

# CONTROLLER AREA NETWORK

---

**Vragotuk, Boris**

**Undergraduate thesis / Završni rad**

**2021**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Karlovac University of Applied Sciences / Veleučilište u Karlovcu**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:128:127172>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-11-23**



**VELEUČILIŠTE U KARLOVCU**  
Karlovac University of Applied Sciences

*Repository / Repozitorij:*

[Repository of Karlovac University of Applied Sciences - Institutional Repository](#)



zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

# CONTROLLER AREA NETWORK

---

**Vragotuk, Boris**

**Undergraduate thesis / Završni rad**

**2021**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Karlovac University of Applied Sciences / Veleučilište u Karlovcu**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:128:127172>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2023-02-13**



**VELEUČILIŠTE U KARLOVCU**  
Karlovac University of Applied Sciences

*Repository / Repozitorij:*

[Repository of Karlovac University of Applied Sciences - Institutional Repository](#)



zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

**VELEUČILIŠTE U KARLOVCU**  
STROJARSKI ODJEL  
STRUČNI STUDIJ MEHATRONIKE

BORIS VRAGOTUK

# **CONTROLLER AREA NETWORK**

ZAVRŠNI RAD

KARLOVAC, 2021.



**Veleučilište u Karlovcu**

Strojarski odjel

Stručni studij mehatronike

Boris Vragotuk

# **Controller Area Network**

Završni rad

Mentor: dr. sc. Adam Stančić, v. pred.

Broj indeksa studenta: 0314673031

Karlovac, siječanj 2021.

## **PREDGOVOR**

Ja, **Boris Vragotuk**, ovime izjavljujem da je moj završni rad pod naslovom **Controller Area Network** rezultat vlastitog rada i istraživanja te se oslanja na izvore i radove navedene u bilješkama i popisu literature. Ni jedan dio ovoga rada nije napisan na nedopušten način, odnosno nije prepisan iz necitiranih radova i ne krši autorska prava.

Sadržaj ovoga rada u potpunosti odgovara sadržaju obranjenoga i nakon obrane uređenoga rada.

Karlovac, 18. siječnja 2021.

Ime i prezime studenta

---

## **Sažetak**

Proizvodnja automobila je kompleksan zadatak dodatno otežan visokom cijenom samog postupka. Sirovine i poluproizvodi potrebni za obavljanje tog zadatka čine gotovo 50% totalne cijene proizvodnje vozila. Bakrene žice potrebne za upravljanje raznim modulima osim same cijene zahtijevaju i rad koji se zbog složenosti postupka montiranja ne može efektivno automatizirati. Da bi se ovi troškovi snizili te da se zadovolji potreba za učinkovitim upravljanjem sve većeg broja elektroničkih komponenata u vozilu, osmišljen je CAN bus.

Controller Area Network je prva i najčešće korištena komunikacijska mreža u automobilima. Današnji automobili sadrže otprilike 1,5 kilometara bakrene žice; bez komunikacijske mreže ta brojka bi bila znatno veća.

U ovom radu bit će objašnjen način funkcioniranja CAN-a, te demonstriran rad ABS-a, sustava koji je neophodan za današnja vozila koji funkcionira preko CAN-a.

Ključne riječi: ABS, Automobil, CAN bus, Controller Area Network

## **Abstract**

Car production is a complex task further complicated by the high cost of the process itself. The raw materials and semi-finished products needed to perform this task make up almost 50% of the total cost of vehicle production. The copper wires needed to control the various modules, in addition to the cost itself, also require work that cannot be effectively automated due to the complexity of the installation process. In order to reduce these costs and to meet the need for efficient management of a growing number of electronic components in the vehicle, CAN bus has been designed.

CAN (Controller Area Network) is the first and most frequently used communication network in cars. Today's cars contain about 1.5 kilometers of copper wire; without a communication network that figure would be significantly higher.

This paper will explain the way CAN works, and demonstrate the operation of ABS, a system that is necessary for today's vehicles that works through CAN.

Keywords: ABS, Car, CAN bus, Controller Area Network



# Sadržaj

<b>1. UVOD.....</b>	<b>1</b>
<b>2. Struktura i način rada CAN-a .....</b>	<b>3</b>
2.1. Arhitektura CAN-a .....	3
2.2. Prijenos podataka.....	3
2.3. Mjerenje vremena bitova.....	5
2.4. Slojevi CAN-a .....	5
2.5. Format i tip CAN okvira .....	9
2.6. Vrste CAN okvira.....	10
2.7. Utor za potvrdu primitka.....	15
2.8. Međuokvirni prostor .....	15
2.9. Sigurnosne značajke CAN-a .....	15
<b>3. Demonstracija funkcioniranja ABS sustava .....</b>	<b>16</b>
3.1. Korišteni softver i hardver .....	17
3.2. Računalna demonstracija kočenja.....	23
<b>4. Zaključak.....</b>	<b>35</b>
<b>5. Literatura.....</b>	<b>36</b>
<b>6. Prilozi.....</b>	<b>38</b>
6.1 Popis slika .....	38
6.2 Popis tablica.....	39



## 1. UVOD

CAN bus, skraćeno za Controller Area Network je specijalizirana komunikacijska mreža koja spaja komponente u vozilu. Dizajnirana je da omogući komuniciranje između mikrokontrolera i raznih uređaja unutar vozila bez glavnog računala. CAN je message-based protocol, odnosno, protokol zasnovan na poruci, isprva namijenjen kao mjera uštede bakra tako da smanji količinu žičnih vodiča potrebnih u vozilu [1]. Međutim odmah je postalo jasno da je iskoristiv u različitim situacijama. Svaki uređaj u mreži prima svaki paket koji je poslan unutar mreže, bez obzira je li taj paket bio namijenjen tom uređaju ili ne. Paketi funkcioniraju po stupnju prioriteta, što znači da u slučaju da više od jednog uređaja šalje pakete informacija istovremeno, uređaj sa većim prioritetom ima prednost rabljenja mreže.

Razvoj CAN-a započeo je 1983. u uredima elektroničke tvrtke Robert Bosch GmbH [7]. Završen je i prezentiran 1986. na konferenciji Društva Automobilnih Inženjera (engl. Society of Automobile Engineers) u Michiganu. Prvi CAN kontroleri proizvedeni su od strane kompanije Intel 1987. godine, te ih je i kompanija Philips ubrzo počela proizvoditi. Godine 1991. Mercedes Benz W140 bio je prvi automobil proizveden koji je bio opremljen sustavom baziranim na CAN-u [8].

Bosch je objavio nekoliko verzija CAN specifikacija te je 1993. godine CAN standardiziran od strane ISO-a (International Organization for Standardization – Međunarodna organizacija za standardizaciju) [9]. Bosch je i dalje veoma aktivan u istraživanju i proširenju CAN tehnologije te je 2012. izdao CAN FD 1.0 (Controller Area Network Flexible Data-Rate) [10]. Ova specifikacija koristi drugačiji format paketa koji omogućuje drugačije količine podataka u paketu uz mogućnost prelaska na veću brzinu prijenosa podataka. Zahvaljujući standardizaciji, CAN FD je kompatibilan sa postojećim CAN mrežama tako da novi CAN FD uređaji mogu raditi na istoj mreži kao i već postojeći CAN uređaji [2].

CAN se koristi na mnogo područja u industriji, najznačajniji su:

- Putnička vozila, kamioni, autobusi (električni i na unutarnje izgaranje)
- Poljoprivredna oprema
- Elektronička oprema za avijaciju i navigaciju
- Industrijska automatizacija
- Dizala, pokretne stepenice
- Medicinski instrumenti i oprema

Moderni automobil može imati više od 70 elektronskih upravljačkih jedinica (Electronically Controlled Unit-ECU). Najveći ECU je najčešće upravljačka jedinica motora. Ostali ECU-i koriste se za zračne jastuke, upravljanje prijenosom brzine, servo upravljače, audio sustave, prilagodbu retrovizora, prozora i vrata, baterije i sustave za punjenje u hibridnim i električnim vozilima, te za ABS sustav (engl. Antilock Breaking System – hrv. Sustav protiv sprječavanja blokiranja kočnja) čiji rad će biti demonstriran. Neki od ovih sustava čine vlastite nezavisne podsustave, podsustav možda mora kontrolirati aktuatore ili dobiti povratne informacije od senzora stoga je komunikacija između ECU-ova ključna. Ključna prednost je da povezanost između različitih sustava pruža širok raspon sigurnosnih, ekonomskih i praktičnih svojstva koji se mogu primjeniti koristeći isključivo softver, te svojstva koja bi imala visoke troškove te dodatno povećala kompleksnost proizvodnog postupka da su primjenjena tradicionalnim elektrotehničkim metodama.

Primjerice "hill hold" funkcionalnost dobiva ulazni signal iz senzora nagiba i brzinomjera putem CAN-a te tako odlučuje da li je vozilo stalo na usponu. Signalom od tih istih senzora protuprovalni alarm procjenjuje da li je provala u tijeku te u slučaju da je, šalje signale da se aktivira alarm. Signalom od opisanih te ovisno o potrebi, ostalih senzora, sustav za kontrolu proklizavanja odlučuje mora li intervenirati. Senzori mogu biti postavljeni na prikladnim lokacijama s obzirom na njihovu zadaću, i biti spojeni s CAN mrežom, tako da svaki ECU ima pristup njihovim mjerenjima. Primjerice senzori za temperaturu koji su tradicionalno bili smješteni na prednjoj strani vozila sada mogu biti smješteni na retrovizorima, izbjegavajući toplinu od motora.

## **2. Struktura i način rada CAN-a**

U narednim točkama će biti objašnjenja arhitektura, funkcionalnost te fizički sloj CAN-a. Posebna pozornost posvetiti će se prijenosu podataka tj. arbitraži poruka koja se određuje pomoću dominantnih i recesivnih bitova.

### **2.1. Arhitektura CAN-a**

CAN je "multi-master serial bus" standard za spajanje ECU-a koji se u praksi nazivaju čvorovima [3]. Multi-master serial bus je sabirnica na kojoj se nalazi više komponenta koje imaju pravo vidjeti i djelovati na svu komunikaciju koja se odvija na sabirnici. Jednostavnije rečeno, čvorovi na CAN mreži imaju pravo pristupiti svim podacima koji se šalju na toj mreži, te se ti podatci šalju sekvencijalno, jedan bit za drugim. Ovakav način funkcioniranja je idealan za specijalizirani zatvoreni sustav poput onog u automobilu ili radnom stroju. U standardnom fizičkom sloju svi čvorovi su međusobno povezani sa dvije žice koje imaju 120  $\Omega$  otpora na svakom kraju gdje se međusobno povezuju. Ti otpornici služe suzbijanju refleksije signala te vraćanju sabirnice u svoje recesivno stanje i tako čine terminaciju CAN sustava.

### **2.2. Prijenos podataka**

Ključni dio CAN poruke čini identifikator, ID (engl. identifier) se koristi za prepoznavanje poruke, određivanje njezinog prioriteta te rješavanje arbitraže.

ID vrijednost poruke mora biti jedinstvena na jednoj CAN sabirnici, inače će dva čvora nastaviti prijenos dalje od kraja arbitražnog polja (ID) uzrokujući pogrešku.

Početakom 90-ih, izbor ID-ova za poruke formiran je na temelju identifikacije vrste podataka i čvora za slanje [4]. Međutim, kako se ID koristi i kao prioritet poruke, to je dovelo do loših performansi u stvarnom vremenu. U tim scenarijima obično se traži nisko opterećenje CAN sabirnice od oko 30% kako bi se osiguralo da sve poruke budu dostavljene u zadovoljavajućem roku. Međutim, ako se ID-ovi umjesto toga određuju na temelju roka poruke, manji je brojčani ID i stoga je veći prioritet poruke, tada se opterećenje sabirnice od 70 do 80% obično može postići prije nego što se propusti bilo koji rok za primanje poruke.

CAN prijenos podataka koristi metodu arbitraže rješavanja „prepirke“ bez gubitaka bitova. Ova metoda zahtjeva da svi čvorovi na CAN mreži budu sinkronizirani da prepoznaju svaki bit na mreži u svakom trenutku. Iz ovog razloga je CAN protokol netočno smatran sinkronim protokolom, jer se podatci prenose u asinkronom formatu tj. bez signala vremena. CAN-ove specifikacije koriste izraze "dominantni" i "recesivni" bitovi, gdje je dominantna logička nula (aktivno prebačena na napon odašiljačem), a recesivna je logička jedinica (otpornik ju pasivno vraća na napon). Stanje u mirovanju predstavljeno je recesivnom razinom (logička jedinica). Ako jedan čvor prenosi dominantni bit, a drugi čvor prenosi recesivni bit, tada dolazi do sudara i dominantni bit "pobjeđuje". To znači da nema odgode za poruku višeg prioriteta, a čvor koji odašilje poruku nižeg prioriteta automatski će ponovno pokušati poslati poruku šest bitnih taktova (šest kvadratnih valova) nakon završetka dominantne poruke. To čini CAN idealnim kao prioritetni komunikacijski sustav u stvarnom vremenu.

Točni naponi za logičku nulu ili jedinicu ovise o korištenom fizičkom sloju, ali osnovni princip CAN-a zahtjeva da svaki čvor sluša podatke u CAN mreži, uključujući i sam čvor koji šalje podatke. Ako logičku jedinicu odašilju svi čvorovi za prijenos istovremeno, tada logičku jedinicu vide svi čvorovi, uključujući sve čvorove odašiljače i prijemne čvorove. Ako logičku nulu odašilju istovremeno svi čvorovi odašiljači, tada logičku nulu vide svi čvorovi. Ako logičku nulu prenosi jedan ili više čvorova, a logičku jedinicu prenosi jedan ili više čvorova, tada logičku nulu vide svi čvorovi uključujući i čvorove koji odašilju logičku jedinicu. Kad čvor prenosi logičku jedinicu, ali vidi logičku nulu, „shvaća da postoji prepirka“ i prestaje s prijenosom. Korištenjem ovog postupka, svaki čvor koji prenosi logičku jedinicu kad drugi čvor prenosi logičku nulu "ispadne" tj. izgubi arbitražu. Čvor koji izgubi arbitražu automatski ponovno pokušava poslati podatke šest bitnih taktova kasnije, a protok CAN podataka se nastavlja bez pogreške dok ne ostane samo jedan čvor odašiljač. To znači da čvor koji prenosi prvu jedinicu gubi arbitražu. Budući da 11-bitni identifikator (ili 29-bitni za CAN 2.0B standard) prenose svi čvorovi na početku CAN okvira, čvor s najnižim identifikatorom prenosi više nula na početku okvira, to je čvor koji pobjeđuje arbitražu tj. ima najviši prioritet [11]. Na primjer, kao što je prikazano u tablici ispod, uzima se u obzir 11-bitna ID CAN mreža, s dva čvora s ID-om 15 (binarno predstavljanje, 00000001111) i 16 (binarno predstavljanje, 00000010000). Ako ova dva čvora istodobno odašilju, svaki će najprije poslati početni bit, a zatim prenijeti prvih šest nula svog ID-a bez arbitražne odluke.

	Početni bit	ID bitovi											Ostatak okvira
		10	9	8	7	6	5	4	3	2	1	0	
Čvor 15	0	0	0	0	0	0	0	0	1	1	1	1	
Čvor 16	0	0	0	0	0	0	0	1	Kraj prijenaosa				
CAN podatci	0	0	0	0	0	0	0	0	1	1	1	1	

Tablica 1: 11-bitna ID CAN mreža s dva čvora s ID-om 15 i 16

Kada se prenosi sedmi ID bit, čvor s ID-om 16 odašilje jedan (recesivno) za svoj ID, a čvor s ID-om 15 odašilje nula (dominantno) za svoj ID. Kad se to dogodi, čvor s ID-om 16 zna da je prenio jedan, ali „vidi“ dominantnu vrijednost nula čime je došlo do „sudara“ te da je izgubio arbitražu. Čvor 16 prestaje prenositi što omogućuje čvoru s ID-om 15 da nastavi prijenos bez gubitka podataka. Čvor s najnižim ID-om uvijek će pobijediti u arbitraži i stoga ima najveći prioritet.

### 2.3. Mjerenje vremena bitova

Svi čvorovi u CAN mreži moraju raditi s istim nominalnim bitovima, ali šum, fazni pomaci, tolerancija oscilatora i slične pojave utječu na stvarnu brzinu prijenaosa bita koji onda možda ne čini nominalnu brzinu. Kombinacija navedenih smetnji može rezultirati propuštenim CAN porukama. Budući da se zasebni signal sata ne koristi, potrebno je na određen način sinkronizirati čvorove. Sinkronizacija je važna tijekom arbitraže jer čvorovi u arbitraži moraju imati uvid u prenesene podatke vlastitog i drugih čvorova. Sinkronizacija je također važna kako bi se osiguralo da varijacije u vremenu oscilatora između čvorova ne uzrokuju pogreške.

### 2.4. Slojevi CAN-a

CAN protokol određuje pravila za implementaciju fizičkog i transportnog sloja OSI modela kako bi se izvršio serijski prijenos podataka između dva ili više uređaja. Apstraktni slojevi su način skrivanja radnih detalja podsustava, omogućavajući razdvajanje problema kako bi se olakšala interoperabilnost i neovisnost platforme. Apstraktni slojevi CAN protokola su slijedeći:

➤ **Aplikacijski sloj**

➤ **Sjednički sloj**

- Filtriranje poruka
- Rukovanje porukama i statusom

➤ **Transportni sloj**

Većina standarda CAN-a spada u transportni sloj. Transportni sloj prima poruke iz fizičkog sloja i transmitira ih prema objektnom sloju. Transportni sloj je odgovoran za mjerenje vremena bitova i sinkronizaciju, uokviravanje poruka, arbitražu, potvrdu, uočavanje grešaka i signalizaciju te zadržavanje kvara. Funkcije transportnog sloja su:

- Zadržavanje kvara
- Uočavanje grešaka
- Validaciju poruka
- Potvrđivanja
- Arbitražu
- Uokviravanje poruka
- Stopa transfera i timing
- Usmjeravanje informacija

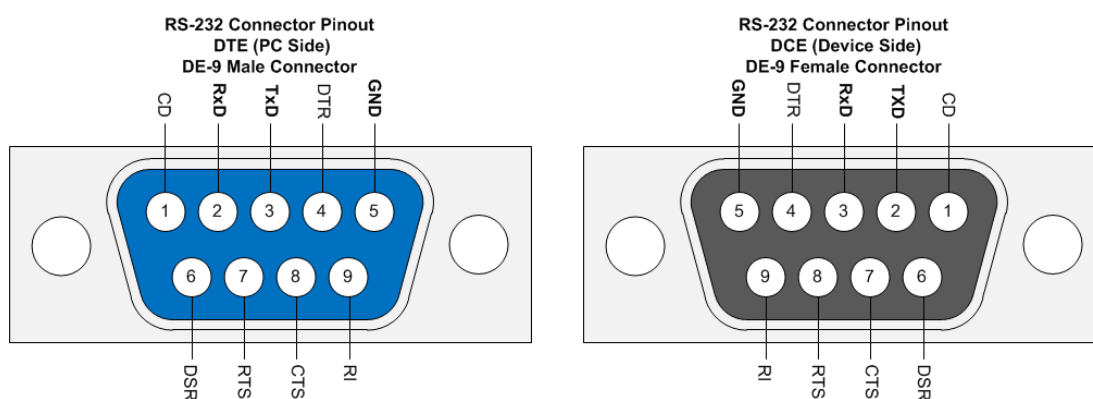
➤ **Fizički sloj**

CAN sabirnica (međunarodni standard: ISO 11898 – 1:2003) je izvorno određivao protokol sloja za veze samo s apstraktnim zahtjevima za fizički sloj npr. isticanje korištenja medija s višestrukim pristupom na razini bita korištenjem dominantnih i recesivnih stanja. Električni aspekti fizičkog sloja (napon, jakost struje, broj vodiča) navedeni su u standardu ISO 11898 – 2:2003 koji je danas široko prihvaćen. Međutim, mehanički aspekti fizičkog sloja (vrsta i broj konektora, boje, oznake, pin-outovi) tek trebaju biti formalno navedeni. Kao rezultat toga, automotivni ECU obično će imati određene – često prilagođene – konektore s raznim vrstama kabela od kojih su dvije veze CAN sabirnica. Ipak nekoliko de facto standarda za mehaničku



implementaciju su se pojavili, najčešći kao 9-pinski D-sub konektor sa sljedećim pribadačama (pin – ovima) kako je prikazano na slici 1:

- pin 2: CAN-Low (CAN -)
- pin 3: GND (Ground)
- pin 7: CAN-High (CAN+)
- pin 9: CAN V+ (Power)



Slika 1: Tip 9-pin muški (s lijeva) i ženski (s desna) konektor

Ovaj de facto mehanički standard za CAN može biti implementiran s čvorom imajući oba muška i ženska pin-9 D-sub konektora električki spojenih jedan prema drugome paralelno s čvorovima. Snaga sabirnice napaja se s muškim konektorom čvora, a sabirnica crpi snagu iz čvora ženskog konektora. Prihvatanje ovog standarda izbjegava nuždu za stvaranjem prilagođenih razdjeljivača da spoje dva skupa sabirničkih žica na jedan D konektor na svaki čvor. Takvi nestandardni (prilagođeni) kabljski svežnjici koji spajaju vodiče izvan čvora smanjuju pouzdanost sabirnice uklanjajući zamjenjivost kabela, smanjuju kompatibilnost kabljskih svežnja i povećavaju troškove.

Odsutnost potpune specifikacije fizičkog sloja (mehaničkog uz električni) oslobodilo je specifikaciju CAN sabirnice od ograničenja i složenosti fizičke implementacije. Međutim, zbog mehaničke nespojivosti, implementacija CAN sabirnice otvorena je za pitanja interoperabilnosti. Kako bi poboljšali interoperabilnost, mnogi proizvođači vozila generirali su specifikacije koje opisuju skup dopuštenih CAN primopredajnika u kombinaciji sa zahtjevima za otpornost prema parazitskim kapacitetom na liniji. Dopušteni parazitski kapacitet uključuje oba kondenzatora,

kao i ESD (engl. Electrostatic discharge, hrv. Elektrostatski izboj) zaštitu. Uz parazitski kapacitet, sustavi od 12 V i 24 V nemaju iste zahtjeve u pogledu maksimalnog napona vodova. Tijekom jump starta (način pokretanja vozila s ispražnjenom baterijom) napon sustava lakih vozila se može popeti do 24 V, dok sustavi kamiona mogu doseći čak 36 V.

Otpornost na radiofrekvencijsku buku prema standardu ISO 11898 – 1:2003 postiže se održavanjem diferencijalne impedancije sabirnice na niskoj razini s otpornicima male vrijednosti (120  $\Omega$ ) na svakom kraju sabirnice. Međutim, za vrijeme mirovanja, sabirnica niske impedancije poput CAN-a povlači više struje (i snage) od ostalih signalnih sabirnica zasnovanih na naponu. U sustavima CAN sabirnica, uravnoteženoga rada linije, jakost struje je u jednoj signalnoj liniji točno uravnotežena odgovarajućom jakošću struje u suprotnom smjeru. Signali uravnoteženih parova CAN sabirnice se prenose u žicama upletenih parova u oklopljenom kabelu kako bi se smanjila RF (radiofrekvencijska) emisija i smanjila osjetljivost na smetnje u ionako bučnom RF okruženju automobila.

Standard ISO 11898 – 1:2003 pruža određenu otpornost na oscilacije napona tokom uobičajenog načina rada odašiljača i prijammika tako što vodič od 0 V prolazi duž sabirnice kako bi se održao visok stupanj povezanosti napona između čvorova. Također, u gore spomenutoj mehaničkoj konfiguraciji uključena je opskrba sabirnica za raspodjelu snage na svaki od čvorova primopredajnika. Dizajn osigurava zajedničku opskrbu svih primopredajnika. Stvarni napon koji će primijeniti sabirnica i koji se čvorovi na nju primjenjuju su specifični za aplikaciju i nisu formalno navedeni. Uobičajena praksa dizajna čvorova pruža svakom čvoru primopredajnike koji su optički izolirani od čvora domaćina i izvode 5 V linearno reguliranog napona napajanja za primopredajnike iz univerzalne opskrbe tračnice koju pruža sabirnica. To omogućuje interoperabilnost na mnogim vrstama čvorova. Tipične vrijednosti napona napajanja na takvim mrežama su 7 do 30 V. Međutim, nedostatak formalnog standarda znači da su dizajneri sustava odgovorni za kompatibilnost opskrbnih vodova.

Standard ISO 11898 – 1:2003 opisuje električnu implementaciju oblikovanu od konfiguracije višepadajuće jednostrane žice s prekidanjem otpornika sa svake strane sabirnice. U ovoj se konfiguraciji dominantno stanje utvrđuje jednim ili više odašiljača koji prebacuju CAN<sup>-</sup> na napajanje 0 V i (istovremeno) prebacuju CAN<sup>+</sup> na napon sabirnice +5 V, čime se stvara strujni put kroz otpornike koji prekidaju rad sabirnice. Kao takvi, prekidajući otpornici čine bitnu komponentu signalnog sustava i uključeni su ne samo kako bi ograničili refleksiju valova na visokoj frekvenciji.

Za vrijeme recesivnog stanja kabelski signal i otpornici ostaju u visokom impedancijskom stanju u odnosu na obje spojnice. Recesivno stanje je prisutno u sabirnici samo kada nijedan transmiter na sabirnici ne zauzima dominantno stanje.

Za vrijeme dominantnog stanja kabelski signal i otpornici prelaze u stanje niske impedancije u odnosu na spojnice tako da struja prolazi kroz otpornike. Napon CAN+ ide do +5 V, a CAN- 0 V. Ova se signalna strategija bitno razlikuje od ostalih tehnologija uravnoteženog linijskog prijenosa kao što su RS-422/3, RS-485 koje upotrebljavaju pogonske sklopove / prijemnike diferencijalnih linija i koriste signalni sustav temeljen na naponu diferencijalnog načina uravnotežene linije koja prelazi zamišljenih 0 V. Višestruki pristup na takvim sustavima obično se oslanja na medije koji podržavaju tri stanja (aktivno visoko, aktivno nisko i neaktivno stanje) i on se rješava u vremenskoj domeni. Višestruki pristup na CAN sabirnici postiže se električnom logikom sustava koji podržava samo dva stanja koja su konceptualno analogna „Wired AND“ mreži.

## **2.5. Format i tip CAN okvira**

CAN mreža može se konfigurirati za rad s dva različita formata poruke (ili "okvira"): standardni ili osnovni format okvira (CAN 2.0 A i CAN 2.0 B) i prošireni format okvira (CAN 2.0 B). Jedina razlika između dva formata je da "CAN osnovni okvir" podržava duljinu od 11 bita za identifikator, a "CAN prošireni okvir" podržava duljinu od 29 identifikatorskih bitova, sastavljen od 11-bitnog identifikatora ("osnovni identifikator") i 18-bitno proširenje ("proširenje identifikatora"). Razlika između CAN osnovnog formata okvira i CAN proširenog formata okvira vrši se korištenjem IDE (engl. Identifier extension flag) bita, koji se prenosi kao dominantan u slučaju 11-bitnog okvira, a prenosi se kao recesivan u slučaju 29-bitnog okvira. CAN kontroleri koji podržavaju poruke proširenog okvira također mogu slati i primiti poruke u CAN osnovnom formatu okvira. Svi okviri počinju s bitom početka kadra SOF (engl. start-of-frame) koji označava početak prijenosa okvira.

## 2.6. Vrste CAN okvira

CAN ima četiri vrste okvira tj. poruka:

- Okvir podataka (Data frame): Okvir koji sadrži čvorove podataka za njihov prijenos
- Udaljeni okvir (Remote frame): Okvir koji zahtjeva prijenos specifičnog identifikatora
- Okvir pogreške (Error frame) Okvir prenešen od strane bilo kojeg čvora koji detektira grešku
- Okvir za signalizaciju zagušenja (Overload frame): Okvir koji ubacuje vremensko odgađanje između podataka i udaljenog okvira

### ➤ Okvir podataka

CAN standardi zahtjevaju da implementacija mora prihvaćati osnovni okvirni format i može prihvaćati prošireni okvirni format ali mora tolerirati prošireni okvirni format.

Okvir podataka je jedini okvir za prijenos podataka. On ima dva formata:

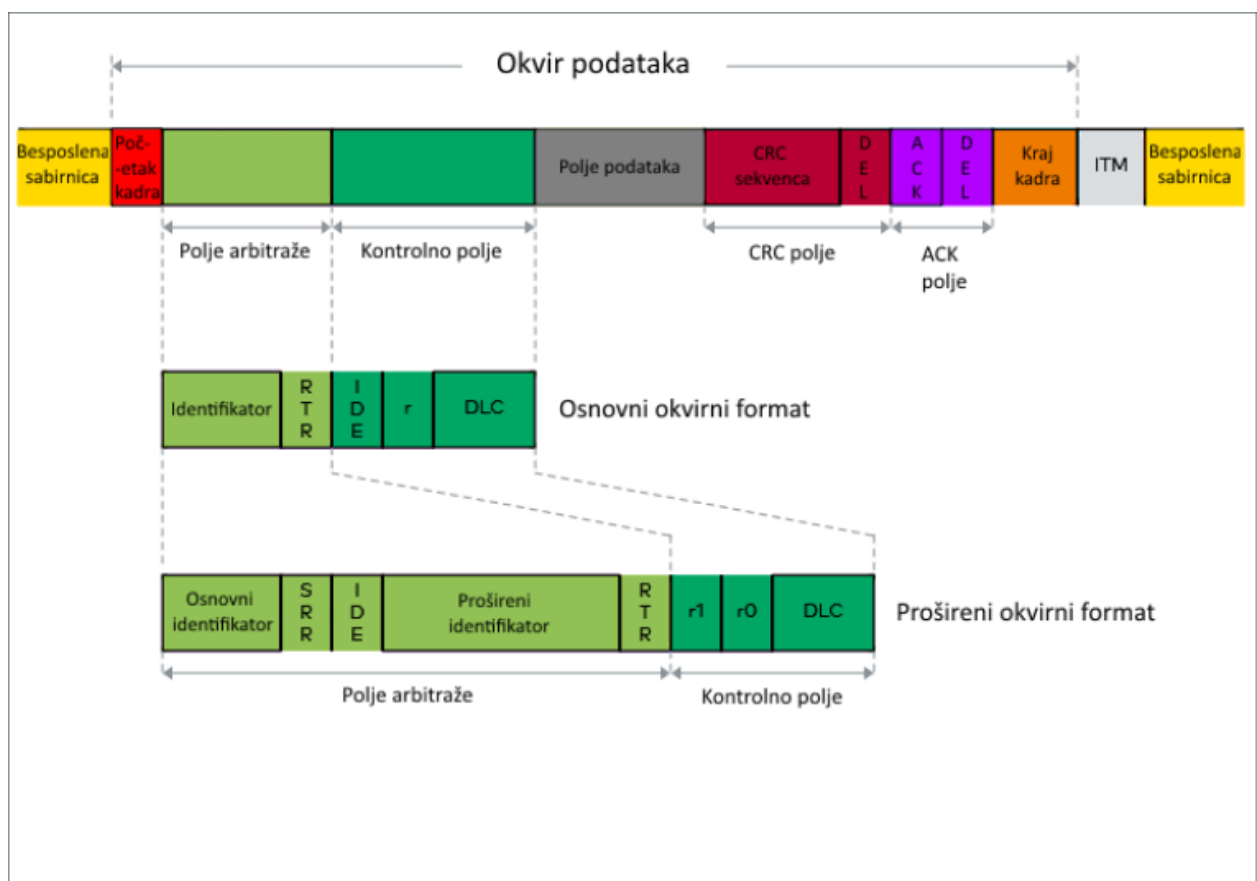
- Osnovni okvirni format: s 11 identifikatorskih bitova

Format okvira je slijedeći. Vrijednosti bitova opisane su za CAN-LO signal.

Naziv polja	Dužina (bit)	Svrha
Početak kadra	1	Označava početak prijensa okvira
Identifikator	11	Jedinstveni identifikator koji također predstavlja prioritet poruke
Zahtjev za daljinski prijenos (RTR)	1	Mora biti dominantan (0) za podatkovne okvire i recesivni (1) za zahtjeve udaljenog okvira
Identifikatorski produžni bit (IDE)	1	Mora biti dominantan (0) za osnovni format okvira s 11-bitnim identifikatorima
Rezervirani bit (r)	1	Rezervirani bit. Mora biti dominantan (0), ali je prihvaćen bez obzira je li dominantan ili recesivan.
Kod duljine podataka (DLC)	4	Daje do znanja broj bajtova podataka (0–8 bajtova)
Polje podataka	0-64 (8 bajtova)	Podaci za prijenos (duljina u bajtovima je diktirana DLC poljem)

CRC sekvenca	15	Kod za otkrivanje pogrešaka koji se obično koristi u digitalnim mrežama za otkrivanje slučajnih promjena podataka
CRC graničnik (DEL)	1	Mora biti recesivno (1)
ACK utor	1	Odašiljač šalje recesivan bit (1), bilo koji prijemnik može izjaviti dominantan bit (0)
ACK graničnik (DEL)	1	Mora biti recesivno (1)
Kraj kadra	7	Mora biti recesivno (1)

Tablica 2: Osnovni okvirni format



Slika 2: Grafički prikaz osnovnog i proširenog okvira podataka

- Prošireni okvirni format: s 29 identifikatorskih bitova

Format okvira je slijedeći. Vrijednosti bitova opisane su za CAN-LO signal.

Naziv polja	Dužina (bit)	Svrha
Početak kadra	1	Označava početak prijena okvira
Osnovni identifikator	11	Prvi dio jedinstvenog identifikatora koji također predstavlja prioritet poruke
Zamjenski daljinski zahtjev (SRR)	1	Mora biti recesivno (1)
Identifikatorski produžni bit (IDE)	1	Mora biti recesivno (1) za format proširenog okvira s 29-bitnim identifikatorima
Prošireni identifikator	18	Drugi dio jedinstvenog identifikatora koji također predstavlja prioritet poruke
Zahtjev za daljinski prijenos (RTR)	1	Mora biti dominantno (0) za podatkovne okvire i recesivno (1) za zahtjeve udaljenog okvira
Rezervirani bitovi (r1) (r0)	2	Rezervirani bitovi koji moraju biti postavljeni kao dominantni (0), ali su prihvaćeni bez obzira jesu li dominantni ili recesivni
Kod duljine podataka (DLC)	4	Broj bajtova podataka (0–8 bajtova)
Polje podataka	0-64	Podaci za prijenos (duljina u bajtovima diktirana DLC poljem)
CRC sekvenca	15	Kod za otkrivanje pogrešaka koji se obično koristi u digitalnim mrežama za otkrivanje slučajnih promjena podataka
CRC graničnik (DEL)	1	Mora biti recesivno (1)
ACK utor	1	Odašiljač šalje recesivan bit (1), bilo koji prijemnik može izjaviti dominantan bit (0)
ACK graničnik (DEL)	1	Mora biti recesivno (1)
Kraj kadra	7	Mora biti recesivno (1)

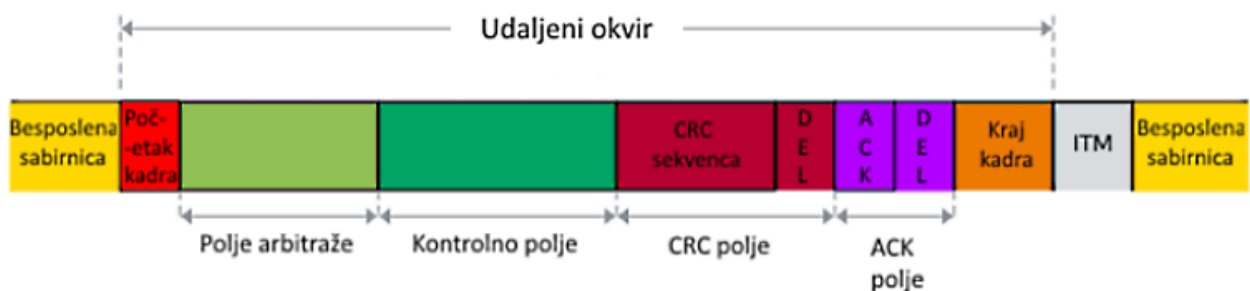
Tablica 3: Prošireni okvirni format

### ➤ Udaljeni okvir

Udaljeni okvir je gotovo identičan podatkovnom okviru, s dvije važne razlike, a to su:

- Izričito je označen kao udaljeni okvir (RTR bit u polju arbitraže je recesivan)
- Ne sadrži polje podataka

Svrha udaljenog okvira je tražiti prijenos odgovarajućeg podatkovnog okvira. Ako, na primjer čvor A pošalje udaljeni okvir s poljem arbitraže postavljenim na 234, tada čvor B, ako je pravilno inicijaliziran, može odgovoriti podatkovnim okvirom s poljem arbitraže također postavljenim na 234. Udaljeni okviri mogu se koristiti za implementaciju zahtjev-odgovor vrste upravljanja. Udaljeni okvir ima jedan specifičan problem: kod duljine podataka (DLC) mora biti postavljen na duljinu očekivane poruke odgovora. Inače arbitraža neće uspjeti. U slučaju da se istodobno prenose okvir podataka i udaljeni okvir s istim identifikatorom, okvir podataka dobiva arbitražu zbog dominantnog RTR bita koji slijedi identifikator.



Slika 3: Grafički prikaz udaljenog okvira

### ➤ Okvir pogreške

Okvire pogrešaka prenose bilo koji čvorovi kada otkriju pogrešku s okvirom podataka ili udaljenim okvirom. Ako je čvor u aktivnom stanju pogreške, on će poslati aktivni okvir pogreške. Ako je čvor u stanju pasivne pogreške, prenijet će pasivni okvir pogreške.

Ovi okviri pogrešaka će "uništiti" trenutne okvire podataka ili udaljene okvire na sabirnici. Čvor odašiljača znat će da poruku nisu svi čvorovi primili ispravno i automatski će pokušati ponovni prijenos u sljedeće dostupno "tiho vrijeme" na sabirnici (tj. kada je sabirnica u besposlenom stanju).

Okvir pogreške sastoji se od samo dva različita polja:

- Prvo polje je Zastave pogrešaka (engl. Error flags, 6–12 dominantnih / recesivnih bitova)
- Sljedeće polje je Graničnik pogrešaka (engl. Error delimiter, 8 recesivnih bitova)

Postoje dvije vrste zastave pogrešaka:

- Oznaka aktivne pogreške:

šest dominantnih bitova - Prenosi ih čvor koji otkriva grešku na mreži koja je u stanju pogreške "pogreška aktivna"

- Oznaka pasivne pogreške:

šest recesivnih bitova - Prenosi ih čvor koji otkriva aktivni okvir pogreške na mreži koji je u stanju pogreške "pogreška pasivna".

### ➤ Okvir za signalizaciju zagušenja

Okvir za signalizaciju zagušenja je posebna verzija okvira pogreške koja neće uzrokovati ponovni prijenos posljednje "uništene" poruke. Umjesto toga, čvor ga koristi za traženje odgode između okvira podataka ili udaljenih okvira tijekom međuokvirnog prostora ili tihog vremena na sabirnici. Ako čvoru treba više vremena za obradu poruke na razini protokola, on može prenijeti do dva okvira signalizacije zagušenja prije nego što sljedeći okvir podataka ili udaljeni okvir budu prisutni na sabirnici. Taj okvir zahtijeva od svih čvorova na mreži da odgode slanje sljedećih okvira podataka ili udaljenih okvira čime se tom čvoru daje više vremena za obradu trenutne poruke.

Kako je posebna verzija okvira pogreške, okvir za signalizaciju zagušenja isto sadrži samo dva različita polja međutim drugačijih naziva. Zastava preopterećenja (engl. Overload Flag) i Graničnik preopterećenja (engl. Overload Delimiter). Dvije su vrste stanja preopterećenja koje mogu dovesti do prijena zastave preopterećenja:

- Unutarnji uvjeti prijarnika, koji zahtijevaju kašnjenje sljedećeg okvira podataka ili udaljenog okvira
- Otkrivanje dominantnog bita tijekom prekida

Zastava preopterećenja sastoji se od šest dominantnih bitova. Opći obrazac odgovara obliku aktivne zastave pogreške. Razgraničnik za preopterećenje sastoji se od osam recesivnih bitova. Graničnik preopterećenja istog je oblika kao graničnik pogreške.



## **2.7. Utor za potvrdu primitka**

Utor za potvrdu (acknowledge slot-ACK) koristi se za potvrdu primanja valjanog CAN okvira. Svaki čvor koji prima okvir, bez detekcije greške, prenosi dominantnu razinu u ACK utoru i na taj način nadjačava recesivnu razinu odašiljača. Ako odašiljač otkrije recesivni nivo u utoru ACK, „zna“ da nijedan prijemnik nije pronašao valjani okvir. Primajući čvor može poslati recesiv da naznači da nije primio važeći okvir, ali drugi čvor koji je primio valjani okvir može ga nadjačati s dominantnim. Odašiljajući čvor nema informaciju da su poruku primili svi čvorovi na CAN mreži.

## **2.8. Međuokvirni prostor**

Okviri podataka i udaljeni okviri su odvojeni od predhodećih okvira preko binarnih vrijednosti koje se zove međuokvirni prostor. Međuokvirni prostor sastoji se od najmanje tri uzastopna recesivna (1) bita. Nakon toga, ako se otkrije dominantan bit, smatrat će se bitom "Početak okvira" sljedećeg okvira. Okvirima za signalizaciju zagušenja i pogrešaka ne prethodi međuokvirni prostor, a višestruki okviri za signalizaciju zgušenja nisu odvojeni međuokvirnom prostorom. Međuokvirni prostor sadrži bitove polja s prekidom te obustavlja prijenos za pasivne stanice s pogreškama koje su bile odašiljači prethodnih poruka.

## **2.9. Sigurnosne značajke CAN-a**

CAN je protokol niske razine i ne podržava nijednu sigurnosnu značajku. Također nema enkripcije u standardnim CAN implementacijama čime se mreža ostavlja izložena okvirnim presretanjima između protokola. U većini implementacija očekuje se da će aplikacije primijeniti vlastite sigurnosne mehanizme; npr. za potvrdu dolaznih naredbi ili prisutnosti određenih uređaja u mreži. Neprovođenje primjerenih sigurnosnih mjera može rezultirati raznim vrstama napada ako napadač uspije umetnuti poruke u sabirnicu. Iako postoje lozinke za neke sigurnosno kritične funkcije, kao što su izmjena upravljačkog softvera, programske tipke ili upravljanje pokretačima kočnica protiv blokiranja, ovi se sustavi općenito ne primjenjuju.

## 2.10. Razvojni alati

Pri razvoju ili rješavanju problema CAN sabirnice, ispitivanje hardverskih signala može biti vrlo važno. Logički analizatori i analizatori sabirnice su alati koji prikupljaju, analiziraju, dekodiraju i pohranjuju signale. Na tržištu su priustni specijalizirani alati za analizu CAN sabirnice. Pregledavač CAN sabirnice je analitički alat, često kombinacija hardvera i softvera, koja se koristi tijekom razvoja hardvera koja koristi CAN sabirnicu. Tipični pregledavač CAN sabirnice će „slušati“ promet na CAN sabirnici radi prikaza na korisničkom sučelju. Često pregledavači CAN sabirnice nude mogućnost simulacije aktivnosti CAN sabirnice šaljući CAN okvire u sabirnicu. Pregledavač CAN sabirnice stoga može biti korišten da potvrdi očekivani CAN promet iz danog uređaja ili da simulira CAN promet da potvrdi reakciju iz danog uređaja spojenog na CAN sabirnicu.

## 3. Demonstracija funkcioniranja ABS sustava

Za praktični dio završnog rada demonstriran je način funkcioniranja ABS-a preko CAN-a. Važno je napomenuti da je ABS znatno stariji od CAN-a. Početak ABS-a je 1928. godine kada je Nijemac Karl Wessels patentirao mehanizam za regulaciju sile kočenja u automobilu, međutim ovaj koncept je postojao samo na papiru [5]. Prvi regulator blokiranja kotača testiran je tek 1941. godine te je postigao tek osrednje rezultate. Usprkos tome što je u početku doživio neuspjeh, ovaj je sustav postao konstrukcijska baza za buduće mehanizme. Ideja o sensorima koji prate okretanje kotača i kontrolnoj jedinici koja upravlja kočnicama bila je uspješna, ali pretvoriti koncept u funkcionalan i financijski isplativ mehanizam je predstavljao velik izazov. Problem senzora je riješen 1952. godine u ABS sustavu ugrađenom u avion, ali i u Knorrov sustav za lokomotive u 1954. godini. Ugradnja u automobile još nije bila moguća zato što su postojali zahtjevi koji su se postavljali pred mehanički senzor zbog nepreciznosti i nepouzdanosti u zavojima, na neravnim podlogama ili pri nepovoljnim vanjskim uvjetima.

Razvoj ABS-a je znatno unaprijeđen od strane Robert Bosch GmbH tvrtke kada je 1978. godine na tržište izbacila elektronski sustav, koji se počeo ugrađivati u automobile Mercedes S klase (W116) a nešto kasnije i u BMW 7 (E23) kao dodatna oprema. Prvi automobil u koji se ABS ugrađivao serijski bio je Ford Scorpio 1985. godine.

1981. godine Mercedes Benz je zajedno sa firmom WABCO razvio ABS za vozila sa zračnim kočnicama (teretna vozila i autobusi). U sve autobuse i teretna vozila je obavezna ugradnja ABS-a od 1991. godine.

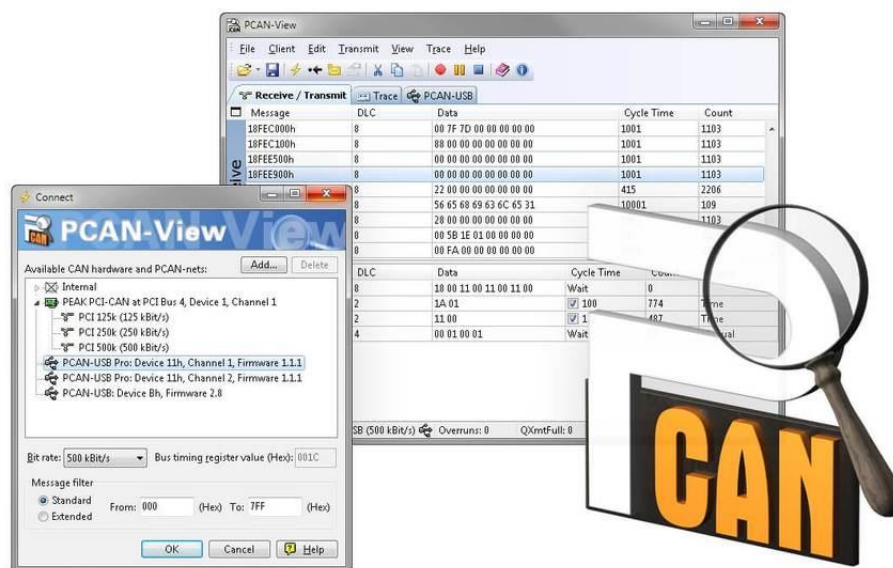
Pošto je ABS veoma kompliciran sustav kojeg čine velik broj signala i komponenata, fokus je na CAN poruci koja sadrži određene ABS podatke, posebno na signal koji mjeri brzinu vozila. Glavna uloga ABS-a jest da spriječi blokiranje kotača prilikom kočenja. Blokiranje kotača onemogućava osobi koja upravlja vozilom zadržavanje smjera i tako dolazi do klizanja vozila i gubljenja kontrole nad istim. ABS poboljšava kontrolu nad vozilom prilikom kočenja na sklizavim i suhim površinama te smanjuje zaustavni put, dok na površinama kao makadam produkuje duljinu kočenja i to bez gubitka kontrole nad vozilom. Međutim ABS nije potreban ukoliko se koči pri maloj brzini na stabilnoj, čvrstoj i suhoj površini, takav sustav bi produžio zaustavni put vozila što bi moglo uzrokovati prometnu nesreću. Iz tog razloga se pred ABS sustav postavlja niz uvjeta koji moraju biti ispunjeni da bi došlo do periodičnog aktiviranja kočnica. Putem Dashboard, PCAN-View i Notebook++ programa postavljeni su, mijenjani i promatrani neki od tih uvjeta, koji omogućavaju neperiodično aktiviranje kočnica pri nižim brzinama.

### **3.1. Korišteni softver i hardver**

Rad ABS-a bit će demonstriran korištenjem nekoliko uređaja i programa te će biti dan kratak uvid u njihov rad. Softver korišten je PCAN-View, Dashboard i Notepad++. Hardver koji je korišten je zapisničar podataka logiRECORDER, PCAN-USB Pro FD te muško-ženski DE9 konektor. Kratko objašnjenje korištenog softvera slijedi.

## PCAN-View

PCAN-View prikazan na slici 4 ispod je jednostavan program za prijenos i nadzor CAN podatkovnog prometa. Poruke se mogu prenijeti ručno i periodično prema brzini koju utvrđuje korisnik (jedna poruka / x milisekundi) te se prenose na željeni izlaz preko sabirnice uređaja. Pogreške sabirnice sustava i prepunjenost memorije u CAN hardveru prikazuju se tijekom postupka prijenosa poruka u stvarnom vremenu. Funkcija praćenja može se koristiti za snimanje i spremanje CAN podataka. PCAN-View softver isporučuje se sa svim hardverskim proizvodima PCAN tvrtke. Sva dostupna sučelja PEAK CAN su navedena u okviru za povezivanje. Nakon odabira hardvera i brzine prijenosa, korisnik može pristupiti svim softverskim funkcijama, hardverskim postavkama i informacijama. PCAN-View će biti korišten da generira i šalje CAN poruke specifičnog ID-a, sadržaja, dužine i brzine.

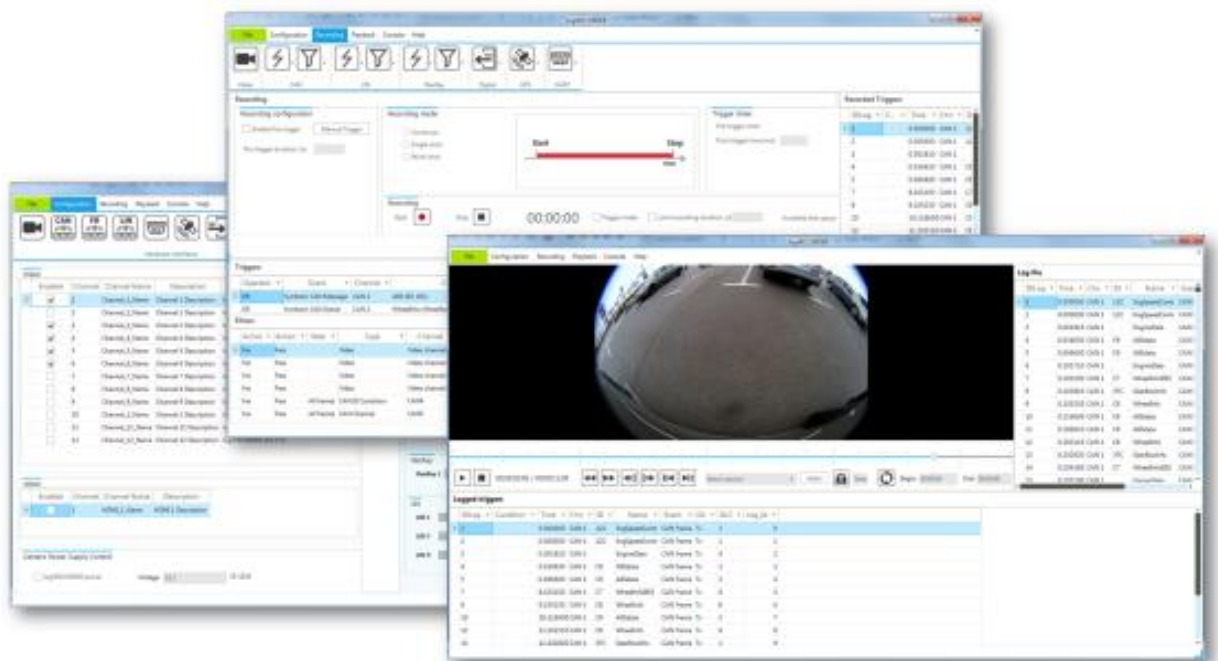


Slika 4: PCAN-View [6]

## Dashboard

Dashboard je PC aplikacija logiRECORDERa zapisničara podataka tj. prikazana ispod na slici 5 koja omogućuje centraliziranu kontrolu zapisničara, te analizu podataka i snimke. Okidači i filteri mogu se postaviti kao signali i poruke iz XML (Extensible Markup Language), ARXML (AUTOSAR Extensible Markup Language), DBC (Decibels Relative to the Carrier), LDF (Log Database File) i FIBEX (Field Bus EXchange) baze podataka vozila. Dashboard se može koristiti za odabir video ulaza za snimanje, prebacivanje između prikaza, kontinuiranog načina snimanja i događaja koji se pokreću određenim radnjama, upravljanje reprodukcijom video zapisa, podešavanjem napajanja itd. Izabrani logiRECORDER prikazi mogu se prikazati u Dashboard aplikaciji i na vanjskom monitoru pričvršćenom na zapisničaru.

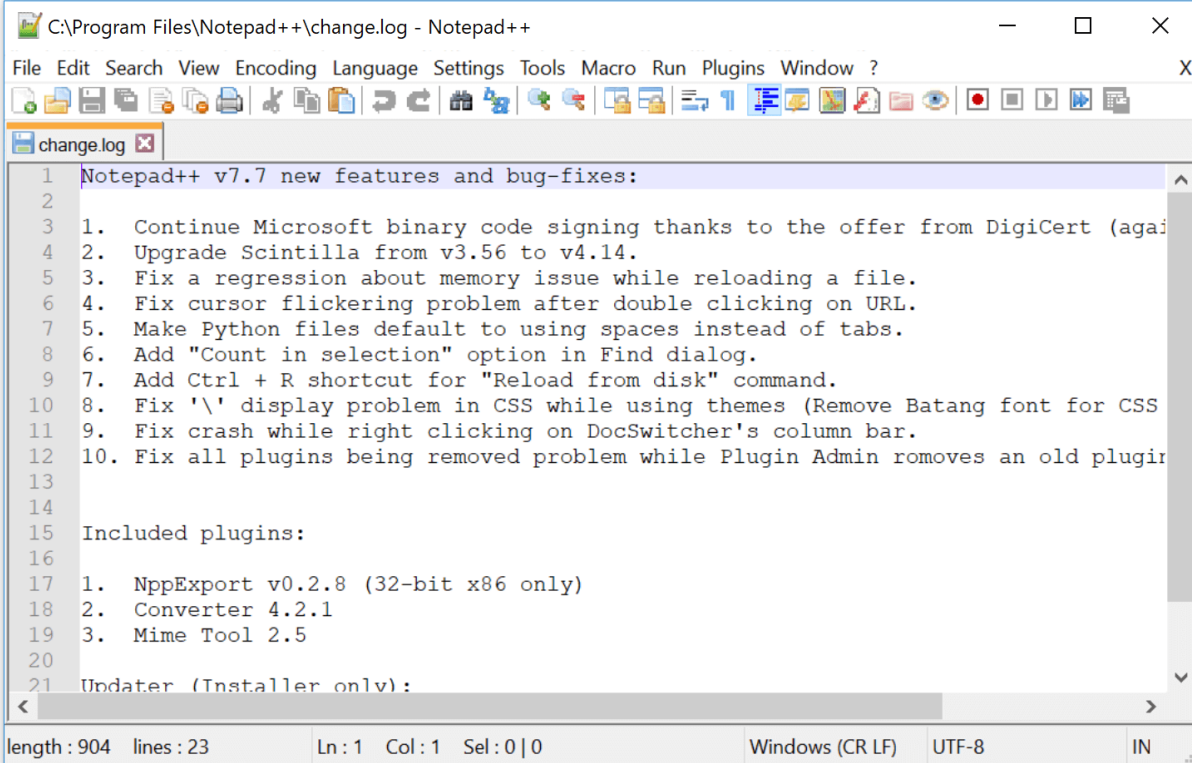
Dashboard će biti korišten da prima i analizira CAN poruke, koristi i demonstrira rad baze podataka, postavlja okidače i filtere, te kao dokaz da su određene vrijednosti CAN poruka primljene.



Slika 5: Dashboard PC aplikacija logiRECORDERa

## Notepad++

Notepad ++ prikazan ispod na slici 6 je besplatni editor izvornog koda i zamjena za aplikaciju Notepad koji podržava nekoliko jezika. U radu se koristi verzija na Microsoft Windows operativni sustav, njegova upotreba određena je GPL (engl. General Public License) licencom (licenca za korištenje za sve svrhe). Notepad ++ napisan je u C ++ jeziku i koristi čisti Win32 API (engl. Application Programming Interface) i STL (engl. Standard Template Library) koji osigurava veću brzinu izvršenja i manju veličinu programa. Notepad++ će se koristiti isključivo za kreiranje podataka potrebna za demonstraciju rada ABS-a putem CAN protokola. Kratko objašnjenje korištenog hardvera slijedi:



The image shows a screenshot of the Notepad++ application window. The title bar reads "C:\Program Files\Notepad++\change.log - Notepad++". The menu bar includes "File", "Edit", "Search", "View", "Encoding", "Language", "Settings", "Tools", "Macro", "Run", "Plugins", and "Window". The toolbar contains various icons for file operations and editing. The main text area displays the following content:

```
1 Notepad++ v7.7 new features and bug-fixes:
2
3 1. Continue Microsoft binary code signing thanks to the offer from DigiCert (agai
4 2. Upgrade Scintilla from v3.56 to v4.14.
5 3. Fix a regression about memory issue while reloading a file.
6 4. Fix cursor flickering problem after double clicking on URL.
7 5. Make Python files default to using spaces instead of tabs.
8 6. Add "Count in selection" option in Find dialog.
9 7. Add Ctrl + R shortcut for "Reload from disk" command.
10 8. Fix '\ ' display problem in CSS while using themes (Remove Batang font for CSS
11 9. Fix crash while right clicking on DocSwitcher's column bar.
12 10. Fix all plugins being removed problem while Plugin Admin removes an old plugin
13
14
15 Included plugins:
16
17 1. NppExport v0.2.8 (32-bit x86 only)
18 2. Converter 4.2.1
19 3. Mime Tool 2.5
20
21 Updater (Installer only):
```

The status bar at the bottom shows "length : 904 lines : 23", "Ln : 1 Col : 1 Sel : 0 | 0", "Windows (CR LF)", "UTF-8", and "IN".

Slika 6: Notepad ++

## Zapisničar podataka LogiRECORDER

Na slici 7 je prikazan zapisničar podataka LogiRECORDER koji povezuje s različitim mrežnim sabirnicama vozila i priključuje se između različitih vrsta kamera vozila i ECU-a u svrhu neinvazivnog snimanja i reprodukcije višekanalnih nekomprimiranih video i mrežnih podataka s niskom latencijom. Upravlja se putem Dashboard programa.



*Slika 7: Zapisničar podataka LogiRECORDER*

## PCAN-USB Pro FD

Pro FD proizvođača PEAK System prikazan je na slici 8. Uređaj omogućava spajanje CAN, CAN FD (engl. Controller Area Network Flexible Data-Rate) i LIN (engl. Local Interconnect Network) mreža sa računalom putem USBa, slanje poruka sa raznim ID-ima, podacima, brzinom itd. Upravlja se putem PCAN-View programa.



*Slika 8: PCAN-USB Pro FD (PEAK System)*

**Muško-ženski DE9 konektor** prikazan na slici 9 prenosi CAN signale između aplikacije PCAN – a i logiRECORDER – a.



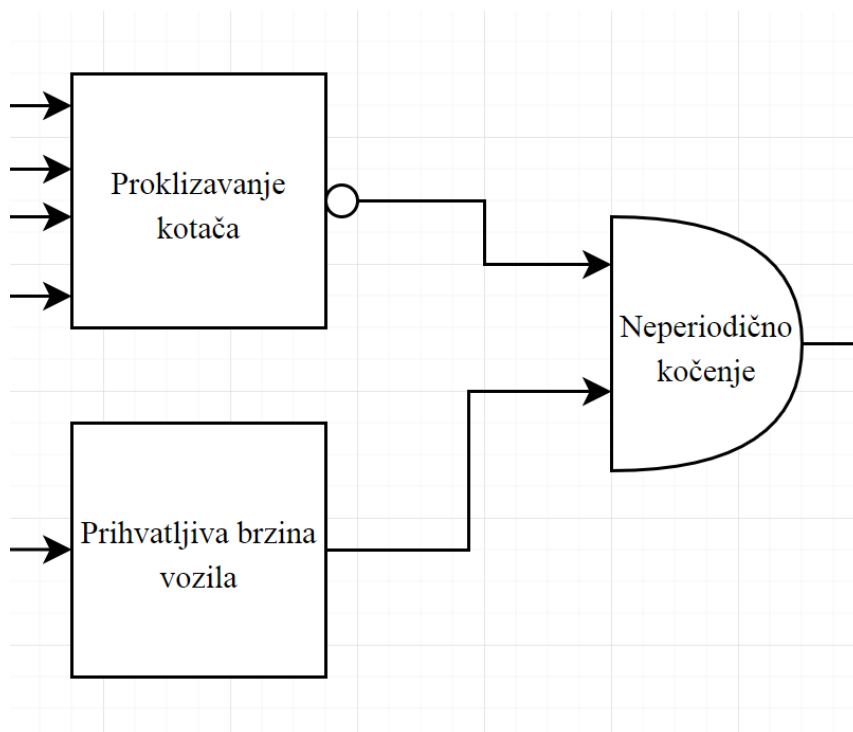
*Slika 9: Muško-ženski DE9 konektor*



### 3.2. Računalna demonstracija kočenja

U ovom potpoglavlju će se demonstrirati rad ABS – a. Specifično uvjet u sustavu koji definira raspon brzine vozila za neperiodično kočenje. Neperiodično kočenje je tzv. "normalno kočenje" gdje se kočnice samo uključe i blokiraju okretanje kotača. Slika 9 prikazuje dijagram toka. Takvo kočenje se u praksi javlja kada ne dolazi do proklizavanja kotača te je brzina vozila između 0-30 km/h. Ta dva uvjeta su spojena na logička I vrata, dakle oba uvjeta moraju biti ispunjena da bi sustav aktivirao neperiodično kočenje.

Sustav koji regulira proklizavanje kotača čini to tako da prima CAN poruke koje sadrže brzinu svakog kotača, te kut pod kojim je volan. Kut volana daje informaciju o tome skreće li vozilo i u kojem smjeru, što bi značilo da je razlika u brzini kotača nužna, jer pri skretanju vanjski kotači pređu veću putanju od unutarnjih tako da vozilo promjeni smjer gibanja. Ako je vozilo u zavoju to znači da je veća razlika u brzini kotača dopuštena. Ako vozilo nije u zavoju ta tolerancija je znatno manja, a ako se brzina kotača ne podudara, sustav procjenjuje da je došlo do proklizavanja vozila. Kad je taj uvjet ispunjen, ne može doći do neperiodičnog kočenja. Kada taj uvjet nije ispunjen, te je brzina vozila u "prihvatljivom" rasponu, neperiodično kočenje je moguće.



Slika 10: Dijagram toka

Zbog ograničenja Dashboard programa, čiji okidači se za sada mogu isključivo postaviti kao logička ILI vrata za postavljene okidače, može se demonstrirati rad samo na jedno od ta dva uvjeta (proklizavanje vozila ili brzina vozila između 0 – 30 km/h) neovisno o drugo, stoga će se demonstrirati podešenje prihvatljive brzine vozila sa pretpostavkom da ne dolazi do proklizavanja kotača.

Uvjet prihvatljive brzine za neperiodično kočenje će biti demonstriran tako da će biti postavljen raspon brzine vozila za neperiodično kočenje, slati će se CAN poruke koje sadrže tri različite brzine vozila, te će sustav reagirati samo na raspon brzine koji je postavljen i aktivirati neperiodično kočenje kada se vozilo kreće isključivo željenom brzinom.

Fizički sloj CAN mreže je standardiziran, međutim softverski ne postoji standardizacija za CAN baze podataka, koje mogu biti raznih oblika, raznih specifikacija te u raznim formatima datoteka. Svaka firma koja se bavi automobilima (a time obavezno i CAN-om) stvara vlastite CAN baze podataka, najčešće u DBC, XML i ARXML formatu. Ove baze podataka omogućuju čitanje zaprimljenih CAN poruka te korištenje podataka iz njih za željene radnje. Za potrebe završnog rada bit će kreirana jednostavna baza podataka u XML formatu pomoću aplikacije Notebook++ naziva Završni\_rad.xml prikazanog na slici 11.

Za demonstraciju rada sustava analiziraju se isključivo CAN ABS poruke. Radi poboljšanja uvida u funkcioniranje baza podataka u praksi kreirati će se nekoliko poruka u Notebook++-u. U praksi se nastoji donekle predvidjeti potrebe projekta ili proizvoda i ostaviti razmak između broja ID-eva za buduće integracije novih sustava koji moraju funkcionirati u bazi. Tako bi se, u ovom slučaju, moglo veoma lako dodati poruka za regulaciju određenog senzora na ID 168 ako su zadnji zauzeti ID-evi 165 i 170.

Primjer visokog ID-a iz prakse bi bio bilokoji sustav čije stanje nije nužno znati tijekom vožnje kao npr. ECU koji regulira visinu stražnjeg desnog prozora u automobilu. Primjer niskog ID-a iz prakse bi bio okvir koji određena firma koristi da zaštiti svoja vozila od krađe, ID je najmanji u bazi i iznosi između tri i petnaest. Ako je vozilo prijavljeno kao ukradeno zaseban ECU prima radio signal te se tako uključuje sustav koji preopteretiti sabirnicu svojim niskim ID-om i daje naredbe GPS (eng. Global Position System, hrv. Globalni sustav za pozicioniranje) modulu za neprestano emitiranje lokacije. Nijedna druga poruka ne može koristiti sabirnicu jer odmah izgubi arbitražu sa porukom tako niskog ID-a stoga se vozilo elektronički zablokira i ne može se koristiti.

```
4 <Name>Zavrsni_rad</Name>
5 <Attribute>
6   <Name>BusType</Name>
7   <Value>CAN</Value>
8   <DefaultUsed>No</DefaultUsed>
9 </Attribute>
10 <Node DB_ID="5">
11   <Name>ABS</Name>
12   <TxMessage DB_ID="6">
13     <Name>ABSdata</Name>
14     <ID>0xC9</ID>
15     <FrameType>CAN Standard</FrameType>
16     <DLC>3</DLC>
17     <Comment></Comment>
18     <Attribute>
19       <Name>CycleTime</Name>
20       <Value>50</Value>
21       <DefaultUsed>No</DefaultUsed>
22     </Attribute>
23     <Signal DB_ID="17">
24       <Name>CarSpeed</Name>
25       <Bitposition>0</Bitposition>
26       <Bitsize>10</Bitsize>
27       <Byteorder>Intel</Byteorder>
28       <Valuetype>Unsigned</Valuetype>
29       <Factor>0.5</Factor>
30       <Offset>0</Offset>
31       <Minimum>0</Minimum>
32       <Maximum>511.5</Maximum>
33       <Unit>km/h</Unit>
34       <Comment></Comment>
35     <Receiver DB_ID="4">
36       <Name>DashBoard</Name>
37     </Receiver>
```

Slika 11: Prikaz datoteke Zavrsni\_rad.xml

U nastavku će biti opisani bitniji dijelovi koda oko kojeg se ovaj rad fokusira:

Redak 13 određuje ime CAN poruke, koja sadrži željeni broj signala u poruci.

Redak 14 je ID te poruke, određen uvijek u heksadekadskom sustavu, u ovom slučaju je C9, što u dekadskom sustavu odgovara broju 201. S razlogom je na visokom broju jer ima sustava u automobilu koji imaju znatno viši prioritet, npr. sustav za kontrolu zračnih jastuka koji ne smije imati niži prioritet od ABS-a.

Redak 15 definira format CAN poruke. Standardni (base) ili prošireni (extended) okvir, standardni okvir koristi 11 identifikacijskih bitova dok prošireni okvir koristi 29. Radi stvaranja poruke u maloj bazi podataka s malim rasponom koristit će se standardni okvir.

Redak 16 definira dužinu poruke, obične CAN poruke mogu biti duljine 1 – 8 bajtova. Tri bajta će dati dovoljan raspon za prikazivanje realistične brzine sa faktorom (varijablom koja se množi sa data vrijednosti signala u poruci) od 0.5. Precizna fizička vrijednost signala ovisi o faktoru, jer se podešavanjem istog može proširiti ili suziti raspon vrijednosti signala. Faktor 0.5 omogućuje da data vrijednost raspona 0-1000 daje pravu, fizičku vrijednost signala 0-500, međutim fizička vrijednost signala onda može biti izražena u polovinama (npr. 190.5 km/h).

Redak 24 određuje ime glavnog signala koji je predmet interesa, CarSpeed signal će se pratiti te će se postvaljati okidači koji će pokrenuti određene radnje kada signal postigne željene vrijednosti.

Redak 25 određuje poziciju prvog bita ovog signala u ABS data poruci.

Redak 26 određuje veličinu signala u bajtovima, što više bitova signal sadrži, može imati veći raspon ili preciznija mjerenja, ovisno o faktoru koji se koristi. Naravno poruka mora sadržavati onoliko bitova koji su suma njezinih signala.

Redak 27 određuje poredak bajtova. Računala skladište informacije u grupama binarnih bitova raznih veličina. Svakoj grupi je dodijeljen broj koji djeluje kao njezina adresa, koju računalo koristi da bi pristupilo toj informaciji. Postoje dva načina koja računala koriste da numeriraju pojedine bajtove u grupi, počevši od bilo kojeg kraja. Little endian format se često (pa tako i u ovom slučaju) naziva Intel byte order. Dok se big endian format naziva Motorola. Intel byte order sprema najmanje bitni bajt na najmanjoj adresi. Razlika između big endiana i little endiana je u tome što se u big endian formatu najznačajniji bit skladišti prvo (na najnižu skladišnu adresu) dok se u little endian formatu najmanje značajan bit skladišti prvo [12].

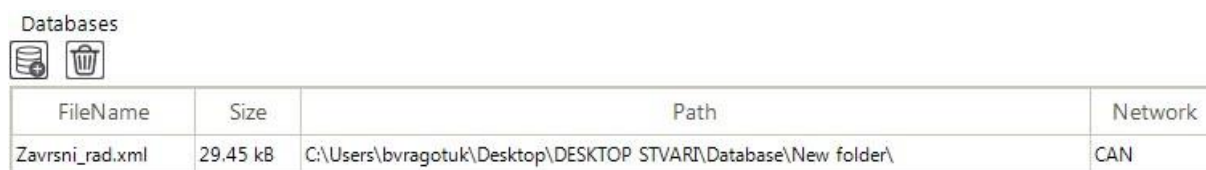
Redak 29 određuje faktor koji se množi sa signalnim podacima da bi postigao realnu vrijednost koja nas zanima.

Redak 30 određuje otklon koji se koristi za postizanje početne vrijednosti drugačije od 0. Npr. pri mjerenju temperature u Kelvinima, offset će biti -273,15. Na taj način se mogu i dalje koristiti cijelobrojne pozitivne vrijednosti za dobivanje negativnih vrijednosti.

Redci 31 i 32 određuju minimum i maksimum realne (fizičke) vrijednosti ovog signala.

Redak 33 određuje mjernu jedinicu u kojoj je signal bio mjereno.

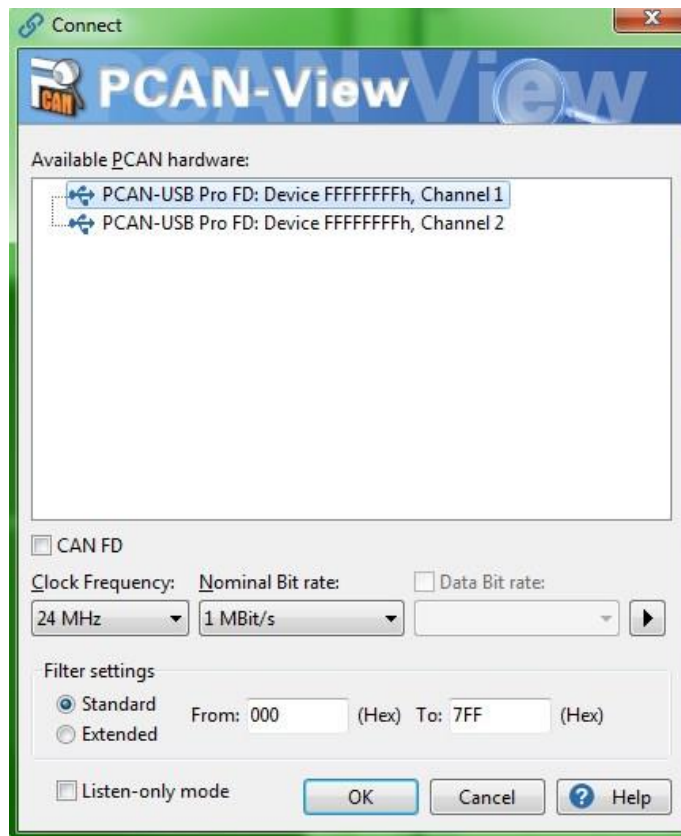
Ova će bazu biti pohranjena i transferirana u Dashboard aplikaciju kako bi je logiRECORDER mogao koristiti prikazano na slici ispod.



FileName	Size	Path	Network
Zavrsni_rad.xml	29.45 kB	C:\Users\bvragotuk\Desktop\DESKTOP_STVARI\Database\New folder\'	CAN

*Slika 12: Spremljeni Zavrsni\_rad.xml transferiran u Dashboard*

Nakon kreiranja baze podataka, mogu se slati poruke i signali koji se očekuju. Za to će se koristiti PCAN-USB Pro FD uređaj te njegov PCAN-View program prikazan na slici 13. LogiRECORDER i PCAN-USB Pro FD će biti spojeni putem muško-ženskog DE9 kabela i obavezno biti postavljeni na istoj nominalnoj stopi bitova (Bit rate).



Slika 2: Aplikacija PCAN-View

U Dashboard aplikaciji omogućimo rad CAN modula te podesimo Baud rate i Sample point kao što je prikazano na slici ispod:

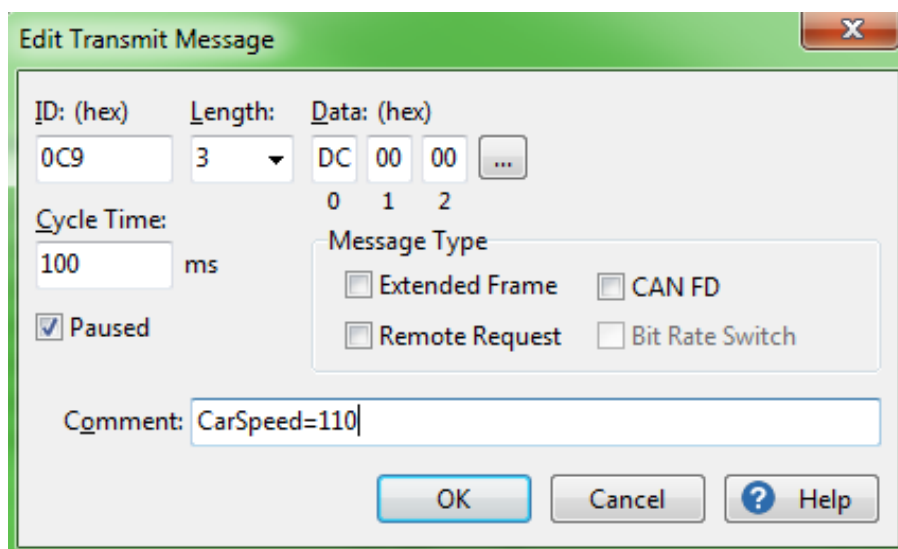
CAN Channels							
En	LR	Ch	Name	Description	CANFD	Baud rate	Sample Point
<input type="checkbox"/>	LR1	1	CANFD1		<input type="checkbox"/>	1.000.000 Bd	75
<input type="checkbox"/>	LR1	2	CANFD2		<input type="checkbox"/>	1.000.000 Bd	75
<input type="checkbox"/>	LR1	3	CANFD3		<input type="checkbox"/>	1.000.000 Bd	75
<input checked="" type="checkbox"/>	LR1	4	CANFD4		<input type="checkbox"/>	1.000.000 Bd	75
<input type="checkbox"/>	LR1	5	CANFD5		<input type="checkbox"/>	1.000.000 Bd	75
<input type="checkbox"/>	LR1	6	CANFD6		<input type="checkbox"/>	1.000.000 Bd	75

Slika 14: Uključeni i podešeni CAN modul u Dashboard aplikaciji

U praksi se uvjet za neperiodično kočenje pokreće unutar vrijednosti od 0 do 20-30 km/h, međutim u ovom radu taj uvjet je postavljen na vrijednost od 222 do 321 km/h u svrhu demonstriranja fleksibilnosti sustava.

U PCAN - View programu stvara se CAN poruka koja će sadržavati potrebne informacije za računalnu demonstraciju kočenja. CAN poruka u pitanju je okvir podataka.

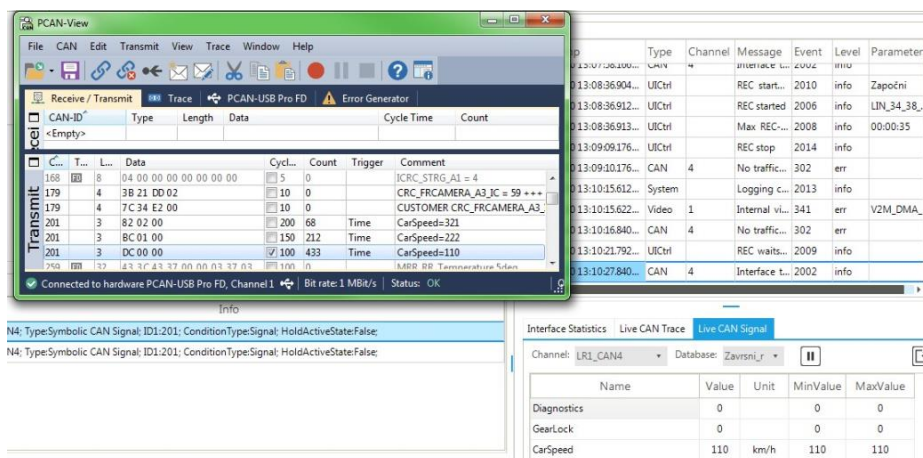
Pod polje ID upisan je ID poruke u heksadekadskom zapisu, dok polje length odgovara duljini CAN poruke te je duljine tri bajta kao što je prikazano na slici 15. Vrijednost polja Cycle Time je u ovom slučaju nebitna jer u bazi nisu određene stroge zadane vrijednosti. Već iz data vrijednosti se može odrediti prava fizička vrijednost CarSpeed signala prema bazi, jer CAN nosi podatke na određenim lokacijama u poruci, iz kojih onda Dashboard protumači o kojem signalu je riječ te koja je vrijednost tog signala. Bitposition signala je broj bita u CAN poruci gdje se vrijednost za željeni signal nalazi, u ovom slučaju to je 0. Heksadekadski broj DC odgovara broju 220 u dekadskom sustavu, međutim faktor koji je unesen u bazu je 0.5, stoga prava fizička vrijednost CarSpeed signala iznosi 110.



*Slika 15: Uređivanje transmitirane poruke*

Slanjem poruke na logiRECORDER potvrđuje se točnost te tvrdnje u Dashboard aplikaciji, u kojoj je transferirana Završni\_rad.xml baza podataka kako vidimo na slici 16. Radi se o Live

CAN Signal kartici koja očitava dolazeće poruke tj. vrijednosti signala u tim porukama. Vidljivo je da vrijednost CarSpeed signala iznosi 110.

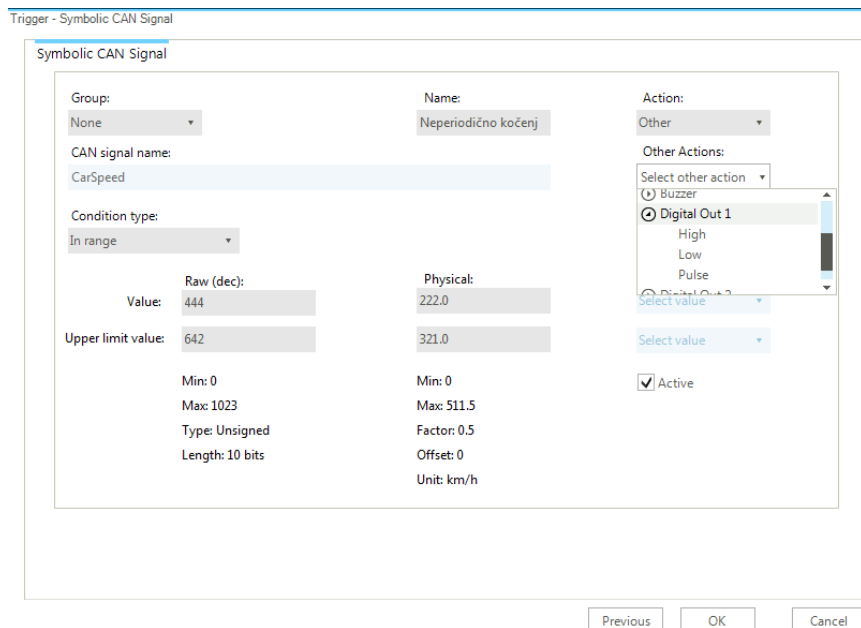


Slika 16: Provjera točnosti tvrdnje, vrijednost CarSpeed signala iznosi 110

Kako je već rečeno, pri povoljnim uvjetima i niskoj brzini periodično kočenje nije nužno, štoviše uzrokuje duži zaustavni put koji, u okolici gdje su ti uvjeti najčešće ispunjeni (uske gradske ulice) može izazvati prometnu nesreću. Stoga za demonstraciju fleksibilnosti softverskog rješenja problema umjesto raspona od 0 do 30 km/h kako je inače postavljen sustav u praksi za neperiodično kočenje, postavljen je raspon od 220 do 321 km/h za neperiodično kočenje.

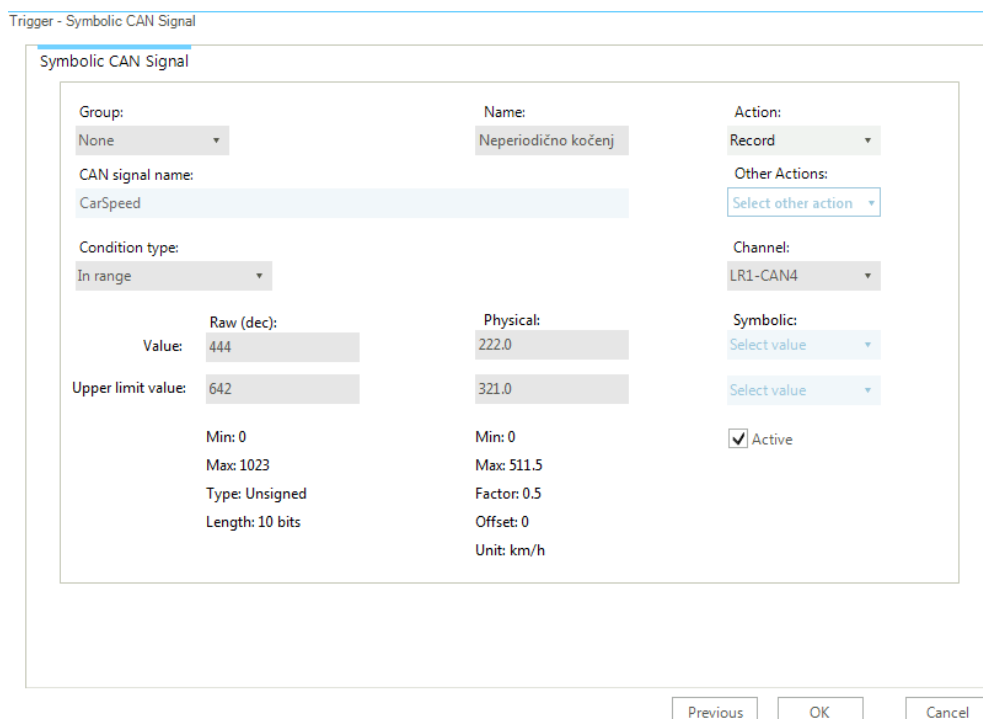
Slika 16 prikazuje Dashboard aplikaciju koja omogućuje slanje digitalnog signala na izlaz logiRECORDERa putem kojeg se može uživo osciloskopom demonstrirati da je postavljeni uvjet ispunjen te da je signal poslan. Pod Action opcijom se odabere "Other" te pod "Other Actions" željeni signal na "Digital Out 1". Time zapisničar daje željeni signal na izlazu u slučaju da je uvjet koji smo postavili ispunjen tj. u slučaju da je brzina vozila između 220 i 321 km/h.





Slika 17: Okidač može slati jednostavni digitalni signal na jedan od tri outputa

Međutim pri odabiru "Record" opcije iz "Action" izbornika, izabrana radnja je početak snimanja videa, koji softverski dokazuje ispunjenje postavljenog uvjeta kako vidimo na slici 18.



Slika 18: Okidač za početak snimanja sesije

Postavljanjem još jednog uvjeta koji zaustavlja snimanje demonstriran je raspon periodičnog kočenja prikazan na slici 19.

Trigger - Symbolic CAN Signal

**Symbolic CAN Signal**

Group: None	Name: Periodično kočenje	Action: Stop record
CAN signal name: CarSpeed		Other Actions: Select other action
Condition type: Out of range		Channel: LR1-CAN4
Value: 444	Raw (dec): 444	Physical: 222.0
Upper limit value: 642	642	321.0
Min: 0	Min: 0	<input checked="" type="checkbox"/> Active
Max: 1023	Max: 511.5	
Type: Unsigned	Factor: 0.5	
Length: 10 bits	Offset: 0	
	Unit: km/h	

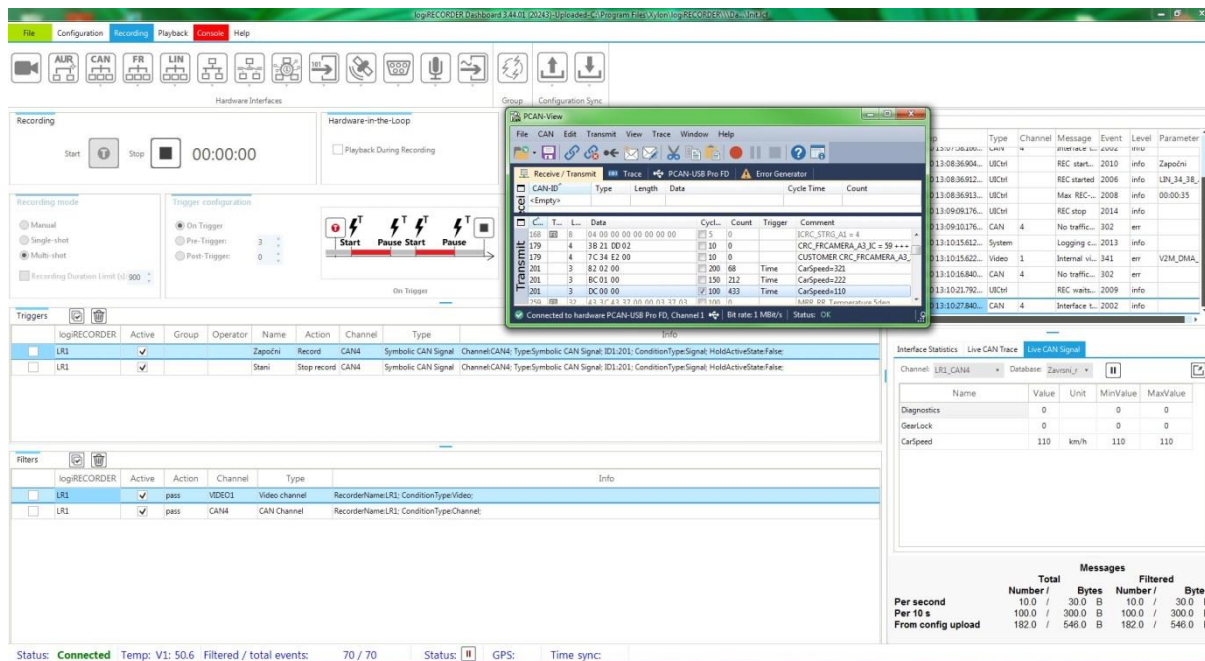
Slika 19: Okidač za zaustavljanje snimanja sesije

Uvjeti tj. okidači su postavljeni, korištene su tri vrijednosti ABS data poruke. Svaka od njih sadrži vrijednost signala koji je opisan u "komentaru" tj. desnoj strani slike prikazanoj na slici 20.

201		3	82 02 00	<input type="checkbox"/> 200	110	Time	CarSpeed=321
201		3	BC 01 00	<input type="checkbox"/> 150	364	Time	CarSpeed=222
201		3	DC 00 00	<input type="checkbox"/> 100	157	Time	CarSpeed=110

Slika 20: ABS data poruke i njihove vrijednosti

Kao što je vidljivo na slici 21, vrijednosti CarSpeed signala se kreću od 110 do 321 km/h. Okidači koji su postavljeni će početi snimati sesiju tek kada se vrijednost tog signala kreće od 222 do 321 km/h.

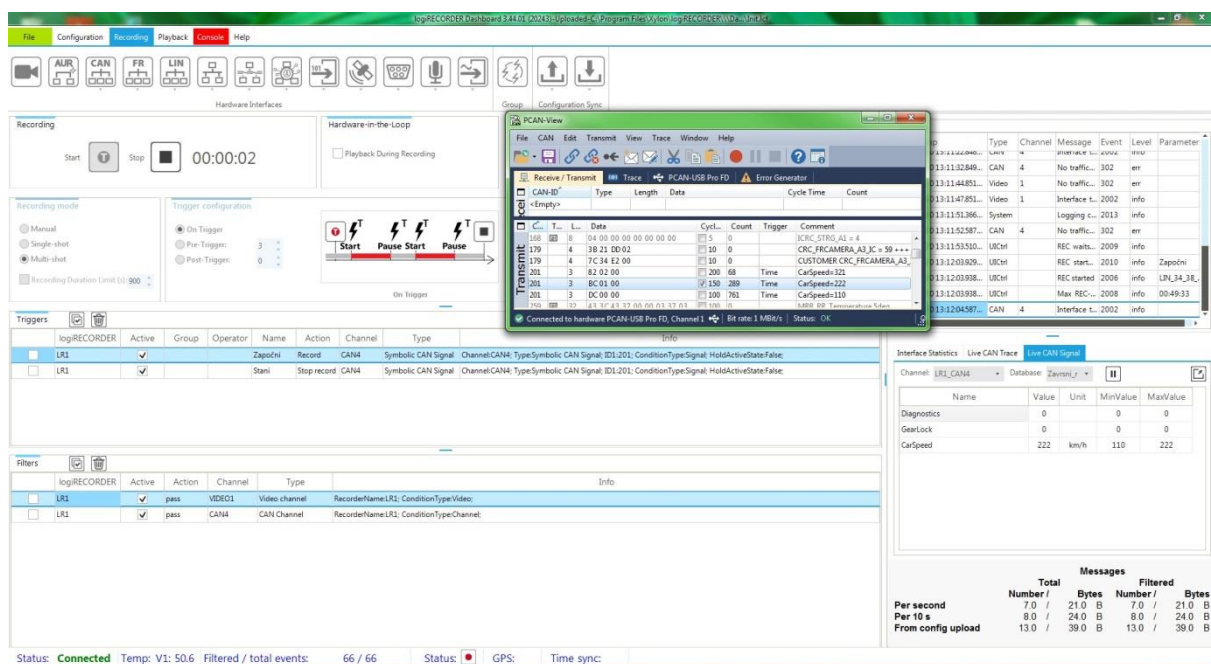


Slika 21: Slanje CAN poruke čija vrijednost signala ne aktivira okidač

Slanjem poruke sa CarSpeed vrijednosti izvan tog raspona snimanje ne počinje, poruke su zaprimljene i uređaj ih jasno vidi. Vrijednost CarSpeed signala je izračunata putem Završni\_rad.xml baze podataka koja je transferirana na uređaj.

Međutim, istog trena kada uređaj zaprimi poruku čija vrijednost CarSpeed signala odgovara okidaču koji je postavljen, ispunjava zadanu dužnost te započinje snimanje sesije kako je predhodno definirano.

U donjem desnom kutu prozora aplikacije prikazanog na slici 22. vidljiva je trenutna vrijednost signala te se lako može potvrditi da okidač koji je postavljen funkcionira.



Slika 22: Slanje CAN poruke čija vrijednost signala aktivira okidač

## 4. Zaključak

Kroz godine razne automobilske komunikacijske mreže su razvijene u pokušaju da konkuriraju ili zamjene CAN. Neke od tih mreža su sigurnije i učinkovitije, međutim CAN je ostao industrijski standard uvelike iz razloga što je prva komunikacijska mreža razvijena za automobile. Njegova pouzdanost i učinkovitost su i dalje adekvatni za izazove modernog tržišta. Demonstrirana je ključna karakteristika softverskog rješavanja problema koja često prođe neprimjećena, a to je fleksibilnost sustava. Prebacivanje kontrole ABS-a iz isključivo mehaničkog u električni sustav omogućava proizvođaču automobila slobodu pri određivanju parametara za svaki model i marku vozila koje želi. Dok su se sa mehaničkim kontroliranim ABS-om parametri morali unaprijed podesiti u pojedine komponente tj. odrediti nužne vrijednosti svake mehaničke komponente da se postignu željeni rezultati, CAN pruži mogućnost brze prilagodbe sustava putem računala spojenog na mrežu. To je samo jedna od mnogih prednosti, uz CAN razni drugi uvjeti i sustavi mogu biti postavljeni ili modificirani koji u mehaničkom sustavu naprosto nebi bili mogući. To je dokazano postavljanjem okidača i uvjeta za neperiodično kočenje koje je u praksi 0-30 km/h međutim, u ovom je radu raspon brzine postavljen od 220-321 km/h. Slanjem CAN poruka raznih vrijednosti CarSpeed signala dokazano je da će okidač reagirati samo kada je vrijednost tog signala u rasponu kojeg smo postavili, dok će sve ostale vrijednosti tog signala biti zanemarene. Za samo par sekundi se može taj uvjet podesiti za bilo koji drugi željeni raspon.

## 5. Literatura

[1] 5 Advantages of CAN bus protocol, dostupno na:

<https://www.totalphase.com/blog/2019/08/5-advantages-of-can-bus-protocol/>

pristupljeno 6.6.2020.

[2] Analytical and Experimental Performance Evaluations of CAN-FD Bus, dostupno na:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8338047>

pristupljeno 10.6.2020.

[3] CAN BUS PROTOCOL, dostupno na:

<https://store.chipkin.com/articles/can-bus-protocol-10-minute-lesson>

pristupljeno 4.7.2020.

[4] Road vehicles — Controller area network (CAN), dostupno na:

<https://www.iso.org/obp/ui/#iso:std:iso:11898:-1:ed-2:v1:en>

pristupljeno 13.7.2020.

[5] Lawes J.: Car Brakes, Crowood, 2014.

[6] PCAN-View description, dostupno na:

<https://www.peak-system.com/PCAN-View.242.0.html?&L=1>

pristupljeno 15.7.2020.

[7] History of CAN technology, dostupno na:

<https://www.can-cia.org/can-knowledge/can/can-history/>

pristupljeno 19.7.2020.

[8] Mercedes W140: First car with CAN, dostupno na:

[https://can-newsletter.org/engineering/applications/160322\\_25th-anniversary-mercedes-w140-first-car-with-can/](https://can-newsletter.org/engineering/applications/160322_25th-anniversary-mercedes-w140-first-car-with-can/)

pristupljeno 19.12.2020.

[9] A CAN Physical Layer Discussion, dostupno na:

<http://ww1.microchip.com/downloads/en/appnotes/00228a.pdf>

pristupljeno 24.8.2020.

[10] CAN with Flexible Data-Rate, dostupno na:

<https://can-newsletter.org/assets/files/ttmedia/raw/e5740b7b5781b8960f55efcc2b93edf8.pdf>

pristupljeno 3.9.2020.

[11] Introduction to the Controller Area Network (CAN), dostupno na:

[https://www.ti.com/lit/an/sloa101b/sloa101b.pdf?ts=1607317487585&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/an/sloa101b/sloa101b.pdf?ts=1607317487585&ref_url=https%253A%252F%252Fwww.google.com%252F)

pristupljeno 8.9.2020.

[12] Assigned numbers, dostupno na:

<https://tools.ietf.org/html/rfc1700>

pristupljeno 14.9.2020.

## 6. Prilozi

### 6.1 Popis slika

Slika 1: Tip 9-pin muški (s lijeva) i ženski (s desna) konektor .....	7
Slika 2: Grafički prikaz osnovnog i proširenog okvira podataka .....	11
Slika 3: Grafički prikaz udaljenog okvira .....	13
Slika 4: PCAN-View .....	18
Slika 5: Dashboard PC aplikacija logiRECORDERa .....	19
Slika 6: Notepad ++ .....	20
Slika 7: Zapisničar podataka LogiRECORDER .....	21
Slika 8: PCAN-USB Pro FD (PEAK System) .....	22
Slika 9: Muško-ženski DE9 konektor .....	22
Slika 10: Dijagram toka .....	23
Slika 11: Prikaz datoteke Zavrzni_rad.xml .....	25
Slika 12: Spremljeni Zavrzni_rad.xml transferiran u Dashboard .....	27
Slika 13: Aplikacija PCAN-View .....	28
Slika 14: Uključeni i podešeni CAN modul u Dashboard aplikaciji .....	28
Slika 15: Uređivanje transmitirane poruke .....	29
Slika 16: Provjera točnosti tvrdnje, vrijednost CarSpeed signala iznosi 110 .....	30
Slika 17: Okidač može slati jednostavni digitalni signal na jedan od tri outputa .....	31
Slika 18: Okidač za početak snimanja sesije .....	31
Slika 19: Okidač za zaustavljanje snimanja sesije .....	32
Slika 20: ABS data poruke i njihove vrijednosti .....	32
Slika 21: Slanje CAN poruke čija vrijednost signala ne aktivira okidač .....	33
Slika 22: Slanje CAN poruke čija vrijednost signala aktivira okidač .....	34



## 6.2 Popis tablica

Tablica 1: 11-bitna ID CAN mreža s dva čvora s ID-om 15 i 16.....	5
Tablica 2: Osnovni okvirni format .....	10
Tablica 3: Prošireni okvirni format .....	12