

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ



DIPLOMOVÁ PRÁCE

Vizualizace a zpracování dat testovací jízdy
vozidla

Praha, 2011

Autor: Bc. Pavel Valenta

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Pavel Valenta**

Studijní program: Elektrotechnika a informatika (magisterský), strukturovaný
Obor: Kybernetika a měření, blok KM1 - Řídicí technika

Název tématu: **Vizualizace a zpracování dat testovací jízdy vozidla**

Pokyny pro vypracování:


Navrhnete koncepci programového vybavení pro vizualizaci a zpracování dat ze záznamů zkušebních jízd vozidel po běžných komunikacích. Program musí podporovat zobrazení ujeté trasy v prostředí Google Earth, zobrazení uživatelem definovaných grafů závislosti jednotlivých měřených veličin, export dat do simulátoru trasy. Vybrané funkce musí být k dispozici i v režimu on-line, kdy jsou data publikována v reálném čase prostřednictvím protokolu UDP.

Seznam odborné literatury:

- [1] Novotný, M.: Technické vybavení záznamníku parametrů trasy, Diplomová práce ČVUT FEL, 2009
- [2] Kočí, J.: Programové vybavení záznamníku parametrů trasy, Diplomová práce ČVUT FEL, 2009
- [3] Haasz, V., Roztočil, J., Novák, J.: Číslicové měřicí systémy, vydavatelství ČVUT, 2000

Vedoucí: Doc.Ing. Jiří Novák, Ph.D.

Platnost zadání: do konce zimního semestru 2011/2012


prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




prof. Ing. Boris Šimák, CSc.
děkan

V Praze dne 24. 11. 2010

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne

.....

podpis

Poděkování

Chtěl bych poděkovat především vedoucímu mé diplomové práce, doc. Ing. Jiřímu Novákovi, Ph.D., za náměty, připomínky a rady týkající se vypracování práce a též za ochotu a vstřícnost. Také děkuji Ing. Josefu Kočímu a Ing. Michalu Novotnému, kteří záznamové zařízení Vehicle Data Logger navrhli a vyrobili. V neposlední řadě patří dík též rodině a přátelům, kteří mě v době studia podporovali.

Abstrakt

Tato práce je součástí projektu návrhu zařízení pro záznam parametrů trasy vozidla (VDL – Vehicle Data Logger). Záznamník VDL během jízdy vozidla zapisuje hodnoty polohy, výšky a rychlosti pomocí GPS, hodnoty zrychlení a úhlové rychlosti pomocí tříosého inerciálního senzoru a záznam komunikace na sběrnici CAN. Práce se zabývá návrhem aplikace, která slouží pro vyhodnocení a prezentaci zaznamenaných dat. Jedná se o .NET aplikaci psanou v jazyce C#, která spolupracuje s internetovými mapami Google Maps a Google Earth.

Abstract

This thesis is a part of project that designs device for recording route parameters of the vehicle. (VDL – Vehicle Data Logger). VDL records location, altitude and speed using GPS, acceleration and angular rate using a tri-axes accelerometer and also records a communication on CAN bus. This thesis deals with implementation of an application, that interprets and presents recorded data. It is an .NET application implemented in C# implemented that cooperates with internet map Google Maps and internet globe Google Earth.

Obsah

1 Úvod.....	1
2 Vehicle Data Logger	3
2.1 GPS modul LR9548SLP	3
2.2 Inerční senzor ADIS16350	4
2.3 CAN rozhraní	5
3 Formát záznamových souborů	7
3.1 Hlavička souboru	7
3.2 Formát zprávy z modulu GPS	8
3.3 Formát zprávy z inerciálního senzoru	9
3.4 Formát zprávy z CAN sběrnice.....	9
3.5 DBC Specifikace	10
4 Aplikace VDLviewer	13
4.1 Načtení souboru do aplikace	13
4.2 Zpracování souboru se záznamem	14
4.3 Chybějící údaje ve zprávě	17
4.4 Operace se záznamem	18
4.4.1 Zobrazování signálů v grafech.....	19
4.4.2 Funkce Najít.....	22
4.4.3 Zobrazení spotřeby paliva.....	24
4.5 Zobrazování projeté trasy na mapě	26
4.5.1 Interakce programu s Google Maps a Google Earth.....	26
4.5.2 Výběr úseku záznamu z mapy	34
4.6 Export dat ze záznamu	37
4.7 Editace souboru se záznamem	38
4.8 Online přenos a zobrazování dat	39

5 Závěr	41
6 Literatura.....	43
7 Obsah CD.....	44

Seznam obrázků

Obrázek 1.1: Zařízení Vehicle Data Logger (VDL).....	2
Obrázek 2.1: GPS modul LR9548SLP	4
Obrázek 2.2: Inerční senzor ADIS16350	5
Obrázek 2.3: Struktura zprávy CAN 2.0A.....	6
Obrázek 4.1: Dialogové okno načtení záznamu	14
Obrázek 4.2: Class diagramy reprezentující jednotlivé zprávy	15
Obrázek 4.3: Class diagramy tříd pro popis komunikačních prostředků na sběrnici.....	16
Obrázek 4.4: Hlavní okno aplikace.....	19
Obrázek 4.5: Class diagram třídy GraphControl	20
Obrázek 4.6: Zobrazení signálů v závislosti na čase	21
Obrázek 4.7: Zobrazení signálů v závislosti mezi sebou.....	22
Obrázek 4.8: Class diagramy tříd TrueSelection a Rule.....	23
Obrázek 4.9: Dialogové okno Najít pro funkci hledání v záznamu.....	23
Obrázek 4.10: Výsledek hledání v záznamu.....	24
Obrázek 4.11: Průběh signálu MO5_Verbrauch	24
Obrázek 4.12: Okamžitá spotřeba přepočítaná na l/100 km	26
Obrázek 4.13: Trasa zobrazená v prostředí Google Maps	27
Obrázek 4.14: Class diagram třídy GEarth	29
Obrázek 4.15: Trasa zobrazená v prostředí Google Earth	29
Obrázek 4.16: Projetá trasa s namodulovaným signálem rychlosti, panel pro spolupráci s Google Earth.....	31
Obrázek 4.17: Část projeté trasy s namodulovaným signálem zrychlení ve směru Z (vertikální směr)	31
Obrázek 4.18: Normalizované souřadnice v okně mapy Google Earth.....	35
Obrázek 4.19: Class diagram třídy SelectorCursor.....	35
Obrázek 4.20: Vyznačení úseku trasy na mapě.....	36
Obrázek 4.21: Přehled zaznamenaných zpráv a jejich výskytů	39

Seznam tabulek

Tabulka 3.1: Popis formátu zprávy z GPS modulu.....	8
Tabulka 3.2: Popis formátu zprávy z inerčního senzoru.....	9
Tabulka 3.3: Popis formátu zprávy ze sběrnice CAN.....	9
Tabulka 4.1: Formát exportu ze záznamového souboru	37
Tabulka 4.2: Přehled statistiky pořízených záznamů.....	38

Kapitola 1

Úvod

Tato práce je součástí projektu tvorby záznamového zařízení nazvaného Vehicle Data Logger (VDL). Toto zařízení (viz obrázek 1.1) slouží k záznamu parametrů jízdy primárně automobilu, ale je možné jej využít i v jiném dopravním prostředku. Vehicle Data Logger snímá následující údaje:

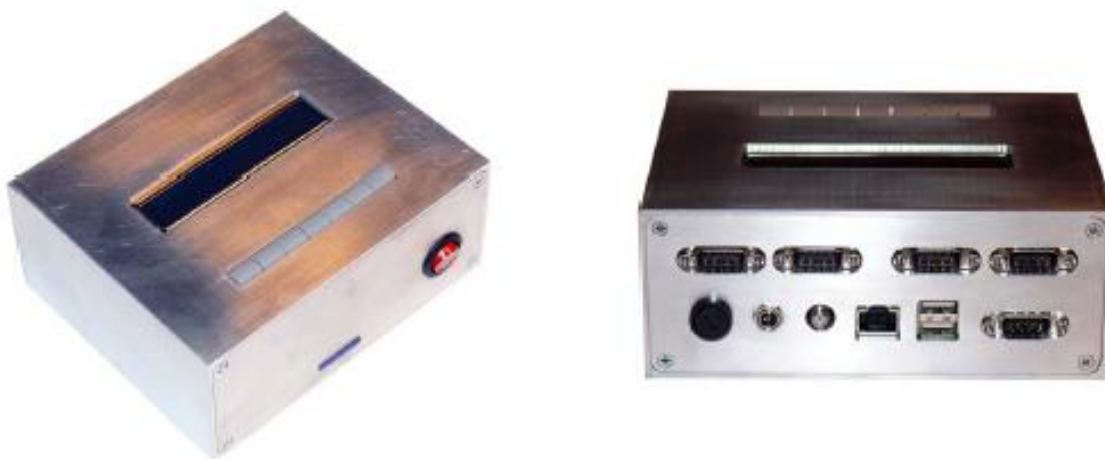
- o poloze dopravního prostředku pomocí systému GPS
- zrychlení ve třech směrech a úhlových rychlostech ve třech osách
- data ze sběrnice CAN, jíž jsou moderní automobily standardně vybaveny

Projekt vývoje zařízení VDL byl rozdělen na tři části. První dvě, zabývající se výrobou zařízení a jeho softwarovým vybavením, byly provedeny v rámci diplomových prací Ing. Josefa Kočího [1] a Ing. Michala Novotného [2]. Surová data v podobě textového souboru, jenž VDL zaznamenává, nejsou pochopitelně uživatelsky přívětivá a bez dalšího zpracování a zobrazení nemají pro uživatele valný smysl. Tato práce je proto zaměřena na prezentaci a vizualizaci naměřených dat.

Výsledkem práce je program, který naměřené hodnoty přehledně prezentuje, zobrazuje jejich průběhy v grafech, vykresluje projetou trasu na mapě v prostředí Google Maps i Google Earth a nabízí další možnosti práce se záznamem. Tímto krokem se teprve dají efektně představit všechny přednosti a klady zařízení VDL.

Čtenář je nejprve v kapitole 2 ve stručnosti seznámen se zařízením VDL, jeho

komponentami a hlavními vlastnostmi. V kapitole 3 následuje popis záznamových souborů a jejich zpracování a uložení zaznamenaných hodnot. Čtenář je také seznámen se souborovým typem DBC, jenž je důležitý pro interpretaci zpráv posílaných po sběrnici CAN. Kapitola 4 se pak věnuje aplikaci *VDLviewer*, popisu její struktury, jejího fungování, vlastností a funkcí.



Obrázek 1.1: Zařízení Vehicle Data Logger (VDL)

Kapitola 2

Vehicle Data Logger

2.1 GPS modul LR9548SLP

Global Positioning System (zkráceně GPS) je vojenský globální družicový polohový systém vyvíjený od roku 1973 a provozovaný Ministerstvem obrany Spojených států amerických, s jehož pomocí je možno s přesností do 10 metrů a větší určit polohu a přesný čas kdekoliv na Zemi nebo nad Zemí. Přesnost GPS lze s použitím dalších metod ještě zvýšit až na jednotky centimetrů. Aby nemohl být systém GPS zneužit proti Spojeným státům, byl signál pro civilní uživatele vysílán s umělou chybou, která měla bránit například přesnému navigování balistických střel.

V roce 2000 byla umělá chyba vypnuta a všichni uživatelé tak mohou využívat přesnosti navigace pomocí signálu v pásmu L1 (1575,42 MHz). Armáda však stále může zaměřovat pozice výrazně přesněji, neboť má k dispozici i šifrované signály v pásmu L2 (1227,62 MHz). Podrobnější informace o systému GPS jsou například v [10].

Pro záznam informací GPS slouží modul LR9548SLP, jenž můžeme vidět na obrázku 2.1. Modul vyrábí firma LEADTEK a jedná se o malý integrovaný obvod o rozměrech 24 mm x 20 mm x 2,9 mm v pouzdře SMD, který se vyznačuje vysokou citlivostí a nízkou spotřebou. Tento 20kanálový GPS přijímač je navržen pro široké využití v OEM aplikacích.



Obrázek 2.1: GPS modul LR9548SLP

Parametry GPS modulu LR9548SLP

- Chipset GSC3f, SIRFstarIII
- 20kanálový přijímač, $f = 1575,42$ MHz
- Citlivost -159 dBm
- Přesnost horizontální 10 m v režimu 2D RMS
- Přesnost rychlosti 0,1 m/s
- Přesnost času 1 mikrosekundy synchronizovaný s GPS
- Čas potřebný pro příjem první hodnoty polohy:
 - Studený start – 42 s
 - Teplý start – 38 s
 - Horký start – 1 s
- Napájecí napětí 3,3 V nebo 5 V se spotřebou ≈ 165 mW
- Sériové rozhraní TTL s protokolem NMEA-0183
- Obnovovací frekvence platných dat = 1 Hz

2.2 Inerční senzor ADIS16350

Inerční senzor ADIS16350 (obrázek 2.2) představuje úplný inerční systém s 6 stupni volnosti. Je vybaven tříosým gyroskopem a akcelerometrem. Jedná se o MEMS (Mikro Elektro Mechanický Systém) systém, který kombinuje možnost kalibrace, předzpracování a poskytnutí měřených dat přes komunikační sběrnici SPI. Umožňuje měření hodnot z akcelerometru, gyroskopu, teploty gyroskopického systému a vstupní napájecí napětí. Hodnoty úhlové rychlosti je možné zaznamenávat v několika rozsazích.



Obrázek 2.2: Inerční senzor ADIS16350

Parametry senzoru ADIS16350

- Rozsah akcelerometru ± 10 g s rozlišením 14 bitů
- Rozsah gyroskopu ± 300 °/s, ± 150 °/s, ± 75 °/s s rozlišením 14 bitů
- Rozsah vzorkovací frekvence od 0,413 – 819,2 Hz
- Napájecí napětí 5 V DC

2.3 CAN rozhraní

Controller Area Network (CAN) je sériový komunikační protokol vyvinutý firmou Bosch pro nasazení v automobilech. Maximální přenosová rychlost sběrnice je 1Mbit/s. Vzhledem k tomu, že přední výrobci integrovaných obvodů implementovali podporu protokolu CAN do svých produktů, stala se sběrnice CAN široce používanou nejen jako vnitřní komunikační síť senzorů a funkčních jednotek v automobilu, ale i v různých průmyslových aplikacích.

Protokol je definován standardem ISO 11898, který popisuje fyzickou a linkovou vrstvu protokolu specifikací CAN 2.0A. Později byla ještě vytvořena specifikace CAN 2.0B, která zavádí dva formáty datové zprávy – standardní a rozšířený. Rozšířený formát používá identifikátor zprávy o celkové délce 29 bitů proti 11 bitům standardního formátu. Detailní popis fyzické a linkové vrstvy je možno najít například v [3]. Pro potřeby interpretace dat přenášených po sběrnici je třeba znát formát posílaných zpráv. Specifikace protokolu CAN definuje celkem čtyři typy zpráv.

Jsou to:

- datová zpráva
- vyžádání dat
- zpráva o chybě
- zpráva o přetížení

VDL během své práce zaznamenává datové zprávy. Jejich struktura je ukázána na obrázku 2.3. Postup při zpracování datové zprávy bude popsán v následující kapitole.



Obrázek 2.3: Struktura zprávy CAN 2.0A, převzato z [3]

VDL je schopen zaznamenávat data až ze 4 různých fyzických vrstev sběrnic CAN. Je možno měřit data jak High Speed, tak Low Speed variant. Ty jsou rozlišeny použitým typem budičů. Monitorování sběrnice je zajišťováno řadiči SJA1000 od firmy NXP.

Kapitola 3

Formát záznamových souborů

Data z VDL se ukládají na USB flash disk, který je k záznamníku připojený. Záznamník generuje klasický textový soubor s příponou *.txt. Podoba záznamového souboru byla určena ve spolupráci s autory VDL. V této kapitole bude popsána struktura souboru a získání užitečných informací z něj.

3.1 Hlavička souboru

Na začátku každého záznamového souboru je zapsána hlavička, která obsahuje základní údaje o daném záznamu. Je tvořena devíti řádky začínajícími znakem #. V hlavičce je uvedeno datum a čas zahájení záznamu a dále nastavení parametrů záznamu z inerciálního senzoru a CAN sběrnice. Podoba konkrétní hlavičky je následující:

```
# cas:2010.06.23-17.28.35
# Zaznam GPS
# Zaznam INS: I time XG YG ZG XA YA ZA
# Nastaveni INS: SMPL_PRD=10 Hz, SENS/AVG=0x402
# Zaznam CAN: C TIME SENSOR ID LENGTH FLAGS
# Nastaveni CAN1: REC=yes ID=0 MASK_ID=0 EXT=no MASK_EXT=no RTR=no MASK_RTR=no
BR=500000 LO=yes
# Nastaveni CAN2: REC=no ID=0 MASK_ID=0 EXT=no MASK_EXT=no RTR=no MASK_RTR=no BR=500000
LO=yes
# Nastaveni CAN3: REC=no ID=0 MASK_ID=0 EXT=no MASK_EXT=no RTR=no MASK_RTR=no BR=500000
LO=yes
# Nastaveni CAN4: REC=no ID=0 MASK_ID=0 EXT=no MASK_EXT=no RTR=no MASK_RTR=no BR=500000
LO=yes
```

Každý následující řádek v záznamu obsahuje zprávu od některého ze tří modulů. První znak v řádku (G, I, C) definuje, z jakého modulu zpráva pochází. Následuje časová značka v mikrosekundách určující dobu příchodu zprávy od spuštění. Dále mají zprávy od jednotlivých modulů specifický tvar.

3.2 Formát zprávy z modulu GPS

Formát zprávy z modulu GPS, význam symbolů v tabulce 3.1:

G X0 GPSD,O=RMC X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14

Příklad zprávy:

G 0878867 GPSD,O=RMC 1277303402.000 0.005 50.102883 14.393528
271.30 25.60 15.20 153.1500 0.000 0.000 ? 51.20 ? 3

Tabulka 3.1: Popis formátu zprávy z GPS modulu

Znak	Význam	Jednotky
X0	časová značka VDL	μs
X1	čas	formát UTC
X2	chyba času	s
X3	zeměpisná šířka	°
X4	zeměpisná délka	°
X5	nadmořská výška	m
X6	odhadovaná chyba polohy	m
X7	odhadovaná chyba výšky	m
X8	kurz	°
X9	horizontální rychlost	m/s
X10	vertikální rychlost	m/s
X11	odhadovaná chyba kurzu	°
X12	odhadovaná chyba horizontální rychlosti	m/s
X13	odhadovaná chyba vertikální rychlosti	m/s
X14	mód	1=no_fix, 2=2D, 3=3D

3.3 Formát zprávy z inerciálního senzoru

Formát zprávy z inerciálního senzoru, význam symbolů v tabulce 3.2:

I X0 X1 X2 X3 X4 X5 X6

Příklad zprávy:

I 0145413 -5 -11 12 -39 62 -396

Tabulka 3.2: Popis formátu zprávy z inerciálního senzoru

Znak	Význam	Jednotky
X0	časová značka VDL	μs
X1	zrychlení X	m/s^2
X2	zrychlení Y	m/s^2
X3	zrychlení Z	m/s^2
X4	úhlová rychlost X	$^\circ/\text{s}$
X5	úhlová rychlost Y	$^\circ/\text{s}$
X6	úhlová rychlost Z	$^\circ/\text{s}$

3.4 Formát zprávy z CAN sběrnice

Formát zprávy z CAN sběrnice, význam symbolů v tabulce 3.3:

C X0 X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12

Příklad zprávy:

C 134739802 1 5e0 8 0 20 ff 1b 2 4c 0 0 0

Tabulka 3.3: Popis formátu zprávy ze sběrnice CAN

Znak	Význam	Jednotky
X0	časová značka VDL	μs
X1	Číslo sběrnice	-
X2	ID zprávy	-
X3	DLC	-
X4	extended ID	-
X5 – X12	datové bajty	-

Na rozdíl od zpráv z GPS modulu a inerciálního senzoru, kde jednotlivé položky přímo vyjadřují hodnoty konkrétních měřených veličin, zprávy z CAN sběrnice obsahují pouze surová data v podobě datových bajtů dané CAN zprávy.

K užitečným informacím se dostaneme pouze správnou interpretací těchto bajtů, k čemuž potřebujeme znát popis komunikačních prostředků na snímané CAN sběrnici. Takový popis lze efektivně zaznamenat pomocí specifikace navržené firmou Vector Informatik GmbH. V následující části bude uveden krátký přehled formátu souboru DBC, který komunikaci na sběrnici definuje.

3.5 DBC Specifikace

DBC soubor popisuje komunikaci na jedné sběrnici CAN. V základní podobě definuje uzly v síti, jimi posílané zprávy a signály, které zprávy obsahují. Soubor je tvořen sekcemi, které popisují jednotlivé prvky komunikace. Podrobný popis specifikace je uveden v [4]. Zde uvádím přehled sekcí, které jsou potřebné pro interpretaci dat tak, aby mohly být programem *VDLviewer* prezentovány.

- Verze a definice symbolů

Tato sekce popisuje verzi DBC souboru. Tato sekce začíná slovem `VERSION` a končí s koncem řádku. Má následující formát:

```
VERSION "version description string" \n
```

```
NS_:
```

```
<keyword>
```

```
<keyword>
```

NS sekce deklaruje klíčová slova pro následující sekce. Všechna klíčová slova, která jsou zde obsažena, by měla být uvedena v následujícím textu. Každé klíčové slovo by mělo být uvedeno na zvláštním řádku.

- Definice uzlů

Tato sekce deklaruje jména všech uzlů na sběrnici CAN. Tato sekce končí klíčovým slovem zahajujícím další sekci. Má následující formát:

```
BU_ : <node name> <node name>
<node name> ... <node name>

<node name> jméno uzlu
```

- Definice zpráv

Tato sekce definuje zprávu CAN. Sekci následuje seznam definic signálů obsažených ve zprávě. Má následující formát:

```
BO_ <id> <id name> : <size> <sender> \n
BO_ 1456 ID_Name : 8 STG_Gate \n
```

<id> ID zprávy v desítkové soustavě

<id name> jméno zprávy

<size> počet datových bajtů

<sender> jméno odesílatele zprávy (CAN uzlu). Pokud je odesílatel neznámý, má hodnotu Vector_XXX.

Pro každou zprávu je definován popis signálů, které zpráva obsahuje. Formát definice signálu je následující:

```
SG_ <signal name> <type> : <first> | <size> @
<format> + ( <scale> , <offset> ) [ <min> | <max> ]
"unit" <receiver name> \n
```

```
SG_ VLI_ABS: 7 | 2@1+ (1, 0) [0 | 1] "kg" Vector_XXX
```

<type> označuje typ signálu:

- „M“ označuje multiplexační signál
- „mX“ kde X je číslo, označuje hodnotu multiplexu
- je-li pole prázdné, jedná se o obyčejný signál

<first> pozice prvního bitu signálu

<size> počet bitů v signálu

<format> formát dat. 0 – Motorola, 1 – Intel.

Hodnota signálu se spočítá jako

$$\langle \text{signal value} \rangle = \langle \text{scale} \rangle * \langle \text{raw data} \rangle + \langle \text{offset} \rangle$$

<min> minimální hodnota signálu

<max> maximální hodnota signálu

"unit" fyzikální jednotka (kg, km/h, atd.)

<receiver name> jméno adresáta signálu

\n označuje nový řádek

- **Komentáře**

Tato sekce obsahuje komentáře pro uzly sběrnice, zprávy a signály.

```
CM_ BU_ <unit name> "comment" ;  
CM_ BO_ <ident> "comment" ;  
CM_ SG_ <ident> <signal name> "comment";
```

<unit name> jméno uzlu

<ident> ID zprávy

<signal name> jméno signálu

- **Definice hodnot signálů**

V této sekci jsou definovány jména či označení pro určité hodnoty signálů. Má následující formát:

```
VAL_ <ident> <signal name> <val 1> "val 1 name" <val  
2> "val 2 name" ... ;
```

<ident> ID zprávy

<signal name> jméno signálu

Kapitola 4

Aplikace VDLviewer

Aplikace, která je cílem této diplomové práce, má za úkol vizualizovat data, která VDL během jízdy zaznamená. To vyžaduje nejprve zpracování dat uložených v textovém souboru, jenž VDL vygeneruje. Následně umožní uživateli prezentovat tato data v grafické formě, tedy v grafu zobrazit průběh zaznamenaných signálů ze všech tří modulů instalovaných ve VDL, a to jak v závislosti na čase, tak i v závislosti na libovolném vybraném signálu ze záznamu.

Dále je možné zaznamenanou trasu prohlížet na mapě v prostředí programu Google Earth. Program také obsahuje funkce pro práci se záznamem a možnost zobrazování zaznamenaných dat v reálném čase po připojení počítače k VDL.

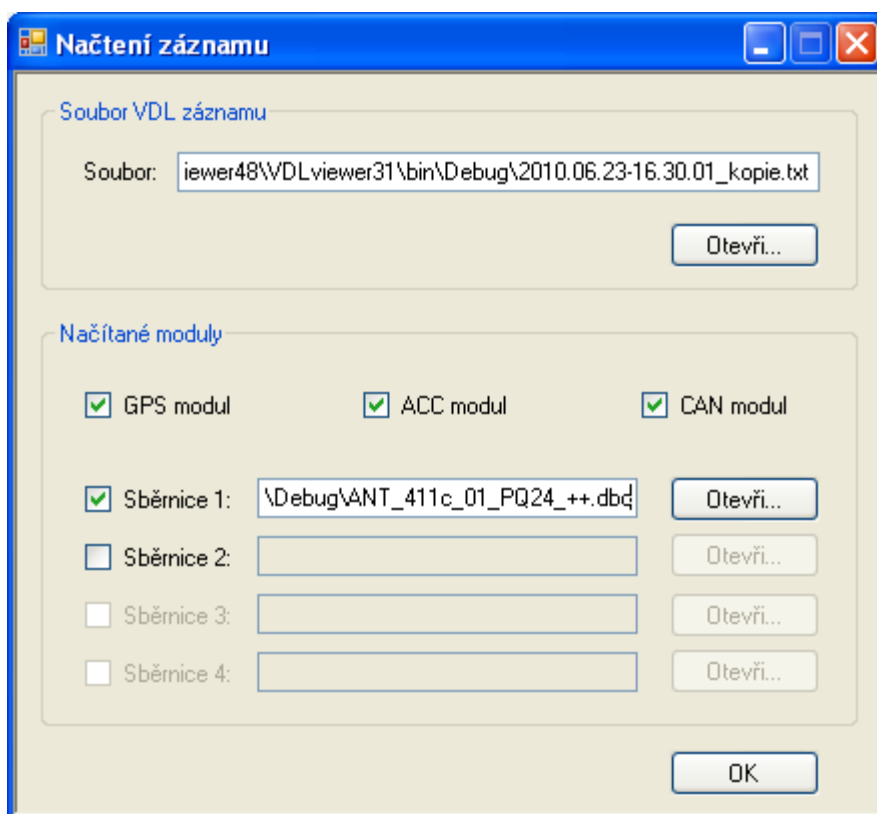
Pro tvorbu programu jsem zvolil prostředí Microsoft .NET Framework 3.5. Pro psaní kódu byl vybrán programovací jazyk C#. Důvodem bylo jeho velké rozšíření a úroveň, vhodná pro tvorbu aplikace tohoto druhu, podpora potřebného grafického rozhraní a kvalitní vývojové prostředí Visual Studio 2008. Následující text seznamuje čtenáře s jednotlivými vlastnostmi aplikace a její strukturou.

4.1 Načtení souboru do aplikace

Práce se záznamem začíná přes dialogové okno (viz obrázek 4.1), ve kterém je uživatel vyzván, aby zadal soubor se záznamem a zároveň DBC soubor příslušející ke sledované sběrnici. Tyto soubory mohou být až čtyři, neboť bylo požadováno, aby

záznamník umožňoval zaznamenávat data až ze čtyř sběrnic současně.

Načtením souboru se vytvoří instance třídy *VDLRecord*, která o záznamu uchovává všechny potřebné informace. Zároveň se v hlavním okně aplikace otevře záložka reprezentující tento záznam. První defaultní podzáložkou týkající se záznamu je záložka „Trasa“. Pokud je k dispozici připojení k internetu, vykreslí se v této záložce projetá trasa v prostředí Google Maps. Tato mapa slouží jako základní orientace v trase, jež je v záznamu obsažena. Po načtení souboru může uživatel se záznamem dále pracovat.



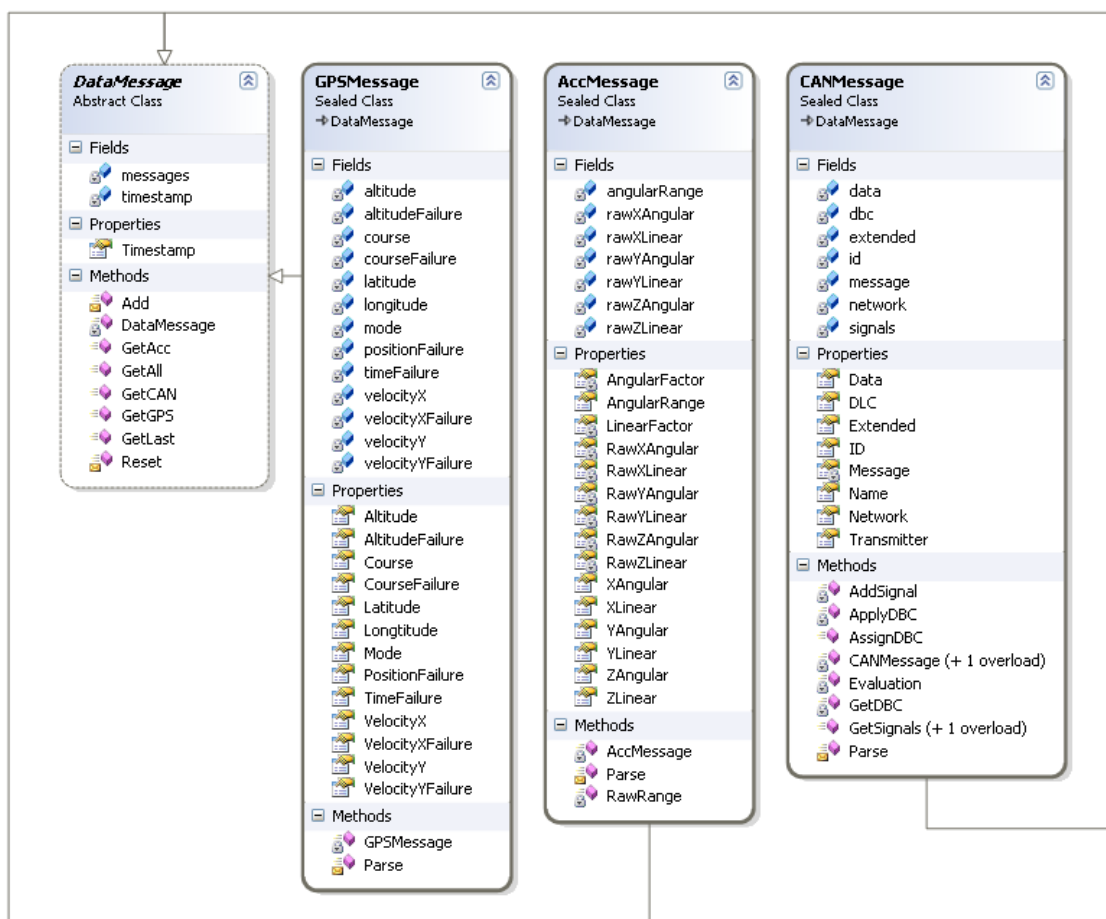
Obrázek 4.1: Dialogové okno načtení záznamu

4.2 Zpracování souboru se záznamem

Ke vstupnímu zpracování textových souborů z VDL slouží dvě knihovny *VLAParser.dll*, pro načítání záznamu, a *DBCParserBasic.dll* vyhodnocující soubory DBC. Autorem těchto knihoven je J. Kočí, který je napsal, ještě než bylo rozhodnuto

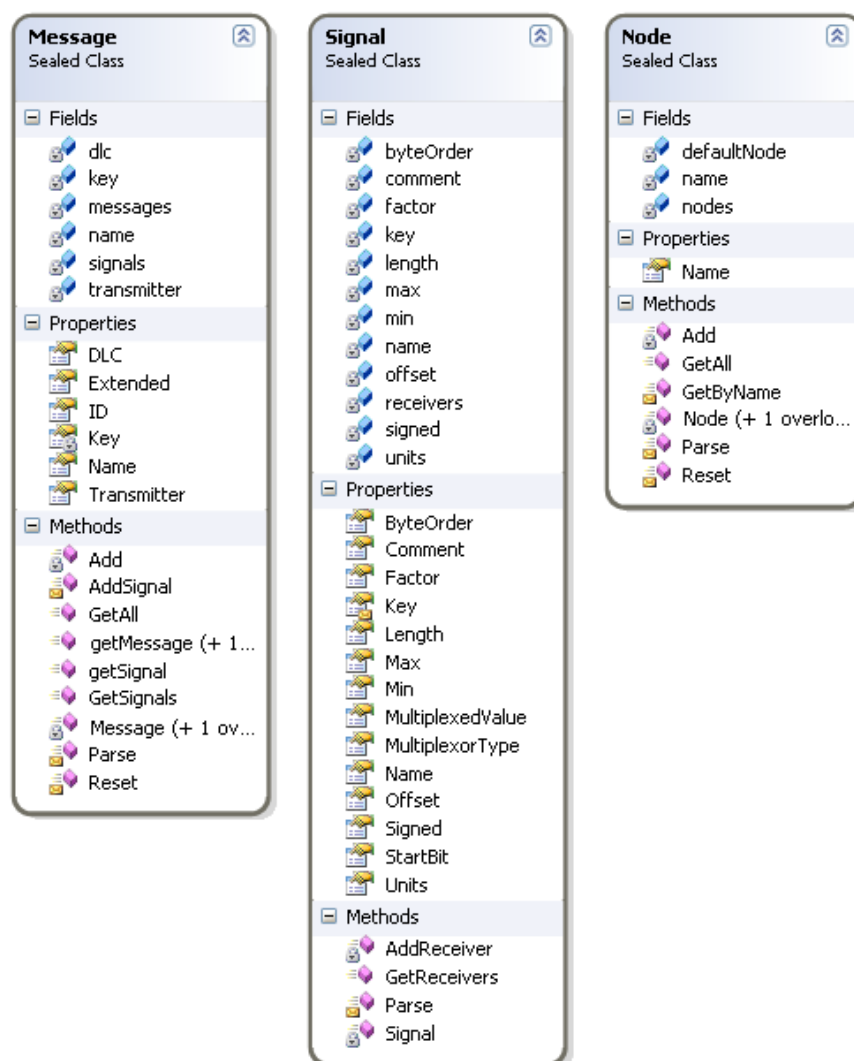
rozdělit projekt *Záznamníku parametrů trasy automobilu* na tři části – část věnující se hardwaru záznamníku, část týkající se jeho softwarového vybavení a třetí část, vizualizaci a prezentaci záznamu. Knihovny bylo třeba dodělat a mírně poupravit, aby vyhovovaly potřebám budoucí aplikace.

O načtení souboru se stará statická třída *VLAParser.Core*, jejíž metoda *OpenFile()* záznam projde a vyhodnotí obsah jednotlivých řádků záznamu. Při zpracování hlavičky souboru se uloží údaj o času započetí záznamu a o nastaveném rozsahu gyroskopu. Poté v záznamu následují jednotlivé zprávy. Zpráva z každého modulu je reprezentována vlastní třídou (*GPSTMessage*, *ACCMessage*, *CANMessage*), jež obsahuje datové položky s hodnotami signálů a údaj o časové značce zděděný od rodičovské třídy *DataMessage*. Obsahuje také metodu pro parsování textového řetězce s danou zprávou. Class diagram tříd je zobrazen na obrázku 4.2.



Obrázek 4.2: Class diagramy reprezentující jednotlivé zprávy

Nejkomplikovanější zpracování představují zprávy ze sběrnice CAN, jež je nutno interpretovat podle popisu sběrnice v souboru DBC. Při požadavku na načítání těchto zpráv ze sběrnice CAN je nejdříve získán popis komunikace po sběrnici ze souboru, resp. souborů DBC. Metoda *CANMessage.AssignDBC()* přiřadí seznam zpráv, které jsou po sběrnici přenášeny do statické proměnné *dbc*, což je pole typu *IEnumerable<DBCParserBasic.Message>* o velikosti 4, v němž položky 1 až 4 představují zaznamenávané sběrnice. Sestavení seznamů zpráv příslušných dané sběrnici obstarává metoda *DBCParserBasic.Core.OpenFile()*, která příslušný soubor DBC zpracuje a zprávy v něm definované uloží do tříděného seznamu typu *SortedList*. Diagramy tříd popisujících uzly sítě, zprávy posílané po sběrnici a signály v nich obsažené jsou na obrázku 4.3.



Obrázek 4.3: Class diagramy tříd pro popis komunikačních prostředků na sběrnici

Zpracovávání samotného záznamového souboru začíná zpracováním hlavičky, ze které se stanoví čas začátku záznamu a měřicí rozsah gyroskopu. Tyto údaje se uloží do nově vytvořené instance třídy *VDLRecord* a pokračuje se zpracováním řádků s jednotlivými zprávami. Řádky jsou procházeny v pořadí, jak byly uloženy, a vždy podle toho, od jakého modulu pocházejí, se zavolá příslušná parsovací metoda. Ta následně vytvoří objekt dané zprávy a zařadí ho do fronty

Queue <DataMessage> messages,

do které jsou uloženy všechny zprávy v záznamu v časové souslednosti. Při parsování zprávy z CAN sběrnice je navíc objektu třídy *CANMessage* přiřazena reference na objekt *DBCParserBasic.Message* (popis zprávy podle DBC), který je vybrán ze seznamu ve výše uvedené proměnné *dbc* podle shodné ID zprávy a čísla sběrnice. Když jsou všechny řádky souboru načteny a zprávy uloženy do fronty, do datových položek objektu *VDLRecord*

IEnumerable <GPSMessage> m_gpsMessages

IEnumerable <AccMessage> m_accMessages

IEnumerable <CANMessage> m_canMessages

IEnumerable <DataMessage> m_dataMessages

se nakonec uloží výčty zpráv z příslušných modulů a tento objekt v sobě již nese kompletní informace o zaznamenané jízdě a je možno s ním pracovat dál.

4.3 Chybějící údaje ve zprávě

Je třeba počítat s tím, že signál GPS nebude v některých okamžicích dostupný. Taková situace může nastat například v tunelu, husté vysoké zástavbě apod. V takovém případě je ve zprávě z GPS nahrazen nedostupný údaj znakem „?“.

Pokud je ve zprávě nedostupný údaj o zeměpisné šířce nebo délce, je taková zpráva z GPS neúčinná a nedojde k jejímu zařazení do fronty zpráv. Pokud není dostupný jiný údaj, je hodnota nastavena na hodnotu 1000, což signalizuje chybu – neplatnou hodnotu.

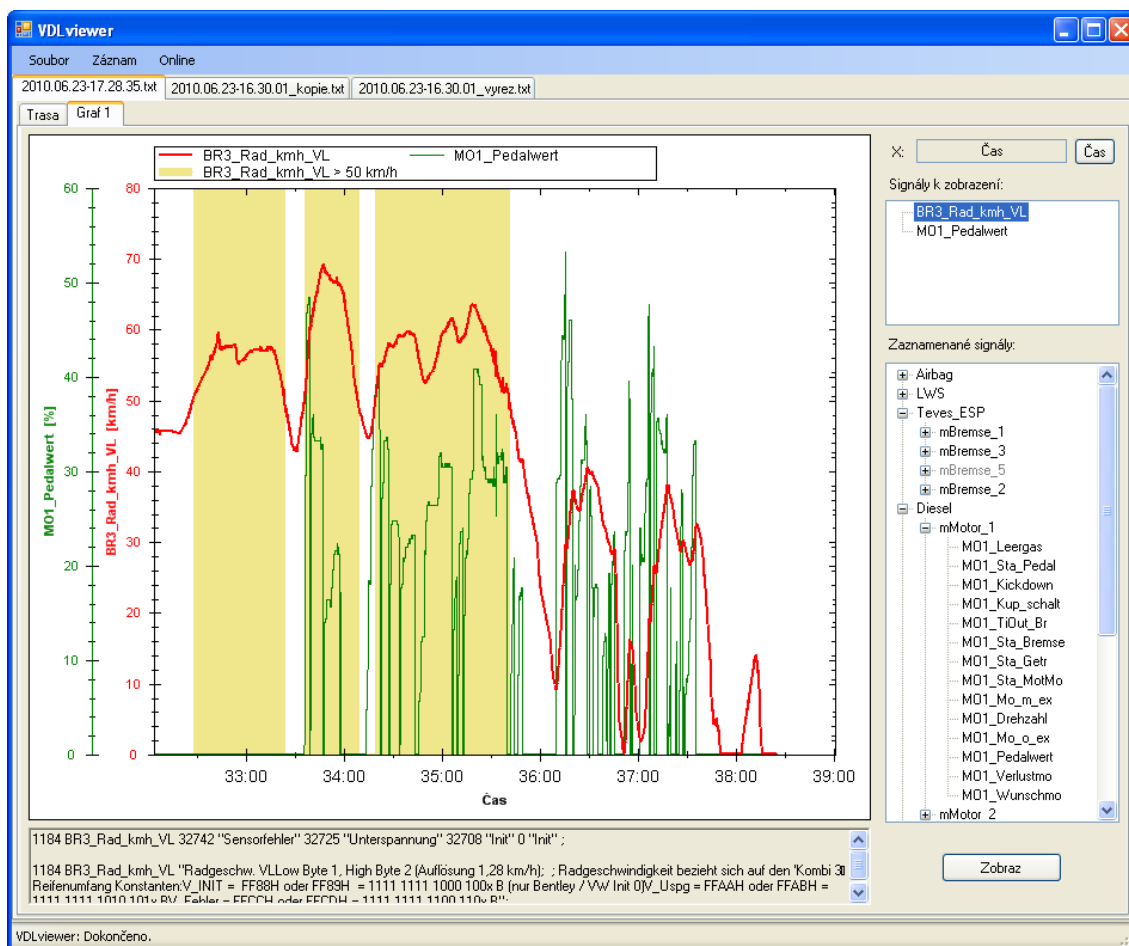
4.4 Operace se záznamem

Program VDLviewer umožňuje následující operace:

- otevření několika souborů současně
 - pro každý načtený záznam prohlížení dat ve více grafech
- zobrazení průběhů signálů v záznamu
 - v závislosti na čase
 - v závislosti na vybraném signálu
- nalezení oblasti v záznamu splňující zadaná kritéria
- zobrazení trasy v aplikaci Google Earth
 - jako prosté vyznačení projeté trasy
 - jako vyznačení projeté trasy s namodulovaným zvoleným signálem ze záznamu
 - zvýraznění vybrané oblasti
 - zvýraznění úseků splňující zadaná kritéria
 - možnost vybrat úsek a zobrazit průběhy signálů v něm v grafu
- editování záznamu
 - vynechat ze záznamu určité zprávy nebo snížit frekvenci jejich výskytu
- exportování dat ze záznamu
 - vybrané signály exportovat do souboru csv pro další zpracování
- přijímání a zobrazování dat z VDL online v reálném čase

Obrázek 4.4 ukazuje hlavní okno aplikace *VDLviewer*. Vidíme tři otevřené záznamové soubory. Každý je reprezentován vlastní záložkou, mezi nimiž je možné se volně přepínat. Pro práci je vybrán záznam 2010.06.23-17.28.35.txt.

Pro každý záznam je defaultně vytvořena záložka „Trasa“ zobrazující projitou trasu na mapě. Dále je vytvořena záložka „Graf 1“ a uživatel má možnost přidávat další záložky s grafy.



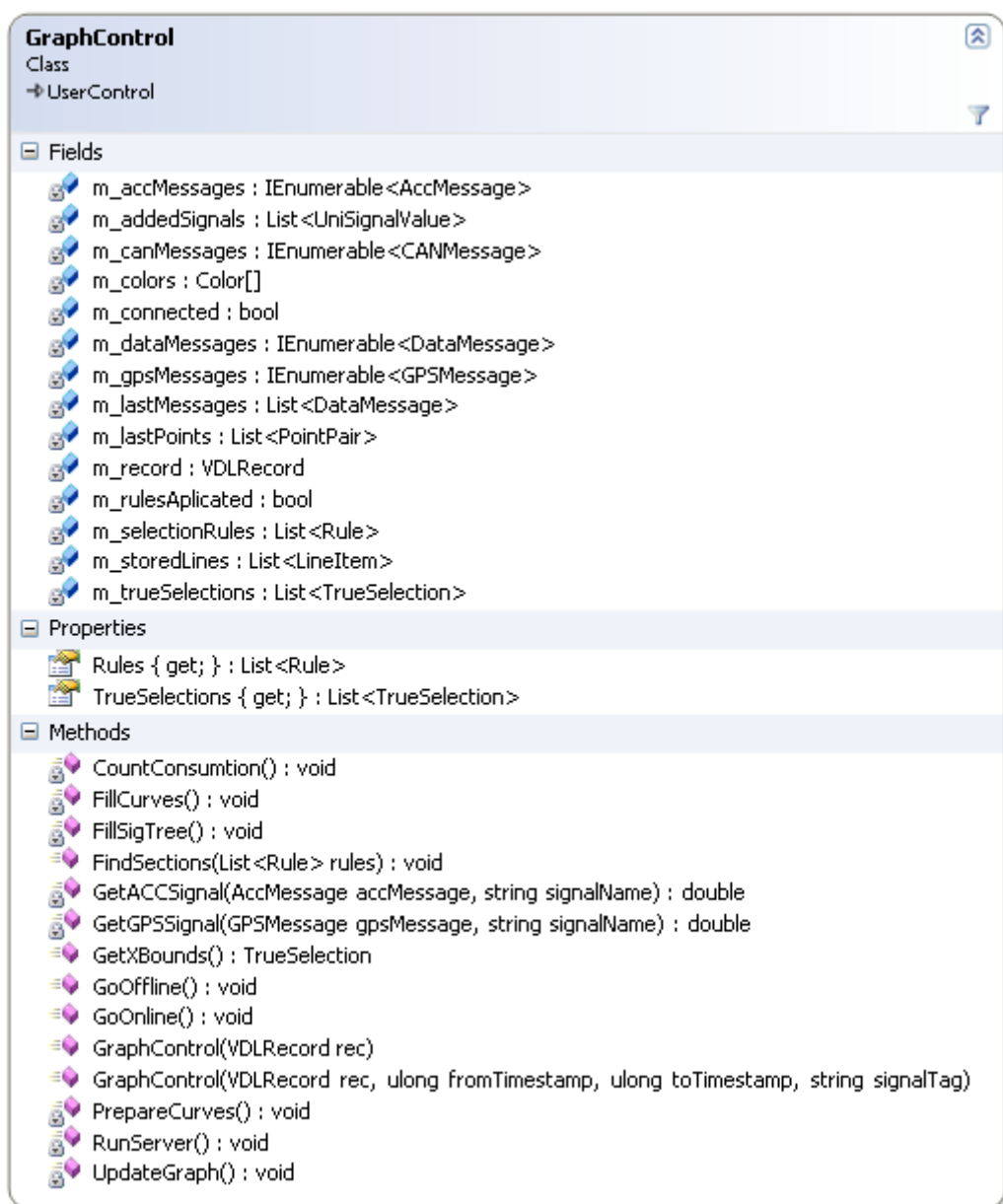
Obrázek 4.4: Hlavní okno aplikace

4.4.1 Zobrazování signálů v grafech

Základní funkcí, která se od programu očekává, je zobrazení časového průběhu signálů. Na obrázku 4.4 vidíme aktivovanou záložku *Graf 2* pro znázorňování průběhů signálů. Seznam *Zaznamenané signály* v pravé části okna obsahuje výčet signálů, které byly ze souboru načteny. Uživatel z tohoto seznamu vybírá signály, které si přeje zobrazit. Pokud je vybrán signál ze sběrnice CAN a v souboru DBC je k němu přiřazen komentář nebo je pro jeho hodnoty definován slovní popis, jsou tyto informace zobrazeny ve spodní části okna. Pokud se určitá CAN zpráva v záznamu nevyskytuje, ale v souboru DBC je definována, je v seznamu vyznačena šedivou barvou a její signály nelze do grafu přidat.

O konkrétní záložku s grafem se stará komponenta *GraphControl* odvozená od třídy *UserControl*. Jedná se o GUI sloužící k ovládání jednoho grafu. Class diagram třídy *GraphControl* je na obrázku 4.5. V diagramu jsou vynechány položky

představující ovládací prvky na formuláři, resp. jejich ovládací metody.

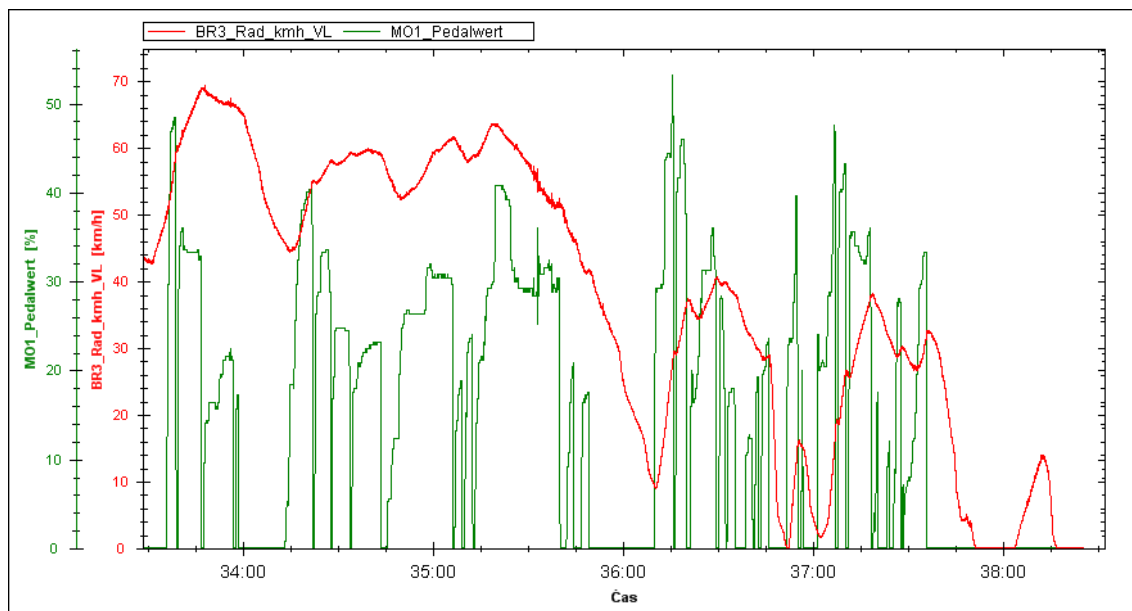


Obrázek 4.5: Class diagram třídy GraphControl

Pro vykreslování grafů jsem původně psal vlastní knihovnu. Ovšem to se ukázalo jako zbytečně složité a s ohledem na požadavky zobrazování jako je zoom, výběr oblasti v grafu atd. S ohledem na toto jsem se rozhodl pro jiné řešení. V duchu „netvořit, co už vytvořené je“ jsem našel knihovnu *ZedGraph*, která se ukázala jako vhodný prostředek pro moji aplikaci. *ZedGraph* je knihovna tříd pro kreslení 2D grafů v prostředí Microsoft .NET. Podrobný popis práce s touto knihovnou je uveden v [9]. Tvorba grafů s tímto nástrojem je velice flexibilní a dovoluje programátorovi požadavky na graf dobře specifikovat. *ZedGraph* je poskytován v licenci LGPL a je

dovoleno jej volně používat ve své aplikaci. Program VDLviewer používá verzi ZedGraph 5.1.5.

Do jednoho grafu je z důvodu přehlednosti možno vykreslit maximálně sedm signálů. Obrázek 4.6 znázorňuje průběhy signálů *BR3_Rad_kmh_VL* (rychlost otáčení předního levého kola) a *MO1_Pedalwert* (sešlápnutí pedálu plynu).



Obrázek 4.6: Zobrazení signálů v závislosti na čase

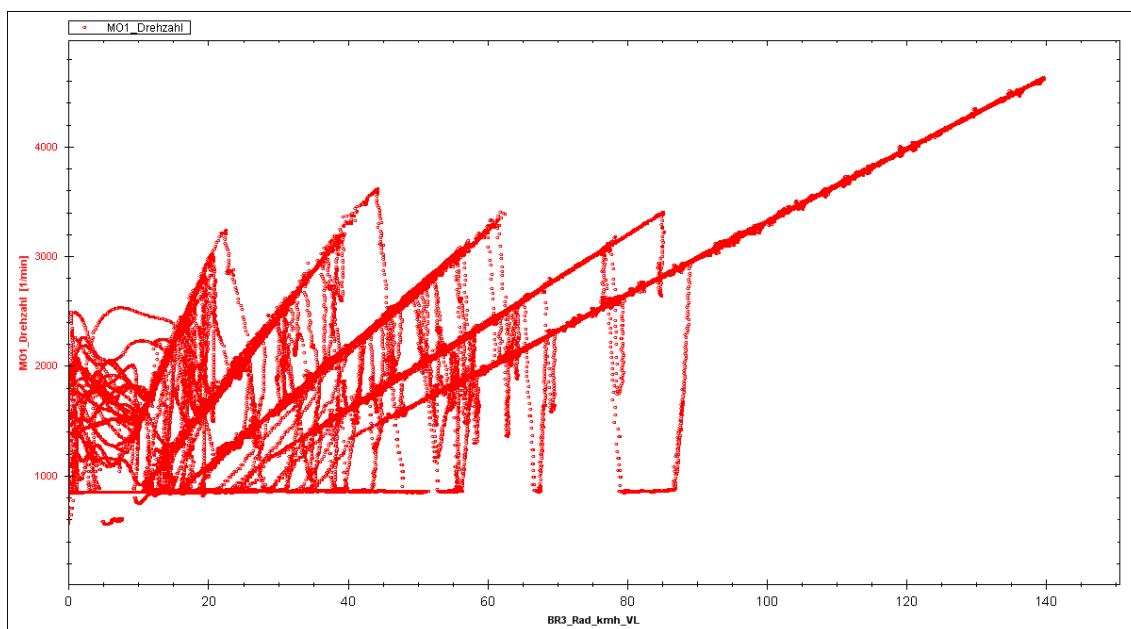
Na obrázku 4.6 zároveň také vidíme celkový vzhled komponenty ZedGraphControl. Oblast grafu je reprezentována třídou *GraphPane*. Ta v sobě uchovává nastavení grafu a spravuje zobrazované křivky.

Vykreslování signálu je prováděno ve dvou hlavních krocích. První má na starosti metoda *PrepareCurves()*, jenž pro zvolené signály vytvoří objekty křivek, resp. formát jejich vzhledu (barvu, tloušťku, styl čáry, legendu, fyzikální jednotky...) a přidá je do seznamu křivek *GraphPane.CurveList*. Jelikož rozdílné velikosti signálů (například otáčky motoru a rychlost) nedovolují vhodně zobrazit takovéto signály v jednotném měřítku na společné ose y, přiřadí každé křivce také její vlastní osu y. Ve druhém kroku pak metoda *FillCurves()* pro jednotlivé křivky v seznamu *CurveList* naplní jejich seznam bodů *Points* uloženými hodnotami signálů ze záznamu.

Druhou možností, jak zobrazit signály, je zobrazení nikoliv v závislosti na čase, nýbrž na zvoleném signálu ze záznamu. V horní části okna na obrázku 4.4 je komponenta *TextBox* indikující, jaká veličina je zvolena pro osu x. Zvolit jinou veličinu

než čas je možno přes kontextovou nabídku u příslušného signálu ve spodním seznamu signálů. Obrázek 4.7 pak ukazuje situaci, při které je jako nezávislá veličina zvolena rychlost. Za závislou veličinu jsou zvoleny otáčky motoru. V grafu je dobře patrných pět polopřímek, jež odpovídají jednotlivým rychlostním stupňům.

V případě zobrazení závislosti právě mezi signály, je vodítkem pro zobrazení bodů v grafu shodnost jejich časových značek. Respektive jde o hledání dvou časově nejbližších zpráv s těmito signály, neboť hodnoty signálů nejsou zaznamenávány ve shodném čase.

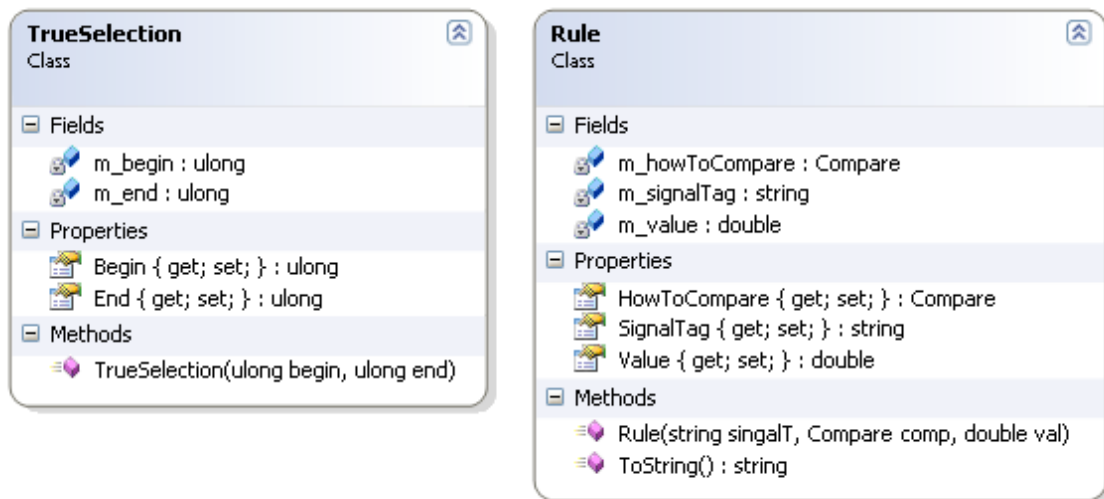


Obrázek 4.7: Zobrazení signálů v závislosti mezi sebou

4.4.2 Funkce Najít

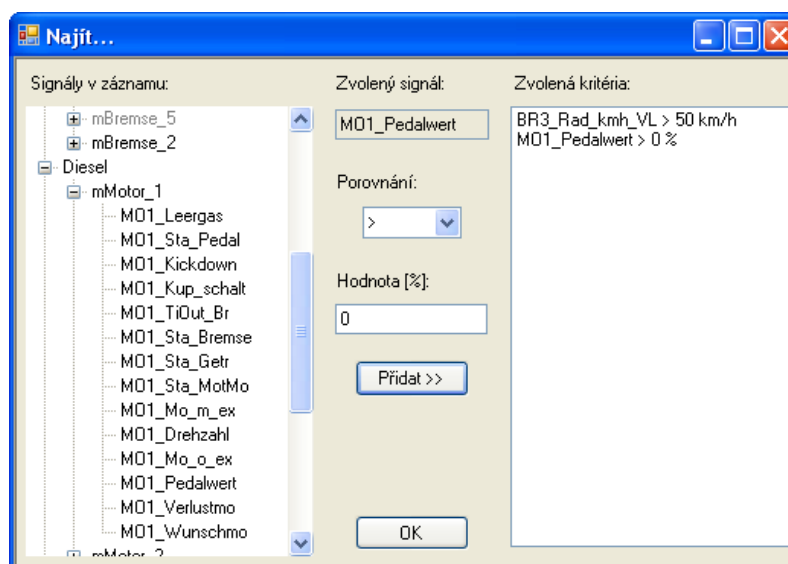
Pro rychlou orientaci v zobrazených signálech byla vytvořena funkce Najít. Touto funkcí může uživatel v záznamu najít a v grafu vyznačit úseky, které hodnotou vybraného signálu odpovídají zadaným kritériím.

Reprezentaci kritérií a úseků, které kritéria splňují, zajišťují třídy *Rule* a *TrueSelection*, jejichž class diagramy jsou na obrázku 4.8. Aplikace zadaných kritérií se děje ve funkci *FillCurves()*, jelikož pruhy vyznačující výběr, jsou ve své podstatě také další křivkou v grafu.

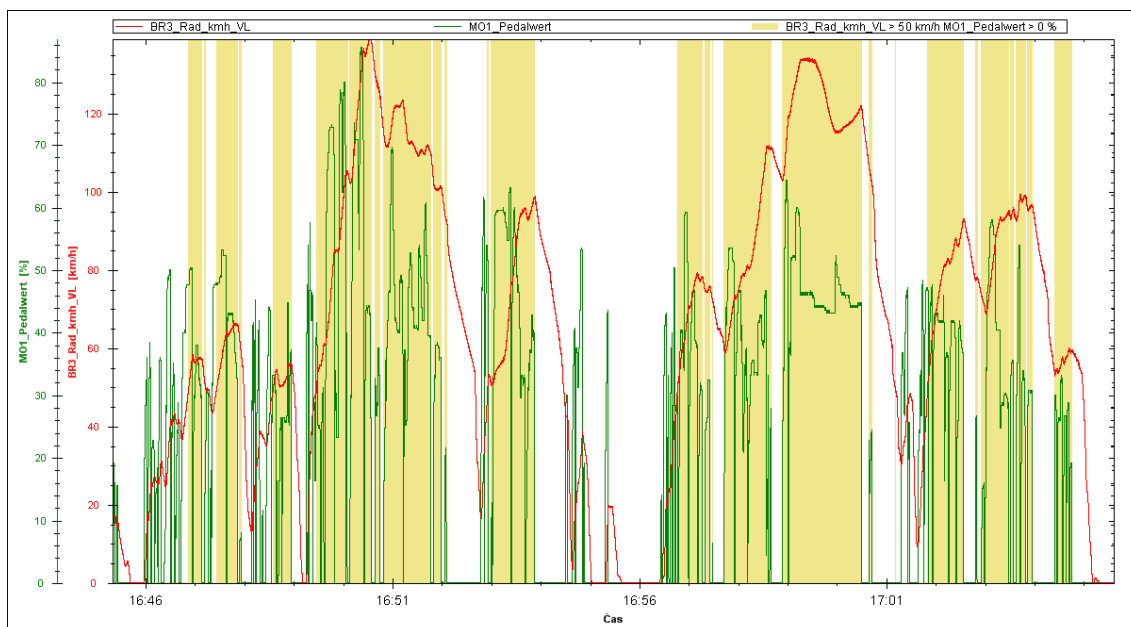


Obrázek 4.8: Class diagramy tříd *TrueSelection* a *Rule*

Je možné zadávat více kritérií, které se na záznam aplikují jako logický součin. Na obrázku 4.9 je zobrazeno dialogové okno se dvěma kritérii. Funkce následně v grafu označí úseky, které splňují tato kritéria, barevným pruhem, jak je vidět na obrázku 4.10.



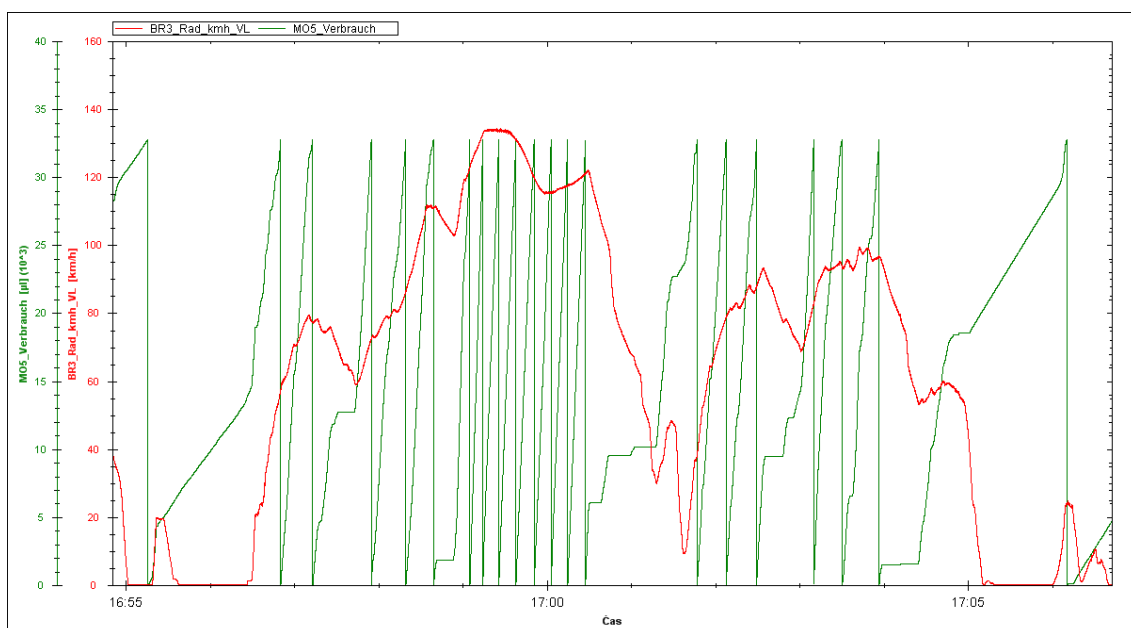
Obrázek 4.9: Dialogové okno Najít pro funkci hledání v záznamu



Obrázek 4.10: Výsledek hledání v záznamu

4.4.3 Zobrazení spotřeby paliva

Velice zajímavým a důležitým parametrem při provozu každého automobilu je jeho spotřeba paliva během provozu. Soubor DBC používaný pro popis sběrnice CAN v automobilu, ve kterém byly prováděny zkušební záznamy, definuje ve zprávě *mMotor_5* signál *MO5_Verbrauch*. Tento signál v sobě uchovává informaci o objemu spotřebovaného paliva. Jeho průběh je ukázán na obrázku 4.11.



Obrázek 4.11: Průběh signálu *MO5_Verbrauch*

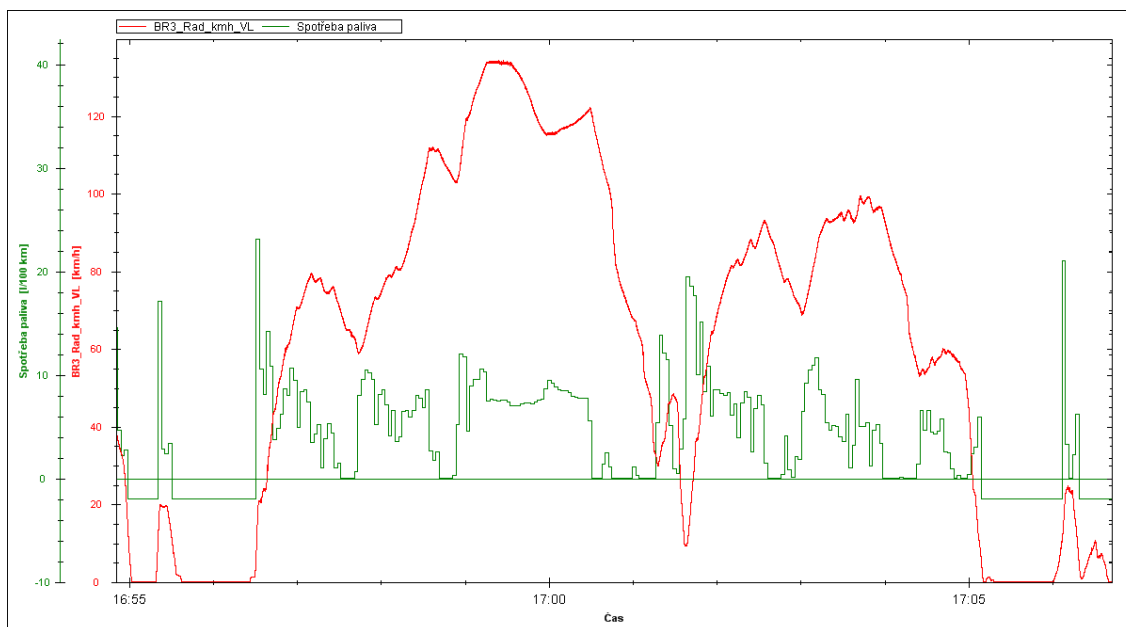
Signál je posílán cyklicky každých 20 ms a jeho hodnota udává množství spotřebovaného paliva v tomto intervalu v μl . Když hodnota signálu překročí jeho maximum 32 767 μl , začíná se počítat opět od nuly.

Podle strmosti růstu jeho průběhu se dá usuzovat na objem spotřebovaného paliva v daném okamžiku, ovšem pro uživatele má větší vypovídací hodnotu udávání spotřeby v tradičním způsobu počtu litrů na 100 km. Proto funkce *CountConsumption()* vypočítává z hodnot signálů *BR3_Rad_kmh_HL* (údaj o rychlosti ze systému ESP) a právě *MO5_Verbrauch* hodnotu spotřeby paliva v l/100 km.

Funkce zobrazení spotřeby je dostupná přes kontextovou nabídku po kliknutí pravým tlačítkem v seznamu „Signály k zobrazení“. Podle údajů o rychlosti spočítá ujetou vzdálenost a podle spotřebovaného paliva ve vybraném časovém intervalu přepočítá údaj na spotřebu na 100 km. Jelikož jsou hodnoty rychlosti pohybu automobilu zaznamenávány pravidelně se stálou periodou, je možné ujetou vzdálenost počítat pomocí aritmetického průměru zaznamenaných hodnot.

Velikost časových intervalů, pro které bude průměrována spotřeba, volí uživatel v dialogovém okně. Může tak zobrazit jak průměrnou spotřebu v průběhu celého záznamu, tak i graf okamžité spotřeby paliva.

V případě počítání okamžité spotřeby v okamžicích rozjíždění automobilu je spotřeba přepočítána na 100 km v poměru ke spotřebě během jízdy neúměrně vyšší a následná amplituda by v grafu působila rušivě. Proto je údaj o okamžité spotřebě počítán jen tehdy, pokud se automobil pohybuje rychlostí vyšší než 15 km/h. Při rychlosti do 15 km/h nebo stání je pak spotřebě přidělena záporná hodnota. Graf na obrázku 4.12 ukazuje výsledný graf s vypočtenou spotřebou.



Obrázek 4.12: Okamžitá spotřeba přepočítaná na l/100 km

4.5 Zobrazování projeté trasy na mapě

Aplikace zobrazuje trasu ve dvou nejznámějších světových mapách, které jsou k dispozici pro volné nekomerční použití. Jedná se o internetovou mapovou aplikaci Google Maps a virtuální glóbus Google Earth.

Oba tyto nástroje poskytují prakticky stejné služby, s tím rozdílem, že ke Google Maps se přistupuje přes okno prohlížeče, kdežto Google Earth je stáhnutelná aplikace, kterou je třeba nainstalovat na uživatelský počítač. Za jistou výhodu u Google Maps můžeme považovat to, že vybrané území zobrazuje pomocí mapy a ne pouze satelitních (resp. leteckých) snímků jako u Google Earth. Toto zobrazení může být někdy pro orientaci přehlednější.

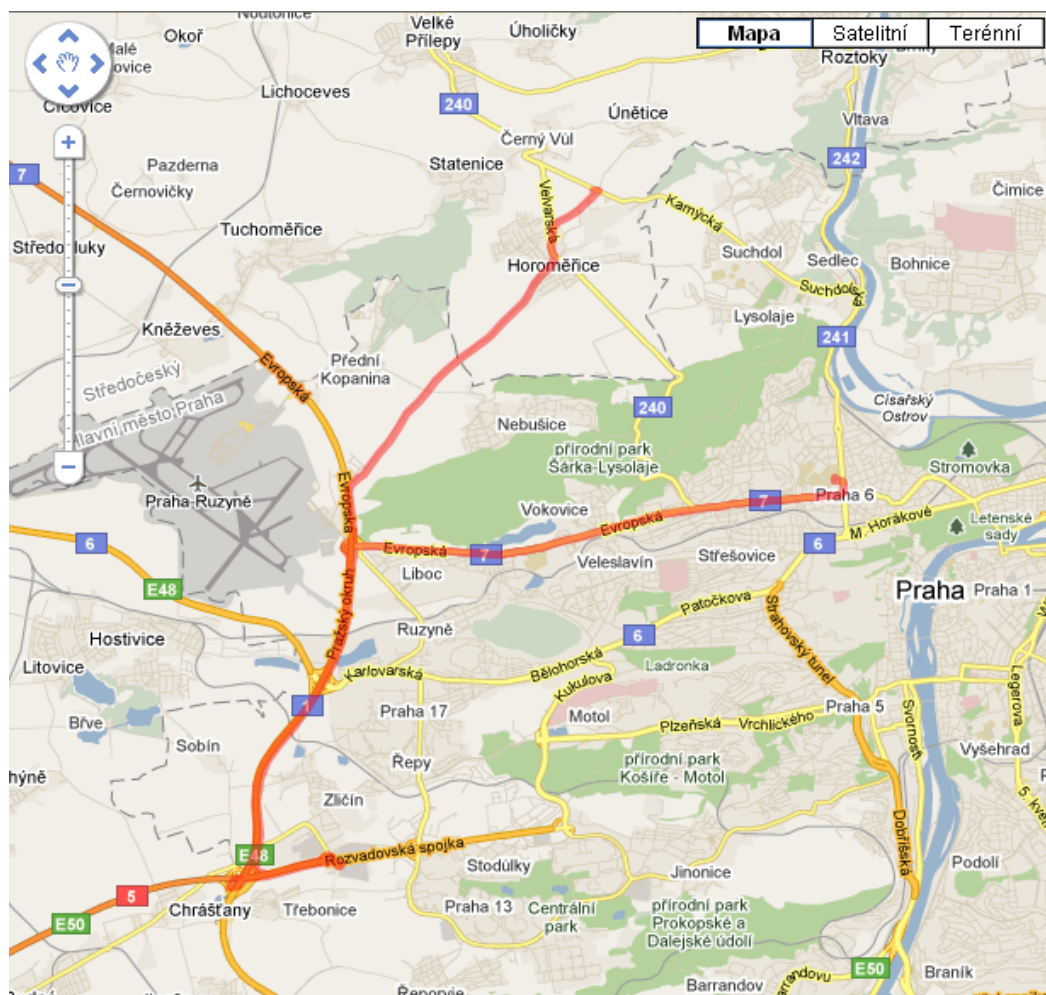
4.5.1 Interakce programu s Google Maps a Google Earth

Program používá jak Google Maps, tak i Google Earth. V Google Maps zobrazuje projetou trasu ihned po načtení záznamu, v Google Earth je pak uživateli k dispozici pokročilé prohlížení trasy.

Nejprve bylo nutné vyřešit komunikaci těchto dvou prostředků s programem psaným v jazyce C#. U Google Maps je k dispozici API využívající JavaScript. Příkazy

pro vykreslení trasy jsou tedy zasílány pomocí scriptu, který je umístěn v HTML stránce. Tato stránka je pak načtena do webového prohlížeče, jenž je do aplikace možno vložit jako komponentu *Webbrowser* do formuláře.

Obrázek 4.13 ukazuje trasu vykreslenou v prostředí Google Maps, tak jak je zobrazená v záložce Trasa po načtení záznamu.



Obrázek 4.13: Trasa zobrazená v prostředí Google Maps

Pro začlenění Google Earth do své aplikace existují tři možnosti:

- pomocí webového prohlížeče umístěného na formuláři, příkazy pak realizovat přes JavaScript
- použít API vytvořené pro ovládání Google Earth v prostředí .NET, okno s mapou GE mít otevřené přímo ve formuláři aplikace
- použít API pro .NET, ale aplikaci Google Earth mít puštěnou zvlášť, mimo uživatelskou aplikaci, a z té pak externě posílat příkazy pro Google Earth

Pro aplikaci VDLviewer jsem upřednostnil použití knihovny s API pro .NET, neboť tento způsob ovládání Google Earth je komfortnější a nabízí širší možnosti než JavaScript. Z možností, jestli mít okno s mapou zobrazené ve formuláři aplikace, nebo mít aplikaci Google Earth spuštěnou zvlášť, jsem vybral druhou možnost s tím, že ovládání bude realizováno přes okno, které bude mít property *TopMost* nastavenou na true, čímž bude tento ovládací panel stále „poruce“. Uvedená varianta má tu přednost, že aplikace Google Earth sama o sobě nabízí řadu funkcí, které by v případě zobrazování pouze okna s mapou nebyly pro uživatele k dispozici.

Nutnou podmínkou pro funkční spolupráci s Google Earth je nainstalování této aplikace na počítači. Propojení s .NET aplikací se pak uskuteční přidáním reference na soubor „googleearth.exe“ do seznamu Reference v projektu VDLviewer. V kódu se potom příkazem

```
EARTHLib.ApplicationGECClass ge = new EARTHLib.ApplicationGECClass()
```

vytvoří instance této třídy, přes niž je možné komunikovat s aplikací Google Earth.

Uživatelská data se do aplikace Google Earth přenášejí prostřednictvím souborů KML (Keyhole Marking Language). Struktura těchto souborů je založená na značkách (tzv. tagy) stejně jako je tomu u známějších formátů HTML a XML. Podle definovaných pravidel se do těchto souborů ukládají geografické informace, popisky, styly atd., které poté aplikace Google Earth zpracovává. Kompletní popis formátu KML a jeho použití je k dispozici v [7].

Spolupráci programu s aplikací Google Earth zajišťuje třída *GEarth*. Její class diagram je na obrázku 4.14.

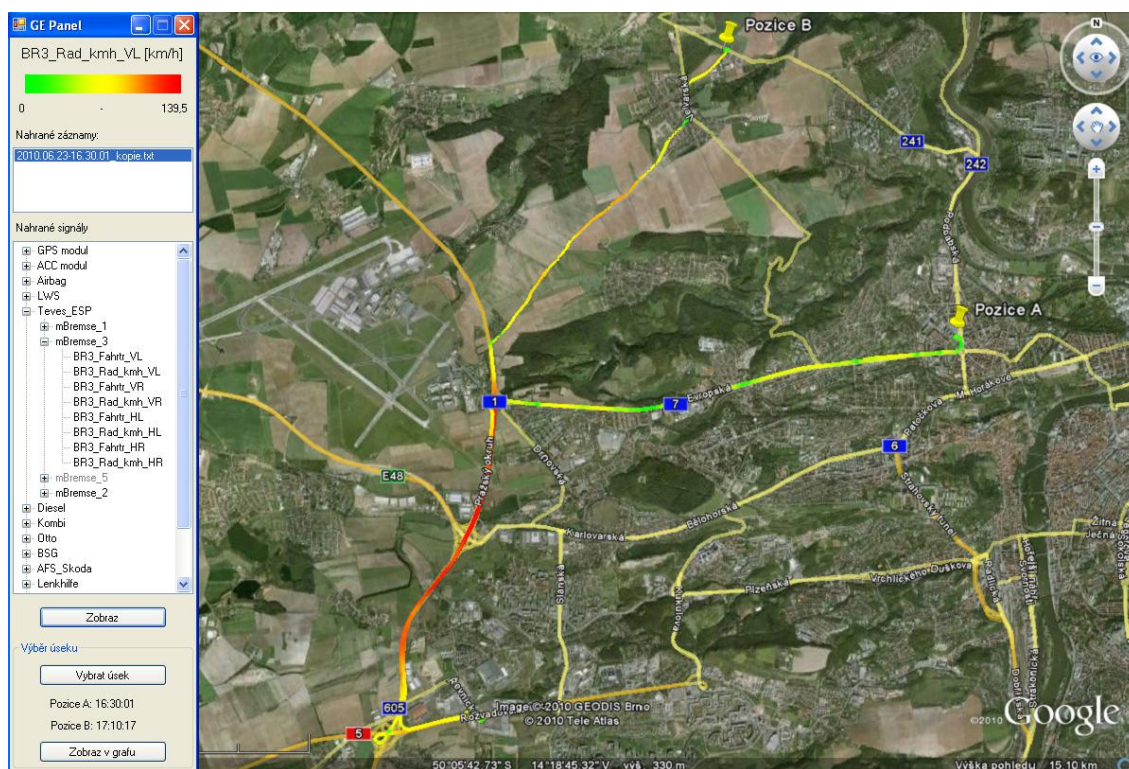
V programu VDLviewer používám soubory KML k vyznačování projetých tras, zvýrazňování jejich úseků a zobrazování značek určujících uživatelem vybraný úsek. Na obrázku 4.15 je vykreslena trasa z pořízeného záznamu.

Soubor vygenerovaný funkcí *MakeKml()* má následující strukturu:

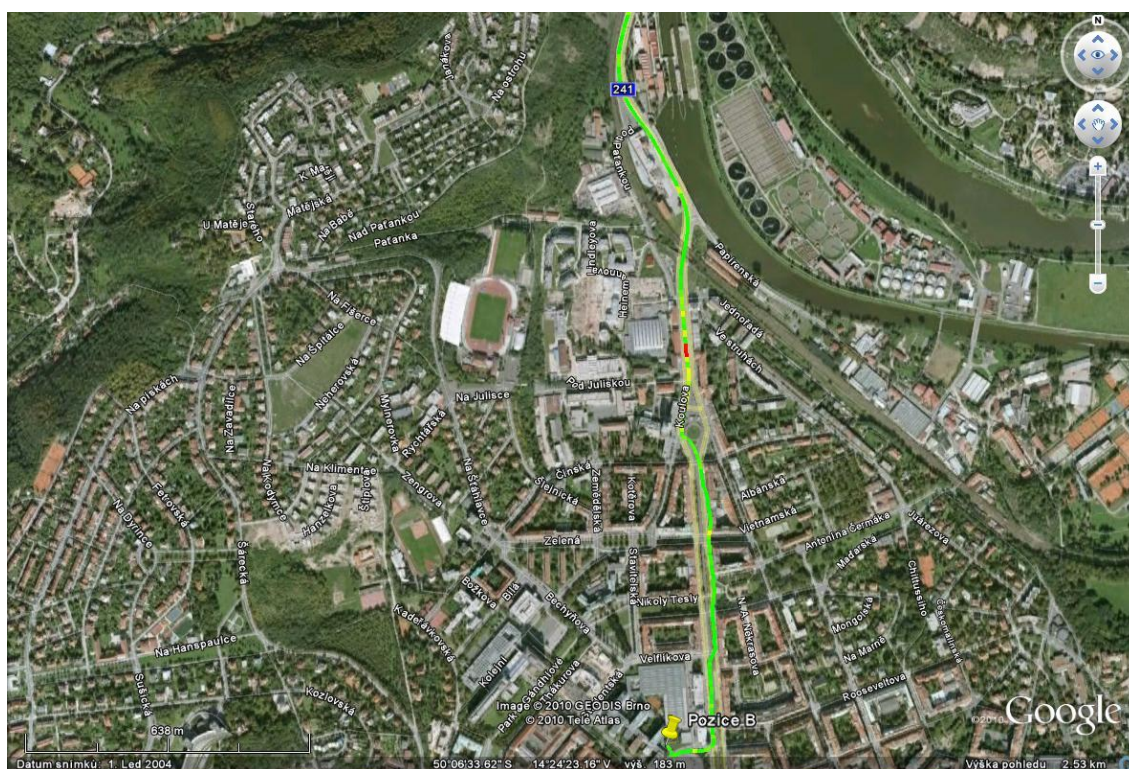
```
<?xml version="1.0" encoding="Windows-1252"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>2010.06.23-17.28.35.txt.kml</name>
    <Style id="LineStyle1">
      <LineStyle>
        <color>FFFF0000</color>
        <width>4</width>
      </LineStyle>
    </Style>
    <Placemark>
      <styleUrl>#LineStyle1</styleUrl>
      <LineString>
        <coordinates>
          14.3462,50.14131,295.9
          14.34634,50.14127,295.8
          14.34648,50.14122,295.6
          .
          .
          .
          14.39344,50.10268,234.9
          14.39344,50.10268,234.9
          14.39344,50.10268,234.9
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

V sekci Style je definovaná tloušťka a barva čáry. Na tento styl je pak uveden odkaz v sekci Placemark, v níž jsou obsažena geografická data, konkrétně zeměpisné souřadnice bodů trasy v sekci Coordinates.

Definování více stylů čar a možnost přiřazovat každé čáře jiný styl vedlo k myšlence vytvoření funkce, která by na zobrazenou trasu navíc namodulovala libovolný signál ze záznamu. Uživatel si pak díky barevnému gradientu zelená-žlutá-červená může udělat představu o hodnotách vybraného signálu v průběhu jízdy lépe než z grafu, neboť tímto způsobem jsou hodnoty signálu vizuálně zobrazeny přímo na mapě a tím pádem logicky svázány s danou polohou automobilu. Obrázek 4.16 ukazuje zobrazení trasy uvedeným způsobem, přičemž namodulovaným signálem je v tomto případě rychlost. Zároveň je na něm vidět plovoucí panel pro spolupráci s aplikací Google Earth. Na obrázku 4.17 je dobře patrný skok na nerovnosti ve vozovce.



Obrázek 4.16: Projetá trasa s namodulovaným signálem rychlosti, panel pro spolupráci s Google Earth



Obrázek 4.17: Část projeté trasy s namodulovaným signálem zrychlení ve směru Z (vertikální směr)

Podoba KML souboru, jenž na projetou trasu moduluje další signál, je následující:

```
<?xml version="1.0" encoding="Windows-1252"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>2010.06.23-16.30.01_kopie.txt.kml</name>
    <Style id="0">
      <LineStyle>
        <color>ff00FF00</color>
        <width>4</width>
      </LineStyle>
    </Style>
    <Style id="1">
      <LineStyle>
        <color>ff00FF00</color>
        <width>4</width>
      </LineStyle>
    </Style>
    <Style id="2">
      <LineStyle>
        <color>ff00FF00</color>
        <width>4</width>
      </LineStyle>
    </Style>
    .
    .
    <Style id="2415">
      <LineStyle>
        <color>ff00FFFB</color>
        <width>4</width>
      </LineStyle>
    </Style>
    <Placemark>
      <styleUrl>#0</styleUrl>
      <LineString>
        <coordinates>
          14.39353,50.10288,271.3
          14.39353,50.10288,271.3
        </coordinates>
      </LineString>
    </Placemark>
    .
    .
    <Placemark>
      <styleUrl>#2415</styleUrl>
      <LineString>
        <coordinates>
          14.34442,50.14151,295.2
          14.34442,50.14149,295.3
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

Zde je pro každé dvě za sebou jdoucí zaznamenané GPS pozice definován vlastní styl (resp. barva) a ten je pak v sekcích Placemark aplikován na příslušný segment trasy. Barva čáry příslušná úseku mezi dvěma souřadnicemi GPS se pak vypočítá podle hodnoty zvoleného signálu v daném úseku vzhledem k celkovému rozsahu jeho hodnot v záznamu. Výpočet barvy čáry v úseku mezi dvěma souřadnicemi GPS se u signálů z každého modulu provádí odlišně. Je to dáno jejich rozdílnými periodami zaznamenávání a jinou povahou vzhledem k požadovanému výsledku. Jedná-li se o signál z :

- modulu GPS:
 - použije se hodnota signálu obsažená v první z daných dvou zpráv GPS
- modulu inerciálního senzoru:
 - zrychlení ve směru X: použit součet hodnot signálu v časovém rozmezí daných dvou GPS zpráv podělený počtem příchozích zpráv s údajem o zrychlení (vypovídá o zrychlení/zpomalení automobilu ve směru jízdy)
 - zrychlení ve směru Y: použit součet absolutních hodnot signálu v časovém rozmezí daných dvou GPS zpráv podělený počtem příchozích zpráv s údajem o zrychlení (vypovídá o velikosti odstředivé síly)
 - zrychlení ve směru Z: Nejdříve je vypočten součet absolutních hodnot změn hodnot signálu (rozdílu dvou po sobě jdoucích hodnot signálu) v časovém rozmezí daných dvou GPS zpráv. Hodnota pro výpočet barvy je pak získána podělením tohoto součtu počtem zpráv s hodnotou zrychlení, které byly v tomto časovém rozmezí přijaty. (Interval mezi dvěma zprávami z modulu GPS je zhruba 1 s, tento způsob výpočtu nám sdělí, jak moc se automobil v tomto úseku „otřásal“ ve vertikálním směru.)
 - u senzoru úhlové rychlosti: použit součet absolutních hodnot signálu v časovém rozmezí daných dvou GPS zpráv podělený počtem příchozích zpráv s údajem o úhlové rychlosti (vypovídá o tom, jak moc se automobil v tomto úseku otáčel)
- sběrnice CAN:
 - použit aritmetický průměr hodnot signálu přijatých v časovém rozmezí daných dvou GPS zpráv

4.5.2 Výběr úseku záznamu z mapy

Jedním z požadavků na aplikaci byla také možnost vybrat z trasy vykreslené na mapě úsek a data z tohoto vybraného úseku pak zpětně zobrazit v grafu.

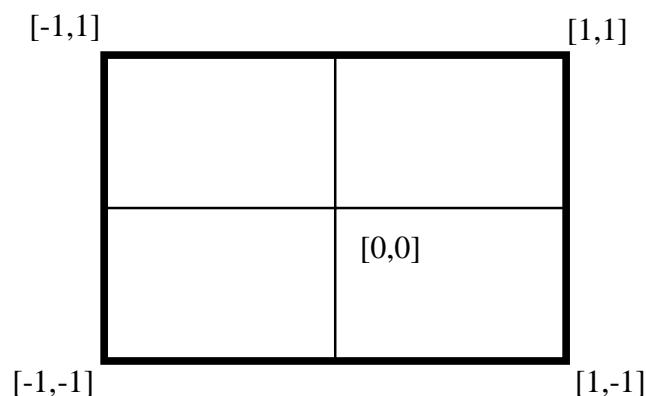
K provedení této operace jsem se rozhodl použít dvojici ukazatelů (Placemarks) „Pozice A“ a „Pozice B“ (jsou vidět i na obrázcích 4.15 a 4.16), které bude moci uživatel na mapě přemísťovat kliknutím v mapě a které budou označovat hranice vybraného úseku. Tyto ukazatele do aplikace Google Earth nahrávám opět souborem KML:

```
<?xml version="1.0" encoding="Windows-1252"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Hranice vyberu</name>
    <Placemark>
      <name>Pozice A</name>
      <Point>
        <coordinates>
          14.3462,50.14131,295.9
        </coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>Pozice B</name>
      <Point>
        <coordinates>
          14.39344,50.10268,234.9
        </coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Třída *EARTHLib.ApplicationGEClass* nabízí funkci, pomocí které je možné zjistit zeměpisné souřadnice bodu na mapě, na nějž ukazuje kurzor myši.

EARTHLib.PointOnTerrainGE point = ge.GetPointOnTerrainFromScreenCoords(x, y);

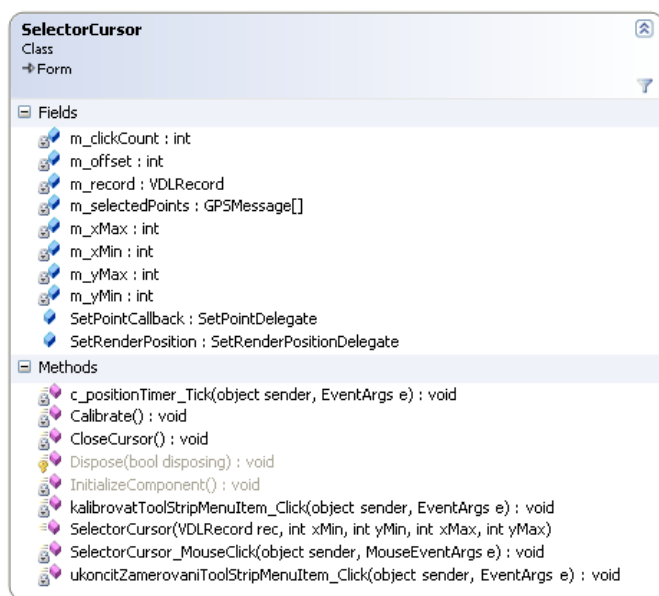
kde *x* a *y* jsou proměnné typu *double* a představují normalizované souřadnice okna mapy aplikace Google Earth, jak je znázorněno na obrázku 4.18.



Obrázek 4.18: Normalizované souřadnice v okně mapy Google Earth

Touto funkcí se tedy určí souřadnice bodu mapy, na který ukazuje kurzor. Bylo ovšem třeba vyřešit problém, jak zachytit událost kliknutí myši (což signalizuje uživatelské přání vybrat bod na trase), když C# umožňuje zachytávat tuto událost pouze, je-li kliknuto na nějakou komponentu nebo formulář programu.

Tento problém jsem vyřešil vytvořením nového formuláře *SelectorCursor*, jehož Property *FormBorderStyle* je nastavena na *None*, čili nemá ohraničení jako klasické okno. Jeho velikost je 7x7 pixelů a svou polohou sleduje pozici kurzoru myši, díky čemuž je již možné odchyťvat událost kliknutí myši i na ploše, která patří mapě Google Earth. Uživateli se tento formulář jeví jako zaměřovací ukazatel. Class diagram třídy *SelectorCursor* je na obrázku 4.19.

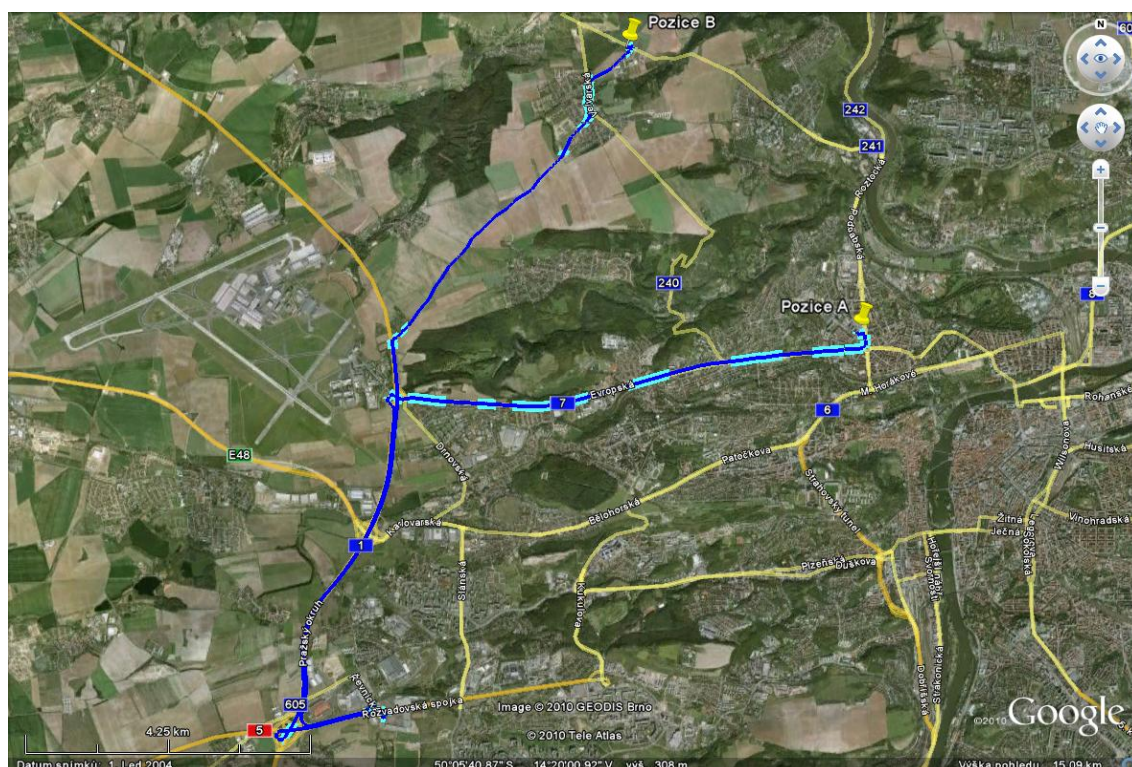


Obrázek 4.19: Class diagram třídy *SelectorCursor*

Pro získání přesných zeměpisných souřadnic bodu z mapy je potřeba znát umístění mapy aplikace Google Earth na obrazovce počítače. Jelikož velikost a pozici této mapy explicitně neznám, je po prvním zmáčknutí tlačítka „Vybrat úsek“ uživatel vyzván ke kalibraci, čili zaměření pozice mapy. Hodnoty se následně uloží, takže tento krok již není nutné dělat opakovaně při každém dalším výběru úseku. Nicméně při změně polohy mapy je možné kalibraci opět spustit přes kontextovou nabídku.

Po provedené kalibraci je možné vybírat úsek. Po každém kliknutí v oblasti mapy je vyhodnoceno, zdali je vybraný bod součástí trasy, či ne. Pokud ano, je bod vybrán jako první ukazatel Pozice A. Stejným způsobem dojde i k výběru Pozice B.

Po úspěšném vyznačení úseku dojde k přiblížení na jeho detail. Uživatel má možnost zobrazit data z vybrané části záznamu v separátním grafu. Je-li na čáru trasy namodulován nějaký signál, je automaticky do nově vytvořeného grafu zobrazen. S tímto grafem pak má uživatel možnost pracovat, jako kdyby se jednalo o samostatný záznam. V programu je vytvořena i opačná cesta pro přepínání mezi zobrazením v grafu a na mapě. Je tedy možné vybrat také oblast v grafu a jí příslušný úsek trasy nechat zobrazit na mapě. Třetí možností je zobrazení výběru vzniklého použitím funkce Najít. Tento případ ilustruje obrázek 4.20, kde jsou zvýrazněny úseky, kde rychlost nepřekročila 55 km/h.



Obrázek 4.20: Vyznačení úseku trasy na mapě

4.6 Export dat ze záznamu

Program VDLviewer slouží k obecné vizualizaci dat pořízených záznamníkem. Informace, které o jízdě záznamník pořídí, budou ovšem sloužit pro výzkum chování automobilu i v dalších, užším způsobem specializovaných aplikacích.

Předávání dat těmto aplikacím je plánováno přes soubory CSV (Comma-separated values), proto program umožňuje z otevřeného záznamu export vybraných signálů do tohoto formátu. Souborový formát csv slouží k výměně tabulkových dat. Sestává se z řádků, ve kterých jsou jednotlivé položky odděleny definovaným znakem, nejčastěji čárkou. Jelikož ale právě čárka v češtině figuruje jako oddělovač desetinných míst, je pro oddělování položek použit středník.

Po otevření exportového souboru v některém tabulkovém editoru jako např. MS Excel má tabulka formu jak ji můžeme vidět v tabulce 4.1. První sloupec tvoří časová značka, další sloupce patří signálům vybraným do exportu. V druhém řádku jsou uvedeny fyzikální jednotky signálů. Protože hodnoty různých signálů nejsou zaznamenávány v identický čas, jsou buňky signálů, pro které není v daný čas v záznamu definovaná hodnota, ponechány prázdné.

Tabulka 4.1: Formát exportu ze záznamového souboru

Timestamp	Zrychlení Z	BR3_Rad_kmh_VL	BR3_Rad_kmh_HL	MO1_Drehzahl
us	m/s2	km/h	km/h	1/min
1671				2117
8834		39,35	39,2	
12991				2124
21557				2124
28932		39,4	39,25	
32744	-1,04			
31838				2131
41554				2131
48799		39,53	39,38	
51555				2137
61587				2137
68895		39,53	39,45	
71580				2139
81585				2139
88871		39,6	39,5	
91565				2142
101555				2142
108820		39,66	39,68	
111571				2141
121606				2141
128904		39,68	39,93	
132468	-0,95			
131535				2142

4.7 Editace souboru se záznamem

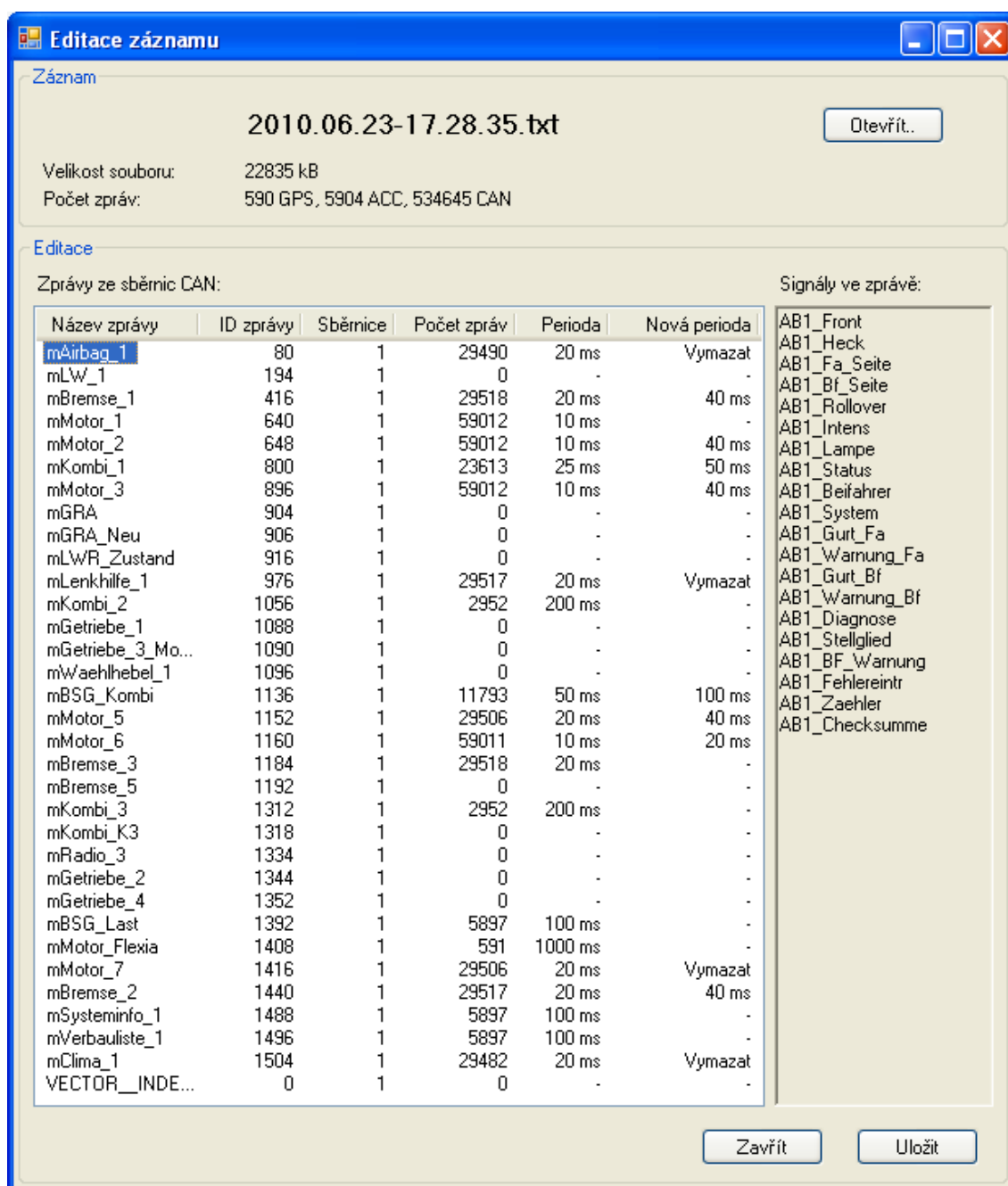
Velikosti záznamových souborů jsou relativně velké, jak můžeme vidět v tabulce 4.2. Největší podíl na velikosti souboru mají data ze sběrnice CAN. Zpracovávání tak velkých souborů je časově náročná operace a i následná práce s objemným záznamem je pomalejší.

Tabulka 4.2: Přehled statistiky pořízených záznamů

Délka záznamu	Snímané moduly	Velikost souboru	Počet zpráv
40 min	GPS, ACC, CAN	92,5 MB	$2,2 \cdot 10^6$
10 min	GPS, ACC, CAN	22,3 MB	$5,4 \cdot 10^5$
40 min	GPS, ACC	1,6 MB	$2,7 \cdot 10^4$
10 min	GPS, ACC	0,4 MB	$6,5 \cdot 10^3$

Aplikace VDLviewer proto umožňuje záznam „zeštíhlit“ podle konkrétních požadavků na přítomnost jednotlivých signálů. Funkce je přístupná přes menu Soubor, položka „Editace záznamu..“. Uživateli se zobrazí okno, ve kterém může vybrat záznam, jenž chce editovat. Tento záznam se pak zanalyzuje, jak ukazuje obrázek 4.21.

Ke každé CAN zprávě, která je definována v DBC souboru, se zobrazí počet jejích výskytů v záznamu a s jakou periodou je zaznamenávána (respektive se jedná o medián všech intervalů mezi jejími hodnotami v záznamu). Přes kontextovou nabídku vyvolanou pravým tlačítkem na zvolené zprávě se dá její perioda v záznamu zdvojnásobit, zčtyřnásobit, nebo lze zprávu ze záznamu zcela odstranit. Změny v záznamu, které zachycuje obrázek 4.21, zmenšily velikost souboru z 22,3 MB na 10,4 MB.



Obrázek 4.21: Přehled zaznamenaných zpráv a jejich výskytů

4.8 Online přenos a zobrazování dat

Záznamník umožňuje předávat snímaná data ke zpracování také v reálném čase. Po připojení VDL k počítači síťovým kabelem začne záznamník automaticky zasílat nahrávaná data také přes síťové rozhraní. Přenos se provádí přes protokol Telnet. Jedná se o protokol na aplikační vrstvě používaný v počítačových sítích pro spojení typu server-klient pomocí protokolu TCP.

Pro připojení k VDL slouží v menu nabídka Online – Připojit k VDL. Vytvoří se nová záložka s názvem „ONLINE“ a je-li síťové připojení k dispozici, naváže program se záznamníkem spojení (IPv4 adresa záznamníku je 192.168.9.1, port 1313) a chová se jako server, kterému záznamník posílá data ke zpracování.

Po přechodu do režimu online se v programu spustí nové vlákno pro příjem zpráv přes protokol TCP. V online režimu záznamník nejprve pošle souborovou hlavičku popsanou v kapitole 3 a následně posílá jednotlivé zprávy v textové podobě tak, jako když jsou zapisovány do souboru. Jelikož frekvence, s jakou zprávy přicházejí, je velká na to, aby každá přijatá hodnota signálu byla okamžitě vykreslena do grafu, je spuštěno ještě další vlákno. Toto vlákno každých 200 ms vykreslí do grafu hodnoty, které VDL v tomto čase odeslal.

Signály, které si uživatel přeje zobrazit v grafu, můžou být vybírány ze seznamu stejně jako při prohlížení záznamu uloženého v souboru. Odpojení se provede pomocí volby v menu Online – Odpojit od VDL. Po odpojení zůstávají hodnoty přijímaných signálů uloženy a je možno s nimi dále pracovat.

Kapitola 5

Závěr

Tato práce představuje třetí část projektu realizace zařízení, které během jízdy dopravním prostředkem, primárně automobilem, zaznamenává parametry projeté trasy. Výsledný záznam obsahuje údaje ze systému GPS, dále naměřené hodnoty zrychlení ve třech na sebe kolmých směrech, úhlové rychlosti ve třech na sebe kolmých osách a v poslední řadě záznam komunikace po sběrnici CAN.

Cílem třetí fáze projektu bylo vytvoření aplikace pro již zkompleťovaný záznamník, která bude zaznamenaná surová data vyhodnocovat a přehledně zobrazovat uživateli. Za účelem testování záznamníku VDL a aplikace VDLviewer byly pořízeny dva záznamy, které obsahovaly všechny výše zmíněné údaje. Prezentované výsledky v této práci pocházejí právě z těchto záznamů.

Aplikace popsaná v diplomové práci nabízí možnost zobrazovat naměřené hodnoty jednak ve formě grafu a jednak vykreslováním hodnot do mapy. S ohledem na zadání a účel používání aplikace byl důraz kladen na uživatelskou přívětivost aplikace a flexibilitu práce se záznamem. Té je dosaženo vzájemným propojením mezi dvěma uvedenými variantami zobrazení. Uživatel tedy může na mapě zobrazit úsek trasy odpovídající libovolné části záznamu označené v grafu a naopak. K měřeným údajům je možné přistupovat a zobrazovat je buď offline, z pořízeného záznamu, nebo online přes síťové rozhraní, v reálném čase tak, jak je VDL aktuálně zaznamenává. Z pořízených záznamů lze exportovat hodnoty signálů do souboru csv. Je možné také ze záznamu odstranit vybrané signály nebo snížit jejich četnost. To je výhodné zejména pokud chceme pracovat s dlouhým záznamem, který obsahuje velké množství dat.

Aplikace VDLviewer ze zaznamenaných údajů podává ucelený obraz o průběhu

jízdy v názorné a přehledné formě. Kombinací informací z modulů záznamníku VDL můžeme získat informaci o nerovnosti terénu, o akceleracích při rozjezdu, prudkosti projetí zatáček atd. Zajímavé informace poskytuje zkoumání informací ze sběrnice CAN. K dispozici tak jsou údaje o otáčkách motoru, sešlápnutí pedálu plynu, zařazeném rychlostním stupni a mnoho jiných. Z naměřených hodnot umí aplikace také spočítat průměrnou i aktuální spotřebu. Jisté omezení použití informací ze sběrnice CAN představuje nutnost znát popis komunikace na dané sběrnici. Takové popisy komunikace obsahují soubory DBC (viz kapitola 3), jejichž sehnání ovšem není samozřejmou záležitostí. Pro účely tohoto projektu nicméně příslušný soubor k dispozici byl a aplikace VDLviewer je tak funkční v celém rozsahu nabízených funkcí.

6 Literatura

- [1] KOČÍ, Josef. *Programové vybavení zápisníku parametrů trasy*. Praha, 2010. 46 s. Diplomová práce. ČVUT.
- [2] NOVOTNÝ, Michal. *Technické vybavení zápisníku parametrů trasy*. Praha, 2010. 49 s. Diplomová práce. ČVUT.
- [3] SPURNÝ, František. Controller Area Network. *Automatizace*. 1998, 41, 7, s. 397-400.
- [4] *DBC file format documentation*. [Online]
http://mon.email.free.fr/QIC/FTP/Docs/Caetec/NoticeDBC_File_Format_Documentation.PDF [cit. 2010-12-12].
- [5] Analog Devices. *ADIS16350/16355 Datasheet*. [Online] září 2009.
http://www.analog.com/static/imported-files/data_sheets/ADIS16350_16355.pdf.
- [6] TROELSEN, Andrew. *C# a .NET 2.0 profesionálně*. Brno: Zoner Press, 2006. 1197 s. ISBN 80-86815-42-0.
- [7] *KML reference*. [Online]
<http://code.google.com/intl/cs/apis/kml/documentation/kmlreference.html>
[cit. 2010-12-12].
- [8] *Google Maps Javascript API V3 Reference*. [Online]
<http://code.google.com/intl/cs/apis/maps/documentation/javascript/reference.html>
[cit. 2010-12-12].
- [9] *ZedGraph*. [Online] <http://zedgraph.org> [cit. 2010-12-12].
- [10] Global Positioning System. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2010-12-16]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Global_Positioning_System>.

7 Obsah CD

K této práci je přiloženo CD, na kterém je uložena elektronická verze dokumentu v pdf a aplikace *VDLviewer*.

- \valenpa2_DP2011.pdf – diplomová práce ve formátu pdf
- \VDLviewer – aplikace VDLviewer
- \VDLviewer_source – zdrojové kódy aplikace
- \Zaznamy – zkušební záznamy pořízené záznamníkem VDL