|  |  |
| --- | --- |
| |  | | --- | | **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: Филип Јашић**  **Број индекса: РА46/2014**  **Тема рада: TODO**  **Ментор рада: доц. др Богдан Павковић**  **Нови Сад, јун, 2019** | |

|  |  |  |  |
| --- | --- | --- | --- |
| 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**: | | **Филип Јашић** | |
| Ментор, **МН**: | | **доц. др Богдан Павковић** | |
| Наслов рада, **НР**: | | TODO | |
| Језик публикације, **ЈП**: | | Српски / ћирилица | |
| Језик извода, **ЈИ**: | | Српски | |
| Земља публиковања, **ЗП**: | | Република Србија | |
| Уже географско подручје, **УГП**: | | Војводина | |
| Година, **ГО**: | | **2019.** | |
| 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) | | **7/48/14/0/12/0/0** | |
| Научна област, **НО**: | | Електротехника и рачунарство | |
| Научна дисциплина, **НД**: | | Рачунарска техника | |
| Предметна одредница/Кључне речи, **ПО**: | |  | |
| **УДК** | |  | |
| Чува се, **ЧУ**: | | У библиотеци Факултета техничких наука, Нови Сад | |
| Важна напомена, **ВН**: | |  | |
| Извод, **ИЗ**: | | TODO | |
| Датум прихватања теме, **ДП**: | |  | |
| Датум одбране, **ДО**: | |  | |
| Чланови комисије, **КО**: | Председник: | проф. др |  |
|  | Члан: | доц. др | Потпис ментора |
|  | Члан, ментор: | доц. др |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 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**: | | **Filip Jašić** | |
| Mentor, **MN**: | | **Bogdan Pavkovic, PhD** | |
| Title, **TI**: | | TODO | |
| 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**: | | **2019.** | |
| Publisher, **PB**: | | Author’s reprint | |
| Publication place, **PP**: | | Novi Sad, Dositeja Obradovica sq. 6 | |
| Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes) | | **7/48/14/0/12/0/0** | |
| 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**: | | TODO | |
| Accepted by the Scientific Board on, **ASB**: | |  | |
| Defended on, **DE**: | |  | |
| Defended Board, **DB**: | President: | , PhD |  |
|  | Member: | , PhD | Menthor's sign |
|  | Member, Mentor: | ,PhD |  |

**Захвалност**

**Садржај**

[1. Увод 1](#_Toc10732430)

[2. Теоријске основе 2](#_Toc10732431)

[2.1 CAN протокол 2](#_Toc10732432)

[2.1.1 Физички слој CAN протокола 4](#_Toc10732433)

[2.1.2 Електрична својства, каблови и утичнице 6](#_Toc10732434)

[2.1.3 Порука, формат оквира и пренос података 7](#_Toc10732435)

[2.1.4 Арбитража 11](#_Toc10732436)

[2.1.5 Убацивање бита, препознавање грешака, сигурност, формат без повратка на нулу 12](#_Toc10732437)

[2.2 LIN протокол 14](#_Toc10732438)

[2.3 SPI протокол 14](#_Toc10732439)

[2.4 I²C протокол 14](#_Toc10732440)

[2.5 УПОРЕДНО ПОРЕЂЕЊЕ ПРОТОКОЛА 14](#_Toc10732441)

[2.6 ОСЦИЛОСКОП 14](#_Toc10732442)

[3. Концепт решења 16](#_Toc10732443)

[*3.1* *ПОВЕЗИВАЊЕ CAN-A И МЕРЕЊЕ СА ОСЦИЛОСКОПОМ,НАЧИН ДЕКОДОВАЊА* 16](#_Toc10732444)

[*3.2* *-||- LIN -||-* 16](#_Toc10732445)

[*3.3* *-||- SPI -||-* 16](#_Toc10732446)

[*3.4* *-||- I2C -||-* 16](#_Toc10732447)

[4. Програмско решење 17](#_Toc10732448)

[*4.1* *МОЈЕ РЕШЕЊЕ ЗА МЕРЕЊЕ,ТРИГЕРИ,МАЛО КОДА КОЈИ ОБЈАШЊАВА ПРИКУПЉАЊЕ ПОДАТАКА И ОБРАДУ* 17](#_Toc10732449)

[4.2 CANoe 17](#_Toc10732450)

[*4.3* *ДЕКОДОВАЊЕ CAN-a са кодом* 17](#_Toc10732451)

[*4.4* *-|| - LIN -||-* 17](#_Toc10732452)

[*4.5* *-||- SPI -||-* 17](#_Toc10732453)

[*4.6* *-||- I2C - ||-* 17](#_Toc10732454)

[5. Тестирање и резултати 18](#_Toc10732455)

[6. Закључак 19](#_Toc10732456)

[7. Литература 20](#_Toc10732457)

**Списак слика**

Слика 2.1 Брза CAN мрежа,ISO 11898-2 4

Слика 2.2 Физичка репрезентација бита у брзој CAN мрежи,ISO 11898-2 4

Слика 2.3 Омеђавање линије за брзу CAN линију 5

Слика 2.4 Омеђавање линије за спору CAN линију 5

**Скраћенице**

CAN – **C**ontroller **A**rea **N**etwork

LIN – **L**ocal **I**nterconnect **N**etwork

SPI – **S**erial **P**eripheral **I**nterface

I²C – **I**nter-**I**ntegrated **C**ircuit

ISO - **I**nternational **O**rganization for **S**tandardization

Mb/s – **m**ega**b**yte per **s**econd

kb/s – **k**ilobytes **p**er **s**econd

CRC - **C**yclic **r**edundancy **c**heck

ECU - **E**lectronic **C**ontrol **U**nit

RTR – **R**emote **T**ransmission **R**equest

MSB **- M**ost **S**ignificant **B**it

SRR – **S**ubstitute **r**emote **r**equest

IDE - **I**dentifier **e**xtension **b**it

UART – **U**niversal **a**synchronous **r**eceiver-**t**ransmitter

# Увод

*О АУТОМОБИЛСКОЈ ИНДУСТРИЈИ И ПОТРЕБИ ЗА ПРОТОКОЛИМА*

# Теоријске основе

## CAN протокол

CAN протокол је најпознатији аутомобилски протокол. Развијен од стране Robert Bosch GmbH 1983. године, док је званично пуштен у употребу од 1986. године на Society of Automotive Engineers (SAE) конферецији у Детроиту. Bosch је објавио више верзија CAN спецификације, а најновија је CAN 2.0, објављена 1991. и на тој верзији спецификације се базира декодовање CAN сигнала урађено у овом дипломском раду. Ова верзија спецификације се састоји из 2 дела:

* А (покривен ИСО 11519 стандардом) – овај део спецификације описује стандардни формат са једанестобитним идентификатором. CAN уређај који користи овакав идентификатор се често назива CAN 2.0 А и тако ћемо га звати надаље.
* Б (покривен ИСО 11898 стандардом) – овај део спецификације описује проширени формат са дватесетдеветобитним идентификатором. CAN уређај који користи овај продужени идентификатор се често назива CAN 2.0 В и тако ћемо га звати надаље.

ISO је 1993. објавио стандард за CAN под називом ISO 11898 који се касније реструктурирао у 2 дела:

* ISO 11898-1 који покрива Data Link Layer (део са подацима)
* ISO 11898-2 који покрива Physical Layer (физички део) – овај слој се користи за брзи CAN протокол (брзине од 125Kb/s до 1Mb/s).Овај протокол се углавном користи у аутомобилској индустрији и индустријама где магистрала повезује више окружења.
* ИСО 11898-3 је објављен 2006. и покрива Physical Layer за спорији CAN (брзине мање од 125Kb/s), али који је отпоран на грешке. Овај протокол се се често користи где групе електониских контролних јединица треба да се повежу заједно.

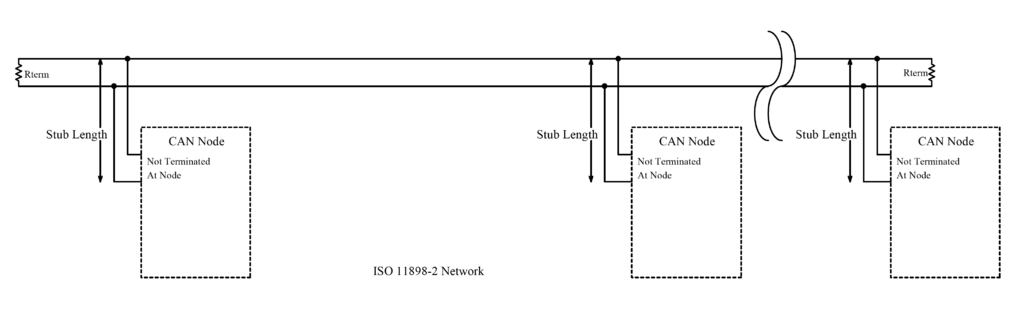
Стандарди ИСО 11898-2 и ИСО 11898-3 нису део Bosch-ове CAN 2.0 спецификације.

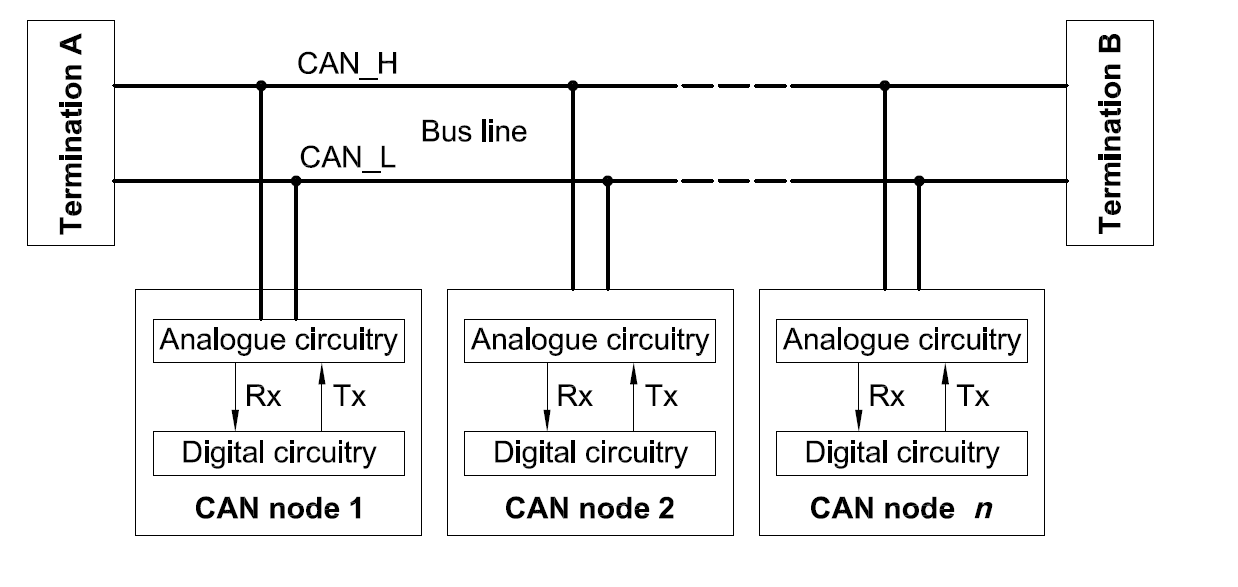
ИСО/ОСИ 1. Физички слој (енгл. *Physical Layer*) и 2. Слој са подацима (енгл. *Data Link Layer*) слој припадају стандардној CAN имплементацији(урађено у хардверу ,што омогућава драматичну предност за процесе у реалном времену).CAN може да се делимично имплеметира са вишим слојевима ИСО/ОСИ модела (од 3. слоја до 6.).

Неке од карактеристика CAN продокола су:

* Брза,серијска,заједничка магистрала(брзине до 1Mbs) за више уређаја(до 32 уређаја по ИСО 11898; либералан приступ – било ко приступа када је медијум слободан), коришћење упредене парице(домет до 40m)
* Асинхрона комуникација(окидање на догађаје).
* Величина пакета: 0-8 октета.
* Филтрирање на пријему,препознавање грешке (коришшењем ЦИКЛИЧНА ПРОВЕРА РЕДУНДАНСЕ), мере за ограничавање грешака(гашење физички оштећених).
* Одзив у реалном времену.
* Недеструктивна арбитража(100% искоришћеност пропусне моћи)
* Приоритет на основу идентификатора:
  + Већа кашњења за поруке ниског приоритета.
  + Мала кашњења за поруке високог приоритета.

### Физички слој CAN протокола





Слика 2.1 Брза CAN мрежа,ISO 11898-2

Ц

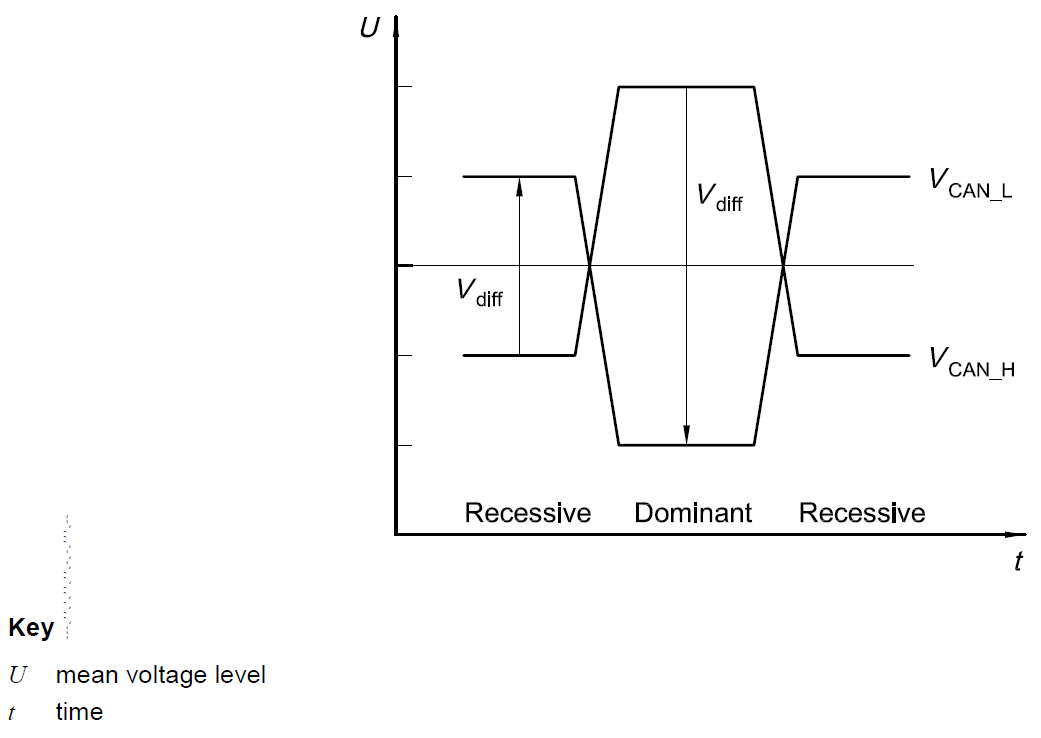
C

CAN је мрежа са могућношћу да има више власника(master) за повезивање са електронском управљачком јединицом. Све електронске управљачке јединице се међусобно повезују преко магистрале са 2 жице(једна се зове CAN-H, друга CAN-L). Ове 2 жице су упрадена парица(оклопљена или неоклопљена) и потребно је омеђити их отпорником са номиналном карактеристичном импеданцом од 120 Ω у случају брзе CAN магистрале, док у случају споре CAN магистрале се користи отпорник од бар 100 Ω (отпорност се индивидуално израчунава).

Логичка стања се процењују на основу разлике напона. Напонски нивои зависе од конкретног сучеља:

* За брзу CAN магистралу(ISO 11898-2):
  + Логичка 1 представља разлику од 0V(у теорији) између CAN-H и CAN-L жице, у пракси се то креће од 0.5V. Напон на CAN-H жици иде ка 5V док на CAN-L жици иде ка 0V. Логичка 1 представља рецесивни бит (високо стање).
  + Логичка 0 представља разлику од 2V (мора биту у распону између 1.5V и 3.5V, разлика напона иде чак и преко 0.9V уколико је напајање +/-12V) између CAN-H и CAN-L жице. Напон CAN-H жице иде ка 5V док напон CAN-L жице иде ка 0V. Логичка 0 представља доминантни бит(ниско стање). Представљање доминатног бита нулом се ради да би електронске контрлне јединице са најнижом идентификацијоним бројем имали највећи приоритет.
* За спору CAN магистралу(ISO 11898-3):
  + Логичка 1 представља разлику од бар 0.6V између CAN-H и CAN-L жице.
  + Логичка 0 представља разлику од бар 2.3V између CAN-H и CAN-L жице. Напон CAN-H жице иде ка 5V, док напон жице CAN-L иде при преносу логичке 0. Она представља доминантни бит.

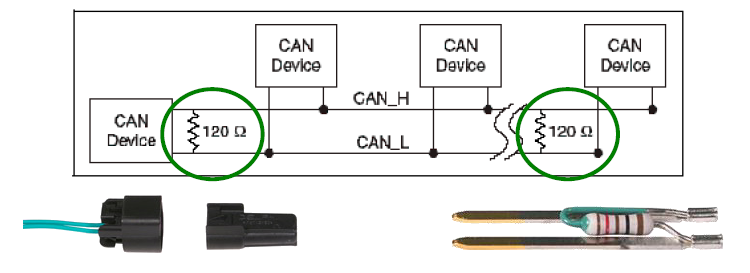
Разлике у ограђивању брзе и споре CAN магистрале су следеће:



Слика 2.2 Физичка репрезентација бита у брзој CAN мрежи,ISO 11898-2

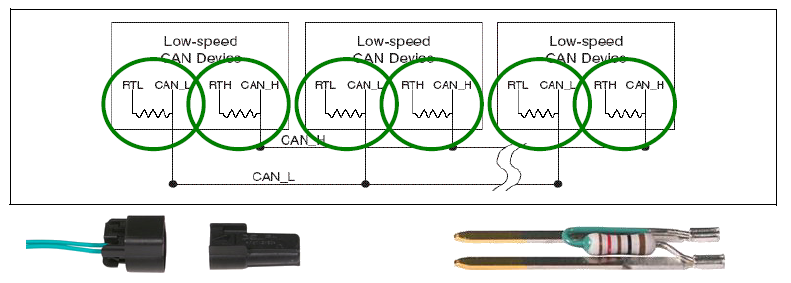
Слика 2.2 Физичка репрезентација бита у брзој CAN мрежи,ISO 11898-2

* За брзу CAN магистралу је потребно да се оба краја две линије (CAN-H и CAN-L) ограде, док је средина слободна.Начин повезивања за овај тип магистале је илустрован на слици 2.3.
* За спору CAN магистралу нема потребе за ограђивањем крајева линије. Омеђавају се уређаји и свака линија посебно: RTH-CAN-H и RTH-CAN-L. Начин повезивања за овај тип магистрале је илустрован на слици 2.4.

****

Слика 2.3 Омеђавање линија за брзу CAN линију

Слика 2.3 Омеђавање линије за брзу CAN линију

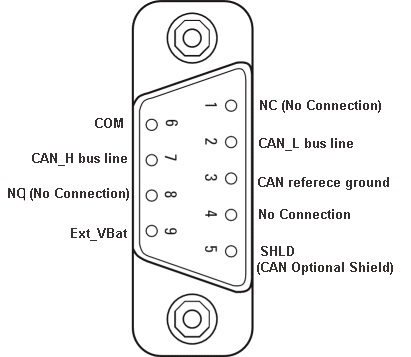


Слика 2.4 Омеђавање линија за спору CAN линију

### Електрична својства, каблови и утичнице

Брзина транзиције је већа када се деси промена из рецесивног у доминантни бит, без обзира да ли је у питању брза или спора CAN мрежа. Брзина транзиције из доминантног у рецисвни бит зависи од дужине саме CAN мреже и капацитивности коришћене жице. Спецификације захтевају да магистрала буде унутар минималног и максималног заједничког напона, али не дефинишу саме вредности тих граница.

Најчешће се за каблове користе оклољене упредене парице. Користи се један кабел,осим у случају када постоји потреба за додатним напајањем. Сам CAN не одређује врсту физичког медијума међутим, типично се користе деветопинске D-sub и петопинске mini style утичице. Пример обележене утичнице која се најчешће користи у аутомобилској индустрији са CAN протоколом (DE-9 тип D-sub утичнице) се може видети на слици 2.5.



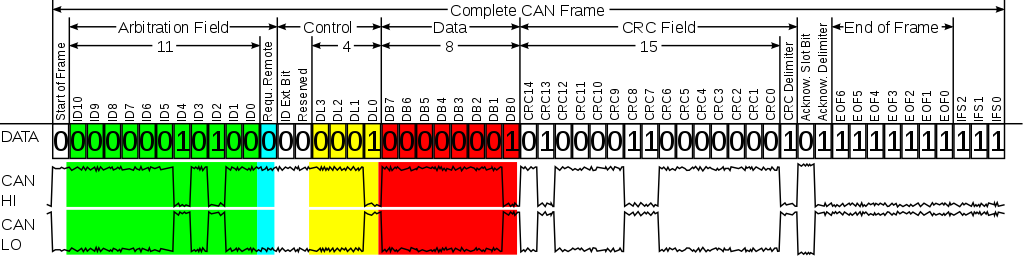
Слика 2.5 Утичница DE-9

### Порука, формат оквира и пренос података

Уређаји повезани на CAN мрежу комуницирају слањем порука. Пренос порука је контролисан са 4 типа оквира:

* Оквир за податке (Data frame) носи податке од предајника до пријемника.
* Оквир за даљинско управљање (Remote frame) се шаље на магистралу као захтев за оквиром за податке са одређеним идентификатором.
* Оквир за грешке (Error frame) се шаље од било ког чвора уређаја који детектује грешку на магистрали.
* Оквир за пренатрпаност (Overload frame) се користи да омогући кашњење између 2 послата оквира за податке или оквира за грешке.

Илустративни пример CAN 2.0 A без уметања бита се може видети на слици 2.5.



Слика 2.5 Илустративни пример CAN 2.0 A без уметања бита и његова корелација са напоном на CAN-H и CAN-L

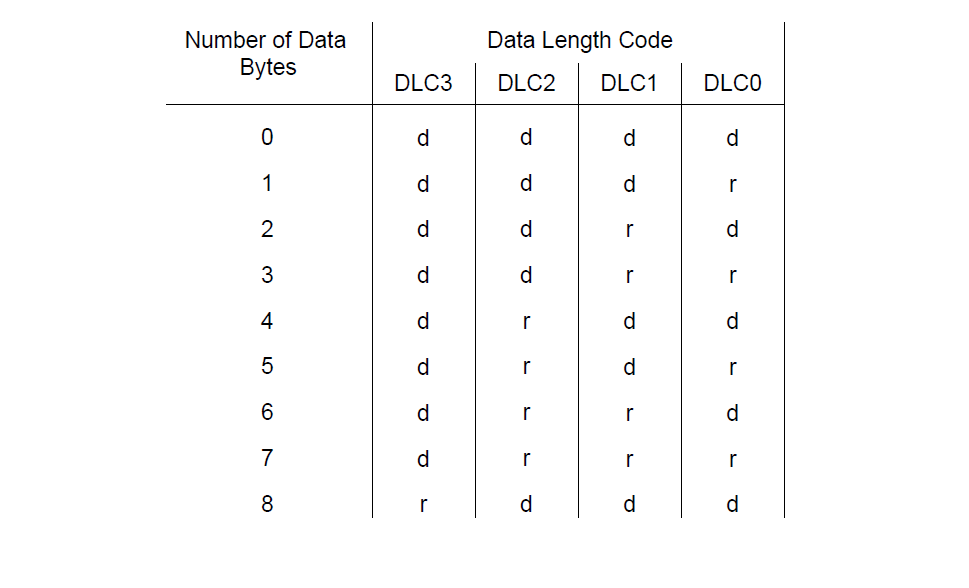
Оквир за податке се састоји од 7 поља(разлике између CAN 2.0 A и CAN 2.0 B ћемо посебно навести где је потребно, иначе ћемо подразумевати одређена поља имају исту структуру):

* Почетак оквира (Start of frame) означава почетак поља за оквир за податке или оквира за даљинско управљање. Састоји се од једног доминантног бита (логичка 0). Уређај на магистрали може да почне да шаље када је магистрала у мирном режиму рада (Bus Idle). Сви уређаји морају да се синхронизују са водећом ивицом која је проузрокована од стране овог поља од уређаја који је први започео пренос.
* Поље за арбитражу (Arbitration field). За ово поље постоје разлике између CAN 2.0 A и CAN 2.0 B:
  + Ово поље се састоји од идентификатора и RTR бита у случају CAN 2.0 A при чему идентификатор се састоји од 11 бита, ови бити се гледају у MSB редоследу при чему свих 7 (дакле од ID-10 до ID-4) најзначајних бита не смеју бити рецесивни. RTR бит мора да буде доминантан када се шаље оквир за податке, иначе када се шаље оквир за даљинско управљање онда је тај бит рецесиван.
  + У случају CAN 2.0 B идентификатор се састоји од 29 бита, подељен на 2 дела: идентификатор А (11 бита) и идентификатор Б(18 бита). Између ова 2 дела продуженог идентификатора се налази SRR (1 рецесиван бит) и IDE (1 рецесиван бит).
* Контролно поље (Control field) . За ово поље такође постоје разилике између 2 подврсте протокола:
  + Код бржег типа протокола, контролно поље се састоји од 6 бита. Оно укључује 4 бита који означавају дужину послатих података (енгл. Data length code) у бајтима и 2 бита који су резервисани за будућа проширења протокола, IDE (мора бити доминантан) и r0 (може бити или доминантан или рецесиван).
  + Код споријег типа протокола ово поље се такође састоји од 6 бита, при чему прво стижу 2 резервисана бита – r0, r1 (могу бити или доминанти или рецесивни). Након њих у оквиру се налазе 14 бита који означавају дужину послатих података и фунцкионишу на исти начин као у бржем протоколу.

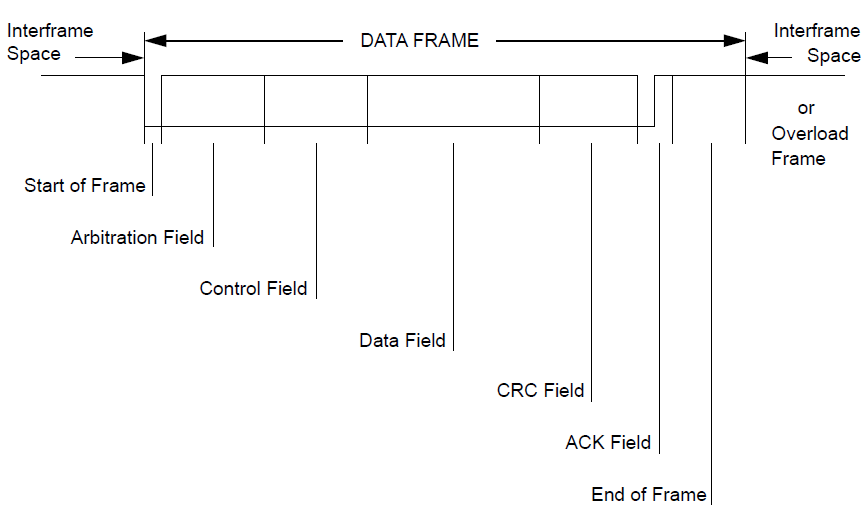
Илустративно објашњење како фунционише део котнролног поља које означава дужину података је приказано на слици 2.6 при чему су са d обележени доминантни бити док су са r обележени рецесивни бити.

* Поље за податке (Data field) се састоји од 0 до 8 бајта, при чему 1 бајт има 8 бита. Подаци се шаљу у MSB редоследу.
* Поље за цикличну проверу редудансе (CRC field) се састоји од 15 бита који заправо представљају израчунату цикличну проверу редудансе и делимитера који заузима 1 бит, при чему је тај бит рецесиван.
* Поље за потврду о пријему (ACK field) се састоји од ACK slot, ACK делимитера, сваки заузима по 1 бит, при чему пошиљаоц шаље рецесивни бит у ACK slot,а примаоц поруке може да нареди слање доминантног бита. ACK делимитер „раздваја“ ово поље са крајем оквира и оно садржи 1 рецесиван бит.
* Крај оквира (End of frame) се састоји од 7 рецесивних бита.

Илустративни приказ оквира за податке може да се види на слици 2.7.



Слика 2.6 Илустративно објашњење како фунционише DLC



Слика 2.7 Илустративни приказ оквира за подаке

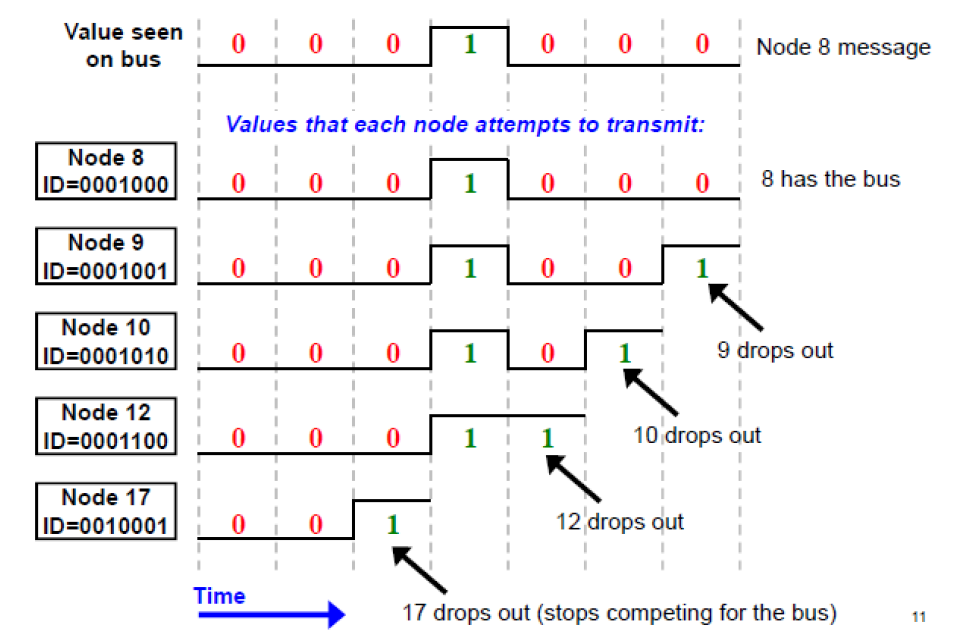
Разлика између CAN 2.0 A и CAN 2.0 B се може боље упоредити у следећој табели.

|  |  |  |  |
| --- | --- | --- | --- |
| CAN 2.0 А (једанаестобитни идентификатор) | | CAN 2.0 B (двадесетдеветобитни идентификатор) | |
| Назив поља | Дужина поља | Назив поља | Дужина поља |
| Почетак оквира | 1 бит | Почетак оквира | 1 бит |
| Идентификатор | 11 bita | Идентификатор А | 11 бита |
| RTR | 1 бит | SRR | 1 бит |
| IDE | 1 бита | IDE | 1 бит |
| Резервисан бит | 1 бита | Идентификатор Б | 18 бита |
| DLC | 4 бита | RTR | 1 бит |
| Подаци | 0-64 бита | Резервисани бити | 2 бита |
| CRC | 15 бита | DLC | 4 бита |
| CRC делимитер | 1 бит | Подаци | 0-64 бита |
| ACK | 1 бит | CRC | 15 бита |
| ACK делимитер | 1 бит | CRC делимитер | 1 бит |
| EOF | 1 бит | ACK | 1 бит |
|  |  | ACK делимитер | 1 бит |
|  |  | EOF | 1 бит |

* Поље за почетак оквира
* Поље за арбитражу
* Контролно поље
* Поље за податке
* Поље за цикличну проверу редудансе
* Поље за потврду о пријему
* Крај оквира

### Арбитража

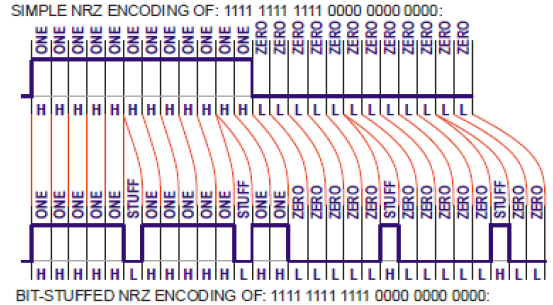
Када год је магистрала слободна, било која електронска управљачка јединица може да почне са слањем поруке. У случају да 2 или више ових јединица почне са слањем порука у исто време, настаје конфликт који се решава механизмом арбитраже за појединачне бите коришћењем идентификатора. Овај механизам обезбеђује да се не губи ни време ни подаци. Ако оквир за пренос података и оквир за даљинско управљање имају исти идентификатор и ако су иницијализовани у исто време, предност има оквир за слање података. Током арбитраже, сваки пошиљаоц пореди на ком нивоу се налази бит идентификатора који је он послао, наспрам нивоа бита идентификатора који се налази на магистрали. Ако су ови бити на истим нивоима једнаки, пошиљаоц може да настави са слањем. Када је рецесивни бит на истом нивоу послат и доминантни бит на истом нивоу се налази на магистрали, тада је пошиљаоц „изгубио“ арбитражу и мора да се повуче, при чему се прекида пренос преосталих бита у оквиру. Односно,побеђују уређаји са најмањим идентификатором. Идентификатор се шаље у MSB формату, да би се брже уочила арбитража и да би медијум био ефикаснији за слање. Илустративни пример арбитраже се може видети на слици 2.8.



Слика 2.8 Пример арбитраже

### Убацивање бита, препознавање грешака, сигурност, формат без повратка на нулу

Убацивање бита се врши на следећи начин: након 5 идентичних бита се убацује супротан бит, ово побољшава усклађивање и додати бити не мењају податке пошто је убацивање хардверски урађено. Оно генерално није потребно, пошто врло ретко иду истоветне вредности. Проблем са убацивањем бита је зато што инверотовани бити се тешко могу препознати, али и зато што могу да се јаве такозвани „лавински“ бити(нпр. 5 рецисних бита које прати 5 доминантих бита) који могу да збуне механизам за декодовање убачених бита. Понекад CRC може да помогне али уз ограничења. Једноставан пример убацивања бита је приказан на слици 2.8.



Слика 2.9 Пример убацивања бита

Статистичка учестаност грешака CAN протокола зависи од укупног броја уређаја, физичког ожичења и распореда, спољашњих електромагнетних сметњи. Учестаност непрепознатих грешака је 1 у 1000 година, и то је један од разлога зашто је ово најкоришћенији протокол у аутомобилској индустрији која захтева велику и дуготрајну поузданост.

Ток препознавања грешака између пошиљаоца (Tx) и примаоца (Rx) је независан од филтрирања и маскирања на пријему и се састоји из 5 механизама:

1. Надгледање појединачних бита (Tx) - препознавање локалних и глобалних грешака код пошиљаоца, упоређује се стање на магистрали са послатим битом,

не примењује се на поље за арбитражу .

1. Провера структуре – типична поља: CRC delimiter, ACK delimiter и EOF–увек рецесивни бити.
2. Провера кодовања (уметање бита) - пријемник проверава ток бита, на сваких 5 узастопних истоветних бита мора да дође убачена промена,

укључује све бите од SOF do краја CRC.

1. Провера потврде (Tx) је обавеза пошиљаоца -

потребно бар да пристигне једна потврда, пошиљалац подеси рецесивни бит, потребно да пријем обори са доминантним.

1. Циклична провера редудансе - пошиљалац рачуна пре слања, пријемник проверава подударање при пријему.

Циклична провера редудансе се рачуна на следећи начин:

полином, чији су коефицијенти представљени комбинацијом претходних поља(при чему је урађено избацивање бита) и за 15 најнижих коефицијената, са 0. Овај полином се потом дели (коефицијенти су израчунати модулом двојке) са генератором-полиномом :

x15 + x14 + x10 + x8 + x7 + x4 + x3 + 1

Oстатак овог дељења представља петнаестобитну вредност CRC секвенце која се шаље на магистралу. Да би се применила ова функција, 15 – битни померачки регистар CRC\_RG (14:0) се може користити. Псеудо-код ове функције се описује са:

CRC\_RG = 0; //initialize shift register

REPEAT

CRCNXT = NXTBIT EXOR CRC\_RG(14); //NXTBIT denotes the next bit of the bit stream

CRC\_RG(14:1) = CRC\_RG(13:0); //shift left by

CRC\_RG(0) = 0; //1 position

IF CRCNXT THEN

CRC\_RG(14:0) = CRC\_RG(14:0) EXOR (4599hex);

ENDIF

UNTIL (CRC SEQUENCE starts or there is an ERROR condition)

Након слања/пријема последњег бита од поља за податке, CRC\_RG садржи CRC секвенцу.

Сам протокол не подржава енкрипцију у стандардној имплементацији, у већини примена, од уређаја који користе CAN се очекује да имају своје сигурносне механизме. Ако ово није урађено, урађаји су подобни раличитим врстама малициозних напада, нпр. ако неко пошаље своје поруке на магуистралу. Постоје шифровани системи за одређене фунцкионалности које су од критичког значаја за сигурност, као што су модификовање системског софтвера, програмерски кључеви или контролисање кочница. Међутим, ови системи нису универзално имплементирани.

## LIN протокол

LIN је серијски мрежни протокол, развијен од стране LIN конзорцијума (5 произвођача аутомобила - BMW, Volkswagen Group, Audi Group, Volvo Cars, Mercedes-Benz; 1 добављач полупроводиника – Motorola; 1 добављач алата - VCT) основаног 1998 ., док је прва, у потпуности имплементирана верзија LIN спецификације (LIN 1.3) , објављена 2002. године. Иницијално је развијена као CAN подмрежа са циљем да се смањи оптерећење, данас има широку употребу у раличитим индустријама: у аутомобилима, медицинској опреми, белој техници...

Најважније одлике LIN прокола су:

* Брзина до 20kb/s на дужинама до 40 метара.
* Руководилац контролише медијум, због тога постоји гаранција кашњења и нема судара. Усклађивање такта је одговорност зависних уређаја(због тога нема потребе за кварцним или керамичким резонаторима).
* Варијабилност у дужини послатих података(2,4 или 8 бајта, при чему је 1 бајт 8 бита).
* Флексибилност у конфигурацији и динамичка топологија.
* Величина мреже је углавном до 12 чворова (могуће је и до 64, али уз низак проток).
* Детекција дефектних чворова, детекција грешке и контролни збир података.
* Оперативни напон од 12V.
* Ниска цена, силиконска имплементација заснована на UART стандарду.

## SPI протокол

## I²C протокол

## УПОРЕДНО ПОРЕЂЕЊЕ ПРОТОКОЛА

## ОСЦИЛОСКОП

# Концепт решења

## *ПОВЕЗИВАЊЕ CAN-A И МЕРЕЊЕ СА ОСЦИЛОСКОПОМ,НАЧИН ДЕКОДОВАЊА*

## *-||- LIN -||-*

## *-||- SPI -||-*

## *-||- I2C -||-*

# Програмско решење

## *МОЈЕ РЕШЕЊЕ ЗА МЕРЕЊЕ,ТРИГЕРИ,МАЛО КОДА КОЈИ ОБЈАШЊАВА ПРИКУПЉАЊЕ ПОДАТАКА И ОБРАДУ*

## CANoe

## *ДЕКОДОВАЊЕ CAN-a са кодом*

## *-|| - LIN -||-*

## *-||- SPI -||-*

## *-||- I2C - ||-*

# Тестирање и резултати

*РЕЗУЛТАТИ КОЈЕ САМ ДОБИО И ПОТЕШКОЋЕ НА КОЈЕ САМ НАИШАО ПРИЛИКОМ ПРИКУПЉАЊА ПОДАТАКА*

# Закључак

*О ТОМЕ ШТА ЈЕ УРАЂЕНО И КАКО БИ ЈОШ МОГЛО ДА СЕ ПОБОЉША*

# Литература

[1] Слика 2.1 Брза CAN мрежа, ISO 11898-2.[Online]. Доступно:<https://upload.wikimedia.org/wikipedia/commons/thumb/b/bc/CAN_ISO11898-2_Network.png/1024px-CAN_ISO11898-2_Network.png> [приступљено: јун 2019.]

<https://neweagle.net/support/wiki/images/9/9c/DB9-CAN-pinout.jpg>