Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.38
DOI: https://doi.org/10.24867/31BE25Petrovic

SISTEM ZA DETEKCIJU GRUBE VOŽNJE BAZIRAN NA TROSLOJNOJ ARHITEKTURI

DRIVING STYLE DETECTION SYSTEM BASED ON THREE-TIER ARCHITECTURE

Aleksandar Petrović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom članku opisana je realizacija sistema za detekciju grube vožnje. Pod grubom vožnjom se smatra naglo kočenje, naglo ubrzanje i grubo skretanje. Sam sistem se sastoji iz tri dela: aplikacije implementirane na ESP32 mikrokontroleru, Android aplikacije i veb servera. Esp32 deo služi za pribavljanje OBD2 podataka iz vozila. Android aplikacija beleži događaje koje smatramo grubom vožnjom zajedno sa koordinatama na kojima su se desili. Da bi sistem za detekciju grube vožnje uspešno funkcionisao potrebno je podatke sa ESP32 i telefona prikupiti i učiniti dostupnim korisniku u tu svrhu razvijen je veb server.*

Ključne reči: *Android, ESP32, CAN, OBD2, MySQL, JavaScript, Node.js, HTML, CSS, GPS, TWAI*

Abstract – *This article describes the implementation of driving style detection system. Events that are interesting for us are hard braking, sudden acceleration and sharp turning. This project consists of three parts: application implemented on ESP32 microcontroller, Android app and web server. Part implemented on ESP32 is responsible for collecting vehicle-related data. The Android app is used for detecting events that are considered important for evaluating driving style. For system to function properly data from both the ESP32 and the phone must be gathered and made accessible to the user, for which the web server was developed.*

Keywords: *Android, ESP32, CAN, OBD2, MySQL, JavaScript, Node.js, HTML, CSS, GPS, TWAI*

1. UVOD

Transport ljudi i robe je neizostavni deo funkcionisanja savremenog društva. Problem porasta nezgoda u saobraćaju dovodi do velikih finansijskih gubitaka kao i do lakših i težih povreda učesnika u saobraćaju. Nasilnička i gruba vožnja jedan je od glavnih razloga nezgoda.

Posmatrano iz ugla vlasnika kompanija sa velikim brojem vozila, informacije o stilu vožnje zaposlenih omogućavaju preventivno reagovanje i uštedu novca.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Predrag Teodorović, vanredni profesor.

Da bi gruba vožnja mogla da se detektuje prvo je moramo definisati. Pod grubom vožnjom se smatra naglo kočenje, naglo ubrzanje i grubo skretanje, stvari koje se lako mogu detektovati uz pomoć akcelerometra, dok bi uz to korisno bilo znati i lokaciju gde se sam prekršaj desio, to su podaci koji se lako mogu dobiti sa senzora pametnih telefona. Pored tih podataka korisni podaci bi bili i oni koje vozilo daje tokom same vožnje.

Projektovani sistem se sastoji iz tri dela:

1. *Android* aplikacije – *GPS* senzor i akcelerometar u samom telefonu su pogodni za detekciju, pošto se sama pozicija telefona u automobilu ne zna pre nego što praćenje krene neophodno je koordinatni sistem akcelerometra rotirati tako da se poklapa sa koordinatnim sistemom vozila
2. Aplikacije na *ESP32* mikrokontroleru – *TWAI/CAN* kontroler unutar same pločice uz dodatni transiver se može koristiti za čitanje *OBD2* podataka iz samog automobila (broj šasije, obrtaji motara, brzina vozila,...), ti podaci sa sobom nose dodatne informacije o samoj vožnji
3. Veb servera – koji služi za vizuelizaciju podataka dobijenih od ostatka sistema, gde se mogu videti sve rute koje su vožene, zajedno sa svim prekršajima koji su se desili na tim rutama, takođe se mogu videti poslednji pristigli podaci koje je poslao ESP32 mikrokontroler za izabrano vozilo i gde se može podešavati interval semplovanja podataka koje vrši ESP32

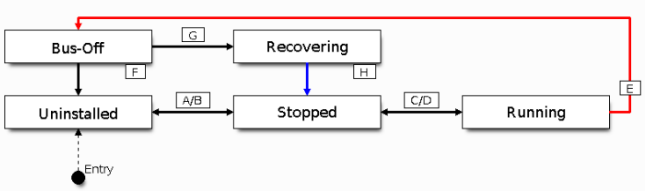
2. *ESP32*

Ovo poglavlje bavi se komunikacijom sa vozilom, protokolima koji se u tu svrhu koriste kao i načinom na koji je željena funkcionalnost implementirana.

2.1. *TWAI*

Deo sistema koji komunicira sa vozilom realizovan je korišćenjem *ESP32* mikrokontrolera. *ESP32* mikrokontroleri su pogodni za aplikacije u kojima se koristi *CAN* interfejs jer u sebi imaju integrisan *TWAI* (eng. *Two-Wire Automotive Interface*) kontroler koji je kompatibilan sa *ISO11898-1* protokolom i *CAN2.0* interfejsom. *TWAI* je serijski protokol koji je asinhron, ima mogućnost detekcije grešaka i obaveštavanje ostalih nodova o tome, podržava više mastera, tako da transfer

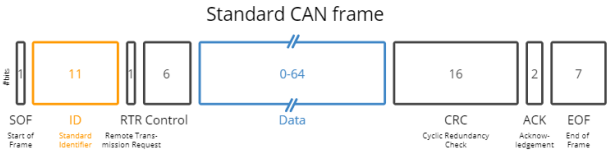
može biti iniciran od strane bilo kog noda, a svi ostali nodovi će dobiti poruku [1]. Da bi se *ESP32* koristio na *TWAI/CAN* magistrali potrebno mu je dodati eksterni transiver kao što je *MCP2551*. *TWAI* kontroler poseduje 4 signala *TX, RX, BUS-OFF i CLKOUT*, od koji su samo prva dva obavezna. Nizak logički nivo (0V) smatra se dominantim i na prijemnom i na predajnom pinu. Svaka *TWAI* poruka sastoji se od sledećih elemenata: identifikatora, dužine korisne poruke - *DLC (eng. Data Length Code)* i 8 bajtova podataka korisne poruke. Rad drajvera kontrolera je prikazan na slici 1.



Slika 1. *Prikaz stanja TWAI kontrolera [1]*

2.2. *CAN*

CAN (eng. Controller Area Network) je jedan od najčešće korišćenih standarda za komunikaciju između različitih elektronskih komponenti u automobilu (eng . *Electronic Control Units - ECU*). Nastao je osamdesetih godina prošlog veka, da bi se eliminisale mane direktne, analogne komunikacije između nodova. *CAN* magistrala predstavlja komunikacioni sistem baziran na porukama, koji omogućava brzu i pouzdanu interakciju između različitih delova automobila. Svaki modul može da prima poruke, ali i da ih šalje svim ostalim modulima u mreži (eng. *broadcast-based* protokol) [2]. Svaki *ECU* za sebe odlučuje da li mu je neka poruka potrebna ili ne. Arbitraža se vrši dodelom prioriteta porukama, manji identifikacioni broj poruke, znači da je njen prioritet veći. Svaki podatak koji se šalje na magistrali nalazi se u frejmu, koji sadrži jasno definisana polja, prikazana na slici 2.



Slika 2. *Izgled jednog CAN frejma [2]*

U zavisnosti od performansi postoji nekoliko vrsta *CAN* protokola, a u ovom radu se koristi *High-speed CAN 2.0* koji podržava brzinu prenosa do 1 *Mbit/s* (*boud rate*) i fiksnu veličinu podataka do 8 bajtova. *CAN* je u skladu sa *ISO11898* podeljen u dve podgrupe: *ISO11898-1* za opis sloja za prenos podataka (eng. *data link layer*) i *ISO11898-2* za opis zahteva fizičkog sloja (eng. *physical layer*).

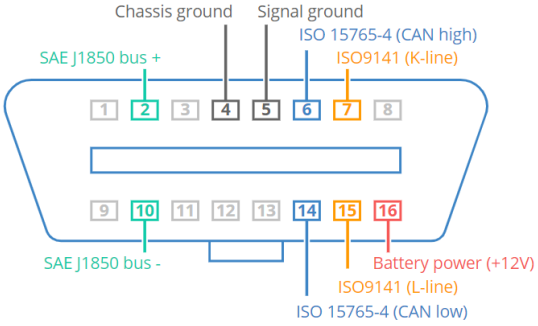
CAN fizički gledano čini par uvrnutih žica, *CAN High* i *CAN Low*. *CAN High* se obično nalazi u opsegu od 2.5V do 3.75V, dok je *CAN Low* od 1.25V do 2.5V. Ukoliko obe linije očitavaju 2.5V signal je recesivan (nivo logičke jedinice), dok ukoliko očitavaju različite vrednosti smatra se da je signal dominantan (logička nula). Važnost prenosa putem ovakvog diferencijalnog signala leži u otpornosti na šum, a u cilju očuvanja integriteta poruke.

2.3. *OBD2*

OBD2 (eng. *On Board Diagnostic*) je standardizovani protokol koji služi za iščitavanje kodova grešaka (eng. *DTC - Diagnostic Trouble Codes*) i podataka vezanih za trenutno stanje automobila (eng. *current data*) preko *OBD2* konektora. [3] Odnos *OBD2* i *CAN* je takav da je *OBD2* protokol višeg nivoa, a *CAN* predstavlja metod komunikacije. *OBD2* definiše da:

- Brzina na *CAN* magistrali mora biti 250K ili 500K bita po sekundi
- Identifikatori poruka moraju imati jedanaest (u automobilima) ili dvadeset i devet bita i koriste se u komunikaciji
- Poruka mora biti dužine do osam bajtova

Postoje dva tipa *OBD2* konektora, tip A i tip B, najveća razlika je napon na 16. pinu koji kod tipa A iznosi 12V, a kod tipa B 24V. Na slici 3. Prikazan je konektor tipa A, koji se sastoji od 16 pinova, gde su pinovi 6 i 14 određeni za *CAN High* i *CAN Low*.



Slika 3. *OBD2 konektor tipa A [3]*

OBD2 PID (eng. *Parameter IDs*) su predefinisani kodovi koji se koriste za traženje podataka od automobila zajedno sa servisima, to jest modovima. *OBD2* zahtevi kao što su zahtevi za broj šasije (eng. *Vehicle Identification Number - VIN*) i zahtevi za iščitavanje kodova grešake u svom odgovoru imaju više od osam bajtova, a kako jedna poruka u protokolu može imati maksimalno osam bajtova, potrebno je koristiti takozvani *ISOTP* (*ISO 15765-2*) transportni protokol. *ISOTP* definiše četiri različite vrste frejma: *Single frame*, *First frame*, *Consecutive frame* i *Flow control frame*.

2.4. Implementirani softver

Implementirani softver podeljen je na dva dela:

- *HAL* (eng. *Hardware Abstraction Layer*) – u kom su implementirani delovi softvera zaduženi direktno za hardver
- *MainApp* – u kom je implementirana glavna funkcionalnost

HAL u sebi sadrži četiri modula: modul koji je zadužen za funkcije koje se koriste za konekciju na internet, modul za logovanje, modul zadužen za sinhronizaciju vremena sa serverom (*„pool.ntp.org“*) i modul zadužen za *CAN* funkcije. Softver implementiran u ovom sloju radi direktno sa *API*-jem (eng. *Application programming interface*) koji je *ESP-IDF* pružio i vrši svu potrebnu inicijalizaciju i proveru podataka kako se na višem sloju ne bi vodilo računa o tome.

MainApp je implementirana kroz dve mašine stanja, jednu koja kontroliše pristup internetu i mašinu stanja koja implementira komunikaciju sa automobilom.

3. ANDROID

Deo sistema koji prikuplja podatke sa akcelerometra i *GPS*-a implementiran je u okviru *Android* aplikacije. *Android* je prvenstveno platforma za mobilne telefone koja se koristi za različite namenske uređaje. Za razvoj *Android* aplikacije korišćen je *Java* programski jezik i *Android Studio* kao razvojno okruženje.

3.1. Senzori

Senzori u mobilnim telefonima su malih dimenzija i potrošnje i spadaju u grupu mikro-elektro-mehaničkih senzora (eng. *micro-electro-mechanical sensors – MEMS*). U ovom radu je korišćen akcelerometar i *GPS* prijemnik.

Akcelerometar

Akcelerometar je sastavni deo svakog mobilnog telefona i predstavlja uređaj za merenje ubrzanja (akceleracije). Meri promene sila koje utiču na uređaj u metrima po sekundi na kvadrat u tri ose (*x, y* i *z*).

GPS

GPS (eng. *Global Positioning System*) je globalni navigacioni satelitski sistem. *GPS* prijemnik (u mobilnom telefonu) može na osnovu radio signala koje prima sa *GPS* satelita da odredi svoju tačnu lokaciju, zahvaljujući činjenici da je brzina kretanja radio talasa kroz vazduh poznata i konstantna. Naravno, da bi proračun bio moguć, potrebni su podaci od minimalno tri satelita.

3.2. Implementirana *Android* aplikacija

Na slici 4. dat je prikaz početne aktivnosti aplikacije. Kako bi prešao na glavnu aktivnost korisnik mora da unese *VIN* u očekivanom formatu. Jedan od primera formata koje se očekuje je 2T3-RFREV7D-W108177.

Slika 4. *Početna aktivnost aplikacije*

Na slici 5. Dat je izgled glavnog dela aplikacije u poljima ***Lat***, ***Lon***, ***Altitude***, ***Speed*** i ***Address*** upisani su podaci dobijeni sa lokacionih senzora.

Pritiskom na dugme ***START TRIP*** počinje prikupljanje podataka sa senzora lokacije i beleže se koordinate početne lokacije. Pritiskom na ***STOP TRIP*** zaustavljamo praćenje vožnje i šaljemo podatke na server, sa svim zabeleženim događajima, tačnije prekršajima naglog kočenja, ubrzanja ili skretanja i koordinate početka vožnje i krajnje destinacije.

SHOW MAP dugme nam omogućava da vidimo mapu svih zabeleženih događaja, dok nam ***SHOW OBDII DATA*** pribavlja podatke koji su poslednji pristigli sa vozila čiji smo *VIN* uneli u početnoj aktivnosti.

Slika 5. *Izgled glavne aktivnosti*

Takođe u samoj aplikacije je moguća i vizuelizacija podataka sa akcelerometra preko grafa i direktnih vrednosti za ose koje će biti ispisane na ekranu.

4. VEB SERVER

U ovom poglavlju biće reči o realizaciji veb servera. Da bi sistem za detekciju grube vožnje uspešno funkcionisao, potrebno je podatke sa senzora mikrokontrolera i pametnog telefona prikupiti i ceo sistem povezati u jedinstvenu celinu. Veb server je podeljen na dve celine, serversku stranu (*back-end*) i klijentsku stranu (*front-end*).

4.1. *Back-end*

Back-end ili serverski deo veb sajta može se podeliti na dve celine veb server i bazu podataka. Uloga mu je da odgovara na *HTTP* (eng. *HyperText Transfer Protocol*) zahteve korisnika. Klijent i server komuniciraju po principu: klijent šalje zahtev, server ga prihvata i na njega odgovara. *Back-end* je realizovan koristeći funkcije koje obezbeđuje *Express.js (Node.js framework)*. [4] Za skladištenje informacija korišćena je *MySQL* baza podataka.

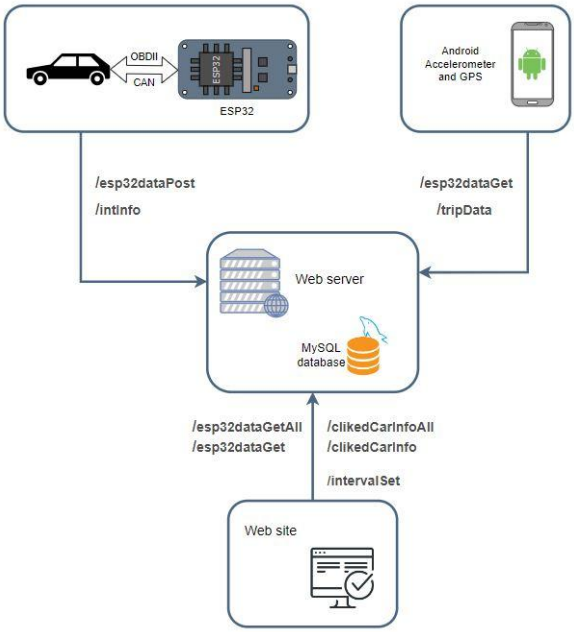
Izgled *back-end* dela veb sajta

Na slici 6. nalazi se šematski prikaz veb servera. Server prima i obrađuje zahteve koji mogu da stignu od strane:

- Android aplikacije, to jest mobilnog telefona korisnika
- *ESP32* koji prikuplja podatke sa *OBD2* porta i prosleđuje ih na server
- *Front-end*-a

Android aplikacija sa serverom komunicira u dva slučaja. Prvi slučaj je kad je završena vožena ruta (eng. *trip*) da dostavi informacije o toku vožnje i događajima, a drugi slučaj je kad želi da dobije informacije o vozilu koje je *ESP32* prethodno dostavio na server. *ESP32* sa serverom

takođe komunicira u dva slučaja, kad želi da dostavi podatke pročitane sa *OBD2* porta i kad želi da dobije interval za slanje podataka za određeno vozilo.



Slika 6. *Šematski prikaz veb servera*

4.2. *Front-end*

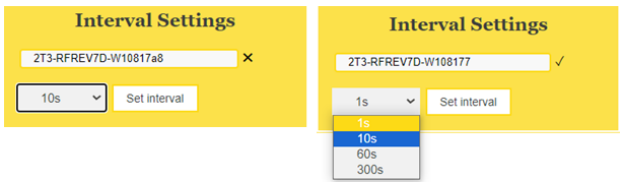
Front-end je termin koji opisuje proces kreiranja dela veb sajta koji je vidljiv korisniku i sa kojim on ima neposrednu interakciju. Implementacija grafičkog interfejsa veb sajta može se podeliti na tri celine: opis, dizajn i dodavanje funkcionalnosti stranici. U tu svrhu se koriste redom: *HTML* (eng. *Hypertext Markup Language*), *CSS* (eng. *Cascading Style Sheets*) i *JavaScript*. [5]

Izgled *front-end* dela veb sajta

Veb stranica je vizuelno podeljena na tri celine, to jest naslova:

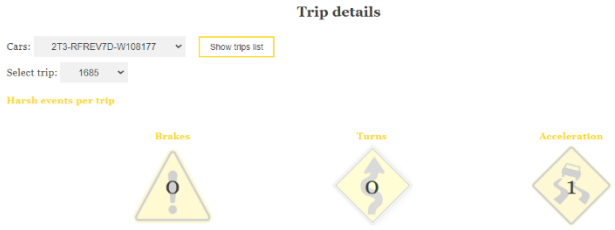
- *OBDII data* – deo o podacima sa automobila
- *Interval Settings* – deo o podešavanju intervala
- *Trip details* – deo za prikaz informacija o putovanjima

Na slici 7. prikazan je segment stranice vezan za odabir intervala. U polje je potrebno uneti *VIN* u odgovarajućem formatu. Ukoliko je format ispravno odabran iz padajućeg menija selektuje se željeni interval i pritiskom na dugme *Set interval* se sačuva.



Slika 7. *Postavljanje intervala za određeni automobil*

Centralni deo stranice zauzima deo o informacijama o rutama. Prvo je potrebno odabrati automobil, potom označiti željeno putovanje, nakon čega se dobije ispis o broju zabeleženih događaja, kao na slici 8.



Slika 8. Prikaz dela veb stranice za ispis zabeleženih događaja

Ispod informacija o događajima na veb stranici se prikazuje i mapa, sa označenim lokacijama početka i kraja putovanja, kao i tačkama u kojima se detektovani događaj desio.

5. ZAKLJUČAK

Cilj rada bila je implementacija sistema za detekciju grube vožnje. Tokom testiranja nije se naišlo na probleme ili greške. Demo snimci testiranja kao i spisak celokupne literature koja je korišćena pri implementaciji nalazi se na linku [6].

U nekim nadogradnjama sistema treba razmisliti o povećanju broja parametara koji se uzorkuju iz vozila, zatim njihovoj daljoj obradi na serverskoj strani. ESP32 delu sistema mogu se dodati senzori za lokaciju i akcelerometar i samim tim ceo sistem možemo učiniti jednostavnijim, ukloniti potrebu za *Android* aplikacijom. Server se može unaprediti dodavanjem novih funkcionalnosti gde bi korisnik mogao da bira parametre koji bi se očitavali.

6. LITERATURA

[1] https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/twai.html, pregledano septembar 2024.

[2] https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial, pregledano septembar 2024.

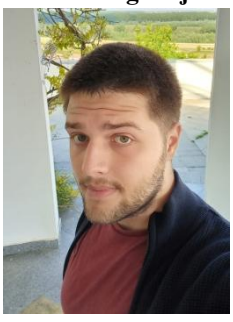
[3] https://www.csselectronics.com/pages/obd2-explained-simple-intro, pregledano septembar 2024

[4] Shah, Dhruti . Node.JS Guidebook. BPB Publications, 2018.

[5] https://www.elektronika.ftn.uns.ac.rs/umrezeni-embeded-sistemi/wp-content/uploads/sites/176/2018/03/UES-04-Front-end-i-Back-end.pdf , pregledano septembar 2024.

[6]https://drive.google.com/drive/folders/1qxiSQiPd3afc4I2gNT2fz6Po85KV1k5W?usp=drive_link

Kratka biografija:



Aleksandar Petrović rođen je 1997. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo – Embeded sistemi i algoritmi odbranio je 2020. god.

kontakt:
aleksandarpetrovic21897@gmail.com