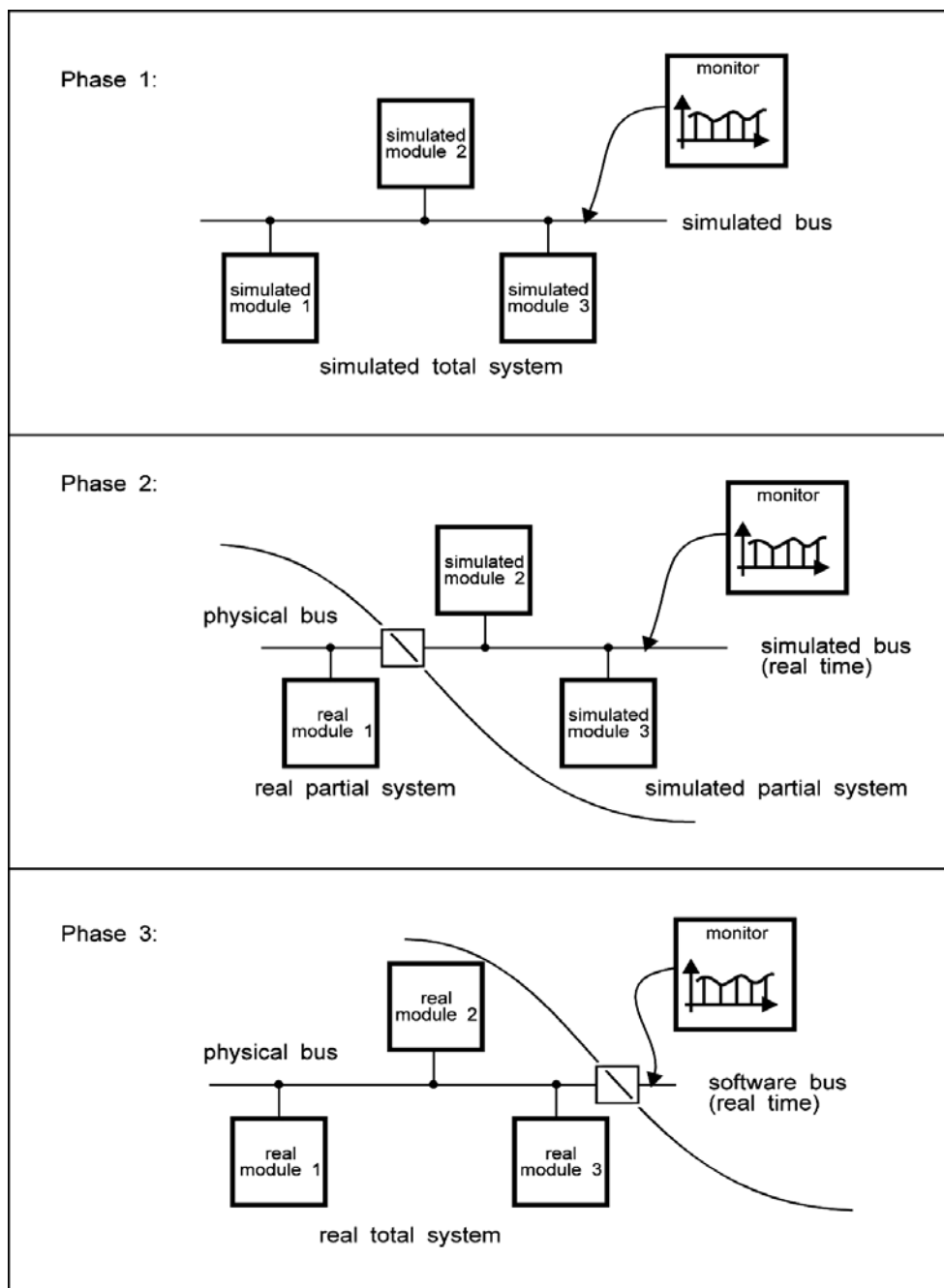


# - CANoe -

## 1.1 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, ktoré sa rozlišujú v troch vývojových stupňoch:



Obr.1: 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 počiatočných 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.

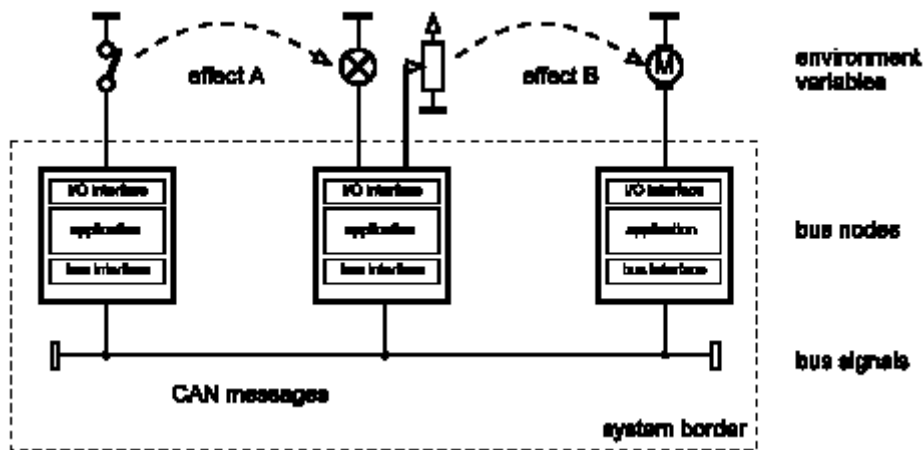
### **Fáza 3:** Integrovanie celého systému.

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 opísované pomocou premenných okolia. CANoe rozlišuje diskrétne a spojité premenné. Pozícia prepínača môže byť reprezentovaná ako diskrétna premenná 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.2 ozrejmuje funkcie, ktoré CANoe poskytuje pre simuláciu a testovanie systémov na CAN zbernici.



Obr.2 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 spojitkej 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.

### 1.2.1 Menu

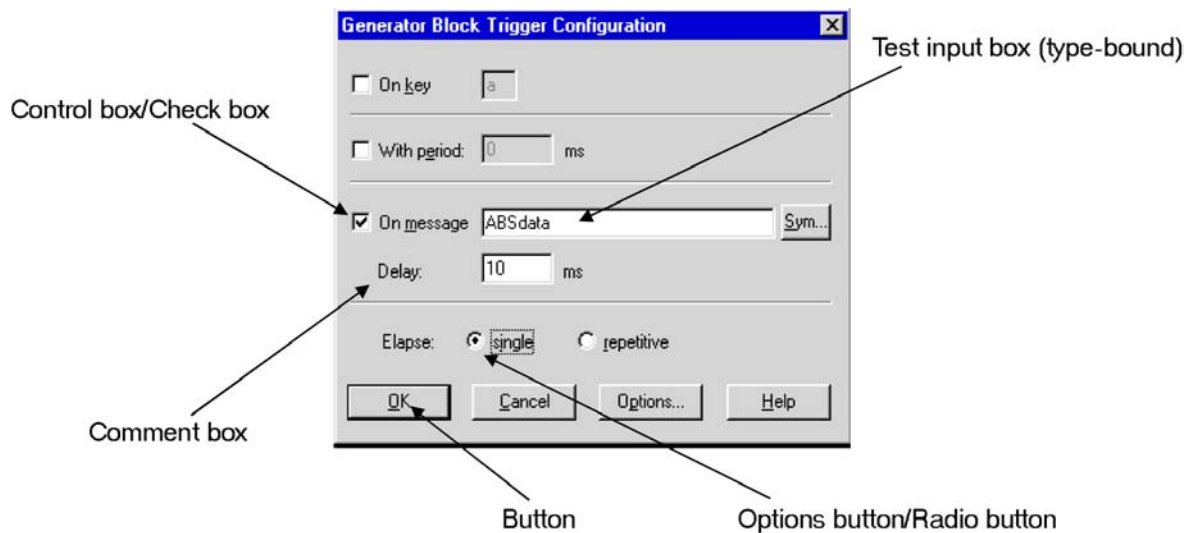
CANoe sa ovláda použitím základného menu. Individuálne príkazy sú podrobne popísané v online pomoci.

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 dvojitým kliknutím na aktívny blok, alebo stlačením <ENTER>.

### 1.2.2 Dialogy

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



Obr. 3: Dialogy

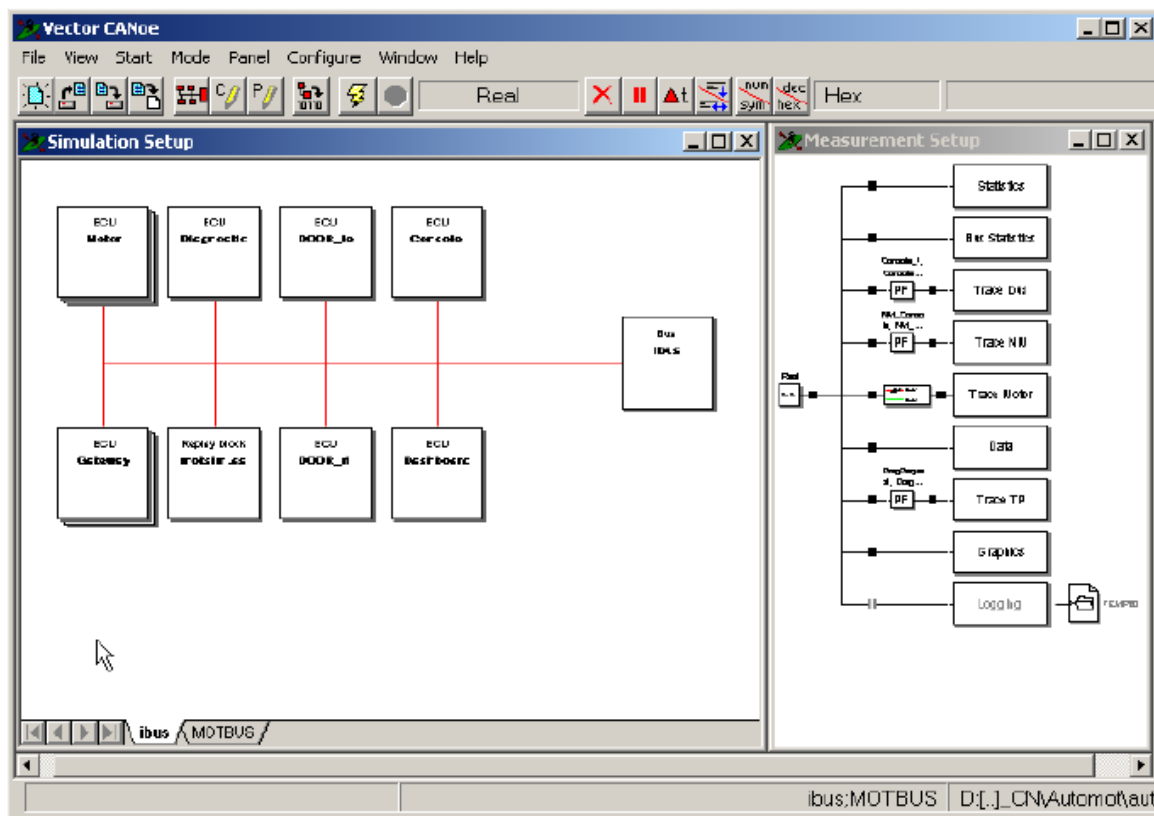
- Comment box** Táto časť hovorí užívateľovi aké sú vstupné dáta. Políčka sú v pasívnom móde a nereagujú na kliknutie myšou. Nemôžu byť taktiež prístupné z klávesnici.
- Text input box (type-bound)** alfanumerické pole, napr. na zadanie názvu súboru. Číselné pole, napr. na zadanie čísel v celočíselnom 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** (zaškrtávacie 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, ovplyvníme 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.

### 1.2.3 (Measurement and Simulation Setup)

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



Obr.4: Nastavenie merania a simulácie v CANoe

#### 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 dvojité kliknutím ľavého tlačidla myši.

#### Ovládanie klávesnicou

Keď je aktívne okno nastavenia merania alebo okno nastavenia simulácie, voľbou <Tab>, <Backspace> alebo ak je aktívovaný 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.

### 1.3 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 vysielача, tak aj prijímača. 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.

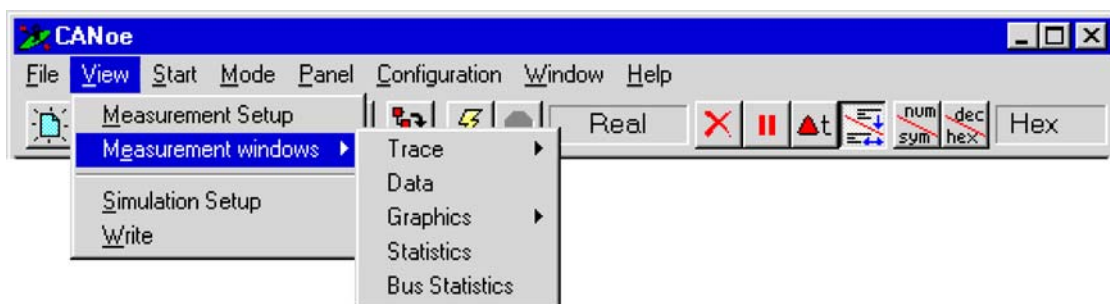
V zložitých reálnych systémoch sa predpokladajú obidve úlohy CANoe. Program môžete používať ako zdroj dát pre prenos do iných riadiacich jednotiek, ale simultánne ho môžete použiť pre sledovanie, vytváranie záznamov hlásení a vyjadrovanie vlastností prenosu dát po CAN zbernici.

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

#### 1.3.1 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. 5: View menu

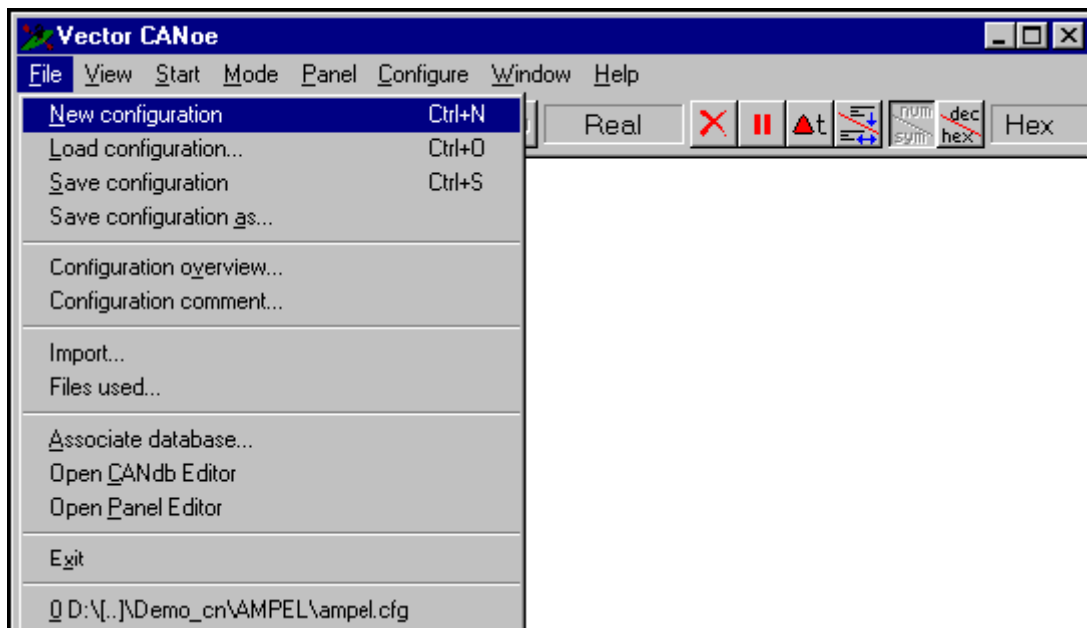
V okne **simulation setup** je graficky znázornený celý systém s CAN zberniciu a sieťovými uzlami. Simulovaná zbernica je reprezentovaná červenou horizontálnou čiarou. Čierna čiara, ktorá je pod ňou symbolizuje reálnu zbernicu. Dve zbernice sú navzájom prepojené prostredníctvom PC karty. Pre prenos dát z CANoe na zbernicu, vložte blok prenosu v

simulačnom nastavení a tento musí byť prepojený červenou čiarou. Blokova 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 prepojujúce 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 vkladať ďalšie funkčné bloky pre manipuláciu s tokom dát (filtrovanie, bloky opakovania a generovania alebo užívateľom definované bloky programu napísané pomocou CAPL).

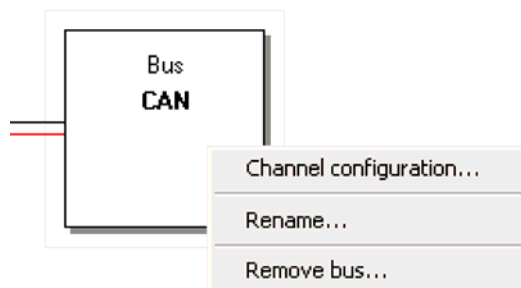
Presvedčte sa, že nasledujúce oboznámenie začínate s novou konfiguráciou voľbou položky menu **File | New configuration**. Spustí sa sprievodca nastavenia simulácie (Simulation Setup), ktorý pre dané oboznámenie nepotrebujete. Preto tohto sprievodcu ukončíte voľbou **Cancel**.



Obr. 6: File/New configuration

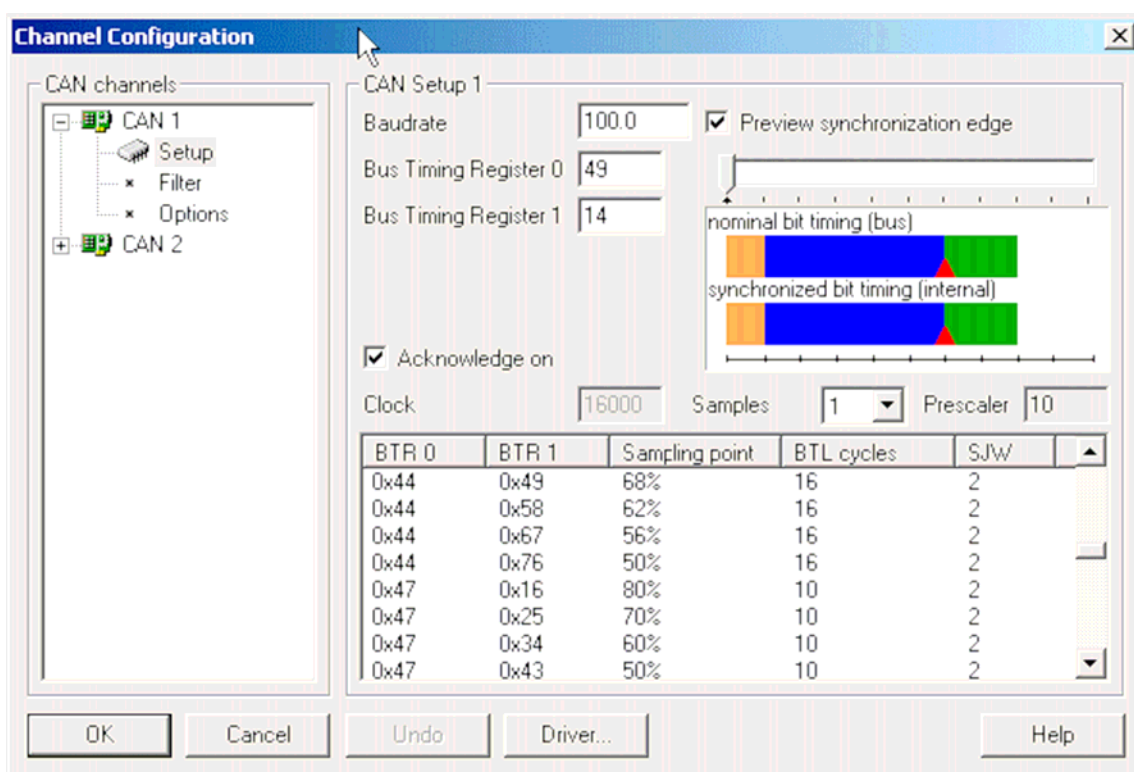
### 1.3.2 Nastavenie CAN zbernice

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 rolovacía ponuka 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.7: Rolovacie menu symbolu zbernice

Vyberte **Channel configuration**



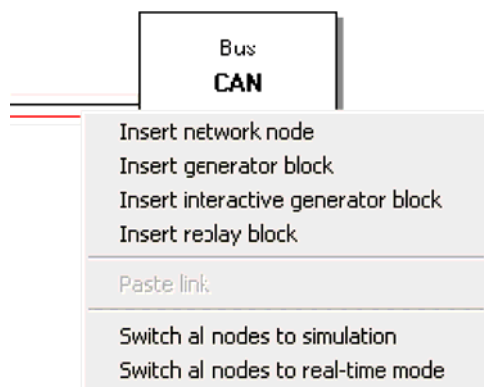
Obr.8: Konfigurácia parametrov zbernice

### 1.3.3 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.

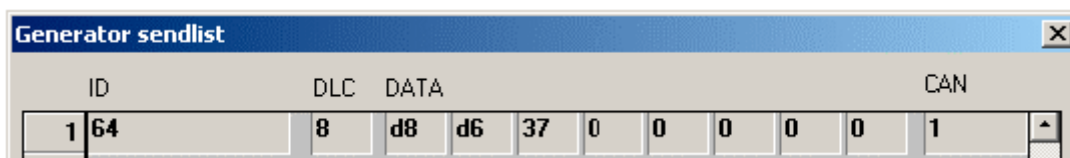
Dáta ktoré budeme prenášať musíme najprv vytvoriť a to pomocou generátora. Ten vytvoríme v okne **Simulation setup** kde kliknutím pravým tlačidlom myši na červenú čiaru v rolovacom menu zvolíme **insert generator block**.





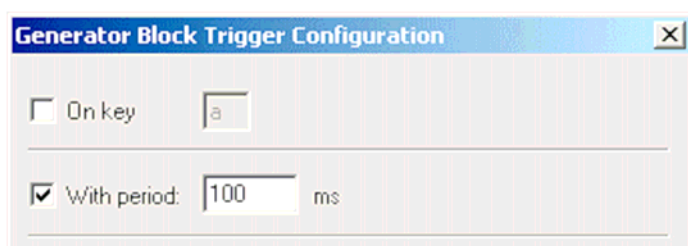
Obr.9: Symbol zbernice v okne **Simulation setup** s kontextovým menu

Konfiguráciu vyvoláme stlačením pravého tlačidla myši na generátore a výberom konkrétneho nastavenia . Najprv si nastavíme **transmit list**. 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. 10: Transmit list generátora

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




Obr.11: Spúšťanie bloku generátora

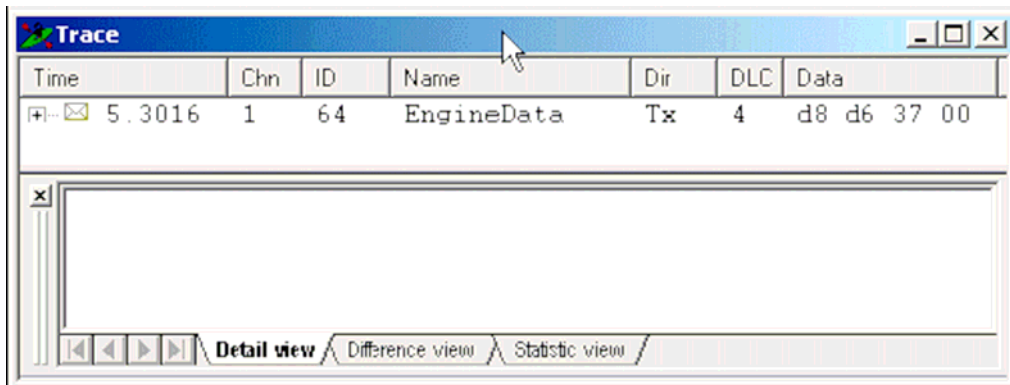
Pred tým ako začneme merať je potrebné si konfiguráciu vykonanú v tomto bode uložiť a to príkazom menu **File/Save configuration**. Tým je možné zvolenú konfiguráciu kedykoľvek vyvolať.

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 rolovaciego 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** 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.12: 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 vyzerat' tak ako je to znázornené ďalej.

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. 13. 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 rozbaľovcích kontextových menu sú nevyrazne 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 zložité) v programovacom jazyku CAPL.

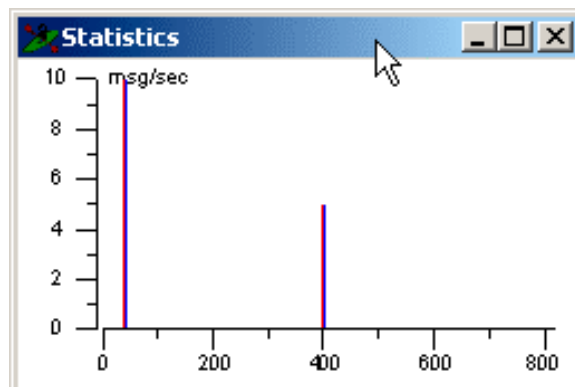
### 1.3.4 Okná vyhodnocovania

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

Povieme 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.

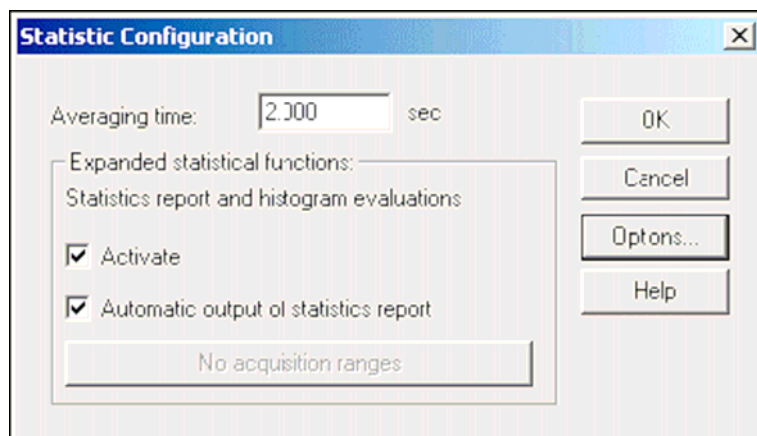
Ďalej, štyri tlačidlá na pravej strane panela nástrojov sa tiež môžu použiť na konfiguráciu **Trace window**. Napríklad, s tlačidlom  je možné meniť **stationary mode** (statický mód) na **scroll mode** (rolovací mód), v ktorom každá prichádzajúca správa je zapísaná v novom riadku. Toto tlačidlo  umožňuje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

Okno štatistických údajov (**Statistics window**) tiež poskytuje informácie o zbernici. 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.

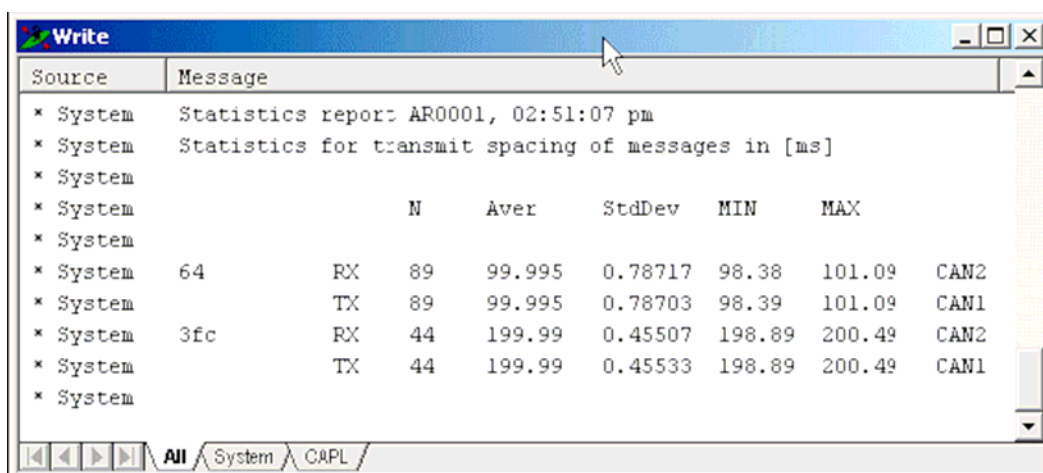


Obr.14: Štatistické okno

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** čím dosiahneme po skončení merania zobrazenie štatistickej správy vo **Write window**.



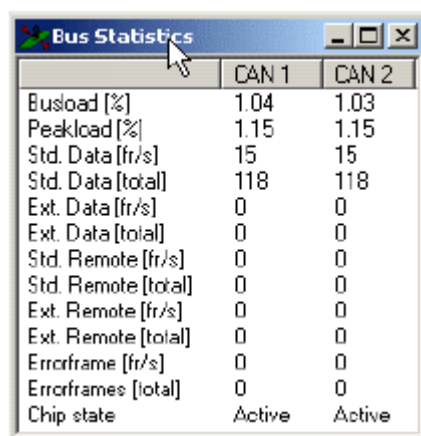
Obr.15: Aktivácia štatistickej správy



Obr.16: 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**. 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%.




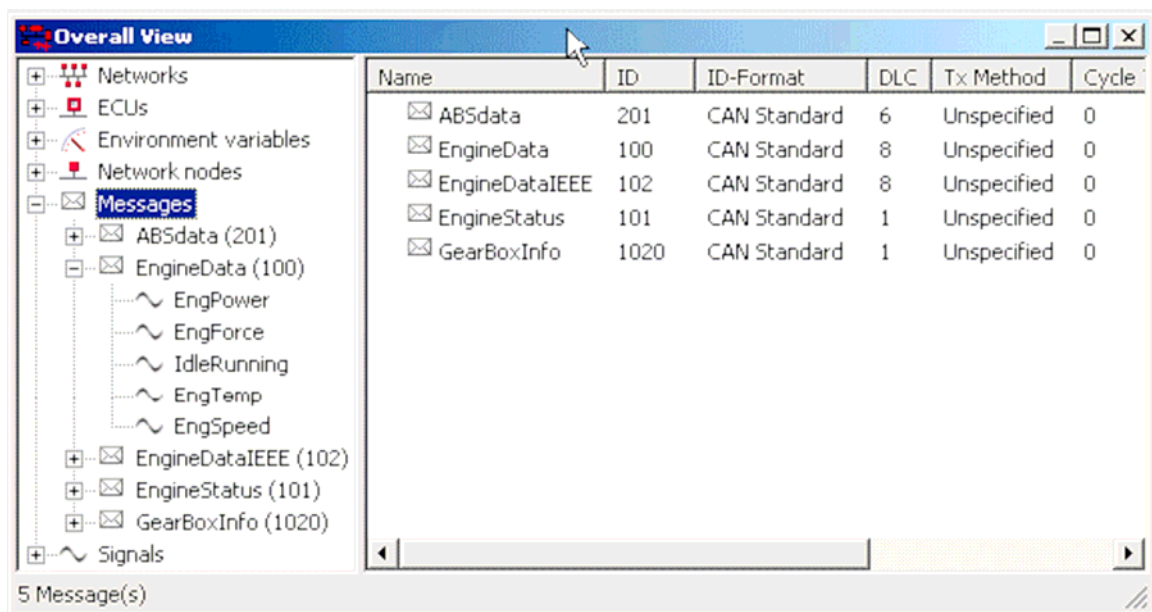
Obr. 17 Okno štatistiky zbernice (**Bus Statistics**)

### 1.3.5 Práca so symbolickými dátami

Prvotným záujmom analýzy CAN systémov (okrem informácií v z'ahovaným 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  tlačidla na panele nástrojov. Otvorí sa CANdb++ editor a obsah databázy MOTBUS.DBC sa zobrazí v okne **Overall View**




Obr.18: 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.

### 1.3.6 Analýza hodnôt signálu v okne Data Window

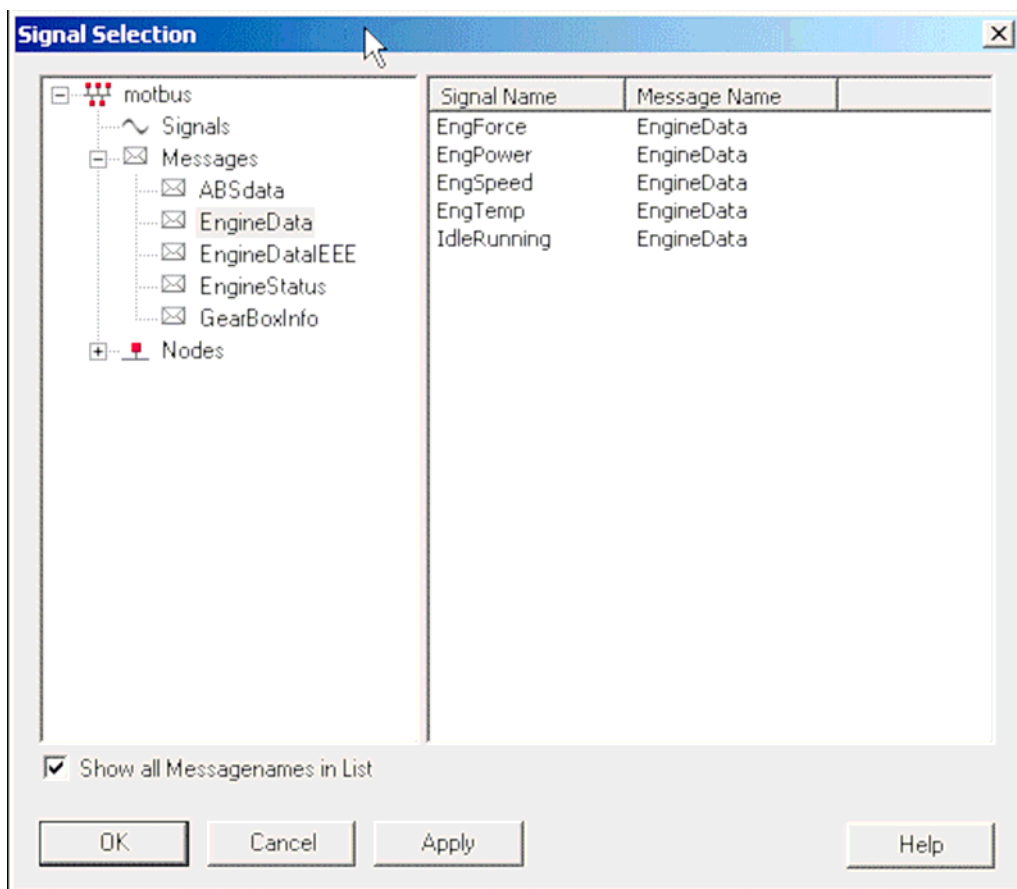
Besides the use of symbolic message names, the associated database can also be used to analyze signal values. The purpose of the Data window is to assist in the study of momentary signal values.

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.

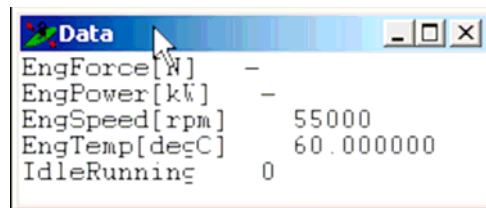
Pri riešení tejto úlohy postupujeme nasledovne. Najprv otvoríme rolovacie menu **Data window** v ktorom zvolíme **Configuration...**. Zobrazí sa nám konfiguračný dialóg ktorý obsahuje na začiatku prázdne okno správ. Pomocou tlačidla **New signal** sa spustí signálový prieskumník ktorý umožní vybrať požadovaný 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.19: Prieskumník signálov

Po ukončení konfigurácie je vidieť že názov zvoleného signálu zo signálového prieskumníka je zobrazený v **Data window**. 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**.



Parameter	Value
EngForce[W]	-
EngPower[kW]	-
EngSpeed[rpm]	55000
EngTemp[decC]	60.000000
IdleRunning	0

Obr.20: Dátové okno

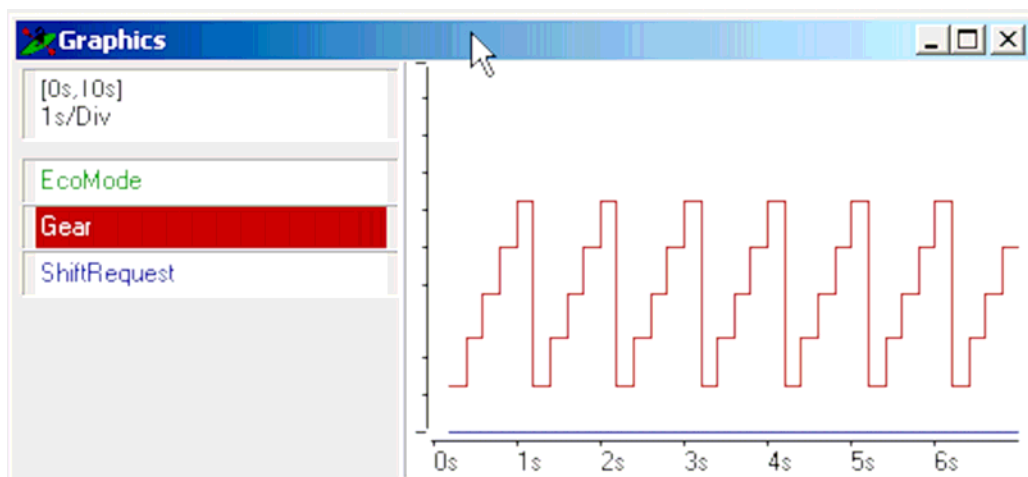
Pomocou prevodového vzorca obsiahnutého v databáze je rýchlosť motora zobrazená v otáčkach, zatiaľ čo teplota je v stupňoch Celzia. Hodnota všetkých troch signálov zostáva po celý čas konštantná pokiaľ je správa vždy prenášaná s rovnakými data bytmi D8, D6, 37 a 0.

### 1.3.7 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**. **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

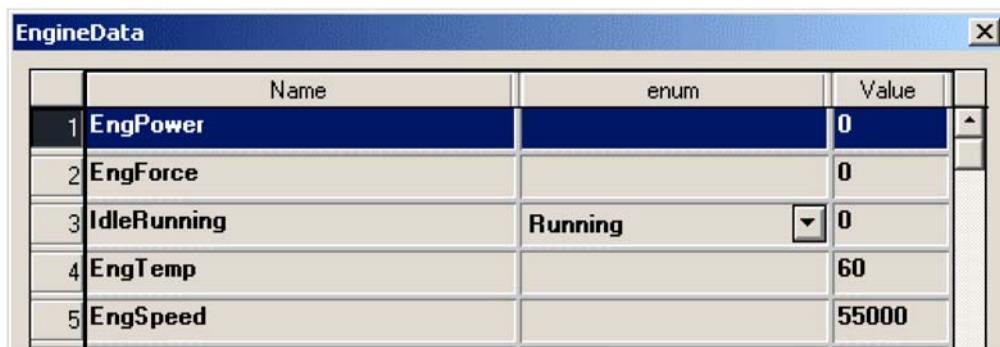


Obr.21: Okno Grafu

### 1.3.8 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 kolonke. Čiže správu možno zadať priamo z databázy použitím tlačidla **Symbol...** (**Simulation 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. 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. 22: Hodnoty v bloku generátora

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

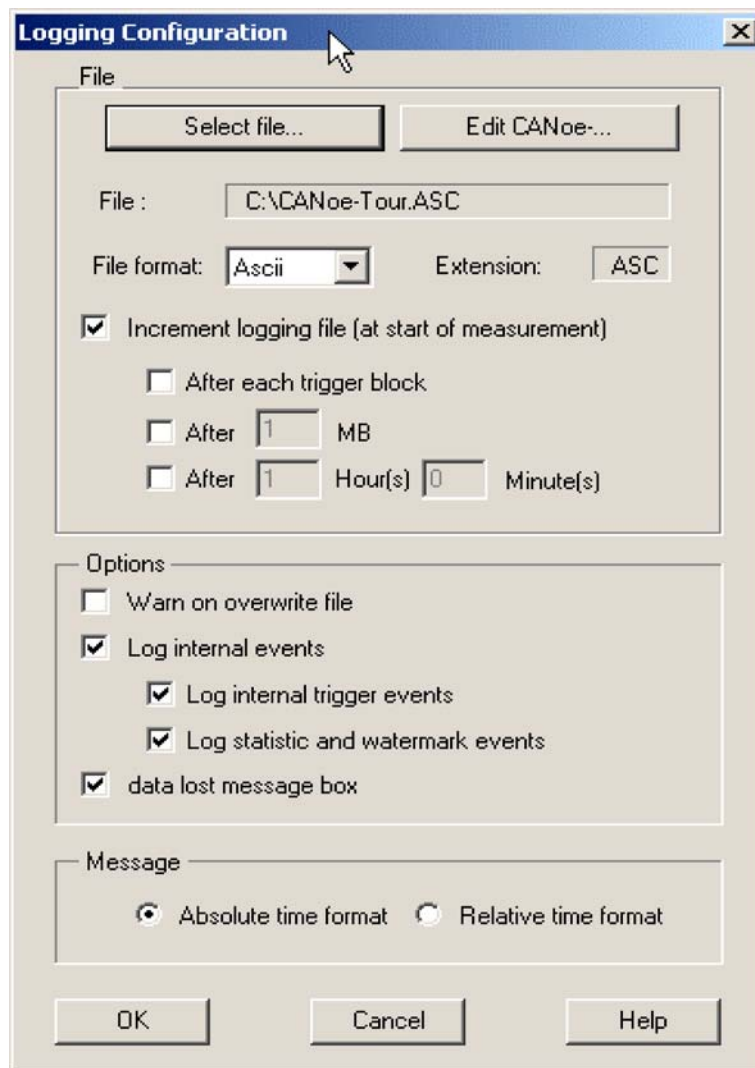
### 1.3.9 Záznam merania

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**. Tu možno zadať meno pre záznam merania, a tiež jeho formát. Vyberte formát ASCII.





Obr. 23: 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.

Popis ako definovať spúšťacie podmienky a časové okná ja uvedené v časti 2.7.1.1. Na zápis merania je postačujúce zmeniť režim zo **Single Tigger** na **Entire Measurement** v dialógu **trigger configuration**.

Kliknutím na OK opustiť dialóg a potom spustiť 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**.

### 1.3.10 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, načítať 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.

### 1.3.11 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 („Panels“) 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.

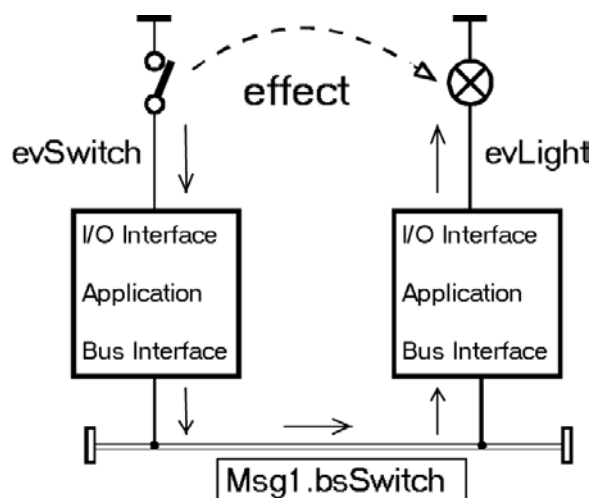
### 1.3.12 Simulácia distribuovaných systémov v CANoe

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, prvý uzol informuje o tejto akcii druhý uzol. Druhý uzol potom aktivuje indikačnú žiarovku ako svoju perifériu.




Obr. 24 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

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

### 1.3.12.1 Tvorba databázy

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

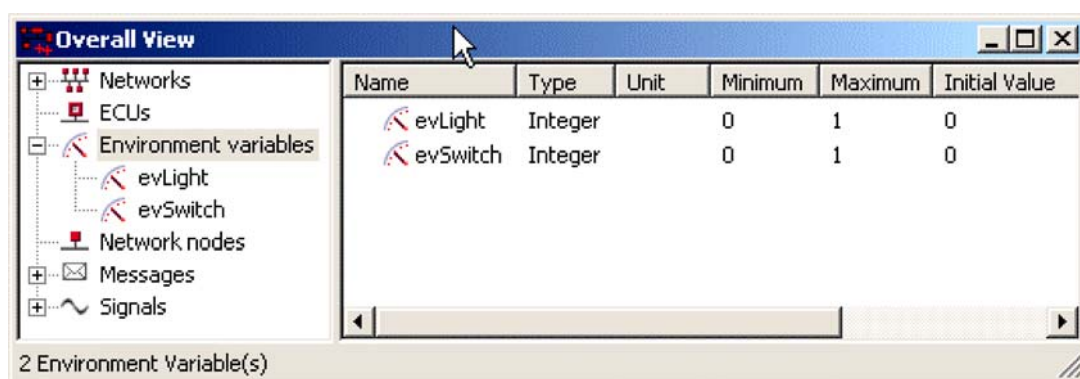
- výmena informácií medzi dvoma uzlami cez CAN zbernicu
- I/O rozhranie k perifériám, t.j. prepojenie medzi každým uzlom a jeho vstupnou a výstupnou jednotkou

Databázová správa a objekty signálov sú vhodné a schopné na opis výmeny informácií cez CAN zbernicu. Jednoduchá funkcionálna vpríklade 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, napr. *evSwitch* a *evLight*



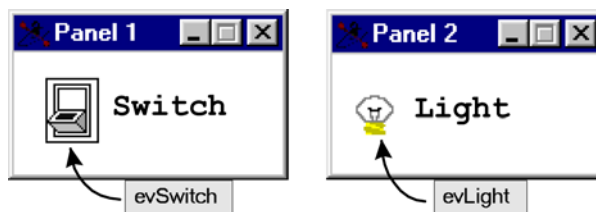
Obr. 25: Premenné v databáze

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)**.

### 1.3.12.2 Tvorba panelov (periférií)

Panel Editor slúži v CANoe na vytvorenie periférií uzlov (**node's periphery**). V kapitole 5 je k tomuto editoru detailný úvod.



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




Obr. 26: 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.

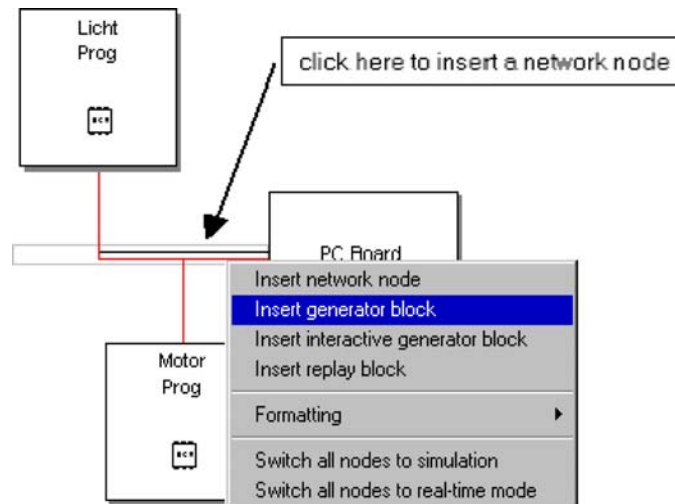
Pred tým ako panel uložíme napr. pod názvom LIGHT.CNP, možno žiarovku pomenovať vložením textu podobne ako v predchádzajúcom prípade.

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.

### 1.3.12.3 Tvorba modelov uzlov

Model uzla možno vytvoriť v **Simulation setup**. Model pre prvý uzol musí posielat' 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**.



Obr. 27: Vloženie uzla v simulation setup

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

Možno vybrať **configuration dialog** pre dva uzly, stlačením pravého tlačítka myši. 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.

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*.