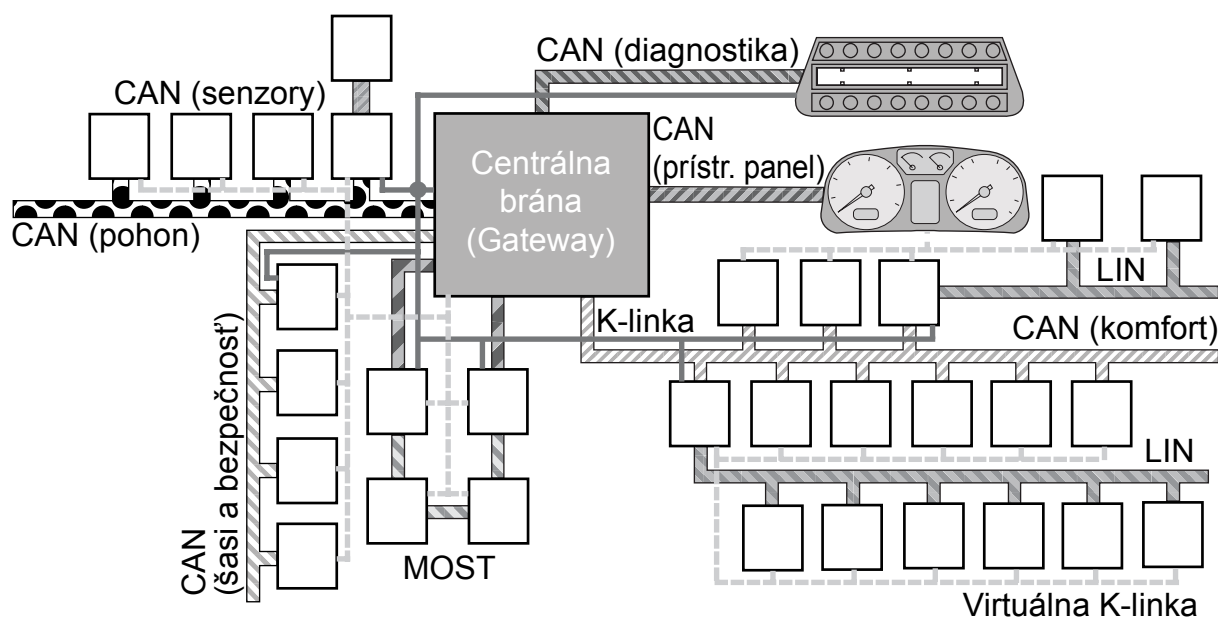


CAN a CANoe



*Ján Gamec
Mária Gamcová*

2014

Tiráž

NÁZOV: CAN a CANoe

AUTORI: Gamec Ján, Gamcová Mária

VYDAVATEL: Technická univerzita v Košiciach

ROK: 2014

VYDANIE: prvé

NÁKLAD: 50 ks

ROZSAH: 70 strán

ISBN: 978-80-553-1734-2

RECENZENTI: Ing. Emil Matúš, PhD.

Ing. Daniel Urdzík, PhD.

Obsah

ÚVOD	7
1 ARCHITEKTÚRA ELEKTRONICKÝCH SYSTÉMOV	8
1.1 História	8
1.1.1 Technológie súčasnosti	8
1.1.2 Vývojové trendy	9
1.2 Metódy architektúry elektronických systémov	10
1.2.1 Architektúra	10
1.2.2 Modely E/E architektúry	11
1.2.3 Sieť funkcie	12
1.2.3.1 Model funkcie	12
1.2.4 Sieť komponentov	12
1.2.4.1 Technologický model	12
1.2.4.2 Model uzlov	13
1.2.4.3 Model hardvéru riadiacej jednotky	13
1.2.4.4 Model softvéru riadiacej jednotky	13
1.2.4.5 Sieťový model komunikácie	14
1.2.5 Schéma zapojenia	14
1.2.5.1 Sieťový model napájania	14
1.2.6 Kabeláž a priestor	15
1.2.7 E/E vývojový proces	16
1.2.7.1 Manažment požiadaviek	17
1.2.7.2 Vývoj E/E architektúr	17
1.2.7.3 Hodnotenie modelov	17
1.3 Ciele a koncepcie AUTOSAR	18
2 ZBERNICE V AUTOMOBILOCH	19
2.1 FlexRay	19
2.1.1 Prenosové médium	20
2.1.2 Topológia	20
2.1.3 Prístup na zbernicu a časovanie	21
2.1.4 Synchronizácia	21
2.1.5 Ustanovenia v dynamickom segmente	22
2.1.6 Dátový rámec	22
2.1.6.1 Header	22
2.1.6.2 Uživatelské dáta - Payload	23
2.1.6.3 Prídavok -Trailer	23
2.1.7 Generovanie bitového toku rámca	23
2.1.8 Pracovné stavy	24
2.2 LIN	24
2.2.1 Systém prenosu	24
2.2.2 Prístup na zbernicu	25
2.2.3 LIN protokol	25
2.2.3.1 Dátový rámec	25
2.2.3.2 Hlavička	26
2.2.3.3 Dátové pole	27
2.2.3.4 Súbor opisu LIN	27
2.2.3.5 Plánovanie správ	27
2.2.3.6 Manažment siete	27

2.3	Ethernet, IP	27
2.3.1	Prenosový systém	28
2.3.2	Topológia	28
2.3.3	Protokol Ethernetu	28
2.3.3.1	Pristup na zbernicu	28
2.3.3.2	Dátový rámec	29
2.4	PSI5	29
2.4.1	Prenosový systém	29
2.5	MOST zbernica	31
2.5.1	Prenosový systém	32
2.5.1.1	Fyzická vrstva	32
2.5.1.2	Špeciálne vlastnosti MOST 150	32
2.5.2	Protokol	32
2.5.2.1	Prenos dát	32
2.5.2.2	Dátové rámce	32
2.5.2.3	Prenos riadiacich správ	33
2.5.2.4	Prenos multimediálnych dát	33
2.5.2.5	Prenos paketových dát	33
2.5.2.6	Topológia	34
2.5.2.7	Adresovanie	34
2.5.2.8	Administrátorské funkcie	34
2.5.2.9	MOST aplikačná vrstva	34
2.5.2.10	Funkčný blok	35
2.5.2.11	Triedy funkcií	35
2.5.2.12	Aplikácie	35
2.5.2.13	Štandardizácia	35
3	ZBERNICA CAN	36
3.1	Princíp prenosu dát	36
3.2	Súčasti dátovej zbernice CAN	37
3.2.1	Úlohy jednotlivých súčastí	37
3.2.2	Priebeh dátového prenosu	38
3.2.3	Dátový protokol	39
3.3	Funkcia	41
3.3.1	Prideľovanie dátovej zbernice CAN	41
3.4	Rušivé zdroje	42
4	PRÍKLAD ZNAKOV DÁTOVÝCH ZBERNÍC AUTOMOBILOV	43
4.1	Znaky dátovej zbernice CAN hnacieho ústrojenstva (Škoda)	43
4.2	Znaky dátovej zbernice CAN komfortného systému (Škoda)	43
5	CANOE	44
5.1	Prehľad	44
5.1.1	Menu	46
5.1.2	Dialogy	47
5.1.3	(Measurement and Simulation Setup)	48
5.1.4	Ovládanie myšou	49
5.1.5	Ovládanie klávesnicou	49

5.2	CANoe – stručné oboznámenie	49
5.2.1	Prípravy	49
5.2.2	Nastavenie CAN zbernice	51
5.2.3	Prenos dát	52
5.2.4	Okná vyhodnocovania	54
5.2.5	Práca so symbolickými dátami	56
5.2.6	Analýza hodnôt signálu v okne Data Window	57
5.2.7	Analýza odpovedí signálu v okne grafiky(Graphics window)	59
5.2.8	Použitie databázy vo vysielaných správach	59
5.2.9	Záznam merania	60
5.2.10	Vyhodnotenie Log File	62
5.3	Vytváranie CAPL programu	63
5.4	Simulácia distribuovaných systémov v CANoe	64
5.5	Tvorba databázy	65
5.6	Tvorba panelov (periférii)	67
5.7	Tvorba modelov uzlov	68
	POUŽITÁ A ODPORÚČANÁ LITERATÚRA	70

Úvod

Za posledné štvrtstoročie sa nárast automatizácie vo všetkých odvetviach priemyslu prudko rozbehol a pokračuje nezadržateľným tempom. Mikroprocesorové systémy dnes poskytujú dostatočnú podporu i pre najzložitejšie elektronické riešenia. Tak je tomu aj v oblasti automobilového priemyslu. Samozrejme aj dnešné požiadavky na sériový automobil sú neporovnateľne vyššie ako v minulosti.

Práve elektronika a jej široké možnosti dali priestor novým systémom, ktoré z obyčajného automobilu vytvárajú dômyselný, do istej miery i inteligentný a samostatne pracujúci celok. Vďaka jej dopadu na riadenie horenia, bezpečnosť, spoľahlivosť a komfort je možné vyrábať automobily spĺňajúce tie najprísnejšie normy a požiadavky. Automobil dostal vďaka elektronike nový rozmer. Svoju inteligenciu a praktickosť demonštruje dnes radom asistenčných, protikolíznych a núdzových systémov. Tiež je v súčasnosti trendom znižovať energetickú závislosť automobilov, čomu elektronika jednoznačne napomáha. Pre splnenie, často protikladne pôsobiacich požiadaviek je nevyhnutná výmena značných objemov dát v relatívne krátkom čase, a to v prostredí, ktoré často býva degradované rušením a náročnými fyzickými podmienkami. Špeciálne vyvinuté sieťové technológie sú jedným z predpokladov pre splnenie týchto požiadaviek.

1 Architektúra elektronických systémov

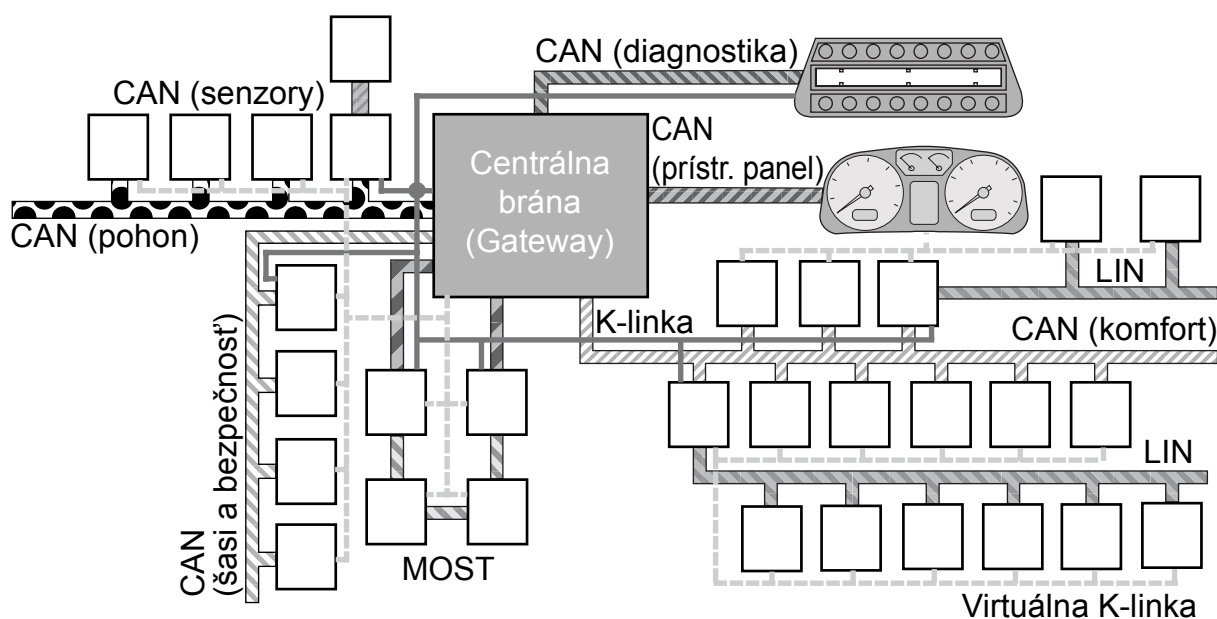
1.1 História

Kvôli značnému počtu komunikujúcich elektronických častí je v automobile použitý sieťový systém, ktorý je schopný pracovať s relatívne vysokou rýchlosťou (až 1Mb/s) v tzv. „nehostinnom“ prostredí. Príkladom takého sieťového systému je CAN (Controller Area Network), používaný pre prepájanie riadiacich jednotiek automobilu, ktorý vyhovuje požiadavkám na komunikáciu z hľadiska ich zložitosti, parametrov, jednoduchosti konštrukcie rozhrania resp. vedenia i jednoduchosti prenášania informácií. V automobiloch sa využívajú a nepretržite vyvíjajú aj nové systémy ako drôtovej tak aj bezdrôtovej komunikácie. Za jeden z prvých príkladov sieťového prepojenia elektronických systémov možno považovať vývoj riadiaceho systému trakcie (TCS). Toto sieťové prepojenie bolo v počiatkoch realizované výlučne mechanickým spôsobom. Poloha klapky v nasávacom potrubí motora s vnútorným spaľovaním bola nastavovaná zariadením, ktoré mohlo byť ovládané systémom riadenia trakcie. Riadenie motora nemohlo rozlíšiť či pohyb klapky spôsobil vodič alebo TCS. Ďalší stav priniesla realizácia elektronického prepojenia riadiacej jednotky motora prostredníctvom interfejsu s impulzovo-šírkovou moduláciou (PWM-Pulse Width Modulation) pre zlepšenie dynamickej odpovede. Tak sa dal dosiahnuť prenos signálu do riadiacej jednotky motora, aby sa redukoval krútiaci moment motora. Daný efekt sa dosahoval škrtaním prívodu vzduchu klapkou, blokovaním vstrekovania paliva a zmenou doby predstihu zapalovania.

V dôsledku silnejúceho tlaku na emisné normy neboli možnosti, ktoré poskytovala väzba medzi manažmentom motora a TCS dostačujúce. Bolo preto potrebné nájsť a prijať výkonnejšie prepojenie, ktoré by dovoľovalo rýchlejší prenos medzi manažmentom motora a TCS za účelom optimalizácie požadovaného momentu a dynamickej odozvy. Do riadiacej jednotky TCS bola, na rozdiel od predchádzajúceho, prenášaná informácia o aktuálnom krútiacom momente, otáčkach a aktuálne nastavená rezerva. To predstavovalo nárast zložitosti a nákladov vo vzťahu k počtu káblov, ktoré by boli nevyhnutné pre prenos takého množstva dát napr. prostredníctvom PWM rozhrania. V roku 1991 bol zavedený systém zberníc CAN ako alternatíva káblového prepojenia jednotlivých signálov. Tak boli položené základy moderných sieťových systémov v automobiloch.

1.1.1 Technológie súčasnosti

V súčasných automobiloch sú riadiace jednotky (ECU-Electronic Control Unit) medzi sebou sieťovo prepojené priamo alebo nepriamo (napr. prostredníctvom brán- Gateway), ako je to znázornené na obr. 1. Sieťová štruktúra pozostáva z mnohých riadiacich jednotiek (napr. 60 alebo aj viac), ktoré medzi sebou komunikujú prostredníctvom CAN zberníc alebo ďalších komunikačných systémov ako sú FlexRay, MOST (Media Oriented Systems Transport) alebo LIN (Local Interconnect Network), K-linka (modifikácia sériovej linky pre automobily) alebo virtuálnej K-linky. Tak napríklad riadiaca jednotka ESP (Electronic Stability Program) poskytuje do siete informáciu o rýchlosti vozidla. Autorádio môže použiť túto informáciu na prispôbenie hlasitosti v závislosti od rýchlosti vozidla vo všetkých prípadoch.



Obr. 1. Prepojenie riadiacich jednotiek vo vozidle

Vďaka výkonnému sieťovému prepojeniu medzi jednotlivými riadiacimi jednotkami sa môže dosiahnuť množstvo nových funkcionalít, a to bez nutnosti ďalšieho prídavného hardvéru, iba prostredníctvom dátovej komunikácie a softvéru. Jedným takým príkladom je otváranie bočných okien prostredníctvom dlhšieho pôsobenia diaľkového rádiového ovládača pre systém centrálného zamykania. Tak napríklad vozidlo môže byť rovnomerne vetrané v letných mesiacoch ak sú dvere otvorené. Pohonná jednotka dverí a jednotka centrálného uzamykania si takto môžu vymieňať informácie. Softvér beží buď v jednej alebo druhej riadiacej jednotke. V mnohých riešeniach oba systémy zdieľajú jednu riadiacu jednotku takže spomínaná funkcionalita môže byť implementovaná ešte jednoduchšie.

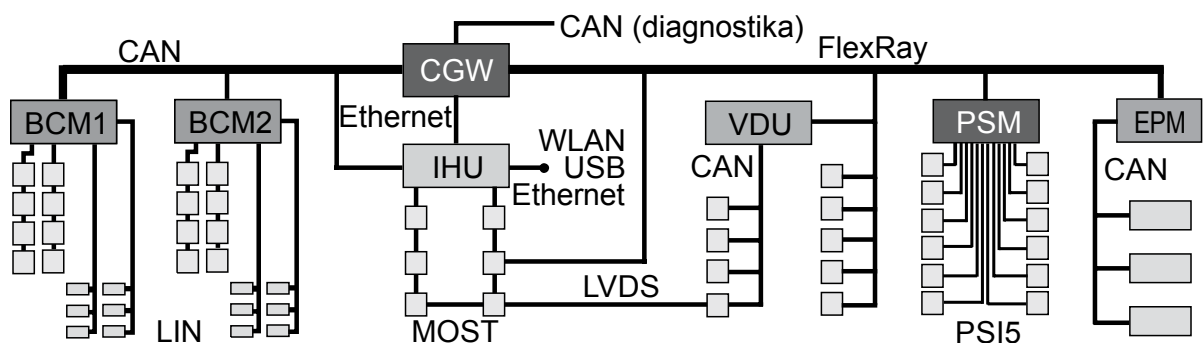
To demonštruje trend, ktorý sa najskôr objavil v elektronike karosérie (integrácia individuálnych ECU do centrálnej ECU). Tieto centrálné ECU sú spojené so senzormi a aktuátormi buď prostredníctvom individuálnych analógových prepojení alebo prostredníctvom zberníc. Zbernice významne redukovujú počet kontaktov v konektoroch riadiacich jednotiek a taktiež náklady na kabeláž. Sensory a aktuátory, ktoré sú pripojené prostredníctvom zberníc sa tiež označujú ako „inteligentné“. Tieto musia byť z dôvodu možnosti pripojenia na zbernicu vybavené elektronickými obvodmi, ktoré často obsahujú aj obvody predspracovania signálov senzorov alebo aktuátorov. Súčasne však použitie elektronických obvodov zvyšuje cenu senzorov a aktuátorov. Minimalizácia celkových nákladov, ktoré pozostávajú z nákladov na elektroniku a kabeláž reprezentuje dôležitú úlohu pri definovaní sieťových konceptov.

Tak napríklad logický obvod, ktorý plní funkciu ochrany prstov pred privretím do okna je mnohokrát začlenený priamo do ECU, ktorá riadi motor pre posun okna. Aktivačný signál pri normálnej prevádzke, t.j. spomínané otváranie okna pomocou diaľkového ovládača centrálného uzamykania dverí, je prenášaný prostredníctvom LIN zbernice z centrálnej ECU elektroniky karosérie (BCM-Body Computer Module). V tomto ohľade sa hovorí o architektúre typu klient-server.

1.1.2 Vývojové trendy

Predtým spomínaná idea centralizácie a využívania inteligentných senzorov a aktuátorov v oblasti elektroniky karosérie našli uplatnenie aj v iných sférach funkcií vo vozidle

(informácie pre vodiča, jazdná dynamika a bezpečnosť) a bude pokračovať v rozširovaní v nadchádzajúcich generáciách vozidiel. V ďalšom sú pre kombináciu funkcií odlišných ECU v jednej riadiacej jednotke využívané počítače hlavných domén (obr. 2). Riadiace jednotky (ECU) inteligentných senzorov a aktuátorov sú závislé od týchto hlavných (master) počítačov (BCM, IHU, atď.). Funkcie, ktoré vykazujú vysoký stupeň výmeny povelov po sieti sú prioritne zabudované v softvéri týchto centrálnych počítačov. Aby mohli byť tieto funkcie prevádzkované na rozličných platformách ECU, a tak opätovne využité, je nevyhnutné využívať štandardnú softvérovú architektúru. Danú požiadavku sa podarilo dosiahnuť pomocou AUTOSAR iniciatívy (AUTomotive Open System Architecture-štandardizačný proces a špecifikácie partnerov a členov automobilového priemyslu celého sveta pri konsolidácii prác na automobilových aplikáciách, ktoré sú základom pre vývoj v oblasti otvorenej štandardizovanej softvérovej architektúry).



BCM-Body Computer Modul (Modul počítača karosérie),
 CAN-Controller Area Network,
 CGW-Centrálny Gateway,
 EPM-Engine & Powertrain Manager (Manažér motora a pohonu),
 IHU-Integrated Head Unit (Jednotka projekcie v zornom poli vodiča),
 LIN-Local Interconnected Network (Prepojenie medzi jednotkami sériovou zbernicou),
 LVDS-Low Voltage Differential Signaling (Signalizačný systém s diferenčným prenosom informácií),
 MOST-Media Oriented Systems Transport (Optická zbernica pre prenos multimediálnych informácií),
 PSI-Peripheral Sensor Interface (Rozhranie pre pripojenie periférnych senzorov),
 PSM-Passive Safety Manager (Manažér psívnej bezpečnosti),
 VDU-Vehicle Dynamics Unit (Jednotka dynamiky jazdy),
 WLAN-Wirless Local Area Network (Bezdrôtová sieť)

Obr. 2. Vývojový trend štruktúry elektroniky v budúcich automobiloch

Hlavné počítače domény (oblasti pôsobnosti) sú medzi sebou navzájom sieťovo prepojené prostredníctvom výkonnej tzv. chrbticovej „backbone“ siete, ktorú v obr. 3 reprezentujú zbernice CAN a FlexRay vzájomne prepojené centrálnou bránou – Central Gateway-om. Prístup k centrálnej sieti ako aj diagnostické pripojenie a aktualizácia softvéru sa taktiež zabezpečuje prostredníctvom gateway.

1.2 Metódy architektúry elektronických systémov

1.2.1 Architektúra

Ako narastá množstvo elektroniky a sietí v automobile, tak stúpajú požiadavky na účinnosť vývojového procesu a metód opisu architektúry elektrických a elektronických systémov.

Pojem „architektúra“ sa vo všeobecnosti vzťahuje k umeniu výstavby. V priemysle konštrukcií architekt navrhuje budovy kreslením plánov v rozličných pohľadoch z hľadiska dodávateľov jednotlivých prác pričom stanovuje podmienky v styčných bodoch na základe požiadaviek klienta. Plán predstavuje abstrakciu reality z pohľadu partikulárnych aspektov

(napr. geometrické podmienky, rozvody káblov, atď.). Budova nakoniec môže byť postavená na základe plánov, v ktorých sú zahrnuté všetky aspekty.

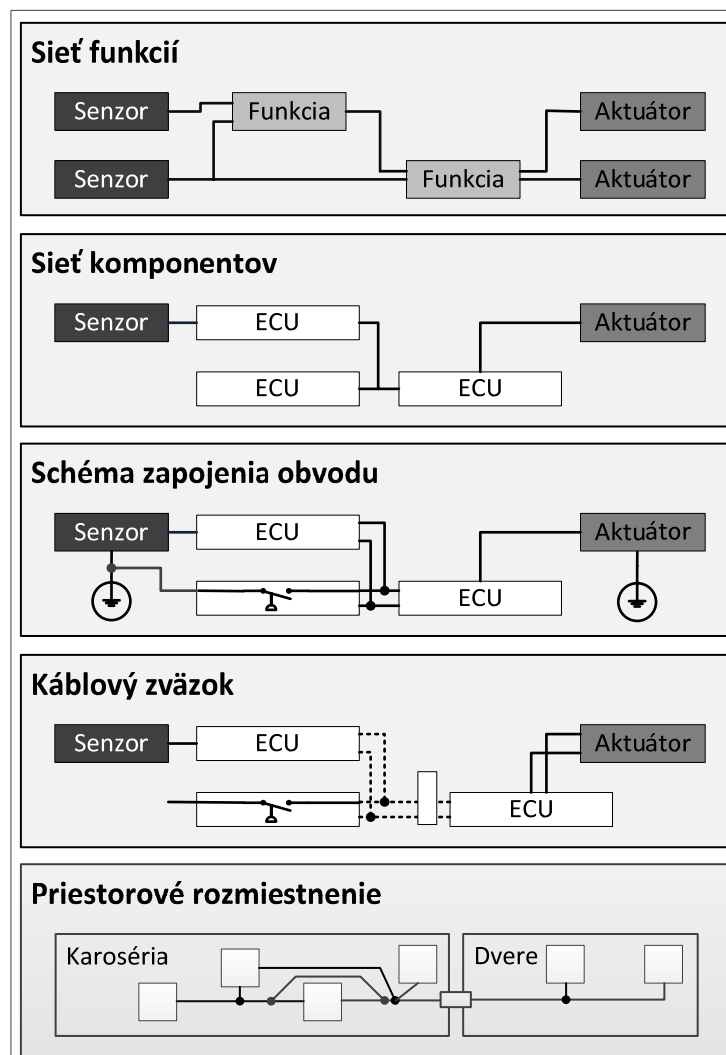
Keď sa to realizuje v automobile, potom sa to označuje ako „E/E architektúra“. „E/E“ znamená elektrické a elektronické aspekty motorového vozidla. „Plány“ E/E architekta sú v ďalšom vo všeobecnosti označované ako „model“.

Výrobcovia automobilov a ich dodávatelia majú odlišné pohľady na to, koľko modelov a akého typu je potrebných pre konkrétny opis elektrických a elektronických systémov vo vozidle. Modely, ktoré budú ďalej predstavené preukázali úspešnosť v praxi a sú potrebným rámcom v E/E oblasti.

Pojem architektúra je často používaný v literatúre a publikáciách na opis modelov ako takých. Je treba jasne rozlíšiť medzi pracovným postupom (vývoj architektúry) a prezentáciou výsledkov (model).

1.2.2 Modely E/E architektúry

Modely E/E architektúry odrážajú výsledky rozličných integračných aspektov (aspektov prepojenia) elektronických systémov vo vozidle (obr.3.).



Obr. 3. Modely E/E architektúry

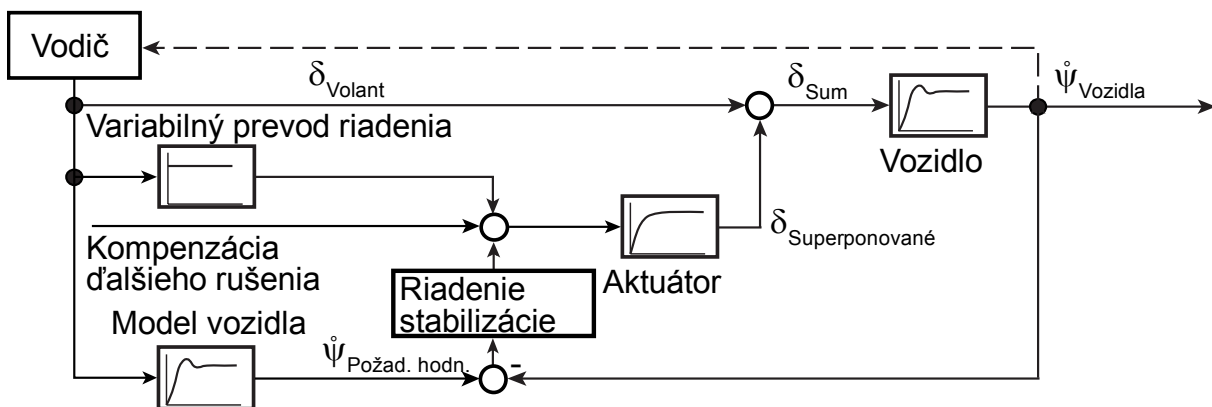
1.2.3 Sieť funkcie

1.2.3.1 Model funkcie

Modely funkcie sú predbežné stavy konkrétnych technických systémov. Opisujú prvky prenosov, ktoré sú potrebné na realizáciu požadovaných výrazných technických vlastností bez zachádzania do konkrétnych technológií. Napríklad v prípade pokročilého riadenia s posilňovačom to znamená rozklad na prvky prenosov ako sú:

- Variabilný prevod riadenia;
- Riadenie stabilizácie;
- Model vozidla;
- Aktuátor;
- Vozidlo;
- Vodič.

Model funkcií (obr. 4.) sú zvyčajne vytvárané ako blokové schémy resp. grafy toku signálov podľa príslušnej normy (napr. DIN 19226).



Obr. 4. Bloková schéma štandardných prvkov pokročilého riadenia s posilňovačom (δ -uhol, $\dot{\psi}$ - uhlové zrýchlenie)

1.2.4 Sieť komponentov

1.2.4.1 Technologický model

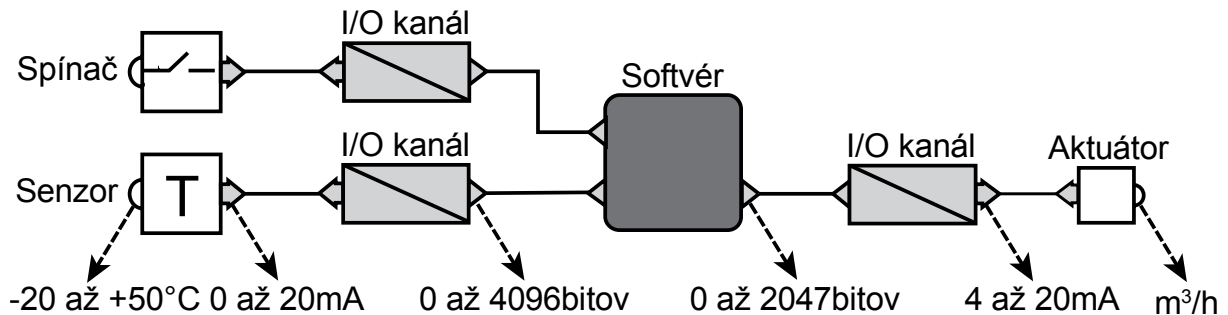
Technologický model opisuje aká technická realizácia je použitá na špecifikované prvky prenosu bez toho, aby sa vopred kombinovali do modulov, akými sú elektronické riadiace jednotky. Vytvárajú sa „technologické bloky“.

Filtrácia signálu môže byť tak realizovaná z diskretných prvkov ako číslicový obvod alebo ako filter, ktorý ako softvér je implementovaný v mikroprocesore. Rovnako funkcia radiča môže byť realizovaná z diskretných súčiastok alebo ako mikroradič (mikropočítač). Stabilizácia napätia sa dá dosiahnuť buď filtračným kondenzátormi alebo DC/DC meničom.

Rozhodnutie pre tú-ktorú realizáciu je na jednej strane ovplyvňované funkciou a na druhej strane nákladmi. Predtým, kým sa technologické bloky pospájajú do modulov vo forme ECU, je prvým krokom hľadanie synergie s ďalšími technologickými blokmi, ktoré sú súčasťou integrácie. Vytvorí sa technologicky aktívny reťazec (obr. 5). Ak napríklad je pre linku aktívneho reťazca vhodná špecifická sensorová technológia, ktorého signál je potrebný pre iný aktívny reťazec tak sa využije. To sa stáva aj keď je daný senzor nadmerne špecifikovaný pre ďalšieho používateľa, t.j. existujú nižšie požiadavky napr. dosiahnuteľný rozsah signálu alebo presnosť.

Napriek tomu je dôležité uchovať pôvodné požiadavky v databáze aj keď tato synergia sa nemusí vyskytovať v iných vozidlách.

Automobilový priemysel pre opis hardvéru zvyčajne používa nomenklatúru podľa normy DIN EN 60617.



Obr. 5. Príklad technologicky aktívneho reťazca

1.2.4.2 Model uzlov

Zlúčenie technologicky aktívnych reťazcov sú realizované kombináciou do skupín prostredníctvom uzlov („node“), ktoré sú na odlišných miestach. Dbá sa pritom na striktnú optimalizáciu nákladov pri integrácii technologických blokov. Tak napríklad sa integrujú časti softvéru niekoľkých technologicky aktívnych reťazcov v jednom spoločnom mikropočítači. Keď je možné, tak signály senzorov sú opätovne využité a podobne, ak je to možné sú spoločne využité aj aktuátory. História však ukazuje, že existuje synergia aj v mechanickej oblasti ako napríklad vytváranie podtlaku pre pneumatický posilňovač brzd cez vývod v nasávacom potrubí zážihového motora.

1.2.4.3 Model hardvéru riadiacej jednotky

Tento model reprezentuje štruktúru elektronického hardvéru individuálnej ECU. Je vytvorený alokáciou špecifických elektronických komponentov z technologicky aktívneho reťazca do elektronického modulu v uzle. ECU je preto vo všeobecnosti zberným bodom elektronických komponentov odlišných systémov, je „integračnou platformou“.

Softvér pre riadenie jednotlivých systémov pochádzajúci od odlišných zdrojov (výrobcov automobilov alebo ich dodávateľov) je integrovaný do mikropočítačov, ktoré sú súčasťou ECU. Tým, že sú ECU sieťovo prepojené, je možná podpora zložitých distribuovaných funkcií, ktoré využívajú senzory a aktuátory, umiestnené na rôznych pozíciách vo vozidle. V procese vývoja ECU sa na začiatku budú využívať obvyklé schémy obvodov ich elektrických alebo elektronických častí. Následne sa vytvorí mechanická časť návrhu a technológia konektorov. V prvotnej fáze konceptu je model obmedzený na veľmi hrubú reprezentáciu.

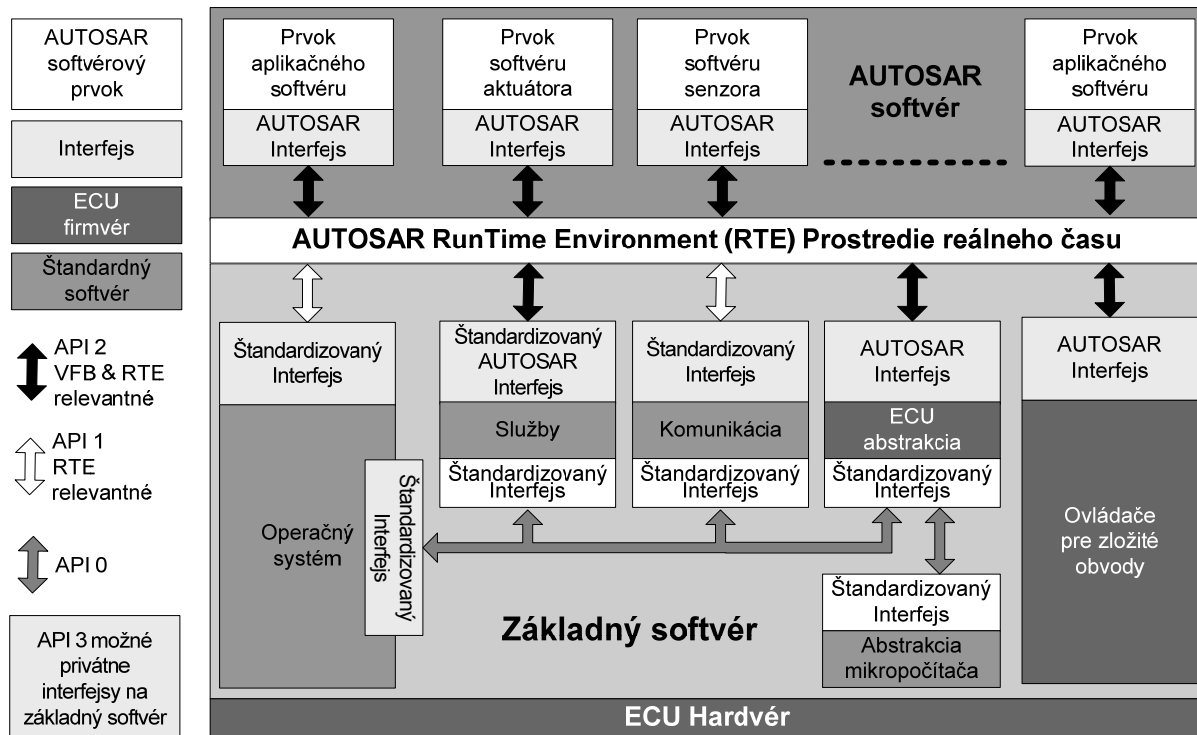
1.2.4.4 Model softvéru riadiacej jednotky

Pochádza z klasickej informačnej technológie (pre PC systémy). V pridružených modeloch sa dajú rozoznať niektoré metódy pre vývoj softvérovej architektúry (napr. „Product Line Approach“ – nástroj a technika pre tvorbu podobných softvérových systémov). Doteraz však v automobilovom priemysle nebol stanovený štandard pre vývoj architektúry. Vznikom AUTOSAR bola priamo definovaná softvérová architektúra pre motorové vozidlá.

AUTOSAR štandard definuje členenie softvéru blízko k hardvérovej úrovni a jeho rozhranie - interfejs s aplikačnými funkciami a stanovuje rozhrania medzi aplikačnými funkciami (obr.6).

AUTOSAR navyše definuje štandardizované formáty výmeny, ktoré sú podporované populárnymi nástrojmi na tvorbu modelov.

Typicky sa rozlišuje medzi základným a aplikačným softvérom. Blokmi základného softvéru sú napríklad softvérové ovládače - drajvre k obvodom, komunikačný softvér, operačný systém a abstrakcia hardvéru.



Obr. 6. Znáozornenie architektúry AUTOSAR

1.2.4.5 Sieťový model komunikácie

Teraz je na mieste sieťové prepojenie ECU a ich komunikačné vzťahy, pretože v predchádzajúcich krokoch boli pridelené technologické bloky vozidla do jednotlivých riadiacich jednotiek – ECU. Sieťový model komunikácie reprezentuje všetky ECU vo vozidle, ktoré je vybavené komunikačnou zbernicou a tak ich priamo alebo nepriamo medzi sebou prepája.

Každý signál, ktorý sa vymieňa medzi dvomi alebo viacerými ECU je priradený do vhodného systému zberníc. AUTOSAR pre tento účel definuje štandardizované formáty výmeny, ktoré umožňujú opis komunikácie po zbernici. Výmenné formáty AUTOSAR obsahujú od revízie 3.0 ASAM (Association for Standardization of Automation and Measuring Systems – Asociácia pre štandardizáciu systémov automatizácie a merania) štandard FIBEX (ASAM MCD-2 NET - štandard nazývaný FIBEX poskytuje zjednotený, XML - orientovaný opis rozhrania pre konfiguráciu softvéru siete v automobiloch).

1.2.5 Schéma zapojenia

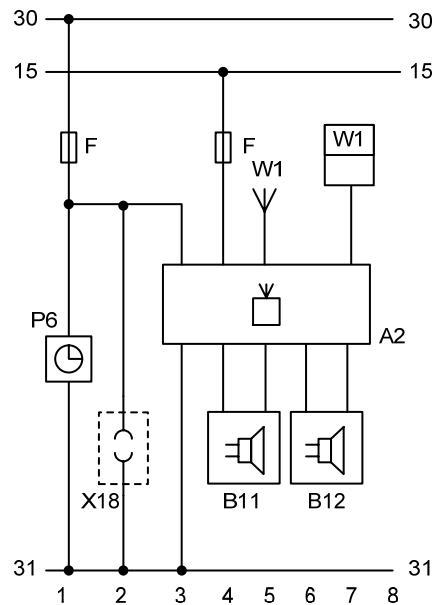
1.2.5.1 Sieťový model napájania

Priradenie technologických blokov riadiacim jednotkám a rozmiestnenie modulov senzorov aj aktuátorov má dopad na sieť elektrických záťaží/spotrebičov, ktoré vyžadujú vhodné

napájanie energiou. Na jednej strane je dôležité istiť jednotlivé elektrické obvody tak aby prípadný skrat nepoškodil celú sieť. Na druhej strane nie všetky obvody by mali byť napájané v každom pracovnom stave. Z toho dôvodu bol zavedený princíp „svoriek“. Napríklad na svorke 15 je napájacie napätie iba ak je zapnuté zapalovanie.

Na elektrickej schéme zapojenia (obr. 7) je znázornený elektrický rozvod a istenie jednotlivých modulov bez zohľadnenia pozície inštalácie. Tu je možné vidieť (v tejto schéme nie je znázornené) farebné označenie vodičov priradenie a istenie jednotlivých svoriek. Označovanie svoriek sa riadi podľa pravidiel, daných DIN 72552.

Kladný pól napájania je zvyčajne situovaný v hornej polovici zobrazenia, kým záporný pól (zem) je v dolnej polovici.



15, 30, 31 Označenie svoriek vozidla;
 1-8 Identifikácia sekcie
 A2 Rádio vozidla;
 B11, B12 Reproduktory;
 F Poistka;
 P6 Časovač;
 W1 Anténa vozidla;
 X18 Diagnostická zásuvka;

Obr. 7. Príklad schémy zapojenia rádia vo vozidle

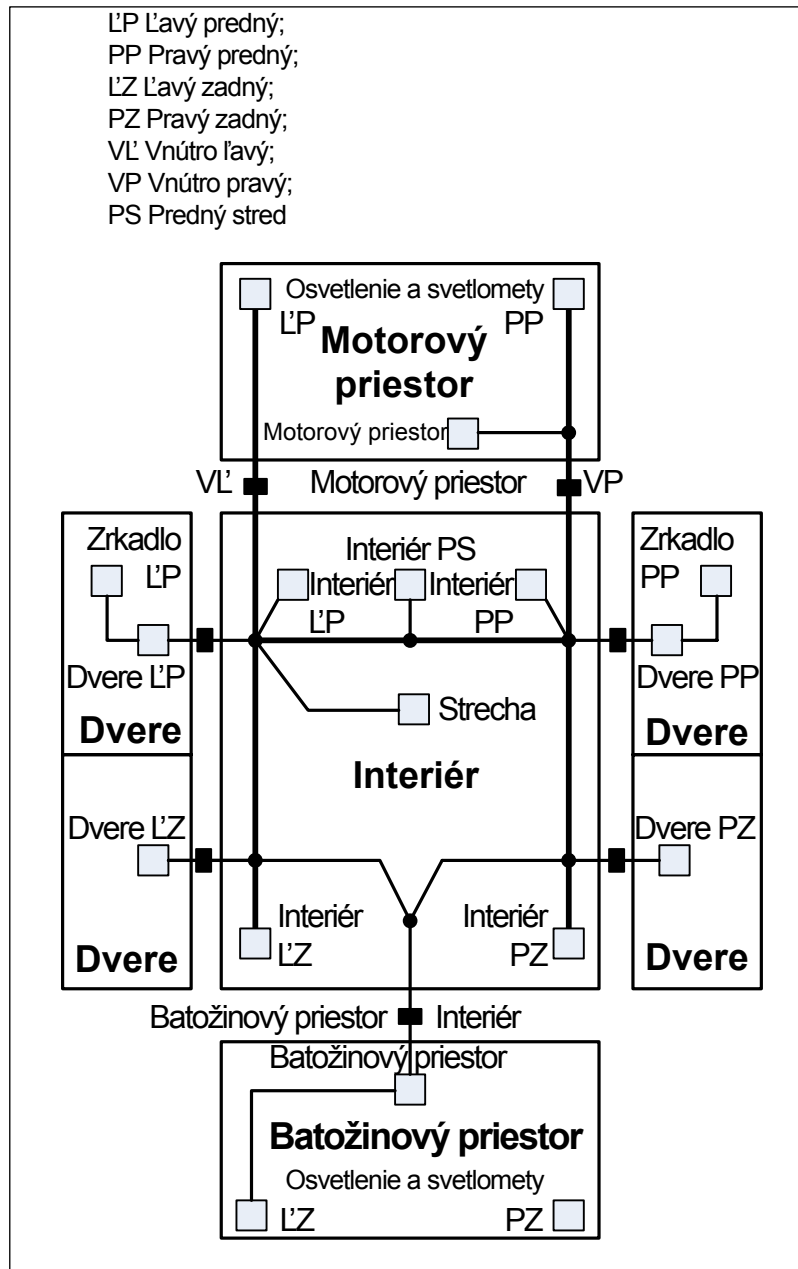
1.2.6 Kabeláž a priestor

Tento model zoskupuje elektrické a elektronické moduly podľa jednotlivých pozícií vo vozidle (obr. 8.). Týmto spôsobom sú prepojovacie káble medzi jednotlivými ECU a prívody napájania záťaží/spotrebičov prenesené spolu do vypletacích šablón káblových zväzkov. To vytvára kabeláž. Pritom sa musí sledovať množstvo rozličných hraničných podmienok ako napr.:

- výrobný postup (kabeláž v jednom celku alebo vo viacerých celkoch),
- prierez káblového zväzku (flexibilita),
- elektromagnetická kompatibilita (EMC),
- rozptyl tepla,
- hmotnosť,
- náklady (napr. med'),
- štruktúra kabeláže vo vozidle.

Štruktúra opisuje možné dráhy, ktorými sa kabeľáž vedie v karosérii ako napr. H štruktúra, ktorá pozostáva z dvoch hlavných prepojení, vedúcich spredu dozadu vozidla a priečky, ktorá spája ľavú a pravú stranu vozidla.

Vo fáze konceptu vozidla sú zvyčajne postačujúce dvojrozmerné modely.



Obr. 8. Príklad dvojrozmerného modelu priestoru

1.2.7 E/E vývojový proces

V E/E vývojovom procese sa prepájajú jednotlivé projektové stavy medzi sebou na logickom aj časovom základe a poskytuje kvalitatívne kritéria na začiatku aj na konci projektového stavu.

Pretože E/E architektúra automobilových aplikácií je ešte stále mladá disciplína, ešte stále sa výrazne líšia procesy jednotlivých výrobcov automobilov a ich dodávateľov. S tým súvisí množstvo a sled projektových stavov a kritérií kvality.

1.2.7.1 Manažment požiadaviek

Požiadavky nepopierateľne určujú rozhodnutia E/E architekta. Je vhodné rozlišovať medzi funkcionálnymi a nefunkcionálnymi požiadavkami. Funkcionálne požiadavky sa vzťahujú k požadovanému súhrnu vlastností keď je vozidlo používané. Nefunkcionálne požiadavky sa vzťahujú k technickému riešeniu a preto sú tiež známe ako projektové obmedzenia.

Takým obmedzením môže byť napr. dostupný priestor v stredovej konzole pre inštaláciu riadiacich jednotiek. Iným obmedzením môže byť prípustný rozptyl tepla v miestach, ktoré sú ovplyvňované výkonovou elektronikou, umiestnenou v danom priestore. Tak napr. audio zosilňovač vo vozidle je často umiestnený v batožinovom priestore, pretože vyvíjané teplo nemôže byť v priestore pre pasažierov primerane rozptýlené.

Keď je vytvorená dokumentácia funkcionálnych a nefunkcionálnych požiadaviek, začína sa konkrétnych E/E architektúr.

1.2.7.2 Vývoj E/E architektúr

Vývoj E/E architektúry sa môže uberať dvomi cestami. Prístup zdola nahor, ktorý začína od existujúcich komponentov a prístup zhora nadol, ktorý predstavuje implementáciu všetkých predtým opísaných krokov tvorby modelov a začína od funkcionálnych a nefunkcionálnych požiadaviek.

Prístup zdola nahor znamená – v priebehu tvorby E/E architektúry sa začína od funkcionality existujúcich komponentov – tieto komponenty sú v ďalšom doplnené funkčnými a komunikačnými aspektmi a vykonajú sa príslušné kroky modelovania. Tento prístup sa typicky volí pri tvorbe E/E architektúr následných generácií existujúcich platforiem vozidiel.

Prístup zhora nadol sa zameriava na zložitosť funkcií a typicky sa volí pre tvorbu E/E architektúr nových platforiem vozidiel.

Použitie nástrojov pre tvorbu E/E konceptov dovoľuje výmenu dát s vývojovými partnermi elektronických komponentov alebo pre tvorbu kabeláže.

1.2.7.3 Hodnotenie modelov

Vo všetkých prístupoch musí byť naplnené nasledovné: Pri prechode od jednej hierarchie modelu k nasledujúcej (napr. od modelu funkcie k technologickému modelu) je porovnaný zoznam hodnotiacich kritérií (napr. opakovateľnosť alebo testovateľnosť) s portfóliom vzorových riešení (napr. technológie zberníc). Hodnotenie vzorových riešení použitím kritérií dovoľuje aby riešenie malo formu na základe čisto požiadaviek funkcionality a nezvratných hraničných podmienok („MUST“ kritérium). Tento postup je tiež známy ako QFD (Quality Function Deployment – dislokácia funkcie kvality - metóda na transformáciu kvalitatívnych požiadaviek používateľa na kvantitatívne parametre, aby sa dislokovali funkcie upravujúce kvalitu a aby sa dislokovali metódy na dosiahnutie kvality návrhu do subsystémov a častí komponentov).

Alternatívna procedúra spočíva v porovnaní referenčného riešenia (napr. predchádzajúci sieťový model) použitím hodnotiaceho kritéria s alternatívnymi riešeniami. To poskytuje rýchle výsledky, ale možno nie globálne optimálne.

Pretože hodnotiace kritéria majú u automobilových výrobcov odlišnú váhu, elektronické systémy vozidiel sa niekedy jeden od druhého značne líšia.

1.3 Ciele a koncepcie AUTOSAR

Hlavnými cieľmi iniciatívy AUTOSAR sú štandardizácia základnej infraštruktúry ECU (základný softvér), výmenné formáty a rozhrania funkcionalít. Ich poslaním je náhrada predchádzajúcich, pre spoločnosti špecifických riešení. Použité sú modelovo orientované koncepcie a metódy, aby sa na najvyššej úrovni udržal kontinuálny rast so zložitou, ktorú prinášajú nové funkcionality. Požiadavky kvality a spoľahlivosti sú napĺňané opätovným používaním prijatých štandardov.

Na základe štandardizovanej softvérovej infraštruktúry, ktorá primárne pozostáva zo štandardných modulov, môže každý výrobca automobilov implementovať jeho špecifický obsah (aplikačný softvér).

V konkrétnych podmienkach sú AUTOSAR sledované tieto ciele:

- opätovné použitie softvéru v odlišných ECU, platformách vozidla a u výrobcov automobilov,
- podpora integrácie softvéru výrobcov tzv. tretích strán, ako v základnej tak aj aplikačnej oblasti,
- podpora presunu aplikačného softvéru medzi ECU (staticky),
- zámena štandardných hardvérových blokov (napr. CAN transceiver - vysielač-prijímač) bez toho, aby to viedlo k zmene aplikačného softvéru.

Softvérová architektúra, definovaná AUTOSAR, podporuje jasné oddelenie základného softvéru od aplikačného. To sa dosahuje niekoľkými úrovňami abstrakcie resp. oddelení základného softvéru – počnúc ovládačmi hardvéru cez zložitú infraštruktúru služieb až po AUTOSAR prostredie reálneho času (RTE – Runtime Environment).

Štandardné hardvérové bloky a im prináležiace ovládače môžu byť nahradené bez nutnosti zmien v aplikačnom softvéri, pretože rozhrania väčšiny modulov základného softvéru v jednotlivých vrstvách sú štandardizované.

Naopak, aplikačný softvér, ktorý je limitovaný, aby používal tieto štandardné rozhrania, môže byť v procese vývoja jednoduchšie vložený do ECU alebo tiež presunutý do inej ECU. Tak je napríklad možné pre ten istý aplikačný softvér radiča rýchlosti vozidla v závislosti od platformy vozidla aby bežal v riadiacej jednotke motora, riadiacej jednotke prevodovky alebo v inej riadiacej jednotke bez nutnosti jeho zmeny. To samozrejme vyžaduje dostatočne výkonné sieťové prepojenie.

Štandardizácia základného softvéru a jeho konfigurovateľnosť, ktorá závisí od aplikačného softvéru, dovoľuje opätovné použitie modulov základného softvéru v odlišných platformách automobilov a automobilových výrobcov. To zvyšuje kvalitu softvéru, keďže neexistujú produktovo špecifické zmeny, redukuje náklady na vývoj opätovným využitím vytvorením stabilného základu pre neustále narastajúcu zložitú a sieťové prepojenie aplikačných funkcií.

Zoskupenie AUTOSAR vyvíja modelovo orientované koncepty pre skoré potvrdzovanie návrhu systému. Kontrola sa vykonáva na základe formalizovaného opisu o tom či rozhrania aplikačného softvéru sú konzistentné vzhľadom ku ostatným, bez toho či aplikačný softvér môže vystupovať ako samostatný program.

2 Zbernice v automobiloch

V tabuľke 1 je stručný prehľad zbernic a ich častá klasifikácia v oblasti automobilov.

Tab.1. Klasifikácia systémov zbernic

Trieda A	
Rýchlosť prenosu dát	Nízke prenosové rýchlosti (do 10kbit/s)
Aplikácie	Sieťové prepojenie aktuátorov a senzorov
Reprezentant	LIN, PSI5
Trieda B	
Rýchlosť prenosu dát	Stredné prenosové rýchlosti (do 125kbit/s)
Aplikácie	Zložité mechanizmy pre ošetrovanie chýb, sieťové prepojenie riadiacich jednotiek v oblasti komponentov komfortu
Reprezentant	Pomalá CAN
Trieda C	
Rýchlosť prenosu dát	Vysoké prenosové rýchlosti (do 1Mbit/s)
Aplikácie	Požiadavky systémov reálneho času, sieťové prepojenie riadiacich jednotiek v oblasti pohonných agregátov a šasi
Reprezentant	Rýchla CAN
Trieda C+	
Rýchlosť prenosu dát	Veľmi vysoké prenosové rýchlosti (do 10Mbit/s)
Aplikácie	Požiadavky systémov reálneho času, sieťové prepojenie riadiacich jednotiek v oblasti pohonných agregátov a šasi
Reprezentant	FlexRay
Trieda D	
Rýchlosť prenosu dát	Veľmi vysoké prenosové rýchlosti (> 10Mbit/s)
Aplikácie	Sieťové prepojenie riadiacich jednotiek v oblasti telematických systémov a multimédií
Reprezentant	MOST

2.1 FlexRay

Je zbernica, ktorá bola navrhnutá v oblasti riadenia v oblasti automobilov. Zvláštny dôraz bol kladený na jej využitie v systémoch aktívnej bezpečnosti bez spätného mechanického zabezpečenia (X-by-Wire), kde sa vyžaduje determinizmus a odolnosť voči poruchám. Predpokladom jej využitia v oblasti prenosu audia a vysoko komprimovaného videa je hlavne jej schopnosť prenosu dát vysokou prenosovou rýchlosťou (neredundantná až do 20Mbit/s).

Hlavnými črtami sú:

- garantovaná latencia s časovým spúšťaním,
- možnosť riadenia prenosu informácií udalosťami so stanovovaním priorít,
- prenos informácie prostredníctvom jedného alebo dvoch kanálov,
- vysoká prenosová rýchlosť až do 10Mbit/s na jednom kanáli, paralelne na dvoch kanáloch až do 20Mbit/s,

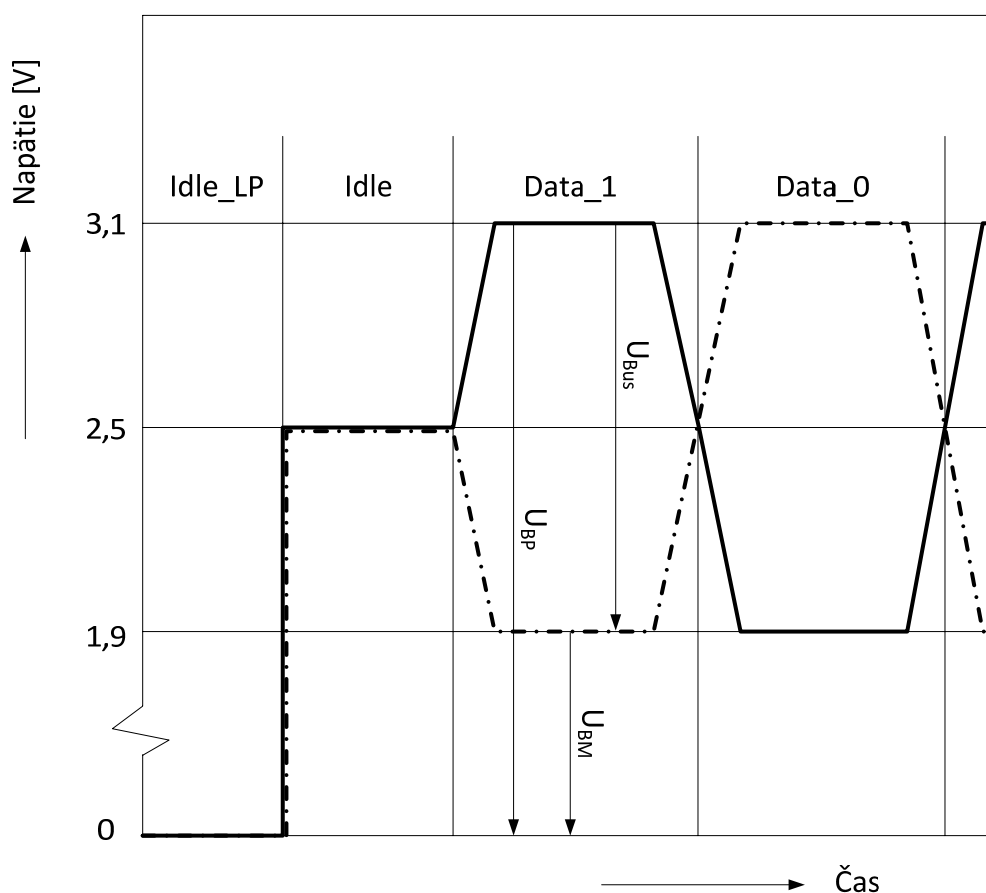
- štruktúra ako líniová sieť - zbernica, hviezdicová konfigurácia alebo ako zmiešaná forma.

Bližšie špecifikácie sú dostupné na webových stránkach konzorcia FlexRay.

2.1.1 Prenosové médium

Prenosovým médium FlexRay systému sú stáčané páry vodičov buď v tienenej alebo netienenej forme. Každý FlexRay kanál pozostáva z dvoch vodičov, Bus-Plus (BP) a Bus-Minus (BM). FlexRay používa NRZ (Non Return to Zero – bez návratu k nule) tvar signálu. Stav zbernice sa vyhodnocuje meraním diferencie napätia medzi BP a BM vodičom. Dátový prenos je tým menej citlivý na externé elektromagnetické rušenie, pretože to pôsobí rovnako na oba vodiče a pri odčítaní sa jeho vplyv ruší.

Po pripojení dvoch hodnôt napätí na dva vodiče jedného kanála zbernice môžu nastať na zbernici štyri stavy. Tieto stavy sú označované ako „Idle_LP“ (LP-Low Power, nízky príkon), „Idle“ (nečinný), „Data_0“ a „Data_1“ (obr. 9). Stav Idle_LP je stav, kedy je na vodičoch BP a BM napätie -200mV a 200mV vzhľadom k zemi. V stave „Idle“ na oboch vodičoch napätie 2,5V a maximálna diferencia medzi BP a BM je 30mV. Aby bol kanál v stave „Data_0“, musí byť na vysielačom uzle medzi vodičmi BP a BM rozdiel napätí -600mV a v stave „Data_1“ +600mV.



Obr. 9. Stavy a napätia na zbernici FlexRay

2.1.2 Topológia

Siete FlexRay môžu mať štruktúru ako zbernice alebo ako hviezdicové siete. Dve hviezdy môžu byť spojené do kaskády, pričom sa berie do úvahy časové oneskorenie. Taktiež je

možná topológia, kde je niekoľko zberníc spojených do hviezdy. Pretože obidva kanály systému FlexRay sú nezávislé, je možná ich odlišná topológia. Napríklad jeden kanál je štruktúrovaný ako aktívna hviezda a druhý ako zbernica - líniová sieť.

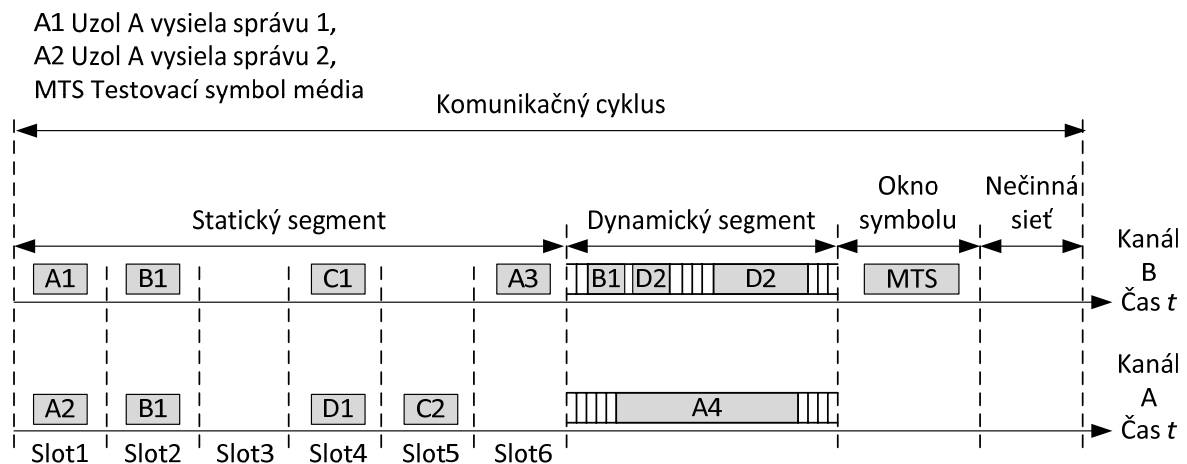
Vzhľadom na využívané frekvencie, ktoré môžu byť až desať násobne vyššie ako v prípade CAN (bude vysvetlené neskôr), je pri návrhu FlexRay sietí nevyhnutné vo všetkých topológiach voliť parametre ako dĺžka vedení a hodnota ukončovacích odporov tak, aby sa vznik skreslenia udržal na prípustnej miere.

2.1.3 Prístup na zbernicu a časovanie

Pre dodržanie determinizmu – správnej časovej následnosti, t. j. garancie maximálnej doby prenosu správy, je pri komunikácii na FlexRay zbernici využívané časové riadenie cyklov s konštantnou dĺžkou ich trvania. Tiež sa to niekedy označuje ako časovo multiplexovaná zbernica. V každom cykle je na začiatku zaradený statický segment, ktorý sa delí na časové úseky – „time slot“-y rovnakej dĺžky (obr. 10). Každý time slot je trvalo priradený maximálne jednému uzlu - „nod“, ktorý môže v tomto čase vysielat’.

Za statickým segmentom nasleduje dynamický segment, v ktorom je prístup ku zbernici regulovaný podľa priority správ. Rozdelenie na statický a dynamický segment je voľne konfigurovateľné, ale nesmie sa meniť v priebehu činnosti. To isté platí aj pre dĺžku time slotov, ktoré sú konfigurovateľné, ale tiež sa nesmú meniť v priebehu vykonávania činnosti.

„Okno symbolov“ (Symbol Window) môže byť nepovinne definované ako tretí prvok v cykle. Môže byť využité na prenos jedného symbolu. Symboly sú určené na aktiváciu siete a testovanie funkcionality.



Obr. 10. Časové radenie FlexRay

2.1.4 Synchronizácia

Pre každý uzol siete je potrebný jeho vlastný generátor hodín, aby mohol stanoviť čas vysielania a trvanie bitov. Interné generátory hodín jednotlivých uzlov môžu mať vzájomné odchýlky v dôsledku teploty kolísania napätia a výrobných tolerancií. V zbernícových systémoch akým je FlexRay, v ktorých je prístup na zbernicu riadený v závislosti od time slotov je potrebné zabezpečiť, aby sa odchýlky hodín pomocou pravidelných korekcií, udržiavali v prípustných medziach. Z toho dôvodu, niektoré uzly siete plnia funkciu generátorov hodín, kým ostatné pravidelne synchronizujú svoje interné hodiny. Procedúra prispôbuje aj čas nula (ofset) hodín aj ich rýchlosť. Môže prebiehať aj keď jednotlivé individuálne uzly zlyhajú. Pre zabezpečenie procedúry synchronizácie každý cyklus končí

krátkou fázou NIT (Network Idle Time – čas nečinnosti siete), kedy môže posúvaný bod nula cyklu.

Vďaka tejto procedúre môže byť udržiavaný tzv. „globálny čas“ vo všetkých uzloch. Ten čas je udávaný tzv. „makrotikmi“. Synchronizačný mechanizmus spôsobuje, že dĺžka jedného makrotiku môže byť v priemere rovnaká vo všetkých uzloch.

Keď sa sieť zapne najskôr sa musí zabezpečiť na časti všetkých uzlov spoločný počiatok času. Na splnenie tejto požiadavky sa spustí štartovací proces, ktorý trvá krátko. Rovnako, Uzol, ktorý sa sám má synchronizovať na bežiacu sieť potrebuje zohľadniť určitý časový úsek.

2.1.5 Ustanovenia v dynamickom segmente

V dynamickom segmente môžu byť správam pridelené rozličné priority. Preto nemožno garantovať trvanie prenosu správy. Priorita je stanovená rámcom ID, ktorý môže byť v sieti alokovaný iba raz. Správy sú vysielané postupne podľa ich rámca ID. Z toho dôvodu ma každý uzol čítač (slot ID), ktorého hodnota narastá keď sa prijme správa. Ak slot ID predpokladá hodnotu rámca ID pripravenej správy v danom uzle tak sa vyšle. Ak dĺžka dynamického segmentu nie je postačujúca pre všetky správy, proces prenosu sa musí posunúť do neskoršieho cyklu.

Rámce v dynamickom segmente môžu dosahovať rozličnú dĺžku. Limity dynamických slotov sú v oboch kanáloch navzájom nezávislé. Čiže v kanáloch v rovnakom čase sa môžu vyskytnúť správy s odlišnými ID slotmi správ.

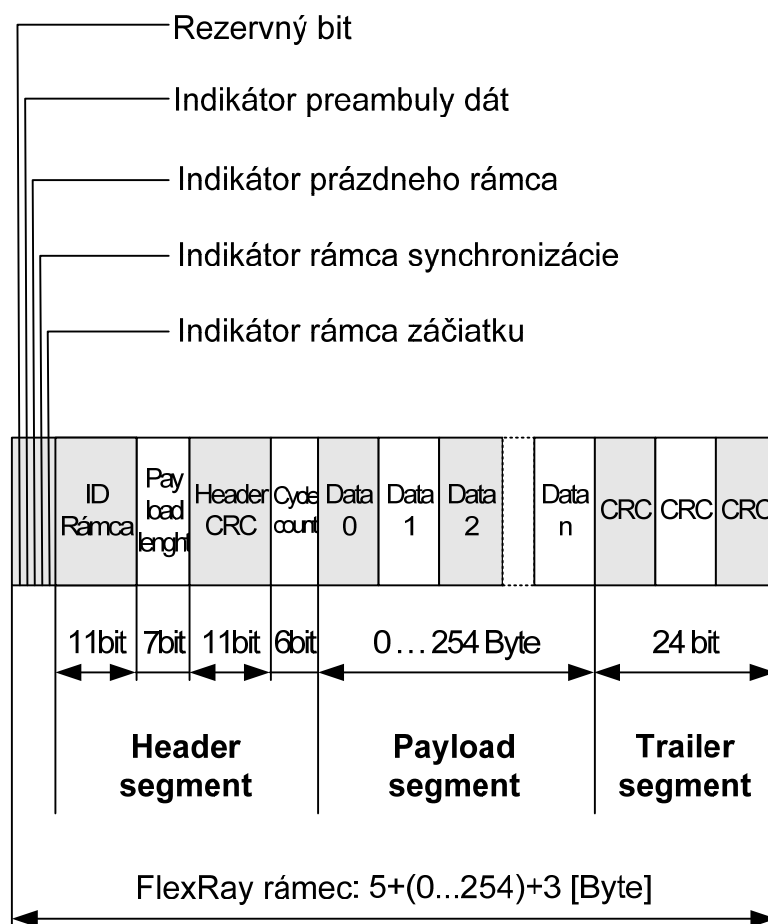
2.1.6 Dátový rámec

FlexRay využíva aj v statickom aj v dynamickom segmente rovnaký formát rámca, ktorý je možné rozdeliť do troch sekcií, a to „Header“ (hlavička), „Payload“ (samotné dáta) a Trailer“ (prídavok), ako je to zrejme z obr. 11.

2.1.6.1 Header

Header alebo hlavička pozostáva z:

- rezervného bitu pre zmeny budúceho protokolu,
- indikátora preambuly dát, ktorý identifikuje, či dáta obsahujú vektor manažmentu siete,
- indikátor prázdneho rámca, ktorý identifikuje, že sa dáta od posledného cyklu nezmenili,
- indikátor rámca synchronizácie, ktorý signalizuje, že tento rámec bude použitý na synchronizáciu systému,
- indikátor rámca začiatku, ktorý identifikuje, že tento rámec je použitý na fázu štartovania siete,
- ID rámca: odpovedá číslu slotu, v ktorom sa vysielala rámec,
- dĺžka dát (Payload length) ktorá vyjadruje objem užívateľských dát. Pre všetky sloty statického segmentu dané pole obsahuje vždy rovnakú hodnotu. Rámce v dynamickom segmente môžu mať odlišné dĺžky.
- Kontrolný súčet hlavičky (Header CRC), zabezpečuje prídavnú ochranu tejto časti rámca pretože je zvlášť citlivá na chybovosť bitov,
- počítadlo cyklov (Cycle count), v tomto poli je vysielané číslo cyklu, počas ktorého sieťový uzol vysielala do siete.



Obr. 11. Dátový rámec FlexRay

2.1.6.2 Uživateľské dáta - Payload

Uživateľské dáta, ktoré sú ďalej spracovávané hosťiteľmi sú prenášané v časti, ktorá je označovaná „Payload segment“. Ako voľba, však môžu byť pre správu siete prenášané dáta, označované ako vektor manažmentu (Network Management Vector) alebo tiež sa môžu prenášať 16 bitové ID správy.

Uživateľské dáta majú maximálnu dĺžku 254 byte-ov a sú prenášané v 2 byte-ových slovách.

2.1.6.3 Prídavok -Trailer

Prídavok resp. „Trailer“ obsahuje 24 bitový kontrolný súčet (Frame CRC), ktorý sa vzťahuje vlastnému rámcu.

2.1.7 Generovanie bitového toku rámca

Predtým, kým uzol môže vysielať rámec s dátami pre hosťiteľov, sa rámec musí skonvertovať na tok bitov - „bitstream“. Z toho dôvodu je rámec najskôr rozčlenený na byte-y. Na začiatok rámca sa vloží postupnosť začiatku prenosu (TSS – Transmission Start Sequence) a po tej nasleduje postupnosť začiatku rámca (FSS – Frame Start Sequence). Potom sa z byte-ov rámca generuje rozšírená postupnosť byte-ov, pričom sa pred každý byte rámca vloží postupnosť začiatku byte-u (BSS – Byte Start Sequence).

Na ukončenie toku bitov sa nakoniec pridá postupnosť konca rámca (FES – Frame End Sequence).

V prípade, že rámec je dynamickým segmentom, je možné pripojiť k toku bitov ďalšiu postupnosť dynamického prídavku (DTS – Dynamic Trailing Sequence), ktorá zamedzuje, aby ďalší uzol začal predčasne vysielat' do kanála.

2.1.8 Pracovné stavy

FlexRay môže byť nastavená do stavu, v ktorom uzly potrebujú iba minimálnu energiu a v ktorom sú všetky operácie kódovania a dekódovania zastavené, ale môžu byť signálom na zbernici prebudené. V tomto stave je radič zbernice ešte schopný detegovať špeciálne signály a potom tiež aktivovať jeho hostiteľa príslušným signálom. Každý uzol je schopný vysielat' signál prebúdzania.

2.2 LIN

Zbernica LIN (Local Interconnect Network) bola navrhnutá aby pokrývala požiadavky komunikácie systémov triedy A (tab. 1) s najefektívnejším hardvérom uzla z hľadiska nákladov. Typickými aplikáciami sú:

- modul dverí so zámkom,
- jednotky pohonu okien,
- nastavovanie spätných zrkadiel v dverách,
- systém klimatizácie (prenos signálov z riadiaceho prvku, aktivácia ventilátora).

Aktuálna špecifikácia LIN môže byť dostupná na web stránkach konzorcia LIN.

Dôležitými charakteristikami zbernice LIN sú:

- koncepcia jednej „master“ a viacerých „slave“ jednotiek,
- nízka náklady na hardvér kvôli prenosu dát prostredníctvom jednovodičových netienených káblov,
- autosynchronizácia jednotiek „slave“ bez nutnosti kryštálových oscilátorov,
- komunikácia formou veľmi krátkych správ,
- prenosová rýchlosť max. 20 kbit/s,
- dĺžka zbernice až 40 m a až 16 uzlov.

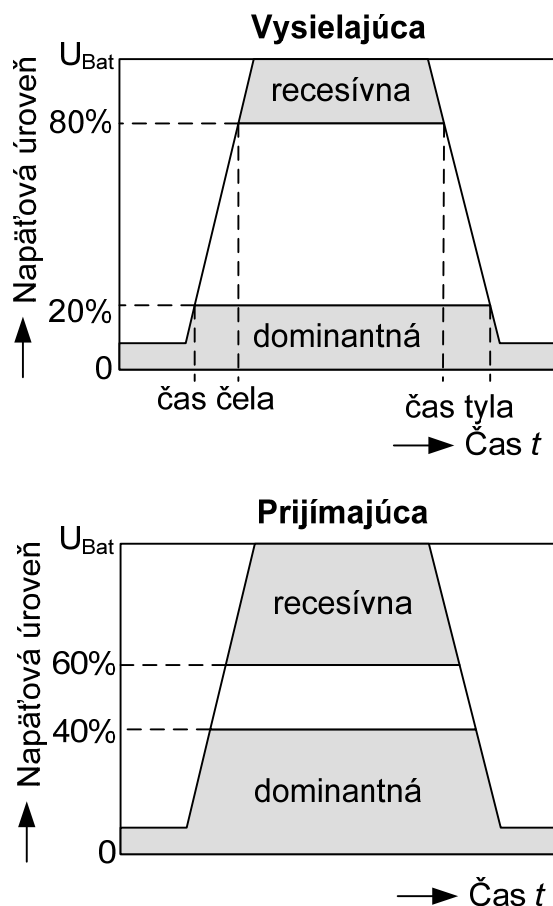
2.2.1 Systém prenosu

LIN zbernica je navrhnutá ako netienený jednovodičový kábel. Na zbernici sa môžu predpokladať dva logické stavy. Dominantnej úrovni zodpovedá napätie približne 0V (zem) a reprezentuje logickú „0“. Recesívnej úrovni zodpovedá napätie batérie U_{bat} a reprezentuje logický stav „1“.

Úrovne môžu mať odchýlky, pretože existujú variácie návrhu obvodov. Definovanie tolerancií pri vysielaní a prijímaní v prípade prenosu recesívnych a dominantných úrovní zaisťuje stabilný prenos dát. Tolerančné pásma sú širšie na strane prijímača (obr. 12), a tak môžu byť prijímané správne signály napriek rušivému žiareniu.

Prenosová rýchlosť zbernice LIN je obmedzená na 20 kbit/s. Je to kompromis medzi požiadavkou na veľkú strmosť hrán, aby sa jednotky „slave“ jednoducho synchronizovali, na jednej strane a malú strmosť hrán kvôli EMC, na strane druhej. Odporúčané prenosové rýchlosti sú 2400, 9600 a 19200 bit/s. Najnižšia dovolená prenosová rýchlosť je 1 kbit/s.

Maximálny počet uzlov nie je v špecifikácii LIN dohodnutý. Teoreticky je limitovaný počtom identifikátorov správ súvisiacich s ich obsahom. Kapacity uzlov, vedení a strmosti hrán limitujú kombináciu dĺžky a počtu uzlov LIN siete. Odporúčaný maximálny počet uzlov je 16. Používatelia zbernice sú zvyčajne usporiadané v líniovej topológii zbernice, avšak táto topológia nie je explicitne predpísaná.



Obr. 12. Napät'ové úrovně na dátových vodičoch zbernice LIN

2.2.2 Prístup na zbernicu

Na zbernicu LIN sa prístup zabezpečuje metódou prístupu „master – slave“. Sieť zriaďuje „master“, ktorý iniciuje každú správu. „Slave“ má príležitosť odpovedať. Správy sa vymieňajú medzi „master-om“ a jedným, niekoľkými alebo všetkými „slave-mi“.

Medzi uzlami „master“ a „slave“ sú počas komunikácie možné nasledujúce vzťahy:

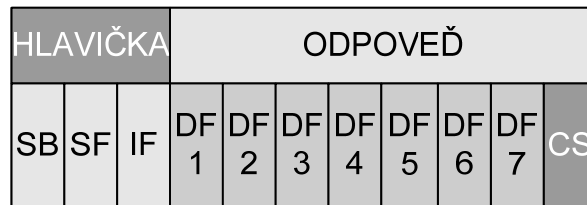
- Správa s odpoveďou uzla „slave“: „Master“ vyšle jednému alebo viacerým uzlom „slave“ správu a žiada dáta (napr. prepínacie stavy meraných hodnôt).
- Správa s inštrukciou uzla „master“: Master vydá inštrukciu jednotke „slave“ (napr. zapni servomotor).
- Správa pre inicializáciu: „Master“ iniciuje komunikáciu medzi dvomi jednotkami „slave“.

2.2.3 LIN protokol

2.2.3.1 Dátový rámec

Informácia je na LIN zbernici vložená do definovanej rámca (LIN rámec), ktorý je zrejmy z obr. 13. Správa, ktorá je iniciovaná uzlom „master“ začína hlavičkou (Header). Pole správy (odpoveď) obsahuje rozličné informácie, ktoré závisia od typu správy. Ak „master“ vysiela riadiace inštrukcie pre uzly „slave“, opisuje pole správy dáta, ktoré majú byť využité v jednotkách „slave“. V prípade žiadosti o dáta, adresovaná jednotka „slave“ vyplní pole správy dátami, požadovanými uzlom „master“.

SB Synchronizačné prerušenie
 SF Synchronizačné pole
 IF Identifikačné pole
 DF Dátové pole
 CS Kontrolný súčet



Obr. 13. Rámec LIN

2.2.3.2 Hlavička

Hlavička je vytvorená zo synchronizačného prerušenia (Synch Break), synchronizačného poľa (Synch Field) a identifikačného poľa (Ident Field).

Synchronizácia:

Synchronizácia je na začiatku každého rámca, aby sa zabezpečila konzistentnosť prenosu dát medzi uzlom „master“ a uzlami „slave“. Prvý začiatok rámca je jasne identifikovaný synchronizačným prerušením (Synch Break). To vytvára najmenej 13 po sebe nasledujúcich dominantných úrovní, za ktorými nasleduje jedna recesívna úroveň. Po synchronizačnom prerušení „master“ vyšle synchronizačné pole, ktoré pozostáva z postupnosti bitov „01010101“. Jednotky „slave“ majú možnosť adaptovať sa na časovú základňu jednotky „master“. Hodinový impulz jednotky „master“ by sa nemal líšiť od nominálnej hodnoty o viac než $\pm 0,5\%$. Hodinový impulz jednotiek „slave“ sa môže pred synchronizáciou líšiť o $\pm 15\%$. Ak sa dosiahne synchronizácia odchýlka môže na konci správy dosiahnuť $\pm 2\%$. Jednotky „slave“ tak môžu byť navrhnuté bez drahých kryštálových oscilátorov, napr. ako cenovo efektívne RC obvody.

Identifikátor:

Tretí byte hlavičky je využitý ako LIN identifikátor. Podobne ako v prípade CAN zbernice (vysvetlené v ďalšom) je použité adresovanie založené na obsahu – Identifikátor tak poskytuje informáciu o obsahu správy. Všetky uzly, ktoré sú pripojené k zbernici sa rozhodujú na základe tejto informácie, či chcú prijať a spracovať správu alebo ju ignorovať (acceptance filtering).

Šesť z ôsmich bitov identifikačného poľa tvoria identifikátor samotný a tým je možné vytvoriť 64 identifikátorov (ID). Tieto majú nasledovný význam:

- ID = 0-59: Prenos signálov.
- ID = 60: Požiadavka jednotky „master“ pre príkazy a diagnostiku.
- ID = 61: Odpoveď jednotky „slave“ na ID 60.
- ID = 62: Rezervovaný pre špecifickú komunikáciu výrobcu.
- ID = 63: Rezervovaný pre budúce rozšírenie protokolu.

Zo 64 možných správ, 32 môže obsahovať iba dva dátové byte-y, 16 štyri dátové byte-y a zvyšných 16 osem dátových byte-ov.

Posledné dva bity identifikačného poľa obsahujú dva kontrolné súčty, ktorými je identifikátor chránený proti chybám prenosu, vedúcim k nesprávnej alokácii správy.

2.2.3.3 Dátové pole

Prenos aktuálnych dát začína po hlavičke, ktorú vysiela uzol „master“. Uzly „slave“ identifikujú z vysielaného identifikátora, či sú adresované a ak je potrebné vysielajú späť odpoveď v dátovom poli.

Niekoľko signálov môže byť zoskupených do rámca. Tu má každý signál exaktne jeden generátor t.j. je vždy vložený jedným uzlom siete. Nie je dovolené počas prevádzky meniť alokáciu signálu na iný generátor ako je to možné v iných časovo riadených sieťach.

Dáta v odpovedi jednotky „slave“ sú zabezpečené kontrolným súčtom (CS).

2.2.3.4 Súbor opisu LIN

Konfigurácia zbernice LIN, t.j. špecifikácia používateľov siete, signály a rámce sú zaznamenané v súbore opisu LIN. Špecifikáciu LIN k tomuto účelu umožňuje vhodný konfiguračný jazyk.

Zo súboru opisu LIN je pomocou nástrojov automaticky generovaná séria C kódov a tzv. „header“ súborov. Tieto kódy a súbory sú využité na implementáciu „master“ a „slave“ funkcií do ECU jednotiek, pripojených ku zbernici. Súbor opisu LIN slúži na konfiguráciu celej LIN siete. Je to bežné prepojenie medzi výrobcom automobilov a dodávateľmi „master“ a „slave“ modulov.

2.2.3.5 Plánovanie správ

Tabuľka rozpisu v súbore opisu LIN určuje poradie a čas rámca v ktorom sú správy prenášané. Často potrebné informácie sú prenášané priebežne. Keď sa prešlo celou tabuľkou, „master“ začína znovu prvou správou. Postupnosť spracovania sa môže zmeniť v závislosti od prevádzkového stavu (napr. diagnostika aktívna alebo neaktívna, zapalovanie vypnuté alebo zapnuté).

Teda prenosový rámec každej správy je známy. Vlastnosť determinizmu je garantovaná skutočnosťou, že všetky prenosy sú v prípade riadenia „master-slave“ iniciované jednotkou „master“.

2.2.3.6 Manažment siete

Uzly siete LIN môžu byť uvedené do nečinného stavu (sleep mode) aby sa minimalizoval prúd v uzavretom obvode. „Sleep“ mód sa dá dosiahnuť dvomi spôsobmi. „Master“ vyšle príkaz „Go to Sleep“ s rezervovaným identifikátorom 60 alebo „slave“ automaticky prejde do „sleep“ módu ak nevysielal dáta na zbernicu v priebehu rozšíreného časového intervalu (4 sekundy). Aj „master“ aj „slave“ môžu opätovne aktivizovať sieť. Za týmto účelom sa musí vysielať aktivačný signál. Ten predstavuje dátový byte reprezentujúci číslo 128. Po prerušení 4 až 64 násobkom bitu (oddeľovač aktivácie) musia byť inicializované všetky uzly a musia byť schopné odpovedať jednotke „master“.

2.3 Ethernet, IP

Výraz Ethernet sa vzťahuje na rodinu zberníc, v ktorých sú adresovanie, formát správa a riadenie prístupu identické (ustanovenia IEEE 802). Ethernet a internet protokol (IP) boli vyvinuté pre dátovú komunikáciu medzi počítačmi alebo perifériami, ktoré lokálne oddelené a kde sa môže vyskytovať počas prevádzky vyskytnúť rekonfigurácia siete z dôvodu pripojenia nových alebo výpadku jestvujúcich používateľov. Ethernet zbernice sú identifikovateľné podľa nasledujúcich črt:

- Prenosová rýchlosť je v rozsahu 10 Mbit/s až 10Gbit/s.

- Prenos dát sa realizuje prostredníctvom rozličných médií ako koaxiálny kábel, stáčaný dvojvodičový kábel, optické vlákno alebo rádio.
- Príslušná technológia je štandardizovaná a veľmi rozšírená.
- Možné je jednoduché zaradzovanie alebo odstraňovanie uzlov.
- Časová odozva v prípade aplikácií, pracujúcich v reálnom čase nie je garantovaná.

Ethernet sa využíva v sériovej produkcii vozidiel napr. BMW série 7, kde sa využíva na zavedenie dát vozidla na konci výroby.

2.3.1 Prenosový systém

Verzie ethernetu sa líšia vzhľadom na prenosovú rýchlosť, fyzický návrh kanála a kódovanie. Koaxiálne káble, káble so stáčanými párami vodičov s jedným alebo viacerými párami, optické vlákna, rádiové cesty alebo aj káble rozvodu elektrickej siete sú špecifikované ako kanály. Podľa toho sa líši kódovanie.

Pôvodne sa ako médium v topológii zbernice požíval koaxiálny kábel. Tu sa transceivre (pár prijímač-vysielač) uzlov pripájali ku káblu buď priamo alebo pomocou T-spojok. Dnes sa v širokej miere používajú káble stáčaných párov vodičov. Prenosové rýchlosti narastali z pôvodnej 10 Mbit/s cez tzv. Fast Ethernet so 100 Mbit/s a Gigabit Ethernet s 1000 Mbit/s až po 10 Gbit/s.

2.3.2 Topológia

Rozmery siete sú limitované skutočnosťou, že čas šírenia signálu medzi dvomi uzlami ovplyvňuje proces rozhodovania. Dá sa to obchádzať rozdeľovaním do segmentov, ktoré sú prepojené špeciálnymi prvkami – rozbočovačmi (hub) a prepínačmi (switch). Hub pôsobí ako zosilňovač, ktorý nanovo upraví na ideálny tvar signál, ktorý bol skreslený interferenciou alebo disperziou prenosového média. Switch kontroluje správnosť prichádzajúcich paketov na základe kontrolného súčtu a smeruje pakety bez kolízie na iný výstup, na ktorom je adresa ich doručenia. Z toho dôvodu musí mať možnosť dočasného uchovania uloženia správy. Nevýhodou použitia takýchto prvkov je okrem nárastu nákladov aj vnášanie oneskorenia do dátového toku. Napriek tomu sa môžu pripájať uzly s odlišnými prenosovými rýchlosťami.

Dnešné siete sú navrhované tak, že každý uzol je pripojený na výstup switch-a, t.j. neexistuje priame prepojenie medzi uzlami. Samotné switch-e sú následne pripojené prostredníctvom switch-ov vyššej úrovne a takým spôsobom sa vytvára stromová štruktúra.

2.3.3 Protokol Ethernetu

2.3.3.1 Prístup na zbernicu

Uzol pre prenos kontroluje či existuje signál na zbernici. Začne vysielat' ak usúdi, že na vedení sa nič nedeje. Z dôvodu času šírenia sa signálu medzi dvomi uzlami môže nastať situácia, že dva uzly vyhodnotia nečinnosť na vedení a začnú vysielat' virtuálne simultánne. Rámce vyslané v tomto prípade sa poškodia. Uzly to vyhodnotia, prerušia vysielanie, čakajú určitý čas (odlišný v každom uzle) a pokúsia o nové vysielanie. Táto deštrukcia rámcov redukuje efektívnu prenosovú rýchlosť na tolerovateľnú mieru za predpokladu ak zaťaženie zbernice nie je príliš vysoké.

Tento proces posudzovania obmedzuje dĺžku správ a čas šírenia t.j. dosah. Neexistujú priority medzi správami. Preto nemôže byť garantované maximum trvania prenosu.

Každý uzol si zo všetkých správ prevezme tú, ktorá obsahuje jeho vlastnú adresu ako adresu určenia a spracuje ju.

2.3.3.2 Dátový rámec

Na obr. 14 je znázornená mierne zjednodušená štruktúra dátového rámca. Preambulou je periodická postupnosť bitov (101010 - 1011), ktorou sa generuje signál na synchronizáciu prijímača. Správy obsahujú adresu ich zdroja a ich určenia. Každá sieťová karta má jedinečnú adresu. Prijímajúce uzly porovnávajú adresu určenia s ich vlastnou adresou karty a akceptujú rámec ak sú zhodné. Niekoľko prijímačov môže byť adresovaných pomocou tzv. „multicast“ a „broadcast“ adres.

Preambula	Adresa určenia	Adresa zdroja	Dĺžka, typ	Dáta, vyplňujúce byte-y	CRC
8 byte	6 byte	6 byte	2 byte	46 – 1500 byte	4 byte

Obr. 14. Formát rámca Ethernet protokolu

2.4 PSI5

Rozhranie periférnych senzorov PSI5 (Peripheral Sensor Interface 5) je číslicové rozhranie, ktoré bolo publikované konzorciom PSI5 a slúži pre aplikácie senzorov v automobiloch. Môže byť zaradené do triedy A (tab. 1). PSI5 vychádza z už existujúcich rozhraní pre periférne senzory airbagov, ale bolo vytvorené ako tzv. „open“ - otvorený štandard, ktorý môže byť využívaný a implementovaný bez ďalších poplatkov. Technické charakteristiky, ktoré sú uvedené v ďalšom, vytvárajú PSI5 ako atraktívnu voľbu pre senzorové aplikácie v automobiloch, vzhľadom na nízke implementačné náklady a nízke ďalšie náklady v porovnaní s analógovým prepojením senzorov.

2.4.1 Prenosový systém

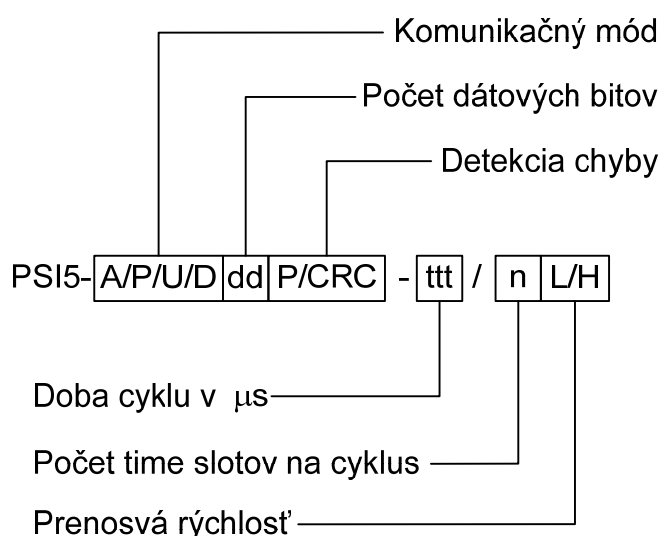
PSI5 je dvojvodičové prúdové rozhranie, v ktorom sú tie isté vodiče použité na napájanie senzorov a prenos dát v Manchester kóde. K tomuto účelu „master“ zbernice v ECU moduluje napätie pre senzory. Prenos dát zo senzora sa vykonáva prúdovou moduláciou napájacieho vedenia. Tým sa dosahuje vysoká EMC robustnosť a dosahuje sa nízke elektromagnetické vyžarovanie. Je možná podpora širokého rozsahu napájacích prúdov senzorov.

Rozlične prevádzkové módy resp. režimy PSI5 definujú topológiu a parametre komunikácie medzi ECU a senzormi (obr. 15):

- Komunikačné módy: Pre spojenie jedným smerom typu bod – bod môže byť použitý asynchrónny mód. V troch synchronných módoch zbernice (paralelný, univerzálny alebo „daisy chain“ – sériovo zreťazený) môže niekoľko senzorov komunikovať s „master“ uzlom obojsmerne na základe riadenia času, využitím TDMA procesu.
- Dĺžka dátového slova: PSI5 podporuje 8, 10, 16, 20 alebo 24 bitovú dĺžku dátového slova.
- Detekcia chyby: To sa môže realizovať buď paritným bitom s párnou paritou alebo kontrolným súčtom s tromi CRC bitmi.
- Doba cyklu – špecifikovaná v μ s.
- Počet časových okien – „time slotov“ na cyklus.

- Prenosová rýchlosť je ako štandard stanovená na 125 kbit/s alebo voliteľná 189 kbit/s.

A Asynchrónny mód,
 P Paralelný, synchronný mód,
 U Univerzálna kabeláž,
 D Daisy-chain kabeláž,
 P Paritný bit,
 CRC Cyclic Redundancy Check-
 zabezpečenie chýb,
 L Nízka prenosová rýchlosť,
 H Vysoká prenosová rýchlosť.



Obr. 15. Označovanie prevádzkových módov PSI5

Napríklad pracovný mód „PSI5-P10P-500/3L“ znamená paralelný synchronný mód s 10 bitmi na dátové slovo a paritným bitom na detekciu chyby. Dáta sú prenášané každých 500 μs s tromi „time slotmi“ na cyklus a s nízkou prenosovou rýchlosťou. Počas komunikácie od senzora k ECU je nízka úroveň reprezentovaná normálnym (nie oscilujúcim) prúdovým vstupom senzora. Vysoká úroveň sa generuje zvýšeným odberom prúdu (typicky 26 mA) senzora. Táto modulácia prúdu je detegovaná prijímačom v ECU.

Každý dátový paket PSI5 pozostáva z N bitov, v ktorých sú zahrnuté v každom prípade dva štart bity, jeden paritný bit (alebo tri CRC bity) a N-3 (alebo N-5) dátové bity. Dátové bity sa vysielajú s najmenším významovým bitom napred. Detekcia paritným bitom sa odporúča pre osem alebo desať bitov, tri CRC bity pre dlhšie dátové slová.

V PSI5 správe majú dáta a rozsahy hodnôt rozdielne významy. Jeden rozsah slúži na vysielanie výstupného signálu senzora (~94%), jeden rozsah na stav a chybové správy (~3%) a jeden rozsah na inicializáciu dát (~3%).

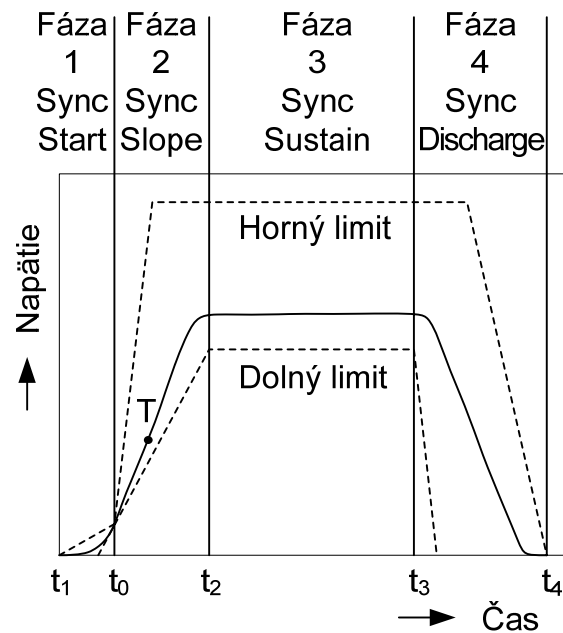
Po každom štarte alebo resete poklesom napätia senzor vykoná vnútornú inicializáciu, po ktorej je senzor vo vykonávacom móde.

Zatiaľ čo je komunikácia senzora s ECU realizovaná prúdovými signálmi, moduláciou napätia napájacieho vedenia sa realizuje komunikácia ECU so senzormi. Logická jednotka je reprezentovaná synchronizačným signálom a logická nula je reprezentovaná absenciou

očakávaného synchronizačného signálu v určenom „time slot“. Synchronizačný signál pozostáva zo štyroch napät'ových fáz (obr. 16):

- Sync Start – začiatok synchronizácie (nom. $3 \mu\text{s} < 0,5 \text{ V}$),
- Sync Slope – stúpanie (nom. $7 \mu\text{s} > 0,5 \text{ V}$),
- Sync Sustain – udržiavanie (nom. $9 \mu\text{s} > 3,5 \text{ V}$),
- Sync Discharge – pokles (nom. $19 \mu\text{s} < 3,5 \text{ V}$).

T Bod spúšťania



Obr. 16. Synchronizačný signál PSI5

2.5 MOST zbernica

Zbernica MOST (Media Oriented Systems Transport – Transportné systémy orientované na multimédia) bola špeciálne vyvinutá na sieťové prepojenie multimediálnych aplikácií v automobiloch („Infotainment“ zbernica). Popri klasických funkciách zábavy ako sú rádiov prijímače a CD prehrávače, infotainment poskytuje funkcie videa (DVD a TV), navigácie a prístup k mobilným komunikáciám a informáciám. Most zbernica podporuje logickú sieť až 64 zariadení a ponúka fixnú aj rezervovanú prenosovú šírku pásma. MOST definuje , protokol, hardvér, softvér a systémové vrstvy. MOST je vyvinutá a štandardizovaná v MOST Cooperation za účasti automobilových výrobcov a dodávateľov. S prenosovou rýchlosťou viac ako 10 Mbit/s sa MOST radí k systémom triedy D.

MOST pre prenos dát podporuje nasledovné prenosové kanály:

- Riadiaci (Control) kanál na transport riadiacich povelov.
- Multimediálny kanál (synchronný kanál) na prenos dát audia a vide.
- Kontajnerový kanál (asynchronný kanál) na prenos napr. konfiguračných dát pre navigačný systém a pre aktualizáciu softvéru riadiacich jednotiek.

Prenos multimediálnych dát, či už videa alebo audia, vyžaduje vysoké prenosové rýchlosti a tiež synchronizáciu prenosu dát medzi zdrojom a koncovým zariadením alebo aj medzi viacerými koncovými zariadeniami.

2.5.1 Prenosový systém

2.5.1.1 Fyzická vrstva

Štandard MOST špecifikuje ako optickú tak aj elektrickú technológiu fyzickej vrstvy (prenosová vrstva). Optická prenosová vrstva je rozšírená a v súčasnosti používa káble optických vlákien (Polymer Optical Fibers, POF), vyrobených z polymetylovaného metakrylátu ako prenosového média. Tieto majú hrúbku jadra 1 mm a sú používané v kombinácii s LED a kremíkovými fotodiódami ako prijímačmi.

Význačným rysom MOST 50 je jej vhodnosť pre elektrický prenos dát. To dovoľuje prenos dát prostredníctvom káblov netienených stáčaných párov medených vodičov (UTP, Unshielded Twisted Pairs). Zatiaľ čo v Európe sa mnoho rokov pokračovalo vo vývoji technológie MOST 25 a zavedená bola na kórejskom trhu, japonský trh uprednostňuje MOST 50, čo je druhá generácia multimediálneho štandardu.

Identifikačné číslo, napr. v prípade MOST 25 znamená prenosovú rýchlosť približne 25 Mbit/s. Exaktná prenosová rýchlosť závisí od vzorkovacej frekvencie, ktorú systém využíva. Pri vzorkovacej frekvencii 44,1 kHz je rámec MOST prenášaný 44100 krát za sekundu: dĺžka rámca 512 bitov vedie k prenosovej rýchlosti 22,58 Mbit/s. Pri MOST 50 tá istá vzorkovacia frekvencia vedie k dvojnásobnej prenosovej rýchlosti ak je rámec dlhý 1024 bitov. V súčasnosti sú dostupné aj vyššie prenosové rýchlosti (MOST 150 – 150 Mbit/s).

2.5.1.2 Špeciálne vlastnosti MOST 150

MOST 150 okrem vyššej prenosovej rýchlosti (150 Mbit/s) má izochrónny transportný mechanizmus na efektívny prenos komprimovaných dát HD videa. Transportné toky MPEG (Moving Picture Expert Group) sú tu prenášané priamo. Pomocou MPEG-4 kodeka je možné dosiahnuť rozlíšenie 1080p (1080 riadkov obrazu), ktoré napr. poskytujú BluRay prehrávače. MOST 150 poskytuje Ethernet kanál pre účinný prenos IP paketových dát.

Na rozdiel od MAMAC protokolu (MOST Asynchronous Medium Access Control), používaného v prípade MOST 25, Ethernetový kanál je schopný prenášať Ethernetové rámce. Ethernetový kanál prenáša nemodifikované dátové bloky, čo znamená, že softvérové balíky a aplikácie z oblasti spotrebnej elektroniky a IT môžu byť v automobile integrované bez problémov s oveľa kratším inovačným cyklom. TCP/IP balíky alebo protokoly, ktoré používajú TCP/IP (TCP – Transmission Control Protocol) môžu tak prostredníctvom MOST 150 komunikovať bez zmien.

MOST radič rozhrania siete (NIC – Network Interface Controller) je hardvérový radič, ktorý je zodpovedný za riadenie fyzickej vrstvy a implementuje dôležité mechanizmy prenosu.

2.5.2 Protokol

2.5.2.1 Prenos dát

Prenos dát na zbernici MOST je organizovaný v rámcoch, ktoré sú vytvárané podľa časovania jednotky „master“ s konštantnou prenosovou rýchlosťou a prechádzajú zariadeniami usporiadanými do kruhu.

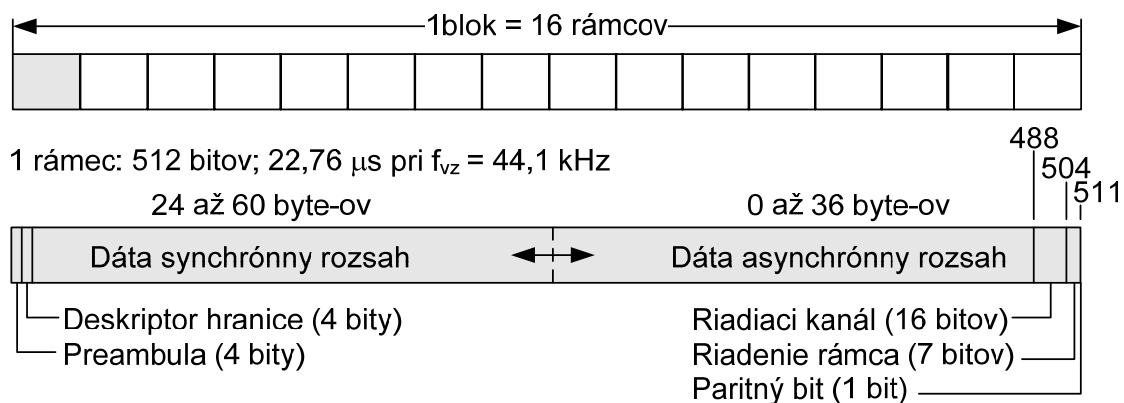
2.5.2.2 Dátové rámce

Master, ktorý generuje časovanie vytvára dátové rámce s hodinovou frekvenciou 44,1 kHz, zriedkavo tiež 48kHz. Rozmer dátových rámcov tak určuje rýchlosť zbernice MOST systému. V prípade MOST 25 je rozmer dátového rámca 512 bitov (obr. 17). Synchronná a asynchronná oblasť spoločne využíva 60 byte-ov dátového rámca. Delenie medzi synchronne a asynchronne kanály je určené hodnotou deskriptora hranice s rozlíšením

4 byte-y. Synchronná oblasť musí mať najmenej 24 byte-ov (6 stereo kanálov). To znamená, že pre synchronnú oblasť je vyhradených 24 až 60 byte-ov a pre asynchronnú oblasť je vyhradených 0 až 36 byte-ov. Pre synchronizáciu sa používa preambula a pre zabezpečenie chýb sa používa paritný bit.

2.5.2.3 Prenos riadiacich správ

Na signalizovanie stavu zariadení a pre správy potrebné na administráciu systému sa využíva riadiaci kanál. Takže riadiaci kanál nezaberá príliš veľkú časť šírky pásma na snímku a je distribuovaný do 16 rámcov zoskupených do bloku. Každá snímka transportuje dva byte-y kanála (obr. 17). Aby sa zaistilo rozpoznanie začiatku bloku, preambula prvého rámcu bloku má špeciálnu postupnosť bitov. V prípade MOST 25 má riadiaci kanál celkovú šírku pásma 705,6 kbit/s.



Obr. 17. Štruktúra MOST rámcu

2.5.2.4 Prenos multimediálnych dát

Synchronne kanály sa používajú pre komunikáciu audio a video dát v reálnom čase, pričom je výmena dát riadená prostredníctvom príslušných riadiacich príkazov v riadiacom kanáli. Synchronnému kanálu môže byť pridelená určitá šírka pásma, ktorej najmenšie delenie je jeden byte dátového rámcu. Stereo audio kanál s rozlíšením 16 bitov vyžaduje napr. 4 byte-y. V prípade MOST 25 v závislosti od hodnoty deskriptora hranice, ktorého maximálna hodnota je 60 byte-ov pre synchronný kanál, odpovedá 15 stereo audio kanálov.

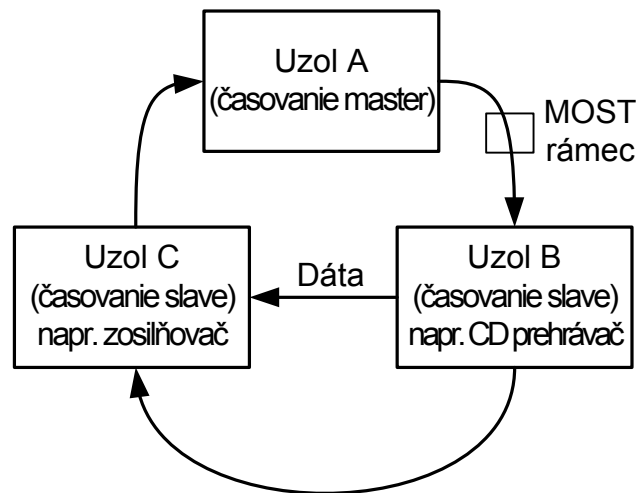
2.5.2.5 Prenos paketových dát

Paketovo orientované dáta sú prenášané na asynchrónnom kanále. Preto je vhodný na prenos informácie, ktorá nemá konštantnú prenosovú rýchlosť ale potrebuje vysoké prenosové rýchlosti v krátkom intervale. Príkladmi sú prenos informácií o stope MP3 prehrávačov alebo aktualizácia softvéru.

V prípade MOST 25 má asynchrónny kanál celkovú šírku pásma až 12,7 Mbit/s a aktuálne podporuje dva módy: pomalší 48 byte-ový, v ktorom je na prenos sieťových dát dostupných 48 byte-ov pre každý balík a 1014 byteový mód, ktorý je zložitejší na implementáciu. Aby sa zaistil spoľahlivý prenos a riadenie toku pre typicky veľké objemy dát asynchrónneho kanála, zvyčajne je zaradený ďalší transportný protokol (DLP – Data Link Protocol). Ten je implementovaný vo vrstve radiča na vyššej úrovni. Buď je to MHP (MOST High Protocol), ktorý bol špeciálne vyvinutý pre MOST alebo bežný TCP/IP ktorý je situovaný na odpovedajúcu adaptačnú vrstvu a nazýva MOST Asynchronous Medium Access Control (MAMAC).

2.5.2.6 Topológia

Zbernica MOST je organizovaná v kruhovej resp. prstencovej štruktúre (obr. 18). Je to systém s tokom dát bod – multibod (point-to-multipoint) t.j. tok dát je nepretržite vysielaný (streaming) z jedného zdroja do mnohých koncových zariadení a všetky zariadenia zdieľajú spoločný systém frekvencií, ktorý si odvodzujú z dátového toku. Zariadenia sú tak vo fáze a môžu prenášať všetky dáta synchronne. To vytvára mechanizmy pre odbúravanie potreby vyrovnávacích pamätí a spracovanie signálu. Stanovené zariadenie pracuje ako „master“ hodín a generuje dátové rámce, využívané na prenos dát a nato aby sa ostatné zariadenia synchronizovali.



Obr. 18. Kruhová štruktúra zbernice MOST

2.5.2.7 Adresovanie

Na zbernici MOST sú zariadenia adresované pomocou 16 bitovej adresy. Dostupné sú rôzne typy adresovania: logické, fyzické a skupinové pre simultánne adresovanie definovanej skupiny riadiacich jednotiek.

2.5.2.8 Administrátorské funkcie

MOST štandard definuje mechanizmy manažmentu (siete a pripojenia „master“ jednotiek), potrebné pre činnosť MOST systému. Tieto mechanizmy sú opísané v ďalšom.

Master siete:

Master siete je implementovaný v označenom zariadení MOST systému a je zodpovedný za konfiguráciu systému. V súčasných systémoch „master“ siete zvyčajne implementovaný v čelnom paneli (t.j. ovládací panel) „infotainment“ systému. Toto zariadenie je často súčasne aj „master“ časovania. Ostatné zariadenia MOST systému sú v tom kontexte označené ako „slave“.

Master spojenia:

Master spojenia manažuje synchronne spojenia, ktoré existujú v MOST systéme v ľubovoľnom špecifickom bode v čase.

2.5.2.9 MOST aplikačná vrstva

MOST štandard na aplikačnej vrstve definuje na prenos riadiacich povelov, informácií o stave a udalosti príslušný protokol. Daný protokol dovoľuje spúšťanie určitých funkcií aplikačného

rozhrania (t.j. FBlock= funkčný blok) ktorý je poskytovaný ľubovoľným zariadením MOST systému.

Protokol pre riadenie MOST správ obsahuje nasledovné prvky riadiacej správy:

- adresa zariadenia MOST systému (DeviceID),
- identifikátor FBlock-u (FBlockID), implementovaný týmto zariadením a jeho inštanciu v MOST systéme (InstID),
- identifikátor funkcie vyžiadanej v FBlock-u (FunctionID),
- typ činnosti (OpType).

2.5.2.10 Funkčný blok

Funkčný blok (FBlock) definuje rozhranie určitej aplikácie alebo systémovej služby. Koncové zariadenia a zdroje multimediálnych dát sú priradené k FBlock-u, ktorý poskytuje príslušné funkcie pre ich administráciu. FBlock preto môže mať niekoľko zdrojov a koncových zariadení, ktoré sú označené pomocou čísla zdroja a koncového zariadenia.

FBlock má funkcie, ktoré doručujú informácie vzhľadom na číslo a typ zdroja a koncového zariadenia, ktorý ich poskytuje (SyncDataInfo, SourceInfo a SinkInfo). Navyše každý FBlock so zdrojom má funkciu „Allocate“, ktorý ju využíva na žiadosť o synchronný kanál a pripojí zdroj k tomuto kanálu. Taktiež FBlock s koncovým zariadením má funkciu „Connect“ na pripojenie tohto zariadenia k určitému synchronnému kanálu a funkciu „Disconnect“ na prerušenie tohto spojenia.

FBlock je adresovaný 8 bitovým FBlockID, ktorý špecifikuje typ FBlocku a ďalším 8 bitovým InstID.

2.5.2.11 Triedy funkcií

Aby sa štandardizoval spôsob akým sú definované funkcie, štandard MOST špecifikuje pre vlastnosti sériu tried funkcií. Tieto určujú ktoré vlastnosti funkcia má a aké operácie sú prístupné.

2.5.2.12 Aplikácie

Tak ako je potrebné pre prenos dát definovať spodné vrstvy, tak aj MOST štandard definuje rozhrania pre typické aplikácie z oblasti systémov infotainmentu v automobiloch, napr. CD menič, zosilňovač alebo rádioner.

FBlock-y, definované MOST Cooperation sú sumarizované v katalógu funkcií.

2.5.2.13 Štandardizácia

Štandard MOST je udržiavaný organizáciou MOST Cooperation, ktorá taktiež publikuje príslušné špecifikácie. Špecifikácie sú dostupné na domovskej stránke organizácie.

MOST Cooperation bola založená v roku 1998 spoločnosťami BMW, Daimler, Becker Radio a OASIS Silicon Systems so zámerom štandardizovať MOST technológiu.

3 Zbernica CAN

Ako je zrejmé z predchádzajúcich častí, automobil je možné navrhovať a analyzovať ako inteligentný systém pomocou komunikačného rozhrania a pristupovať len k číslicovo reprezentovaným dátam, no tiež ako analógový obvod na fyzickej úrovni, ako tomu bolo u automobilov predošlých generácií, kde elektronika nebola rozvinutá do súčasnej podoby.

Mnohí výrobcovia automobilov a komponentov pre automobily (Bosch, VW, Opel, Ford, BMW, Nippon Seiki... - AUTOSAR) pri vývoji vlastných riadiacich jednotiek a ich začleňovaní do celkového distribuovaného systému, v ktorom je komunikácia realizovaná prostredníctvom zberníc a sietí využívajú nástroje pre simuláciu a testovanie sietí a riadiacich jednotiek od spoločnosti Vector. K takýmto nástrojom patrí ako hardvérový (CANCardXL) tak aj softvérový produkt s označením CANoe.

V ďalšom budú stručne vysvetlené otázky spojené so zbernicou CAN do úrovne, ktorá je nevyhnutná pre možnosť samostatnej práce na diagnostickom rozhraní automobilu. Práca s nástrojmi CANCardXL aj CANoe je stručne demonštrovaná v ďalšej časti. Vniknutiu do danej problematiky napomáhajú priložené príklady aj celkové riešenia, ktoré je možné odskúšať v demo verzii balíka programov CANoe. Každému elektronickému systému v automobile prislúcha číslicová riadiaca jednotka; napr. pre zapalovanie a vstrekovanie, pre ABS alebo pre automatickú prevodovku.

Ku každej riadiacej jednotke sú priradené určité špeciálne snímače, senzory a akčné členy.

Postupy riadené jednotlivými riadiacimi jednotkami musia byť vzájomne skĺbené. Napríklad, ak má byť zmenou okamihu zapálenia zmesi v priebehu radenia zmenšený krútiaci moment. Ďalším príkladom je zmenšenie hnacieho momentu v priebehu zamedzovania preklzavania hnacích kolies pri akcelerácii, prípadne decelerácii (ASR-tzv. regulácia preklzavania pohonu). Je výhodné využívať snímače a senzory všetkých riadiacich jednotiek spoločne.

Neustále sa zvyšujúca výmena informácií medzi riadiacimi jednotkami má pre celkový systém vozidla nesmierny význam. Totiž, pri výmene informácií medzi jednotlivými riadiacimi jednotkami po samostatných vodičoch je vždy jeden vodič určený k prenosu jedného druhu informácie. S každým ďalším druhom prenášanej informácie, rastie počet potrebných vedení a počet vývodov na svorkovniciach riadiacich jednotiek. Aby elektrická a elektronická časť vozidla zostala aj napriek tomu prehľadná a nezaberala v ňom veľa miesta, je nutné nájsť a uplatniť jednoduchý systém. Jedným z nich je dátová zbernica CAN (Controller Area Network). Systém, ktorý využíva dátové zbernice, bol vyvinutý špeciálne pre použitie v motorových vozidlách.

Pri výmene informácií pomocou dátovej zbernice CAN sa všetky dáta prenášajú len po dvoch vedeniach. Na oboch bidirekcionálnych (obojsmerných) vedeniach sa prenášajú rovnaké dáta, a to nezávisle od počtu riadiacich jednotiek a množstve dopravovaných údajov. Prenos informácií pomocou dátovej zbernice CAN má význam vtedy, ak je potrebné predávať veľa informácií medzi mnohými riadiacimi jednotkami.

3.1 Princíp prenosu dát

Prenos dát pomocou dátovej zbernice CAN si môžeme predstaviť ako telefónnu konferenciu. Kým jeden účastník konferencie (napr. riadiaca jednotka 1) posiela do siete dáta – „hovorí“, statní účastníci dáta prijímajú – „počúvajú“ a prijaté dáta vyhodnocujú. Niektorý z účastníkov konferencie vyhodnotí posielané údaje ako zaujímavé a potrebné a využije ich. Ostatným účastníkom nehovorí nič, a preto zostanú pasívni. Do „telefónnej konferencie“ môžu byť zapojení dvaja alebo viacerí účastníci. Dátová zbernica CAN je druh prenosu medzi riadiacimi jednotkami. ňou sú jednotlivé riadiace jednotky spojené do uceleného systému. Pre použitie CAN v motorových vozidlách existujú v podstate tri základné oblasti:

- dátová zbernica hnacieho ústrojenstva (prepojenie napr. riadiacich jednotiek motora, ABS, automatickej prevodovky atď.),
- dátová zbernica komfortnej elektroniky (prepojenie napr. riadiacich jednotiek dverí, klimatizácie atď.) a
- dátová zbernica mobilného komunikačného systému (prepojenie napr. riadiacich jednotiek, autorádia, telefónu, navigačného systému atď.)

Ak sú v automobile viaceré zbernice, zvyčajne existuje aj centrálna riadiaca jednotka, ktorá slúži na vzájomné prepojenie tokov dát alebo informácií medzi jednotlivými zbernicami, ktoré môžu používať iné prenosové rýchlosti alebo parametre prenosu.

K prednostiam dátovej zbernice CAN patrí:

- značné zjednodušenie prepojení,
- veľmi rýchly prenos dát medzi riadiacimi jednotkami,
- úspora miesta vďaka menším rozmerom riadiacich jednotiek a menším svorkovniciam (konektorom),
- zníženie počtu porúch v dôsledku sústavnej kontroly dopravovaných dát riadiacimi jednotkami,
- ak je potrebné rozšíriť prenosový protokol o ďalšie informácie, stačí len upraviť software a
- dátová zbernica CAN je normalizovaná v celom svete; má to tu výhodu, že výmena dát je možná aj medzi riadiacimi jednotkami rôznych výrobcov.

3.2 Súčasti dátovej zbernice CAN

Hlavnými súčasťami dátovej zbernice CAN sú:

- radič,
- vysielač,
- ukončenia dátovej zbernice a
- vedenia dátovej zbernice.

S výnimkou vedení dátovej zbernice sa všetky časti nachádzajú v riadiacich jednotkách (obr. 19).

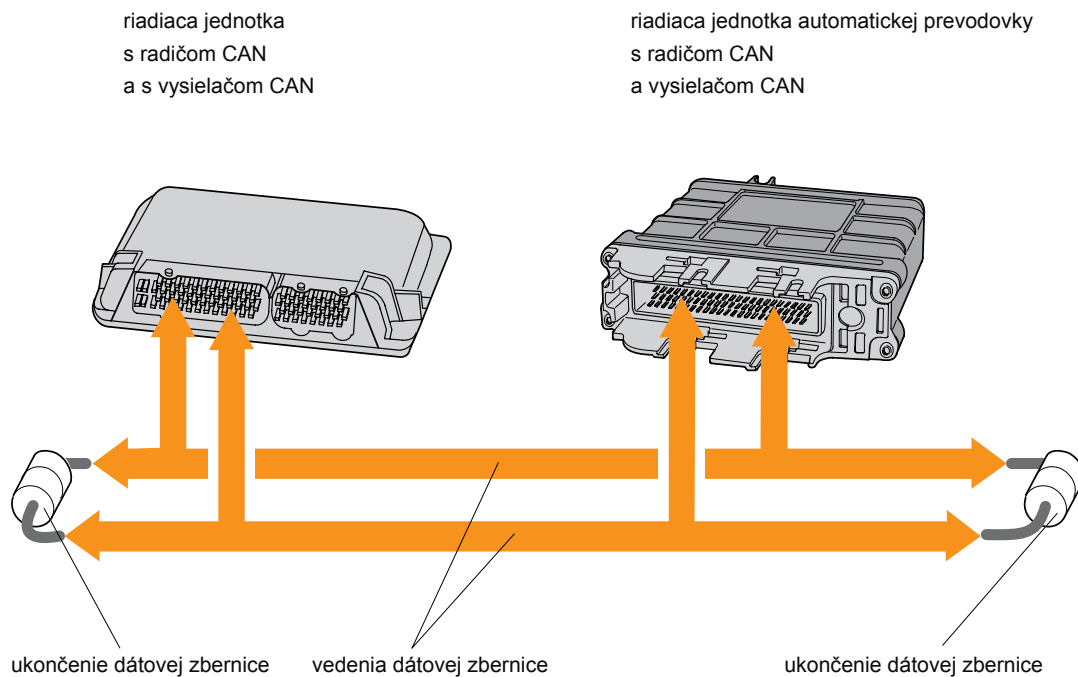
3.2.1 Úlohy jednotlivých súčastí

Radič dostáva od mikropočítača v riadiacej jednotke tie údaje, ktoré majú byť odoslané. Radič ich pripravuje a predáva ďalej do vysielača CAN. Súčasne, ale dostáva od vysielača dáta, ktoré pripravuje a predáva ďalej do mikropočítača v riadiacej jednotke.

Vysielač CAN je vysielačom (*transmitter*) aj prijímačom (*receiver*) v jednom. Mení dáta radiča CAN na elektrické signály. Podobne prijíma elektrické signály, ktoré mení na dáta pre radič.

Ukončenia dátovej zbernice predstavujú odpory (impedančné prispôsobenie), ktoré svojou impedanciou zamedzujú vzniku odrazov na vedeniach, čo spôsobuje skreslenie signálov na vedeniach a tým preberanie chybných dát.

Vedenia dátovej zbernice sú biderkcionálne a slúžia k prenosu dát.



Obr.19. Pripojenie riadiacich jednotiek k zbernici CAN s ukončovacími odporami

Pri prenose dátovou zbernicou sa príjemca neurčuje. Dáta sú do dátovej zbernice vysielané, prijímané a vyhodnocované spravidla všetkými účastníkmi.

3.2.2 Pribeh dátového prenosu

Postup prenosu dát je znázornený na obr.20.

Príprava dát

Správy (dáta) vychádzajú vždy z riadiacej jednotky. Riadiaca jednotka predáva svojmu radiču dáta, ktoré majú byť odoslané.

Odoslanie dát

Vysielač CAN dostane tieto dáta od radiče CAN, premení ich na sériové elektrické signály a pošle ich ďalej.

Prijatie dát

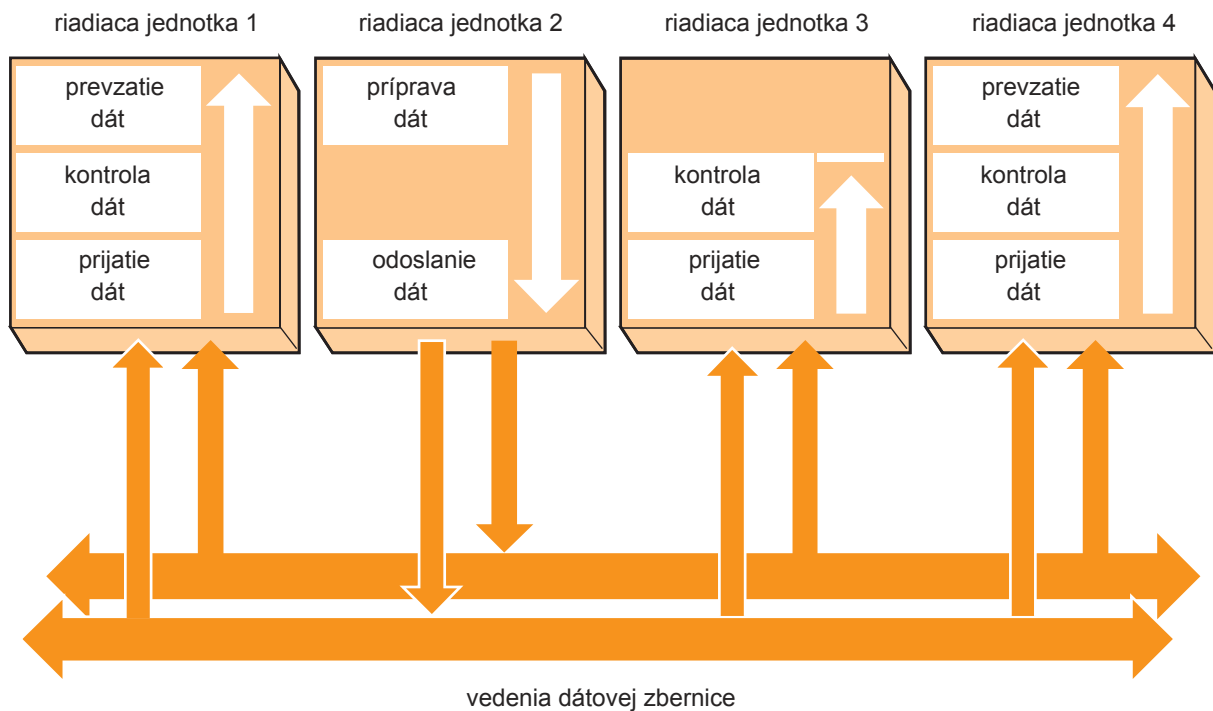
Ostatné riadiace jednotky, ktoré sú zapojené do siete dátovej zbernice CAN, odoslaná dáta prijmu.

Kontrola dát

Riadiace jednotky preverujú, či sú prijaté dáta pre ich činnosť potrebné.

Prevzatie dát

Ak sú prijaté dáta pre riadiacu jednotku potrebné, prevezme ich a ďalej spracuje. Ak sú nepotrebné, tak na ne riadiaca jednotka nereaguje.

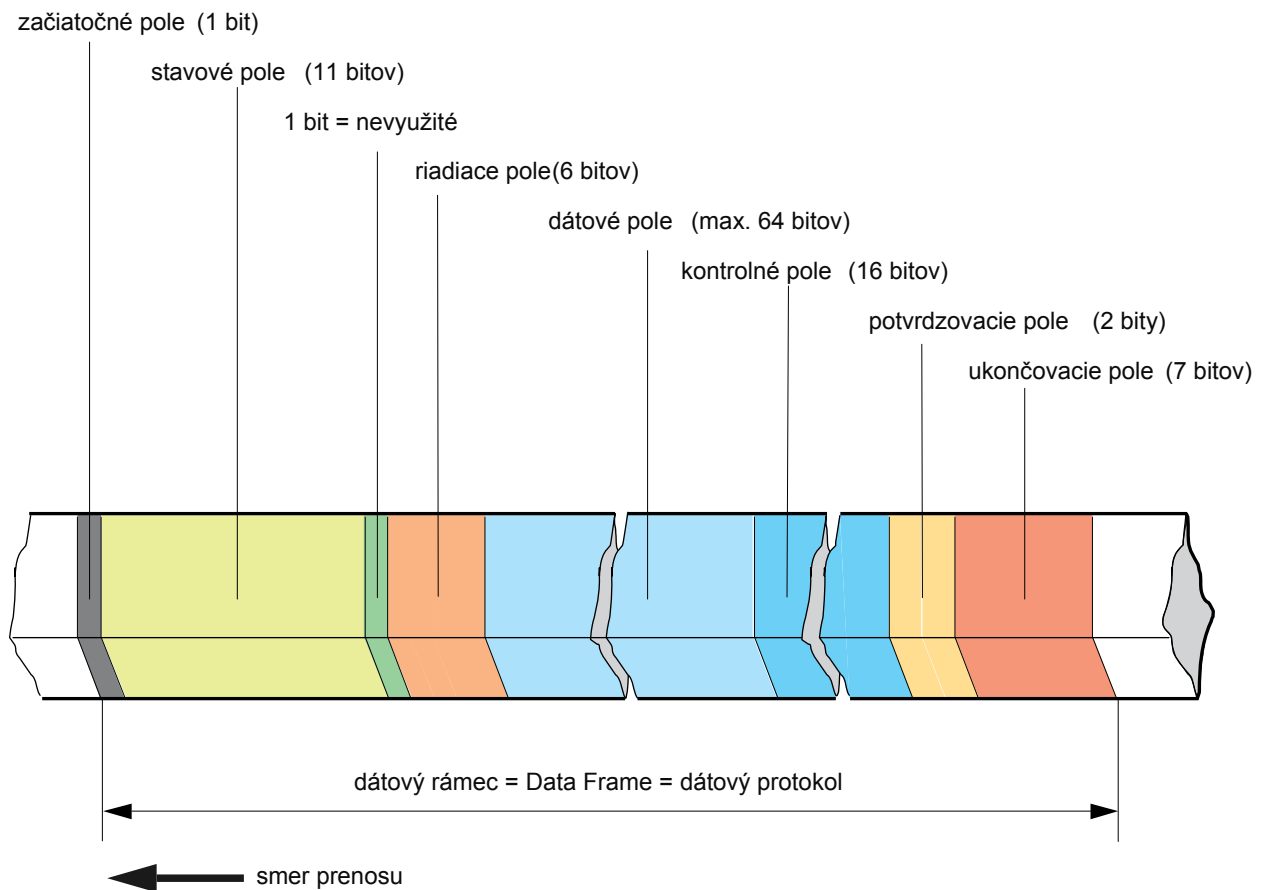


Obr. 20. Proces vysielania a prijímania dát na zbernici CAN

3.2.3 Dátový protokol

Dátová zbernica CAN prenáša vo veľmi krátkych časových intervaloch medzi riadiacimi jednotkami dátový protokol (dátový rámec) nazývaný tiež „správa“. Dátový protokol sa skladá z mnohých po sebe nasledujúcich bitov. Počet bitov dátového protokolu je závislý od dátového poľa.

Na obr. 21 je schematicky znázornený dátový protokol. Stavba dátového protokolu je na oboch vedeniach dátových zberníc rovnaká. Tento dátový protokol je vytváraný vždy podľa rovnakého dátového rámce = Data Frame. Tento rámec je tvorený **siedmimi** po sebe nasledujúcimi **poľami**.



Obr. 21. Dátový rámec CAN zbernice

Polia

Začiatkové pole (Start of Frame) - označuje začiatok dátového protokolu.

Stavové pole (Arbitration Field) - stanovuje prioritu dátového protokolu. Ak chcú dve riadiace jednotky odoslať svoj dátový protokol súčasne, má ten, ktorého priorita je vyššia, prednosť. Zároveň je v ňom označený obsah správy (napr. otáčky motora).

Riadiace pole (Control Field) - obsahuje ako kód počet informácií, ktoré sú obsiahnuté v dátovom poli. Vďaka tomu môže prijemca skontrolovať, či mu došli všetky.

Dátové pole (Data Field) - V dátovom poli sú prenášané informácie, ktoré sú dôležité pre ostatné riadiace jednotky. Obsahom informácií 0 až 64 bit (0 až 8 byte) je polom s najväčším množstvom informácií.

Kontrolné pole (CRC Field) (CRC = Cyclical Redundancy Check) – slúži na zisťovanie chýb v prenose. Jedná sa o metódu založenú na cyklickom výpočte kontrolného kódu dát pred prenosom a po prenosom.

Potvrdzovacie pole (ACK Field) (ACK = Acknowledgement) - Potvrdenie prijatia. Prijemca signalizuje objektu, ktorý správu vyslal, že dátový protokol bol správne prijatý. Ak bola zistená chyba, je to vysielaciemu objektu ihneď oznámené a ihneď dochádza k opätovnému odoslaniu správy.

Ukončovacie pole (End of Frame) - v tomto poli kontroluje vysielateľ svoj dátový protokol a potvrdí objektu, ktorý správu vyslal, či je v poriadku. Ak nie je, dôjde okamžite k prerušeniu a opakovanému začatiu prenosu. Tým je dátový prenos protokolu ukončený.

3.3 Funkcia

3.3.1 Pridelovanie dátovej zbernice CAN

Ak chce viacero riadiacich jednotiek poslať svoj dátový protokol súčasne, je potrebné rozhodnúť, ktorý bude mať prednosť (obr. 22). Prednosť bude daná protokolu s najvyššou prioritou. Tak napríklad dátový protokol z riadiacej jednotky ABS/ABS s EDS je z bezpečnostných dôvodov dôležitejší, než dátový protokol z riadiacej jednotky automatickej prevodovky.

Priorita dátového protokolu sa určí postupným vyhodnocovaním radu príslušných bitov (logická „0“ s prioritou, logická „1“ bez priority). Priorita dátového protokolu sa zaisťuje tak, že každému dátovému protokolu je podľa jeho dôležitosti priradený v stavovom poli jedenásť bitový kód. V nasledujúcej tabuľke (tab.2) je znázornená priorita troch dátových protokolov.

Tab.2

brzda	001	1010 0000
motor	010	1000 0000
prevodovka	100	0100 0000

Tri riadiace jednotky začnú súčasne posilať svoj dátový protokol. Súčasne však začnú postupne bit za bitom vyhodnocovať bity na vedení dátovej zbernice. Ak zistí riadiaca jednotka v stavovom poli bit s prioritou na mieste, kde je jej bez priority, prestane vysilať a začne prijímať.

Príklad:

Bit 1 v stavovom poli (Arbitration Field)

- riadiaca jednotka automatickej prevodovky vysila bit bez priority. Na vedení dátovej zbernice však zistí, že na nej stojí bit s prioritou. V tom okamihu stráca právo na pridelenie vedení dátovej zbernice. Prestane vysilať a začne prijímať. Bity 2 a 3 odpadnú a nebudú sa už vyhodnocovať.
- riadiaca jednotka ABS/ABS s EDS vysila bit s prioritou.
- riadiaca jednotka Motronic tiež vysila bit s prioritou.

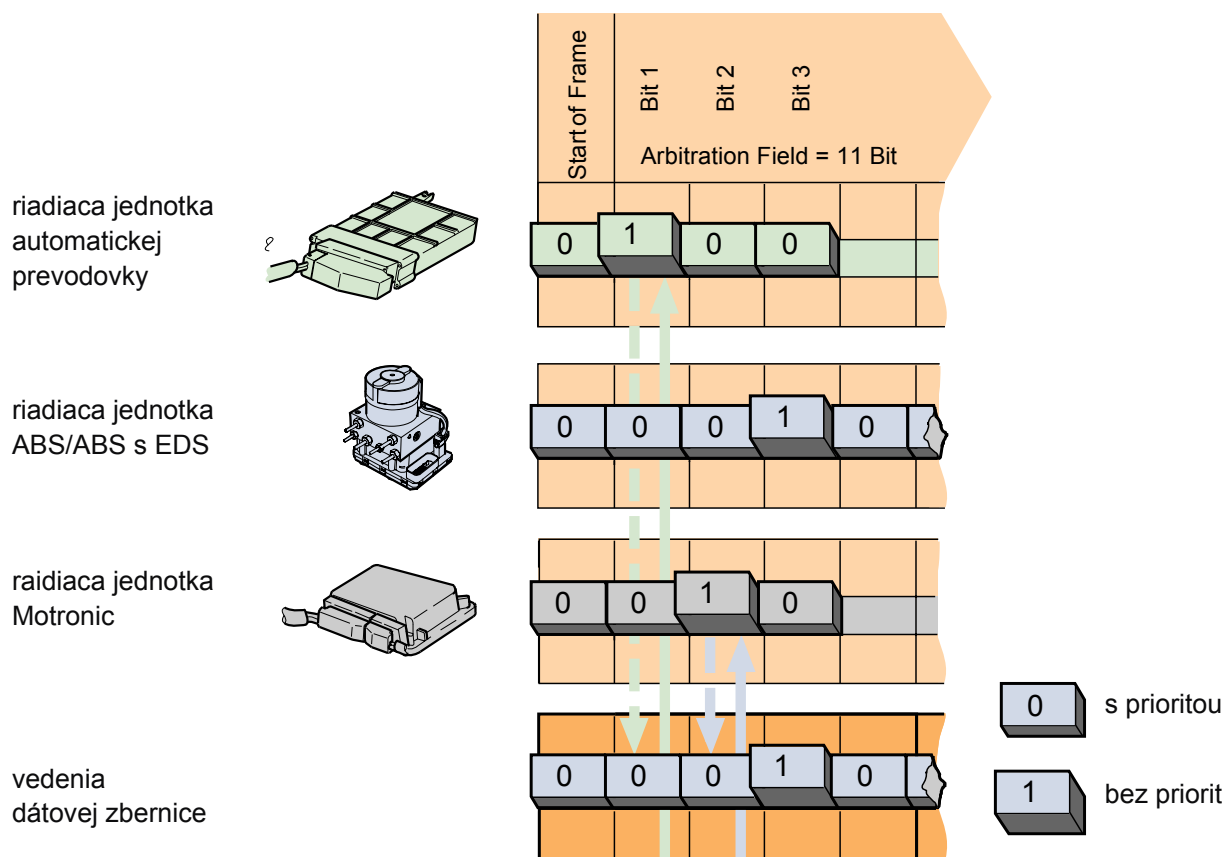
Bit 2 v stavovom poli

- riadiaca jednotka ABS/ABS s EDS vysila bit s prioritou.
- riadiaca jednotka Motronic vysila bit bez priority. Na vedení dátovej zbernice však rozpozná, že na nej stojí bit s prioritou. V tom okamihu stráca právo na pridelenie vedení dátovej zbernice. Prestane vysilať a začne prijímať. Zvyšný bit 3 odpadne a už sa nebude vyhodnocovať.

Bit 3 v stavovom poli

- riadiaca jednotka ABS/ABS s EDS mala najvyššiu prioritu a získala tak právo na pridelenie vedení dátovej zbernice. Preto vo vysielaní dátového protokolu pokračuje.

Ako náhle riadiaca jednotka ABS/ABS s EDS dokončí odosielanie dátového protokolu, pokúsia sa zvyšné riadiace jednotky poslať svoj dátový protokol znovu.

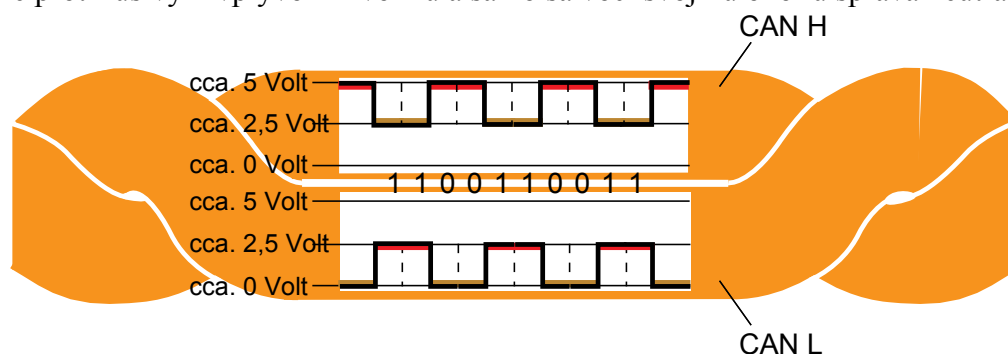


Obr. 22. Proces prideľovania zbernice CAN a vyhodnocovanie priority

3.4 Rušivé zdroje

Vo vozidle existujú súčasti, ktoré sa javia ako rušivé zdroje napr. tým, že vytvárajú iskry alebo otvárajú či uzavierajú elektrický obvod. Ďalšími rušivými zdrojmi sú popripade mobilné telefóny, krátkovlnné vysielачky – skrátka všetko, čo vytvára elektromagnetické vlny. Rušivé zdroje môžu prenos dát ovplyvniť a prenášané dáta dokonca pozmeniť.

Aby sa zabránilo prenosu rušivých vplyvov na prenášané dáta, sú dve netienené vedenia dátovej zbernice vzájomne spletené. Spletenými vedeniami je prenášaný diferenčný (rozdielový) signál, tzn. na vedeniach sa nachádza vždy opačné napätie (obr. 23). Ak je na jednom vedení dátovej zbernice napätie približne 0 V, je na druhom približne 5 V. Inak je na oboch vedeniach približne rovnaké napätie s veľkosťou asi 2,5 V. Tým je zaistené, že súčet napätí má v každom okamihu rovnakú hodnotu - je konštantný a elektromagnetický vplyv pole oboch vedení dátovej zbernice sa vzájomne vruší. Vedenie dátovej zbernice je tak chránené proti rušivým vplyvom z vonku a samo sa voči svojmu okoliu správa neutrálne.



Obr. 23. Diferenčné signály vedení zbernice CAN

4 Príklad znakov dátových zberníc automobilov

4.1 Znaky dátovej zbernice CAN hnacieho ústrojenstva (Škoda)

- Prenosové médium, dátová zbernica je tvorená dvoma vodičmi, ktorými sa prenášajú informácie.
- Aby sa zabránilo vplyvu rušivých polí, a aby vedení samo nebolo zdrojom rušenia, sú obidva vodiče dátovej zbernice spolu vzájomne spletené.
- Dátová zbernica hnacieho ústrojenstva pracuje s rýchlosťou prenosu od 500 kb/s. Tým sa radí do rýchlostného rozpätia veľkých rýchlostí (high speed 125 až 1000 kb/s. Prenos rámca trvá asi 0,25 milisekúnd. Dátová zbernica komfortnej električky pracuje s prenosovou rýchlosťou 62,5 kb/s. Z toho vyplýva, že dátovú zbernicu hnacieho ústrojenstva a dátovú zbernicu komfortnej elektroniky nie možné priamo vzájomne prepojiť.
- Podľa druhu riadiacej jednotky sa v rozmedzí 7 až 20 milisekúnd pokúša riadiaca jednotka vyslať správu.
- Poradie priorít:
 1. Riadiaca jednotka ABS/ABS s EDS ———>
 2. Riadiaca jednotka motora ———>
 3. Riadiaca jednotka automatickej prevodovky.

Poradie priorít sa stanovuje na základe hodnotenia časovej dôležitosti prepravovanej správy. Z tohto hľadiska stojí tiež aktívne zabránenie nehody v poradí priorít na prvom mieste.

Aby bolo možné posielané dáta v rámci hnacieho ústrojenstva skutočne optimálne využívať, je potrebné ich prenášať veľkými rýchlosťami. Preto sa používa vysielateľ s vysokým výkonom. Vysielané dáta sa prenášajú v časovom intervale medzi dvoma zápalmi. Tým sa umožňuje prijaté informácie využívať už pri nasledujúcom zápale.

4.2 Znaky dátovej zbernice CAN komfortného systému (Škoda)

- Prenosové médium, dátová zbernica je tvorená dvoma vodičmi, ktorými sa prenášajú informácie.
- Aby sa zabránilo vplyvu rušivých polí, a aby vedení samo nebolo zdrojom rušenia, sú obidva vodiče dátovej zbernice spolu vzájomne spletené.
- Systém dátovej zbernice komfortnej električky pracuje s prenosovými rýchlosťami od 62,5 kb/s. Patrí teda do rýchlostného rozsahu malých rýchlostí low speed 0 - 125 kb/s. Prenesenie dátového rámca trvá asi 1 milisekundu. (Pre porovnanie: dátová zbernica hnacieho ústrojenstva pracuje s rýchlosťou 500 kb/s.)
- Každá riadiaca jednotka sa pokúša každých 20 milisekúnd poslať svoje dáta.
- Poradie priorít:
 4. centrálna riadiaca jednotka
 5. riadiaca jednotka dverí vodiča
 6. riadiaca jednotka dverí spolujazdca
 7. riadiaca jednotka zadných ľavých dverí
 8. riadiaca jednotka zadných pravých dverí

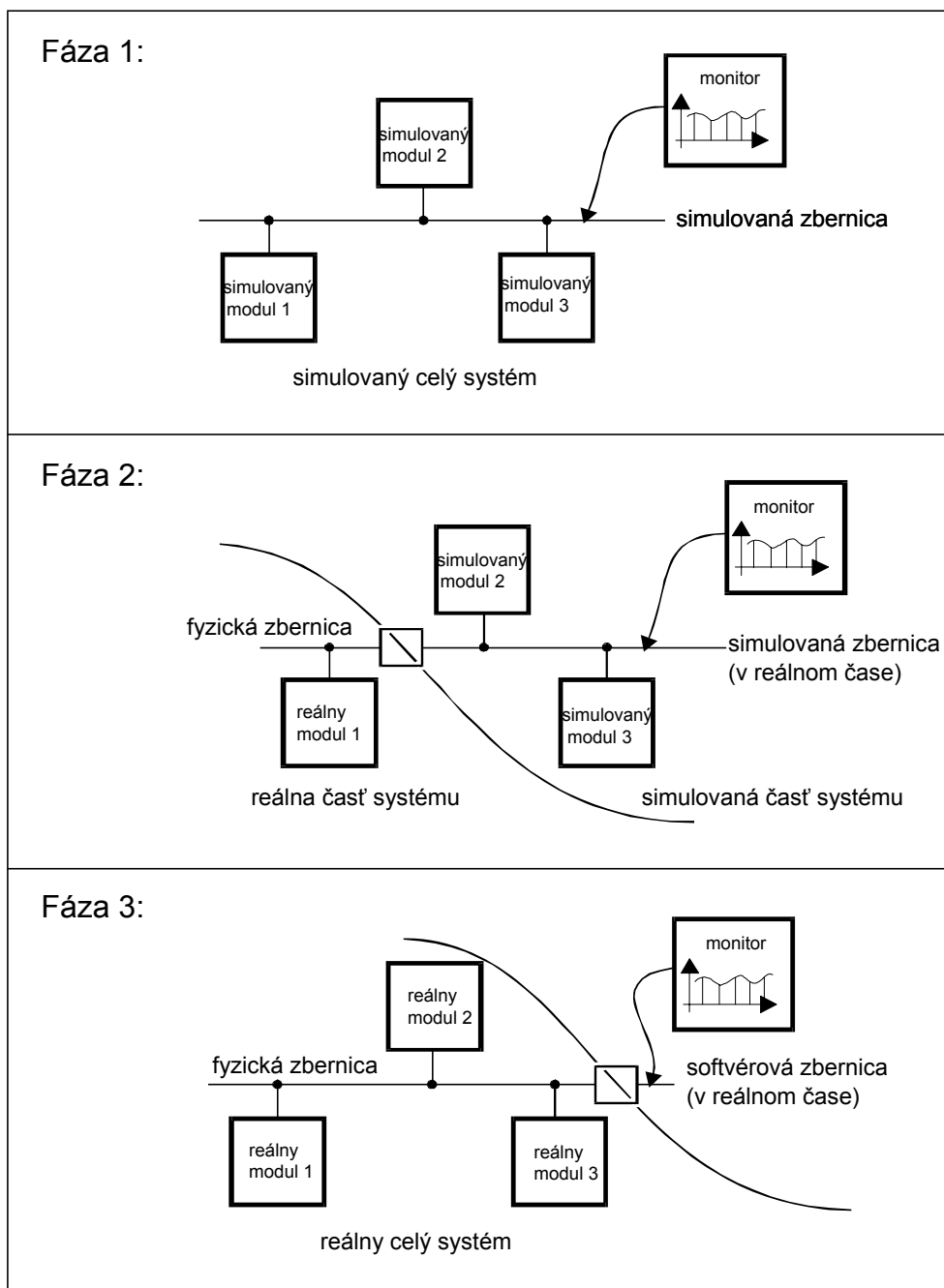
Vzhľadom k tomu, že v systéme komfortnej električky sa k prenosu dát používajú pomerne malé rýchlosti, je možné použiť vysielateľ s menším výkonom. Má to tú výhodu, že v prípade výpadku jedného z vedení dátovej zbernice, dôjde k prepnutiu na prenos iba jedným vodičom a dáta je možné prenášať ďalej.

5 CANoe

5.1 Prehľad

CANoe je univerzálne prostredie pre vývoj, testovanie a analýzu systémov CAN zbernice, ktoré robí dostupným vývojový proces všetkým jeho účastníkom. Výrobca systému má podporu pre distribúciu funkcií, kontrolu funkcií a integráciu všetkých systémov. Dodávateľ tak získa ideálne prostredie pre testovanie simulácií ostatných častí zbernice a okolia.

Vývojový proces vychádza z vývojových fáz (obr. 24), ktoré sa rozlišujú v troch vývojových stupňoch:



Obr. 24. Model vývojového procesu

Fáza 1: Analýza požiadaviek a návrh systému siete.

Najskôr sa rozdelia a načrtnú dôležité celky, medzi jednotlivé sieťové uzly tak, aby sa zabezpečila celková funkčnosť systému a vyladí sa návrh na úrovni sieťových uzlov. To zahŕňa definovanie správ a výber prenosových rýchlostí zbernice. Nakoniec musí byť špecifikované správanie individuálnych sieťových uzlov z pohľadu zbernice, napr. vo forme časov cyklov, alebo zložitejších protokolov. Potom môže byť táto informácia vyhodnotená najskôr pomocou simulačného nástroja pre získanie počiatkových odhadov zaťaženia zbernice a časov oneskorenia pre predpísanú prenosovú rýchlosť. Potom tieto špecifikácie môžu byť tiež použité pre testovanie následných fáz.

Pre presnejšie štúdium je vytvorený funkčný model celého systému. To umožňuje špecifikovanie správania sieťových uzlov s ohľadom na vstupné a výstupné premenné a správy, ktoré sú tak vysielané ako aj prijímané. Tu je zvlášť užitočný model, riadený udalosťami s procedurálnym opisom správania. Napríklad môže opisovať ako (po prijatí správy/udalosti) budú prijaté dáta ďalej spracované (procedurálne) a aký výsledok, ako riadená premenná, bude na výstupe.

Fáza 2: Implementácia komponentov so simuláciou zvyšku zbernice.

Po dokončení prvej fázy, je návrh a vývoj individuálnych sieťových uzlov zvyčajne realizovaný všetkými účastníkmi nezávisle a paralelne. Modely sieťových uzlov môžu byť použité na simuláciu zvyšku zbernice pri testovaní vyvíjaného sieťového uzla. Nástroj potrebuje rozhranie pre pripojenie k reálnej zbernici a musí byť schopný vykonať simuláciu v reálnom čase.

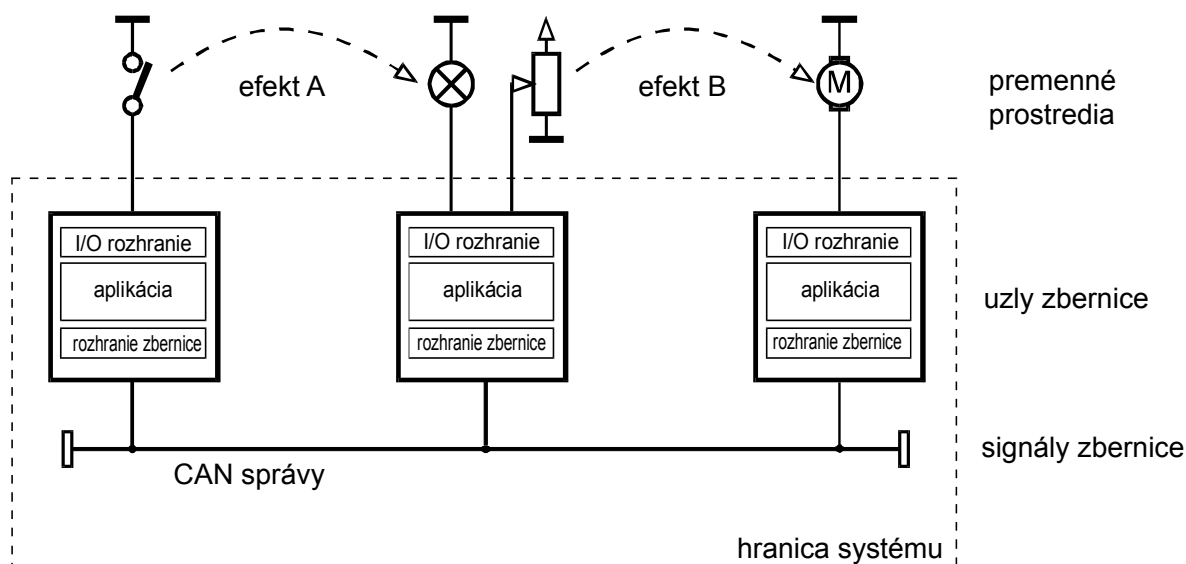
Fáza 3: Integrovanie celého systému.

V tejto poslednej fáze sú všetky reálne sieťové uzly pripájané jeden po druhom. Aby sa to dalo vykonať, musí existovať možnosť postupného odpájania modelov jeden po druhom, pre simuláciu zvyšku zbernice. Nástroj čoraz viac plní úlohu inteligentného nástroja, ktorý sleduje prechod a poskytovanie správ medzi jednotlivými reálnymi sieťovými uzlami na zbernici a porovnáva výsledky so špecifikovanými požiadavkami.

Správanie sa sieťových uzlov s ohľadom na vstupné a výstupné signály, je opisované pomocou premenných okolia. CANoe rozlišuje diskkrétne a spojité premenné. Pozícia prepínača môže byť reprezentovaná ako diskkrétne premenné zariadenia. Spojitou premennou zariadenia môže byť opísaná teplota, rozmery, alebo otáčky motora.

Ovládacie panely ponúkajú užívateľsky jednoduchý interfejs k premenným okolia. Používateľ môže vytvárať nezávislé panely pomocou editora panelov (**panel editor**). Počas simulácie môžu byť premenné okolia zobrazované (žiarovky, počítadla) a tiež interaktívne modifikované (prepínače, potenciometre).

Príklad na obr. 25 ozrejmuje funkcie, ktoré CANoe poskytuje pre simuláciu a testovanie systémov na CAN zbernici.



Obr. 25. Komponenty simulovaného systému

Stlačením tlačidla na ľavom ovládacom paneli, je diskretná premenná zariadenia **push button** nastavená na hodnotu "1". Uzol zbernice na ľavej strane reaguje odoslaním správy na zbernicu CAN. Prostredný uzol prijme túto správu a nastaví diskretnú premennú **light** na "1". To spôsobí, že sa malá žiarovka na prostrednom paneli rozsvieti.

Analogicky používateľ môže tiež nastaviť potenciometer na strednom paneli, čím je hodnota spojitej premennej **potentiometer** modifikovaná. To spôsobí, že prostredný uzol siete vyšle na zbernicu nové dáta. Táto správa je prijatá sieťovým uzlom na pravej strane. Z obsahu signálu je vypočítaná nová hodnota premennej **engine rpm**. Nakoniec to spôsobí, že zobrazenie rýchlosti (otáčok) motora na pravom paneli motora sa zmení.

Postup a správanie prezentované v predchádzajúcej časti, môže byť opísané veľmi jednoducho, funkciami dostupnými v CAPL. Týmto postupom je možné implementovať simuláciu zložitých systémov, s relatívne malým úsilím.

5.1.1 Menu

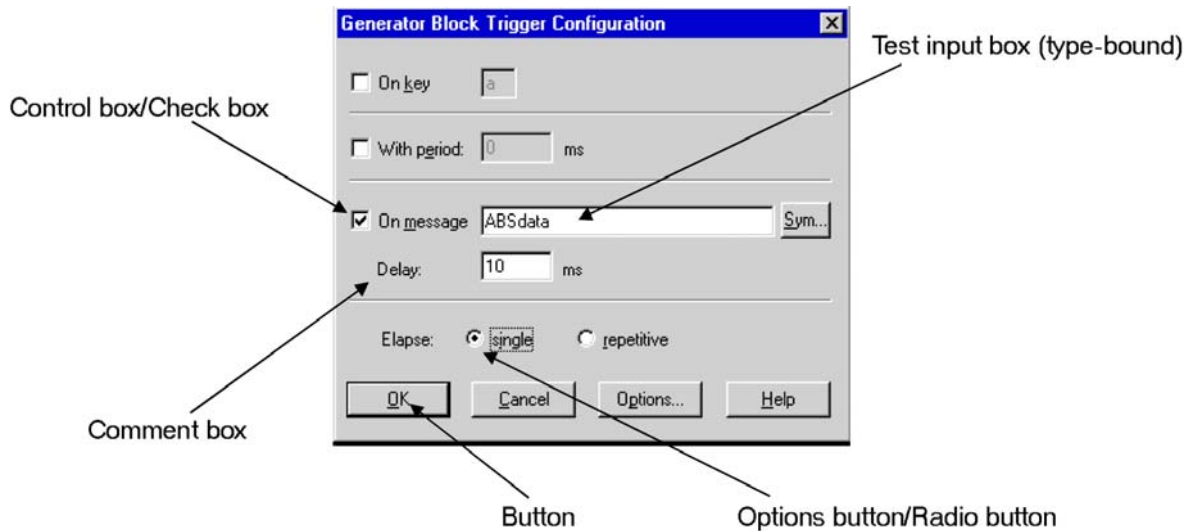
CANoe sa ovláda použitím základného menu. Individuálne príkazy sú podrobne popísané v online pomoci.

Ďalej existujú iné kontextovo závislé ponuky, opísané v oknách simulácie, dátových tokov a nastavení merania. Tieto ponuky dovoľia používateľovi špecificky nastaviť dané objekty. Otvoriť ich možno kliknutím na aktívny blok v aktívnom okne, alebo v meracom okne, pravým tlačidlom myši. Prípadne použitím klávesnice, stlačením <F10>.

Väčšina blokov môže byť parametrizovaná v nastaveniach simulácie a merania (simulation and measurement setup) voľbou prvej položky v kontextovom menu (context menu) **Configuration**. K tomuto účelu je tak otvorený konfiguračný dialóg bloku. Tento dialóg možno štartovať priamo, bez prechádzania menu, a to dvojitém kliknutím na aktívny blok, alebo stlačením <ENTER>.

5.1.2 Dialogy

Navyše, pre zadávanie príkazov, ktoré sa zvyčajne vykonáva použitím menu existuje možnosť zadávania parametrov (obr. 26). Zvyčajne sú parametre vkladané do dialógových okienok. Dialógové okienka (políčka) sú obvykle šiestich typov, z ktorých každý sa môže vyskytnúť viac ako raz:



Obr. 26. Dialogy

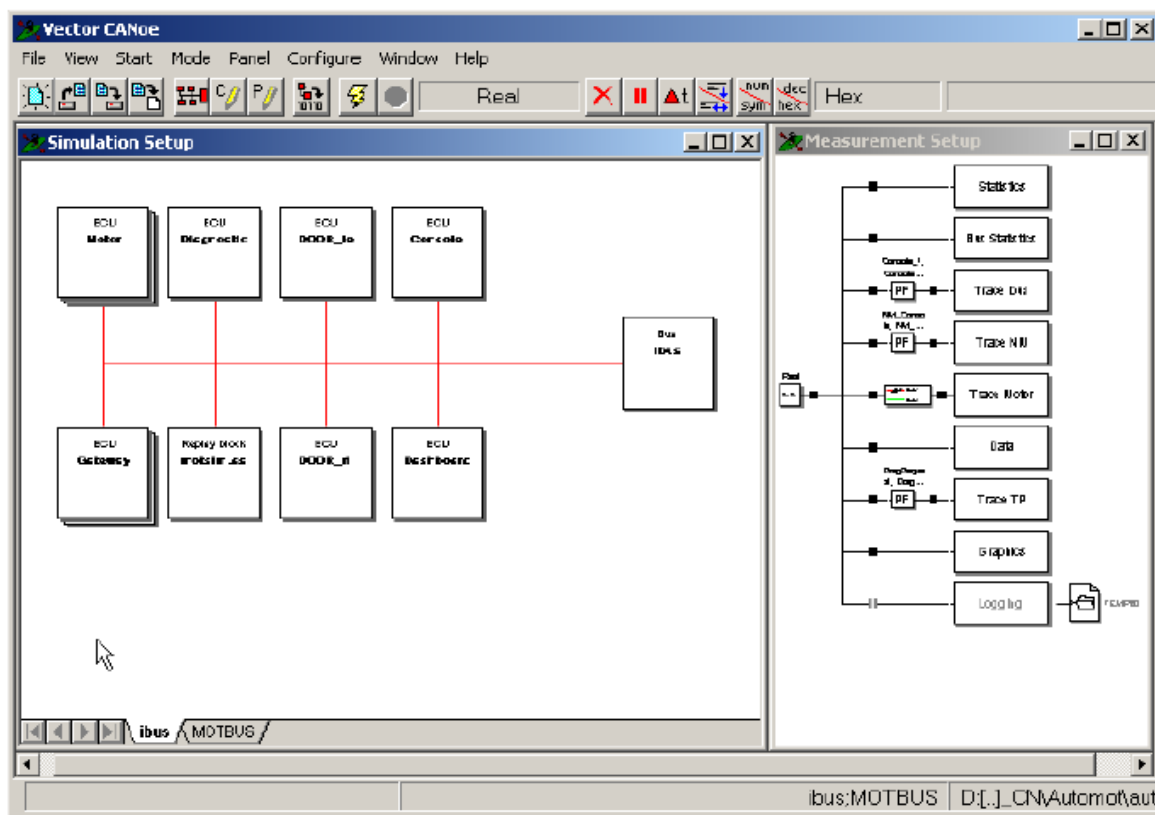
- Comment box** Táto časť hovorí užívateľovi aké sú vstupné dáta. Políčka sú v pasívnom móde a nereagujú na kliknutie myšou. Nemôžu byť taktiež prístupné z klávesnici.
- Text input box (type-bound)** alfanumerické pole, napr. na zadanie názvu súboru. Číselne pole, napr. na zadanie čísel v celočíselnom tvare alebo s pohyblivou rádovou čiarkou
- Drop-down list** Po kliknutí na vyznačené tlačidlo pozdĺž pravého okraja sa zobrazí rozbaľovací zoznam, v ktorom môžeme vybrať hodnotu z predpísaných súborov hodnôt
- Options button/Radio button** Tieto tlačidlá reprezentujú vzájomné vylučujúce nastavenia. Môžeme si vybrať iba jednu alternatívu. Ak si zvolíme jednu z možností druhá sa stáva neaktívnou. Aktuálne vybraná možnosť je označená čiernou bodkou.
- Control box/Check-box** **Check-box** (zaškrťavacie políčko vedľa voľby) slúži na možnosť indikácie či táto voľba môže byť aktivovaná alebo deaktivovaná. V tomto prípade môžeme aktivovať toľko check-boxov koľko si budeme želať. Aktívny check-box je označený ako "x" alebo "✓".
- Button** Tlačidlá slúžia na vykonanie určitej akcie, napr. na ukončenie dialógových okien alebo na otvorenie podružných dialógových okien.

Všetky dialógy majú tlačidlá označené <OK>, <Cancel> a <Help>. Ak stlačíte <OK>, tak nastavenia ktoré ste pre daný dialóg urobili, budú akceptované pre príslušný blok. Ak stlačíte <Cancel>, všetky nastavenia naposledy otvoreného dialógového okna budú stratené. Pri stlačení <Help> získate nápovedu o dialógovom okne s ktorým práve pracujete. Po zatvorení nápovedy môžete pokračovať s prácou v dialógu. Všetky nastavenia ostávajú nezmenené. Mnoho CANoe dialógov má tlačidlo <Options> - voľby (možnosti). S týmto tlačidlom možno meniť celkové nastavenia v CANoe (desiatkové/šestnástkové zobrazenie čísel, symbolický/číselný spôsob). Zmenou celkových nastavení, ovplyvňujeme reprezentáciu dát vo všetkých oknách a dialógoch.

Tam kde sú viacnásobné vstupy a pracovné okná v dialógových oknách, musí byť najskôr zvolené želané okno. Použitím myši, kliknutím na vhodné okno. Pre vstupné okno to spôsobí že textový kurzor je umiestnený na pozíciu ukazovateľa myši. Pri obsluhu klávesnicou je pohyb medzi jednotlivými položkami ovládaní pomocou <Tab> alebo <Shift-Tab>. **Check boxes** môže potom byť prepínané použitím klávesy **spacebar** (medzerníka). Klávesom <Enter> uzatvárame dialógové okná a realizuje všetky vybrané akcie v pracovnom okne.

5.1.3 (Measurement and Simulation Setup)

CANoe je primárne konfigurovaný do usporiadania okien pre nastavenie resp. zobrazenie okien merania a simulácie (obr. 27), v ktorých sú zobrazované blokové schémy tokov dát programu.



Obr. 27. Nastavenie merania a simulácie v CANoe

5.1.4 Ovládanie myšou

Všetky bloky a niektoré obrázky v meracom a simulačnom okne sa dajú ovládať myškou. Keď na zvolené okno klikneme ľavým tlačidlom myši je toto predvolené a v simulačnom usporiadaní definované ako **Active element**. Keď naň zatlačíme pravým tlačidlom myši, objaví sa rolovací menu v ktorom je objekt konfigurovaný vyššie popísanými metódami. Iným spôsobom pre volanie konfiguračného menu daného bloku môže byť priame volanie menu a to dvojitým kliknutím ľavého tlačidla myši.

5.1.5 Ovládanie klávesnicou

Keď je aktívne okno nastavenia merania alebo okno nastavenia simulácie, voľbou <Tab>, <Backspace> alebo ak je aktivovaný jeden z kurzorových klávesov, potom sa predvolenou štruktúrou stáva štruktúra v tesnom okolí aktuálne aktívneho prvku. <Tab> vedie k doprednému označovaniu (<Backspace> označovanie smerom vzad) spracovania v poradí, ktoré je vnútorne dané. Kurzorové klávesy označujú dopredne ku najbližšej časti geometricky danej (podľa polohy) v smere šípky. <F10> aktivuje kontextové menu označeného prvku. Alternatívou je (<Enter> kláves ktorý vyvoláva konfiguračný dialóg aktívneho bloku priamo. Na deaktiváciu predvoleného funkčného bloku v okne nastavenia merania a nastavenia simulácie usporiadaní používame kláves **spacebar** (medzerník). Opätovne aktivovane dosiahneme stlačením tej istej klávesy. Prostredníctvom <Ctrl-F6> a <Ctrl-Shift-F6> je možné umiestniť ktorékoľvek okno CANoe do popredia a aktivovať ho.

5.2 CANoe – stručné oboznámenie

Ak ste spustili CANoe po prvý krát a funkcie, ako aj ovládanie je vám úplne nové, prejdite nasledujúce oboznámenie.

Najskôr budete nastavovať veľmi jednoduchú CAN zbernicu, pričom sa predpokladá, že CANoe plní ako funkciu vysielateľa, tak aj prijímateľa. V prvom kroku je CANoe konfigurované ako zdroj dát, t.j. ako vysielacia stanica. Potom sa oboznámite s podmienkami analýzy CANoe, štúdiom generovaných dát v následnom meracom okne.

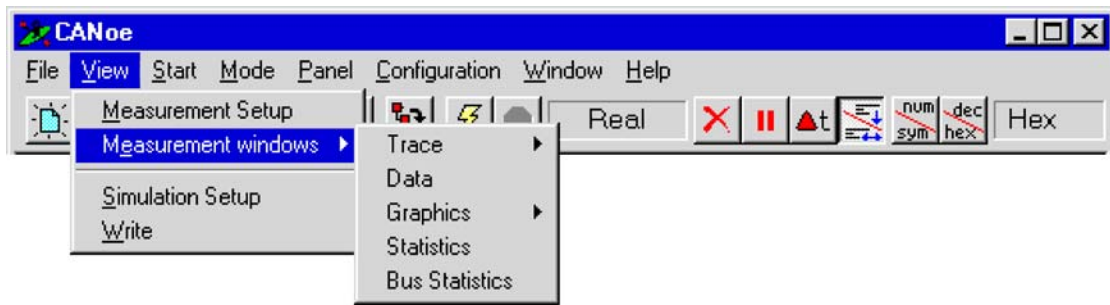
V zložitých reálnych systémoch sa predpokladajú obidve úlohy CANoe. Program môžete používať ako zdroj dát pre prenos do iných riadiacich jednotiek, ale simultánne ho môžete použiť pre sledovanie, vytváranie záznamov hlásení a vyjadrovanie vlastností prenosu dát po CAN zbernici.

V poslednej časti sa bližšie oboznámite s programovacím jazykom CAPL a vytvoríte dva sieťové uzly distribuovaného systému pre vytvorenie jednoduchej simulačnej úlohy v CANoe.

5.2.1 Prípravy

Aby ste spustili CANoe, musíte dvojklikom spustiť CANOE32.EXE, príslušnou ikonou v programovej skupine CANoe.


CANoe disponuje rôznymi oknami výsledkov (TRACE, DATA, GRAPHICS, STATISTICS a BUS STATISTICS), ako aj oknom pre nastavenie merania a oknom pre nastavenie simulácie, ktoré znázorňujú tok dát a simultánne dovoľujú konfigurovať CANoe. Ku všetkým oknám programu je možné pristupovať prostredníctvom **view menu** v hlavnom paneli nástrojov (obr. 28).



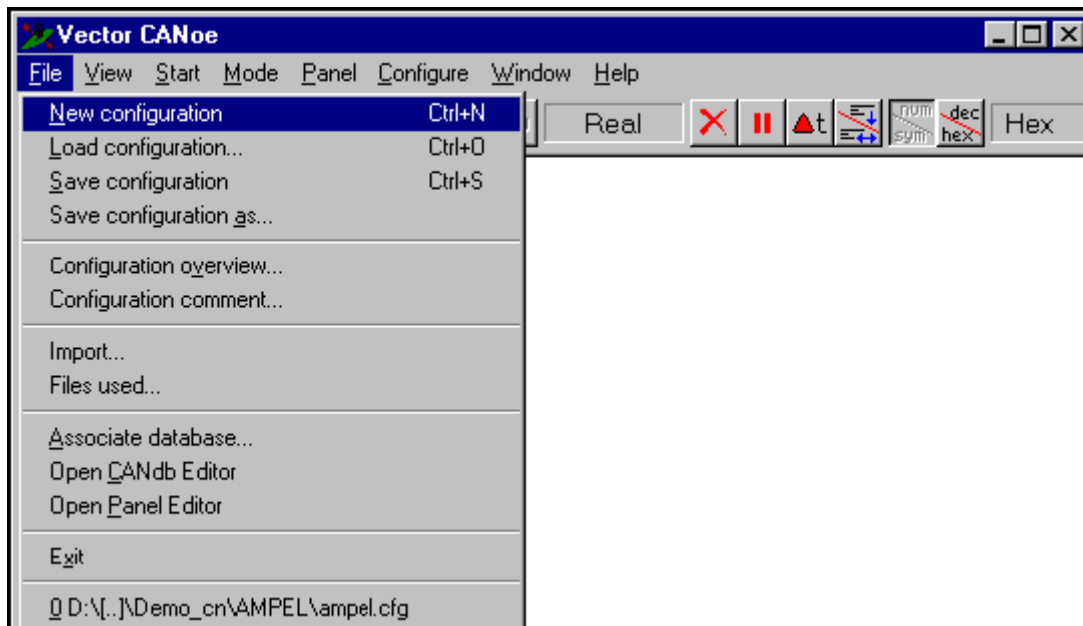
Obr. 28. View menu

V okne **simulation setup** je graficky znázornený celý systém s CAN zbernicou a sieťovými uzlami. Simulovaná zbernica je reprezentovaná červenou horizontálnou čiarou. Čierna čiara, ktorá je pod ňou symbolizuje reálnu zbernicu. Dve zbernice sú navzájom prepojené prostredníctvom PC karty. Pre prenos dát z CANoe na zbernicu, vložte blok prenosu v simulačnom nastavení a tento musí byť prepojený červenou čiarou. Bloková schéma **measurement setup** CANoe má prepojenie na **simulation setup** na ľavej strane (je znázornené symbolom >>) a rozličné bloky vyhodnocovania na pravej strane slúžia ako prijímače dát. To znamená, že tok dát je zľava doprava. Pre objasnenie toku dát sú medzi jednotlivými prvkami nakreslené prepojovacie čiary a vetvy. Prechádzajúce informácie na každý vyhodnocovací blok sú zobrazené vo vyhodnocovacom okne bloku. Napr. v okne **trace** sa zobrazujú všetky informácie prichádzajúce do bloku **trace**, kým okno **graphics** znázorňuje informáciu prichádzajúcu do bloku **graphics**.

Výnimkou je blok **logging**, ktorý nie je označený ako okno, ale ako súbor, v ktorom sú zaznamenávané (**logged**) prichádzajúce informácie do tohto bloku.

V blokovej schéme môžete zaregistrovať malé čierne štvorčeky . Do týchto bodov vkladania (**insertion points-hot spots**) môžete vkladat' ďalšie funkčné bloky pre manipuláciu s tokom dát (filtrovanie, bloky opakovania a generovania alebo užívateľom definované bloky programu napísané pomocou CAPL).

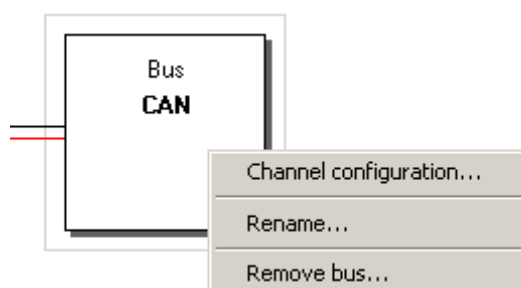
Presvedčte sa, že nasledujúce oboznámenie začínate s novou konfiguráciou voľbou položky menu **File | New configuration** (obr. 29). Spustí sa sprievodca nastavenia simulácie (Simulation Setup), ktorý pre dané oboznámenie nepotrebujete. Preto tohto sprievodcu ukončíte voľbou **Cancel**.



Obr. 29. File/New configuration

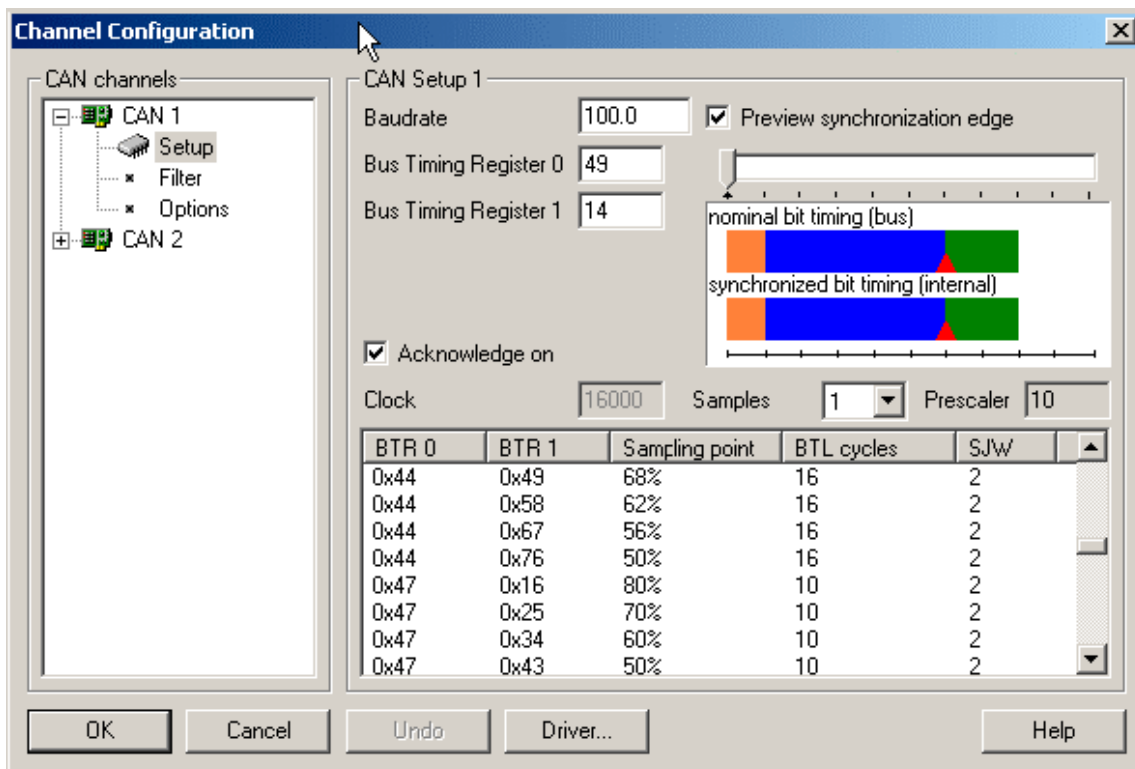
5.2.2 Nastavenie CAN zbernice

CAN zbernica je definovaná niekoľkými parametrami (rýchlosť, snímací bod, prenos), ktoré musia byť nastavené pre každý z dvoch účastníckych radičov. Menu pre nastavenie týchto parametrov vyvoláme ak na CAN zbernicu v okne **simulation setup** klikneme ľavým tlačidlom myši pričom sa nám zobrazí rollovacia ponuka (obr. 30), v ktorej zvolíme **channel configuration**. Potom nastavíme pre CAN 1 hodnotu **baudrate**(prenosová rýchlosť) na 100kBaud. Nachádzajú sa tu aj ďalšie parametre ktoré môžeme definovať (vzorkovacie body, BTL cyklus atd.). Pre správnu funkčnosť systému musia byť niektoré hodnoty CAN 1 prijaté aj pre CAN 2.



Obr.30. Rolovacie menu symbolu zbernice

Vyberte **Channel configuration** (obr. 31).

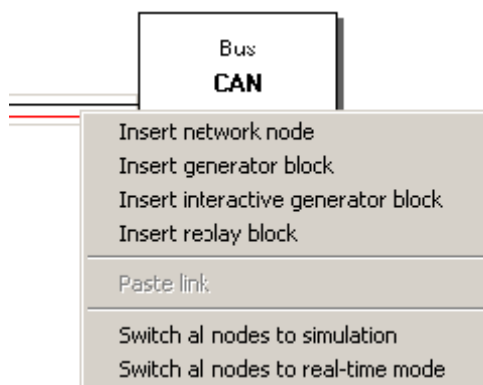


Obr.31. Konfigurácia parametrov zbernice

5.2.3 Prenos dát

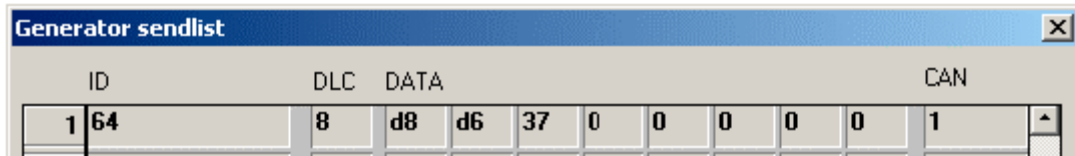
Úloha 1: Nakonfigurujte CANoe tak, aby po začatí merania bola vysielaná CAN správa s identifikátorom 64 (hex) každých 100 milisekúnd. V danom prípade by správa mala obsahovať práve 4 byty dát s hodnotami d8 (hex) d6 (hex) 37 (hex) a 0.

Dáta ktoré budeme prenášať musíme najprv vytvoriť a to pomocou generátora. Ten vytvoríme v okne **Simulation setup** kde kliknutím pravým tlačidlom myši na červenú čiaru v rolovacom menu (obr. 32) zvolíme **insert generator block**.



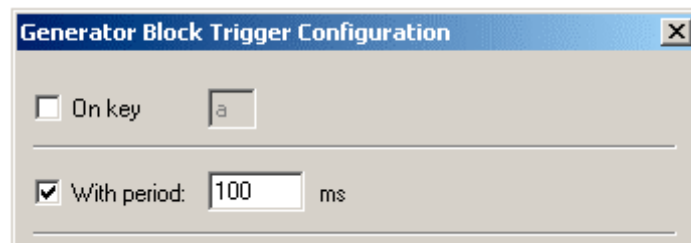
Obr. 32. Symbol zbernice v okne **Simulation setup** s kontextovým menu

Konfiguráciu vyvoláme stlačením pravého tlačidla myši na generátore a výberom konkrétneho nastavenia . Najprv si nastavíme **transmit list** (obr. 33). Zapišeme číslo 64 ako ID(názov správy) (uistite sa či formát čísel je nastavený na hexadecimal ak nie, zmenu vykonáme stlačením **option**). Potom zadáme do kolónky **DLC** číslo 4 (**DLC** číslo závisí od dĺžky vst. dát). Nakoniec nastavíme hodnotu byty dát v prvých štyroch okienkach na hodnoty **d8, d6, 37** a **0**.



Obr. 33. Transmit list generátora

Zmeny uložíme stlačením OK. Pre správnu činnosť generátora je potrebné nastaviť spúšťanie- v menu zvolíme **generator block trigger configuration** (obr. 34) kde zadáme hodnotu periódy na 100ms.




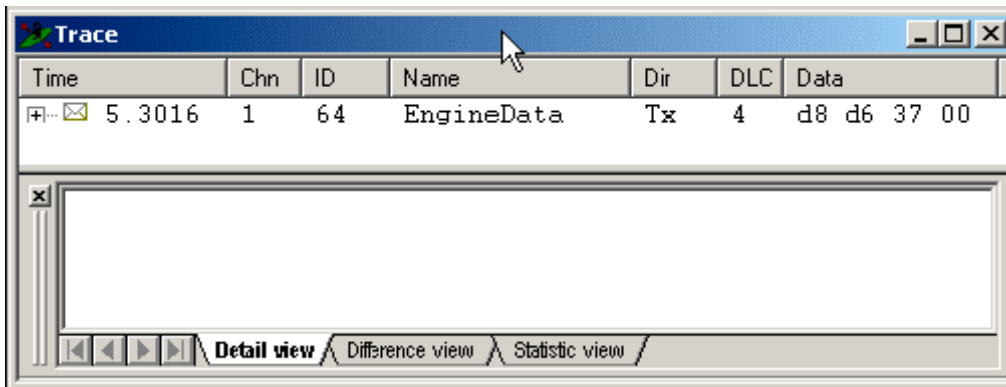
Obr. 34. Spúšťanie bloku generátora

Pred tým ako začneme merať je potrebné si konfiguráciu vykonanú v tomto bode uložiť a to príkazom menu **File/Save configuration**. Tým je možné zvolenú konfiguráciu kedykoľvek vyvolať.

Ďalej, CANoe vyžaduje aby pred začatím merania bola konfigurácii pridružená databáza. Preto je na začiatku potrebné priradiť databázu MOTBUS.DBC umiestnenej v adresári DEMO_CAN_CN\AUTOMOT\CANDB.

CANoe ponúka možnosť editácie databáz v **simulation setup** (pridávanie, mazanie, priradovanie atď.) Pri tejto demo verzii máme k dispozícii niekoľko preddefinovaných databáz, ktoré si zavoláme tak, že v **simulation setup** pravým kliknutím myši na **Databases** vyberieme z rolovacieho menu príkaz **Add**. Potom už stačí nájsť príslušnú databázu a kliknúť na **Open**. Máme možnosť tiež načítania niekoľkých databáz.

Začiatok merania je podmienený stlačením štart tlačidla na panely nástrojov . Okamžite po spustení začne generátor cyklicky posielať nakonfigurovaný signál. Ten je možné pozorovať v **Trace window** (obr. 35), ktoré sa pri meraní automaticky zobrazí. V prvej kolónke je zobrazený relatívny čas prenosu ku začiatku merania, v druhej ktorý z CAN kanálov je použitý na prenos.



Obr. 35. Trace okno

Okrem zbernice CAN 1 je možnosť využitia aj zbernice CAN 2 ale len pri hardwarovom prevedení.

Úloha 2: Rozšírte konfiguráciu predchádzajúcej úlohy tak, že ďalšia správa s identifikátorom 3fc (hex) je vysielaná každých 200 ms. Hodnota prvého dátového bytu sa cyklicky mení od 1 do 5.

Túto úlohu môžete riešiť vložení iného bloku generátora v okne **simulation setup**. Zvoľte 200 ms ako hodnotu cyklického spúšťania **cyclic triggering**. Zoznam prenosu (**transmit list**) by mohol vyzeráť tak ako je to znázornené ďalej (obr. 36).

ID	DLC	DATA	CAN
1	3fc	1	1
2	3fc	2	1
3	3fc	3	1
4	3fc	4	1
5	3fc	5	1

Obr. 36. Zoznam prenosu bloku generátora



Nezabudnite zastaviť meranie predtým, než rekonfigurujete nastavenie simulácie (**simulation setup**). Počas merania nie je možné robiť zmeny v konfigurácii dátového tuku. Položky relevantných rozbaľovacích kontextových menu sú nevýrazne sivé.

Okrem blokov generátorov CANoe taktiež ponúka dva ďalšie typy blokov ako zdroje dát. Blokom prehrávania môžete opätovne prehrávať dáta na zbernicu, ktoré boli napr. zaznamenané pomocou funkcie **logging**. Blok programu umožňuje integrovať do CANoe vlastné funkcie (môžu byť aj celkom zložitú) v programovacom jazyku CAPL.

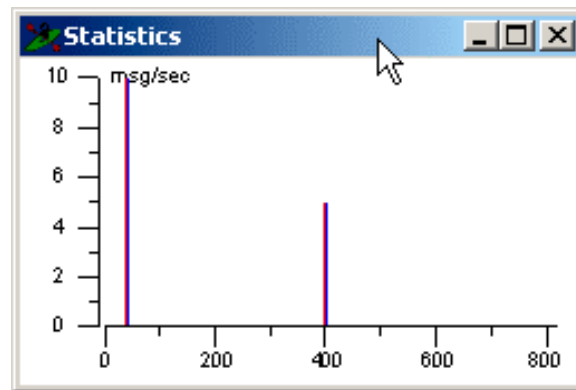
5.2.4 Okná vyhodnocovania

Na analýzu dát generovaných generátorom v **simulation setup** sú použité okná vyhodnocovania.

Poviem si niečo o funkcii **Trace window**. Dáta ktoré prechádzajú **Trace** blokom sú v **Trace window** zobrazené ako CAN správa v zbernicovom tvare. Okrem časovej značky obsahuje číslo CAN kontroléra, identifikátor, na rozlíšenie či sa jedná o prichádzajúcu alebo odchádzajúcu správu a byty dát CAN správy. Konfiguračné menu **Trace window** spustíme pravým kliknutím myši v okne.

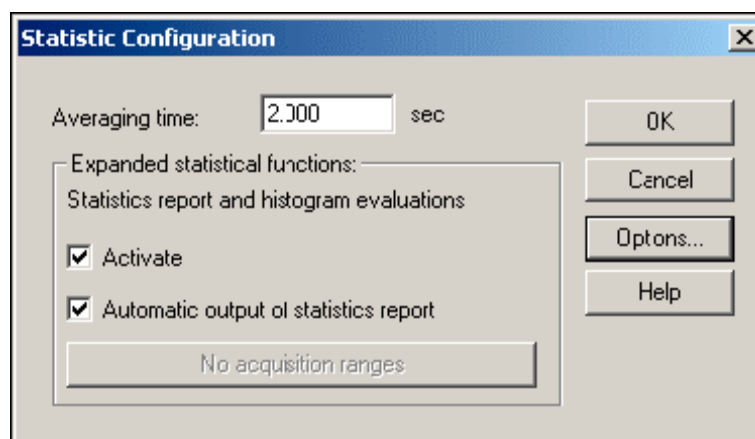
Ďalej, štyri tlačidlá na pravej strane panela nástrojov sa tiež môžu použiť na konfiguráciu **Trace window**. Napríklad, s tlačidlom  je možné meniť **stationary mode** (statický mód) na **scroll mode** (rolovací mód), v ktorom každá prichádzajúca správa je zapísaná v novom riadku. Toto tlačidlo  umožňuje prepínanie medzi absolútnym a relatívnym časovým zobrazením.

Okno štatistických údajov (**Statistics window**) tiež poskytuje informácie o zbernici (obr. 37). Tu možno sledovať frekvenciu prenosu správ kódovaných identifikátormi. Po začatí merania je v okne štatistických údajov vidieť dve vertikálne čiary (v danom príklade), ktoré zobrazujú frekvencie prenosu dvoch generovaných správ 64(hex) a 3FC(hex). 10 správ/sekunda (msg/sec) je pre správu s identifikátorom 64(hex) a 5msg/sec pre 3FC(hex). Tento výsledok odpovedá periódam 100 a 200ms blokom generátorov.



Obr. 37. Štatistické okno

Keď je v **Graphics window** zobrazenie príliš nepresné, pomocou **statistic block**-u získame štatistickú správu, ktorá uvádza presnejšie informácie o intervale prenosu každej správy. Zastavíme meranie a pri konfigurácii **Statistic block**-u (**Measurement Setup**) zaznačíme kolónku **Activate** (obr. 38), čím dosiahneme po skončení merania zobrazenie štatistickej správy vo **Write window** (obr. 39).



Obr. 38. Aktivácia štatistickej správy

Source	Message	N	Aver	StdDev	MIN	MAX
* System	Statistics report: AR0001, 02:51:07 pm					
* System	Statistics for transmit spacing of messages in [ms]					
* System						
* System						
* System	64 RX	89	99.995	0.78717	98.38	101.09
* System	TX	89	99.995	0.78703	98.39	101.09
* System	3fc RX	44	199.99	0.45507	198.89	200.49
* System	TX	44	199.99	0.45533	198.89	200.49
* System						

Obr. 39. Štatistická správa

Tá okrem celkového počtu správ zobrazuje priemernú hodnotu, smerodajnú odchýlku, a maximum a minimum zaznamenaného intervalu prenosu.

Iným oknom, ktoré sa vzťahuje ku zbernici je okno **Bus Statistics** (obr. 40). Poskytuje prehľad o toku dát na zbernici. Sú tu zobrazené celkové informácie o frekvenciách dát, chybné a pretečené rámce, zaťaženie zbernice a stav radiča zbernice. Pretože v našom prípade je jedna správa vysielaná každých 100ms a druhá každých 200ms celková frekvencia je 15 rámcov za sekundu s priemernou dĺžkou 70 bitov na rámec (pribl. $15 \cdot 70 \approx 1000$ bit/s). Pri prenosovej rýchlosti 100kbit/s pre náš príklad je zaťaženie zbernice približne 1%.

	CAN 1	CAN 2
Busload [%]	1.04	1.03
Peakload [%]	1.15	1.15
Std. Data [fr/s]	15	15
Std. Data [total]	118	118
Ext. Data [fr/s]	0	0
Ext. Data [total]	0	0
Std. Remote [fr/s]	0	0
Std. Remote [total]	0	0
Ext. Remote [fr/s]	0	0
Ext. Remote [total]	0	0
Errorframe [fr/s]	0	0
Errorframes [total]	0	0
Chip state	Active	Active

Obr. 40. Okno štatistiky zbernice (**Bus Statistics**)

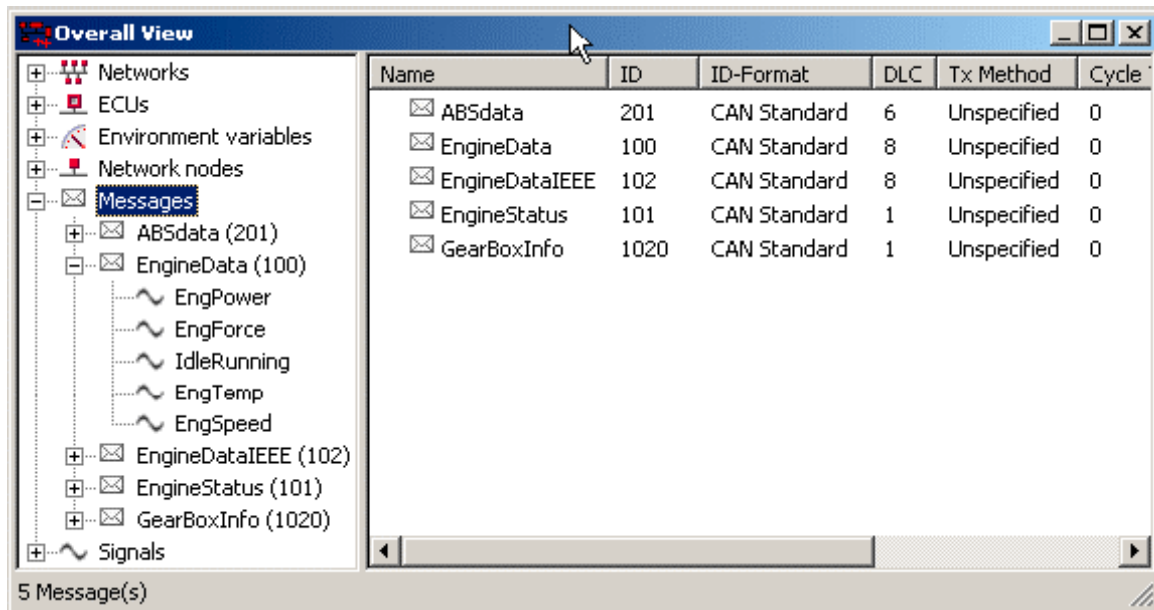
5.2.5 Práca so symbolickými dátami

Prvotným záujmom analýzy CAN systémov (okrem informácií v zťahovaným k zbernici ako správy, chybné rámce, frekvencie správ) je informácia o využívaní dát ako napr. signály o otáčkach, teplote a zaťažení motora, ktoré sú poskytované jednotlivými radičmi CAN a sú vysielané na zbernicu prostredníctvom CAN správ.

Pre symbolický opis týchto informácií CANoe ponúka databázu formátu DBC a databázový editor s ktorým je možné čítať, vytvárať a modifikovať CAN databázy.

V tomto kroku by sme radi použili databázu MOTBUS.DBC, ktorá je momentálne pridružená aktívnej konfigurácii. Táto databáza bude použitá na interpretáciu dátových bytov správy generovanej generátorom. Za týmto účelom, najprv otvoríme databázu použitím tohto

tlačidlá na panele nástrojov. Otvorí sa CANdb++ editor a obsah databázy MOTBUS.DBC sa zobrazí v okne **Overall View** (obr. 41).



Obr. 41. Okno editora CANdb++

Potom dvojitým kliknutím na záložku **Messages** na ľavej strane okna **Overall View** dosiahneme zobrazenie všetkých dostupných správ ktoré databáza obsahuje. Tie sú tiež zobrazené aj na pravej strane okna s ich systémovými parametrami. Ako prvé prepne formát čísel na hexadecimal v **Options/Settings**.

Kliknutím na správu *EngineData* na ľavej strane okna sa na pravej strane zobrazia systémové parametre prenášaných signálov obsiahnutých v tejto správe. Voľbou ľubovoľného signálu z ľavej strany okna sa na pravej strane zobrazia podrobné informácie o tomto signále, ktoré v CANoe interpretované príslušným spôsobom.

Správy, ktoré sú generované v našom príklade dvoma generátormi sú pomocou databázy MOTBUS.DBC interpretované ako *EngineData* (64 hex) a *GearBoxInfo* (3FC hex).

Teraz ak spustíte meranie môžete prepnúť program do symbolického módu aktiváciou tlačidla



. V okne **Trace** sa objavia symbolické mená správ spolu s ich numerickým identifikátorom.

5.2.6 Analýza hodnôt signálu v okne Data Window

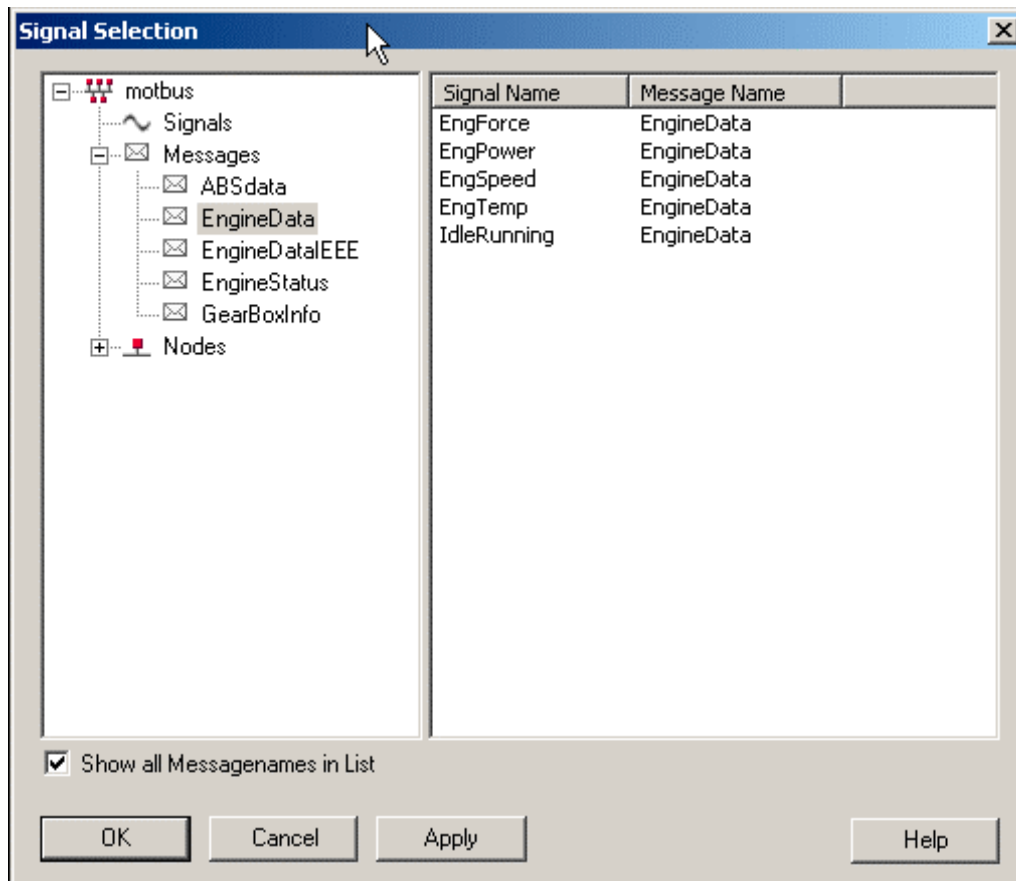
Besides the use of symbolic message names, the associated database can also be used to analyze signal values. The purpose of the Data window is to assist in the study of momentary signal values.

Okrem využitia na symbolické názvy správ, možno pridruženú databázu použiť na analýzu hodnôt signálu. Cieľom okna **Data window** je prispievať pri sledovaní momentálnych hodnôt signálov.

Toto vysvetľuje prečo je pri novej konfigurácii **Data window** zo začiatku prázdne. Zobrazené hodnoty signálu sú výlučne založené na informáciách z databázy. Vy ako užívateľ musíte rozhodnúť len o tom ktoré hodnoty signálu budú zobrazené.

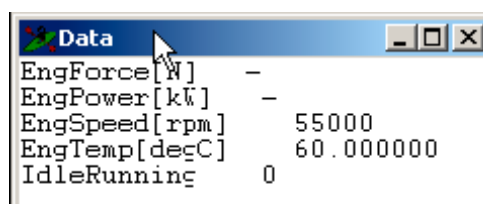
Úloha 3: Nakonfigurovať **Data window** tak aby zobrazoval hodnoty signálu správy *EngineData* ktoré je generované v prenosovej vetve.

Pri riešení tejto úlohy postupujeme nasledovne. Najprv otvoríme rolovacie menu **Data window** v ktorom zvolíme **Configuration...**. Zobrazí sa nám konfiguračný dialóg ktorý obsahuje na začiatku prázdne okno správ. Pomocou tlačidla **New signal** sa spustí signálový prieskumník (obr. 42), ktorý umožní vybrať požadovaný signál z databázy. Hierarchia tohto prieskumníka je rozdelená na 2 časti. Ľavá strana obsahuje okno na vyhľadávanie špecifickej správy ktorá obsahuje signály a tie sú zobrazené na pravej strane prieskumníka.



Obr. 42. Prieskumník signálov

Po ukončení konfigurácie je vidieť že názov zvoleného signálu zo signálového prieskumníka je zobrazený v **Data window** (obr. 43). Spustením merania začne generátor cyklicky posielat správu *EngineData* obsahujúcu byty D8, D6, 37 a 0 do zbernice. Podľa vlastností správy v databáze, Data blok v **measurement setup** preloží tieto hodnoty bytov ako rýchlosť motora, teplota a zobrazenie príslušných hodnôt signálu v **Data window**.



Obr. 43. Dátové okno

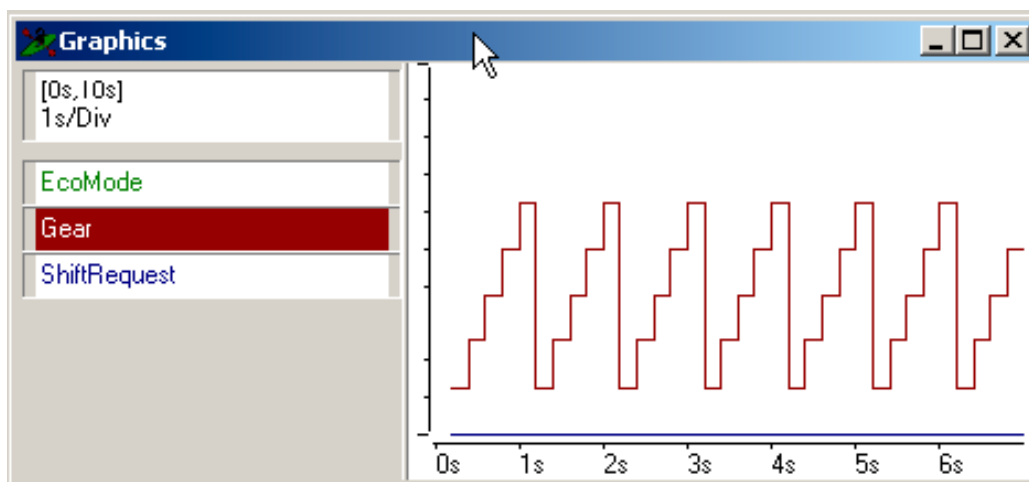
Pomocou prevodového vzorca obsiahnutého v databáze je rýchlosť motora zobrazená v otáčkach, zatiaľ čo teplota je v stupňoch Celzia. Hodnota všetkých troch signálov zostáva po celý čas konštantná pokiaľ je správa vždy prenášaná s rovnakými data bytmi D8, D6, 37 a 0.

5.2.7 Analýza odpovedí signálu v okne grafiky(Graphics window)

Pokiaľ **Data Window** zobrazuje okamžité hodnoty signálu, máte možnosť zobrazovať časové odozvy signálu v okne **Graphics window**.

Úloha 4: Nakonfigurujte **Graphics window** tak, že hodnoty signálu sú zobrazené v správe 3FC ktorá je generovaná vo vetve prenosu.

Druhá správa generovaná v prenosovej vetve je tiež opísaná v pridruženej databáze. Z databázy je zjavne že identifikátor 3FC zodpovedá symbolickej správe s názvom *GearBoxInfo* obsahujúci signály *Gear*(prevodový stupeň), *ShiftRequest*(požiadavka na radenie) a *EcoMode*(ekonomický mód). Máte hneď možnosť sledovať časové odozvy týchto signálov v **Graphics window** (obr. 44). **Graphics window** môže byť nakonfigurovaný rovnako ako **Data window**. Z rolovacieho menu zvolíme **signal selection**, zobrazí sa nám dialógové okno kde stlačíme **Add signal**(pridať signál). Tým sa otvorí signálový prieskumník(podobne ako u konfigurácii Data window). V ňom zvolíme správu *GearBoxInfo* a z pravej strany okna vyberieme 3 signály. V **Graphics window** sú potom tieto signály zobrazené ako legenda na pravej strane okna. Po začatí merania pozorujeme že sa signál *Gear* cyklicky opakuje od hodnoty 1 po 5 pokiaľ ďalšie dva signály zostanú konštantné



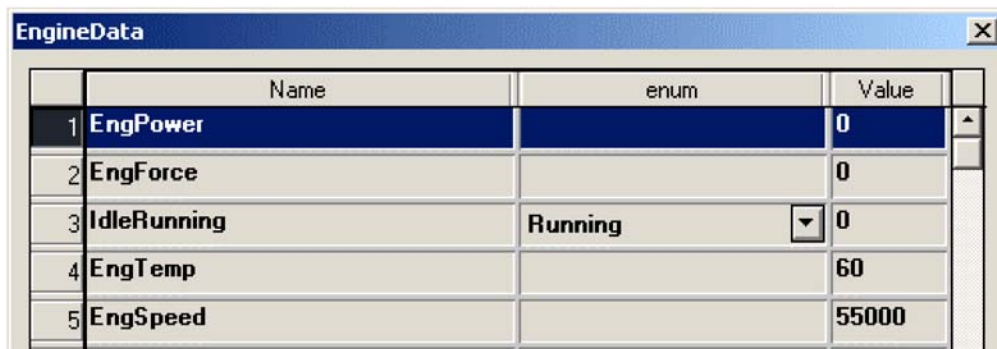
Obr. 44. Okno Grafu

5.2.8 Použitie databázy vo vysielaných správach

Doteraz sme používali na pozorovanie hodnôt signálu symbolickú databázu. Napríklad otvoriť **transmit list** bloku generátora z úlohy 1. Namiesto toho ako ste spúšťali transmit list (64), teraz môžete zistiť **symbolic name** v prvej kolónke. Čiže správu možno zadať priamo z databázy použitím tlačidla **Symbol...** (**Simukation setup - Configuration of transmit list...**), bez potreby pracovať s identifikátorom.

Hodnoty signálu môžu byť teraz menené priamo v **transmit list-e**. V **transmit list-e** vyberte prvý riadok a potom kliknite na tlačítko signál (obr. 45). V dialógovom okne teraz môžete

priamo zadať hodnoty signálu. Znova bude jasné, že hodnoty D8, D6, 37 a 0 z prvého riadku odpovedajú hodnotám signálu *EngSpeed* = 55000 rpm, *EngTemp* = 60 stupňou Celzia a *IdleRunning* = 0.




	Name	enum	Value
1	EngPower		0
2	EngForce		0
3	IdleRunning	Running	0
4	EngTemp		60
5	EngSpeed		55000

Obr. 45. Hodnoty v bloku generátora

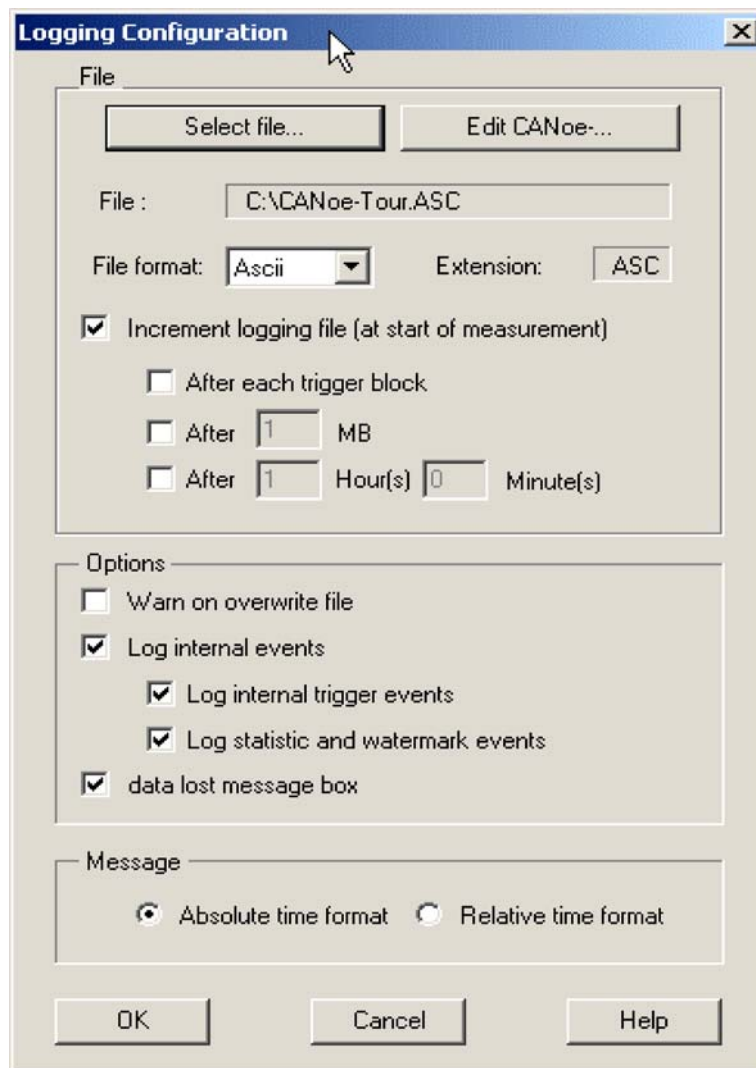
Napríklad: Ak teraz nastavíte hodnotu *EngSpeed* na 1000 rpm, blok generátora bude automaticky používať databázu informácií na počítanie príslušných dátových byte-ov (10, 27, 37, a 0)

5.2.9 Záznam merania

CANoe má rozsiahle množstvo funkcií pre zaznamenávanie dát. V štandardnom okne nastavenie merania (**Measurement setup**) je táto vetva umiestnená v dolnej časti obrazovky.

Možno to ľahko rozoznať podľa ikony , ktorá symbolizuje **log file**. **Log file** je počas merania plnený dátami.

Na zaznamenanie dát, ktoré prídu do CANoe počas merania, treba najprv aktivovať **logging branch**. Možno to vykonať dvojitým kliknutím na break symbol, alebo v rolovacom menu. Z rolovacieho menu ikony súboru, otvoriť **configuration dialog** (obr. 46). Tu možno zadať meno pre záznam merania, a tiež jeho formát. Vyberte formát ASCII.



Obr. 46. Konfiguračný dialóg v časti Logging

Zápis v binárnej forme zaberie menej miesta na harddisku, ale obyčajný textový editor ho neprečíta. Ponuka **Offline mode** ponúka na vyhodnotenie zápisu dve formy.

Okrem ikony súboru môžete presne určiť spúšťacie podmienky pre zápis súboru v **logging block**-u. Toto je často vhodné, pretože často to nieje prenos dát po CAN zbernici počas merania čo nás zaujíma, ale sú to časové intervaly.

Na zápis merania je postačujúce zmeniť režim zo **Single Trigger** na **Entire Measurement** v dialógu **trigger configuration**.

Kliknutím na OK opustí dialóg a potom spustí meranie, ktoré po 20 sekundách zastavíte. Teraz dvojitým kliknutím na ikonu **log file** možno otvoriť zaznamenaný súbor ASCII. Okrem zaznamenaného merania možno vidieť, že sú zaznamenané aj štatistické informácie. Tieto riadky korešpondujú presne k informáciám, ktoré sa zobrazili počas merania v okne **Bus Statistics**.

5.2.10 Vyhodnotenie Log File

Log Files vo formáte ASCII si skutočne možno prezrieť v textovom editore, ale často je múdrejšie využiť vybavenie, ktoré CANoe poskytuje pre Offline analýzu **Log Files**.

Úloha 6: Prehrať **Log File** zaznamenaný pri poslednej úlohe v Offline mode a pozorovať odpoveď signálu v grafickom okne.

Na riešenie tejto úlohy treba najprv prepnúť CANoe do **Offline mode**. V menu **main Mode** sú dve položky: **To Offline** a **To Offline (Copy)**. Konfiguráciu grafického okna pripravenú v **Online mode**, je vhodné kopírovať všetky konfigurované voľby zo zložky analýzy do **Offline mode**, použitím **To Offline (Copy)**.

Namiesto symbolu zbernice je ikona súboru. Inak všetky nastavenia merania z **Online mode** boli prijaté. Dvojitým kliknutím na **file icon** (ikonu súboru) možno zostaviť zdroj dát, naľavo od nastavenia merania a v poslednej úlohe zadaného mena **Log file-u**.

Tiež je treba separovať **Logging block**. To je možné spraviť dvojitým kliknutím na symbol aktívneho bodu, vpredu (vľavo) **Logging block-u**, alebo pomocou rozbaľovacieho menu tohto aktívneho bodu.

Stlačením <F9> je možné prehrať meranie. Na rozdiel od **Online mode**, tu CANoe ponúka voľbu reprodukcie merania v spomalenom pohybe (Start/položka **Animate**, alebo F8), alebo v krokovom režime (Start/položka **Step**, alebo F7).

Tie isté analýzy funkcií sú k dispozícii tak v **Offline** ako aj v **Online mode**. Totiž nahrané dáta sú zobrazené vo formáte týkajúceho sa zbernice v **Trace window**, zatiaľ čo možno pozorovať zaznamenané odpovede signálu v grafickom okne.

Samozrejme, do nastavenia merania je možné vložiť filtre, alebo CAPL programy, na zredukovanie dát, alebo vložiť ďalšie používateľom definované funkcie analýz.

5.3 Vytváranie CAPL programu

Aj keď CANoe poskytuje veľký počet funkcií, ktorých parametre možno konfigurovať, skôr či neskôr bude potrebné rozšíriť funkčnosť CANoe, pomocou vlastných funkcií, pre špeciálne požiadavky úloh. Preto CANoe ponúka programovací jazyk CAPL.

Nasledujúca úloha pozostáva z tvorby jednoduchého CAPL programu na počítanie správ, ktoré sú vytvárané v nastavení **CANoe simulation**. Nájdeme kompletný popis programovacieho jazyka spolu s početnými príkladmi v online pomoci.

Úloha 7: Zostavte CAPL program, s ktorým budete môcť spočítať počet správ typu *EngineData* (ID 64 hex) a výpočet počtu správ; odpoveď sa zobrazí stlačením klávesy

Najprv treba prepnúť CANoe do **Online mode**. V **simulation setup** blok generátora, ktorý cyklicky posiela *EngineData* do zbernice, by mal byť stále zdrojom dát.

Treba sa rozhodnúť, kde v blokovej schéme si prajeme CAPL program vložiť. V zásade každý aktívny bod v **measurement setup** (v nastavení merania), alebo v **simulation setup** je k dispozícii. Avšak, keďže program spĺňa všetky náležitosti pre analýzu a sám negeneruje nejaké správy, ale ich len počíta, je vhodné vložiť program na pravú stranu **measurement setup**, asi pred blok štatistiky. V rolovacom menu aktívneho bodu treba zvoliť funkciu **Insert CAPL node**. Na vybranom bode v **measurement setup** sa znázorňuje funkčný blok so symbolom **P**. Možno pristúpiť ku konfigurácii uzla, cez rolovacie menu. Najprv vybrať názov programu, napr. COUNTER.CAN a potom spustiť prehliadač CAPL, oba z **configuration dialog**, tlačidlom <Edit...>, alebo priamo dvojitým kliknutím na blok programu v **measurement setup**.

CAPL je programovací jazyk založený na udalosti. Každý CAPL program pozostáva z procedúr udalostí, s ktorými možno reagovať na vonkajšie udalosti (napr. výskyt správy na zbernici alebo stlačenie klávesu na klávesnici). Prehliadač CAPL je detailne opísaný v Online pomoci. Pomocou jeho pod-okien („Panes“) je možné tvoriť a upravovať CAPL programy rýchlo a ľahko.

V princípe možno na tvorbu CAPL programu použiť aj vlastný textový editor. CAPL programy sú ASCII súbory s príponou názvu *.CAN, ktoré musia byť skompilované pred spustením merania použitím kompilujúceho programu, poskytovaného programom CANoe.

Pre daný program sú potrebné celočíselné premenné, ktoré počítajú správy. Napríklad, nazveme ju **counter**. Ideme do pravého horného políčka prehliadača a zadáme tento názov do bloku premenných. Následne sa v políčku ukáže:

```
variables {  
  int counter;  
}
```

Po štarte merania sa tak, ako všetky globálne premenné, aj táto inicializuje na hodnotu 0.

Je treba rozšíriť CAPL program zahrnutím udalosti typu ON MESSAGE (reaguje na udalosť správy). Za týmto účelom treba kliknúť na CAN MESSAGES, použitím pravého tlačidla myši a vložiť novú udalosť použitím príkazu **New** v rolovacom menu.

Následne sa objaví v okne textového editora procedúr šablóna procedúry: najprv nahradiť text <newMessage> symbolickým názvom *EngineData*, ktorý možno prevziať priamo z databázy v rolovacom menu, položka **CANdb Message**.

Počas kompilácie CAPL kompilujúci program nahradzuje symbolický názov príslušným identifikátorom 0x64.

Teraz je potrebné definovať, ktoré akcie majú byť vykonané keď prípad nastane. Od vtedy čo program počíta správy, premená veličina `counter` musí byť pričítaná vždy keď je správa registrovaná. Kompletná metodika vyzerá:

```
on message EngineData
{
    counter++;
}
```

Ako posledný krok, musí byť zrealizovaný výstup k **Write window**. Nakoniec program by nemal len počítať správy, ale tiež sledovať, koľko správ je spočítaných.

Výstup k **Write window** by sa mal objaviť pri stlačení <a>. Preto je treba definovať inú udalosť pre prípad: “**Press key <a>**”. V stromovom prehliadači sa vyberie **Keyboard**. Toto spôsobí, že predtým definované **on message** zmizne, pretože sa jedná o iný typ udalosti. Samozrejme to stále ostáva zložkou programu CAPL a znova sa objaví pri voľbe udalosti **Messages**.

Teraz vložte udalosť **Keyboard** v CAPL programe z rolovacieho menu. Nová procedúra sa objaví v textovom editore, ktorý sa vyplní nasledovne:

```
on key 'a'
{
    write("%d EngineData messages counted",counter);
}
```

Formát %d odkazuje na celočíselnú premennú `counter`, ktorá je zadaná za čiarkou.

Teraz uložte svoj program voľbou **save program** a aktivujte kompilačný program pomocou <F9> alebo v **main menu** príkazom **Compiler/Compile** alebo pomocou ikony v **Toolbar-e**. Ak je v programe chyba, otvorí sa okno so správou a zobrazí chybu. Dvojitým kliknutím na túto správu sa dostaneme na miesto kde sa chyba nachádza. Po opravení chyby a uložení programu treba program znova skompilovať. Akonáhle je program skompilovaný a je bez chyby, zobrazí sa v stavovom riadku správa **COMPILED**.

Teraz spustíme meranie. Generátor začne cyklicky vysielat' správy typu *EngineData*, ktoré sú teraz počítané vaším programom. Pri stlačení <a> môžeme vo **Write window** vidieť text: “*n EngineData messages counted*”, kde *n* reprezentuje počet spočítaných správ.

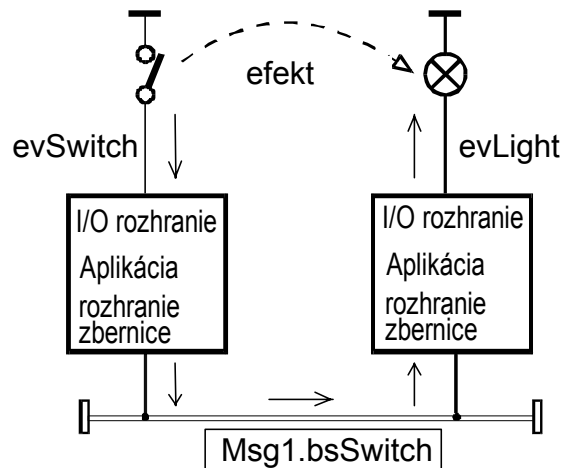
5.4 Simulácia distribuovaných systémov v CANoe

CANoe poskytuje premenné okolia pre možnosť modelovania funkčného správania uzlov na zbernici. Tieto premenné okolia sú opísané udalosťami a stavmi okolia systému (napr. vonkajším tlakom, teplotou, polohou spínača atď.). Môžete ich sledovať a úmyselne meniť tieto stavy t.j. hodnoty premenných okolia, pomocou používateľom definovaných ovládacích panelov.

Na prácu v CAPL s premennými okolia sa používa procedúra typu **on envVar** (reaguje na zmenu v premenných). Funkcie *getValue()* a *putValue()* sa používajú na čítanie a zápis premenných premenných.

Tieto nástroje jazyka a symbolický prístup k premenným, definovaným v databáze umožňujú vytvoriť jednoduché prototypy modelov sieťových uzlov.

Nasledujúca úloha pozostáva z tvorby kompletnej CANoe konfigurácie s dvoma modelmi sieťových uzlov a pridružených periférií, t.j. ovládacích panelov. Po tom ako používateľ zapne spínač (obr. 47), prvý uzol informuje o tejto akcii druhý uzol. Druhý uzol potom aktivuje indikačnú žiarovku ako svoju perifériu.




Obr. 47. Konfigurácia CAN zbernice s dvoma modelmi uzlov

Táto najjednoduchšia funkcionálna bola vybraná na to, aby zamerala používateľov na tvorbu modelov a nie ich funkcií. Komplexnejšie decentralizované systémy sa môžu stavať podľa toho istého vzoru.

Model pre vytvorenie decentralizovaného systému v CANoe môže byť účinne vytvorený v troch krokoch:

1. Vytvoriť databázu so správami, signálmi a premennými okolia
2. Vytvoriť periférie sieťového uzla, t.j. ovládacie panely
3. V CAPL vytvoriť model uzla

Prípraviť sa na túto úlohu možno napríklad vytvorením novej, prázdnej konfigurácie, kliknutím na  v paneli nástrojov (toolbar).

5.5 Tvorba databázy

Prvý krok zahŕňa vytvorenie databázy, ktorá opisuje nasledujúce dva významné aspekty systému:

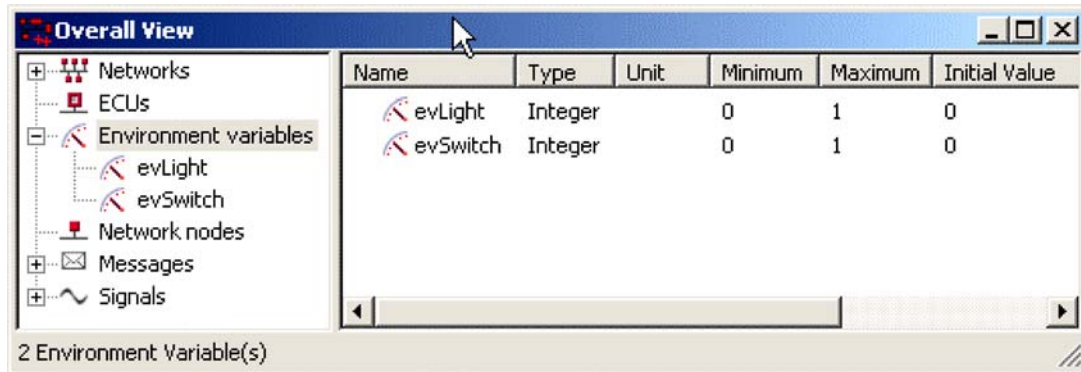
- výmena informácií medzi dvoma uzlami cez CAN zbernicu
- I/O rozhranie k perifériám, t.j. prepojenie medzi každým uzlom a jeho vstupnou a výstupnou jednotkou

Databázová správa a objekty signálov sú vhodné a schopné na opis výmeny informácií cez CAN zbernicu. Jednoduchá funkcionálna v príklade môže byť riadená 1-bitovým signálom, ktorý opisuje stav spínača v prvom uzle. Tento signál je zabalený v CAN správe a je vysielaný len ak sa zmení pozícia spínača (spontánne vysielanie).

Pomocou **CANdb++** editora vytvoríte novú databázu. V databáze vytvoríte správu s menom napr. *Msg1* a identifikátorom *100*, ktorý je vysielaný prvým uzlom. Vytvoríte signál *bsSwitch* na opis pozície spínača, a spojte ho (*link*) so správou *Msg1*. V tomto prípade je postačujúca dĺžka signálu 1 bit, keďže je potrebné prenášať len dva stavy: *On (1)* a *Off (0)*.

Databáza poskytuje prostredníctvom premenných okolia opis I/O rozhrania medzi uzlami a ich perifériami. Každý periférny prvok (spínač, indikátor, posuvný volič atď.) je priradený k premenným prostredia (ku CAPL programu sieťového uzla).

V tomto prípade máme dva periférne prvky: v prvom uzle spínač, v druhom žiarovka. Preto v databáze musia byť vytvorené dve premenné veličiny (obr. 48), napr. *evSwitch* a *evLight*.



Name	Type	Unit	Minimum	Maximum	Initial Value
evLight	Integer		0	1	0
evSwitch	Integer		0	1	0

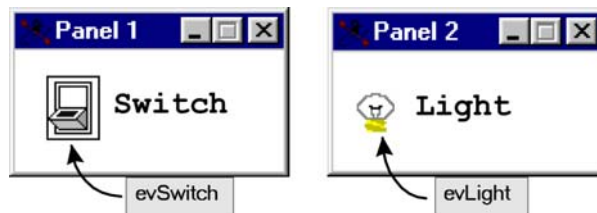
Obr. 48. Premenné v databáze

Uložte (**Save**) databázu napr. pod názvom TOUR.DBC a asociujte ju s vašou prázdnu konfiguráciou. V **Simulation setup** v okne **System view**, možno pridať databázu. V stromovom menu aktuálnej konfigurácie pravým tlačítkom myši klikneme na **Databases** a spustíme príkaz **Add (Pridať)**.

5.6 Tvorba panelov (periférií)



Panel Editor slúži v CANoe na vytvorenie periférií uzlov (**node's periphery**).


Pre každý z dvoch uzlov musí byť vytvorený jeden panel. Prvý panel má vlastné ovládanie (**control**), spínač, druhý má ako indikačnú jednotku žiarovku (obr. 49).



Obr. 49. Panely v CANoe


Panel Editor môžete spustiť cez **toolbar**  (lišta nástrojov). To zaisťuje, že bude k dispozícii databáza s veličinami *evSwitch* a *evLight*.

V **Panel Editor**-e najprv zadáme meno prvého panela (**Options/Panel size, Name, Colors and Fonts**), napr. Panel 1. Na lište **Panel Editor** zvolíme spínač  a umiestnime ho na panel. Spínač možno prepojiť s premennou okolia dvojitým kliknutím a následnou voľbou premennej okolia *evSwitch*. Spínaču možno priradiť popis cez lištu nástrojov prostredníctvom , jeho umiestnením na panel (napr. vľavo od spínača) a konfigurovať text (napr. "Switch") –dvojklikom na objekt popisu. Tiež možno zmeniť veľkosť panela, kliknutím na jeho okraj s následným ťahaním (nevoľte väčšie panely než ako je nevyhnutné - zbytočne zaberajú miesto).

Uložte panel napr. pod názvom SWITCH.CNP. Potom rovnako vytvorte panel pre druhý uzol rovnakým spôsobom. Namiesto spínača sa vloží žiarovka ako zobrazovací prvok. Dosiahne sa to voľbou  na paneli nástrojov a dvojklikom na už vložený prvok sa konfiguruje ako zobrazovací prvok (type-**Display**) s dvoma stavmi (**2 states**). V konfiguračnom okne sa tiež musí zvoliť bitmapový obrázok, ktorý bude použitý ako indikátor. To sa dosiahne voľbou súboru obrázka napr. LAMP_2.BMP z adresára `../DEMO_CAN_CN/AUTOMOT//PANELS/BITMAPS/BMP_2`.

Pred tým ako panel uložíme napr. pod názvom LIGHT.CNP, možno žiarovku pomenovať vložením textu podobne ako v predchádzajúcom prípade.

Ukončením tejto úlohy je integrovanie vytvorených panelov do konfigurácie CANoe (**CANoe configuration**). Vykoná sa to voľbou príkazu **Panel/Configure panels** v konfiguračnom dialógu.

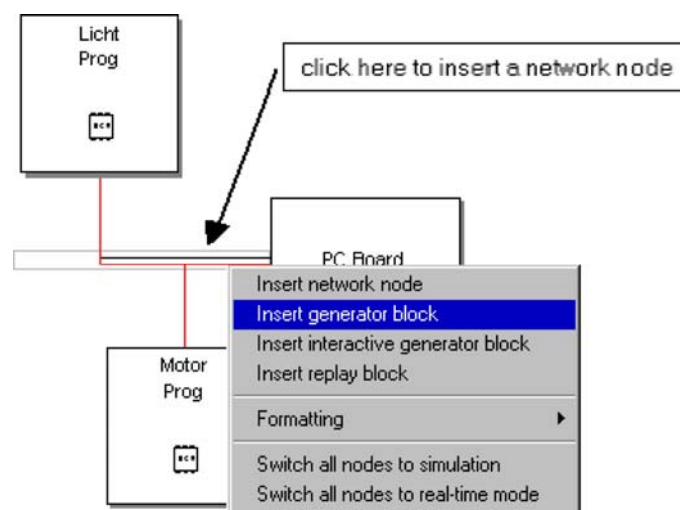
Pridajte súbory **<Add panel...>** SWITCH.CNP a LIGHT.CNP do zoznamu panelov a otvorte panely pomocou tlačidla **<Open all panels>**. Panely si na obrazovke umiestnite podľa vašich potrieb a túto pozíciu uložte pomocou príkazu **Panel/Save panel position**. Uložte konfiguráciu pomocou tlačidla  v paneli nástrojov.

5.7 Tvorba modelov uzlov

Model uzla možno vytvoriť v **Simulation setup**. Model pre prvý uzol musí posielať správu keď je spínač zapnutý a preto nesmie byť vložený do nastavenia merania (**measurement setup**). Pre vloženie nového modelu uzla kliknite na zbernicu v **simulation setup**.

V tomto príklade potrebujete v **simulation setup** dva uzly: prvý uzol je pre spínač, druhý na to reaguje na aktiváciu alebo deaktiváciu žiarovky.

Možno vybrať **configuration dialog** pre dva uzly, stlačením pravého tlačítka myši (obr. 50). Tu sa vloží meno uzla (napr. ECU 1 alebo ECU 2) a priradí meno súboru ku každému z dvoch uzlov (napr. ECU1.CAN alebo ECU2.CAN). Mená uzlov je vidieť v ikonkách uzlov; mená súborov sa odvolávajú na CAPL programy, ktoré simulujú funkciu dvoch uzlov. Dvojitým kliknutím na uzol otvoríme prehliadač CAPL.



Obr. 50. Vloženie uzla v simulation setup

Prvý CAPL program patrí k uzlu, ku ktorému je priradený spínač. Keď sa zmení poloha spínača, program nadobúda novú hodnotu a hneď sa to preukáže na zbernici:

```
// Reaction to change of environment var. evSwitch
on envVar evSwitch {
// Declare a CAN message to be transmitted
message Msg1 msg;
// Read out the value of the light switch,
// Assign to the bus signal bsSwitch
msg.bsSwitch = getValue(this);
// Output message on bus (spontaneous transmission)
output(msg);
}
```

Druhý uzol reaguje na túto správu. CAPL program číta hodnoty signálu zo spínača na zbernici a aktivuje, alebo deaktivuje žiarovku. Hodnota spínača je získaná cez hodnotu signálu na

zbernici. Hodnota premennej *evSwitch* je CAPL programu neznáma. To znamená, komunikácia medzi dvoma uzlami nastane jedine cez CAN zbernicu:

```
// Reaction to receipt of the CAN message M1
on message Msg1 {
// Read out a bus signal and
// set the environment variable
putValue(evLight, this.bsSwitch);
}
```

Teraz spustite meranie v CANoe. Vždy keď aktivujete spínač na paneli 1, žiarovka sa rozsvieti. Vždy keď spínač vypnete, žiarovka zhasne. **Trace window** (Grafické okno) ukazuje komunikáciu zbernice (spontánny prenos správy Msg1 keď sa zmení pozícia spínača) a hodnoty premenných *evSwitch* a *evLight*.

Použitá a odporúčaná literatúra

- [1] Ronald K. Jurgen: Automotive Electronics Handbook (2nd ed.). McGraw-Hill Professional, 1999.
- [2] Bosch: Automotive Handbook (8th ed.). Bentley Publishers, 2011.
- [3] Vector Informatik GmbH: CANoe,DENoe .CAN.LIN.MOST.FlexRay, Manual Version 4.1, Firemná literatúra spoločnosti VECTOR.
- [3] Firemná literatúra spoločnosti VOLKSWAGEN.