Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.38
DOI: https://doi.org/10.24867/27IH03Dobrijevic

ISPITIVANJE CAN MEHANIZMA UPRAVLJANJA GREŠKAMA PRIMENOM METODE FAULT INSERTION U AUTOMOBILSKOJ INDUSTRIJI

TESTING CAN ERROR CONFINEMENT MECHANISM USING FAULT INSERTION METHOD IN AUTOMOTIVE INDUSTRY

Nikola Dobrijević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA, ROBOTIKA I AUTOMATIZACIJA

Kratak sadržaj – *Predmet ovog rada predstavlja opis, primenu i prikaz rezultata testa koji analizira stanje uređaja tokom i nakon kritične greške „BusOff" detektovane od strane CAN transivera. U radu se opisuje uloga testiranja u automobilskoj industriji, CAN protokol, mehanizmi za detektovanje greške, test slučaj, test okruženje, plan napada i praktična izvedba testa sa rezultatima*

Ključne reči: *Industrijske komunikacione mreže i protokoli, CAN protokol, Metoda ubacivanja grešaka, Automobilska industrija*

Abstract – *The main part of this paper presents the description, application and display of test results that analyze the state of the device during and after the critical "BusOff" error detected by the CAN transceiver. The paper describes the role of testing in the automotive industry, CAN protocol, error detection mechanisms, test case, test environment, plan of attack and practical execution of the test with results.*

Keywords: *Industrial communication networks and protocols, CAN protocol, Fault Insertion Method, Automotive industry*

1. UVOD

Pojam sigurnosti je tradicionalno jedan od najvažnijih atributa u sklopu automobilske industrije koji se neprestano unapređuje u skladu sa rastom i razvojem kompjuterske tehnike i tehnologije. Zbog sigurnosti i pouzdanosti pojedinačnog uređaja u sklopu automobila, posebna pažnja je posvećena ovoj grani, kroz razvoj i testiranje uređaja koji čine jednu funkcionalnu celinu. Testiranje je proces fokusiran ka cilju da se pronađu greške u sistemu (IEEE 829 Definition). Da bi razumeli važnost testiranja i metode prvo treba napomenuti da je broj ECU-a dostigao preko stotinu u zavisnosti od klase automobila. Tolika grupa uređaja sadrži preko 2000 funkcija, a komunikacija se odvija na više različitih magistrala. Trend ne-autorizovanom pristupu podacima, poznatijem kao hakovanjem (cyber attack), je porastao a sa njim i odbrambeni mehanizmi. Jedni od najpoznatijih su MAC (Message Autentification Code) i IDS (Intrusion Detection.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bila dr Gordana Ostojić, red. prof.

Systems) sistemi koji povećavaju sigurnost mreže unutar automobilskog sistema. Međutim, oni delimično obuhvataju ranjivosti celokupne komunikacione mreže. Mnogo veća pretnja, u praksi, odvija se na nivou fizičke arhitekture i zahteva primenu protokola.

Mehanizam upravljanja greškama (CAN Fault Confinement Mechanism) se upravo odvija na nivou bita i smatra se jednim od glavnih atributa u robusnosti CAN (Controller Area Network) protokola, a definisan je standardom ISO 11898 (International Organization for Standardization).

Izazov koji predstoji jeste pažljivo kreiranje sekvence napada i pokušaj da se metodom umetanja greške izazove predviđeno ponašanje DUT-a prema sistemskim sahtevima.

2. CAN PROTOKOL

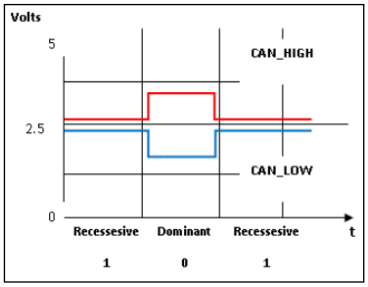
CAN protokol je razvijen od strane kompanije Robert Bosch GmbH, 1982. kao multi-master, sistem za emitovanje poruka maksimalnom brzinom od 1 megabit po sekundi (Mbit/s). Razlozi velike primene i popularnosti su pouzdanost, performanse protokola, ogromna fleksibilnost pri dizajniranju i unapređivanju sistema, kao i njegova cena.

Međutim njegovu popularnost je dodatno pospešila visoka imunost na električne interferencije kao i sposobnost samo-diagnostikovanja i ispravljanja u slučaju greške nastale pri slanju podataka. CAN magistrala je serijska magistrala koja podatke emituje istovremeno (Bidirectional Half Duplex) ka svim svojim periferijama – čvorovima (Node).

Fizička arhitektura sastoji se od para upletene bakarne žice visokog CAN-H (High) i niskog CAN-L (Low) napona. Vrednost logičkih stanja 0 i 1 određena je razlikom naponskih nivoa (Slika 1), gde 0 predstavlja dominantno stanje u kome čvor pravi naponsku razliku ((CANH – CANL) ≥ 1,5 V) i dovodi magistralu u dominantno stanje. Logično stanje 1, je ujedno i stanje mirovanja (Idle) magistrale, gde je naponska razlika približna nuli.

2.1. Detektovanje greške na CAN magistrali

Kako se poruke na CAN magistrali "dele" svim čvorovima neophodno je uspostaviti konzistenciju podataka u okviru odluka koje ECU-ovi donose. Postoje četiri različita tipa poruka (ramova) shodno njihovom sadržaju i nameni:



Slika 1. *Naponski nivoi za CAN HS*

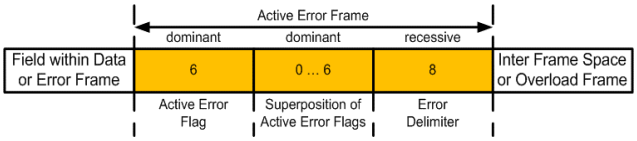
• DATA FRAME – Poruka sa sadržajem koju treba distribuirati jednom ili više čvorova.

• REMOTE FRAME – Upitna poruka koja traži poruku sa sadržajem sa istim identifikatorom.

• ERROR FRAME – Čvor šalje ovaj tip poruke ukoliko detektuje grešku.

• OVERLOAD FRAME – Služi za kontrolu toka poruka tako što zahteva dodatno vreme čekanja, pre nego što se DATA ili REMOTE poruka pošalje.

Dva tipa rama su od značaja za ovaj rad, ram sa podacima i ram greške. Ram sa podacima će biti meta napada, dok će se ram greške posmatrati kroz analizu detekcije greške na CAN magistrali.
Protokol poseduje pet metoda detekcije greške od kojih su dva na nivou bita, a tri na nivou poruke: Bit Error, Stuff error, CRC error, ACK Error, Form Error. Prema planu i realizaciji napada u poglavlju 4, cilj i jeste kreiranje Stuff Error greške i ona će primarno biti prisutna tokom izvršavanja sekvence napada. Ostali čvorovi će takođe koristiti Stuff Error za signalizaciju greške na magistrali sto će dovesti do superpozicije dominantnih bitova (Slika 2).



Slika 2. *Superpozicija*

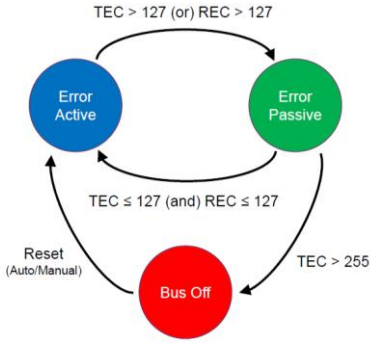
Ova pojava je specifičnost mehanizma upravljanja greškama tokom aktivnog rada transivera. Superpozicija će izmeniti oblik rama greške dodavanjem dominantnih bitova u deo zastavice aktivne greške.

2.2. Mehanizam upravljanja greškama

Cilj mehanizma upravljanja greškama jeste da očuva funkcionalnost i dostupnost magistrale čak i u slučaju neispravnog čvora. Zato se strategija detektovanja greške zasniva na detektovanju razlika između privremene i trajne greške, i isključivanju ugroženog čvora. Postoji niz pravila [1] koja moraju biti ispoštovana prilikom detektovanja greške.

Svaki kontroler ima dva nezavisna brojača: TEC (Transmit Error Counter) broji greške detektovane na čvoru prilikom slanja poruke; REC (Receive Error Counter) broji greške detektovane na čvoru prilikom primanja poruke. Detektovanjem bilo koje greške, čvor šalje Error Frame na magistralu i menja vrednosti nekog od ova dva brojača, u zavisnosti gde je greška detektovana.

Čvor, odnosno transiver, kada detektuje grešku tokom slanja poruke povećava TEC brojač za 8 sa bilo kojom detektovanom metodom greške. U slučaju detektovanja greške prilikom primanja poruke, povećava se REC brojač za 1. Međutim, metodama Bit Error, Stuff Error i detektovovanim Error Frame-om REC brojač se povećava za 8. Sva slanja i primanja koja se izvrše bez greške smanjuju brojače za 1. Da bi se greške lokalizovale i njima upravljalo, svaki ECU mora da bude u jednom od tri stanja greške (Error Active, Error Passive, Bus Off) prikazanih na slici 3.



Error! No text of specified style in document. 3. *Dijagram stanja kontrolera*

Tabela 1. *Test slučaj*

Precondition	Procedure	Expected Results
Battery.**SetVoltage**(13) ECU.**Set**(Operating Mode = Awake) CAN.**Set**(Msg = Command_for_Output_1, Val = 0) ECU.**Monitor**(Signal =OUTPUT_SAFETY_1, State = Gnd, Time = 1000)#ms	#1. CAN.**Set**(State = BusOFF, Status = Active) Time.**Wait**(20) #2. CAN.**Set**(State = BusOFF, Status = Inactive); #3. DIAG.**SendRequest**(19 02 0C);	#1. ECU.**Monitor**(Signal =OUTPUT_SAFETY_1, State = Vbatt, Time = 14000)#ms CAN.**CheckCommunication**(Inactive); #2. CAN.**CheckCommunication**(Active); ECU.**Monitor**(Signal =OUTPUT_SAFETY_1, State = Gnd, Time = 1000)#ms #3. DIAG.**CheckResponse**(Type = ReadbyMask, Value = DF9F31, Status = 2C);

3. TEST SLUČAJ

Osnovni atributi koji opisuju scenario su:

- preduslovi (Precondition),
- procedure (Procedure) i
- očekivani rezultati (Expected Results).

U tabeli1 vidi se primer testa kreiranog za ovaj rad i svi koraci za validiranje OEM zahteva. Od najvišeg značaja je prvi korak (Tabela 1 - Korak #1.) iz procedure i njegov očekivani rezultat. Unutar prvog koraka se izvršava uzurpiranje mreže, a implementacija ovog koraka će biti detaljno opisana u poglavlju 4.

Prvi korak test slučaja zapravo predstavlja metodu namernog ubacivanja greške u sistem (Fault Insertion Method). U daljem tekstu, reč „napad“ se odnosi na uzurpiranje poruke poslate od strane DUT-a. Prvi preduslov (Tabela 1 – Precondition) za ovaj test zahteva da se na ulazne periferije napajanja uređaja dovede napon od 13 V, što je i optimalna vrednost u toku normalnog rada. Drugi preduslov definiše operativno stanje ECU-a, a u ovom slučaju je to „budno” (Awake) stanje u kome komunikacija i sve funkcije uređaja rade bez ograničenja. Treći preduslov je da se putem simuliranog okruženja preko CAN-a pošalje komanda za deaktivaciju izlazne periferije OUTPUT_SAFETY_1 i proveri da je njena vrednost jednaka 0 V.

Unutar prvog koraka se konfiguriše i aktivira hardverski uređaj VH6501 [2] koji šalje prethodno definisane sekvence na CAN magistralu i time počinje uzurpiranje ECU-a. U očekivanom rezultatu, prema specifikaciji OEM-a, OUTPUT_SAFETY_1 ima zadatak da se aktivira ukoliko je greška BusOff detektovana. Ovo ponašanje je tipičan primer sigurnosnog cilja (Safety Goal) usled gubitka komunikacije uređaja sa ostatkom automobila. Odašiljač ECU-a na kome se vrši testiranje ima mehanizam za automatski oporavak iz BusOff stanja pa je čekanje od 20 ms i više nego dovoljno da se detektuje odsustvo ECU-a sa magistrale.

Drugi korak obustavlja napad i proverava da li je ispitivani uređaj ponovo uspostavio komunikaciju na magistrali. Takođe proverava da li je izlaz OUTPUT_SAFETY_1 povratio svoju inicijalnu vrednost iz preduslova testa. Ovo ponašanje je očekivano obzirom da su informacije o mreži ponovo dostupne, a uređaj treba da prođe kroz “oporavak” i dovede svoje izlaze na traženo stanje.

Treći korak proverava postojanje greške putem diagnostike. Kada je greška detektovana, poseban protokol UDS (Unified Diagnistic Services - ISO 14229) nalaže da se informacija o trenutnoj ili već postojećoj grešci sačuva u dodeljenom delu memorije DUT-a.

4. PLAN, REALIZACIJA I REZULTAT NAPADA

Prema zahtevima proizvođača, grešku BusOff, ECU treba da prepozna nakon 16 uzastopnih pokušaja reaktivacije (Restart) i sačuva u vidu DTC-a. ECU nad kojim se vrši napad, određene poruke šalje po događaju dok češći slučaj je da poruke šalje ciklično. Prve tri posmatrane poruke na ECU-u se šalju brzinom od 10 ms ciklično u skladu sa prioritetom. U ovom radu targetirana je druga poruka po prioritetu, pod nazivom HS4_IDB_INPUT_2. Razlog tome je prikaz uspešne komunikacije pri slanju prve poruke i jasno prikazan napad druge.

Napad ima za cilj da targetira RTR polje poruke i kreira 6 uzastopnih dominantnih stanja na magistrali, što će rezultovati pojavljivanjem greške Suff Bit Error i primoranim slanjem Error Frame-a. Ono što se očekuje tokom napada je da ECU nakon prve neuspešno poslate poruke, u Active Error stanju primopredajnika, proba da ponovi slanje iste.

Nakon 16 repeticija i uvećanja TEC brojača do više od 127, očekuje se nastavak slanja poruka u Error Passive obliku što ce prouzrokovati da Error Frame-ovi duže traju. Nakon novih 16 uzastopnih grešaka očekuje se da TEC brojač dostigne vrednost 256 i CAN mehanizam upravljanja greškama prepozna DUT kao nefunkcionalan modul na mreži i pređe u stanje Bus Off. Takođe, cilj je da se prikaže brzina reagovanja sigurnosnog mehanizma koji aktivira digitalni izlaz OUTPUT_SAFETY_1 na ECU kako je predviđeno po zahtevima proizvođača i u samom test slučaju.

4.1 Programsko rešenje

Programsko rešenje je realizovano kroz CANoe okruženje koristeći CAPL proceduralni programski jezik. Za kontrolu VH6501 CAN Disturbance Interface-a, postoje predefinisane klase, funkcije, metode i sistemske promenljive.

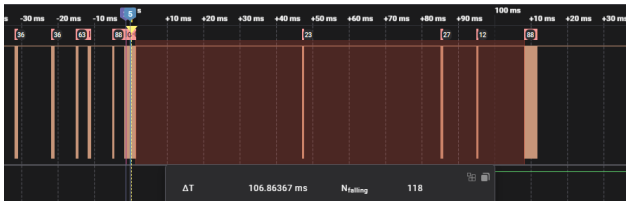
Osnovni parametri koji su neophodni za konstruisanje jednog napada su sekvenca (Sequence) i okidač (Trigger), a treći opcioni parametar je repeticija okidača (Trigger Repetitions). Spram toga u kodu su definisani objekti sequence, frameTrigger i triggerRepetitions.

U CAPL okruženju, sequence predstavlja objekat klase CanDisturbanceSequence. Može da ima ukupnu dužinu od 4096 segmenata, a svaki segment je sastavljen od 1 do 65535 FPGA pulseva (ticks). Jedan FPGA puls traje 6.25 ns što rezultuje maksimalnom mogućem segmentu dužine ~409 ns. Ovakva rezolucija podešavanja sekvenci omogućuje da se veštački produže ili skrate bit-ovi i testira tolerancija samih transivera. Dodavanje segmenata u sekvencu se izvršava funkcijom AppendToSequence, gde je prvi atribut broj 1920 sto predstavlja 6 bitova i drugi atribut „d“ koji predstavlja dominantne bitove.

Okidač frameTrigger treba da sadrži dve informacije. Masku prepoznavanja poruke koja je targetirana i polja unutar poruke nad kojim odpočinje sekvenca. Dodavanje ovih parametara je izvršeno funkcijama SetMessage i TriggerFieldType nad objektom frameTrigger.

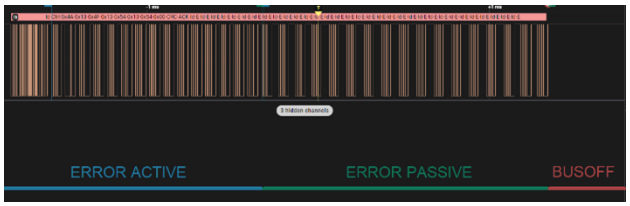
Treći parametar od značaja jeste objekat repeticije okidača (triggerRepetitions). On ima za cilj da definiše broj ciklusa, zadržavanja ciklusa, repeticija i zadržavanja repeticija. Za potrebe napada je dovoljno definisati atribute Cycles = 16 i Repetitions = 32.
Aktivacija je poslednji korak u kome se koristi funkcija canDisturbanceTriggerEnable koja prihvata parametre deviceID, frameTrigger, sequence i triggerRepetitions i njenim pozivom odpočinje napad.

4.2 Prva sekvenca i status transivera



Slika 4. *Prikaz dva uzastopna ciklusa*

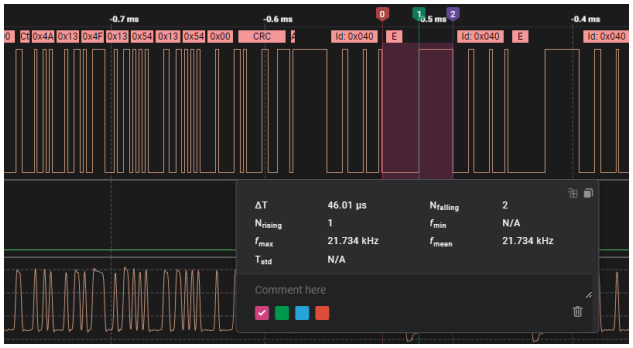
Na slici 4 uređaj VH6501 je detektovao prvu HS4_IDB_INPUT_2 poruku i izvršio napad. Transiver sa strane DUT-a je nakon prvog napada presao u *BusOff* sanje i obustavio slanje poruka ka magistrali. U crvenoj zoni sa slike 4 se vidi da je simulacija uspesno otpočela slanje poruka nakon oporavka svojih REC brojača. Prvi sledeći napad je usledio kada se transiver DUT-a reaktivirao automatski nakon ~106 ms. Ujedno, ovo je i prvi ciklus napada koji je uspešno uspostavljen. Dodatnim uvećavanjem po vremenskoj skali (prema žutoj strelici) sa merenja (Slika 4), može se uvideti celokupni prikaz 32 uzastopne repeticije (Slika 5).



Slika 5. *Prikaz dva uzastopna ciklusa*

Promene stanja transivera označene su plavom (Error Active), zelenom (Error Passive) i crvenom (Bus Off) bojom. Uvećavanjem na nivo bita (Slika 6) jasno se vidi uspešno poslata poruka HS4_IDB_INPUT_1(ID 0x000), a zatim pokušaj HS4_IDB_INPUT_2(ID 0x040) i kontrolno polje tokom kojeg nastaje greška. Kako je definisano standardom, Error Frame poruku automatski šalje transiver tokom Error Active stanja čvora. Crvena zona traje 46.01 us i predstavlja prvi Error Frame na merenju (Slika 6). Ovo ponašanje je očekivano obzirom da je 6 dominantnih bitova ubačeno od strane sekvece kršeći BitStuffing pravilo.

Prema mehanizmu upravljanja greškama narednih 6 dominantnih, 8 recesivnih bitova generisanih od strane novonastalog Error Frame-a i 3 recesivna Interframe Space bita čine ukupno niz od 23 bita. Nakon ovog vremena otpočinje retransmitovanje HS4_IDB_INPUT_2 (ID 0x040) poruke i novog napada.



Slika 6. *Oblik napada tokom aktivnog stanja*

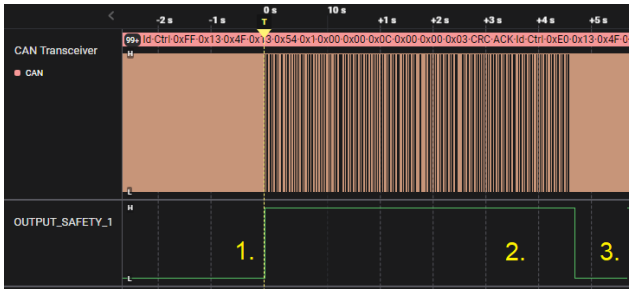
4.3 Rezultat

Ishod i realizacija napada metodom umetanjem greške je bila uspešna u smislu usmerene i precizne obstrukcije CAN mrežnog transivera kao što je prikazano u poglavlju 4.2. Preostalo je dokazati adekvatnu reakciju izlaznog kola tokom trajanja test slučaja i prikazati celokupno ponašanje ispitivanog uređaja. Korišćenjem istog alata, Saelea logički dekoder, jasno i u paraleli se vidi ispravan i očekivani rad programske podrške tokom napada (Sl. 7).

Ako se posmatra prvi korak iz procedure u tabeli 1, cilj je aktivacija izlaza OUTPUT_SAFETY_1 za vreme od 20 ms i gubitak komunikacije od strane DUT-a.

Drugi korak iz test slučaja proverava povratak izlaznog signala na nulti potencijal prestankom napada na DUT i uspešno prolazi (Slika 7 – Tačka 2).

Poslednji korak test slučaja jeste normalizacija komunikacije, što ukupno čini da je ponašanje DUTa u skladu sa sistemskim zahtevima.



Slika 7. *Celokupan napad i ponašanje ispitivanog uređaja*

5. ZAKLJUČAK

Iako se CAN protokol smatra jednim od najpouzdanijih, kada je u pitanju prenos podataka, mnogi istraživači nalaze načine da direktno ugroze ECU-ove na mreži bez dekodovanja i pristupanja internim podacima. Dovoljno je uzurpirati fizički nivo i nivo podataka i ECU može biti ugrožen do te mere da je primoran da se isključi sa mreže.

CCT (CAN Conformance Testing) i CTA (Critical Timing Analysis) su neizostavni deo testiranja pouzdanosti magistrale usled analognih ili digitalnih smetnji.

Prvenstveno, uzurpiranje mreže se radi u sklopu CAN Conformance testing-a [3], ali za potrebe funkcionalnog testiranja neophodno je proizvesti pojedine greške poput BusOff, CAN-Mute, itd i uporediti sa stvarnim odzivom DUT-a na sistemskom novou. Primenjena metoda odnosno vrsta tehnike, ubacivanje greške (Fault insertion from FMEA/FMEDA), izuzetno je poželjna u trenucima sistemskog testiranja i testiranja vozila. Mada, ono zahteva specifične uslove i opremu kao što je slučaj u ovom radu.

Povećanje pouzdanosti rada programske podrške ECU-a, konkretno za detektovanje Bus Off greške, svakako može biti postignuto kombinovanjem različitih test metoda sa Fault Insertion metodom. Dodavanjem Stress And Robusness metode na primer, uključuje repetativno izvršavanje ovakvog jednog test slučaja kroz niz operacionih stanja (High to Low Power transtionts) ili naponskih nivoa (Undervoltage ili Overvoltage).

Poboljšanje u smislu implementacije napada je svakako posebna oblast. Kao primer daljeg istraživanja može biti targetiranje različitih poruka sa prostorom zadrške između dva napada. Na ovaj način bi ispitali ponašanje i rad TEC (ili REC) brojača transivera.

Postoji jako širok spektar načina kako se mehanizam upravljanja greškama CAN protokola može testirati ali ovaj rad je sveden na jednostavniji primer. Proces testiranja je uslovno beskonačan ciklus, što dovodi do zaključka da uvek treba voditi računa o balansu utrošenog vremena, prekomernog/nedovoljnog testiranja kao i raspoloživih resursa.

6. LITERATURA

[1] Kyong-Tak Cho and Kang G. Shin: *Error Handling of In-vehicle Networks Makes Them Vulnerable*, The University of Michigan, Ann Arbor, Michigan, United States

[2] *Precise Disturbance Hardware for CAN/CAN FD and Network Interface for CANoe*, https://cdn.vector.com/cms/content/products/VH6501/Docs/VH6501_FactSheet_EN.pdf (pristupljeno u oktobru 2023.)

[3] Wolfhard Lawrenz: *CAN Conformance Testing – The Developing ISO Standard and Necessary Extensio*ns - Communication and Systems Group, University of Applied Science at Wolfenbüttel, Germany

Kratka biografija:

Nikola Dobrijević rođen je u Novom Sadu 1993. god. Master rad na Fakultetu tehničkih nauka iz oblasti Industrijske komunikacione mreže i protokoli – Mehatronika robotika i automatizacija odbranio je 2023.god.
kontakt: dobrinikola@gmail.com