ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ

Katedra elektromechaniky a výkonové elektroniky

BAKALÁŘSKÁ PRÁCE

GPS logger s mikrokontrolérem ARM

Ondřej Šedivec **2013**

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: **2012/2013**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Ondřej ŠEDIVEC**
Osobní číslo:	**E10B0081K**
Studijní program:	**B2644 Aplikovaná elektrotechnika**
Studijní obor:	**Aplikovaná elektrotechnika**
Název tématu:	**GPS logger s mikrokontrolérem ARM**
Zadávající katedra:	**Katedra elektromechaniky a výkonové elektroniky**

Zásady pro vypracování:

Datalogger bude ukládat data o poloze získaná z GPS na paměťovou kartu. Data by měla být ve formátu vhodném pro další zpracování. Zařízení bude vybaveno vhodným uživatelským rozhraním a bude napájeno z akumulátoru.

1. Uveďte základní principy globálního polohového systému GPS.
2. Sestavte a naprogramujte funkční prototyp pomocí vývojového kitu s mikrokontrolérem ARM.
3. Zhodnoťte navržené zařízení (případně uveďte možnosti dalšího vylepšení či rozšíření).

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah pracovní zprávy: **20 - 30 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Mann, B.: C pro mikrokontroléry**

Vedoucí bakalářské práce: **Ing. Lukáš Valda**
Katedra aplikované elektroniky a telekomunikací

Datum zadání bakalářské práce: **15. října 2012**

Termín odevzdání bakalářské práce: **7. června 2013**

Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan

Západočeská univerzita v Plzni
Fakulta elektrotechnická
4

Prof. Ing. Václav Kůs, CSc.
vedoucí katedry

V Plzni dne 15. října 2012

Abstrakt

První část bakalářské práce popisuje základní principy fungování globálního polohového systému (GPS). Další část je věnována využití mikrokontroléru s jádrem ARM Cortex-M3, jenž zajišťuje komunikaci s GPS modulem pomocí protokolu NMEA0183 a rovněž umožňuje záznam uskutečněné trasy na SD paměťovou kartu. Poslední část se zabývá návrhem uživatelského rozhraní a popisem jeho ovládání.

Klíčová slova

GPS, ARM, Cortex-M3, OLIMEX, STM32, NMEA0183, GUI

Abstract

The first part of bachelor thesis describes basic principles of the Global Positioning System (GPS). The next part is devoted to the use of ARM microcontroller with Cortex-M3 core, which provides communication with the GPS module by NMEA0183 protocol and also allows recording of made track to the SD memory card. The last part deals with the design of the graphic user interface and describes its controlling.

Key words

GPS, ARM, Cortex-M3, OLIMEX, STM32, NMEA0183, GUI

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 5.6.2013 Jméno, příjmení

……...…..……..

Poděkování

Děkuji panu Ing. Lukáši Valdovi za četné poznámky a připomínky, které vedly ke zkvalitnění této práce po obsahové i formální stránce.

Také bych chtěl poděkovat svým rodičům za bezmeznou podporu, a to nejen při zpracování mé bakalářské práce, ale i po celou dobu mého studia.

Obsah

Seznam obrázků

Seznam tabulek

Seznam příloh

Seznam symbolů a zkratek

ADC.....	**A**nalog to **D**igital **C**onverter
C/A.....	**C**oarse/**A**cquisition Code
CID.....	**C**ard **I**dentification **D**ata
CSD.....	**C**ard **S**pecific **D**ata
DSP.....	**D**igital **S**ignal **P**rocessor
FAT.....	**F**ile **A**llocation **T**able
FSMC.....	**F**lexible **S**tatic **M**emory **C**ontroller
GPS.....	**G**lobal **P**ositioning **S**ystem
GRAM.....	**G**raphic **R**andom **A**cces **M**emory
GUI.....	**G**raphic **U**ser **I**nterface
HAL.....	**H**ardware **A**cces **L**ayer
HOW.....	**H**and **O**ver **W**ord
JPO.....	**J**oint **P**rogram **O**ffice
JTAG.....	**J**oint **T**est **A**ction **G**roup
LCD.....	**L**iquid **C**rystal **D**isplay
LED.....	**L**ight **E**mitting **D**iode
MBR.....	**M**ain **B**oot **R**ecord
MCS.....	**M**aster **C**ontrol **S**tation
MCU.....	**M**icrocontroller **U**nit
MS.....	**M**onitor **S**tation
NVIC.....	**N**ested **V**ector **I**nterrupt **C**ontroller
P(Y).....	**P**rotected **C**ode
PWM.....	**P**ulse **W**idth **M**odulation
RGB.....	**R**ed-**G**reen-**B**lue
RTC.....	**R**eal **T**ime **C**lock
SDIO.....	**S**ecure **D**igital **I**nput **O**utput
SRAM.....	**S**tatic **R**andom **A**ccess **M**emory
SV.....	**S**pace **V**ehicle
SWD.....	**S**erial **W**ire **D**ebug
TCP/IP.....	**T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol
TLM.....	**T**elemetry **W**ord
UHF.....	**U**ltra **H**igh **F**requency
USART.....	**U**niversal **S**ynchronous/**A**synchronous **R**eciever and **T**ransmitter
USB.....	**U**niversal **S**erial **B**us
VBR.....	**V**olume **B**oot **R**ecord

Úvod

Toto téma jsem si vybral především z důvodu edukativního průzkumu možnosti komunikace mikrokontroléru s periferiemi a vytvoření grafického uživatelského rozhraní (GUI). Navržený GPS logger nabízí možnost ověření ovládání sériového portu, displeje, ADC převodníku a použití souborového systému.

Původně byl uvažován 8-bitový mikrokontrolér Atmel ATmega, ale v poslední době je trendem přechod na 32-bitové MCU s jádry ARM. Jejich výhody spočívají především v úspornosti, výkonnosti a v počtu dostupných periferií. Za nevýhodu lze považovat přílišnou složitost periferií, na což výrobci reagují tím, že dávají k dispozici jejich ovladače. Použit je dnes značně rozšířený mikrokontrolér řady STM32F1 s jádrem ARM Cortex-M3 od spol. ST Microelectronic. Tato společnost na svých webových stránkách nabízí ke stažení, kromě výše uvedených ovladačů ovládající periferie mikrokontroléru, také velké množství knihoven, které usnadňují vývoj aplikací, např. pro tento projekt použitou knihovnu k vytvoření GUI, včetně jejího návrhového software a také knihovnu ovládající přístup k paměťové kartě.

Prototyp GPS loggeru je sestaven a naprogramován na vývojovém kitu OLIMEX STM32-LCD, jehož nedílnou součástí je barevný dotykový displej o rozlišení 320x240 bodů, který v dnešní době tvoří moderní ovládací prvek a zobrazovací rozhraní. Výhodou tohoto displeje je především možnost zobrazit velké množství informací a ovládacích prvků najednou bez potřeby jejich přepínání. Dále tento vývojový kit nabízí již vestavěný slot na SD kartu.

Práce je rozdělena do tří částí. První se zabývá stručným popisem požadavků na vytvoření systému GPS a základními principy jeho fungování. Jsou popsány druhy uživatelů, jimi přijímané frekvence a kódování signálů, které mají vliv na přesnost určení polohy. Je uveden obsah navigační zprávy přijímané z družice a chyby zaměření.

Druhá část se pokouší teoreticky nastínit obecnou problematiku všech jednotlivých prvků GPS loggeru. Uvádí základy řízení barevného LCD displeje s řadičem, detekci polohy pomocí odporové dotykové vrstvy, komunikaci SD karty s mikrokontrolérem za použití standardu SDIO a přenos dat po sériovém portu z GPS modulu ve standardu NMEA0183. Dále se zabývá návrhem vhodných obvodů k řízení napájení vývojového kitu s GPS modulem a nabíjení li-pol akumulátoru.

Poslední část prakticky popisuje uživatelské rozhraní, realizaci výstupního souboru s uloženými geodaty pro mapový software Google Earth a důležité částí kódů, především inicializace periferií MCU, použité knihovní funkce a funkce zajišťující chod gps loggeru.

1 Globální polohový systém GPS

Vývojové práce na GPS byly zahájeny v roce 1973 jako výsledek fúze několika programů (TIMATION, TRANSIT, 621B) výzkumu a vývoje v rámci amerického ministerstva obrany. První satelit byl vypuštěn v roce 1978. O celý systém se stará U.S. Air Force – divize vesmírných systémů, Joint Program Office (JPO), na základně leteckých sil v Los Angeles.

Cílem JPO bylo vyvinout nepřetržitě fungující, na počasí nezávislý, celosvětově dostupný navigační systém na podporu lokalizačních schopností ozbrojených sil, který splňuje následující podmínky:

- Vhodný pro všechny druhy ozbrojených sil (letecké, pozemní, námořní, vesmírné)
- Schopný zachytit velkou dynamiku pohybu
- Určení pozice a rychlosti v reálném čase
- Nejvyšší přesnost omezena dle třídy uživatele
- Odolnost proti rušení
- Redundance – systém funkční i v případě poškození některých jeho součástí
- Nahradit v té době stárnoucí TRANSIT a ostatní navigační systémy něčím komplexnějším

1.1 Pozemní kontrolní část

Hlavní kontrolní stanice (MCS) je umístěna v Coloradu - monitoruje a řídí polohou GPS družic, stejně tak i jejich synchronizaci přesného času a korekce na základě matematického modelu celého systému. V případě nefunkčnosti některého ze satelitů je řídící stanicí označen v efemeridech jako unhealth, tedy nezdravý. Přijímač GPS toto zaznamená a přestane lokalizační data považovat za korektní.

Data o ostatních družicích, které nejsou viditelné z MCS, jsou získávána/odesílána z tzv. monitorovacích stanic (MS) rozmístěných po celém světě při každém jejich průletu.

1.2 Vesmírná část

Satelitní síť obsahuje celkem 32 družic označovaných jako Space Vehicle (SV). Aby bylo dosaženo celosvětové pokrytí, je jich potřeba alespoň 24. Zbylé slouží pro zpřesnění a pro případ výpadku některé z družic. Družice jsou umístěny ve výšce 20 200 km, pohybují se na 6 drahách se sklonem k rovníku 55 až 60 stupňů a Zemi oběhnou přibližně dvakrát za jeden den. Z každého místa na povrchu můžeme zaměřit najednou přibližně 8-9 satelitů, což

zajišťuje dostatečnou redundanci, vezmeme-li v potaz, že k zaměření polohy je potřeba minimálně 4 satelitů.

Každý SV obsahuje [1]:

- Atomové hodiny s rubidiovým oscilátorem
- 12 antén pro vysílání navigačních dat v pásmu L (1000-2000 MHz)
- Antény pro komunikaci s pozemními kontrolními stanicemi v pásmu S (2204,4 MHz)
- Anténu pro vzájemnou komunikaci družic v pásmu UHF
- Rentgenové, optické a elektromagnetické senzory pro detekci jaderných výbuchů a startů balistických raket
- Solární panely a baterie jako zdroj energie

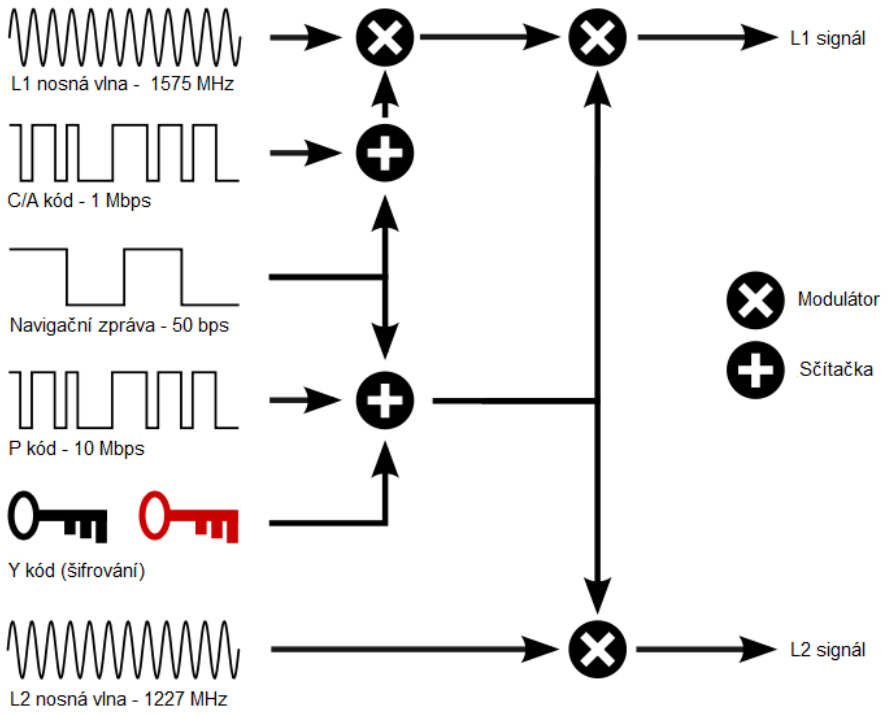
1.3 Signál GPS družic

Všechny satelity vysílají navigační data na dvou stejných základních frekvencích 1575,42 MHz (L1 signál) a 1226,60 MHz (L2 signál). Pro modulaci nosné se používá binární fázové klíčování (BPSK) - při každé změně binárního čísla se změní fáze nosné o 180 stupňů.

Standardní (nevojenské) přijímače využívají pásmo L1. Navigační zpráva je modulována společně s nešifrovaným C/A (Coarse/Acquisition - hrubá lokalizace) kódem o taktovací frekvenci 1,023 MHz. Nosná vlna je pro standardní přijímače posunuta o 90 stupňů, z důvodu namodulování dvou signálů (navigační zprávy s C/A a P(Y) kódováním) na jednu frekvenci.

Vojenské přijímače využívají L1 i L2 pásmo šifrované P(Y) kódem (Protected) s taktovací frekvencí 10,23 MHz. Toto přináší značné výhody, neboť kromě vyšší přenosové rychlosti, tedy vyšší přesnosti, je možné provádět i korekce chyb způsobených zpožděním signálu průchodem ionosférou (ionosférickými refrakcemi) při příjmu lokalizačních dat ze dvou různých frekvencí současně.

P(Y) i C/A kód jsou vlastně pseudonáhodné kódy, díky kterým lze identifikovat družici vysílající na stejné frekvenci jako ostatní - využívá se tzv. kódového multiplexu. Pseudonáhodný kód má vlastnosti obdobné jako šum (rozprostřené spektrum), jenž slouží jako ochrana před rušením. Schéma principu modulace je na obr. 1-1.



Obr. 1-1 Schéma modulace signálu [1]

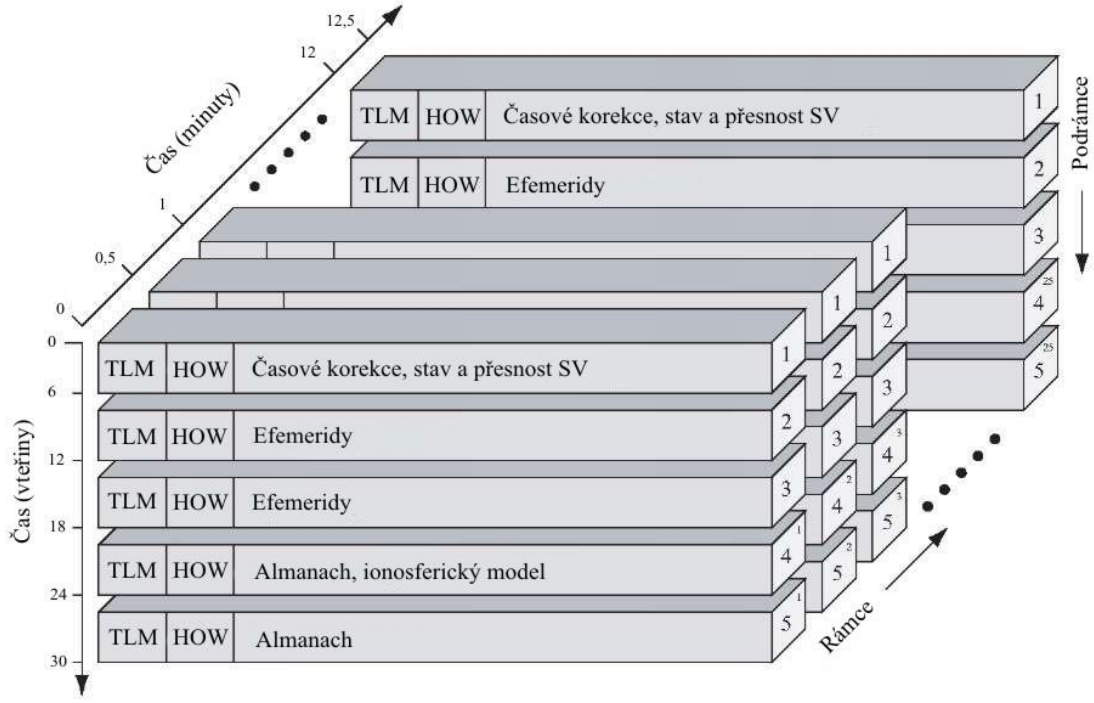
Dále je využito pásmo L3, které slouží pro přenos informací ze senzorů jaderných výbuchů a rozpoznávání startu balistických raket. U novějších satelitů je počítáno i s pásmy L4 a L5. L5 je vyhrazené pásmo určené leteckému provozu, kde musí být garantováno minimum rušení. Bude v něm využíváno vyšších přenosových rychlostí, delšího kódu a vyšších vysílacích výkonů. Pásmo L4 slouží pro vědecké účely. Spolu s modernizací SV jsou do budoucna plánována další rozšíření, např. zavedení civilního pásma L2C, kde bude možno využívat příjmu ve dvou pásmech, jak bylo již popsáno i pro nearmádní využití. Přehled všech frekvencí je uveden v tab 1-1.

Tab. 1-1 Používaná pásma GPS satelitů [1]

Pásmo	Frekvence	Popis
L1	1575,42MHz	C/A kód, šifrovaný P(Y) kód
L2	1226,60MHz	šifrovaný P(Y) kód
L3	1381,05MHz	používané systémem (NUDET) – detekování nukleárních výbuchů
L4	1379,913MHz	je studováno pro budoucí využití ionosferických korekcí
L5	1176,45MHz	Safety of Life

1.3.1 Navigační zpráva

Navigační zpráva obsahuje 25 rámců dat, každý rámec obsahuje 1500 bitů a je rozdělen do pěti podrámců po 300 bitech. Přenosová rychlost je 50 bit/s, přijmutí jednoho podrámce tedy trvá 6 vteřin, jednoho rámce 30 vteřin, všech rámců 12,5 minuty. Podrámce 1 až 3 jsou specifické pro konkrétní satelit, obsahují stejná data pro všech 25 rámců a opakují se po 30 vteřinách. Podrámce 4,5 jsou společné pro celou síť družic, v každém rámci obsahují jiná data a opakují se po 12,5 minutách (viz obr. 1-2).



Obr. 1-2 Rámce navigační zprávy [3]

Všechny podrámce obsahují na začátku slova TLM (Telemetry Word) a HOW (Hand Over Word). TLM nese informaci o začátku podrámce, HOW jeho pořadí od začátku navigačního týdne. První podrámec určuje nutné časové korekce, zdraví satelitu a přesnost určení pseudovzdálenosti. Druhý a třetí obsahuje tzv. efemeridy, což jsou předpovědi polohy družice na oběžné dráze, podle předem vypočítaného matematického modelu. Čtvrtý a pátý tvoří tzv. almanach (informace o efemeridech ostatních družic), informace o ionosféře a zdraví ostatních družic.

1.4 Princip zaměření polohy

Uvažujme, že měříme naši vzdálenost od jednoho satelitu, a zjistíme, že jsme 22 000 km daleko, to znamená, že se nacházíme někde na povrchu pomyslné koule. Poté přidejme k

prvnímu druhý satelit, který je od nás vzdálený 23 000 km a zjistíme, že se nacházíme na průsečíku obou dvou sfér, tedy kdekoliv na kružnici. Když přidáme i třetí satelit, třeba ve vzdálenosti 24 000 km, průsečíky všech třech sfér budou tvořit dva body. Z toho jeden můžeme zanedbat za předpokladu, že se pozorovatel nachází na zemi. Je využíváno tzv. triangulace.

Vzdálenost družice od pozorovatele je určena dobou šíření elektromagnetického vlnění od satelitu do přijímače za známé rychlosti:

$$\boldsymbol{l = c.(t_p - t_d)} \tag{1-1}$$

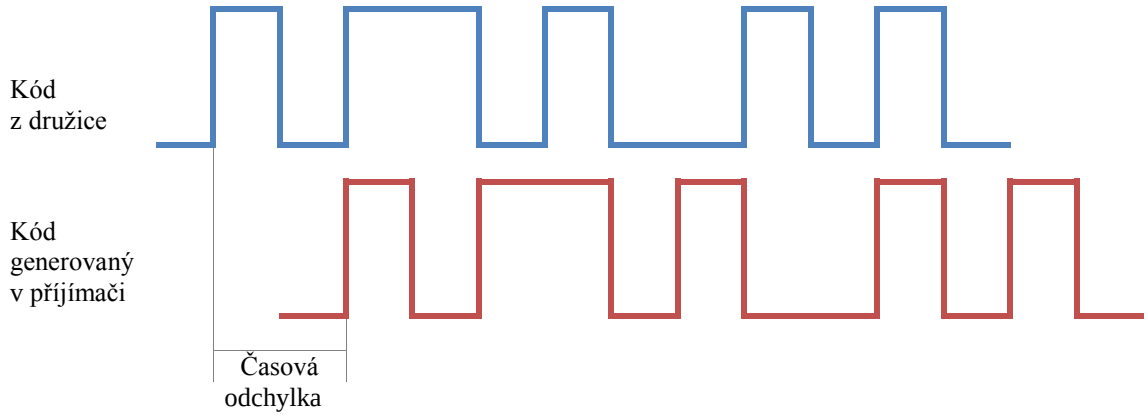
, kde c - rychlost světla

t_p - čas družicových hodin v okamžiku odeslání signálu

t_d - čas hodin přijímače v době zachycení signálu

l - vzdálenost mezi družicí a přijímačem

Satelity vysílají pseudonáhodný kód, který přijímač porovnává se svým vlastním generovaným pseudonáhodným kódem. Vzdálenost posuvu kódu přijímače, tak aby se překrýval (byl synchronní) s vyslaným kódem, se poté bude rovnat časové odchylce (obr. 1-3). Z té je pak možno určit vzdálenost od satelitu tzv. pseudovzdálenost.



Obr. 1-3 Synchronizace pseudonáhodných kódů

Křemíkové hodiny v přijímači však nejsou natolik přesné, aby pouze tato metoda byla dostačující. Pokud bychom si představili, že měříme dobu, za kterou přijmeme signál ze satelitu nacházejícího se přímo nad námi, došli bychom k závěru, že odchylka jedné tisíciny vteřiny by znamenala chybu 321km [2].

K zpřesnění se používá čtvrtý satelit, který odstraní chybu hodin v přijímači. Je vypočítán tzv. fix koeficient tak, aby se všechny naměřené pseudovzdálenosti protnuly v jednom bodě (obr. 1-4) a dostaly skutečnou vzdálenost ke všem satelitům, a tedy i přesnou polohu.

Obr. 1-4 a) vzdálenosti naměřené třemi družicemi b) vzdálenosti naměřené čtyřmi družicemi c) vzdálenosti naměřené po opravě chyby času [2]

Pseudonáhodný kód je poté odfiltrován a je získán C/A kód, ze kterého je vyčtena navigační zpráva.

1.5 Chyby zaměření

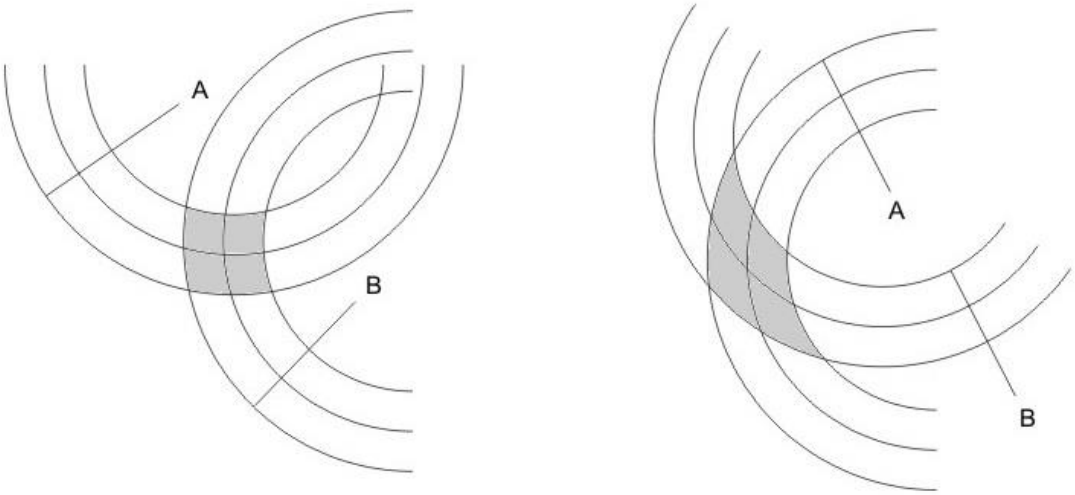
Vznikají především nehomogenitou atmosféry, tedy rozdílnou rychlostí šíření signálu v různých prostředích. Tyto chyby jsou nejmenší, pokud se sledovaná družice nachází přímo nad námi a největší, pokud je blízko horizontu, jelikož signál musí urazit velkou vzdálenost skrz atmosféru.

Rozložení chyb je následující [2]:

- Vliv ionosféry ±5 m
- Chyby efemeridů ±2,5 m
- Chyba hodin satelitů ±2 m
- Zkreslení vlivem vícecestného šíření signálu ±1 m
- Vliv troposféry ±0,5 m
- Numerické chyby ±1 m

Dále je nutné počítat i se vzájemnou geometrickou polohou satelitů, které přijímáme. Dokud budou rozprostřeny ve velkém úhlu od sebe, odchylka bude minimální. Naopak při výhledu přijímače, např. jen do úzkého pásma před nás, získáme mnohem nepřesnější

informaci o naší poloze. Situace je vidět na obr. 1-5, kde vyšrafovaný pás značí naši přibližnou polohu a střed kružnice polohu satelitu.



Obr. 1-5 a) neomezený výhled b) omezený výhled do úzkého pásu

Chyby ale vznikají i ne úplně přesným porovnáním přijímaného a generovaného pseudokódu. GPS modul a jeho DSP je schopen změřit rozdíl dvou bitových pulzů s odchylkou jednoho procenta jejich šířky, což u C/A kódu taktovaném na frekvenci 1,023 MHz vede k odchylce ±3 m a u P(Y) kódu taktovaném na frekvenci 10,23 MHz k odchylce ±0,3 m, tedy 10x menší.

Uvádí se, že celková teoretická přesnost GPS pro civilní použití by v 95% měření měla být do 13 m v horizontálním a do 22 m ve vertikálním směru [3].

2 Návrh GPS loggeru

Původně byl pro toto zařízení uvažován mikrokontrolér Atmega48, nicméně se ukázalo, že pro zamýšlené použití barevného dotykového displeje s GUI by byl naprosto nedostačují z hlediska jeho omezené kapacity paměti SRAM. Z tohoto důvodu bylo nutné najít jiné řešení, a to rovnou v podobě 32-bitového MCU, kde výběr padl na mikrokontrolér s jádrem ARM Cortex-M3 od firmy ST Microelectronic, která nabízí nezanedbatelnou výhodu ve formě jednoduše použitelné knihovny periferií, umožňující vyhnout se zapisování přímo do registrů. Je vytvořeno jakési přemostění HW vrstvy do pohodlnější formy. Toto je jistá konkurenční výhoda, hlavně pro vcelku rychlé a pochopitelné učení se při přechodu od jiného výrobce čipu.

Další výhody lze těžit z rozsáhlé komunity uživatelů a podpory výrobce, který nabízí volně ke stažení další knihovny značně urychlující vývoj, např. pro implementaci obsluhy

USB, pamětí, TCP/IP a GUI. Cena tohoto 32-bitového MCU je mnohdy srovnatelná s cenou 8 bitového, ale nabízí vyšší výkon a nižší spotřebu.

Pro aplikaci GPS loggeru jsem se rozhodl použít vývojový kit především kvůli možnosti ušetření času (není potřeba se zabývat návrhem, výrobou a osazením desky plošných spojů) a požadavku na výrobek, jelikož se předpokládá, že prototyp může být ještě při vývoji hardwarově doplněn. O součásti rozšiřující jeho funkčnost.

2.1 Vývojový kit

Na vývojový kit je kladeno několik požadavků, především dostupná dokumentace, dostatečně velké pouzdro MCU s možností využít FSMC (Flexible Static Memory Controller) na ovládání LCD. Tímto odpadá využití cenově nejdostupnějšího vývojového kitu STM32VL Discovery postaveného na jádře ARM Cortex-M3, jelikož je vyráběn pouze v malém pouzdře. MCU STM32F4 Discovery FSMC umožnuje, nicméně pro tuto aplikaci se jeví jako příliš výkonný.

Vývojový kit by měl také obsahovat co nejvíce komunikačních periferií vyvedených na pin lišty. Neméně důležité je i 5V napájení celého kitu s dostatečně dimenzovaným 3,3V regulátorem i pro GPS modul a případné další periferie. Tyto specifikace nejlépe splňuje kit od společnosti OLIMEX, model STM32-LCD, který je již osazen LCD dotykovým displejem s možností využít PWM regulaci jasu a slotem na micro SD kartu. Bohužel není připojen spínač detekce vložení karty.

Kit má následující parametry:

- Mikrokontrolér STM32F103ZE s maximální taktovací frekvenci 72 MHz, 512kB FLASH pamětí a 64kB SRAM
- JTAG/SWD konektor pro připojení programátoru/debuggeru
- 2x40 pin, 2x10 pin header s vyvedenými IO piny MCU
- Slot pro mikro SD kartu
- LCD TFT 320x240 s dotykovou vrstvou
- Akcelerometr
- 8MHz krystal
- Napájení možno Mini USB/JST/PIN header konektorem

2.1.1 MCU STM32F103ZE

Mikrokontrolér je umístěn v LQFP pouzdře se 144 vývody. Jeho taktovací frekvence je 72 MHz. Velikost Flash/SRAM paměti je plně postačujících 512, resp. 64 Kbytů. Napájen může být v rozmezí 2 V až 3,6 V. Pro jeho programování je na vývojovém kitu vyveden JTAG konektor. Obsahuje porty označené GPIOA až GPIOG. Každý port tvoří 16 pinů. Piny mohou být nastaveny jako vstupní, výstupní nebo jim může být přiřazena integrovaná periferie. Blokové schéma lze najít v příloze 2 na str. 5.

MCU obsahuje tyto nejdůležitější periferie a součásti:

- 8MHz a 40KHz RC krystal, 32KHz krystal pro generování reálného času
- Deset 16-bitových časovačů
- Tři 12-bitové A/D převodníky se 16 kanály s rozsahem 0 až 3,6 V
- Dva 12-bitové D/A převodníky
- Velké množství komunikačních rozhraní (2x I2C, 5x USART, 3x SPI, CAN, SDIO, USB 2.0)
- 12ti kanálový DMA řadič s podporou téměř všech periferií
- FSMC

ARM Cortex-M3 je označení pro samostatné řešení jádra vyvinuté firmou ARM. Výrobci čipů si sami volí velikost paměti a periferie, které je budou doplňovat, a tvořit tak jednočipový počítač. Je založeno na Harvardské architektuře, která má oddělenou instrukční a datovou sběrnici. To umožňuje přístup k datům a instrukcím v jeden okamžik. Jádro obsahuje NVIC (Nested Vector Interrupt Controller), který dokáže vygenerovat až 240 přerušení s 256 úrovněmi priorit a dynamicky priority přerušení měnit. Jeho součástí je i 24-bitový časovač (SysTick) a také zodpovědnost za napájení a úsporné režimy. Jádro je navrženo s ohledem na maximální úspornost. Všechny periferie jsou po startu bez povolení hodinového taktu, tedy neaktivní. Čtyřgigabytový adresový prostor je pevně rozdělen do několika částí - programová část, SRAM, systémové periferie, přídavné periferie. Část je rezervována pro budoucí modernizaci jádra. Jádro je postaveno na sadě instrukcí Thumb-2. Ta obsahuje, jak 32-bitové, tak i 16-bitové instrukce. Neobsahuje ale původní ARM instrukce, takže není zpětně kompatibilní.

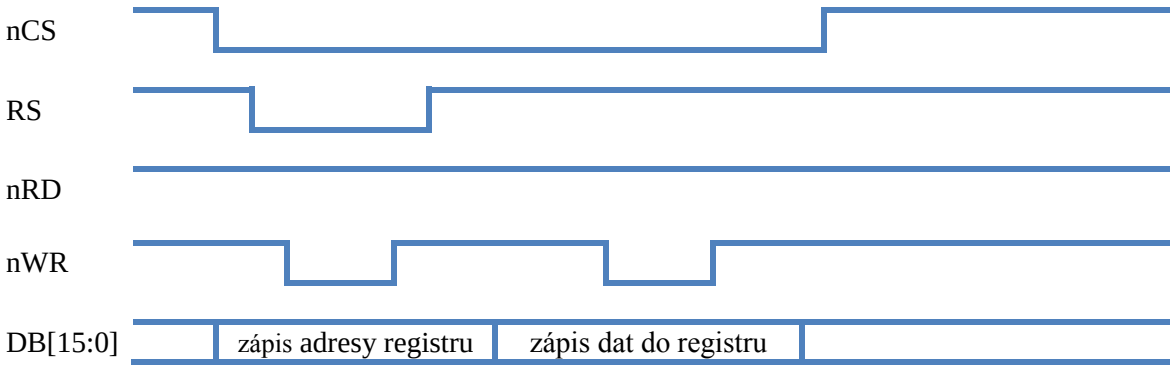
2.1.2 LCD

Barevný grafický displej je osazen řadičem ILI9320, jenž nabízí několik možností komunikace s řídícím prvkem. V tomto případě je použit interface i80 od Intelu se 16-bitovou paralelní datovou sběrnicí. Tab 2-1 popisuje funkce jednotlivých pinů.

Tab. 2-1 Popis funkce pinů LCD

Pin	Popis funkce
nCS	Výběr čipu, log0 povolen
RS	Při log1 čte/zapisuje data, při log0 čte/zapisuje index (adresu) registru
nWR	Povoluje zápis při log0
nRD	Povoluje čtení při log0
Reset	Při přivedení log0 nastává reset LCD
D0:D15	Datové piny

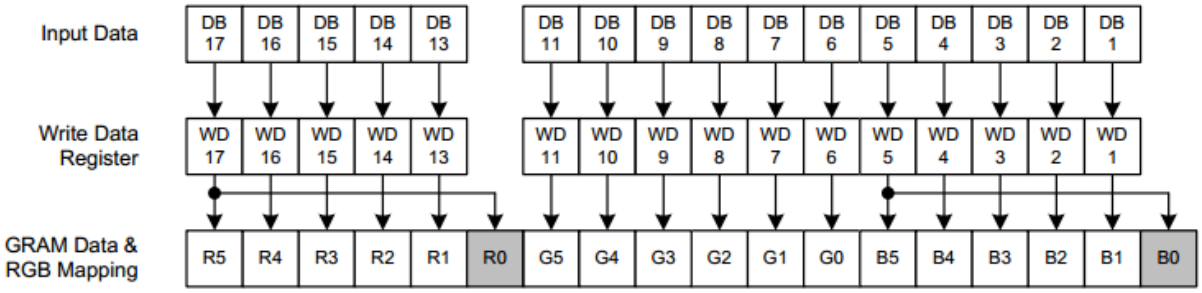
Obecně platí, že při požadavku zápisu (obr. 2-1) je nutné na pinu CS nastavit log0 (výběr čipu povolen), na paralelní datové sběrnici požadovanou adresu registru v bitovém formátu, pin RS a WR nastavit na log0 (výběr indexu, povolení zápisu). Takto zapíšeme adresu požadovaného registru. Poté WR nastavíme na log1 (zákaz zápisu), RS nastavíme na log1 (výběr dat), zaměníme adresu na paralelní sběrnici za požadovaná data, WR na log0 (povolení zápisu), tím zapíšeme datovou část. Nakonec nastavíme řídící piny zpět na log1.



Obr. 2-1 Zápis do řadiče LCD [5]

Čtení probíhá obdobně až do místa zápisu dat, kde je místo nastavení pinu WR do log0 (povolení zápisu) nastaven pin RD do log0 (povolení čtení). Nicméně je nutné dodržovat minimální a maximální doby zpoždění mezi jednotlivými kroky dle dokumentace použitého LCD řadiče.

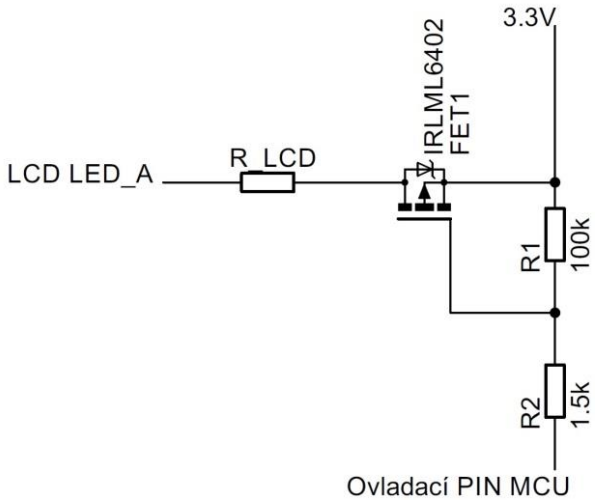
Při 16-bitové barevné hloubce je možno použít 65536 různých barevných kombinací, při 18-bitové 262144 barev. Displej pracuje nativně s 18-bitovou, což znamená pro každou z barevných složek RGB šest bitů, avšak datová sběrnice je jen 16-bitová, tedy je nutno u dvou složek nastavovat dva bity jedním bitem. Toto je řešeno dle obr. 2-2.



Obr. 2-2 16-bitové RGB [5]

Řízení a nastavení chování LCD probíhá pomocí zápisů parametrů do registrů řadiče. Např. pro řadič ILI9320 a editaci pixelu na pozici x,y, zapíšeme do registru na adrese 0x20, resp. 0x21 požadovanou polohu pixelu. Přepneme registrem 0x22 na zápis do GRAM paměti a zapíšeme data ve formátu RGB 5-6-5 bitů.

Podsvícení obvykle u těchto typů LCD bývá tvořeno LED diodami. Pokud by bylo potřeba ovládat intenzitu podsvícení, lze využít PWM modulaci připojením P-MOSFET tranzistoru na anodu led diod pro řízení jejich jasu (obr. 2-3).



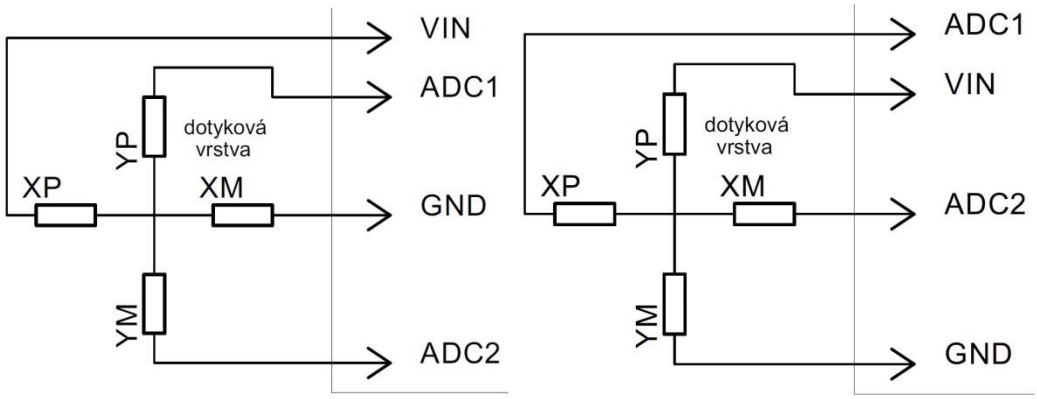
Obr. 2-3 Schéma zapojení řízení LCD podsvícení

2.1.3 Dotyková vrstva

U tohoto typu LCD je použita rezistivní dotyková vrstva bez řadiče. Je tedy nutné využít ADC převodník použitého mikrokontroléru. ADC převodník někdy bývá součástí dotykové vrstvy, pak komunikace probíhá např. po SPI sběrnici, což může být výhoda v případě nedostatku volných pinů mikrokontroléru.

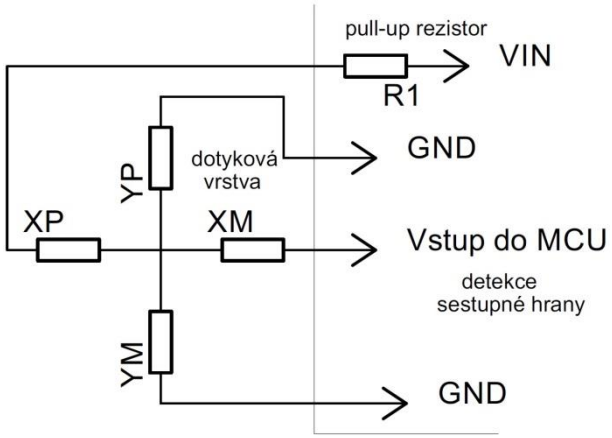
Rezistivní dotyková vrstva je typu 4-wire, tedy má čtyři vstupy/výstupy. Odpory os X a Y jsou vůči sobě zapojeny paralelně. Měření bodu dotyku (obr. 2-4) probíhá tak, že na vstup XP a XM připojíme napětí, resp. zem. Po přitlačení, např. stylusem, se odpory obou os propojí a na výstupech YP, resp. YM naměříme napětí závislé na poloze stylusu v X souřadnicích. Poté připojíme napětí na vstup YP, zem na vstup YM a na výstupech XP, resp.

XM získáme napětí závislé na poloze stylusu v Y souřadnicích. Získaná napětí poté pomocí ADC převodníku převedeme na číselnou hodnotu pro další zpracování [4].

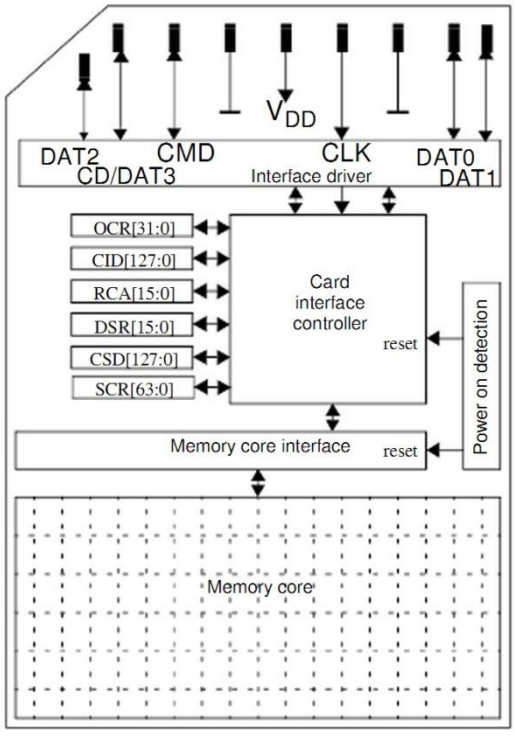


Obr. 2-4 a) měření polohy dotyku X souřadnice b) měření polohy dotyku Y souřadnice [4]

Nutné je však ještě ošetřit případ, kdy je dotyková vrstva delší čas nevyužívána a není třeba mít aktivní ADC převodník (obr. 2-6). Připojíme-li napětí s pull-up rezistorem (cca 5k) na vstup XP a zem na vstup YM, pak při dotyku detekujeme na výstupu XM sestupnou hranu. Tuto hranu můžeme využít pro generování přerušení a aktivaci měření ADC převodníku.



Obr. 2-6 Detekce dotyku [4]



Obr. 2-5 Schéma SD karty [7]

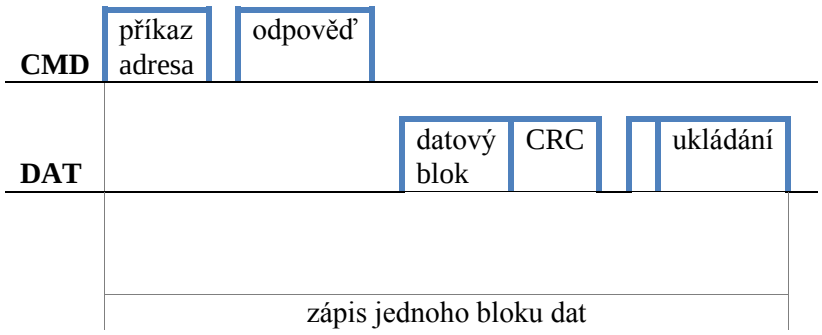
2.1.4 Paměťová karta

SD karta (obr. 2-5) obsahuje řadič, který se stará o komunikaci s vlastní pamětí typu NAND FLASH. Řadičem je spravováno rovnoměrné využití všech bloků paměti pro dosažení co nejvyšší životnosti a jsou v něm obsaženy některé informace o paměti uložené v registrech.

Například CID (Card Identification Data) - výrobce, datum výroby, CSD (Card Specific Data Register) - velikost, max. takt, velikost bloku.

Komunikace může probíhat dvěma způsoby: buďto přes standardní SPI rozhraní, nebo přes specifické rozhraní SD karty, které bude použito, jelikož MCU nabízí dostatek volných pinů a SDIO periferii. Rozdíl je v počtu použitých vodičů a především v rychlosti přenosu dat. SPI využívá signály CLK, MISO, MOSI, CS, přenos probíhá sériově vždy po osmi bitech. Naproti tomu rozhraní SD karty je tvořeno šesti signály. Každým taktem hodin (CLK) je přenesen jeden bit příkazu (CMD) a jeden bit dat (DAT). Maximální taktovací frekvence může být u nových rychlých paměťových karet až 50MHz, při inicializaci je však nutný takt do 400KHz. Pomocí CMD zasíláme sériově SD kartě příkazy, na které je následně odpovězeno tzv. answer tokenem. Ten je dlouhý 48-136 bitů a obsahuje informace dle odeslaného příkazu. Například odešleme-li CMD13 (žádost o stavu karty), dostaneme zpět po CMD 48-bitovou odpověď. Bity 45-40 označují číslo zpracovaného příkazu, 39-8 tvoří skutečnou zprávu o stavu a 7-1 tvoří CRC kontrolu dat. Ostatní jsou využity na rozpoznání začátku/konce zprávy [7].

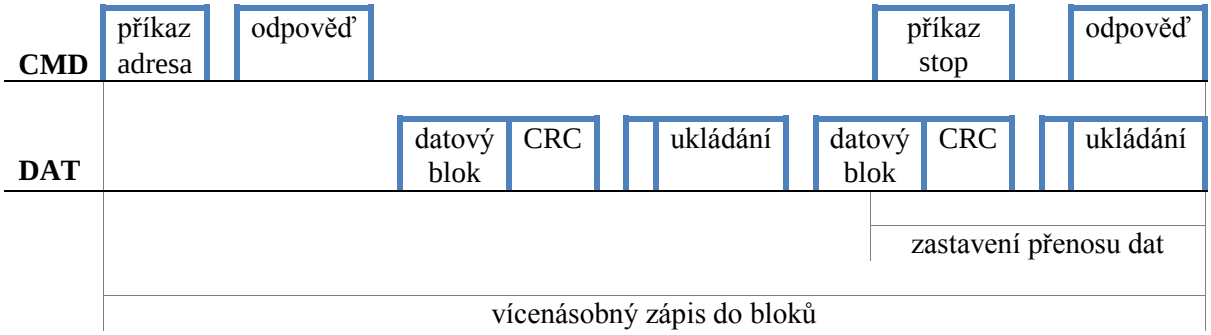
Chceme-li na kartu zapsat data (obr. 2-7), musíme postupovat po blocích (standardně bývá blok 512 bytů). Po CMD vyšleme příkaz CMD24 s argumentem tvořícím 32-bitovou adresu bloku paměti. Pokud nenastala nějaká chyba, vrátí se odpověď o připravenosti k zápisu. Nyní je možno začít posílat data. Přenos probíhá po čtyřech bitech. Start je signalizován uvedením DAT0-3 do log0, poté následuje zápis do celého bloku paměti, který je ukončen CRC kontrolním součtem. Probíhající ukládání do paměti je signalizováno log0 na DAT0 pinu.



Obr. 2-7 Zápis jednoho bloku dat [7]

Při zápisu většího množství dat (obr. 2-8), je vhodnější využít vícenásobný zápis do bloků. Ten probíhá obdobně jako zápis do jednoho bloku s tím rozdílem, že dokud akci nezastavíme (CMD příkazem), je při dokončení zápisu do jednoho bloku adresa bloku

automaticky inkrementována a je pokračováno dalším blokem. Čtení probíhá obdobně, jen odpadá část čekaní na uložení.



Obr. 2-8 Vícenásobný zápis do bloků [7]

Detekce fyzického vložení karty může probíhat buďto spínačem umístěným v patici, nebo DAT3 pinem, který obsahuje pull-up rezistor. Sledováním vstupu MCU lze vyvolat přerušení při změně stavu na log1 a až poté spustit obsluhující periferii.

2.2 Souborový systém FAT

Základní ideou souborového systému je zpřístupnění a ukládání dat pomocí hiearchicky organizovaného systému souborů a adresářů. Data musí být jednoznačně určena svým jménem. Jména souborů jsou doplněna dalšími daty, např. datovými známkami, jejich velikostmi a povolením zápisu. FAT systém je rozdělen na několik částí:

1. MBR (Master Boot Record - Hlavní spouštěcí záznam) se nachází na počátku každého svazku (mimo datovou oblast). Obsahuje informace o počtu diskových oddílů a o umístění spouštecích oddílů (VBR). Můžou být vytvořeny maximálně 4 oddíly, z toho jeden označen jako aktivní, kterému je předáno řízení systému (toto platí pro PC).
2. VBR (Volume Boot Record - Spouštěcí záznam oddílu) se nachází na počátku každé části disku označené jako oddíl (mimo datovou oblast). Obsahuje blok parametrů média - velikost, počet sektorů, velikost alokační jednotky a název svazku.
3. Kořenový adresář - databáze souborů a jejich počátečních alokačních jednotek nacházejících se ve FAT tabulce, včetně atributů.
4. FAT tabulka - každá její buňka odpovídá jedné alokační jednotce (obsahuje tolik buněk, jako je alokačních jednotek) a hodnota v ní uložená ukazuje na další

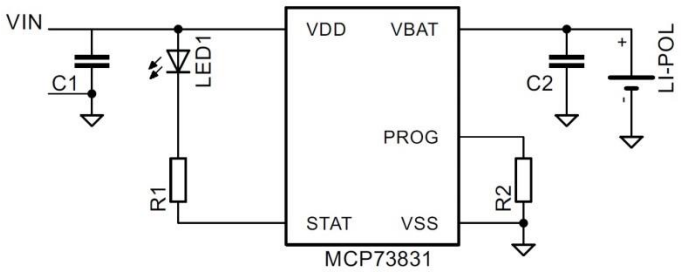
alokační jednotku, ve které je soubor. Pokud již žádná další není, obsahuje značku EOC.

5. Datová oblast - Slouží k ukládání samotných dat, je rozdělena do částí o velikosti alokačních jednotek.

Například naformátovaná paměťová SD karta o velikosti 2 GB (1,83 GB skutečné) má velikost sektoru 512 bytů. Alokační jednotka je tvořena jedním sektorem a obsahuje 3 838 535 alokačních jednotek, což je také velikost FAT tabulky. Alokační jednotka je nejmenší možný úložný prostor. Pokud bychom uložili soubor o velikosti 513 bytů, již budou obsazeny dvě.

2.3 Napájení

Vývojový kit vyžaduje 5V napájení, přičemž nejvyšší možný celkový odběr GPS loggeru je cca 150 mA. Ideálním a rozšířeným zdrojem se jeví USB rozhraní, které je dostačující i pro nabíjení. Nabízí se možnost použití buď 3,7V (jednočlánkového – 1S) li-pol akumulátoru, nebo 7,4V (dvoučlánkového – 2S). Oproti Ni-MH mají vyšší energetickou hustotu, vyšší počet nabíjecích cyklů a nižší hmotnost. Nevýhodou je potřeba sledování poklesu napětí na cca 2,6 V. Při příliš velkém vybití hrozí jeho nevratné poškození. V prvním případě je však nutno použít zvyšující měnič napětí (step-up konvertor), ve druhém snižující měnič napětí (step-down konvertor). Společně s potřebou použití balanceru (obvodu pro vyrovnání napětí jednotlivých článků) a méně dostupnými hotovými nabíjecími/napájecími moduly je však toto řešení nevhodné. Z výše uvedeného důvodu je použit jeden 3,7V článek.



Obr. 2-9 Schéma zapojení nabíjecího integrovaného obvodu MCP73831 [8]

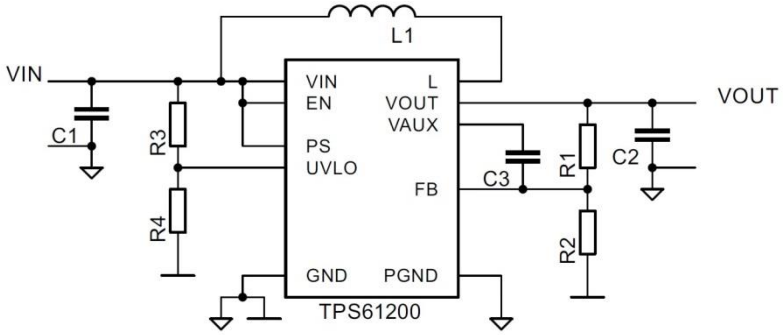
O správu nabíjení li-pol článku se stará integrovaný obvod Microchip MCP73831. Jeho nabíjecí proud je dán vztahem 2-1[8]:

$$\boldsymbol{I_{REG}} = \frac{\mathbf{1000}}{\boldsymbol{R_{PROG}}} \qquad (2\text{-}1)$$

, kde I_{REG} - Nabíjecí proud [mA]

R_{PROG} - Rezistor [kΩ]

Tento proud může být 15 až 500 mA. Připojením rezistoru o hodnotě 2kΩ k výstupu PROG (obr. 2-9) tedy získáme maximální nabíjecí proud 500 mA[8]. LED dioda připojená ke STAT pinu bude svícením informovat o probíhajícím nabíjení.



Obr. 2-10 Step up-regulátor TPS61200 [9]

Step-up konvertor TPS61200 má maximální proudovou zatížitelnost 300 mA při 3,3 V resp. 600 mA při 5 V. Regulovatelné výstupní napětí o rozsahu 1,8 V až 5,5 V je nastavitelné pomocí napěťového děliče z odporů R1, R2 (obr. 2-10). Podle vztahu 2-2[9]:

$$\boldsymbol{R_1 = R_2 * \left(\frac{V_{OUT}}{V_{FB}} - 1\right)} \qquad (2\text{-}2)$$

, kde R_1 - Odpor rezistoru R1

R_2 - Odpor rezistoru R2

V_{OUT} - Požadované výstupní napětí

V_{FB} - Typicky 500mV

Odpor R2 by se měl dle dokumentace [9] pohybovat v mezích okolo 200kΩ. Při potřebě napětí 5 V na výstupu by R2 měl hodnotu 220kΩ a R1 2MΩ.

Součástí spínaného regulátoru je i podpěťová ochrana akumulátoru, která jej odpojí při poklesu napětí pod nastavenou hodnotu. Tato hodnota je nastavena napěťovým děličem složeného z rezistorů R3 a R4 (obr. 2-10). Výpočet je následující (2-3[9]):

$$\boldsymbol{R_3 = R_4 * \left(\frac{V_{INMIN}}{V_{UVLO}} - 1\right)} \qquad (2\text{-}3)$$

, kde R_3 - Odpor rezistoru R3

R_4 - Odpor rezistoru R4

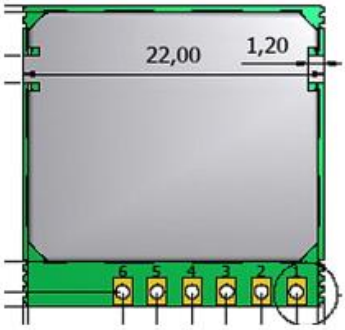
V_{INMIN} - Aktivační napětí podnapěťové ochrany

V_{UVLO} - Referenční napětí (250mV)

Odpor R4 by se měl dle dokumentace [9] pohybovat okolo 250kΩ. Pro účinnou ochranu li-pol akumulátoru by aktivační napětí podpěťové ochrany mělo dosahovat hodnoty okolo 2,6 V. Dle výše uvedeného vztahu, by pak odpor R3 mohl být 2MΩ a R4 220kΩ.

2.4 GPS modul

Jako GPS modul byl vybrán čipset MediaTek MT3329 (obr. 2-11), především z důvodu jeho nízké ceny a dobré dostupnosti. Je schopný přijímat a dekódovat L1 C/A kód. S MCU komunikuje asynchronně po sériovém portu (RXD, TXD). Napájecí napětí může být v rozsahu od 3 V do 4,2 V (VDD). Záložní napájení (VDD_B) umožňuje uchovávat informace o efemeridech, sloužící k rychlejšímu startu a pomocí vnitřních RTC udržovat čas. Pin PPS generuje každou 1 vteřinu puls synchronizovaný s GPS časem. Samotný čip je vyveden na tzv. breakout plošný spoj, osazený integrovanou anténou a umožňující snadnější propojení s vývojovým kitem.

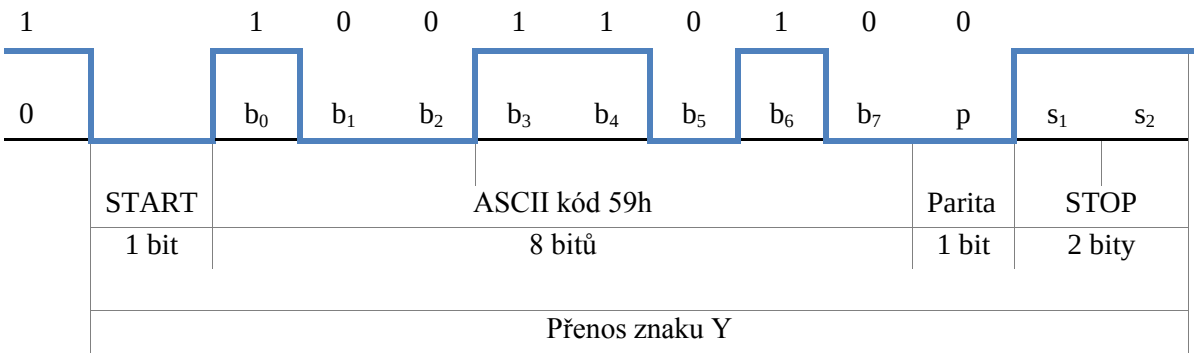


Obr. 2-11 Zapojení GPS modulu [10]

Tab. 2-2 Popis funkce pinů GPS

Pin	Popis funkce
1 - RXD	UART asynchronní vstup
2 - TXD	UART asynchronní výstup
3 - GND	Zem
4 - VDD	Napájení
5 - VDD_B	Záložní napájení
6 - PPS	Generátor pulsů

Asynchronní sériová komunikace (obr. 2-12) probíhá následovně: pro zajištění synchronizace mezi přijímačem a vysílačem předchází každému vysílanému znaku tzv. start bit s logickou hodnotou 0 a následuje alespoň jeden stop bit s logickou hodnotou 1. Start a stop bity obklopují každý odeslaný znak. Časový interval mezi vysláním dvou znaků je různý. Během této doby je komunikační linka v logické 1. Synchronizace na bitové úrovni je dosaženo pomocí hodinových signálů stejné frekvence u vysílače i přijímače. Když přijímač detekuje začátek start bitu, spustí oscilátor, který umožní správné vzorkování bitů. Odběr vzorku probíhá ve středu každého přeneseného bitu z důvodu spolehlivosti. Součástí přenášených dat může být i paritní bit, sloužící k detekci chyb, což je vlastně redundantní bit přidaný k datovému slovu obsahující počet jeho jedničkových bitů.



Obr. 2-12 Sériový přenos

K přenosu informací se používá tzv. NMEA 0183 standard, což je původem námořní komunikační standard, sloužící ke sjednocení komunikace všech lodních systémů. Zprávy jsou tvořeny ASCII znaky přenášenými po osmi bitech s jedním stop bitem, bez paritních bitů, rychlostí 9600 baudů. Každá zpráva začíná znakem $, následuje prefix zařízení vysílajícího zprávu (pro GPS je GP) a poté tyto hlavičky [10]:

- GGA (Global positioning system fix data)
- RMC (Recommended Minimum Specific GNSS Data)
- GSV (Satellites in view)
- GSA (Active satellites)

Pokračují data oddělovaná čárkou, konec dat signalizuje hvězdička (*), za kterou je kontrolní součet, řídící znaky <CR> - návrat na začátek řádky a <LF> - posun na novou řádku. Například zpráva GPRMC (Souhrn navigačních dat) vypadá takto:

```
        1          2  3       4  5         6  7   8    9       10  11
        |          |  |       |  |         |  |   |    |       |   |
$GPRMC,hhmmss.ss,A,llll.ll,a,yyyyy.yy,a,x.x,x.x,ddmmyy,x.x,a*

12 13  14
|  |   |
hh<CR><LF>
```

1) Čas UTC 2) Stav GPS, A = data validní, V = data nevalidní 3) Zeměpisná šířka 4) Polokoule, N = severní, S = jižní 5) Zeměpisná délka 6) Polokoule, E = východní, W = západní 7) Rychlost v uzlech 8) Kurz 9) Datum 10) Magnetická variace 11) E = východní, W = západní 12) Kontrolní součet 13) Řídící znak návratu na začátek řádky 14) Řídící znak posunu na novou řádku

Nastavení GPS modulu probíhá pomocí příkazů. Začínají znakem $, následuje identifikace zařízení, číslo příkazu, parametry příkazu, hvězdičkou oddělený kontrolní součet,

návrat na začátek řádky a posun na novou řádku. Například změna přenosové rychlosti na 38400 baudů by probíhala následovně:

```
 1    2    3       4  5    6
 |    |    |       |  |    |
$PMTK251,38400*27<CR><LF>
```

1) Identifikační hlavička volaného zařízení 2) Číslo příkazu 3) Nová hodnota (parametry) 4) Kontrolní součet 5) Řídící znak návratu na začátek řadky 6) Řídící znak posunu na novou řádku

3 Software

Pro vývoj softwaru je k dispozici několik placených vývojových nástrojů, např. IAR Workbench, Keil uVision, Atollic TrueSTUDIO, Reisonance Ride7, a neplacených, např. CooCox. Z důvodu rozšířenosti ve firemní sféře a osobní preference je využito vývojové prostředí Keil uVision. Jako programátor/debugger slouží ST-LINK V2. Zdrojové kódy jsou součástí přílohy 1.

3.1 main.c

V main.c jsou inicializovány všechny použité periferie. Běh programu je uskutečněn v nekonečné smyčce. Nejprve je ověřeno, zda-li není k dispozici nová GPS zpráva funkcí *gps_poll()*. Pokud ano, je zpracována a uložena do gprmc struktury funkcí *gps_get_gprmc(&gprmc)*. Zeměpisné souřadnice jsou převedeny pomocí *gps_print_polar(lat, gprmc.pos.latitude)* na tvar vhodnější k zobrazení na LCD. Následuje ověření dotyku *ts_poll(&x, &y, &pr)*, případně jeho zpracování GUI *ProcessInputData()*. Obnovení geodat, pokud jsou zobrazeny na *LCD Refresh_s1()*. Funkce pro zápis na SD kartu, když je spuštěno logování *Logger(&fil,&gprmc)*. Nakonec ověření stavu SD karty *log_status_check()*.

3.2 comm.c

Tento soubor obsahuje zdrojový kód pro sériovou komunikaci mezi MCU a GPS modulem, nastavení všech přerušení a funkci přerušení pro zpracování všech příchozích GPS zpráv.

Pro inicializaci USART (Universal Synchronous/Asynchronous Receiver and Transmitter) je nejprve třeba přivést hodiny na port GPIOA a USART1:

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA |
RCC_APB2Periph_USART1, ENABLE);
```

Inicializovat I/O piny PA09 pro Tx do režimu výstupu push-pull a PA10 pro Rx do režimu plovoucího vstupu pro použití společně s USART1:

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

Předchozí příklad platí pro Tx. Pro Rx by vypadal obdobně:

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

Následuje samotná inicializace USART1:

```
USART_InitTypeDef USART_InitStructure;
USART_InitStructure.USART_BaudRate = 9600;
USART_InitStructure.USART_WordLength = USART_WordLength_8b;
USART_InitStructure.USART_StopBits = USART_StopBits_1;
USART_InitStructure.USART_Parity = USART_Parity_No;
USART_InitStructure.USART_HardwareFlowControl =
USART_HardwareFlowControl_None;
USART_InitStructure.USART_Mode = USART_Mode_Rx |
USART_Mode_Tx;
USART_Init(USART1, &USART_InitStructure);
USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
USART_Cmd(USART1, ENABLE);
```

Nejprve je definována struktura *USART_InitStructure* obsahující položky jednotlivých nastavitelných parametrů, které jsou posléze definovány (rychlost, délka slova v bitech, počet stop bitů, paritní bit, kontrola toku dat, směr toku dat). Poté je volána funkce inicializace USART1 z předem definované struktury, konfigurace přerušení (v tomto případě povolení přerušení při příjmu) a nakonec povolení samotného USART1.

Dále soubor obsahuje nastavení priorit všech přerušení, celkem jsou použity 4. Nejvyšší prioritu má příjem znaku z USARTu, následované změnou stavu SD karty, dokončení přenosu DMA z/na SD kartu a RTC. Následující příklad ukazuje inicializaci priority přerušení pro hodiny reálného času mající prioritu 4, tedy nejnižší. Všechna přerušení s nižší hodnotou *NVIC_IRQChannelPreemptionPriority* (vyšší priority) mohou do tohoto přerušení vstoupit a vytvořit tzv. vnořené přerušení.

```
NVIC_InitStructure.NVIC_IRQChannel = RTC_IRQn;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 4;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
```

Poslední část tohoto souboru tvoří funkce obsluhy přerušení při příchozím znaku po sériovém portu - *USART1_IRQHandler*. GPS zprávy jsou zpracovávány na základě detekce /n /r řídících znaků umístěných na konci každé z nich a ukládány do FIFO bufferu tvořeného polem *lines[NUM_LINES][LINE_SIZE]* definovaného v comm.h.

3.3 gps.c

Tato knihovna je součástí demo programu dodáváného s vývojovým kitem. Jsou zde obsaženy funkce pro zpracování gps zpráv. *Gps_poll* zpracovává data uložená v poli *lines*, uvolní místo již zpracovaných dat a volá podfunkce, které data roztřídí do jednotlivých pracovních struktur. Funkce *gps_get_xxxx* poté vloží vybranou zprávu do námi zvolené struktury. Také jsou zde funkce na převod zeměpisné délky/šířky do různých tvarů. Pro zobrazení na LCD se využívá tvaru - stupně, minuty, sekundy k tomu je volána *gps_print_polar*. Ve formátu KML je ale nutno zapisovat zeměpisnou délku/šířku ve tvaru stupňů + jejich desetinná hodnota, proto je volána *gps_norm_pos*.

3.4 lcd.c

V tomto souboru jsou umístěny funkce pro řízení podsvícení, inicializační funkce displeje a jeho základní funkce. Displej je řízen periferií FSMC. Ovládání intenzity podsvícení je zajištěno pomocí PWM modulace, kdy log0 na I/0 pinu znamená rozsvíceno.

3.4.1 Intenzita podsvícení

Inicializace probíhá takto, nejprve je nutno přivést hodiny:

```
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4 , ENABLE);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD |
RCC_APB2Periph_AFIO, ENABLE);
```

Poté inicializovat pin PD13, zde je potřeba ho kompletně remapovat, jelikož bude využit TIM4_CH2:

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOD, &GPIO_InitStructure);
GPIO_PinRemapConfig(GPIO_Remap_TIM4, ENABLE);
```

Následně bude probíhat inicializace časovače:

```
TIM_TimeBaseStructure.TIM_Period = 2000;
TIM_TimeBaseStructure.TIM_Prescaler = 0;
TIM_TimeBaseStructure.TIM_ClockDivision = 0;
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit(TIM4, &TIM_TimeBaseStructure);
```

Čítač je připojen na zdroj hodinového taktu, to z něj vytváří časovač. Perioda je zvolena pro 36 kHz, jelikož frekvence jádra je 72 MHz a bylo nutné ji nastavit na 2000 (72MHz/36kHz = 2000). Předdělička není nastavena - není potřeba nižších frekvencí, čítač počítá nahoru.

Dalším krokem je inicializace samotné PWM modulace:

```
TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_Low;
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = bl_intensity;
TIM_OC2Init(TIM4, &TIM_OCInitStructure);
```

TIM_OCMode nastavuje typ výstupu, *TIM_OCPolarity* charakteristiku pulsu(log0 nebo log1), *TIM.Pulse* délku trvání pulsu (intenzita podsvícení).

3.4.2 Komunikace s LCD pomocí FSMC

Displej obsluhuje FSMC (Flexible Static Memory Controller) periferie, což je ovladač pro externí paměti, např. typu NOR Flash, NAND Flash, ale i SRAM, která je použita jako součást ILI9320 ovládající LCD. Důležité je nastavení časovnání pro čtení:

```
pr.FSMC_AddressSetupTime = (5/T_HCK+1);
pr.FSMC_AddressHoldTime = (5/T_HCK+1);
pr.FSMC_DataSetupTime = (100/T_HCK+1);
pr.FSMC_BusTurnAroundDuration = 0;
pr.FSMC_CLKDivision = 0;
pr.FSMC_DataLatency = 0;
pr.FSMC_AccessMode = FSMC_AccessMode_A;
```

Pro zápis:

```
pw.FSMC_AddressSetupTime = (10/T_HCK+1);
pw.FSMC_AddressHoldTime = (5/T_HCK+1);
pw.FSMC_DataSetupTime = ((20+15)/T_HCK+1);
pw.FSMC_BusTurnAroundDuration = 0;
pw.FSMC_CLKDivision = 0;
pw.FSMC_DataLatency = 0;
pw.FSMC_AccessMode = FSMC_AccessMode_A;
```

Hodnota T_HCK je perioda hodinového taktu procesoru (1/72Mhz = cca 14ns). Konkrétní minimální hodnoty časování zápisu a čtení SRAM displeje jsou uvedeny v dokumentaci [5] na str. 106. Samotná konfigurace FSMC je řešena strukturou *FSMC_NORSRAMInitTypeDef* dle vlastností paměti.

K zápisu(čtení) je využit adresový prostor FSMC_Bank1, náležící SRAM paměti na adrese 0x60000000. Pro nastavení indexu (adresy) registru je PE3 (A19 FSMC) pin (RS pin

na LCD) v log0. Pro přenos dat je adresa nastavena na 0x60100000, tedy pin PE3 (odpovídající 20. bitu adresy paměti) je uveden do log1 při zachování povoleného adresového rozsahu.

```
#define LCDBUS_RSLOW_ADDR   ((vu16 *)0x60000000)
#define LCDBUS_RSHIGH_ADDR  ((vu16 *)(0x60000000 |
(1<<(19+1))))
```

3.4.3 Ovládání LCD

Celá inicializační sekvence LCD je popsána v [6] na str. 28. Nastavení indexu registru a zápis dat je uskutečněn touto funkcí:

```
void LCD_WriteReg(u16 regn, u16 val)
{
     *LCDBUS_RSLOW_ADDR = regn;
     *LCDBUS_RSHIGH_ADDR = val;
}
```

Proměnná regn specifikuje index registru a val jeho hodnotu. Nejnižší HW funkce obsluhy LCD pro STM Graphic Library HAL tvoří:

Nastavení pozice pro vykreslení pixelu na hodnoty Xpos a Ypos pomocí indexů 0x20, 0x21:

```
LCD_SetCursor(Xpos, Ypos)
```

Přepnutí na zápis do obrazové RAM indexem 0x22:

```
LCD_WriteRAM_Prepare()
```

Zapsání barvy pixelu ve tvaru BGR, při opakovaném zápisu je adresa pixelu inkrementována:

```
LCD_WriteRAM(u16 color)
```

Nastavení oblasti vykreslování pixelů - x pozice, y pozice, výška, šířka:

```
LCD_SetDisplayWindow(uint8_t Xpos, uint16_t Ypos, uint8_t
Height, uint16_t Width)
```

Smazání displeje zvolenou barvou:

```
LCD_Clear(uint16_t Color)
```

3.5 touchscreen-hw.c , touchscreen-sw.c

Obsluha dotykové vrstvy je realizována ADC3 převodníkem, kanály 10-13. Vodorovnou/svislou polohu měří vždy dva kanály současně, pak jsou sečteny a vyděleny dvěma pro zprůměrování měření. Změření jednoho kanálu probíhá následovně:

```
ADC_RegularChannelConfig(TS_ADC, ch, 1,
ADC_SampleTime_55Cycles5);
ADC_Cmd(TS_ADC, ENABLE);
while(!ADC_GetFlagStatus(TS_ADC,ADC_FLAG_EOC));
val = ADC_GetConversionValue(TS_ADC);
ADC_Cmd(TS_ADC, DISABLE);
```

Je nutno čekat na dokončení konverze a až poté vložit hodnotu do proměnné. ADC převodník je zapínán a vypínán, jelikož piny jsou přenastaveny a zároveň využity i k detekci dotyku:

```
GPIO_SetOutput(TS_PIN_YD, 0);
GPIO_SetOutput(TS_PIN_YU, 0);
GPIO_SetInputPulledUp(TS_PIN_XL | TS_PIN_XR);
```

Při stisku se na pinech TS_PIN_XL a TS_PIN_XR vytvoří log0, která je detekována funkcí *void tshw_poll(u16 *adc_x, u16 *adc_y, int *pressed)*, zapisující aktuální stav dotyku a souřadnice do proměnných.

Vyšší vrstvu tvoří touchscreen-sw.c, kde jsou funkce pro kalibraci displeje a použití vícenásobného měření pro zpřesnění. Soubory touchscreen-hw.c a touchscreen-sw.c obsahují modifikované části kódu demo programu vývojového kitu.

3.6 STM Graphic Library

Knihovna pochází z dílny ST Microelectronics a je převážně určena pro její řadu mikrokontrolérů STM32Fx. Umožňuje vytvoření GUI s ovládacími prvky, včetně podpory dotykové vrstvy. Je rozdělena na dvě části.

První část, Embedded_GUI_HAL, obsahuje střední funkce, jako tisk znaku, čáry, vykreslení bitmapy, atd. Volá hw funkce souboru lcd.c. Součástí jsou i nižší funkce pro použití touchpadu. Ty ale nejsou použity, jelikož předpokládají použití touchscreenu s A/D převodníkem a komunikaci po SPI rozhraní.

Druhá část, Embedded_GUI_Library, definuje zapouzdřené funkce k řízení celého GUI. Výhodou je, že vytvoření ovládacích prvků i jednotlivých zobrazitelných rámců, je vcelku snadno realizovatelné i bez hlubších procesních znalostí samotné knihovny.

Věškerá dokumentace k této knihovně je součástí přílohy 4.

3.6.1 uiappuser.c

Slouží k definici chování po aktivaci ovládacích prvků. Například akci po zmáčknutí tlačítka přesunu na obrazovku s ovládáním zajišťuje tato funkce:

```
void s1_DesignButton01_Click()
{
   Show_s2();
}
```

3.6.2 uiframework.c

Všechny použité ovládací prvky (jejich rozmístění, názvy, popisky) a stránky jsou vytvářeny zde. Nejvyšší místo v hiearchii zastává funkce *Create_PageObj (GL_Page_TypeDef* pThis)*, což je vlastně vytvoření jedné zobrazované strany na LCD. Ta je tvořena objekty, které jsou na ní umístěny. Takto vypadá část kódu tvořící jednu stránku se zobrazenými geodaty:

```
void Create_s1()
{
   GL_PageControls_TypeDef* DesignButton01=
   NewButton(1,(uint8_t*)"-->",s1_DesignButton01_Click);
   GL_PageControls_TypeDef* LATLabel =
   NewLabel(11,(uint8_t*)"LATITUDE",GL_HORIZONTAL,
   GL_FONT_SMALL,GL_Black);
   Create_PageObj( &s1 );
   AddPageControlObj(34,212,DesignButton01,&s1);
   AddPageControlObj(318,2,LATLabel,&s1);
}
```

Nejprve je vytvořen objekt *DesignButton01*, což je tlačítko k přepnutí na druhou stránku s ovládáním, jeho popisek je --> a funkce popisující chování při zmáčknutí se jmenuje *s1_DesignButton01_Click*. Druhá řádka obsahuje vytvoření objektu textu LATITUDE, který je pojmenován *LATLabel*, jeho parametry tvoří směr (horizontální), velikost fontu (malý) a barva (černá). Následně je vytvořena strana s1. Na posledních řádcích probíhá umístění prvků a jejich přiřazení ke straně. Takto je však vložen statický text. Geodata se dynamicky mění. Pak je nutno vytvořit ještě funkci, která bude zajišťovat obnovu hodnot:

```
void Refresh_s1(void)
{
   if(CurrentScreen == &s1) {
      if (gprmc.status == 'A') {
         Set_Label(&s1,21,(uint8_t*) stringtoformat("%s %c
         ", lat, gprmc.pos.ns));
      }
      else {
         Set_Label(&s1, 21, (uint8_t*) "N/A              ");
      }
      RefreshPageControl(&s1, 21);
   }
}
```

Funkce testuje, zda-li jsme na první straně. Pokud ano a zároveň máme gps fix (zaměření), pak jsou obnoveny údaje o zeměpisné šířce, pokud gps fix nemáme, je zobrazeno N/A. Podfunkce *Set_Label* mění text, jejími parametry jsou adresa strany, identifikační číslo textu a řetězec, který je nutný vytvořit (v tomto případě vlastně dva spojit). Z toho důvodu je zde použita také podfunkce stringtoformat ze souboru help.c.

3.7 sdcard.c

Obsahuje STM knihovnu ovládající paměťovou kartu a využívá periferního SDIO rozhraní MCU. Jsou zde nízké funkce pro FatFS, např. zápis a čtení surových dat, funkce obsluhy přerušení dokončení DMA přenosu a stavu SD karty:

Inicializace SD karty:

```
SD_Error SD_Init(void)
```

Přepnutí na 4bitový paralelní přenos (po inicializaci probíhá 1bitový):

```
SD_EnableWideBusOperation(uint32_t WideMode);
```

Načtení informací o kartě (CSD, CID, typ, velikost):

```
SD_Error SD_GetCardInfo(SD_CardInfo *cardinfo)
```

Datové funkce budou uvedeny v kapitole 3.8 FatFS.

3.8 FatFS

Autorem této knihovny je Elm Chan. Tato knihovna byla vytvořena pro embedded systémy. Je nezávislá na hardwarové architektuře, jelikož má plně oddělenou diskovou I/O vrstvu, tvořenou souborem diskio.c. Jako naprosté minimum pro zajištění základní funkčnosti knihovny bylo potřeba doplnit hal funkce z sdcard.c pro zápis a čtení SD karty:

Zápis:

```
SD_WriteBlock((uint8_t *)(&buff[0]),sector << 9 ,SECTOR_SIZE);
Status = SD_WaitWriteOperation();
while(SD_GetStatus() != SD_TRANSFER_OK);
```

Čtení:

```
SD_ReadBlock((uint8_t *)(&buff[0]),sector << 9 ,SECTOR_SIZE);
Status = SD_WaitWriteOperation();
while(SD_GetStatus() != SD_TRANSFER_OK);
```

Parametry funkce *SD_Writeblock* tvoří adresa začátku pole přenášených dat, adresa na paměťové kartě (zde je bitový posuv o 9 míst doleva, tedy o 1 vyšší hodnota proměnné *sector* bude odpovídat o 512 vyšší adrese) a velikost sektoru (bloku) (512bytů).

SD_WaitWriteOperation() čeká na dokončení DMA přenosu. *SD_GetStatus()* ověřuje, zda-li přenos proběhl v pořádku. Čtení probíhá obdobně, stejně tak i zápis/čtení několika bloků:

```
SD_WriteMultiBlocks((uint8_t *)(&buff[0]),sector <<
9,SECTOR_SIZE,count);
```

Rozdíl je pouze v existenci proměnné *count*, která udává počet bloků k zápisu. Je možné doplnit funkci získávání času, pak jsou soubory označeny časovou známkou.

3.9 filehandler.c

Soubor filehandler.c ošetřuje zapisování a celkové chování logování, např. v případě nedostupnosti GPS signálu nebo vyjmuté SD karty.

3.9.1 Ovládání zápisu souboru

K ovládání zápisu na paměťovou kartu slouží funkce *Logger(FIL *fil, struct gps_gprmc *gprmc1)*. Ta je volána z main.c, řídí vytvoření a zapisování do kml souboru, ověřuje gps fix, zobrazuje počet uložených geodat a celkovou dobu logování. Je rozdělena do tří podfunkcí.

První vytvoří soubor (přiřadí oddíl 0 SD karty) a zapíše hlavičku:

```
void File_print_header(FIL *fil, struct gps_utc_time time1 )
{
   char fname[10];
   FRESULT res1;
   FAT_FILE_SET();
   sprintf(fname,"%02d%02d%02d%02d.kml",gprmc.utc_date.day,
   time1.hour, time1.min,time1.sec);
   res1=f_open(fil,fname, FA_CREATE_ALWAYS | FA_WRITE);
   f_printf(fil,"<?xml version=\"1.0\"
   encoding=\"ANSI\"?>\n");
   .....
}
```

Přiřadí se oddíl 0 za pomoci funkce *FAT_FILE_SET* (obsahující funkci *f_mount* z knihovny FatFS + je alokována paměť funkcí memset). Vytvoří se řetězec složený z data a času, uložený do proměnné fname, který slouží jako název souboru. Pomocí *f_open* (z FatFS) je otevřen soubor (parametry FA_CREATE_ALWAYS a FA_WRITE nastavují, že bude vždy vytvořen nový a bude do něj povolen zápis). Nakonec následuje několik volání *f_printf* zapisujících hlavičku kml souboru.

Druhá zapisuje geodata, obnovuje hodnotu čítače logů a kontroluje, zda-li nastala doba k zápisu podle nastavení obnovení :

```
void File_print_middle(FIL *fil,struct gps_gprmc gprmc1, char
lat[16], char lon[16])
{
   char plat[20],plon[20];
   double lat1;
   double lon1;
   int alt;
   if(timer1 >= refresh) {
     gps_norm_pos(&gprmc.pos, &lat1, &lon1);
     alt=gps_real_int(gpgga.msl_altitude);
     sprintf(plon,"%f",lon1);
     sprintf(plat,"%f",lat1);
     f_printf(fil,"                    %s,",plon);
     f_printf(fil,"%s,",plat);
     f_printf(fil,"%d\n", alt);
     f_sync(fil);
     timer1 = 0;
     counter++;
     Set_Label(&s2, 46,(uint8_t*) stringtoformat("%d
     ",counter));
        if(CurrentScreen == &s2)
           RefreshPageControl(&s2, 46);
     }
}
```

Gps_norm_pos převede souřadnice polohy do desetinného formátu stupňů. *Sprint* vytvoří řetězec z float proměnné *lon* a *lat*, *f_print* je zapíše do souboru. Následně je volána *f_sync* (z FatFS), která okamžitě data zapíše na SD kartu. Nakonec je navýšena hodnota čítače záznamů a vynulován časovač obnovení.

Třetí podfunkce zapíše patičku, ukončí soubor, odpojí oddíl, pokud je logování zastaveno:

```
void File_print_footer (FIL *fil)
{
   ....
   f_printf(fil,"</kml>\n");
   res =  f_close(fil);
   f_mount(NULL, &fs32);
}
```

F_close zaktualizuje a uzavře soubor, *f_mount* odpojí oddíl 0 SD karty (karta může být bezpečně vyjmuta; obě funkce pocházejí z knihovny FatFS).

3.9.2 Testování chyb před povolením logování

Testování chyb před logováním zajišťuje funkce *log_status_check()*. Je volána z main.c. Uvnitř ní probíhá kontrola přítomnosti SD karty (i její inicializace v případě, že je nalezena) a GPS signálu. Také indikuje stav zaznamenávání výpisem na LCD.

3.9.3 RTC přerušení

RTC přerušení nastává po jedné vteřině. *TimeOutCalculate()* (z STM GUI Library) ovládá zhasnutí displeje, pokud není detekován žádný dotyk po dobu 40 vteřin. Proměnná *timer* je použita na časování zápisu geodat do souboru, dle nastavené hodnoty *refresh*. *Elap_time(&eltime)* obnovuje informaci o době logování (elapsed time).

```
void RTC_IRQHandler(void)
{
  TimeOutCalculate();
     if(logging == TRUE) {
          timer1++;
          elap_time(&eltime);
     }
     RTC_ClearITPendingBit(RTC_IT_SEC);
}
```

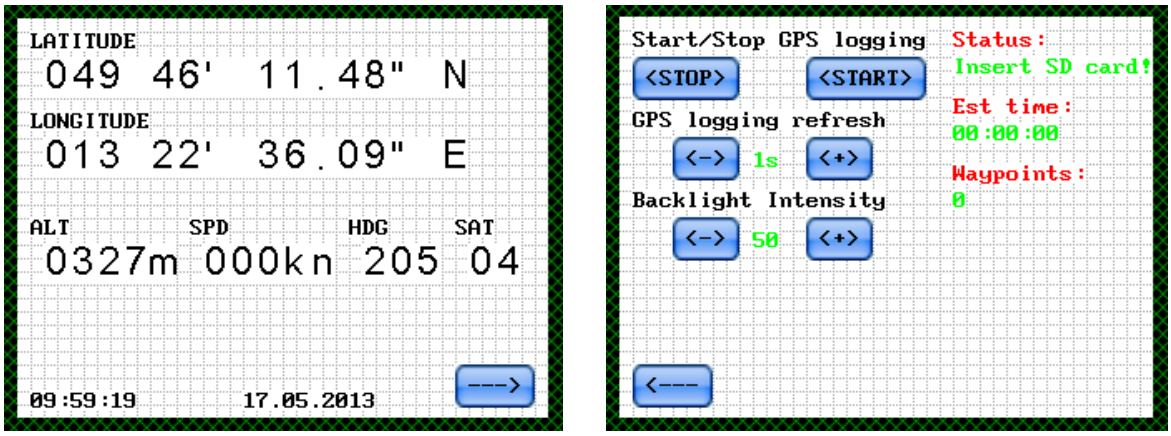
3.10 Formát uložených dat

Cílem bylo zvolit takový formát výstupních dat, aby je bylo možné jednoduše prohlížet v některém z mapových softwarů. Proto byl použit formát KML podporující program Google Earth a webové rozhraní Google mapy. KML je skriptovací jazyk. V tomto případě tvoří parametrickou část programu. Byl vytvořen za účelem zobrazování geodat a od roku 2008 schválen Geoprostorovým konsorciem jako mezinárodní standard.

Po každém spuštení logování je vytvořen soubor ddhhmmss.kml, první dva znaky tvoří den v měsíci a zbylé aktuální čas (hodina, minuta, vteřina). Dále je zapsána hlavička vytvořeného souboru, jejíž součástí jsou použitá znaková sada, název a parametry zobrazení sledovaného úseku. Následují samotné koordináty předcházené typem jejich zobrazení uzavřené patičkou.

3.11 GUI

Ovládání a celkové rozložení bylo navrženo v programu Embedded Resource Editor GUI patřící ke grafické knihovně. Ten je rozdělen na dvě části. Na první (obr. 3-1a) je přehled geodat, čas, datum (pokud GPS není dostupné, je zobrazeno N/A). K přepnutí do druhé části slouží tlačítko --> a zpět tlačítko <--. Druhá část je ovládací (obr. 3-1b), obsahuje povely START/STOP logování, jak často má být poloha zapisována (volitelný rozsah je od 1s do 30min) a intenzitu podsvícení (10 – 90%). Vpravo nahoře se nacházejí informace o stavu (Ready!/Loggging!/Insert SD card!/No GPS lock!), pod nimi uplynulá doba záznamu a jejich počet. Zhasínaní LCD je nastaveno po 40 vteřinách neaktivity dotykové vrstvy.



Obr. 3-1 a) Přehledová strana b) Strana s ovládáním

Závěr

Podařilo se sestavit a naprogramovat plně funkční prototyp GPS loggeru. Soubory uložené na SD kartě, které byly vytvořeny v GPS loggeru pomocí souborového systému FAT, jsou po vložení do čtečky paměťových karet na PC čitelné. Lze je otevřít v mapovém softwaru Google Earth. Zaznamenaná trasa odpovídá skutečnosti. Ovládání a čtení geodat je vzhledem k použití velkého displeje pohodlné. K pohodlnosti ovládání značně přispívá i použitá grafická knihovna, která umožňuje jednoduše vytvořit zobrazené ovládací prvky. Při reálném nasazení by bylo rozumné provést několik následujících úprav.

Je nevhodné testovat události tzv. pollingem, může být zbytečné stále opakovat neměnnou funkci. Například je stále ověřováno, zda-li byl proveden dotyk na LCD, toto by bylo možno provést pomocí přerušení při dotyku. Totéž platí o LCD displeji, který je stále překreslován, ač se zobrazovaná data nezměnila. Tedy je zbytečně využíván výpočetní výkon.

Bylo by vhodné mikrokontrolér uvádět cyklicky mezi intervaly zápisu polohy na paměťovou kartu do některého z režimů se sníženou spotřebou. Tím dojde ke snížení spotřeby a tedy zvýšení životnosti použitého akumulátoru.

Dále by bylo pro tuto aplikaci vhodné použít některý z monochromatických LCD. Nejlépe LCD využívající technologii tekutého inkoustu známou ze čteček elektronických knih, jenž nabízí nejnižší možnou spotřebu. Jeho statická spotřeba je téměr nulová a proud je odebírán jen během překreslování LCD. Toto by vedlo k dalšímu snížení spotřeby. K ovládání by mohla být použita tlačítka nebo joystick.

Napájecí/nabíjecí modul by měl umožnovat indikovat stav nabíjení v kooperaci s mikrokontrolérem, tedy schopnost informování o stavu nabíjení na displeji společně s indikací zbývající kapacity akumulátoru. Orientačně by toto šlo řešit pomocí ADC převodníku.

Název vytvořeného souboru obsahuje pouze identifikaci dne v měsíci a čas, je tedy nutno paměťovou kartu jednou za měsíc zkopírovat/smazat, aby byl rozeznatelný datum zaznamenání.

Výše uvedené úpravy by vedly ke zlepšení výdrže akumulátoru a ke zpříjemnění obsluhy zařízení.

Seznam literatury a informačních zdrojů

[1] Rádiové signály GPS. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-06-03]. Dostupné z: http://cs.wikipedia.org/wiki/R%C3%A1diov%C3%A9_sign%C3%A1ly_GPS

[2] MANLEY, Pat. *Practical navigation for the modern boat owner*. Hoboken, NJ: J. Wiley, c2008, xv, 182 p. ISBN 04-705-1613-5.

[3] *Understanding GPS: principles and applications*. 2nd ed. Editor Elliott D Kaplan, C Hegarty. Boston: Artech House, c2006, xvii, 703 s. ISBN 978-1-58053-894-7.

[4] CYPRESS MICROSYSTEMS. *AN2173 Application Note: Touch Screen Control and Calibration – Four Wire, Resistive*. 2004. Dostupné z: http://www.psocdeveloper.com/uploads/tx_piapappnote/an2173_01.pdf

[5] ILI TECHNOLOGY CORP. *ILI9320 Datasheet*. V0.46. 2007. Dostupné z: http://www.displayfuture.com/Display/datasheet/controller/ILI9320.pdf

[6] ILI TECHNOLOGY CORP. *ILI9320 Application Notes*. V0.92. 2007. Dostupné z: http://www.densitron.com/uploadedFiles/Displays/Support/ILI9320AN_V0.92.pdf

[7] SD GROUP. *SD-Memory Card Specifications: Part 1 Physical Layer Specification*. V1.0. 2000. Dostupné z: http://www.scribd.com/doc/13134577/SD-Memory-Card-Specifications-V10

[8] MICROCHIP TECHNOLOGY INC. *MCP73831/2 Datasheet*. Rev. E. 2008. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/21984e.pdf

[9] TEXAS INSTRUMENTS. *Low Input Voltage Synchronous Boost Converter with 1.3-A Switches TPS61200/ TPS61201/TPS61202*. Rev. 2013. 2013. Dostupné z: http://www.ti.com/lit/ds/symlink/tps61200.pdf

[10] FASTRAX LTD. *NMEA Manual for Fastrax IT500 Series GPS recievers*. Rev. 1.7. 2010. Dostupné z: http://elecfreaks.com/store/download/datasheet/rf/UP501/NMEA%20manual%20for%20Fastrax%20IT500%20Series%20GPS%20receivers_V1.7.pdf

[11] ST MICROELECTRONICS. *RM0041 Reference Manual: STM32F100xx advanced ARM-based 32-bit MCUs*. Rev. 4. 2011. Dostupné z: http://www.st.com/web/en/resource/technical/document/reference_manual/CD00246267.pdf

[12] ST MICROELECTRONICS. *AN3128 Application Note: STM32 embedded graphic objects/touchscreen library*. Rev. 5. 2011. Dostupné z: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application_note/CD00259585.pdf?s_searchtype=keyword