

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

Přenos dat mezi aplikacemi a jejich zpracování

Bakalářská práce

Aplikovaná informatika

vedoucí diplomové práce
Ing. Michal Šerý

vypracoval
Josef Maxa

České Budějovice, 2006

Knihovna JU - PF



3115172698

JIHOČESKÁ UNIVERZITA
V ČESKÝCH BUDĚJOVICÍCH
PEDAGOGICKÁ FAKULTA
KATEDRA INFORMATIKY
- 33 -

28. 4. 2006

Josef Maxa

Anotace

Obsahem práce je popis systému GPS(Jeho vznik, historie, složení. Další kapitoly se zabývají GPS přístroji a přesností udávané polohy.), jazyku Visual Basic.NET(Dozvíme se o zásadních změnách Visual Studia 2005 oproti starším verzím a konkrétněji se zaměříme na jazyk Visual Basic.NET.) a mnou vytvořené aplikace(Poslední část pojednává o aplikaci, kterou jsem vytvořil. Nalezneme zde popis vizuálního prostředí, popis činnosti funkcí a principy některých algoritmů.).

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně, a že jsem veškeré použité zdroje uvedl v Seznamu použitých zdrojů.

Josef Maxa

Josef Maxa

1.	Úvodem	1
2.	GPS.....	2
2.1.	ÚVODEM.....	2
2.2.	HISTORIE	2
2.3.	STRUKTURA SYSTÉMU.....	3
2.3.1.	<i>Kosmický subsystém</i>	3
2.3.2.	<i>Řídící podsystém</i>	4
2.3.3.	<i>Uživatelský podsystém</i>	4
2.3.4.	<i>NMEA protokol</i>	4
2.3.4.1.	\$GPRMC.....	4
2.3.4.2.	\$GPGGA.....	5
2.3.4.3.	\$GPGGL	5
2.3.4.4.	\$GPGSV	5
2.3.4.5.	\$GPGSA	6
2.3.4.6.	\$GPVTG	6
2.4.	PŘESNOST SYSTÉMU	7
2.4.1.	<i>Obecně</i>	7
2.4.2.	<i>Prakticky</i>	8
2.4.3.	<i>Horizontální poloha</i>	8
2.4.4.	<i>Nadmořská výška</i>	8
2.4.5.	<i>Shrnutí</i>	8
2.5.	GPS PŘIJÍMAČE	9
2.6.	ROZDĚLENÍ	10
2.6.1.	<i>Ruční</i>	10
2.6.2.	<i>Námořní</i>	10
2.6.3.	<i>Aplikační</i>	10
2.6.4.	<i>Letecké</i>	11
2.6.5.	<i>Pro kapesní počítače PDA</i>	11
3.	Vývojové prostředí	12
3.1.	ÚVOD	12
3.2.	TEAM FOUNDATION SERVER	13
3.3.	ANALÝZA KÓDU	13
3.4.	REFAKTOROVÁNÍ A ZNÁZORŇOVÁNÍ	13
3.5.	CODE SNIPPETS.....	14
3.6.	KOMPONENTY	14
3.7.	ZABEZPEČENÍ A ZVEŘEJNĚNÍ APLIKACE	15
3.8.	TECHNOLOGIE CLICKONCE	15
3.9.	DROBNOSTI KTERÉ POTĚŠÍ.....	16
3.10.	POŽADAVKY NA PROVOZ	17
4.	Aplikace GPS	18
4.1.	ÚVOD	18
4.2.	UŽIVATELSKÉ ROZHRAŇÍ	19
4.2.1.1.	Záložka „GPS zařízení“	19
4.2.1.2.	Lišta záložek.....	20
4.2.1.3.	Připojení.....	20
4.2.1.4.	Test.....	20
4.2.1.5.	Doba zpracování.....	20
4.2.1.6.	Mapa	20
4.2.1.7.	Stav	20

4.2.1.8.	Spustit	20
4.2.2.	Záložka ,Data‘	21
4.2.2.1.	Nahrávání dat	21
4.2.2.2.	Data	22
4.2.3.	Záložka ,Údaje‘	22
4.2.3.1.	Záložky s NMEA větami	22
4.2.3.2.	Tělo NMEA věty	22
4.2.4.	Záložka ,Simulátor‘	23
4.2.4.1.	Data	23
4.2.4.2.	Volby	23
4.2.4.3.	Spuštění simulátoru	23
4.2.5.	Záložka ,Přehled satelitů‘	24
4.2.5.1.	Vykreslení satelitů	24
4.2.5.2.	Detaily satelitů	24
4.2.5.3.	Informace o zpracování cyklu	24
4.2.6.	Formulář ,Mapa‘	25
4.2.6.1.	Kotvící body	26
4.2.6.2.	Informace	26
4.2.6.3.	Ovládací tlačítka	26
4.2.6.4.	Vyčištění plochy	26
4.3.	POPIS PROGRAMOVÉ ČÁSTI	27
4.3.1.	Struktura aplikace	27
4.3.1.1.	Třída ,Protokol‘	27
4.3.1.2.	Třída ,funkce‘	28
4.3.1.3.	Formulář ,Main‘	30
4.3.1.3.1.	Funkce ,Připojit‘	30
4.3.1.3.2.	Funkce ,Odpojit‘	31
4.3.1.3.3.	Funkce ,Čti‘	32
4.3.1.3.4.	Funkce ,Čti větu‘	33
4.3.1.3.5.	Jádro systému	34
4.3.1.4.	Formulář ,Zobrazování na mapě‘	34
4.3.1.4.1.	Zobrazení formuláře	34
4.3.1.4.2.	Nastavení kotvících bodů	35
4.3.1.4.3.	Algoritmus vykreslování	35
5.	Závěr	36
6.	Použité zdroje	37

1. Úvodem

Primárním cílem bakalářské práce bylo využít dovednost programování v praxi a předvést schopnosti jazyka Visual Basic.NET, který je v nové verzi implementován ve Visual Studiu 2005.

Sekundárním cílem práce bylo proniknout do problematiky systému GPS a porozumět, z čeho se systém skládá a jak funguje, dále vytvořit aplikaci, která bude schopná číst a zpracovávat data z GPS zařízení.

Dokument je rozdělen do tří hlavních částí.

1. Popis systému GPS (vznik, složení, využití,...)
2. Vývojové prostředí (VB.NET, novinky ve Framework 2.0)
3. Popis vytvořené aplikace

2. GPS

2.1. Úvodem

GPS (Global Positioning System¹) je družicový navigační systém a patří mezi pasivní dálkoměrné systémy². Pomocí GPS přístroje(přijímače) určíte svoji polohu kdekoli na zemském povrchu, bez ohledu na počasí a na dobu, kdy měříte.

2.2. Historie

Program družicové navigace rozvíjely americké vzdušné síly a námořnictvo od začátku 60. let. Jeho vývoj probíhal ve třech hlavních fázích. Čtvrtá představuje zásadní milník, kdy se systém stal plně přístupný široké veřejnosti.

1. 1973-1979 byly zahájeny práce na systému GPS. První vypuštěné družice sloužily pro testování a vývoj celé technologie systému. První družice byla vypuštěna počátkem roku 1978 a na konci tohoto roku byly k dispozici družice čtyři. Umožňovaly třírozměrnou navigaci po omezenou dobu a to jen na testovacím polygonu v Arizoně. Družice vyrobené v této fázi nesly označení družice bloku I.
2. 1979 -1985 probíhalo budování řídicího a monitorovací střediska. Začalo ostré testování GPS přijímačů a jejich další vývoj.
3. 1985-1994 byly vypouštěny družice II. generace(bloku) a v roce 1993 bylo dosaženo počátečního operačního stavu³. Po důkladném zkoušení systému bylo v roce 1995 dosaženo plného operačního stavu⁴.
4. 2000 1.květen. Americká armáda zrušila selektivní dostupnost⁵, tím se pro civilní obyvatelstvo zvýšila přesnost určení polohy téměř 10- ti násobně (0-15 m).

¹ Globální polohový systém. Někdy bývá také označován jako NAVSTAR (Navigation System using Time And Range)

² Družice vysílají signály a uživatel zjistí čas jejich příjmu. Vzdálenost ke družicím je určena z doby, která uplynula mezi vysláním a příjmem signálů. Družice vyšlou svou polohu. Z polohy družic a ze vzdálenosti družic zjistí uživatel svou polohu.

³ Na oběžných drahách je 24 družic a v libovolném místě na Zemi lze určit polohu.

⁴ Plný bezproblémový provoz.

⁵ (Selected Availability) - záměrné zhoršování přesnosti určení polohy a zavedení tzv. přesného P/Y - kódu, kterým je šířen signál pouze pro vojenské aplikace(hlavně kvůli obavám ze zneužití teroristy).

2.3. Struktura systému

Systém GPS se skládá ze 3 podsystémů.

2.3.1. Kosmický subsystém

V současné době je tvořen 24 družicemi, z čehož 3 slouží jako záložní (další záložní družice jsou na Zemi a lze je zavést do systému během 2 dnů). Ty krouží kolem Země ve výšce přibližně 20 000 km na 6 oběžných, velmi přesně vypočítaných drahách, skloněných vždy o 55 stupňů vzhledem k rovníku. Dráhy družic jsou vypočteny tak, aby v daném okamžiku byl v kterémkoliv místě na zemské kouli viditelný nad obzorem dostatečný počet družic, nezbytný k přesnému zaměření (bylo ověřeno, že v oblasti celé naší republiky je v kteroukoliv denní či noční hodinu nad obzorem nejméně 6 až 8 družic navigačního systému GPS). Každá družice je vybavena přijímačem, vysílačem, atomovými hodinami a řadou přístrojů, které slouží pro navigaci nebo jiné speciální úkoly (např. pro detekci výbuchu jaderných náloží). Doba oběhu je přibližně 11 hodin 58 minut. Navigační signály družice se vysílají na kmitočtech 1575,42 MHz a 1227,6 MHz. Družice přijímá, zpracovává a uchovává informace předávané z pozemního řídicího centra, na základě kterých koriguje svoji dráhu raketovými motorky, dále sleduje stav vlastních systémů a podává o těchto skutečnostech informace zpět do řídicího centra. Pro případné problémy je každá družice vybavena záložními zdroji, palubní baterie jsou dobíjeny dvěma slunečními panely.

Samotný princip určování polohy systémem GPS je následující: družice vysílá signály pro uživatele v podobě složitěho signálu. Každá družice vysílá zprávy o své poloze a přibližné polohy ostatních družic systému. K určení aktuální polohy Váš přijímač počítá tzv. pseudovzdálenosti, což jsou vzdálenosti mezi vaším přijímačem a viditelnými družicemi (nad obzorem). Výpočet pseudovzdálenosti vychází ze znalosti rychlosti šíření družicového signálu a rozdílu času mezi vysláním a příjmem signálu. Termín pseudovzdálenost se zavádí proto, že je nutné zavádět další doplňující výpočty, které určení výsledné polohy dále zpřesňují. Pro určení dvojrozměrné polohy (nejčasněji zeměpisná délka a šířka) postačí příjem signálu z min. tří družic (výpočet tří pseudovzdáleností), pro určení trojrozměrné polohy (navíc výška) minimálně ze čtyř družic.

Příjem menšího počtu družic znemožňuje výpočet polohy, vyšší počet družic naopak určení polohy dále zpřesňuje.

2.3.2. Řídící podsystém

Monitoruje funkce družic a získané údaje předává zpět družicím. Řídící podsystém tvoří 5 monitorovacích stanic a 3 pozemní řídicí stanice vhodně rozmístěných po celém světě, které spolupracují s hlavní řídicí stanicí, neustále přijímá signály z družic a předává je hlavní řídicí stanici ukryté v opevněném bunkru ve Skalistých horách v Colorado Springs, USA.

Cílem celého řídicího podsystému je monitoring funkcí každé družice, sledování a výpočet dráhy družice, komunikace a zajištění přesného chodu atomových hodin na družicích. Jakákoliv závada na družici musí být co nejrychleji operativně řešena, při ceně 50 milionů dolarů za družici je to pochopitelné.

V současné době existuje několik nezávislých monitorovacích sítí, které umožňují další přesnější určování polohy, především pro velmi přesné aplikace (geodézie, geodynamika). Tyto sítě se nepodílejí na řízení a činnosti systému GPS.

2.3.3. Uživatelský podsystém

Pro příjem a zpracování GPS signálů byli vyvinuty speciální přijímače. Kromě speciálních přijímačů určených pro vojenské aplikace, existuje dnes řada dalších typů GPS přijímačů.

2.3.4. NMEA protokol

V té to sekci se budu zabývat specifikací NMEA protokolu. Uvedu zde jen ty věty, které využívá aplikace.

2.3.4.1. \$GPRMC

Doporučené minimum přenášovaných informací z GPS (RMC).

\$GPRMC,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,<10>,<11> *hh<CR><LF>

<1> UTC čas, formát hhmmss (hodiny|minuty|vteřiny)

<2> stav zobrazovaných informací, A = platná pozice, V = pozice není platná

<3> zeměpisná šířka, formát ddmm.mmmm (stupně|minuty.desetinné vyjádření zbytku)

<4> určení zemské polokoule u zeměpisné šířky N/S (severní/jižní)

<5> zeměpisná délka, formát ddmm.mmmm (stupně|minuty.desetinné vyjádření zbytku)

<6> určení zemské polokoule u zeměpisné délky W/E (západní/východní)

<7> rychlost nad zemí 0.0 až 999.9 uzlů

<8> azimut pohybu 000.0 až 359.9 stupňů

<9> UTC datum, formát format ddmmrr (den |měsíc|rok)

<10> hodnota magnetické odchylky, rozpětí 000.0 až 180.0 stupňů

<11> směr magnetické variace, VÝCHOD-ZÁPAD

2.3.4.2. \$GPGGA

Informace o naměřené poloze a výšce (GGA).

\$GPGGA,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,<10>,<11>,<12> *hh<CR><LF>
<1> UTC čas, formát hhmmss (hodiny|minuty|vteřiny)
<2> zeměpisná šířka, formát ddmm.mmmm (stupně|minuty.desetinné vyjádření zbytku)
<3> určení zemské polokoule u zeměpisné šířky N/S (severní/jižní)
<4> zeměpisná délka, formát ddmm.mmmm (stupně|minuty.desetinné vyjádření zbytku)
<5> určení zemské polokoule u zeměpisné délky W/E (západní/východní)
<6> kvalita GPS signálu 0 = bez signálu, 1 = bez diferenčních korekcí (DGPS), 2 = s diferenčními korekcemi
<7> družice použité pro výpočet polohy, čísla 00 až 12
<8> HDOP (Horizontal Dilution of Precision) - horizontální složka ukazatele geometrie rozložení družic, které budou použity pro výpočet
<9> výška antény vztažená nad nebo pod hladinu moře, -9999.9 až 99999.9 metrů
<10> výška antény vztažená ke geoidu (fyzikální def. tvaru Země), -999.9 až 9999.9 metru
<11> stáří diferenčních korekcí DGPS (RTCM - SC104), počet vteřin od posledního platného přenosu RTCM (0 = nemá DGPS)
<12> identifikační číslo diferenční referenční stanice, 0000 až 1023 (0 = nemá DGPS)

2.3.4.3. \$GPGGL

Pozice(GGL).

\$GPGGL <1>,<2>,<3>,<4>,<5>,<6>*hh<CR><LF>

<1> zeměpisná šířka, formát ddmm.mmmm (stupně|minuty.desetinné vyjádření zbytku)
<2> určení zemské polokoule u zeměpisné šířky N/S (severní/jižní)
<3> zeměpisná délka, formát ddmm.mmmm (stupně|minuty.desetinné vyjádření zbytku)
<4> určení zemské polokoule u zeměpisné délky W/E (západní/východní)
<5> UTC čas, formát hhmmss (hodiny|minuty|vteřiny)
<6> stav zobrazovaných informací, A = platná pozice, V = pozice není platná

2.3.4.4. \$GPGSV

Viditelné družice na obloze (GSV)

\$GPGSV <1>,<2>,<3>,<4>,<5>,<6>,<7>.....<4>,<5>,<6>,<7> *hh<CR><LF>

<1> celkový počet GSV přenášených vět
<2> pořadové číslo právě přenášené GSV věty
<3> celkový počet viditelných satelitů, rozsah 00-08
<4> označení PRN (pseudonáhodný kód - signál, který vysílá každá družice a který může být použit pro výpočet polohy. PRN je pro každou družici unikátní), rozsah 01-32
<5> elevační úhel, udávající výšku satelitu (ve stupních) nad obzorem, rozsah 00-90 stupňů
<6> azimut (úhel měřený od severu po směru hodinových ručiček) polohy satelitu. rozsah 0-359 stupňů
<7> poměr šumu v signálu (C/No - signal to noise ratio), rozsah od 00 do 99 dB, prázdná položka znamená, že není přenášen signál

2.3.4.5. \$GPGSA

Informace o DOP a aktivních družic (GSA)

DOP (Dilution of Precision) - ukazatel geometrie rozložení družic, které budou použity pro výpočet, družice které jsou více "rozprostřeny po nad viditelným obzorem" dávají lepší DOP. Aktivní družice jsou všechny družice, které přístroj uzná, že je může použít pro výpočet polohy (kritériem pro uznání takové družice je mnoho, lze uvést např. intenzita přijímaného signálu, okamžitá výška satelitu nad obzorem v době příjmu signálu, atd.).

\$GPGSA,<1>,<2>,<3>,<3>,<3>,<3>,<3>,<3>,<3>,<3>,,,,,<4>,<5>,<6> *hh<CR><LF>

<1> mód, M = manuální, A = automatický

<2> mód (typ) navigace, 0 = navigace není možná, 1 = 2D navigace, 2 = 3D navigace

<3> identifikátory družic (podle PRN kódu, rozsah 01-32), které označují družice použité pro výpočet polohy, max. 8 najednou v jedné větě

<4> DOP, rozsah 1.0 - 99.9

<5> HDOP - horizontální složka DOP, rozsah 1.0 - 99.9

<6> VDOP - vertikální složka DOP, rozsah 1.0 - 99.9

2.3.4.6. \$GPVTG

Rychlost nad zemí(VTG)

\$GPVTG,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,*hh<CR><LF>

<1> skutečný kurz 0-399,9°

<2> orientace (T= true north)

<3> magnetický kurz 0-399,9°

<4> orientace (M= magnetic north)

<5> rychlost 0- 999.9

<6> jednotka (N= knots)

<7> rychlost 0- 999.9

<8> jednotka (K= Km/h)

2.4. Přesnost systému

2.4.1. Obecně

Měření k více družicím zpřesňuje údaj o poloze. Složitý anténní systém zajišťuje komunikaci družice s řídicími středisky na Zemi, vysílání měřicích signálů a u poslední generace družic i jejich vzájemnou komunikaci (vedoucí třeba k rychlejšímu odhalení chybně pracujících družic). Vysílány jsou dva vojenské a jeden civilní signál. Šifrované vojenské signály nabízejí přesnější určení polohy, ale mohou je použít pouze autorizovaní uživatelé (jako jsou armády států NATO). Signály jsou složeny z nosné frekvence (na ni je naladěn přijímač), speciálního kódu (známy jsou kódy pro civilní signál) a vlastní navigační zprávy, která nese informaci o poloze družice, čase, technickém stavu družice a dalších parametrech. Speciální kód dovoluje vysílat všem družicím na stejném kmitočtu bez vzájemného rušení a tento signál je odolnější i vůči záměrnému rušení.

Nedávným vypnutím záměrného znepřesňování polohy se velmi zvýšila přesnost civilních přístrojů GPS. Do 1.5.2000 byla přesnost určení horizontální polohy přibližně 100 m.

Nyní určíte polohu i pomocí nejlacinějšího přístroje s přesností přibližně 15 m. Což výrazně rozšiřuje možnosti využití systému GPS i pro účely pro které byl donedávna nepoužitelný.

Pokud potřebujete ještě přesnější určení polohy máte možnost připojit k GPS přístroj DGPS, který přijímá diferenciální korekce a s jejich pomocí upřesňuje polohu. Pro ještě přesnější měření se používá speciálních geodetických GPS.

Signál na Zemi je však již poměrně slabý, což způsobuje potíže při určování polohy v tunelech nebo budovách. Zlepšení v tomto směru má přinést nová generace družic, které budou vypouštěny v následujících letech.

2.4.2. Prakticky

Při běžném používání GPS bez použití DGPS - při turistice, jízdě v autě atd. se chyba ve většině případů vyjde menší než 10 m.

Odchylka je závislá na poloze satelitů v okamžiku měření, a v neposlední řadě na místě měření - čím je horší výhled na oblohu, tím je GPS méně přesný.

2.4.3. Horizontální poloha

Při testování na místě s dokonalým příjmem byla chyba v 90 % menší než 10 m. Na druhou stranu v místě s extrémně špatným příjmem bývá chyba až 90 m.

2.4.4. Nadmořská výška

Při určování nadmořské výšky jsou chyby zhruba o 50 % větší než při určování horizontální polohy.

Proto se GPS dá využít spíše k orientačnímu měření nadmořské výšky. Pro přesnější měření nadmořské výšky je vhodnější klasický výškoměr, nebo GPS s integrovaným výškoměrem.

Přesnost měření se dá zvýšit zaznamenáváním polohy po delší dobu a následným zprůměrováním výsledků.

2.4.5. Shrnutí

Na základě praktických zkušeností lze říci, že ruční GPS se dá používat výhradně tam, kde je volný, nezakrytý výhled alespoň na část oblohy. Přičemž nejcitlivější na dostatek signálu je zejména zapnutí přijímače, kdy se přístroj inicializuje a zjišťuje svou polohu. Proto, chystáte-li se používat GPS v místě se zhoršeným příjmem, je vhodné ho zapnout ještě dříve, než se do místa s nedostatkem signálu dostanete (například při použití GPS v hustém lese provést zapnutí ještě před vstupem do lesa nebo na mytíně). Když se GPS již inicializuje, tak pro další provoz mu stačí podstatně menší signál. Nutno však počítat s tím, že čím je slabší příjem, tím je i horší přesnost určené polohy. GPS přijímač také není vhodné stínit tělem, zejména nacházíte-li se v oblasti s horším příjmem, proto, nenesete-li ho v ruce před sebou (ideální poloha), dejte ho například do horní kapsy batohu anténou vzhůru, nebo, máte-li ho v pouzdře s klipsnou přichyťte ho na ramenní popruh batohu. Další relativně vhodné místo je v náprsní kapse. Přichycení GPS k pásku na boku není z důvodu velkého stínění příliš ideální. Ani hustý les nebývá pro dvanácti kanálové přijímače větším problémem, pokud se ovšem neskloňuje dohromady hustý les s další překážkou, například zastíněním skálou. O poznání horší je situace ve městě, speciálně na chodníku, hned u stěny budovy, kde výhled na satelity je značně stíněn. Opět je vhodné přístroj nechat inicializovat na otevřeném místě (např. náměstí). Pak už by GPS mělo pracovat, nutno se ovšem smířit s tím, že speciálně v úzkých uličkách se bude chvílemi ztrácet, ovšem na nejbližší křižovatce, kde získá výhled na satelity z jiných směrů se opět chytne. Co se týká budov, tak uvnitř nepočítejte, že by se dalo GPS využívat. Výjimkou mohou být chaty s tenkou dřevěnou či bakelitovou střechou. V "normální" budově vám GPS může fungovat leda přímo u okna, nebo na balkóně, a to pouze za předpokladu, že z okna máte nestíněný výhled na oblohu a ve směru výhledu je v okamžik používání GPS dostatek satelitů. Celkem bezproblémově se GPS dá používat

v autě a to i bez externí antény, přičemž je však nutné mít GPS umístěný poblíž okénka. Ideální místo je na desce pod předním sklem. Kupodivu GPS při průjezdu městem funguje lépe, než při pěší procházce - zřejmě jednak na silnici je lepší příjem než na chodníku ve stínu budovy a navíc přes "kritické" místa se autem dostanete rychleji než pěšky, a tak GPS ani nestačí ztratit signál. Při jízdě autem tedy normálně ztratíte signál pouze při průjezdech tunely, v podzemních garážích, popřípadě při delším stání pod nízkým mostem. V dopravním letadle funguje GPS bez problémů, pokud sedíte přímo u okénka. Na místě do uličky s GPS nepochodíte. Obdobně se chová GPS ve vlaku - pokud ho přitisknete přímo k okénku, tak funguje, akorát nesmějí být přímo u trati skály nebo budovy, které by stínily výhled na oblohu. Problémem občas bývá inicializace přístroje. V autobusu je situace o něco lepší než ve vlaku - přeci jenom okna bývají větší a GPS má lepší výhled na oblohu, takže problémy s nedostatkem signálu nebývají. Opět je ovšem nutné mít GPS co nejbližší u okénka. V jeskyních a jiných uzavřených prostorách GPS nefunguje vůbec. Co se týká povětrnostních vlivů, tak jsem používal GPS v hustém dešti, při hustém sněžení i mlze, aniž bych pozoroval jakoukoli změnu citlivosti nebo přesnosti. Většina dnešních GPS má tzv. satelitní obrazovku - stránku s grafickým vyobrazením satelitů, kde můžete sledovat, kde se v danou dobu jaký satelit nachází a podle níž se brzy naučíte předvídat, místa, kde hrozí ztráta signálu a naopak místa s perfektním příjmem. Na závěr ještě jedna rada - nedaří-li se vám inicializovat GPS na místě se zhoršeným příjmem, zkuste ho vypnout a znovu zapnout o pár kroků vedle s anténou směrem k obloze. Mnohdy i pár kroků pomůže...

2.5. GPS přijímače

Přístroje GPS obsahují velmi citlivý přijímač signálů v pásmu 1.5 GHz, přesný zdroj časového signálu (hodiny) a vysoce výkonný matematický procesor. Vše se nachází v ultrazvukově hermeticky zapouzďřeném obalu, který je naplněn dusíkem. Veškerá zobrazení a komunikace s uživatelem probíhá pomocí přehledné LCD obrazovky, obrazového menu systému, akustických signálů a plochých tlačítek na panelu přístroje. Rozměry všech kapesních navigačních přístrojů se pohybují okolo 10x5x3 cm a jejich hmotnost je přibližně kolem 200 g. Napájení je řešeno pomocí běžných tužkových článků (alkalických, NiCd, NiMH) nebo vnějšího napájecího zdroje 10-40 V (např. v automobilu). Spotřeba GPS přístrojů se však pohybuje nad hranicí 100 mA, což klade na výběr napájecích článků obzvláště přísné požadavky. Některé přístroje navíc odebírají z článků velmi malý proud i ve vypnutém stavu, což souvisí s nutností napájení paměti s uživatelskými údaji.

Jednotlivé přístroje se navzájem poněkud liší, především maximální velikostí napájecího napětí (8-40 V), typem miniaturní antény přijímače (vnitřní, vnější), počtem přijímacích kanálů (8,12), maximální měřitelnou rychlostí (166 Km/h, 1850 Km/h), maximálním počtem uložitelných trasových bodů (250, 500) a výdrží baterií (12-24 hodin).

Většina přístrojů je opatřena konektorem k propojení dvou přístrojů navzájem nebo přístroje a osobního počítače (PC). Dodaný software pak umožní nejen načítat údaje z přístroje, zobrazovat je a uchovávat v počítači, ale přímo na počítači zadávat jednotlivé trasové (otočné) body, sestavovat z nich trasy různých cest a určovat vzdálenosti k zajímavým cílům. Po přenesení těchto dat z počítače do přístroje již nezbyvá, než odvážně, bez mapy a kompasu, vyrazit na cesty nebo do vzduchu vstříc novým dobrodružstvím.

2.6. Rozdělení

GPS přijímače se dají rozdělit do několika skupin

2.6.1. Ruční

- nejširší oblast přijímačů vhodných pro turistiku, cestování, navigaci do auta, jednoduché přístroje na loď, paragliding, UL, motorku, kolo...
- cenově výhodnější než specializované modely
- široká škála přístrojů, mezi kterými se dá vybírat



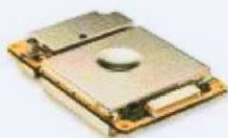
2.6.2. Námořní

- neboli tzv. "mapové plottery", přístroje většinou s větší obrazovkou vhodnou pro zobrazení vyššího detailu, často pouze s externím napájením
- sonary nebo kombinované GPS se sonary



2.6.3. Aplikační

- GPS přijímače pro aplikace jako např. řízení přesného času počítačových sítí, sledování pohybu objektů...



2.6.4. Letecké

- GPS přijímače pro použití v letadlech. Vestavěné nebo ruční.
- Hlavní odlišností je, že obsahují vestavěné letecké mapové prvky (Jeppesen databáze, VOR, NBD..)



2.6.5. Pro kapesní počítače PDA

Výhody GPS pro PDA:

- velmi levné
- velké množství mapových podkladů
- jednoduchá obsluha pomocí dotykového displeje
- možnost širšího využití PDA (ne pouze pro navigaci)



3. Vývojové prostředí

3.1. Úvod

Jako vývojové prostředí jsem použil Microsoft Visual Studio 2005⁶. Součástí tohoto studia je Framework⁷ ve verzi 2.0. Jako programovací jazyk jsem zvolil Visual Basic.NET⁸.

Cílem této části není popisovat samotný jazyk Visual Basic.NET ani vývojové studio Microsoft Visual Studio, ale upozornit na změny oproti starší verzi studia a Frameworku (1.x).

Nová verze .NET Frameworku přináší početně novinek. Zde uvádím některé z nich a některými se budu dále zabývat.

- Team Foundation Server
- Analýza kódu
- Refaktorování a znázorňování
- Code Snippets⁹
- Komponenty
- ASP.NET 2.0
- Podpora pro 64bitové platformy
- Podpora pro SSL komunikaci
- Šifrování citlivých dat aplikace
- Aplikace změn kódu při ladění bez restartu aplikace
- Vylepšena podpora pro síťovou komunikaci (FTP, ping, SMTP ...)
- Generics (šablony)
- Kompresní formát GZIP

⁶ Microsoft Visual Studio 2005 je komplexní vývojové prostředí, které v sobě integruje úplné spektrum nezbytné pro efektivní tvorbu softwaru – od nástrojů pro aplikační modelování až po zátěžové testy.

⁷ **Microsoft .NET Framework** je nová technologie od Microsoftu pro vývoj aplikací, jak na desktop, tak i pro web. Jeho hlavním motorem je CLR (Common Language Runtime). Na tomto motoru jsou provedeny jednotlivé prvky a jejich komunikace. Takzvaně spravuje běžící kód. Celý kód je kompilován do MSIL (Microsoft Intermediate Language – obdoba strojového kódu). Platforma dokáže také nabídnout širokou škálu programovacích jazyků. Sami si můžete vybrat mezi C# Sharp, C++.Net, VB.Net a J# JAVA. Další jazyky jsou v přípravě od třetích stran. Už nyní je vidět snaha zakomponovat další jazyk a to Delphi.NET od Borlandu.

⁸ **Visual Basic.NET** představuje novou generaci programovacího prostředí Visual Basicu, které bylo od základu předěláno pro moderní vývojářskou platformu .NET. Umožňuje rychle vyvíjet plnohodnotné aplikace Windows, služby Web i aplikace Web a budovat vícevrstvé distribuované systémy na úrovni korporací.

⁹ Předpřipravený kód

3.2. Team Foundation Server

Asi tou největší novinkou pro vývojáře z předchozích verzí Visual Studia je Team Foundation Server. Je to právě ta součást nového Studia, která má vyplnit mezeru způsobenou absencí vestavěných nástrojů pro koordinaci vývoje v týmu.

Team Foundation Server obsahuje nový systém správy zdrojových kódů, která nahrazuje již značně zastaralý Visual SourceSafe (nic nám však nebrání ho nadále používat). Kromě lepší integrace do IDE umožňuje provádět "check-in"¹⁰ změněných souborů jako jeden celek a stejným způsobem provést i případné vrácení změn (rollback¹¹). Data jsou ukládána do databáze MS SQL Server 2005, který vychází společně s Visual Studií a jeho základní verze MS SQL Server 2005 Express je součástí instalace Visual Studia.

Dále zde můžeme najít podporu pro plánování, zadávání úkolů, jejich přidělování, sledování, reportování a vůbec kontrolu nad chodem vývoje. Team Foundation Server tedy může nahradit doposud používané nástroje s výše uvedenou funkcionalitou, je ale otázkou, nakolik se vyplatí tyto fungující, mnohdy za draho koupené produkty rušit a pracně převádět do nového systému data za léta vývoje. Přinejmenším nová správa zdrojových kódů za to však stojí. Své místo si zde našlo i centralizované vytváření buildů¹².

3.3. Analýza kódu

Nikdo není dokonalý, a proto je vhodné nechat si v kódu vyhledat chyby a nedostatky, které by mohly způsobit nejednu nepříjemnost. Ačkoli sám kompilátor dokáže odhalit nepovolené praktiky jako je např. použití neinicializovaných proměnných, odhalení řady nebezpečných konstrukcí mu uniká. Od toho je tu ale modul Code Analysis, který dokáže těchto problémů odhalit bezmála dvě stě. Jeho použití nejenže snižuje riziko vzniku chyby, ale zlepšuje i čitelnost kódu, což se vyplatí dvojnásob, při vývoji v týmu.

3.4. Refaktorování a znázorňování

Visual Studio 2005 poskytuje několik základních refaktorovacích úkonů, čítaje přejmenování, vyjmutí metody, zapouzdření proměnné, vyjmutí rozhraní, přesunutí položky do parametru, odebrání parametru a přeuspořádání parametrů. Podstatné na použití těchto refaktorovacích funkcí je automatická úprava celého kódu na základě provedeného refaktorování. Zapomeňte na "Najdi a nahrad"!

Abychom neztratili přehled nad naším rozrůstajícím se dílem, můžeme si pomocí Class Diagramu přehledně znázornit moduly aplikace a vyvodit z toho případné změny v návrhu. A když už nic jiného, bude vhodnou součástí projektové dokumentace.

¹⁰ Zamknutí používaných zdrojů. Tím se předejde kolizím zdroje, který chce měnit více programátorů na jednou. Jeden programátor zdroj zamkne a ostatní musí počkat, až zdroj zase odemkne.

¹¹ Vrácení zdroje do původní podoby (beze změn).

¹² Přeložení zdrojových kódů.

3.5. Code Snippets

Modul Code Snippets obsahuje řadu předpřipraveného kódu, který zrychlí rutinní vytváření opakovaných úseků kódu. Převážně se jedná o blokem ohraničené příkazy typu foreach, try, #region apod. Můžeme si ale vytvořit libovolné vlastní. Snippet lze vložit na jedno místo, nebo jím lze obklopit (pokud to snippet podporuje) označenou část kódu.

Vytváření zdrojů (resources) ve Visual Studiu 2002 a 2003 bylo poněkud nezvyklé, a proto není divu, že Microsoft vyslyšel přání vývojářů a vrátil do Studia práci se zdroji způsobem, na jaký byli zvyklí z Visual Studia 6.0. To znamená, že můžeme pohodlně v IDE přidávat do projektu různé typy zdrojů včetně zvuků a obrázků v různých formátech.

3.6. Komponenty

Značného vylepšení se dostalo také mnoha prvkům pro práci s daty. Úplně novým nástrojem z této oblasti je tzv. designér datasetů. Jedná se v podstatě o grafickou reprezentaci relací mezi datasetem a zdrojovými daty. Umožňuje jednoduše provádět změny těchto mapování, aniž bychom byli nuceni editovat přímo XML schéma, tak jako v předchozích verzích. Dále je v něm možné navrhnout typový datový adaptér a nechat k němu automaticky vygenerovat metodu Fill k naplnění datasetu. Metod Fill, popřípadě Get, je možné vytvořit i více – např. pro získávání dat na základě nějakého filtru. Tyto metody je také možné namapovat přímo na SQL příkazy nebo uložené procedury.

Dataset je možné rozšířit pomocí částečných tříd. Jejich smyslem je oddělit psaný kód, aby nedošlo k jeho znehodnocení při regenerování automatického kódu z designéru – automaticky generovaný kód je spojen s částečnou třídou až v době kompilace. Tuto novou vlastnost je možné využít např. při vytváření vlastní validační logiky datasetu.

Dalším rozšířením datasetu je typový dataset. Při vytváření datasetu se zároveň vygeneruje třída, která reprezentuje datovou tabulku napojenou k datasetu. Programátor se tak v kódu může odkazovat přímo na metody této třídy generované z názvů sloupců databázové tabulky. Tato třída je zároveň zanalyzována vývojovým prostředím Visual Studia a dané metody se automaticky začlení do IntelliSense menu.

Jednou z nových komponent pro práci s daty je BindingSource. Tato komponenta slouží k zapouzdření napojení na datový zdroj, kterým může být objekt, webová služba nebo databáze. Ačkoliv se v podstatě nejedná o nic výjimečného, ušetří programátorovi psaní spousty rutinního kódu vytvářejícího přemostovací datasety pro zobrazení dat ve formulářových prvcích.

Závěrem bych zmínil komponentu DataGridView určenou pro zobrazování a editaci dat. Jako datový zdroj lze použít SQL Server, případně nějakou jinou databázi, webovou službu nebo uživatelsky definovaný objekt. Pokud datový zdroj obsahuje velké množství záznamů, lze ji použít v tzv. virtuálním režimu, kdy komponenta načte vždy jen ta data, která jsou skutečně zobrazena. Kromě sloupců z datového zdroje v ní lze vytvářet i sloupce vlastní. Události je možné nastavit pro jednotlivé buňky a ne pouze celé řádky. Komponenta také plně podporuje grafická schémata Windows.

Za zmínku jistě stojí menší revoluce v datových komponentách, kde při návrhu formuláře budete marně hledat svoje staré komponenty pro přístup k jednotlivým providerům – rozhraní pro přístup k datům bylo zjednodušeno a vazba na konkrétní datový zdroj je nyní zanořena v nových zobecněných komponentách DataGridView, BindingSource, BindingNavigator a ReportViewer. Z nových komponent by mohla zaujmout kupříkladu komponenta BackgroundWorker usnadňující paralelní zpracování některých úloh nebo třeba komponenta SerialPort, která – jak je z názvu zřejmé – zajišťuje komunikaci se sériovým portem.

3.7. Zabezpečení a zveřejnění aplikace

Visual Studio 2005 umožňuje z hlediska bezpečnosti nadefinovat oprávnění pro přístup aplikace k prostředkům systému, jako např. přístup k souborům, registrům, internetu, nebo tisku. Tato oprávnění se definují pro vybranou zónu (zpravidla místní síť, nebo internet). Následně si můžeme nechat pravidla analyzovat, aby byla vzájemně kompatibilní. Samozřejmostí je i digitální podepsání aplikace a tím ze strany klienta možnost ověřit si její původ.

Pokud jde vše dobře, nadejde jednoho dne čas ke zveřejnění naší práce, ať už z důvodů testovacích, nebo kvůli doručení aplikace zákazníkovi. Visual Studio 2005 to hravě zvládne a nezáleží na tom, jestli je cílová složka na místním disku, webové stránce, nebo ftp. Automaticky je zajištěno i verzování. Lahůdkou by se dala nazvat správa aktualizací. Pro aplikaci si lze zvolit čas instalace aktualizace, a hlavně – aplikace se sama může dotazovat na dostupné aktualizace při startu, nebo pravidelně podle zvoleného intervalu.

3.8. Technologie ClickOnce

Dále bych se věnoval technologii ClickOnce, která je určená k nasazení a aktualizaci aplikací. Jejím cílem je odstranit typické problémy nasazení, např. velmi časté konflikty DLL knihoven, a také celý proces nasazení a případných aktualizací co nejvíce zautomatizovat, aby vyžadoval ze strany uživatele minimální interakci. Použití ClickOnce je velmi snadné a pokud si vývojář vystačí se základními schopnostmi, lze tuto technologii zintegrovat do aplikace prakticky bez napsání jediné řádky kódu. Její nastavení se provádí na záložkách publish a security v nástroji project designer.

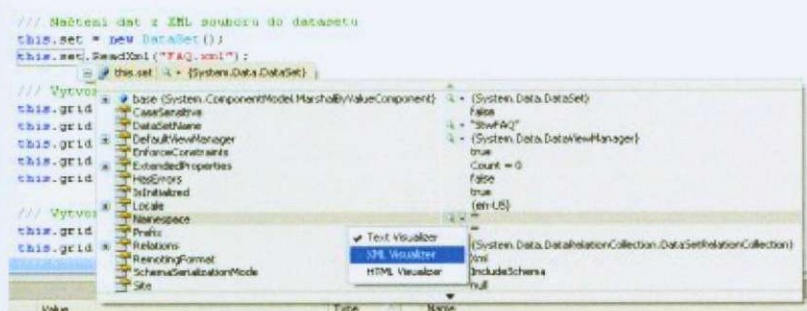
Je nutné zde nastavit, zda se má aplikace spouštět přímo z webu a nebo zda je možné ji nainstalovat i pro off-line použití. V případě instalace je možné nastavit způsob aktualizací, k dispozici jsou volby pro zkontrolování aktualizací před spuštěním samotné aplikační logiky – v takovém případě dojde k eventuální aktualizaci okamžitě, nebo až za běhu aplikace s tím, že se případný update provede před příštím spuštěním. Také je možné využít ClickOnce API a přidat položku pro manuální zkontrolování aktualizací např. do menu. Další přínosnou možností je nastavení minimálního čísla verze. Pokud pak aplikace zjistí, že je na webu aktualizace s vyšší zadanou minimální verzí, nebude již možné starou verzi použít, aniž by uživatel povolil její update. Vývojář tak vlastně může donutit uživatele k instalaci různých kritických oprav.

Druhá skupina nastavení se týká bezpečnosti. Aplikace postavená na .NET Frameworku standardně běží v tzv. sandboxu, což je obdoba kontejneru ve kterém se spouštějí applety v rámci webového prohlížeče. Obvykle ale aplikace vyžaduje další práva, např. pro přístup k registrům nebo souborovému systému. V nastavení je možné zadat minimální práva, která aplikace potřebuje pro svůj běh. Při spuštění je pak uživatel dotázán, zda chce aplikaci tyto práva přidělit a nebo jí ukončit. Tím se zamezí jejímu selhání z důvodu odmítnutí přístupu k nutným systémovým zdrojům. Minimální práva je možné ve Visual Studiu nastavit buď ručně a nebo je nechat vygenerovat automaticky pomocí tzv. kalkulátoru oprávnění. Dále je možné k aplikaci přidat různé bezpečnostní certifikáty, které umožňují ověřit její pravost a zajistit, že se budou stahovat aktualizace pouze od původního vydavatele.

Po provedení těchto nastavení, a případně vlastní implementaci ClickOnce API v programovém kódu, se pomocí nástroje publish wizard aplikace se všemi nutnými součástmi nakopíruje na web. Odtud si ji uživatel může spustit nebo nainstalovat. Čímž se dostáváme k poslednímu podstatnému prvku technologie ClickOnce. Uživatel nestahuje z webu přímo danou aplikaci, ale tzv. bootstrapper. Jedná se o program, jehož účelem je zajistit aplikaci veškerý potřebný software. Po stažení a spuštění bootstrapper nejprve zkontroluje, zda je na počítači nainstalován .NET Framework 2.0. Pokud ne, automaticky se postará o jeho stažení a instalaci. Stejným způsobem jsou obstarány všechny další softwarové součásti, které aplikace potřebuje, např. Microsoft DirectX nebo MDAC. Pokud je to potřeba, postará se bootstrapper během tohoto procesu i o restarty systému. Teprve když je připraveno softwarové prostředí, přejde bootstrapper k samotnému spuštění nebo instalaci původní aplikace.

3.9. Drobnosti které potěší

U nápovědy došlo ke zlepšení její dynamické části (generuje se během psaní kódu) – můžeme si zvolit oblasti, ve kterých se bude vyhledávat i jejich pořadí. Dále došlo k rozšíření zdrojů nápovědy. Celkem jsou čtyři: Local Help, MSDN Online, Codezone Community a Questions. Zvláště ty "neoficiální" zdroje mohou být užitečné při hledání odpovědí v krátkém čase na záludnosti, se kterými se často potýká větší množství uživatelů současně (typicky u nového software). Podobně jako u dynamické nápovědy si můžeme vybrat, které ze zdrojů se mohou použít a v jakém pořadí.



Data si lze během ladění přehledně zobrazit

Seznam drobných vylepšení, která při rozhodování o koupi Visual Studio 2005 nebudou rozhodující, ale především pro vývojáře v předešlých verzích Visual Studio budou příjemnou novinkou:

- Možnost rozdělit definici třídy do více souborů
- Inicializační kód formuláře implicitně skryt, oddělen od zbytku kódu
- Snazší přepínání mezi kódem a návrhem formuláře pomocí myši
- Zlepšeno zarovnávání prvků při návrhu formuláře
- Vylepšený IntelliSense
- Zobrazení datových zdrojů během krokování aplikace

3.10. Požadavky na provoz

Požadavky na instalaci a provoz Visual Studio se liší podle používané edice. Většinou jsou jako minimální požadavky uváděny: procesor 2 GHz, operační systém Microsoft Windows 2000 SP4 a novější (XP, 2003 s aktuálními service packy), 384 MB operační paměti, 2 GB volného místa na disku, DVD-ROM mechanika, monitor s rozlišením 1024x768 a kompatibilní klávesnice a myš. Asi nemusíme připomínat, že pro opravdu pohodlnou práci jsou uváděné parametry operační paměti, diskového prostoru a monitoru nedostatečné.

4. Aplikace GPS

4.1. Úvod

Cílem této kapitoly je seznámit vás se zobrazovací aplikací, kterou jsem vytvořil ve Visual Studiu 2005 za pomoci programovacího jazyka Visual Basic.NET.

Pro běh aplikace je nutné mít nainstalované běhové prostředí Visual Studio 2005, což je Framework verze 2.0. Tato verze není kompatibilní s předešlými verzemi, z důvodu některých změn.

Aplikace nabízí připojení GPS přijímače a to na sériovém portu nebo na virtuálním sériovém portu například přes BlueTooth¹³ rozhraní. Komunikace s aplikací probíhá podle protokolu NMEA(viz. kapitola o protokolu NMEA) a podporuje tyto věty: \$GPRMC, \$GPGGA, \$GPGGL, \$GPGSV, \$GPGSA a \$GPVTG. Tyto věty jsou totožné téměř u všech výrobců GPS(někteří mají další své specifické) a poskytují komplexní data systému GPS. GPS zařízení, které jsem měl zapůjčené¹⁴ je podporuje všechny. V aplikaci je možné si otestovat jaké věty dané zařízení podporuje.

Jako další funkci bych rád zmínil operace s daty. Data si můžeme zobrazit real-time¹⁵, jak je posílá připojené GPS zařízení a pokud jsou pro nás užitečná, máme možnost si je uložit do textového souboru¹⁶. Další možností je rovnou zapnout nahrávání dat. Data se průběžně s chodem GPS zařízení nahrávají do zvoleného souboru. Tento soubor je obyčejný textový soubor, neboť je to nejběžnější interpretace protokolu NMEA a lze využít i soubory získané jinými programy při zachování zpětné kompatibility formátu protokolu NMEA.

Tyto data si lze pomocí **Simulátoru**¹⁷ zpětně prohlédnout či zobrazit, zkrátka jako kdyby bylo ono GPS zařízení připojeno k aplikaci. Při využití simulátoru jsou dostupné všechny funkce jako pro GPS zařízení.

NMEA protokol obsahuje mimo jiné i data o družicích(satelitech) z kterých je postaven kosmický subsystém systému GPS. Toho se dá využít na zobrazení přehledu satelitů. Z dat lze i vyčíst jaké satelity jsou právě využívány.

Jako poslední je zobrazení polohy na mapě. Zde je třeba říci, že mapa musí být v jednom z formátů obrázků¹⁸ a souřadnicovém systému **WGS-84**. Z toho plyne jedna velká výhoda. Potřebnou mapu si lze jednoduše naskenovat nebo stáhnout z internetu. Zde ovšem pozor na autorská práva. Pro zvláště šikovné a aktivní jedince je i možnost, že si mapu udělají úplně sami, čímž otázka s autorskými právy odpadá. Nicméně zdrojů, kde se dají mapy sehnat zadarmo je nepřeberně.

¹³ Jedná se o interní nebo externí zařízení(většinou se zasouvá do USB portu) a slouží pro připojení a přenos dat mezi počítačem a dalším zařízením, které podporuje technologii Bluetooth

¹⁴ GPS zařízení Fortuna Clip-On připojitelné přes BlueTooth rozhraní

¹⁵ real-time data, nebo-li data v reálném čase bez zpoždění

¹⁶ pokud máme zapnuto „zobrazování dat“, uložení provedeme klávesovými zkratkami CTRL+C(kopírovat) a CTRL+V(vložit) nebo pravým tlačítkem myši v kontextovém menu

¹⁷ Simulátor je součástí aplikace, která simuluje GPS zařízení z dat ve formátu protokolu NMEA

¹⁸ Podporované formáty jsou BMP, GIF, JPEG, WMF. Dále ICO, ale ten není pro tento případ reálně využitelný. (zdroj: nápověda Visual Studio)

Nyní když víme, co aplikace umí, se vrátíme zpátky k vlastnímu popisu.

Popis se bude skládat ze dvou částí:

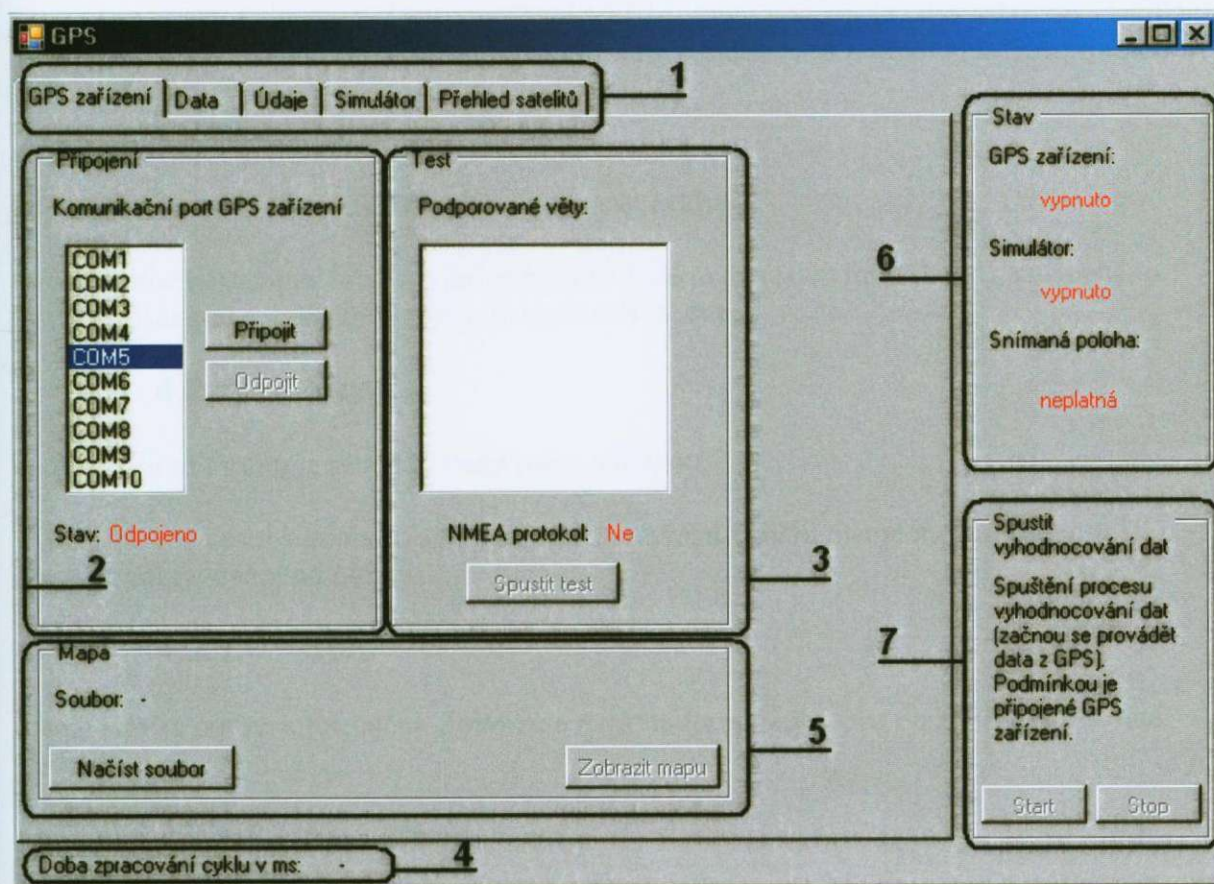
- Ovládání a vizuální popis
- Popis struktury aplikace, principu a vybraných algoritmů

4.2. Uživatelské rozhraní

Po spuštění aplikace se nám zobrazí hlavní formulář. Formulář je rozdělen na oddíly, které jsou označeny pořadovými čísly usnadňující popis aplikačního rozhraní.

Jednotlivé sekce budou podrobně popsány.

4.2.1.1. Záložka „GPS zařízení“



4.2.1.2. Lišta záložek

Lišta záložek slouží k navigaci mezi funkcemi aplikace a je určena pro rychlou navigaci mezi nimi.

V každé záložce a jejímu příslušnému tělu je zobrazeno ovládání a popis pro jednotlivé funkce.

4.2.1.3. Připojení

Ve Framu připojení je nastavení portu¹⁹, na kterém se je připojeno GPS. Pokud je vybrán správný port, stiskneme tlačítko připojit. Pokud vše proběhne v pořádku (zařízení je zapnuté, je vybrán správný port, popřípadě je v dosahu Bluetooth,...) proběhne připojení zařízení a bude indikováno stavem Připojeno u popisku **Stav**. Pokud je zapnutý simulátor automaticky se vypne.

4.2.1.4. Test

Frame test slouží k otestování vět protokolu NMEA, které podporuje GPS zařízení. Tlačítkem spustit test se do seznamu vypíše podporované věty a indikuje se podpora NMEA protokolu.

4.2.1.5. Doba zpracování

Informuje nás za kolik milisekund byl zpracován cyklus.

Pokud máme nastaveno zobrazování jednou za 1 sekundu (1000 milisekund), znamená to: Kolik milisekund z 1 sekundy program zpracovával data.

4.2.1.6. Mapa

Funkce tohoto Framu je přednastavení práce s mapou.

Vybereme zde cestu k souboru s mapou a po správném výběru máme možnost otevřít si Frame k zobrazování na mapě.

4.2.1.7. Stav

Frame stav je pouze informační. Zobrazuje nám status aplikace, pro rychlejší přehled nad aplikací.

4.2.1.8. Spustit

Jeden z nejdůležitějších Framu v celé aplikaci.

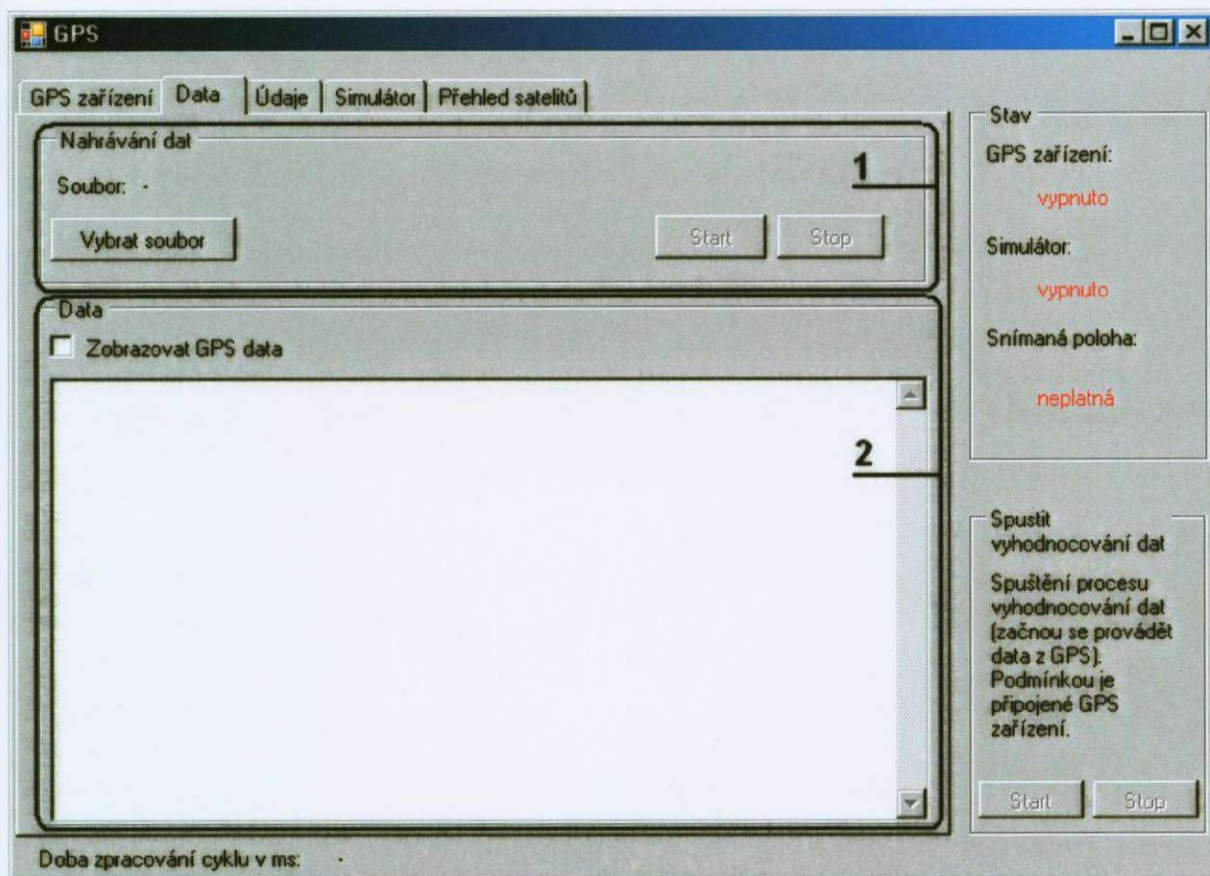
Obstarává spuštění výkonného jádra aplikace.

¹⁹ Pokud používáme připojení přes Bluetooth číslo portu nalezneme ve Windows → Ovládací panely → Zařízení Bluetooth → „Vlastní GPS zařízení“ → Vlastnosti

Začnou se číst data z připojeného GPS zařízení, převádět se do zpracovatelné podoby a to do podoby čistého textu. Poté se z těchto převedených dat vyberou souvislé NMEA věty a zobrazí do náležitých oblastí(viz. dále).

Pokud je zapnuto zobrazování na mapě, zobrazují se body polohy na mapě ve formuláři mapa (viz. dále).

4.2.2. Záložka ,Data‘



4.2.2.1. Nahrávání dat

Frame sloužící k záznamu dat z GPS zařízení.

Vybereme cílový soubor a tlačítkem start dáme aplikaci na vědomí, že se chystáme nahrávat data. Neznamená to ovšem, že se data okamžitě začnou nahrávat! Data se začnou nahrávat až při spuštění tlačítka **Start** v sekci **Spustit vyhodnocování dat**. Pokud je již spuštěno vyhodnocování dat, data se nahrávají.

Při vypnutí tlačítkem **Stop** a následném znovuzapnutí při ponechání stejného souboru se soubor nepřemaže novými daty, nýbrž se data přidají na konec souboru. Tzn., přihrajou se k původnímu souboru. Z toho plynou výhody i nevýhody. Pokud chceme mít nový záznam v čistém souboru, je třeba zvolit jiný, nový.

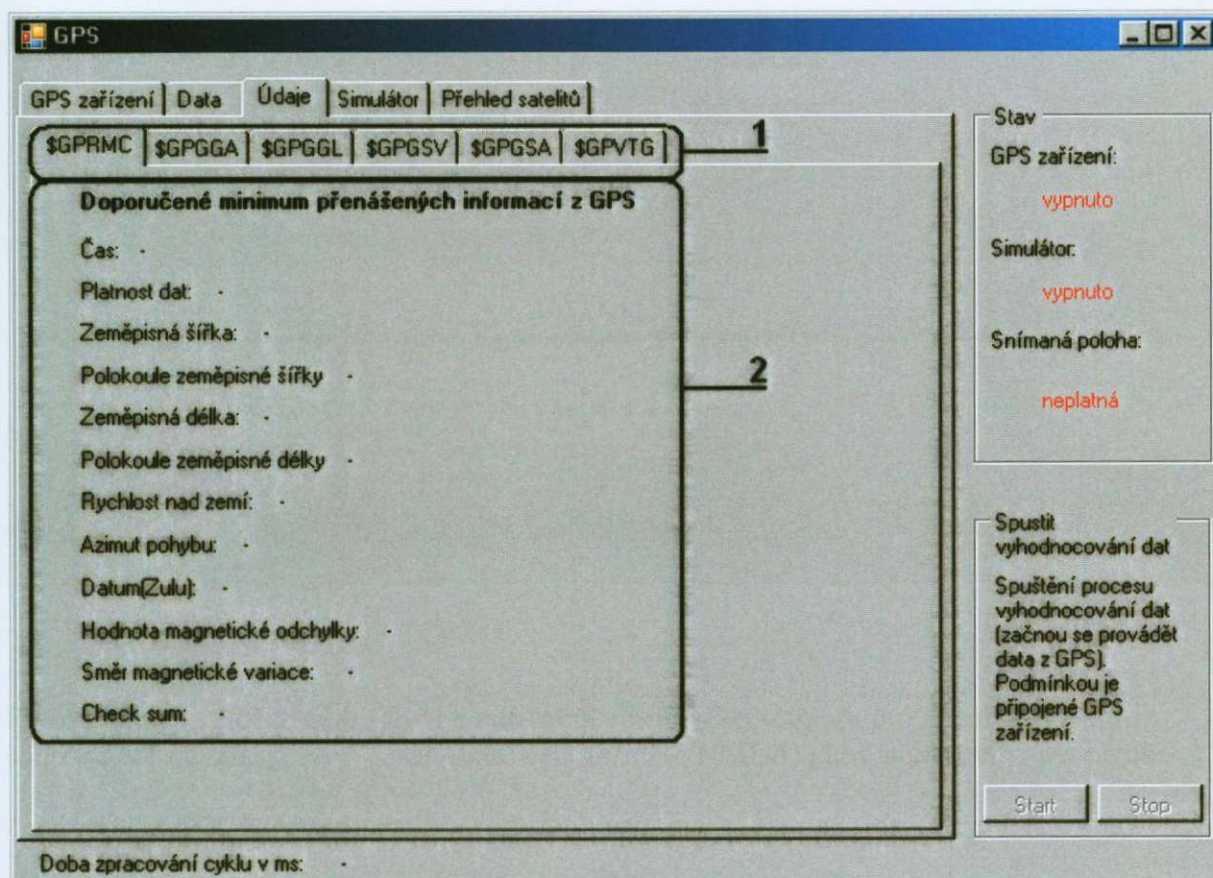
4.2.2.2. Data

Zde si můžeme po zatržení zaškrťovacího políčka zobrazit čistá data v textovém poli.

Samozřejmě data se zobrazují, pokud je aktivováno jádro aplikace podobně jako u nahrávání.

Z textového pole lze data libovolně kopírovat běžnými operacemi operačního systému.

4.2.3. Záložka 'Údaje'



4.2.3.1. Záložky s NMEA větami

Zde se nachází jednotlivé záložky k jednotlivým NMEA větám.

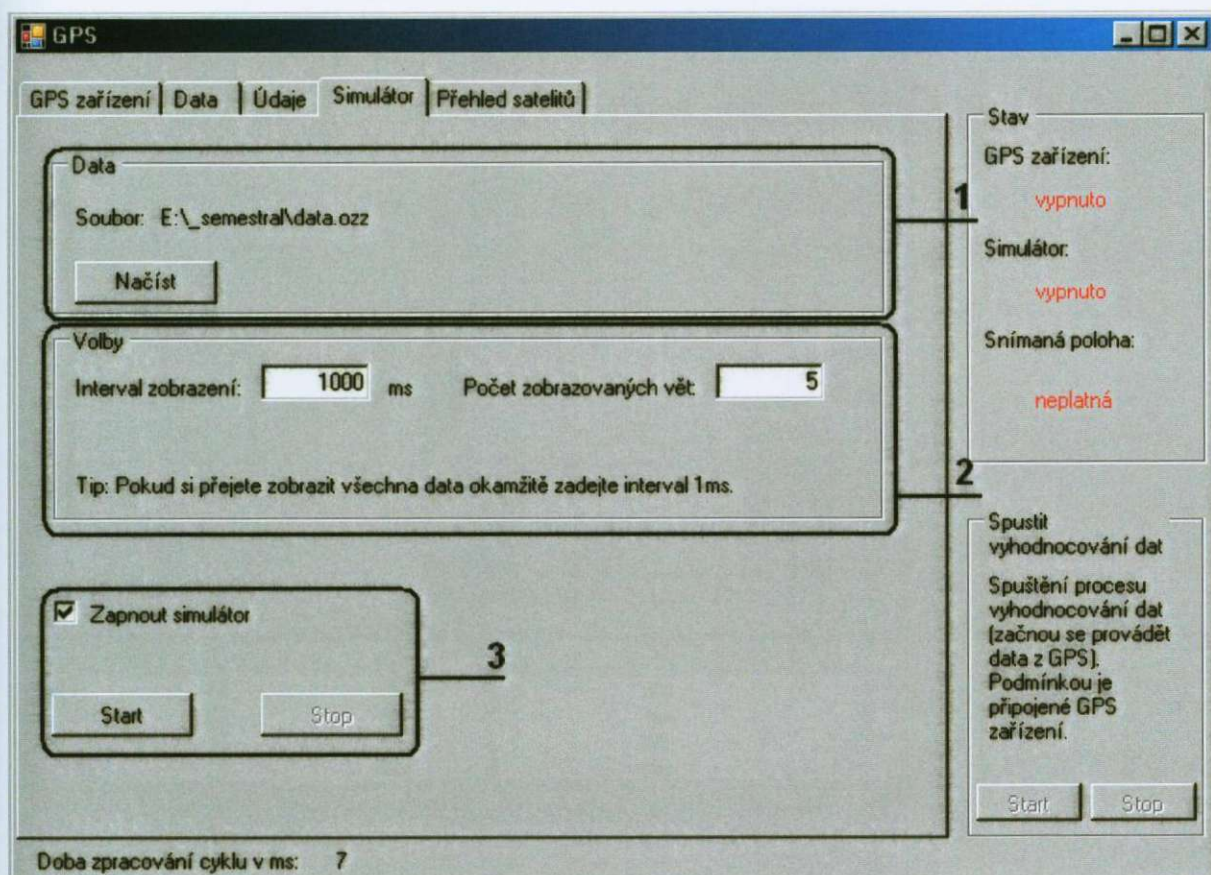
Údaje z těchto vět si lze prohlédnout, pokud tedy máme aktivované jádro aplikace.

Legenda k větám je uvedena v kapitole zabývající se NMEA protokolem.

4.2.3.2. Tělo NMEA věty

V těle máme atributy, které se dají vyextrahovat z věty.

4.2.4. Záložka ,Simulátor‘



4.2.4.1. Data

První Frame v záložce nám slouží k nastavení zdrojového souboru pro simulátor. Pokud soubor neobsahuje správný obsah (tzn. věty formátu NMEA) jádro aplikace ho jen přečte.

4.2.4.2. Volby

Ve volbách jsou dvě nastavení. Nastavení ,Interval zobrazení‘ udává po kolika milisekundách aplikace vykoná cyklus zpracovávání dat. Například si můžeme podrobně nasimulovat naši cestu a pozorně si ji prohlédnout, nastavíme vyšší interval. Pokud chceme naopak vše zpracovat a třeba jen co nejrychleji naši cestu vykreslit na mapě, nastavíme nízký interval a aplikace se postará o zbytek.

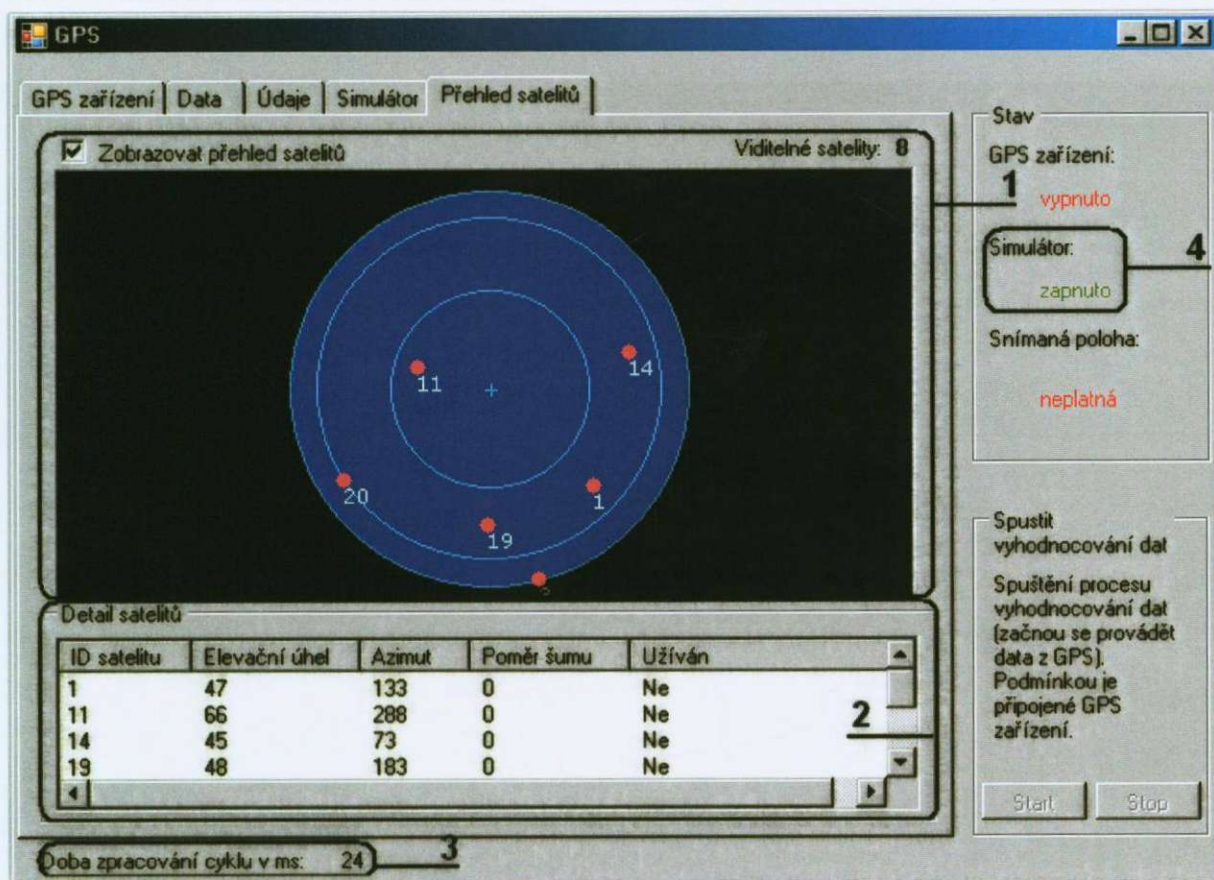
Aplikaci také můžeme nastavit počet zpracovávaných vět, standardně nastaveno na 5 (Z vlastní zkušenosti. Průměrný objem dat za jeden cyklus – omezeno přenosovou rychlostí portu.)

4.2.4.3. Spuštění simulátoru

Pokud je vše nastavené přistoupíme ke spuštění simulátoru.

Zatrhne ,Zapnout simulátor‘ a poté start. Pokud je připojeno zařízení GPS, provede se jeho odpojení.

4.2.5. Zálůžka ‚Přehled satelitů‘



4.2.5.1. Vykreslení satelitů

Po zatržení ‚Zobrazovat přehled satelitů‘ se nám začnou zpracovávat údaje o satelitech. Vidíme zde údaj o viditelných satelitech (nutně to neznamená, že jsou všechny vykresleny²⁰).

Používané satelity jsou vykreslovány zelenou barvou, nepoužívané barvou červenou.

Satelity jsou vykresleny ze znalosti jejich elevačního úhlu a azimutu vzhledem k GPS.

4.2.5.2. Detaily satelitů

V tabulce jsou uvedeny specifické údaje o satelitech a zda je satelit využit naším GPS zařízením.

4.2.5.3. Informace o zpracování cyklu

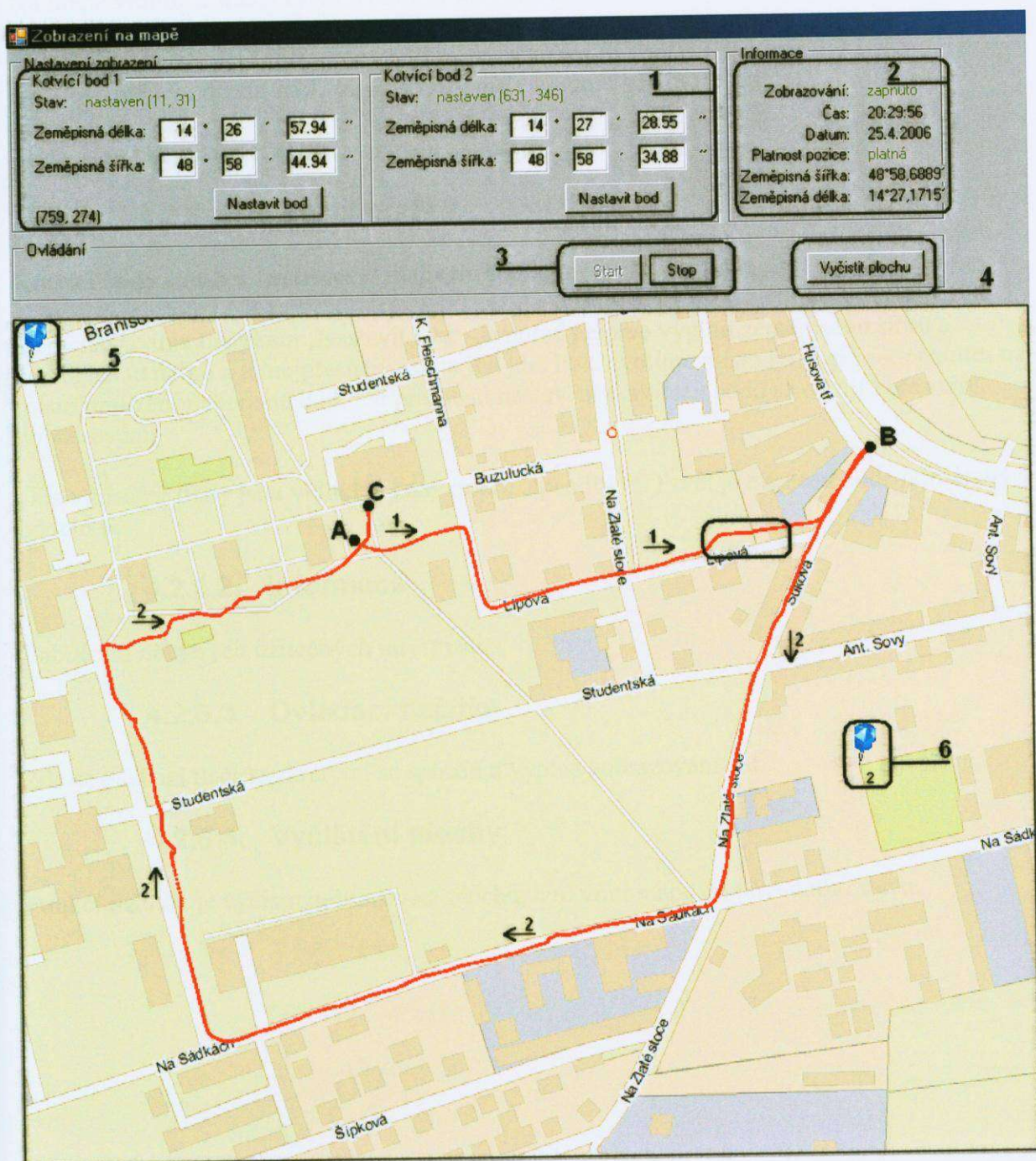
Zařízení GPS je v zapnuté což je indikováno (viz bod 4). Vidíme, že zde je doba zpracování cyklu 24ms.

Můžeme si i všimnout, že snímaná pozice je neplatná.

²⁰ NMEA věta o konkrétních údajích o satelitech obsahuje jen 4 pozice

4.2.6. Formulář ‚Mapa‘

Účel formuláře je zobrazovat data na mapě. Zobrazení formuláře najdeme v záložce GPS zařízení (viz. [kapitola 4.2.1](#)).



Jednotlivé souřadnice pozičních bodů(červená barva) se vypočítávají ze známých odchylek obou kotvících bodů.

Na mapě vidíme 2 trasy. První z bodu **A** do bodu **B** a druhá z bodu **B** do bodu **C**, obě orientované podle šípek.

Ve zvýrazněném území na 1. trase si můžeme všimnout většího zkreslení²¹.

Trasy jsou zobrazené postupně ze dvou soborů.

4.2.6.1. Kotvící body

Kotvící body slouží k „uchycení“ mapy do souřadnicového systému.

Body nastavíme tlačítkem „Nastavit bod“. Je nutné nejprve vyplnit zeměpisnou šířku a zeměpisnou délku a to nejpřesněji, jak to jen jde. Poté zvolíme místo na mapě a klepnutím na místo umístění vybereme bod. Od přesnosti nastavení kotvících bodů se odvíjí i přesnost zobrazování.

Oba umístěné body jsou viditelné jako modré „špendlíky“. První je označen bodem **5**, druhý bodem **6**.

4.2.6.2. Informace

Zobrazení některých užitečných informací.

4.2.6.3. Ovládací tlačítka

Zde se nachází tlačítka, kterými se spouští a vypíná zobrazování dat.

4.2.6.4. Vyčištění plochy

Funkcí tlačítka je vyčistit zobrazovací plochu, tzn. vrátí mapu do původního stavu.

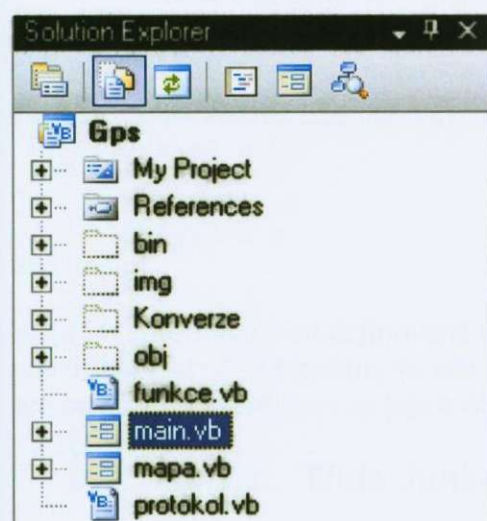
²¹ Ke zkreslení zde došlo z důvodu nedostatku signálu. Tento úsek je obklopen dvěma vysokými budovami.

4.3. Popis programové části

Tato část je určena především programátorům nebo osobám majícím přehled o programovacích technikách, konstrukcích a také syntaxí²² jazyka. V této části se budu zabývat vnitřní strukturou programu a popíši zde vybrané algoritmy.

Cílem této kapitoly není podrobný popis zdrojového kódu, ale jen vysvětlit funkcionalitu a vnitřní strukturu aplikace. Popis zdrojového kódu nalezneme v samotném zdrojovém kódu aplikace ve formě komentářů.

4.3.1. Struktura aplikace



Jak vidíme v Solution Exploreru aplikace se skládá ze dvou formulářů a dvou tříd.

Třídy **funkce** a **protokol** obsahují definice NMEA protokolu a funkcemi nad ním prováděným.

Formuláře²³ **main** a **mapa** jsou vizuální částí aplikace a obsahují řadu funkcí pro své ovládání.

4.3.1.1. Třída „Protokol“

V této třídě je naprogramována definice vět NMEA protokolu ve formě datových struktur a výčtových typů.

²² Syntaxe jazyka Visual Basic. NET se liší, v některých částech podstatně, od dalších programovacích jazyků. Na druhou stranu zkušený programátor může vyčíst funkcionalitu a podobnosti si dát do souvislosti se „svým“ programovacím jazykem.

²³ Formulář se skládá ze dvou hlavních částí. A to jsou:

1. Kód grafického rozhraní, které Visual Studio generuje na pozadí při vytváření prvků. Tento kód můžeme samozřejmě editovat a měnit. Je možné i vytvářet nové prvky na formulář za běhu aplikace(musíme samozřejmě napsat příslušný kód)
2. Funkce a metody naprogramované programátorem

Například definice věty \$GPGLL. Vidíme, že je to struktura a její členi odpovídají definici věty z NMEA protokolu. Vidíme i že v této struktuře je i další struktura(Poloha) a výčtový typ(Polokoule).

```
Public Structure GPGLLData
    Public ZemepisnaSirka As Poloha
    Public PolokouleZemepisneSirky As Polokoule
    Public ZemepisnaDelka As Poloha
    Public PolokouleZemepisneDelky As Polokoule
    Public Hodiny As Integer
    Public Minuty As Integer
    Public Sekundy As Integer
    Public PlatnostDat As Platnost
    Public CheckSum As String
End Structure

Public Structure Poloha
    Public Stupne As Double
    Public Minuty As Double
End Structure

Public Enum Polokoule As UShort
    Sever = 0
    Jih = 1
    Východ = 2
    Západ = 3
End Enum
```

Stejným způsobem jsou definované i ostatní věty. Možná vznikne otázka: „Proč nejsou použity objekty?“. Odpovím, že zde nejsou objekty nutné, protože jsou tyto data jsou jen pro čtení. V celé aplikaci se jejich obsah nemění.

4.3.1.2. Třída ,funkce‘

Třída obsahuje funkci pro každou větu. Jako vstupní parametr je použito pole vět získaných z GPS odkazované referencí²⁴. Výstup funkce je struktura příslušné věty.

Princip: V poli vět se identifikuje příslušná věta a její obsah se ,rozseká‘ do odpovídajících úseků ve struktuře. Tato struktura se poté vrátí jako výstup.

²⁴ Reference se používá pro zamezení kopírování větších datových bloků v paměti. V tomto případě lze použít bezpečně, neboť data se používají jen pro čtení.

Například větě \$GPVTG přísluší funkce getGPVTG. Můžeme si též všimnout importu třídy protokol.

```
Imports Gps.protokol
Public Function getGPVTG(ByRef data As String()) As GPVTGData
    Dim tm, s As New String("")
    Dim GPVTG As New GPVTGData()
    Dim i As UInt32
    Dim hash As Integer = 0
    'Pole vet
    For Each s In data
        'ukončení vety
        s = s & ", "
        'GPVTG veta
        If Mid(s, 2, 5) = "GPVTG" Then
            'Prohledavani po vete
            For i = 8 To s.Length
                If Mid(s, i, 1) = ", " Then
                    hash = hash + 1
                    Select Case hash
                        Case 1
                            GPVTG.Kurz = Val(Mid(tm, 3, tm.Length))
                            tm = ""
                        Case 3
                            GPVTG.MagnetickySklon = Val(Mid(tm, 3, _25
tm.Length))
                            tm = ""
                        Case 5
                            GPVTG.RychlostUzly = Val(Mid(tm, 3, _
tm.Length))
                            tm = ""
                        Case 7
                            GPVTG.RychlostKmh = Val(Mid(tm, 3, _
tm.Length))
                            tm = ""
                        Case 9
                            GPVTG.CheckSum = Mid(tm, 3, tm.Length)
                            tm = ""
                    End Select
                Else
                    tm = tm & Mid(s, i, 1)
                End If
            Next i
        End If
    Next s
    Return GPVTG
End Function
```

Každá věta má opět takovouto funkci.

²⁵ Znak ,_‘ značí ve VB.NET rozdělení příkazu na další řádek.

4.3.1.3. Formulář ‚Main‘

Kromě vizuálního vzhledu se uvnitř formuláře **Main** skrývá jádro celé aplikace.

4.3.1.3.1. Funkce ‚Připojit‘

Funkce slouží k připojení GPS zařízení k aplikaci.

Funkci předchází vytvoření objektu sériového portu.

Ve funkci probíhá nastavení portu (kompatibilní s GPS přijímači) a otevření námi zvoleného portu se seznamu (viz. [kapitola 4.2.1](#), bod 2).

Je volána tlačítkem ‚Připojit‘.

Při jakékoliv chybě otevírání portu se otevře chybové hlášení s popisem chyby a funkce se ukončí- zařízení není připojeno.

Po úspěšném otevření sériového portu se indikuje úspěch a zpřístupní se další možnosti v ovládání aplikace (test, čtení dat, atd.).

Znepřístupní se znovupřipojení.

```
Private port As New SerialPort()
Private Function pripojit() As Boolean
    cmdPripojit.Enabled = False
    lstPorty.Enabled = False
    port.PortName = CType(lstPorty.SelectedItem, String)
    port.Parity = Parity.None
    port.BaudRate = 4800
    port.StopBits = StopBits.One
    port.DataBits = 8
    Try
        port.Open()
    Catch ex As Exception
        MsgBox("Chyba při otevírání portu." & Chr(10) & ex.Message & _
Chr(10) & ex.StackTrace, MsgBoxStyle.Critical)
        cmdPripojit.Enabled = True
        lstPorty.Enabled = True
        Exit Function
    End Try
    cmdOdpojit.Enabled = True
    cmdTest.Enabled = True
    lblStav.Text = "Port " & port.PortName & " otevřen"
    lblStav.ForeColor = Color.Green
    lblStavGps.Text = "zapnuto"
    lblStavGps.ForeColor = Color.Green
    'Zarizeni pripojeno
    cmdStart.Enabled = True
    'Call test()
End Function
```

4.3.1.3.2. Funkce ‚Odpojit‘

Funkce slouží k odpojení GPS zařízení od aplikace.

Je volána tlačítkem ‚Odpojit‘ (viz. [kapitola 4.2.1](#), bod 2).

Při jakékoliv chybě zavírání portu se otevře chybové hlášení s popisem chyby a funkce se ukončí- zařízení není odpojeno.

Při úspěšném zavření sériového portu se zavření indikuje a znepřístupní se možnosti, které byly dostupné při otevření portu a naopak.

```
Private Function odpojit() As Boolean
    Try
        port.Close()
    Catch ex As Exception
        MsgBox("Chyba při uzavírání portu." & Chr(10) & ex.Message & _
Chr(10) & ex.StackTrace, MsgBoxStyle.Critical)
        Exit Function
    End Try
    lblStav.Text = "Odpojeno"
    lblStav.ForeColor = Color.Red
    lblNmea.Text = "Ne"
    lblNmea.ForeColor = Color.Red
    lstVety.Items.Clear()
    lstPorty.Enabled = True
    cmdPripojit.Enabled = True
    cmdOdpojit.Enabled = False
    cmdTest.Enabled = False
    lblStavGps.Text = "vypnuto"
    lblStavGps.ForeColor = Color.Red
    'vypnuti zpracovavani dat
    tmr.Enabled = False
    'zarizeni odpojeno
    cmdStart.Enabled = False
    cmdStop.Enabled = False
End Function
```

4.3.1.3.3. Funkce „Čti“

Čte data z GPS zařízení a vrací jejich řetězec.

Funkci předchází vytvoření objektu kódování.

Ze sériového portu se do **bufferu** načtou data jako pole **Byte** a pomocí kódování se převedou²⁶ na **řetězec**.

Při chybě zapříčiněné nemožností čtení dat, vráceno chybové hlášení.

```
Private encoding As System.Text.Encoding =  
System.Text.ASCIIEncoding.GetEncoding(1252)  
Private Function cti() As String  
    Dim temp As New String("")  
    Dim bData(479) As Byte  
  
    Try  
        port.Read(bData, 0, 479)  
    Catch ex As Exception  
        MsgBox("Chyba při čtení dat." & Chr(10) & ex.Message & Chr(10) _  
& ex.StackTrace, MsgBoxStyle.Critical)  
    End Try  
    temp = encoding.GetString(bData)  
    Return temp  
End Function
```

²⁶ Funkce GetString (bytes () As Byte) převede pole Byte na typ String

4.3.1.3.4. Funkce ‚Čti větu‘

Z dat, které jsme získali z předchozí funkce ‚Čti‘ nám tato funkce rozdělí data do pole vět typu řetězec.

Nejprve zjistíme, jestli se v datech nachází znak uvozující věty(,\$'). Poté procházíme data až do dalšího výskytu uvozujícího znaku a prošlá data si ukládáme do pomocně proměnné ‚veta‘. Když dojde na výskyt uvozujícího znaku(,\$') předimenzujeme výsledné pole o 1 a vložíme do něj dočasnou proměnou věta, která uchovává obsah mezi uvozujícími znaky(,\$'), tedy větu.

Tímto máme připravené věty ke zpracování funkcemi pro zpracování vět(viz [kapitola 4.3.1.2](#))

```
Private Function ctiVetu(ByVal data As String) As String()
    Dim tmp() As String = {}
    Dim veta As New String("")
    Dim i, pocet As UInt16
    i = InStr(data, "$")
    If i < 1 Then
        'Throw New Exception("Zařízení vysílá nesouvislá data!")
        Return tmp
    End If
    For i = i To data.Length - 1
        If Mid(data, i, 1) = "$" Then
            If Len(veta) > 6 Then
                pocet = pocet + 1
                ReDim Preserve tmp(pocet - 1)
                tmp(pocet - 1) = veta
                veta = ""
            End If
            veta = veta & Mid(data, i, 1)
        End If
    Next i
    Return tmp
End Function
```

4.3.1.3.5. Jádru systému

Chod celého jádra systému obstarává časovač ,tmr‘, který v intervalu, který si zvolíme vykonává funkce, které jsme nastavili.

```
Private Sub tmr_Tick(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles tmr.Tick
```

Vzhledem k rozsáhlosti zde nebudu uvádět jeho zdrojový kód, ale jen popíši jeho princip.

Zdrojový kód je dostatečně okomentován a vysvětlení, které uvedu je pro pochopení funkčnosti dostačující.

1. Vytvoří se struktury vět
2. a) Pokud není simulátor(viz. [kapitola 4.2.4](#)) spuštěn, načtou se data z GPS zařízení.
b) Pokud je spuštěn, načte se soubor s daty. Tyto data jsou předávána dále k zpracování po počtu vět, který si zvolíme. Po dosažení konce dat ze souboru se simulátor automaticky vypne.
3. Další krok plní data získaná z vět do jejich příslušných popisek v aplikaci, do záložky ,Údaje‘(viz. [kapitola 4.2.3](#)).
4. Zobrazování satelitů(pokud máme tuto možnost zaškrtnutou, viz. [kapitola 4.2.5](#))
5. Zobrazování dat(pokud máme tuto možnost zaškrtnutou, viz. [kapitola 4.2.2](#))
6. Nahrávání dat(pokud máme tuto možnost zaškrtnutou, viz. [kapitola 4.2.2](#))
7. Zobrazování na mapě(pokud máme tuto možnost zaškrtnutou, viz. [kapitola 4.2.6](#))
8. Všechny tyto úkony měří časovač a výsledek měření zobrazuje do popisku(viz. [kapitola 4.2.1](#), bod 4)
9. Zobrazování satelitů(pokud máme tuto možnost zaškrtnutou, viz. [kapitola 4.2.5](#))

4.3.1.4. Formulář ,Zobrazování na mapě‘

Formulář sloužící ke zobrazování dat na mapě.

4.3.1.4.1. Zobrazení formuláře

Zobrazení (viz. [kapitola 4.2.6](#)).

Při zobrazení se vytvoří záloha původního obrázku mapy, pro pozdější využití.

4.3.1.4.2. Nastavení kotvících bodů

Nastavení se provádí vyplněním údajů do textových políček a nastavení pozice.

Pozice je odčítána podle polohy kurzoru. Následně je na mapu vykreslen obrázek pivotu, s průhlednou vrstvou pozadí. Souřadnice souřadnicového systému Visual Studio(pixely) se uloží jako objekt **Point**.

Z důvody obnovování **PictureBoxu**, ve kterém se nachází mapa, jsem musel použít dočasný objekt typu **Bitmap**, do kterého se provádí veškeré vykreslování a výsledná podoba se zobrazuje do **PictureBoxu**.

4.3.1.4.3. Algoritmus vykreslování

1. Známe pozice obou bodů v pixelech a jejich zeměpisné šířky a délky.
2. Nejprve provedeme odečet obou bodů, tím zjistíme jejich odchylky.
3. Poté vypočítáme jaká hodnota zeměpisné délky a šířky připadá na jeden pixel.
4. Nově příchozí bod odečteme taktéž od pivotu, tím získáme také jeho odchylku.
5. Tuto odchylku vynásobíme poměrem hodnot, které připadají na jeden pixel a dostaneme výsledné souřadnice v pixelech.
6. Bod vykreslíme.

5. Závěr

Cílem práce bylo poskytnout čtenáři informace o systému GPS a to v komplexnější míře. Pro vytvoření aplikace bylo nutné pochopit některé principy systému. Pro uživatele, který bude pracovat s aplikací ovšem toto nutné není. Záleží na každém jedinci, zda se chce dozvědět něco nového.

Část věnovaná Visual Studiu 2005 je souhrnem, který je zřejmě nejvíce užitečný aktivním programátorům. Využívání některých novinek opravdu zlepši a zpříjemní práci a zvýší tak efektivitu.

Samostatná aplikace byl pro mě asi největší oříšek. Byl to nejen test mých schopností programování a myšlení, ale také trpělivosti. Nejvíce obtížná část byla zobrazování dat na mapě. Testování bylo obtížné z několika důvodů. Mnohdy jsem nevěděl, zda je nepřesnost zobrazení způsobená příjmem dat z GPS, výpočty, či nepřesností mapy. Když vše mělo fungovat jak má, ale nefungovalo, jsem po značném pátrání zjistil, že data přijímaná GPS zařízením jsou ve formátu mapového systému **WGS-84**, ve kterém již vše funguje bez sebemenších problémů.

6. Použité zdroje

<http://nb.vse.cz/~zelenyj/>
<http://www.connect.cpress.cz/>
<http://gps.slansko.cz/>
<http://www.skyfly.cz/gpspraxe/>
<http://obchod.sunnysoft.cz/>
<http://www.palmserver.cz/>
<http://www.iabc.cz/>
<http://gis.vsb.cz/vojtek/>
<http://home.zf.jcu.cz/>
http://www.commlinx.com.au/NMEA_sentences.htm
<http://msdn.microsoft.com/vbasic/>