Рачунарска техника и рачунарске комуникације  
Аутомобилске мреже

Вежба 5 – *CANdb++ Editor, CAN* протокол у склопу *CANoe* и *ТТX Connexion-Configurator* aлата, *CAN-2-CAN* рутирање сa *TTX Connexion* уређајем

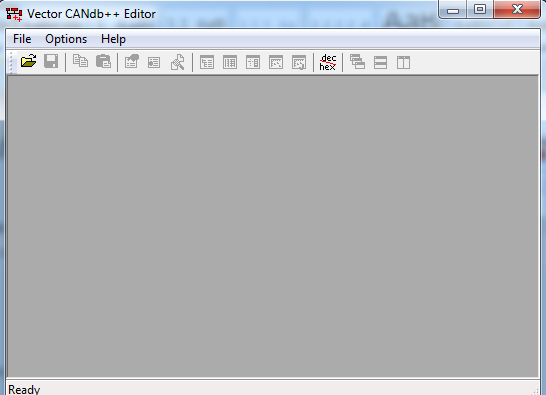
**Циљеви вежбе:**

* Упознавање са *CANdb++ Editor* алатом
* Употреба *CAN* протокола у *CANoe* и *TTX Connexion-Configurator* алатима
* Рутирање *CAN* порукама и сигналима са *TTX Connexion* уређајем

# *CANdb++ Editor* aлат

## Увод

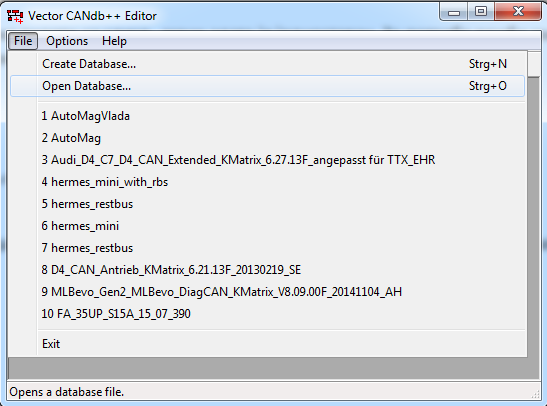
Целокупна комуникација у *CAN* мрежи једног аутомобила описана је у специјалнoј датотеци која je препознатљива по томе што се завршава са екстензијом *.dbc*(енг. *Data Base for CAN*)*.* Унутар *.dbc* датотеке налазе се својства *CAN* мреже, *ЕCUs(*енг*. еlectronic control unit)* повезани на *CAN* магистралу као и *CAN* поруке и сигнали. Алат који нам омогућава да учитавамо већ постојеће *.dbc* датотеке, модификацију истих као и креацију нових *.dbc* датотека, назива се *CANdb++ Editor.* На слици испод може се видети изглед прозора након покретања алата.



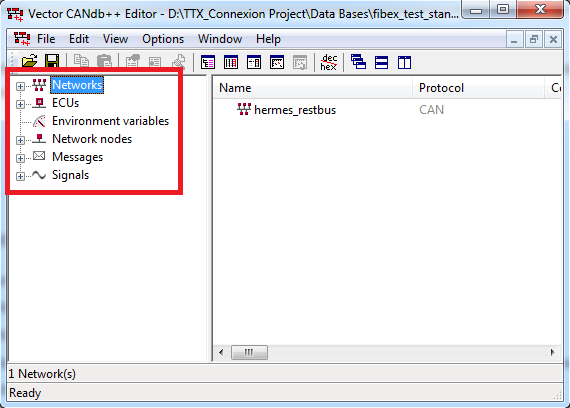
Као што се може видети са слике, мени алата је једноставан. За потребе вежби користи ћемо читање већ постојеће *.dbc* датотеке као и креацију сопствене.

## Учитавање *.dbc* датотеке

Учитавање *.dbc* датотеке реализује се бирањем ***File*** опције из основног менија, па затим избором ***Open Database…***опције.



Након учитавања датотеке добија се увид у целокупну *CAN* мрежу са свим релевантним параметрима који је описују. На слици испод, параметри *CAN* мреже се налазе унутар црвеног правоугаоника.



Пареметри који описују једну *CAN* мрежу су:

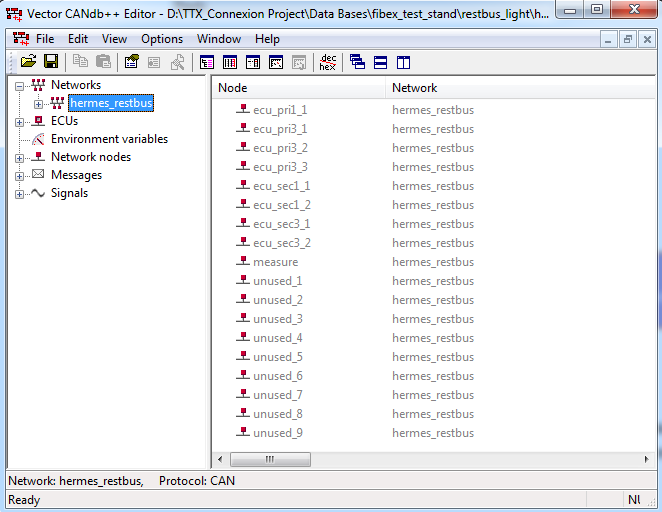
* Мреже
* Контролне јединице
* Променљиве окружења
* Мрежни чворови
* Поруке
* Сигнали

**Напомене:**

* Као показна, већ креирана *.dbc* датотекa, користиће се „*hermes\_restbus.dbc*“

### *CAN* мреже

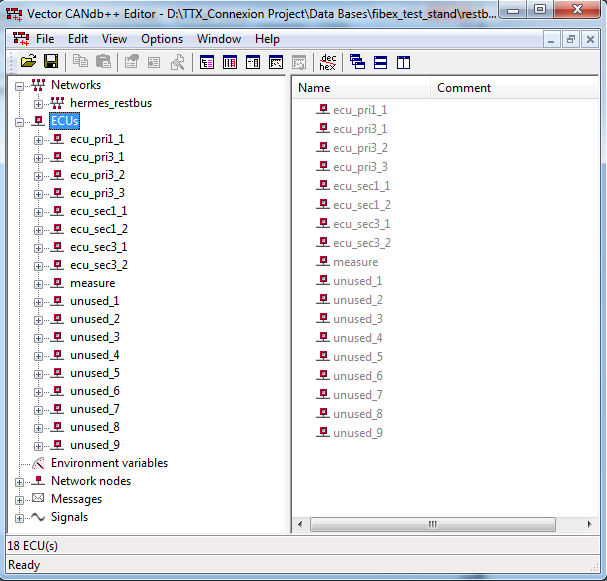
Свака *CAN* мрежа може бити дефинисана са више мањих под мрежа, такозваних група (енг. *cluster*). Све групе у *CAN* мрежи налазе се у под секцији ***Networks***. У примеру *.dbc* датотеке коју ћемо користити, као и већина *.dbc* датотека, целокупна *CAN* мрежа састоји се од једне групе, “*hermes\_restbus*” као што је приказано на слици испод.



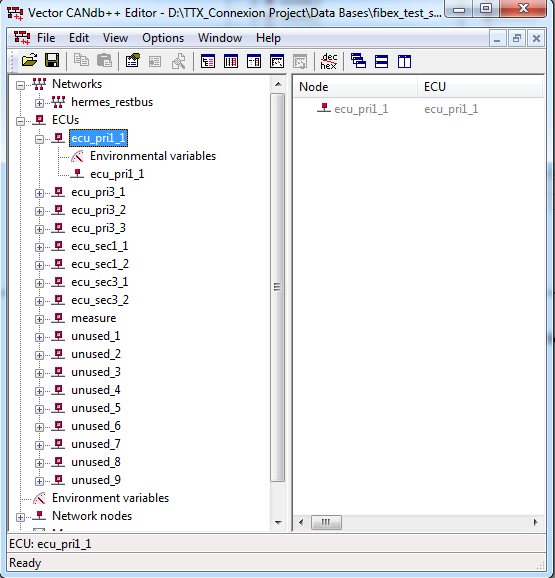
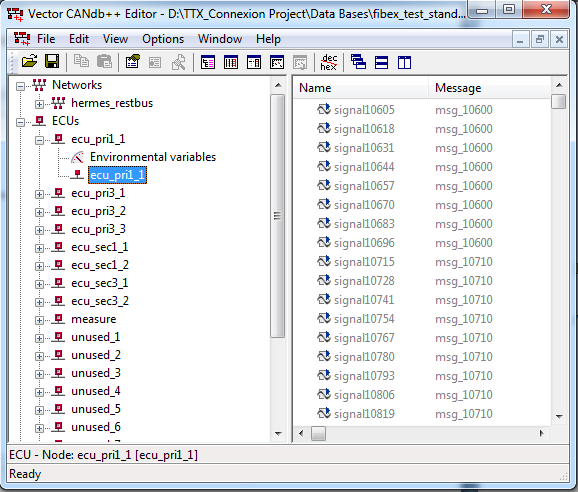
Кликом на назив групе, у десном делу прозора постају видљиви сви мрежни чворови који се налазе унутар изабране групе.

### *CAN контролне јединице*

*CAN* контролне јединице представљају дистрибуиране процесорке јединице унутар *CAN* мреже.Подаци које контрална јединица размењује са *CAN* мрежом, шаљу се и примају преко мрежног чвора на који је повезана контролна јединица. У пракси је углавном на један мрежни чвор повезана једна контролна јединица.



Након клика на ***ECUs****,* све контолне јединице постају видљиве, како у прозору са десне стране тако и као под секције испод ***ECUs****.*

Селектовањем било које контолне јединице постају видљиве променљиве окружења као и мрежни чворови на које је повезана изабрана контролна јединица. Као што је раније напоменуто, у већини случајева је једна конторлна једица повезана на један мрежни чвор.

Даљим избором мрежног чвора на који је повезана контролна јединица, у десном делу прозора постају видљиви сви сигнали као и поруке којима припадају сигнали који се размењују преко изабране контролне јединице преко одговарајућег мрежног чвора.

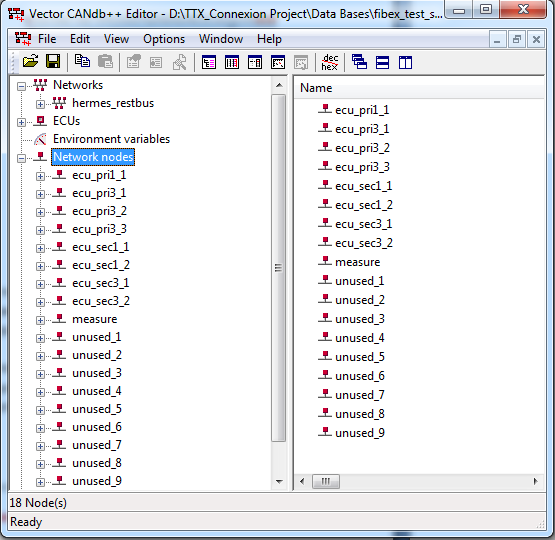
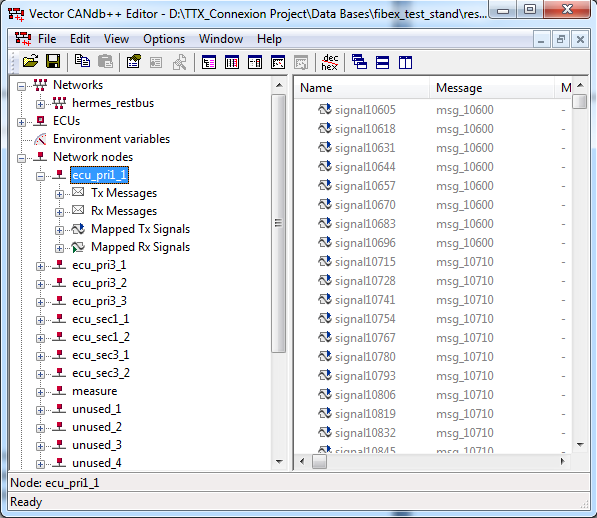
### *CAN* променљиве окружења

Променљиве окружења јесу променљиве које су придружене конторлној јединици.

### *CAN* мрежни чворови

*CAN* мрежни чворови представљају спрегу за комуникацију контролних јединица са *CAN* мрежом.Све поруке и сигнале које контролна јединица шаље на мрежу и прима са мреже, обрађују се преко мрежних чворова.

Као и код конторлних јединица, списак свих мрежних чворова доступан је након селектовања ***Network nodes****.* Приметите да су називи мрежних чворова као и конролних јединца исти, што представља и праксу приликом креирања *.dbc* датотеке.

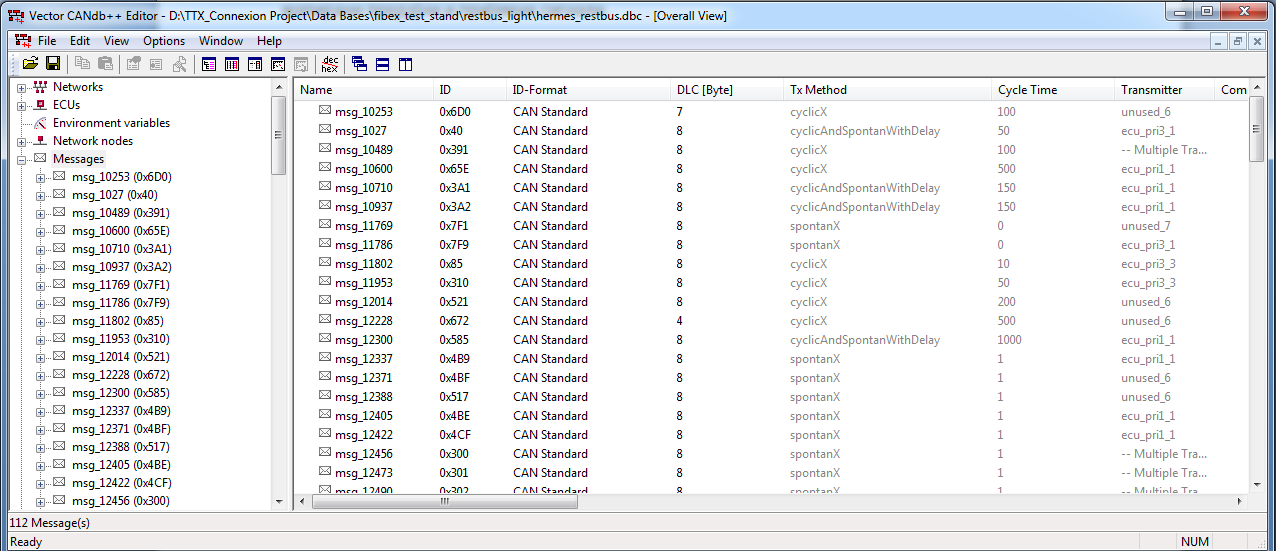
 

За сваки изабрани мрежног чвор могу се видети предајне поруке, пријемне поруке, мапирани предајни и пријемни сигнали.

### *CAN* поруке

Свака *CAN* порука у *.dbc* датотеци дефинисана је следећим параметрима:

* Симболичним именом поруке
* Јединственим идентификационим бројем(oзначено са ***ID*** на слици испод) , стандардним или проширеним (***ID-Format***)
* Бројем бајтова у сегменту за податке ***DLC***(енг. *Data Length Code*)
* Типом предаје(опционо)
* Временом циклуса(опционо)
* Сигналима који су дефинисани унутар поруке



Слика изнад приказује преглед основних параметара *CAN* порука.

#### Симболично име поруке

Симболично име поруке остављено је креатору на избор.

#### Јединствени идентификациони број

Свака *CAN* порука је у мрежи одређена јединственим идентификационим бројем. Идентификациони број може бити у основном формату (означено са ***CAN Standard***у колони ***ID-Format*)** или у проширеном формату (означено са ***CAN Extended***у колони ***ID-Format***)

#### Број бајтова у делу за податке

Величина дела за податке *CAN* поруке дефинисан је бројем бајтова који може бити у распону од 0 до 8 бајтова. Колона ***DLC[Byte]***означава број бајтова података у *CAN* поруци.

#### Типови предаје CAN порука

Постоји више типова предаје порука кроз *CAN* мрежу. У пракси се најчешће користе следећи типови предаје:

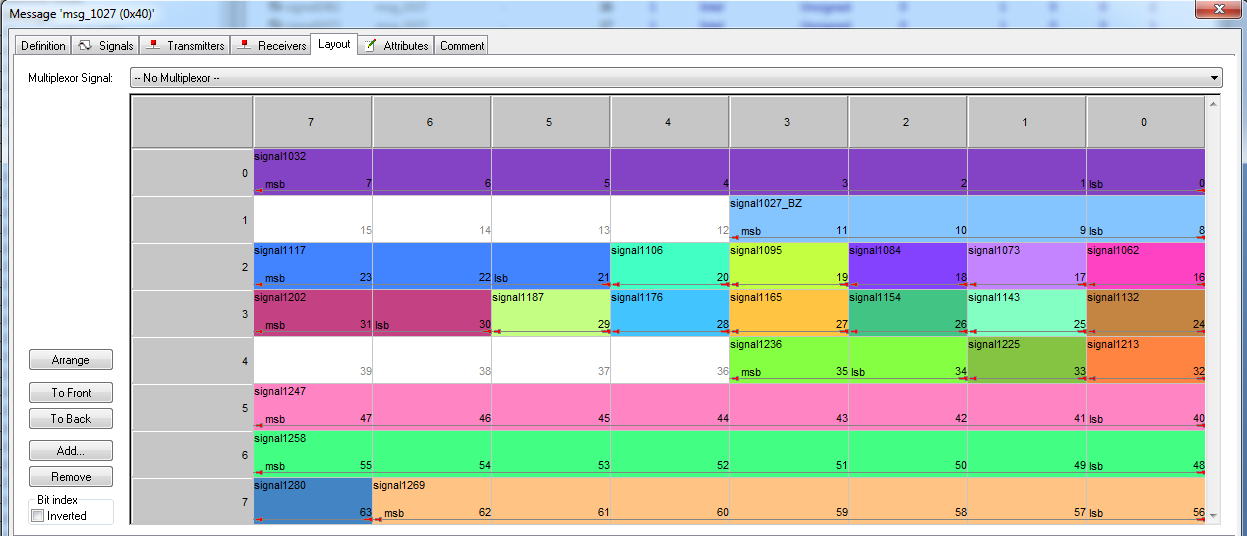
* Циклично - порука се шаље периодично на унапред дефинисану периоду (нпр. *10 ms*)
* Спонтано - супротно од периодичног слања, порука се шаље када нпр. се захтева нека сервисна информација унутар мреже па порука проследи ту информацију
* Циклично и спонтано – порука може да се шаље на оба претходно описана начина

#### Време циклуса CAN поруке

Време циклуса *CAN* поруке дефинише периоду генерисања поруке. Време генерисања сваке поруке назначено је у колони ***Cycle Time****.*

#### Сигнали унутар CAN поруке

Двоструким кликом на било коју поруку па затим избором картице ***Layout***, као што је приказано на слици испод, могу се видети сви дефинисани сигнали унутар изабране поруке. Сваки сигнал представљен је различитом бојом. Дат је почетни бит сваког сигнала као и распоред бита унутар сигнала, ***lsb***(енг. *least significant bit*)или ***msb***(енг. *most significant bit*).

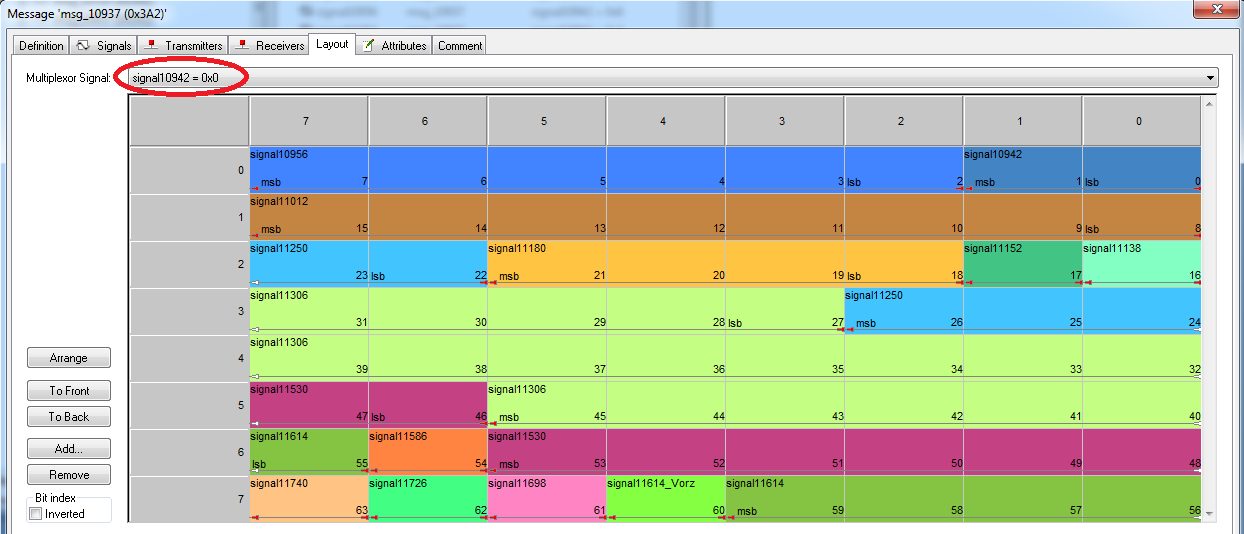


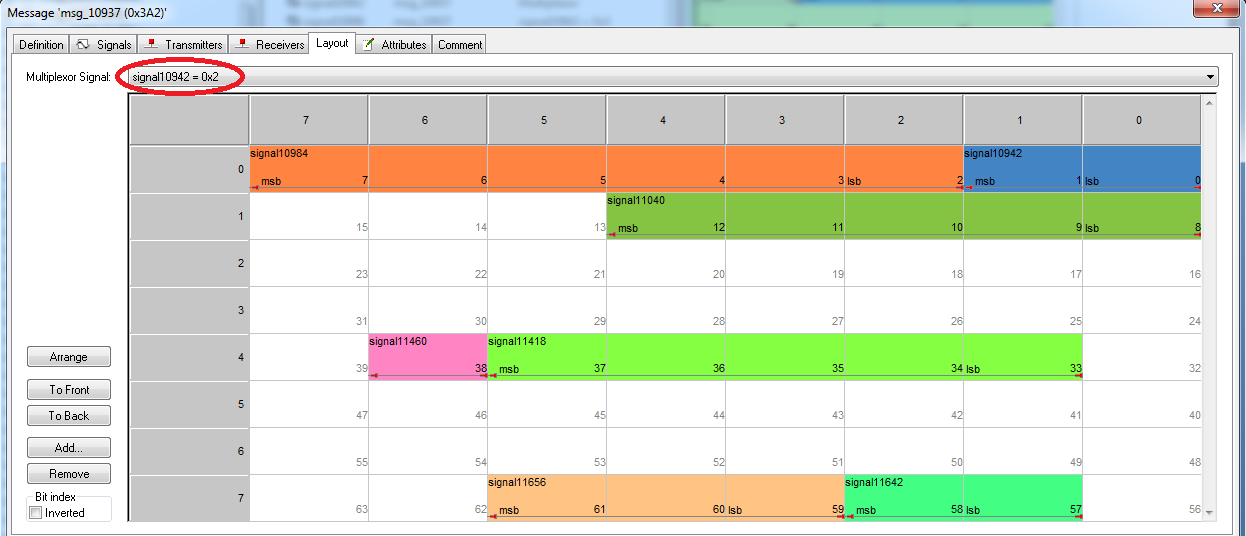
#### CAN multiplex поруке

У пракси постоји један специфичан формат *CAN* порука које се називају ***CAN multiplex*** поруке.Ове поруке специфичне су по томе што je унутар поруке дефинисан такозвани ***multiplex signal****.* У зависности од вредсности ***multiplex signal****-а* мења се и садржај поруке.

Један пример поруке која у себи има дефинисан *multiplex signal* је “*msg\_10937 (0x3A2)*”. Мултиплекс сигнал је *“signal\_10942”* и може имати следеће вредности: *0x0, 0x1, 0x2* и *0x3.*

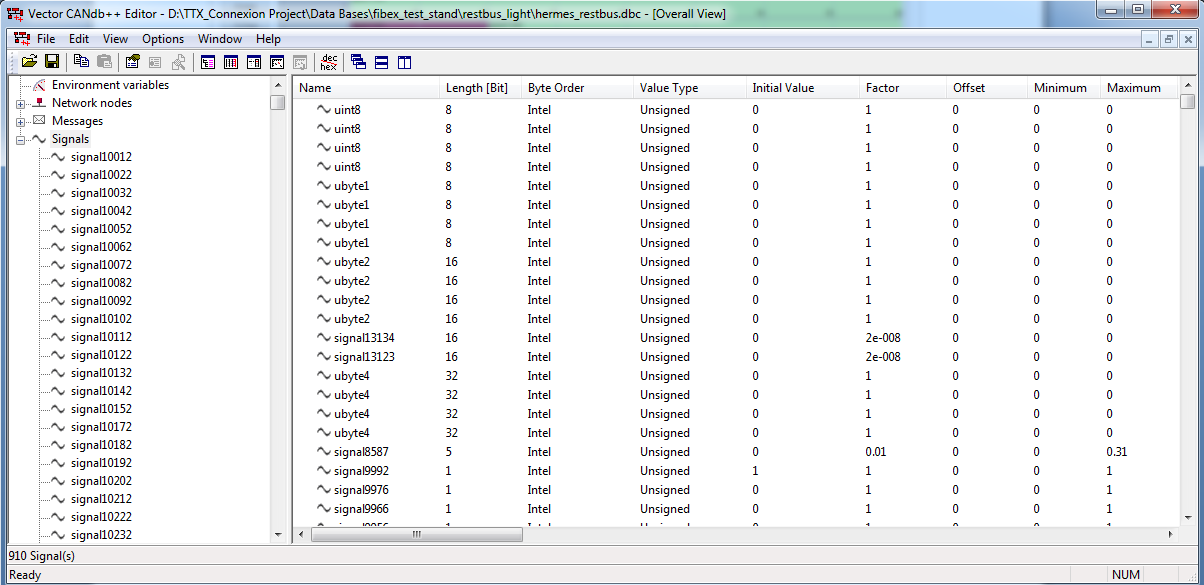
На сликама приказаним испод, уочите различите вреднсоти *multiplex signal-а* и садржаја порука у зависности од вредности споменутог сигнала.





### *CAN сигнали*

Све релевантне парамете за *CAN* сигнале можемо исчитати из *CANdb++ Editor* алата. На слици испод приказан је преглед карактеристика сигнала у *.dbc* датотеци.



Сваки сигнал окарактерисан је са:

* Симболичним именом
* Дужином која је изражена у битима
* Са распоредом бита унутар сигнала
* Са типом вртедности
* Иницијалном вредношћу
* Параметрима за конверзију из *raw* у *phisical* вредност
* Минималном и максималном вредности

#### Симболично име CAN сигнала

Симболично име сигнала остављено је креатору на избор.

#### Дужина сигнала

Дужина сигнала изражава се у битима и не сме бити већа од дужине поруке у којој се налази.

#### Распоред бита унутар сигнала

Распореда бита унутар сигнала може бити:

* ***Little endian*** *(Intel)*
* ***Big endian*** *(Motorola)*

#### Типови вредности сигнала

Преглед типова сигнала као и опсега за сваки од типова дат је у табели у наставку.

|  |  |
| --- | --- |
| Тип сигнала | Опис |
| ***Signed*** | *Signed integer*  *Вредност се креће у опсегу : 2^(дужина сигнала-1) до +2^( дужина сигнала -1)-1* |
| ***Unsigned*** | *Unsigned Integer*  *Вредност се креће у опсегу*: *0 до 2^дужина сигнала-1* |
| ***IEEE Float*** | *32 Bit IEEE Float* *Вредност се креће у опсегу*: *3,4 \* 10^-38 до 3,4 \* 10^38* *прецизност: 7 цифара* |
| ***IEEE Double*** | *64 Bit IEEE Double* *Вредност* *се креће у опсегу: 1,7 \* 10^-308 до 1,7 \* 10^308 прецизност: 15 цифара* |

#### Иницијална вредност сигнала

Иницијална вредност сигнала дефинише се од стране креатора *.dbc* датотеке.

#### Конверзија из raw у phisical вредност сигнала

Сваки сигнал на *CAN* магисталу унутар *CAN* мреже је послат у *raw* формату. Физичка вредност сигнала је вредност физичке количине (нпр. бризина, температура) која представља сигнал.

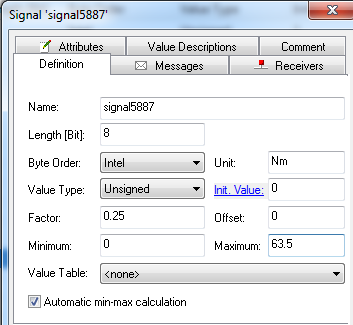
Формула која дефинише претварање сигнала из једног у други формат је:

[физичка вредност] =([сирова вредност]\*[фактор]) + [офсет]

где се параметри фактор и офсет дефинишу при креирању сигнала у *.dbc* датотеци.

Пример дефинисаних параметара може се видети на слици испод. До приказаних података се долази двоструким кликом на сигнал.

Уколико желимо да вредност са *CAN* магистрале буде и физичка вредност сигнала, параметар фактор се поставља на вредност 1 а параметар офсет на 0.

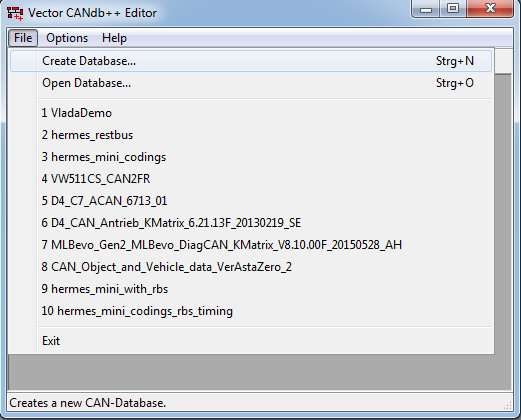
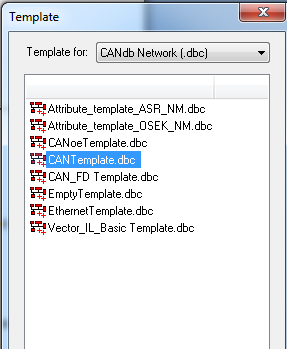


#### Минимална и максимална вредност сигнала

Минимална и максимална вредност сигнла се такође дефинишу од стране корисника. Битно је нагласити да се ове вредности односе на **физичке** вредности сигнала.

## Креација *.dbc* датотеке

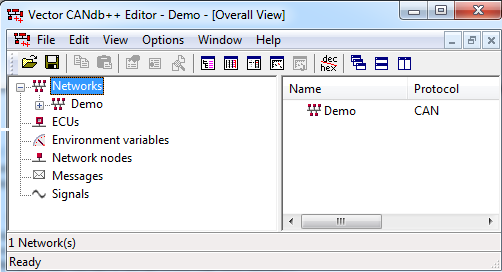
Креирање сопствене *.dbc* датотеке реализује се бирањем ***File*** опције из основног менија, па затим избором опције ***Create Database…***. Након отварања прозора, изабрати опцију ***CANTemplate.dbc***  као врсту *.dbc* датотеке коју ћемо креирати.

#### Назив CAN мреже

Након избора ***CANTemplate.dbc***  отвара се нови прозор где је потребно уписати назив .*dbc* датотеке коју ћете правити. Име које дате датотеци ће уједно бити и име *CAN* групе.

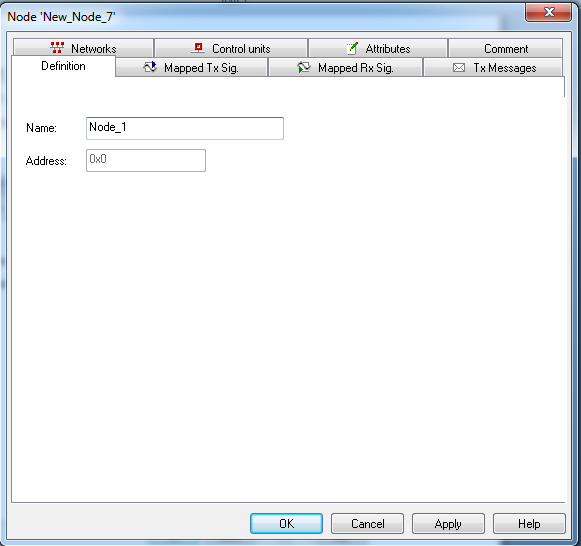
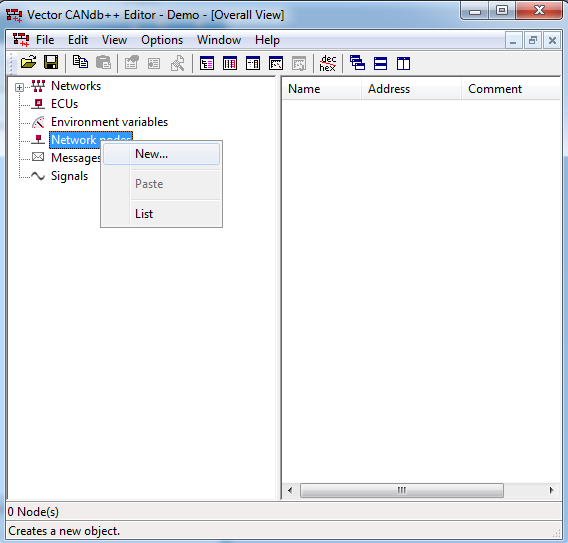
Након именовања *.dbc* датотеке и након клика на ***OK***дугме, појавиће се прозор као што је приказан на слици испод.



За потребе ових вежби, новој *.dbc* датотеци додељено је име *“Demo.dbc”*. Приликом креације својих *.dbc* датотека, студентима је остављен на избор назив својих *.dbc* датотека.

#### Креирање мрежног чвора

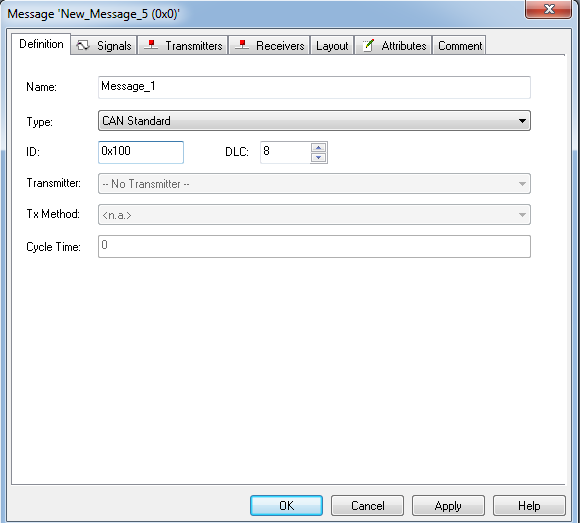
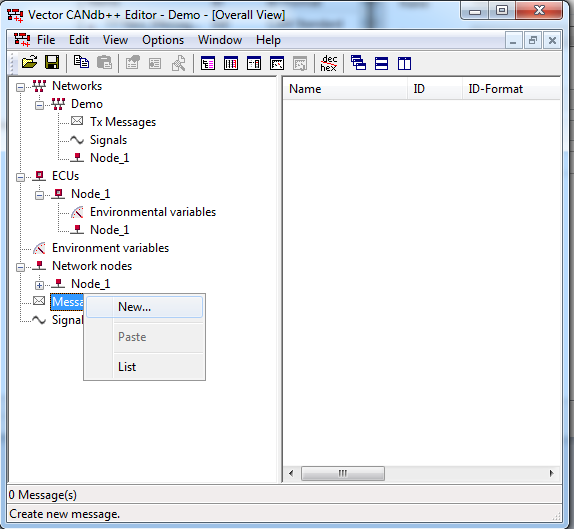
Kреацију нове .*dbc* датотеке најједноставније је извести додавањем новог мрежног чвора. Додавање новог мрежног чвора постиже се тако што се кликне десни клик на ***Network nodes*** па након тога из менија бира опција ***New…***Након избора опције ***New…*** приказаће се прозор као што је приказано на десној слици испод.



У склопу картице ***Definition****, у* поље ***Name*** унесите, по сопственом избору, назив мрежног чвора. Прођите по свим картицама. Тренутно једина видљива инфорамција се може наћи под картицом ***Networks*** а то је назив мреже којој дефинисани мрежни чвор припада. Картице предајних порука, мапираних предајних и пријемних сигнала ће бити, за сада, празне.

#### Креирање нове поруке

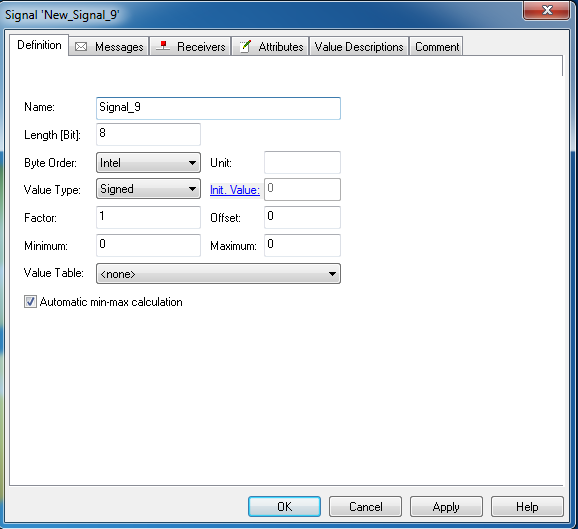
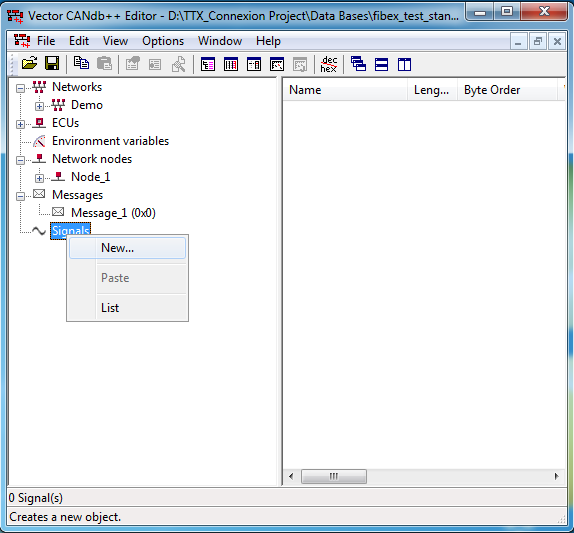
Као и код креирања мрежног чвора, додавање нове поруке постижемо десним кликом на ***Messages*** па након тога, изабрати опцију ***New...*** из понуђеног менија.



Након избора опције ***New…*** приказаће се прозор као што је приказано на десној слици изнад. Као што се може видети на слици, могуће је доделити име поруци(***Name***), изабрати формат поруке(***Type***) који може бити стандардан или проширен, доделити јединствени идентификациони број поруке(***DLC***) и одредити дужину корисног дела поруке у бајтовима (***DLC***).

#### Креирање новог сигнала

Додавање новог сигнала постижемо десним кликом на ***Signals*** и затим изабором опције ***New...*** из понуђеног менија.

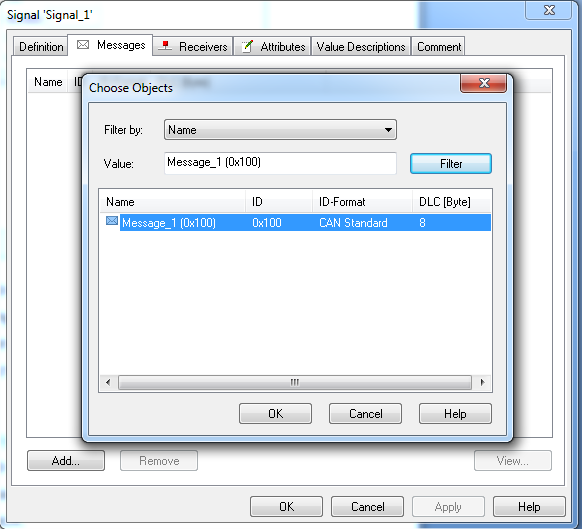
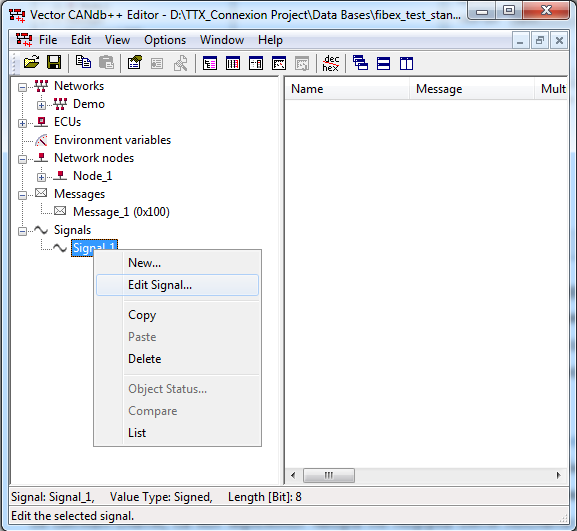


Након потврде за креацијом новог сигнала, приказаће се прозор као на десној слици изнад. Потребно је доделити име сигналу(***Name***), дефиниисти дужину сигнала у битима(***Length[Bit]***), дефинисати редослед бита унутар сигнала(***Byte Order***), тип сигнала(***Value Type***), минималну и максималну вредност као и параметре за конверзију вредности сигнала из сирове у физичку вредност.

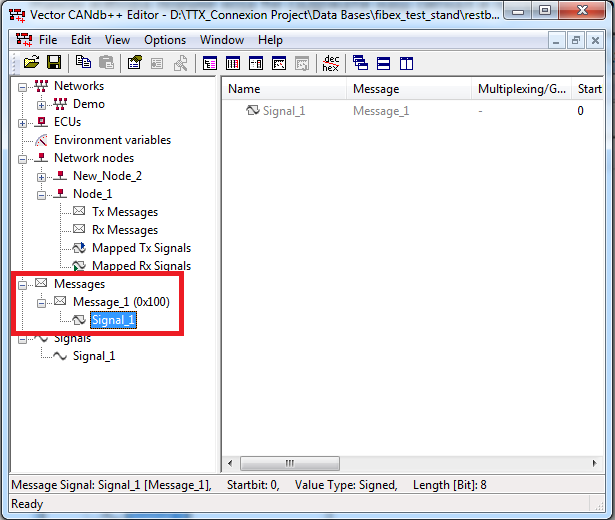
#### Међусобно повезивање сигнала, порука и мрежних чворова.

Након дефинисања мрежног чвора, поруке и сигнала, потребно је повезати ове величине. Када се каже повезати, мисли се на релације између ових величина. Дефинисати у којој поруци ће се сигнал слати, са ког мрежног чвора ће порука бити послата исл.

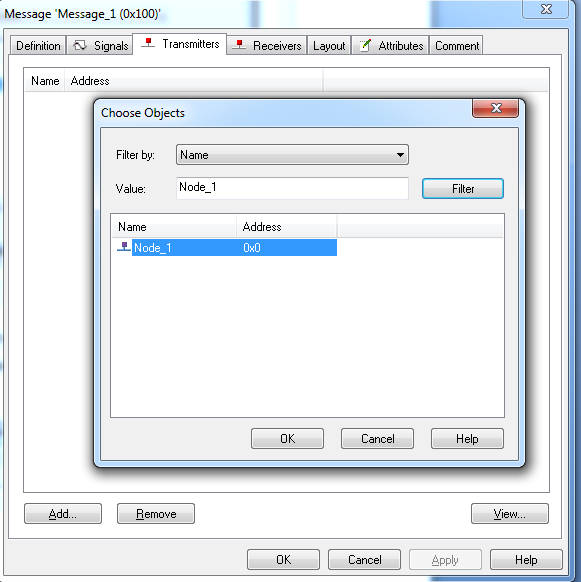
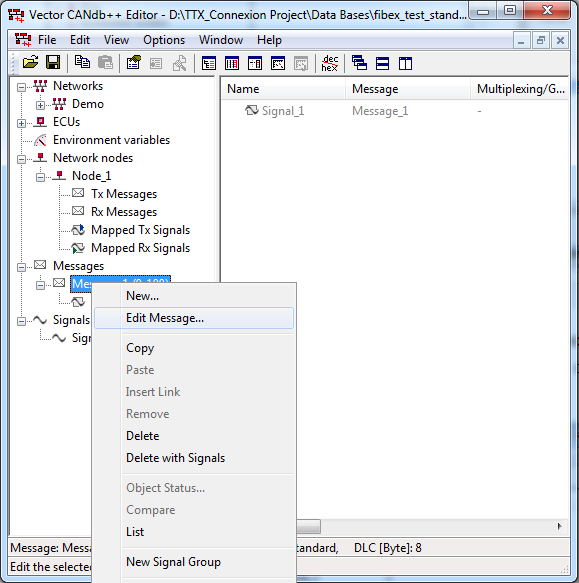
Кренућемо од убацивања сигнала у одговарајућу поруку. Десним кликом на претходно креирани сигнал и избором ***Edit Signal...*** из понуђеног менија отвара се нови прозор где је неопходно одабрати картицу ***Messages.*** Кликом на дугме ***Add…*** отвара се нови прозор, слика десно испод, где можете изабрати поруку која ће садржати сигнал.



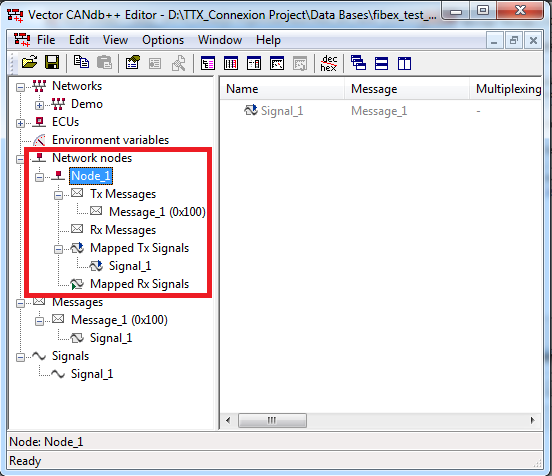
Након избора поруке која ће садржати наш сигнал и потврдом избора, кликом на ***OK*** у прозору, испод поруке у којој смо додали сигнал, додати сигнал постаје видљив као што се може видети на слици испод.



Следећи корак је избор мрежног чвора који ће бити предајни за поруку. Десним кликом на поруку и избором ***Edit Меssage*** из понуђеног менија отвара се нови прозор где је неопходно одабрати картицу ***Transmitters.*** Кликом на дугме ***Add…*** отвара се нови прозор, слика десно испод, где можете изабрати предајни мрежни чвор за изабрану поруку.



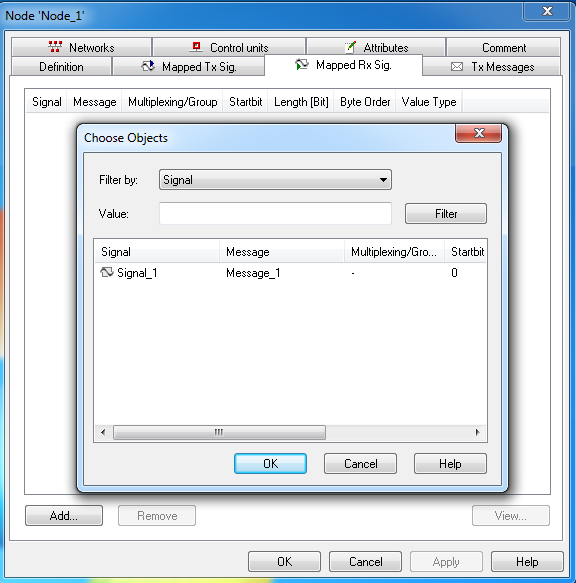
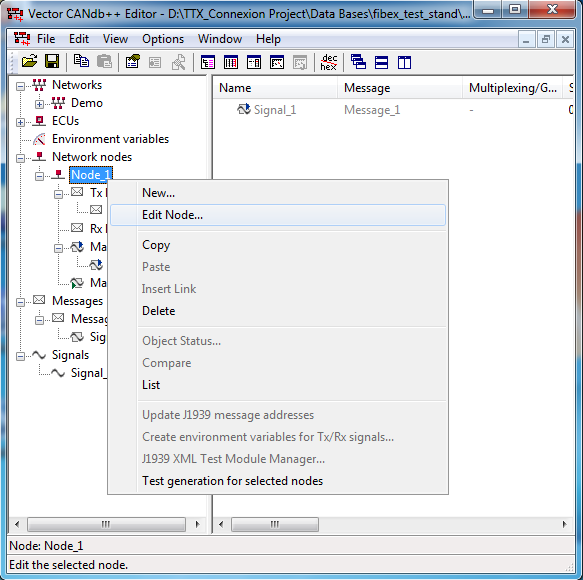
Након избора мрежног чвора и кликом на ***OK*** дугме, може се приметити да је у секцији предајних порука и сигнала мрежног чвора, доступна порука којој смо доделили мрежни чвор као предајни, као и сигнали који се налазе унутар порука, су излистани у листи мапираних предајних сигнала, као што је приказано на слици испод.



#### Задатак

Задатак овог дела вежби је да се заврши креирање започете *.dbc* датотеке. Већ креирана порука треба да садржи пет сигнала. Креирати два сигнала типа *Signed*, два сигнала типа *Unsigned* и ставити да сигнали буду дижине осам бита.Као пети сигнал креирати сигнал типа *Float.* По избору, за један синал дефинишите параметре за конверзију из *raw* у *phisical* вредност сигнала.

Следећи корак је да креирате пет **нових** сигнала. Типови и дужине сигнала нека буду исти као и претходно креираних пет сигнала. Креирајте нову поруку и у њој дефинишите креиране сигнале. Сви сигнали из ове поруке, као и сама порука, треба да буду у листи **пријемних** сигнала и порука већ креираног мрежног чвора. Мапирање пријемних сигнала и порука за одређени мрежни чвор се врши мапирањем само пријемних сигнала. Алат ће сам, на основу мапираног пријемног сигнала, у листу пријемних порука додати поруку којој припада изабрани сигнал. Да бисмо мапирали пријемни сигнал на жељени мрежни чвор неопходно је из менија који се добија десним кликом на мрежни чвор изабрати опцију ***Edit…*** као што је приказано на слици лево испод.



Након избора картице ***Mapped Rx Sig.*** и притиском на дугме ***Add..*** отвара се нови прозор са листом сигнала, приказано на слици горе десно. Након избора пријемних сигнала и потврдом на дугме ***OK***, изабрани сигнал као и порука у којој је дефинисан сигнал ће бити видљиви у листи пријемних сигнала и порука мрежног чвора.

Следећи задатак је да додате још један мрежни чвор. Поруке и сигнали који су били пријемни за предходан мрежни чвор треба да буду предајни док поруке и сигнали који су били предајни за предходни мрежни чвор треба да буду пријемни за ново креирани мрежни чвор.

# Употреба креиране *.dbc* датотеке

Након успешно креиране *.dbc* датотеке, на конкретном примеру упознаћемо се са алатима који користе *.dbc* датотеке. Обавићемо једноставно рутирање *CAN* порука и сигнала. Рутирање ћемо обавити помоћу два *CANcaseXL* уређаја(црно црвени уређаји на слици испод) и једног *TTX Connexion* уређаја(плаво црни уређај на слици испод).



## *CANcaseXL* уређај

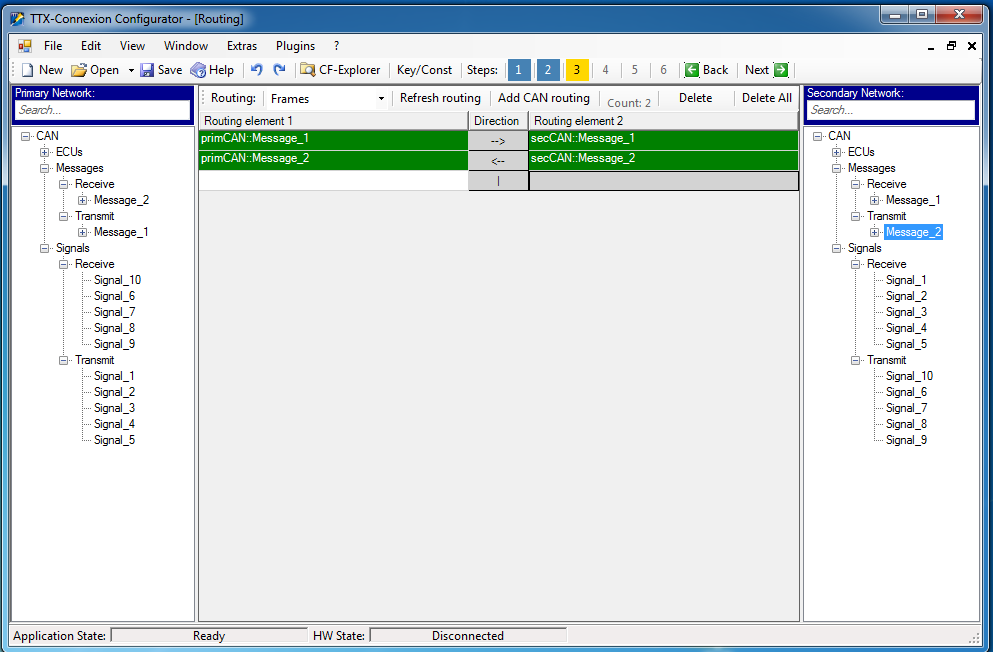
*CANcaseXL* уређај се састоји од два *CAN* канала обележена са *Channel 1* и *Channel 2*. На уређајима које ћемо користити на вежбама оспособљен је само први канал(лиценце су јако скупе).

Пре коришћења уређаја, неопходно га је конфигурисати. Алат који се користи за конфигурацију *CANcaseXL* уређаја је *CANoe.*

## *TTXConnexion* уређај

*TTXConnexion* уређај садржи два *FlexRay* канала(више о њима на следећим вежбама) и два *CAN* канала обележена као *CAN Primary* и *CAN Secondary*.

Као и у случају *CANcaseXL* уређаја, пре коришћења уређаја неопходно га је конфигурисати. Алат који се користи за конфигурацију *TTXConnexion* уређаја назива се *ТТX-Connexion Configurator.*



На слици изнад приказан је пример алата *ТТX-Connexion Configurator* у којем је дефинисано рутирање порука са *CAN Primary* на *CAN Secondary* канал и обрнуто.

**Напомена:**

* Алати *CANoe* и *TTX-Conexion Configurator* су у оквиру овог документа описани врло површно због своје обимности. На вежбама ће асистент детаљно показати креације кофигурационих датотеке са споменутим алатима и проћи све битиније детаље везане за алате.

### *CAN* протокол у *CANoe* алату

### Сам рад у *CANoe* алату можемо поделити на две велике целине:

* Рад у Offline моду
* Рад у Online моду

#### Offline мод

*Offline* мод у *CANoe* алату се користи за мониторинг и анализу саобраћаја целокупне или дела *CAN* мреже која је претходно снимљена у одговарајућу датотеку. Предност овавквог начина рада је тај, што неко може да сними саобраћај *CAN* мреже, сачува у одговарајућу датотеку и пошаље особи која то треба да анализира.

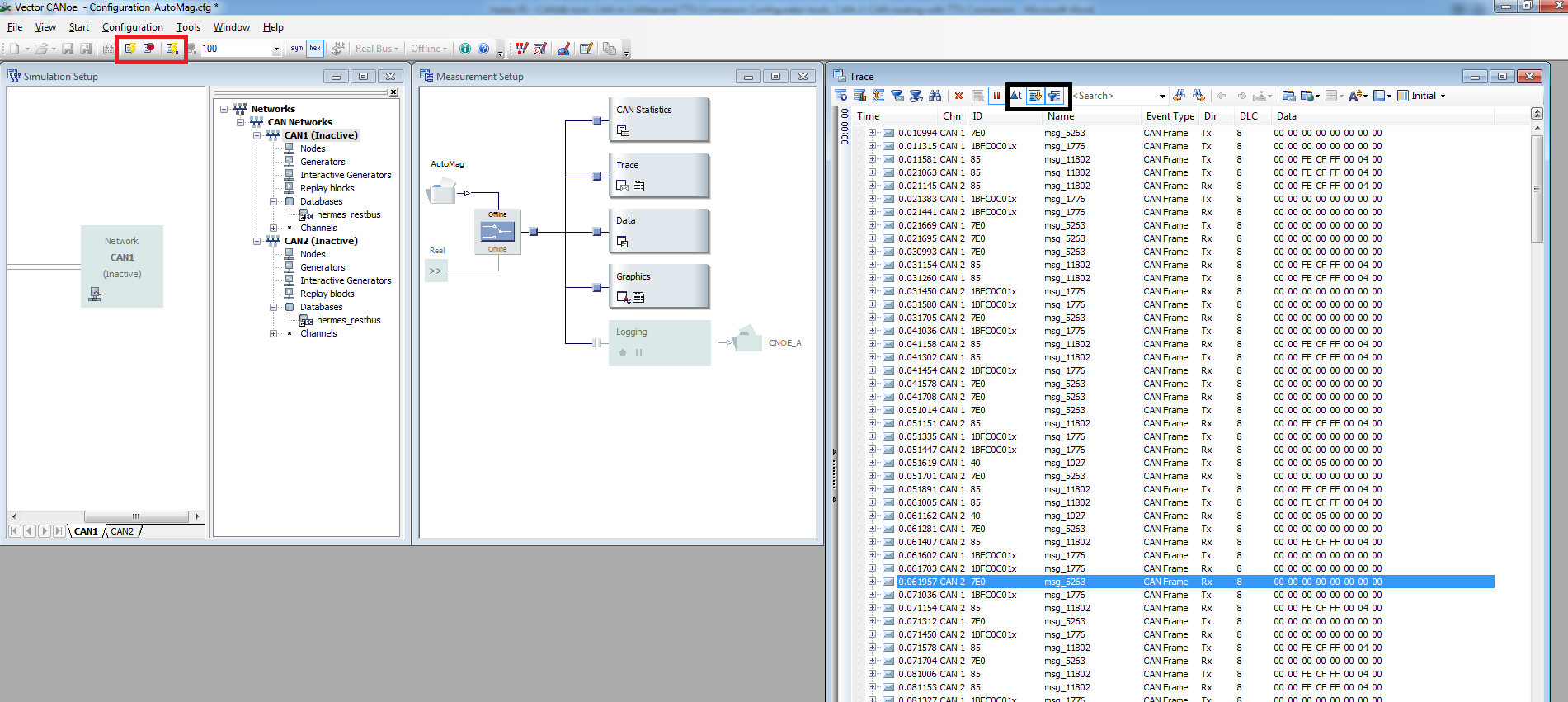
Може се снимити рад једне контролне јединице, начин на који контролна јединица генерише поруке и сигнале, периодичност слања порука на *CAN* магистралу исл.

У примеру који ћемо ми анализирати, снимљен је саобраћај који је добијен на следећи начин. *TTX Connexion* уређај је конфигурисан да рутира *CAN* саобраћај са своје примарне на секундарну страну. Први *CANcaseXL* уређај користио се као генератор *CAN* саобраћаја и он је био повезан на примарну страну *TTX Connexion* уређаја. Други *CANcaseXL* уређајкористио се као ***sniffer*** *CAN* порука и он је био повезан на секундарну страну *TTX Connexion* уређаја.

##### Задатак

Задатак овог дела вежби је да анализирате *CAN* саобраћај који је снимљен на начин описан у претходној секцији. Прво отворите већ направљену *CANoe* конфигурацију под називом *Configuration\_AutoMag*. Уколико вам алат затражи да изаберете *database*, изаберите *hermest\_restbus.dbc*.

Након успешног отварања сценарија, требало би да добијете прозор као на слици испод. Пробајте да кликнете на опције које су на слици уоквирене црвеним правоугаоником. Шта представња свака од тих опција и које промене уочавате након активације истих?



Проучите и опције под *Trace* прозором које су на слици уоквирене црним правоугаоником. Шта представња свака од тих опција и које промене уочавате након активације истих?

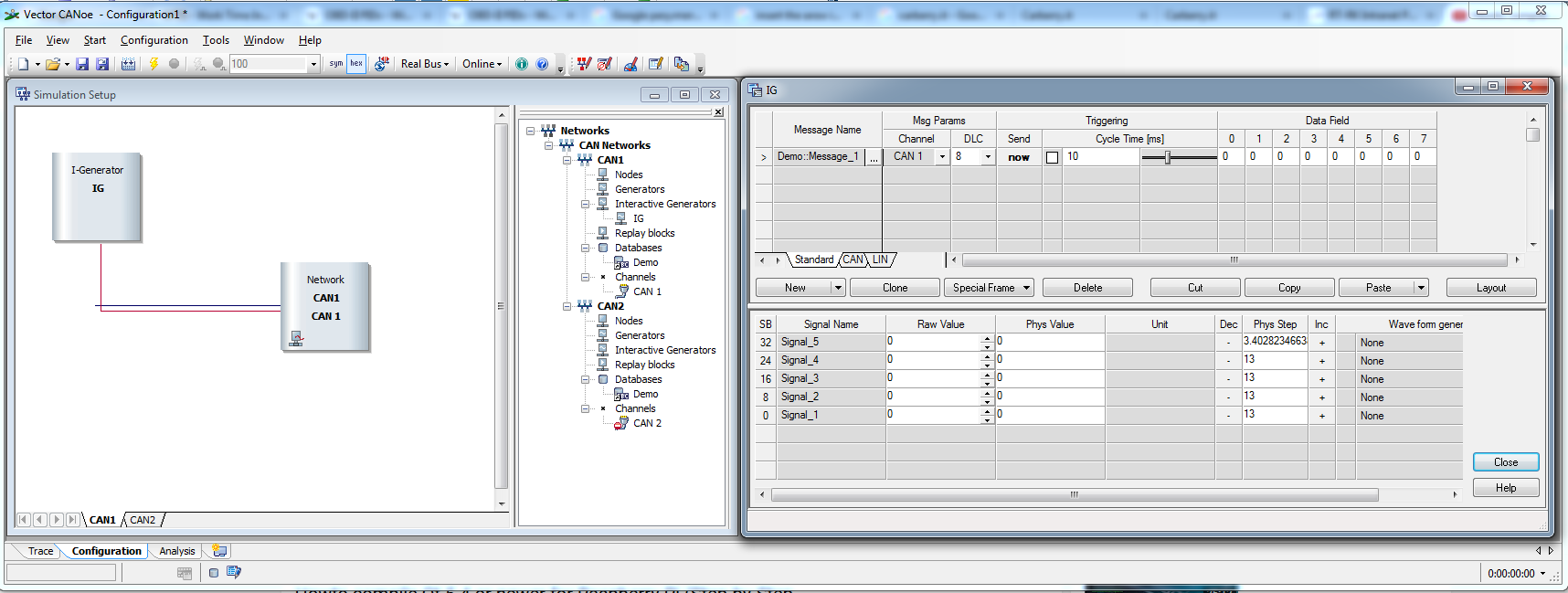
Посматрајте поруке унутар *Trace* прозора. Можете ли да уочите које су поруке стандардне а које проширене?

Анализирати сваку од колона унутар *Trace* прозора и схватити шта представља.

#### Online мод

### За потребе наших вежби, један *CANcaseXL* уређај ћемо користити као генератор *CAN* порука и сигнала(уређај из кога је усмерена црвена стрелица на слици изнад) док ћемо други *CANcaseXL* уређај користи као пријемник *CAN* порука и сигнала(уређај у кога је усмерена зелена стрелица на слици изнад).

Изглед једне већ креиране конфигурације у *CANoe* алату приказан је на слици испод.



##### Задатак

Задатак овог дела вежби је да студенти сами направе *CANoe* и *TTX Connexion* конфигурације које ће, као што је и делом написано у претходним поглављима, за циљ имати рутирање порука у оба правца, са *CAN Primary* на *CAN Secondary* канал и обрнуто. Пошто смо лимитирани са опремом, идеја је да по две групе студената раде заједно. За креирање конфигурационих сценарија користиће се .*dbc* датотеке које сте креирали у првом делу вежби.

Студенти који чекају да дођу на ред, слободно нека истраже поменуте алате и, уколико нису завршили реализацију претходних вежби, завршавају задатке датим у претходним вежбама.