|  |
| --- |
| Роберт Шандор  **Реализација софтверске магистрале за дистрибуцију видео сигнала у возилу на „Adaptive AUTOSAR“ платформи**  МАСТЕР РАД  Нови Сад, 2019 |

|  |  |  |  |
| --- | --- | --- | --- |
| Редни број, **РБР**: | |  | |
| Идентификациони број, **ИБР**: | |  | |
| Тип документације, **ТД**: | | Монографска документација | |
| Тип записа, **ТЗ**: | | Текстуални штампани материјал | |
| Врста рада, **ВР**: | | Дипломски – мастер рад | |
| Аутор, **АУ**: | |  | |
| Ментор, **МН**: | |  | |
| Наслов рада, **НР**: | |  | |
| Језик публикације, **ЈП**: | | Српски / ћирилица | |
| Језик извода, **ЈИ**: | | Српски | |
| Земља публиковања, **ЗП**: | | Република Србија | |
| Уже географско подручје, **УГП**: | | Војводина | |
| Година, **ГО**: | |  | |
| Издавач, **ИЗ**: | | Ауторски репринт | |
| Место и адреса, **МА**: | | Нови Сад; трг Доситеја Обрадовића 6 | |
| Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога) | |  | |
| Научна област, **НО**: | | Електротехника и рачунарство | |
| Научна дисциплина, **НД**: | | Рачунарска техника | |
| Предметна одредница/Кqучне речи, **ПО**: | |  | |
| **УДК** | |  | |
| Чува се, **ЧУ**: | | У библиотеци Факултета техничких наука, Нови Сад | |
| Важна напомена, **ВН**: | |  | |
| Извод, **ИЗ**: | |  | |
| Датум прихватања теме, **ДП**: | |  | |
| Датум одбране, **ДО**: | |  | |
| Чланови комисије, **КО**: | Председник: |  |
|  | Члан: |  | Потпис ментора |
|  | Члан, ментор: | Проф. Др. Милан З. Бјелица |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 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**: | |  | |
| Mentor, **MN**: | |  | |
| Title, **TI**: | |  | |
| 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/appendixes) | |  | |
| Scientific field, **SF**: | | Electrical Engineering | |
| Scientific discipline, **SD**: | | Computer Engineering, Engineering of Computer Based Systems | |
| Subject/Key words, **S**/**KW**: | |  | |
| **UC** | |  | |
| Holding data, **HD**: | | The Library of Faculty of Technical Sciences, Novi Sad, Serbia | |
| Note, **N**: | |  | |
| Abstract, **AB**: | |  | |
| Accepted by the Scientific Board on, **ASB**: | |  | |
| Defended on, **DE**: | |  | |
| Defended Board, **DB**: | President: |  |
|  | Member: |  | Menthor's sign |
|  | Member, Mentor: |  |  |

**Захвалност**

**Садржај**

[1. Увод 1](#_Toc21031959)

[2. Теоријске основе 4](#_Toc21031960)

[2.1 *AUTOSAR* 4](#_Toc21031961)

[2.1.1 *AUTOSAR* *Classic* 5](#_Toc21031962)

[2.1.2 *AUTOSAR Adaptive* 6](#_Toc21031963)

[2.1.2.1 *Execution* *Management* 9](#_Toc21031964)

[2.1.2.2 *State* *Management* 10](#_Toc21031965)

[2.1.2.3 *Communication* *Management* 10](#_Toc21031966)

[2.2 Магистрале 10](#_Toc21031967)

[2.2.1 Хардверске магистрале 11](#_Toc21031968)

[2.2.2 Софтверске магистрале 11](#_Toc21031969)

[2.3 Хардверска платформа 12](#_Toc21031970)

[2.3.1 *ALPHA Automotive Development platform* 13](#_Toc21031971)

[3. Концепт решења 14](#_Toc21031972)

[3.1 Софтверска магистрала за дистрибуцију видео сигнала 14](#_Toc21031973)

[3.2 Слој за апстракцију камера наменске платформе 16](#_Toc21031974)

[3.3 Слој за апстракцију дистрибуције добављеног видео сигнала 16](#_Toc21031975)

[3.4 Слој за информисање о доступности и карактеристикама прибављеног видео сигнала 18](#_Toc21031976)

[4. Програмско решење 19](#_Toc21031977)

[5. Резултати 20](#_Toc21031978)

[6. Закључак 21](#_Toc21031979)

[7. Литература 22](#_Toc21031980)

**Списак слика**

Слика 2.1 Детаљни приказ чланова *AUTOSAR* конзорцијума [7] 5

Слика 2.2 Приказ комуникације у *SOA* адаптивне платформе [12] 8

Слика 2.3 Преглед доступних компоненти адаптивне платформе [12] 9

Слика 2.4 Дијаграм прелаза стања приликом покретања *ЕМ* 9

Слика 2.5 *ALPHA* развојна платформа [5] 12

Слика 2.6 Дијаграм хардверских компоненти *ALPHA* развојне платформе [5] 13

Слика 3.1 Слојевита архитектура концепта решења софтверске магистрале 15

Слика 3.2 Приказ зависности слоја за апстракцију хардвера и хардверске платформе 16

Слика 3.3 Зависност слоја за дистрибуцију видео сигнала од осталих компоненти система 17

**Списак табела**

**No table of figures entries found.**

**Скраћенице**

**ADAS – A**dvanced **D**river-**a**ssistance **s**ystems – Напредни системи за помоћ возачу у вожњи

**API** – **A**pplication **P**rograming **I**nterface – апликативна програмска спрега/интерфејс

**ARA** – **A**UTOSAR **R**untime for **A**daptive **A**pplications – *AUTOSAR* извршно окружење за апликације адаптивне платформе

**АUTOSAR - AUT**omotive **O**pen **S**ystem **A**rchitecture – Стандард отворене архитектуре у аутомобилској индустрији

**CM** – **C**ommunication **M**anagement

**DSP** – **D**igital **S**ignal **P**rocessor - наменски процесор за обраду дигиталног сигнала

**EM – E**xecution **M**anagement

**IPC - I**nter **P**rocess **C**ommunication – Међупроцесна комуникација

**LiDAR – Li**ght **D**etection **a**nd **R**anging – детекција објеката и њихове удаљености путем светлосног снопа

**OEM** – **O**riginal **E**quipment **M**anfuacturer – произвођач оригиналне опреме

**POSIX – P**ortable **O**perating **S**ystem **I**nterface – Интерфејс преносивог оперативног система

**RADAR – Ra**dio **D**etection **a**nd **R**anging – детекција објеката и њихове удаљености путем радио таласа

**RGB – R**ed **– G**reen **– B**lue – формат енкодовања видео сигнала, где сваки од канала представља вредност једне боје (црвене, зелене и плаве)

**RPC** - **R**emote **P**rocedure **C**all – Позивање функционалности удаљеног модула

**RTOS** *-* **R**eal-**t**ime **o**perating **s**ystem – оперативни систем за рад у реалном времену

**SOA** – **S**ervice **O**riented **A**rchitecture – сервисно оријентисана софтверска архитектура

**SM** – **S**tate **M**anagement

**SoC - S**ystem **o**na **C**hip – систем заснован на чипу

**USB – U**niversal **S**erial **B**us – универзална серијска веза

**YUV –** формат енкодирања видео сигнала где **Y** компонента представља осветљај, а **U** и **V** компоненте представљају боје.

# Увод

Решавајући бројне изазове аутономне вожње, аутомобилска индустрија данашњице се у многоме ослања на помоћ напредних алгоритама за помоћ возачу (енг. *ADAS -**Advanced Driver-assistance systems*). Иако се *ADAS* алгоритми разликују по степену безбедности који морају да испуне, свака информација од значаја мора правовремено бити достављена, како самом систему који управља возило, тако и самом путнику. Функционална безбедност [1] самих алгоритама се значајно разликује. Тако, одређени алгоритми могу бити информативног типа, попут оних који приказују окружење возила у току његовог мировања [2], док са друге стране постоје алогритме за избегавање судара [3], чије отказивање може имати катастрофалне последице по путнике. Велики број оваквих алгоритама се развија и тестира у контролисаним лабораторијским условима. Ипак, крајњи циљ како произвођача возила, тако и твораца ових алгоритама, јесте њихово извршавање на платформама специјализованим за аутономну вожњу.

Како би извршавање сваког од ових алгоритама било оптимално, потребно је на циљним платформама извршити дистрибуцију послова, попут прикупљања, припреме и обраде информација од интереса. Овим се уводи јасна граница између различитих сензора, попут камера, ласерских (енг. *LiDAR – Light Detection and Ranging*) и радио даљинометра (енг. *RADAR – Radio Detection and Ranging*), уређаја који преводе сигнал из аналогног у дигитални и циљне платформе која врши обраду сигнала и екстрахује корисне информације из истог.

Овакав приступ омогућује оптималан рад и уску специјализацију сваке од компоненти у систему. Ипак, иако су перформансе сваке од компоненти значајно побољшане, архитектура система је хетерогена и као један од већих проблема истиче се дистрибуција података. Произвођачи хардверских компоненти тренутно немају униформни одговор на овај проблем, у својим решењима ослањају се на хардверске магистрале великих брзина преноса података, које најчешће прати програмска подршка затвореног кода.

Ипак, зарад олакшавања развоја и стандардизације софтвера за аутономну вожњу стварају се конзорцијуми који окупљају велике произвођаче аутомобила и аутомобилске индустрије. Један такав конзорцијум представља *AUTOSAR* [4]*,* који окупља произвођаче аутомобила као што су *BMW, Volkswagen, Toyota, Chrysler* и произвођаче опреме као што су *Bosch, Continental, Siemens VDO.* Циљ овог конзорцијума јесте развој стандарда којег сваки произвођач мора да се придржава, али задржава слободу у имплементацији свог решења. Један од стандарда који овај конзорцијум развија, *Adaptive AUTOSAR*, тиче се платформи које имају велику процесну моћ и намењене су управо за ADAS алгоритме.

Како је овај стандард још увек у настајању, те захтеви који морају бити испуњени нису строго дефинисани, овај рад представља једно могуће решење дистрибуције видео сигнала у оквиру *AUTOSAR Adaptive* платформе. Рад се ослања на употребу софтверске магистрале која је реализована апстракцијом етернет (енг. *ethernet*) магистрале и апстракције платформе која врши прикупљање видео сигнала, зарад вршења дистрибуције истог кроз остатак система. Као циљна платформа, употребљена је Алфа [5] развојна плоча (енг. *ALPHA Automotive Development platform)*, заснована на *TDA2x* систему на чипу (енг. *SoC, System on a Chip*), компаније *Texas Instruments*.

Овај рад је организован у следећих 5 целина:

1. **Теоријске основе –** преглед *AUTOSAR* стандарда и *AUTOSAR Adaptive* платформе. Опис магистрала, као и поређење софтверских и хардверских магистрала. Опис хардверске платформе на којој је тестирано идејно решење.
2. К**онцепт решења -** поглавље даје увид у реализоване модуле софтверске магистрале. Преглед и опис слојева за апстракцију хардверске платформе.
3. **Програмско решење –** реализација модула софтверске магистрале.
4. Т**естирање и валидација –** поглавље даје увид у перформансе реализоване софтверске магистрале као и њену функционалну употребу у два различита сценарија.
5. **Закључак –** ово поглавље даје преглед реализоване софтверске магистрале, резултата који су остварени, као и кораци за даље унапређење софтверске магистрале.

# Теоријске основе

Ово поглавље даје увид у неопходне теоријске основе неопоходне за разумевање реализованог решења. Посебна пажња посвећена је софтверској платформи *AUTOSAR Adaptive* на којој је заснована ова реализација софтверске магистрале за дистрибуцију видео сигнала. Како је *AUTOSAR Adaptive* платформа сервисно оријентисане архитектуре, део поглавља објашњава парадигме заступљене у оваквом развоју софтвера. Поред тога, направљен је осврт на хардверске и софтверске магистрале, као и на хардверску платформу на којој је реализовано и тестирано ово решење.

## *AUTOSAR*

*AUTOSAR* представља конзорцијум који окупља велике произвођаче возила, као и опреме за возила. Оформљена 2003. године, ова групација је имала за циљ да смањи како трошкове производње, тако и потребан труд и време за преношење постојећих решења на друге платформе [6]. У оквиру овог конзорцијума, прави се разлика између група чланова. Подела се врши на следећи начин:

1. Чланови оснивачи (енг. *Core Partner)* – чине девет првобитних чланова који су започели формирање овог конзорцијума.
2. Премијум чланови (енг. *Premium Members)* – чини их 46 чланова, где они могу бити произвођачи возила, произвођачи оригиналне опреме (енг. *Original Equipment Manufacturers, ОЕМ*), произвођачи различитих софтверских алата, полупроводничких компоненти, итд. Ови чланови су се касније прикључили конзорцијуму.
3. Партнери за равој и производњу (енг. *Development partners*) – ову групу чини 28 чланова који су задужени за развој компоненти које задовољавају прописане стандарде ове групације.
4. Сарадници (енг. *Associates and Attendees*) – последњу групацију у овој хијерархији чине компаније које сарађују са члановима свих осталих група и користе тренутно доступни стандард.

Слика 2.1 даје детаљнији увид у хијерархију ове групације.



Слика . Детаљни приказ чланова *AUTOSAR* конзорцијума [7]

Ова групација изродила је два стандарда. Први, старији, *AUTOSAR Classic* и други,  *AUTOSAR Adaptive*, који се још увек развија. Зарад стварања јасне слике и намене, како једног, тако и другог стандарда, следећа два потпоглавља позабавиће се сваким од стандарда, како би се поставила основа за решење које овај рад износи.

### *AUTOSAR* *Classic*

Овај стандард представља првобитну платформу коју је оформљени конзорцијум створио. Циљна хардверска платформа подразумева микроконтролерске јединице које имају малу процесорску моћ, ограничене меморијске ресурсе. На оваквим платформама извршава се оперативни систем за рад у реалном времену (енг. *Real-time operating system*, RTOS). Задатак ових система јесте вршење временски критичних радњи, обзиром да сваки оперативни систем за рад у реалном временом има јасно дефинисане оквире у којима се одређена операција може извршити. Управо су овакве хардверске и софтверске платформе идеалне за разноврсне функционалност које се тичу управљања возила и на њима се заснивају системи попут кочионог система возила

Иако се овај стандард показао као поуздан у пракси, обзиром да га многобројни произвођачи аутомобила управо уграђују у своја возила, недостатак овог стандард и саме платформе јесте његова непримењивост на платформе велике процесне моћи.

Потреба за великом процесном моћи јавља се као последица унапређења самих сензора, који су сада у могућности да прикупе значајно више информације из свог окружења и самим тиме омогуће алгоритмима на које ослања возач да буду знатно прецизнији. Ова појава за последицу има потребу за новим стандардом, који би подржап како софтверске, тако и хардверске платформе.

*AUTOSAR Adaptive* управо представља овакав стандард. Стварањем њега, не укида се потреба за постојањем *AUTOSAR Classic*. Циљ стандарда, а и самог конзорцијума јесте коегзистирање ових двају платформи.

Наредно поглавље детаљније описује *AUTOSAR Adaptive.*

### *AUTOSAR Adaptive*

Као што је већ напоменуто, ова платформа представља искорак аутомобилске индустрије у смеру стандардизације софтверске платформе која ће омогућити знатно већу процесорску моћ и комуникацију са спољним светом.

Задатак самог конзорцијума јесте стварање стандарда који ће омогућити комуникацију разноврсних електронских контролних јединица (енг. *ECU – Electronic Control Unit*) путем етернет магистрале [8]. Управо 100-мегабитни и гигабитни етернет представљају основе за пренос информација кроз систем и управо је циљ да оне преузму примат у комуникацији. Претходно коришћене магистрале у систему неће бити избачене из употребе, већ ће њихова намена бити ограничена. Управо употреба етернет магистрале омогућује флексибилнији приступ у протоку информација које стижу са разноврсних сензора. Поред тога, етернет магистрала овде омогућује и софтверске подршке на електронским компонентама, у току животног века једног возила, што претходна платформа није пружала.

Поред напретка у комуникацији који овај стандард доноси, велики напори су уложени у омогућавању употребе процесора и компоненти високих перформанси, како би потребе комплексних алгоритама биле задовољене. Управо овај стандард у својој имплементацији стога не представља засебни оперативни систем, већ искључиво његово проширење. Као што је случај код *AUTOSAR Classic* платформе, тако *AUTOSAR Adaptive* платформа подржава већи број хардверских платформи на којима ће се она налазити.

Како би овакво проширење могло бити примењено на оперативни систем, потребно је да систем подржава *POSIX* [9]стандард, специфично *POSIX51.* Како *Linux* оперативни систем задовољава овај стандард, он представља један од оперативних система који је погодан за ово проширење.

Ово проширење оперативног система са собом носи *AUTOSAR* извршно окружење за апликације адаптивне платформе (енг. *ARA – AUTOSAR Runtime for Adaptive Applications*). Оно пружа решења које се тичу основних функционалности апликација, као што су системски позиви, комуникација са локалним и удаљеним сервисима, као и сервисима које пружа сама платформа. Овај апликативни интерфејс (енг. *API* – *Application programing interface*) пружа могућност лаке интеграције већ постојећих апликација у саму адаптивну платформу. Поред тога, постојећи *ARA* *API* је стандардом дозвољено проширити уколико нека од потребних компоненти за функционисање апликација не улази у скуп функционалности *POSIX51*.

Адаптивна платформа је заснована на сервисно оријентисаној софтверској архитектури (енг. *SOA* – *Service Oriented Architecture*) [10]. Одлика овакве архитектуре огледа се могућности постојања већег броја независних софтверских компоненти, тј. сервиса, који посредством мреже или међупроцесне комуникације размењују информације [11]. Оваква софтверска архитектура омогућује велику модуларност и поновну употребу већ постојећег изворног кода из потребне измене. Због своје модуларности, ова архитектура се врло често користи у *Web* технологијама.

Механизам комуникације између два сервиса дат је на слици 2.2.



Слика . Приказ комуникације у *SOA* адаптивне платформе [12]

Као што се на слици може приметити, потребно је да постоји је постоји једна софтверска компонента која прати постојање сервиса који пружају различите услуге, тј. информације. Компоненте које пружају услуге (енг. *Provider*), морају да буду видљиве остатку система. То чине тиме што се представе (енг. *Register*) модулу који је задужен за праћене доступних сервиса. Сервиси који зависе од информација, такозвани претплатници (енг. *Subscriber*), прво морају претражити доступне сервисе. Уколико је тражени сервис доступан, тада је могуће добити вршити удаљене позиве њихових функционалности (енг. *RPC - Remote Procedure Call*). Сама адаптивна платформа функционише по овом принципу.

Адаптивна платформа се ослања на овај принцип рада. Доступност сервиса адаптивне платформе је такође могуће мењати у току времена рада саме платформе (енг. *Runtime*) као и у току дизајнирања сервиса. Особине сервиса покрива датока која описује сваки од доступних сервиса (енг. *Service Instance Manifest*). Ова датотека се испоручује са самим извршним кодом, како би се јасно нагласили сви протоколи које апликација користи у својој комуникацији. Поред овог манифеста, који је од већег значаја за овај рад, у адаптивној платформи се користе још два. *Execution manifest* чији је задатак да опише када се који сервис и апликација извршава на адаптивној платформи и *Machine Manifest*, чији је задатак да пружи детаљнији опис саме хардверске платформе и стања у којима ова платформа може да се налази.

Слика 2.3 даје детаљнији увид у сервисе адаптивне платформе који су доступни.



Слика . Преглед доступних компоненти адаптивне платформе [12]

Детаљнији преглед функционалности адаптивне архитектуре дат је у следећим поглављима, кроз опис функционалности сервиса и модула који су од значаја за овај рад.

#### *Execution* *Management*

Ова компонента представља модул задужен покретање, како платформе, тако сервиса и апликација које се налазе на платформи. Приликом покретања оперативног система, ова компонента је прва прозвана. Њен задатак је да, из манифеста који описују платформу и остале апликације, распореди када је потребно да се која апликација покрене. *EM* нема за задатак да врши распоређивање апликација у току њиховог извршавања, како је то дужност оперативног система, чије је адаптивна платформа проширење. Дијаграм прелаза стања дат је на слици 2.4.



Слика . Дијаграм прелаза стања приликом покретања *ЕМ*

Поред свог задужења покретања адаптивне платформе и њених апликација у оквиру хладног (енг. *Cold boot*)или поновног покретања (енг. *Reboot/Restart*) оперативног система, *ЕМ* је компонента која је такође задуже за руковање животним циклусом апликација адаптивне платформе. Ова компонента, на основу информација које су доступне из датотека *Machine Manifest* и *Execution Manifest*, доноси закључак која апликација, односно сервис, треба да буде покренута у зависности од стања у којем се како адаптивна, тако и сама хардверска платформа налази. *EM* не врши прелазе платформе из једног стања у друго, већ се на то ослања на другу компоненту, *State Management* (*SM*).

Како је задужен за праћење стања адаптивних апликација и платформе, *ЕМ* је такође задужен и за враћање апликација у њихово претходно стање (енг. *Recovery*) када платформа прелази из једног стања у друго.

#### *State* *Management*

Ова компонента представља део адаптивне платформе који је задужен за праћење и прелазак између дефинисаних стања. *SM* је компонента која свој финални облик добија од стране систем интегратора и зависи од саме хардверске платформе за коју је адаптивна платформа намењена. Као што је већ речено, ова компонента у многоме зависи од саме платформе и намене исте, те стандард не поставља детаљне захтеве за реализацију ове компоненте, већ само обавезује ову компоненту да обавести остатак система преко којих интерфејса се са њом комуницира.

За комуникацију остатком система, *SM* се ослања на функционалности компоненте *Communication Management* (*CM*). Ова компонента задуже је комуникацију апликација и сервиса адаптивне платформе.

#### *Communication* *Management*

Као што је већ споменуто, ова компонента је задужена за комуникацију осталих компоненти у систему. Задатак *CM* јесте апстракција начина комуникација једне апликације, тј. сервиса са апликацијама и сервисима који се налазе на истој адаптивној платформи или чак на некој другој, удаљеној адаптивној платформи.

Начин комуникације апликација, тј. сервиса адаптивне платформе могуће је одредити у тренуцима дизајнирања апликације, њеног покретања или у току њеног самог рада, променом конфигурације апликације, односно сервиса.

## Магистрале

Магистрала (енг. *Communication bus*) представљају физичку компоненту чија је намена повезивање екстерних или интерних уређаја процесором. Под магистралом, у рачунарству, подразумевају се подједнако софтверске компоненте, тачније комуникациони протоколи, и хардверске компоненте, као што су оптичка влакна, каблови и многобројни други медијуми. Како би било могуће руковање самом магистралом, потребно је да постоји руковалаоц (енг. *Driver*) који пружа одређену дозу апстракције саме магистрале, како би њена употреба била лакша, за крајњег корисника.

Наредна два поглавља даће мало детаљнији опис ова два типа магистрала.

### Хардверске магистрале

Хардверске магистрале представљају директну везу између процесора и компоненте која може бити интерна или екстерна у односу на читав систем. Овај тип магистрала, поред потребе за постојањем физичког медијума преко којег се врши комуникација, такође захтева и постојање руковалаоца. Управо реализација руковалаоца олакшава употребу физичког медијума, јер он врши апстракцију комуникационог протокола оваквих магистрала.

Како би се омогућио оптималан рад и функционисање магистрале, потребно је да управо оне буду стандардизоване.Најбољи пример тога јесу магистрале попут универзалне серијске магистрале (енг. *USB – Universal Serial Bus*) [13], *PCI (Peripheral Component Interconnect)* и *PCI-Е (Peripheral Component Interconnect - Express).* Одлика ових магистрала у систему јесте могућност преношења података великим брзинама. Оне се налазе у персоналним рачунарима и њихова намена је разнолика.

Поред магистрала које су карактеристичне за искључиво за персоналне рачунарске системе, на наменским платформама постоје стандардизоване магистрале, које су карактеристичне различите гране индустрије. Тако магистрала попут *CAN (*енг. *Controller Area Network)* [14], *LIN* (енг. *Local Interconnected Network*) [15] представљају неке од стандарда који су карактеристични за аутомобилску индустрију и користе се како за контролне информација са микроконтролера, тако и за информације са различитих сензора.

### Софтверске магистрале

Софтверске магистрале представљају начин на који се остварује међупроцесна комуникација (енг. *IPC -* *Inter process communication*) [16]. Када посматрамо овај тип магистрала, оне не морају поседовати посебан физички медијум, како би се вршила размена информација. Са друге стране, као што је то случај и са хардверским магистралама, потребно је да постоји установљени протокол у комуникацији између пријемне и предајне стране, како би информација која тече системом била валидна.

Како софтверска магистрала не мора поседовати физички медијум за комуникацију, потребно је постојање оперативног система. Тако на пример *Linux*, оперативни систем отвореног кода (енг. *Open source*), нуди механизме комуникације као што је *D-bus* [17] система за размену порука, као и проточна обрада (енг. *Pipes*) [18] подржана у језгру (енг. *Kernel*) самог оперативног система. Поред ових видова комуникације, употреба дељене меморије (енг. *Shared memory*) која је такође подржана у самом оперативном систему или реализација неких других комуникационих протокола који нису део *Linux* оперативног система такође представљају својеврсну реализацију софтверске магистрале.

## Хардверска платформа

Како би наменска хардверска платформа задовољила потребе адаптивне платформе, на којој се заснива ово решење софтверске магистрале, она мора да задовољи неколико критеријума. Прво, као што је већ напоменуто, адаптивна платформа представља проширење оперативног система заснованог на најмање *PSE51* стандарду. Поред тога, платформа такође мора да поседује велику процесорску моћ и специфично за овај рад, мора поседовати камере за добављање видео сигнала који ће бити дистрибуиран кроз систем.

Једна оваква платформа је *ALPHA Automotive Development platform* дизајнирана и развијена на институ за Рачунарску Технику и рачунарске комуникације (РТ-РК).

Следеће поглавље даје детаљнији опис платформе хардверских компоненти и софтверске подршке за дату платформу.

На слици 2.5 налази се *ALPHA* развојна платформа.



Слика . *ALPHA* развојна платформа [5]

### *ALPHA Automotive Development platform*

Развојна платформа *ALPHA* заснована је на систему на чипу *TDA2x* [19] компаније *Texas Instruments.* Сама платформа поседује три оваква чипа, која поседују 2 *ARM Cortex A15* и 2 *ARM Cortex M4* језгра. Два чипа ове хардверске платформе покрећу оперативни систем за рад у реалном времену, *SysBios*, док преостали чип ове платформе покреће *Linux* оперативни систем. На платформи се налазе и два наменска процесора за обраду дигиталног сигнала (енг. *DSP – Digital Signal Processor*). Поред тога, како би се комуникација одвијала са што мањим кашњењем, на плочи се налази интерфејс који подржава гигабитни етернет.

Поменути чипови на којима је поменути *SysBios* задужени су за извршавање алгоритама за надгледање стања возача, детекцију слободног простора, детекцију возила у околини и сл. Преостали чип који покреће *Linux* оперативни систем задужен је за покретање адаптивне платформе и апликација.

Детаљан дијаграм компоненти *ALPHA* платформе налази се на слици 2.6.



Слика . Дијаграм хардверских компоненти *ALPHA* развојне платформе [5]

# Концепт решења

Ово поглавље даје опис решења постављеног проблема. Дат је приказ решења софтверске магистрале за дистрибуцију видео сигнала у два сценарија. Први представља дистрибуцију сигнала на платформи која генерише сам видео сигнал, док други описује дистрибуцију истог сигнала до удаљене платформе. Такође, у овом поглављу дат је логички приказ сваке од функционалних компоненти реализоване софтверске магистрале.

## Софтверска магистрала за дистрибуцију видео сигнала

Основна идеја софтверске магистрале за дистрибуцију видео сигнала јесте олакшавање и апстракција достављања овог садржај крајњем кориснику, тј. апликацији која врши манипулацију истог.

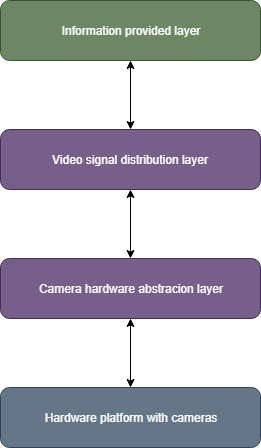
Наиме, вршењем апстракције добављања сигнала знатно се смањује вероватноћа недозвољеног приступа одређеним сегментима меморије. На тај начин је елиминсана могућност нарушавања интегритета и квалитета добављених података, јер апликација која врши обраду поменутог сигнала није у могућности да приступи оригиналним подацима, већ користи копију истих. Поред тога, постиже се и апстракција читаве платформе и на омогућује се фокус инжењера који развија софтвер базиран на алгоритму за асистенцију возачу. С тога, инжењер, па самим тиме и алгоритам, искључиво мора бити упознат са карактеристикама сигнала који добија.

Из приложених проблема, софтверска магистрала најпре мора да изврши апстракцију сензора, тј. камера са којих се добија видео сигнал, како би крајњег корисника исте лишила потребе познавања детаља хардверске платформе на којој се извршава алгоритам. Након тога, потребно је извршити апстракцију достављања видео сигнала кориснику, тј. апликацији која обрађује исти. Овиме се постиже додатни ниво апстракције којим се ограничава писање и читање одређених зона меморије који су намењени искључиво за сам сигнал. На крају, потребно створити пружити информације о сигналу који се доставља, као и његовој доступности. Те информације о сигналу представљају његов формат (нпр. *YUV422*, *YUV444*, *RGB*, итд.), затим меморијске захтеве за прихватања сигнала, као и у случају да је већи број доступних камера са једне хардверске платформе, информација са које камере, тј. из које перспективе долази сам видео садржај.

Из свих ових захтева уочава се да је потребно решење разложити на три слоја, где постоје следећи слојеви:

1. Слој за апстракцију камера наменске платформе;
2. Слој за апстракцију дистрибуције добављеног видео сигнала;
3. Слој за информисање о доступности и карактеристикама прибављеног видео сигнала.

Слика 3.1 даје увид у слојеве софтверске архитектуре овог решења.



Слика . Слојевита архитектура концепта решења софтверске магистрале

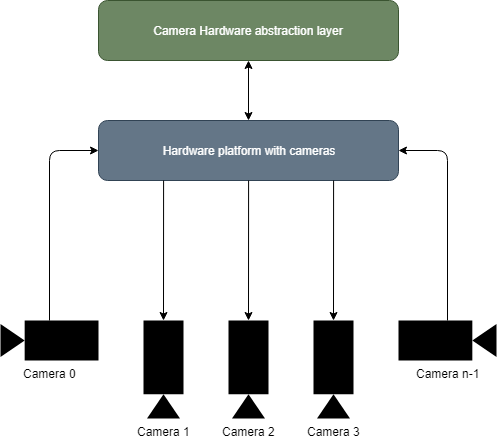
Следећа поглавља даће детаљнији увид у сваку од наведених компоненти, као и њихове међусобне зависности.

## Слој за апстракцију камера наменске платформе

Као што је већ напоменуто, први корак у реализацији софтверске магистрале представља сама апстракција хардверске платформе на којој се извршава софтверска магистрала. Тиме, инжењер који управо развија неки од алгоритама не мора детаљно да познаје саму хардверску платформу на којој се извршава алгоритам, обзиром да су детаљи попут конфигурације платформе и камера које се налазе на њој сакривени у овом слоју. Такође, инжењер који развија тај алгоритам, није оптерећен ни детаљима складиштења добављених слика са камере, већ то овај слој такође омогућује.

С тога, овај слој софтверске магистрале мора бити уско повезана са хардверском компонентом на којој се извршава.

Слика 3.2 даје детаљнији увид у организацију овог слоја софтверске магистрале.



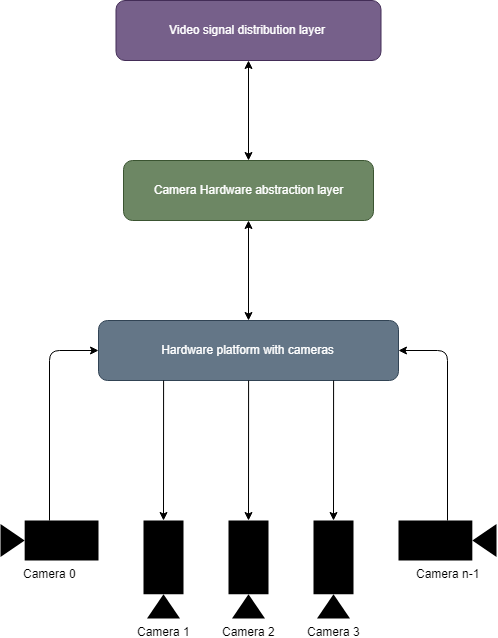
Слика . Приказ зависности слоја за апстракцију хардвера и хардверске платформе

## Слој за апстракцију дистрибуције добављеног видео сигнала

Овај слој софтверске магистрале је, за разлику од претходно описаног, у потпуности независан од хардверске платформе на којој се извршава. Међутим, овај слој се ослања на функционалности оперативног система за који је развијан.

Наиме, слој за апстракцију дистрибуције видео сигнала мора бити у могућности да пружи исти свакој апликацији која њега захтева, независно од платформе на којој се апликација извршава. Односно, овај слој мора бити у стању да комуницира како са апликацијама на локалној, тако и са апликацијама на удаљеној платформама.

Слика 3.3 представља зависност овог слоја од осталих компоненти система.



Слика . Зависност слоја за дистрибуцију видео сигнала од осталих компоненти система

Како овај слој мора бити способан да рукује меморијом у којој је складиштен добављени сигнал, зарад његове даље дистрибуције до апликације која њиме рукује, он је такође зависан од оперативног система на којем се извршава. Логички, овај слој се дели у две компоненте.

## Слој за информисање о доступности и карактеристикама прибављеног видео сигнала

# Програмско решење

# Резултати

# Закључак

# Литература

|  |  |
| --- | --- |
| [1] | R. Debouk, "Overview of the 2nd Edition of ISO 26262: Functional Safety – Road Vehicles," in *36th International System Safety Conference*, Phoenix, AZ , 2018. |
| [2] | B. Zhang, V. Appia, I. Pekkucuksen, A. U. Batur, P. Shastry, S. Liu, S. Sivasankaran, K. Chitnis and Y. Liu, "A surround view camera solution for embedded systems," *IEEE Conference on Computer Vision and Pattern Recognition Workshops,* 2014. |
| [3] | K. Lee and H. Peng, "Evaluation of automotive forward collision warning andcollision avoidance algorithms," *Vehicle System Dynamics,* vol. 43, no. 10, pp. 735-751, 2005. |
| [4] | S. Fürst and J. Mössinger, "AUTOSAR – A Worldwide Standard is on the Road.," [Online]. Available: https://www.win.tue.nl/~mvdbrand/courses/sse/0809/papers/AUTOSAR.pdf. [Accessed 12 Септембар 2019]. |
| [5] | "Развојна платформа Алфа," [Online]. Available: http://www.rt-rk.com/services/automotive. [Accessed 12 Септембар 2019]. |
| [6] | H. Fennel, S. Bunzel, H. Heinecke, J. Bielefeld, S. Fürst, K.-P. Schnelle, W. Grote, N. Maldener, T. Weber, F. Wohlgemuth, J. Ruh, L. Lundh, T. Sandén, P. Heitkämper, R. Rimkus, J. Leflour, A. Gilberg, U. Virnich, S. Voget, K. Nishikawa, K. Kajio, K. Lange, T. Scharnhorst and B. Kunkel, "Achievements and exploitation of the AUTOSAR development partnership," 2006. [Online]. Available: https://tinyurl.com/atsr-prtnrshp. [Accessed 16 Септембар 2019]. |
| [7] | H. Hellgren, "AUTOSAR in a Nutshell," 14 Април 2018. [Online]. Available: https://hackernoon.com/adaptive-autosar-in-a-nutshell-1cc609c1c5f5. [Accessed 16 Септембар 2019]. |
| [8] | M. Tische, "The Computing Center in the Vehicle - AUTOSAR Adaptive," Септембар 2018. [Online]. Available: https://assets.vector.com/cms/content/know-how/\_technical-articles/AUTOSAR/AUTOSAR\_Adaptive\_ElektronikAutomotive\_201809\_PressArticle\_EN.pdf. [Accessed 21 Септембар 2019]. |
| [9] | Austin Common Standards Revision Group, "POSIX® 1003.1 Frequently Asked Questions," 8 Јун 2017. [Online]. Available: http://www.opengroup.org/austin/papers/posix\_faq.html. [Accessed 21 Септембар 2019]. |
| [10] | AUTOSAR, "Explanation of Adaptive Platform Design," 27 Октобар 2017. [Online]. Available: https://www.autosar.org/fileadmin/user\_upload/standards/adaptive/17-10/AUTOSAR\_EXP\_PlatformDesign.pdf. [Accessed 21 Септембар 2019]. |
| [11] | IBM Knowledge Center, "Service Oriented Architecture," IBM, [Online]. Available: https://www.ibm.com/support/knowledgecenter/en/SSMQ79\_9.5.1/com.ibm.egl.pg.doc/topics/pegl\_serv\_overview.html. [Accessed 21 Септембар 2019]. |
| [12] | AUTOSAR, "AUTOSAR," Март 2019. [Online]. Available: https://www.autosar.org/standards/adaptive-platform/adaptive-platform-1903/. [Accessed 22 Септембар 2019]. |
| [13] | Compaq,Hewlett-Packard,Intel,Lucent,Microsoft,NEC,Philips, "Universal Serial BusSpecification," 27 Април 2000. [Online]. Available: http://sdphca.ucsd.edu/lab\_equip\_manuals/usb\_20.pdf. [Accessed 29 Септембар 2019]. |
| [14] | Texas Instruments, "Introductionto the Controller Area Network(CAN," Мај 2016. [Online]. Available: http://www.ti.com/lit/an/sloa101b/sloa101b.pdf. [Accessed 29 Септембар 2019]. |
| [15] | LIN Consortium, "LIN - Local Interconnect Network," 13 Јун 2015. [Online]. Available: http://www.interfacebus.com/Design\_Connector\_LIN\_Bus.html. [Accessed 29 Септембар 2019]. |
| [16] | L. Lamport, "Interporcess communication," 11 Јун 1985. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a156337.pdf. [Accessed 28 Септембар 2019]. |
| [17] | "D-bus," freedesktop.org, 7 Јул 2018. [Online]. Available: https://www.freedesktop.org/wiki/Software/dbus/. [Accessed 28 Септембар 2019]. |
| [18] | D. A. Rusling, The Linux Kernel, 2001. |
| [19] | Texas Instruments Incorporated, "TDA2x ADAS System-on-Chip," 2013. [Online]. Available: http://www.ti.com/lit/ml/sprt681/sprt681.pdf. [Accessed 29 Септембар 2019]. |