Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

DIPLOMOVÁ PRÁCE

Zpracování dat z GPS modulu

PLZEŇ, 2013 JAN BOHATÝ

PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 1.1.2013 .....

vlastnoruční podpis

Poděkování

Chtěl bych poděkovat firmě VECTRA Electronic s.r.o. zejména konzultantovi panu Jiřímu Knížeti za poskytnutí informací a veškerého zázemí pro vývoj a Ing. Tomášovi Rybovi za vedení této diplomové práce.

Abstrakt

Tématem této práce je zpracování dat z GPS modulu. Zpracování dat je provedeno pomocí mikrokontroléru. Data jsou zobrazována na displeji. Pro displej je vytvořeno menu. Ovládání zařízení je pomocí tlačítek. Data je možné přenášet a zobrazovat na počítači.

Klíčová slova: GPS, LCD, mikrokontrolér, C8051, NMEA

Abstract

The subject of this thesis is processing data from a GPS module. Data are processed by a microcontroller and displayed on a LCD. A menu is created for the display. The device is controlled by buttons. Data can be transferred and shown on a computer.

Keywords: GPS, LCD, microcontroller, C8051, NMEA

Obsah

Zpracování dat z GPS modulu

1 Úvod

Tato práce vznikla ve spolupráci s firmou VECTRA Electronic s.r.o., zabývající se vývojem elektroniky, elektrotechnických a elektronických zařízení na zakázku.

Jedná se o základní část projektu z oblasti přesného zemědělství, který se zabývá řízením dávkování hnojiva pro postřikovače. Dávkování postřiku je regulováno podle rychlosti pojezdu. Zařízení přijímá data z družic GPS a hodnotu rychlosti využívá k nastavení výkonu čerpadla zajišťujícího rozprašování roztoku. Tímto zařízením je tak zajištěno přesnější dávkování a úspora nákladů na hnojení.

Cílem této práce je:

1. Zpracovat data z GPS modulu pomocí mikrokontroléru,
2. zobrazit data na displeji,
3. vytvořit menu pro displej,
4. přenést data do PC,
5. zobrazit data z GPS v PC.

Základem celého zařízení je mikroprocesor C8051F020, který zpracovává data z GPS modulu, odesílá je do počítače a na displej. Přenos dat do počítače zajišťuje převodník USB – UART. Pro příjem a zobrazení hodnot v počítači je vytvořen program (GPSReceiver.exe). Zobrazení dat na podsvětleném LCD je ovládáno pomocí dvou tlačítek, kterými se vybírá z menu hodnota pro zobrazení. Přenos dat přes USB je indikován LED (příjem a vysílání). Příjem dat z GPS je indikován zelenou LED, neplatnost přijímaných GPS dat je signalizována červenou LED.

Veškeré programování (mikroprocesoru a softwaru pro počítač) je provedeno v jazyce C.

2 Teoretické informace

2.1 GPS

GPS (Global Positioning System) je vojenský globální družicový polohový systém provozovaný Ministerstvem obrany USA. Tento systém umožňuje určení polohy a přesného času. Určování polohy je založeno na principu měření doby šíření signálu z družice k přijímači. Tímto způsobem je zjištěna vzdálenost přijímače od družice. Poloha přijímače je dána vzdáleností od více družic. Dochází tedy k průniku signálů z odlišných družic. Jedná se o princip jednosměrného dálkoměru. Tento radionavigační systém, označovaný též jako NAVSTAR (Navigation System using Time and Ranging), umožňuje určit polohu přijímače v trojrozměrných souřadnicích a jeho rychlost v reálném čase. Lze také získat potřebnou informaci o čase UTC. Systém umožňuje v reálném čase okamžité a přesné určení polohy (řádově 10m) libovolného počtu i rychle se pohybujících předmětů.

V systému GPS je velmi důležitá časová synchronizace časových stupnic, které se používají při jednostranném určování vzdálenosti. Časové stupnice na družici a v přijímači nemohou být přesně synchronizovány, tím vzniká v určení vzdálenosti systematická chyba. Měřená vzdálenost se označuje jako pseudovzdálenost. Doba šíření družicového signálu mezi vysílačem na družici a anténou přijímače se určuje s využitím frekvenčních standardů časových stupnic na družici a v přijímači. Obě časové stupnice jsou porovnávány vzhledem k systémovému času, označovaném jako GPST. Je třeba si uvědomit, že chybě 1 ns v šíření signálu odpovídá chyba 0,3 m v měřené vzdálenosti. Časové stupnice na družicích mohou být synchronizovány s časovou stupnicí GPST pomocí koeficientů polynomu, které jsou v navigační zprávě. Polynom určuje časové posunutí družicových hodin ke vztaženému času.

Budování systému započalo v roce 1973, v roce 1978 se připojilo ještě devět členských států NATO. V roce 1983 bylo rozhodnuto o zpřístupnění systému i pro civilní využití. Dalším zlomovým okamžikem byl rok 2000, kdy došlo k odstranění uměle generované nepřesnosti signálu pro civilní využití.

Jako vojenský systém je odolný proti rušení a ve svých špičkových možnostech je nedostupný neoprávněným uživatelům.

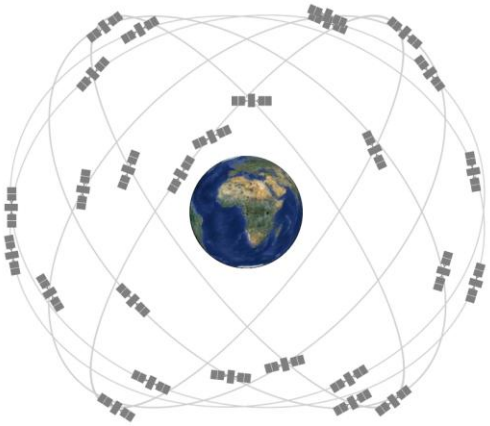
2.1.1 Struktura systému

Kosmický segment

Kosmický segment zahrnuje 24 družic, které obíhají Zemi ve výšce 20 200 km a jsou rozmístěny do šesti rovnoměrně rozmístěných drah (viz. Obr. 1). Každý satelit obkrouží Zemi dvakrát denně. Družice vysílají radiový signál směrem k uživateli. Každá družice vysílá informace o své poloze a přibližné poloze ostatních družic. Přesný čas a přesné kmitočty vysílaných frekvencí zajišťují cesiové nebo vodíkové oscilátory.

Používáno je několik generací satelitů:

- **blok IIA** – upgradovaná verze satelitů bloku II vypuštěných v letech 1989-1990, série IAA vypuštěna v letech 1990 – 1997, celkem 19 satelitů
- **blok IIR** – určeny pro nahrazení série II/IIA, vypuštěny v letech 1997 – 2004, celkem 13 satelitů
- **blok IIR(M)** – upgradovaná verze série IIR vypuštěna v letech 2005 – 2009, celkem 8 satelitů
- **blok IIF** – slouží k rozšíření série IIR(M), první vypuštění proběhlo v roce 2010, celkem 12 satelitů, jsou vybaveny kombinací rubidiových a cesiových atomových hodin udržující přesnost 8 biliontin vteřiny za den
- **blok III** – ve vývoji, celkem 8 satelitů



Obr. 1: *Oběžné dráhy družic (převzato z [10])*

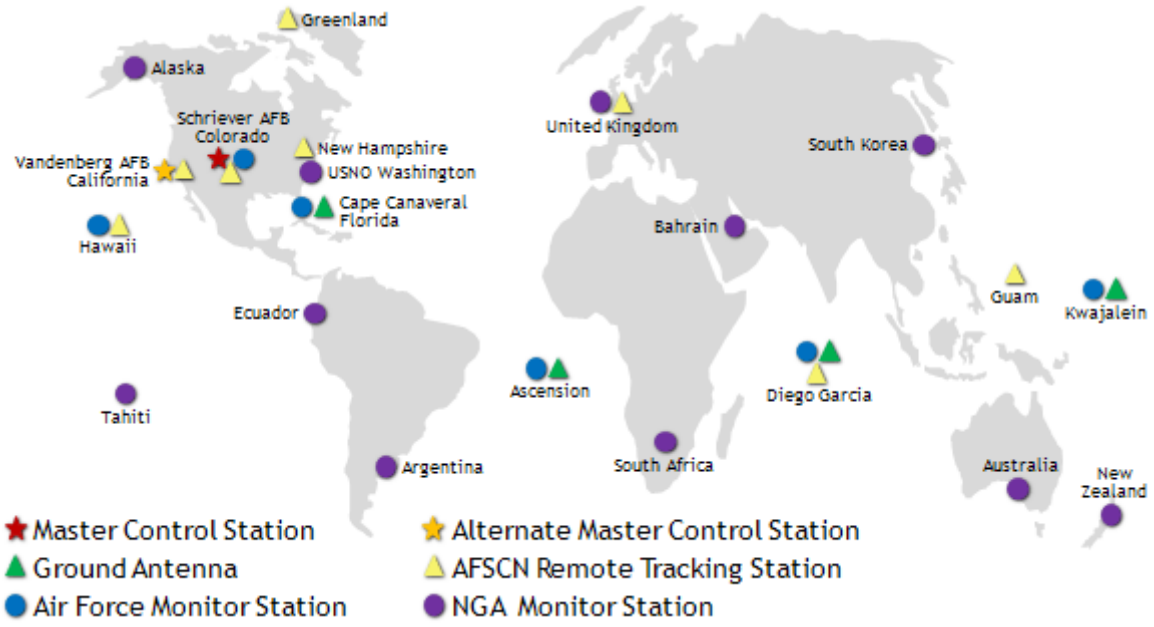
Řídicí segment

Řídicí segment je složen ze sítě pozemních stanic rozmístěných po celém světě (viz. Obr. 2). Tento segment monitoruje a řídí kosmický segment (monitoruje signál satelitů, provádí analýzy a zasílá data a příkazy družicím).

Součástí je hlavní řídicí stanice (Master Control Station), jedna náhradní řídicí stanice (Alternate Master Control Station), 12 povelových stanic (Ground Antenna, AFSCN Remote Tracking Station) a 16 monitorovacích míst (Air Force Monitor Station, NGA Monitor Station).

Hlavní řídicí stanice uchovává časový systém, ve kterém pracuje celý systém GPS. Pro výpočet dráhových parametrů se používá týdenní pozorování monitorovacích stanic. Pomocí Kalmanova filtru[1] jsou odvozovány parametry drah družic, určovány korekce družicových hodin a parametry ionosférického modelu. Z odvozených parametrů se předpovídají dráhové parametry platné na 26 hodin. V intervalu osmi hodin se předávají z pozemních řídicích stanic pomocí antén povely pro řízení provozního režimu družic a korekce drah družic do procesorů na družicích. Monitorovací stanice sledují pohyb satelitů a zasílají tato informace řídicí stanici.

Pozemní antény slouží ke komunikaci s družicemi. Umožňují odesílání a přijímání signálu.



Obr. 2: *Mapa rozmístění stanic řídicího segmentu (převzato z [10])*

[1] Podrobnější informace lze získat na http://www.mendeley.com/catalog/introduction-kalman-filter/

Uživatelský segment

Uživatelé přijímají signál z družic, které jsou nad obzorem, pomocí pasivních GPS přijímačů (jednosměrná komunikace od družice k uživateli).

Přijímač GPS je tvořen anténou, radiofrekvenční jednotkou, mikroprocesorem, komunikační jednotkou paměti a zdrojem napětí. Signály GPS jsou slabé, z toho důvodu je anténa doplněna předzesilovačem. Radiofrekvenční jednotka zpracovává přijaté signály (na jedné frekvenci pro civilní využití, na dvou frekvencích pro vojenské účely). Základem je oscilátor, který generuje referenční frekvenci. Tato frekvence je použita k vytvoření referenčního signálu, jenž se porovnává se signálem přijímaným.

Důležitou částí přijímače jsou tzv. přijímací kanály, které umožňují příjem a odlišení signálů z různých družic. Přijatý signál přijímač zpravidla rozlišuje podle kódu, který je pro každou družici charakteristický.

Jsou 3 druhy přijímacích kanálů:

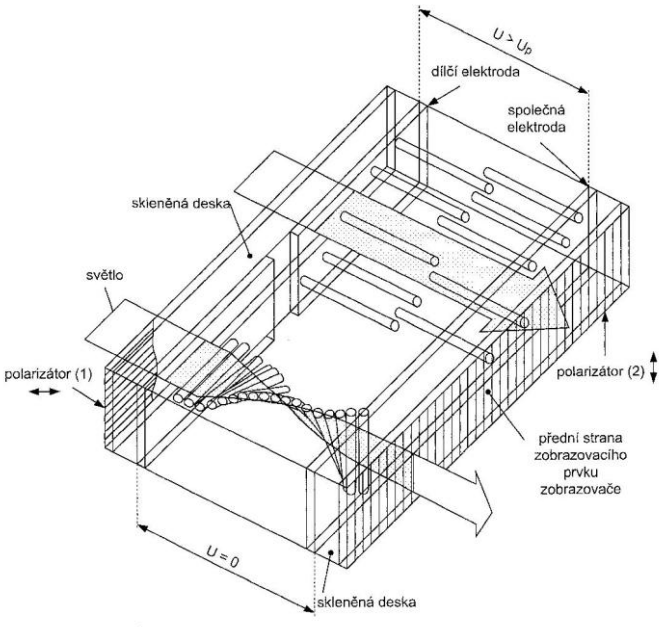
- kódový korelační kanál – umožňuje měřit pseudovzdálenost ke družici a umožňuje dekódování navigační zprávy
- kódový fázový kanál – dovoluje měřit zlomky fází nosných frekvencí L1 a L2 zvolené družice a umožňuje dekódování navigační zprávy
- násobný korelační kanál – umožňuje měřit zlomky fází na nosných frekvencích současně z několika družic, příjem signálů se neustále přepíná z jedné družice na druhou, rychlost přepínání umožňuje jejich využití pro kinematické měření a pro měření v reálném čase

Přesnost GPS signálu pro civilní využití (SPS) je stejná jako přesnost signálu pro vojenské využití (PPS). Nicméně SPS využívá pouze jednu frekvenci (nosná frekvence L1 = 1575,42 MHz) na rozdíl od PPS, kde jsou využity frekvence dvě (L1 a L2 = 1227,6 MHz). Pro vojenské účely lze tedy využít ionosférickou korekci. Jedná se o techniku, která omezuje degradaci signálu způsobenou zemskou atmosférou. Využívá se rozdílu zpoždění těchto dvou frekvencí. Z toho tedy plyne, že PPS poskytuje větší přesnost oproti SPS. Podrobnější informace lze získat v [10] a [11].

2.2 LCD

LCD (Liquid Crystal Display) je displej z tekutých krystalů. Kapalné krystaly jsou zvláštní látky, které si v jistém rozsahu teplot udržují typické vlastnosti jak pro pevnou látku, tak pro kapalinu (látka, která stojí na pomezí pevného a tekutého stavu). Mají krystalickou strukturu a přitom jsou ještě tekuté. Samotný tekutý krystal vzniká smícháním několika základních látek, jako jsou například bifenyly či dioxiny, s dalšími. To je důležité proto, aby výsledný produkt získal všechny potřebné vlastnosti, ke kterým patří např. elasticita, viskozita či správný index odrazu. Molekuly tekutého krystalu tvoří zvláštní podlouhlé útvary (mají tyčový tvar), které se jakoby vznášejí v tekutině a mohou být elektricky polarizovány.

Toto zobrazovací zařízení je rozděleno na jednotlivé pixely (barevné nebo monochromatické buňky) uspořádané do řádkových a sloupcových struktur se systémem adresovatelného buzení. Základem zobrazovače jsou dvě skleněné destičky. Mezi těmito destičkami je roztok kapalných krystalů. Z vnitřní strany je na skleněných deskách nanesena rýhovaná vrstva oxidu křemičitého. Rýhování je na obou deskách na sebe vzájemně kolmé, a to způsobuje, že se krajní molekuly roztoku orientují ve směru rýh a molekuly mezi nimi vlivem mezimolekulárních sil utvoří šroubovici (Schadt-Helfrichův jev). Na vnitřní strany skleněných desek jsou napařeny průhledné vodivé elektrody pro přívod napětí. Povrchy obou vnějších stran jsou dále opatřeny polarizačními filtry s navzájem kolmou polarizací, která odpovídá směru drážek na vnitřní straně desek (viz. Obr. 3).



Obr. 3: *Struktura LCD (převzato z [12])*

Tekuté krystaly samy o sobě světlo nevydávají, jejich úkolem je správně světlo zpracovat (blokovat nebo propouštět). Existují 3 typy LCD: reflexní, transmisivní a transreflexní. U transmisivních je zdrojem světla panel (např. elektroluminiscenční výbojka), u reflexních je pod spodní vrstvou displeje navíc reflexní vrstva odrážející světlo a transreflexní je kombinace obou zmíněných. Světlo je zpracováno dvěma polarizačními filtry a vrstvou tekutých krystalů. V klidovém stavu (bez připojení vnějšího zdroje) prochází světlo ze zadního světelného zdroje polarizátorem. Průchodem tímto polarizátorem získáme světlo polarizované v horizontální rovině. To dále prochází tekutým krystalem. Protože jsou ve vrstvě tekutého krystalu jednotlivé molekuly pootočeny, je průchodem světla změněna i jeho polarizace z horizontální na vertikální. Světlo s touto vertikální polarizací je pak propuštěno i druhým polarizátorem (který má vertikální polarizaci) a zobrazovač svítí. V případě, že připojíme na elektrody tekutého krystalu zdroj napětí, změní se jeho vnitřní struktura. Molekuly krystalu již nejsou vzájemně pootočeny, ale napřímeny. Světlo procházející vrstvou tekutého krystalu tedy nemůže změnit svou polarizaci z horizontální na vertikální a je tak zablokováno na polarizátoru, který propouští pouze světlo s polarizací vertikální. Zobrazovač tedy zůstává tmavý, neboť světlo ze zadního zdroje neprojde. Postavením molekul tekutého krystalu se tedy ovládá průchod světla. V praxi však nestačí pouze mezní stavy (světlo projde/neprojde), nutností je také regulace množství propuštěného světla, resp. změna jasu. Toho lze docílit velikostí napětí připojeného k elektrodám. Podrobnější informace o LCD lze nalézt v [12].

2.3 Mikroprocesor

Mikroprocesor je složitý logický obvod vykonávající sled aritmetických a logických operací podle zadaného programu a tak realizuje námi požadovanou funkci. Program je uložen v paměti ve formě instrukcí, které jsou postupně načítány a vykonávány. Mikroprocesor zpracovává data v paměti, řídí tok vstupních a výstupních dat a zajišťuje jejich zpracování.

Mikroprocesor je nejčastěji realizován v integrovaném obvodu zapouzdřeném do plastového (příp. keramického) pouzdra (viz. Obr. 4). Základem je křemíková destička, na které jsou vytvářeny jednotlivé elektronické obvody. První mikroprocesory byly vyvinuty kolem roku 1971.

Úkolem procesoru je provádění instrukcí. Instrukce jsou nejmenší jednotky, ze kterých je složen program. Základní části instrukce jsou:

- operační kód
- operační kód a adresa operandu
- operační kód a přímá data

Ve všech případech je první částí operační kód (instrukční kód), který přesně definuje další činnost při provádění instrukce.

Některé instrukce obsahují jen operační kód. V těchto případech jsou prováděny jen vnitřní operace procesoru (např. inkrementuj obsah vnitřního datového registru atp.).

V druhém případě obsahuje instrukce ještě adresu operandu. Operandy jsou uloženy v datové paměti. Příkladem mohou být instrukce typu „přesuň obsah vnitřního datového registru do datové paměti na adresu ...“ apod..

Ve třetím případě obsahuje instrukce ještě přímá data. Jedná se vždy o konstanty, které jsou součástí programu. Příkladem může být instrukce „naplň vnitřní datový registr číslem ...“.

Existují i složitější instrukce obsahující dvě nebo tři adresy, adresu i data apod.. Záleží vždy na instrukčním souboru procesoru. Instrukce mají tedy obecně různou délku.

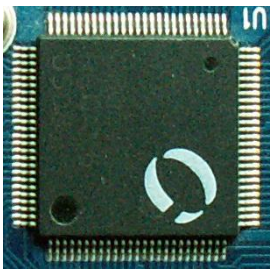
Celý průběh instrukce se skládá ze dvou fází – přečtení instrukce a provedení instrukce. Přečtení instrukce znamená vyvolání čtecích cyklů z paměti programu při postupně rostoucích adresách tak, aby se postupně přečetly všechny části instrukce. Počet čtecích cyklů odpovídá délce instrukce. Adresa instrukce odpovídá adrese, na které je uložen operační kód. Provedení instrukce znamená vykonání vnitřních operací v procesoru včetně získání potřebných operandů z datové paměti a následné uložení výsledků do datové paměti.

Z výše uvedeného vyplývají některé potřebné obvody v procesoru. Velmi důležitý je čítač instrukcí, který slouží k adresování paměti programu. Čítač je postupně inkrementován a program je tak postupně čten.

Další důležitou částí je instrukční registr, do kterého je uložen operační kód. Na základě obsahu tohoto registru je dále řízen průběh provádění instrukce. Má-li instrukce adresovou část, je uložena do pomocného adresového registru, kde bude k dispozici během provádění instrukce. Tyto a mnohé další pomocné registry jsou uživateli zpravidla nedostupné a neviditelné. Vnitřní obvody si předávají data po vnitřní datové sběrnici. Operace s daty jsou prováděny v aritmeticko-logické jednotce. Výsledek ve tvaru binárního čísla je uložen do střadače.

Potřebné jsou také informace o nulovosti přenosu z nejvyššího bitu, přetečení rozsahu čísla atd.. Všechny tyto informace jsou jednobitové a jsou uloženy do registru příznaků. Pro krátkodobé uložení dat slouží několik datových registrů (registrů zápisníkové paměti). Mohou sloužit k adresaci datové paměti, k odpočítávání počtu opakování programových smyček atd.

Řízení vnitřních obvodů (nastavení cest dat, zápisy do registrů atd.) provádí řadič. Jedná se o velmi složitý sekvenční obvod, řešený buď jako pevně zapojená logika nebo jako mikroprogramový automat. Jednotlivé posloupnosti řídicích signálů, generované řadičem, jsou vyvolávány instrukčním dekodérem, který na základě obsahu instrukčního registru rozliší typ instrukce a tomu přiřadí patřičný počáteční stav sekvenčního obvodu řadiče. Řadič inkrementuje obsah čítače instrukcí tak, aby byly postupně čteny všechny části instrukce a aby byla na konci čtení instrukce k dispozici již adresa následující instrukce. Tak jsou čteny a zpracovávány instrukce s postupně rostoucí adresou v paměti programu. Některé instrukce jsou podmíněné – jsou provedeny jen při splnění jisté podmínky. Jako podmínky slouží stavy v registru příznaků. Příznakové bity jsou zavedeny do řadiče, kde modifikují průběh mikroprogramu.



Obr. 4: *Mikroprocesor (použitý v zařízení)*

Ukazatel zásobníkové paměti (Stack Pointer) slouží pro adresování zásobníku. Zásobník zaujímá část datové paměti. Při operacích se zásobníkem se v programu neudává adresa. Jedná se o paměť typu LIFO (Last In – First Out). Adresa je uložena v ukazateli zásobníkové paměti, který je řadičem inkrementován nebo dekrementován, takže zásobník při vkládání dat postupuje na jednu stranu a při vybírání dat na druhou stranu. Funkce zásobníku umožňuje vložení podprogramů do podprogramů v libovolném počtu úrovní, přitom jsou do zásobníku postupně ukládány návratové adresy a při návratech z podprogramů jsou postupně ze zásobníku vybírány.

Velmi důležitou součástí procesoru jsou obvody pro zpracování přerušení programu. Přerušení je vyvoláno vnitřními obvody procesoru při výjimečných stavech (např. chyba při

čtení instrukce atp.) nebo vnějšími signály. Přerušení vyvolá odskok na podprogram. Rozdíl mezi vyvoláním podprogramu instrukcí a vyvoláním podprogramu přerušením spočívá v tom, že k přerušení může dojít kdykoliv. Jedná se tedy o okamžitou změnu činnosti jako reakci na neočekávaný vnější podnět. Adresa návratu z podprogramu je v obou případech uchována v zásobníku. Přerušení může být maskovatelné nebo nemaskovatelné. Procesor je pak vybaven zvláštními vstupy pro maskovatelné přerušení a pro nemaskovatelné přerušení. U maskovatelných přerušení může být podnět blokován v závislosti na stavu vnitřního maskovacího klopného obvodu. Blokování lze programově nastavovat či nulovat, kromě toho se v některých stavech procesoru přerušení blokuje automaticky. Nemaskovatelné přerušení nelze blokovat a má přednost před přerušením maskovatelným.

Pro synchronizaci vnitřních obvodů procesoru slouží hodinové impulzy. Pro zvýšení výkonnosti je žádoucí pracovat s jejich co nejvyšším kmitočtem. Vždy je proto používán krystalový generátor (oscilátor), neboť přesně definovaný kmitočet umožňuje držet se horní meze rozsahu povoleného výrobcem.

Souhrn všech činností mikroprocesoru, potřebných pro provedení jedné instrukce, se nazývá instrukční cyklus, lze ho rozdělit ještě na dílčí úseky – strojové cykly. V každém strojovém cyklu se právě jednou provede čtení nebo zápis. Strojové cykly se skládají z taktů, daných periodou vnitřních hodinových impulzů procesoru. Podrobnější informace o mikroprocesorech lze nalézt v [13].

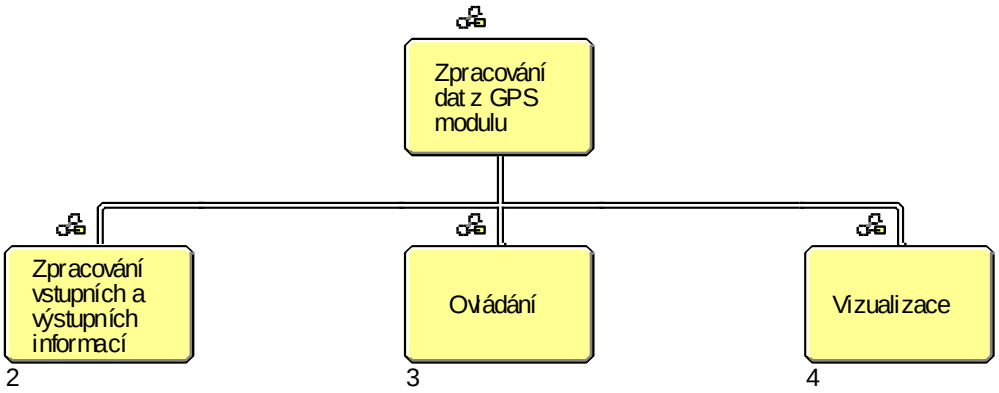
3 Návrh

3.1 Funkční struktura

Tento popis struktury systému zachycuje hierarchickou dekompozici systému na subsystémy a prvky pomocí stromových diagramů. Zachycuje hierarchickou nadřazenost a podřízenost prvků. Umožňuje tak získání přehledného pohledu na analyzovaný systém.

3.1.1 Zpracování dat z GPS modulu

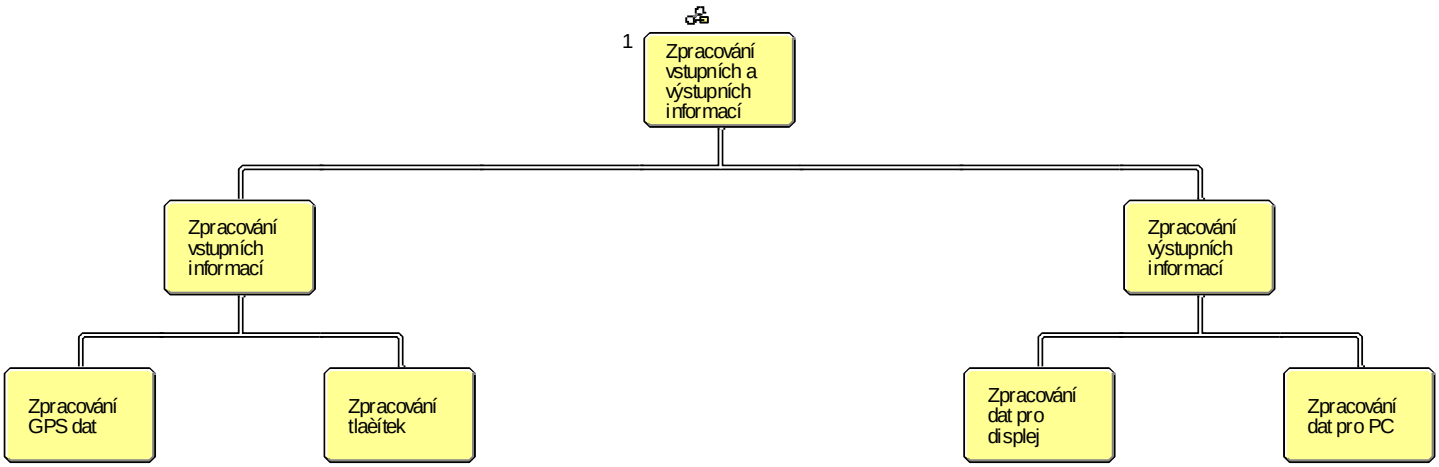
Struktura systému (viz. Obr. 5) zachycuje základní pohled na celý systém, ve kterém bude docházet ke zpracování informací, systém bude možné ovládat a informace zobrazovat. Zpracování vstupních a výstupních informací, ovládání a vizualizace jsou dekomponovány v následujících kapitolách.



Obr. 5: *Funkční struktura zpracování dat z GPS modulu*

3.1.2 Zpracování vstupních a výstupních veličin

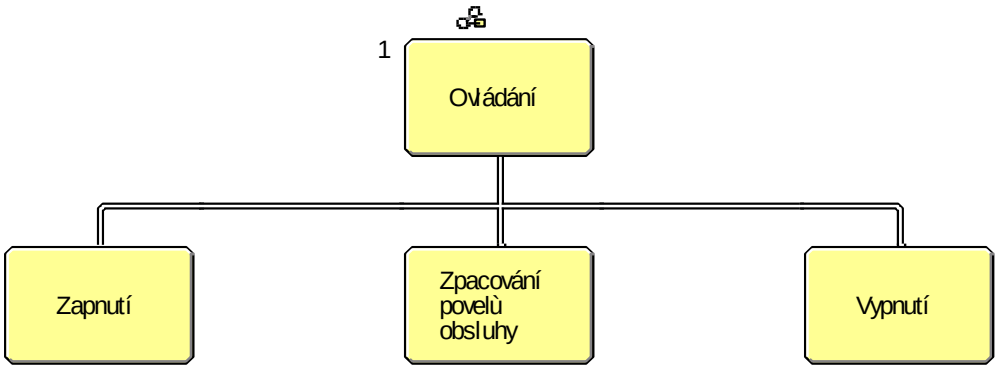
Dekompozice struktury zpracování vstupních a výstupních veličin zobrazuje, že vstupem budou data z GPS a tlačítek. Výstupní veličiny budou určeny pro displej a pro počítač (viz. Obr. 6).



Obr. 6: *Funkční struktura zpracování vstupních a výstupních veličin*

3.1.3 Ovládání

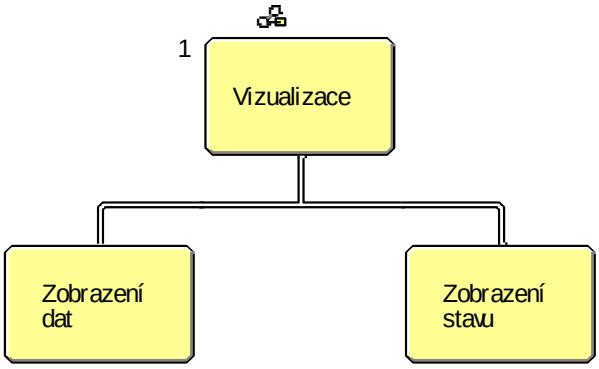
Tato funkční struktura zobrazuje způsoby ovládání systému. Povely obsluhy jsou definovány stiskem tlačítek nahoru a dolu (viz. Obr. 7).



Obr. 7: *Funkční struktura ovládání*

3.1.4 Vizualizace

Tato funkční struktura zobrazuje způsob vizualizace (viz. Obr. 8). Data budou zobrazována na LCD, další potřebné informace pomocí LED.



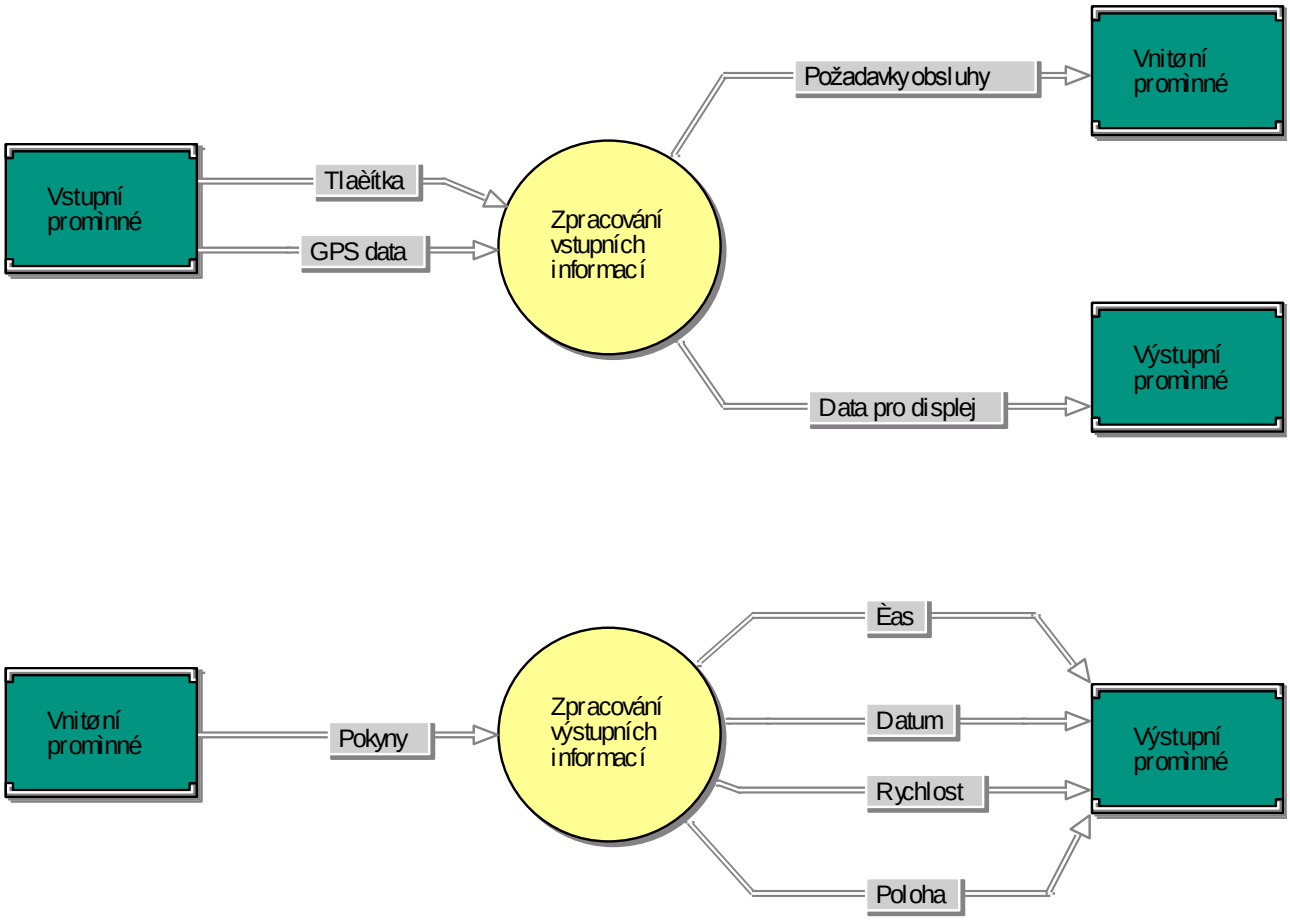
Obr. 8: *Funkční struktura vizualizace*

3.2 Informační toky

Metoda informačních toků zobrazuje hierarchickou dekompozici systému na subsystémy a zachycuje informační vazby mezi těmito prvky.

3.2.1 Zpracování vstupních a výstupních informací

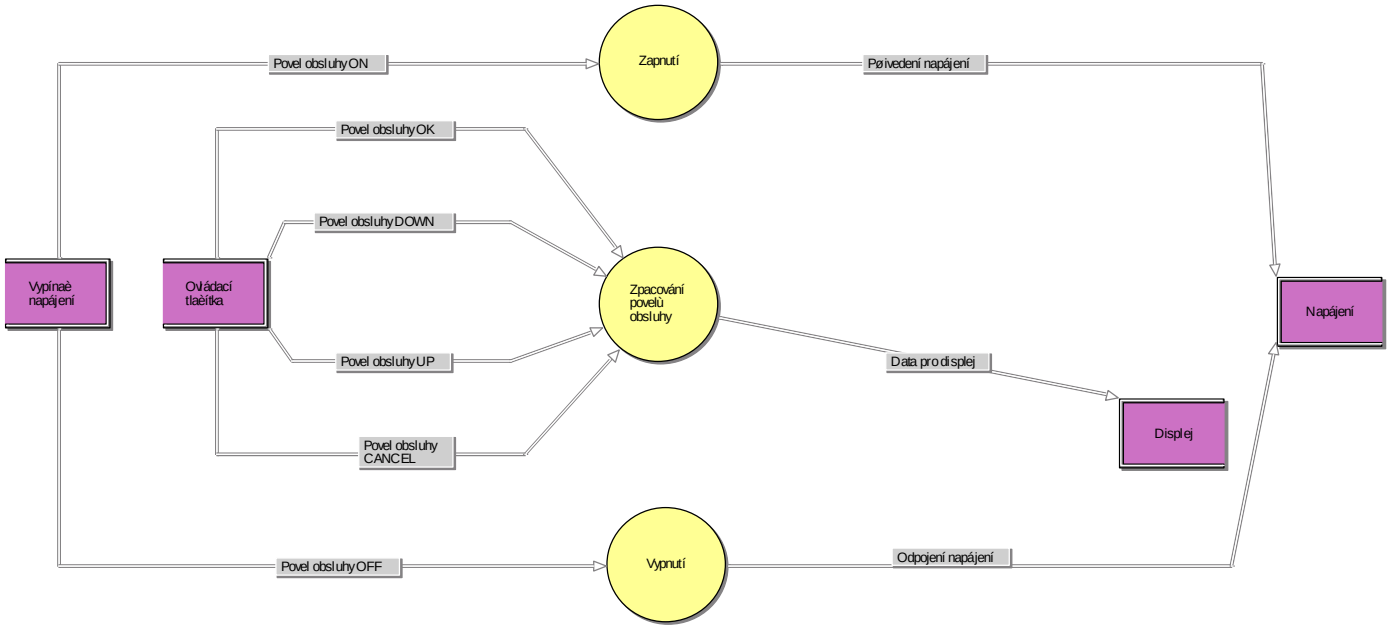
Tato část je věnována předzpracování dat z GPS modulu a informací z tlačítek. Do vnitřních proměnných se ukládají data z GPS v závislosti na požadavku obsluhy (viz. Obr. 9). K přenosu dat z GPS modulu je využíván protokol NMEA. Data jsou zpracována mikroprocesorem 8051.



Obr. 9: *Informační toky zpracování vstupních a výstupních informací*

3.2.2 Ovládání

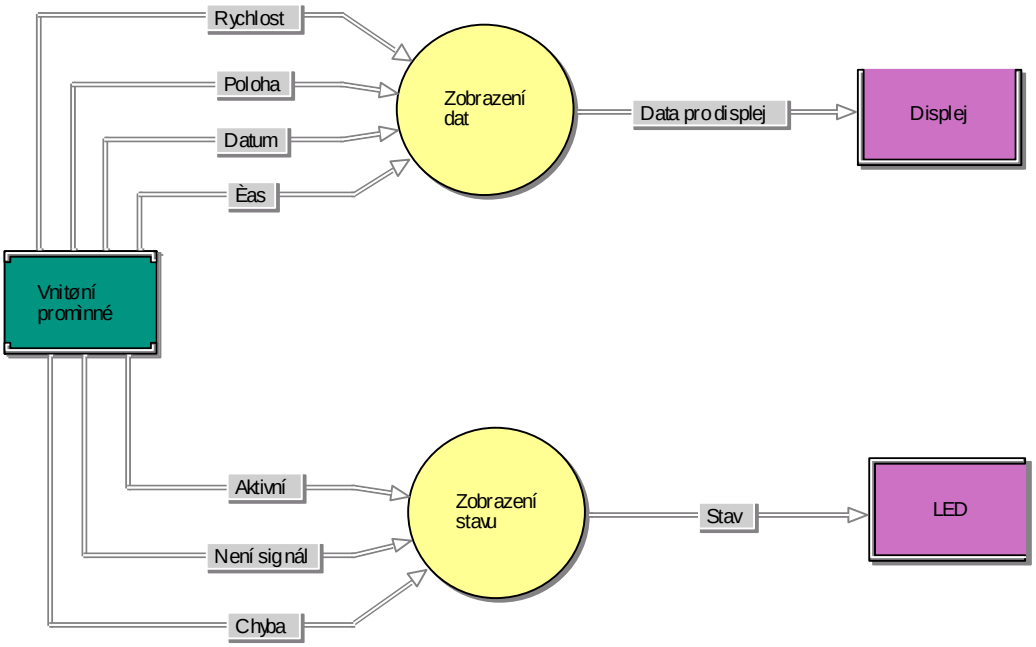
V této kapitole je navrženo ovládání systému (viz. Obr. 10). Zapínání a vypínání zařízení je realizováno pomocí vypínače (nikoliv tlačítka). Ovládání je pomocí tlačítek pro pohyb v nabídce na displeji (nahoru, dolu), potvrzovacího a stornovacího tlačítka. Tímto ovládáním je možné změnit zobrazovaná data na displeji.



Obr. 10: *Informační toky ovládání*

3.2.3 Vizualizace

V této kapitole je popsán návrh vizualizace dat získaných ze zařízení (viz. Obr. 11). Na displeji jsou zobrazována data (rychlost, poloha, datum, čas). Pomocí zelené LED je signalizován příjem signálu ze satelitů, červenou LED je signalizována chyba.



Obr. 11: *Informační toky vizualizace*

4 Vývojová sestava

4.1 Prototyp zařízení

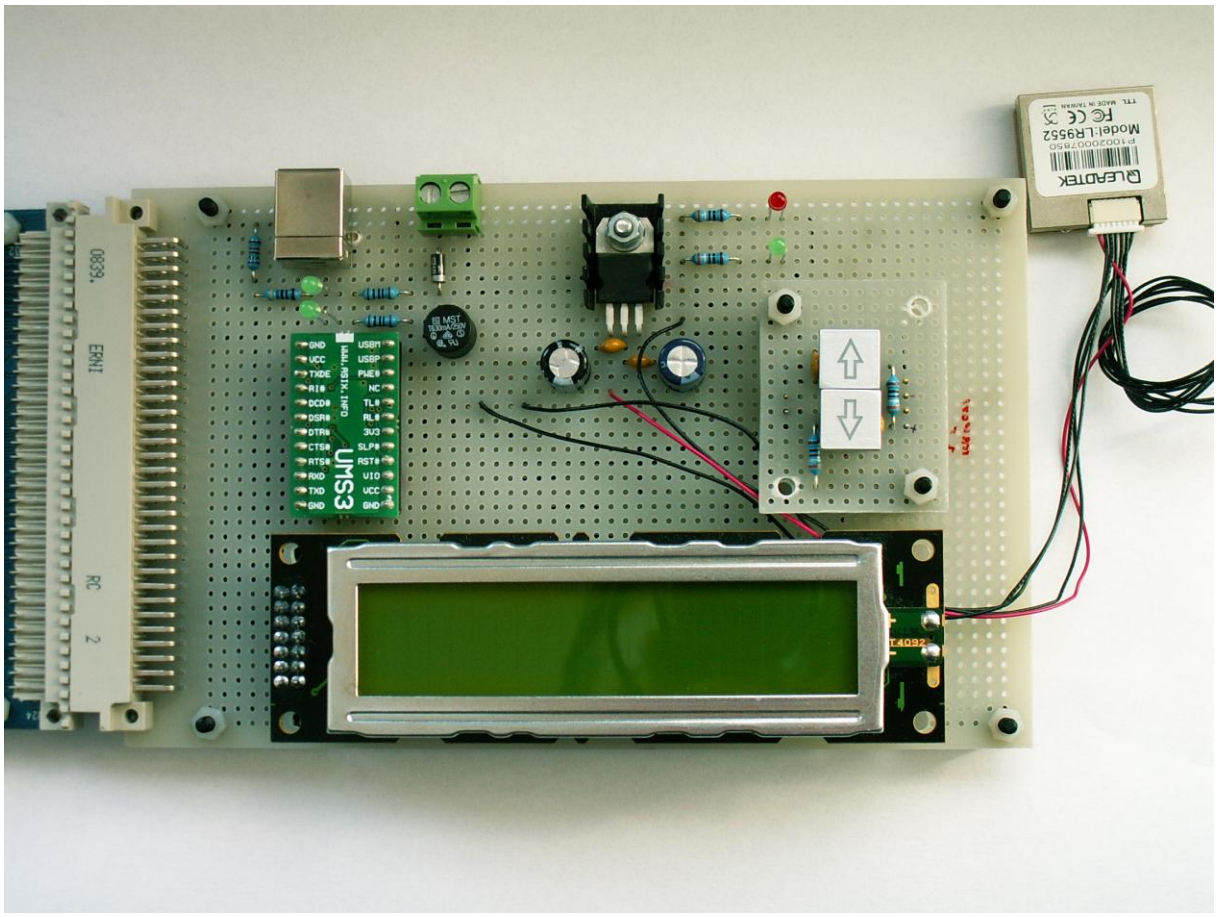
Pro zobrazování dat je použito dvouřádkového 20-ti místného LCD modulu Sharp. Zobrazení dat na displeji je ovládáno tlačítky. Mezi displej a procesor jsou vloženy budiče pro konverzi mezi 3,3V a 5V logikou.

GPS modul má architekturu SiRFStarIII. Jedná se o produkt firmy Leadtek typ LR9552. Ke komunikaci je využito protokolu NMEA. Příjem protokolu z GPS je indikován zelenou LED. Nedostupnost signálu z družic je indikována červenou LED.

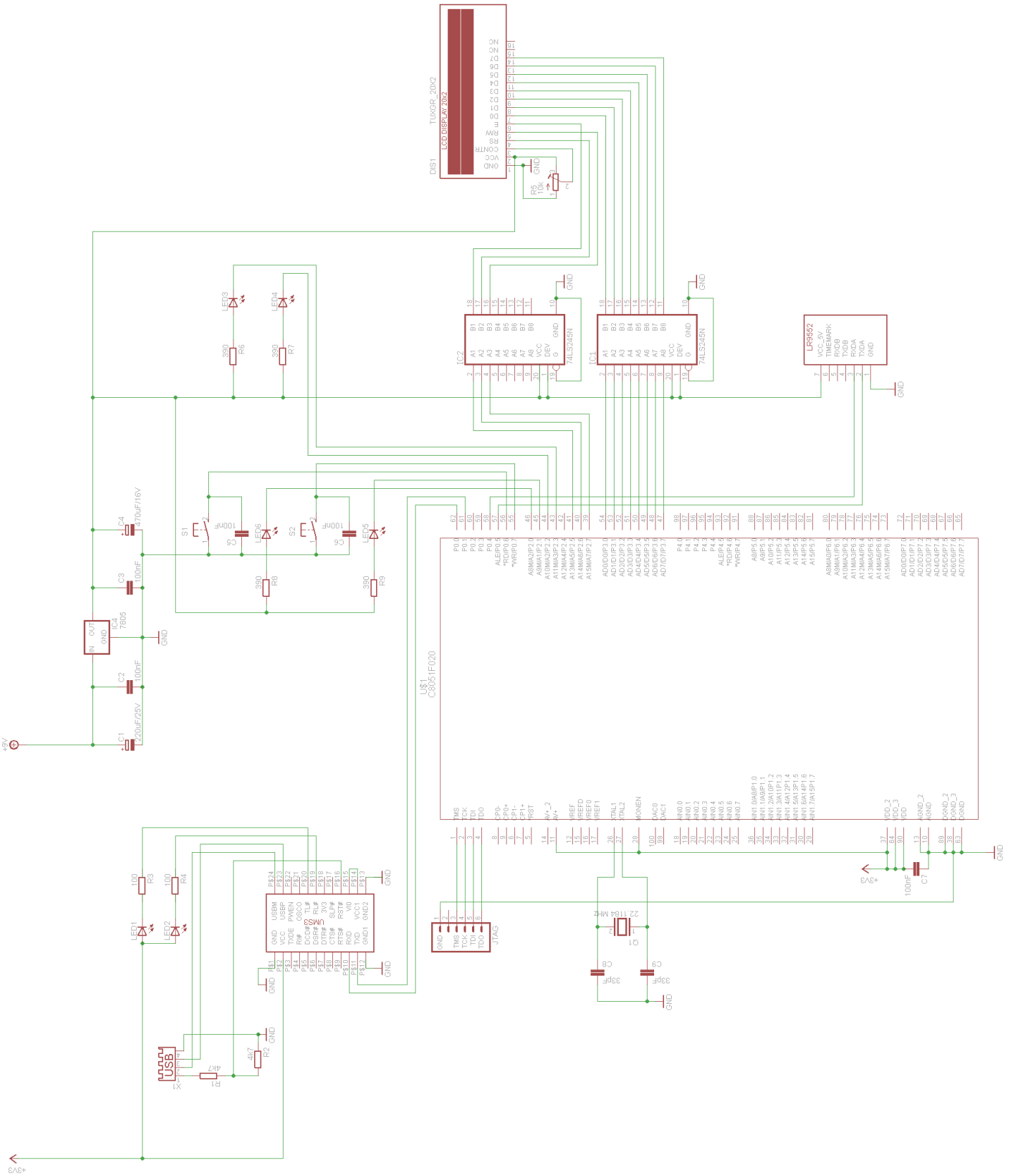
Pro komunikaci s PC je využit převodník USB – UART připojený na USB konektor. Příjem a vysílání dat přes USB je signalizováno dvěma LED.

Napájení zařízení je přes vývojový kit 9V, které je dále stabilizováno na 5V pro LCD a GPS. Pro mikroprocesor a převodník USB - UART je na vývojovém kitu stabilizováno na 3,3V.

Prototyp zařízení je sestaven na univerzálním plošném spoji (viz. Obr. 12). Schéma zapojení zařízení je na Obr. 13.



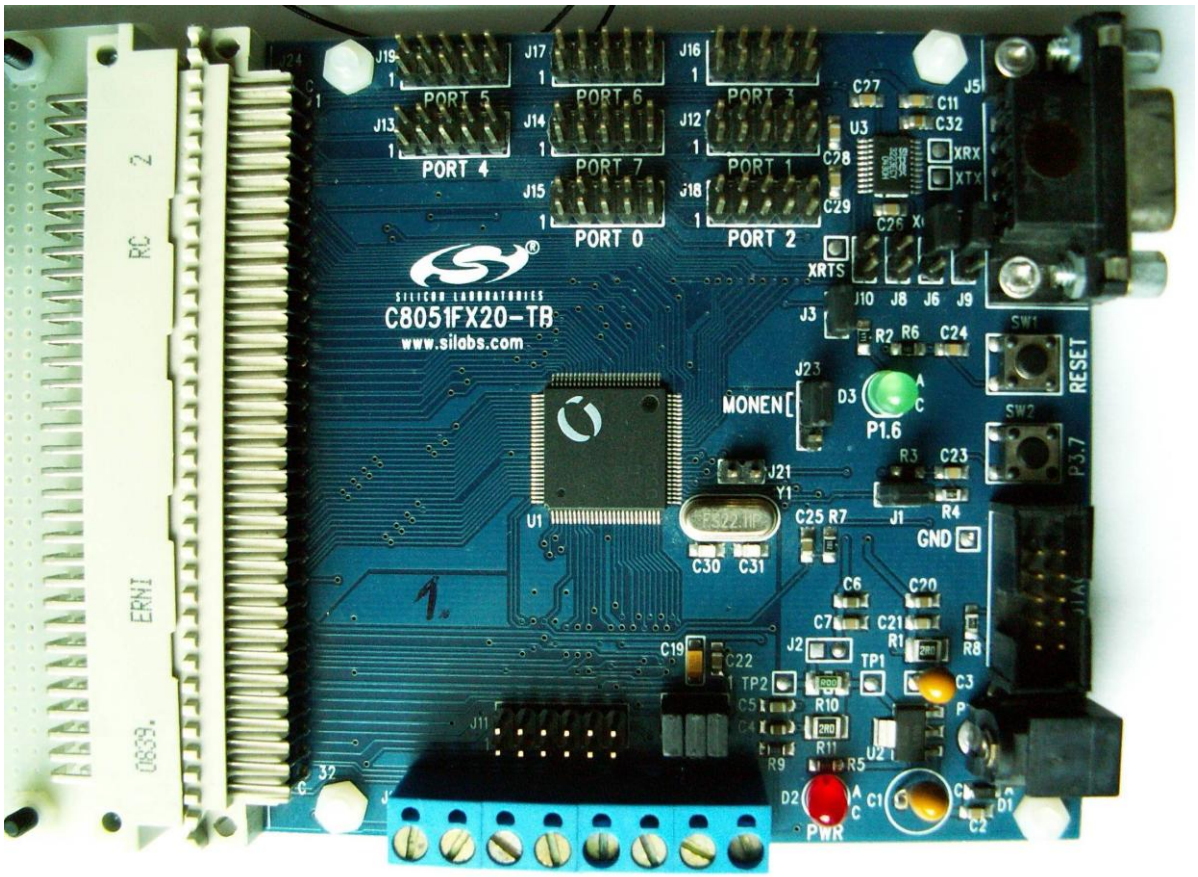
Obr. 12: *Pohled shora*



Obr. 13: *Schéma zapojení*

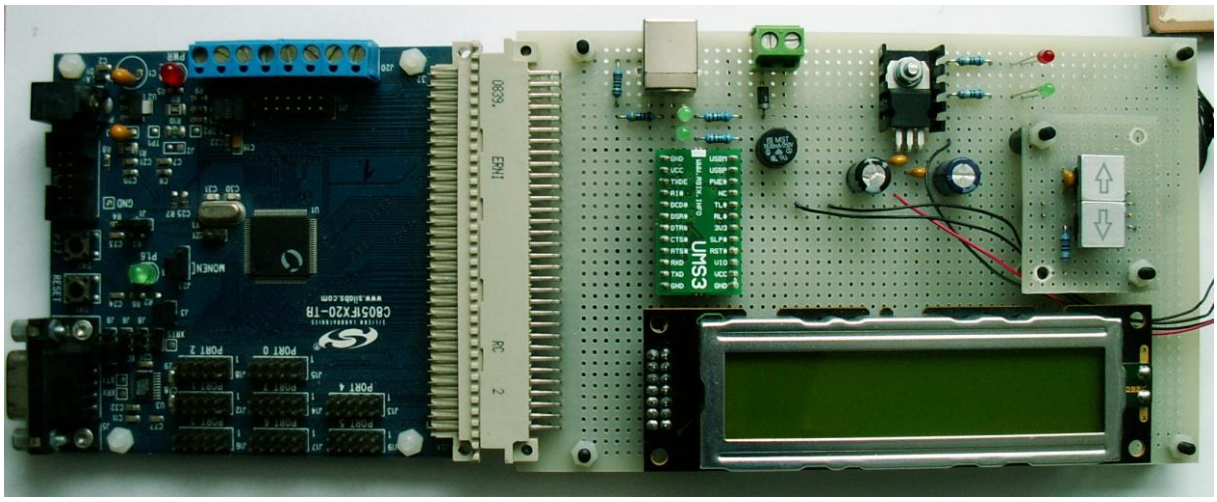
4.2 Vývojový kit

Základem zařízení je mikroprocesor C8051F020 (Silicon Laboratories). Pro vývoj bylo využito kitu, který je složen z debug adaptéru s USB připojením k PC a vývojové desky s procesorem C8051F020 (viz. Obr. 14).

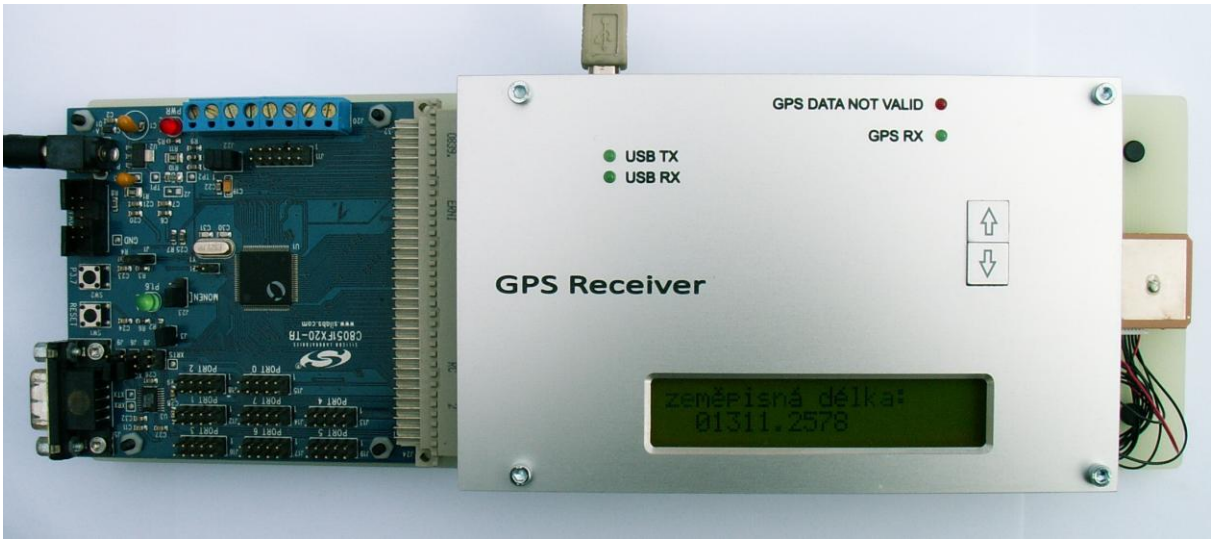


Obr. 14: *Vývojová deska s procesorem*

Obr. 15: *Debug adaptér*



Obr. 16: *Propojení kompletní sestavy zařízení s vývojovým kitem*



Obr. 17: *Konečná verze prototypu zařízení*

5 Použité komponenty

5.1 GPS modul

GPS modul Leadtek LR9952 je založen na architektuře SiRFStarIII, využíván je protokol NMEA-RMC (obsahuje informace: čas, datum, pozice, kurz a rychlost). Modul je přenastaven na zasílání pouze protokolu RMC. Komunikace je sériová, přenosová rychlost 4800 bps a napájení 5V. Přesnost pozice je 10m. Přesnost rychlosti je 0,1 m/s. Modul je malých rozměrů (25 x 23 x 6,9 mm) s integrovanou anténou. Studený start trvá 42s. Modul má integrovanou anténu.

5.2 LCD

Podsvícený displej Sharp LM20A212 je dvouřádkový dvacetimístný. Displej zobrazuje data dostupná z GPS modulu. Jedná se zejména o zobrazení polohy (viz. Obr. 19), data a času. Kromě zobrazení standardních znaků je možné zobrazovat i české znaky (viz. Obr. 18).

Česká lokalizace je doprogramována přímo pro účely tohoto zařízení pro zpracování dat z GPS modulu.

Obr. 18: *Zobrazení českých znaků na LCD*

Obr. 19: *Zobrazení GPS souřadnic na LCD (GPS modul bez signálu)*

5.3 Mikroprocesor

Základní částí zařízení je mikroprocesor od společnosti Silicon Laboratories C8051F020 se 64kB FLASH pamětí, pracující s napětím 3,3V. Výhodou tohoto procesoru je možnost krokování při ladění programu přes JTAG port. Pro potřeby tohoto zařízení je procesor zcela dostačující, co se týče rychlosti, paměti a vybavení.

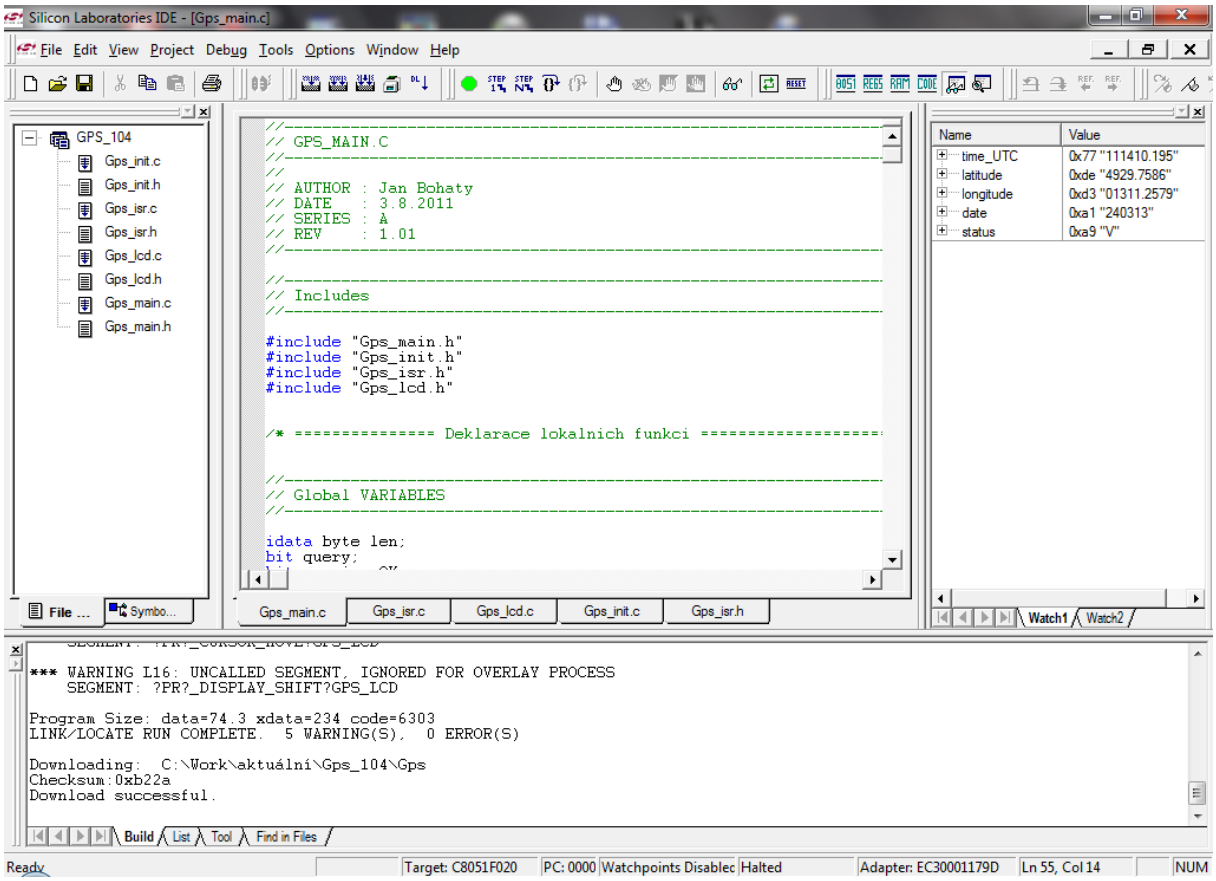
5.4 Převodník USB – UART

Modul UMS3 od českého výrobce Asix umožňuje připojení k PC pomocí USB. Jedná se o převodník mezi sběrnicí USB a UART. Je založen na integrovaném obvodu FT232RL od firmy FTDI Chip, která rovněž poskytuje ovladače pro různé druhy operačních systémů. Datové vstupy a výstupy pracují v rozsahu 1,8 – 5V. Podporuje USB 1.1 i USB 2.0. Modul umožňuje signalizaci pro příjem a vysílání dat mezi počítačem a FTDI čipem. Signalizace je zajištěna pomocí dvou LED.

6 Programování

K programování mikroprocesoru je použito softwaru od výrobce. Jedná se o program Silicon Laboratories Integrated Development Environment. Využívá programovací jazyk C.

Tento program umožňuje zejména ladění, kompilaci a nahrávání programu do mikroprocesoru. V režimu ladění lze program krokovat a zobrazovat vybrané hodnoty (viz. Obr. 20 v pravé části).



Obr. 20: *Programovací prostředí Silicon Laboratories IDE*

6.1 GPS

Tato kapitola obsahuje nejdůležitější informace, které byly zapotřebí při programování zpracování dat z GPS modulu.

6.1.1 Inicializace

GPS modul je zapotřebí nejprve inicializovat. Při inicializaci GPS modulu se deaktivuje zasílání všech protokolů kromě RMC (viz. příloha 1).

Po prvním připojení modulu bylo přednastaveno zasílání všech protokolů. Bylo možné toto nastavení ponechat, tzn. zpracovávat veškeré protokoly a vybrat si z nich pouze potřebná data. Zvolil jsem však druhou variantu, kterou je právě deaktivace zasílání nepotřebných protokolů, čímž se zamezí redundantnosti dat a případné možné chyby s tím spojené.

6.1.2 Zpracování dat

Data z modulu jsou oddělena čárkou, celý řetězec je zakončen znaky <CR><LF>. Počet znaků u hodnot v řetězci je proměnný (např. u rychlosti a kurzu). Přenosová rychlost je nastavena na 4800 bps.

Je zde snížena možnost případné chyby vzniklé načtením nesprávné hodnoty z řetězce. Z původních 6 protokolů je zasílán pouze jeden, čímž byla zásadně omezena možnost chybného načtení.

Využitá data z řetězce – pro zobrazení na displeji a v PC (jdoucí po sobě, oddělena čárkou):

Název	Příklad	Popis
Message ID	$GPRMC	protokol RMC
UTC Time	151130.525	hhmmss.sss
Status	A	A=data valid / V=data not valid
Latitude	4929.7664	ddmm.mmmm
N/S Indicator	N	N=north / S=south
Longitude	01311.2841	dddmm.mmmm
E/W Indicator	E	E=east / W=west
Speed Over Ground	0.15	knots
Course Over Ground	0.00	degrees
Date	210313	ddmmyy

Pro zobrazení na displeji je upraven datum do formátu dd.mm.rrrr a čas je posunut o hodinu napřed (tzn. přechod z UTC na UTC+1) a převeden do formátu hh:mm:ss. Není zajištěn přechod na letní čas.

Příjem dat (protokolu) z modulu signalizuje zelená LED (LED4). Neplatnost dat z družic signalizuje červená LED (LED3).

6.2 LCD

Tato kapitola obsahuje nejdůležitější informace potřebné při programování displeje.

Dvouřádkový dvacetimístný podsvícený displej Sharp je ovládaný následujícími piny:

PIN	Funkce
RS	Data / Instruction register select
R/W	Read / Write
E	Enable signal, start data read/write
DB0 – DB7	Data Bus Line

Displej pracuje s 5V logikou na rozdíl od mikroprocesoru, který pracuje na 3,3V. Z toho důvodu jsou mezi výstupy z mikroprocesoru a vstupy do displeje přidány budiče (74LS245). Pro správný chod displeje je velice důležitá inicializace displeje, správné nastavení prodlev při zasílání příkazů. Pro ovládání displeje je nutné využívat u procesoru režim Push-Pull (při použití Open-Drain by museli být přidány pull-up rezistory).

6.2.1 Inicializace

Nejdříve je zapotřebí displej inicializovat (viz. Obr. 21). Při inicializaci se definuje počet řádků a počet znaků displeje.

Pro inicializaci displeje se používá metoda init_display. Tato metoda je doprogramována, nejedná se o metodu poskytovanou výrobcem. Je řešena způsobem uvedeným v příloze 2.

Zapnutí

Prodleva alespoň 15 ms po nárůstu V_{CC} na 4,5V

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

Prodleva alespoň 4,1 ms

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

Prodleva alespoň 100 µs

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	*	*	*	*

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	N	F	*	*
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	I/D	S

Obr. 21: *Proces inicializace LCD*

Legenda:

*	0 nebo 1 (na hodnotě nezáleží)
N	=1 pro dvouřádkový displej (pro jednořádkový displej N=0)
F	=0 pro znaky 5x8 (pro znaky 5x10 F=1)
I/D	=1 inkrementace adresy DDRAM o 1 při zápisu znaku
S	=1 přepnutí celého displeje doleva

6.2.2 Bitmapy - přednastavené

Displej má v paměti předdefinované znaky (viz. Obr. 22), neobsahuje však české znaky, které bylo nutné dodefinovat.

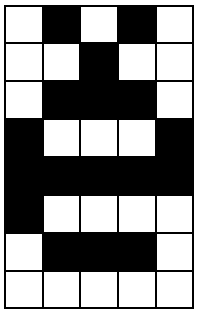
Obr. 22: *Definované znaky v paměti LCD (převzato z [9])*

6.2.3 Bitmapy - české

Na displeji lze zobrazit české znaky ě, š, č, ř, ý, í, á, é. Tyto bitmapy českých fontů je nutné vygenerovat do paměti.

Příklad uživatelsky definované bitmapy (viz. Obr. 23) pomocí kódu, ve kterém každé hexadecimální číslo v závorce definuje řádek bitmapy:

code byte font1[8] = { 0x0a, 0x04, 0x0e, 0x11,0x1f, 0x10, 0x0e, 0x00 };



Obr. 23: *Česká bitmapa - ě*

Inicializace českých fontů je zpracována dle přílohy 3.

6.2.4 Nastavení

Před samotným zápisem na displej se definuje nastavení displeje příkazem:
control_display(D, C, B);

Legenda:

D 1 – zapnutý displej, 0 – vypnutý displej

C 1 – zobrazení kurzoru, 0 – nezobrazení kurzoru

B 1 – blikání kurzoru, 0 – neblikání kurzoru

V našem případě:
control_display(1, 0, 0);

Tato metoda je doprogramována, nejedná se o metodu od výrobce LCD. Je řešena následovně:

```
void control_display( bit display, bit cursor, bit blink )
{
  byte set;
  set = ((byte)display<<2) | ((byte)cursor<<1) | ((byte)blink);
  DATA_LCD = ( 0x08 | set );
  RS = 0;
  RW = 0;
  CE_DS = 1;
  CE_DS = 0;
  LCD_busy();

}
```

6.2.5 Zápis

Pro zobrazení textu na displeji se používá příkaz:

enter_text(L, P, "text");

Legenda:

L	0 – 1. řádek, 1 – 2. řádek
P	pozice prvního symbolu (0 – 9)
text	zobrazovaný text

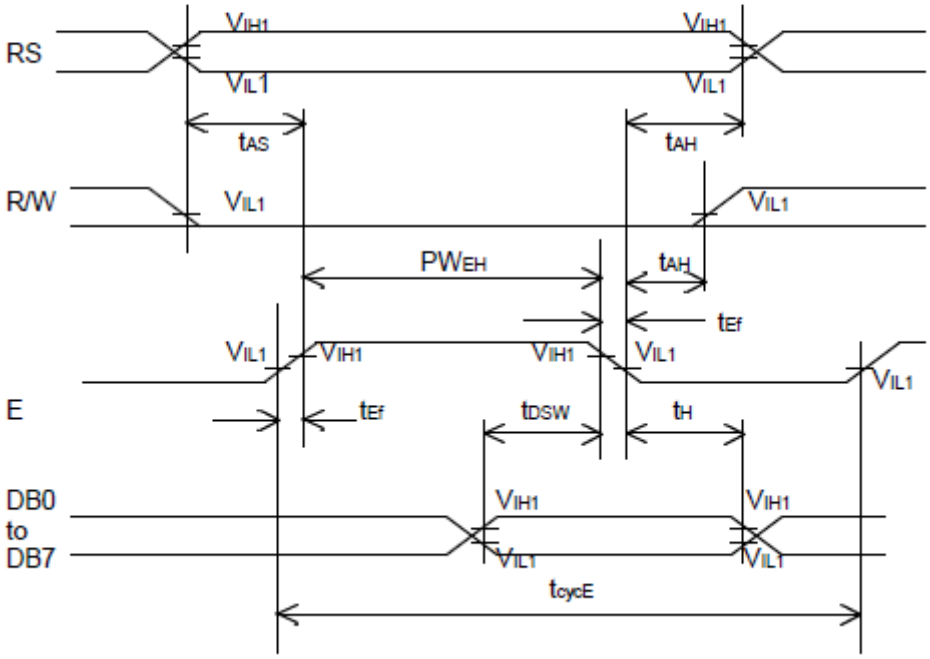
Např.:

enter_text(0, 0, "GPS Receiver v1.0");

Tato metoda je doprogramována, nejedná se o metodu od výrobce LCD. Je řešena dle přílohy 4.

Nutností je dodržení časování pro zápis na displej (viz. Obr. 24)

Item	Symbol	Condition	Min.	Max.	Unit
Enable cycle time	tcycE	VDD = 5V	500	–	ns
Enable pulse width (high level)	PWEH		220	--	
Enable rise/fall time	tEr, tEf		--	25	
Address set-up time (RS, R/W,to E)	tAS		40	--	
Address hold time	tAH		10	–	
Data set-up time	tDSW		60	--	
Data hold time	tH		10	--	



Obr. 24: *Časování zápisu na displej (převzato z [9])*

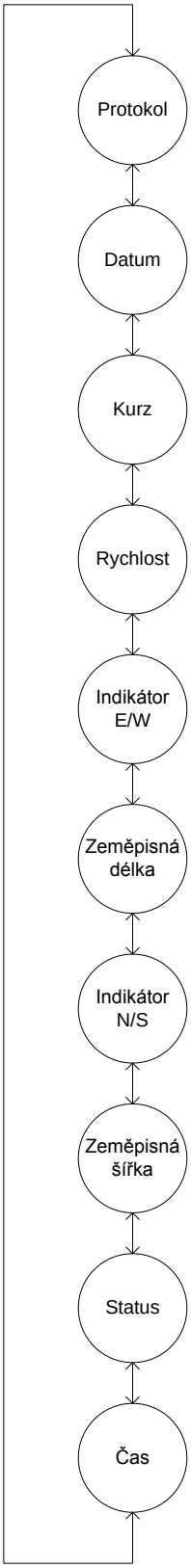
6.2.6 Vymazání

Pro vymazání údajů na displeji se používá metoda clear_display. Tato metoda je doprogramována, nejedná se o metodu poskytovanou výrobcem. Je řešena následujícím způsobem:

```
void clear_display( void )
{
 DATA_LCD = 0x01;
 RS = 0;
 RW = 0;
 CE_DS = 1;
 CE_DS = 0;
 LCD_busy();
}
```

6.2.7 Tlačítka ovládání zobrazení

Tlačítka slouží pro pohyb v menu pro výběr zobrazovaných dat na displeji. Jakmile jsou vybrána požadovaná data pro zobrazení (nedochází ke stiskům tlačítek), dojde k zobrazení vybrané informace. Při procházení menu jsou tlačítka rozsvícena, jakmile je zobrazena vybraná položka z menu, tlačítka zhasnou. Menu displeje je sestaveno v pořadí dle Obr. 25. Menu lze procházet postupně obousměrně (viz. Obr. 26 až Obr. 45).



Obr. 25: *Menu displeje*

Data na displeji jsou zobrazována v menu v následujícím pořadí:

Obr. 26: *Menu displeje - protokol*

Obr. 27: *Displej - protokol*

Obr. 28: *Menu displeje - datum*

Obr. 29: *Displej - datum*

Obr. 30: *Menu displeje - kurz*

Obr. 31: *Displej - kurz*

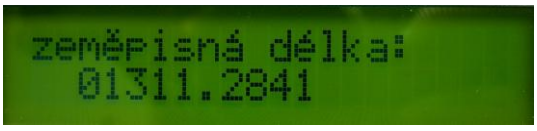
Obr. 32: *Menu displeje - rychlost*

Obr. 33: *Displej - rychlost*

Obr. 34: *Menu displeje – indikátor E/W*

Obr. 35: *Displej – indikátor E/W*

Obr. 36: *Menu displeje – zeměpisná délka*

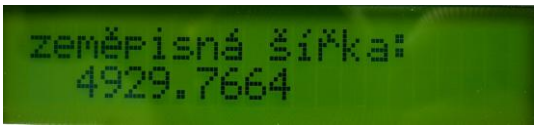


Obr. 37: *Displej – zeměpisná délka*

Obr. 38: *Menu displeje – indikátor N/S*

Obr. 39: *Displej – indikátor N/S*

Obr. 40: *Menu displeje – zeměpisná šířka*



Obr. 41: *Displej – zeměpisná šířka*

Obr. 42: *Menu displeje - status*

Obr. 43: *Displej - status*

Obr. 44: *Menu displeje - čas*

Obr. 45: *Displej - čas*

6.3 Mikroprocesor

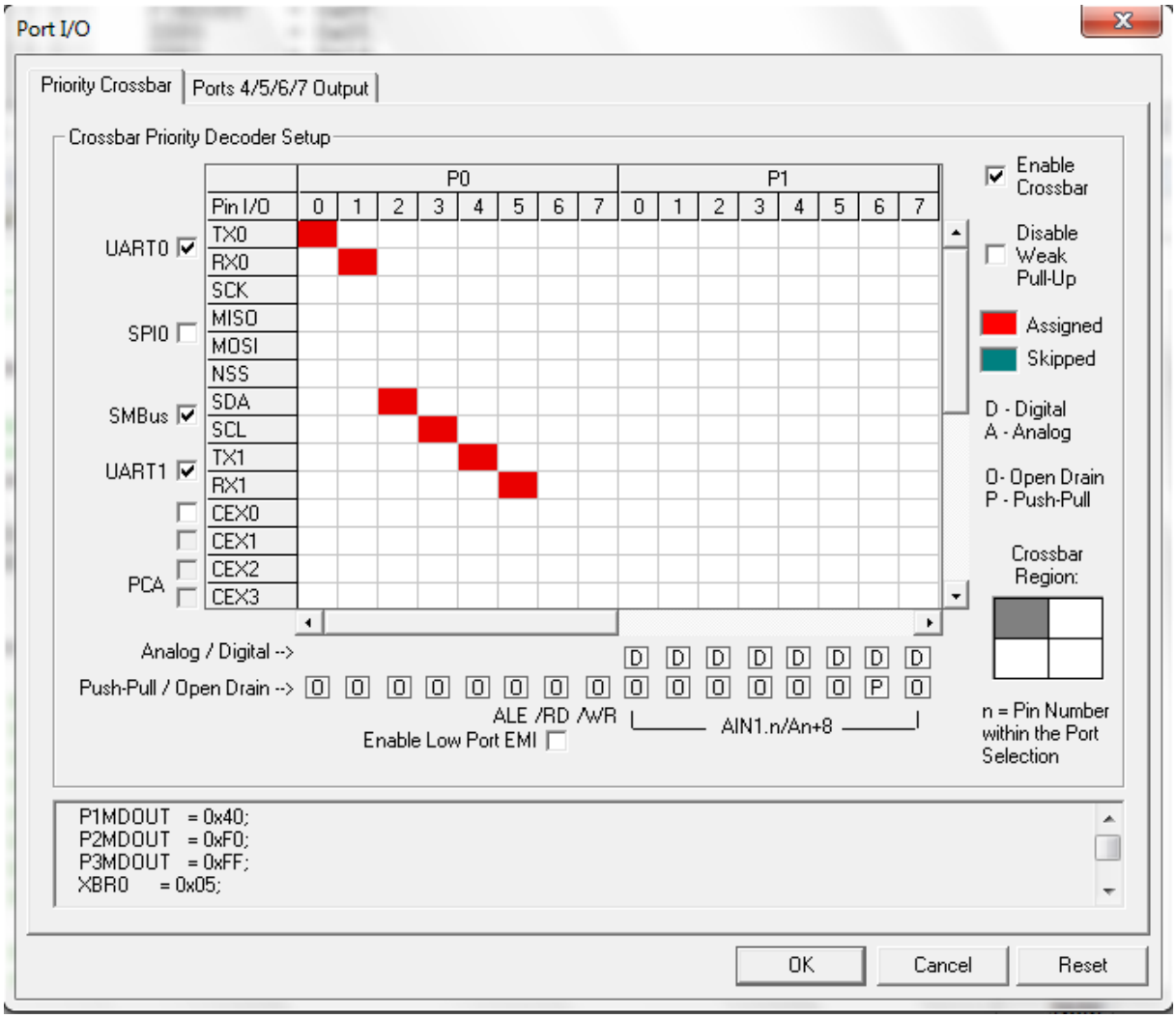
K programování procesoru a ladění bylo použito vývojového kitu (vývojová deska s mikroprocesorem, USB debug adaptér, napájecí adaptér) a programovacího softwaru Silicon Laboratories Integrated Development Environment (verze 4.20.00).

Pro konfiguraci portů, systémových hodin, přerušení bylo využito generátoru Config2 Version 3.00 od Silicon Laboratories. V této aplikaci lze konfigurovat také další periferie (časovače, UART atd.).

6.3.1 Porty

Pro konfiguraci portů bylo využito generátoru Config2 (viz. Obr. 46). Porty pro displej bylo nutné nastavit do režimu push-pull (při logické 0 port přiveden na GND, při logické 1 port přiveden na VDD). Oproti režimu open-drain, kdy je při logické 1 port uveden do stavu vysoké impedance, což způsobovalo problémy zejména při inicializaci displeje, při nevyužití pull-up rezistorů a nevyužití budičů, kdy náběhy pulzů byly příliš dlouhé. Komunikace procesoru s GPS modulem a převodníkem je pomocí sériového rozhraní (UART).

Porty jsou nakonfigurovány dle přílohy 5.



Obr. 46: *Silabs Config2*

6.3.2 Systémové hodiny

Pro konfiguraci systémových hodin bylo využito generátoru Config2.

Zvolená frekvence odpovídá výchozímu nastavení konfigurátoru, pro naše použití je vyhovující.

Nastavení systémových hodin pro používání krystalu 22,1184MHz:

```
void SYSCLK_Init( void )
{
  int i;                          // delay counter

  OSCXCN = 0x67;                  // start external oscillator with
                                  // 22.1184MHz crystal

  for (i=0; i < 256; i++) ;       // wait for XTLVLD to stabilize

  while (!(OSCXCN & 0x80)) ;      // Wait for crystal osc. to settle

  OSCICN = 0x88;                  // select external oscillator as
                                  // SYSCLK
                                  // source and enable missing clock
                                  // detector

}
```

6.3.3 Časovač

Základní kód vygenerovaný Config2 byl modifikován pro potřeby zařízení následovně:

```
void Timers_Init( void )
{

 CKCON  = 0x10;                   // SYSCLK/12 for
                                  // timer0, SYSCLK for timer1
 TMOD   = 0x11;                   // nastaveni modu citacu
```

```
 TH0    = 0xb7;                    // pocatecni nastaveni horniho byte
 TL0    = 0xff;                    // pocatecni nastaveni dolniho byte
 TR0    = 1;                       // spusteni citace CNT 1
 TF0    = 0;                       // shozeni priznaku preteceni

 TH1    = 0xa9;                    // pocatecni nastaveni horniho byte
 TL1    = 0x99;                    // pocatecni nastaveni dolniho byte
 TR1    = 1;                       // spusteni citace CNT 1
 TF1    = 0;                       // shozeni priznaku preteceni

 ET0    = 1;                       // Enable interrupt
 ET1    = 1;                       // Enable interrupt

}
```

6.3.4 Přerušení

Pro konfiguraci přerušení bylo využito generátoru Config2.

```
void Interrupts_Init (void)
{
   IE       = 0x05;                              // Enable IE0 (INT0)
}
```

6.3.5 UART

Konfigurace UART pro převodník USB – UART:

```
void UART0_Init( void )
{
  SCON0  = 0x50;                          // SCON0:mode 1,
```

```
                                        // 8-bit UART, enable RX
  T2CON |= 0x30;                        // Stop Timer; clear int flags;
                                        // enable UART baudrate mode;
                                        // enable 16-bit
                                        // auto-reload timer function;
                                        // disable external count
                                        // and capture modes
  RCAP2 = -(SYSCLK/BAUDRATE_PC/32);     // set Timer reload value for
                                        // baudrate
  T2 = RCAP2;                           // initialize Timer value
  CKCON |= 0x20;                        // Timer2 uses SYSCLK as time
                                        // base
  T2CON |= 0x04;                        // TR2 = 1; start Timer2
  PCON  |= 0x80;                        // SMOD0 = 1
  ES0    = 1;                           // enable UART0 interrupts

}
```

Konfigurace UART pro GPS modul:

```
void UART1_Init( void )
{
  SCON1  = 0x50;                        // SCON1:mode 1, 8-bit
                                        // UART,enable RX
  T4CON  = 0x30;                        // Stop Timer; clear int flags;
                                        // enable UART baudrate mode;
                                        // enable 16-bit
                                        // auto-reload timer function;
                                        // disable external count
                                        // and capture modes
  RCAP4  = -(SYSCLK/BAUDRATE_GPS/32);   // set Timer reload value
                                        // for baudrate
  T4     = RCAP4;                       // initialize Timer value
```

```
    CKCON |= 0x40;                  // Timer4 uses SYSCLK as
                                    // time base

    T4CON |= 0x04;                  // TR4 = 1; start Timer4
    PCON  |= 0x10;                  // SMOD1 = 1
    EIE2  |= 0x40;                  // enable UART1 interrupts

}
```

6.4 Převodník USB – UART

Pro přenos dat z procesoru do počítače je použita sériová komunikace. Do PC je odesílán celý řetězec z GPS modulu, ve kterém jsou jednotlivá data oddělena čárkou. Další zpracování dat je prováděno až v počítači pomocí programu vyvinutého pro tento účel. Ovladače k převodníku jsou dodávány výrobcem.

Přenosová rychlost rozhraní je nastavena na 57600 bps.

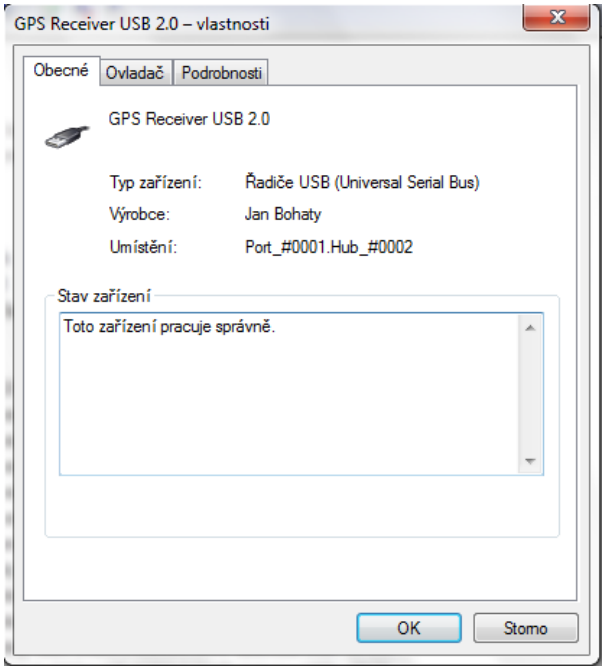
Popis použitých pinů:

PIN	FTDI	Popis
1	GND	zem
2	VCC	napájení 3,3V
10	RXD	data z procesoru
11	TXD	data do procesoru
12	GND	zem
13	GND	zem
14	VCC	napájení 3,3V
15	VCCIO	napájení 3,3V
16	RESET	externí reset
19	RXLED	indikace příjmu
20	TXLED	indikace vysílání
23	USBDP	USB data signal plus
24	USBDM	USB data signal minus

6.5 *Program pro PC*

Pro zobrazení dat z GPS modulu v počítači je vyvinut program pomocí C++ Builderu.

FTDI čip lze nakonfigurovat v programu MProg od výrobce FTDI Chip. V tomto programu lze pro zařízení nastavit název, výrobce atp.. Vlastnosti zařízení po připojení k počítači jsou zobrazeny na Obr. 47.



Obr. 47: *Vlastnosti USB zařízení*

Program zobrazuje „syrová“ data z řetězce vysílaného GPS modulem:

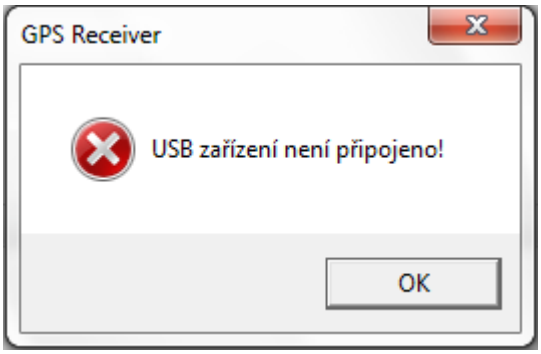
- protokol
- UTC
- status
- zeměpisná šířka
- indikátor N/S
- zeměpisná délka
- indikátor E/W
- rychlost
- kurz
- datum

Dále program zobrazuje upravená data:

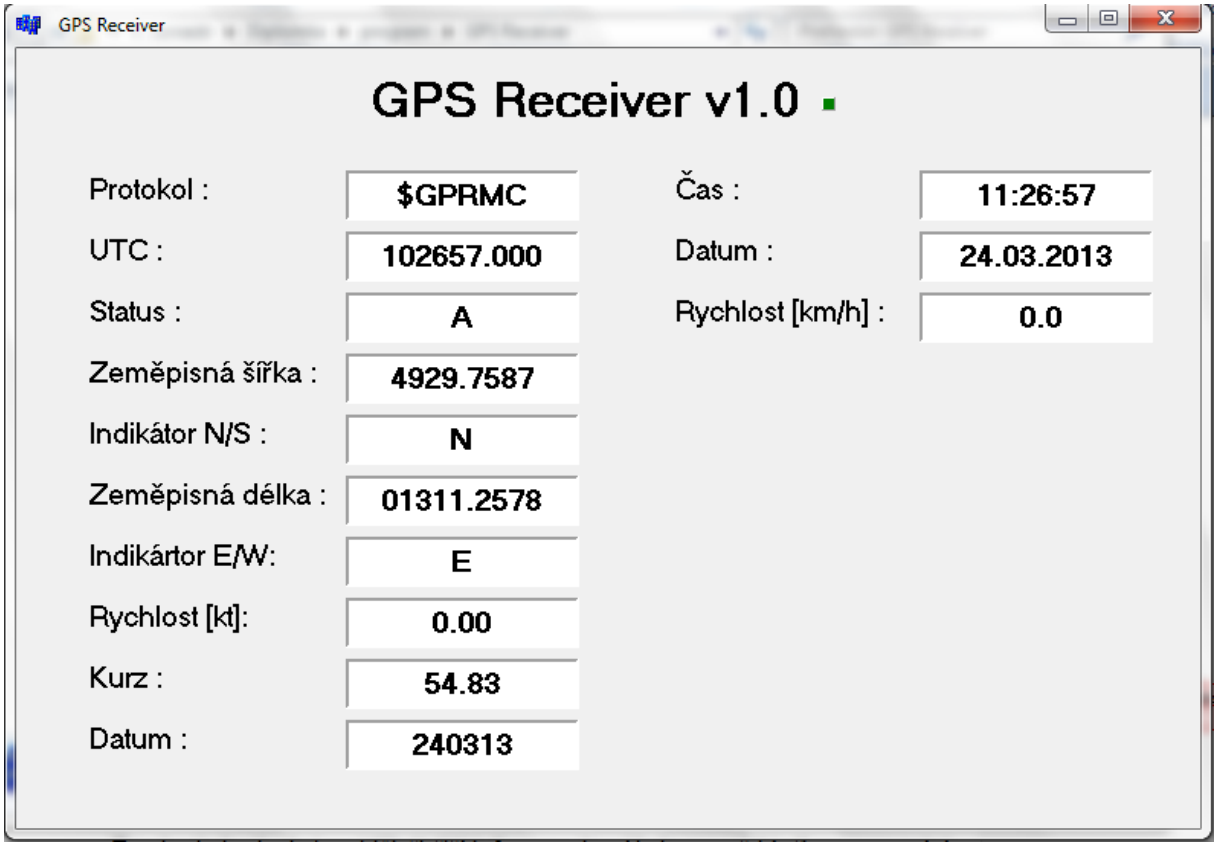
- čas ve formátu UTC+1 (časové pásmo pro ČR – středoevropský čas)
- datum ve formátu dd.mm.rrrr
- rychlost [km/h]

Platnost přijímaných dat signalizuje zelený čtverec vedle nadpisu. Červený čtverec signalizuje neplatnost přijímaných GPS dat.

Po spuštění programu (GPSReceiver.exe) jsou přenášena data ze zařízení do počítače. V případě, že není zařízení připojeno k počítači pomocí USB kabelu, je zobrazena chyba (viz. Obr. 48).



Obr. 48: *Zařízení nepřipojeno*



Obr. 49: *Program pro PC – GPS Receiver (v1.0)*

7 Závěr

Tato práce vznikla jako součást projektu na zakázku, kterou obdržela firma Vectra Electronic s.r.o., a stala se tak jeho základem, který je dále rozvíjen a modifikován dle požadavků zákazníka.

Oproti návrhu došlo ke zjednodušení ovládání pomocí tlačítek. Namísto 4 tlačítek (nahoru, dolu, ok, storno) jsou použita pouze 2 tlačítka (nahoru, dolu). Při výběru z menu se po uvolnění tlačítek zobrazí vybraná hodnota.

Časově nejnáročnější bylo programování části pro LCD. U displeje je nejdůležitější správné nastavení procesu inicializace, kde záleží na časování signálů. V případě, že proces inicializace neproběhne korektně, displej se chová zcela nevyzpytatelně. Nesprávná inicializace se většinou projevuje nezobrazováním žádných znaků, zobrazováním zcela jiných znaků, možností zápisu pouze na první řádek atp.. Porty mikroprocesoru pro displej je vhodné nastavit do režimu push-pull (není poté zapotřebí použití pull-up rezistorů jako v režimu open-drain). Pro zajištění spolehlivosti chodu displeje bylo využito budičů, které konvertují úroveň logické 1 z 3,3V z mikroprocesoru na 5V pro displej.

Pro budoucí využití je vhodné použití jiného GPS modulu, který umožňuje připojení externí antény. Současně použitý GPS modul nedokáže přijímat platný signál z družic při zhoršených provozních podmínkách (např. v těsné blízkosti stavení). Dalším rozšířením zařízení je použití bezdrátového přenosu dat do PC namísto kabelového USB spojení. Velice vhodné by bylo rozšíření o možnost ukládání GPS dat na paměťovou kartu. Tento projekt má rozsáhlé možnosti pro další vývoj, které budou rozvíjeny teoretickými nápady a také zejména použitím v praxi.

8 Použitá literatura

[1] HEROUT, Pavel. *Učebnice jazyka C. III. upravené a rozšířené vydání.* České Budějovice: Nakladatelství KOPP, 1994. ISBN 80-85828-21-9.

[2] ŠUBRT, Vladimír. *Jednočipové mikropočítače INTEL 8048-8096.* GRADA a.s., 1992. ISBN 80-85424-66-5.

[3] HRBÁČEK, Jiří. *Komunikace mikrokontroléru s okolím - 1. díl. 1. vydání.* Praha: BEN - Technická literatura, 1999. ISBN 80-86056-42-2.

[4] 80C51-Based 8-bit Microcontrollers Data Handbook IC20. Eindhoven: Phillips Semiconductors, 1995.

[5] SiRF Binary Protocol Reference Manual. San Jose: SiRF Technology, Inc., November 2008.

[6] NMEA Reference Manual. San Jose: SiRF Technology, Inc., January 2005.

[7] MATOUŠEK, David. *USB prakticky s obvody FTDI - 1. díl. 1. vydání.* Praha: BEN - Technická literatura, 2003. ISBN 80-7300-103-9.

[8] Uživatelská příručka k modulu UMS3. Praha: Asix s.r.o., 2010.

[9] LCD Module Specification. Taipei: Data Image Corporation, October 1998.

[10] National Coordination Office for Space-Based Positioning, Navigation, and Timing: *Global Positioning System* [online], 2013 [cit. 2013-01-01]. Dostupné z WWW: <http://www.gps.gov/>.

[11] ŠVÁBENSKÝ, Otakar. *Základy GPS a jeho praktické aplikace. 1. vydání.* Brno: CERM, 1995. ISBN 80-214-0620-8.

[12] BEZDĚK, Miloslav. *Elektronika: [učebnice]*. *1. vydání*. České Budějovice: Kopp, 2004. ISBN 80-723-2212-5.

[13] PINKER, Jiří. *Mikroprocesory a mikropočítače. 1. vydání.* Praha: BEN - technická literatura, 2004. ISBN 80-730-0110-1.

9 Přílohy

9.1 Příloha 1: Inicializace GPS

```
void init_GPS( void )
{
    ms_timer_array[MS_TIMER] = 100;
    while ( ms_timer_array[MS_TIMER] != 0 );

    while (TX1_ready == FALSE);
    strcpy(TX1_buffer,"$PSRF103,0,0,0,1*24\r\n");          // GGA disable
    len = strlen(TX1_buffer);
    TX1_buffer[len] = 0x00;
    TX1_ready = FALSE;
    SCON1 |= 0x02;
    while (TX1_ready == FALSE);
    strcpy(TX1_buffer,"$PSRF103,1,0,0,1*25\r\n");          // GLL disable
    len = strlen(TX1_buffer);
    TX1_buffer[len] = 0x00;
    TX1_ready = FALSE;
    SCON1 |= 0x02;
    while (TX1_ready == FALSE);
    strcpy(TX1_buffer,"$PSRF103,2,0,0,1*26\r\n");          // GSA disable
    len = strlen(TX1_buffer);
    TX1_buffer[len] = 0x00;
    TX1_ready = FALSE;
    SCON1 |= 0x02;
    while (TX1_ready == FALSE);
    strcpy(TX1_buffer,"$PSRF103,3,0,0,1*27\r\n");          // GSV disable
    len = strlen(TX1_buffer);
    TX1_buffer[len] = 0x00;
    TX1_ready = FALSE;
```

```
    SCON1 |= 0x02;
    while (TX1_ready == FALSE);
    strcpy(TX1_buffer,"$PSRF103,4,0,1,1*21\r\n");          // RMC enable; rate 1 sec
    len = strlen(TX1_buffer);
    TX1_buffer[len] = 0x00;
    TX1_ready = FALSE;
    SCON1 |= 0x02;
    while (TX1_ready == FALSE);
    strcpy(TX1_buffer,"$PSRF103,5,0,0,1*21\r\n");          // VTG disable
    len = strlen(TX1_buffer);
    TX1_buffer[len] = 0x00;
    TX1_ready = FALSE;
    SCON1 |= 0x02;
}
```

9.2 Příloha 2: Inicializace displeje

```
void init_display( void )
{
    RS = 0;                 // Instruction register
    RW = 0;                 // Write
    CE_DS = 1;              // Read/write enable signal
    DATA_LCD = 0x38;        // Function set: 2 Line, 8-bit, 5x8 dots
    for (i=0; i<10; i++);
    CE_DS   = 0;

    for (i=0; i<10000; i++);

    RS = 0;                 // Instruction register
    RW = 0;                 // Write
    CE_DS  = 1;             // Read/write enable signal
    DATA_LCD = 0x38;        // Function set: 2 Line, 8-bit, 5x8 dots
    for (i=0; i<10; i++);
    CE_DS   = 0;

    for (i=0; i<10000; i++);

    RS = 0;                 // Instruction register
    RW = 0;                 // Write
    CE_DS = 1;              // Read/write enable signal
    DATA_LCD = 0x38;        // Function set: 2 Line, 8-bit, 5x8 dots
    for (i=0; i<10; i++);
    CE_DS   = 0;

    for (i=0; i<10000; i++);
}
```

9.3 Příloha 3: Inicializace českých fontů

```
void init_czech_fonts( void )
{
    byte j;
    for (j=0; j<64; j++) {
        DATA_LCD = 0x40 + j;
        RS = 0;
        RW = 0;
        CE_DS = 1;
        CE_DS = 0;
        LCD_busy();

        if (j>=  0 && j<=  7) DATA_LCD = font1 [j-  0];
        if (j>=  8 && j<= 15) DATA_LCD = font2 [j-  8];
        if (j>= 16 && j<= 23) DATA_LCD = font3 [j- 16];
        if (j>= 24 && j<= 31) DATA_LCD = font4 [j- 24];
        if (j>= 32 && j<= 39) DATA_LCD = font5 [j- 32];
        if (j>= 40 && j<= 47) DATA_LCD = font6 [j- 40];
        if (j>= 48 && j<= 55) DATA_LCD = font7 [j- 48];
        if (j>= 56 && j<= 63) DATA_LCD = font8 [j- 56];
        RS = 1;
        RW = 0;
        CE_DS = 1;
        CE_DS = 0;
        LCD_busy();

        }

}
```

9.4 Příloha 4: Zápis textu na displej

```
void enter_text( bit line, byte position, char text[] )
{
    byte j, str;
    int strlen( char *text);

    if ( line ) {
      // nastaveni adresy DD RAM
      RS = 0;
      RW = 0;
      CE_DS = 1;
      DATA_LCD = (0xc0 + position);
      for (i=0; i<5; i++);
      CE_DS = 0;
      for (i=0; i<10000; i++);
    }
    else  {
      RS = 0;
      RW = 0;
      CE_DS = 1;
      DATA_LCD = (0x80 + position);
      for (i=0; i<5; i++);
      CE_DS = 0;
      for (i=0; i<10000; i++);
    }

     for (i=0; i<10000; i++);
     str = strlen(text);
     RS = 1;

  // pro české znaky
  if (DATA_LCD > 0x7f) { // pokud je větší než 127
```

```
  for (j=0; j<str; j++) {
      for ( i=0; i<100; i++) _nop_();
      CE_DS = 1; _nop_();
      DATA_LCD = text[j];
       i = text[j];
       if ( DATA_LCD == 0xec ) //ě
         DATA_LCD = 0x00;
       if ( DATA_LCD == 0x9a ) //š
         DATA_LCD = 0x01;
       if ( DATA_LCD == 0xe8 ) //č
         DATA_LCD = 0x02;
       if ( DATA_LCD == 0xf8 ) //ř
         DATA_LCD = 0x03;
       if ( DATA_LCD == 0xf2 ) //ň = ý
         DATA_LCD = 0x04;
       if ( DATA_LCD == 0xed ) //í
         DATA_LCD = 0x05;
       if ( DATA_LCD == 0xe1 ) //á
         DATA_LCD = 0x06;
       if ( DATA_LCD == 0xe9 ) //é
         DATA_LCD = 0x07;

      CE_DS = 0;
      LCD_busy();

      }
}
      }
```

9.5 Příloha 5: Konfigurace portů mikroprocesoru

```
void PORT_Init ( void )
{

    // P0.0  -  TX0 (UART0), Open-Drain, Digital to PC
    // P0.1  -  RX0 (UART0), Open-Drain, Digital from PC
    // P0.2  -  SDA (SMBus), Open-Drain, Digital not connected
    // P0.3  -  SCL (SMBus), Open-Drain, Digital not connected
    // P0.4  -  TX1 (UART1), Open-Drain, Digital to GPS modul
    // P0.5  -  RX1 (UART1), Open-Drain, Digital from GPS modul
    // P0.6  -  INT0 (Tmr0), Open-Drain, Digital SW Up
    // P0.7  -  INT1 (Tmr1), Open-Drain, Digital SW Down

    // P1.0  -  Unassigned,  Open-Drain, Digital
    // P1.1  -  Unassigned,  Open-Drain, Digital
    // P1.2  -  Unassigned,  Open-Drain, Digital
    // P1.3  -  Unassigned,  Open-Drain, Digital
    // P1.4  -  Unassigned,  Open-Drain, Digital
    // P1.5  -  Unassigned,  Open-Drain, Digital
    // P1.6  -  Unassigned,  Push-Pull , Digital  LED PCB
    // P1.7  -  Unassigned,  Open-Drain, Digital

    // P2.0  -  Unassigned,  Open-Drain, Digital  LED SW Up
    // P2.1  -  Unassigned,  Open-Drain, Digital  LED SW Down
    // P2.2  -  Unassigned,  Open-Drain, Digital  LED GPS Received
    // P2.3  -  Unassigned,  Open-Drain, Digital  LED GPS Valid data
    // P2.4  -  Unassigned,  Push-Pull , Digital
    // P2.5  -  Unassigned,  Push-Pull , Digital  LCD E
    // P2.6  -  Unassigned,  Push-Pull , Digital  LCD RS
    // P2.7  -  Unassigned,  Push-Pull , Digital  LCD RW

    // P3.0  -  Unassigned,  Push-Pull , Digital  LCD D0
```

```
    // P3.1  -  Unassigned,  Push-Pull , Digital  LCD D1
    // P3.2  -  Unassigned,  Push-Pull , Digital  LCD D2
    // P3.3  -  Unassigned,  Push-Pull , Digital  LCD D3
    // P3.4  -  Unassigned,  Push-Pull , Digital  LCD D4
    // P3.5  -  Unassigned,  Push-Pull , Digital  LCD D5
    // P3.6  -  Unassigned,  Push-Pull , Digital  LCD D6
    // P3.7  -  Unassigned,  Push-Pull , Digital  LCD D7

    P1MDOUT   = 0x40;
    P2MDOUT   = 0xF0;
    P3MDOUT   = 0xFF;
    XBR0      = 0x05;
    XBR1      = 0x14;
    XBR2      = 0x44;

}
```