

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

GPS PRO VÝUKOVOU HW/SW PLATFORMU

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

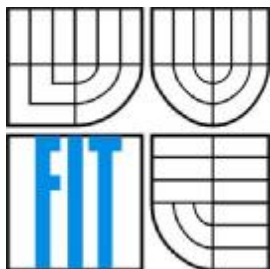
AUTHOR

Pajgrt Ondřej

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

GPS PRO VÝUKOVOU HW/SW PLATFORMU

GPS FOR TEACHING HW/SW PLATFORM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Pajgrt Ondřej

VEDOUCÍ PRÁCE
SUPERVISOR

Fučík Otto, Dr. Ing.

BRNO 2008

Abstrakt

Práce se zabývá popisem spojení výukové platformy FITkit s GPS modulem. Seznámíme se v ní s problematikou systému GPS z teoretického i praktického hlediska, s výukovou platformou FITkit a jejím obecným využitím. Také podrobně popisuje spojení těchto dvou systémů ve spolupracující celek. Rozebírá též různé komunikační rozhraní a protokoly, které byly pro toto spojení využity. Ve druhé části práce je pak uveden postup zpracovávání dat od GPS modulu až po výsledné zobrazení na monitoru, samotná implementace aplikace a popis její obsluhy.

Klíčová slova

GPS, NMEA, platforma FITkit, MCU, FLASH, FPGA, řadič, SPI, VGA, SDRAM, RS232, C, C#, VHDL, libfitkit, bitmapa, RGB.

Abstract

The work deals with a connection between HW/SW platform FITkit and GPS device. Introduces GPS systems from theoretical and practical view describes FITkit teaching platform and its common use and shows in detail the connection of those two systems into a cooperative unit. Work also analyses communication interfaces and protocols used for that connection. Finally it provides a description of the whole system data flow from the very beginning in GPS unit to the final displaying on the screen and description of implementation and application usage.

Keywords

GPS, NMEA, FITkit platform, MCU, FLASH, FPGA, driver, SPI, VGA, SDRAM, RS232, C, C#, VHDL, libfitkit, bitmap, RGB.

Citace

Pajgrt Ondřej: GPS pro výukovou HW/SW platformu. Brno, 2008, bakalářská práce, FIT VUT v Brně

GPS pro výukovou HW/SW platformu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Dr. Ing. Otto Fučíka.

Další informace mi poskytli Ing. Ladislav Čapka, Ing. Michal Pajgrt.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
jméno příjmení
datum

Poděkování

Chtěl bych poděkovat vedoucímu práce, Ing. Otto Fučíkovi, za poskytnutí odborných rad a připomínek při řešení dané práce. Dále bych chtěl poděkovat Ing. Ladislavu Čapkovi za vydatnou pomoc s VGA radičem, Ing. Michalu Pajgrtovi za spoustu užitečných rad při řešení práce a firmě Smart, s.r.o za poskytnutí nástrojů potřebných k ladění.

© Ondřej Pajgrt, 2008.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	3
Úvod	4
1 GPS	5
1.1 Historie.....	5
1.2 Princip určování polohy	5
1.3 Přesnost a chyby	6
1.4 Komunikace	6
2 FITkit.....	7
2.1 Popis kitu.....	7
2.2 Propojovací systém - SPI	8
2.3 RS232.....	8
2.4 VGA.....	9
3 Základ aplikace	10
3.1 Top-level entita.....	10
3.2 GPS modul	10
3.3 Řadič RS232.....	11
3.4 Řadič přerušení.....	11
3.5 Řadič SPI.....	11
4 Zpracování dat	12
4.1 Příjem.....	12
4.2 Třídění.....	12
4.3 Úprava.....	13
5 Vizualizace dat.....	14
5.1 Řadič LCD	14
5.2 Řadič VGA+SDRAM	14
6 Mapy	15
6.1 Formát a uložení	15
6.2 Tvorba map	16
7 Použití aplikace.....	17
7.1 Postup.....	17
Závěr.....	19
Literatura.....	20
Seznam příloh	21

Úvod

Fakulta informačních technologií (FIT) VUT v Brně zavedla v roce 2006 do výuky platformu FITkit, jejíž účelem bylo, aby byli studenti seznámeni s látkou nejen po teoretické, ale i praktické stránce. Každoročně je knihovna ovladačů a aplikací pro kit rozšiřována projekty jak pedagogů, tak studentů a jedním z takových projektů je právě i tato práce. Jejím cílem je zobrazení dat přijatých z GPS modulu na PC za pomoci platformy FITkit.

Téma jsem si zvolil ze zájmu o technologii GPS, která se v dnešní době stále více dostává do povědomí běžných uživatelů mobilních komunikací a pole jejího uplatnění se rychle rozšiřuje. Dalším významným plus pro toto téma byl samotný FITkit, který zastupuje větev informačních technologií, s nimiž se setkáváme téměř na každém kroku, ačkoli si to možná ani neuvědomujeme – Vestavěné systémy dnes tvoří základ velkého množství běžně užívané elektroniky.

Podrobně se podíváme na technologii GPS a platformy FITkit, což nám připraví teoretický základ potřebný k dalšímu porozumění této práce. Mimoto jsou předpokládány alespoň základní znalosti z oblasti počítačových systémů a informačních technologií obecně.

Projdeme postupně celou pomyslnou „cestu“, po které putují data z GPS přijímače do FITkitu, jak se v něm mění jejich struktura a jak pokračují dále do výsledného zobrazení na monitoru v podobě blikající tečky na mapě, která vyznačuje aktuální pozici GPS snímače vzhledem k okolnímu prostředí.

Na závěr si také popíšeme, jak si připravit mapu a další potřebná data pro tuto aplikaci, aby ji bylo možné použít kdekoli a stala se tak další z řady volně využitelných a univerzálních aplikací pro FITkit.

1 GPS

GPS (Global Positioning Systém, viz [1]) je v současné době jediným plně fungujícím systémem satelitní navigace, umožňujícím jeho uživatelům určit aktuální zeměpisnou polohu, rychlost, směr a čas kdekoli na zemském povrchu za předpokladu, že dotyčný má přímý výhled na větší část oblohy.

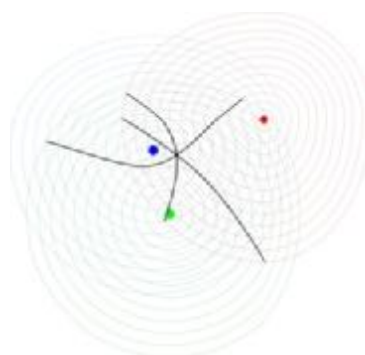
Tvoří jej jednosměrná komunikace mezi vysílači, tj. 24 družic rovnoměrně rozprostřených nad zemským povrchem ve výšce přibližně 20 000km, a přijímačem, který z přijatých dat zprostředkovává uživateli potřebné informace.

1.1 Historie

GPS byl původně vojenským systémem, vyvíjeným a budovaným od roku 1973 Ministerstvem obrany Spojených států. Od roku 1983 byl zpřístupněn pro civilní použití, avšak s jistými omezeními. Nejzásadnější z nich byla tzv. selektivní dostupnost (selected availability) – záměrné zhoršení přesnosti určení výsledné polohy, které mělo zabránit případnému zneužití systému (např. pro navádění balistických střel). Tato opatření byla ovšem v roce 2000 zrušena, čímž se přesnost téměř 10x zvýšila a nabídlo se tak plné uplatnění GPS v civilní sféře. Dnes se s ním nejčastěji můžeme setkat v podobě navigačních přístrojů pro motorová vozidla, turistiku atd.

1.2 Princip určování polohy

K určení přesné polohy je zapotřebí výhled a tudíž možnost přijímat data ze 3 družic (dvojměrná poloha – zeměpisná šířka a délka), popř. 4 družic (navíc výška). Každá družice vysílá složitý signál, obsahující aktuální pozici, čas a přibližné informace o ostatních družicích. Samotná poloha je vypočtena v přijímači. Využívá se rozdílu mezi časem vyslání signálu a jeho přijetím na Zemi. Z něj je následně vypočtena vzdálenost od družice. Aktuální polohu poté získáme průsečíkem těchto vzdáleností podle obrázku:



Obrázek 1.1 – Určení polohy za pomoci 3 satelitů

1.3 Přesnost a chyby

Obecná přesnost v terénu se za ideálních podmínek v případě tzv. „turistických“ GPS pohybuje kolem 5 metrů. Existují i přesnější moduly, spolupracující např. s pozemními stanicemi, jejichž odchylka může být i menší než 20cm. Výsledná přesnost však závisí na mnoha faktorech, které ji negativně ovlivňují.

Tím hlavním a základním je výhled. Pokud přijímač nemá přímý výhled na oblohu (konkrétně aspoň 3 družice na ní), není možné určit zeměpisnou polohu vůbec. Viditelnost může být omezena například i velmi špatným počasím. Přesnost ovlivňuje i postavení satelitů na obloze vůči pozorovateli. Optimální je jeden satelit v nadhlavníku a další tři rovnoměrně rozmístěné na obloze.

Na výslednou polohu mají vliv ještě další aspekty, které zmenšují přesnost měření v řádu několika metrů. Asi největší vliv na to mají atmosférické vrstvy, převážně ionosféra a troposféra, ve kterých je přenášený signál zpomalen. Dalším problémem je zpoždění aktualizace polohy družice (tzv. efemerida), zpoždění popř. předbíhání hodin (tzv. clock drift), rušivé signály, odražené od různých překážek v terénu (tzv. multipath), a chyby numerických výpočtů. V minulosti byla však největším problémem výše zmíněná selektivní dostupnost (selected availability), umělá chyba zavádějící nepřesnost až přes 10 metrů. Následující tabulka ukazuje přibližné hodnoty odchylek způsobených výše uvedenými problémy.

Selektivní dostupnost	~ 10 m
Ionosféra	~ 5 m
Efemerida	~ 2,5 m
Clock drift	~ 2 m
Multipath	~ 1 m
Troposféra	~ 0,5 m
Numerické chyby	~ 1 m

Tabulka 1.2 – Chyby GPS

1.4 Komunikace

Ke komunikaci s dalšími zařízeními používá GPS přijímač armádní protokol NMEA (The National Marine Electronics Association, viz [2]). Byl zaveden Námořnictvem Spojených států za účelem sjednocené komunikace různých armádních přístrojů s PC, popř. mezi sebou.

Skládá se z tzv. vět, tvořených ASCII znaky, které jsou samostatné a nezávislé na ostatních. Pro GPS přijímač je těchto vět přibližně 30, přičemž základní informace, jako například čas a polohu, obsahuje rovnou několik z nich. Je tedy možné vybrat si právě ty, které nejvíce vyhovují našim požadavkům na zpracovávaná data podle toho, jaké detaily o nich potřebujeme vědět. Každá věta začíná symbolem '\$' a není delší než 80 znaků. Podrobný popis věty bude uveden poději.

2 FITkit

FITkit (viz [3]) je HW/SW platforma, která vznikla na Fakultě informačních technologií VUT v Brně k výukovým účelům. Jejím cílem je umožnit studentům naučit se navrhovat a prakticky realizovat softwarové i hardwarové aplikace a lépe pochopit jejich vzájemnou provázanost.



Obrázek 2.1 - FITkit

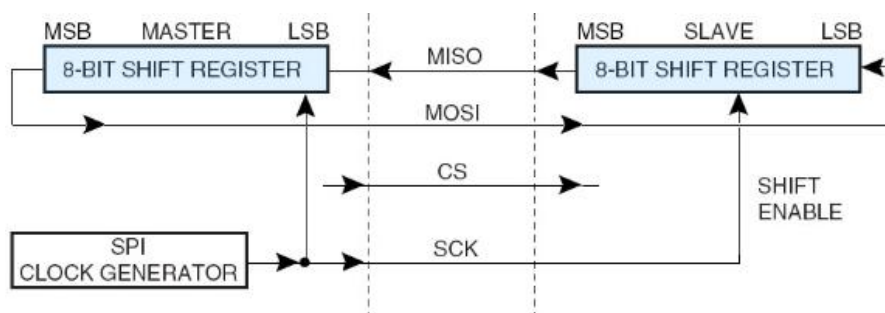
2.1 Popis kitu

Je tvořen výkonným mikrokontrolérem s nízkým příkonem a reprogramovatelným hardwarem na bázi FPGA (Field Programmable Gate Array), umožňujícím opakované přizpůsobení pro různé účely výsledné aplikace. Jeho součástí je i velké množství vstupních a výstupních konektorů, připojených k FPGA, čímž je umožněno, po zavedení správného ovladače, používat periferie jako VGA, RS232, USB, PS/2 atd. Pro práci s těmito periferiemi byla vytvořena sada řadičů, které mají za úkol zjednodušit návrh a implementaci dalších projektů. Základní práci s FITkitem po softwarové stránce pak obstarává knihovna *libfitkit* a sada ovladačů, podporujících práci s jednotlivými periferiemi v FPGA (SPI, LCD, přerušení atd.).

Spojení s kitem probíhá přes virtuální COM port v PC a terminálovou aplikaci. Program pro mikrokontrolér je napsán v jazyce C a přeložen GNU překladačem za pomoci programu GMAKE a jeho dávkového souboru, který výrazně ulehčuje překlad a obstarává samotné naprogramování MCU. Data pro FPGA jsou tvořena profesionálními návrhovými systémy podle popisu v jazyce VHDL. Výsledný binární konfigurační soubor je poté nutné nahrát do kitu přes terminál a vyvolat reset MCU. Podrobnější popis programování kitu je uveden v jeho dokumentaci.

2.2 Propojovací systém - SPI

Interní komunikace ve FITkitu mezi MCU, FLASH a FPGA je zajištěna využitím synchronního rozhraní SPI. Princip spočívá ve výměně dvou osmibitových registrů – jednoho v zařízení typu master (v našem případě MCU) a druhého v zařízení typu slave (FLASH nebo FPGA, rozlišení adresované komponenty probíhá na úrovni adresových bitů).

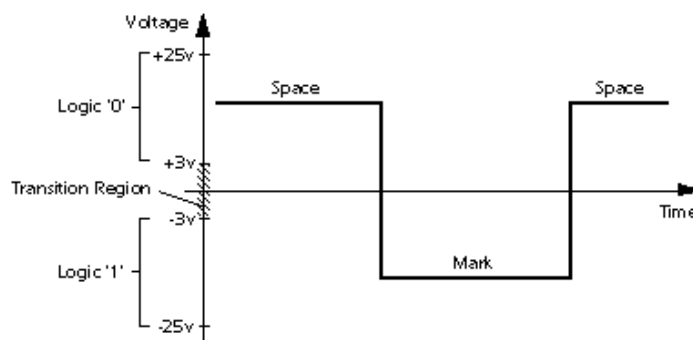


Obrázek 2.2 – Popis rozhraní SPI

Toto rozhraní umožňuje kromě komunikace s FLASH pamětí, uchovávající program pro MCU a konfiguraci FPGA, také komunikaci se všemi komponentami implementovanými v FPGA. Její základ tvoří SPI řadič, který provádí konverzi SPI protokolu na jednoduchý vnitřní sériový protokol. Vnitřní sběrnice může být obsazena libovolným počtem SPI dekodérů, jejichž úkolem je jednak dekódovat adresu cílového zařízení, ale hlavně poskytnout jednoduché paralelní rozhraní umožňující obousměrnou komunikaci. Tyto dekodéry a jim odpovídající zařízení je poté možné jednoznačně adresovat podle šířky adresní části a báze adresy. Bližší informace najdeme na [4].

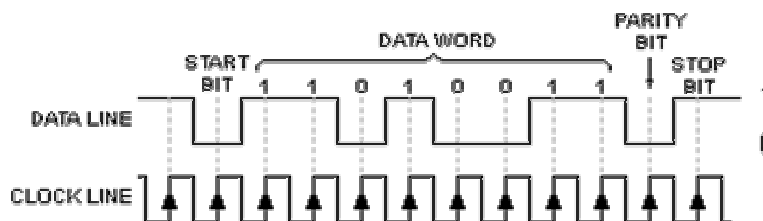
2.3 RS232

Pro přenos dat mezi GPS modulem a FITkitem je použito rozhraní RS232. Jedná se o asynchronní sériové rozhraní pro komunikaci dvou zařízení, řízené pomocí pevně nastavené přenosové rychlosti. Ta se udává v Baudech (počet přenesených bitů za sekundu). Pro větší odolnost proti vnějšímu rušení se používá vyšší napětí, než standardních 5V. Logickým hodnotám zpravidla odpovídá napětí +10V (log. 0, označována jako *space state*) resp. -10V (log. 1, označována jako *mark state*).



Obrázek 2.3 – Napěťové úrovně

Samotná komunikace je pak zahájena tzv. Start bitem, jehož sestupná hrana zároveň slouží jako synchronizační signál hodin pro asynchronní přenos. Následují data (zpravidla 8 bitů), paritní bit (je-li kontrola paritou použita) a stop bit(y). Pro detailnější informace viz [5].



Obrázek 2.4 – Průběh přenosu jednoho datového rámce

2.4 VGA

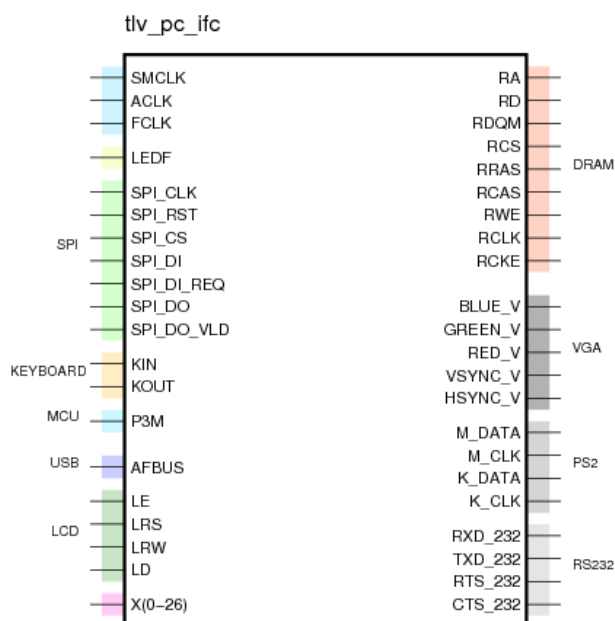
Grafický výstup aplikace je realizován přes port VGA. Jedná se o analogové rozhraní, sloužící pro připojení zobrazovacího zařízení k výstupu grafické karty. Zobrazovaná data jsou reprezentována třemi základními barevnými složkami (červená, zelená a modrá), ze kterých je možné vytvořit širokou škálu barev. Pozice právě vykreslovaného bodu na obrazovce je určena pomocí 2 synchronizačních pulzů: horizontální (je vyvolán vždy na konci řádku) a vertikální (je vyvolán vždy po vykreslení celé obrazovky). Přijímací zařízení nepotřebuje už žádné další informace o režimu, rozlišení atd. Vše si odvodí z přijímaných dat. Bližší informace na [6].

3 Základ aplikace

Samotné jádro aplikace vychází z několika komponent, obsažených v knihovně libfitkit. Jedná se především top-level entitu a základní řadiče pro komunikaci – řadič SPI, přerušení a sériové komunikace RS232. Jejich použitím a za pomoci drobných úprav na nich získáme základní funkčnost a můžeme se plně věnovat rozšiřování této „kostry“ o části specifické pro naši aplikaci.

3.1 Top-level entita

Knihovna *libfitkit* umožňuje pro vývoj aplikací výběr ze 3 základních top-level entit – *bare*, *pc* a *ide*. Našemu použití nejlépe vyhovuje architektura *pc*, která byla navržena právě pro aplikace přistupující k rozhraním VGA a RS232, které v tomto projektu využijeme.



Obrázek 3.1 – Entita `tlv_pc_ifc`

V definici této entity (soubory `tlv_pc_ifc.vhd` a `arch_pc_ifc.vhd`) bylo také zapotřebí opravit chybu, týkající se komunikace RS232, konkrétně prohození směru signálů `RTS_232` (out na in) a `CTS_232` (in na out).

3.2 GPS modul

Jako zdroj GPS dat je v tomto projektu použit modul s čipem SiRF GPS2e/LP-7450, jehož výstup byl z původní TTL logiky předělán na rozhraní RS232, čímž se výrazně zvýšil počet zařízení, které jsou schopny s ním spolupracovat (např. PC nebo FITkit). Jako zdroj napájení +5V byl použit USB konektor pro jednoduchou manipulaci s celým systémem.

Modul není nutné nijak inicializovat. Sám se nakonfiguruje po zapnutí a začne posílat data. V základním nastavení modul posílá každou vteřinu věty GGA (čas, detailní zeměpisná poloha, platnost měření), RMC (čas, zeměpisná poloha, rychlost, azimut) a VTG (detailní rychlost) a každých 5 vteřin větu GSA (informace o dostupných satelitech).

3.3 Řadič RS232

Pro umožnění komunikace modulu s kitem je nutné implementovat řadič sériové komunikace také do FPGA. Byla využita komponenta rozhraní sériové komunikace z knihovny libfitkit, kterou bylo potřeba správně propojit s top-level entitou (konkrétně připojit její signály RXD, TXD, RTS a CTS na konektor CAN9 (JP5)). Řadič byl inicializován na hodnoty, odpovídající nastavení komunikace v GPS:

<i>Rychlost:</i>	4800 Baud
<i>Datové bity:</i>	8
<i>Stop bity :</i>	1
<i>Parita :</i>	0

Tabulka 3.2 – Nastavení řadiče RS232

3.4 Řadič přerušení

Pro správnou obsluhu sériové komunikace je potřeba přidat z knihovny libfitkit také řadič přerušení, propojit ho s řadičem RS232 a nakonfigurovat na filtrování všech přerušení kromě sériového. To nastane v momentě přijetí, resp. odesílání jednoho rámce po sériové lince.

Vzhledem ke starší verzi FITkitu bylo také nutné propojit pin JP10(5) (signál z FPGA) s pinem JP9(26) (signál z MCU). Bez tohoto zásahu by celý systém přerušení nemohl fungovat.

3.5 Řadič SPI

Interní komunikaci mezi MCU a prvky v FPGA zajišťuje rozhraní SPI. Proto je potřeba na každou jednotlivou komponentu, která má být schopna s MCU komunikovat, přidat adresový dekodér SPI a zvolit jim správné báze adresy tak, aby se jejich adresové prostory nepřekrývaly.

4 Zpracování dat

Po spojení FITkitu a GPS přes sériovou linku a zapnutí napájení obou zařízení už modul posílá data, která je nutno zpracovávat, tzn. přijímat je a ukládat, třídit a dále upravovat pro následné použití.

4.1 Příjem

V okamžiku přijetí rámce po sériové komunikaci vyvolá řadič přerušení obslužnou rutinu, která uloží přijatý byte do přijímacího bufferu. V každém průchodu hlavní smyčkou programu se poté kontroluje jeho velikost a v případě, že není prázdný, je jeho obsah přesunut do druhého bufferu, který slouží pro uchování právě přijímané věty. V momentě, kdy je v tomto bufferu rozpoznána celá věta (ukončena zalomením řádku znaky `\r\n`), je vyvoláno její další zpracovávání. Po jeho dokončení je buffer vyprázdněn a může být přijímána další věta.

4.2 Třídění

Přestože modul posílá 4 různé věty, pro účely naší aplikace bude pro zjednodušení stačit zpracovávat jednu z vět GGA (akt. čas, poloha, výška, kvalita signálu) a RMC (akt. čas, poloha, datum, rychlost). Byla vybrána věta GGA kvůli výšce a hlavně informaci o kvalitě a platnosti signálu, která se ukázala být velice užitečnou v době ladění programu. V každé přijaté větě se před zpracováním zkontroluje její identifikace a pokud se nejedná o GGA, popř. je věta neúplná či jinak poškozená, je zahozena a dále se nezpracovává.

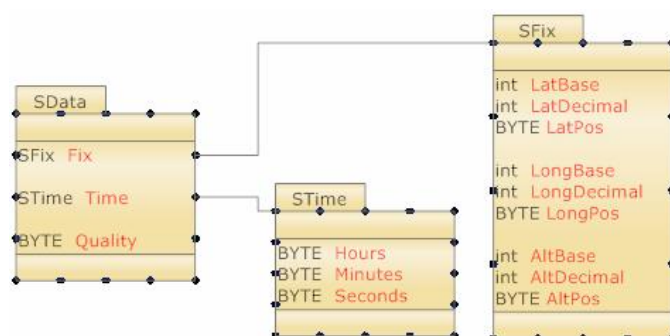
4.3 Úprava

Nyní máme tedy k dispozici platnou GGA větu. Je nutné ji rozložit na jednotlivé položky (parsovat), které jsou odděleny čárkou. Jejich jednotlivý význam uvádí následující tabulka:

\$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47			
Kde:			
	\$	Začátek věty	
	GP	Označení cílového zařízení (GP=GPS)	
	GGA	"Global Positioning System Fix Data"	
	123519	Čas porřízení měření	12:35:19 UTC
	4807.038,N	Zeměpisná šířka	48°07.038' N (N=severní šířky)
	01131.000,E	Zeměpisná délka	11°31.000' E (E=východní délky)
	1	Kvalita měření:	0 = invalid (neplatné)
			1 = GPS fix (platné)
			2-8 = specifické hodnoty
	08	Počet aktuálních satelitů	
	0.9	DOP - vhodnost rozmístění družic (rostoucí=horší)	
	545.4,M	Nadmořská výška v metrech	
	46.9,M	Nadmořská výška geoidu nad elipsoidem WGS84	
	*47	Kontrolní součet,vždy začíná *	

Tabulka 4.1 – GGA věta

Z nich jsou vybrány ty, které mají pro naši aplikaci význam, konkrétně čas, zeměpisná poloha a kvalita. Tyto informace jsou dále zpracovány a uloženy v datové struktuře *Data* typu *SData*. Údaje o zeměpisné poloze (šířka, délka a nadmořská výška) jsou rozděleny na celou část a desetinnou část z důvodu ulehčení následné manipulace s nimi. S těmito údaji se poté pracuje až do přijetí další platné GGA věty, která je aktualizuje.



Obrázek 4.1 – Datová struktura SData

5 Vizualizace dat

Po zpracování údajů z GPS následuje jejich předání uživateli v nějaké rozumné a pochopitelné formě. Tato kapitola popisuje jednotlivé komponenty, které tato aplikace bude využívat k vizualizaci dat. Patří mezi ně řadič LCD displaye a úzce spolupracující řadiče VGA a SDRAM.

5.1 Řadič LCD

Jednoduché zobrazení načtených dat obstará 16ti místný LCD display samotného kitu, jehož ovladač je součástí knihovny libfitkit. Lze na něm zobrazit všechny 4 načtené informace. Přepínat mezi nimi lze přes terminál PC specifickými příkazy:

TIME	Přepne LCD display na zobrazení času.
FIX	Přepne LCD display na zobrazení polohy.
ALT	Přepne LCD display na zobrazení výšky.
QUALITY	Přepne LCD display na zobrazení kvality/platnosti signálu.

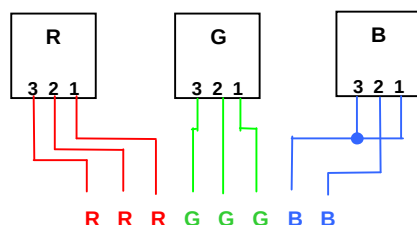
Tabulka 5.1 – Příkazy pro LCD display

Data na displayi se aktualizují po zpracování každé platné věty, tzn. každou vteřinu. Při zobrazení zeměpisných souřadnic se z důvodu malého rozměru průběžně střídá zeměpisná délka a šířka.

5.2 Řadič VGA+SDRAM

Pro složitější zobrazení je využit výstup na monitor konektorem JP2 přes řadič VGA. Jako základ byla použita aplikace *vga_sdram*, ve které ovšem musela být nejdříve za pomoci Ing. Čapky opravena chyba, která způsobovala šum ve výsledném obraze. Aplikace využívá paměť SDRAM simulovanou v FPGA jako zdroj dat. VGA řadič tuto paměť neustále čte, zpracovává a vykresluje na monitor, takže jakákoli změna v SDRAM se ihned projeví na výsledném obraze.

Z kapacitních důvodů jsou data v SDRAM uložena v hustotě 1 Byte/pixel. Samotný VGA řadič ale používá pro vykreslování 3 bity pro každou ze 3 barev (červená, zelená, modrá), dohromady tedy 9 bitů. Z tohoto důvodu jsou data v SDRAM uložena ve standardizovaném 8-bitovém formátu RRR-GGG-BB (3x červená, 3x zelená, 2x modrá, viz [7]). Vzhledem k chybějícímu jednomu bitu pro modrou barvu je jí odpovídající nejnížší bit ve VGA řadiči nastaven podle nejvyššího bitu modré barvy, aby se zajistila její rovnoměrnost.



Obrázek 5.2 – Zapojení 8-bitových dat na 9-bitový řadič

Bez tohoto opatření by nebylo možné zobrazit zcela černou barvu (v případě, že by tento nejnižší bit byl nastaven implicitně na hodnotu 1), resp. zcela bílou barvu (při implicitním nastavení na 0).

Data (v našem případě mapy pro GPS) je nutné po každém zapnutí kitu nahrát z FLASH paměti do SDRAM. Její jednoduchý ovladač ovšem neumožňuje současné čtení a zápis do této paměti. VGA řadič ji však čte téměř neustále (načítá z ní data pro vykreslení), takže pro přenos dat z FLASH do SDRAM se používá zapisovací buffer a musí se počkat na „pauzu“ mezi tímto čtením. Takový okamžik nastává při čekání VGA řadiče na synchronizaci. Obsah zápisového bufferu se uloží do SDRAM, kterou následně opět zabere VGA pro čtení. Z tohoto důvodu je překreslení celé obrazovky o rozměrech 512 x 480 px poměrně časově náročné (trvá přibližně 1 minutu).

6 Mapy

Základní funkcí této aplikace je zobrazení dat, získaných z GPS. Nejlepším postupem bude pravděpodobně svázat tyto informace o zeměpisné poloze přímo s nějakou mapou, na které je budeme zobrazovat. V této kapitole si popíšeme, jak je tato mapa ve FITkitu reprezentována a jak případně vytvořit a přidat nějakou další, vlastní mapu.

6.1 Formát a uložení

Jak už bylo zmíněno výše, po zapnutí kitu je třeba nahrát mapu z FLASH do SDRAM. Uživatelsky přístupná část FLASH, která je pro uchování mapy využita, je ovšem omezena kapacitou přibližně 215 KB (815 stránek po 264 Bytech), což značně omezuje rozměry a kvalitu této mapy. Barevná hloubka uložené bitmapy je 8 bitů ve formátu RRR-GGG-BB, stejně jako formát uložení v SDRAM, a není tedy zapotřebí žádná další konverze. Standardní rozměr mapy je zvolen 500 x 400 px s ohledem na kapacitu FLASH a rozměry zobrazení, které podporuje VGA řadič.

Součástí datového souboru s mapou je i hlavička o velikosti 38 KB, připojená na začátek, ve které jsou uloženy kotvící informace, vztahující se k této mapě, což umožňuje její modularitu a změnu aktuální sledované oblasti pouhým nahráním nové mapy do FITkitu. Obsahuje čtveřici

devítimístných čísel, reprezentujících souřadnice hraničních bodů, a dvojici písmen, určujících směr těchto souřadnic (severní/jižní šířka, východní/západní délka).

Detailní popis hlavičky je v následující tabulce:

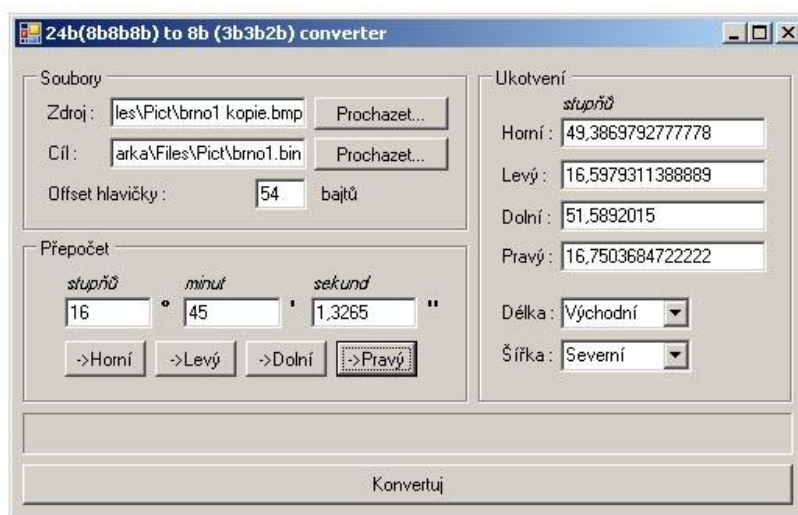
051654284011690323047757166019159964NE			
Pozice	Obsah	Funkce	Hodnota
0-8B	051654284	Horní ohraničení, zeměpisná šířka	51,654284°
9-17B	011690323	Levé ohraničení, zeměpisná délka	11,690323°
18-26B	047757166	Spodní ohraničení, zeměpisná šířka	47,757166°
27-35B	019159964	Pravé ohraničení, zeměpisná délka	19,159964°
36B	N	Směr zeměpisné šířky (N/S)	N=severní
37B	E	Směr zeměpisné délky (E/W)	E=východní

Tabulka 6.1 – Příklad hlavičky mapového souboru

Tyto kotvící informace se načtou do MCU při startu kitu nebo po nahrání mapy z FLASH do SDRAM vyvolaným příkazem LOAD.

6.2 Tvorba map

Díky výše zmíněné možné modularitě map, se kterými FITkit pracuje, byl tento projekt rozšířen o možnost tvorby nových, vlastních mapových souborů. Pro tyto účely byla v jazyce C# vytvořena speciální aplikace *Map Converter*, která ze zadané bitmapy a kotvících souřadnic generuje zdrojový soubor s mapou ve stanoveném formátu.



Obrázek 6.2 – Ukázka aplikace Map Converter

Jako vstupní data (soubor zvolený v poli *Zdroj*) je vyžadována bitmapa o rozměrech 500 x 400 px, uložená ve speciálním 24-bitovém formátu R8 G8 B8 (8 bitů na každou barvu) s převráceným pořadím řádků, které nám umožní jednodušší manipulaci s výslednou mapou ve

FITkitu.. Takto ukládat bitmapy umožňuje např. Adobe Photoshop. Původní soubor navíc obsahuje hlavičku bitmapy, ve které jsou uloženy detailnější informace o ní, její rozměry, paleta barev atd. Tyto informace jsou pro naše použití nadbytečné vzhledem k tomu, že rozměry jsou pevně dané a paletu nepoužíváme. Její standardní velikost je 54 B, ale v programu je možné tuto pozici posouvat pro případ některých atypických, popř. již upravených bitmap.

Dalším požadovaným vstupem jsou zeměpisné souřadnice okrajových bodů zpracovávané mapy a jejich směry. Tyto souřadnice je možné zadávat přímo ve stupních, ale většina zdrojů GPS dat je uvádí ve formátu *stupně-minuty-sekundy*. Proto tato aplikace umožňuje přepočet z takto zadaných souřadnic na aplikací požadovaný formát a dosazení výsledné hodnoty do odpovídajícího pole.

Výstup aplikace (soubor zvolený v poli *Cíl*) je požadovaný mapový soubor, připravený na nahrání do FITkitu.

7 Použití aplikace

Poslední kapitola se zabývá použitím aplikace a její obsluhou.

7.1 Postup

Nejprve musíme zapnout samotný FITkit. To provedeme pouhým zapojením napájecího napětí buď přímo přes USB konektor (JP0), který bude sloužit i pro komunikaci s terminálem v PC, nebo přes napájecí konektor (JP8). Na sériový port RS232 (JP5) připojíme GPS modul a na VGA výstup (JP2) nějaké zobrazovací zařízení (monitor či projektor).

Dalším krokem je připojení se ke kitu přes libovolný terminál a nahrání požadované mapy. To provedeme příkazem *flash w map* a následným transferem souboru (postup tohoto transferu je detailně popsán v dokumentaci FITkitu [8]). Vykreslení mapy a načtení kotvících souřadnic vyvoláme příkazem *load*. Po překreslení celé mapy je aplikace plně funkční a blikáním černé tečky zobrazuje aktuální pozici na mapě.

Následující tabulka obsahuje kompletní seznam příkazů:

HELP	Vypíše do terminálu nápovědu a seznam příkazů.
TIME	Přepne LCD display na zobrazení času, vypíše akt. čas do terminálu .
FIX	Přepne LCD display na zobrazení polohy, vypíše ji do terminálu.
ALT	Přepne LCD display na zobrazení výšky, vypíše ji do terminálu.
QUALITY	Přepne LCD display na zobrazení kvality signálu, vypíše ji do terminálu.
FLASH W MAP	Nahrání souboru s mapou z PC do FLASH paměti FITkitu.
LOAD	Vykreslení mapy (nahrání z FLASH do SDRAM), ukotvení.
PT x y	Vytvoření černého bodu na souřadnicích [X,Y].
CL x y	Obnovení bodu na souřadnicích [X,Y] podle původní bitmapy.

Tabulka 7.1 – Seznam příkazů

Závěr

Ve výsledku tedy máme aplikaci, která převádí data přijatá z GPS modulu do vizuální podoby jako mapu s průběžnou indikací aktuální pozice, podobně, jako to dělají například různé navigační systémy. Tyto mapy lze vytvářet za pomoci k tomu vytvořené doplňující aplikace, což zajišťuje jistou univerzálnost výsledného projektu.

Do budoucna je možné ho rozšířit teoreticky hned v několika různých směrech, např.:

- Zvýšení kvality zobrazované mapy – zavedení mapy s větším rozlišením a barevnou hloubkou, což ovšem předpokládá zvětšení kapacity paměti FITkitu. K tomu by mohla být využita paměť DRAM.
- Zaznamenávání aktuální pozice v nějakém časovém úseku – tyto informace následně zpracovávat a různě vyhodnocovat, např. pro zobrazení ujeté trasy na mapě či zprostředkování různých statistických informací (průměrná rychlost, výškový profil trasy atd.).
- Vytvoření jednoduché navigace – prezentování různých navigačních algoritmů. To by ovšem vyžadovalo použití vektorové mapy místo bitmapové, popř. aspoň ukotvení několika významných bodů na stávající mapu a ohodnocení jejich vzájemného spojení. Toto řešení by bylo mnohem jednodušší, výsledek by ovšem byl o to méně použitelný pro reálnou navigaci (prakticky nepoužitelný v malém měřítku, ve velkém měřítku by mohl najít uplatnění např. na vyhodnocení cesty mezi městy – dálnice je lepší než okresní cesty apod.).

Během práce jsem se detailně seznámil s celým systémem GPS navigace, jak z teoretického, tak z praktického hlediska. Práce s FITkitem mi prohloubila znalosti v oblasti mikrokontrolerů a vestavěných systémů a pomohla pochopit některé dosud pouze teoreticky řešené zákonitosti a principy, které se jich týkají. Prakticky jsem zde také uplatnil znalosti získané v průběhu bakalářského studia, převážně z předmětů INP (návrh počítačových systémů), IMP (mikroprocesorové a vestavěné systémy) a IPZ (periferní zařízení), které se úzce věnují problematice popisu hardware komponent v jazyce VHDL, jejich spolupráce se softwarem a komunikačního rozhraní a protokolů, které k této spolupráci využívají.

Literatura

- [1] WIKIPEDIA CONTRIBUTORS. *Global Positioning System* [online]. Wikipedia, The Free Encyclopedia, [2007-], 17 April 2008 12:01 UTC [cit. 2008-04-20]. Dostupný z WWW: http://en.wikipedia.org/w/index.php?title=Global_Positioning_System&oldid=206224592
- [2] DEPRIEST, Dale. NMEA data [online]. [2007-] [cit. 2008-04-20]. Dostupný z WWW: <http://www.gpsinformation.org/dale/nmea.htm>
- [3] FUČÍK, Otto. *FITkit : Úvod* [online]. Fakulta informačních technologií VUT Brno, 2006-2008 [cit. 2008-04-20]. Dostupný z WWW: <http://merlin.fit.vutbr.cz/FITkit/>
- [4] VAŠÍČEK, Zdeněk. *FITkit : Firmware : Propojovací systém FITkitu* [online]. Fakulta informačních technologií VUT Brno, 2006-2008 [cit. 2008-04-20]. Dostupný z WWW: <http://merlin.fit.vutbr.cz/FITkit/docs/firmware/spifitkit.html>
- [5] RS232 [online]. HW server, 2003 [cit. 2008-04-20]. Dostupný z WWW: <http://rs232.hw.cz/>
- [6] ČAPKA, Ladislav. *FITkit : Firmware : VGA rozhraní* [online]. Fakulta informačních technologií VUT Brno, 2006-2008 [cit. 2008-04-20]. Dostupný z WWW: <http://merlin.fit.vutbr.cz/FITkit/docs/firmware/20060931vga.html>
- [7] WIKIPEDIA CONTRIBUTORS. *8-bit color* [online]. Wikipedia, The Free Encyclopedia, [2007-], 1 November 2007 12:01 UTC [cit. 2008-04-20]. Dostupný z WWW: http://en.wikipedia.org/wiki/8-bit_color
- [8] MARKOVIČ, Ján. *FITkit : Návod : Tutoriál* [online]. Fakulta informačních technologií VUT Brno, 2006-2008 [cit. 2008-04-20]. Dostupný z WWW: <http://merlin.fit.vutbr.cz/FITkit/docs/navody/20060210a.html>
- [9] VAN DER SPIEGEL, Jan. *VHDL Tutorial* [online]. 2001 [cit. 2008-04-20]. Dostupný z WWW: http://www.seas.upenn.edu/~ese201/vhdl/vhdl_primer.html#_Toc526061340
- [10] PETZOLD, Charles. Programování Microsoft Windows Forms v jazyce C# / Vyd. 1. Brno : Computer Press, 2006. 356 s. + ISBN 80-251-1058-3
- [11] Microsoft. *MSDN : Visual C#* [online]. 2008 [cit. 2008-04-20]. Dostupný z WWW: <http://msdn2.microsoft.com/en-us/library/kx37x362.aspx>
- [12] *Mapy.cz* [online]. PLAN Studio, 2005-2007 [cit. 2008-04-20]. Dostupný z WWW: <http://www.mapy.cz/>

Seznam příloh

Příloha 1. Ukázka aplikace, mapa ČR

Příloha 2. Ukázka aplikace, mapa Brna

Příloha 3. Ukázka aplikace, display a zapojení

Příloha 4. CD se zdrojovými texty



Příloha 1 – Ukázka aplikace, ČR



Příloha 2 – Ukázka aplikace, Brno



Příloha 3 – Ukázka aplikace, display a zapojení