler prima CAN podatke sa primopredajnika, pa zatim prihvata ili odbija te podatke, a ako prihvati, šalje podatke na sistem mikrokontrolera. Slično tome, kontroler šalje podatke na CAN liniju kroz CAN primopredajnik.

U nekim savremenim mikrokontrolerima ugrađeni su CAN moduli kontrolera i kola primopredajnika, pa tako mogu biti povezani direktno sa CAN linijom, kao što je prikazano na slici 7.1. Ako mikrokontroler nema ugrađen CAN kontroler i primopredajnik, onda spoljni čipovi kontrolera i primopredajnika mogu da se povežu sa U/I pinovima tog mikrokontrolera, kao što je prikazano na slici 7.2.



Slika 7.1 – Mikrokontroler sa ugrađenim CAN kontrolerom i primopredajnikom



Slika 7.2 – Mikrokontroler sa spoljnim CAN kontrolerom i primopredajnikom

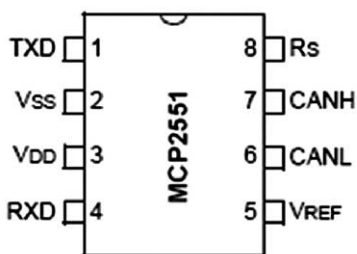
U ovom poglavlju ćemo razmotriti osnovnu strukturu različitih CAN primopredajnika i CAN kontrolera koji su dostupni na tržištu.

7.1 Osnovna struktura CAN primopredajnika

Kao što sam napomenuo ranije u ovom poglavlju, CAN primopredajnik obezbeđuje stvarni fizički interfejs CAN liniji. Tipični CAN primopredajnik sadrži sledeće:

- TX i RX pinove za povezivanje sa kontrolerom
- CAN_H i CAN_L pinove za povezivanje sa CAN linijom
- Pinovi za napajanje

Na slici 7.3 prikazana je konfiguracija pinova MCP2551 tipičnog CAN primopredajnika kompanije Microchip In. (izvor: <http://www.microchip.com>).



Slika 7.3 – Konfiguracija pinova MCP2551 CAN primopredajnika

Unutrašnja struktura MCP25512 čipa prikazana je na slici 7.4. Uočite otvoreni izlaz kolektora pina CAN_H. Čip podržava operacije do 1 Mb/s i pogodan je za 12 V (npr. automobile) i 24 V sisteme. MCP2551 sadrži sklopove za zaštitu od kratkog spoja i mehanizam za automatsko isključenje tokom pregrevanja. Takođe sadrži zaštitu od visokonaponskih tranzijenata. Rad u režimu niskog napajanja omogućava uštedu energije u kritični pogonskim aplikacijama. Na CAN liniji može da se poveže najviše 112 čvorova.

Poglavljje 8 • Programerske alatke CAN linije

Razvoj CAN projekta je lakši kada se koriste programerske alatke. Postoje različite hardverske i softverske alatke koje mogu da pomognu u programiranju, testiranju i izradi projekata zasnovanim na CAN liniji. Ovo su neke od popularnih CAN programerskih alatki:

- razvojne ploče CAN hardvera
- razvojne ploče mikrokrokontrolera za opštu namenu sa CAN modulima
- CAN analizatori
- logički analizatori sa CAN dodatnim opcijama
- CAN simulatori
- CAN stimulatori
- CAN programerske biblioteke za jezike visokog nivoa

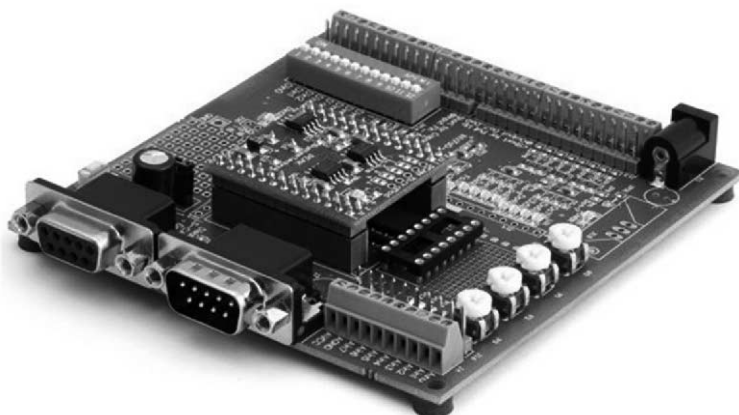
U ovom poglavlju ćemo razmotriti jednu od najčešće korišćenih CAN alatki. Uopšteno govoreći, CAN programerske alatke možemo da podelimo na tri tipa: hardverske programerske alatke, CAN analizatore i softverske programerske alatke. Neke od ovih alatki sadrže i hardver i softver.

8.1 Hardverske programerske alatke

Hardverske programerske alatke su, obično, ploče zasnovane na mikrokontroleru koje se koriste tokom razvoja CAN projekata. Za takve alatke potrebno je poznavanje računarskog programiranja i osnovnog elektronskog inženjerstva. U ovom odeljku ćemo razmotriti neke od hardverskih programerskih alatki.

8.1.1 CAN MicroMOD razvojni komplet

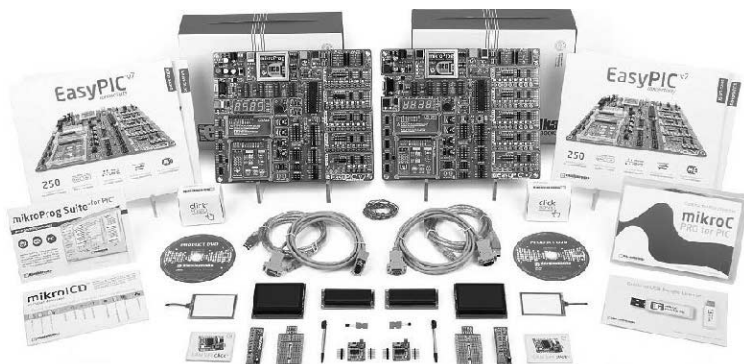
Ovaj razvojni komplet sadrži PC CAN interfejs, MicroMod ploču za evaluaciju (pogledajte sliku 8.1), MicroMod CPU pločicu, adapter za napajanje i CAN kabl od 0,9 m. Korisnici mogu da kreiraju CAN mrežu i eksperimentišu kompletom. Podaci o konfiguraciji se šalju na modul preko CAN linije i svaki čvor na liniji može da se omogućiti za čitanje ili slanje podataka.



Slika 8.1 – CAN MicroMod ploča za evaluaciju (www.peak-system.com)

8.1.2 mikroElektronika CAN komplet za komunikaciju

Ovaj komplet (pogledajte sliku 8.2), koji je proizvela kompanija mikroElektronika, zasniva se na veoma uspešnim razvojnim pločama mikrokontrolera EasyPIC V7. Ovaj paket sadrži kompajler programskog jezika visokog nivoa mikroC Pro za PIC i IDE. Komplet može da se koristi za podešavanje dva čvora CAN mreže (može se proširiti po potrebi).



Slika 8.2 – mikroElektronika CAN komplet za komunikaciju (www.mikroe.com)

Komplet sadrži:

- 2 x EasyPIC v 7 razvojne ploče
- kompajler mikroC PRO za PIC kompajler i IDE sa USB hardverskim ključem
- 2 x CAN SPI Click 3.3V pločica
- 2 x SmarPROTO pločicu
- 2 x EasyConnect pločicu
- 2 x znakovni LCD dimenzije 2 x 16 sa plavim pozadinskim osvetljenjem
- 2 x DS1820 temperaturni senzor
- 2 x grafički LCD dimenzije 128 x 64 koji sadrži TouchPanel
- upredeni kabl (dugačak 2 metra)
- 2 x plastična olovka za TouchPanel
- USB kabl
- štampano uputstvo za upotrebu

8.1.3 RCDK8C CAN razvojni komplet

Ovaj komplet (pogledajte sliku 8.3) je razvila kompanija Renesas i može da se koristi za podešavanje dva čvora CAN mreže. Komplet se isporučuje sa R8C23 ciljnim pločicama sa CAN MCU za rad na mreži, CAN primopredajnicima, CAN kablom sa upredenim paricama i konektorima.

Komplet sadrži sledeće delove:

- 2 pločice sa početnim kompletom (unapred programirane i sadrže demonstracione pločice)
- DC napajanje
- LCD

Poglavlje 9 • Arhitektura ARM mikrokontrolera

Važno je da naučite osnovnu arhitekturu, prednosti, nedostatke i ograničenja mikrokontrolera kako biste mogli da ga koristite u projektu. U CAN projektima iz ove knjige ćemo koristiti veoma popularni 32-bitni ARM mikrokontroler, STM32F407VGT6. Ovaj mikrokontroler je član STM32 porodice ARM mikrokontrolera. U ovom poglavlju ukratko je opisana arhitektura ovog mikrokontrolera i opisane su njegove osnovne funkcije kako biste lako i efikasno izradili projekte.

Konfiguracija radnog takta i ulazno-izlazni portovi su važni delovi svakog mikrokontrolera i koriste se skoro u svim projektima. Kao rezultat toga, oni su detaljno opisani u ovom poglavlju.

9.1 STM32 porodica ARM mikrokontrolera

STM32 porodica 32-bitnog mikrokontrolera se zasniva na ARM Cortex-u i u njoj postoji više od 300 kompatibilnih uređaja. Kao što je opisano u nastavku, porodica obuhvata mikrokontrolere sa arhitekturama Cortex-M4, Cortex-M3 i Cortex-M0.

Cortex-M: Mikrokontroleri iz ove serije počinju nazivima STM32F4 i dizajnirani su za upotrebu u 32-bitnim aplikacijama. Ovo su mikrokontroleri visokih performansi koji imaju DSP i mogućnosti aritmetičke operacije sa pokretnim zarezom, uz 168 MHz radnog takta, do 1 MB fleš memorije, 256 KB SRAM memorije i veliki broj perifernih uređaja, uključujući USB, Ethernet i interfejs kamere. Mikrokontroler PIC32F407CGT6 je član Cortex-M4 porodice.

Cortex-M3: Mikrokontroleri iz ove serije počinju nazivima STM32F1, STM32F2, STM32W ili STM32L1 i dizajnirani su za upotrebu u 16/32-bitnim aplikacijama. STM32F1 uređaji rade na 72 MHz, sadrže do 1 MB fleš memorije, 96 KB RAM memorije i veliki broj perifernih uređaja, uključujući Ethernet i interfejs kamere. STM32W su bežični mikrokontroleri (IEEE 802.15.4) sa frekvencijom radnog takta do 24 MHz, 256 KB fleš memorije i 16 KB SRAM memorije. STM32L1 mikrokontroleri su ultra niskonaponski uređaji koji rade na 32 MHz, sadrže do 384 KB fleš memorije i 48 KB SRAM memorije. Radni napon je do 1,65 V uz struju u standby režimu od samo 0,3 μ A.

Cortex-M0: Mikrokontroleri iz ove serije počinju nazivom STM32F0 i pripadaju manjim uređajima. Frekvencija radnog takta je do 48 MHz i mogu da sadrže do 128 KB fleš memorije i 12 KB SRAM memorije.

9.2 STM32F107VCT6 mikrokontroler

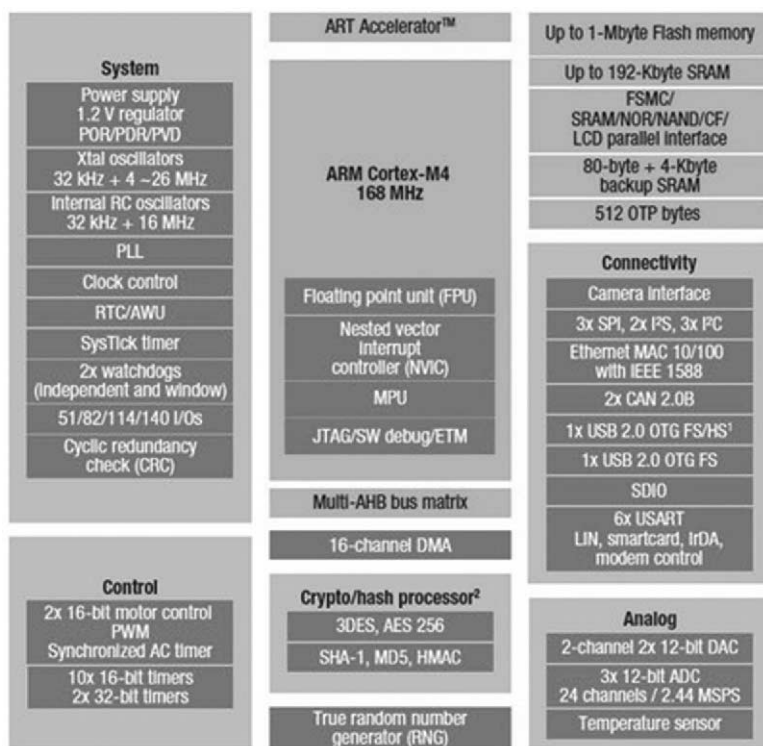
U ovoj knjizi ćemo koristiti veoma popularni ARM mikrokontroler STM32F407VGT6 sa razvojnom pločom *Clicker 2 for STM32* (detalji su priloženi u narednom poglavlju). U ovom poglavlju ćemo razmotriti funkcije mikrokontrolera STM32F407VGT6. Unutrašnja arhitektura ovog mikrokontrolera je veoma složena, pa ćemo pogledati samo važne module koji se koriste u većini projekata, kao što su U/I, tajmeri, ADC i DAC pretvarači, prekidi, I²C, USART i tako dalje. Zainteresovani čitaoci mogu da potraže detaljne informacije u listi sa podacima proizvođača koju mogu preuzeti sa Interneta.

9.2.1 Osnovne funkcije mikrokontrolera STM32F407VCT6

Mikrokontroler STM32F407VCT6 se zasniva na Cortex-M4 arhitekturi i sadrži sledeće osnovne funkcije:

- ARM Cortex-M4 32-bitnu RISC arhitekturu
- jedinicu sa pokretnim zarezom (FPU) i uređaj za digitalnu obradu signala (DSP)
- najviše 168 MHz radne frekvencije
- jednociklično množenje i deljenje hardvera
- do 1 MB fleš memorije
- do 192 KB SRAM memorije
- napajanje od 1,8 do 3,6 V
- rad na -40 °C do -105 °C
- PLL sata
- 4 – 26 MHz kristalni oscilator
- unutrašnji radni takt 16 MHz
- unutrašnja frekvencija oscilatora od 32 KHz za RTC
- nisko napajanje u sleep, stop i standby režimima
- 3 x 12-bitni 24-kanalni ADC pretvarači sa 0 do 3,6 V referentnog napona
- mogućnost uzorkovanja i zadržavanja
- senzor temperature
- 2 x 12-bitni DAC pretvarači
- do 17 tajmera
- do 140 U/I portova (od kojih 138 imaju dopušteno odstupanje od +5 V)
- 16 DMA kontroler toka podataka
- 2 x CAN interfejs (2.0B)
- 6 x USART interfejs (uz LIN i IrDA mogućnosti)
- 3 x SPI interfejs (42 Mb/s)
- 3 x I²S interfejs
- 3 x I²C interfejs
- 2 x USB interfejs
- 2 x nadzorni tajmer
- 2 x 16-bitna PWM kontrola motora
- SDIO interfejs
- 1 x 10/100 Ethernet interfejs
- paralelni interfejs kamere od 8 do 14 bita
- integrisani prekidni kontroler
- generator slučajnih brojeva
- tehnologija Serial Wire Debug i JTAG interfejs
- jedinica za cikličnu proveru redundanse (CRC)

Osnovne funkcije mikrokontrolera STM32F407VGT6 su ukratko prikazane na slici 9.1.



Slika 9.1 – Osnovne funkcije mikrokontrolera STM32F407VCT6

Na slici 9.2 prikazan je raspored pinova (100-pinski paket) mikrokontrolera STM32F407VGT6.

9.2.2 Unutrašnji blok dijagram

Unutrašnji blok dijagram prikazan je na slici 9.3. U gornjem levom uglu se nalazi 168 MHz Cortex-M4 procesor sa fleš i SRAM memorijom, uz DMA kanale, USB i Ethernet module odmah ispod procesora. Interfejs kamere, regulator napona i spoljni kristalni ulazi prikazani su u gornjem desnom uglu slike. Unutrašnja AHB linija (Advanced High Speed Bus) podeljena je na 84 MHz brzu APB2 liniju (Advanced Peripheral Bus 2), koja podržava nekoliko tajmera, SPI liniju, USART-e i ADC kanale. Linija male brzine APB1 (Advanced Peripheral Bus 1) podržava nekoliko tajmera, USART-e, I²C linije, CAN module, DAC i nadzorni tajmer. Radni taktovi za GPIO portove se pokreću iz brze 168 MHz AHB1 linije koja je prikazana u sredini slike. Memorije su prikazane u sredini gornjeg dela slike.

Poglavlje 10 • Programiranje ARM mikrokontrolera

Postoje različiti programski jezici i metode koji su dostupni za programiranje ARM mikrokontrolera. Na primer, mikrokontroleri ARM Cortex-M mogu da se programiraju u jezicima Java, Arduino, C, C++ i drugima. U ovoj knjizi ćemo koristiti programski jezik mikroC Pro for ARM (koji se u ostatku knjige naziva mikroC) i integrisano razvojno okruženje (IDE) koje je razvila mikroElektronika. Jezik mikroC Pro for ARM je moćna programerska alatka bogata funkcijama za ARM mikrokontrolere, dizajnirana tako da programerima obezbedi najlakša moguća rešenja za razvoj aplikacija ugrađenih sistema zasnovanim na ARM-u.

U ovom poglavlju ćemo razmotriti posebne funkcije programskog jezika mikroC kada se koristi u programu STM32 porodice ARM mikrokontrolera (tačnije mikrokontrolera STM32F407VGT6). Ovo poglavlje nije namenjeno za učenje programskog jezika C, jer se pretpostavlja da čitaoci poseduju praktično znanje jezika C u okruženju mikrokontrolera.

10.1 Funkcije jezika mikroC za mikrokontroler STM32F407VCT6

Jezik mikroC je vrlo sličan standardnom jeziku C, ali razvijen je posebno za programiranje mikrokontrolera. Postoje različite verzije jezika mikroC za programiranje PIC mikrokontrolera, serije 8051 mikrokontrolera, AVR porodice, ARM Cortex mikrokontrolera i tako dalje.

- mikroC omogućava programeru da:
 - napiše izvorni kod koristeći ugrađeni uređivač teksta
 - uvrsti sve biblioteke da bi ubrzao razvojni proces
 - lako upravlja projektom
 - prati strukturu programa, promenljive i funkcije
 - generiše sklop i HEX datoteke za programiranje ciljnog procesora
 - koristi integrisane simulatore za debugiranje koda na računaru
 - koristi integrisani hardverski debager da bi ubrzao razvoj programa i testiranje
 - pribavi detaljne izveštaje o potrošnji memorije, stablo poziva, asemblerski listing i još mnogo štošta
 - programira ciljni procesor koristeći integrisani programski softver

mikroC sadrži biblioteke hardvera, uređaje za digitalnu obradu signala, ANSI C i drugo. Neke od biblioteka koje se najčešće koriste su sledeće (postoji više od 60 biblioteka):

- ADC biblioteka
- CAN biblioteka
- EEPROM biblioteka
- Ethernet biblioteka
- GPIO biblioteka
- LCD i grafička LCD biblioteka
- Biblioteka numeričkih tastera
- Biblioteka zvuka
- UART biblioteka
- Biblioteka TFT ekrana
- Biblioteka ekrana osetljivog na dodir

- USB biblioteka
- Biblioteke digitalnih filtera (FIR i IIR)
- FFT biblioteka
- Biblioteka matrice
- ANSI C matematička biblioteka
- Biblioteka dugmadi
- Biblioteka za konverziju
- Biblioteka za podešavanje vremena
- Trigonometrijska biblioteka

mikroC sadrži ugrađenu integrisanu datoteku Help koja pomaže programerima da nauče format različitih naredbi biblioteke i takođe da provere sintaksu naredbi programa. mikroC organizuje aplikacije u projektima i sastoji se od jedne datoteke projekta (nastavak .mcarm) i jedne ili više izvornih datoteki (nastavak .c). IDE pomaže programerima da kreiraju više projekata. Datoteka projekta sadrži sledeće:

- Naziv projekta
- Ciljni mikrokontroler
- Sat uređaja
- Lista izvorne datoteke projekata
- Datoteke zaglavlja
- Binarne datoteke
- Datoteke slika
- Druge datoteke

Na slici 10.1 prikazana je struktura mikroC programa koji je napisan za mikrokontroler STM32F407VGT6. Iako su komentari u ovom programu opcionalni, strogo se preporučuje da ih koristite jer to olakšava razumevanje i održavanje programa. Ovaj veoma jednostavan program uključuje LED svake sekunde. U ovom poglavlju ćemo videti neke od funkcija mikroC jezika za STM32F407VGT6. Većina funkcija opisanih u ovom poglavlju se može primeniti na druge članove porodice STM32.

=====

In this project 16 LEDs are connected to PORTD of a STM32F407VGT6 type ARM Cortex-M4 microcontroller.

The program flashes the LEDs every second.

Author: Dogan Ibrahim

File : LED5.c

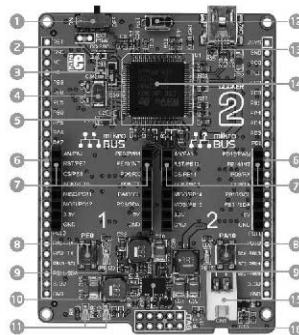
Poglavlje 11 • Razvojna ploča Clicker 2 za STM32

Clicker 2 je moćna razvojna ploča koja koristi STM32F407VGT6 32-bitni ARM Cortex-M4 mikrokontroler koji radi na 168 MHz. Posедуje 1 MB fleš memorije i preko 192 KB SRAM memorije. Ploču (pogledajte sliku 11.1) je razvila i proizvela kompanija mikroElektronika i sadrži sledeće osnovne funkcije:

- STM32F407VGT6 mikrokontroler (100 pinova)
- brzina rada 168 MHz
- 1 MB fleš memorije
- preko 192 KB SRAM memorije
- 25 MHz i 32,768 KHz spoljnih kristala
- 52 programabilnih GPIO pinova
- 2 mikroBUS konektora za Click pločicu (www.mikroe.com)
- USB mini priključak
- 2 LED-a
- 2 tasterska prekidača
- 2 x 26 table za povezivanje
- IC upravljanje napajanjem
- dugme za resetovanje
- spoljni priključak za bateriju

Key features

- 1 CAN/NE switch
- 2 Push button: reset and JTAG/SWD
- 3 Lumps for enabling RTC power supply
- 4 25 MHz crystal oscillator
- 5 32768 KHz crystal oscillator
- 6 2x26 connector pins
- 7 mikroBUS connector pins
- 8 Push buttons
- 9 Address LEDs
- 10 LEDs: USB power, usage, etc.
- 11 Indicator LEDs
- 12 Reset button
- 13 USB mini-B connector
- 14 STM32F407VGT6
- 15 Battery connector
- 16 JTAG programmer connector



Slika 11.1 – Razvojna ploča Clicker 2 za STM32

Mikrokontroler na ploči može da se programira sa računara pomoću programa Bootloader. Razvojna ploča Clicker 2 za STM32 se koristi u svim projektima iz ove knjige koji se zasnivaju na ARM procesoru i CAN komunikacionim linijama. Stoga, više detalja i jednostavan primer programiranja dati su u ostatku ove knjige koji upoznaje čitaoca sa ovom pločom.

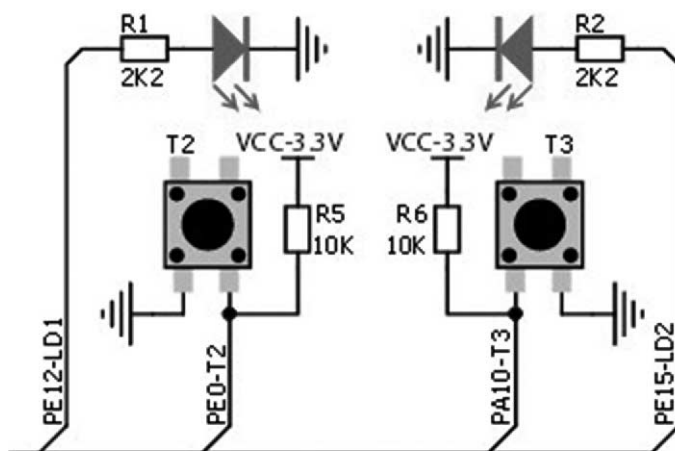
11.1 LED-ovi

Na ploči postoje dva LED-a, LD1 i LD2, povezani sa pinovima porta PE12 i PE15, tim redom. Katode LED-ova su povezane sa uzemljenjem putem otpornika za ograničenje struje od 2,2 K. Dakle, LED-ovi se uključuju kada se logička 1 primeni na odgovarajuće pinove porta.

11.2 Tasterski prekidači

Na ploči postoje dva tasterska prekidača, T2 i T3, povezani sa pinovima porta PE0 i PA10, tim redom. Prekidači se obično postavljaju na logički nivo HIGH kroz otpornike od 10 K. Pritiskanjem prekidača izlaz se prinudno podešava na logičku 0.

Na slici 11.2 prikazane su LED i veze komandnog dugmeta na ploči.



Slika 11.2 – LED i veze komandnog dugmeta

11.3 Napajanje

Ploča može da se napaja iz računara ili odgovarajućeg napajanja od 5 V preko mini USB porta (pogledajte sliku 11.3) ili spoljne litijum polimerne baterije (pogledajte sliku 11.4).

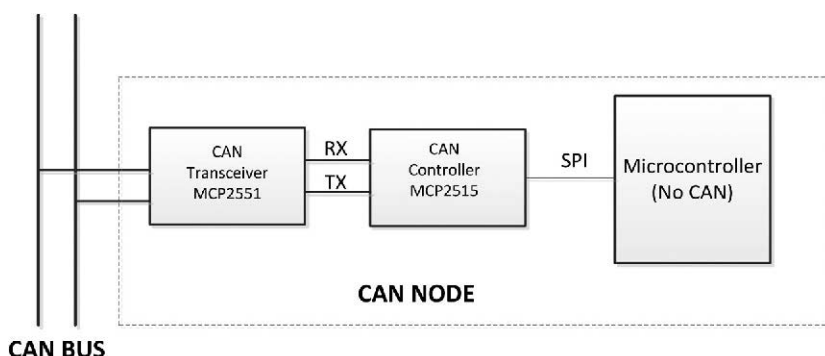


Slika 11.3 – Napajanje iz mini USB porta

Poglavlje 12 • mikroC Pro for ARM sa ugrađenim funkcijama CAN linije

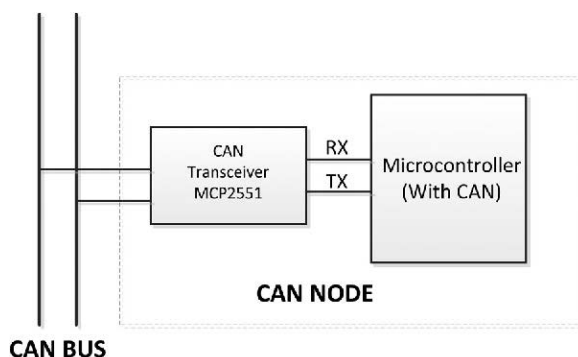
Uopšteno govoreći, u projektima zasnovanim na CAN liniji može da se koristi bilo koji tip mikrokontrolera. Mikrokontroler sa ugrađenim CAN modulom kontrolera (kao što su ARM procesori) će sigurno pojednostaviti dizajn i takođe skratiti vreme razvoja i testiranja.

Na slici 12.1 prikazan je blok dijagram aplikacije CAN linije u kojoj se koristi mikrokontroler opšte namene koji nema ugrađen CAN kontroler. Kao što se može videti na slici, za uspostavljanje CAN čvora potrebni su CAN čip kontrolera (npr. MCP2515) i CAN čip primopredajnika (npr. MCP2551). U takvim aplikacijama komunikacija između mikrokontrolera i CAN kontrolera se, obično, obavlja putem SPI linije.



Slika 12.1 – CAN čvor na kome mikrokontroler nema ugrađeni CAN modul kontrolera

U ovoj knjizi ćemo koristiti mikrokontroler STM32F407VGT6 ARM Cortex-M4 koji ima dva ugrađena CAN modula kontrolera. Jedini dodatni čip koji je potreban za uspostavljanje CAN čvora je CAN čip primopredajnika. Na slici 12.2 prikazan je blok dijagram CAN čvora koji koristi ARM procesor.



Slika 12.2 – CAN čvor na kome mikrokontroler ima ugrađeni CAN modul kontrolera

12.1 Mikrokontroler STM32F407VGT6 sa ugrađenim CAN modulom kontrolera

Važno je poznavati osnovni rad STM32F407VGT6 mikrokontrolera sa CAN modulom pre nego što se razmotre funkcije dostupne za programiranje čipa.

Opis rada koji je dat u ovom poglavlju odnosi se samo na mikrokontroler STM32F407VGT6. Slične mikrokontrolere i više informacija možete pogledati u referentnom priručniku proizvođača (www.STMicroelectronics.com) RM0090 (STM32F405/415, STM32F407/417, STM32F427/437 i STM32F429/439 napredni ARMR 32-bitni MCU), ID dokumenta: DocID018909 Rev 11, 2015.

- Ovo su osnovne funkcije CAN modula koje nas zanimaju:
- 2 CAN modula (CAN1 i CAN2) koja dele 512 bajta SRAM memorije
- podrška za CAN2.0A i CAN2.0B
- podrška za standardne identifikatore (11-bitne) i proširene identifikatore (29-bitne)
- brzina u bitovima je do 1 MB/s
- 3 poštanska sandučeta za prenos
- prioritet slanja koji se može konfigurisati
- 2 prijemna FIFO bafera (sa 3 faze)
- 28 grupa filtera identifikatora (ili adresna filtera) koji se mogu konfigurisati deljena između CAN1 i CAN2

Obezbeđena su tri poštanska sandučeta za prenos radi podešavanja poruka.

CAN čvor koristi adresne filtere da bi odredio da li će prihvatiti primljenu poruku. Sve neprihvaćene poruke se odbacuju. U CAN protokolu adrese čvorova se ne koriste i identifikator je povezan sa sadržajem poruke.

Za skladištenje dolaznih poruka koriste se dva FIFO bafera kojim upravlja hardver, a u svakom od njih mogu se uskladištiti tri kompletne poruke.

CAN modul ima tri režima rada:

Initialization režim: U ovom režimu se inicijalizuje CAN modul. Sve poruke koje se šalju u i iz CAN linije se zaustavljaju i izlazni status linije se podešava na receptivni.

Normal režim: Ovo je normalni režim rada u kome je dozvoljeno slanje i primanje poruka. Normalni režim sledi nakon režima inicijalizacije.

Sleep režim: U ovaj režim se ulazi na zahtev softvera kako bi se smanjila potrošnja energije. U ovom režimu radni takt je zaustavljen.

Pored tri režima, postoje i režimi Test i Debug. Režim Test ima opcije: režim Silent (samo za prijem), režim Loop Back (poslate poruke su primljene i uskladištene) ili režim Loop Back sa režimom Silent.

Poglavlje 13 • Projekti CAN linije u kojima se koriste ARM mikrokontroleri

Razvoj projekta zasnovanom na CAN liniji pomoću mikrokontrolera zahteva dve stvari: hardver i softver. Hardver se, obično, sastoji od razvojne ploče koja sadrži odgovarajući mikrokontroler, a softver je, obično, kompajler jezika visokog nivoa. U ovom poglavlju se kao hardver koristi razvojna ploča *Clicker 2 za STM32* mikrokontroler koja je opisana u Poglavlju 11. Softver projekata se zasniva na *mikroC Pro for ARM* integrisanom razvojnim okruženju koje sadrži kompajler, simulator, debugger i programski softver ARM mikrokontrolera.

U ovom poglavlju je dat dizajn nekoliko projekata zasnovanim na CAN liniji. Svi projekti su opisani uz sledeća podzaglavljja:

- broj i naslov projekta
- blok dijagram projekta
- električna šema projekta
- opis programa pomoću PDL-a
- kompletni listinzi programa
- opis programa
- preporuke za budući rad (ako je potrebno)

U suštini, za razvoj projekata zasnovanim na CAN liniji potrebno je uraditi sledeće:

Prvo napraviti hardver, kreirati čvorove CAN linije i uveriti se da se linija pravilno završava.

Napravite softver čvora CAN linije koristeći integrisano razvojno okruženje *mikroC for ARM* (www.mikroe.com). Softver se sastoji od sledećeg:

- Izračunavanje potrebne vremenske parametre CAN linije.
- Inicijalizacije CAN linije pozivanjem funkcije *CANxInitializeAdvanced*.
- Podešavanje CAN linije u režim inicijalizacije pozivanjem funkcije *CANxSetOperationMode*.
- Podešavanje maske filtera CAN linije pozivanjem funkcije *CANSetFilterScale32*.
- Podešavanje CAN linije u normalni režim pozivanjem funkcije *CANxSetOperationMode*.
- Upotrebe funkcije *CANxWrite* ili *CANxRead* za upisivanje ili čitanje podataka iz CAN linije, tim redom.

Prvi projekat je jednostavan, ali je važan jer opisuje osnovni hardver i softver koji se koriste u svim projektima. Čitaoci treba detaljno da nauče prvi projekat pre nego što pređu na druge projekte.

13.1 Projekat – Daljinska kontrola LED-a

Opis

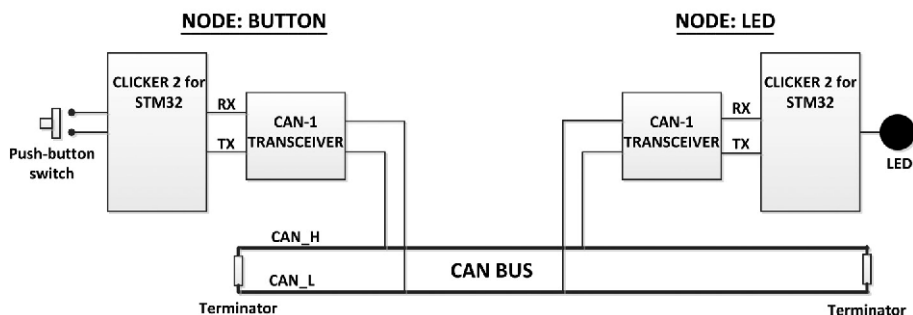
Ovaj projekat se sastoji od dva čvora, koji se zovu BUTTON i LED. Na čvoru BUTTON se nalazi tasterski prekidač. Na čvoru LED nalazi se LED. Pritisak na dugme uključuje LED, a otpuštanje dugmeta isključuje LED. Ovaj proces se kontinuirano ponavlja.

Blok dijagram

Blok dijagram projekta je prikazan na slici 13.1.

Na čvoru BUTTON koristi se *Clicker 2 za STM32 ARM Cortex-M4* kompatibilna razvojna ploča. Ova ploča sadrži dva tasterska prekidača i dva LED-a, kao što je opisano u Poglavlju 11. U ovom projektu koristi se tasterski prekidač povezan sa pinom porta PE0 (bit 0 porta PORTE). Za povezivanje razvojne ploče sa CAN linijom koristi se *CAN-1* ploča primopredajnika (www.mikroe.com). CAN liniju čine kabl sa upredenim paricama dužine 3 m i završni otpornici na svakom kraju.

Na čvoru LED koristi se još jedna *Clicker 2 za STM32* razvojna ploča. U ovom projektu koristi se LED povezan sa pinom porta PE12 (bit 12 porta PORTE). Kao i u prošlom čvoru, za povezivanje drugog čvora sa CAN linijom koristi se *CAN-1* ploča primopredajnika.

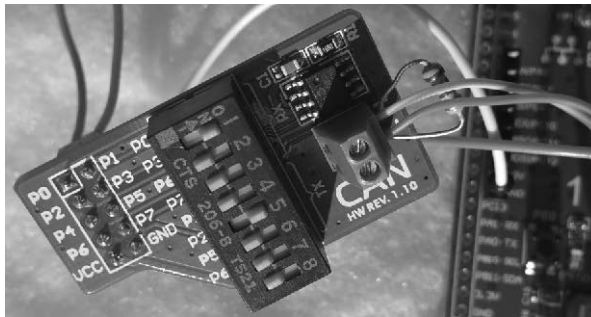


Slika 13.1 – Blok dijagram projekta

Električna šema

Električna šema projekta je prikazana na slici 13.2.

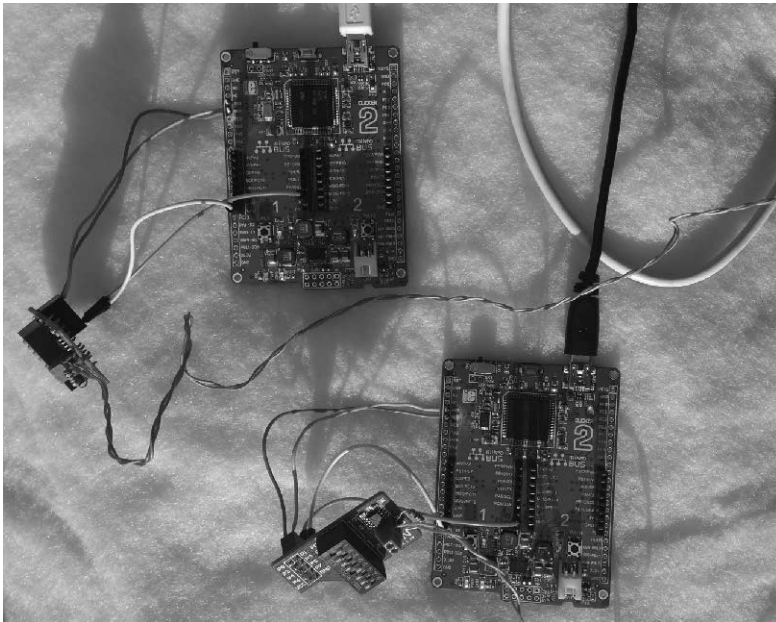
Na čvoru BUTTON koristi se tasterski prekidač povezan sa pinom porta PE0, kao što je opisano ranije. Izlaz ovog dugmeta je podešen na logički nivo HIGH pomoću otpornika, kao što je prikazano na slici 13.3. Dakle, pritiskom na taster podešavamo izlaz prekidača na logički LOW. Pinovi porta PB8 i PB9 su CAN1 modul pinova CAN1_RX i CAN1_TX mikrokontrolera STM32F407VGT6, tim redom, i koriste se za CAN interfejs. Na slici 13.4 prikazana su ova dva pina na ivičnom konektoru ploče Clicker 2 za STM32.



Slika 13.9 – Završetak CAN linije

Hardver čvora na čvoru LED je isti kao na čvoru BUTTON. U ovom projektu se koristi LED povezan sa pinom 12 porta PORTE (PE12) (pogledajte sliku 13.3). LED se uključuje kada mu se pošalje logička 1.

Na slici 13.10 prikazana su dva čvora i CAN linija. U ovom projektu razvojne ploče Clicker 2 za STM32 se napajaju iz USB porta na računaru.



Slika 13.10 – Dva čvora i CAN linija

Opis programa (PDL)

Opis programa je prikazan na slici 13.11 kao jezik za opis programa (PDL), koji se sastoji od naredbi na engleskom jeziku koje opisuju rad programa.

NODE: BUTTON (ID = 3)

```

Configure port pin PE0 as input
Declare CAN bus flags
Declare CAN bus timing parameters
Initialize CAN bus
Set operation mode to initialization
Set filterscale
Set operation mode to normal
    
```

DO FOREVER

```

Get push-button state
Send push-button state to ID 5
Wait 10ms
ENDDO
    
```

NODE: LED (ID = 5)

```

Configure port pin PE12 as output
Declare CAN bus flags
Declare CAN bus timing parameters
Initialize CAN bus
Set operation mode to initialization
Set filterscale
Set operation mode to normal
    
```

DO FOREVER

```

Read push-button state
IF state = 0 THEN
    Turn OFF LED
ELSE
    Turn ON LED
ENDIF
Wait 10ms
ENDDO
    
```

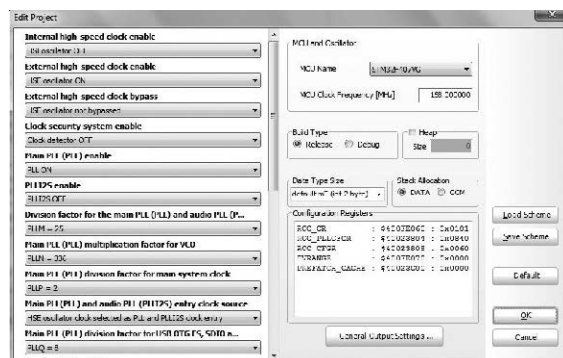
Slika 13.11 – Rad programa (PDL)

Radni takt mikrokontrolera

U ovom i preostalim ARM projektima iz ove knjige razvojna ploča Clicker 2 za STM32 sa mikrokontrolerom STM32F407VGT6 je konfigurisana za rad iz spoljnog kristala od 25 MHz. Mikrokontroler je konfigurisan za efikasan rad sa frekvencijom radnog takta od 168 MHz. Koraci za podešavanje frekvencije radnog takta su ponovljeni u nastavku radi jednostavnosti:

Korak 1: Kliknite *Project -> Edit Project*

Korak 2: Podesite parametre kao što je prikazano na slikama 13.12 i 13.13



Slika 13.12 – Konfigurisanje radnog takta

```

    }
}
}

```

Slika 13.39 – Listing programa čvora DCMOTOR

13.4 Projekat 4 – CAN linija sa 3 čvora

Opis

Ovaj projekat se sastoji od tri čvora, koji se zovu BUTTONS, DCMOTOR i STEPPER. Na čvoru BUTTONS nalaze se dva tasterska prekidača pod nazivom DC_MOTOR i STEPPER_MOTOR. Na čvoru DCMOTOR nalazi se mali DC motor. Na čvoru STEPPER nalazi se jednopolarni koračni motor. Pritiskom na dugme DC_MOTOR aktivira se DC motor 5 sekundi. Slično tome, pritiskom na dugme STEPPER_MOTOR uključuje se jedan ceo obrtaj koračnog motora.

Blok dijagram

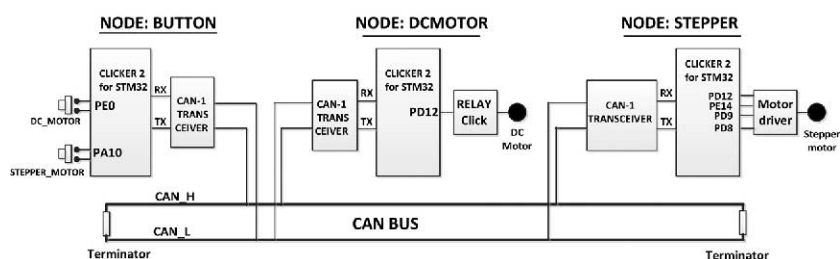
Blok dijagram projekta je prikazan na slici 13.40.

Clicker 2 for STM32 ARM Cortex-M4 kompatibilna razvojna ploča koristi se na svakom čvoru u kombinaciji sa CAN-1 modulom primopredajnika kao u prethodnim projektima. Upredeni kabl CAN linije dužine 3 m koristi se u projektu u kome je svaki čvor povezan preko modula primopredajnika. Mikrokontroler (STM32F407VGT6) na ploči *Clicker 2 za STM32* se napaja iz spoljnog kristala od 25 MHz. Stvarna radna frekvencija takta mikrokontrolera je podešena na 168 MHz, kao što je opisano u projektu 1.

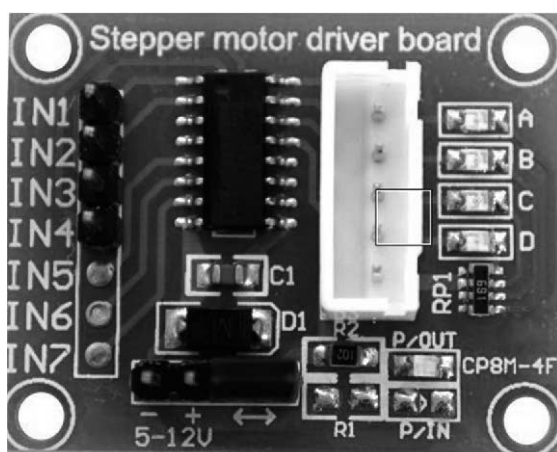
Na razvojnoj ploči *Clicker 2 for STM32* nalaze se dva tasterska prekidača koja su povezana sa pinovima porta PE0 i PA10. Na čvoru BUTTONS tasterski prekidač na pinu porta PE0 naziva se DC_MOTOR, a dugme na pinu porta PA10 se zove STEPPER_MOTOR.

Na čvoru DCMOTOR ploča *Relay Click* se priključuje u mikroBUS Socket 2 razvojne ploče. *Relay Click* ploča sadrži dva releja bez pokretnih delova pod nazivom RELAY1 i RELAY2. Kada se u ovom projektu koristi Socket 2, RELAY1 i RELAY2 se kontrolišu iz pinova porta mikrokontrolera PD12 i PE8, tim redom. U ovom projektu se koristi relej povezan sa pinom PD12 (tj. sa RELAY1) (pomoću dva vijčana priključka u gornjem desnom uglu ploče, pogledajte sliku 13.31). Na slici 13.32 prikazana je ploča *Relay Click* priključena u mikroBUS Socket 2. Kao i u drugim čvorovima, *CAN-1* ploča primopredajnika se koristi za povezivanje CAN linije.

Na čvoru STEPPER koristi se modul drajvera unipolarnog koračnog motora (pogledajte sliku 13.41) u kombinaciji sa koračnim motorom (pogledajte sliku 13.42). Modul drajvera je povezan sa mikroBUS Socket 2 pinovima razvojne ploče *Clicker 2 for STM32* (pogledajte sliku 13.43).



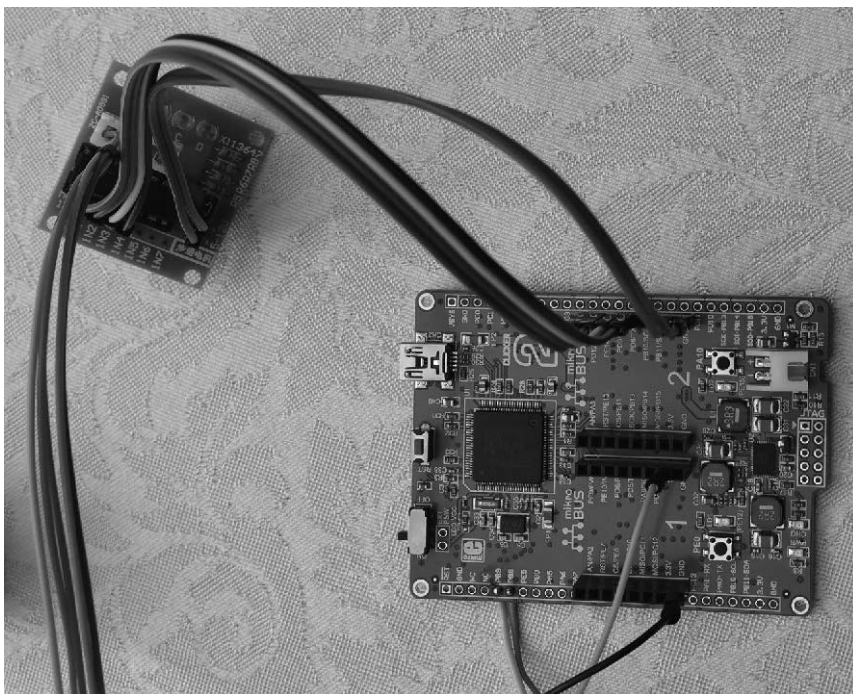
Slika 13.40 – Blok dijagram projekta



Slika 13.41 – Modul drajvera unipolarnog koračnog motora



Slika 13.42 – Koračni motor koji se koristi u ovom projektu



Slika 13.43 – Povezivanje modula dražvera sa razvojnom pločom

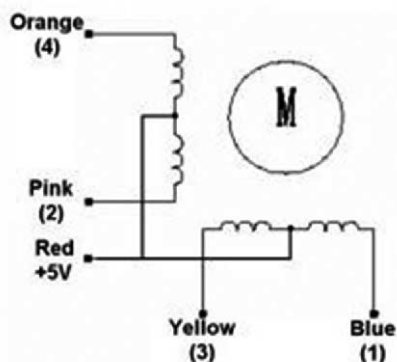
Električna šema

Detalji čvora BUTTONS i DCMOTOR su slični detaljima koji su dati u ranijim projektima.

U ovom projektu se koristi 28BYJ-48 tip malog unipolarnog koračnog motora. Ovaj motor (pogledajte sliku 13.42) ima sledeće funkcije:

- Nazivni napon: 5VDC
- Broj faza: 4
- Odnos promene brzine 1/64
- Koračni ugao 5,625°
- Frekvencija 100 Hz
- Struja 160 mA po namotaju
- Otpor DC-a 50 Ω
- Obrtni momenat vuče >34,3mN.m(120Hz)
- Obrtni momenat samopozicioniranja >34,3mN.m
- Obrtni momenat trenja 600-1200 gf.cm
- Obrtni momenat uvlačenja

Motor ima unutrašnji prenosni mehanizam u približnoj razmeri 1:64. Kao što je prikazano na slici 13.44, postoje dva unutrašnja namotaja sa izvodima koji su označeni ružičastom, narandžastom, žutom, plavom i crvenom bojom.

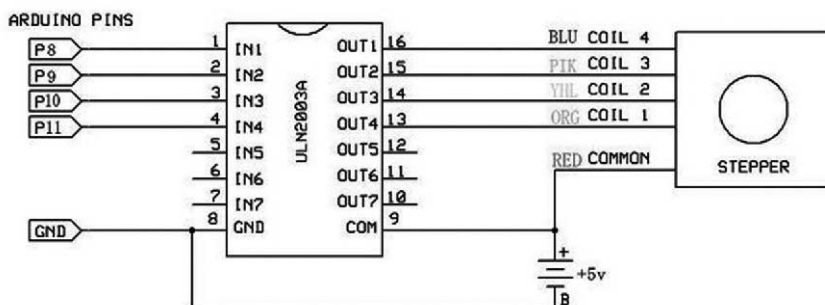


Slika 13.44 – Namotaji 28BYJ-48 koračnog motora

28BYJ-48 motor može da se kontroliše pomoću drajvera unipolarnog koračnog motora prikazanog na slici 13.41. Motor je povezan sa pinovima ABCD na sledeći način:

- A: plavo
- B: ružičasto
- C: žuto
- D: narandžasto

Za rad modula drajvera potrebno je spoljno napajanje od +5 V. Električna šema modula drajvera prikazana je na slici 13.45. Na modulu drajvera koristi se ULN2003A tip IC drajvera.



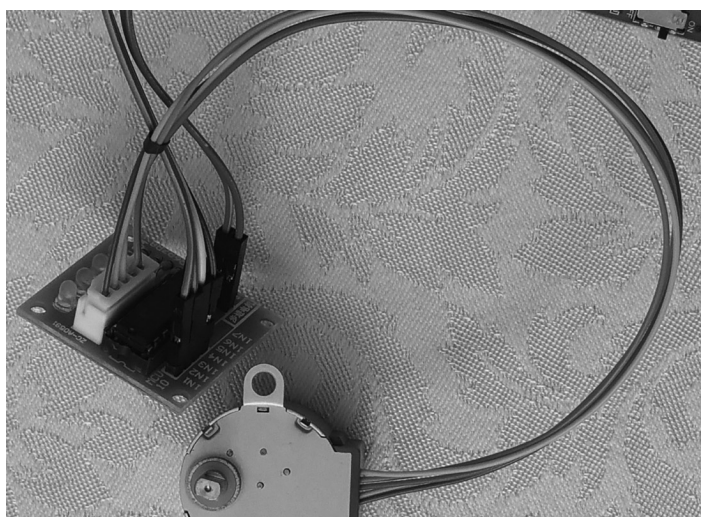
Slika 13.45 – Električna šema modula unipolarnog motora

Modul drajvera motora se kontroliše iz 4 ulazna pina IN1, IN2, IN3 i IN4. Motor može da radi u dva režima: 4-koračni i 8-koračni. U ovom projektu koristi se 8-koračni režim koji čini koračni ugao od $5,625^\circ$. Motor se kontroliše slanjem impulsa na ulaze, kao što je prikazano u tabeli 13.1. Slanjem impulsa obrnutim redom promeniće se smer rotiranja.

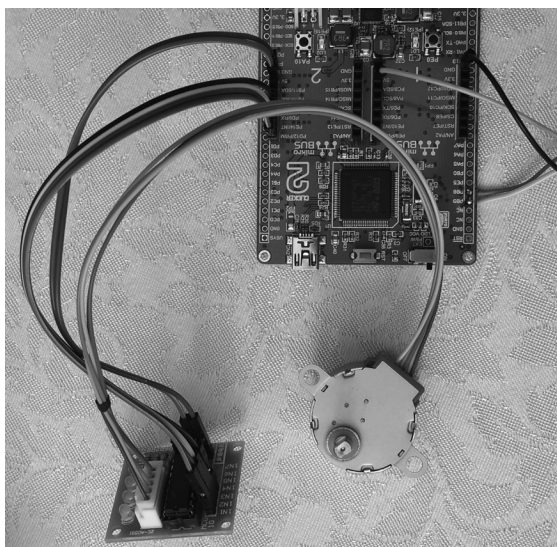
KORACI - >			1	2	3	4	5	6	7	8
D	IN1	Narandžasto	0	0	0	0	0	1	1	1
C	IN2	Žuto	0	0	0	1	1	1	0	0
B	IN3	Ružičasto	0	1	1	1	0	0	0	0
A	IN4	Plavo	1	1	0	0	0	0	0	1

Tabela 13.1 – Impulsi koji su primenjeni na motor za 8-koračni rad

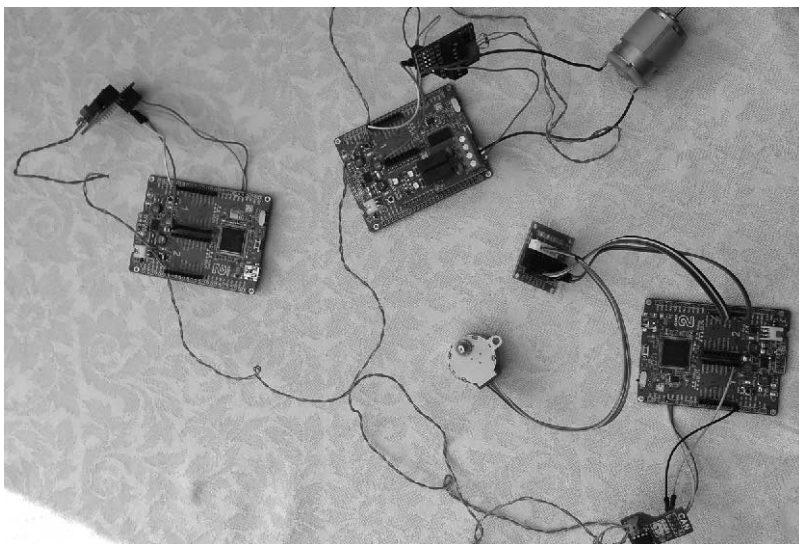
Na slici 13.46 prikazana je veza između motora i modula drajvera. Veza između modula drajvera, motora i razvojne ploče *Clicker 2 for STM32* (čvor: STEPPER) prikazana je na slici 13.47.



Slika 13.46 – Veza motora modula drajvera



Slika 13.47 – Veza između motora, drajvera i razvojne ploče



Slika 13.48 – Sva tri čvora povezana sa CAN linijom

Opis programa (PDL)

Opis programa projekta prikazan je na slici 13.49 kao PDL.

NODE: BUTTONS (ID = 3)

```

Configure pins PE0 and PA10 as input
Declare CAN bus flags
Declare CAN bus timing parameters
Initialize CAN bus
Set operation mode to initialization
Set filterscale
Set operation mode to normal

```

DO FOREVER

```

Get DC_MOTOR state
Send DC_MOTOR state to ID 5
Get STEPPER_MOTOR state
Send STEPPER_MOTOR state to ID7

```

ENDDO**NODE: DCMOTOR (ID = 5)**

```

Configure port pin PD12 as output
Declare CAN bus flags
Declare CAN bus timing parameters
Initialize CAN bus
Set operation mode to initialization
Set filterscale
Set operation mode to normal

```

DO FOREVER

```

Read push-button state
IF state = 1 THEN
    Turn ON relay for 5s

```

ENDIF**ENDDO****NODE: STEPPER (ID = 7)**

```

Configure PD12,PE14,PD8,PD9 as output
Declare CAN bus flags
Declare CAN bus timing parameters
Initialize CAN bus
Set operation mode to initialization
Set filterscale
Set operation mode to normal

```

DO FOREVER

```

Read push-button state
IF state = 1 THEN
    Rotate for 1 revolution

```

ENDIF**ENDDO***Slika 13.49 – Rad programa (PDL)***Radni takt mikrokontrolera**

Radni takt mikrokontrolera je podešen na 168 MHz, kao što je opisano u prethodnim projektima.

Vremenski parametri CAN linije

U ovom projektu se pretpostavlja da CAN linija ima sledeće zahteve (isto kao u prethodnim projektima):

```

Bit rate =          125 kbps
Bus length =        3 m
Bus propagation delay = 5 ns/m

```

Poglavlje 14 • Upotreba analizatora CAN linije

Analizatori linije mogu da budu dragocene alatke prilikom razvoja novog projekta zasnovanom na CAN liniji. Osim toga, ovi uređaji mogu da budu veoma korisni kao sredstvo za učenje pomoću kojih studenti mogu da analiziraju i da nauče strukture prikupljenih poruka i detalje o usklađivanju vremena. Analizatori CAN linije imaju različite funkcije i cene. Uopšteno rečeno, analizator linije se sastoji od malog hardverskog uređaja (pod nazivom hardver analizatora) i namenskog softvera (pod nazivom softver analizatora, koji obično radi na računaru). Jedan kraj hardvera analizatora je priključen na CAN liniju kao čvor, a drugi je obično povezan sa računarom preko USB-a. Nakon što se aktivira softver analizatora, hardver analizatora počinje da prikuplja sve poruke koje su poslate na liniju, uz vremensku oznaku. Prikupljeni podaci se zatim mogu analizirati kako bi se lako detektovale sve greške pri prenosu podataka ili greške pri usklađivanju vremena.

U ovom poglavlju ćemo razmotriti dva različita tipa analizatora CAN linije i primer upotrebe svakog tipa.

14.1 Upotreba logičkog analizatora kao analizatora CAN linije

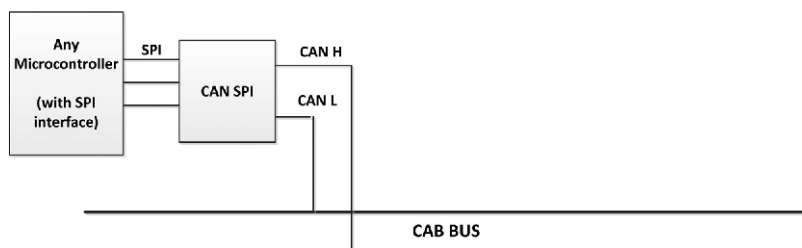
U ovom odeljku ćemo razmotriti upotrebu logičkog analizatora koji sadrži dodatni softver, što omogućava da se koristi kao analizator CAN linije. Uređaj će biti povezan sa CAN linijom da bi se obavilo prikupljanje i analiza poruka koje su poslate na liniju. Ovaj odeljak će biti od koristi čitaocima jer će u njemu biti prikazano slanje stvarnih podataka na CAN liniju. U ovom odeljku je izabran logički analizator LAP-C 16032 kompanije *Zeroplus Logic Cube* (source: <http://www.nkcelectronics.com>). Ovaj analizator je prikazan na slici 14.1.



Slika 14.1 – Logički analizator LAP-C 16032

Poglavlje 15 • CAN SPI

CAN SPI je hardver i softver koji se koristi za kreiranje CAN čvora kada se u projektima bez ugrađenog hardvera CAN kontrolera koristi mikrokontroler. Kao rezultat toga, možemo da kreiramo čvor CAN linije koristeći CAN SPI hardver i bilo koji tip mikrokontrolera. Jedino je potrebno da mikrokontroler podržava SPI komunikaciju (u softveru ili hardveru). CAN SPI hardver je pločica koja sadrži CAN čip kontrolera i primopredajnika. On je obično smešten između U/I pinova mikrokontrolera i CAN linije (pogledajte sliku 15.1).



Slika 15.1 – CAN SPI

U ovom poglavlju će biti opisana CAN SPI biblioteka funkcija *mikroC Pro for ARM* jezika koja je dostupna za STM32 ARM Cortex-M4 mikrokontrolere. U narednom odeljku dat je primer koji prikazuje CAN SPI funkcije koje se mogu koristiti u projektu CAN linije sa dva čvora.

15.1 CAN SPI funkcije

U ovom odeljku prikazane su funkcije koje podržava *mikroC Pro for ARM* za STM32 ARM mikrokontroler (napominjemo da su ove funkcije razvijene za hardver pomoću čipova kontrolera CAN linije MCP2510 ili MCP2515).

Pre nego što razmotrimo ove funkcije, korisno je da pogledamo osnovnu arhitekturu čipova MCP2515 CAN kontrolera.

MCP2515 CAN kontroler sadrži sledeće režime rada:

Configuration Mode: U ovom režimu se inicijalizuje CAN modul. Prenos ili prijem poruka nije dozvoljen u ovom režimu. Brojači greške se brišu u ovom režimu.

Normal Operation Mode: U ovom režimu je dozvoljen prenos i prijem poruka sa podacima.

Listen-Only Mode: Obično se koristi za praćenje statusa komunikacione linije. Modul može da prima poruke, uključujući i greške, ali ne može da šalje poruke.

Loop-Back Mode: Ovaj režim se koristi za testiranje gde poruke mogu da se usmeravaju iz unutrašnjih bafera prenosa radi prijema bafera bez stvarnog slanja.

Poglavlje 16 • Upotreba Arduina sa CAN linijom

Arduino je jedan od najpopularnijih razvojnih sistema mikrokontrolera koji se koristi na računaru i koriste ga studenti elektronskog inženjerstva i hobisti. Prvi Arduino je predstavljen 2005. godine sa ciljem da se omogući jeftina razvojna ploča mikrokontrolera jednostavna za korišćenje.

Arduino ploča se sastoji od Atmel 8, 16 ili 32-bitnog AVR mikrokontrolera sa konektorima i priključcima za spoljni interfejs i programiranje. Dostupan je veliki broj modula koji mogu da se postave na vrhu Arduino ploče i obezbede dodatne periferne mogućnosti, kao što su 7-segmentni ekran, LCD, kontroleri motora, CAN kontroleri, ADC i DAC pretvarači i mnogi drugi.

Arduino ploča podržava integrisano okruženje koje sadrži skriptni jezik, kompajler i softver za programiranje. Ploče su programirane preko serijskih portova. Trenutno postoji mnogo tipova Arduino ploča koje su dostupne u različitim veličinama i sa različitim mogućnostima. Ovo su neki od popularnih tipova:

Pločice za početnike: Arduino Uno, Arduino Pro, Arduino Nano, Arduino starter kit

Poboljšane ploče: Arduino Mega, Arduino Zero, Arduino Due

Pločice kompatibilne sa Internet of Things (Internet Stvari): Arduino Yun, Arduino MKR1000, Arduino Ethernet Shield, Arduino Wi-Fi Shield

Pločice koje se nose kao deo odeće: Arduino Gemma, Lilypad Arduino USB, Lilypad Arduino Simple

U ovom poglavlju nećemo razmatrati detalje o hardveru i softveru. Pretpostavlja se da su čitaoci upoznati sa Arduino Uno pločicom i da su je koristili u najmanje jednom projektu. Postoje mnoge knjige, tekstovi, projekti i podaci na internetu o Arduino hardveru i softveru, pa zainteresovani čitaoci mogu da pogledaju te podatke. U ovom poglavlju ćemo razmatrati upotrebu jedne od najpopularnijih Arduino ploča, Arduino Uno (slika 16.1) u primeru projekta koji se zasniva na CAN liniji. U ovom primeru se koristi modul CAN linije da bi se obezbedio interfejs CAN kontrolera linije za Arduino razvojnu ploču. Napominjemo da je primer dat u ovom poglavlju samo za Arduino Uno pločicu, pa taj primer možda neće raditi sa drugim Arduino pločicama. Takođe, modul CAN linije koji se koristi u ovom poglavlju se može primeniti samo na Arduino Uno.



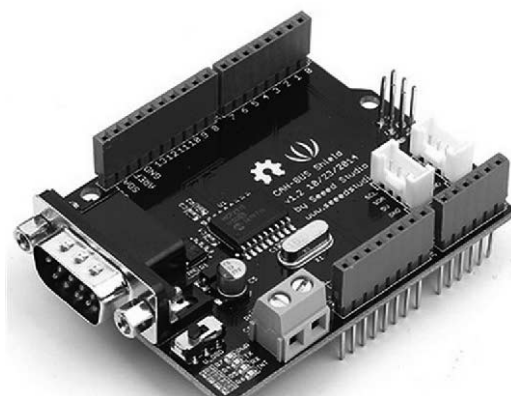
Slika 16.1 – Arduino Uno pločica

16.1 Arduino modul CAN linije

Arduino modul CAN linije je hardver koji se može postaviti na vrhu Arduino Uno pločice i obezbeđuje interfejs CAN linije za Arduino ploču.

Na tržištu je dostupno nekoliko pločica modula CAN linije. U ovom poglavlju se koristi pločica modula koju je proizvela kompanija Seeed Studio (www.seeedstudio.com), koja je prikazana na slici 16.2. Pločica može da se poveže sa CAN linijom preko 9-pinskog konektora ili 2-pinskog vijčanog priključka na ivici ploče. Pločica prihvata MCP2515 CAN kontroler linije sa SPI interfejsom i MCP2551 čipa primopredajnika. Glavne funkcije pločice su sledeće:

- brzina prenosa podataka do 1 MB/s
- SPI interfejs
- standardne (11-bitne) i proširive (29-bitne) poruke sa podacima i zahtevom za podacima
- dva prijemna bafera
- dva LED indikatora
- završni otpornik linije (120 oma) na ploči



Slika 16.2 – Seeed Studio modul CAN linije