

**PRIMENA UDS PROTOKOLA U AUTOMOBILSKOJ INDUSTRIJI**  
**APPLICATION OF UDS PROTOCOL IN AUTOMOTIVE INDUSTRY**Pavle Janjić, Zoran Ivanović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

**Kratak sadržaj** – U radu je predstavljen UDS protokol kao primarno sredstvo dijagnostičke komunikacije u savremenoj automobilske industriji. Objasnjen je princip rada UDS protokola i dat pregled najbitnijih servisa koje sadrži. Takođe, predložen je ISO transportni protokol kao podloga za dijagnostičku komunikaciju na transportnom nivou. U praktičnom delu data je potvrda pomenutih koncepata kroz konkretan primer dijagnostičke komunikacije izveden u Canoe programskom okruženju.

**Ključne reči:** UDS protokol, Dijagnostika, ISO transportni protokol, Automobilna industrija

**Abstract** – This paper introduces the UDS protocol as the primary method of diagnostic communication in the automotive industry. It describes the operational approach of the UDS protocol and the most important diagnostic services contained within it. Additionally, the ISO transport protocol is presented as the foundation for diagnostic communication at the transport level. The practical part involves the verification of the aforementioned through an example of diagnostic communication created in the Canoe environment.

**Keywords:** UDS protocol, diagnostics, ISO transport protocol, Automotive industry

**1. UVOD**

Od svih tehničkih inovacija u automobilske industriji najznačajniju predstavlja kompjuterizacija automobilske komponenti. Tokom godina, današnji moderni automobili evoluirali su od čisto mehaničkih naprava do naprednih sistema koji se u mnogome oslanjaju na softver kako bi mogli pravilno da funkcionišu. Počevši od sedamdesetih godina prošlog veka broj elektronskih uređaja unutar jednog vozila je neprestano rastao od jednog centralnog računara do oko 100 elektronskih kontrolnih jedinica (engl. *electronic control unit* - ECU) koliko poseduje jedan moderan današnji automobil. Sve kontrolne jedinice zajedno sa svojim senzorima, međusobno povezane preko CAN (engl. *controller area network*) mreže, kontrolišu većinu podsistema unutar vozila, te se njihov skup može smatrati i „mozgom“ jednog današnjeg modernog automobila.

Kako bi se osiguralo da celokupan sistem funkcioniše bez grešaka neophodne su redovne dijagnostičke provere.

**NAPOMENA:**

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Zoran Ivanović, vanr. prof.

Dijagnostika vozila podrazumeva metodologiju uključenu u identifikaciju i evaluaciju problema koji mogu negativno uticati na normalan rad vozila. Ona može da obuhvata širok spektar tehnika i alata u rasponu od površnih fizičkih provera do složenijih, kompjuterski zasnovanih analiza. S obzirom na ponekad skrivenu prirodu problema, efikasna dijagnostika je ključna za uspešne popravke i nesmetan rad svakog vozila. Sa uslozljavanjem kompjuterskih sistema u vozilima broj potencijalnih problema se eksponencijalno povećavao, ali samo njihovo otkrivanje je postalo značajno jednostavnije. Svako današnje vozilo dolazi opremljeno sa dijagnostičkim konektorima, te mehaničar uz pomoć odgovarajućeg instrumenta može da prikupi podatke o greškama iz svih kontrolnih jedinica u automobilu. Redovne dijagnostičke kontrole omogućavaju preventivno otkrivanje problema te se na ovaj način često mogu izbeći kvarovi na putu, ugrožavanje života putnika i skupe popravke.

Usled uslozljavanja procesa dijagnostike, stanje vozila tokom dijagnostičkih provera postaje sve značajnije. Različiti dijagnostički parametri mogu imati drugačije vrednosti u zavisnosti od toga da li je vozilo u pokretu ili miruje u garaži. Kako bi se osigurala maksimalna bezbednost putnika i ispravan rad u svakom trenutku dijagnostiku možemo podeliti na dve grane:

- Dijagnostika na tabli (engl. *on-board diagnostic*)
- Dijagnostika van table (engl. *off-board diagnostic*)

**2. OBD PROTOKOL**

Za potrebe standardizacije načina komunikacije između dijagnostičkog alata i samog vozila nastao je OBD protokol koji definiše izgled i sadržaj poruka kojima se komunikacija odvija kao i sam tip konektora koji se koristi. Uprkos svim dobrim stranama, nedostatak OBD2 protokola predstavlja njegova fokusiranost na pogon i emisije te pomoću njega nije moguće očitati korisne podatke iz ostalih kontrolnih jedinica kao što su status sigurnosnog pojasa ili vazдушnih jastuka. Takođe, protokol dozvoljava samo očitavanje podataka, ali nije omogućeno njihovo upisivanje, te ne može biti od pomoći pri samom otklanjanju greške nego isključivo prilikom detektovanja.

**3. UDS PROTOKOL**

Usled povećanja broja ECU-a koji se ne odnose na pogon i emisije, nedostaci OBD2 protokola postaju sve očigledniji. Porastom broja problema kojih kontrolne jedinice koje ne pripadaju pomenutoj grupi mogu izazvati, nastaje potreba za standardizovanjem načina njihove

komunikacije sa dijagnostičkim alatima. Zbog toga, internacionalna organizacija za standardizaciju – ISO (engl. *International organization for standardization*) razvija komunikacioni protokol namenjen dijagnostičkoj analizi rada i konfiguraciji kontrolnih jedinica u automobilske industriji nazvan Ujedinjeni dijagnostički servisi – UDS (engl. *Unified Diagnostic Services*). Sam UDS protokol specificiran je u ISO14229 standardu i predstavlja protokol višeg reda zasnovan na klijent – server topologiji, gde je u ulozi klijenta dijagnostički alat povezan najčešće na standardizovani OBD2 konektor u vozilu.

Čitava komunikacija zasniva se na principu zahteva od strane klijenta (mehaničar) i odgovora od strane servera (ECU). UDS standard definiše: sve dostupne servise, formate poruka zahteva, formate poruka odgovora, vremenske parametre tokom komunikacije, kao i sam način korišćenja svih definisanih servisa. Propisan je standardan format i funkcija ali ne i način na koji implementacija mora biti izvršena te ga iz tog razloga mnogi nazivaju i UDS interfejsom.

Za razliku od OBD2 protokola koji omogućava isključivo mogućnost iščitavanja podataka, UDS obezbeđuje takođe i interfejs za njihov upis. Pored standardnih opcija pronalazjenja i analize grešaka sačuvanih u EEPROM memoriji mikrokontrolera UDS nudi i niz drugih značajnih mogućnosti kao što su:

- Transfer velike količine podataka. Najčešće se koristi u slučaju kada je potrebno spustiti novi softver na ECU
- Kontrola ulaza i izlaza mikrokontrolera
- Pokretanje predefinisanih rutina
- Čitanje brojnih podataka iz memorije mikrokontrolera i upis podataka u istu

### 3.1. Format dijagnostičkog zahteva i odgovora

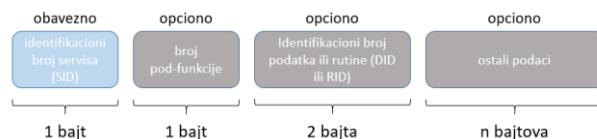
UDS protokol predstavlja kolekciju servisa od kojih svaki poseduje svoj jedinstveni identifikacioni broj (engl. *service identifier* - SID). Identifikacioni broj je veličine jednog bajta i uvek se nalazi na početku zahteva.

Nakon identifikacionog broja servisa, drugi bajt zahteva namenjen je za pod-funkciju (engl. *sub-function*). Njime se određuje koju tačno funkcionalnost datog servisa je neophodno izvršiti. Pojedini servisi, ne poseduju nijednu pod-funkciju te je samim tim ovo polje opciono u formatu zahteva.

Kako UDS protokolom klijent i server mogu da komuniciraju samo putem brojeva svakom podatku koji klijent želi pročitati ili upisati dodeljen je određen identifikacioni broj. Taj broj se naziva identifikacioni broj podatka iliti skraćeno DID (engl. *data identifier*), veličine je dva bajta i u samom zahtevu nalazi se posle pod-funkcije.

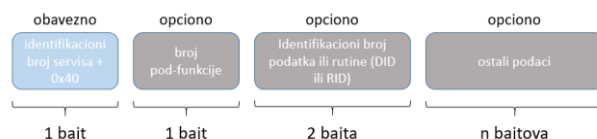
Slično DID-ovima, ukoliko je zahtevan servis za kontrolu rutine, kako bi se specificiralo koja tačno rutinu je potrebno izvršiti, koristi se identifikacioni broj rutine ili skraćeno RID (engl. *routine identifier*). Pošto se DID-ovi i RID-ovi ne koriste u svakom servisu oni predstavljaju opcione bajtove u poruci zahteva.

Nakon DID-a ili RID-a u poruci zahteva se nalaze svi ostali podaci (engl. *data record*). Šta će se nalaziti u ovom polju kao i sama njegova veličina i postojanje zavisi od servisa do servisa. U skladu sa tim i samo polje je opciono.



Slika 1. Format dijagnostičkog zahteva

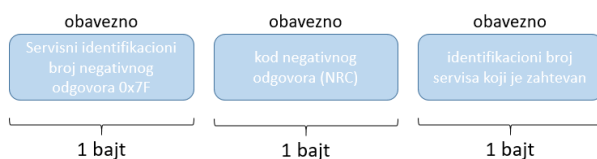
Po pristizanju zahteva ECU najpre proverava njegovu validnost (da li je podržan dati servis ili pod-funkcija, da li je zahtev odgovarajuće dužine, da li su svi preduslovi za ispunjenje zahteva ispunjeni itd.). U slučaju da su sve provere uspešno sprovedene server prelazi na izvršavanje zahteva. U trenutku kada je zahtev uspešno izvršen server vraća testeru pozitivan odgovor. Prvi bajt pozitivnog odgovora sadrži servisni identifikacioni broj odgovora koji se preračunava uvećavanjem identifikacionog broja servisa zahteva za 0x40 (šesti bit setovan). Ukoliko polje pod-funkcije postoji u zahtevu, na mestu drugog bajta će se naći broj pod-funkcije koja je zahtevana (identična vrednost kao u zahtevu). Nakon pod-funkcije slede DID ili RID (takođe opciona polja u zavisnosti od njihovog postojanja u zahtevu). Na kraju dolazi opcioni prostor namenjen za ostale podatke čije se postojanje, veličina i upotreba razlikuje od servisa do servisa.



Slika 2. Format pozitivnog odgovora

Ukoliko bilo koja od provera zahteva nije bila uspešna ili ukoliko server iz nekog razloga nije u mogućnosti da izvrši dati zahtev uslediće negativan odgovor. Svaki negativan odgovor se sastoji od 3 bajta i započinje sa predefinisanim vrednošću servisnog identifikacionog broja negativnog odgovora 0x7F.

Nakon njega sledi kod negativnog odgovora iliti skraćeno NRC (engl. *negative response code*) veličine jednog bajta. On uzima jednu od niza predefinisanih vrednosti u zavisnosti od razloga zbog koga zahtev nije mogao da bude ispunjen. Uz pomoć njega klijent može naslutiti zbog čega je došlo do greške. Vrednosti svih NRC-ova i njihova konkretna namena definisani su u ISO 14229-1 dokumentu [1]. Poslednji bajt negativnog odgovora namenjen je za identifikacioni broj servisa koji nije uspešno izvršen (identična vrednost kao u poruci zahteva).



Slika 3. Format negativnog odgovora

### 3.2. Dijagnostički servisi

UDS standard u sebi sadrži definisan veliki broj servisa [1] među kojima se kao najbitniji ističu:

- *Diagnostic Session Control* (SID 0x10) – Servis namenjen promeni dijagnostičkih sesija. Pod pojmom sesije se podrazumeva stanje ECU-a koje određuje šta je dostupno za dijagnostiku. Određene dijagnostičke funkcionalnosti dostupne su samo u pojedinim sesijama.
- *ECU Reset* (SID 0x11) - Servis namenjen izvršavanju reseta mikrokontrolera. Željena vrsta reseta se odabira uz pomoć pod-funkcije zahteva.
- *Security Access* (SID 0x27) – Servis namenjen otključavanju potrebnog nivoa sigurnosnog pristupa. Čitav sigurnosni koncept je zasnovan na principu semena (eng. *seed*) i ključa (eng. *key*), a željeni nivo sigurnosnog pristupa se odabira vrednošću pod-funkcije.
- *Read Data By Identifier* (SID 0x22) – Servis namenjen iščitavanju podataka iz memorije. Traženi podatak je određen vrednošću DID-a prosleđenog u zahtevu.
- *Write Data By Identifier* (SID 0x2E) - Servis namenjen upisu podataka u memoriju. Podatak koji će biti upisan odabira se vrednošću DID-a prosleđenog u zahtevu.
- *Routine Control* (SID 0x31) – Servis namenjen pokretanju, zaustavljanju i prikupljanju rezultata rutina. Rutina u smislu dijagnostike predstavlja predefinisani niz instrukcija kojim se izvršava određena funkcionalnosti.
- *Read DTC Information* (SID 0x19) – Servis namenjen iščitavanju različitih informacija o sačuvanim kodovima greški (engl. *diagnostic trouble code* - DTC) unutar mikrokontrolera.

Celokupan aplikativni sloj UDS-a, uključujući sve prethodno navedene servise, može se ravnopravno i neizmerno koristiti bez obzira na protokol nižeg reda koji je zastupljen na fizičkom nivou. S obzirom na rasprostranjenost primene nastavak rada će biti fokusiran na primenu UDS-a koja se oslanja na CAN mrežu na fizičkom nivou.

### 4. ISO TRANSPORTNI PROTOKOL

Ograničenje u maksimalnoj dužini poruke predstavlja značajan nedostatak CAN komunikacije. Ponajviše u pogledu same dijagnostike, pojedini zahtevi mogu značajno premašivati maksimalnih 8 ili 64 bajta koliko može da sadrži jedna CAN ili CAN FD poruka. Kako bi se ovaj nedostatak nadomestio međunarodna organizacija za standardizaciju u svom ISO 15765-2 daje predlog transportnog protokola (ISO TP) namenjenog više-paketnom prenosu podataka veličine do 4095 bajtova.

Kako bi komunikacija putem transportnog protokola bila moguća u deo CAN poruke namenjen smeštanju podataka, a pre samih podataka, umetnuto je zaglavlje koje sadrži dodatne protokolne informacije. ISO transportni protokol [2] razlikuje 4 različita tipa poruka:

- Pojedinačna poruka (engl. *single frame*)
- Prva poruka (engl. *first frame*)

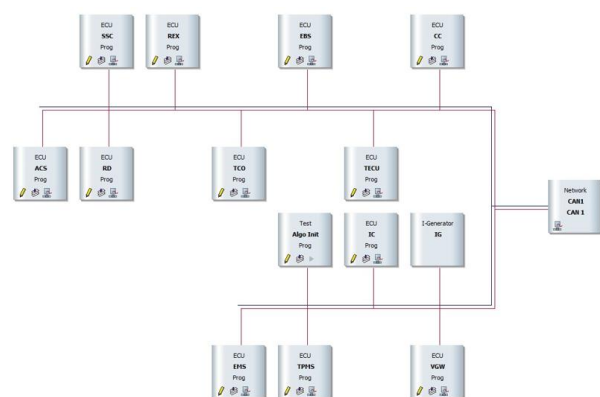
- Uzastopna poruka (engl. *consecutive frame*)
- Poruka za kontrolu toka (engl. *flow control frame*)

Ukoliko je potrebno poslati podatke dužine do 7 bajtova za to će biti korištena pojedinačna poruka. Sa druge strane, ako je potrebno poslati podatke duže od 7 bajtova koristi se više-paketni prenos. Njega otpočinje pošiljalac slanjem prve poruke u kojoj naglašava veličinu podataka koje želi poslati. Primalac je dužan da odgovori sa porukom kontrole toka u kojoj će obavestiti pošiljaoca da li je spreman za prijem i proslediti potrebne komunikacione parametre. Nakon prijema potvrde dalji prenos se nastavlja uzastopnim porukama.

### 5. CANOE SOFTVERSKI ALAT

U današnje doba, elektronske kontrolne jedinice preuzimaju vodeću ulogu u upravljanju radom jednog modernog vozila te u skladu sa tim od uspešnog rada njihovog softvera umnogome zavisi ugodnost tokom vožnje, a u nekim slučajevima i život samih putnika. Zbog toga je nad svakim ECU-om neophodno sprovesti čitav niz provera kako bi se osiguralo da sve funkcioniše kako što je i planirano. Kako bi rezultati bili merodavni potrebno je ECU dovesti u okruženje što približnije okruženju u stvarnom vozilu. Rešenje pomenutog problema nudi kompanija Vector, sa svojim programskim alatom nazvanim Canoe [3]. Uz pomoć Canoe-a moguće je simulirati kompletnu komunikacionu mrežu jednog vozila, a samim tim i realno okruženje ECU-a za potrebe testiranja. Pored simulacije, Canoe pruža platformu za praćenje kompletnog mrežnog saobraćaja u realnom vremenu, kao i za detekciju i analizu greški nastalih tokom komunikacije.

U konfiguracionom prozoru Canoe programa moguće je kreirati željenu topologiju komunikacionog sistema. Izgled jedne praktične realizacije CAN mreže dat je na slici 4. Prikazani čvorovi simuliraju slanje svih potrebnih poruka koje bi unutar vozila bile slate od strane drugih elektronskih kontrolnih jedinica.



Slika 4. Realizacija CAN mreže u Canoe okruženju

Usled velikog značaja, obezbeđivanje ispravnog rada dijagnostike nalazi se visoko na listi prioriteta svakog proizvođača motornih vozila. U skladu sa tim, Canoe programsko okruženje sadrži i alat za dizajn, razvoj, testiranje i simulaciju rada dijagnostičkih funkcionalnost jednog ECU-a. Za upotrebu dijagnostičkog sloja Canoe programa potrebno je standardnoj konfiguraciji pridodati i

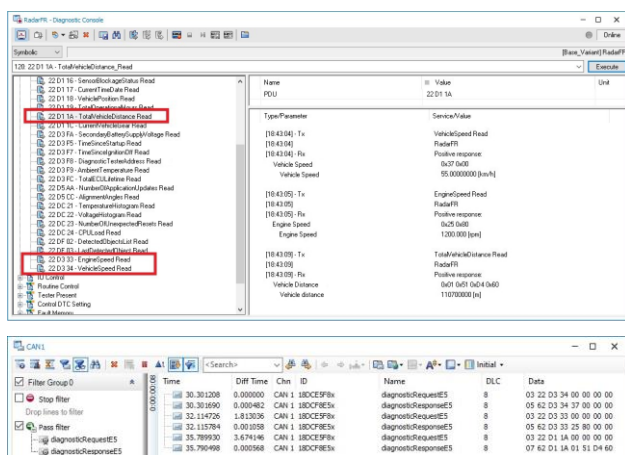


dijagnostički opis, najčešće u vidu CDD fajla (engl. *CANdel* Diagnostic Description - CDD). Dijagnostički opis se može posmatrati kao vrsta baze podataka za dijagnostičke zahteve koja u sebi sadrži definiciju i uputstvo za interpretaciju svih, od strane ECU-a podržanih, dijagnostičkih servisa i njihovih parametara (kako zahteva tako i odgovora). Na osnovu ubačenog dijagnostičkog opisa, Canoe automatski generiše dijagnostičku konzolu koja predstavlja intuitivni grafički interfejs namenjen jednostavnom slanju dijagnostičkih zahteva i simboličkoj interpretaciji dobijenih odgovora. Nakon odabira iz liste definisanih, zahtev se šalje dvoklikom miša, a dobijeni odgovor biva ispisan u samoj konzoli. Dijagnostička konzola se može smatrati softverskom realizacijom dijagnostičkog alata koji koristi server.

## 6. DIJAGNOSTIKA U CANOE OKRUŽENJU

U ovom poglavlju kroz će kroz Canoe simulaciju ECU-a radarskog sistema i njegovog okruženja, biti data verifikacija prethodno objašnjenih dijagnostičkih koncepta.

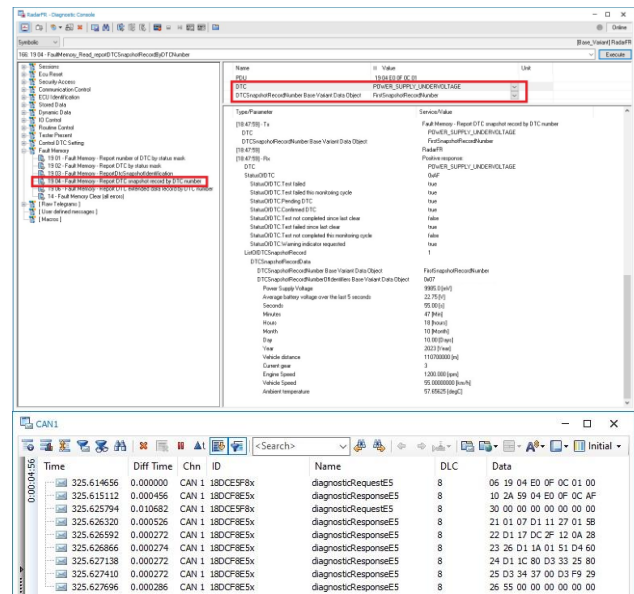
Simulacija je podešena tako da je trenutna brzina vozila postavljena na 55 km/h, broj obrtaja motora na 1200 obrtaja u minuti, a pređena kilometraža na 110.700 km. Sve za sam rad bitne informacije, uključujući i prethodno navedene, ECU dobija preko poruka na CAN magistrali ili sa svojih internih senzora. Trenutne vrednosti pomenutih veličina možemo očitati i sa samog ECU-a preko dijagnostike ukoliko ih zahtevamo preko servisa 0x22 uz pomoć odgovarajućih DID-ova (u ovom slučaju 0xD333, 0xD334 i 0xD11A) što je i prikazano na slici 5. Pored ispisa iz dijagnostičke konzole dat je i pregled CAN komunikacije koja se odigrala u pozadini. Na slici se jasno može uočiti da su za komunikaciju korišteni pojedinačne poruke. Takođe, može se uvideti ogroman značaj dijagnostičkog alata (u ovom slučaju konzole) koji niz bajtova uz pomoć konverzionih formula prevodi u korisniku čitljive podatke.



Slika 5. Očitavanje stanja sistema putem dijagnostike

Ukoliko se u ovom trenutku u sistem unese greška spuštanjem napona napajanja simuliranog ECU-a ispod dozvoljene granice ona će rezultovati odgovarajućim kodom greške 0xE00F0C. Sada je moguće zatražiti i informacije o stanju sistema u trenutku inicijalnog nastanka greške uz pomoć servisa 0x19 njegovom pod-funkcijom

0x04, što je i prikazano na slici 6. Može se primetiti da se sačuvano stanje trenutka nastanka greške poklapa sa podacima koje su očitani neposredno pre unošenja greške u sistem. Kako je odgovor duži od 7 bajtova za njegovo slanje je morao biti korišten više-paketni prenos.



Slika 6. Stanje sistema u trenutku nastanka greške

## 7. ZAKLJUČAK

Evolucijom automobilske industrije elektronske kontrolne jedinice preuzimaju kontrolu nad svim bitnijim funkcijama jednog savremenog vozila. Za proveru ispravnosti rada, analizu greški, i parametrizaciju samih kontrolnih jedinica ključnu ulogu ima dijagnostika. U ovom radu predstavljen je UDS protokol kao osnovno sredstvo dijagnostičke komunikacije u automobilske industriji. Protokol je verifikovan kroz praktične primere izvedene u Canoe okruženju.

## 8. LITERATURA

- [1] International Organization for Standardization, ISO 14229-1 - Unified diagnostic services (UDS) - Part 1: Specification and requirements, 2013
- [2] ISO 15765-2 - Diagnostic communication over Controller Area Network (DoCAN) - Part 2: Transport protocol and network layer services, 2016
- [3] [https://cdn.vector.com/cms/content/products/canoe/canoe/docs/Product%20Informations/CANoe\\_ProductInformation\\_EN.pdf](https://cdn.vector.com/cms/content/products/canoe/canoe/docs/Product%20Informations/CANoe_ProductInformation_EN.pdf) (pristupljeno u oktobru 2023.)

## Kratka biografija:



**Pavle Janjić** rođen je u Zrenjaninu 1997. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2023.god.

kontakt: pavlejanjic97@gmail.com