



УНИВЕРЗИТЕТ У НОВОМ САДУ ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД
Департман за рачунарство и аутоматику
Одсек за рачунарску технику и рачунарске комуникације

ЗАВРШНИ (BACHELOR) РАД

Кандидат: Тијана Срејић

Број индекса: РА 30-2012

Тема рада: Реализација механизма за испоруку догађаја у апликативном окружењу возила

Ментор рада: доц. др Милан Бјелица

Нови Сад, септембар, 2016



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Тијана Срејић		
Ментор, МН:	Милан Ђелица		
Наслов рада, НР:	Реализација механизма за испоруку догађаја у апликативном окружењу возила		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2016		
Издавач, ИЗ:	Ауторски репринг		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилога)	7/33/5/5/12/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	Аутомобилска индустрија, спрега човек-машина		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	<p>Модерна возила карактеришу инфо-забавни системи (енг. <i>infotainment</i>), који укључују све више екрана. Екрани захтевају догађаје са различитих уређаја. Поставља се питање како униформно распоређивати догађаје ка свим екранима.</p> <p>Рад приказује једно решење за испоруку догађаја у апликативном окружењу возила. Циљ реализације је униформно слање догађаја од уређаја до апликација које користе догађаје. Описаны су и могући догађаји у возилу, као и начин њихове испоруке.</p>		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	проф. др Илија Башичевић	
	Члан:	доц. др Иван Каштелан	Потпис ментора
	Члан, ментор:	доц. др Милан Ђелица	



KEY WORDS DOCUMENTATION

Accession number, ANO:		
Identification number, INO:		
Document type, DT:	Monographic publication	
Type of record, TR:	Textual printed material	
Contents code, CC:	Bachelor Thesis	
Author, AU:	Tijana Srejić	
Mentor, MN:	Milan Bjelica, PhD	
Title, TI:	Implementation of event dispatching mechanism in vehicle application environment	
Language of text, LT:	Serbian	
Language of abstract, LA:	Serbian	
Country of publication, CP:	Republic of Serbia	
Locality of publication, LP:	Vojvodina	
Publication year, PY:		
Publisher, PB:	Author's reprint	
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)	7/33/5/5/12/0	
Scientific field, SF:	Electrical Engineering	
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, S/KW:	Automotive, HMI	
UC		
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, N:		
Abstract, AB:	<p>Modern vehicle consists of an infotainment system including a multi-screen environment. These screens require events from various HMI devices. The problem is how to dispatch these events to different screens.</p> <p>This paper describes implementation of event dispatching mechanism in vehicle application environment. The goal is to implement a uniform event dispatching mechanism from devices to applications.</p> <p>This paper also gives a listing of typical vehicle events and provides methods for event dispatching.</p>	
Accepted by the Scientific Board on, ASB:		
Defended on, DE:		
Defended Board, DB:	President:	Ilija Bašičević, Ph.D
	Member:	Ivan Kaštelan, Ph.D
	Member, Mentor:	Milan Bjelica, Ph.D
		Menthor's sign

Zahvalnost

Zahvaljujem se doc. dr Milanu Bjelici i Mileni Milošević na stručnoj pomoći i savetima za izradu ovog rada.

Posebno zahvaljujem svojoj porodici na pruženoj podršci u toku čitavog školovanja.

Na kraju zahvaljujem se svima onima koji su na bilo koji način doprineli izradi ovog rada.



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



SADRŽAJ

1.	Uvod	7
2.	Teorijske osnove.....	9
2.1	Pregled događaja u vozilu.....	9
2.2	HMI.....	11
2.2.1	Mehaničke sprege.....	11
2.2.2	Akustičke sprege	11
2.2.3	Vizuelne sprege	12
2.3	Pregled protokola za komunikaciju u vozilu.....	13
2.3.1	Controller Area Network (CAN).....	14
2.3.2	Local Interconnect Network (LIN).....	14
2.3.3	FlexRay.....	15
2.3.4	Ethernet.....	15
2.4	Publish-subscribe šablon.....	15
3.	Koncept rešenja	17
3.1	Opisa aplikativnog okruženja.....	17
3.2	Prikaz softverske arhitekture HMI agenta.....	19
4.	Programsko rešenje.....	20
4.1	Opis komunikacije sa pretplatnicima	20
4.2	Pregled programskih modula	21
4.2.1	Modul Agent.....	22
4.2.2	Moduli Steering_wheel, Sensor, Pedals	24
4.2.3	Modul IVI.....	25
	Modul App	25
4.2.4	Car_simulation modul	26



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



5.	Rezultati.....	27
5.1	Opis okruženja za ispitivanje	27
5.2	Ispitni slučajevi	28
6.	Zaključak	30
7.	Literatura	31

SPISAK SLIKA

Slika 2-1 Primeri mehaničkih HMI uređaja (volan, ručni kontroler, pedale)	11
Slika 2-2 HUD (Head-up display)	12
Slika 2-3. Klaster ekran	12
Slika 2-4 IVI ekran	13
Slika 2-5 Dijagram publish-subscriber šablona	16
Slika 3-1 Aplikativno okruženje HMI Agenta.....	17
Slika 3-2 Softverska arhitektura HMI Agenta	19
Slika 3-3 Softverska arhitektura HMI Agenta	19
Slika 4-1 MCS dijagram uspostave veze pretplatnika sa HMI Agentom	21
Slika 5-1 Okruženje za ispitivanje	27
Slika 5-2 Filmora okruženje za merenje odziva	28
Slika 5-4 Prikaz rezultata meranja sa i bez HMI Agenta.....	29
Slika 5-3 Prikaz rezultata merenja odziva na klaster ekranu	29

SPISAK TABELA

Tabela 1. Pregled HMI događaja u vozilu	10
Tabela 2. Poređenje protokola na osnovu prenosa podataka	14
Tabela 3. Lista dostupnih topika	18
Tabela 4. Pregled modula	21
Tabela 5. Funkcije programske sprege	22

SKRAĆENICE

ABS	- <i>Anti-lock braking system</i> , Sistem za sprečavanje blokiranja točkova
CAN	- <i>Controller Area Network</i>
HMI	- <i>Human Machine Interface</i> , Dispečer
IVI	- <i>In-Vehicle-Infotainment</i> , Multimedijalni ekran
LIN	- <i>Local Interconnect Network</i>
OLED	- <i>Organic Light Emitting Diode</i> , Organska svetleća dioda
RPM	- <i>Revolution per minute</i> , Obrtaj (motora) u minuti

1. Uvod

Današnja moderna vozila uključuju različite info-zabavne sisteme (eng. *infotainment*). Info-zabavni sistem je izraz kojim se opisuje kombinovanje informacionog sadržaja (eng. *information*) sa zabavnim sadržajem (eng. *entertainment*). Takođe, postoji tendencija povećavanja broja ekrana za prikaz ovakvog sadržaja. Aplikacije izvršavane na ovim ekranima zahtevaju informacije o događajima sa različitih HMI (eng. *Human machine interface* – Sprega čovek-mašina) uređaja. Primer ovih uređaja su pedale, volan, spoljni uređaj kao što je mobilni telefon, tablet i sl. Komunikaciju između HMI uređaja i aplikacija na različitim ekranima usložnjava činjenica da se različiti ekran mogu nalaziti na različitim hardverskim uređajima na kojima se nalaze različiti softverski stekovi. Ovim se stvara zahtev za mehanizmom koji će omogućiti uniformno slanje događaja ka različitim softverskim stekovima ali i omogućiti nezavisnost HMI uređaja od ovakvog okruženja sa više ekrana.

U ovom radu opisano je jedno moguće rešenje ovog mehanizma. Ovo rešenje podrazumeva postojanje agenta, koji će omogućiti uvođenje apstrakcije svih HMI uređaja prema aplikativnim slojevima. Agent oponaša publish-subscribe šablon, gde se HMI uređaji ponašaju kao pružaoci usluga (eng. *publishers*). U ovom slučaju usluge su HMI događaji. Dok se aplikacije iz različitih aplikativnih okruženja ponašaju kao korisnici usluga (eng. *subscribers*).

Ovaj rad je sačinjen od 7 poglavlja.

U drugom poglavlju je opisano šta predstavljaju HMI uređaji, zatim su opisani protokoli za komunikaciju. Dat je i opis mogućih događaja u vozilu kao i opis publish-subscribe šablon-a.

U trećem poglavlju dat je prikaz aplikativnog okruženja kao i softverska arhitektura rešenja.

Četvrto poglavlje opisuje softverske module, dok peto opisuje način ispitivanja softvera i rezultate testiranja.

Šesto poglavlje sadrži kratak opis šta je urađeno, predstavljeni su nedostaci rešenja kao i planovi za dalje unapređenje rešenja.

2. Teorijske osnove

U ovom poglavlju dat je pregled tipičnih događaja u vozilu, koji bi se mogli dostavljati do ekrana. Zatim je dat opis okruženja vozila i HMI uređaja u vozilu kao i pregled protokola za komunikaciju u vozilu. Takođe je i navedeno koji događaji se najčešće prenose navedenim protokolima. Pored toga, opisan je i šablon publish-subscriber, koji je korišćen u ovom rešenju za dostavljanje događaja.

2.1 Pregled događaja u vozilu

Događaji u vozilu se mogu svrstati u više vrsta:

1. Aplikativni događaji, koji se razmenjuju između ekrana;
2. Događaji koji stižu sa HMI uređaja. Ti događaji su prikazani u Tabeli 1;
3. Dogadaji koji obaveštavaju vozača o parametrima rada vozila, koji olakšavaju vožnju i čine je bezbednijom. Ovi dogadaji takođe mogu stizati sa nekih HMI uređaja i senzora.

Neki primeri aplikativnih događaja su:

- Obaveštenje o promeni radio stanice;
- Obaveštenje o telefonskom pozivu;
- Obaveštenje o trajanju telefonskog poziva.

HMI uređaj	Dogadjaj
Pedale	Pritisak pedale Gas
	Pritisak pedale Kvačilo
	Pritisak pedale Kočnica
Volan	Pritisak tastera na volanu: Levo/desno/gore/dole
	Pritisak tastera na volanu: Enter/back
	Pritisak tastera na volanu za pojačanje ili smanjenje zvuka
	Pritisak tastera na volanu za davanje glasovne komande
	Pritisak tastera za meni
Senzor	Položaj volana (ugao za koji je volan okrenut)
Ručni kontroler	Pritisak tastera za navigaciju kroz meni
	Pritisak tastera za Radio
	Pritisak tastera za CD
	Pritisak tastera za Telefonski poziv
	Pritisak tastera za pokretanje navigacije
Spoljni uređaj	Informacije o profilu korisnika

Tabela 1. Pregled HMI događaja u vozilu

Primeri dogadaja promene stanja vozila su:

- prelasci između 6 mogućih stanja: P-R-N-D-2-1 (parking – reverse – neutral – driving - second gear - first gear);
- obaveštenja o stanju goriva u vozilu;
- informacije o brzini;
- informacije nadmorskoj visini;
- temperatura motora;
- spoljna temperatura;

- obaveštenje o promeni vrednosti RPM-a(eng. *Revolution per minute* – obrtaj (motora) u minuti) itd.

Takođe, mogu se prenositi i komande za zaključavanje vrata, kontrolu klima uređaja, kontrolu sedišta i krova i sl.

2.2 HMI

Najveći broj navedenih događaja u vozilu dolazi sa HMI uređaja [1]. Ove sprege mogu imati upravljačku ulogu, najčešće su to mehaničke sprege. Takođe, mogu služiti i za pružanje informacija koje olakšavaju vožnju i čine je bezbednjom. HMI tehnologije se mogu podeliti na 3 grupe: Mehaničke, akustičke i vizuelne.

2.2.1 Mehaničke sprege

U mehaničke sprege svrstavaju se pedale, menjač, volan, ručni kontroler (eng. *jog shuttle*) kao i različite vrste tastera. Na slici 2-1 su prikazani primeri mehaničkih uređaja.



Slika 2-1 Primeri mehaničkih HMI uređaja (volan, ručni kontroler, pedale)

2.2.2 Akustičke sprege

Akustičke sprege su dugo bile najčešća izlazna sprega u vozilima. Njihova prednost je u tome što ne zahtevaju da vozač skreće pogled, što ih čini sigurnijim od vizuelnih sprega. Neke od ovih su sprega koje daju različite zvučne signale upozorenja (npr. vrata su otvorena, upozorenje da je pojas za vezivanje neupotrebljen, svetla su ostala uključena i sl.) kao i različiti ulazi za glasovne komande.

2.2.3 Vizuelne sprega

Vizuelnim spregama pripadaju različite vrste ekrana (eng. *display*) kao što su LCD, oLED, Head-Up Display (HUD) i instrument tabla (klaster). Pored nabrojanih vizuelnih sprega, u vozilima se često pojavljuje i in-vehicle infotainment (IVI) ekran.

Head-up Display (Slika 2-2) se razlikuje od ostalih ekrana po tome što ne zahteva skretanje pogleda vozača, jer se sadržaj projektuje na staklo ispred vozača. Ovaj displej obično prikazuje informacije kao što su brzina vozila, ograničenje brzine, odstojanje vozila itd.



Slika 2-2 HUD (Head-up display)

Analogna instrument tabla se zamenjuje digitalnom (Slika 2-3), u obliku klaster ekrana. Instrument tablu čine instrumenti kao što su brzinomer, obrtomer, digitalni merač pređenog puta i sl. Pored ovih informacija digitalne instrument table prenose i informacije kao što su navigacija, nadmorska visina, različite informacije o gorivu kao što su stanje, potrošnja i upozorenje o trenutnom nivou.



Slika 2-3. Klaster ekran

In-vehicle infotainment (Slika 2-4) [2] omogućuje funkcionalnosti kao što su upravljanje audio sadržajima i njihova reprodukcija, korišćenje navigacije pri vožnji, primanje i slanje SMS poruka, telefoniranje kao i pristup internetu i povezivanje sa pametnim telefonima preko USB-a ili Bluetooth-a.



Slika 2-4 IVI ekran

2.3 Pregled protokola za komunikaciju u vozilu

Razvoj tehnologije u automobilskoj industriji doveo je do pojave sve više elektronike u vozilima. Najzastupljeniji mrežni komunikacioni protokoli u automobilskoj industriji su Controller Area Network (CAN), FlaxRay, Local Interconnect Network (LIN) i Ethernet [3].

Tabela 2. prikazuje najčešće korišćene magistrale i podatke koji se kroz njih šalju. U tabeli je data i brzina prenosa ovih podataka [4].

Magistrala	Brzina	Aplikacija
LIN	10K to 125Kbps (body)	Lampe, svetlo, zaključavanje vrata, podešavanje sedišta itd.
CAN	125K to 1 Mbps (status information)	Elektronski indikatori, informacije o vožnji, automatsko podešavanje klime, detekcija grešaka itd.
FlexRay	1M to 10Mbps (real-time control)	Kontrola kočnice i gasa, ABS.
MOST/1394	Preko 10 Mbps (multimedia)	Informacije vezane za navigacioni sistem, audio sistem itd.
Ethernet	Preko 1 Gbps (multimedia)	Informacija sa radara, infotainmenta, IP kamere itd.

Tabela 2. Poređenje protokola na osnovu prenosa podataka

2.3.1 Controller Area Network (CAN)

CAN je serijski komunikacioni protokol sa podrškom za distribuirano slanje komandi u realnom vremenu sa veoma velikom sigurnošću i brzinom do 1 Mbps. CAN magistrala je idealna u sredinama zagađenim šumom. CAN protokol je osetljiv na događaje (eng. *event-triggered*). Neke od karakteristika ove magistrale su:

- CAN je magistrala sa više vodećih kontrolera (eng. *multi-master*), što omogućava izgradnju složenih sistema, bez potrebe uvođenja centralizovanog vodećeg kontrolera. Svi kontroleri mogu da šalju i primaju poruke ali ne u isto vreme.
- Za distribuciju podataka se koristi emiterski (eng. *broadcast*) prenos, uz garanciju da će svi umreženi uređaji primiti poslatu poruku u istom obliku.
- CAN je prihvaćen kao standard od strane Evropske automobilske industrije.

CAN magistrala se koristi pre svega za komunikaciju različitih kontrolera ili procesora putem poruka. Primer uređaja koji se povezuje na CAN je ručni kontroler.

2.3.2 Local Interconnect Network (LIN)

LIN je serijska magistrala u kojoj postoji jedan glavni i tipično do šesnaest podređenih kontrolera. Komunikacija počinje tako što glavni kontroler emituje zaglavje sa

identifikatorom poruke. Zaglavljene dobijaju svi kontrolери. Kontrolер koji je zadužen da šalje poruku sa tim identifikatorom, počinje sa slanjem podataka. Celokupnu komunikaciju inicira glavni kontrolер tako da ne postoji mogućnost kolizije. Glavna motivacija pri razvoju LIN magistrale je bila jeftinija serijska komunikacija za funkcionalnosti koje nisu zahtevne sa stanovišta brzine prenosa podataka. LIN poruka nosi maksimalno 8 bajtova korisnih podataka, a maksimalna brzina prenosa podataka je $\sim 19,6\text{ kBit/s}$. Fizička veza je uglavnom realizovana preko jedne žice.

LIN magistrala prenosi informacije vezane za podešavanje okruženja kao što su podešavanje sedišta, zaključavanje vrata i sl. Ove informacije mogu biti korisne klaster aplikaciji.

2.3.3 FlexRay

FlexRay je komunikacioni protokol promovisan od strane mnogih proizvođača vozila. FlexRay je serijska magistrala u kojoj svaki kontroler ima određen vremenski okvir u kojem ima ekskluzivno pravo pristupa magistrali. Vremenski okviri se ciklično ponavljaju. Prednost FlexRay magistrale u odnosu na CAN i LIN magistralu su veća brzina prenosa podataka, veća otpornost na greške, determinizam i tačno definisani vremenski okvir kad svaka EKJ šalje podatke. Svaka poruka nosi maksimalno 254 bajtova korisnih podataka. Maksimalna brzina prenosa podataka je $\sim 10\text{ MBit/s}$.

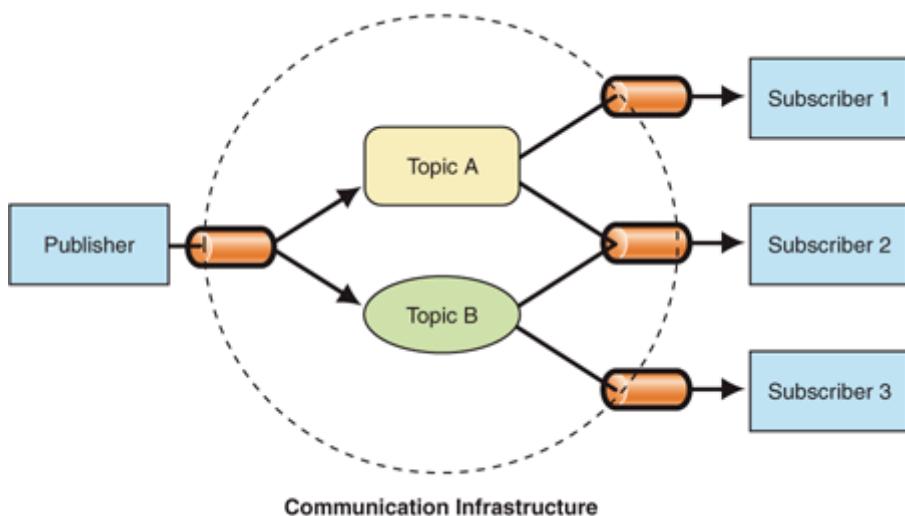
2.3.4 Ethernet

Za mrežnu komunikaciju u automobilskoj industriji u upotrebi je i Ethernet. Komunikacija preko Ethernet veze je zasnovana na događajima, kao kod CAN protokola. Najveća prednost Ethernet veze je veoma visok propusni opseg, zbog čega se Ethernet koristi za dostavljanje informacija sa radara, IP kamera, kao i informacija sa nabrojanih magistrala na ekrane kao što su IVI, klaster, head-up displej.

2.4 Publish-subscribe šablon

Publish-subscribe [5] je šablon za razmenu poruka, gde pošiljalac poruke, koji se naziva pružalac usluga (eng. *publisher*), ne šalje poruku direktno ka određenom primaocu, koji se naziva pretplatnik (eng. *subscriber*), već se poruke za slanje grupišu u klase odnosno topike (eng. *topics*). S druge strane, pretplatnik označava koji od ponuđenih topika su mu od interesa i samo te topike će i dobijati.

Prednost ovog šablonu je nezavisnost pružaoca usluga i pretplatnika, gde pružaoci usluga ne moraju znati za postojanje pretplatnika. Ova osobina je naročito značajna kod sistema koji se mogu proširivati, kakavi su i info-zabavni sistemi. U info-zabavnim sistemima može se javiti potreba za dodavanjem novih HMI uređaja, pa dodavanje novog HMI uređaja neće uticati na pretplatnike. Takođe, ovaj šablon omogućuje veću skalabilnost nego neka druga tradicionalna rešenja (npr. klijent-server arhitektura).



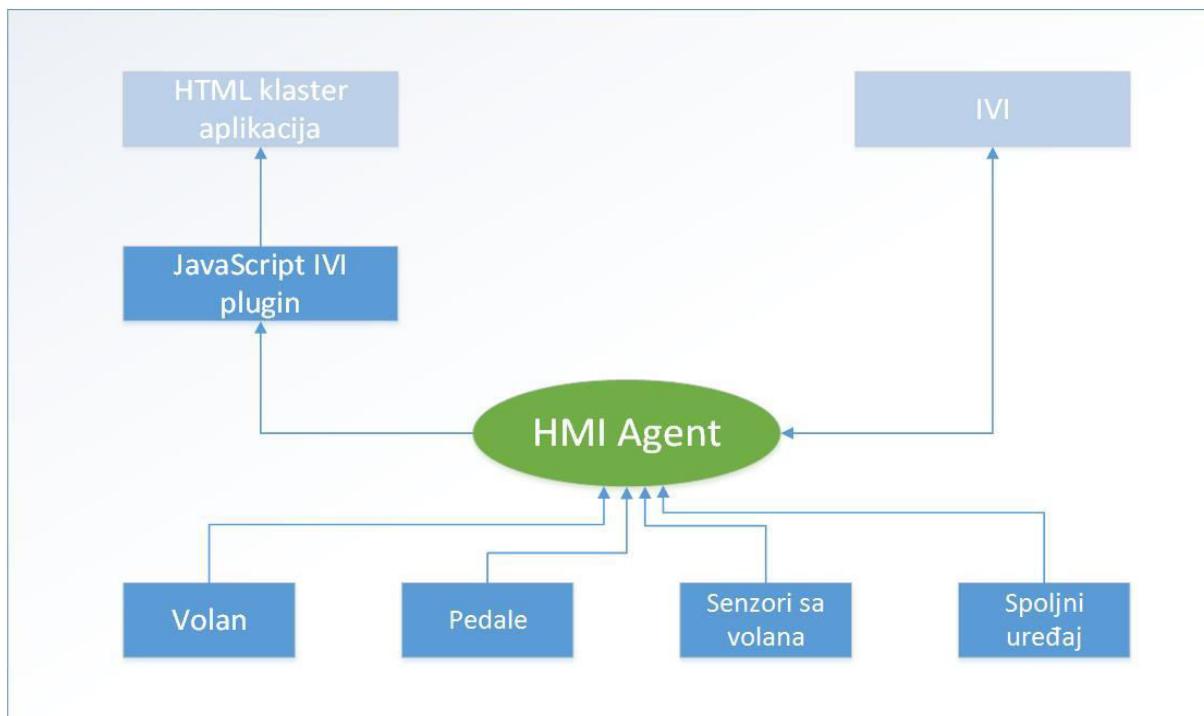
Slika 2-5 Dijagram publish-subscriber šablonu

3. Koncept rešenja

Rešenje je u formi HMI Agenta. U ovom poglavlju opisano je aplikativno okruženje agenta. Kao i moduli od kojih je sačinjen HMI Agent i veze između njih.

3.1 Opisa aplikativnog okruženja

Na slici 3-1. je prikazano aplikativno okruženje u kome se nalazi HMI agent. HMI agent se povezuje na dostupne HMI uređaje. Iznad HMI agenta nalaze se pretplaćene aplikacije klaster i IVI. Klaster aplikacija sa HMI Agentom komunicira preko JavaScript dodatka.



Slika 3-1 Aplikativno okruženje HMI Agenta

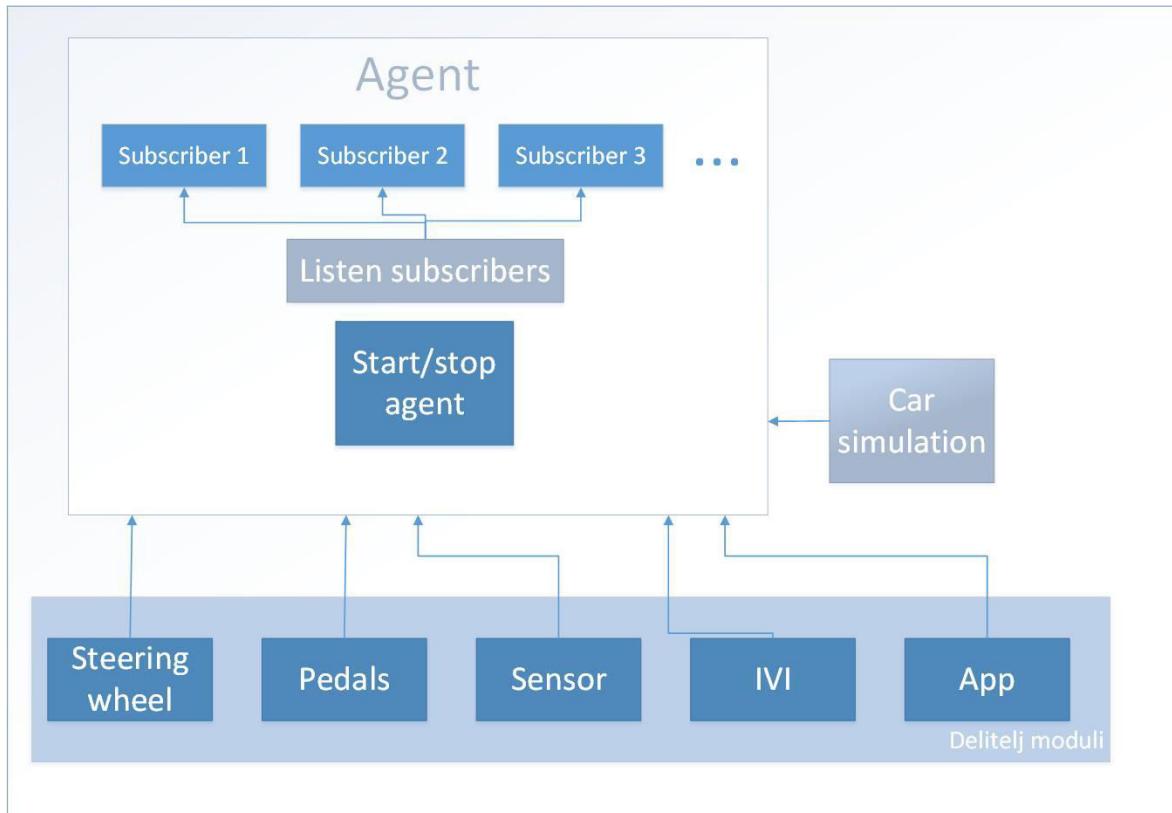
JavaScript dodatak se povezuje sa HMI Agentom i preplaćuje se na sve topike potrebne klaster aplikaciji. Pored Klaster aplikacije na HMI Agentu povezuje se i IVI, koji takođe očekuje određene topike. Pored zahteva za događajima sa HMI uređaja, klaster aplikacija zahteva i događaje sa IVI-a, što znači da se IVI ponaša i kao preplatnik i kao pružalac usluga. Ovim aplikacijama su dostupni sledeći topici prikazani u tabeli 2.

	Naziv topika	Opis	Događaji
1.	Steering wheel	pritisak tastera na volanu	Pritisak tastera levo/desno/gore/dole/enter/back, Pritisak tastera za pojačanje i smanjenje zvuka, pritisak tastera za davanje glasovne komande, pritisak tastera za meni;
2.	Pedals	pritisak pedale i informacija o intenzitetu pritiska	Pritisak gasa, pritisak kočnice;
3.	Steering wheel sensor	položaj volana	Ugao za koji je volan zaokrenut;
4.	IVI	informacije koje šalje IVI klasteru	Promena radio stanice, telefonski poziv, trajanje telefonskog poziva;
5.	App	informacije sa spoljašnjeg uređaja (mobilnog telefona)	Promena profila korisnika;
6.	Car simulation parameters	simulirani parametri vozila	Prelasci između 6 mogućih stanja (P-R-N-D-2-1), obaveštenje o stanju goriva u vozilu, informacije o brzini, informacije u nadmorskoj visini, temperatura motora, spoljna temperatura, obaveštenje o promeni vrednosti RPM-a.

Tabela 3. Lista dostupnih topika

3.2 Prikaz softverske arhitekture HMI agenta

Na slici 3-3. je prikazana softverska arhitektura HMI agenta.



Slika 3-2 Softverska arhitektura HMI Agenta

HMI agent se može izdeliti na sledeće module:

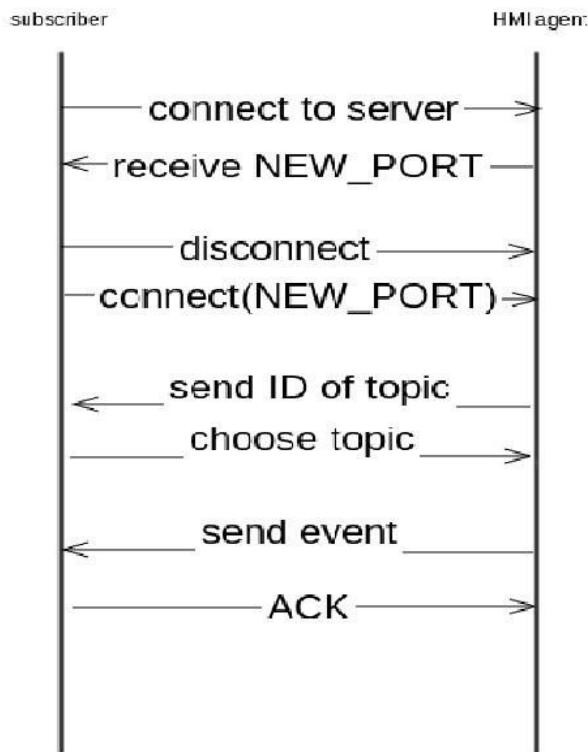
- Modul *Agent*, čija je uloga inicijalizacija i pozivanje ostalih modula, prihvatanje zahteva za uspostavom veze od strane preplatnika, kao i slanje događaja preplatnicima;
- Delitelj moduli, koji se dalje mogu podeliti u 3 vrste:
 - Moduli za prikupljanje podataka sa HMI uređaja i smeštanje u strukture za slanje ka preplatnicima. U ove module ubrajaju se Modul *Steering wheel*, Modul *Pedals* i Modul *Sensor*;
 - Moduli za komuniciranje sa aplikacijama na drugim ekranima (Modul *IVI*);
 - Moduli za komuniciranje sa spoljnim uređajima, npr. aplikacija na mobilnom telefonu (Modul *App*);
- Modul *Car simulation*, za simulaciju nekih parametara u vozilu, kao što su brzina, RPM, stanje goriva u vozilu, trenutna potrošnja goriva, nadmorska visina, temperatura vozila itd.

4. Programsко rešenje

Za izradu softvera korišćen je Ubuntu Linux 14.04 operativni sistem, dok se C prevodi za Yocto operativni sistem koji se izvršava na PC računaru. U ovom radu dat je predlog rešenja za prihvat i prosleđivanje HMI događaja sa ciljem da se obezbedi nezavisnost HMI uređaja od aplikacija. U ovom poglavlju opisani su izrađeni programski moduli i dat je pregled funkcija aplikativne programske spreme.

4.1 Opis komunikacije sa preplatnicima

Za svaki HMI uređaj implementirana je biblioteka koja omogućuje pristup uređaju i realizuje njegove funkcionalnosti. HMI agent koristi te biblioteke, povezuje se na uređaj i prihvata događaje sa uređaja. Ove uređaje HMI agent posmatra kao pružaoce usluga (eng. *publishers*). Za potrebe testiranja implementirana je biblioteka koja implementira funkcionalnosti korisnika usluga ili preplatnika (eng. *subscriber*). Ovu biblioteku mogu koristiti komponente koje koriste usluge HMI agenta (Plug-in, IVI). Takođe je moguće implementirati sopstvenu subscriber biblioteku, poštujući proceduru povezivanja, koja je prikazana na slici 4-1. Na slici je prikazan proces uspostave veze, zatim po uspostavi veze preplatnik obaveštava HMI agenta koji događaji su mu od interesa. Svi događaji su svrstani u grupe. Ova grupa se u publish-subscribe obrascu naziva topik (eng. *topic*). Posle odabira grupe HMI događaja, HMI agent po stizanju sa HMI uređaja prosleđuje događaje preplatnicima.



Slika 4-1 MCS dijagram uspostave veze pretplatnika sa HMI Agentom

4.2 Pregled programskih modula

Pregled svih modula u softveru prikazan je u Tabeli 4.

Moduli	Datoteke izvornog koda
<i>Agent</i>	Agent.h Agent.c
<i>Steering_wheel</i>	Steering_wheel.h Steering_wheel.c
<i>Sensor</i>	Sensor.h Sensor.c
<i>Pedals</i>	Pedals.h Pedals.c
<i>IVI</i>	Ivi.h Ivi.c
<i>App</i>	App.h App.c
<i>Car simulation</i>	Car_simulation.h Car_simulation.c

Tabela 4. Pregled modula

U tabeli 5. date su sve funkcije programske sprege i moduli kojima pripadaju.

Funkcija	Modul
void* PSS_listenSubscribers(void* ptr);	Agent
void* PSS_sendTopics(void* ptr)	
tPSS_Error PSS_receiveTopics()	
tPSS_Error PSS_initTopics()	
tPSS_Error PSS_StartServer()	
tPSS_Error PSS_StopServer()	
void PSS_getSteeringWheelEvents(tARDKEYB_KeyboardEvent* a)	Steering wheel
void PSS_getSensorEvents(int SensorEvent)	Sensor
void PSS_getPedalsEvents(tARDP_pedalsEvent*pedalsEvent)	Pedals
void* PSS_receiveIVIconnection(void* ptr)	IVI
void* PSS_receiveAppConnection(void* ptr)	App
void PSS_callCarUpdate(union sigval signalArg)	Car simulation

Tabela 5. Funkcije programske sprege

4.2.1 Modul Agent

Modul *Agent* je zadužen za inicijalizaciju i deinicijalizaciju agenta. Tokom inicijalizacije pokreće se programska nit zadužena za osluškivanje zahteva za uspostavljanjem veze od strane preplatnika. Veza između HMI Agenta i preplatnika je zasnovana na TCP vezi. U ovoj programskoj niti nalazi se TCP server, zadužen za osluškivanje veze na prethodno definisanom portu. Sa druge strane preplatnik mora instancirati modul TCP klijent, pomoću kojeg će uspostaviti vezu na prethodno definisanom portu. Kada se uspostavi veza, HMI agent zauzima jedan port koji će dodeliti preplatniku a potom mu šalje taj port preko uspostavljene veze. Posle slanja novog porta, prethodna veza se raskida a uspostavlja se veza

na novom portu. Posle uspostavljanja veze, stvara se nova programska nit, koja će upravljati vezom sa preplatnikom.

```
typedef struct
{
    int      topicID;
    char     topicName[kPSS_MAX_NAME_SIZE];
    char*   topicData;
    int      topicSubscribers[kPSS_MAX_NUM_OF_SUBSCRIBERS];
} tPSS_Topic;
```

Ova struktura služi za smeštanje informacija o topiku. Kao polja sadrži ID i ime topika, podatke koje treba poslati i ID preplatnika koji očekuju ovaj topik.

```
typedef struct
{
    char*   subData;
    int      subIsActive;
} tPSS_Subscriber;
```

Ova struktura služi za smeštanje informacija o preplatniku, kao polja sadrži podatke iz svih topika koje treba poslati kao i njegovo stanje da li je aktivan.

```
void* PSS_listenSubscribers(void* ptr);
```

Ova funkcija je zadužena za prihvatanje veze na podrazumevanom portu, poziva se u posebnoj programskoj niti. Po prihvatanju veze, unutar nje se formiraju posebne programske niti za svakog preplatnika.

Opis funkcije:

- **Argument:** void* - mogućnost konverzije u bilo koji tip
- **Povratna vrednost:** - pokazivač na nit

```
void* PSS_sendTopics(void* ptr)
```

Uloga ove funkcije je da upravlja slanjem događaja ka preplatniku. Ova funkcija se takođe nalazi u posebnoj programskoj niti. Za svakog preplanika se formira posebna programska nit koja izvršava ovu funkciju. Ove niti se kreiraju unutar *PSS_listenSubscribers* funkcije.

```
tPSS_Error PSS_initTopics()
```

Zadatak ove funkcije je pregled raspoloživih topika i slanje tih topika preplatniku.

Opis funkcije:

- **Argument:** -
- **Povratna vrednost:** kod greške

```
tPSS_Error PSS_receiveTopics()
```

Zadatak ove funkcije je prihvatanje strukture sa rednim brojevima odabralih topika od strane preplatnika, njeno parsiranje i smeštanje u strukturu PSS_topic.

Opis funkcije:

- **Argument:** -
- **Povratna vrednost:** kod greške

```
tPSS_Error PSS_StartServer()
```

Ova funkcija je zadužena za inicijalizaciju struktura u serveru i pokretanje programskih niti.

Opis funkcije:

- **Argument:** -
- **Povratna vrednost:** kod greške

```
tPSS_Error PSS_StopServer()
```

Zadatak ove funkcije je deinicijalizacija servera i gašenje programskih niti.

Opis funkcije:

- **Argument:** -
- **Povratna vrednost:** kod greške

4.2.2 Moduli **Steering_wheel, Sensor, Pedals**

U modulima *Steering_wheel, Sensor, Pedals* implementirane su funkcije povratnog poziva (eng. *callback*), koje pozivaju biblioteke za pristup HMI uređaju. Unutar ovih funkcija prihvataju se događaji i smeštaju u odgovarajuće strukture za slanje preplatnicima.

```
void PSS_getSteeringWheelEvents(tARDKEYB_KeyboardEvent* a)
```

Opis funkcije:

- **Argument:** pokazivač na strukturu koja sadrži informacije: kod tastera i informaciju o pritisku tastera
- **Povratna vrednost:** -

```
void PSS_getSensorEvents(int SensorEvent)
```

Opis funkcije:

- **Argument:** ugao za koji je volan zaokrenut
- **Povratna vrednost:** -

```
void PSS_getPedalsEvents(tARDP_pedalsEvent* pedalsEvent)
```

Opis funkcije:

- **Argument:** pokazivač na strukturu koja sadrži informacije o pritisku pedale kao i stepenu pritiska
- **Povratna vrednost:** -

4.2.3 Modul IVI

U modulu *IVI* realizovana je funkcija za uspostavu veze i prihvata događaja sa *IVI* ekrana. Događaji koji se trenutno šalju su: Promenjena radio stanica, Desio se poziv, Poziv je u toku itd. Postoji mogućnost proširivanja ovih događaja, a nove događaje definišu aplikacije na *IVI* ekranu.

```
void* PSS_receiveIVIconnection(void* ptr)
```

Opis funkcije:

- **Argument:** - mogućnost konverzije u bilo koji tip
- **Povratna vrednost:** - pokazivač na niti

Modul App

Modul *App* realizuje iste funkcionalnosti kao modul *IVI*, ali se u Modulu *App* uspostavlja veza sa spoljnim uređajem, u ovom slučaju to je mobilni telefon. Neki od događaja koje se prenose su: Obaveštenje o kvaru vozila, obaveštenje o stanju goriva, profil vozača... I za ove događaje postoji ista mogućnost proširivanja kao i za *IVI* skup događaja.

```
void* PSS_receiveAppConnection(void* ptr)
```

Opis funkcije:

- **Argument:** - mogućnost konverzije u bilo koji tip
- **Povratna vrednost:** - pokazivač na niti

4.2.4 Car_simulation modul

Modul *Car_simulation* simulira neke parametre u vozilu. Ti parametri su dati u tabeli 4.

Parametar	Merna jedinica
Brzina	Km/h
Predeni put	Km
Mod vožnje	P/R/N/D/1/2
RPM	Rev/min
Ukupno pređeni put	Km
Preostali put	Km
Stanje goriva	L
Trenutna potrošnja goriva	l/100 km
Nadmorska visina	M
Temperatura motora	°C
Spoljna temperatura	°C

Tabela 4. Simulirani parametri

5. Rezultati

U ovom poglavlju opisano je okruženje za ispitivanje, prikazani su ispitni slučajevi i performanse programskog rešenja.

5.1 Opis okruženja za ispitivanje

Programsko rešenje je testirano na Yocto operativnom sistemu (distribucija Linux operativnog sistema), koji se nalazi na Fujitsu računaru sa procesorom Intel i7 i 8G RAM-om. Na slici 5-1 prikazano je okruženje za ispitivanje.

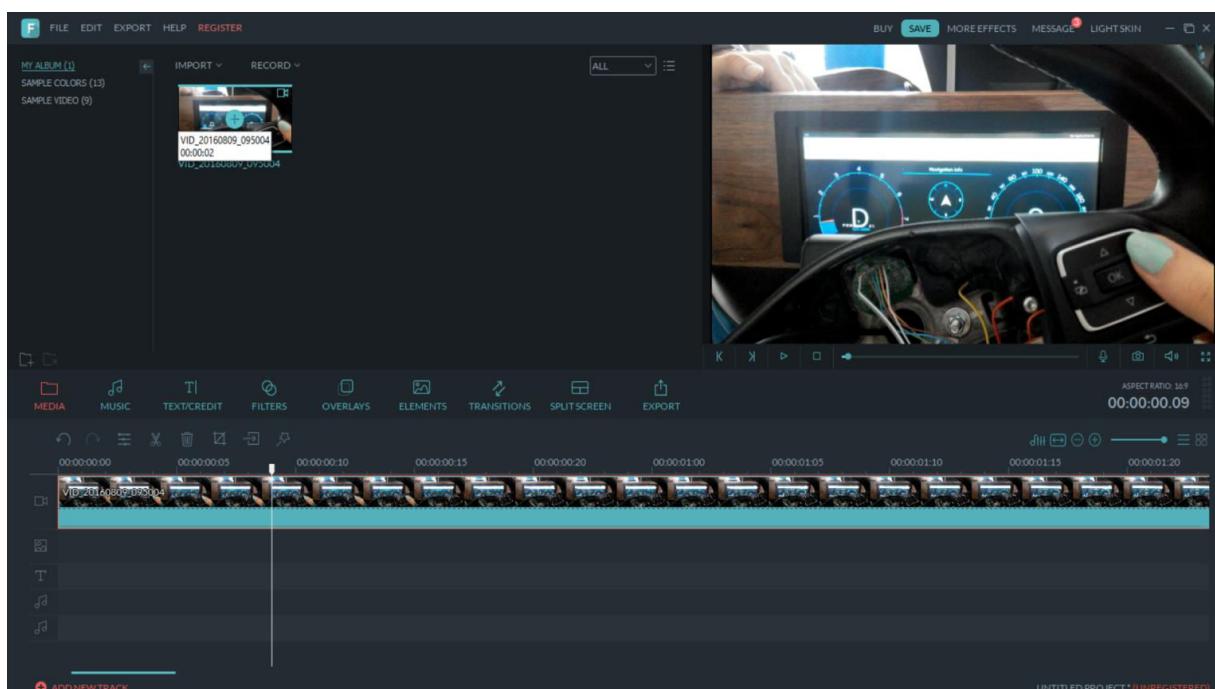


Slika 5-1 Okruženje za ispitivanje

5.2 Ispitni slučajevi

Radi testiranja robusnosnosti aplikacije formirani su unapred zadati ispitni slučajevi.

- Na HMI Agentu je povezano 6 pretplatnika pretplaćenih na sve raspoložive topike, što bi trebalo da predstavlja najveće moguće opterećenje u realnim uslovima korišćenja. Ispitivanje je trajalo 15 sati. Testirana je stabilnost aplikacije i ispitivano da li dolazi do curenja memorije. Pokazalo se da aplikacija radi stabilno i nije primećeno curenje memorije. Ispitivanje je prošlo.
- Ispitano je i ponašanje HMI Agentu u slučaju stalnog povezivanje pretplatnika i raskidanja veze. Ispitivani su portovi koji se dodeljuju novim pretplatnicima. Ispitivanje je pokazalo da se portovi uspešno oslobođaju i dodeljuju sledećem prijavljenom pretplatniku. Ispitivanje je prošlo.



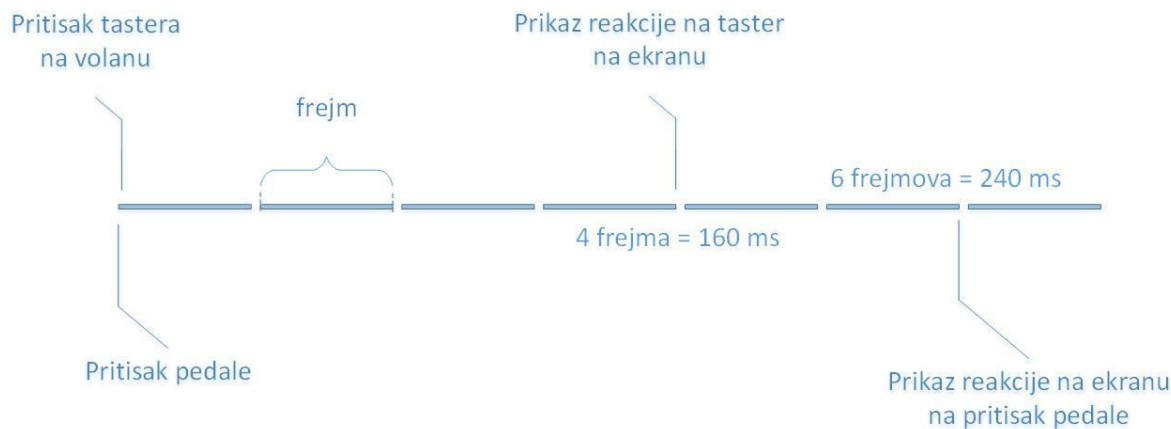
Slika 5-2 Filmora okruženje za merenje odziva

Performanse i potrošnja celokupne aplikacije mereni su uz pomoć posebnog watchdog-a za Chromium Web pregledač. Na osnovu te provere ustanovljeno je da aplikacija troši prosečno 38% od ukupnog CPU-a, što je 41% u odnosu na ceo sistem. Prosečna potrošnja RAM memorije je 1.8MB, što je 0.2% u odnosu na ceo sistem.

Ukupan odziv meren je pomoću telefonske kamere, snimanjem događaja tako da se u istom kadru nalaze i HMI uređaj čiji se odziv testira i ekran na kojem se sadržaj prikazuje. Zatim je snimak obrađen u video editoru, pomoću kojeg je utvrđeno vreme koje protekne od

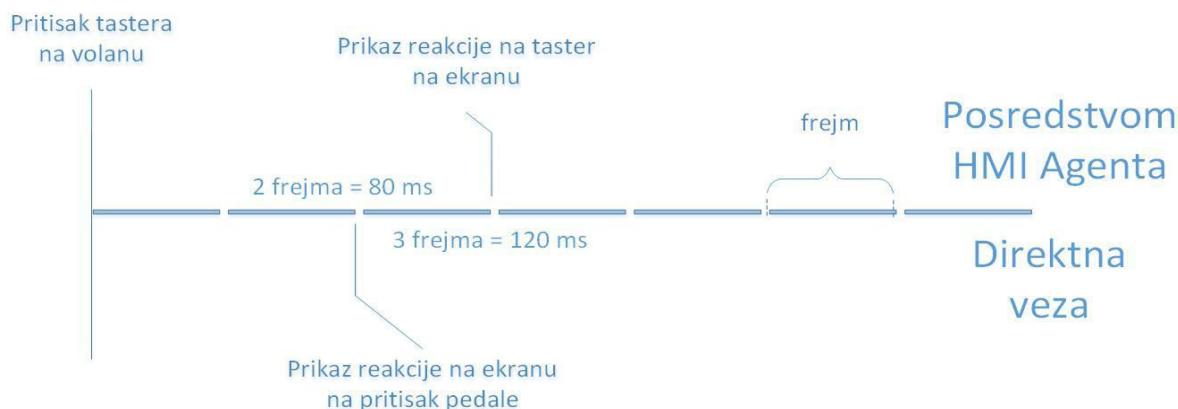
pritiska tastera, odnosno pedale, do prikaza događaja na ekranu. Korišćen je Filmora editor (Slika 5-2), koji prikazuje 25 frejmova u sekundi.

Slika 5-3 prikazuje vreme odziva na klaster ekranu za dva slučaja: pritisak tastera na volanu i pritisak pedale.



Slika 5-3 Prikaz rezultata merenja odziva na klaster ekranu

Kako bi se utvrdilo koliko kašnjenje unosi HMI Agent u celokupan sistem, vršeno je sledeće merenje na dva načina: merenje vremena od pritiska na taster volana do prikaza reakcije na IVI ekranu, sa i bez HMI Agenta. U prvom slučaju kod prisustva HMI agenta događaj prolazi od volana kroz HMI Agent do IVI ekrana, s tim da je okruženje maksimalno opterećeno (preko Ethernet veze pored komunikacije agenta sa ekranima, vrši se i prenos videa ka IVI ekranu). U drugom slučaju volan je direktno povezan na IVI ekran. Razlika je predstavljena na slici 5-4.



Slika 5-4 Prikaz rezultata meranja sa i bez HMI Agenta

Od ukupnog vremena, merenjem unutar softvera utvrđeno je da događaj unutar samog HMI Agenta provede manje od 2 milisekunde, a ostatak kašnjenja unosi mrežna komunikacija. Merenje unutar softvera je izvršeno tako što je mereno vreme od trenutka kada neki od događaja stiže sa HMI uređaja do trenutka pre slanja događaja ka aplikacijama.

6. Zaključak

Razvoj info-zabavnih sistema prati pojava novih ekrana i HMI uređaja. Ovaj rad prikazuje jedno moguće rešenje komunikacije između ovih komponenti. U okviru projektnog zadatka realizovan je mehanizam za uniformno slanje događaja od HMI uređaja do ekrana. Rešenje je osmišljeno da bude proširivo, što podrazumeva mogućnost dodavanja novih HMI uređaja i ekrana. Prednost rešenja je i mogućnosnost da se na različitim ekranima nalaze različiti softverski stekovi.

Dalji rad na rešenju se odnosi na integraciju sa stvarnim komponentama u vozilu. U trenutnom rešenju podaci o vozilu se softverski simuliraju, dok u realnim okruženjima ovi podaci dolaze sa CAN magistrale.

7. Literatura

[1] Human-machine interface pristupljeno avgust 2016. URL:

http://www.mogi.bme.hu/TAMOP/jarmurendszerek_iranyitasa_angol/math-ch04.html

[2] In-vehicle infotainment pristupljeno avgust 2016. URL:

<http://www.webopedia.com/TERM/I/in-vehicle-infotainment-ivi.html>

[3] Miladin Jovanović, Dejan Čotra, Radisav Stamenković, Velibor Ilic, "Programska implementacija obrade signala elektronskih kontrolnih jedinica u automobilskoj mreži", 23nd Telecommunications Forum (TELFOR 2015),, At Belgrade, Serbia, Volume: 23, Novembar 2015.

[4] LIN, CAN, FlexRay, MOST, Ethernet comparation pristupljeno avgust 2016. URL:

http://www.eetimes.com/author.asp?doc_id=1322342

[5] P. Eugster, P. Felber, R. Guerraoui, A. Kermarrec, "The Many Faces of Publish/Subscribe", ACM Computing Surveys, Vol. 35, No. 2, Jun 2003.