

# **TECHNICKÁ UNIVERZITA V LIBERCI**

*Fakulta mechatroniky, informatiky a mezioborových studií*

*Studijní program: B2646 - Informační technologie*

*Studijní obor: 1802R007 - Informační technologie*

## **Automatická kalibrace řídicí jednotky airbagu**

## **Automatic calibration of the airbag control unit**

### **Bakalářská práce**

*Autor: Michal Martin*

*Vedoucí práce: Ing. Josef Grosman*

*Konzultant: Ing. Michal Vojtěchovský*

*V Liberci 13.4. 2012*

## Zadání bakalářské práce

<b>Příjmení a jméno studenta</b> (osobní číslo - nepovinné)	<i>Martin Michal</i>
<b>Zkratka pracoviště</b>	<i>MTI</i>
<b>Datum zadání BP/DP</b>	<i>14.10.2011</i>
<b>Plánované datum odevzdání</b>	<i>18.5.2012</i>
<b>Rozsah grafických prací</b>	<i>dle potřeby dokumentace</i>
<b>Rozsah průvodní zprávy</b>	<i>cca 40 – 50 stran</i>
<b>Název BP/DP (česky)</b>	<i>Automatická kalibrace řídicí jednotky airbagu</i>
<b>Název BP/DP (anglicky)</b>	<i>Automatic calibration procedure for Airbag Central Unit</i>
<b>Zásady pro vypracování BP/DP</b> (text nijak neformátujte, pouze očísľujte jednotlivé body .. 1) ... 2) ... atd. a každý bod uveďte jako nový odstavce):	
<i>1, Seznamte se s procesem kalibrace řídicí jednotky airbagu pomocí aplikace C-tool pro projekt Blue car Bollore</i>	
<i>2, Navrhněte a naprogramujte vlastní řešení, které bude umožňovat automatický proces kalibrace řídicí jednotky airbagu tak aby odpovídal specifikaci projektu Blue car Bollore</i>	
<i>3, Otestujte program na sadě vzorků a spočítejte zrychlení procesu automatické kalibrace</i>	
<b>Seznam odborné literatury</b> (text nijak neformátujte, pouze každou položku uveďte jako nový odstavce):	
<i>[1] Dokumentace k použitým vývojovým prostředím (C-tool, C++,...)</i>	
<i>[2] Dokumentace k aplikaci C-tool</i>	
<i>[3] Specifikace procesu kalibrace řídicí jednotky airbagu pro Blue car Bollore</i>	
<b>Vedoucí BP/DP</b>	<i>Ing. Josef Grosman</i>
<b>Konzultant BP/DP</b> (u externích pracovníků uveďte plný název pracoviště – firmy)	<i>Ing. Michal Vojtěchovský</i> <i>TRW Automotive Czech s.r.o.</i> <i>Aftermarket Operations Frýdlant</i>

## **Prohlášení**

*Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon*

*č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.*

*Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.*

*Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.*

*Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.*

*Datum*

*Podpis*

## ***Poděkování***

*Na tomto místě bych rád poděkoval vedoucímu mé bakalářské práce Ing. Josefu Grosmanovi za rady, připomínky a čas, který mé práci věnoval. Také bych chtěl poděkovat Ing. Michalu Vojtěchovskému, za poskytnuté konzultace. Dále bych chtěl poděkovat firmě TRW, že mi umožnila udělat bakalářskou práci u nich a vyšla mi vstříct, když jsem potřeboval.*

*A chtěl bych také poděkovat svým rodičům a celé rodině za podporu při studiu.*

## **Abstrakt**

*Práce shrnuje postup vytvoření skriptu pro automatickou kalibraci centrální jednotky airbagu. Práce je rozvržena do čtyř hlavních částí. V první části se nachází úvod, který shrnuje veškeré informace o práci. Dále se zde nacházejí informace o firmě, centrální jednotce airbagu a také popis CAN komunikace. V další části je popsáno seznámení se s kalibrací centrální jednotky airbagu. Tato část obsahuje i obecný popis kalibrace vykonávané operátorem. Ve třetí části se nachází popis návrhu a naprogramování vlastního řešení. V této části jsou umístěny ukázky kódů, které byly použity ve skriptu. Poslední částí práce je měření zrychlení, kterého docílila automatická kalibrace. Práci uzavírá závěr, ve kterém je zhodnocena celá práce.*

## **Abstract**

*The work summarizes how to create a script for automatic calibration of the central unit airbag. The work is divided into four main parts. The first part is an introduction which summarizes all information about the work. Referred to herein located information about the company, Project Blue Car, Central airbag unit and a description of CAN communication. The next section describes the familiarity with calibration of the central unit airbag. This section contains a general description of the calibration performed by the operator. The third part is about the design and programming of my skript. In this section are located examples of codes that were used in the script. The last part of this work is to measure acceleration, which achieved automatic calibration. The work is closed with the conclusion in which the evaluation of the whole work.*

## **Klíčová slova**

*ECU, DTC, airbag, CAN, Blue Car*

## **Keywords**

*ECU, DTC, airbag, CAN, Blue Car*

## Obsah

<i>Prohlášení</i> .....	3
<i>Poděkování</i> .....	4
<i>Abstrakt</i> .....	5
<i>Seznam obrázků</i> .....	8
<i>1. Úvod</i> .....	9
<i>2. Projekt Blue Car</i> .....	10
<i>3. Firma TRW</i> .....	11
<i>4. Centrální jednotka airbagu</i> .....	12
<i>5. CAN</i> .....	12
<i>5.1 Potřeba sériové komunikace v automobilech</i> .....	12
<i>5.2 Základní vlastnosti protokolu CAN</i> .....	13
<i>6. Seznámení se s kalibrací</i> .....	16
<i>6.1 Okno Process Values</i> .....	19
<i>6.2 Okno RxTxMon</i> .....	20
<i>6.3 Hlavní okno aplikace</i> .....	20
<i>6.4 Postup kalibrace</i> .....	21
<i>7. Návrh a naprogramování skriptu</i> .....	24
<i>7.1 Časová nenáročnost</i> .....	25
<i>7.2 Seznámení se s jazykem skriptu</i> .....	25
<i>7.3 Návrh skriptu</i> .....	30
<i>7.4 Programování</i> .....	31
<i>8. Měření zrychlení kalibrace</i> .....	37
<i>9. Závěr</i> .....	38
<i>Reference na použitou literaturu</i> .....	40

**Seznam obrázků**

<i>Obr. 1 Elektromobil firmy Bolloré.....</i>	<i>10</i>
<i>Obr. 2 Toleranční pásmo úrovní logických stavů na sběrnici CAN.....</i>	<i>14</i>
<i>Obr. 3 Jednotka simulující ECU.....</i>	<i>17</i>
<i>Obr. 4 Aplikace C-Tool.....</i>	<i>18</i>
<i>Obr. 5 Ukázka Process Values Window.....</i>	<i>19</i>
<i>Obr. 6 Ukázka RxTxMon.....</i>	<i>20</i>
<i>Obr. 7 Ukázka hlavního okna aplikace.....</i>	<i>21</i>
<i>Obr. 8 Vývojový diagram kalibrace.....</i>	<i>22</i>
<i>Obr. 9 Ukázka souboru *.db.....</i>	<i>26</i>



## 1. Úvod

*V automobilovém průmyslu se stále snaží snížit náklady na výrobu automobilu. Díky tomu jsou nové automobily skládány ze součástí ze starších, nebo podobných modelů. Z tohoto důvodu se spíše vyvíjejí zařízení na upravení kalibrace a přepisování licencí těchto dílů. To je případ mé bakalářské práce. Moji bakalářskou práci jsem dělal pro firmu TRW ve Frýdlantu. V této firmě jsem měl za úkol vytvořit automatickou kalibraci centrální jednotky airbagu pro projekt Blue car Bolloré. Centrální jednotka bude využita v elektromobilu firmy Bolloré. Tato firma sídlící ve Francii se rozhodla vyrobit vlastní elektromobil. Tento elektromobil je postaven pro městský provoz a je určen pro veřejnost. Automatická kalibrace, by měla především odstranit lidský faktor. Jedná se o napsání skriptu, který provede automatickou kalibraci centrální jednotky. V automobilovém průmyslu je snaha zautomatizovat veškeré postupy a tudíž odstranit veškeré faktory, které mohou způsobovat chyby. Dalším aspektem je i zrychlení kalibrace, z čehož vyplývá i více překalibrovaných jednotek za směnu. Kalibrace se provádí v aplikaci C-Tool. Tato aplikace byla vytvořena pro diagnostiku jednotek. V tomto programu je uživatelské menu pro diagnostiku centrální jednotky airbagu. Jednotlivé úkoly jsou zde řešeny přes sérii tlačítek a výběrových menu. Díky flexibilitě této aplikace je zde možná i kalibrace jednotky. Tato aplikace umožňuje spouštět skripty, které mohou obsahovat kalibraci. Jedná se o řazení jednotlivých úkolů za sebe, které se mají vykonat. Mezi důležité kroky patří především upravení citlivosti senzorů a změna systému. Systém je třeba změnit, jelikož ECU (vestavěný počítač pro řízení automobilových systémů) má vlastní systém. Tyto systémy musí být kompatibilní, proto je zapotřebí změnit systém u jednotky airbagu. Dalšími kroky kalibrace jsou přepsání licenčních údajů, vymazání chyb a kontrola přepsaných údajů. Veškeré kroky kalibrace jsou vypisované a lze tedy uložit informace o změně jednotky. Uložené informace jsou použity k případné reklamaci.*

## 2. Projekt Blue Car

*Projekt Blue Car je pod záštitou firmy Bolloré. Firma Bolloré je jedním z předních evropských výrobců autobaterií a baterií do elektromobilů. Proto se tato firma rozhodla vytvořit si vlastní elektromobil. Koncepce elektromobilu se objevila už v roce 2006 na autosalónu v Ženevě, kde Bolloré chtělo ukázat potenciál lithium-iontové baterie. Tato koncepce je navržena především pro městský život, proto se také poprvé objevilo v Paříži. Došlo k tomu v prosinci minulého roku a to v počtu 250 kusů. Firma Bolloré ovšem plánuje přidávat každý měsíc 300 nových vozů. Tento elektromobil je zkonstruován výhradně pro veřejnost, tudíž by se mělo v Paříži objevit až 1120 stanic pro parkování a dobíjení baterie.*

*Celý projekt spočívá ve veřejném půjčování elektromobilů. Princip zapůjčení elektromobilu je jednoduchý. Lidé s přístupovou kartou od automobilu si ho mohou zapůjčit, v libovolné stanici elektromobilů.. Pomocí čipové karty mohou s tímto vozem odjet a po vrácení na libovolnou stanici, je jim stržena částka za půjčení vozu. Tohoto způsobu využívá v Paříži čím dál více lidí. Proto chce firma Bolloré doplňovat počty vozů až do června tohoto roku.*



Obr 1 Elektromobil firmy Bolloré

*Vůz je navržen na podvozku z automobilu Smart. Hlavním pohonným článkem je elektromotor. Elektromobil používá 30 kW elektromotor napájený 28 kWh lithium-iontovým akumulátorem umístěným ve středu vozu pod sedadly a před zadní nápravou. Tato baterie umožňuje dojezdovou vzdálenost až 250 km s maximální rychlostí 130 km/h.*

### **3. Firma TRW**

*Firma TRW Automotive je jedním z předních světových dodavatelů autopříslušenství, a mezi dodavateli technologií aktivní a pasivní bezpečnosti pro výrobce vozidel po celém světě představuje špičku.*

*Společnost TRW Automotive ve Frýdlantu je jedním z předních výrobců brzdových kotoučů pro náhradní spotřebu, renovace diskových brzd, renovace dílů řízení. Tyto díly jsou dodávány výrobcům vozidel, ale také autorizovaným servisům a prodejčům automobilových dílů.*

*V rámci splňování evropských standardů a norem zahájil TRW Automotive ve Frýdlantu renovaci diskových brzd pro náhradní spotřebu. Tyto renovované brzdy jsou testovány za stejných podmínek jako nové diskové brzdy. V současnosti začínají být renovované brzdy dodávány i výrobcům automobilů pro jejich náhradní spotřebu.*

*V roce 2004 byl v závodě ve Frýdlantu společností TRW Automotive rozšířen výrobní program zaměřený na renovaci dílů řízení a to jak manuálních tak i s posílením. Další součástí tohoto programu je i renovace hydraulických pump řízení. Vzhledem k tomu, že se jedná o bezpečnostní díly, je každý výrobek po ukončení renovace testován dle stejných technických specifikací jaké platí v prvovýrobě. TRW Automotiv své produkty dodává, jak přímo výrobcům automobilů do jejich značkových servisů, tak i na nezávislý trh s náhradními díly.*

## 4. Centrální jednotka airbagu

Nejdůležitější součástí řídicího systému vozu, je z pohledu bezpečnosti posádky, řídicí jednotka airbagu. Ta disponuje snímači kinematických veličin, jako jsou víceosé akcelerometry a gyroskopy. Díky těmto senzorům pozná, že došlo k nárazu, ať už do překážky při pohybu nebo naopak došlo-li k nárazu od jiného vozidla. Dále sleduje pohyb vozidla po nárazu a na základě rychlého vyhodnocení signálů ze snímačů rozhodne o tom, jaké vaky aktivuje a v jakém pořadí. Mezi akční členy patří kromě vaků (přední, boční, hlavové, kolenové a další) také předpínače bezpečnostních pásů a aktivní hlavové opěrky. Při nárazu řídicí jednotka vyšle na sběrnici tzv. „Crash signál“, který způsobí reakci ostatních částí řídicího systému. Dojde k odpojení palivového čerpadla, zapnutí výstražných ukazatelů, odemčení vozidla, rozsvícení vnitřního osvětlení a v některých případech, pokud je touto funkcí vozidlo vybaveno, na závěr dojde k odpojení baterie. Čímž se odstaví všechny elektrické systémy, ale zabrání se případnému zkratu, který by mohl způsobit požár havarovaného vozu.

## 5. CAN

V automobilovém průmyslu se s velkou efektivitou využívá sběrnice typu CAN. Projekt Blue Car není výjimka a také jednotlivé jednotky spolu komunikují přes CAN sběrnici. Jednotka airbagu je připojena CAN sběrnici k centrální jednotce automobilu, ta zajišťuje veškerou komunikaci.

### 5.1 Potřeba sériové komunikace v automobilech

Většina vozidel je vybavena celou řadou elektronických řídicích systémů. Vývoj

*elektroniky v automobilovém průmyslu je podmíněn jak vzrůstajícími nároky uživatelů, tak také tlakem jednotlivých vlád na neustálé snižování spotřeby zdrojů a snahy snížit vypouštěné emise do ovzduší.*

*Komplexnost využívaných funkcí implementovaných v těchto nejrůznějších systémech si vynutila potřebu vzájemné komunikace mezi těmito systémy. V konvenčních systémech je pro každý přenášený signál vyhrazena jedinečná přenosová linka, což se ale pro velký počet přenášených signálů stává z finančního hlediska neúnosné. Navíc to přináší mnohé komplikace vyplývající z takto vysokého počtu vodičů určených pro přenos dat. Toto vyřešila sběrnice CAN.*

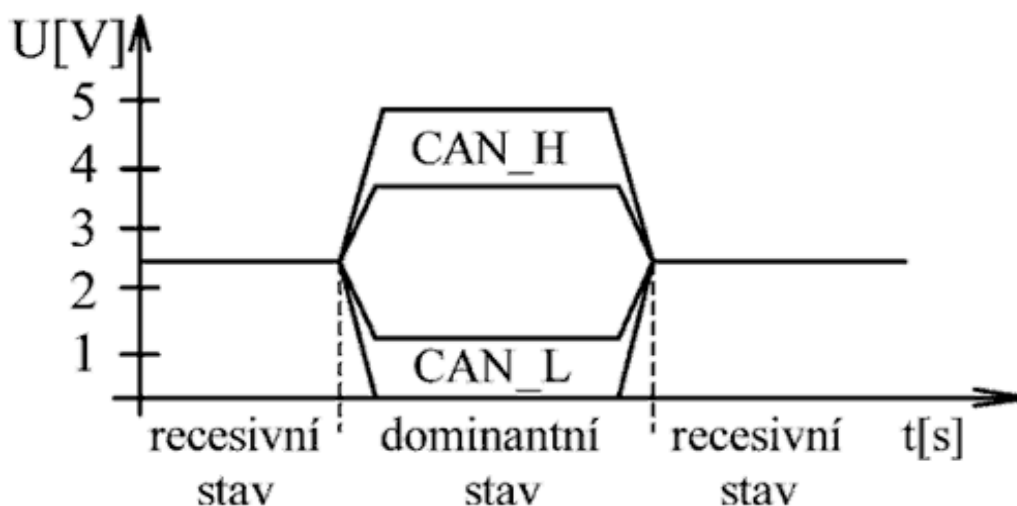
*Veškeré jednotky, které mají potřebu komunikovat ať už mezi sebou, či s jednotlivými senzory zajišťujícími sběr informací, jsou propojeny navzájem právě pomocí sběrnice CAN. Účelem použití této sběrnice v automobilovém průmyslu je zajištění komunikace mezi jednotlivými jednotkami tak, aby nedocházelo k velkému zatížení centrálního procesoru.*

## **5.2 Základní vlastnosti protokolu CAN**

*CAN je sériový komunikační protokol umožňující distribuované řízení systémů v reálném čase s vysokou mírou zabezpečení proti chybám. Jedná se o protokol typu multi-master, kde každý uzel sběrnice může být master a řídit tak chování jiných uzlů. Není tedy nutné řídit celou síť z jednoho nadřazeného uzlu, což přináší zjednodušení řízení a zvyšuje spolehlivost, (při poruše jednoho uzlu může zbytek sítě pracovat dál). Pro řízení přístupu k médiu je použita sběrnice s náhodným přístupem, která řeší kolize na základě prioritního rozhodování. Po sběrnici probíhá komunikace mezi uzly pomocí zpráv (datová zpráva a žádost o data), a management sítě (signalizace chyb,*

pozastavení komunikace) je zajištěn pomocí dvou speciálních zpráv (chybové zprávy a zprávy o přetížení).

Sběrnici se přenáší dva logické stavy: aktivní (dominant - dominantní) a pasivní (recessive - recesivní), přičemž dominantní stav představuje log.0, recesivní stav log.1. Sběrnice je v dominantním (aktivním) stavu, je-li alespoň jeden její uzel v dominantním stavu. V recesivním (pasivním) stavu je sběrnice tehdy, když všechny její uzly jsou v recesivním stavu. V recesivním stavu je rozdíl napětí mezi vodiči CAN\_H a CAN\_L nulový. Dominantní stav je reprezentován nenulovým rozdílem napětí. Spínače signálových vodičů jsou konstruovány tak, aby v dominantním stavu na vodiči CAN\_H bylo napětí v rozsahu 3,5 až 5V a na vodiči CAN\_L napětí v rozsahu 0 až 1,5V. V recesivním stavu je napětí vodičů CAN\_H a CAN\_L stejné a je zajištěno odporovou sítí na vstupu přijímače. Na obr. 2 je na časové ose průběhu signálu znázorněno toleranční pásmo napěťových úrovní logických stavů na sběrnici CAN. Je též patrné, že signálové vodiče CAN\_H a CAN\_L jsou vzájemně logicky invertované.



Obr. 2 Toleranční pásmo úrovní logických stavů na sběrnici CAN

*Zprávy vysílané po sběrnici protokolem CAN neobsahují žádnou informaci o cílovém uzlu, kterému jsou určeny, a jsou přijímány všemi ostatními uzly připojenými ke sběrnici. Každá zpráva je uvozena identifikátorem, který udává význam přenášené zprávy a její prioritu. Protokol CAN zajišťuje, aby zpráva s vyšší prioritou byla v případě kolize dvou zpráv doručena přednostně a dále je možné na základě identifikátoru zajistit, aby uzel přijímal pouze ty zprávy, které se ho týkají.*

*Pro zajištění transparentnosti návrhu a flexibility implementace je sběrnice CAN rozdělena do tří rozdílných vrstev:*

- *CAN vrstvy objektů,*
- *CAN transportní vrstvy,*
- *fyzické vrstvy.*

*Vrstva objektů a transportní vrstva zahrnují veškeré služby a funkce poskytované v rámci linkové vrstvy, tak jak je definována modelem ISO/OSI. Vrstva objektů je odpovědná za:*

- *nalezení zprávy, která má být vyslána,*
- *rozhodnutí, které přijaté zprávy od transportní vrstvy mají být použity,*
- *poskytování rozhraní aplikační vrstvě související s hardwarem.*

*Úkolem transportní vrstvy je především přenosový protokol. Například řízení rámců, řízení, kontrola chyb, signalizace chyb. Uvnitř transportní vrstvy je rozhodnuto, zda je sběrnice volná pro nový přenos dat či naopak jejich příjem. Také několik obecných vlastností týkajících se časování bitů je svěřeno transportní vrstvě. Je možné prohlásit, že vzhledem k povaze transportní vrstvy zde není žádný prostor*

pro její modifikaci ze strany uživatele.

Úkolem fyzické vrstvy je vlastní přenos jednotlivých bitů mezi jednotlivými uzly s respektováním všech elektrických vlastností. Uvnitř jedné sítě má fyzická vrstva stejné parametry pro všechny uzly, nicméně je možné zvolit si její parametry tak, aby co nejlépe vyhovovaly dané aplikaci.

## 6. Seznámení se s kalibrací

Firma Bolloré, je jedním z předních výrobců baterií pro elektromobily. Tato firma se rozhodla vytvořit svůj elektromobil nazvaný Blue Car. Pro tyto účely byly vybrány některé části z jiných vozů. Elektromobil je založený především na automobilu Smart. Z tohoto vozu je použit podvozek a centrální jednotka airbagu. Firma TRW pro firmu Bolloré překalibrovává centrální jednotku airbagu. Jedná se o kalibraci airbagu, aby se dal použít ve zmíněném elektromobilu. Centrální jednotka airbagu je kalibrována do jiného vozu a tak je potřeba změnit její tovární nastavení. Je zapotřebí především změnit citlivost senzorů a nastavení nového systému jednotky. Aby mohla jednotka komunikovat s ECU, je zapotřebí, aby se shodoval jejich systém. Systém je zpětně kompatibilní, tudíž se v novém automobilu mohou vyskytnout starší díly, ale ne naopak. Kalibraci zajišťuje firma TRW. Pro účely kalibrace je sestrojen i hardware pro připojení k airbagu. K airbagu samotnému se nelze připojit bez potřebné součásti. Lze se připojit pouze přes ECU. Jelikož není možné kalibrovat bez ECU, firma TRW využívá vlastní sestrojenou jednotku simulující ECU. Pomocí této jednotky se může připojit jednotka airbagu k PC.

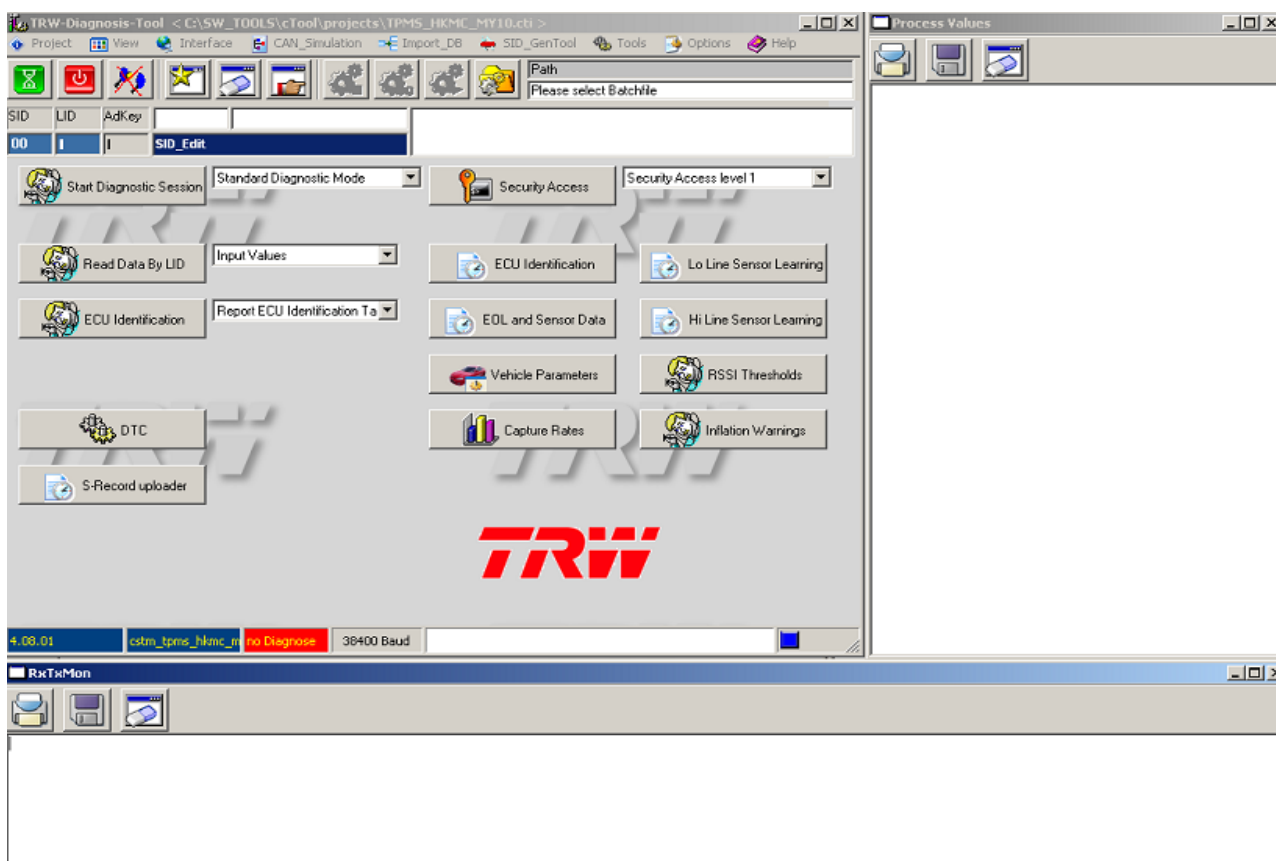




Obr. 3 Jednotka simulující ECU

Tato jednotka je napájena 12V jako v automobile. Je navržena tak, že může simulovat různé stavy, které mohou nastat v automobilu. Airbag je připojen pomocí CAN sběrnice. Připojení s počítačem zajišťuje transceiver na USB. Komunikace je řízena přes aplikaci C-Tool. Příkazy jsou překládány transceiverem do ECU. ECU se poté dotáže na airbag, který mu odpoví a ECU pošle odpověď zpět přes transceiver do počítače. Zde je odpověď vyhodnocena aplikací C-Tool a vypsána na display. Celá kalibrace se provádí v programu C-Tool. Jedná se o specifický program, který neslouží právě k tomuto účelu. Aplikace C-Tool je především diagnostickou aplikací, která se díky své flexibilitě, dá použít ke kalibraci. C-Tool nelze spustit bez připojení airbagu. Z důvodu, že ECU je navržena tak aby fungovala s airbagem. Po připojení airbagu je nutné nahrát konfigurační soubor, který určuje o jaký typ jednotky airbagu se jedná. Po spuštění se zobrazí aplikace, která se skládá z hlavního a dvou výpisových oken, viz Obr. 4. V hlavní okně se nachází panel nástrojů a jednoduché menu tvořené z tlačítek. Další dvě výpisová okna slouží k informovanosti uživatele která operace byla provedena, popřípadě s jakým výsledkem. Aplikace dále umožňuje ukládat výsledky celého procesu, což nám slouží jako historie provedení

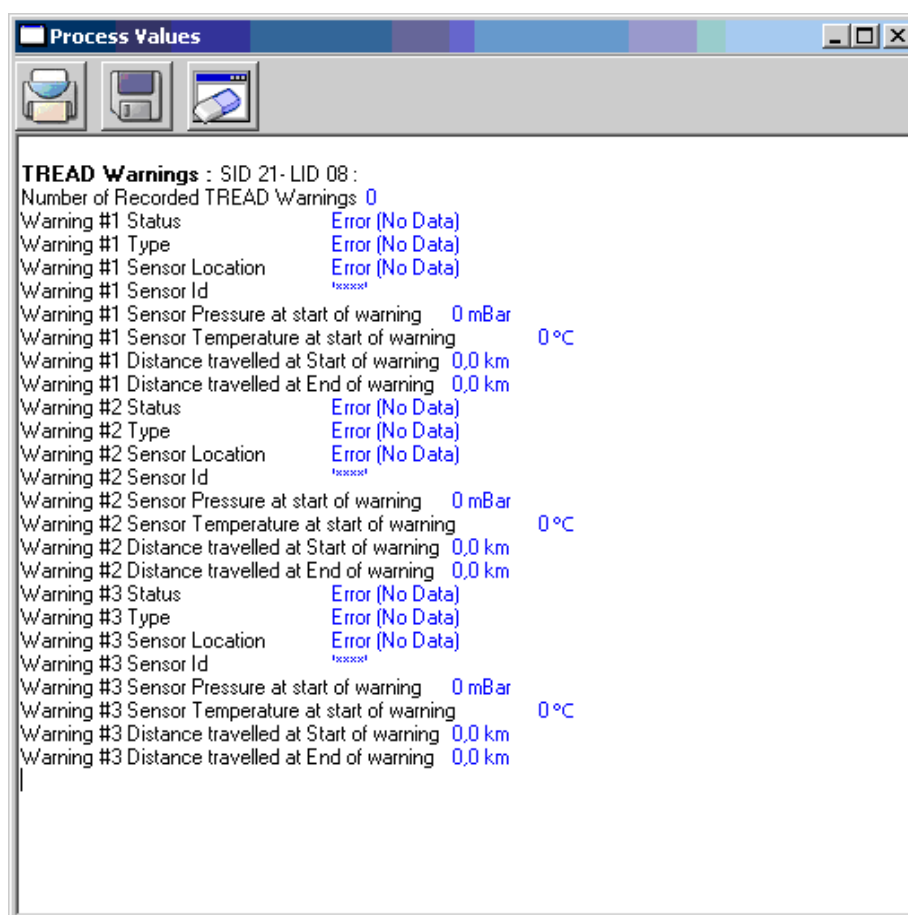
kalibrace. Hlavním úkolem aplikace je ovšem pouze diagnostika jednotky. Kalibrace se v této aplikaci provádí obtížně. Proto je neefektivní. Z tohoto důvodu aplikace obsahuje i možnost vytvoření spustitelného skriptu. V tomto skriptu se dá kalibrace provést. Skript nám nejen ulehčí čas kalibrace, ale především odstraní operátora. Díky odstranění operátora je kalibrace provedena s větší přesností. Jelikož se kalibrace skládá ze zápisu a kontroly zapsaných dat může zde dojít k nesprávné kalibraci. Takto kalibrovanou jednotku je poté potřeba celou resetovat a kalibrovat znovu.



Obr. 4 Aplikace C-Tool

### 6.1 Okno Process Values

V tomto okně se vypisují zadané příkazy s jejich odpověďmi. Je to výpisové okno pro uživatele. Výsledky jsou zde prezentovány v psané podobě. Zobrazují se zde i údaje o odeslaném příkazu v hexadecimálním tvaru. Na obrázku je ukázka příkazu "\$21,\$08" a následný výpis odpovědi z airbagu. Toto okno překládá a vypisuje hodnoty z okna RxTxMon, kde jsou zapsány v hexadecimální hodnotě. Uživatel při kalibraci využívá toto okno především ke zjištění původního nastavení a ke kontrole změn, které provedl.



Obr. 5 Ukázka Process Values Window

## 6.2 Okno RxTxMon

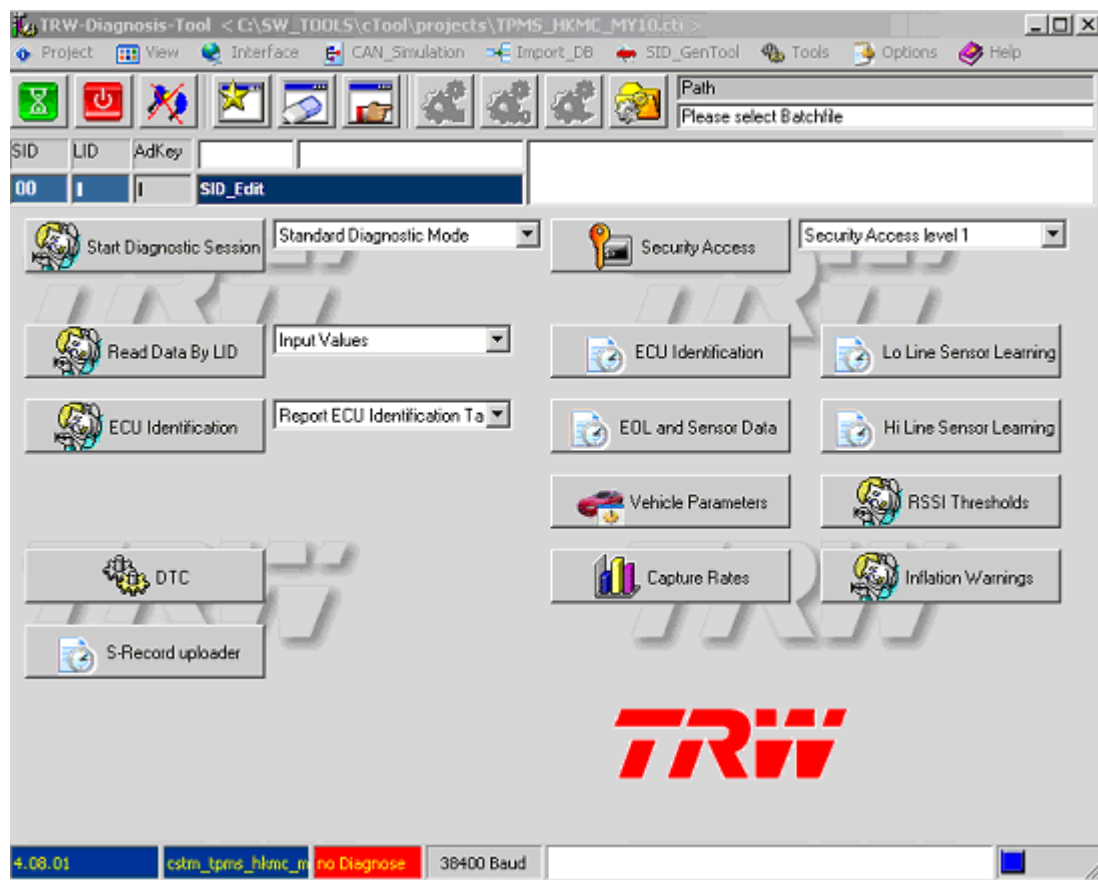
V tomto okně se vypisuje odeslaný příkaz v originální formě, tedy ve formě hexadecimálního řetězce. Toto okno slouží spíše ke kontrole, zda airbag odpovídá nebo nikoli. Odeslaný příkaz je zvýrazněn zeleně a příchozí odpověď modře.



Obr. 6 Ukázka RxTxMon

## 6.3 Hlavní okno aplikace

V hlavním okně, jak bylo už popsáno, se nachází panel nástrojů, kde najdeme asi nejzajímavější položku, viz. Obr. 7. Tou je *SID\_GenTool*. Tato záložka umožňuje přímé vložení příkazu, tzn. zapsání příkazu přímo do konzole, pokud není tento krok reprezentován tlačítkem v uživatelském menu. *SID\_GenTool* slouží především k zapisování hodnot do centrální jednotky airbagu. Dalším výrazným prvkem v hlavním okně je uživatelské menu. Menu je vytvořeno tak, aby byla zajištěná rychlá a přehledná diagnostika jednotky. Skládá se z několika tlačítek a rolovacích menu. Každé tlačítko reprezentuje určitý příkaz. Kalibrace se tedy provádí pomocí určitého sledu těchto příkazů. Uživatel postupuje podle příručky krok za krokem. Postup je dán specifikací kalibrace. Tento postup se musí dodržet, jinak nedojde ke správnému překalibrování. Kalibrace je tedy prováděna podle manuálu, který vlastní firma TRW. Jak bylo zmíněno, aplikace slouží pouze k diagnostice jednotky. Proto je kalibrace přes aplikaci neefektivní.

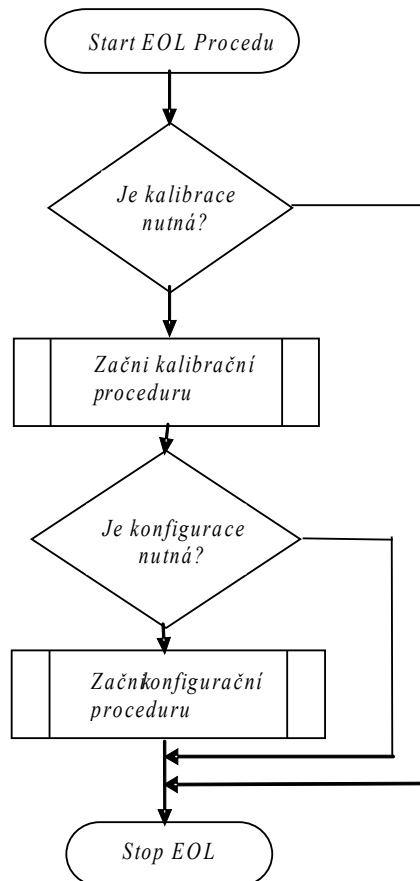


Obr. 7 Ukázka hlavního okna aplikace

#### 6.4 Postup kalibrace

Kalibrace se provádí podle specifikace projektu Blue Car. Prvním krokem kalibrace je připojení centrální jednotky airbagu k PC. Proces kalibrace se může provádět dvěma způsoby. Prvním způsobem je posílání příkazů přes CAN zprávy bez využití programu C-Tool. Jelikož je pro kalibraci vytvořena aplikace C-Tool nepoužil jsem tento způsob.. Druhým a uživatelsky přívětivějším způsobem je kalibrace přes aplikaci C-Tool. Po připojení centrální jednotky airbagu k PC a nahrání konfiguračního souboru se spustí požadovaná aplikace. Centrální jednotka airbagu rozeznává dva způsoby session ( módy ), jeden je “Normal“ a druhý je “Extended“ . Každý se používá k jiné části procesu. Jak kalibrace funguje nám zobrazuje vývojový

diagram.



Obr. 8 Vývojový diagram kalibrace

Kalibrační procedura je složená z několika částí. Hlavní dvě části jsou čtení a zápis. Začátek kalibrace je spíše kontrola, zda kalibrace už proběhla nebo je potřeba. Zjišťujeme tedy příkazy zda byla kalibrace provedena nebo nikoliv. Při zjištění, že kalibrace neproběhla, začneme kalibraci. Zkontrolujeme obsah centrální jednotky airbagu a necháme si je vypsát do okna Process Values. Tyto informace určují o jakou centrální jednotku se jedná. Po získání těchto informací je potřeba přejít do speciální úrovně “Extended“, která nám umožňuje zapisování dat do centrální

jednotky. Po přechodu do této úrovně se začnou kalibrovat jednotlivé části jednotky. Změna dat se provádí spuštěním speciálního souboru, ve kterém se nacházejí nová data. Data v jednotce jsou nahrazena novými daty tak, aby jednotka odpovídala projektu Blue Car. Po zapsání těchto dat se jednotka resetuje, aby se data zapsali do paměti. Tento krok se provádí, přes speciální tlačátko Access Key, které je popsáno níže. Po opětovném připojení k jednotce se zkontrolují zapsaná data, zda odpovídají. V případě neshody se musí tento krok provést znovu. Dalším krokem je přepsání kódování jednotky, toto se provádí přes položku SID\_GenTool (popsána výše), kde se zadá příkaz pro zapsání dat společně s dalšími 4 bajty dat. (př. 3B 22 XX XX XX XX ). Po zapsání těchto dat se opět provede reset jednotky. Dalším krokem je vymazání DTC (Diagnostic Trouble Code), které udává chyby, které mohly vzniknout při kalibraci nebo při používání jednotky. Nejprve si necháme vypsát počet DTC. V případě nenulového výsledku je nutné je vymazat. Po vymazání DTC zkontrolujeme data co jsme zapsali přes SID\_GenTool. Důležitým krokem je i zapsání nového sériového čísla této jednotky. Znovu je zapotřebí se nejprve přepnout do stavu "Extended". Zapsání se provádí opět přes SID\_GenTool. V tomto případě, se ale zapisuje 26 bajtů ( př. 3B 21 XX XX XX XX XX ..... ). Příkazy pro zapisování dat se trochu liší, rozdíl je v tom, kolik se zapisuje dat, proto jsou příkazy odlišné. Po zapsání dat se jednotka opět resetuje a po načtení se zkontroluje sériové číslo, zda odpovídá. Po zapsání sériového čísla už jen zbývá zapsat poslední data. Jedná se o 36 bajtů dat od firmy ve Frýdlantu, příkaz se opět liší (př. 3B 03 XX XX XX XX XX ... .. ). Po zapsání dat následuje opět reset a kontrola. Poslední věcí kalibrace je zkontrolování statusu centrální jednotky airbagu. Výsledek by měl být v pořádku, pokud by nebyl, musí si nechat vypsát počet DTC. V případě, že bude nenulové, necháme si jej vypsát. V Process Values se poté zobrazí o jaké chyby se jedná. Díky této informaci se dozvíme jaká chyba nastala a podle seznamu chyb ji můžeme odstranit. Pokud se chyba nenachází v seznamu chyb je potřeba kompletní reset

*jednotky a celá kalibrace se musí provést znovu. Po těchto všech krocích kalibrace je jednotka funkční a připravena pro projekt Blue Car. Celá kalibrace se provádí několik minut a je řízená operátorem což není optimální. Tudiž bylo zapotřebí zautomatizovat tuto kalibraci pomocí spustitelného skriptu.*

## **7. Návrh a naprogramování skriptu**

*Před návrhem skriptu bylo nutné se nejprve seznámit s kalibrací jednotky airbagu. Kalibraci jsem poprvé viděl ve firmě ve Frýdlantu. Na první pohled bylo zřejmé, že takto dělaná kalibrace je neefektivní. Prvním krokem při návrhu bylo nutné vymyslet kontrolu zapsaných dat. Při manuální kalibraci, která není optimální, je problém čtení a kontrola výsledků, jelikož operátor může v této fázi přehlédnout maličkost a tím udělat chybu. Proto dalším krokem návrhu je čtení vypsaných dat. Poslední částí kalibrace je zápis dat, který rovněž nesplňoval podmínky na nenáročnost. Při seznamování s kalibrací jsem zjistil, že každý požadavek na jednotku se skládá ze dvou hodnot. Ze dvou hexadecimálních čísel SID(Service Identifier) a LID(Local Identifier). Hodnota SID je definována podle standardů KWP2000. Jedná se o diagnostický protokol automobilů, definován v normě ISO 14230. Využívá se zejména ve vozidlech od roku 1998. Tento protokol sjednocuje veškeré komunikační protokoly. LID je lokální identifikátor a odpovídá specifikacím ECU. Tyto dvě hodnoty určují požadavek na jednotku. Jelikož každý požadavek se skládá z těchto dvou hodnot, bylo nutné si zjistit, veškeré kroky kalibrace reprezentovány právě těmito čísly. Tyto požadavky v aplikaci představují jednotlivá tlačítka, za nimiž se skrývají právě hodnoty SID a LID. V aplikaci se dají tyto hodnoty zapsat ručně. V případě nenalezení správného tlačítka, je zde možnost zapsat požadavek přímo. Používá se k tomu položka SID\_GenTool, která umožňuje odeslat do jednotky požadavek.*



## 7.1 Časová nenáročnost

*Tak aby bylo docíleno časové nenáročnosti, muselo se nejprve zjistit jednotlivé časové nároky na vykonání příkazů. Tyto časy se pohybovaly v ms, tudíž největší časová prodleva vznikala u operátora. Aby se mohli nahradit úkony co dělal operátor, bylo za potřebí zjistit co vše operátor dělá. Časové prodlevy vznikaly především v kontrole a zápisu dat. Zde musel uživatel vše kontrolovat a zapisovat ručně. Skript tedy musel určitě obsahovat funkce na čtení a kontrolu a především na zápis. V případě zápisu se nejednalo především o časovou náročnost, ale hlavně o nepřehlednost kontroly zapsaných dat. Zapisování probíhá přes záložku `SID_GenTool`, kde je za potřebí zapsat data od 4 do 36 bajtů. Tento krok byl potřeba nahradit skriptem.*

## 7.2 Seznámení se s jazykem skriptu

*Skript je určen pro aplikaci C-Tool tudíž je psán v jazyku určeném speciálně pro tuto aplikaci. Jazyk ve kterém, se má skript psát je podobný jazyku C. Podobnost je především v zapisování funkcí a v deklaraci proměnných. Vytvořený skript je v souboru s příponou `.ccs`. Pro účely kalibrace jednotky airbagu jsou předem vytvořeny některé funkce. Tyto funkce slouží především ke čtení dat z jednotky, ale také k nastavení CAN komunikace a připojení k databázi příkazů. Většina funkcí využívá přístupu k databázi, to je přístup k souboru `*.db`. V těchto souborech se nachází databáze příkazů. Na Obr. 9 je vidět z jakých hodnot je složena databáze. Veškeré položky jsou popsány níže.*

SID	LID	AddKey	Mnemonic	Title	Title_2Lng	Enable	ResponseID	SIDlocalID	PDU
33	151	0	RDBLID	PostCrash 5 Data	PostCrash 5 Data	Wahr	97	Falsch	21,97
33	152	0	RDBLID	PreCrash 6 Data	PreCrash 6 Data	Wahr	97	Falsch	21,98
33	153	0	RDBLID	PostCrash 6 Data	PostCrash 6 Data	Wahr	97	Falsch	21,99
33	155	0	RDBLID	Crash Counter	Crash Counter	Wahr	97	Falsch	21,9B
33	156	0	RDBLID	Read PODS Version	Read PODS Version	Wahr	97	Falsch	21,9C
33	225	0	RDBLID	Meßwert Block 225	MW-Block 225	Wahr	97	Falsch	21,E1
49	1	0	SRBLID	Max_ADC_Values	Max_ADC_Values	Wahr	113	Falsch	31,01
49	2	0	SRBLID	Max_Min_ADC_Values_uc2	Max_Min_ADC_Values_uc2	Wahr	113	Falsch	31,02
49	3	0	SRBLID	Init_QD_Counters	Init_QD_Counters	Wahr	113	Falsch	31,03
49	17	0	SRBLID	Min_ADC_Values	Min_ADC_Values	Wahr	113	Falsch	31,11
49	184	1	SRBLID	Äbrufen der unterstützten Fkt.	Start actuator test	Wahr	113	Falsch	31,B8,00,00
49	184	2	SRBLID	Start Stellglied Test	Start actuator test	Wahr	113	Falsch	31,B8,01,02
49	185	0	SRBLID	Routine weiterschalten	Switch Routine	Wahr	113	Falsch	31,B9,01,02

Obr. 9 Ukázka souboru \*.db

**SID**

*Hodnota SID je definována podle standardů KWP2000.*

**LID**

*LID je lokální identifikátor a odpovídá specifikacím ECU.*

**AddKey**

*Třetí a poslední hodnota identifikátoru příkazu. Tato hodnota nám umožní mít pro stejná SID/LID jiný příkaz. To nám umožní mít například pro příkaz \$3B, \$21 dvě různé odpovědi od jednotky.*

**Mnemonic**

*Mnemotechnické pomůcky pro tuto službu v souladu s KWP2000, zobrazené v informativním okně.*

**Title**

*Název pro tuto službu v souladu s KW2000, zobrazí ji v informativním okně spolu s výkladem.*

**Title\_2Log**

*Stejně jako Title jen v druhém jazyku.*

## **Enable**

*Určuje zda je příkaz přístupný k použití. Obsahuje dvě hodnoty False a True (“Wahr” = True, “Falsch”=false).*

## **ResponseID**

*Tato hodnota nám udává pozitivní odpověď k požadovanému příkazu. Pomocí této hodnoty se vyhledává v interpretační databázi.*

## **SIDlocalID**

*Hodnota určující zda se jedná o lokální příkaz.*

## **PDU (Process Data Units )**

*Toto pole obsahuje kompletní PDU v souladu s KW2000 (ISO 142302 ).*

*Funkce jsou vytvořené především na základní příkazy, které se opakují a je třeba je opakovat. Po prostudování jednotlivých předem vytvořených funkcí jsem došel k závěru, že jsou pro psaní skriptu vhodné jen některé. Tyto funkce zde popíši.*

**SetCANLink** ( *String sCANTypeLink , String sCustomer, int iTesterID, int iEcuID, int i1281Address, int iCANBdRate, int iCANChannel, int iCANCtrlReg, bool bExtendedID*)

*Tato funkce slouží k nastavení CAN fyzické spojení.*

- **Parametry:**

- *sCANTypeLink* – string hodnota typu CAN protokolu,

- *sCustomer* – string hodnota, jméno zákazníka,
- *iTesterID* – integer value of Tester ID (viz. ECU specifikace a KWP2000),
- *iEcuID* – integer hodnota ECU ID (viz. ECU specifikace a KWP2000),
- *iCANBdRate* – integer hodnota CAN baudrate,
- *iCANChannel* – integer hodnota CAN kanálu,
- *bExtendedID* – bool hodnota zda použít rozšíření CAN ID, nebo ne.

### **ExecSID** ( *int iSID, int iLID, int iAddKey, bool bSuppIntp, String sDBName* )

Tato funkce slouží k zasílání požadavků na jednotku airbagu. Žádost má být uvedena v příslušné databázi. Funkce vrací odpověď na žádost jako hodnotu typu *variant*.

- **Parametry:**

- *iSID* – integer hodnota Service Identifier, odpovídá standardům KWP2000
- *iLID* – integer hodnota Local Identifier (viz. ECU specifikace),
- *iAddKey* – integer hodnota Additional Key (viz. AddKey u databáze),
- *bSuppIntp* – bool hodnota zda zobrazovat výsledky v Process Values nebo ne.
- *sDBName* – jméno databáze.

### **ClearProcVal()**

Vymaže veškeré záznamy z Process Value

### **ClearRxTxMon()**

Vymaže veškeré záznamy z RxTxMon okna

### **ClearAllVal()**

Vymaže záznamy z obou oken zároveň

**SaveProcVal**(*AnsiString asTempPath, AnsiString asTempFile, int iNextIndex*)

*Uloží veškeré informace v Process Value.*

- **Parametry:**

- *asTempPath* – Umístění souboru.
- *asTempFile* – Název souboru.

**ExecBatch**(*AnsiString asPath*)

*Spustí spustitelný soubor, parametrem je cesta k souboru.*

- **Parametr:**

- *asPath* – Umístění souboru

**GenerateSID**(*Variant request, int reqLength*)

*Tato funkce generuje požadavek, parametry jsou pole hexadecimálních hodnot typu variant a parametr určující délku pole. Toto slouží k zapisování dat.*

- **Parametry:**

- *request* – V této proměnné se nacházejí data ve formě pole.
- *reqLength* – Tato hodnota udává délku pole.

**WriteProcValAtt**(*string sText, string sColor, bool bBold*)

*Výpisová funkce do Process Values s atributy.*

- **Parametry:**

- *sText* – Vypisovaný text
- *sColor* – Barva jakou má být text vypsán (“red“, “yellow“).
- *bBold* – Hodnota určující zda má být text tučný.

### 7.3 Návrh skriptu

Při návrhu jsem musel postupovat především podle původní kalibrace. Tato kalibrace spočívala v určitých krocích. Jednotlivé kroky se skládají ze zasílání požadavků. Proto bylo nutné si nejprve obstarat tyto požadavky, složené ze dvou hexadecimálních čísel SID a LID. Po získání všech těchto požadavků bylo nutné znovu projet původní kalibraci a zapsat si posloupnost těchto požadavků. Skript měl nevýhodu toho, že nemohl obsahovat některé důležité části, jelikož nebyly nadefinovány veškeré funkce. Ovšem existoval zde druhý typ skriptu a tím byl tzv. Batch file skript. Tento skript mohl obsahovat jen strohé zasílání požadavků. Je to soubor, ve kterém jsou požadavky ve frontě a vykonají se postupně za sebou. Proto je skript složen ze souboru .ccs a souborů .bbt. V souborech s příponou .bbt jsou batch skripty, které jsou využívány na funkce, které nebyli předem vytvořeny. Jednou z funkcí je zaslání tzv. Access Key (přístupového klíče), který se odesílá automaticky jedním tlačítkem. Toto tlačítko nám simuluje přepnutí do Extended session, resetování jednotky a dále odeslání přístupového klíče. Toto tlačítko bylo tedy nutné nahradit batch skriptem. Další batch skript zajišťoval přepsání kalibračních údajů. V předchozí kalibraci bylo nutné na určité operace měnit session (úroveň zabezpečení). Existovali zde dvě úrovně a to Normal a Extended. Bylo tedy nutné zjistit požadavky na změnu session. Tyto požadavky měnili celou strukturu komunikace a proto bylo důležité pracovat s nimi opatrně. Dalším krokem byl zápis jednotlivých dat do jednotky. Jelikož se zápis prováděl v aplikaci přes SID\_GenTool, kde bylo možné zadat celá data bez ohledu na délku, ve skriptu se zápis musel provádět jinak. Nejdůležitějším krokem návrhu bylo samotné uspořádání veškerých funkcí, aby byla zaručena bezchybná kalibrace. Proto se skript musel vytvářet v závislosti na původní kalibraci. Jednotlivé kroky musely následovat za sebou. Vytvořil jsem si tedy jednoduchý seznam veškerých kroků a přiřazoval jsem k nim jejich funkce. Díky tomuto seznamu jsem mohl přejít od návrhu k programování.

## 7.4 Programování

*Jak bylo zmíněno, programoval jsem podle seznamu jednotlivých kroků. Skript byl tedy tvořen posloupností funkcí. Tento skript je zapsán na přiloženém CD. Prvním krokem bylo vytvoření jednodušších částí. Byli to batch soubory. Programování batch souborů se skládá z jednotlivých kroků. Můžeme si nechat vypsat něco na Process Values pomocí info\_line, dále je rozhodující KofService, který určuje jaký požadavek se bude vykonávat. Dále je zde SubDataBase, to nám určuje z jaké databáze budeme kontrolovat požadavky. Poslední částí jsou hodnoty Service\_ID a Local\_ID. Tyto hodnoty jsou zadány v dekadickém tvaru, tzn. přepočítat hexadecimální hodnoty na dekadické. Musel jsem si naprogramovat automatické odesílání Access Key. Jedná se o potvrzování zapsaných dat. V batch skriptu, který odesílá Access Key jsem musel zadat i jednu hodnotu navíc a to Security\_Code což je právě klíč k přístupu. Níže přikládám ukázkou souboru sec\_access.bbt, který zasílá přístupový klíč.*

```
[DATABASE]
DataBase      = kw2000_srs_req.db
NoofCommands  = 3

[-001-]
info_line     = Switch to Extended Session
KofService    = EXEC_SID
Service_ID    = 16
Local_ID      = 146

[-002-]
KofService    = SEC_ACCESS_SEED
SubDataBase   = srs_req.db
Service_ID    = 39
Local_ID      = 9

[-003-]
KofService    = SEC_ACCESS_KEY
SubDataBase   = srs_req.db
Service_ID    = 39
Local_ID      = 10
Security_Code = 86902

[-004-]
KofService    = START_TESTER_PRESENT

[EOF]
```

*Tento soubor se skládá z pěti částí. Na začátku souboru je připojení k databázi a číslo určující počet příkazů. Prvním příkazem je přepnutí do Extended session. V příkazu je vidět, že se požadavek provádí přímo přes službu Exec\_SID, což znamená přímé zaslání hodnot SID a LID bez odpovědi. Druhým příkazem je získání přístupu k zaslání přístupového klíče. Třetí krok je samotné zaslání přístupového klíče. Ten obsahuje batch soubor k odeslání Access Key. Tento soubor se poté volá ve skriptu často, jelikož je zapotřebí vždy po zapsání nebo vymazání nějakých dat. Samotná kalibrace senzorů a čidel se provádí samotným batch souborem, který zde nemohu ukázat z důvodu ochrany dat firmy TRW. Nastínit ho, ale mohu. Jednalo se o přepsání veškerých dat v paměti jednotky na nové. Data byla zapsána v blocích, které byly postupně jeden po druhém přepisovány novými daty. Celý proces přepisu dat se*



provádí v chráněném *Extended session* po zaslání přístupového klíče. Při zapisování dat se využívalo dat minulých. Jak je dále vidět na ukázce části skriptu. Jedná se o metodu pro zápis sériového čísla. Sériové číslo se nepřepisuje celé, jen se změní posledních 8 znaků. Prvním příkazem, který se zde nachází je příkaz pro zjištění předchozího sériového čísla. Sériové číslo nám bude vráceno jako pole. Musíme si tedy nadeklarovat proměnnou do které zapíšeme předchozí sériové číslo. Proměnná *Response\_var* je datového typu *variant*. Typ *variant* lze použít právě jako pole. Do této proměnné se zapíše předchozí sériové číslo ve formátu pole o 31 položkách. V každé položce je umístěna hexadecimální hodnota. První dvě hodnoty jsou nedůležité, ale hodnota na indexu 3 je důležitá. Určuje nám totiž kolik znaků obsahuje sériové číslo. Každé políčko reprezentuje jeden znak ze sériového čísla. Toto pole nám dále poslouží i k zápisu. Po zjištění a uložení předešlého sériového čísla, si nadeklarujeme proměnnou *Diagnostic\_request*. Tato proměnná je také typu *variant*, ale už je dána její velikost. Jedná se o pole o 28 položkách. Pole je větší o dvě položky. První dvě položky, které jsou nenulové nám určují *SID* a *LID*, tedy hodnoty požadavku, že se jedná o zápis o velikosti 26 položek. Zbytek je nulový a připraven k naplnění. Pro naplnění je zde *for* cyklus. Tento cyklus bude prováděn tolikrát, podle čísla v *response\_var* na indexu 3. V cyklu se provede zápis dat do proměnné *Diagnostic\_request*. Do proměnné *Diagnostic\_request* se zapisuje až za první dvě položky. Po skončení tohoto cyklu se v *Diagnostic\_request* nachází celé předešlé sériové číslo. Díky tomu nám teď zbývá pouze přepsat posledních 8 položek novými. K tomuto je zapotřebí další *for* cyklus. Tentokrát bude prováděn jen 8krát. Postupně se na posledních 8 položek zapíší nové znaky sériového čísla. Znaky jsou ještě upraveny a to tím, že je k nim přičtena hodnota 30h. Takto je zapsáno celé sériové číslo. Po uložení celého nového sériového čísla nezbývá než zapsat ho do jednotky. Zapsání se provádí funkcí *GenerateSID()*, kde se jako parametr vloží proměnná, která musí být datového typu *variant*. Druhým parametrem je velikost tohoto pole. Po

zapsání nového sériového čísla je zapotřebí pauza jednotky a to na dobu minimálně 400ms. V mém případě jsem zvolil 500ms. Poté co je zápis dokončen je potřeba znovu zaslat přístupový klíč. Tento krok obstará výše uvedený batch soubor.

```
void write_serial_number()
{
    Response_var = ExecSID(33,225,0,false, "kw2000_srs_req.db");

    Diagnostic_request=
[0x3B,0xE1,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
|0x00,0x00,0x00,0x00,0x00,0x00];
    for(i = 0; i < Response_var[3]-1; i++)
    {
        Diagnostic_request[i+1] = Response_var[i+5];
    }

    for(i = 0; i < 8; i++)
    {
        Diagnostic_request[i+15] = ECU_New_serial[i] +0x30;
    }
    Vysledek = GenerateSID (Diagnostic_request,28);

    Pause(500);
}
```

Další ukázkou zápisu je přepsání kódování. Provádí se podobně jako zápis nového sériového čísla. S tím rozdílem, že zde nepotřebujeme znát předchozí hodnoty. Nadeklarujeme si proměnnou, kterou naplníme 10 položkami. První dvě položky nám určují SID a LID tudíž musí být zadány. Zbylé ponecháme nulové. Pomocí for cyklu naplníme nulové položky novými hodnotami z proměnné hodnoty. Po té odešleme celý požadavek přes GenerateSID(). Poté následuje pauza 500ms a aby se data zapsala a mohl jsem s nimi dále pracovat musíme zaslat přístupový klíč. K tomu slouží opět batch soubor `sec_access.bbt`. Skript se zavolá pomocí funkce `ExecBatch()`. Po vykonání batch souboru se opět pokračuje ve vykonávání dalších metod skriptu.

```
void write_variant_coding() {  
    Diagnostic_request = [0x3B,0x22,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00];  
    for(i = 0; i < 8; i++)  
    {  
        Diagnostic_request[i+2] = hodnoty[i];  
    }  
    Vysledek = GenerateSID (Diagnostic_request,10);  
    Pause(500);  
    ExecBatch("eeprom\\sec_access.bbt");  
}
```

Skript umožňuje dva různé způsoby jak odeslat požadavek. První způsob je přes funkci `ExecSID()`, popsanou výše. Příklad zaslání požadavku na přečtení `Calibration_ID`.

```
Vysledek = ExecSID(33,150,0,false, "kw2000_srs_req.db");
```

Druhý způsob je použití proměnné typu `variant` jako pole o dvou položkách. V tomto poli by byly zapsány `SID` a `LID`. K zaslání požadavku by se využilo funkce `GenerateSID()`. Příklad zaslání stejného požadavku jako výše.

```
Variant Request = [0x21][0x96];  
Vysledek = GenerateSID (Request,2);
```

Rozdíl mezi oběma použitými je ve formě zasláního `SID` a `LID`. Zatímco

u prvního příkladu je SID a LID zasláno v dekadické formě, v příkladu dvě je zasláno v hexadecimálním tvaru. Nevýhodou u prvního příkladu je především to, že pokud neznáme správnou databázi tak nám nemusí být požadavek odeslán. V druhém případě je nevýhoda ta, že pokud nepožadujeme výpis do Process Valuse, stejně tam bude. Existuje ještě třetí způsob jak odeslat požadavek a to ,jak bylo ukázáno výše, přes batch soubor. Nevýhoda je stejná jako u příkladu dvě.

Veškeré informace jsou vypisovány do Process Value. Vypisují zde o jaký krok se jedná. Výpis stringového řetězce se provádí přes funkci WriteProcValAtt(). Tato funkce obsahuje parametry. Tyto parametry nesou informace o barvě a tučnosti textu. Pomocí této funkce mohu informovat o dění při probíhání skriptu. Vypisují do okna vždy informaci o úspěšném či neúspěšném provedení příkazu. Ukázka je výpis informace o zápisu nového sériového čísla. Na ukázce je názorně vidět využití parametrů. Text bude tedy červený a písmo bude tučné.

```
WriteProcValAtt("#09+\"Zapisuji nové sériové číslo\n\n\", \"red\", true);
```

## 8. Měření zrychlení kalibrace

Posledním krokem práce bylo změření zrychlení, kterého automatická kalibrace dosáhla oproti kalibraci vytvářené operátorem. Bylo předem jasné, že automatická kalibrace zrychlí tento postup. Důležité bylo změřit původní čas kalibrace a čas automatické kalibrace.

Kalibrace vytvářená operátorem byla časově pomalejší. V případě operátora šlo o jasné důvody. Kontrola dat zabírala většinu času kalibrace. Tato kontrola se musela provádět po každém příkazu. Operátor mohl kontrolovat data přes okno *Process Values*, ve kterém musel hledat veškeré potřebné informace. Dále musel zkontrolovat hodnotu DTC. Vznikne-li při kalibraci chyba zapíše se do DTC. Operátor tedy musel kontrolovat tyto dvě věci. S celkového času toto činí 70%. Při původních kalibracích operátor zvládl kalibraci za 2-3 minut. S balením jednotky je čas 6 minut. Tato kalibrace s sebou, ale nesla riziko přehlédnutí nějaké chyby. Tudíž by se musela kalibrace znovu opakovat a došlo by k zbytečné časové prodlevě. Díky této kalibraci se za směnu dokázalo překalibrovat cca 60 jednotek.

Časově výhodnější se stala automatická kalibrace. Při automatické kalibraci se provedou stejné kroky, jako u kalibrace s operátorem. Jelikož je automatická kalibrace řízená skriptem, můžeme zde provádět kontrolu. Pomocí kontroly ve skriptu je ušetřeno nejvíce času z původní kalibrace. Skript provede veškeré kroky mnohem rychleji. Díky tomuto zrychlení se kalibrace provádí v několika sekundách. Zápis dat je nejdelším časovým úsekem automatické kalibrace. Z tohoto důvodu kalibrace trvá přibližně 6-8 vteřin. Díky tomuto času jedna směna dokáže vyrobit mnohonásobně více jednotek. To přináší kvalitní práci za co nejkratší dobu.

*Zhodnotím-li obě kalibrace, musím vyzdvihnout především tu automatickou, která především zkvalitnila postup. V automatické kalibraci nedojde k přehlédnutí chyby a je tedy zaručena bezchybná kalibrace. Nesmíme, ale zapomínat i na časový faktor, díky kterému je možné kalibrovat více jednotek.*

## **9. Závěr**

*Poté co jsem se dozvěděl o možnosti dělat bakalářskou práci pro firmu TRW, chtěl jsem ji využít. Věděl jsem, že firma TRW se pohybuje v automobilovém průmyslu, o který se zajímám. Práci jsem si vybral podle toho co mě zajímalo nejvíce. Proto jsem zvolil automatickou kalibraci centrální jednotky airbagu. Práce mě hned ze začátku nadchla. Poté co jsem mohl poprvé programovat jednotku airbagu jsem poznal, že to nebude jednoduchá práce. Seznámení s kalibrací proběhlo ve firmě ve Frýdlantu. Byl jsem mile překvapen přístupem a moderním zařízením ve firmě. Ve firmě mě ve všech ohledech vycházeli vstříc. Skript jsem konzultoval především se zaměstnanci firmy, kteří prováděli kalibraci. Skript jsem začal vytvářet předběžně a jezdil si ho testovat do Frýdlantu. Důležitým krokem bylo naučit se programovací jazyk, který se podobal jazyku C. Tento jazyk nebyl tak odlišný, proto seznámení proběhlo rychle. Jelikož jsem neměl přístup k zařízení pro připojení jednotky, které bylo pouze ve Frýdlantu, musel jsem skript psát po částech a jezdit si ho otestovat. Testování jsem prováděl vždy ve formě návštěvy ve firmě. Skript jsem touto cestou vytvořil. Skript slouží především k odstranění operátora, což znamená odstranit chybový faktor. Dalším krokem práce bylo změření zrychlení, o které je automatická*

*kalibrace rychlejší. Toto měření jsem provedl ve Frýdlantu. Bylo předem jasné, že automatická kalibrace zrychlí postup. Při měření se ukázalo, že automatická kalibrace zrychlila kalibraci o několik minut. Díky tomuto zrychlení kalibrace, firma ve Frýdlantu může překalibrovat mnohonásobně více jednotek.*

## **Reference na použitou literaturu**

[1] *Torregrossa, Michaël. Bolloré Bluecar,*

URL: <<http://www.avem.fr/news?id=2995>>, 7.3.2012

[2] *TRW automotive. URL: <<http://www.trw.cz/>>, 2006*

[3] *Bosch, Centrální řízení airbagu,*

URL:<[http://rb-kwin.bosch.com/cz/cs/safety\\_comfort/drivingsafety/capscombinedactivepassivesafety/basiscaps/centralairbagcontrolunit.html](http://rb-kwin.bosch.com/cz/cs/safety_comfort/drivingsafety/capscombinedactivepassivesafety/basiscaps/centralairbagcontrolunit.html)>, 7.3.2012

[4] *Polák, Karel. Sběrnice CAN, elektro revue,*

URL: <<http://www.elektrorevue.cz/clanky/03021/index.html>>, 16.6.2003

[5] *Taraba, Radek. Aplikování sběrnice CAN,*

URL: <<http://www.hw.cz/Rozhrani/ART1173-Aplikovani-sbernice-CAN.html>>,  
9.11.2004