

Univerza v Ljubljani  
Fakulteta za elektrotehniko

Zoran Abram

# **Uporaba komunikacijskega protokola J1939 v vozilih**

Diplomska naloga

Mentor: prof. dr. Danijel Vončina

Ljubljana, 2016



## **Zahvala**

Zahvaljujem se mentorju prof. dr. Danijelu Vončini za vso pomoč pri izdelavi diplomske naloge. Prav tako bi se rad zahvalil tudi vsem sodelavcem na projektu, ki so mi nesebično pomagali pri izzivih.



## Vsebina

1.	UVOD.....	5
2.	CAN BUS KOMUNIKACIJA .....	6
3.	VIŠJENIVOJSKI PROTOKOLI .....	16
3.1	CANopen .....	16
3.2	DeviceNet.....	18
3.3	MilCAN .....	20
4.	VIŠJENIVOJSKI PROTOKOL J1939.....	23
4.1	Zgradba sporočila komunikacijskega protokola J1939.....	25
4.2	Načini komunikacije .....	29
4.3	Številka skupine parametrov (PGN) .....	30
4.4	Prenosne funkcije protokola .....	48
4.5	Upravljanje z omrežjem J1939 .....	56
5.	DEMONSTRACIJSKA PLOŠČA PICDEM CAN-LIN 2 .....	70
6.	PRIMERJAVA KOMUNIKATORJEV KVASER LEAF RIGHT IN PEAK SYSTEM PCAN-USB.....	74
7.	REALIZACIJA KUMUNIKACIJE S POMOČJO DEMONSTRACIJSKE PLOŠČE PICDEM CAN-LIN 2.....	79
8.	ZAKLJUČEK.....	93
9.	UPORABLJENA LITERATURA .....	94

## Seznam slik

Slika 2.1: Shematski prikaz CAN vodila in priključitve naprav [1].....	6
Slika 2.2: Grafični prikaz OSI/ISO modela [1]. ....	7
Slika 2.3: NRZ kodiranje [22]. ....	8
Slika 2.4: RZ kodiranje [22]. ....	9
Slika 2.5: Nominalne vrednosti in izgled signala na CAN vodilu [21].....	9
Slika 2.6: Zgradba bita [1].....	10
Slika 2.7: Prikaz segmenta časa razširjanja [1]. ....	10
Slika 2.8: Prikaz razmerja med različnimi časi [1].....	11
Slika 2.9: Shematski prikaz poteka informacije [2]. ....	13
Slika 2.10: Prikaz poteka arbitražnega procesa [9]. ....	13
Slika 2.11: Shema oblike okvirja CAN 1.0A [1].....	14
Slika 3.1: Primer CANopen omrežja [14]. ....	17
Slika 3.2: Prikaz sklopa naprav, povezanih z DeviceNet vodilom [18].....	18
Slika 3.3: Angleško inženirsko vojaško vozilo Terrier [14].....	20
Slika 3.4: Principielna zgradba celotnega sistema [14].....	21
Slika 4.1: Primer omrežja J1939 z napravami, priključenimi nanj [2].....	24
Slika 4.2: Povezava med SPN in PGN parametri [1]. ....	24
Slika 4.3: Podrobna zgradba podatkovnega paketa [2]. ....	25
Slika 4.4: Slika podatkovnega paketa z 11 bitnim ID-jem na vodilu. ....	27
Slika 4.5: Prikaz prepletanja različnih komunikacijskih protokolov znotraj posamezne celote avtomobila [25]. ....	32
Slika 4.6: Zgradba 24 bitne številke skupine parametrov [2].....	36
Slika 4.7: Prikaz preslikave med splošnim razširjenim CAN podatkovnim okvirjem in podatkovnim okvirjem CAN višjenivojskega protokola J1939 [2].....	37
Slika 4.8: PDU 1 format številke skupine parametrov [2]. ....	39
Slika 4.9: PDU 2 format številke skupine parametrov [2]. ....	40
Slika 4.10: Prikazuje arhitekturo CAN sporočila [2]. ....	49
Slika 4.11: Prikazuje zgradbo CAN kontrolnega polja [2].....	49
Slika 4.12: Zgradba večbajtnega sporočila [2]. ....	50
Slika 4.13: Sosledje zaporednih števil in podatkovnih bajtov [2]. ....	50
Slika 4.14: Prikazuje zakasnitve pošiljanj večbajtnega sporočila [2].....	53
Slika 4.15: Zgradba imena [2]. ....	57
Slika 4.16: Odvisnosti posameznih polj znotraj imena [1].....	61

Slika 4.17: Primer CAN bus omrežja [1]. .....	62
Slika 4.18: Proces potrjevanja naslova [2]. .....	64
Slika 4.19: Pošiljanje sporočila v istem trenutku [2]. .....	65
Slika 5.1: CAN-LIN demonstracijska plošča. ....	70
Slika 5.2: PICDEM CAN-LIN demonstracijska plošča [8]. ....	71
Slika 6.1: PCAN-USB komunikator [12]. .....	74
Slika 6.2: Razporeditev kontaktov v konektorju komunikatorja na CAN strain [12]. ....	74
Slika 6.3: Okno za povezavo komunikatorja z osebnim računalnikom. ....	75
Slika 6.4: Glavno okno programa PCAN-View. ....	75
Slika 6.5: Vmesnik za sestavljanje novega sporočila. ....	76
Slika 6.6: Kvaser Leaf Light CAN komunikator [11]. .....	76
Slika 6.7: Izbirno okno, kjer izbiramo med predlogami. ....	77
Slika 6.8: Vmesnik za generiranje sporočil. ....	78
Slika 6.9: Slika prikazuje okolje CANKing vmesnika. ....	78
Slika 7.1: Oprema za nadzor CAN vodila. ....	79
Slika 7.2: Prikaz povezave demonstracijske plošče na osebni računalnik. ....	80
Slika 7.3: Programsko okolje za nadzor komunikacije. ....	81
Slika 7.4: Pregled CAN vodila preko CAN komunikatorja. ....	82
Slika 7.5: Prikaz strojne opreme, potrebne za programiranje mikrokrmilnikov. ....	83
Slika 7.6: Slika prikazuje vezalno shemo demonstracijske plošče [8]. .....	84
Slika 7.7: Prikaz blokovne zgradbe demonstracijske plošče [8]. ....	84
Slika 7.8: Format sporočila J1939 [7]. ....	88
Slika 7.9: Prikaz stanja vodila, ko mikrokrmilnik ne oddaja ničesar. ....	91
Slika 7.10: Prikaz povezanega omrežja s podatkovnim paketom. ....	91
Slika 7.11: Slika prikazuje ne sinhronizirano vodilo. ....	92

## Seznam tabel

Tabela 1: Veličine in enote.....	ix
Tabela 2: Primerjava vrednosti pri standardu ISO-11898-2 in MCP2551 [24]. ....	12
Tabela 3: Prikaz usklajene tabele PGN-jev. ....	28
Tabela 4: Prikaz razčlenitve ene skupine parametrov po posameznih bitih in bajtih v sporočilnem delu enega komunikacijskega okvirja.....	31
Tabela 5: Prikaz usklajene tabele SPN-jev.....	35
Tabela 6: Povezava med PDUf in PDUs [2]. ....	37
Tabela 7: Obseg števil skupin parametrov [2].....	39
Tabela 8: Tip sporočila: prošnja, podrobne pomembne vrednosti [2].....	44
Tabela 9: Kode skupine industrije [2]. ....	60
Tabela 10: Tabela priporočenih naslovov [2].....	62
Tabela 11: Sporočila upravljanja naslovov [2].....	66



## Seznam uporabljenih simbolov

V diplomski nalogi sem uporabil naslednje veličine in simbole:

Tabela 1: Veličine in enote.

Veličina/oznaka		Enota	
Ime	Simbol	Ime	Simbol
Vrtilna hitrost	$N$	Vrtljaji na minuto	vrt/min
izkoristek	$\eta$	Odstotek	%
napetost	$U$	Volt	V
moč	$P$	Watt	W
tok	$I$	Amper	A
temperatura	$T$	Stopinje Celzija	°C
čas	$t$	Milisekunda	ms
frekvenca	$F$	Hertz	Hz
hitrost prenosa	$V$	Kilobit na sekundo	kbit/s

## **Povzetek**

Namen in cilj te diplomske naloge je raziskati komunikacijske protokole. V prvem delu naloge sem opisal 3 komunikacijske protokole, ki imajo za svojo osnovo CAN bus komunikacijo. Opisal sem CANopen, ki je odprtokodni komunikacijski protokol in je ravno zaradi svoje odprtosti zelo zanimiv za širšo javnost. Drugi je DeviceNet, to je komunikacijski protokol, ki se uporablja v industriji. Zadnji komunikacijski protokol, ki sem ga opisal, je MilCAN, ki se uporablja v vojaški industriji, kar je bil glavni razlog, da sem se odločil tudi za njegovo predstavitev.

Najobširnejši je osrednji del naloge, ki podrobno opisuje in razčlenjuje komunikacijski protokol J1939. Podrobneje sem ga opisal s pomočjo arhitekture sporočila, načinov komunikacije, številke skupine parametrov, prenosnih funkcij protokola in na koncu še upravljanja z omrežjem. Vse ugotovitve sem poizkušal podkrepiti s konkretnimi primeri. Osnovo celotne diplomske naloge predstavlja projekt, ki sem ga izvajal. To je bila realizacija komunikacijskega protokola za projekt čolna, zato so tudi vse številke skupine parametrov, SPN-ji, PDU f-ji in PDU s-ji, realni. Zadnja tretjina naloge je realizacija projekta. Najprej je predstavljen pregled demonstracijske plošče, nato pregled strojne in programske opreme ter njihova primerjava in na koncu sledi še realizacija komunikacije.

Ključne besede: Control Area Network, CAN bus, CANopen, MilCAN, DeviceNet, J1939, PIC, mikrokrmilnik, omrežje, komunikacija, tovornjak, avtomobil, industrijska komunikacija



## **Abstract**

My final thesis is about communication protocols. In first part I described three communication protocols, which are based on CAN bus communication. As first, I described CANopen, that is an open code communication protocol and because of its open access, it's very interesting for public. Second one is Device net, which is communication protocol used in industry and the third one is MilCAN, used in military industry and this is the main reason why I chosen it as a focus of my thesis.

Major part of my thesis is detailed description of communication protocol J1939, which is a higher level communication protocol, based on CAN bus communication. I detailed described J1939 through the architecture of message, ways of communication, numbers of parameter groups, transportation functions of protocol and in the end also network management. As much as possible, I tried to base theoretical description on cases of practical use and on a personal project of realization of this communication protocol for implementation in a boat, so numbers of parameter groups SPN, PDU f and PDU are based on real data.

Last part of my thesis is realization. First an overview of demonstration board, followed by an overview of hardware and software and its comparison and in the end realization of communication protocol with its implementation in function is presented.

Key words: Control Area Network, CAN bus, CAN open, MilCAN, DeviceNet, J1939, PIC, micro controller, network, communication, truck, car, industrial communication



## 1. UVOD

Danes je na trgu veliko različnih komunikacijskih protokolov, ki postajajo vedno bolj specifični in kompleksni. Komunikacijski protokol je »računalniški jezik«, ki je strukturiran v obliki določenih pravil in postopkov, ki skrbijo za prenos informacij. Tako poznamo najrazličnejše vrste komunikacijskih protokolov, v računalništvu so najpogostejši TCP/IP, POP, SMTP, FTP, HTTP. Na področju elektrotehnike in elektronike pa so najpogostejše uporabljeni RS232/RS485/RS422, MODBUS, PROFIBUS, CANBUS in drugi. Vsem protokolom je skupno eno, in sicer kako spraviti informacijo od oddajnika do sprejemnika ali obratno, pa naj bosta sprejemnik in oddajnik človek ali elektronska naprava.

V diplomski nalogi se bom osredotočil na CAN bus komunikacijski protokol, podrobneje na njegov višjenivojski protokol J1939. Kratica CAN pomeni Controller Area Network, kar v slovenščini pomeni področno omrežje krmilnikov. CAN je bil razvit v nemškem Boschu (Robert Bosch GmbH) v osemdesetih letih prejšnjega stoletja (1987). Zaradi njegove zgradbe je CAN na področju serijskih vodil postal hitro zelo popularen in veliko proizvajalcev ga je takoj sprejelo. Trdno se je zasidral v avtomobilski industriji. Zaradi integriranega vmesnika za detekcijo napak je zelo zanesljiv, njegova prednost je tudi v hitrosti prenosa podatkov od 100 kbit/s pa do 1 Mbit/s do 1km dolžine. Cenovno ugodni krmilniki so se razširili na področje industrijske avtomatizacije. Nemotena komunikacija med napravami, v CAN bus omrežju, poteka tako, da imajo vse naprave v omrežju enake pravice. Vsaka naprava bo v trenutku oddajanja na vodilo nadrejena ostalim, ostale pa podrejene in bojo poslušale. Za tekoč in hiter prenos podatkov skrbi postopek izogibanja trkov ter arbitražno določanje pomembnosti in s tem nujnosti sporočila. Nujna sporočila imajo prednost.

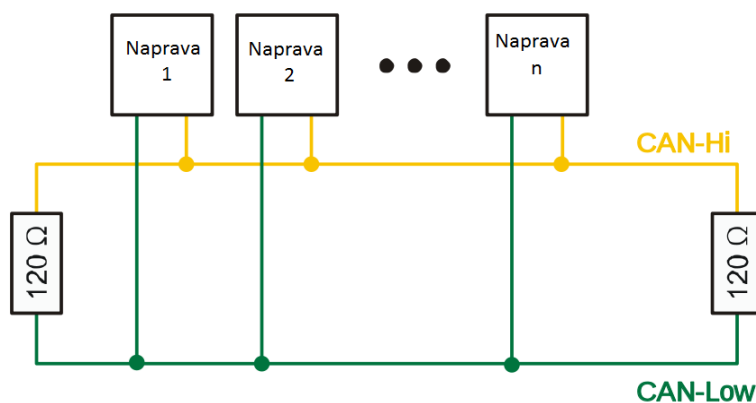
Primer komunikacije bom realiziral na demonstracijski plošči PICDEM CAN-LIN 2, katere osnova bo PIC-ev mikrokrmilnik PIC18F4680, ki podpira omenjeno komunikacijo.

## 2. CAN BUS KOMUNIKACIJA

CAN je bil razvit za potrebe avtomobilske industrije. S tem, ko so postali avtomobili in druga prevozna sredstva vse bolj kompleksni, je postal tudi sistem povezovanja senzorjev in krmilnih enot v avtomobilu vse bolj nepregleden. Z vpeljavo komunikacijskega protokola CAN, kot glavnega povezovalnega vodila, prek katerega potujejo vse informacije znotraj zaključene celote – avtomobila, se je le-ta poenostavil. CAN predstavlja dvožično vodilo, po katerem poteka diferencialni signal, na katerega lahko priključujemo naprave.

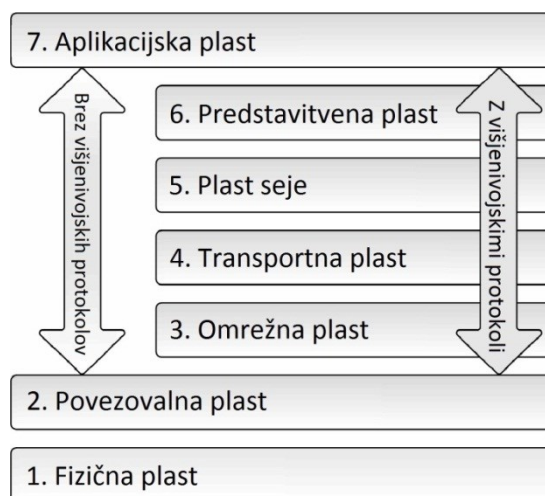
Lastnost, ki odlikuje CAN vodilo, je realizacija prenosa podatkov v realnem času. To je mogoče, ker so sama sporočila razvrščena po pomembnosti. V sami identifikaciji sporočila se skriva prioriteta sporočila, ki predstavlja pomembnost. Tako se na primer ne more zgoditi, da bi zavore morale čakati na prenos sporočila avtoradiu.

CAN je vodilo in teoretično nima omejitev, koliko naprav lahko priključimo na vodilo, omejen je le doseg. Pri postavljanju omrežja moramo upoštevati, da z dolžino vodila povečujemo čas prenosa podatkov. Če na primer hočemo vzpostaviti hitrost prenosa 1Mb/s, je lahko vodilo dolgo 50 m, pri hitrosti prenosa 500 kbit/s je lahko vodilo dolgo že 100 m. CAN vodilo je fizično zgrajeno iz dveh žic. Vrednost nivoja se lahko razlikuje med sprejemno in oddajno stranjo. Višji nivo mora biti nekje med 2,7 in 4,5 V, nižji nivo pa med 0,5 in 2,7 V. Diferencialna razlika mora biti vsaj 1,5 V. Slika 2.1 predstavlja, da moramo na začetek in konec vodila vgraditi zaključni upor.



Slika 2.1: Shematski prikaz CAN vodila in priključitve naprav [1].

CAN je standardiziran po ISO 11898 in ISO 11519-1. Sama komunikacija se nanaša na spodnji dve plasti OSI modela.



Slika 2.2: Grafični prikaz OSI/ISO modela [1].

Na sliki 2.2 vidimo grafični prikaz OSI modela, po katerem je zgrajena tudi CAN komunikacija. Iz slike lahko razberemo, da obstaja možnost nadgradnje osnovnega protokola CAN z višjenivojskimi protokoli, ki so po navadi komercialno naravnani. Vsak avtomobilski proizvajalec vgrajuje v svoja vozila CAN komunikacijo, vendar pa se razlikuje na višjih nivojih. Med enega izmed višjenivojskih protokolov spada tudi J1939, ki se mu bom kasneje v nalogi še posebej posvetil [1].

OSI/ISO model je sestavljen iz naslednjih plasti:

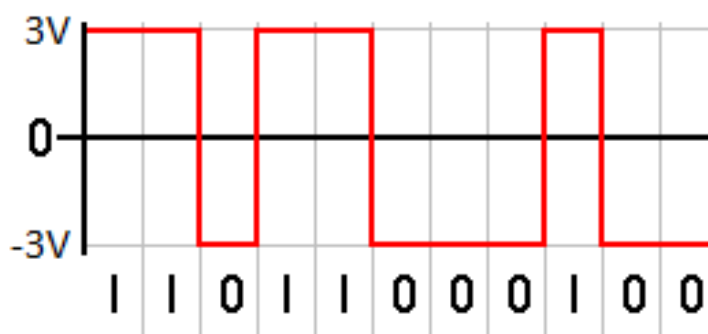
- Aplikacijska plast: je vmesnik med uporabnikom in komunikacijskim omrežjem. Določa protokole, ki omogočajo prenos informacij (datoteke, elektronska pošta ...).
- Predstavitevna plast: skrbi za različne načine kodiranja ali kriptiranja podatkov. Na tej stopnji se pretvarjajo podatki iz osnovnega protokola, ponavadi BCD, v informacijo oblike, ki je znana aplikacijam.
- Sejna plast: nadzira komunikacijo med vmesniki in določa smer komunikacije.
- Transportna plast: definira način prenosa. Gradi posamezne »pakete« informacij. Skrbi tudi za pravilen prenos podatkov.
- Omrežna plast: skrbi za povezavo med uporabniki.
- Povezovalna plast: ta določa enote sporočila, način ugotavljanja napak in kontrolo prenosa.



- Fizična plast: je najnižja plast v OSI modelu in jo opisuje standard ISO 11898. Plast lahko delimo na dve podplasti. V prvi plasti so določila o vrsti signalov in o mehanskih priključkih na medije. Druga podplast pa vsebuje procedure povezovanja računalnikov na omrežja. Glavne lastnosti CAN fizične plasti so: diferenčna dvožilna povezava oziroma vodilo ali sukani par z ali brez opleta. CAN sprejemno/oddajna enota spremeni signal v npr.: FlexCAN, ki je CAN komunikacijski protokol, transformiran v način, ki je znan mikrokrmilniku (Freescape). ECAN pa je signal, ki je znan mikrokrmilniški družini PIC. Na preizkusni plošči PICDEM-LIN 2 je vgrajen CAN oddajnik MCP 2551, ki pretvori signal z vodila v signal, ki je poznan mikrokrmilniku. Pod fizično plast spada sinhronizacija in kodiranje, CAN bus uporablja NZR obliko kodiranja (ang. Non-Return-To-Zero). Nivoji in izgled signala so vidni na sliki 2.5, ki prikazuje nominalne vrednosti in nivoje CAN vodila.

CAN se nanaša na spodnja sloja OSI modela in definira način dostopa do vodila. Ostali sloji so še v razvoju in so definirani z višjenivojskimi protokoli [1].

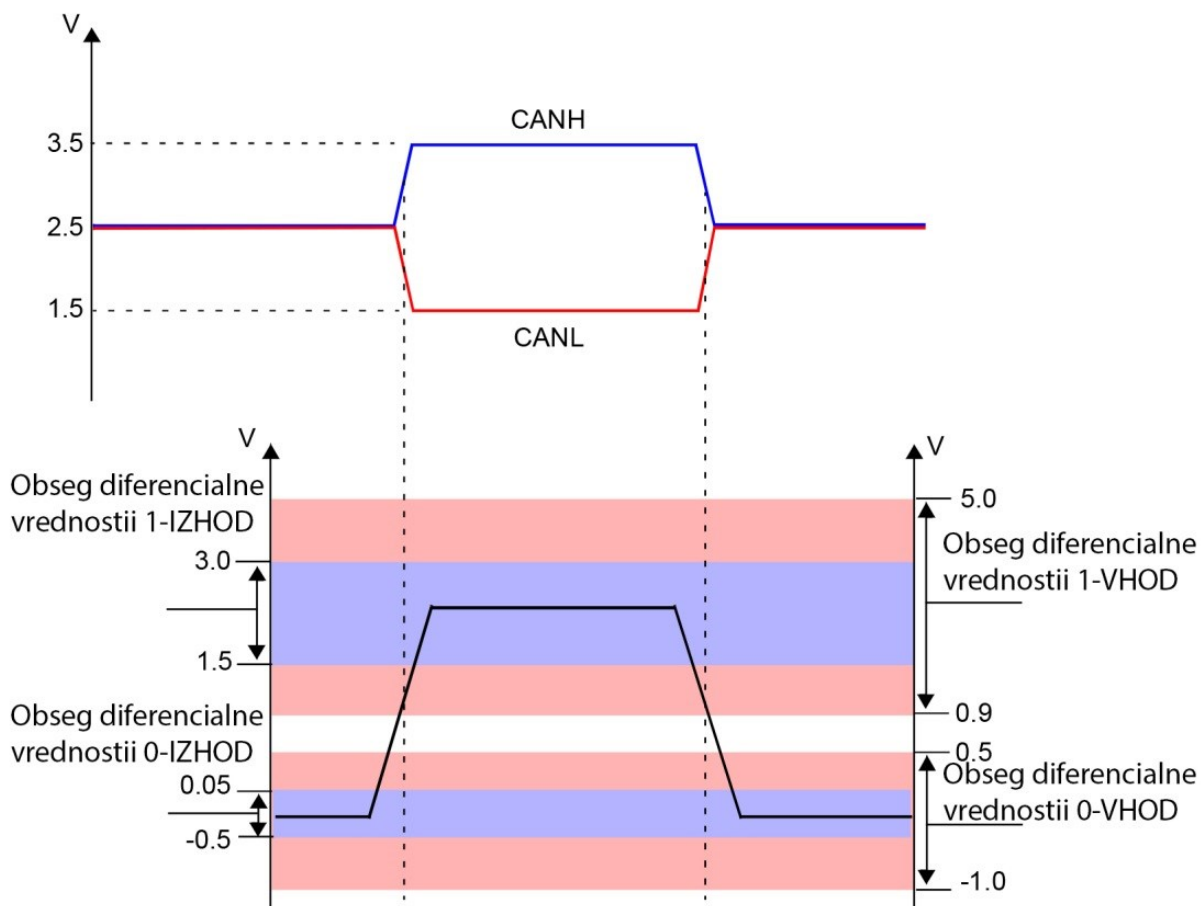
NZR kodiranje prikazuje slika 2.3. Imamo zaporedje binarnih znakov, eno binarno stanje je 1, kar je +3 V, drugi znak je 0, kar pa je -3 V. Ta vrsta kodiranja potrebuje več energije kot npr. RZ način kodiranja (ang. Return To Zero), kjer sta logična ena 3 V in logična nič 0 V. Pri NZR težje pride do napake, saj če pride do izpada in pade nivo na 0 V, je posledično zaznana napaka. Na sliki 2.3 lahko vidimo primer naključnega zaporedja logičnih 1 in 0 v obliki diferencialnega signala od -3 V do 3 V. RZ kodiranje prikazuje slika 2.4 [1].



Slika 2.3: NRZ kodiranje [22].



Slika 2.4: RZ kodiranje [22].

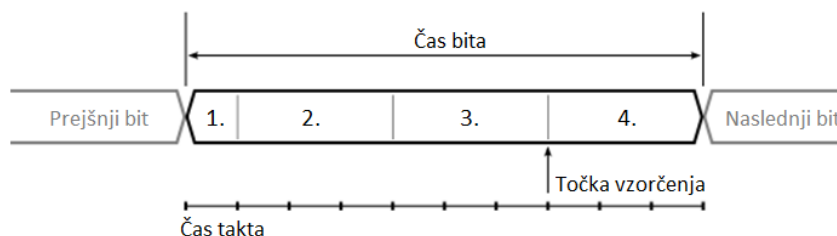


Slika 2.5: Nominalne vrednosti in izgled signala na CAN vodilu [21].

Najprej bomo pogledali na najnižji nivo fizične plasti. CAN uporablja sinhroni prenos podatkov. Za sinhroni prenos podatkov velja, da je samo en začetni ali sinhronizacijski bit na začetku podatkovnega paketa. Medtem ko je pri asinhronskem prenosu podatkov potreben sinhronizacijski bit pred vsakim znakom. Sinhronizacija se vrši z bitom SOF (Slika 2.11).

Vsaka CAN naprava v omrežju ima svoj generator takta. Ker na takt vpliva več dejavnikov, se ti od naprave do naprave razlikujejo. Na njih vplivajo tudi zunanji dejavniki, kot so temperatura, napetostno nihanje ali druge motnje, ki se pojavljajo na napravi. Dokler so ta

nihanja takta znotraj nekih oscilatorskih toleranc, so jih same naprave zmožne kompenzirati. Nastale razlike v bitnih hitrostih rešujejo z resinhronizacijo bitnega okvirja, kajti vsak bit ima na CAN vodilu neko določeno strukturo ali drugače povedano bitni čas. Strukturo bita delimo v štiri segmente, ki jih prikazuje slika 2.6 (sinhronizacijski segment, segment časa razširjanja, fazni segment 1 in fazni segment 2)[2].

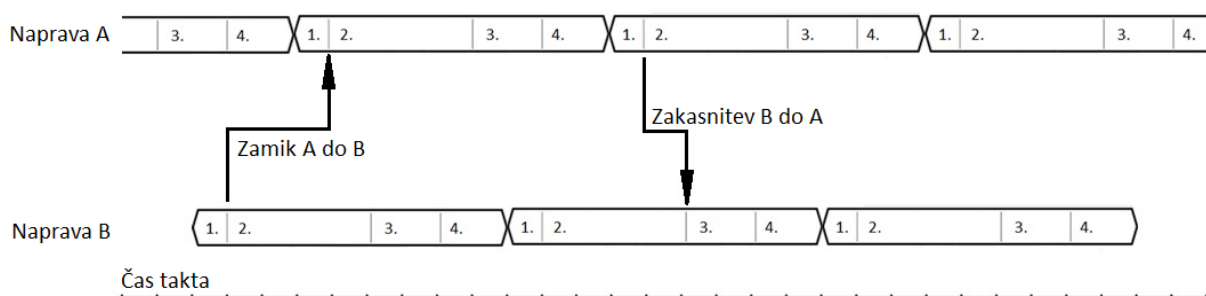


Slika 2.6: Zgradba bita [1].

Na sliki 2.6 lahko podrobneje vidimo samo zgradbo bita:

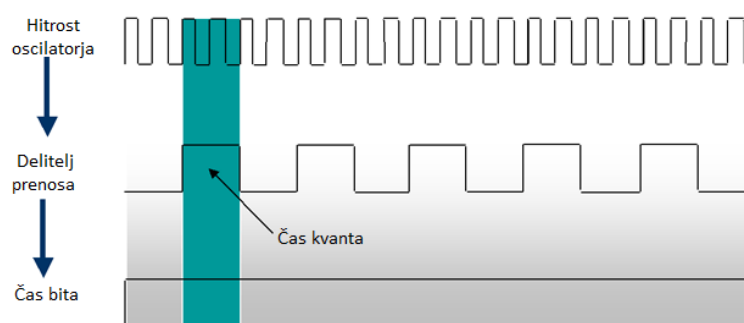
- 1. prikazuje prvi sinhronizacijski segment,
- 2. je segment časa razširjanja,
- 3. je fazni segment 1 in
- 4. je fazni segment 2.

Kot je tudi razvidno s slike, je vsak segment sestavljen iz določljivega časa oziroma določenega števila kvantov. Dolžina kvanta (kvant je čas najmanjša enota časa trajanja takta)  $t_q$  je definirana kot kvocient sistemske ure  $f_{sys}$  CAN naprave in BRP-jevega (ang. Boud Rate Prescaler) delitelja hitrosti prenosa. Če hočemo, da komunikacija poteka brez izpadov, moramo zagotoviti kontinuirano resinhronizacijo, kar pa dosežemo z ustavitvijo omenjenih štirih segmentov pred in po točki vzorčenja. Točka vzorčenja je točka, kjer ugotavljamo vrednost logičnega bita, in sicer, ali je to logična ena ali logična nič. Sinhronizacijski signal traja en kvant in v tem času se sinhronizirajo naprave na vodilu. V sinhronizacijskem segmentu se tudi zgodi sprememba stanja na vodilu. Segment časa traja od 1 do 8 časovnih kvantov in je sestavljen iz fizične zakasnitve, ki je čas razširjanja signala od oddajnika do sprejemnika in internih zakasnitev CAN naprav [1].



Slika 2.7: Prikaz segmenta časa razširjanja [1].

Na sliki 2.7 vidimo dve oddajni napravi (Naprava A in Naprava B), ki izvajata arbitražo. Postaja A je pred postajo B oddala startni bit za manj kot celotni bitni interval, zato se mora postaja B sinhronizirati s sprejeto fronto. To pomeni, da so časovni segmenti postaje B premaknjeni v odvisnosti od postaje A (Zamik A do B). V naslednjem pošiljanju pa postaja B pošlje paket z višjo prioriteto in zmaga v arbitražnem procesu. To se zgodi v trenutku, ko pošlje naprava B dominantni bit, naprava A pa recesivnega. Dominantni bit iz naprave B bo k postaji A prispel po zakasnitvi B do A. Do sinhronizacije pride na prehodih iz recesivnega stanja v dominantno. Sprememba stanja je sinhronizirana le, če se zgodi znotraj sinhronizacijskega segmenta. Samo spremembo zaznamo z vzorčenjem stanja, ki se izvaja na vsak kvant. Razmerje med trajanjem kvanta prikazuje slika 2.8 [1].



Slika 2.8: Prikaz razmerja med različnimi časi [1].

Fazna napaka je razdalja med fronto in med koncem sinhronizacijskega signala. CAN uporablja dva tipa sinhronizacije. Prvi je trda sinhronizacija in se dogaja na začetku paketa. S trdo sinhronizacijo prisilimo, da sprememba stanja leži znotraj sinhronizacijskega intervala. Druga vrste sinhronizacije je resinhronizacija, ki podaljšuje ali krajša čas bita. Točka branja se premakne na začetek spremembe logičnega stanja. Če je fazna napaka, ki je povzročila resinhronizacijo, negativna, se fazni segment 2 skrajša, če pa je fazna napaka pozitivna, pa se fazni segment 1 podaljša.

ISO 11898-2 je standard fizične plasti, ki se najpogosteje uporablja in opisuje funkcijski dostop do vodila. Kot vidimo na sliki 2.1, je impedanca vodila  $120\ \Omega$ , nivoji na CAN vodilu pa so razvidni na sliki 2.5. Primerjava vrednosti med standardom ISO-11898-2 in vrednostmi, ki jih uporablja CAN sprejemnik MPC2551, je vidna v tabeli 2 [1].

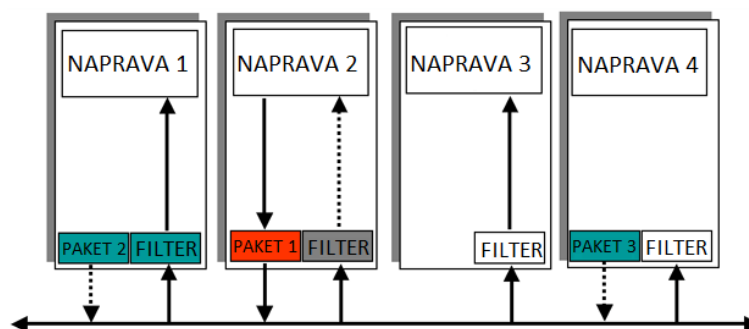
Tabela 2: Primerjava vrednosti pri standardu ISO-11898-2 in MCP2551 [24].

Parameter	ISO-11898-4		MCP2551		Enote
	min	max	min	max	
Enosmerna napetost na CANH in CANL	-3	32	-40	40	V
Prehodna napetost na CANH in CANL	-150	100	-250	250	V
Normalna napetost na vodilu	-2,0	7,0	-12	12	V
Izhodna napetost ob vrednosti logične 0	2,0	3,0	2,0	3,0	V
Izhodna diferencialna vrednost ob logični 0	-500	50	-500	50	mV
Razlika notranje upornosti	10	100	20	100	kΩ
Vhodna upornost v normalnem načinu	5,0	50	5,0	50	kΩ
Izhodna diferencialna vrednost ob logični 1	1,5	3,0	1,5	3,0	V
Vrednost CANH pri logični vrednosti 1	2,75	4,50	2,75	4,50	V
Vrednost CANL pri logični vrednosti 1	0,50	2,25	0,50	2,25	V

Vsak podatkovni paket ima stalno strukturo, razlikuje se le po protokolu. Poznamo dva protokola, ki sta CAN 2.0A in CAN 2.0B. Razlika je v velikosti samega paketa. Po standardu CAN 2.0A se paket začne z 11 bitno identifikacijo, CAN 2.0B pa z 29 bitno identifikacijo.

Leta 2012 pa je Bosch izdal standard CAN FD 1.0. posebnost tega standarda so sporočila z različno velikostjo podatkovnega dela. Same naprave so kompatibilne s standardom CAN 2.0, do omenjene nadgradnje pa je prišlo, da bi se omrežje pohitri, ker ne bi bilo potrebno vsakič pošiljati celotnega podatkovnega dela, temveč se prilagaja dejanski potrebi.

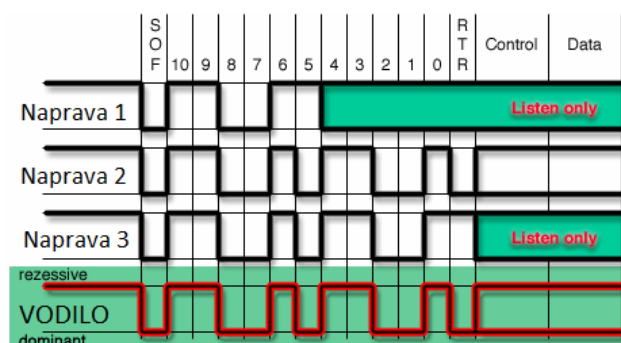
CAN ne uporablja naslovov vozlišč, vendar razpoznavna številka pove, komu so informacije namenjene ter njihovo pomembnost. Primer pošiljanja lahko pogledamo s pomočjo slike 2.9. Naprava 1, naprava 2 in naprava 4 imajo pripravljeno informacijo za pošiljanje. Ker ima naprava 2 informacijo z višjo prioriteto, napravi 1 in 4 odnehata s pošiljanjem in bosta nadaljevali takoj, ko se bo vodilo sprostilo. Hkrati tudi vsi poslušajo na vodilu in če je podatek zanje pomemben ali ne. Razni filtri se nastavijo preko programske opreme. Naprava 2 posluša svoje oddajanje zaradi preverjanja pravilnosti odpošiljanja, vsi ostali pa preverjajo, če so podatki namenjeni njim. Kot je razvidno iz slike 2.9, podatke, ki jih pošlje naprava 2, sprejmeta naprava 1 in naprava 3.



Slika 2.9: Shematski prikaz poteka informacije [2].

Primer arbitražnega procesa poteka po načelu logičnih IN vrat. V mirovanju je na vodilu predpostavljena vrednost logična 1, torej višji nivo. Ko začne oddajati sporočilo več naprav, kar lahko vidimo na sliki 2.9, bodo oddajale vse naprave do trenutka, dokler ne bo ena od naprav oddala vrednost logične 0. Le-ta je namreč dominantna vrednost in jo vodilo prevzame kot bolj pomembno od vrednosti 1. Primer poteka arbitražnega procesa je razviden s slike 2.10. Na sliki 2.9 vidimo shematski potek oddajanja, medtem ko nam slika 2.10 prikazuje podrobneje po bitih, kako dejansko poteka arbitražni proces. Vodilo ima v mirovanju predpostavljeno vrednost 1. Prvi bit, ki ga naprave oddajo, je SOF (ang. Start Of Frame), ki nakazuje začetek oddajanja in je hkrati sinhronizacijski bit. Nato pa se začne oddajanje 11 bitne identifikacije sporočila. Kasneje bomo v poglavju, kjer bo predstavljen višjenivojski protokol J1939, govorili o 29 bitih prepoznavne številke sporočila [13].

Arbitraža na sliki 2.10 poteka na naslednji način; vse tri naprave kot prva dva bita oddajo logično 1. Naslednja dva bita vseh naprav je logična 0, nato naprava 1 odda logično 1, napravi 2 in 3 pa oddata logično 0. Ker je 0 nad logično 1, naprava 1 izgubi arbitražo in začne s poslušanjem, če je sporočilo mogoče namenjeno njej. Nato sta v stanju oddajanja še napravi 2 in 3. Do razlike v oddajanju pride šele pri 12. RTR bitu, kjer tudi naprava 3 začne poslušati. Sam pomen posameznih bitov pa bo opisan v nadaljevanju naloge.



Slika 2.10: Prikaz poteka arbitražnega procesa [9].

Kot lahko vidimo iz slike 2.10, CAN uporablja NRZ obliko signala. Razlika med CAN 1.0A in CAN 2.0B je v identifikacijski številki. CAN 1.0A ima 11 bitno predstavitevno številko, CAN 2.0B pa 29 bitno (11 bitno število + 18 bitni podaljšek). Samo zgradbo 29 bitne identifikacijske številke si bomo pogledali v poglavju o komunikacijskem protokolu J1939. Zgradbo paketa po protokolu CAN 1.0A prikazuje slika 2.11 [13].



Slika 2.11: Shema oblike okvirja CAN 1.0A [1].

SOF napoveduje začetek paketa (ang. Start OF Frame), hkrati pa služi kot sinhronizacijski bit. Temu sledi 11 identifikacijskih bitov, zatem je bit RTR, z aktivacijo katerega zahtevamo odgovor (ang. Remote Transmission Request). Sledi bit IDE (ang. Id Extendet), ki napoveduje, ali v nadaljevanju sledijo informacije oziroma podatki v sporočilu, ali pa sledi še 18 identifikacijskih bitov. R0 je rezervirani bit, njegova vrednost je predpostavljena s protokolom. DLC (ang. Data Length Code) določa število zlogov in obsega 4 bite. V podatkovnem paketu sledi Data, to je podatkovno polje, ki vsebuje od 0 do 8 bajtov podatkov. V primeru, ko je bit RTR (ang. Remote Request) postavljen, v podatkovnem delu sporočila ni podatkov. Polju Data sledi polje CRC (ang. Cyclic Redundancy Check). CRC obsega 15 bitov za preverjanje pravilnosti poslanega sporočila in en ločilni recesivni bit. Za preverjanje je izbran polinom 15. stopnje in je takšne oblike, kot prikazuje enačba 2.1.

$$x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1 \quad (2.1)$$

CRC ostanek se računa od vključno SOF bita. En recesivni bit, ki ga kasneje vse naprave, ki pravilno sprejmejo paket, spremenijo v dominantnega, imenujemo ACK bit ter drugega, ki je ločilna recesivna 1, sestavljata polje za potrditev. Sledi naznanitev konca okvirja EOF (ang. End Of Frame). EOF je sestavljen iz sedmih bitov, ki so recesivne enice, sledi časovni premor, ki traja tri bite recesivnih enic. Zatem pa se začne novo oddajanje ali sprejemanje, lahko pa vodilo pade v stanje pripravljenosti [1].

Reakcijski čas na prejeto sporočilo CAN komunikacije je odvisen od prioritete, ki ga samo sporočilo ima. Ker CAN komunikacija pošlje najprej sporočila z višjo prioriteto, ima določeno sporočilo fiksno prioriteto. Zaradi omenjenih razlogov se lahko zahtevana odzivnost

zagotovi samo sporočilom z višjo prioriteto, za vsa sporočila pa se lahko določi zgornja meja odzivnega časa na osnovi poznanih parametrov omrežja.

S pomočjo Tindellovega pristopa lahko določimo zgornjo mejo odzivnega časa. V omrežju je znano število sporočil, poznamo tudi, koliko naprav je v omrežju. Vsa sporočila so urejena po padajočih prioritetah. Upoštevati moramo, da bo skozi potek komunikacije prihajalo do novih sporočil z najrazličnejšimi prioritetami, saj je izvor sporočil vezan na dogodke, ki se odvijajo. Taka sporočila so sporadična. Ker je sam sistem zgrajen in teče z nekim taktom, poznamo najkrajši čas generiranja nekega istega sporočila. Za vsako sporočilo pa imamo podan tudi rok, in sicer kakšen je najdaljši čas, ko mora biti sporočilo poslano. Nas zanima odzivni čas kot najdaljši možni čas oddaje sporočila. Ker je CAN komunikacija, ki poteka v realnem času, moramo zagotoviti, da je vsako sporočilo preneseno do skrajnega roka, tudi tisto, ki ima najnižjo prioriteto. Da je pogoj izpolnjen mora biti skrajni rok v vsakem primeru večji ali vsaj enak odzivnemu času [2].

Da se sporočilo ne izgubi, mora biti oddano prej, kot se na isto mesto generira novo sporočilo. Ker ne moremo zagotovo trditi, v kakšnem času bo samo sporočilo oddano, govorimo o negotovosti razvrstitve. Zagotoviti moramo le, da bo odzivni čas manjši od razlike ponovitvenega cikla in časa nastajanja sporočila. Odzivni čas za najneugodnejše razmere je seštevek časa prenašanja sporočila in časa čakanja na oddajo sporočila, dodati pa moramo še čas, ki je potreben za razvrstitev. Čas trajanja enega bita je čas, ki je odvisen od velikosti samega sporočila. Čas sporočila v CAN 1.0A je 47 začetnih in zaključnih bitov ter število vseh podatkovnih bitov v osnih bajtih. Sama komunikacija poteka tako, da na vsakih 5 enakih bitov CAN sam vrine še en bit, zato je hitrost prenosa odvisna od seštevka vseh naštetih. Čas čakanja na oddajo sporočila je seštevek čakanja na konec prenosa tekočega sporočila in skupnega časa prenašanja sporočil z višjo prioriteto [2].



### 3. VIŠJENIVOJSKI PROTOKOLI

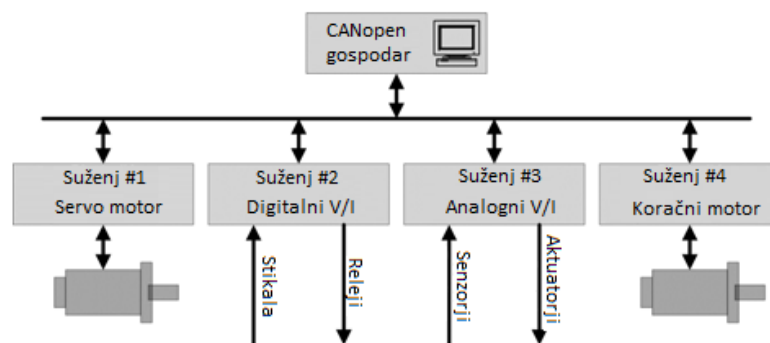
Ker pri CAN protokolu ni definiranih vmesnih nivojev, jih definira vsak sam. Zato se je tudi razvilo večje število višjenivojskih protokolov. Zlasti v avtomatiki je manjkala možnost nastavljanja master/slave. Osnovni CAN protokol ima predpostavljeno, da so v omrežju vse naprave enakovredne in je oblike master/master (multimaster). Najbolj razširjeni višjenivojski protokoli, ki so nastali kot nadgradnja CAN protokola so: CANopen, DeviceNet, J1939, Isobus, MilCAN ...

CAN definira samo okvirje in predstavitvena števila posameznih sporočil. Definicije objektov vozlišč, zgradbe podatkovnega dela sporočila pa so definirana preko višjenivojskih sporočil [2].

#### 3.1 CANopen

CANopen je odprtokodni protokol, čigar specifikacije so brezplačne. Ima isti najnižji stopnji OSI modela, isto ima tudi zmožnost komunikacije v realnem času in podpira iste hitrosti prenašanja podatkov kot CAN bus. Sam standard ponuja možnosti in mehanizme za sinhronizacijo in konfiguracijo obnašanja omrežja. CANopen predstavlja standardizirano izvedbo specifikacije CAL (ang. Can Application Layer), ki so ga razvili člani združenja CiA (ang. Can In Automotion) leta 1995. Na začetku sta CANopen definirala dva standarda. Prvi je bil DS 301, ki definira aplikacijsko plast. Drugi pa je DSP 302, ta pa govori o načrtu standarda za ogrodje programskih naprav [14].

Dodatki k standardu so profili, ki opisujejo modele komunikacije in podatke same. Objekt, ki je definiran v okviru profila, je lahko procesni podatek, ukaz ali funkcija. Slika 3.1 prikazuje primer CANopen omrežja.



Slika 3.1: Primer CANopen omrežja [14].

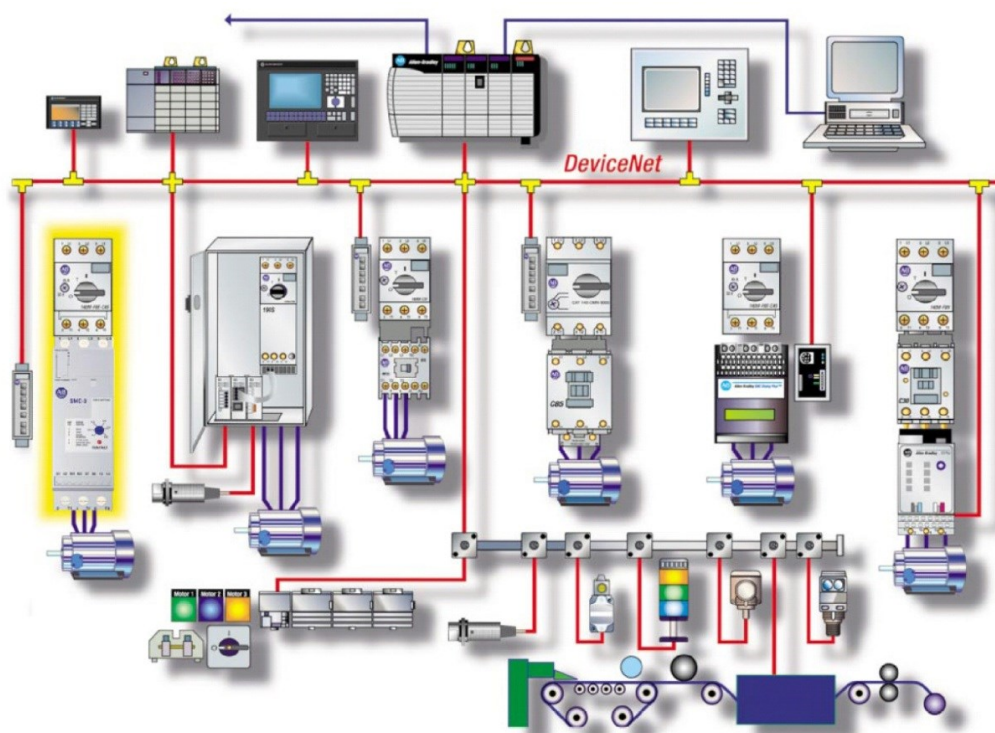
CANopen uvaja pojem vozlišče, s pomočjo identifikacije komunikacijskega objekta, od CAN-a pa sistem uporablja prepoznavne številke, ID-je. Za CANopen je značilen COB-ID, to je vsota identifikacijske številke okvirja za določen komunikacijski objekt in predstavitvene številke določenega vozlišča. Sama komunikacija poteka tako, da najprej definiramo identifikacijo posameznega objekta ali naprave, s tem omogočimo identifikacijo vozlišča in identifikacijo naprave [14].

Zgradbo opisuje standard ISO 11898. Pri protokolu CANopen so definirani objekti, s pomočjo katerih poteka komunikacija. Sami objekti pa omogočajo različne storitve, da lahko aplikacije, ki želijo med seboj komunicirati, le-to izvajajo. Aplikacijska plast ravno tako, kot pri samem CAN protokolu, skrbi za konfiguracijo, komunikacijo in sinhronizacijo v realnem času. Komunikacija poteka preko t. i. komunikacijskih objektov in ustreznih mehanizmov za sam prenos na in po vodilu. Mehanizme sestavlja slovar objektov, v tem slovarju so navedeni vsi podatki, ki vplivajo na obnašanje aplikacijskih in komunikacijskih objektov. Slovar objektov predstavlja vmesnik med aplikacijo in komunikacijo, sestavljen je iz več delov. Za naslavljanje različnih delov (rezervirani del, del statičnih podatkovnih tipov, del, namenjen kompleksnim podatkovnim tipom, sledi mu del specifičnih statičnih podatkovnih tipov v okvirju profila naprave, specifični kompleksni podatkovni tip v okviru profila naprave ima tudi svoj rezervirani del, del, ki je rezerviran za kasnejšo uporabo, del komunikacijskega profila, profil proizvajalca ima rezerviran svoj del, en del je predpostavljen za standardizirani profil naprave in na koncu sledi še del, ki je rezerviran za kasnejšo uporabo) slovarja se uporablja 16 bitni indeks in 8 bitni podindeks. Naslednji model, ki se tudi razlikuje oziroma nadgrajuje model, ki ga vsebuje sama CAN komunikacija, je komunikacijski model. Ker CANopen povezuje različne predmete in opravlja različne operacije, tudi ponuja različne načine povezovanja. CANopen omogoča nastavljanje inhibiranih časov, to so časi, ki določajo

najnižji časovni interval med dvema zaporedno poslanima sporočiloma. CANopen omogoča način komunikacije master/slave. Tak način vsebuje samo eno primarno vozlišče in ponuja funkcionalnost ostalim. Master upravlja ostala vozlišča. Samo master/slave vozlišče lahko naprej razdelimo na različne modele. Pri modelu tipa potisni (ang. Push) potrošniki zahteve za neko akcijo ne potrdijo. Pri modelu povleci (ang. Pull) so potrošniki tisti, ki sami zahtevajo določeno akcijo. Tretji model je model oblike odjemalec/strežnik (ang. Client/Server), kjer se vozlišči med seboj pogovarjata. Odjemalec pošlje zahtevo, na katero potrošnik odgovori [14].

### 3.2 DeviceNet

DeviceNet je objektni komunikacijski protokol, ki se najpogosteje uporablja na področju avtomatizacije industrije in za medsebojno povezovanje krmilnih naprav. DeviceNet uporablja kot hrbtenico CAN protokol, vpeljuje pa nov pojem aplikacijske plasti za pokrivanje večjega obsega različnih naprav. Naprave vsebujejo sklop za izmenjavo podatkov, sklop varnostnih naprav ter sklop večjega števila vhodnih ali izhodnih naprav. Sklop takih naprav prikazuje slika 3.2 [16].



Slika 3.2: Prikaz sklopa naprav, povezanih z DeviceNet vodilom [18].

DeviceNet je razvilo ameriško podjetje Allen-Bradley. Je višjenivojski protokol, ki za osnovo uporablja CAN komunikacijski protokol, hkrati pa vsebuje in je prilagojen na ControlNet

protokol, ki je ravno tako industrijski komunikacijski protokol. ControlNet je ravno tako razvila družba Allen-Bradley. Zaradi združljivosti s protokoli, grajenimi na platformi protokola RS 485, je cenovno ugoden ter robusten.

Za spodbujanje uporabe DeviceNet protokola je podjetje Rockwell Automation, v čigar lasti je danes družba Allen-Bradley, sklenilo, da bo DeviceNet odprt koncept protokola. Danes protokol upravlja neodvisna organizacija ODVA (ODVA je mednarodna organizacija, ki podpira računalniško povezljivost na osnovi CIP protokola. Vključuje pa DeviceNet, EtherNet/IP, CIP Sync in pa CIP Safety.). Združenje nadzira razvoj protokola, poleg tega zagotavlja skladnost s standardi DeviceNet in zagotavlja izvedbo testiranj. DeviceNet spada pod okrilje skupnih industrijskih protokolov, ki ga sestavljajo še: Ethernet/IP, ControlNet, CompoNet [18].

DeviceNet določa ISO/OSI, sedemplastni model. Podrobneje določa spodnje tri plasti (fizična, povezovalna ter omrežna plast) in zgornjo plast modela (aplikacijska plast). Vodilo je sestavljeno iz napajalnega in signalnega dela, po navadi je naslednje oblike: dva prepletena para in oklop. Hitrost prenosa je med 125 kbit/s in 500 kbit/s, dolžina prenosa pa je obratno sorazmerna s hitrostjo in znaša med 500 in 100 m za omenjene hitrosti. V protokolu je predpostavljen ploski šest žilni kabel, ki omogoča hitro namestljive konektorje. Na eno omrežje lahko priključimo do 64 naprav, ker ima predpostavljenih od 0 do 63 naslovov. DeviceNet omogoča komunikacijo tipa master/slave ali komunikacijo vsak z vsakim, večina naprav uporablja takšen tip komunikacije. Omogoča pa tudi več gospodarjev na enem vodilu (ang. Multimaster). Za manjše naprave, kot so razne fotocelice ali manjša stikala, pa je možno po mrežnem signalnem kablu pridobiti tudi napajanje. Naprave, priključene na omrežje, se lahko nadzirajo in nastavljajo preko komunikacijskih paketov. Protokol je zasnovan tako, da je odporen proti motnjam. DeviceNet po CAN-u povzema tudi bitno arbitražo in s tem zagotavlja prednost sporočilnim paketom z nujnimi informacijami. Drugače pa je DeviceNet združitev CIP protokola in CAN fizične plasti. Po CIP protokolu se vsaka naprava, priključena na vodilo, predstavlja kot skupek objektov. Vsak objekt je skupina sorodnih vrednosti v napravi. Poznamo tri glavne skupine sporočil, to so:

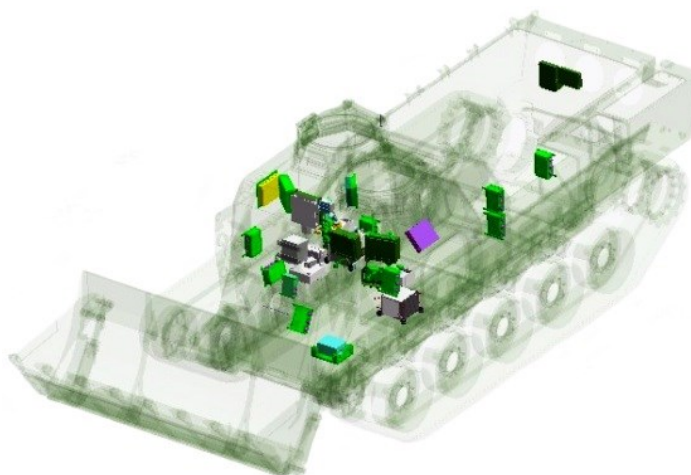
- zahtevani objekti, to so sporočila, ki vsebujejo predstavitevni objekt,
- sporočilni objekt in
- omrežni objekt.

Ponavadi je zahtevani objekt definiran v samih specifikacijah. Naslednja skupina so aplikacijski objekti. Ti določajo pomen podatkov v sporočilu. Na primer: motorni objekt v sistemu nekega vozila prenaša podatke, kot so frekvenca, trenutna hitrost ali vrsta motorja. Aplikacijski objekti so definirani za večje število podobnih naprav. Omogoča, da določeno sporočilo sprejme točno določena naprava ter hkrati omogoča predpostavljeni profil same naprave. Poznamo pa tudi objekte, ki ne spadajo v nobeno objektno skupino, to so posebni objekti ponudnikov. Predmeti, ki napravi niso na voljo za normalno delovanje, spadajo v skupino posebnih predmetov, to so posebne funkcije naprave. Do njih dostopamo na običajni način, kot je definirano v Cip protokolu [17].

### 3.3 MilCAN

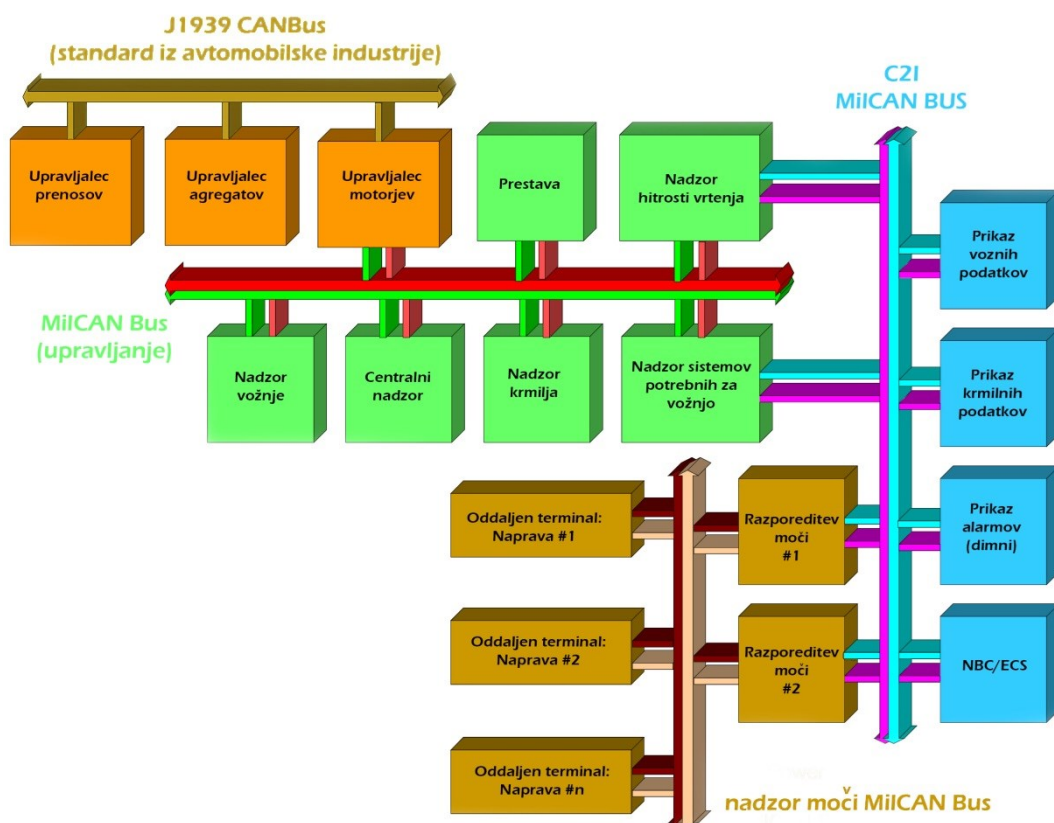
Zaradi pomembnosti opravljanja nalog komunikacijskega omrežja v najtežjih pogojih je bil razvit protokol MilCAN. MilCAN je nadgradnja na komunikacijskem protokolu CAN bus, hkrati pa je smiselna nadgradnja prej obstoječega Mil-Std 1553 bus protokola. Predvsem je stabilnejši, njegova izgradnja pa je cenejša. MilCAN deluje v realnem času, hkrati pa poskrbi, da so v zahtevanem času preneseni tudi komunikacijski paketi z najnižjo prioriteto.

V sistem povezujemo od najpreprostejših pa do kompliciranih naprav, ki opravljajo najrazličnejše naloge. Slika 3.3 prikazuje angleško vojaško inženirsko vozilo Terrier in shematično prikazuje enote, ki so med seboj povezane po MilCAN standardu. Nekatere izmed nalog lahko vplivajo na človeška življenja, zato so postavljene višje na lestvici prioritet, nekatere manj zahtevne, vendar ne nepomembne, pa lahko na izvedbo čakajo nekoliko več časa. Da bi samo delovanje celotnega sistema čim bolj uskladili, naprave razdelimo v segmente.



Slika 3.3: Angleško inženirsko vojaško vozilo Terrier [14].

Posamezni segment skrbi za nemoteno delovanje znotraj sebe, nato pa se povezuje z drugimi. Segmenti so med seboj ločeni po pomembnosti. Tako dobimo zelo širok spekter pomembnosti delovanja celotnega sistema. Slika 3.4 prikazuje principiarno zgradbo sistema vojaškega vozila Terrier. Po barvah vidimo, kako se naprave smiselno povezujejo v podsisteme znotraj celotnega sistema. Na sliki so vidne tudi dvojne zanke, ki so obvezne v sistemih v avtomobilski industriji, saj ne smemo dopustiti, da izgubimo nadzor nad vozilo, če en sistem odpove.



Slika 3.4: Principiarna zgradba celotnega sistema [14].

Razlikujemo dve različici MilCAN protokola. Prvi je MilCAN A, ki uporablja 29-bitov ID naslova, zgradba pa je podobna višje nivojskemu protokolu J1939. ID naslov MilCAN B protokola obsega 11 bitov in dovoljuje samo prenos periodičnih podatkov. Naprave MilCAN B lahko komunicirajo z napravami, ki go grajene za komuniciranje v CANopen omrežju. Razlikujeta se v hitrosti prenosa, MilCAN A podpira hitrosti med 250 kbit in 1 Mbit, medtem ko MilCAN B podpira hitrosti med 10 kbps in 1 Mbps.

Fizična, povezovalna in aplikacijska plast MilCAN komunikacijskega protokola se zgleduje po standardu ISO 11898-1. Vojaško združenje NATO stremi k temu, da bi bile naprave komunikacijskih protokolov J1939, CanOPEN in MilCAN, kar se da povezljive.

Fizična plast protokola, to je najnižja plast OSI/ISO modela, omogoča, da se po kablu omrežja napajajo tudi senzorji. S tem se deloma poenostavi tudi sama vezalna shema celotnega sistema, saj do odročnih senzorjev povlečemo samo en kabel in ne dveh. Hkrati pa za celoten sistem ne potrebujemo več manjših napajalnikov, ampak enega samega, ki skrbi, da se vse naprave napajajo v tolerančnem območju [15].

## 4. VIŠJENIVOJSKI PROTOKOL J1939

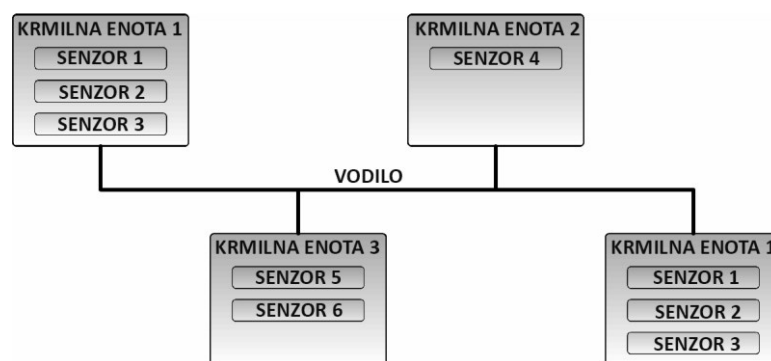
J1939 je višjenivojski protokol, ki za osnovo uporablja osnovni CAN komunikacijski protokol, verzije CAN 2.0B.

CAN bus je znan po svoji robustnosti in zanesljivosti, zmožen je delovanja v najzahtevnejšem okolju z zelo visokimi hitrostmi (do 1Mbit/s). Ker je J1939 nadgradnja le-tega, vse napisane trditve držijo tudi zanj. Razlikuje se le v nekaterih lastnostih. Največja hitrost prenosa podatkov je pri J1939 250 kbit/s, medtem ko je CAN bus zmožen komuniciranja pri hitrosti 1 Mbit/s. Poleg maksimalne zanesljivosti pa ima J1939 razvito odlično detekcijo napak in zmožnost preprečevanja izgube podatkov na vodilu (trki med podatkovnimi paketi). Vodilo je ponavadi prepleten par žic z opletom proti zunanjim motnjam. Dolžina vodila je lahko 40 m, nanj pa je lahko priklopljenih 30 naprav (ECU). J1939 podpira dolžine sporočil do 1785 bajtov. Prenos takega sporočila se izvede preko pošiljanja večjega števila zaporednih sporočil [9].

V nekaterih lastnostih protokola lahko pride do razhajanja med ISO standardom ISO 11898 in standardom, ki ga zagovarja združenje SAE, ki skrbi za nadzorovan razvoj komunikacije. Medtem ko SAE deklarira hitrost pretoka podatkov 250 kbit/s, po standardu ISO 11898 ni razloga, ki bi zmanjševalo najvišjo hitrost 1 Mbit/s. Tudi dolžina vodila se razlikuje, SAE zapoveduje maksimalno dolžino 40 m, medtem ko je v standardu zabeležena dolžina 250 m. Do razlike pride predvsem, ker SAE poizkuša obdržati promet po vodilu v nekih lahko nadzorovanih mejah, medtem ko ISO standard opisuje zgornjo mejo, ki se bliža že teoretičnemu stanju vodila.

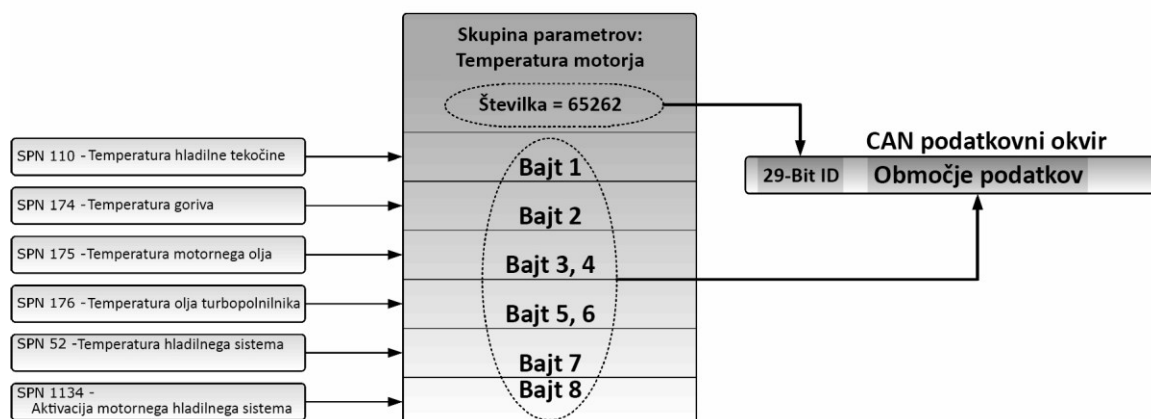
Protokol J1939 predpostavlja 8 bitni naslov naprave, kar nam odpira 256 različnih možnosti. SAE je omejen na 30 različnih naprav v omrežju. Iz slike 4.1 vidimo, da je lahko v omrežju več naprav z istim naslovom. J1939 omogoča določanje ID neznanim napravam. Podatke pošljemo v posebnem sporočilu, tako se naprava naslovi na želeni ali prvi prosti naslov. Glavna gradnika sporočila sta PGN – številka skupine parametrov in SPN – predvidena številka parametra. Sama povezava med SPN in PGN je vidna na sliki 4.2.





Slika 4.1: Primer omrežja J1939 z napravami, priključenimi nanj [2].

PGN in SPN sta določena s standardom. SAE J1939 je skupek standardov, ki nam določajo parametre, obsege parametrov in podatkovne tipe, ki jih pošiljamo ali prejemamo v posameznih sporočilih. PGN je pri komunikacijskem protokolu J1939 dolžine 29 bitov.

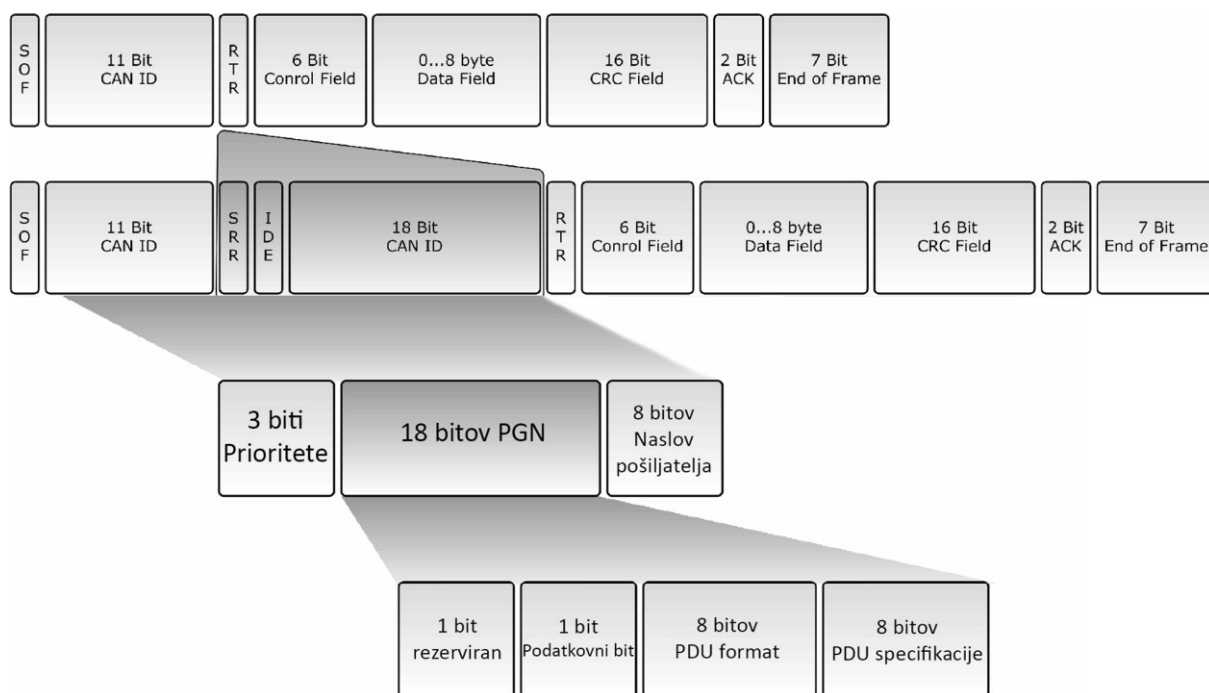


Slika 4.2: Povezava med SPN in PGN parametri [1].

Zgornji primer prikazuje povezavo PGN-ja in SPN-ja za sporočilo, ki ima podatek, kakšna je temperatura motorja. PGN in SPN sta logično povezana. Pod številko skupine parametrov 65262 imamo v 8 bajtnem podatkovnem delu različne SPN-je, vsi pa so v povezavi s temperaturo motorja. Prvi bajt nam pove temperaturo hladilne tekočine in ima SPN številko 110. Tretji in četrti bajt vsebujeta informacijo o temperaturi olja v motorju, SPN pa ima številko 175 [2].

## 4.1 Zgradba sporočila komunikacijskega protokola J1939

Bitni okvir posameznega podatkovnega paketa J1939 se začne s SOF, ki označuje začetek podatkovnega okvirja. Sledi mu 29 bitov, ki sestavljajo identifikacijo sporočila. Prvi trije biti ID naslova so prioriteta in so povezani s pomembnostjo informacije, sledita 2 bita, ki sta pomembna za samo obliko sporočila, ali sporočilo v nadaljevanju prenaša tudi podatke ali je le zahteva, ki povprašuje po podatkih. Zadnjih 16 bitov PGN naslova nam pove o podatkih, kaj se prenaša v podatkovnem polju. Zadnjih 8 bitov ID naslova pa je naslov pošiljatelja. ID naslovu sledi RTR bit (ang. Remote Request), ki je glavni pokazatelj, ali sporočilo zahteva informacijo ali pa jo prenaša. Nato sledi 6 kontrolnih bitov, ki nam povedo, koliko od maksimalno 8 bajtov podatkov vsebuje sporočilo, in ali je sporočilo z 11 ali 29 bitnim identifikacijskim naslovom. Temu sledi podatkovno polje v velikosti od 0 pa do 8 bajtov. 16 bitov CRC preverja pravilnost sporočila, če je prenesena informacija pravilna. Slika 4.3 prikazuje zgradbo komunikacijskega paketa.



Slika 4.3: Podrobna zgradba podatkovnega paketa [2].

ACK zaznamuje pravilnost sprejetosti sporočila, ali je neka naprava pravilno prebrala sporočilo iz vodila. Na koncu pa je še 7 bitov, ki označujejo konec okvirja [2].

J1939 je komunikacija, ki lahko poteka na dva načina, prvi je pogovor s točno določeno napravo (ang. Peer To Peer), drugi pa je naslavljanje vseh naprav (ang. Broadcast).

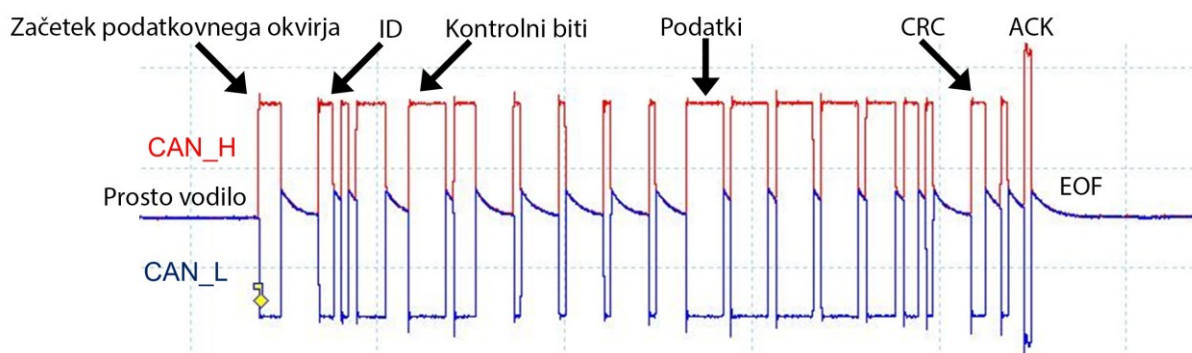
Informacije po vodilu potekajo tako, da jih določena naprava pošilja ciklično na določen časovni interval ves čas vklopa. Druga možnost pa je, da primarna naprava sprašuje ostale naprave, ki so priklopljene na vodilo in od njih zahteva informacije. Standard SAE J1939/71 je obsežna knjižnica, ki nam določa, katera številka parametrov (PGN) vsebuje določene predvidene številke parametrov (SPN). Standard SAE J1939/X je povezan s sedemplastnim ISO/OSI modelom. Tako je v standardu SAE J1939/1x opisana fizična plast vodila, v standardu SAE J1939/2x povezovalna plast, knjižnica standarda SAE J1939/3x pa opisuje omrežno plast. SAE J1939/4x je v povezavi s transportno plastjo ISO/OSI modela. SAE J1939/5x, 6x in 7x opisujejo plast seje, predstavitevni nivo in aplikacijski nivo. X na koncu je verzija standarda oziroma področje. Standard ni mrtev, ampak se z razvojem in s časom nadgrajuje, vse spremembe in nadgradnje pa so usklajene, pregledane in preverjene s strani združenja SAE. Pri iskanju gradiva sem naletel tudi na različne standarde s podobno oznako (J1939/12, J1939/14, J1939/72), ki pa jih nisem našel v uradni knjižnici SAE. To so interne knjižnice določenih proizvajalcev, ki jih uporabljajo za svoje namene in niso usklajene s strani združenja, zato niso namenjene širši uporabi [10].

CAN standard ne podpira direktnega naslavljanja. Prejemnik sporočila spozna sporočilo po ID naslovu, znotraj katerega pa se skriva 8 bitov, ki pripadajo pošiljatelju. Naprava ima na sprejemni strani možnost nastavljanja različnih mask, ki delujejo hkrati kot filter za sporočila. Ena izmed poglavitnih lastnosti, zakaj J1939 uporablja 29 bitno naslavljanje, je število različnih sporočil. 11 bitna identifikacija omogoča  $2^{11}$  različnih sporočil, 29 bitni ID pa omogoča  $2^{29}$  različnih sporočil.

Različne izvedbe CAN komunikacijskega protokola omogočajo povezanost med osnovnim in razširjenim identifikacijskim naslovom, tak primer je MilCAN komunikacijski protokol. J1939 pa te kompatibilnosti ne podpira, čeprav sama fizična zgradba omogoča tovrstno povezanost. V primeru povezanosti bi imela sporočila z 11 bitnim ID naslovom prednost v arbitražnem postopku. Sporočila z razširjenim ID naslovom imajo minimalno dolžino daljšo za časovni interval trajanja dvajsetih bitov.

Pošiljanje določenega sporočila poteka na naslednji način, in sicer najprej moramo zgraditi podatkovni paket. Začetek označuje bit SOF, ki na vodilu naznanja, da biti, ki mu sledijo, sestavljajo podatkovni paket. Podatkovni paket na vodilu je prikazan na sliki 4.4. SOF-u sledijo trije biti prioritete, kjer je vrednost 0 (000) najvišja prioriteta, vrednost 8 (111) pa je

sporočilo z najnižjo prioriteto. Sporočila oziroma podatki imajo določeno prioriteto s SAE standardom, tako da imajo sporočila, ki prenašajo podatke pomembne za varnost ljudi ali stroja višjo prioriteto od podatkov, ki vplivajo na udobnost vožnje. SAE J1939 je bil primarno mišljen za tovornjake oziroma težko mehanizacijo [20].



Slika 4.4: Slika podatkovnega paketa z 11 bitnim ID-jem na vodilu.

Arbitražnim bitom sledi številka skupine parametrov PGN. PGN je tabela, s pomočjo katere razvozlamo in dekodiramo podatke v sporočilu. Pove nam tudi, katera naprava pošilja in za katero napravo so podatki pomembni. Ko načrtujemo določeni izdelek, napravo ali senzor, ki mora komunicirati z ostalimi napravami, si pomagamo s standardom SAE J1939/71, ki je knjižnica vseh zbranih PGN-jev in izberemo le tiste, ki so za nas pomembni. Po možnosti interno dopolnimo kakšnega, ki ga ni v standardu, vendar je za nas pomemben. Primer usklajene in dopolnjene tabele je v tabeli 3, manjkajo ji določeni stolpci, ki niso bistveni za njeno razumevanje.

Tabela 3: Prikaz usklajene tabele PGN-jev.

PGN (h) PDUf (d) PDUs (d)	Name	Repet. Rate [ms]	Byte	Byte Function	SPN	Format	Comment	Unit
<b>00F00C</b> 254 12	Motor Power	100	1, 2	Motor_Power		Int	Motor DC-link average power power smoothed by low path filter with 0.5s time constant	W
			3, 4	Motor_Current		Int	Motor DC-link current	A
			5, 6	Motor_PhaseCurrent		Int	Motor phase current amplitude	A
			7, 8	Mot_MessageCounter		Int		
<b>00FEDA</b> (DODAN) 254 218	Battery Software Identification	on request	1	BMS_SW_IdentFieldsNumber		Int	Number of software identification fields	
			2 - n	BMS_SW_Identification	234	ASCII	Software identification number	
<b>00FEEB</b> 254 235	Component Identification	on request			586	ASCII	Field a: Make	
					587	ASCII	Field b: Model	
					588	ASCII	Field c: Serial number	
					233	ASCII	Field d: Unit number (Power unit) Not necessary to include all fields, however, the delimiter is allways required	

Znotraj tabele lahko vidimo PGN 00FEDA, ki ga ne najdemo v standardu, ampak je usklajen z vsemi proizvajalci naprav in senzorjev za omenjeni projekt. V prvem stolpcu tabele najdemo PGN številko, ki je razdeljena na PDUf in PDUS, ki ju potrebujemo pri sestavljanju določenega sporočila. Drugi stolpec tabele nam pove ime oziroma funkcijo, na katero se nanašajo podatki. Tretji stolpec predstavlja, na kakšen časovni interval se določeno sporočilo pošilja. Obstaja tudi možnost, da se sporočilo ne pošilja periodično, temveč na zahtevo. Četrti stolpec razdeli podatkovni del sporočila in nam pomaga razumeti, kateri podatki se skrivajo v katerem od 8 bajtov in koliko le-teh obsega. Peti stolpec nam pove ime funkcije določenega bajta, šesti je SPN številka, sedmi pa je tip oziroma format podatka. Osmi stolpec nam poda podrobno razdelitev določenega bajta po bitih. Deveti stolpec je komentar, ki nam v večini primerov razloži, kaj določena vrednost osmega stolpca pomeni. Zadnji štirje stolpci pa predstavljajo obseg podatkov, najmanjši korak, območje podatkov in enote. V številki skupine parametrov sledi naslov pošiljatelja, ki obsega 8 bitov. 29 bitni identifikacijski naslov omogoča 253 različnih naslovov. Naslovi naprav v omrežju se morajo med seboj razlikovati, saj lahko le tako sledimo, katera naprava je poslala določeni podatek. V primeru, da bi obe zavori imeli isti naslov, bi lahko prišlo do tega, da bi ena zavora ves čas zavirala, druga pa bi ostala nesprožena, in ker bi obe imeli isti naslov, glavni računalnik vozila ne bi zaznal napake.

## 4.2 Načini komunikacije

Standard SAE J1939 podpira tri komunikacijske tipe, vsak pa je zadolžen, da pokriva svoje področje.

- Prvi je ciljno usmerjena komunikacija (ang. Destination Specific Communication).
- Drugi način je komunikacija širšega področja (ang. Broadcast Communication).
- Tretji način pa je namenska komunikacija (ang. Proprietary Communication).

Ciljno usmerjena komunikacija uporablja PDU1 format, to pomeni vrednosti od 0 do 239, hkrati pa se lahko uporablja tudi globalno naslavljanje z naslovom 255. Obstajajo primeri, kjer ta metoda zahteva uporabo formata PDU specifikacije, to se lahko zgodi, če imamo v omrežju dva ali več enakovrednih motorjev, neko zahtevo pa moramo poslati samo enemu izmed njih [2].

Komunikacija širšega področja uporablja naslove PDU2, to so vrednosti med 240 in 255. Že samo ime pove, da je ta komunikacijski tip namenjen komuniciranju in pošiljanju sporočil enega ali večjih oddajnikov enemu sprejemniku oziroma pošiljanje enega oddajnika enemu ali večjemu številu sprejemnikov [2].

Tretja je namenska komunikacija. Le-ta uporablja naslove PDU1 in PDU2. Potrebujemo jo le izjemoma, in sicer takrat, ko standardna ciljno usmerjena komunikacija ali komunikacija širšega področja ne ustreza zahtevam. Ker uporablja obe skupini naslovov, jo lahko delimo na dve podskupini, to sta namenska komunikacija širšega področja in ciljno usmerjena namenska komunikacija, odvisno od skupine naslovov, ki jih uporabimo [2].

Natančneje sta sprejem in nadaljnja obdelava sporočil opisana v standardu SAE J1939/21 (povezovalna plast) in J1939/7X (aplikacijska plast) [10].

Sprejeto sporočilo je torej obdelano glede na eno od komunikacijskih metod:

- Specifična zahteva (ang. Destination Specific Request Or Command)

Vsaka naprava (ECU) mora preveriti, ali naslov, ki je naslovljen v sporočilu, ustreza lastnemu naslovu naprave, nato pa mora le-ta sporočilo sprejeti in ustrezno odreagirati ter poslati potrdilo.

- Splošna zahteva (ang. Global Request)

Če je poslano sporočilo s splošno zahtevo, morajo vse naprave na vodilu sprejeti in obdelati sporočilo, to pomeni, da naprava preveri, ali so podatki, potrebni za oddajo, že pripravljeni.

- Splošna oddaja (ang. Broadcast)

Če se oddaja sporočilo splošne oddaje, morajo vse naprave poslušati in preveriti, ali je sporočilo namenjeno tudi njim, če jim je, se morajo odzvati tako, kot je v sporočilu zahtevano ali zabeleženo. Če pa sporočilo ni namenjeno njim, pa lahko poslušanje prekinejo [10].

### **4.3 Številka skupine parametrov (PGN)**

SAE J1939 je domiselno zgrajen protokol, ki iznajdljivo uporablja 29 bitni ID naslov. Višjenivojski protokol J1939 raje, kot bi uporabljal veliko število funkcij, uporablja tabelo parametrov, ki je bistvo protokola. Torej, če hočemo ustvariti določen sistem, ki bo splošno povezljiv v komunikacijskem protokolu CAN bus, moramo biti pazljivi, da so komunikacije, s katerimi bosta naprava ali modul operirala, skladna s tabelo parametrov [2].

Skupina posameznega parametra tvori smiselno celoto, kot jo lahko vidimo v tabeli 3. Kot primer lahko izpostavimo številko skupine parametra (PGN) 00F00B, kar je skupina parametrov za status motorja. V podatkovnem okvirju statusa motorjev se prenašajo podatki o statusnih bitih motorja (napake motorja, opozorila, smer vrtenja, kontrola hitrosti, kontrola navora, zgornja omejitev hitrosti in navora), navoru motorja, hitrosti motorja, napetosti motorja in vrednost števca sporočil. Podrobneje prikazuje zgradbo tabela 4.

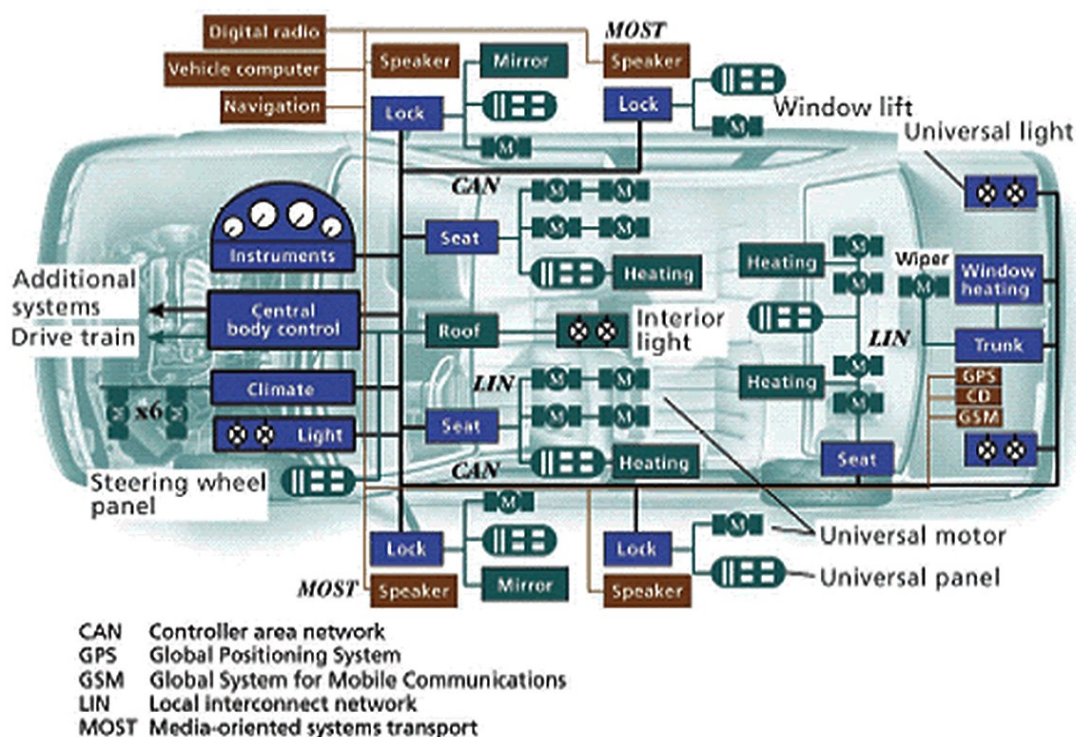
Tabela 4: Prikaz razčlenitve ene skupine parametrov po posameznih bitih in bajtih v sporočilnem delu enega komunikacijskega okvirja.

PGN (h) PDUf (d) PDU <sub>s</sub> (d)	Name	Repet. Rate [ms]	Byte	Byte Function	SPN	Format	Bit Function	Comment	Range	Scaling	Offset	Unit
00F00B 240 11	Motor Status	20	1	Mot_StatusBits		bit	1 Failure	See parameter indices				
						bit	2 Warning	See parameter indices				
						bit	3 Rotation direction	0 CW (clock wise)				
								1 CCW (counter clock wise)				
						bit	4	Spare				
						bit	5 Inactive					
						bit	6 Speed control					
						bit	7 Torque control					
						bit	8 Speed/Torque limited control					
			2	Mot_Torque	513	Int		Motor torque, calculated based on motor model	-125 to +125	1	-125	%
			3, 4	Mot_Speed	190	Int		Motor speed	0 to +8191.875	0.125	0	1/min
			5, 6	Mot_Voltage	158	Int		DC-link voltage of propulsion system	0 to +3212.75	0.05	0	V
			7, 8	Mot_ResponseMessageCounter		Int		Response @ Mot_Message_Counter				

Seznam skupine parametrov je izdan znotraj dokumenta SAE J1939/21, skupaj obsega blizu 300 strani dolgo tabelo, kjer so zbrani vsi uradno potrjeni parametri. SAE J1939/71 pa je dokument, kjer so podrobno opredeljeni vsi posamezni parametri in obsega okoli 800 strani. Tabeli 3 in 4 sta samo kratka izvlečka. Tabela skupine parametrov nam pove oziroma splošno določa zgradbo in pomen posameznih bitov znotraj 8 bajtov podatkov, ki jih posamezno sporočilo prenaša. Če hočemo, da komunikacija poteka brez težav, tekoče in z najvišjo možno hitrostjo, potem je treba, da so vse naprave v omrežju definirane po istem ključu. Vsaka naprava ima v spodnjih plasteh nastavljene maske, ki prepuščajo samo sporočila, ki so pomembna za posamezno napravo ali senzor. Mikrokrmilnik obdela sporočilo v najkrajšem



možnem času in pripravi ter odpošlje pogovor, če je bila taka zahteva, oziroma čaka na zahteve uporabnika, ki jih preko nekega komandnega komunikacijskega vmesnika vnaša in pričakuje odziv. Treba se je zavedati, da so danes sistemi, kamor se vgrajujejo moderni komunikacijski protokoli (med katere spada tudi CAN bus z višjenivojskim protokolom J1939) zelo zapleteni in prepleteni z različnimi komunikacijskimi protokoli. Slika 4.5 prikazuje podrobno zgradbo vozila in kako se različni komunikacijski protokoli prepletajo znotraj celote.



Slika 4.5: Prikaz prepletanja različnih komunikacijskih protokolov znotraj posamezne celote avtomobila [25].

Skupina parametrov pa poleg podatkovnega dela vsebuje tudi informacije o pogostosti generiranja posameznega sporočila in sami prioriteti sporočila. Tabela 4 prikazuje realno razčlenbo komunikacijskega paketa. Tako razčlenjene parametre najdemo v dokumentu SAE J1939/71. Slika nam definira celotno sporočilo, kjer je v prvem stolpcu zapisana številka skupine parametrov, ki je za naš primer 00F00B, spodaj pa sta zapisana še PDUf (240) in PDUs (11). Drugi stolpec vsebuje ime skupine. Tretji stolpec nam pove, na koliko časa se sporočilo generira. Sporočilo iz tabele 4 se generira na 20 [ms]. Četrty stolpec nam izda, na katero funkcijo se posamezni bajt nanaša. V stolpcu šest so zapisane pričakovane številke parametra (ang. Suspect Parameter Number – SPN). V stolpcu osem je razčlenba po posameznih bitih. Stolpec devet je razlaga in komentar stolpca osem. V stolpcih deset, enajst, dvanajst in trinajst pa so podatki oziroma definicije vrednosti, ki jih posamezni bit ali bajt

prenaša. V stolpcu deset je obseg določenih podatkov, stolpec enajst definira najmanjši korak, stolpec dvanajst morebitni odmik iz izhodišča, v stolpcu trinajst so zapisane enote posameznih podatkov v vrstici.

PGN 61451	Status motorja
Hitrost prenosa:	20 ms
Število podatkovnih bajtov:	8 bajtov
Vrednost bita DP:	0
Vrednost PDUf:	240
Vrednost PDUs:	11
Prioriteta:	6
Številka skupine parametrov:	00F00B
Opis vrednosti podatkovnih bajtov:	
Bajt: 1: Mot_StatusBits → Status bitov v motorju:	
	Bit 1: napake
	Bit 2: opozorila
	Bit 3: smer vrtenja
	Bit 4: rezerva
	Bit 5: neaktiven
	Bit 6: kontrola hitrosti
	Bit 7: kontrola navora
	Bit 8: kontrola omejevanja hitrosti in navora
2: Mot_Torque → navor motorja	
3, 4: Mot_Speed → hitrost motorja	
5, 6: Mot_Voltage → napetost motorja	
7, 8: Mot_ResponseMessageCounter → števec odzivov motorja	

Pričakovana številka parametra je vrednost, predpisana s strani družbe SAE in je vrednost posameznega parametra v tabeli. Tabela je zelo obsežna. Posamezne parametre iščemo po SPN vrednostih. Del SNP tabele prikazuje tabela 5. V tabeli pod določeno pričakovano številko parametra najdemo vse pomembne podatke za obdelavo in generiranje sporočila v razvojni fazi. V prvem stolpcu parametrov je zapisana pričakovana številka parametra. Drugi stolpec je naziv parametra, tretji stolpec je njegov komentar ali razlaga. V četrtem stolpcu je zapisana informacija o dolžini, koliko podatkovnih bajtov je potrebnih za zapis tega

parametra. V stolpcih pet, šest, sedem in osem pa so podatki o obsegu, najmanjšem koraku, odmiku iz izhodišča in enote. SPN-ji, ki opisujejo karakteristike podobnih parametrov, so združeni v posamezno skupino parametrov. Po vodilu bodo poslani v poročilu s skupno številko parametra.

Poglejmo, iz katerih SPN-jev in katerih parametrov je sestavljeno sporočilo s PGN 61451. Prvi bajt v sporočilu nima definiranega SPN-ja, ker se pošiljajo statusi posameznih bitov, ki se navezujejo na različne status motorja. V drugem bajtu se prenaša SPN s številko 513. To je dejanski parameter, katerega navor je izražen v odstotkih. Vrednost je izračunana glede na motorske parametre. Podatek obsega 1 bajt, vrednost pa se giblje med -125 in +125 odstotki. V bajtih 3 in 4 PGN 61451 se prenaša SPN 190. To je parameter hitrosti motorja. Hitrost je izračunana vrednost, ki se giblje med 0 in  $+8191,875 \text{ min}^{-1}$ , in sicer s korakom po 0,125. Predzadnja dva bita (bajt 5 in bajt 6) PGN 61451 prenašata parametre o napetosti baterij. Napetost je izmerjena na vhodu razsmernika, podatki obsegajo 2 bajta, vrednost se nahaja med 0 in 3212,75 V. Najmanjši korak je 1 V. Zadnja dva bajta PGN 61451 pa ne vsebujeta nobenega SPN-ja, saj je njuna funkcija štetje odzivov motorja. Samo povezavo med SPN-ji in PGN-ji prikazuje slika 4.2. Katere pričakovane številke parametrov tvorijo določeno skupino parametrov, je izbrano po logičnem sklepanju, kateri parametri smiselno spadajo skupaj. Posamezno sporočilo mora imeti neko smiselno celoto, hkrati pa mora biti sama arhitektura protokola prijazna do vgradnje v sisteme.

Tabela 5: Prikaz usklajene tabele SPN-jev.

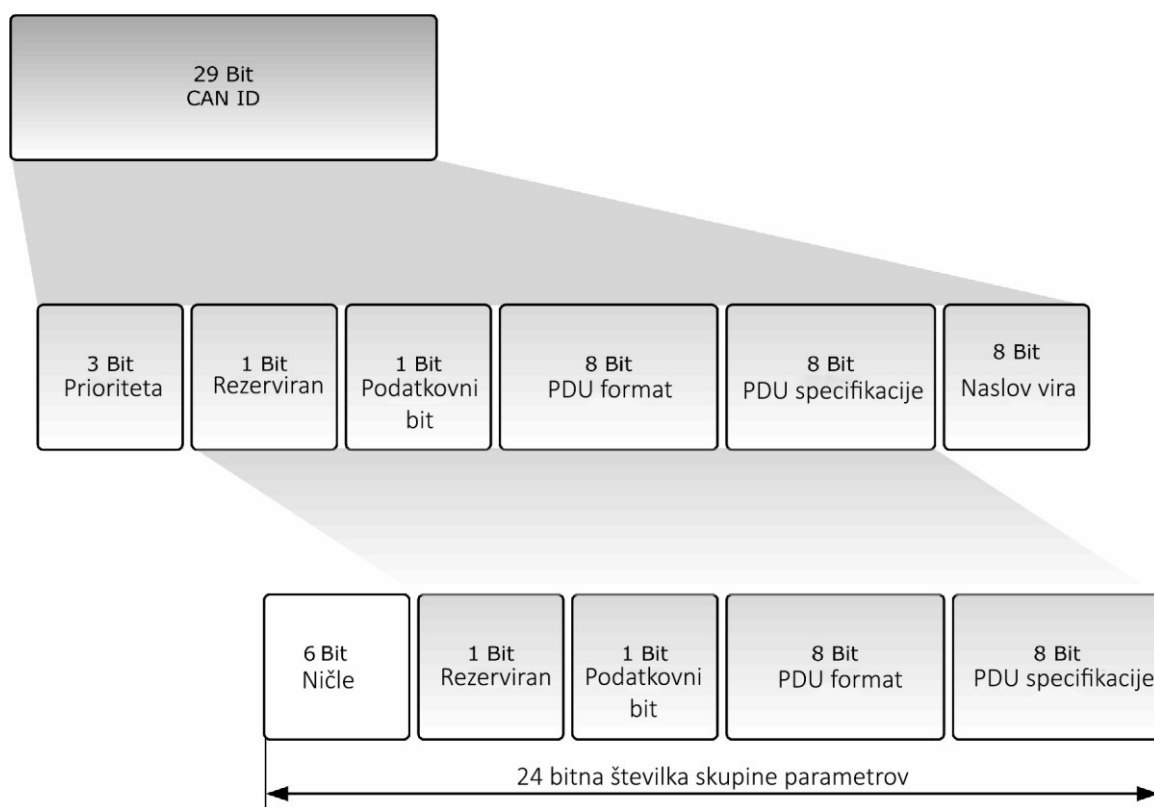
SPN	Parameter	Comment	Data Length [byte]	Range	Scaling [1 / digit]	Offset	Unit
110	Engine Coolant Temperature	Temperature of liquid found in cooling system	1	-40 to +210	1	-40	°C
158	Battery potential (voltage) switched	Electrical potential measured at the input of the electronic control unit supplied through a switching device	2	0 to +3212.75	0.05	0	V
168	Electrical potential (voltage) of the battery	Measured electrical potential of the battery	2	0 to +3212.75	0.05	0	V
189	Rated engine speed	Maximum governed rotational velocity of the engine crankshaft	2	0 to +8191.875	0.125	0	1/min
190	Engine speed	Actual calculated engine speed	2	0 to +8191.875	0.125	0	1/min

Zgradba številke skupine parametrov dopušča do 8672 različnih skupin. Številka skupine parametrov določa oziroma sporoča, katera skupina parametrov se pošilja v posameznem sporočilu. Vsaka skupina parametrov definira, kateri parametri se prenašajo in kaj se z njimi zgodi. S sporočilom lahko spreminjamo samo vrednost parametra, z njim lahko kličemo vrednosti, kateri parametri so trenutno shranjeni v določeni napravi ali senzorju. S PGN pa je določena tudi hitrost in smer komunikacije in prioriteta. Ko določeni ECU sprejme sporočilo, uporabi PGN v identifikacijski številki, da razbere tip podatkov v podatkovnem delu sporočila. Slika 4.6 prikazuje zgradbo številke skupine parametrov [2].

Bit R, ki je rezerviran, je po dokumentu SAE J1939 namenjen kasnejši obdelavi in mora biti v trenutku pošiljanja sporočila vedno nastavljen na vrednost 0. V dokumentu SAE J1939/21 je ta bit označen kot razširjeni podatkovni okvir (ang. Extended Data Page (EPD)). Ta bit nakazuje, katera sporočila so poslana kot funkcija razširjenega podatkovnega okvirja. Več o pošiljanju informacije v več komunikacijskih paketih bom opisal kasneje [2].

Podatkovni okvir (ang. Data Page (DP)) deluje kot pokazatelj sledečemu formatu enot podatkovnega protokola. Vrednost bita je 0 in kaže na stran 0, ki kaže na vsa trenutno definirana sporočila. Stran 1 bo v uporabi, ko bomo želeli pošiljati razširjene podatkovne okvirje [2].

Format enot podatkovnega protokola (ang. Protocol Data Unit format (PDUf)) definira funkcijo PDUf. Dolžina je 8 bitov, razdeljeni pa so na dva dela zaradi konstrukcije 29 bitnega ID-ja. 29 bitna identifikacijska številka je zgrajena iz 11 bitov osnovne identifikacijske številke, 1 bita SRR, 1 bita IDE in 18 bitov razširjene identifikacijske številke. SRR in IDE sta določena s strani standarda in sta v nalogi že opisana. Zaradi omenjenih bitov pa je v literaturi včasih PF označen kot 10 bitna številka.



Slika 4.6: Zgradba 24 bitne številke skupine parametrov [2].

Specifikacija enot podatkovnega protokola (ang. Protocol Data Unit specific (PDUs)) nam pomaga podatkovni del razumeti v pravilni obliki. Vrednosti formata enot podatkovnega protokola med 0 in 239 (PDU1) kažejo ciljni naslov v specifikaciji enot podatkovnega protokola ob komunikaciji vsak z vsakim. Vrednosti PDUf med 240 in 255 (PDU2) ob komunikaciji širšega področja nakazujejo razširitev skupine (ang. Group Extension (GE))

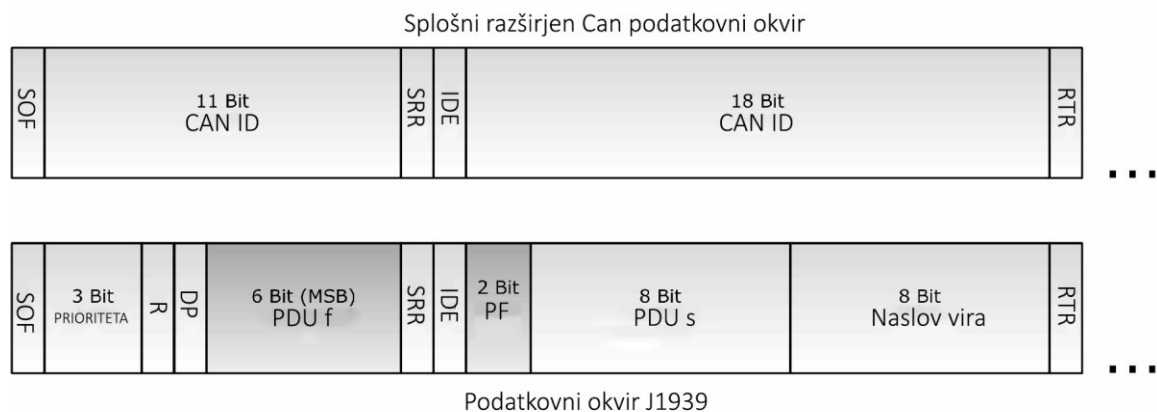
PDUf. Razširitev skupine je uporabljena za povečanje števila prostih sporočil v tipu komunikacije širšega področja. Tabela 6 prikazuje povezavo med PDUf in PDUs.

Tabela 6: Povezava med PDUf in PDUs [2].

	PDUf	PDUs	Tip komunikacije
PDU 1 format	0 - 239 0(hex) - EF(hex)	ciljni naslov	vsak z vsakim
PDU 2 format	240 - 255 F0(hex) - FF(hex)	razširitev skupine	komunikacija za širše področje

Medtem, ko s ciljnim naslovom naslavljamo točno določeno napravo ali senzor v omrežju, pa lahko ta isti tip komunikacije uporabimo ob želji naslavljanja več naprav hkrati. Naslov 255 je tako imenovani DGA (ang. Global Destination Address), ki je globalni naslov in z njegovo uporabo naslovimo vse naprave v omrežju. Po standardu SAE J1939 se najpogosteje uporablja komunikacija za širše področje in PDU 2 format (razširitev skupine). Ta sporočila ne morejo biti namenjena le točno določeni napravi [2].

Slika 4.7 prikazuje preslikavo podatkovnega okvirja med podatkovnim okvirjem splošnega razširjenega CAN okvirja v podatkovni okvir J1939.



Slika 4.7: Prikaz preslikave med splošnim razširjenim CAN podatkovnim okvirjem in podatkovnim okvirjem CAN višjenivojskega protokola J1939 [2].

Kot je razvidno iz tabele 6, PDUf in PDUs podpirata 8672 različnih skupin. Obseg številke skupine parametrov je razdeljen v dve skupini:

1. Posebne številke skupine parametrov za komunikacijo vsak z vsakim (PDU 1 format)  
Obseg: 00<sub>hex</sub> - EF<sub>hex</sub>, (PDUs ni vključen)  
Število PGN-jev: 240

2. Generične številke skupine parametrov za razširjeni način komunikacije (PDU 2 format)

Obseg: F000<sub>hex</sub> - FFFF<sub>hex</sub>, (PDUs ni vključen)

Število PGN-jev: 4096

Upošteva bit podatkovnega okvirja, je skupno število skupine parametrov 8672. Podroben izračun prikazuje enačba 4.3 [2].

$$(240+4096)*2=8672 \quad (4.3)$$

Bit podatkovnega okvirja je pokazatelj formata enot podatkovnega protokola. Ob pošiljanju sporočila mora biti ta bit postavljen na 0, kar kaže na vsa definirana sporočila. Stran 1 je v uporabi samo ob uporabi razširjenega načina, ko so izrabljene vse možnosti vrednosti bita DP 0. Tabela 7 prikazuje vseh 8672 skupin [2].

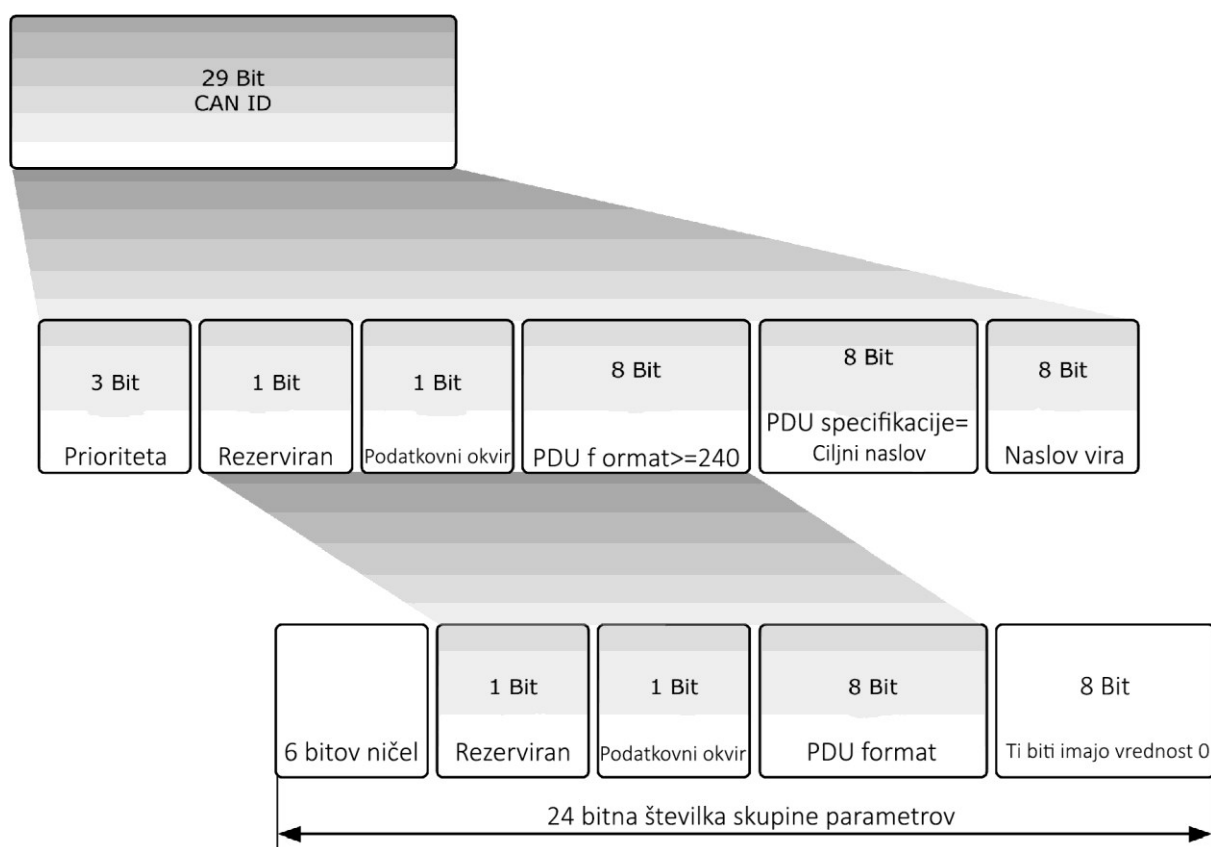
Trenutni obseg števil skupin parametrov je po standardu SAE J1939/71 od najnižje vrednosti PGN 0, ki opisuje parametre navora in hitrosti, pa do PGN 65279, ki opisuje informacije o stanju goriva in vode.

Za nekatere interne namene je številka skupine parametrov razširjena na 24 bitov, kjer je 6 najpomembnejših bitov vedno postavljenih na 0. Ta proces mora biti izvršen posamezno, saj ni del standardiziranega CAN procesa. Da prevedemo in povežemo prave parametre s pravo skupino parametrov ter da naslovimo prave naprave v omrežju, moramo razumeti oba tipa pošiljanja. PDU1, ki je komunikacija vsak z vsakim, in pa PDU 2, ki je komunikacija širšega tipa oddajanja sporočila, saj sporočilo oddamo na vodilo z namenom, da mu prisluhnejo vse naprave, ki so v določenem trenutku priključene na omrežje [2].

Komunikacija PDU 1, vsak z vsakim, ima posebno zgradbo, ki jo prikazuje slika 4.8. Vrednost formata enot podatkovnega protokola je manjša od 240. Pri tem tipu komunikacije je 8 najmanj pomembnih bitov postavljenih na vrednost 0. Specifikacijo enot podatkovnega protokola pa razumemo kot naslov naslovnika. Ko z naslovom naslovnika nakazujemo na točno določeno napravo v omrežju, pa lahko s točno določeno vrednostjo naslova naslovnika naslavljamo vse naprave v omrežju. Ta vrednost je 255 in jo imenujemo globalni ciljni naslov [2].

Tabela 7: Obseg števil skupin parametrov [2].

DP	Obseg PGN [hex]	Število PGN-jev	SAE ali Združenje proizvajalcev	Tip komunikacije
0	000000 - 00EE00	239	SAE	PDU 1=Vsak z vsakim
0	00EF00	1	ZP	PDU 1=Vsak z vsakim
0	00F000 - 00FEFF	3840	SAE	PDU 2= Komunikacija širšega področja
0	00FF00 - 00FFFF	256	ZP	PDU 2= Komunikacija širšega področja
1	010000 - 01EE00	239	SAE	PDU 1=Vsak z vsakim
1	01EF00	1	ZP	PDU 1=Vsak z vsakim
1	01F000 - 01FEFF	3840	SAE	PDU 2= Komunikacija širšega področja
1	01FF00 - 01FFFF	256	ZP	PDU 2= Komunikacija širšega področja

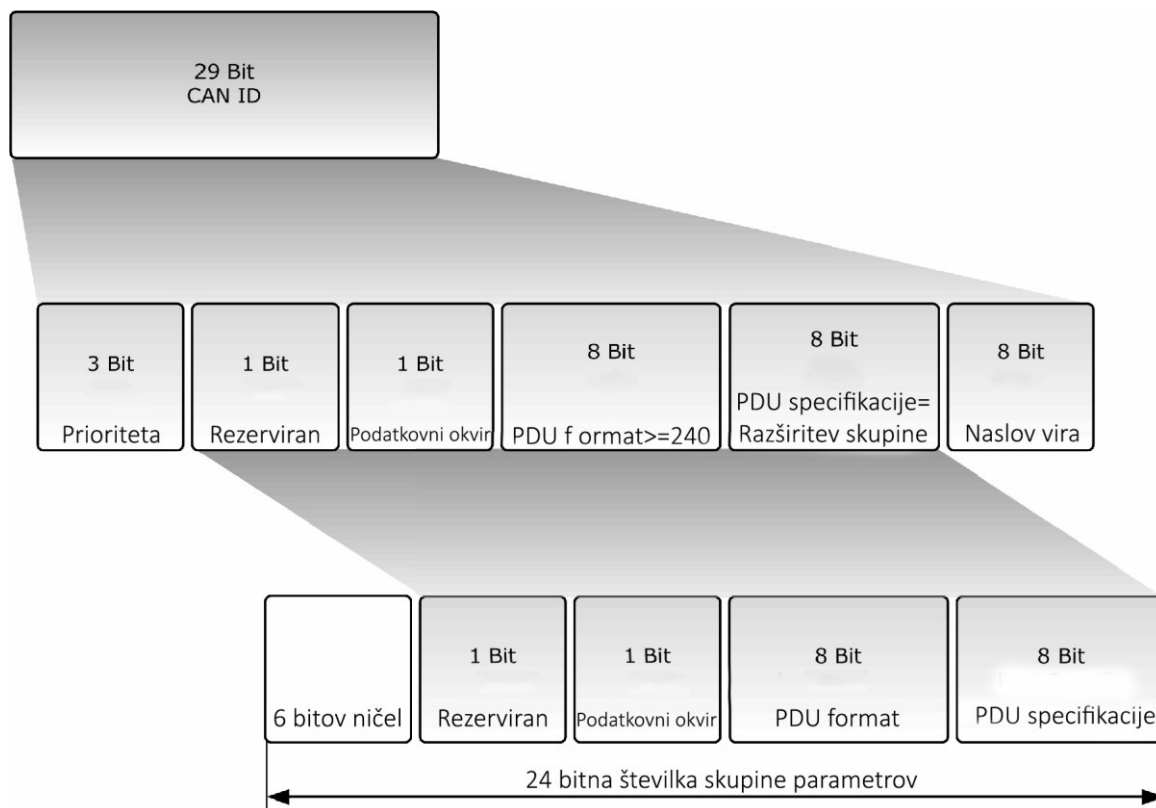


Slika 4.8: PDU 1 format številke skupine parametrov [2].

Slika 4.9 prikazuje zgradbo sporočila v PDU 2 formatu, ki je sporočilo, namenjeno pošiljanju širšega tipa, lahko bi rekli splošno oddajanje sporočila v omrežje. Vrednost formata enot



podatkovnega protokola je enaka ali večja od 240. Vrednost specifikacije enot podatkovnega protokola razumemo kot vrednost razširjene skupine [2].



Slika 4.9: PDU 2 format številke skupine parametrov [2].

Standard SAE J1939 vključuje številke skupine parametrov, ki so proste in so na voljo za interne parametre naprav. Te številke skupine parametrov imenujemo registrirane številke skupine parametrov, saj so v standardu registrirane in proste ter so namenjene parametrom, ki jih standard ne obsega. Registrirane številke skupine parametrov obsegajo PDU 1 in PDU 2. Edino navodilo s strani SAE je, da v omrežju naj ne bosta več kot 2 odstotka takih sporočil, saj je v omrežje, kjer je večje število naprav s posebnimi parametri, težje vključiti standardizirane naprave. Treba je ročno nadgraditi vse knjižnice naprav, da razumejo omenjena sporočila. V nekateri literaturi zasledimo, da so to predvidene številke parametrov med 520192 in 524287. Registrirana števila skupine parametrov imajo povsem enako strukturo sporočila kot ostala sporočila, kar prikazuje primer [2].

Primer sporočila z registrirano številko skupine parametrov:

Ime skupine parametrov:	Registrirano sporočilo A
Številka skupine parametrov:	61184 (00EF00 <sub>hex</sub> )
Definicija:	Registrirano število skupine parametrov uporablja format PDU 1 in tip komunikacije vsak z vsakim
Hitrost prenosa:	Določi proizvajalec
Dolžina podatkovnega dela:	Od 0 pa do 1785 bajtov, podpira tudi pošiljanje enega sporočila, deljenega v več paketih
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	0
PDUf in PDU s:	239 in 8 bitni naslov naslovnika, ki ga določi proizvajalec
Prioriteta in opis podatkovnega dela:	6, določi proizvajalec

Težava, ki se lahko pojavi v sklopu registrirane številke skupine parametrov je, da različni proizvajalci uporabijo iste številke za različne parametre. Ko napravo proizvajalca A priključimo na omrežje proizvajalca B, naprava ne bo javila nobene napake ob prebiranju, vendar pa bo javljala napačne podatke oziroma napaka bo v delovanju, saj bo interpretirala prejete zahteve ali parametre na napačen način. Da se izognemo morebitnim nepravilnostim, se moramo v času konstrukcije in načrtovanja zavedati in uporabljati standardizirane parametre. Za primer vzemimo plovilo. Podjetje izdelava pogonski elektromotor, komunikacijsko vodilo, panel. Skratka, zaključeno celoto. Kasneje pa lahko uporabnik kupi na trgu določene naprave in jih priključi na omrežje, kar pa lahko povzroči, da zaključena celota ne deluje več brez težav, saj se lahko začne na vodilu pojavljati »vsiljivec«, ki kvira robustnost omrežja, vse to zaradi nespametne uporabe registriranih številke skupin parametrov. Številke skupine parametrov uporabljajo dva formata. Izbiramo med PDU 1 in PDU 2 formatom. Zavedati pa se moramo, da ko neko sporočilo oblikujemo in mu določimo format, tega med delovanjem ne moremo spremeniti. Z arhitekturo sporočila določimo tudi prioriteto, hitrost posodabljanja, pomembnost podatkov do drugih naprav v omrežju in dolžino podatkovnega dela sporočila. Čeprav CAN standard s svojimi pravili in podstandardi poskuša pokriti zahteve, ki so na trgu, se ob hitro razvijajočih se napravah dnevno pojavljajo situacije, kjer sam standard nima odgovora na potrebo. Takrat je potrebno poseči po izjemah. Za primer uporabimo že opisanega, in sicer ko imamo omrežje enega proizvajalca z nekaterimi

napravami in priključimo večje število naprav drugega proizvajalca, ki pa med seboj komunicirajo predvsem z registriranimi števili skupine parametrov. V tem primeru nam standard priporoča uporabo registriranih skupin parametrov. Ravno tako lahko izbiramo med dvema formatoma, in sicer med PDU 1, PDU 2 in registriranim formatom. PDU 1 uporabimo za informacije splošnega tipa, torej sporočila, ki bodo znana vsem napravam v omrežju. Zahteva je, da uporabimo točno naslavljanje naslovnikov. PDU 2 uporabimo, ko želimo splošno informacijo prenesti po omrežju, vendar tukaj ne sme biti zahteva za točno naslavljanje naslovnika. Registrirani format pa uporabljamo samo za prenašanje sporočil z registrirano številko skupine parametrov. V takem omrežju se priporoča kot primarni tip formata PDU 2 format, ostali pa samo v primeru, ko nam PDU 2 ne odgovarja na potrebe.

Kot sem že omenil je hitrost višjenivojske J1939 CAN bus komunikacije 250 kbit/s, kar pa ni nikakršna ovira ali problem za mikrokrmilnike, ki so danes na trgu. Hitrost je omejena zgolj in samo, da se na vodilu ustvarja promet, ki je dovolj tekoč, da tudi najšibkejši člen omrežja zmore sprejemati in oddajati sporočila, pomembna za nemoteno in kakovostno delovanje celotnega sistema. Standardna CAN sporočila omogočajo nastavljen podatkovni del sporočila. Pošiljamo torej sporočilo brez podatkovnega dela, to so sporočila, v katerih pošiljamo samo zahtevo določenemu uporabniku, naj posreduje določene informacije. Lahko pa pošiljamo sporočila z dolžino 8 bajtov podatkovnega dela. Zgradbo sporočila prikazuje slika 4.3. Cilj standarda SAE J1939 je pošiljanje, kar se da polnih podatkovnih okvirjev, podatkovni okvirji dolžine 8 bajtov, saj s tem zmanjšujemo število sporočil, potrebnih za prenos podatkov. Manj podatkovnega okvirja kot uporabimo, več sporočil moramo poslati, s tem pa povečujemo promet vodila in povečujemo verjetnost napak. Že sam standard je zamišljen tako, da prek številke skupine parametrov v eno sporočilo združujemo sorodne parametre, podatke v sporočilu pa lahko združujemo tudi po skupnosti podsistema. Vsi podatki, namenjeni določeni napravi, se pošiljajo v enem sporočilu. Podatke pa lahko združujemo tudi po intervalu pošiljanja. Če imamo neko manjše število podatkov, ki se pošiljajo na enak interval, jih združimo v eno sporočilo in tako zmanjšujemo ozko grlo na omrežju. Poznamo več vrst sporočil:

- zahteva in prošnja,
- oddaja/odziv,
- potrditev,
- funkcije skupine.

Sporočila, s katerimi pošiljamo razne prošnje, potrditve ali funkcije skupine, imajo predpisane posebne PGN-je. Sporočila, s katerimi pa pošiljamo zahteve ali oddaje/odzive, pa imajo lahko katerikoli PGN [2].

Definicija sporočila, v katerem pošiljamo zahtevo, je navadno sporočilo. PGN razumemo kot zahtevo in ni točno določen. Sporočilo, s katerim pošiljamo zahtevo, nima točno določene številke skupine parametrov ali obsega številke skupine parametrov, ampak je lahko vsak PGN. Tako sporočilo tudi podpira oba formata PDU 1 in PDU 2 [2].

Sporočilo, v katerem pošiljamo prošnjo po informaciji točno določene naprave ali pa pošiljamo prošnjo po informaciji splošno v omrežje, na katerega se mora odzvati več naprav iz omrežja. Sporočilo s prošnjo podpira samo komunikacijo vsak z vsakim (PDU 1), odziv pa podpira oba tipa komunikacije vsak z vsakim in komunikacijo širšega tipa. Komunikacija, v kateri pošiljamo prošnjo pa je povezana s točno določenim PGN-jem, kar prikazuje spodnji primer.

Ime skupine parametrov:	Prošnja
Številka skupine parametrov:	59904 (00EA00 <sub>hex</sub> )
Definicija:	Prošnja skupine parametrov posamezne naprave ali vseh naprav v omrežju
Hitrost prenosa:	Določi proizvajalec, vendar ne večkrat kot 2 ali 3 krat na sekundo.
Dolžina podatkovnega dela:	3 bajte (CAN DLC=3)
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	0
PDUf:	234
PDU s:	Ciljni naslov (uporabimo globalni ali specifični naslov)
Prioriteta:	6
Opis podatkovnega dela:	Bajti 1, 2 in 3 = zelena številka skupine parametrov

Ciljni naslov definira točno določeno napravo v omrežju. Lahko pa s sporočilom kličemo vse naprave v omrežju. Podrobne vrednosti in razlike med naslavljanjem ene ali več naprav v omrežju pa prikazuje tabela 8 [2].

Tabela 8: Tip sporočila: prošnja, podrobne pomembne vrednosti [2].

Tip sporočila	PGN	Ciljni naslov	Naslov vira	Bajt 1	Bajt 2	Bajt 3
Splošna prošnja	59904	255	Zahtevnik	PGN (LSB)	PGN	PGN (MSB)
Točno določena naprava	59904	Odzivnik	Zahtevnik	PGN (LSB)	PGN	PGN (MSB)

Ko naprava sprejme zahtevo, mora odgovoriti ne glede na to, v kakšnem stanju ali fazi je. Ima možnost poslati negativni odziv (NACK), to pride v poštev takrat, kadar PGN ne obstaja. Splošna zahteva v omrežju ne bo sprožila negativnega odziva, ko določena naprava ne bo prepoznala PGN-ja, NACK je odziv v komunikaciji vsak z vsakim [2].

Sporočilo oddaja/odziv ima PGN definiran kot podatkovno sporočilo. Ne navezuje se na posebno določen PGN ali obseg točno določenih PGN-jev. Oddaja/odziv je normalno sporočilo v normalnem komunikacijskem toku. Podpira oba formata komunikacije, vsak z vsakim in komunikacijo širšega tipa.

Standard SAE J1939 opisuje dva načina potrditve. Prvi, vgrajen v sam CAN okvir, in drugi, potrjen s potrditvenim sporočilom. Potrditven del sporočila potrjuje uspešno preverjanje uspešnosti poslanega sporočila, to je uspešno preverjanje CRC. Potrditveno sporočilo J1939 je odzivno sporočilo na zahtevo ali željo in obsega točno določene PGN-je [2].

Ime skupine parametrov:	Potrditev
Številka skupine parametrov:	59392 (00E800 <sub>hex</sub> )
Definicija:	Predvideva potrditev pogovora med dvema napravama
Hitrost prenosa:	Po prejemu zahteve
Dolžina podatkovnega dela:	8 bajtov
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	0

PDUf: 232

PDU s: Ciljni naslov (globalni, 255)

Prioriteta: 6

Opis podatkovnega dela: Bajti 1 do 8= pozitivna potrditev, negativna potrditev, zavrnjen dostop ali nemogoč odziv.

Pozitivna potrditev (ACK):

Bajt	Vrednost
1	Kontrolni bajt je 0.
2	Vrednost skupine funkcij.
3–4	Rezervirano (FF <sub>hex</sub> ).
5	Naslov potrditve
6	PGN zahtevanih podatkov - LSB
7	PGN zahtevanih podatkov
8	PGN zahtevanih podatkov - MSB

Negativna potrditev (NACK):

Bajt	Vrednost
1	Kontrolni bajt je 1.
2	Vrednost skupine funkcij.
3–4	Rezervirano (FF <sub>hex</sub> ).
5	Naslov negativne potrditve.
6–8	PGN zahtevanih podatkov.

Zavrnjen dostop (PGN podprt, varnostna zavrnitev dostopa):

Bajt	Vrednost
1	Kontrolni bajt je 2.
2	Vrednost skupine funkcij.
3–4	Rezervirano (FF <sub>hex</sub> ).
5	Naslov zavrnjenega dostopa.
6–8	PGN zahtevanih podatkov.

Nemogoč odziv (PGN podprt, ECU se ne odziva, kasnejša zahteva po podatkih):

Bajt	Vrednost
1	Kontrolni bajt je 3.
2	Vrednost skupine funkcij.
3–4	Rezervirano (FF <sub>hex</sub> ).
5	Naslov zasedene naprave.
6–8	PGN zahtevanih podatkov.

Poznamo pa še en tip sporočila, to je skupina funkcij. Ta tip sporočila se uporablja samo v določenih primerih. Ti primeri so, ko hočemo pošiljati sporočilo z registrirano številko skupine parametrov. Drugi tip je sporočilo, s katerim nastavljamo omrežje, tretji tip je pošiljanje velikih sporočil. To so sporočila, kjer podatkovni okvir presega velikost 8 bajtov in vsebino pošiljamo z več komunikacijskimi paketi. Poznamo tri različna registrirana sporočila. To je registrirano sporočilo A, registrirano sporočilo A2 in registrirano sporočilo B. Razlika pa je vidna iz spodnjih primerov.

Ime skupine parametrov:	Registrirano sporočilo A
Številka skupine parametrov:	61184 (00EF00 <sub>hex</sub> )
Definicija:	Registrirano število skupine parametrov uporablja format PDU 1 in tip komunikacije vsak z vsakim.
Hitrost prenosa:	Določi proizvajalec.
Dolžina podatkovnega dela:	Od 0 pa do 1785 bajtov. Podpira tudi pošiljanje enega sporočila deljenega v več paketih.
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	0
PDUf:	239
PDU s:	8 bitni naslov naslovnika, ki ga določi proizvajalec.
Prioriteta	6
Opis podatkovnega dela:	Določi proizvajalec.

Ime skupine parametrov:	Registrirano sporočilo A2
Številka skupine parametrov:	126720 (01EF00 <sub>hex</sub> )
Definicija:	Registrirano število skupine parametrov uporablja format PDU 1 in tip komunikacije vsak z vsakim.
Hitrost prenosa:	Določi proizvajalec.
Dolžina podatkovnega dela:	Od 0 pa do 1785 bajtov. Podpira tudi pošiljanje enega sporočila deljenega v več paketih.
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	1

PDUf:	239
PDU s:	8 bitni naslov naslovnika, ki ga določi proizvajalec.
Prioriteta	6
Opis podatkovnega dela:	Določi proizvajalec.

Ime skupine parametrov:	Registrirano sporočilo B
Številka skupine parametrov:	65280 - 65535 (00FF00 <sub>hex</sub> - 00FFFF <sub>hex</sub> )
Definicija:	Registrirano število skupine parametrov uporablja format PDU 2 in širši tip komunikacije.
Hitrost prenosa:	Določi proizvajalec.
Dolžina podatkovnega dela:	Od 0 pa do 1785 bajtov. Podpira tudi pošiljanje enega sporočila deljenega v več paketih.
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	0
PDUf:	255
PDU s:	Razširitev skupine - določi proizvajalec.
Prioriteta	6
Opis podatkovnega dela:	Določi proizvajalec.



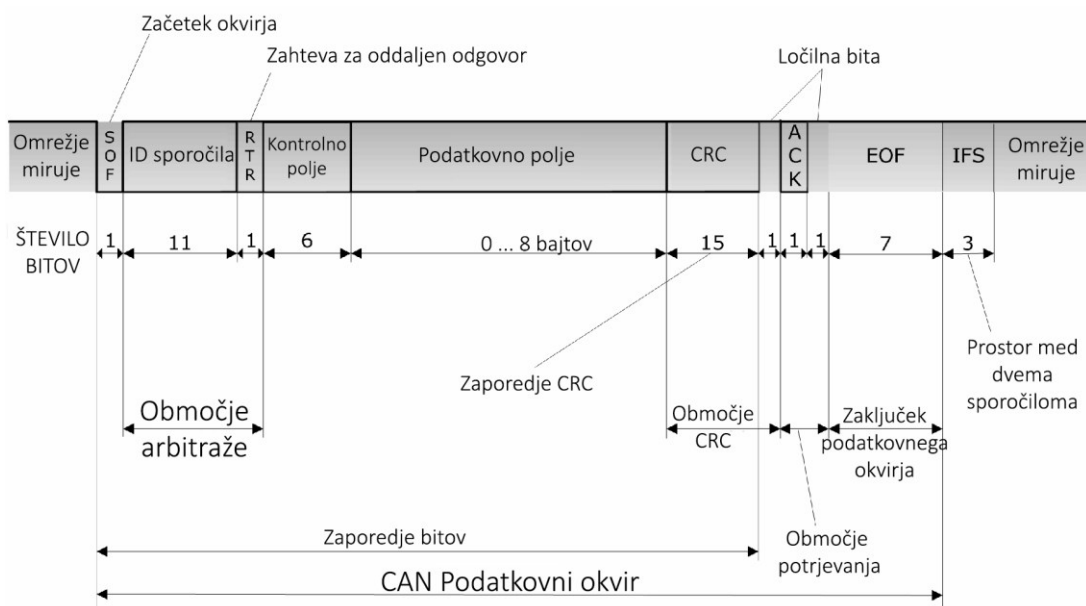
#### 4.4 Prenosne funkcije protokola

Čeprav je CAN bus komunikacija zelo učinkovita in se je v manjših vozilih ter v industrijskih aplikacijah izkazala za zelo učinkovito, pa ne pokriva potreb v težki mehanizaciji, kamionih in avtobusih. Prav zaradi slabosti in pomanjkljivosti se je razvil višjenivojski protokol J1939, ki nadgrajuje osnovno CAN bus komunikacijo na točkah, kjer so se izkazale pomanjkljivosti. Največja pomanjkljivost je pošiljanje podatkov, ki obsegajo velikost, večjo od 8 bajtov. To pomanjkljivost J1939 rešuje s pošiljanjem več zaporednih podatkovnih okvirjev v velikosti 8 bajtov. S pošiljanjem več zaporednih sporočil lahko pošljemo do 1785 bajtov v enem sporočilu. To storimo tako, da zgeneriramo sporočilo v dolžini do 1785 bajtov, ga »razrežemo« in na vodilo pošljemo v podatkovnih okvirjih z velikostjo 8 bajtov, nato pa ga sprejemnik nazaj sestavi v celoto. Funkciji pošiljanja več zaporednih sporočil rečemo prenosna funkcija protokola (TP). Prenosno funkcijo podrobneje delimo v dve podfunkciji:

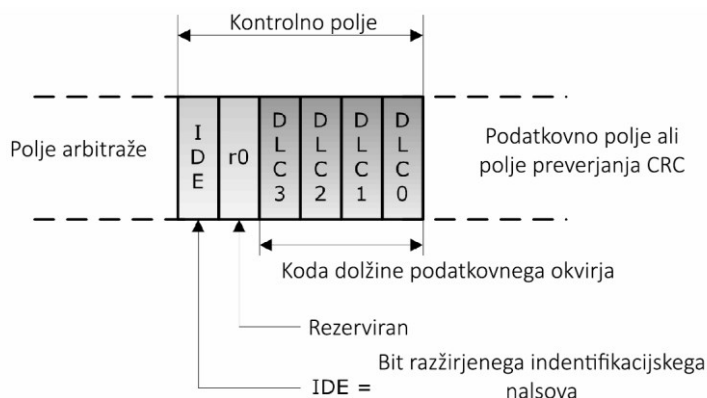
- sestavljanje in razstavljanje sporočila in
- upravljanje povezave.

Dodatna prenosna funkcija protokola skrbi tudi za nadzor prometa pošiljanja, potek pošiljanja in končni prevzem pri uporabniku. Prenosne funkcije protokola podpirajo samo PDU 1 format, to je komunikacija vsak z vsakim, če pa hočemo sporočilo poslati globalno vsem napravam, pa moramo uporabiti ciljni naslov 255 [2].

Posamezna številka skupine parametrov zahteva podatkovni okvir, ki je večji od 8 bajtov. To težavo osnovnega CAN bus protokola rešuje višjenivojski protokol J1939. S prenosno funkcijo lahko prenašamo sporočila dolžine do 1785 bajtov. Vsi bajti od 9 do 1785 so definirani kot več bajtno sporočilo in ga je treba najprej sestaviti, pred oddajo na omrežje razstaviti, saj gredo na omrežje samo sporočila dolžine 8 bajtov, nato pa ga morajo naprave še sestaviti nazaj v celoto. Sporočilo dobi pravi pomen šele, ko je sporočilo ponovno sestavljeno in so vse informacije interpretirane na pravilni način. Slika 4.10 prikazuje zgradbo CAN sporočila. Slika 4.11 pa prikazuje zgradbo CAN kontrolnega polja. Kontrolno polje nam poda informacijo, koliko od 0 do 8 bajtov podatkovnega okvirja je uporabljenih. V zgradbi sporočila vidimo, da obstaja vrsta sporočila, kjer kontrolnemu polju sledi preverjanje CRC [2].



Slika 4.10: Prikazuje arhitekturo CAN sporočila [2].



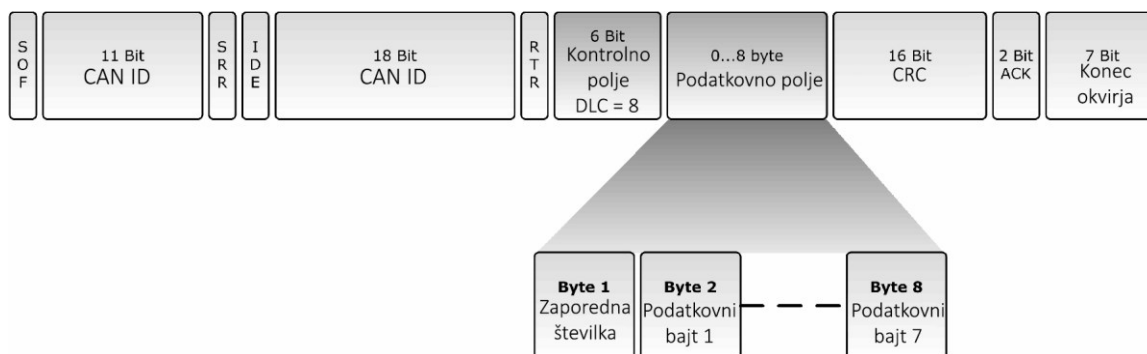
Slika 4.11: Prikazuje zgradbo CAN kontrolnega polja [2].

Če hočemo, da bo nemoteno potekala komunikacija pošiljanja in prejemanja več bajtnega sporočila, se moramo držati določenih navodil:

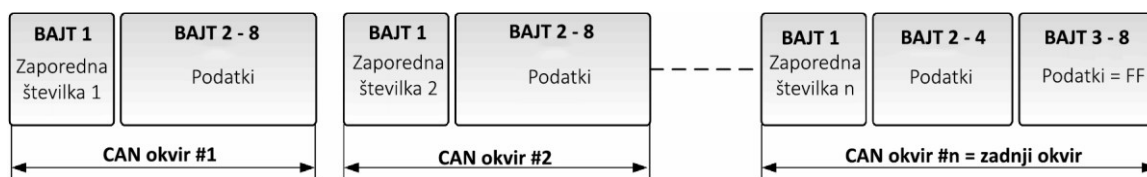
- Vsako posamezno sporočilo uporablja točno določeno številko skupine parametrov za prenos podatkov (ang. Data transfer PGN) (60160, TP.DT = Transfer Protocol Data Transfer). Vsa sporočila znotraj enega več bajtnega sporočila uporabljajo isto identifikacijsko številko.
- Kontrola pretoka informacij je nadzorovana preko točno določene številke skupine parametrov PGN = 60146 TP. CM = Transfer Protocol Communication Management.
- Dolžina podatkovnega dela v večbajtnih sporočilih mora vedno biti 8 bajtov (DLC = 8).
- Prvi bajt v podatkovnem okvirju vsebuje zaporedje števil 1 do 255.
- Ostalih 7 bajtov se zapolni z informacijami, ki jih je treba prenesti.

- Vse neuporabljene podatkovne bajte se zapolni z FF<sub>hex</sub>.

Končno dolžino sporočila določa ustrezna številka skupine parametrov. Dolžino lahko izračunamo tako, da vzamemo zaporedno število poslanega podatkovnega okvirja in dodamo še 7 bajtov, s katerimi dejansko prenašamo informacije, tako skupaj znesse 1785 bajtov, s katerimi lahko prenašamo informacije [2].



Slika 4.12: Zgradba večbajtnega sporočila [2].



Slika 4.13: Sosledje zaporednih števil in podatkovnih bajtov [2].

Sliki 4.12 in 4.13 podrobneje prikazujeta, kako v praksi sestavljamo večbajtno sporočilo. V prvem bajtu sporočila prenašamo podatek, za kateri podatkovni okvir po vrsti gre, ostalih 7 bajtov pa nosi informacije. Vsako izmed večbajtnih sporočil pa vsebuje vso periferno, razlika je samo v podatkovnem okvirju, kjer zaporedna številka teče naprej vse do konca, podatkovni del pa vsakič vsebuje nadaljevanje še ne poslanih podatkov. Ko naprava prebere vsa sporočila, sporoči napako, če kakšno izmed vmesnih sporočil manjka, saj moramo vedno poslati in prevzeti vsa sporočila, prav zaradi tega je konstruirana varovalka, da četudi imamo za pošiljanje namenjenih manj kot 1785 bajtov podatkov, jih moramo poslati vseh 255, neuporabljene pa zapolnimo z FF<sub>hex</sub>. Šele, ko je prevzeto celotno sporočilo, to pomeni, da je naprava oziroma sprejemnik prejel vseh 255 sporočil uspešno, jih naprava sestavi v celoto in posreduje naprej aplikacijski plasti v obdelavo [2].

Upravitelj povezave višjenivojskega protokola J1939 opredeljuje komunikacijo širšega tipa za pošiljanje večbajtnih sporočil, še posebej pa nadzor prenosa za večbajtna sporočila vsak z vsakim. Nadzor prenosa ima funkcijo najave večbajtnega sporočila v omrežje, nato pošiljanje

zahteve za pošiljanje, sledi znak za pošiljanje in na koncu najavi konec sporočila. Če pa pride do izpada, pa najavi prekinitve zaradi izpada omrežja [2].

Komunikacija širšega področja je namenjena oddajanju sporočila na omrežje in takrat vse naprave v omrežju poslušajo. Njen namen ni pošiljanje na točno določeni naslov, zato se tudi uporablja globalni naslov 255. Ker pa za pošiljanje večbajtnega sporočila uporabljamo komunikacijo širšega področja, moramo sporočilo najprej najaviti (ang. Broadcast Announce Message → BAM), če hočemo to sporočilo poslati točno določeni napravi ob najavi sporočila pošiljamo naslednje podatke:

- PGN večbajtnega sporočila,
- velikost sporočila in
- število paketov.

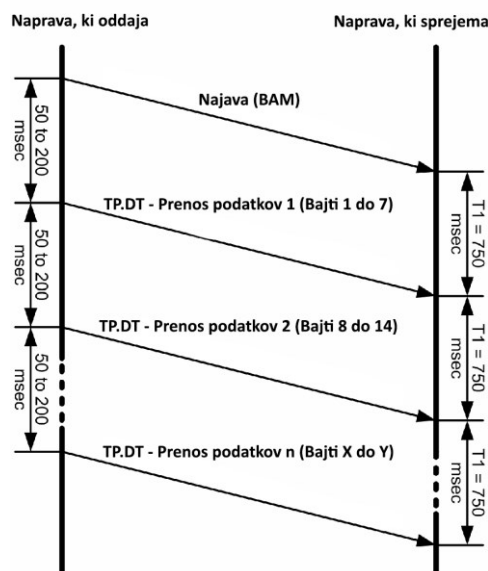
BAM sporočilo dovoljuje vsem napravam, za katere bo večbajtno sporočilo pomembno, da se pripravijo za sprejem. Vse naprave, ki bodo začele sprejemati večbajtno sporočilo, morajo zagnati posebno rutino, s katero bodo sprostile dovolj spomina, da bodo celotno sporočilo lahko sprejele in ga kasneje obdelale. Najava sporočila je vgrajena v transportni protokol, natančneje v upravitelju povezave. (TPCM) PGN 60416 se nadaljuje s prenosom podatkov, ki pa ima PGN 60160 – prenos podatkov [2].

Primer pomembnih skupin parametrov, ki so potrebni za prenos večbajtnega sporočila, so prikazani v nadaljevanju naloge.

Ime skupine parametrov:	Protokol za prenos podatkov – prenos podatkov (TP.DT)
Številka skupine parametrov:	60160 (00EB00 <sub>hex</sub> )
Definicija:	Uporablja se za prenos podatkov v večbajtnem sporočilu
Hitrost prenosa:	Odvisno od številke skupine parametrov
Dolžina podatkovnega dela:	8 bajtov.
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	0
PDUf:	235
PDU s:	Ciljni naslov
Prioriteta	7
Opis podatkovnega dela:	Določi proizvajalec

Podrobno po bajtih:	1 – zaporedna številka sporočila 2-8 – podatki
Ime skupine parametrov:	Protokol za prenos podatkov – upravitelj povezave (TP.CM)
Številka skupine parametrov:	60416 (00EC00 <sub>hex</sub> )
Definicija:	Uporablja se za upravljanje povezave za kontrolo pretoka, to sporočilo je najava v omrežje
Hitrost prenosa:	Odvisno od številke skupine parametrov
Dolžina podatkovnega dela:	8 bajtov
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	0
PDUf:	236
PDU s:	Ciljni naslov (=255 za splošno oddajo)
Prioriteta	7
Opis podatkovnega dela:	Samo za najavo sporočila
Podrobno po bajtih:	1 - Kontrolni bajt = 32 2, 3 - Velikost sporočila (število bajtov) 4 – Skupno število paketov 5 – Rezerviran (zapolni se z vrednostjo FF <sub>hex</sub> ) 6-8 – Številka skupine parametrov več bajtnega sporočila (6=LSB, 8=MSB)

Da poteka pošiljanje večbajtnega sporočila nemoteno, je potrebno definirati časovne zamike med pošiljanjem najave in potem med samimi podatkovnimi sporočili. Te zakasnitve so prikazane na sliki 4.14 [2].



Slika 4.14: Prikazuje zakasnitve pošiljanj večbajtnega sporočila [2].

Perioda pošiljanja je med 50 in 200 ms. Prekinitev se bo pojavila, če bo časovni interval med pošiljanjem dveh sporočil daljši od 750 ms. Prekinitev bo povzročila dokončno prekinitev pošiljanja vseh nadaljnjih sporočil. Povezava bo zaključena, če pošiljatelj pošlje zadnje sporočilo, ali če se pojavi prekinitev, ki je daljša od 750 ms, ki prekine pošiljanje.

Večbajtno sporočilo lahko pošljamo tudi s pomočjo komunikacije tipa vsak z vsakim. To sporočilo pošljamo na točno določen naslov, hkrati pa je predmet nadzora pretoka. Nadzor pretoka preverja tri naj pomembnejše mejnike pošiljanja:

- Inicializacija povezave: pošiljatelj najprej pošlje zahtevo za pošiljanje, sprejemnik odda odgovor prosto za pošiljanje, to pomeni, da se je naprava z inicializirala in lahko sprejme večbajtno sporočilo, če pa je karkoli, zaradi česar naprava ne bi mogla sprejeti celotnega sporočila, pa pošlje odgovor, da je povezava prekinjena.
- Prenos podatkov: pošiljatelj pošlje številko skupine parametrov za prenos podatkov, nato pa sledijo podatkovni okvirji sporočila. Še vedno pa se lahko zgodi, da se med pošiljanjem večbajtnega sporočila povezava prekine ali pa naprava, ki sporočilo sprejema, ne more več sprejemati, zato pošlje sporočilo, da je povezava prekinjena.
- Zaključek: ko sprejemnik sprejme zadnji podatkovni okvir, odgovori s sporočilom konec sporočila – ACK.

Nadzor pretoka preverja tudi časovne intervale, saj more med dvema sporočiloma preteči dovolj časa, da sprejemnik lahko sporočilo obdela, vendar ne preveč, ker se lahko razume, da je na vodilu ali oddajniku prišlo do napake in se pošiljanje prekine. Za nemoteno in kvalitetno pošiljanje so pomembni naslednji časovni intervali:

- $T_r=200$  ms; to je časovni interval, na katerega se morata sprejemnik ali oddajnik odzvati s podatkovnim sporočilom ali nadzorom pretoka. Če preteče več časa, komunikacija pade, zato se javi napaka.
- $T_h=500$  ms; če sprejemnik iz kakršnegakoli razloga potrebuje zakasnitev sprejemanja, pošlje sporočilo prosto za pošiljanje, kjer je število paketov postavljeno na nič. Če potrebuje zakasnitev za 500 ms, sprejemnik pošlje sporočilo prosto za pošiljanje, ki se ponavlja vsakih 500 ms. Ko pa je sprejemnik zmožen normalnega sprejemanja, pošlje običajno sporočilo, ki je prosto za pošiljanje.
- $T_1=750$  ms; do prekinitve pride, če je časovni interval večji od  $T_1$ , kadar je naprava sredi pošiljanja in se pričakujejo naknadna sporočila.
- $T_2=1250$  ms; po preteku 1250 ms od prejemu sporočila, prostega za pošiljanje pride do prekinitve, če oddajnik ne začne s pošiljanjem sporočil.
- $T_3=1250$  ms; časovni interval  $T_3$  povzroči prekinitev. To se nanaša na čas, ki preteče po poslanem zadnjem sporočilu. Če preteče več časa, kot ga določa časovni interval  $T_3$  oz. od prejema zadnjega sporočila v več bajtnem sporočilu, in sprejemnik ne pošlje sporočila, da je konec sporočila ali prosto za pošiljanje, pride do prekinitve.
- $T_4=1050$  ms; časovni interval, ki je daljši od  $T_4$  povzroči prekinitev, če po zakasnitvi s funkcijo pošiljanja prosto za pošiljanje vsakih 500 ms funkcije ne deaktiviramo s pošiljanjem prosto za pošiljanje.

Obstajajo pa tudi drugi razlogi za prekinitve. Prekinitev se proži, ko pošiljatelj pošlje zadnji podatkovni okvir, oziroma ko sprejemnik prejme zadnji podatkovni okvir in nato preteče časovni interval  $T_1$ . Do prekinitve pride, ko pošiljatelj prejme sporočilo konec sporočila, ali ko prejme sporočilo padec komunikacije.

Sporočilo kontrole prenosa se zahteva za pošiljanje, prosto za pošiljanje in druge zgoraj opisane. Ta sporočila so vgrajena v prenosni protokol – upravitelj povezave (TP.CM) PGN 60416, prenos podatkov pa opravi številka skupine parametrov za prenos podatkov 60160. V nadaljevanju bom predstavil zgradbo sporočil, pomembnih za pošiljanje večbajtnega sporočila [2].

Ime skupine parametrov:	Protokol za prenos podatkov – upravitelj povezave (TP.CM)
Številka skupine parametrov:	60416 (00EC00 <sub>hex</sub> )
Definicija:	Uporablja se za upravljanje povezave (zahteva za pošiljanje, prosto za pošiljanje, prekinitev ...)
Hitrost prenosa:	Odvisno od številke skupine parametrov
Dolžina podatkovnega dela:	8 bajtov
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	0
PDUf:	236
PDU s:	Ciljni naslov (=255 za splošno oddajo)
Prioriteta	7
Opis podatkovnega dela:	Samo za najavo sporočila

Podrobno po bajtih (odvisno od sporočila):

TP.CM_RTS:	<p>Način povezave: prošnja za pošiljanje</p> <p>1 – kontrolni bit = 16.</p> <p>2, 3 – velikost sporočila (število bitov)</p> <p>4 – skupno število paketov</p> <p>5 – maksimalno število paketov, odziv na prosto za pošiljanje</p> <p>6–8 – PGN večbajtnega sporočila</p>
TP.CM_CTS:	<p>Način povezave: prosto za pošiljanje</p> <p>1 – kontrolni bit = 17</p> <p>2 – skupno število paketov (ne sme presegati 5 bajtov v RTS)</p> <p>3 – številka naslednjega paketa zapovrstjo</p> <p>4, 5 – rezervirano (zapolni se z vrednostjo FF<sub>hex</sub>)</p> <p>6-8 – PGN večbajtnega sporočila.</p>



TP.CM_EndOfMsgACK:	Način povezave: potrditev konca sporočila 1 – kontrolni bit = 19 2, 3 – velikost sporočila (število bajtov) 4 – skupno število paketov 5 – rezervirano (zapolni se z vrednostjo FF <sub>hex</sub> ). 6-8 – PGN večbajtnega sporočila
TP.CM_Abort:	Način povezave: Prekinitev. 1 – kontrolni bit = 255 2 – razlog za prekinitev (opisano spodaj) 3-5 – rezervirano (zapolni se z vrednostjo FF <sub>hex</sub> ) 6-8 – PGN večbajtnega sporočila

Razlogi za prekinitev so:

- 1 – Naprava je zaposlena z drugo sejo in ne more obdelovati še dodatnih informacij.
- 2 – Naprava se ubada s pomanjkanjem kapacitet.
- 3 – Predolg časovni interval.
- 4 – 250 – Rezerviran za standard SAE.

#### 4.5 Upravljanje z omrežjem J1939

Upravljanje z omrežjem opisuje dokument SAE J1939/81. Tema upravljanja z omrežjem se v višjenivojskem protokolu J1939 navezuje predvsem na proces zahteve po potrditvi naslova (ang. Address Claiming Process). Dinamično naslavljanje oziroma potrjevanja naslova ne podpira noben drug CAN bus višjenivojski protokol. Osnovni CAN komunikacijski protokol ne podpira naslovov naprav, uporablja samo 11 bitno identifikacijsko številko, v primeru CAN 2.0B pa 29 bitni ID [2].

Vsaka krmilna enota v omrežju mora imeti obvezno eno ime in en naslov, da lahko nemoteno in kvalitetno komunicira v omrežju. 8 bitni naslov nam pove naslov naprave, ki sporočilo oddaja ali pa ciljni naslov, komu je sporočilo namenjeno. V omrežju pa imamo lahko več naprav z isto nalogo oziroma opravljajo isto delo, zato si delijo tudi sporočila. V tem primeru pa moramo uporabljati tudi indikatorje, ki so vključeni v ime naprave. Prek indikatorjev izbiramo točno določeno napravo. Na primer, če imamo v omrežju dva motorja, ki sta nameščena v kolesu, morata imeti v zavijanju različno hitrost, da bo naprava zavila, zato za

desni motor uporabimo indikator 1, za levi motor pa indikator 2. Standard J1939 dovoljuje do 255 naprav v istem omrežju z istimi funkcijami, vendar z različnimi naslovi in imeni. SAE J1939 definira 64 bitno ime. Zgradbo imena prikazuje slika 4.15 [2].



Slika 4.15: Zgradba imena [2].

Naprave bi lahko za svojo registracijo in registracijo svojih funkcij v omrežje uporabljale 64 bitno ime, ki je najbolj primerno, vendar pa zahteva nerazumna sredstva za nemoteno komunikacijo, zato se ime uporablja samo, ko se v omrežje prijavlja nova naprava, a to le v bolj zakompliciranih sistemih. Navadna preprosta omrežja te funkcije ne uporabljajo. Rešitev, ki jo uporabljajo bolj zapletena omrežja, je proces potrjevanja naslova. To je proces, kjer naprava uporabi samo 8 bitov naslova za identifikacijo vira sporočila ali za naslavljanje naslovnika. Potrjevanje naslova se izvede takoj po inicializaciji omrežja. Takoj, ko se vse naprave »postavijo«, se izvede potrjevanje naslova, s tem zagotovimo, da imajo vse naprave v omrežju različne naslove. Lahko pa pride do primerov, ko mora imeti posamezna naprava več naslovov, saj mora ista naprava posredovati informacije na različne zahteve. Vsak naslov pa ima tudi svoje ime, saj gre za različno funkcijo. Kot primer navedimo navor električnega vozila. Senzor bo meril navor, ki ga motor proizvaja, da vozilo poganja, meril pa bo tudi negativen navor ob zaviranju. Ker gre za dve različni funkciji, potrebujemo dva naslova in dve imeni [2].

Da se proces dodajanja naslovov in poimenovanja funkcij ter naprav poenostavi, je združenje SAE J1939 definiralo izraz predpostavljenih naslovov. To je tabela naslovov, ki je osnova ob generiranju novih naprav. Z uporabo predpostavljenih naslovov pohitimo postopek priklopa naprave v omrežje, hkrati pa zmanjšamo verjetnost uporabe istih naslovov, saj naprava izbere ime glede na funkcijo, ki jo opravlja. Če je v omrežju več enakih naprav, izbere na začetku osnovni naslov, če pa je le-ta že zaseden, pa gre na prvega naslednjega [2].

Potrjevanje naslova v omrežju se lahko izvaja na dva različna načina.

- Prvi način je pošiljanje sporočila za potrjevanje naslova. Proces poteka tako, da ko se naprava inicializira v omrežje, pošlje na vodilo sporočilo za potrjevanje naslova.

Naprave v omrežju prejmejo sporočilo in primerjajo naslov s svojim, če nima nobena obstoječa naprava istega naslova, je postopek zaključen, nova naprava pa obdrži naslov. Če pa pride do primera, ko se v omrežju pojavi naprava z enakim naslovom, pa napravi po preverjanju naslova preverita še ime, naprava z najnižjo vrednostjo imena bo obdržala naslov, naprava z višjim naslovom, pa ga mora zamenjati.

- Drugi način pa je pošiljanje prošnje za potrjevanje naslova. Ta način se uporablja predvsem v namene preverjanja naprav, ki so trenutno priključene v omrežje, in za potrebe zunanjega nadzora omrežja (razvojna faza). Tukaj pa se na omrežje pošlje sporočilo z zahtevo, da naprave pošljejo svoje naslove, uporabnik ali naprava, ki je to sporočilo poslala pa preverja prispele naslove, če kateri sovпада z njenim. Če prejme isti naslov, kot ga ima sama že iz omrežja, se ponovno preverja vrednost imena in naprava z nižjo vrednostjo imena zmaga. Naprava z višjo vrednostjo imena pa mora spremeniti naslov.

Zahteva po potrjevanju naslovov tudi hoče, da so v omrežju naprave, ki so sposobne same spreminjati naslove. Če bi imeli v omrežju napravo, ki bi poslala zahtevo za potrjevanje naslova in bi izgubila arbitražo ter prejela zahtevo po spremembi naslova, le-tega pa sama ne bi bila sposobna, bi izpadla iz komunikacije oziroma bi šla v način napake. Ko pride na omrežju do konflikta naslovov, ko imata dve napravi isti naslov, situacijo lahko razvrstimo v dve skupini [2].

Prva skupina je tako imenovana zmožnost enega naslova naprave. Druga skupina pa je arbitraža naslova naprave.

Zmožnost enega naslova naprave zaokroža več metod, po kateri rešujemo nastalo situacijo. Vsem metodam pa je skupno to, da se mora izvesti neki dodaten proces, hkrati pa je obseg naslovov za potrjevanje omejen. V omenjeno skupino spadajo naslednji načini reševanja:

- Samonastavljive naprave

Samonastavljive naprave imajo možnost, da ko pride do »trka« naslovov, same spremenijo naslov, takim napravam rečemo, da imajo dinamično naslavljanje, saj naslov spremenijo samodejno.

- Nastavljive naprave na ukaz

Z raznimi servisnimi napravami, komunikatorji lahko kot skrbnik omrežja zahtevamo, da neka naprava spremeni naslov, temu procesu rečemo, da je nastavljanje naslova na zahtevo.

- Nastavljive naprave prek servisnega vmesnika

Večini naprav se lahko naslov spreminja prek servisnega vmesnika, to pomeni, da se na napravo priklopi ali poveže usposobljena oseba in prek servisne platforme nastavi naslov. Druga možnost pa je, da ima naprava strojno rešeno nastavljanje naslova prek DIP stikal.

- Nenastavljive naprave

Te naprave pa nimajo možnosti spreminjanja naslova in se ob trku, če ne zmagajo arbitraže, preklopijo v način napake.

Druga skupina, arbitraža naslova naprave, so naprave, ki sodelujejo v arbitražnem procesu imena, ko pride do trka naslovov. Zmaga naprava z najnižjo vrednostjo naslova. Te naprave so najbolj razširjene [2].

Dinamično naslavljanje je najbolj uporabna funkcija oziroma proces, saj se v omrežju vzpostavi ravnotežje samodejno, naprave se same »pogovorijo« in naslovijo ter so tako pripravljene na izvajanje komunikacije.

Vsaka naprava ima v omrežju J1939 vsaj eno ime in en naslov, ki se uporabljata za naslavljanje in identifikacijo. Lahko pa ena naprava upravlja več naslovov in več imen. Kot prikazuje slika 4.15, 8 bitni naslov naprave definira pošiljatelja in naslovnika sporočila. Ime naprave vsebuje indikator, ki nakazuje glavno funkcijo naprave, hkrati pa vpliva na arbitražni postopek naslova. [2]

Če podrobneje pogledamo pomene polj, ki jih prikazuje slika 4.15 – Zgradba imena. SAE J1939/81 definira 64 bitno ime, ki je sestavljeno iz desetih polj, pet od njih je definiranih s strani SAE, ostalih pet pa definira omrežje ali proizvajalec. Ime je pomembno zaradi nastopanja v arbitražnem postopku, druga prav tako pomembna stran pa je, da enoznačno definira aplikacijo krmilnika.

Razčlemba imena posamezno po sklopih:

- Zmožnost arbitražnega naslova: je pokazatelj, ki pokaže, kdaj lahko naprava opravi arbitražo (1-lahko, 0-ne more).

- Skupina industrije: koda skupine industrije je določena s strani združenja SAE. Poznamo več skupin industrije in izdelke razvrščamo glede na funkcije, ki jih opravljajo, ali glede na procese, v katerih bodo obratovali. Poznamo: oprema za avtoceste, oprema za kmetijstvo, oprema za gozdarstvo (podrobneje prikazuje tabela 9).

Tabela 9: Kode skupine industrije [2].

Koda skupine industrije	Industrija
0	splošno, podpira vse industrije
1	avtocestna oprema
2	kmetijska in gozdna oprema
3	gradbena oprema
4	vodna oprema
5	oprema za industrijske procese in nadzorna oprema
6, 7	rezervirano

- Primer sistema vozila; to polje tesno sodeluje z naslednjim poljem, ki je sistem vozila. V omrežju J1939 je treba združevati več različnih naprav v enem omrežju. Če podamo primer avtomobila; v avtomobilu je veliko identičnih naprav v enem sistemu, vse te naprave je treba uskladiti, da delujejo nemoteno, predvsem pa pravilno, saj bi lahko prišlo do velike katastrofe, če bi voznik v napačnem trenutku aktiviral napačno napravo z mislijo, da bo uporabil pravilno. Na primer 4 bitni sistem vozila ima vrednosti od 0 do 15, ki se dodelijo za vsako stopnjo posebej.
- Sistem vozila; ta 7 bitni sklop je določen s strani SAE. Kot smo že omenili v prejšnjem sklopu, ta dva sklopa sodelujeta in skupaj določata namen naprave.
- Rezervirano; tudi ta sklop je določen s strani SAE. Polje rezervirano je vedno postavljeno na nič. Namenjeno je bodočim podrobnejšim razčlembam s strani SAE.
- Funkcija; polje funkcija je določena s strani SAE. Ima vrednosti od 0 do 255, vendar z nekaterimi izjemami. Vrednosti enake ali večje od 128 so odvisne od panožne skupine. Vrednosti, ki so manjše 128, pa se ne navezujejo na nobeno drugo polje in so odvisne same od sebe.  
Primer: funkcija 133 pomeni tok produktov v kmetijski in gozdarski panožni skupini, v gradbeni panožni skupini pa pomeni prikaz vrednosti merjenja tal.
- Primer funkcije; polje primer funkcije deluje v povezavi s poljem funkcija. Ker omrežje J1939 podpira, da ima lahko več naprav iste funkcije, jih ločimo s primeri.

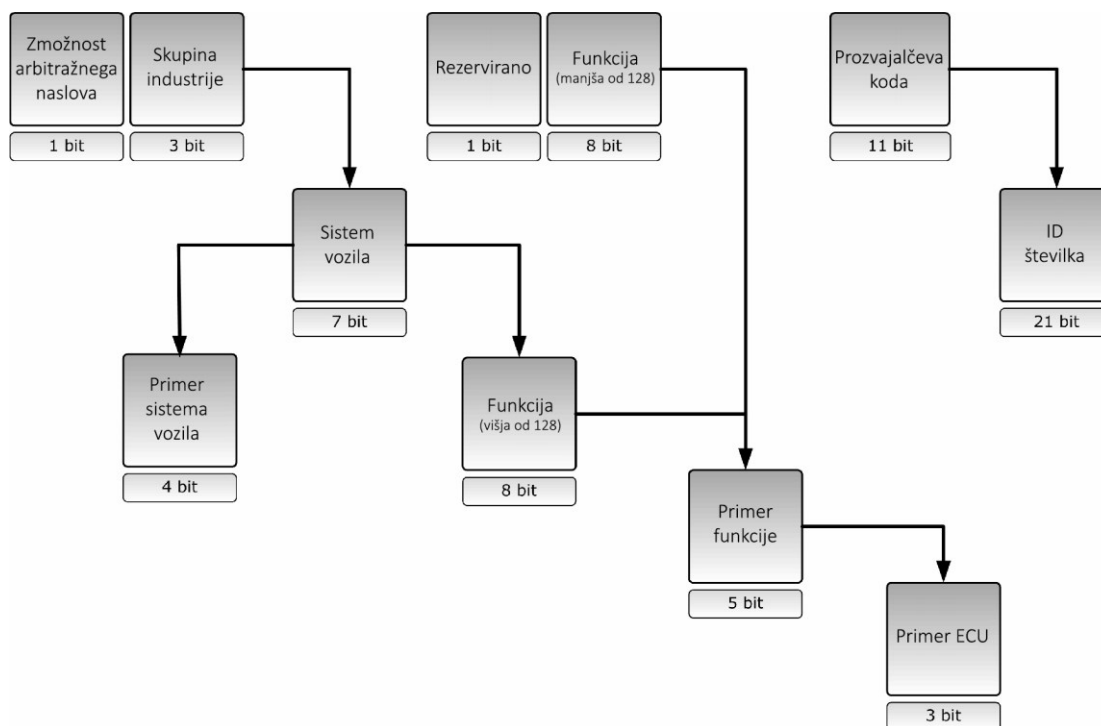
Prva funkcija v omrežju ima vrednost 0, vse naslednje pa imajo vrednost  $n+1$ . polje je 15 bitno.

- Primer ECU; s primeri ločimo tudi naprave v omrežju. V omrežju imamo identične naprave, ki opravljajo isto delo, le da podatki niso isti.

Primer: v omrežju avtomobila imamo dva identična merilnika hitrosti, le da prvi meri hitrost avtomobila, drugi pa meri hitrost priklopnika. Da lahko razberemo podatke, kdo je pošiljatelj, jih ločimo s primeri.

- Proizvajalčeva koda; 11 bitna proizvajalčeva koda je določena s strani SAE. Vsak proizvajalec ima svojo kodo, v kateri so po navadi zaobjeti naslednji podatki: kupčeva koda v sistemu, datum zadnje operacije, revizijska številka, ki sovпада z revizijo zadnje, aktualne dokumentacije in šarža materiala.
- ID številka; 21 bitna identifikacijska številka je določena s strani proizvajalca in mora omogočiti edinstveno ime naprave.

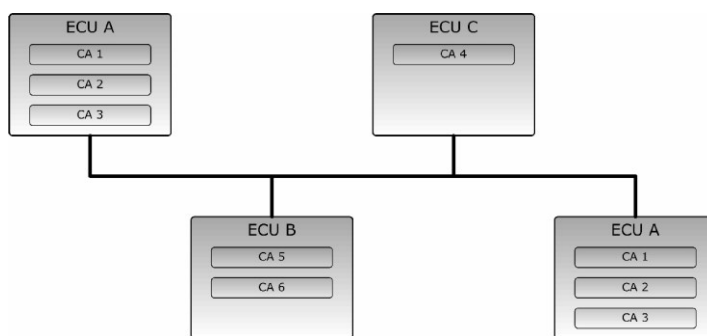
Odvisnosti posameznih polj znotraj imena prikazuje slika 4.16.



Slika 4.16: Odvisnosti posameznih polj znotraj imena [1].

Po standardu SAE J1939 je naslov določen z 8 biti, kar teoretično omogoča do 256 različnih naslovov. Če predpostavimo, da ima vsaka naprava v omrežju različen naslov, lahko sklepamo, da v posamezno omrežje priključimo 256 naprav, vendar pa je združenje omejilo

število naprav v omrežju na 30. To so storili, da bi bilo omrežje čim bolj pretočno (Slika 4.17) [2].



Slika 4.17: Primer CAN bus omrežja [1].

Slika prikazuje naprave – ECU, v katerih so narisane različne aplikacije krmilnikov – CA. Vsaka naprava ima lahko več aplikacij. Ker pa je naslov, naslov aplikacije in ne samo naprave, se številka 256 hitro zmanjša. Kot je že omenjeno v nalogi imamo dva naslova že vnaprej določena, naslov 255 je splošni naslov za naslavljanje vseh naprav v omrežju, naslov 254 je namenjen upravljanju z omrežjem, tako da je prostih še 253 naslovov. Na sliki 4.17 vidimo, da imamo v omrežju dve enaki napravi, to pomeni, da sta z arbitražnim postopkom uveljavljali naslove, zato je ena naprava mogla le-tega spremeniti. V omrežju imamo dve enaki napravi z drugačnim naslovom, natančneje, različne naslove imajo aplikacije, ki delujejo znotraj krmilnika. Za določanje naslovov pa obstaja tabela priporočenih (Tabela 10). Priporočeni naslovi so smiselno razdeljeni v skupine, delijo se po funkcijah naprav. Naprava se bo naslovila na prvi priporočeni naslov šele, če izgubi arbitražo, poviša vrednost naslova za vrednost 1 [2].

Tabela 10: Tabela priporočenih naslovov [2].

Skupina industrije	Priporočen naslov
skupina industrije 0: splošno za vsa področja	0–84: predpisani 85–127: rezervirani 248, 252–255: rezervirani
skupina industrije 1: avtocestna oprema	128–160: dinamični 161–247: predpisani
skupina industrije 2: kmetijska in gozdna oprema	128–207: dinamični 208–247: predpisani
skupina industrije 3: gradbena oprema	128–207: dinamični 208–247: predpisani
skupina industrije 4: vodna oprema	128–207: dinamični 208–247: predpisani
skupina industrije 5: oprema za industrijske procese in nadzorna oprema	128–207: dinamični 208–247: predpisani

Naslov 254 je torej rezerviran za upravljanje z omrežjem, imenujemo ga naslov nič. Ta naslov 254 je tudi v uporabi za sporočilo nezmožnosti potrditve naslova. 255 pa je t. i. splošni naslov za hkratno naslavljanje vseh naprav v omrežju [2].

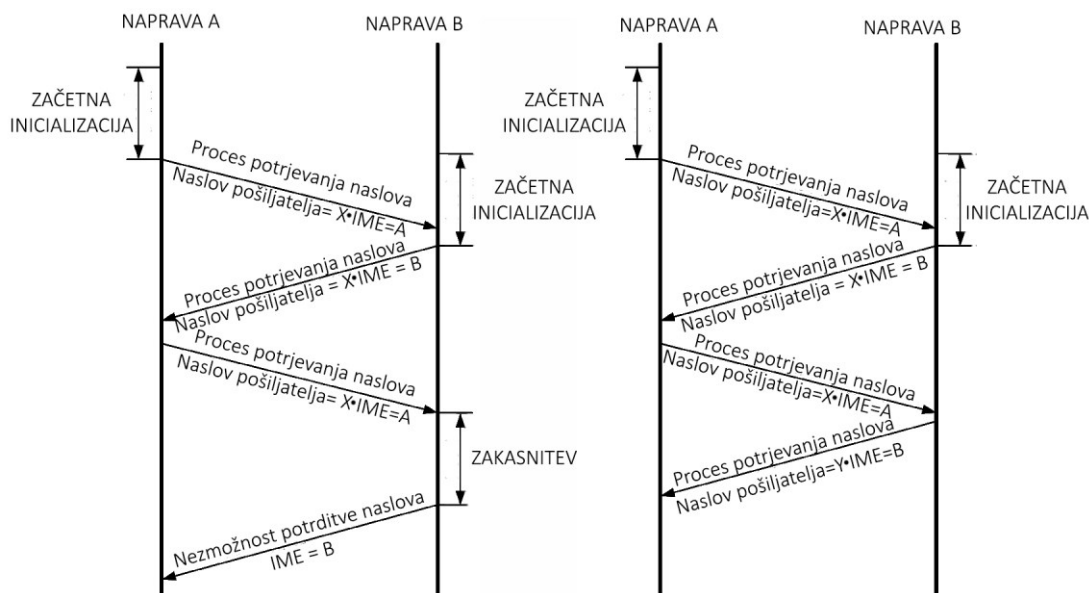
Proces upravljanja z omrežjem se navezuje predvsem na postopke potrjevanja naslovov. Za potrjevanje naslova uporabljamo tri različne številke skupine parametrov:

- PGN 59904 – prošnja po potrditvi naslova,
- PGN 60928 – sporočilo: naslov potrjen ali naslov ni potrjen,
- PGN 65240 – zahteva za potrditev naslova.

Sam proces potrditve lahko poteka na dva različna načina. Prvi način je pošiljanje sporočila za potrjevanje naslova. Ta način je običajno potrjevanje naslova v omrežju J1939. Takoj po registraciji naprave v omrežje, naprava pošlje sporočilo v omrežje, kjer naprave preverjajo enakost naslovov, če katera od naprav prepozna enakost, se potrjevanje naslova nadaljuje z arbitražnim postopkom. Naslov obdrži naprava, katere ime ima nižjo vrednost, druga naprava pa prek dinamičnega naslavljanja spremeni naslov in ponovi postopek [2].

Drugi način pa je preko pošiljanja sporočila s prošnjo po potrditvi. Tukaj novo priklopljena naprava pošlje prošnjo po potrditvi vsem preostalim napravam v omrežju in čaka odgovore ter jih primerja s svojim naslovom. Če pride do situacije, da imata dve napravi enak naslov, se postopek nadaljuje v arbitražo. Ta postopek se uporablja predvsem za komunikatorje ali druge servisne naprave, s katerimi se preverja in nadzoruje omrežje. Šele po končanem zagonu in potrditvi naslova prične novo priklopljena naprava komunicirati z omrežjem. Sam proces in časovni intervali potrjevanja so prikazani na sliki 4.18 [2].

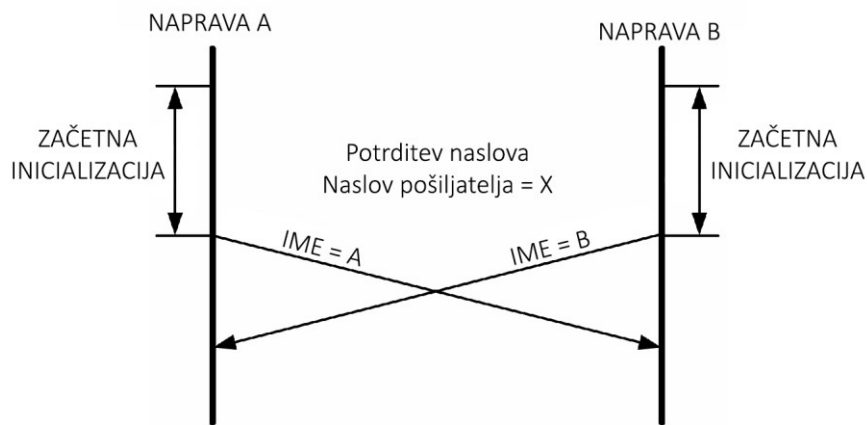




Slika 4.18: Proces potrjevanja naslova [2].

Na sliki 4.18 sta prikazana oba načina potrjevanja naslova. Dve napravi A in B potrjujeta isti naslov. Naprava A ima ime z višjo prioriteto. Naprava B v primeru na levi strani pa nima možnosti dinamičnega spreminjanja naslovov, treba je posredovanje fizičnega operaterja, da prek uporabniškega vmesnika ali DIP stikal spremeni naslov. V desnem primeru pa ima naprava B možnost dinamičnega spreminjanja naslovov, kar se vidi tudi na sliki, saj na koncu prevzame drug, Y, naslov. Pri obeh primerih je postopek enak. Naprava A začne z začetno inicializacijo nekaj trenutkov pred napravo B. Medtem ko naprava B zaključuje začetno proceduro, pa naprava A že pošlje zahtevo po potrjevanju naslova. Napetost B ugotovi, da ima isti naslov kot naprava B, zato pristopi k procesu potrjevanja. Po pristopu naprave B k potrjevanju, naprava A ugotovi, da ima ime višje prioritete kot naprava B, zato naprava A obdrži osnovni naslov. Naprava B prejme odgovor, da je ime naprave A višje prioritete, zato naprava B ne sme obdržati trenutnega naslova. Razlika je samo v zadnjem koraku. V levem primeru naprava B pošlje sporočilo nezmožnosti potrditve naslova. V desnem primeru pa naprava B poviša vrednost naslova za 1 in pošlje sporočilo o potrjevanju naslova ter postopek se ponovi od začetka, vendar ker napravi nimata več istih naslovov, bo postopek zaključen po prvem koraku [2].

V prejšnjem primeru napravi ne pošljeta sporočila o potrjevanju sočasno, kar lahko privede do težave. Sicer pa ne obstaja velika verjetnost, da bi dve napravi poslali zahtevo po potrjevanju naslova v istem trenutku. Primer pošiljanja sporočila v istem trenutku prikazuje slika 4.19.



Slika 4.19: Pošiljanje sporočila v istem trenutku [2].

V tem primeru obe napravi začneta oddajati v istem trenutku. Ker CAN ne dopušča takega ravnanja, bo to povzročilo napako. Napravi bosta zaznali napako in bosta ponovno poskušali s pošiljanjem sporočila za potrjevanje naslova, to bosta počeli toliko časa, dokler se ne bo generirala napaka na vsaki napravi posebej. Temu rečemo pasivno stanje napake. Tik preden bosta napravi preklopili stanje, bo omrežje obremenjeno do vrhnje zmožnosti prenosa, saj se naprava, takoj ko prejme povratek za napako, loti pošiljanja novega sporočila, še preden odpošlje prejšnje. V tem primeru pa pridemo v situacijo, ko to počneta dve napravi, tako da močno zapolnimo samo vodilo. To bo trajalo nekaj časa, preden napravi spremenita stanje obratovanja. Iz aktivnega se spremenita v stanje napake, po določenem času pa zopet preideta v aktivno stanje in se poskušata inicializirati, ter nato potrditi naslov. Verjetnost, da bi se postopek pošiljanja v istem trenutku ponavljal, je minimalna oziroma z vsakim novim krogom se zmanjšuje [2].

SAE J1939/81 priporoča upravljanje s trki po potrjevanju naslova, ta proces preverja pojave napak zaradi trkov sočasno poslanih sporočil. Ta proces povzroči dodatno umetno zakasnitev, če spozna, da je prišlo do napake zaradi sočasno poslanega sporočila.

Sporočila upravljanja naslovov imajo popolnoma enake lastnosti kot vsa ostala.

Tabela 11: Sporočila upravljanja naslovov [2].

<b>Sporočilo</b>	<b>PGN</b>	<b>PDU f</b>	<b>PDU s</b>	<b>Naslov pošiljatelja</b>	<b>Dolžina podatkovnega okvirja</b>	<b>Podatkovni okvir</b>
prošnja po potrjevanju naslova	59904	234	ciljni naslov	naslov pošiljatelja	3 bajti	PGN 60928
naslov potrjen	60928	238	255	naslov pošiljatelja	8 bajti	ime
nezmožnost potrditve naslova	60928	238	255	254	8 bajti	ime
potrjevanje na zahtevo	65240	254	216	naslov pošiljatelja	protokol prenosa, da prenesemo 8 bajtov	ime in nov naslov pošiljatelja

Prošnja po potrjevanju naslova je enaka navadni zahtevi po sporočilu, ki je opisano v poglavju o zgradbi komunikacijskega protokola. Ta vrsta sporočila je uporabljena za pošiljanje prošnje po potrjevanju sporočila. Sporočilo je lahko namenjeno točno določeni napravi, v tem primeru govorimo o komunikaciji vsak z vsakim, ali pa je namenjeno celotnemu omrežju, torej vsem napravam v omrežju. Sporočilo se uporabi v primeru, ko se določena naprava inicializira na omrežje in želi izvedeti, ali lahko uporablja predpostavljeni naslov ali je ta naslov mogoče že zaseden. Odziv na sporočilo takega tipa je lahko različen. Odzvala se bo vsaka naprava z istim naslovom, s katerim se bo nato začel postopek arbitraže. Pri tem se bodo odzvale vse naprave, ki ne morejo potrditi naslova.

Ime skupine parametrov:	Prošnja po potrjevanju naslova
Številka skupine parametrov:	59904 (00EA00 <sub>hex</sub> )
Definicija:	Prošnja po sporočilu v omrežju
Hitrost prenosa:	Definira uporabnik, vendar ne več kot 2 do 3 krat na sekundo
Dolžina podatkovnega dela:	3 bajti (CAN DLC = 3)
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	0
PDUf:	234
PDU s:	Ciljni naslov
Prioriteta:	6
Opis podatkovnega dela:	Zahtevana številka skupine parametrov PGN 60928

Potrjen naslov in nezmožnost potrditve naslova imata enako zgradbo sporočila. Uporabljata se v primeru, ko naprava potrdi naslov, ali ko naprava ne more potrditi naslova, sta pa obe sporočili odgovora na prošnjo po potrjevanju [2].

Za sporočilo o potrditvi naslova obstajajo določena pravila:

- Sporočilo o potrditvi naslova mora biti vedno naslovljeno na splošni 255 naslov.
- Sporočilo o potrditvi naslova mora biti poslano takoj po inicializaciji naprave na omrežje, uporabi se lahko minimalna časovna zakasnitev 250 ms.
- Takoj ko prejme naprava potrditev naslova, lahko začne s splošnim delovanjem in komunikacijo, ki jo mora izvajati.
- Ko naprava prejme sporočilo o potrditvi naslova, mora najprej preveriti in primerjati prejeti naslov iz sporočila s svojim naslovom, če sta naslova enaka, mora primerjati še imeni. Če si lasti ime z višjo prioriteto (nižjo numerično vrednostjo imena), mora poslati sporočilo o potrditvi naslova z informacijami o naslovu in imenu. Če pa ima naprava ime z nižjo prioriteto (višjo numerično vrednostjo imena), pa pošlje sporočilo o nezmožnosti potrditve naslova. Potem mora sama spremeniti naslov ter začeti postopek potrditve.
- V primeru, da naprava izgubi naslov in je v stanju pošiljanja sporočila transportnega protokola, mora takoj prenehati s pošiljanjem. Sprejemnik transportnega protokola bo zaznal prekinitev zaradi prevelike časovne prekinitve.

Primer sporočila:

Ime skupine parametrov:	Prošnja po potrjevanju naslova
Številka skupine parametrov:	60928 (00EE00 <sub>hex</sub> )
Definicija:	Sporočilo o potrditvi naslova
Hitrost prenosa:	Predpiše proizvajalec
Dolžina podatkovnega dela:	8 bajtov (CAN DLC = 8)
Podaljšan podatkovni okvir:	0
Podatkovni okvir:	0
PDUf:	238
PDU s:	255
Prioriteta:	6
Opis podatkovnega dela:	Samo za najavo sporočila

Podrobno po bajtih:	Ime aplikacije krmilnika (CA)
Bajt 1:	Biti od 8 do 1: LSB identifikacijskega polja
Bajt 2:	Biti od 8 do 1: drugi bajt identifikacijskega polja
Bajt 3:	Biti od 8 do 6: LSB proizvajalčeve kode Biti od 5 do 1: MSB identifikacijskega polja
Bajt 4:	Biti od 8 do 1: MSB proizvajalčeve kode
Bajt 5:	Biti od 8 do 4: primer funkcije Biti od 3 do 1: primer ECU
Bajt 6:	Biti od 8 do 1: funkcija
Bajt 7:	Biti od 8 do 2: sistem vozila Bit 1: rezerviran
Bajt 8:	Bit 8: zmožnost arbitražnega procesa Biti od 7 do 5: skupina industrije Biti od 4 do 1: primer sistema vozila

Sporočilo nezmožnost potrditve naslova ima enako zgradbo kot sporočilo potrditve naslova. Razlika je le v tem, da sporočilo nezmožnosti potrditve uporablja ničelno naslovitev, to je naslov 254, kot je naslov vira. Za sporočilo o nezmožnosti potrditve naslova obstajajo določena pravila:

- Ko naprava nima možnosti arbitražnega preverjanja imena in bo zahteva po le-tem, bo poslala sporočilo o nezmožnosti potrditve naslova.
- Naprava, ki ima možnost arbitražnega preverjanja imena, bo poslala sporočilo o nezmožnosti potrditve naslova, ko ne bo imela nobenega prostega naslova.
- Če je sporočilo o nezmožnosti potrditve naslova odgovor na prošnjo po potrditvi, mora naprava zagotoviti dodatno časovno zakasnitev med 0 in 153 ms, preden pošlje omenjeni odgovor. To je eden izmed ukrepov preprečitve pošiljanja istočasnega sporočila.

Zadnje je poročilo potrditev na zahtevo. To sporočilo ni v uporabi ob dinamičnem naslavljanju ali normalni, samostojni komunikaciji v omrežju. Potrjevanje na zahtevo se uporablja kot razvojno orodje in ga uporablja gradnik (oseba, ki gradi omrežje) med ustvarjanjem omrežja. Ko neka naprava prejme potrditev na zahtevo, to se lahko zgodi med delovanjem, bo prevzela zahtevani naslov in takoj začela z rutino potrditve. Lahko pa se zgodi, da naprava zahteve ne bo sprejela in tudi ne bo poslala nobene. V tem primeru se je

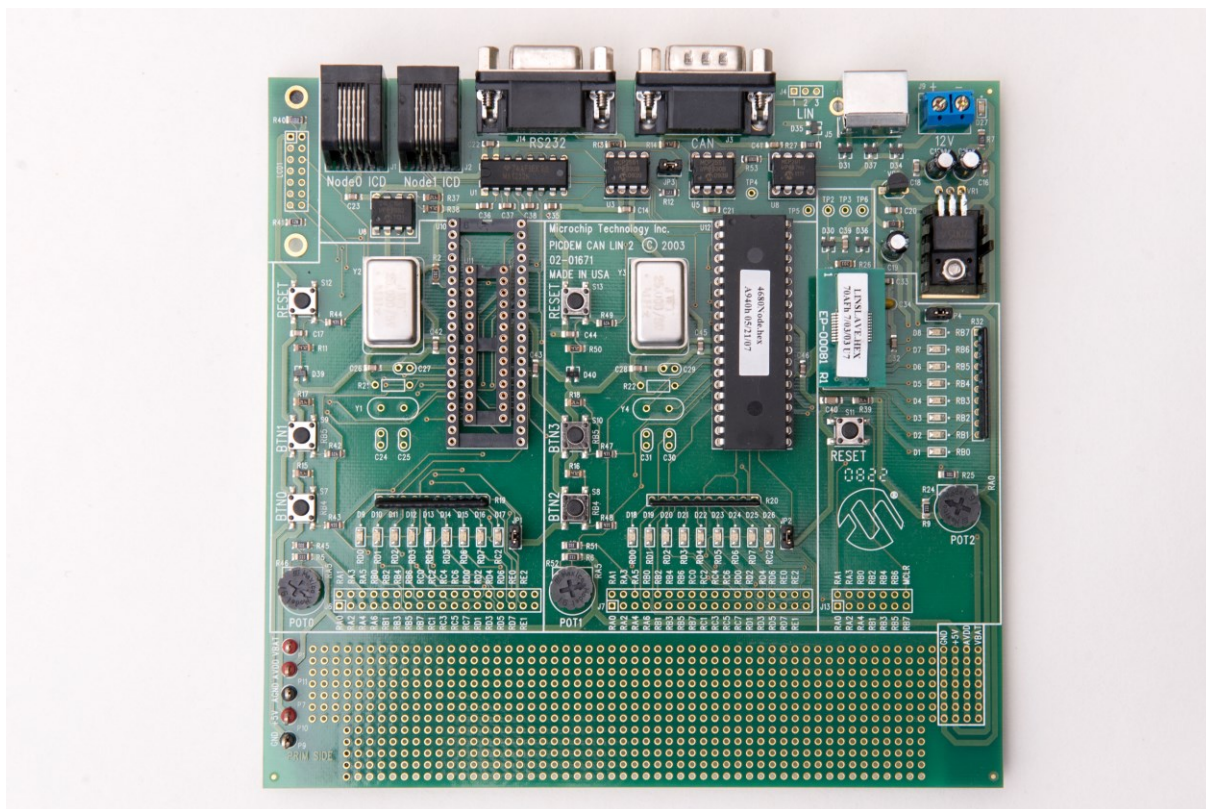
treba fizično priključiti na napravo in prek reprogramiranja spremeniti naslov. Ko napravo priključimo nazaj na omrežje, bo po zagonski inicializaciji začela z rutino potrjevanja naslova [2].

Primer sporočila:

Ime skupine parametrov:		Prošnja po potrjevanju naslova
Številka skupine parametrov:		60928 (00EE00 <sub>hex</sub> )
Definicija:		Sporočilo o potrditvi naslova
Hitrost prenosa:		Predpiše proizvajalec
Dolžina podatkovnega dela:		8 bajtov (CAN DLC = 8).
Podaljšan podatkovni okvir:		0
Podatkovni okvir:		0
PDUf:		238
PDU s:		255
Prioriteta		6
Podrobno po bajtih:		Potrditev na zahtevo
	Bajt 1:	Biti od 8 do 1: LSB identifikacijskega polja
	Bajt 2:	Biti od 8 do 1: drugi bajt identifikacijskega polja
	Bajt 3:	Biti od 8 do 6: LSB proizvajalčeve kode Biti od 5 do 1: MSB identifikacijskega polja
	Bajt 4:	Biti od 8 do 1: MSB proizvajalčeve kode
	Bajt 5:	Biti od 8 do 4: primer funkcije Biti od 3 do 1: primer ECU
	Bajt 6:	Biti od 8 do 1: funkcija
	Bajt 7:	Biti od 8 do 2: sistem vozila. Bit 1: rezerviran
	Bajt 8:	Bit 8: zmožnost arbitražnega procesa Biti od 7 do 5: skupina industrije Biti od 4 do 1: primer sistema vozila
Podrobno po bajtih:		Potrditev na zahtevo
	Bajt 9:	Biti od 8 do 1: nov naslov vira

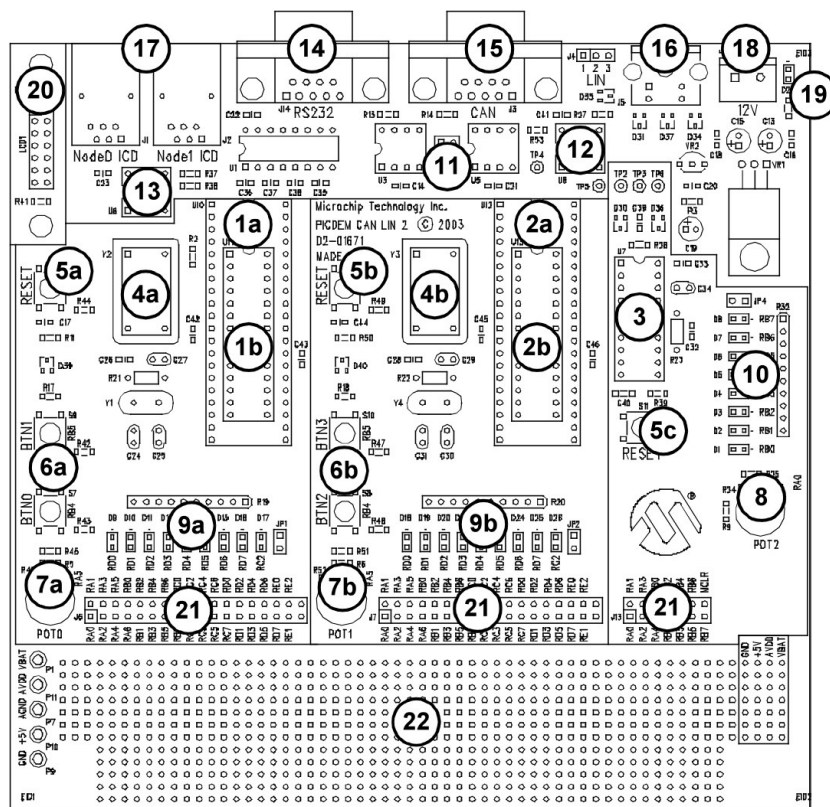
## 5. DEMONSTRACIJSKA PLOŠČA PICDEM CAN-LIN 2

Pri prikazu delovanja CAN bus komunikacije si bom pomagal z demonstracijsko ploščo PICDEM CAN-LIN 2, ki podpira dve vrsti komunikacije. Na njej lahko realiziramo tako LIN komunikacijo kot CAN bus. Slika 5.1 prikazuje CAN-LIN demonstracijsko ploščo.



Slika 5.1: CAN-LIN demonstracijska plošča.

Ostali vmesniki oziroma demonstracijske plošče imajo samo en mikrokrmilnik, zato potrebujemo več plošč, da lahko sestavimo omrežje. Omenjena demonstracijska plošča pa vsebuje tri naprave. Naprava 0 vsebuje 40 pinski mikrokrmilnik PIC 18F4680. Ta naprava ima tudi povezavo na RS 232. Naprava 1 vsebuje isti mikrokrmilnik kot naprava 0. Naprava 1 ima povezavo neposredno na DB9 konektor, kjer je dostop do CAN vodila. Naprava 1 je tudi master LIN komunikacije, saj ima tudi povezavo do naprave 2. Naprava 2 je slave LIN komunikacije. Napravi 0 in 1 sta povezani preko CAN omrežja, ki je integrirano na demonstracijsko ploščo. Slika 5.2 podrobneje prikazuje ploščo in označuje posamezne dele.



Slika 5.2: PICDEM CAN-LIN demonstracijska plošča [8].

Podrobneje po točkah:

1. Naprava 0, ima pripravljen sedež za 40 in 28 pinske mikrokrmilnike. Na mestu je integriran CAN modul. Naprava 0 je povezana s 14 na sliki, ki je konektor za RS 232 povezavo, ima pa tudi možnost priključitve zunanjega EPROMa ter zunanjega LCD zaslona.
2. Naprava 1 ima pripravljen sedež za 40 in 28 pinske mikrokrmilnike. Na mestu je integriran CAN modul. Naprava 0 je povezana s 15 na sliki, ki je konektor za zunanjo CAN komunikacijo prek DB 9 konektorja. Ima pa tudi LIN povezavo in na demonstracijski plošči služi kot master v LIN omrežju.
3. Naprava 2 ima pripravljen sedež za 20 pinski mikrokrmilnik. Na mestu je integriran LIN oddajnik. Naprava 2 je podrejena v LIN omrežju.
4. Oscilatorji; napravi 0 in 1 podpirata različne vrste oscilatorjev: kristal, RC ... plošča je opremljena s 25 MHz oscilatorjem, za kakšne druge zahteve pa je treba spremeniti demonstracijsko ploščo.
5. Reset tipka; vsak mikrokrmilnik ima svojo reset tipko. Tipka je vezana na MCLR priključek.



6. Proste tipke; napravi 0 in 1 imata dve prosti tipki, ki jih lahko nastavlja vsak uporabnik posebej. Tipki sta vezani na CAN naprave in simulirata digitalni vhod.
7. Potenciometra; napravi 0 in 1 imata vezana potenciometra, s katerima lahko simuliramo analogne vhode.
8. Potenciometer naprave 2; pod označbo 8 se nahaja potenciometer naprave 2, s katerim simuliramo analogni vhod.
9. Led diode; pod zaporedno označbo 9 se nahaja 9 led diod. 8 diod kaže status izhoda D (napravi 0 in 1), PWM kontrolirane LED diode pa kažejo stanje potenciometra.
10. LED diode naprave 2; pod zaporedno oznako 10 so diode naprave 2. 8 diod je vezanih na izhod B naprave 2.
11. CAN oddajnik; MCP2551 CAN oddajnik spremeni signal iz CAN omrežja v digitalni signal, ki je poznan mikrokrmilniku.
12. LIN oddajnik; MCP201 spremeni visokonapetostni signal iz LIN omrežja v signal, ki je znan mikrokrmilniku.
13. Zunanji EPROM; EPROM serije 24LC16 je vključen v shemo plošče za razvijanje najrazličnejših sistemov. Je 16 kbit naprava, ki je preprogrameljiva z I2C vmesnikom. EPROM je povezan z napravo 0.
14. RS 232; na zgornji strani plošče je DB 9 konektor, preko katerega se lahko povežemo na RS 232 komunikacijo. Ta konektor lahko uporabljamo tudi kot serijsko povezovanje v aplikacijah.
15. CAN konektor; Pod zaporedno številko 15 je DB 9 konektor, s katerim se lahko priključimo na CAN omrežje demonstracijske plošče.
16. LIN konektor; 3-pinski konektor za povezavo na LIN omrežje demonstracijske plošče.
17. ICD konektor; RJ 11 konektor za povezavo programatorja na mikrokrmilnik.
18. Konektor za priključitev napajanja; napajanje je 12 V enosmerne napetosti.
19. Indikator napajanja; pod zaporedno številko 19 se nahaja LED dioda rdeče barve, ki je pokazatelj, ali je napajanje priključeno.
20. Konektor za LCD; v levem zgornjem vogalu je mesto, kamor lahko priključimo zunanji LCD zaslon.
21. Prosto mesto konektorjev; pod zaporedno številko 21 je mesto, kamor lahko namestimo konektorje, ki jih potrebujemo v fazi razvoja.
22. Prosto mesto; pod zaporedno številko 22 je mesto, kamor lahko namestimo komponente, ki jih potrebujemo v fazi razvoja, v neposredni bližini je tudi napajanje.

Zraven demonstracijske plošče pa prejmemo tudi program, omejeno različico CANKing, ki ga uporabljamo za nadziranje CAN omrežja. Program bom natančneje opisal v naslednjem poglavju diplomskega dela [8].

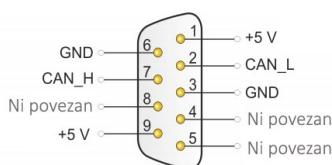
## 6. PRIMERJAVA KOMUNIKATORJEV KVASER LEAF RIGHT IN PEAK SYSTEM PCAN-USB

Prva oprema, s katero sem se srečal ob reševanju in vzpostavitvi CAN bus komunikacije, je bila oprema Peak system. Komunikator PCAN-USB je komunikacijski pripomoček za nadzor omrežja, kot je prikazan na sliki 6.1.



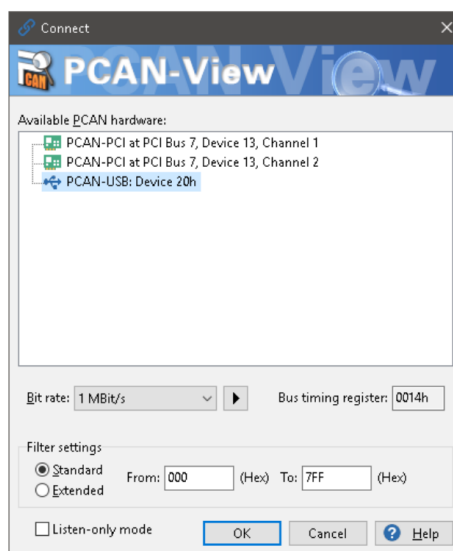
Slika 6.1: PCAN-USB komunikator [12].

S komunikatorjem brez galvanske ločitve moramo biti pazljivi, saj se po navadi v fazi razvijanja uporabljajo nedokončana tiskana vezja. Ob neprevidnosti lahko pride do kratkega stika, ki poškoduje računalnik. Adapter podpira USB verzije 1.1, 2.0 in 3.0. Na CAN strani ima DB 9 (moški) konektor za preprosti priklop na CAN omrežje. Podpira hitrosti od 5 kbit/s do 1 Mbit/s. Podpira CAN komunikacijo z 11 bitnim in z 29 bitnim identifikacijskim številom. Uporablja CAN oddajnik NXP PCA82C251. Napaja se prek USB priključka iz računalnika, omogoča pa tudi napajanje na CAN strani. Slika 6.2 prikazuje razporeditev kontaktov v konektorju na CAN strani [12].



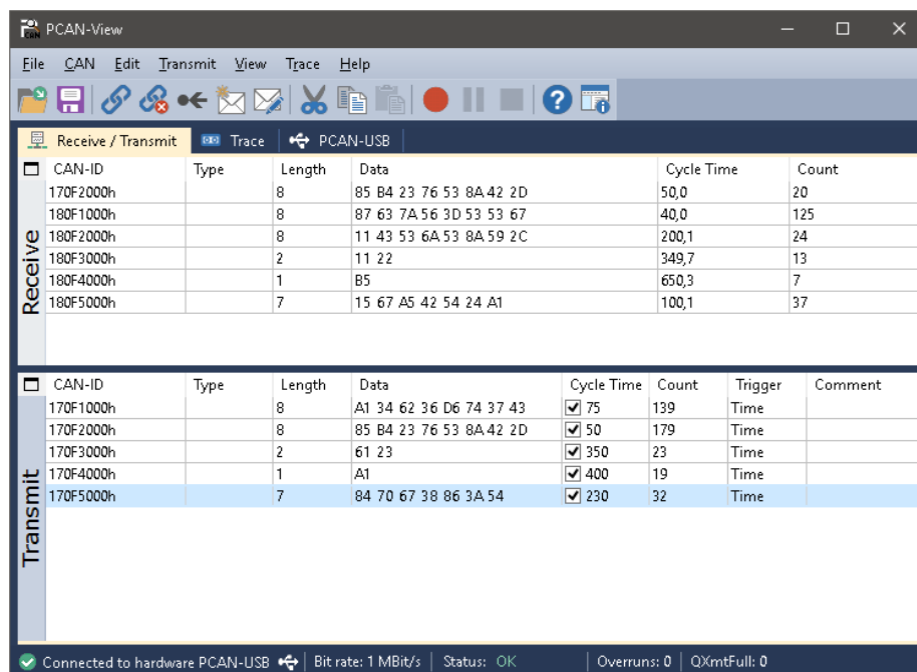
Slika 6.2: Razporeditev kontaktov v konektorju komunikatorja na CAN strain [12].

PCAN komunikator je kompatibilen s programsko opremo PCAN View. Programska oprema je brezplačna, vendar ne omogoča veliko nastavitvev, z opremo lahko nadziramo omrežje, pošiljamo in prejemamo sporočila. Podpira 11 bitni in 29 bitni ID. Slika 6.3 prikazuje okno, kjer komunikator povežemo z osebnim računalnikom [12].



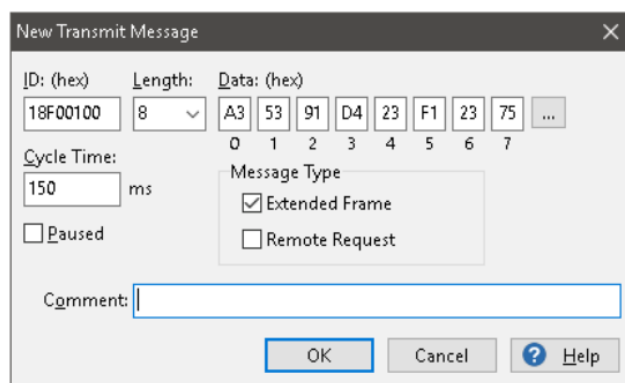
Slika 6.3: Okno za povezavo komunikatorja z osebnim računalnikom.

Tukaj izberemo hitrost povezave in nastavimo tip CAN komunikacije, za komuniciranje v omrežju J1939 izberemo Extended in nastavimo filter za sporočila. Slika 6.4 prikazuje glavno okno programa.



Slika 6.4: Glavno okno programa PCAN-View.

Glavno okno je razdeljeno na dva dela, in sicer na oddajnega in sprejemnega, kjer spremljamo promet na vodilu. Tukaj nadziramo sporočila, ki jih pošiljamo. Novo sporočilo ustvarimo preko vmesnika, ki je prikazan na sliki 6.5.



Slika 6.5: Vmesnik za sestavljanje novega sporočila.

Slika 6.4 prikazuje tudi del, kjer nadziramo sporočila, ki jih prejemamo. Preko filtrov nastavimo, katere vrednosti ID-jev naj prepušča, le-te se nam izpisujejo na zaslonu. Edina slaba lastnost je ta, da podatkovnega dela ne moremo dodatno razdeliti in definirati, kaj določena informacija pomeni. Izpisujejo se nam šestnajstiške vrednosti po bajtih. Slika 6.5 prikazuje vmesnik za sestavljanje sporočila, ki je tudi pregleden in uporabniku prijazen, saj so vsa pomembna polja jasno vidna in lahko nastavljiva.

Naslednja oprema je bila od proizvajalca Kvaser. Komunikator Kvaser Leaf Light je prikazan na sliki 6.6.

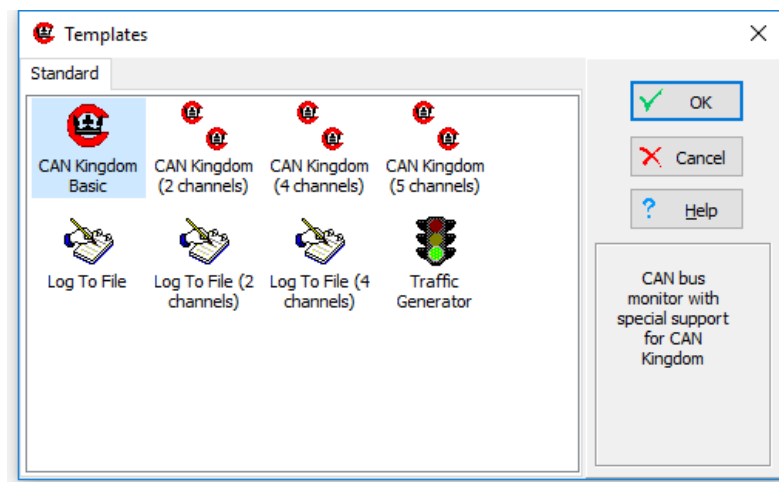


Slika 6.6: Kvaser Leaf Light CAN komunikator [11].

Izdelki Kvaser so višjega cenovnega razreda in so namenjeni profesionalni uporabi. Kvaser ponuja na trgu veliko različnih komunikatorjev od osnovnega, kar je Leaf Light, pa do Leaf Professional, ki podpira CAN komunikacijo višje hitrosti, CAN komunikacijo nižje hitrosti, SWC komunikacijo (to je enožična CAN komunikacija) in LIN komunikacijo. Naprava ima na ohišju vmesnika 2 diodi, ki sta indikatorja delovanja, zelena je napajanje, rumena pa

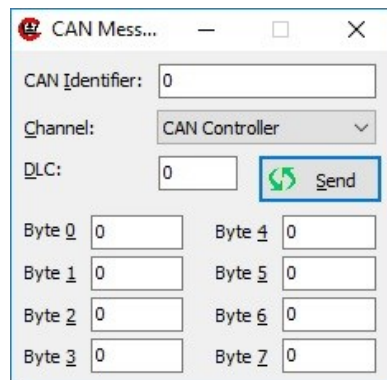
sprejemno/oddajna in utripa ob sprejemanju in oddajanju. Druge lastnosti so podobne kot pri Pcanu.

Kvaser ponuja programsko opremo CANKing. Slika 6.7 prikazuje izbirno okno CANKinga, kjer izbiramo med želenimi predhodno nastavljenimi predlogami.

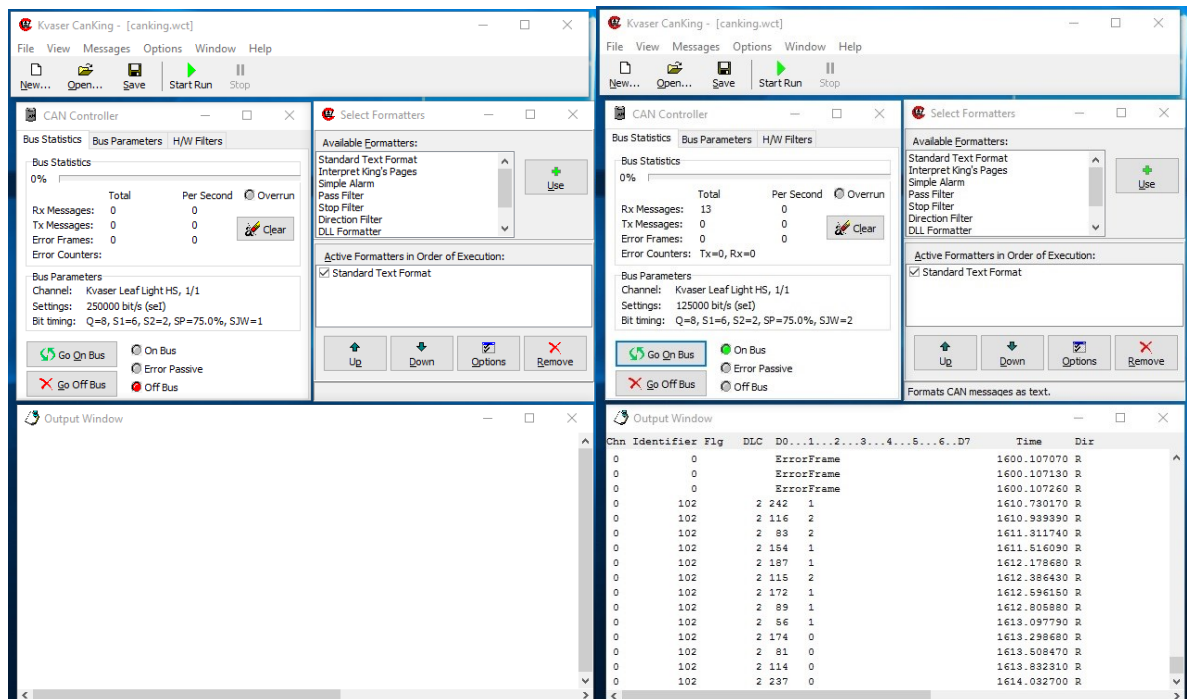


Slika 6.7: Izbirno okno, kjer izbiramo med predlogami.

Can Kingdom Basic je osnovna komunikacija za nadzor prometa na vodilu. Sporočila z omrežja se ne izpisujejo v arhiv. Nastavljamo lahko parametre omrežja, kot so hitrost, mesto vzorčenja ter filtre za oddajo in sprejem sporočil. CAN Kingdom (2 channels) je isti program kot prejšnji, le da tukaj nadzorujemo 2 CAN vodili. Ista sta naslednja dva, le da tukaj nadziramo 4 oziroma 5 CAN vodil. Trafic generator generira promet na vodilu. Za nadzor na vodilu izberemo CAN Kingdom Basic, kjer se nam odpre okno, kot je prikazano na sliki 6.9. Prvi del slike 6.9 kaže prazno okno. Ko nastavimo vse potrebne parametre in se povežemo na vodilo, vmesnik izgleda tako, kot prikazuje desni del slike 6.9. Program je malo manj uporabniku prijazen, kot prej omenjeni PCan View, vendar pa nudi veliko več možnosti nadzora omrežja. Slika 6.8 prikazuje vmesnik za pošiljanje sporočil. V CANKingu lahko izbiramo med velikim naborom predlog, ki nam zelo olajšajo delo generiranja sporočila. Za delo s programom uporabnik potrebuje malo več časa. Ko ga enkrat osvoji, pa mu delo močno olajša.



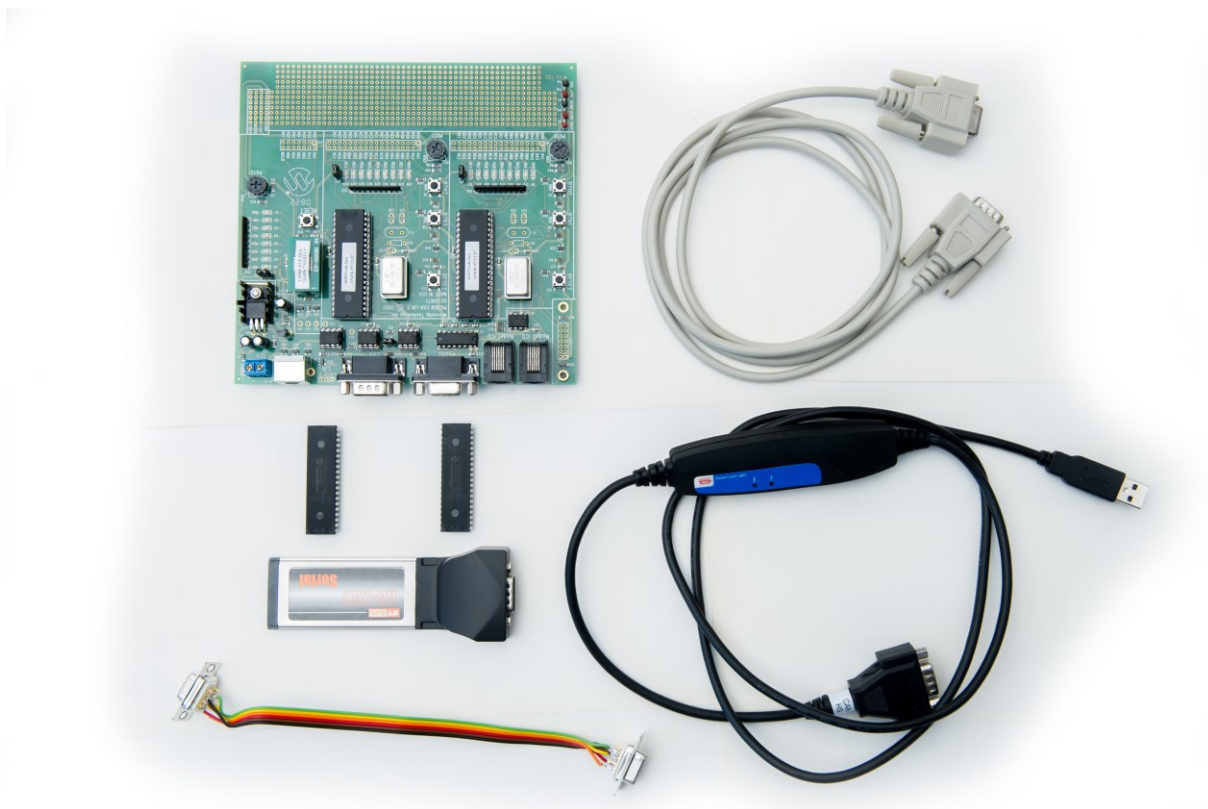
Slika 6.8: Vmesnik za generiranje sporočil.



Slika 6.9: Slika prikazuje okolje CANKing vmesnika.

## 7. REALIZACIJA KUMUNIKACIJE S POMOČJO DEMONSTRACIJSKE PLOŠČE PICDEM CAN-LIN 2

Realizacija CAN bus komunikacije je potekala v dveh delih. Prvi del je bil spoznavanje z demonstracijsko ploščo in programsko opremo, ki jo podpira. V drugem delu pa je bilo programiranje mikrokrmilnikov. Slika 7.1 prikazuje opremo in povezave za pregled delovanja prednastavljenih programov. Zraven demonstracijske plošče spada CANKing, s pomočjo katerega lahko nadzorujemo in nastavljamo vodilo CAN.



Slika 7.1: Oprema za nadzor CAN vodila.

Uporabljena strojna oprema je bila:

- demonstracijska plošča PICDEM CAN-LIN 2,
- dva mikrokrmilnika 18F4680,
- Express card RS-232 (kartica, ki poveže port RS-232 direktno na port računalnika),
- dodatno ustvarjen podaljšek konektorja DB 9 (ženski – ženski),
- CAN komunikator Kvaser Leaf Light,
- serijski kabel.



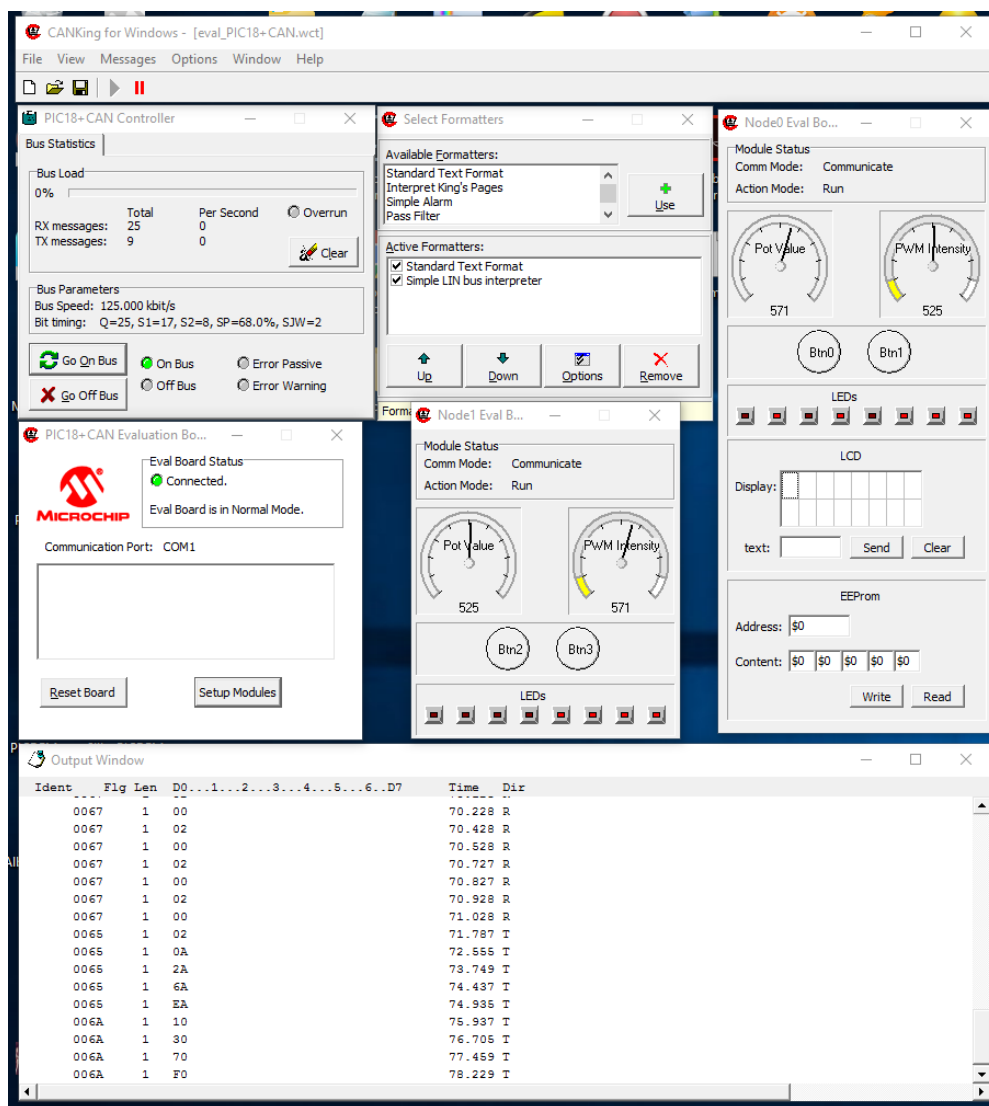
Pri zagonu sistema sem imel največ težav pri vzpostavitvi zunanjih serijskih vrat na osebem računalniku. Preden sem prišel do delujoče rešitve, sem preizkusil veliko USB – RS-232 pretvornikov. Največja težava je bila, ker ob priklopu nisem pomislil, da je težava v samem pretvorniku in stabilnosti njegovega delovanja, prej sem namreč iskal težave v nepoznavanju sistema. Omenjena celotna strojna oprema deluje brez težav in je dokaj enostavna za uporabo.

Sam postopek postavitve sistema je naslednji: na računalnik namestiš programsko opremo CANKing, ki je priložena demonstracijski plošči, na računalniku je treba še zagotoviti serijsko povezavo in to je, kar se tiče programske opreme, vse. Demonstracijsko ploščo najprej povežemo s serijsko komunikacijo, nato pa priključimo še napajanje. Ob priklopu napajanja preverimo, ali LED indikator (prisotnosti napajanja), sveti. Še prej preverimo, ali so mikrokrmilniki dobro na svojem mestu, da kasneje ne bi prišlo do težav. Ko je priključeno vse potrebno, zaženemo CANKing na osebem računalniku in spremljanje komunikacije steče. Sam sem dodatno, za spremljanje komunikacije, uporabil še CAN komunikator. Najprej sem poskušal na enem osebem računalniku izvajati oba programa. CANKing je nadgrajen izključno za delo z demonstracijsko ploščo PICDEM, Kvaser Leaf Light pa ravno tako podpira CANKing. Osnovna različica, ki je brezplačna programska oprema, ki jo Kvaser ponuja za uporabo s svojimi izdelki. Kljub temu mi ni uspelo zagnati obeh programov na enem računalniku.



Slika 7.2: Prikaz povezave demonstracijske plošče na osebni računalnik.

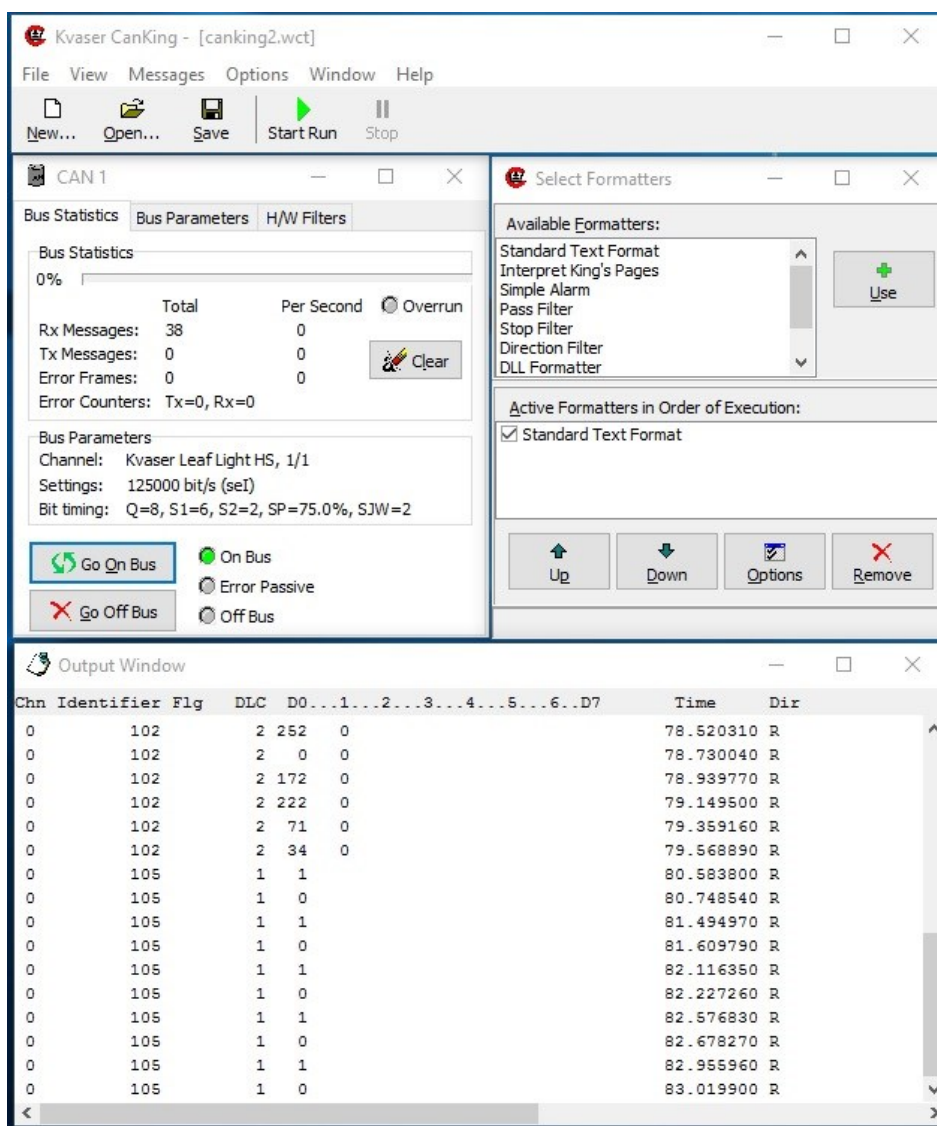
Na sliki 7.2 vidimo izvedeno povezavo na osebni računalnik. Na demonstracijsko ploščo je priključeno napajanje, in sicer Rs-232 povezava z računalnikom, kjer obratuje CANKing, izdelan posebej za nadzor in upravljanje z demonstracijsko ploščo. Preko CAN komunikatorja je drugi računalnik povezan na CAN vodilo. Slika 7.3 prikazuje programsko okolje za upravljanje z obstoječo komunikacijo. Zgornji okvir je glavna programska vrstica, kjer imamo dostop do vseh nastavitev. Če hočemo sami generirati sporočilo, dostopamo do vmesnika za generiranje preko glavne programske vrstice, in sicer se generator skriva pod zavihkom Message → Universal, kjer izberemo universal za generiranje splošnega sporočila.



Slika 7.3: Programsko okolje za nadzor komunikacije.

PIC 18+CAN Controler je okno, ki prikazuje vodilo, zasedenost vodila ter števca uspešno poslanih sporočil in napak. PIC 18+CAN Evaluation Bord, je glavni upravljevec povezave. S pritiskom na Setup Modules, se povežemo z mikrokrmilnikom prek CAN bus povezave. V oknu Output Window spremljamo vsa prenesena sporočila po vodilu. V oknu »Select

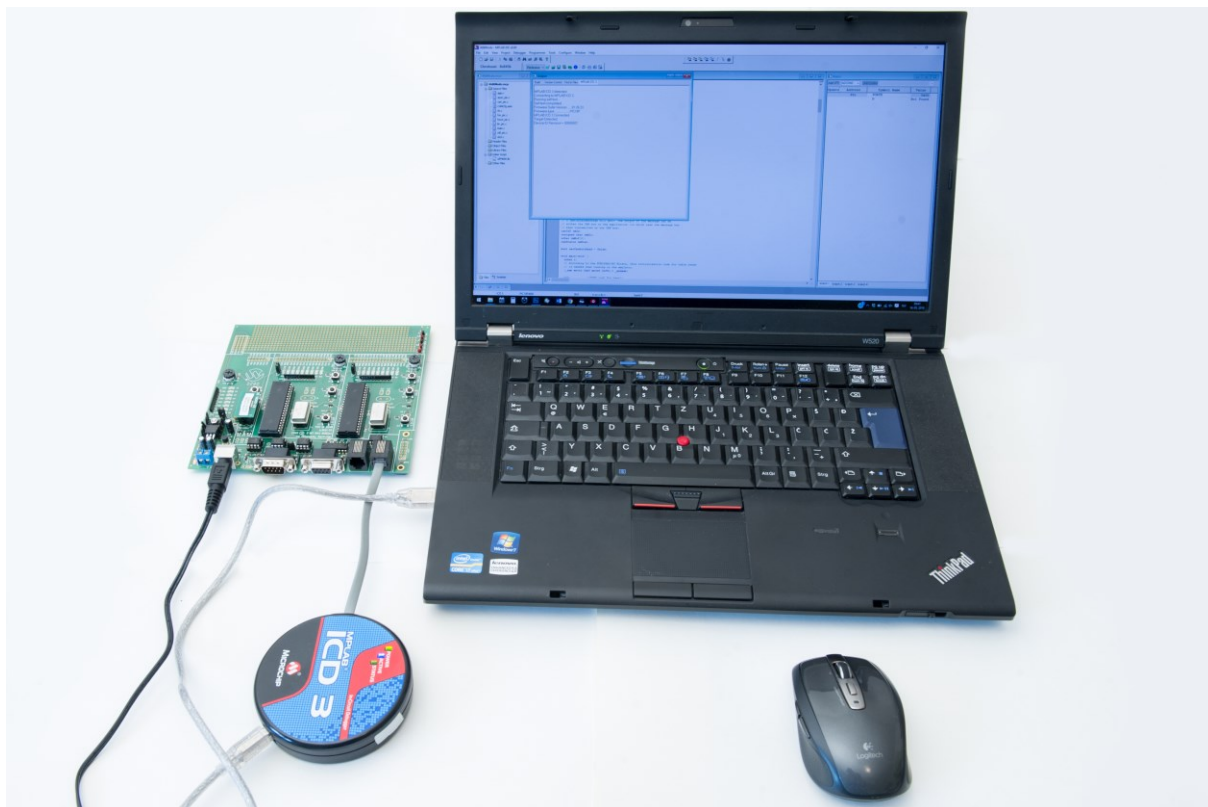
formatters« nastavljamo format sporočil, ki jih želimo poslati oziroma, ki jih sprejemamo na vodilu. Nato sta še dve okni, in sicer uporabniška vmesnika, ki predstavljata stanje na mikrokrmilniku 1 (naprava 0) in mikrokrmilniku 2 (naprava 1). Slika 7.4 prikazuje pregled vodila preko CAN komunikatorja.



Slika 7.4: Pregled CAN vodila preko CAN komunikatorja.

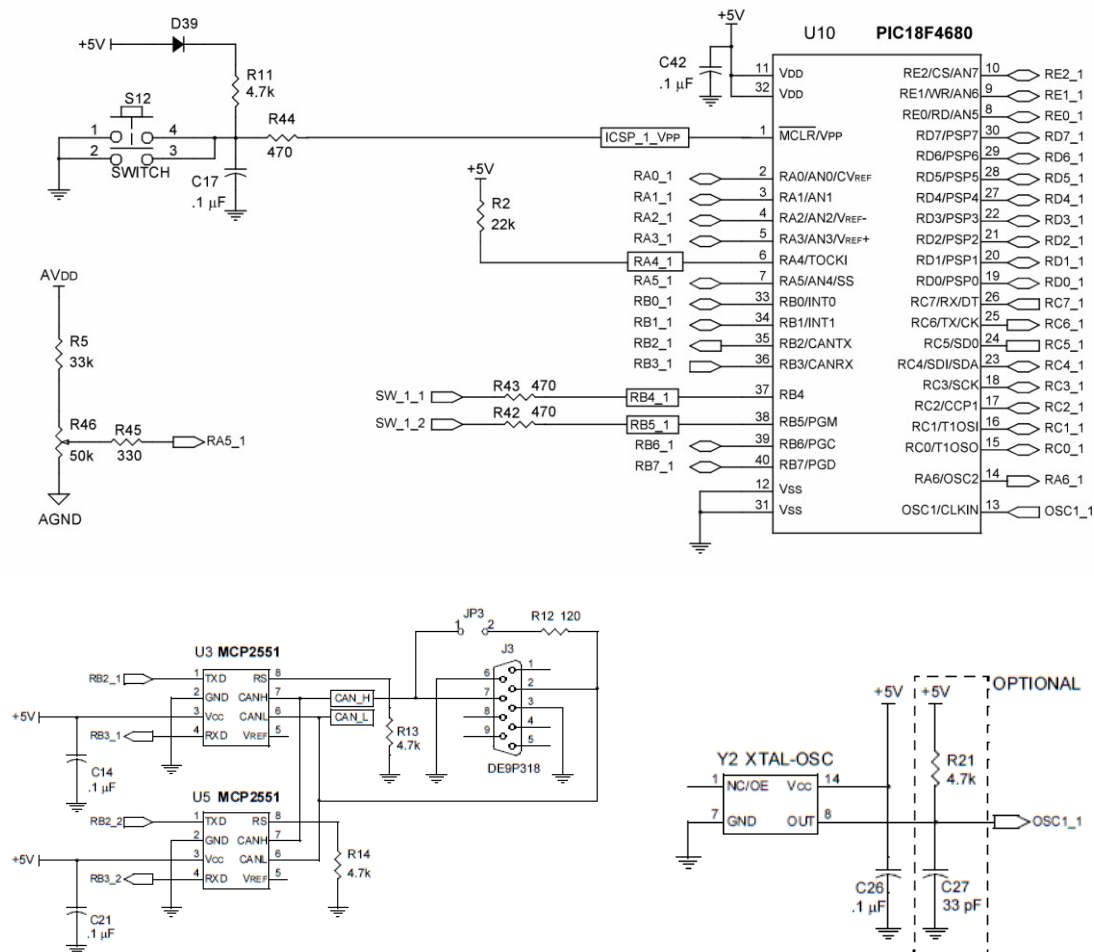
Drugi del uporabe PICDEM je bil sprogramirati mikrokrmilnik, da bo deloval v CAN omrežju. Najprej je bilo treba namestiti programsko opremo, ki podpira razvoj in programiranje PIC mikrokrmilnikov. Odločil sem se za MPLAB IDE verzijo 8.60. Za nadgradnjo tudi ni zadostovala strojna oprema, saj PICDEM ne omogoča programiranja krmilnikov. Za programiranje sem se odločil, da vzamem programator MPLAB ICD 3, ki je splošni PIC programator in podpira vse družine PIC mikrokrmilnikov, omogoča razhroščevanje in omogoča napajanje preko programatorja. Slika 7.5 prikazuje uporabljeno strojno opremo pri programiranju novih mikrokrmilnikov. Sam potek je bil naslednji: na

demonstracijski plošči sem zamenjal mikrokrmilnik, priključil programator, vključil napajanje, na računalniku zagnal programsko opremo MPLAB IDE. S čarovnikom sem ustvaril nov projekt in se lotil inicializacije mikrokrmilnika.

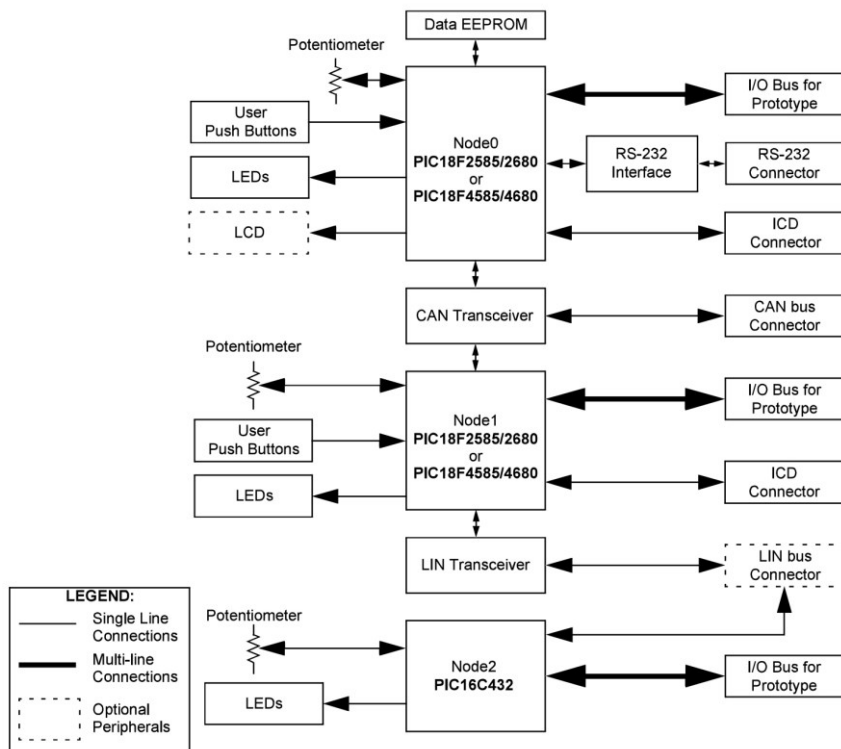


Slika 7.5: Prikaz strojne opremo, potrebne za programiranje mikrokrmilnikov.

Ker je demonstracijska plošča postavljena in ne omogoča spreminjanja, je bilo treba najprej pregledati diagram, kako je plošča sestavljena. Diagram je prikazan na sliki 7.7. Na sliki 7.6 pa je prikazana vezalna shema demonstracijske plošče.



Slika 7.6: Slika prikazuje vezalno shemo demonstracijske plošče [8].



Slika 7.7: Prikaz blokovne zgradbe demonstracijske plošče [8].

Pomembno je poznavanje arhitekture izdelka, saj le tako lahko spoznamo in izdelamo program, ki bo v skladu s strojno opremo. Če ne upoštevamo kompatibilnosti in omejitev strojne opreme, lahko hitro izgubimo rdečo nit, največja težava pa bo, ker izdelek ne bo deloval pravilno ali pa ne bo dosegal svoje preciznosti. Pri komunikacijskih vmesnikih je visokega pomena takt. Če zaradi konstrukcijske napake, kristal pomaknemo daleč stran ali pa na območje, ki je močno obremenjeno z najrazličnejšimi motnjami, lahko povzročimo, da bo naprava delovala slabše, kot če bi se zavedali vseh pomembnih lastnosti. Če pa gledamo skozi perspektivo strojev, v strojogradnji, pa morata konstruktor mehanike in elektro projektant sodelovati z roko v roki, saj le tako pridemo do kvalitetnega izdelka.

Programerski del naloge je potekal na naslednji način; najprej je bilo treba nov mikrokrmilnik nastaviti, to sem storil preko kontrolnih bitov. Pri delu z mikrokrmilniki, nam delo olajšajo uporabniška navodila. V njih so podrobno opisani vsi registri in lastnosti delovanja mikrokrmilnika. Pri nalogi se z napajanjem, nivoji napetosti in tokov, nivoji signalov nisem ukvarjal, saj je bil mikrokrmilnik pritrjen na demonstracijsko ploščo, tako da je bila moja naloga krmilnik pravilno nastaviti, vklopiti ECAN modul, in nato pravilno implementirati knjižnico AN930 v kodo.

Pri nastavljanju moramo biti še posebej pazljivi, saj je za stabilno delovanje treba nastaviti pravilne registre, in sicer samo tiste, ki jih potrebujemo. Hkrati se moramo zavedati, da nam lahko povzročajo težave tisti, ki jih pustimo proste. Lahko se namreč zgodi, da preide mikrokrmilnik v stanje, ki nam ni posebej jasno, razlog pa je lahko kateri od registrov, ki smo ga napačno ali pa ga sploh nismo nastavili. Za nastavljanje registrov sem si pomagal z uporabniškimi navodili ter s pomočjo, ki je del razvojnega okolja MPLAB IDE. Nato sem nastavil vhode in izhode, Tukaj je bila potrebna največja pozornost, da se ne nastavljajo celotni registri vhodov, saj je na posameznem priključku digitalni vhod ali izhod ter CAN sprejemna ali oddajna povezava mikrokrmilnika.

Po uspešni inicializaciji krmilnika sem s testnim programom preizkusil vhode in izhode. Ko je vse stabilno delovalo, sem se lotil vklopa ECAN modula. ECAN modul mikrokrmilnika 18F4680 je zelo zmogljiva funkcija, saj omogoča:

- Hitrost prenosa sporočil do 1 Mbps.
- Podpira specifikacije CAN 2.0A in 2.0B, definirana v Boschevem standardu, podpira tudi CAN 1.2, Pasivna CAN 2.0A in 2.0 B.

- Je kompatibilen z vsemi moduli PIC družine 18 (18FXXX8).
- Podpira več načinov delovanja (sprotna obdelava, FIFO ...).
- Ima predpostavljene tri oddajne medpomnilnike za nemoteno delovanje.
- Ima predpostavljena dva sprejemna medpomnilnika.
- Šest spremenljivih sprejemno/oddajnih medpomnilnikov.
- Ima tri nastavljive 29 bitne maske.
- 16 polno nastavljivih (29 bitov) sprejemnih filtrov.
- Podpira DevideNet.
- Avtomatsko rokovanje z oddaljenimi okvirji.
- Napredna obdelava in upravljanje z napakami.

CAN modul je sestavljen iz gonilnika protokola in medpomnilnikov za sporočila. Avtomatsko upravlja vse funkcije, ki so povezane s sprejemanjem ali oddajanjem sporočila. Sam proces pa poteka na naslednji način; takoj ko modul zazna sporočilo na vodilu, sporočilo najprej pregleda, če so kakšne napake, nato ga primerja z nastavljenimi filtri. Preko filtrov nastavljamo prepustnost za sporočila, povedano z drugimi besedami, sporočilo, ki ga filtri prepustijo, je namenjeno napravi, tisto, ki ga pa filtri zadržijo na zunanji strani, pa ji ni namenjeno. Če filtri prepustijo sporočilo, pomeni, da je relevantno za napravo, zato ga modul razvrsti v enega od dveh sprejemnih medpomnilnikov. Ob dodelitvi v medpomnilnik se prožijo zastavice, da lahko program mikrokrmilnika zazna, da je prispelo sporočilo, ki ga je treba obdelati. Sproži se prekinitev, ki poskrbi, da se sporočilo obdeli, nato pa vrne v akcijo, kjer se je pred prekinitvijo ustavila.

ECAN modul podpira naslednje tipe sporočil:

- Običajne podatkovne okvirje z 11 bitnim ID-jem.
- Običajne podatkovne okvirje z 29 bitnim ID-jem.
- Sporočila z zahtevo (remote frame).
- Sporočila, ki sporočajo o napaki.
- Sporočila, ki sporočajo o zasičenosti (Overload Frame Reception).

CAN modul uporablja dva priključka RB2/CANTX in RB3/CANRX za povezavo s CAN omrežjem. Pri nastavljanju je pomembno, da uporabnik postavi register TRIS <3> na vrednost 1. Potek inicializacije poteka na naslednji način:

1. Zagotoviti je treba, da postavimo ECAN modul v način delovanja nastavljanja.
2. Izberemo Operacijo ECAN modula.
3. Nastavimo hitrost komunikacije.
4. Nastavimo filtre in registre.
5. Izklopimo način nastavljanja in preklopimo v normalen način delovanja.

Rutina nastavitve ECAN modula izgleda tako:

```
void initCAN (void) //Zanka rutine, ki jo zaženemo v glavnem programu.
```

```
{
    setCANMode(CAN_CONFIG_MODE); // Modul postavimo v stanje
nastavljivosti.
    BRGCON1=0x44; // Nastavimo bitno hitrost na 250 kbps.
    BRGCON2=0xBD;
    BRGCON3=0x04;
    RXM0SIDH=0; // Nastavimo sprejemne registre na 0.
    RXM0SIDL=0;
    RXM0EIDH=0;
    RXM0EIDL=0;
    RXM1SIDH=0;
    RXM1SIDL=0;
    RXM1EIDH=0;
    RXM1EIDL=0;
    RXB0CON=0; // Nastavimo kontrolne sprejemne registre.
    RXB1CON=0;
    RXF0SIDL=0; // Nastavimo medpomnilnik nič na vrednost
0, za branje 11
    RXF1SIDL=0; // bitnih identifikacijskih naslovov.
    RXF2SIDL=0x08; // Nastavimo medpomnilnik nič na vrednost
0, za branje 29
    RXF3SIDL=0x08; // bitnih identifikacijskih naslovov.
    RXF4SIDL=0x08;
    RXF5SIDL=0x08;
```



```

    TRISBbits.TRISB2=0; // CANTX - nastavitve vhodov in
    izhodov CAN modula.

    TRISBbits.TRISB3=1; // CANRX- nastavitve vhodov in
    izhodov CAN modula

    PIE3=0x03; // Vklop vhodnih prekinitev.
    IPR3=0x03; // Nastavitev prioritete.
    PIR3=0; // Postavljanje zastavic na 0.
    CIOCON=0x30; // Vklop zajema na CAN vodilu.

    setCANMode(CAN_NORMAL_MODE); // Modul postavimo v
    normalno stanje.
}

```

Po nastavljenem ECAN modulu je bilo treba v projekt uvoziti knjižnico AN930. Knjižnica podpira komunikacijo na višjenivojskem protokolu J1939. Dodatna knjižnica J1939 (AN930) je obširna knjižnica in odličen pripomoček za razvijanje CAN komunikacije. Zastavljena je v takšni smeri, da celotno manipulacijo sporočila opravi algoritem sam, zato se razvojniki ni treba posebej ukvarjati z dodatnimi funkcijami. Knjižnica J1939 je spisana v programskem jeziku C in opravi večino dela, ki je potrebno za nemoteno delovanje protokola.

Knjižnica podpira vse naslove, imena in možnosti naslavljanja, ki sem jih že omenil v nalogi. Zgradba sporočila je zasnovana po osnovni predpisani zgradbi sporočila, ki ga predpisuje standard. Natančno grajenje, predvsem kako se štejejo posamezni biti pa prikazuje slika 7.8.

S O F	Identifier 11 Bits											S R R	I D E	Identifier Extension 18 Bits																		R T R		
S O F	Priority			R	D P	PDU Format (PF) 6 Bits (MSB)						S R R	I D E	PF (cont.)		PDU Specific (PS) (Destination Address, Group Extension or Proprietary)										Source Address								R T R
	3	2	1			8	7	6	5	4	3			2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	23	25	26	27	28	29	30	31	32	33		
	28	27	26	25	24	23	22	21	20	19	18			17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			

Slika 7.8: Format sporočila J1939 [7].

Knjižnica J1939 je namenjena uporabi z družino mikrokontrolerov PIC 18 in razvojno programsko opremo MPLAB C18. Knjižnica pokriva tako rutine inicializacije kot rutine, ki so nujne za nemoteno delovanje omrežja. Sporočilo, ki se generira nekje v kodi, je treba kopirati vrsto, od kjer se ga prenese v medpomnilnik za pošiljanje in sprejemanje sporočil. Rutina knjižnice opravi celotno delo. Mi kopiramo sporočilo v medpomnilnik, od tam naprej

pa prevzame algoritem knjižnice, sam sporočilo odpošlje in ob tem postavi in pobriše potrebne zastavice.

Glavna datoteka knjižnice je J1939.c, ki je izvorna koda in jo moramo implementirati v lastno kodo programa. Povezati moramo tudi J1939.h, ki je »leader« osnovne kode. V teh datotekah so shranjene in definirane vse spremenljivke ter fizični naslovi mikrokrmilnika. Tretja datoteka, ki pripada knjižnici, pa je J1939.def, v tej datoteki so sistemske datoteke in ostala periferija, ki je potrebna za nemoteno delovanje knjižnice. Najpogostejša je namestitev v načinu, da knjižnica za svoje delovanje uporablja prekinitve. Možna je sicer tudi namestitev, da knjižnica kliče programsko funkcijo, kar imenujemo povpraševanje (polling).

Potek namestitve in koraki klicanja posameznih funkcij, potrebnih za delovanje; ko naš program opravi vse začetne rutine, kličemo datoteko knjižnice J1939\_Initialization, s pomočjo katere nastavimo ime in naslov naprave. Ko je inicializacija končana, se sprožita prekinitvi GIEH/GIE in/ali GIEL/PEIE, ki sta pokazatelja, da je inicializacija končana. Nato v programu kličemo funkcijo J1939\_Poll, to funkcijo moramo klicati zelo pogosto, vsaj vsakih nekaj milisekund, dokler ni bit WaitingForAddressClaimContention postavljen nazaj na vrednost 0. Ko je omenjena vrednost 0, program začne gledati bit CannonClaimAddress, če je tudi ta bit postavljen na 0, potem je sistem uspešno povezan na komunikacijsko vodilo. Funkcija J1939\_Poll ne glede na to, ali je sporočilo poslano ali prejeto, nadzira samo bite, ki so pomembni ob potrjevanju naslova in imena. Funkcija J1939\_ISR preverja, če je kateri izmed PIR3 bitov postavljen, saj to je pokazatelj, da imamo sprejeto sporočilo, hkrati pa prestavi prejeto sporočilo iz vhoda v medpomnilnik vhodnih sporočil, kjer sporočila čakajo na obdelavo.

Sporočilo definiramo preko funkcije J1939\_MESSAGE MyMessage; ta funkcija je definirana v datoteki J1939.h. Ročno moramo izpolniti naslednja polja:

- MyMessage.DataPage, (1 bit),
- MyMessage.Priority, (3 biti),
- MyMessage.PDUFormat, (8 bitov),
- MyMessage.PDUSpecific, (8 bitov).
  - Zadnje polje je lahko zamenjano z:
  - DestinationAddress ali GroupExtension.
- MyMessage.SourceAddress, (8 bitov),

- MyMessage.DataLength, (4 biti),
- MyMessage.Data, (64 bitov),

Knjižnica sama manipulira z dohodnimi sporočili. To pomeni, da ko sporočilo pride, pregleda ustreznost in ga nato prenese v medpomnilnik dohodnih sporočil. Ko imamo v medpomnilniku sporočilo, ki ga moramo obdelati, se nam postavi RXQueueCount, da pa preverimo omenjeni bit, pa moramo klicati rutino J1939\_DequeueMessage. Zastavica J1939\_flags.ReceivedMessagesDropped nam pove, ali smo katero sporočilo izpustili. Za pošiljanje sporočil pa kličemo J1939\_EnqueueMessage, da se sporočilo kopira v medpomnilnik za pošiljanje. Rutina J1939\_TransmitMessages izvede pošiljanje sporočila.

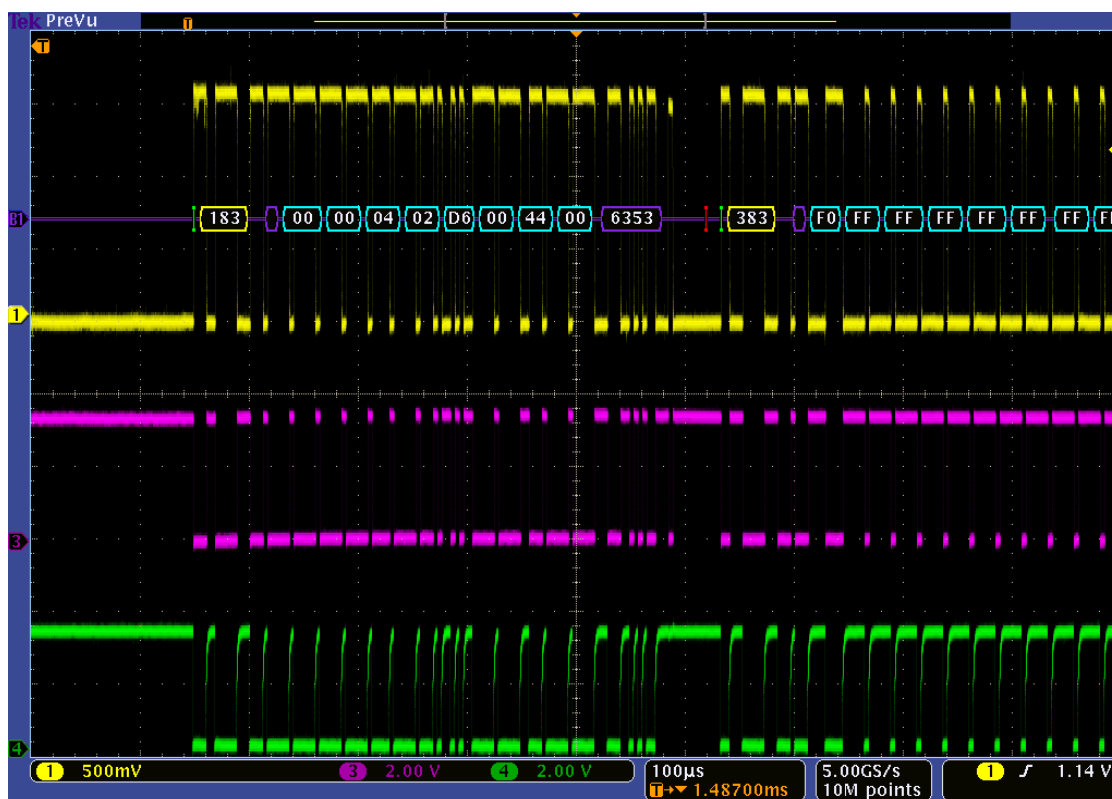
Za osnovo svojega programa sem vzel primer, ki je priložen knjižnici. Zaradi nevednosti manipuliranja s prekinitvami sem se odločil, da bom v programu sam klical funkcije za pregled prispelih sporočil in pregled potrebe po oddaji sporočila.

Sama rutina je postavljena tako, da naprava najprej potrdi svoje ime in naslov, nato pa je pripravljena za pošiljanje. Slika 7.9 prikazuje stanje na vodilu, ko mikroprocesor ne oddaja nič in je v stanju pripravljenosti. Rumena - bus linija (parici), vijola - gledano na mikrokrmilniku sprejem RX in zelena - gledano na mikrokrmilniku oddaja-TX).



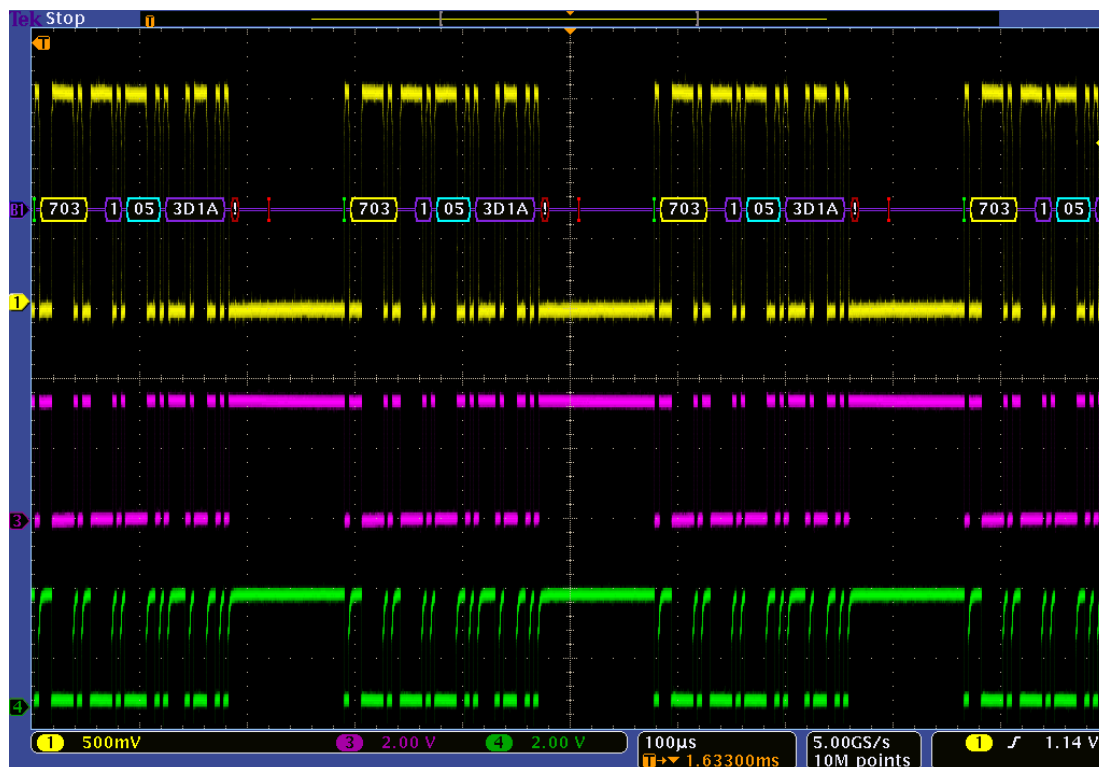
Slika 7.9: Prikaz stanja vodila, ko mikrokrmilnik ne oddaja ničesar.

S pritiskom tipke pošljemo sporočilo prek omrežja, Slika 7.10 prikazuje podatkovni paket v omrežju.



Slika 7.10: Prikaz povezanega omrežja s podatkovnim paketom.

Veliko težav sem imel z nastavitvijo omrežja. Največ preglavic sem imel s sinhronizacijo, z nastavljanjem časa takta prek oscilatorja. Ko naprava pravilno oddaja paket, in ko sta napravi priključeni, vsaka s svojo hitrostjo oddajanja, pa prikazuje slika 7.11.



Slika 7.11: Slika prikazuje ne sinhronizirano vodilo.

Signal, gledan na parici (zelene barve), je lepo viden, in sicer prikazuje, kako naprava pošilja sporočilo, ki ga druga naprava zaradi nesinhroniziranosti ne prebere. Ko zazna prva naprava, da sporočilo ni bilo prebrano, prekine pošiljanje, in ga ponovno pošlje, to počne, vse dokler ne preklopi v način delovanja napake, ali ko uporabnik odklopi napravo.

## **8. ZAKLJUČEK**

Tematika diplomske naloge je komunikacija na višjenivojskem protokolu J1939. Naloga sledi zaporedju, ki so si sledili v realnem razvoju omenjenega komunikacijskega protokola. Razlika je le v tem, da je bil realni projekt zgrajen na Freescale-ovem mikrokrmilniku, diplomska naloga pa je realizacija na PIC-ovem mikrokrmilniku. Sam potek je šel v isti smeri. Najprej je bilo treba raziskati literaturo s področja CAN bus komunikacije, kar je vzelo največ časa, nato raziskati in poiskati okolje, na katerem sem lahko začel razvijati komunikacijo. Nato naročiti vso opremo, se spoznati z opremo ter jo začeti uporabljati.

Sam namen diplomske naloge je prikazati začetek določenega projekta v industriji, hkrati pa obnoviti in razširiti znanje programiranja.

## 9. UPORABLJENA LITERATURA

- [1] W. Voss, A Comprehensible Guide to Controller Area Network, Copperhill Technologies Corporation, 2008.
- [2] W. Voss, A Comprehensible Guide to J1939, Copperhill Technologies Corporation, 2008.
- [3] J. Škvarč, PIC Programirajmo Mikrokontrolerje, AX Elektronika, 2004.
- [4] MPLAB ICD 3 In Circuit Debugger User's guide, Microchip Technology Inc., 2008.
- [5] MPLAB IDE User's guide, Microchip Technology Inc., 2008.
- [6] PIC18F2585/2680/4585/4680 Data Sheet, Microchip Technology Inc., 2007.
- [7] J1939 C Library for CAN-Enabled PICmicro Microcontrollers, Microchip Technology Inc., 2008.
- [8] PICDEM CAN-LIN 2 Development Kit User's Guide, Microchip Technology Inc., 2003.
- [9] J1939 Presentation:  
<http://www.simmasoftware.com/j1939.html>.
- [10] SAE J1939 Standard collection:  
<http://www.sae.org/>.
- [11] PEAK SYSTEM, User's manual:  
<http://www.peak-system.com>.
- [12] KVASER:  
<https://www.kvaser.com/canking/>.
- [13] ROBERT BOSCH GmbH, Bosch CAN Specification 2.0B, 1991.F
- [14] CanOPEN:  
<http://www.can-cia.org/>.
- [15] MilCAN:  
<http://www.milcan.org/>.
- [16] DeviceNET:  
<https://en.wikipedia.org/wiki/DeviceNet>.
- [17] DeviceNET:  
<http://www.rtaautomation.com/technologies/devicenet/>.
- [18] DeviceNET:  
<http://www.smar.com/en/devicenet>.
- [20] J1939:  
<http://www.j1939.org/>.

- [21] M. Kariž, Praktična aplikacija CAN komunikacijskega standarda del 1 in del 2, 2010.
- [22] Načini kodiranja:  
<https://en.wikipedia.org/wiki/Non-return-to-zero>.
- [23] A CAN Physical Layere Discussion, Microchip Technology Inc., 2002.
- [24] MCP2551 High-Speed CAN transceiver Data Sheet, Microchip Technology Inc., 2010.
- [24] CAN basics,  
<http://techiscafe.blogspot.si/p/the-controller-area-network-can.html>.
- [25] Controler Area Network for Cars,  
[http://www.aalcar.com/library/can\\_systems.htm](http://www.aalcar.com/library/can_systems.htm).