

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MIKROPOČÍTAČ V ULTRALEHKÉM LETADLE/VRTULNÍKU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ADAM ŠIROKÝ

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MIKROPOČÍTAČ V ULTRALEHKÉM LETADLE/VRTULNÍKU

MICROCOMPUTER IN ULTRAMICRO AIRCRAFT/HELICOPTER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ADAM ŠIROKÝ

VEDOUcí PRÁCE

SUPERVISOR

Prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2013

Abstrakt

Tato bakalářská práce se zabývá vývojem vestavěného systému pro ultralehká letadla a vrtulníky. Cílem je navrhnout, vytvořit a otestovat systém, pomocí kterého by bylo možné zaznamenávat sadu leteckých dat. Ta je tvořena GPS souřadnicemi, nadmořskou výškou, rychlostí a náklony ve dvou osách. Systém je vybudován na mikrokontroléru z rodiny MSP430 od firmy Texas Instruments. Čtenář bude v tomto textu seznámen s použitým hardwarem, návrhem systému, jeho realizací, způsobem ověření funkčnosti a možnostmi využití navrženého systému.

Abstract

This thesis deals with the development of embedded system for ultra light airplanes and helicopters. The aim is to suggest, make and test the system with help of which it would be possible to take notes of a flight data set. The set is formed by GPS coordinates, altitude, speed and leans in both axes. The base of the system is created on microcontroller of the MSP430 family by Texas Instruments. A reader will be offered the used hardware, the system suggestion, its realization, the way of function verification and the possibility of using of the suggested system.

Klíčová slova

Mikrokontrolér, MSP430, eZ-430, eZ430-RF2500, Texas Instruments, vestavěný systém, GPS, Leadtek, LR9540G, NMEA, akcelerometr, Kionix, KXTC9, trajektorie letu

Keywords

Microcontroller, MSP430, eZ-430, eZ430-RF2500, Texas Instruments, embedded system, GPS, Leadtek, LR9540G, NMEA, accelerometer, Kionix, KXTC9, flight trajectory

Citace

Adam Široký: Mikropočítač v ultralehkém letadle/vrtulníku, bakalářská práce, Brno, FIT VUT v Brně, 2013

Mikropočítač v ultralehkém letadle/vrtulníku

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana profesora Pavla Zemčíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Adam Široký
9. května 2013

Poděkování

Rád bych zde poděkoval vedoucímu mé bakalářské práce za poskytnutou pomoc při tvorbě tohoto textu, cenné rady z oblasti hardwaru a umožnění otestovat zařízení v ultralehkém letadle a vrtulníku.

© Adam Široký, 2013.

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

Obsah

1 Úvod	4
2 Dosavadní poznání	6
2.1 Mikrokontroléry MSP430	6
2.2 GPS moduly	27
2.3 Akcelerometry	30
2.4 Vybavení letadel letovými zapisovači	32
2.5 Dostupná zařízení pro sběr leteckých dat	32
3 Shrnutí současného stavu a požadavků	34
3.1 Stanovení požadavků	34
3.2 Návrh požadavků	34
3.3 Způsob ukládání dat	36
3.4 Možnosti využití navrhovaného řešení	36
4 Popis systému	37
4.1 Princip činnosti	38
4.2 Použité hardwarové vybavení	39
4.3 Obsluhy přerušení	41
4.4 Způsob zpracování a ukládání dat do paměti	43
4.5 Výstup	44
4.6 Testování	48
4.7 Cenové aspekty zařízení	54
5 Závěr	56
Seznam příloh	60
A Obsah CD	61
B Obsluha přerušení portu P1	62
C Obsluha přerušení portu P2	63
C.1 Příznak P2IGF_bit.P7	63
C.2 Příznak P2IGF_bit.P6	63

Seznam obrázků

2.1	Blokové schéma mikrokontroléru MSP430 (převzato z [12])	7
2.2	Hardwarďská a von Neumannova architektura (převzato z [2])	7
2.3	Blokové schéma modulu BCM+ (převzato z [2])	8
2.4	Závislost hodnot registrů RSELx a DCOx na výslednou frekvenci DCO (převzato z [2])	10
2.5	Vliv operačního režimu na spotřebu elektrického proudu u MSP430x41x (převzato z [4])	12
2.6	Blokové schéma modulu ADC10 u MSP430F20x2 (převzato z [2])	13
2.7	Časování během jednorázového převodu jednoho kanálu u modulu ADC10 (převzato z [13])	16
2.8	Datový rámec kanálu USCLA v režimu UART (převzato z [13])	18
2.9	Blokové schéma časovače Timer_A s jedním kanálem (převzato z [2])	20
2.10	Blokové schéma paměťového modulu flash u MSP430 (převzato z [13])	24
2.11	Organizace paměťového prostoru paměti Flash u MSP430 (převzato z [13])	25
2.12	Obsah balení vývojového kitu eZ430-RF2500 (převzato z [23])	27
2.13	Letový zapisovač GAU 2000 od Appareo Systems (převzato z [19])	33
2.14	Mobilní telefon OpenMoko Neo FreeRunner (převzato z [26])	33
3.1	Návrh požadavků	35
4.1	Očekávaný formát výstupu	37
4.2	Fyzická podoba zařízení	38
4.3	Graf konečného automatu	39
4.4	Ukázka formátu nejlepšího vzorku dat v paměti	44
4.5	Ukázka formátu nejhoršího vzorku dat v paměti s chybou zápisu	44
4.6	Závislost pozice letadla z pohledu pilota na velikostech úhlů v osách x a y	45
4.7	Konfigurace nástroje Putty pro logování virtuálního sériového portu	46
4.8	Parametry skriptu pro rekonstrukci dat	47
4.9	Vizualizace výstupu pomocí Google Earth	47
4.10	Ukázka záznamu ladícího výpisu	48
4.11	Test přesnosti GPS souřadnic z konstantní pozice	49
4.12	Test přesnosti GPS souřadnic za pohybu	50
4.13	Ověření činnosti systému s krátkodobými výpadky GPS signálu	51
4.14	Trajektorie prvního pokusného letu	52
4.15	Trajektorie druhého pokusného letu	52
4.16	Trajektorie třetího pokusného letu	53
4.17	Trajektorie čtvrtého pokusného letu v době přistání a vzletu	53
4.18	Trajektorie čtvrtého pokusného letu v době rolování	54
4.19	Trajektorie čtvrtého pokusného letu	54

B.1	Funkční schéma obsluhy přerušení pro port P1	62
C.1	Funkční schéma obsluhy přerušení pro port P2	65

Kapitola 1

Úvod

Sportovní létání, které se v dnešní době těší velké oblibě, přináší kromě příjemných zážitků i ty nepříjemné v podobě leteckých nehod. Dle statistiky pro Českou republiku se bohužel nedá říci, že by jejich počet rok od roku klesal. Naopak civilní letecká doprava je i přes svou složitost dnes chápána jako jeden z nejbezpečnějších druhů dopravy. Na základě těchto skutečností se nabízí otázka: Proč nedokážeme i u sportovních letadel snížit množství leteckých nehod?

Jedním z důvodů, proč nelze tak efektivně zabráňovat haváriím sportovních letadel, je obtížnější způsob stanovení příčin předchozích leteckých nehod. V civilní letecké dopravě je situace zcela odlišná. U většiny nehod se pověřeným úřadům podaří velmi přesně stanovit příčinu, kterou může být například špatný postup v údržbě letadla, chyba pilotáže nebo dokonce i konstrukční vada samotného letadla. Na základě tohoto zjištění je poté možné pomocí příslušných opatření předcházet podobným nebo totožným nehodám.

Hlavním zdrojem informací o průběhu letu u dopravních letadel je zařízení, které známe pod pojmem černá skříňka. To je konstruováno tak, aby odolalo většině destruktivních účinků pádu letadla. Jedná se o velmi komplexní a drahý přístroj, které například umožňuje záznam trajektorie letu, zvuků na palubě letadla nebo lokalizaci místa nehody. Právě tato všestrannost, a tím pádem i složitost ztěžuje použití tzv. černých skříněk na palubách sportovních letadel. Současně žádný právní předpis nenařizuje sportovním letadlům mít na své palubě zařízení tohoto druhu. Není tedy překvapením, že velká část sportovních letadel není vybavena žádným takovýmto přístrojem.

Někteří výrobci již zareagovali na tuto situaci a začali nabízet odlehčené varianty tzv. černých skříněk. Ty jsou již dostatečně levné a malé na to, aby je bylo možné využívat na palubách sportovních letadel. Dále jsou nabízena velmi podobná zařízení, která slouží pro výuku pilotů. Jsou ještě o něco jednodušší konstrukce a neposkytují ochranu vůči poškození v případě nehody. Lze z nich po každém letu rekonstruovat jeho průběh a ukázat chování pilota v daný okamžik. Vhodným rozbořem cvičného nebo jakéhokoliv jiného letu je možné za včas odhalit případné nedostatky v pilotáži, a tím zabránit opakování se stejného chování, které by již mohlo vést k nehodě. Samozřejmě tato zařízení nemusí sloužit čistě jako bezpečnostní prvek. Lze je úplně stejně využít pro informační účely. Veškerá tato zařízení však pracují na stejném principu, díky čemuž je možné první fáze vývoje sloučit. Samotné přizpůsobení konkrétním potřebám je poté již relativně snadné a rychlé.

Mezi sportovní letadla patří i ultralehká letadla a vrtulníky, a právě pro ně je navrhovaný systém určen. Cílem této práce je navrhnout a odzkoušet zařízení, které by dokázalo zaznamenávat základní sadu leteckých dat. Tu tvoří GPS souřadnice, náklon letadla ve dvou osách, nadmořská výška a rychlost. Řídicím členem systému má být mikropočítač,

který by v kombinaci s několika snímači a pamětí dokázal zaznamenávat co nejdélší časový úsek letu a poskytnout tak prostředek pro analýzu určité části letu. Hlavním účelem této práce není vytvořit systém, který by bylo možné ihned začít komerčně využívat, ale prozkoumat možnosti využití mikropočítače na palubě ultralehkého letadla a vyzkoušet si tvorbu systému od návrhu až po testování a vyhodnocení systému.

Předchozí popis odpovídá popisu vestavěných systémů. Ty mají dnes své místo prakticky ve všech elektronických zařízeních, se kterými se setkáváme. Do budoucna se spíše dá očekávat, že jejich počet bude i nadále stoupat. Na základě toho se jeví velmi perspektivní se s těmito systémy co nejvíce seznámit a osvojit si základní principy vývoje. Obojí je v rámci tohoto zadání možné a řekl bych, že nezbytné. Toto zadání je navíc zajímavé v tom, že nechávalo volnou ruku pro řešitele, který si mohl vybrat účel systému. Jediným kritériem bylo, že se musí jednat o systém, který by byl využitelný na palubě ultralehkého letadla nebo vrtulníku. Osobně jsem tuto podmínku nevnímal jako omezující. Naopak ještě o něco více zvýšila atraktivnost tohoto zadání, jelikož umožňovala aktivní účast při testování přímo na palubě ultralehkého letadla.

Jádro samotné práce je rozděleno do tří kapitol. První obsahuje ty nejdůležitější informace, které bylo nutné nastudovat před zahájením práce na samotném systému. Velká část této kapitoly je věnována popisu hardwarového vybavení zařízení. Konkrétně se jedná o popis mikropočítače a dvou typů senzorů pro sběr samotných dat. Poté následuje krátký pohled na platné právní předpisy této země, které se týkají povinného přístrojového vybavení letadel. Závěr této kapitoly je věnován již existujícím a volně dostupným zařízením podobného charakteru. Náplní druhé kapitoly je stanovení požadavků na systém a způsobu jejich řešení. Navíc je přidána část, která popisuje potenciál zařízení. Třetí kapitola je zaměřena na popis struktury, principu činnosti, možností vizualizace zaznamenaných dat a testování systému.

Kapitola 2

Dosavadní poznání

Cílem této kapitoly je shrnout podstatná fakta, která bylo nutné zohlednit při návrhu a implementaci systému. Vzhledem k rozsahu této práce nebylo možné do této části zahrnout veškerá zařízení a možnosti jejich využití.

Samotná kapitola je rozdělena do pěti podkapitol. První je věnována mikrokontrolérům z rodiny MSP430. Mimo jiné popisuje několik modulů, které jsou svázány s navrhovaným systémem. Tématem druhé podkapitoly je problematika GPS modulů. Za touto podkapitolou následuje další, která nahlíží na akcelerometry a možnosti jejich využití. Poslední dvě podkapitoly jsou zaměřeny na problematiku letových zapisovačů.

2.1 Mikrokontroléry MSP430

MSP430 je označení rodiny mikrokontrolérů od firmy Texas Instruments, která používá šestnáctibitové procesory s redukovanou instrukční sadou. Mikrokontroléry z rodiny MSP430 byly navrženy s požadavkem na co nejnižší příkon. Z tohoto důvodu každý mikrokontrolér z této rodiny obsahuje velmi flexibilní modul hodinových signálů [12]. Podrobněji je tento modul popsán v části 2.1.3.

Dále výrobce udává, že mikrokontroléry MSP430 jsou tzv. signálově smíšené [1], což znamená, že mikrokontroléry umí pracovat jak s digitálními, tak i s analogovými signály. Jedné z možností zpracování analogového signálu je věnována část 2.1.4. Další předností mikrokontrolérů z této rodiny jsou jejich nízké ceny, které začínají již na dvaceti pěti centech za kus [12].

2.1.1 Rozdělení rodiny

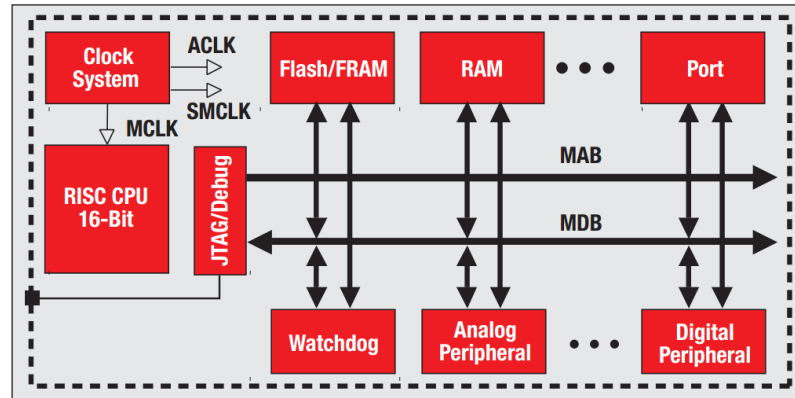
Samotnou rodinu mikrokontrolérů MSP430 výrobce dělí do několika řad, kde každá řada zahrnuje mikrokontroléry s určitou společnou vlastností [2]. Příkladem takové vlastnosti může být typ použité paměti nebo účel, pro který je daná rodina primárně určena.

Příkladem označení jedné konkrétní řady je MSP430F2xx [13]. Z názvu vyplývá, že se jedná o mikrokontroléry z rodiny MSP430 druhé generace, které používají paměť typu Flash [22]. Poslední číslovky, které jsou zde zastoupeny symbolem x, slouží pro určení typu mikrokontroléru v dané rodině. Toto označení ovšem nemusí být konečné. Jednotlivé mikrokontroléry mohou být, a také často jsou, nabízeny ve více variantách. Konkrétní variantu poté označuje několik symbolů, které následují za označením rodiny, řady a typu zařízení. Příkladem přesného označení mikrokontroléru je MSP430F2618ATZQWT-EP. Kompletní

přehled veškerých možných označení, včetně významů jednotlivých symbolů, je dostupný na stránkách výrobce [22].

2.1.2 Architektura

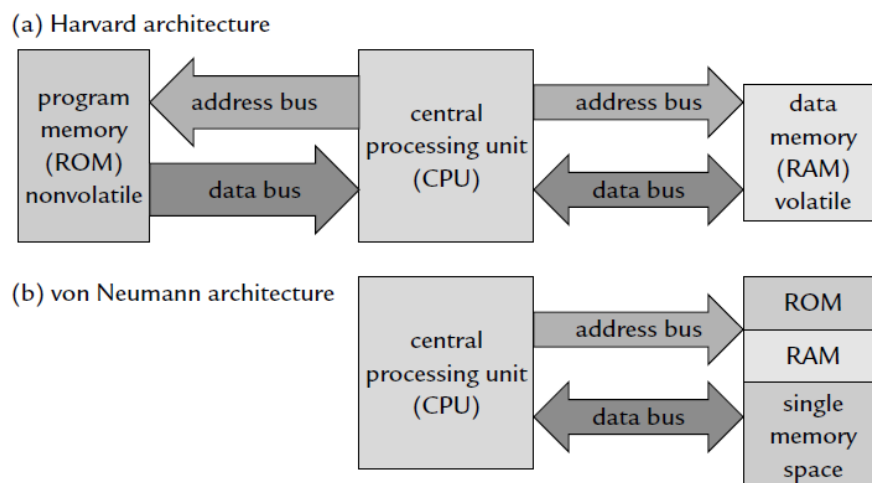
Základní blokové schéma mikrokontrolérů z rodiny MSP430 je na obrázku 2.1.



Obrázek 2.1: Blokové schéma mikrokontroléru MSP430 (převzato z [12])

Na blokovém schématu je vidět, že každý mikrokontrolér z rodiny MSP430 má dvě nezávislé paměti. Ačkoliv toto uspořádání na první pohled připomíná Harwardskou architekturu, mikrokontroléry MSP430 jsou von Neumannovy architektury [2]. Ta v porovnání s Harwardskou архитектурou má společnou paměť pro program a data.

Obě architektury jsou schématicky ukázány na obrázku 2.2. Z tohoto obrázku je vidět, že v případě Harwardské architektury jsou přítomny dvě paměti. Konkrétně se jedná o paměť typu ROM pro program a paměť typu RAM pro data. Každá paměť má vlastní adresovou a datovou sběrnici pro komunikaci s procesorem. Naopak v případě von Neumannovy architektury existuje pouze jeden adresový prostor, se kterým procesor komunikuje pomocí jedné datové a jedné adresové sběrnice.



Obrázek 2.2: Harwardská a von Neumannova architektura (převzato z [2])

Jako hlavní paměť pro program a data slouží u MSP430 paměť typu Flash, případně ROM nebo FRAM. Druhou pamětí je paměť typu RAM, která primárně slouží pro uložení proměnných [2]. Alternativně může paměť RAM obsahovat i část kódu. Tato možnost je využívána v případě, kdy je nutné zajistit, aby mikrokontrolér v době probíhající paměťové operace byl i nadále schopný vykonávat určitou část kódu. Příkladem takovéto situace je zároveň probíhající zápis do paměti a zahájení asynchronní komunikace. V tu dobu nelze zahájit komunikaci, jelikož paměť Flash je zcela vytížena. Přesněji řečeno je vytížen řadič paměti Flash, který provádí paměťovou operaci. CPU v tuto dobu zůstává nezaměstnán. V případě uložení programu v paměti RAM ovšem CPU může nadále vykonávat kód, zatímco řadič paměti Flash provádí zápis do paměti [14].

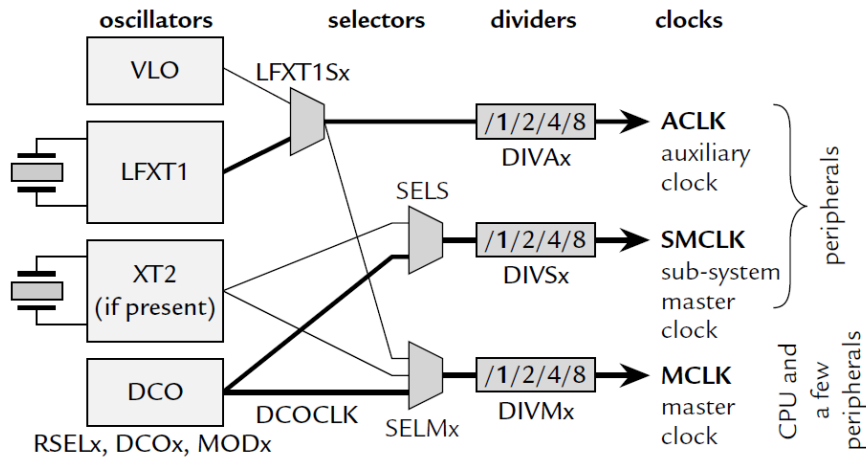
2.1.3 Modul BCM+¹

Každý mikrokontrolér obsahuje hodinový modul, jehož úlohou je konfigurace jednotlivých hodinových signálů. Na obrázku 2.1 je tento modul zastoupen blokem Clock System. Typ hodinového modulu závisí na konkrétním mikrokontroléru. Například u mikrokontrolérů MSP430x2xx se tento modul označuje BCM+ a právě této variantě hodinového modulu je věnována tato část.

Zdrojem informací pro tuto část byly především uživatelské manuály Basic Clock Module, Oscillator and Clock Generator [27], MSP430x2xx Family [13], MSP430 Ultra-Low-Power Microcontrollers [12] a kniha MSP430 Microcontroller Basics [2].

Architektura

Blokové schéma tohoto modulu je na obrázku 2.3.



Obrázek 2.3: Blokové schéma modulu BCM+ (převzato z [2])

Výstupem modulu jsou tři hodinové signály. Modul sám o sobě umožňuje přiřazení oscilátoru, jakožto zdroje hodinového signálu pro jednotlivé výstupní signály. Na blokovém schématu modulu na obrázku 2.3 je tato možnost realizována pomocí multiplexorů. Zároveň každý hodinový signál prochází skrze vlastní děličku hodinového signálu. Ty jsou na blokovém schématu vyobrazeny pomocí bloků, jejichž výplň zobrazuje možné dělitele. Pomocí

¹BCM+ (Basic Clock Module+) - název hodinového modulu u mikrokontrolérů MSP430x2xx

těchto děliček lze například nastavit různou frekvenci dvěma hodinovým signálům, které jsou buzeny stejným zdrojem.

Konfigurace

Ke konfiguraci modulu slouží několik vlastních registrů. Konkrétně se jedná o tři kontrolní registry BCSTLx, registr IFG1 pro příznaky přerušení, registr IE1 pro povolení přerušení a registr DCOCTL pro kalibraci digitálně řízeného oscilátoru.

Hodinové signály

U všech mikrokontrolérů z rodiny MSP430 rozlišujeme tři hodinové signály, které jsou výstupem hodinového modulu. Prvním hodinovým signálem je MCLK², který je výhradně určen pro CPU a některé moduly podporující vyšší frekvence. Dalšími hodinovými signály jsou SMCLK³ a ACLK⁴. Ty jsou určeny především pro jednotlivé moduly. Obvykle platí, že frekvence ACLK je v řádech kilohertz, zatímco SMCLK a MCLK v jednotkách megahertz.

Oscilátory

Zdrojem hodinového signálu je vždy oscilátor. U mikrokontrolérů MSP430 se lze v závislosti na konkrétním zařízení celkem setkat až se dvěma krystalovými a dvěma RC oscilátory.

Jednou z výhod RC oscilátorů je, že nepotřebují žádný další prvek a jsou schopny činnosti v té podobě, ve které jsou uvnitř mikrokontroléru. Oproti tomu krystalové oscilátory vyžadují externě zapojený krystal na piny mikrokontroléru XIN a XOUT. Samotný mikrokontrolér obsahuje totiž pouze obvody pro nastavení a stabilizaci výstupního signálu. Ve většině případů je při připojení externího krystalu ještě nutné připojit externí kondenzátory na oba piny. Z tohoto důvodu byly tyto kondenzátory integrovány již do obvodů oscilátorů mikrokontroléru. Jejich velikost, která závisí na konkrétním zařízení, lze dohledat v podpůrné dokumentaci a následně nastavit pomocí bitů XCPAx z registru BCSTL3.

Z pohledu běžící aplikace jsou především důležité tři vlastnosti oscilátorů. Konkrétně se jedná o přesnost signálu, stabilitu signálu a dobu ustálení frekvence oscilací. Z hlediska přesnosti a stability signálu jsou zcela jistě lepší krystalové oscilátory. Nevýhodou krystalových oscilátorů je ovšem relativně dlouhá doba ustálení frekvence po zahájení činnosti oscilátoru. Tuto zápornou vlastnost je možné vyřešit například vložením zpoždovací smyčky po zahájení činnosti oscilátoru. Je ovšem důležité si uvědomit, že po tuto dobu procesor nevykonává žádnou užitečnou činnost a pouze se pohybuje určitý čas ve smyčce. Konkrétní čas, který je nutný pro ustálení frekvence, je možné najít v dokumentaci pro konkrétní zařízení. V závislosti na zařízení se můžeme setkat s oscilátory:

DCO⁵: RC oscilátor, který je dostupný u všech zařízení z rodiny MSP430. Hlavní výhodou je velmi krátká doba ustálení frekvence po přechodu z úsporného režimu. U řady MSP430F2xx se jedná o přibližně jednu mikrosekundu v případě aktivního DC⁶ generátoru. DC generátor je samostatným celkem určujícím frekvenci výstupního signálu modulu DCO. V případě použití DCO pro MCLK nebo SMCLK je vždy nutné, aby

²MCLK (Main System Clock) - hlavní hodinový signál

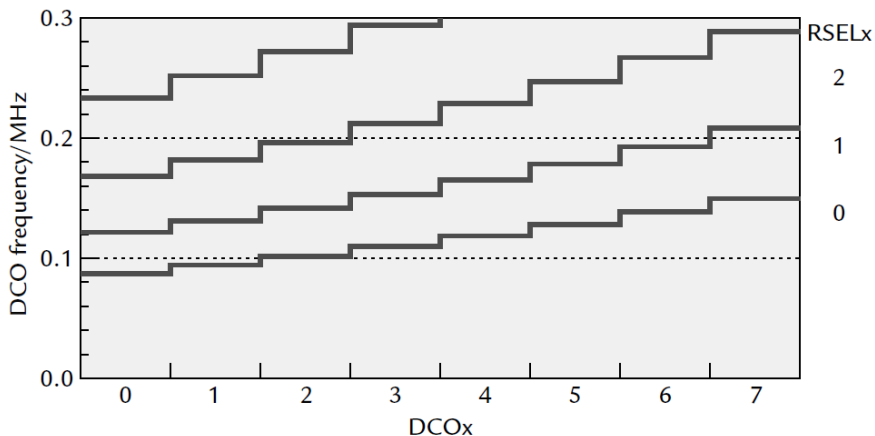
³SMCLK (Sub-system Clock) - sekundární hodinový signál

⁴ACLK (Auxillary Clock) - pomocný hodinový signál

⁵DCO (Digitally Controlled Oscillator) - digitálně řízený oscilátor

⁶DC (Digitally Controlled) - digitálně řízený

byl DC generátor aktivní spolu s DCO. Nevýhodou DCO je poněkud menší přesnost v porovnání s krystalovými oscilátory. Samotná přesnost je ovlivněna mnoha faktory včetně typu použitého zařízení. Například u rodiny MSP430F2xx se při ustálené okolní teplotě a hodnotě napájecího napětí udává přesnost dvě procenta. Výsledná frekvence oscilátoru je dána kombinací bitů RSELx z registru BCSCCTL1 pro nastavení jednoho z šestnácti rozsahů a bity DCOx z registru DCOCTL1 pro nastavení jedné z osmi frekvencí daného rozsahu. Závislost hodnot registrů na výslednou frekvenci je ukázána na obrázku 2.4.



Obrázek 2.4: Závislost hodnot registrů RSELx a DCOx na výslednou frekvenci DCO (převzato z [2])

VLO⁷: RC oscilátor, jenž je dostupný pouze u některých zařízeních. Typicky ho lze nalézt například u většiny zařízeních z řady MSP430F2xx. Frekvenční rozsah 4 - 20 kHz znamená, že tento oscilátor je primárně používán jako zdroj hodinového signálu pro ACLK. Vhodné použití tohoto oscilátoru je především u aplikací, kde je nutné dosáhnout co nejmenší spotřeby zařízení a není vyžadováno úplně přesné časování.

LFXT1⁸: Tento krystalový oscilátor je stejně jako DCO dostupný ve všech zařízeních. Typicky je používán buď s 32 kHz krystalem pro nízké frekvence či vysokofrekvenčním krystalem nad 400 kHz pro vysoké frekvence. Výhodou a důvodem, proč jsou tyto krystaly používány, je jejich vysoká přesnost a stabilita. Stabilitou rozumíme částečnou teplotní nezávislost, kdy nedochází k výrazným změnám stability signálu na základě změny teploty. Přesnost je udávána u krystalových oscilátorů jako maximální možná odchylka mezi dvěma po sobě jdoucími kmity. U LFXT1 je stanovena jako 1 Hz na 10^5 Hz.

XT2⁹: Krystalový oscilátor, který je dostupný pouze v některých zařízeních, je určen pouze pro vysoké frekvence. Jinak je principiálně stejný jako LFXT1.

⁷VLO (Internal Very Low-power, Low-frequency Oscillator) - vnitřní nízkopříkonový a nízkofrekvenční oscilátor

⁸LFXT1 (Low- or High-frequency Crystal Oscillator) - nízko nebo vysokofrekvenční krystalový oscilátor

⁹XT2 (High-frequency Crystal Oscillator) - vysokofrekvenční krystalový oscilátor

Režimy činnosti

Jak již bylo zmíněno v podkapitole 2.1, mikrokontroléry MSP430 byly navrženy tak, aby bylo možné při správném použití dosáhnout co nejnižšího příkonu. Toho je dosaženo tím, že je možné ve vhodný okamžik pozastavit činnosti některých oscilátorů/hodinových signálů mikrokontrolérů. Z tohoto důvodu mikrokontroléry MSP430 disponují několika režimy činnosti, které určují aktivní a neaktivní hodinové signály.

U MSP430 rozlišujeme v závislosti na konkrétním zařízení až osm režimů. Konkrétně je k dispozici jeden aktivní (AM - Active Mode) a sedm úsporných režimů (LPMx - Low Power Mode x). K přechodu aktivního režimu do libovolného úsporného režimu není zapotřebí žádné speciální instrukce. Pouze stačí pomocí běžných instrukcí nastavit příslušnou kombinaci bitů SCGO, SCG1, CPUOFF a OSCOFF ze SR¹⁰. Opětovný návrat z úsporného režimu je možný pouze pomocí přerušení. Z tohoto důvodu je nutné neopomenout v případě přechodu do některého z úsporných režimů povolit i přerušení pomocí bitu pro globální masku přerušení (GIE - General Interrupt Enable).

Kromě LPM3.5 a LPM4.5 jsou všechny níže popsané režimy dostupné ve všech zařízeních z rodiny MSP430. Zároveň u všech režimů s výjimkou těchto dvou zůstává i v případě návratu z úsporného režimu zachován obsah paměti RAM. U mikrokontroléru MSP430 tedy rozlišujeme režimy:

AM: Jedná se o počáteční režim po startu mikrokontroléru, ve kterém jsou aktivní veškeré hodinové signály včetně procesoru. Jediný způsob, jak snížit okamžitý příkon, je snížení napájecího napětí, pokud nám to mikrokontrolér umožní. Je ovšem nutné poté počítat s omezením na maximální frekvenci, na které mikrokontrolér běží.

LPM0: Procesor a MCLK jsou neaktivní. Zbylé hodinové signály zůstávají aktivní. Tento režim je především využíván, pokud není vyžadována činnost procesoru, a je nutné dodávat hodinový signál o vyšší frekvenci, tedy SMCLK, do některých modulů.

LPM1: Tento režim je téměř totožný jako LPM0. Rozdíl spočívá v neaktivním DCO v případě, že neslouží jako zdroj pro SMCLK.

LPM2: Procesor, MCLK a SMCLK jsou neaktivní. Spolu s tím je neaktivní i DCO, avšak DC generátor zůstává aktivní. Díky tomu sice DCO stále spotřebovává energii, avšak dovoluje téměř okamžité zahájení činnosti. ACLK zůstává aktivní.

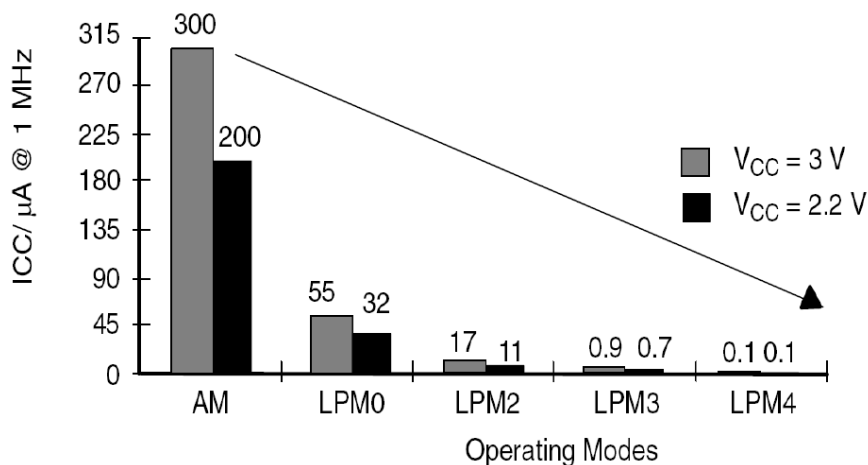
LPM3: Téměř obdobné schéma jako v případě LPM2. Důležitým rozdílem je ovšem neaktivní DC generátor, díky němu DCO nespotebovává energii. Nevýhodou ovšem zůstává časová prodleva při zahájení činnosti DCO (řádově několik mikrosekund).

LPM4: V tomto režimu je vůči LPM3 navíc neaktivní i ACLK. Díky tomu lze mikroprocesor uvést do aktivního režimu pouze pomocí externího přerušení, jelikož veškeré podpůrné obvody, které mohly ve všech předchozích režimech tuto činnost provést, jsou nyní taktéž neaktivní.

LPM3.5 a LPM4.5: Tyto režimy jsou dostupné pouze u zařízení z řady MSP430F5xx. Jak je patrné z názvu, oba jsou pouze rozšířením již představených režimů LPM3 v případě LPM3.5 a LPM4 pro LPM4.5. Důležitou vlastností je, že v těchto dvou úsporných režimech není zachováván obsah paměti RAM.

¹⁰SR (Status Register) - status registr

Příklad rozdílu spotřeby elektrického proudu v závislosti na operačním režimu u mikrokontrolérů řady MSP430x41x je možné zobrazit pomocí obrázku 2.5.



Obrázek 2.5: Vliv operačního režimu na spotřebu elektrického proudu u MSP430x41x (převzato z [4])

2.1.4 Modul ADC10¹¹

Modul ADC10 je jedním z typů analogově digitálního převodníku, se kterým se lze u některých mikrokontrolérů z rodiny MSP430 setkat. Konkrétně se jedná o desetibitový analogově digitální převodník s postupnou aproximací. Stejně jako mikrokontroléry z rodiny MSP430 byl tento modul navržen tak, aby dosahoval co nejnižšího příkonu. Toho je dosaženo pomocí automatické deaktivace jádra a vnitřního oscilátoru převodníku, jakmile není vyžadována jejich činnost. Podrobnější popis obou částí je uveden dále v této části.

Zdrojem informací pro tuto část byl uživatelský manuál MSP430x2xx Family [13], kniha MSP430 Microcontroller Basics [2] a volně dostupné studijní materiály k předmětu Metody vzájemného převodu analogového a digitálního signálu z Fakulty elektrotechniky a komunikačních technologií VUT v Brně [24].

Architektura ADC10

Zjednodušené blokové schéma modulu v podobě, ve které se s ním lze setkat například u mikrokontrolérů z řady MSP430F20x2, je na obrázku 2.6. V této podobě se vyskytuje u velkého počtu mikrokontrolérů z různých řad, kdy jedinou odlišností bývá množství analogových vstupů. Ty jsou vyznačeny jako Ax, kde číslo x udává číslo analogového vstupu. Obecně lze říci, že zařízení vyšších řad mívají větší počet těchto vstupů. Hlavními částmi modulu jsou vzorkovací obvod s pamětí (sample and hold) a desetibitové jádro (SAR core). Oběma je dále věnována samostatná část.

Princip činnosti

Samotný převod, během kterého je vstupní hodnota napětí převedena na desetibitové číslo, lze rozdělit na dvě po sobě jdoucí fáze:

¹¹ADC10 (10-bit Analog-to-Digital Converter) - desetibitový analogově digitální převodník

Referenční napětí

Modul ADC10 potřebuje ke své činnosti dvě napěťové reference. Ty vymezují interval pro metodu půlení intervalů a zároveň i napěťový rozsah, ve kterém se smí nacházet vstupní analogový signál. Horní mez je označována jako V_{r+} a spodní jako V_{r-} .

K výběru zdroje spodní meze V_{r-} slouží bit SREF2 z registru ADC10TLO. Jako defaultní je zvolena zem V_{ss} . Alternativní možností je externí reference V_{ref-} . K nastavení horní meze V_{r+} slouží bity SREF1 a SREF0 z registru ADC10TLO. Defaultní možností je napájecí napětí V_{cc} . Další možností je vnitřní reference V_{ref+} , která se povoluje nastavením bitu REFON. Zde je defaultní hodnotou napětí 1,5 V, kterou lze zvětšit nastavením bitu REF2_5 na 2,5 V. Poslední možností je externí reference V_{ref+} .

Vstup

Každý modul ADC10 v závislosti na konkrétním zařízení obsahuje různé množství analogových vstupů. Z nich je poté pomocí bitů INCHx z registru ADC10CTL1 vybrán jeden, jehož hodnota je převáděna. Jednotlivé vstupy jsou označovány pomocí Ax, kde x značí číslo vstupu od nuly. Jednotlivé vstupy lze rozdělit do dvou skupin:

Všeobecně využitelné vstupy: Tyto vstupy jsou přímo vyvedeny na piny mikrokontroléru a lze na ně přivést libovolný signál. Piny mikrokontroléru jsou ovšem primárně určeny pro digitální signál. Z tohoto důvodu je nutné tyto piny nastavit jako analogové pomocí příslušných bitů z registrů ADC10AE0 a ADC10AE1. Pomocí nich lze ukončit činnost vstupně/výstupních bufferů pro dané piny. V případě vynechání tohoto kroku hrozí možnost vzniku parazitního proudu, který by protékal skrze modul ADC10 na zem a zbytečně zvyšoval spotřebu mikrokontroléru. V případě přivedení externích napěťových referencí V_{ref+} a V_{ref-} do modulu je nutné počítat s menším počtem volně dostupných analogových vstupů. Tyto reference jsou totiž přiváděny na stejné piny mikrokontrolérů jako samotné analogové vstupy. Obvykle se jedná o vstupy A3 a A4.

Interní vstupy: Modul ADC10 obsahuje dva analogové vstupy s předem stanoveným využitím. Na první vstup je přes dělič napětí přivedeno napájecí napětí V_{cc} . Tento vstup slouží pro monitorování napájecího napětí. Druhý vstup je přiveden na teplotní čidlo, které je součástí modulu.

Výstup

Výstupem převodníku je desetibitová hodnota, kterou lze vyjádřit pomocí vztahu:

$$N_{ADC} = \text{nint} \left(1023 * \frac{V_{in} - V_{r-}}{V_{r+} - V_{r-}} \right) \quad (2.1)$$

kde V_{in} je vstupní napětí, V_{r+} horní napěťová mez a V_{r-} spodní napěťová mez. Výsledná hodnota N_{ADC} je celé číslo z rozsahu 0 - 1023.

Na konci převodu je výsledek zapsán do šestnáctibitového registru ADC10MEM v jednom ze dvou formátů, který je zvolen pomocí bitu ADC10DF z registru ADC10CTL1. Výchozím formátem je binární reprezentace čísla v rozmezí 0 - 1023, kde 0 označuje spodní mez rozmezí hodnot na výstupu převodníku. Alternativní variantou je dvojkový doplněk, kde nejvíce významný bit značí znaménko a nejnižších šest bitů je vždy nulových. Hodnota nula poté označuje střední hodnotu rozmezí daného převodníkem.

Vzorkovací obvod

Vzorkovací obvod je tvořen paměťovou a komparační částí. Paměťová část se skládá z paralelně zapojených kondenzátorů, které tvoří kapacitní dělič napětí. Na konci tohoto děliče se nachází invertující komparátor, jehož výstup je přiveden do jádra modulu.

Se vzorkovacím obvodem souvisí pojem vzorkovací čas t_{sample} . Jedná se o čas, který je nutný pro nabití kondenzátorů vzorkovacího obvodu. Vyjádřen je pomocí násobků period hodinového signálu ADC10CLK. Konkrétní hodnotu je nutné nastavit prostřednictvím bitů ADC10SHTx v registru ADC10CTL0. Pomocí kombinace těchto bitů lze nastavit 4, 8, 16 nebo 64 cyklů signálu ADC10CLK. Vzorkovací čas t_{sample} v sekundách je vyjádřen vztahem:

$$t_{sample} > (R_s + R_i) * \ln(2^{11}) * C_i \quad (2.2)$$

kde R_i označuje vnitřní odpor, C_i vnitřní kapacitu a R_s vstupní odpor. Hodnoty R_i a C_i vždy závisí na konkrétním mikrokontroléru a je nutné tyto hodnoty dohledat v dokumentaci pro dané zařízení. Vstupní odpor R_s je odpor zařízení, které je připojeno na analogový vstup.

Pro názornost uvažujme, že: $R_i = 2 \text{ k}\Omega$ (MSP430x2xx), $C_i = 27 \text{ pF}$ (MSP430x2xx) a $R_s = 5 \text{ k}\Omega$ (smyslená hodnota). Perioda hodinového signálu ADC10CLK je $0,2 \mu\text{s}$ ($f_{ADC10CLK} = 5 \text{ MHz}$). Po dosazení těchto hodnot do rovnice 2.2 vychází:

$$t_{sample} > 1,44 \mu\text{s}$$

Při periodě hodinového signálu $0,2 \mu\text{s}$ je tedy potřeba minimálně 7,2 period hodinového signálu pro správné nabití kondenzátorů. V tomto případě je tedy nutné pomocí bitů ADC10SHTx nastavit minimálně 8 cyklů hodinového signálu.

Zahájení převodu

Činnost převodníku je zahájena v případě výskytu nástupné hrany na signálu SHI¹². Zdrojem tohoto signálu může být u modulu ADC10 například bit ADC10SC z ADC10CTL0. Alternativně lze na signál SHI namapovat některý z výstupů časovačů. Důležité je při zahájení konverze nastavit taktéž bit ENC z registru ADC10CTL0. Z tohoto důvodu je vhodné a také doporučené nastavovat bity ADC10SC a ENC z registru ADC10CTL0 v jedné instrukci. Na konci převodu je bit ADC10SC automaticky vynulován a pro zahájení dalšího převodu je nutné ho opět nastavit.

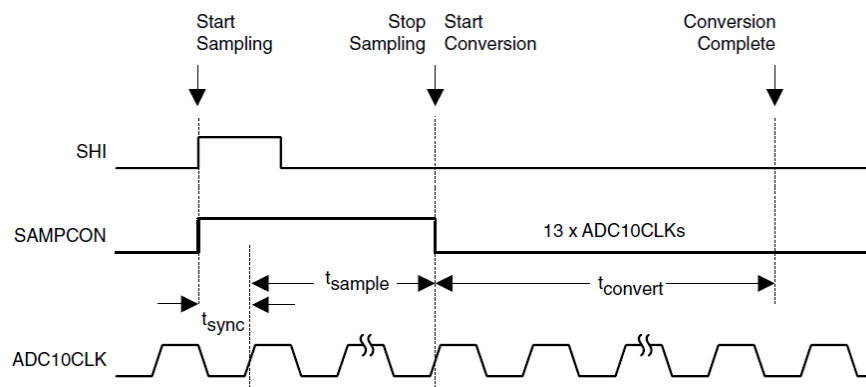
Časování modulu

Průběh převodu je vidět na obrázku 2.7. Na základě tohoto obrázku lze celkovou dobu převodu t_c vyjádřit pomocí vztahu:

$$t_c = t_{sync} + t_{sample} + t_{convert} \quad (2.3)$$

kde t_{sync} je doba nutná k synchronizaci na začátku činnosti převodníku. Tato doba je vždy rovna jednomu hodinovému cyklu signálu ADC10CLK. Druhou složkou je vzorkovací čas t_{sample} , jehož výpočet je uveden v části **Vzorkovací obvod**. Poslední složka $t_{convert}$ je čas, po který je prováděn samotný převod. V případě modulu ADC10 se jedná o třináct hodinových cyklů, během kterých je proveden převod a výsledek překopírován do registru ADC10MEM.

¹²SHI (Sample Input Signal) - vstupní vzorkovací signál



Obrázek 2.7: Časování během jednorázového převodu jednoho kanálu u modulu ADC10 (převzato z [13])

Řadič datového přenosu

Jedním z vybavení modulu ADC10 je i DTC¹³, který umožňuje automaticky distribuovat výsledek převodu do dalších paměťových buněk mikrokontroléru. Činnost DTC je povolena při nenulové hodnotě registru ADC10DTC1. V takovémto případě je ihned po dokončení převodu a nakopírování výsledku do registru ADC10MEM zahájen datový přenos do paměťové buňky na cílové adrese. Cílová adresa je dána obsahem registru ADC10SA a je nutné ji nastavit při konfiguraci DTC. Samotná konfigurace DTC je možná, pouze pokud není modul ADC10 zaneprázdněn. Tato kontrola není prováděna automaticky, ale je čistě v režii programátora.

Přerušení

S modulem ADC10 je spojen právě jeden vektor přerušení. Při neaktivním DTC je příznak ADC10IFG nastaven po skončení zápisu výsledku do registru ADC10MEM. S aktivním DTC je ADC10IFG nastaven až po ukončení veškerého nastaveného přenosu. V případě nastavení bitů ADC10IE v registru ADC10CTL0 a GIE ze SR je nastavením ADC10IFG vyvolána žádost o přerušení. ADC10IFG je vymazán automaticky s vyřízením žádosti o přerušení. Alternativně to lze provést přímo v registru ADC10CTL0 pomocí běžných instrukcí.

Režimy převodu

Modul ADC10 poskytuje celkem čtyři režimy převodu. Výběr režimu je prováděn pomocí bitů CONSEQx z registru ADC10CTL1. Konkrétně se jedná o tyto režimy:

Jednorázový převod jednoho kanálu: Pomocí bitu INCHx z registru ADC10CTL1 je vybrán jeden z kanálů, jehož hodnota je navzorkována a převedena. Na konci převodu je nastaven bit ADC10IFG a výsledek je vložen do registru ADC10MEM.

Opakující se převod jednoho kanálu: Stejně jako u předchozího režimu je vybrán jeden kanál, který je neustále převáděn a výsledek ukládán do registru ADC10MEM. Po ukončení každého převodu je nastaven bit ADC10IFG.

¹³DTC (Data Transfer Controller) - řadič datového přenosu

Jednorázový převod více kanálů: Pomocí bitu INCHx je určen kanál, od kterého je převod zahájen. Postupuje se sekvenčně až po kanál A0. Každý kanál je převeden pouze jednou. Na konci každého převodu je nastaven bit ADC10IFG a výsledek uložen do registru ADC10MEM. Celý převod končí po ukončení převodu kanálu A0.

Opakující se převod více kanálů: Skupina kanálů stejně jako v předchozím režimu je převáděna jednotlivě až po kanál A0. Celý proces je automaticky opakován znovu od kanálu daného bitem INCHx. Na konci každého převodu je nastaven bit ADC10IFG a výsledek uložen do registru ADC10MEM.

Rozlišovací schopnost

Rozlišovací schopnost převodníku ovlivňuje velikost výsledného čísla v bitech a rozsah napětí, ve kterém je prováděno porovnávání během převodu. Zatímco první parametr je daný použitým modulem, druhý je možné efektivně měnit. Jak již bylo zmíněno v části **Referenční napětí**, rozsah napětí je určen pomocí V_{r+} a V_{r-} . Vzhledem k použité metodě pulení intervalu lze s jistotou říci, že se zmenšujícím se rozsahem se zvyšuje rozlišovací schopnost převodníku.

2.1.5 Modul USCI¹⁴

Modul USCI umožňuje asynchronní i synchronní komunikaci s perifériemi. Vychází ze staršího modulu USART¹⁵, který po příchodu modulu USCI ztratil své místo a dnes se s ním již nesetkáme.

Každý modul USCI obsahuje alespoň dva asynchronní kanály USCI_An a dva synchronní kanály USCI_Bn. Symbol n je zástupný symbol pro číslo kanálu. Každý kanál má dva osmibitové kontrolní registry označované UCAnCTL0, ICAAnCTL1 pro kanál USCI_An a UCBICTL0, ICBICTL1 pro kanál USCI_Bn. Většina této části je věnována právě asynchronnímu kanálu USCI_A.

Zdrojem informací pro tuto část byl především uživatelský manuál MSP430x2xx Family [13] a kniha MSP430 Microcontroller Basics [2].

Kanál USCI_A

Kanál USCI_A může sloužit jako univerzální asynchronní přijímač/vysílač (UART - Universal Asynchronous Receiver/Transmitter). Dále kanál USCI_A podporuje infračervený port a synchronní sériové rozhraní SPI. Výskyt synchronního rozhraní SPI v asynchronním kanálu je vcelku matoucí. Toto rozhraní je ovšem natolik využíváno, že bylo přidáno i do tohoto kanálu.

Kanál USCI_B

Kanál USCI_B je čistě synchronní kanál, který podporuje rozhraní SPI a I^2C jak v roli master, tak i slave.

¹⁴USCI (Universal Serial Communication Interface) - univerzální modul pro realizaci sériové komunikace

¹⁵USART (Universal Synchronous/Asynchronous Receiver/Transmitter) - modul pro realizaci synchronně/asynchronní sériové komunikace

Kanál USCI_A v režimu UART

Režim UART pro kanál USCI_A je nastaven smazáním bitu UCSYNC z registru UCAnCTL0. V tomto režimu je poté mikrokontrolér propojen s periférií skrze bity UCAnRXD a UCAnTXD pro příjem a vysílání. Oba tyto bity jsou uvnitř připojeny na osmibitové posuvné registry.

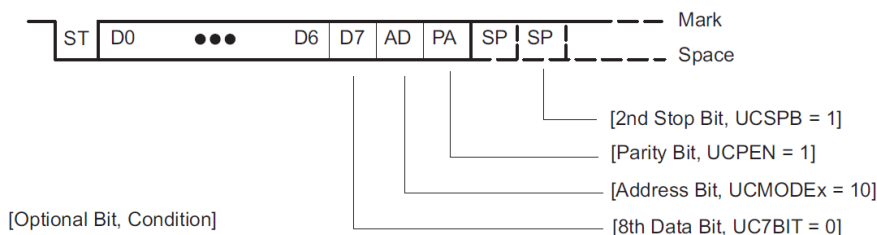
Přenášet je možné sedm nebo osm bitů od nejméně či nejvíce významného bitu. Ty mohou být zakončeny sudou nebo lichou paritou. Zároveň podporuje kanál USCI_A dva protokoly pro víceprocesorové systémy, automatické probuzení z úsporného režimu, sadu stavových příznaků a nezávislé vektory přerušení pro příjem i odesílání.

Konfigurace kanálu USCI_A

Konfiguraci kanálu USCI_A je doporučeno začít nastavením bitu UCSWRST z registru UCAnCTL1. Po nastavení tohoto bitu dojde k resetování hodnot většiny registrů, jež jsou svázány s kanálem USCI_A. Následně je potřeba inicializovat hodnoty registrů kanálu USCI_A, provést konfiguraci portů mikrokontroléru a smazat bit UCSWRST. Po těchto krocích je kanál USCI_A v počátečním stavu a připraven k použití.

Datový rámeček kanálu USCI_A v režimu UART

Formát každého přeneseného rámce přes USCI_A v režimu UART je znázorněn pomocí obrázku 2.8.



Obrázek 2.8: Datový rámeček kanálu USCI_A v režimu UART (převzato z [13])

Každý rámeček se skládá minimálně ze start bitu (ST), sedmi datových bitů (Dx) a jednoho stop bitu (SP). Volitelně je možné přidat osmý datový bit, druhý stop bit, paritní bit (PA) a adresní bit.

Přerušení

USCI modul obsahuje jeden vektor přerušení pro zápis a jeden pro odesílání. Oba vektory jsou sdíleny mezi oběma kanály modulu. Vektor přerušení pro příjem je tedy společný pro kanály USCI_An a USCI_Bn. To samé lze říci i o vektoru přerušení pro odesílání.

Modulační rychlost

Přenosová rychlost asynchronní komunikace je udávána pomocí jednotky modulační rychlosti baud (Bd). Pro funkční komunikaci musí platit, že rychlosti na straně vysílací a přijímací musí být téměř shodné, a pokud možno co nejpřesněji nastaveny. V případě asynchronní komunikace lze tedy říci, že modulační rychlost je číselně rovna přenosové rychlosti,

která určuje počet přenesených bitů za sekundu. Na základě toho je frekvence, při níž jsou jednotlivé bity přijímány/odesílány, označována f_{baund} .

Generátor hodinového signálu kanálu USCI_A

Jako vstupní hodinový signál BRCLK pro USCI_A lze pomocí bitů UCSSELx z registru UCAnCTL1 zvolit UCLK¹⁶, ACLK nebo SMCLK. Tento signál je veden skrze frekvenční děličku s modulátorem, která umožňuje nastavit i neceločíselný dělicí poměr.

Samotné nastavení děličky s modulátorem je prováděno pomocí registrů UCA0BR0, UCA0BR1, UCBRFn a UCBRSn. Jejich obsah pro běžně používané hodnoty frekvence vstupního hodinového signálu a přenosové rychlosti lze nalézt v uživatelském manuálu pro danou řadu mikrokontrolérů. V případě nenalezení požadované kombinace frekvence vstupního hodinového signálu a přenosové rychlosti je v tom samém manuálu velmi podrobný postup stanovení potřebných hodnot registrů. Výstupní hodinový signál z děličky o frekvenci blížíci se f_{baund} je označován BITCLK.

Příjem zprávy přes USCI_A v UART režimu

Na sestupnou hranu signálu UCAnRXD přechází USCI_A do stavu pro příjem dat. Generátor signálu BITCLK zahájí svou činnost a čeká na výskyt validního start bitu. Při nenalezení tohoto bitu přechází modul opět do počátečního stavu a generátor signálu BITCLK je opět uveden do nečinnosti. Při rozpoznání validního start bitu je zahájen příjem dat. V každé periodě signálu BITCLK je přijat právě jeden bit. Po dokončení přenosu dat je výsledek uložen do registru UCAnRXBUF a nastaven příznak UCAnRXIFG. Přerušeni je vyvoláno, pokud jsou nastaveny bity UCAnRXIE a GIE. Po přečtení registru UCAnRXBUF je příznak UCAnRXIFG automaticky smazán.

Odesílání zprávy přes USCI_A v UART režimu

Po zápisu hodnoty do registru UCAnTXBUF přechází modul do stavu pro odesílání dat. Generátor signálu BITCLK zahájí svou činnost a s každým impulzem je odeslán právě jeden bit. Po uvolnění registru je nastaven příznak UCAnTXIFG, který značí, že je registr UCAnTXBUF připraven pro vložení nové hodnoty, která má být odeslána. S nastavením bitů UCAnTXIE a GIE je vyvoláno přerušeni. Po dokončení odesílání je zkontrolována hodnota registru UCAnTXBUF. Pokud byla vložena nová data, je celý proces opakován i pro tuto novou hodnotu z registru UCAnTXBUF. V případě nevložení nové hodnoty do registru je modul automaticky po dokončení předešlého odesílání uveden do počátečního stavu, kdy je připraven pro příjem či odesílání nových dat.

USCI modul a úsporné režimy

Jednou z výhod USCI modulu je automatická aktivace hodinového signálu SMCLK v případě, že je tento signál zvolen jako vstupní hodinový signál pro modul USCI a zároveň je tento hodinový signál neaktivní. Po návratu modulu do počátečního stavu je hodinový signál SMCLK opět uveden do neaktivního stavu.

¹⁶UCLK (Universal Clock) - univerzální hodinový signál

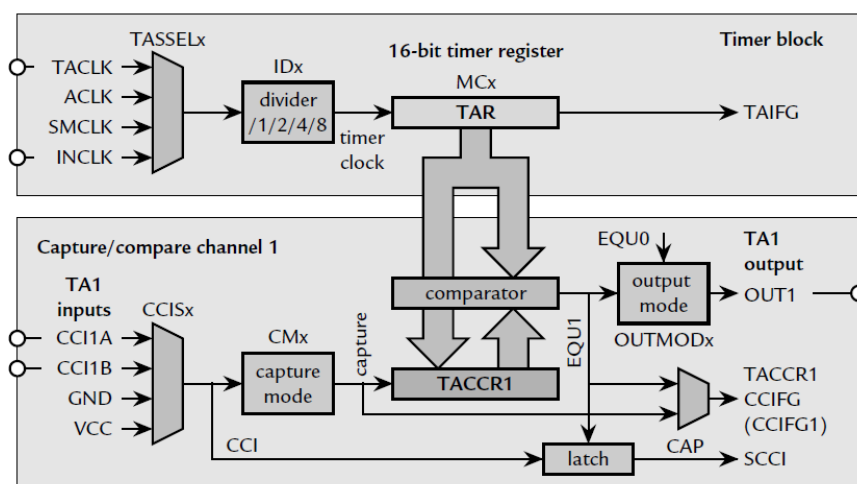
2.1.6 Modul Timer_A

Tato část popisuje časovač Timer_A, který je označován jako jeden z nejuniverzálnějších modulů, se kterým se lze setkat u všech zařízení z rodiny MSP430. Zdrojem informací pro tuto část byla kniha MSP430 Microcontroller Basics [2] a uživatelské manuály MSP430x2xx Family [13] a Using the Timer_A UART Library [17].

Architektura

Jádrum časovače je šestnáctibitový registr TAR¹⁷. Na jeden z jeho vstupů je přiveden hodinový signál, který na základě zvoleného režimu činnosti inkrementuje či dekrementuje hodnotu tohoto registru. Výstupem samotného registru je zjednodušeně řečeno příznak přerušování TAIFG z registru TACTL.

Dále časovač v závislosti na konkrétním zařízení obsahuje alespoň dva kanály pro záchyt hrany vstupního signálu a porovnávání hodnot registrů. Počet kanálů lze velmi často odvodit přímo z názvu modulu. Například název Time_A2 značí přítomnost dvou takovýchto kanálů. Zjednodušené blokové schéma časovače Timer_A s jedním kanálem je na obrázku 2.9.



Obrázek 2.9: Blokové schéma časovače Timer_A s jedním kanálem (převzato z [2])

Konfigurace

Vzhledem k univerzálnosti modulu je zapotřebí většího množství konfiguračních registrů. U každé verze tohoto čítače nalezneme kontrolní registr TACTL, vnitřní registr čítače TAR a registr vektorů přerušování TAIV. Dále je zde vždy pro každý kanál čítače pár registrů TACCTLn a TACCRn, kde symbol n značí číslo kanálu od nuly.

Hodinový signál

Jako zdroj hodinového signálu je možné pomocí bitů TASSELx z registru TACTL vybrat jeden ze čtyř hodinových signálů. Ten je poté přes děličku hodinového signálu přiveden na vstup registru časovače. Samotný dělicí poměr je nastavován pomocí bitů IDx z registru TACTL.

¹⁷TAR (Timer_A Register) - registr časovače Timer_A

Zdrojem hodinového signálu mohou být dva již známé interní hodinové signály SMCLK a ACLK. Dále je možné namapovat do modulu dva externí hodinové signály TACLK a INCLK, jejichž zapojení vždy závisí na použitém zařízení.

Režimy činnosti

Časovač se vždy nachází na základě kombinace bitů MCx z registru TACTL v jednom ze čtyř režimů. Poslední dva níže zmíněné režimy potřebují ke své činnosti registr TACCR0, který náleží kanálu nula. Ten je primárně určen právě pro tyto dva režimy. V případě zvolení jednoho z těchto dvou režimů nelze kanál dále využívat. Jinak je možné využít kanál nula stejně jako například kanál jedna. Jednotlivé režimy činnosti:

Stop: Během tohoto režimu je činnost časovače pozastavena. Hodnoty všech registrů zůstávají po celou dobu tohoto režimu zachovány.

Nepřetržité čítání: Časovač neustále čítá vzestupně v plném rozsahu registru TAR. Hodnota registru TAR nabývá hodnot 0 až 65 535. Příznak TAIFG je nastaven s přetečením hodnoty registru časovače zpět na nulu.

Čítání nahoru: Hodnota registru časovače je postupně inkrementována od nuly až do hodnoty, která je shodná s hodnotou registru TACCR0. Při shodě hodnot obou registrů je nastaven příznak CCIFG z registru TACCTL0. V následujícím hodinovém taktu je hodnota registru časovače opět nulována a čítání probíhá znovu. Perioda čítání je rovna hodnotě registru TACCR0 plus jedna. Příznak TAIFG je nastaven s nulováním hodnoty registru časovače.

Čítání nahoru/dolů: Čítání je provedeno tentokrát dvakrát za jednu periodu časovače. Nejprve opět do hodnoty udané registrem TACCR0 a poté zpět do hodnoty nula. Perioda čítání je rovna dvojnásobku hodnoty registru TACCR0. Příznak TAIFG je nastaven při čítání hodnoty časovače z hodnoty jedna na nulu. Příznak CCIFG je nastaven v případě čítání z hodnoty udané registrem TACCR0 minus jedna na hodnotu TACCRO.

Kanály časovače

Každý kanál smí pracovat v jednom ze dvou režimů činnosti. Konkrétně se jedná o režim záchyty hrany a režim porovnávání. Konkrétní režim je zvolen pomocí bitu CAP z registru TACCTLn.

Režim záchyty hrany

Podstata činnosti časovače s využitím kanálu v tomto režimu spočívá v tom, že v případě výskytu vybrané události na vstupním signálu je překopírována aktuální hodnota registru TAR do registru TACCRn. Zároveň je v době výskytu události nastaven příznak CCIFG registru TACCRn. Při nastavení bitu CCIE registru TACCRn dojde k vyvolání obsluhy přerušení. V té by mělo dojít k přečtení hodnoty z registru TACCRn tak, aby tato hodnota byla vyčtena ještě před tím, než se vyskytne nová událost. Pokud byla hodnota v registru TACCRn nahrazena novou hodnotou registru TAR před přečtením staré hodnoty, dochází k nastavení bitu COV z registru TACCRn.

Vstupní signál, na kterém je zachytávána událost, je vybírán pomocí bitů CCISn z registru TACCTLx. Událostí může být sestupná hrana, nástupná hrana nebo kombinace obou.

Jako vstupní signál je možné přivést signály CCInA, CCInB, VCC a GND. První dva jmenované jsou externí a lze je namapovat přímo na piny mikrokontroléru pomocí nastavení příslušných portů. Poslední dva jmenované signály tvoří napájecí napětí mikrokontrolérů.

Tento režim je možné využít například pro zahájení činnosti některého modulu na základě vnější události na vstupním signálu. Dalším příkladem použití je odměření časového intervalu mezi jednotlivými událostmi na základě rozdílů hodnot registru TAR v době událostí.

Režim porovnávání

V tomto režimu je v každém hodinovém cyklu porovnávána hodnota registru TAR s hodnotou registru TACCRn. Při shodě dojde k nastavení příznaku CCIFG, nastavení vnitřního signálu EQU_n a namapování vnitřního signálu CCI na signál SCCI. Nastavení signálu EQU_n vede ke změně výstupního signálu OUT_n na základě zvoleného výstupního režimu.

Jednotlivých režimů je k dispozici celkem osm. Konkrétní režim je nastavován pomocí bitů OUTMOD_x z registru TACCTL_n. Jedním z nich je výstupní režim, kdy hodnota výstupního signálu OUT_n přímo odpovídá hodnotě vnitřního signálu EQU_n. V tomto režimu tedy slouží čítač jako zdroj digitálního signálu, který je možné využít například pro generování PWM¹⁸ signálu. Další režim umožňuje nastavení hodnoty signálu OUT_n pokaždé, když je nastaven příznak CCIFG. Velmi podobně se chová ještě jeden režim, který ovšem hodnotu signálu nastaví na logickou hodnotu nula.

Přerušeni

S modulem Timer_A jsou svázány dva vektory přerušeni. První slouží výhradně pro kanál nula a má nejvyšší prioritu v rámci tohoto modulu. Příznak přerušeni je smazán automaticky při obsluze. Druhý vektor je vyhrazen příznakům zbývajících kanálů a příznaku přetečení registru TAR. Priorita přerušeni od jednotlivých kanálů klesá s rostoucím číslem kanálu. Nejnižší prioritu má přerušeni, které je svázáno s příznakem TAIFG. Pro zjištění zdroje přerušeni slouží hodnota registru TAIV. Jakýkoliv přístup do tohoto registru zajistí vymazání příznaku přerušeni, které má aktuálně nejvyšší prioritu. Po ukončení obsluhy je automaticky testována hodnota registru. Při nenulové hodnotě je opět volána obsluha těchto přerušeni.

Využití časovače pro řízení asynchronní komunikace

Samotný modul je vzhledem k několika režimům činnosti velmi univerzální. Příklady jednoduchého užití byly uvedeny u jednotlivých režimů. Poněkud složitějším příkladem využití modulu je softwarové řešení rozhraní UART právě pomocí modulu Timer_A. Jedná se tedy o náhradu modulu USCI_A v režimu UART, který byl představen v části 2.1.5.

Základní ideou je využití jednoho nebo více kanálů časovače nejprve v režimu záchytu hrany pro zahájení příjmu datového rámce a poté v režimu porovnávání pro vzorkování a odesílání jednotlivých bitů rámce. Díky tomu lze pomocí vhodně zvolené hodnoty registru TACCR_n a kmitočtu hodinového signálu modulu přesně nastavit přenosovou rychlost, na které dochází ke komunikaci.

Při implementaci pomocí jednoho kanálu, který řídí jak odesílání, tak příjem, lze dosáhnout pouze spojení typu half-duplex (poloviční duplex). V případě nutnosti spojení typu full-duplex (plný duplex) je nutné využít dvou kanálů.

¹⁸PWM (Pulse Width Modulation) - pulzně šířková modulace

V současné době existují minimálně dvě knihovny pro mikrokontroléry MSP430, které se zabývají implementací tohoto rozhraní pomocí modulu `Timer_A`. Obě poskytují plně duplexní komunikaci a možnost nastavení přenosové rychlosti.

První knihovna nese název `Using the Timer_A UART Library`. Tato knihovna je volně dostupná na stránkách firmy Texas Instruments [28] spolu s uživatelským manuálem, který popisuje jednotlivé funkce a způsob použití knihovny. Spolu se zdrojovými kódy jednotlivých funkcí v jazyce assembler jsou dostupné i zdrojové kódy ukazující několik způsobů použití a inicializace celé knihovny. Mimo jiné knihovna obsahuje i hlavičkové soubory, pomocí kterých lze knihovnu použít i přímo z jazyka C.

Druhá knihovna byla publikována v rámci knihy `MSP430 Microcontroller Basics` [2] od Johna H. Daviese. Knihovna je vytvořena pro mikrokontrolér MSP430F2013 v jazyce C a podporuje stejně jako předchozí plně duplexní komunikaci. Po drobných úpravách ji lze použít téměř na libovolné zařízení, které obsahuje modul `Timer_A` s alespoň dvěma kanály. Veškeré informace o možnostech a principu činnosti knihovny jsou dostupné v již zmíněné knize `MSP430 Microcontroller Basics` [2].

2.1.7 Flash paměť

Značná část mikrokontrolérů z rodiny MSP430 má hlavní paměť typu Flash. Ta zde zastupuje nevolatilní datové úložiště jak pro kód, tak i pro data. Hlavní předností je možnost měnit její obsah i za běhu CPU. Díky tomu lze například přeprogramovat mikrokontrolér. U MSP430 lze adresovat jednotlivé bity, bajty nebo slova. Tato část popisuje základní charakteristiku a jednotlivé operace s pamětí Flash u mikrokontroléru z rodiny MSP430.

Zdrojem informací pro tuto část byl uživatelský manuál `MSP430 Flash Self-Programming Technique` [14], tutoriál `MSP430 Microcontroller Tutorial` od Gustava Litovského [11] a uživatelský manuál `MSP430x2xx Family` [13].

Architektura

Paměťový modul Flash u MSP430 obsahuje integrovaný řadič, který řídí veškeré paměťové operace. Řadič obsahuje čtyři řídicí registry `FCTLx` pro čtení a zápis, generátor hodinového signálu (`Timing Generator`) a generátor napětí (`Programming Voltage Generator`). Kromě řadiče modul dále obsahuje samotnou paměť a další podpůrnou logiku pro zpracování adresy a dat. Blokové schéma paměťového modulu Flash je na obrázku 2.10.

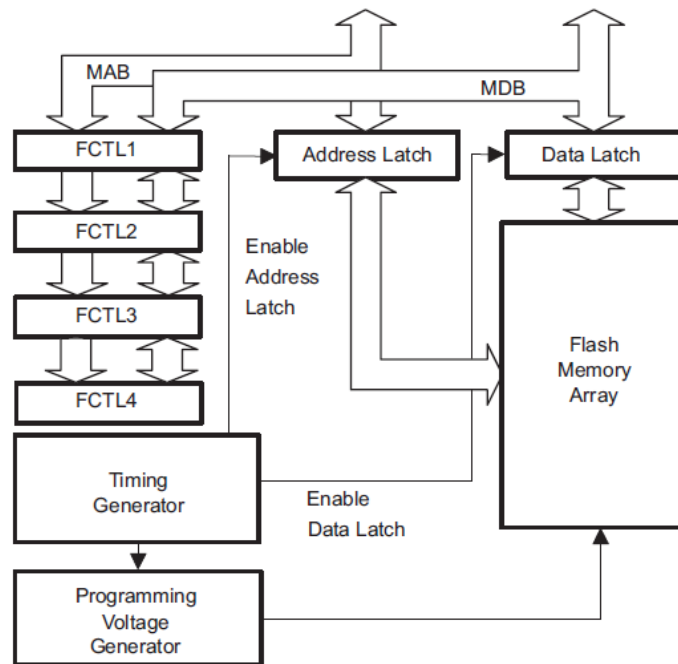
Organizace paměti

Paměťový prostor je rozdělen na hlavní a informační paměť. Hlavní paměť slouží pro ukládání kódu a dat. Informační paměť je určena pro uložení kalibračních konstant, případně je možné ji využít stejně jako hlavní paměť.

Hlavní paměť je dále rozdělena na jeden nebo více segmentů po 512 bajtech. Každý segment je dále rozdělen na osm bloků po 64 bajtech. Informační paměť je tvořena segmenty A-D po 64 bajtech. Příklad organizace paměti je vyjádřen pomocí obrázku 2.11.

Segment A

Segment A informační paměti je primárně určen pro kalibrační konstanty. Z tohoto důvodu je možné tento segment uzamknout nastavením bitu `LOCKA` z registru `FCTL3`, čímž je zne-



Obrázek 2.10: Blokové schéma paměťového modulu flash u MSP430 (převzato z [13])

možněn jakýkoliv druh zápisu do tohoto segmentu. Zároveň jsou zbylé segmenty informační paměti vynechány při hromadném mazání paměti.

V případě vymazání bitu LOCKA je možné se segmentem A pracovat stejně jako s ostatními segmenty informační paměti, které jsou smazány spolu se segmenty hlavní paměti při hromadném mazání. Operace pro zápis, čtení a mazání jsou naprosto shodné pro informační i hlavní paměť.

Konfigurace

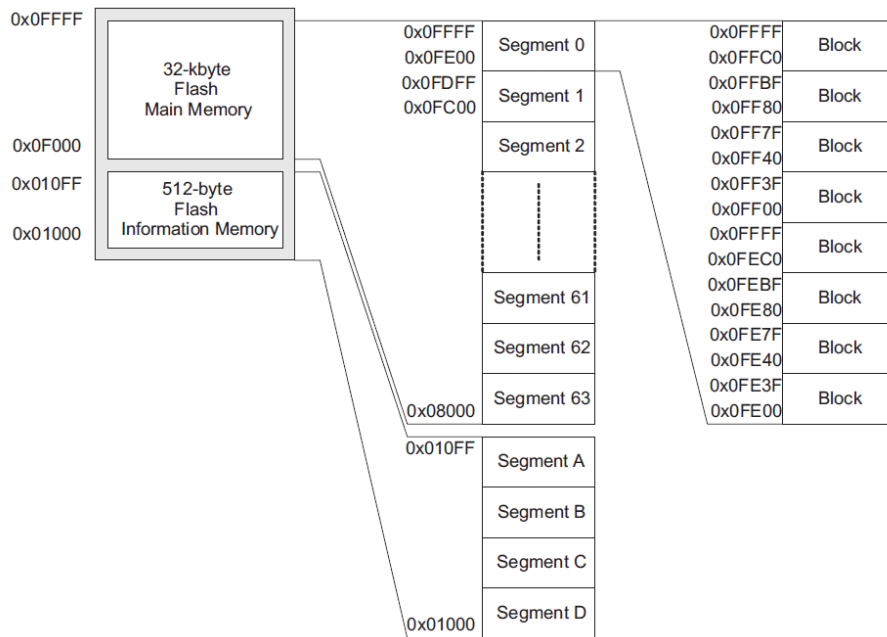
Ke konfiguraci řadiče paměti slouží již zmíněné registry FCTLx. Ty jsou z důvodu zabezpečení chráněny heslem. Každý přístup k těmto registrům musí začínat zadáním hesla FWKEY, které je zpřístupní. V opačném případě dojde k nastavení příznaku KEYV z registru FCTL3 a softwarovému resetu zařízení.

Generátor hodinového signálu

Operace pro zápis a mazání vyžadují hodinový signál o frekvenci 257 kHz až 476 kHz. Z tohoto důvodu je součástí řadiče paměti i generátor hodinového signálu pro paměť Flash, který umožňuje nastavit jak zdroj hodinového signálu, tak i dělicí poměr tohoto signálu.

Pomocí bitů FSSELx z registru FCTL2 je možné vybrat jako zdroj hodinového signálu ACLK, MCLK nebo SMCLK. Následně lze pro dosažení vyžadované frekvence zvolit vhodný dělicí poměr pomocí bitů FNx z registru FCTL2.

Při zápisu nebo mazání a současně snaze uvést procesor do úsporného režimu je zdroj hodinového signálu ponechán aktivní až do konce operace. Díky tomu je zaručeno, že operace s pamětí proběhne celá až do konce tak, jak byla naplánována.



Obrázek 2.11: Organizace paměťového prostoru paměti Flash u MSP430 (převzato z [13])

Princip práce s pamětí Flash

Každá paměťová buňka může nabývat vždy pouze hodnoty logická jedna nebo logická nula. Zápis hodnoty logická jedna je možný pouze pomocí operace mazání. Hodnotu logická nula je možné zapsat pomocí operace pro zápis.

Čtení z paměti Flash

Ve výchozím stavu je paměť připravena pro čtení. V tomto stavu se paměť Flash chová naprosto stejně jako paměť ROM, kdy z ní lze pouze vyčíst aktuální hodnoty. Generátory hodinového signálu a napětí jsou neaktivní.

Zápis do paměti Flash

Zápis do paměti je možný v několika režimech. Konkrétně je možné pomocí bitů LKWRT a WRT z registru FCTL1 zvolit zápis bajtu, slova, případně celého bloku. Ve všech režimech jsou na začátku zápisu uvedeny v činnost generátory napětí a hodinového signálu. Po ukončení operace jsou oba generátory opět uvedeny do nečinného stavu.

Zápis bajtu/slova do paměti Flash

Zápis bajtu či slova je možný v době vykonávání programu z paměti Flash i RAM. Při vykonávání programu z paměti Flash je veškeré časování v roli řadiče paměti. Vykonávání kódu procesorem je pozastaveno až do doby, kdy je dokončena paměťová operace.

V době vykonávání kódu z paměti RAM je nutné před samotným zahájením paměťové operace otestovat, zda není nastaven bit BUSY z registru FCTL3. Tento bit značí stav generátoru hodinového signálu. V případě nastavení bitu BUSY je generátor hodinového signálu zaneprázdněn a zahájení operace zápisu v tuto dobu by vedlo k nastavení příznaku

ACCVIFG z registru FCTL3 a výsledek operace zápisu by byl nepředvídatelný. Test na nulovou hodnotu bitu BUSY je nutný i na konci operace zápisu. Při nenulové hodnotě bitu je zakázáno dokončit operaci zápisu a dále vykonávat kód.

Blokový zápis do paměti Flash

Zápis bloku je možný, pouze pokud je kód vykonáván z paměti RAM. Hodnota napětí nastavená generátorem je konstantní po dobu zápisu celého bloku, tedy 64 bajtů. Tento režim je proto výhodný z pohledu doby zápisu, zejména pokud je nutné zapisovat větší množství dat do paměti.

Po dobu zápisu je nastaven bit BUSY, který je automaticky smazán po dokončení celého procesu. Bit WAIT z registru FCTL3 je nastaven mezi zápisem jednotlivých bajtů/slov, kde jeho nastavení slouží pro zahájení zápisu dalšího bajtu/slova.

Mazání paměti Flash

Smazání paměti znamená uvést všechny buňky dané paměti do stavu, kdy nesou logickou hodnotu jedna. Nejmenší úsek paměti, který je možné smazat, je segment.

Pomocí bitů MERAS a ERAS z registru FCTL1 je vybrán jeden z několika režimů mazání paměti. Konkrétně je možné smazat právě jeden segment pomocí nastavení bitu ERAS nebo všechny segmenty hlavní paměti pomocí nastavení bitu MERAS. Smazání obou částí paměti je provedeno, pokud jsou nastaveny oba bity.

Samotný proces mazání je realizován pomocí tzv. fiktivního zápisu. Ten uvede v činnost generátory hodinového signálu a napájecího napětí a zahájí samotný proces mazání. Po celou dobu mazání je nastaven bit BUSY, který je po ukončení automaticky nulován.

Stejně jako u zápisu bajtu/slova je nutné rozlišovat, odkud je vykonáván kód. V době vykonávání programu z paměti Flash je vykonávání kódu pozastaveno až do doby, než je dokončeno mazání. Časování je opět čistě v roli řadiče paměťového modulu. Při vykonávání kódu z paměti RAM je nutné stejně jako u zápisu přidat před zahájením a před ukončením paměťové operace testovací smyčku na nulovou hodnotu bitu BUSY.

Přerušení

Řadič paměťového modulu má dva příznaky přerušení. Jedná se o již zmíněné bity KEYV a ACCVIFG, které jsou spojeny s neoprávněným přístupem do paměti.

Přerušení, jež je svázáno s příznakem ACCVIFG, patří do skupiny nemaskovatelných přerušení. To znamená, že není nutné nastavení bitu GIE pro vyvolání obslužné rutiny přerušení. Jediný způsob, jak potlačit vyvolání obslužné rutiny, je smazání bitu ACCVIE z registru IE1. V případě nastavení příznaku KEYV dojde k již zmíněnému softwarovému resetu zařízení.

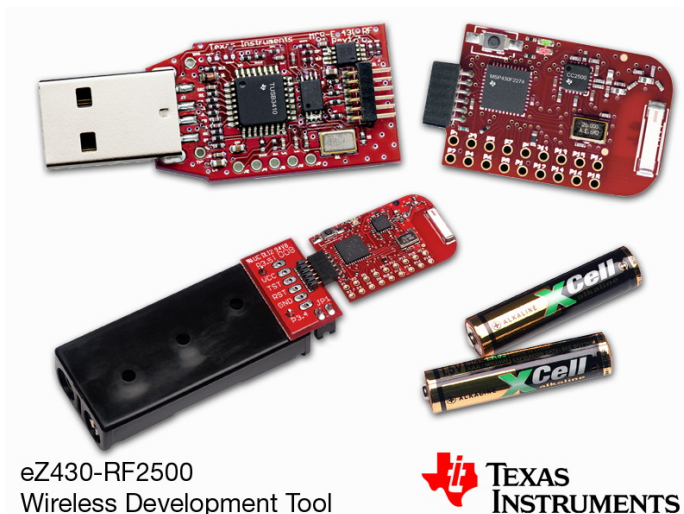
2.1.8 Vývojové prostředky eZ430

Firma Texas Instruments nabízí několik vývojových prostředků z řady eZ430. Každý vývojový prostředek této řady představuje kompletní řešení pro vývoj aplikace na konkrétním typu mikrokontroléru. Obsah balení libovolného kitu z této řady se skládá minimálně z USB programátoru/debuggeru, cílové desky osazené mikrokontrolérem a CD obsahujícího technickou dokumentaci a trial verzi vývojového prostředí.

Zdrojem informací pro tuto část byly informace poskytnuté výrobcem na jeho webových stránkách [23] a uživatelský manuál eZ430-RF2500 Development Tool [3].

eZ430-RF2500

Kit eZ430-RF2500 na obrázku 2.12 je jedním z vývojových kitů z řady eZ430. Balení obsahuje USB programátor/debugger eZ430-RF, dvě cílové desky eZ430-RF2500T, napájecí modul na baterie AAA a CD obsahující dokumentaci, trial verzi vývojového prostředí a vzorovou aplikaci.



Obrázek 2.12: Obsah balení vývojového kitu eZ430-RF2500 (převzato z [23])

Cílová deska eZ430-RF2500T je osazena VF vysílačem/přijímačem CC2500 a mikrokontrolérem MSP430F2274, který mimo jiné disponuje 32 kB pamětí Flash, 1 kB pamětí RAM a moduly USCI, ADC10 a Timer_A3. Dále je součástí eZ430-RF2500T mikrospínač, dvě svítivé diody (LED), osmnáct volně dostupných pinů mikrokontroléru připravených pro usazení patice a šestipinové rozhraní pro komunikaci s eZ430-RF. Šestipinový konektor se skládá ze dvou pinů pro napájecí napětí a zem, dvou pinů pro datový vstup a výstup rozhraní UART a dvou pinů pro rozhraní Spy-Bi-Wire.

Samotný mikrokontrolér na cílové desce je možné napájet buď přes USB programátor/debugger, nebo pomocí přiloženého modulu pro dvě mikrotužkové baterie. Kit je primárně určen pro vývoj aplikací používajících bezdrátový přenos, avšak je možné ho využít pro libovolnou aplikaci na MSP430F2274.

V případě napájení mikrokontroléru přes USB programátor/debugger je možné využít komunikační kanál rozhraní UART mezi stanicí a mikrokontrolérem na cílové desce. Pomocí něho je možné v reálném čase například přenášet informace z teplotního čidla umístěného v mikrokontroléru nebo přenést obsah paměti mikrokontroléru.

2.2 GPS moduly

V současné době jsou GPS moduly hojně využívány například pro určování polohy nebo velmi přesnou časovou synchronizaci zařízení. Jádrem každého modulu je GPS čip, který obsahuje obvody pro příjem a zpracování družicového signálu. Oproti masovému rozšíření

GPS modulů existuje pouze několik architektur a výrobců samotných GPS čipů. Příkladem může být architektura SiRFstarII od společnosti SIRF. Ta spadá do celé série GPS čipů SiRFstar různých generací. Dnes je na trhu již pátá generace těchto čipů, avšak v mnoha zařízeních se stále setkáváme s čipy druhé, případně třetí generace.

Tato část si klade za cíl představit jeden z komerčně používaných GPS modulů. Spolu s ním je značná část prostoru věnována protokolu NMEA 0183 včetně jeho rozšíření od firmy SIRF.

Zdrojem informací pro tuto část byla především dokumentace od firmy Leadtek [9, 10] a SIRF[15]. Dále byly použity informace dostupné na stránkách asociace National Marine Electronics Association [25] a výstup projektu GPS Receiver od Tomáše Götthanse [5].

2.2.1 Leadtek LR9540G

Jedná se o GPS modul od firmy Leadtek, který je osazen GPS čipem LR9805T architektury SiRFstarII s firmwarem SiRFXTrac. Samotná komunikace s modulem probíhá pomocí rozšířené verze protokolu NMEA 0183 od firmy SiRF skrze dvacetipinový konektor modulu. Napěťové úrovně odpovídají logice TTL. Dále modul obsahuje standardní SMA konektor pro připojení aktivní antény.

Maximální frekvence výstupu je jeden herz. Operační teplotní rozsah je od -40 do 85°C . Spotřeba elektrického proudu v aktivním režimu je 32 mA. Hodnota TTFF¹⁹ je vyjádřena pomocí tří hodnot, kdy každá závisí na konkrétním stavu zařízení. V případě horkého startu se jedná v průměru o 4 sekundy. U teplého startu je to již okolo 35 sekund a u studeného dokonce přibližně 45 sekund.

2.2.2 NMEA 0183

Protokol NMEA 0183 pochází od americké asociace National Marine Electronics Association. Tento protokol definuje požadavky na formát přenášených dat, způsob zapojení a samotný přenos dat skrze sériovou linku.

Specifikaci protokolu nelze v plném rozsahu získat bezplatně, je nutné ji zakoupit od asociace National Marine Electronics Association. Cena se aktuálně pohybuje v rozmezí od 150 do 525 dolarů. Výrobci GPS čipů ovšem na tuto skutečnost zareagovali a v rámci svých produktů nabízejí alespoň částečnou specifikaci protokolu. Například firma SIRF s distribucí svých GPS čipů poskytuje popis několika vybraných vět, které jejich čipy podporují. Zároveň existuje na internetu několik neoficiálních verzí, které ovšem nelze brát jako důvěryhodné zdroje.

Zapojení sériového portu

Protokol NMEA 0183 mimo jiné specifikuje způsob zapojení sériové linky. Konkrétně popisuje dvě varianty možného zapojení. Obě využívají napěťové úrovně logiky TTL.

První varianta využívá jeden pár vodičů pro odesílání a druhý pár vodičů pro příjem. Jednotlivé signály jsou vždy v rámci páru v protifázi. Přijímací strana poté vyhodnocuje jejich napěťový rozdíl. Toto zapojení pomáhá odstranit vliv součtového rušení. Z tohoto důvodu obsahuje například LR9540G právě dva kanály A a B pro odesílání a příjem pomocí sériové linky.

Druhou variantou je zapojení pouze pomocí jednoho kanálu, kdy pro příjem a odesílání je vyhrazen vždy jeden vodič. Hodnota těchto signálů je poté vždy vyhodnocována vůči

¹⁹TTFF (Time To First Fix) - doba získání první polohy po zapnutí

zemi, která by měla být pro obě komunikující strany společná. Toto zapojení je méně odolné vůči rušení, avšak umožňuje poněkud jednodušší variantu zapojení.

Datový rámeček

Protokol NMEA 0183 je založen na sériové asynchronní komunikaci. Z tohoto důvodu je každý datový rámeček tvořen start bitem, osmi datovými bity a jedním stop bitem. Přenosová rychlost je typicky 4800 baudů, avšak v případě modulu LR9540G ji lze zvýšit na 9600, 19200, 38400 nebo 57600 baudů.

Formát přenášených dat

Samotné informace z družice jsou organizovány do vět. Ty jsou tvořeny několika ASCII znaky, které jsou poté postupně po jednom přenášeny z modulu pomocí sériové linky v datové části rámečku. Každá věta je ohraničena znakem '\$' na začátku a znaky konce řádku 'CR' a 'LF' na konci. Jednotlivé informace včetně identifikátoru zprávy za počátečním znakem jsou odděleny čárkou. Standard NMEA 0183 definuje mnoho druhů vět, jejichž názvy jsou odvozeny od typu informací, které dané věty nesou.

Výstupní NMEA věty podporované čipem LR9805ST

Firmware, kterým je vybaven čip LR9805ST, podporuje věty typu:

GGA²⁰: Tato zpráva nese informace o pozici GPS přijímače. Konkrétně z ní lze vyčíst hodnotu koordinovaného světového času, GPS souřadnice, počet satelitů, ze kterých byly tyto hodnoty vypočítány, hodnotu parametru HDOP²¹, nadmořskou výšku v metrech vůči elipsoidu dle standardu WGS 84²², čas v sekundách od poslední aktualizace DGPS²³ a identifikační číslo referenční stanice pro DGPS.

GSA²⁴: Tato věta primárně slouží pro sdělení informací o aktivních satelitech prostřednictvím identifikátorů jednotlivých satelitů. Dále věta obsahuje hodnoty parametrů DOP v metrech a informaci o tom, zda je aktuálně možné stanovit dvoj nebo trojrozměrné souřadnice. Taktéž se dozvídáme, zda má nebo nemá GPS čip povolenou automatickou detekci počtu dimenzí souřadnic na základě počtu satelitů.

GSV²⁵: Tyto věty nesou podrobnější informace o jednotlivých viditelných družicích. Vzhledem k množství informací a maximální možné délce jednotlivých vět bývají tyto informace zpravidla rozděleny do více očíslovaných GSV vět. Každá věta pojme informace až o čtyřech družicích. Konkrétně se tedy kromě sekvenčního čísla a celkového počtu GSV vět dozvídáme sadu čtyř hodnot pro každou viditelnou družici. Sada hodnot pro každou družici se skládá z identifikátoru družice, úhlové výšky družice ve stupních, azimutu polohy družice ve stupních a odstupu signálu od šumu v decibelech.

²⁰GGA (Global Positioning System Fixed Data) - věta standardu NMEA 0183 obsahující čas, polohu GPS a fixní data

²¹HDOP (Horizontal Dilution Of Precision) - horizontální rozptyl přesnosti

²²WGS 84 (World Geodetic System 1984) - geodetický standard vydaný v roce 1984

²³DGPS (Differential Global Positioning System) - diferenciatální GPS

²⁴GSA (Global Navigation Satellite System Dilution of Precision and Active Satellites) - věta standardu NMEA 0183 obsahující informace o GPS signálu

²⁵GSV (Global Navigation Satellite System Satellites in View) - věta standardu NMEA 0183 obsahující informace o aktivních satelitech

RMC²⁶: Tato věta obsahuje minimální sadu informací, které jsou doporučeny pro činnost navigace. Stejně jako věta GGA nese hodnotu koordinovaného světového času a GPS souřadnice. Mimo to zpráva obsahuje příznak validnosti poskytnutých informací, rychlost vůči zemi v uzlech a kurz pohybu ve stupních. Dále věta obsahuje datum a hodnotu magnetické deklinace ve stupních. Ta ovšem není podporována GPS čipy od firmy SiRF a toto pole zůstává nevyplněno.

VTG²⁷: Poslední věta typu VTG nese informace o rychlosti a směru pohybu. Konkrétně se lze z věty dozvědět aktuální kurz ve stupních včetně příznaku platnosti. Dále je zde vyjádřen magnetický kurz a horizontální rychlost v uzlech a kilometrech za hodinu. U GPS čipů od firmy SiRF jsou veškeré informace ohledně kurzu vyjádřeny pomocí WGS 84.

Konfigurace GPS čipu SiRFstar pomocí vstupních NMEA vět

Firma SIRF vytvořila rozšíření standardu NMEA 0183 o několik vět, které slouží ke konfiguraci samotného GPS čipu. Konkrétně se jedná o věty PSRF100 až PSRF200, kdy formát vět je stejný jako v případě ostatních vět standardu NMEA 0183. Příkladem jsou:

PSRF100: Tato věta slouží k nastavení vlastností sériového portu A a komunikačního protokolu GPS čipu. Pomocí ní lze nastavit formát rámce, přenosovou rychlost a použitý protokol. Vždy záleží na konkrétní verzi GPS čipu a jeho verzi firmwaru.

PSRF103: Pomocí této věty je možné konfigurovat výstupní NMEA věty, které jsou zasílány z GPS čipu. Pro jednotlivé podporované výstupní věty lze nastavit frekvenci, při níž jsou samotné věty odesílány. Nejvyšší možná frekvence je jeden herz. Naopak nastavením nulové hodnoty frekvence lze zasílání příslušné věty zcela zakázat. Zároveň je možné si touto větou jednorázově vyžádat konkrétní výstupní větu. Tato věta poté slouží čistě jako dotazovací.

2.3 Akcelerometry

Název akcelerometr označuje senzor, pomocí kterého lze měřit statické nebo dynamické zrychlení. Jedním z možných využití je určení náklonu akcelerometru vůči zemskému povrchu na základě měření zemské gravitace. Dalším možným využitím akcelerometru je například měření vybračí nebo setrvačných sil.

Akcelerometr je tvořen integrovaným obvodem, který se skládá ze samotného měřicího ústrojí a několika podpůrných obvodů. Množství a složitost těchto obvodů vždy záleží na konkrétním akcelerometru. Na základě principu činnosti a účelu rozdělujeme akcelerometry do několika kategorií. Příkladem jsou kapacitní, piezoelektrické nebo tepelné akcelerometry.

Zdrojem informací pro tuto část byla technická dokumentace od firmy Kionix [7, 8, 21] a volně dostupné studijní materiály k předmětu Sensorové systémy z Katedry mikroelektroniky FEL ČVUT v Praze [20].

²⁶RMC (Recommended Minimum Specific Global Navigation Satellite System Data) - věta standardu NMEA 0183 obsahující minimální doporučenou sadu hodnot pro navigační přístroje

²⁷VTG (Course Over Ground and Ground Speed) - věta standardu NMEA 0183 obsahující informace kurzu a rychlosti vůči zemskému povrchu

2.3.1 Kapacitní akcelerometry

Kapacitní akcelerometry jsou založeny na principu změny kapacity v závislosti na zrychlení. Výsledná kapacita v daný okamžik je poté pomocí ASIC²⁸ transformována na výstupní analogové napětí. Počet samostatných měřících členů, a tím pádem i výstupních hodnot analogového napětí akcelerometru, je závislý na počtu snímaných os. Například u tříosého akcelerometru dostáváme tři hodnoty analogového napětí, kdy každá reprezentuje právě jednu osu.

Formát výstupu

V závislosti na konkrétním akcelerometru se lze setkat s různou formou výstupu. Například některé akcelerometry od firmy Kionix poskytují analogové nebo digitální rozhraní SPI a I²C.

Parametry

Kromě formátu výstupu a počtu os jsou dalšími důležitými parametry citlivost, dynamický rozsah, teplotní rozsah, napájecí napětí a spotřeba elektrického proudu.

Dynamický rozsah kapacitních akcelerometrů je vyjádřen pomocí kladného a záporného násobku hodnoty tíhového zrychlení. Při překročení tohoto rozsahu hrozí mechanické poškození akcelerometru. Citlivost je udávána pomocí hodnoty napětí na hodnotu tíhového zrychlení. Tato hodnota je vždy svázána s dynamickým rozsahem, kdy platí, že čím větší rozsah, tím menší citlivost. Spotřeba elektrického proudu závisí na konkrétním zařízení a případně i na konkrétním režimu činnosti. Například akcelerometry Kionix z řady KXTC9 mají typicky spotřebu 240 μ A v operačním módu a v pohotovostním pouze 5 μ A.

2.3.2 Kionix EZ430-C9

Stejně jako výrobci mikrokontrolérů i výrobci akcelerometrů nabízejí různé vývojové kity. Jedním z nich je kit EZ430-C9 od firmy Kionix, který obsahuje akcelerometr KXTC9-2050. Tento kit je primárně určen pro vývojový prostředek eZ430-F2013 z řady eZ430 od firmy Texas Instruments. Tato řada vývojových prostředků mikrokontrolérů již byla představena v části 2.1.8. Alternativně ovšem lze zcela bez problému použít kit EZ430-C9 ve spojení s jakýmkoliv mikrokontrolérem, který umožňuje přivést na piny kitu napájecí napětí a zem. Dále musí mikrokontrolér obsahovat potřebné množství analogových vstupů pro jednotlivé měřené osy.

Kionix KXTC9-2050

Jedná se o tříosý kapacitní akcelerometr s napájecím napětím 3,3 V. Výstup je možný pouze pomocí analogové hodnoty napětí pro každou osu zvlášť. Dynamický rozsah akcelerometru je $\pm 2,0$ g. Citlivost při dynamickém rozsahu $\pm 2,0$ g je 660 mV/g. Operační rozsah teploty je od -40 do 85 °C. Citlivost akcelerometru se vzhledem k teplotě mění o $\pm 0,1$ %/°C pro osy x a y. Pro osu z je to $\pm 0,4$ %/°C. Výstupní odpor se pohybuje v rozmezí od 24 do 40 k Ω .

²⁸ASIC (Application-Specific Integrated Circuit) - aplikačně specifický integrovaný obvod

2.4 Vybavení letadel letovými zapisovači

Zákon č. 49/1997 Sb. říká, že „Letový zapisovač je záznamové zařízení instalované v letadle, které slouží k získání údajů usnadňujících odborné zjišťování příčin leteckých nehod nebo incidentů.“ [29]

Jejich přítomnost na palubě letounů majících státní příslušnost tohoto státu udává Hlava 6 Přístroje, vybavení letounu a letová dokumentace leteckého předpisu L 6/I Provoz letadel - Část 1. Z pohledu této práce je nejzajímavější část 6.3 Letové zapisovače. Ta specifikuje jednotlivé typy letových zapisovačů včetně podmínek použití těchto zapisovačů na palubách letounů. Mimo jiné se dozvídáme, že letový zapisovač je komplexní zařízení, které je tvořeno několika systémy. Ty umožňují například snímat letecké údaje nebo hlasy na palubě letounu.

Aplikace, která je předmětem této práce, je vytvářena pro ultralehké letouny. Ten je v tom samém zákoně definován jako „Ultralehký letoun je letoun, který je konstruován maximálně pro dvě osoby, řízený buď přesouváním těžiště pilota, nebo aerodynamickými prostředky, jehož pádová rychlost nepřevyšuje 65 km/h, jehož maximální vzletová hmotnost je 450 kg.“ [29]

Na základě této definice a informací z již zmíněných předpisů je zřejmé, že ultralehké letouny nemají dle platných předpisů žádnou povinnost mít na své palubě jakýkoliv druh letového zapisovače.

2.5 Dostupná zařízení pro sběr leteckých dat

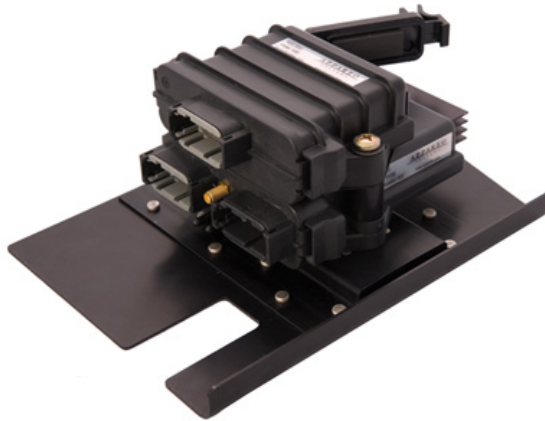
V předchozí podkapitole byl vysvětlen pojem letový zapisovač. Ten zde byl spojován s prostředkem, který je používán pouze ve spojení s vyšetřováním nehod. V současné době ovšem je na trhu celá řada zařízení, která poskytují podobnou funkcionalitu a jsou mimo jiné i pro sportovní účely. Pomocí těchto zařízení lze zpravidla zaznamenávat různé množství letových dat, zvukových a obrazových záznamů z průběhu letu. Ta je možné po každém letu pomocí speciálního softwaru analyzovat. Tato podkapitola přiblíží dvě takováto zařízení, která jsou v současné době na trhu.

Zdrojem informací pro tuto část byly informace dostupné na webových stránkách výrobců Appareo Systems [19], Wi-Flight [26] a elektronický seriál o mobilním telefonu OpenMoko Neo FreeRunner [6].

2.5.1 GAU 2000

Jedná se o zařízení od firmy Appareo Systems, které je možné umístit na jakýkoliv druh letadla. Veškerá letová data jsou měřena pomocí tříosého akcelerometru, gyroskopu, tlakového senzoru a šestnáctikanálového GPS modulu. Jednotlivá data jsou ukládána zároveň na externí SD kartu a interní paměťový modul, který by měl odolávat poškození i při pádu letadla. Veškeré zmíněné prvky jsou umístěny v rámci jednoho modulu, který je na obrázku 2.13.

Samotný výrobce udává, že toto zařízení je již používáno ve stovkách letadel na celém světě. Poskytuje standardní sadu leteckých dat jako je poloha, rychlost, kurz atd. Vzhledem ke své konstrukci je zařízení možné využívat i jako bezpečnostní prvek. Zároveň je možné v kombinaci se softwarem od stejného výrobce analyzovat každý let, kdy bylo toto zařízení v chodu.



Obrázek 2.13: Letový zapisovač GAU 2000 od Appareo Systems (převzato z [19])

2.5.2 GTA02

Druhým zařízením, které je vůči předchozímu úplně odlišné koncepce, je GTA02 od firmy Wi-Flight. Toto zařízení je postaveno na tzv. otevřeném mobilním telefonu OpenMoko Neo FreeRunner, který je na obrázku 2.14.



Obrázek 2.14: Mobilní telefon OpenMoko Neo FreeRunner (převzato z [26])

Ten v sobě obsahuje například tříosý akcelerometr, GPS modul, slot na SD kartu, rozhraní USB a WiFi. Firma Wi-Flight využívá těchto modulů pomocí vlastního softwaru pro získávání letových dat.

Po přistání jsou data ze zařízení přehrána pomocí WiFi na datové úložiště. Tato data jsou poté automaticky do dvou minut analyzována na serveru. Výsledky analýzy jsou posléze prostřednictvím emailu zaslány na určenou adresu. Analýza se soustřeďuje na vyhledání kritických míst letu, jakými jsou nízká letová výška nebo vysoké hodnoty přetížení.

Průběh celého letu je možné posléze zobrazit pomocí 3D aplikace, která prostřednictvím aplikace Google Earth a Google Maps poskytuje věrohodnou rekonstrukci celého letu. Toto zařízení slouží spíše pro sportovní a výukové účely.

Kapitola 3

Shrnutí současného stavu a požadavků

V textu již zaznělo, že navrhovaný systém má sloužit pro záznam leteckých dat. Ty lze chápat jako konečnou množinu bodů v prostoru, ve které je každý bod tvořen sadou výstupních hodnot jednotlivých snímačů v daném čase.

Snahou je vytvořit zařízení, které by bylo odlehčenou variantou letového zapisovače. V této fázi vývoje nemá cenu hovořit o tom, zda navrhovaný systém bude sloužit pro informační nebo bezpečnostní účely. Důležité ovšem je vytvořit funkční zařízení, na kterém by si bylo možné osvojit základní principy vývoje vestavěných systémů a možnosti jejich využití na palubách sportovních letadel.

3.1 Stanovení požadavků

Základním požadavkem je, že systém musí být schopen zaznamenávat GPS souřadnice, nadmořskou výšku, rychlost a náklony v osách x a y . Právě hodnoty náklonů v obou osách mají sloužit jako jednoduchá náhrada umělého horizontu. Minimální frekvence záznamu dat, která musí být stálá, je stanovena na 0,5 Hz. Při této frekvenci je nutné mít dostatek paměti pro uložení alespoň jedné hodiny letu.

Dalším požadavkem je, že řídicím členem celého systému musí být mikrokontrolér. Systém jako celek musí být schopen zpracovávat data v reálném čase a umožňovat jejich pozdější analýzu na počítači. To znamená, že datové úložiště musí být realizováno pomocí nevolatilní paměti. Její obsah musí být možné kdykoliv přenést do počítače, kde bude proveden převod do lidsky čitelného formátu.

Zároveň je vhodné zavést určitý druh kontroly zápisu dat do paměti pro každou sadu dat zvlášť. Jako ideální řešení se jeví využití kontrolního součtu, který bude ukládán v rámci každé sady dat. Posledním požadavkem je možnost napájení celého systému pomocí baterií tak, aby bylo dosaženo nezávislosti na konkrétním cílovém prostředí.

3.2 Návrh požadavků

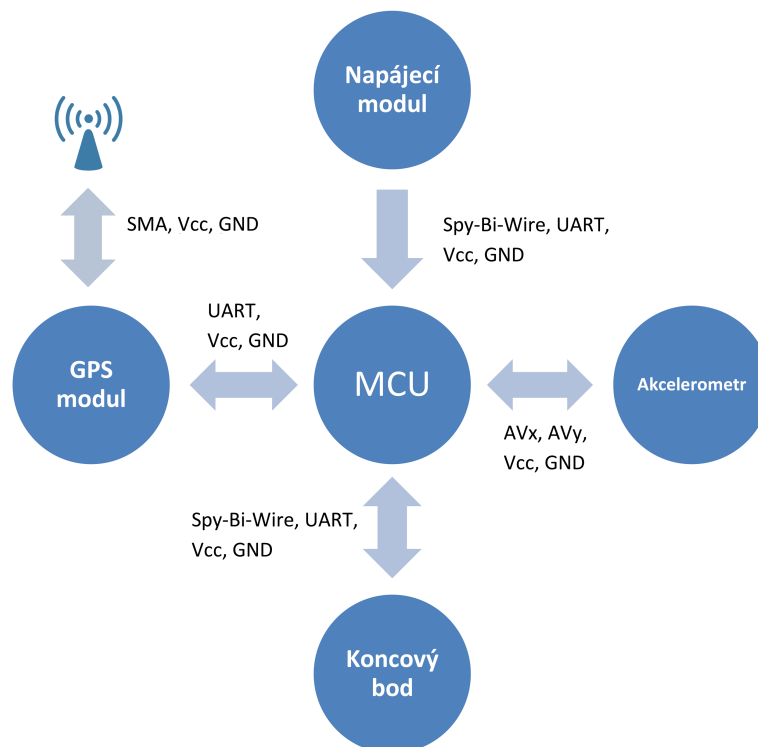
Jako vhodné se zdá využití kitu eZ430-RF2500, který byl v krátkosti představen v části [2.1.8](#). Pro jeho využití především hovoří jeho komplexnost a dostupnost. Drobnou připomínku lze mít k velikosti vnitřní paměti mikrokontroléru (MCU), kterým je osazena cílová deska kitu. V tomto konkrétním případě se bavíme od 32 kB, které jsou využity pro uložení

kódu a lze je také využít i jako datové úložiště. Toto omezení, které pramení z nepříliš velké vnitřní paměti mikrokontroléru, lze vyřešit buď přidáním externího paměťového modulu, nebo zavedením určitého druhu kódování dat.

Externí paměťové úložiště je možné realizovat například pomocí modulu s SD kartou, který využívá rozhraní SPI. To je na straně mikrokontrolérů z rodiny MSP430 podporováno modulem USCI, který byl popsán v části 2.1.5. Nevýhodou je, že toto řešení navyšuje hardwarové požadavky a cenu celého zařízení. Z tohoto důvodu je pozornost zaměřena na využití kódování, pomocí kterého by bylo možné poskytnout požadovanou funkcionalitu za menší cenu.

Pro získání GPS souřadnic, nadmořské výšky a rychlosti je vhodné použít GPS modul LR9540G, který byl popsán v rámci části 2.2.1. GPS souřadnice a nadmořskou výšku je možné získat z věty GGA. Rychlost je možné získat z věty VTG. Alternativně ji lze vypočítat z delta GPS souřadnic. Jediným komunikačním rozhraním GPS modulu LR9540G je rozhraní UART. Vzhledem k cílovému prostředí by mělo být dostačující realizovat tuto komunikaci pouze s využitím jednoho kanálu. Dále je nutné přidat aktivní anténu, kterou lze připojit pomocí standardního SMA konektoru k GPS modulu. Lze použít například aktivní anténu Mounting 32.

K získávání zbývajících hodnot, kterými jsou náklon v osách x a y , je vhodné využít akcelerometr. Konkrétně se nabízí využití kitu EZ430-C9, jehož analogové výstupy AV_x a AV_y lze připojit na neobsazené analogové vstupy cílové desky eZ430-RF2500T. Samotný kit s akcelerometrem je rovnou připraven k použití bez nutnosti přidání dalších podpůrných obvodů. Napájení je opět možné brát z cílové desky eZ430-RF2500T. Výše popsany návrh je ukázán na obrázku 3.1.



Obrázek 3.1: Návrh požadavků

3.3 Způsob ukládání dat

Na začátku této kapitoly bylo zmíněno, že je nutné využít určitý druh kódování dat tak, aby bylo možné co nejefektivněji využít kapacitu vnitřní paměti mikrokontroléru. Jako vhodné se jeví využití delta kódování, jehož princip spočívá v ukládání rozdílu mezi aktuální a předchozí hodnotou [16]. Díky tomu lze v případě, kdy nedochází k prudkým změnám po sobě jdoucích hodnot, využívat pro uložení paměťově méně náročné datové typy.

Hlavní nevýhodu tohoto způsobu kódování dat vidím v možné komplikaci při pozdější rekonstrukci dat. Tento problém spočívá v samotném způsobu ukládání, jelikož chybný zápis hodnot v jednom bodě způsobí zkreslení hodnot ve všech následujících bodech. Oproti tomu lze tímto způsobem, v porovnání s absolutním zápisem, docílit efektivnějšího využití paměťového prostoru. Další nevýhodou, která ovšem v tomto případě nečiní zásadní problém, je větší paměťová náročnost na paměť RAM, ve které je navíc nutné uchovávat absolutní hodnoty jednotlivých měřených veličin v předchozím bodě. Dále je vhodné zmínit složitější implementaci a větší velikost výsledného kódu, který tím pádem snižuje zbývající volnou kapacitu paměti Flash. Obecně platí, že přínos uplatnění kódování vždy závisí na konkrétních podmínkách a účelu aplikace.

Z předchozího popisu je zřejmé, že tento druh kódování není příliš vhodný pro koncovou podobu zařízení. Je ovšem vhodný pro testování a získání co největšího množství dat v relativně krátkém časovém úseku.

3.4 Možnosti využití navrhovaného řešení

Již bylo řečeno, že cílem je vytvořit funkční základ zařízení, které by umožňovalo sběr leteckých dat. Koncepce celého zařízení umožňuje velmi rychlé přizpůsobení navrhovaného zařízení tak, aby bylo možné vyhovět velkému množství požadavků.

Z pohledu využití je možné hovořit o zařízení, které by mělo čistě informační charakter. Množství snímaných veličin by bylo nutné zřejmě navýšit tak, aby informační hodnota zařízení splňovala konkrétní požadavky. Například o audiovizuální záznam paluby letounu. Zda by bylo možné využít stávající mikrokontrolér nebo by bylo nutné použít vyspělejší mikrokontrolér, z té samé rodiny, je už otázkou konkrétních požadavků a způsobu realizace. Z pohledu softwaru se ovšem nejedná o zásadní změnu. Použité technické vybavení mikrokontroléru je totiž téměř standardní výbavou, která je u vyspělejších zařízení spíše rozšiřována. Nedílnou součástí takového zařízení by byl i poněkud pokročilejší nástroj, pomocí kterého by bylo možné zaznamenaná data prohlížet.

Další možné využití zařízení je jako bezpečnostní prvek. Jednou z variant, která byla hodně diskutována na začátku této práce, je vytvoření odlehčené verze černé skříňky. Vnitřní struktura takového zařízení je velmi podobná té, která byla uvedena v předchozím odstavci. Navíc by ovšem bylo nutné zařízení zapouzdřit tak, aby odolávalo destruktivním účinkům pádu letadla. Druhou variantou je aktivní bezpečnostní zařízení, které by dokázalo na základě známých hodnot varovat pilota o překročení mezí letové obálky.

Kapitola 4

Popis systému

Má práce na tomto systému spočívala ve vytvoření návrhu, jeho realizaci pomocí dostupných komponent, vytvoření softwaru pro řídicí člen a otestování celého zařízení. Očekávan byl grafický výstup, který bude zobrazovat trajektorii letu, nejlépe s využitím některých mapových podkladů. Příklad takového výstupu je na obrázku 4.1.



Obrázek 4.1: Očekávaný formát výstupu

Fyzická podoba zařízení, která téměř odpovídá návrhu z části 3.2, je na obrázku 4.2. Systém je tedy tvořen GPS modulem LR9540G, akcelerometrem EZ430-C9, aktivní GPS anténou Mounting 32, cílovou deskou eZ430-RF2500T, USB programátorem/debuggerem eZ430-RF a napájecím modulem. Cílová deska je zapojena do napájecího modulu, pomocí kterého je možné napájet zařízení z baterií nebo z autoadaptéru pro nabíječky ipc-1 a ipc-11 od společnosti FK technics spol. s r.o. na 12 V. Právě přítomnost autoadaptéru je změnou vůči návrhu z části 3.2. Jeho použití doporučuji pokaždé, kdy je možné z palubních přístrojů letadla 12 V získávat. V systému totiž není zabudována žádná ochrana, která by hlídala hodnotu napájecího napětí, jehož pokles zapříčiní selhání systému. Selháním je myšleno ukončení další činnosti zařízení. Již uložená data ovšem zůstanou zachována. Z tohoto

důvodu je lepší využít stálý zdroj napájecího napětí.



Obrázek 4.2: Fyzická podoba zařízení

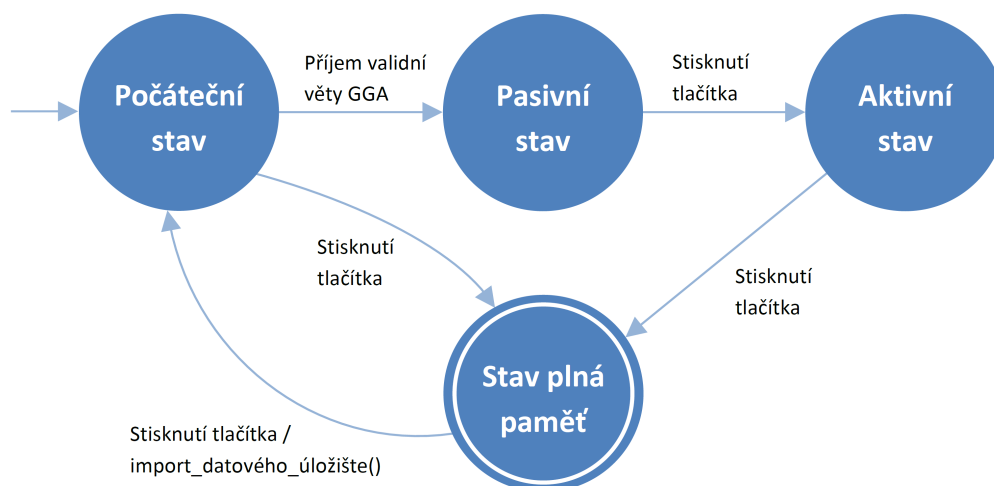
Jumper na napájecím modulu, který slouží pro přívod napětí z baterií, musí být s využitím adaptéru rozpojen. Naopak při napájení zařízení z baterií je nutné jumper zapojit a adaptér rozpojit. Pro migraci dat je nutné odpojit anténu z GPS modulu a cílovou desku eZ430-RF2500T zapojit do USB programátoru/debuggeru eZ430-RF.

4.1 Princip činnosti

Zjednodušeně lze říci, že činnost systému je řízená výhradně uživatelem pomocí tlačítka na cílové desce. Samotný sběr dat v aktivním stavu je poté taktován příchozími větvami z GPS modulu. Z pohledu implementace je klíčové to, že chod celého systému je řízen výhradně pomocí systému přerušení. Veškeré akce, které jsou svázány se zpracováním stisku tlačítka nebo komunikací s GPS modulem, jsou realizovány uvnitř obslužných rutin pro porty P1 a P2. Obě budou ve stručnosti představeny v rámci této části. Podrobnější popis obou obsluh je obsažen v příloze této práce. Mimo již zmíněná přerušení jsou dále rozlišována přerušení od časovače, modulu pro realizaci sériové komunikace a analogově digitálního převodníku. Jejich význam bude v této části taktéž přiblížen. Další možností, která nebyla doposud zmíněna a může ovlivnit chod systému, je softwarový reset zařízení. Ten může nastat například při chybné práci s pamětí nebo načtením instrukce z adres, které nejsou určeny pro uložení kódu. Po tomto resetu je systém uveden do počátečního stavu. Po dobu, kdy není vyžadována činnost procesoru, je mikrokontrolér v úsporném režimu.

4.1.1 Stavby zařízení

Z pohledu implementace je možné systém popsat jako deterministický konečný automat o čtyřech stavech, které jsou popsány níže. Přechod mezi stavy je realizován buď pomocí již zmíněného tlačítka, nebo při splnění určitých podmínek za běhu. Graf konečného automatu, popisující ovládání systému, je na obrázku 4.3.



Obrázek 4.3: Graf konečného automatu

Počáteční stav: Indikován pomocí blikání zelené LED. Výchozí stav po zapnutí, během kterého je provedena inicializace modulů BCM+, USCI, ADC10, Timer_A, portů P1, P2, P3, P4 a GPS modulu. S výjimkou přerušení pro kanál nula časovače Timer_A a modulu ADC10 jsou veškerá přerušení po dobu inicializace zakázána. Po dokončení inicializace jsou veškerá přerušení povolena.

Pasivní stav: Indikován svícením zelené LED. Tento stav značí, že je zařízení připraveno pro zahájení sběru dat. Přejechod z počátečního stavu do tohoto stavu je realizován automaticky v rámci obslužné rutiny pro port P2 po přijetí validní GGA věty z GPS modulu. Kromě přerušení od portu P1 a modulu ADC10 jsou všechna ostatní přerušení zakázána.

Aktivní stav: Indikován blikáním červené LED. Při přechodu do tohoto stavu je provedena inicializace datového úložiště a povolení přerušení pro kanál jedna časovače Timer_A, modulu ADC10 a portů P1, P2. V tomto stavu probíhá záznam leteckých dat s frekvencí 0,5 Hz. Samotný princip záznamu dat je vysvětlen dále v textu v rámci obslužné rutiny pro port P2.

Stav plná paměť: Indikován svícením červené LED. Kromě přerušení pro P1 jsou všechna ostatní přerušení zakázána. Využíván jako výchozí stav pro zahájení importu obsahu datového úložiště. Ten je zahájen opětovným stiskem tlačítka. Po ukončení importu přechází zařízení do počátečního stavu.

4.2 Použité hardwarové vybavení

Ke své činnosti mikrokontrolér využívá moduly: Timer_A pro komunikaci pomocí rozhraní UART s GPS modulem, ADC10 pro převod analogových hodnot výstupu akcelerometru a USCI pro migraci dat z vnitřní paměti Flash do PC skrze rozhraní UART. Dále je použit modul BCM+, interní paměť Flash a porty P1, P2, P3 a P4. S výjimkou akcelerometru je nutné provést inicializaci všech součástí systému.

4.2.1 Konfigurace hodinových signálů

Modul BCM+ je nakonfigurován tak, že zdrojem hodinového signálu pro MCLK a SMCLK je DCO. Ten je pomocí kalibračních konstant, které jsou uloženy v Segmentu A informační paměti, nastaven na frekvenci 8 MHz. Děličky pro oba hodinové signály jsou nastaveny na dělicí poměr jedna. To znamená, že frekvence obou hodinových signálů je rovna výstupní frekvenci DCO.

4.2.2 Úsporný režim

Po dobu, kdy není vyžadována činnost procesoru, je mikrokontrolér v režimu LPM0. Použití tohoto režimu je dáno využitím hodinového signálu SMCLK pro časovač Timer_A a oscilátoru DCO jakožto zdroje hodinového signálu pro SMCLK. Tato koncepce neumožňuje využití vyššího úsporného režimu než je režim LPM1. Ten by ovšem v tomto případě měl zcela totožnou charakteristiku jako použitý režim LPM0.

4.2.3 Datové úložiště

Jako datové úložiště slouží volný paměťový prostor interní Flash paměti mikrokontroléru. V tomto případě se jedná o 26,5 kB, na které je možné uložit až 90 minut letu. Jako zdroj hodinového signálu pro paměťový modul je vybrán signál MCLK, který je uvnitř modulu dělen dvaceti tak, aby splňoval požadavek na frekvenční rozsah hodinového signálu.

4.2.4 Migrace obsahu datového úložiště

Odeslání obsahu datového úložiště do PC je možné pouze skrze USB programátor/debugger eZ430-RF. Ten je připojen k desce s mikrokontrolérem pomocí šestipinového konektoru, který mimo jiné obsahuje datové signály TX a RX pro rozhraní UART. Tyto signály jsou přivedeny na piny P3.4 a P3.5. V tomto případě jsou tyto piny namapovány do modulu USCI, který je v režimu UART použit pro přenos obsahu datového úložiště do PC.

Zdrojem hodinového signálu pro modul USCI je hodinový signál SMCLK. Přenosová rychlost je pevně stanovena na běžně používanou hodnotu 9600 baudů. Je pravdou, že se jedná o relativně nízkou hodnotu, která ovšem má své opodstatnění. U modulu USCI v režimu UART totiž platí, že s rostoucí přenosovou rychlostí a klesající frekvencí hodinového signálu roste chyba, která může vzniknout při odesílání. Ta je vyjádřena jako kladná a záporná odchylka vůči ideálnímu časování. Při těchto hodnotách přenosové rychlosti a frekvenci hodinového signálu je tato chyba rovna -0,1 až 0 %. Například při zvýšení přenosové rychlosti na 38400 baudů dojde k nárůstu chyby na -0,2 až 0,5 %. Velikost přenášených dat a znatelný nárůst možné chyby při přenosu hovoří proti zvýšení přenosové rychlosti z hodnoty 9600 baudů. Datový rámec je tvořen jedním start bitem a osmi datovými bity.

4.2.5 Komunikace s akcelerometrem

Pro zpracování výstupu akcelerometru je použit modul ADC10, jehož výstup je přiveden na piny P4.3 a P4.4. Ty jsou nastaveny jako analogové vstupy. Vstupním hodinovým signálem pro ADC10 je signál MCLK, který je uvnitř modulu pomocí děličky upraven na hodnotu 1 MHz. Při této frekvenci je na základě rovnice 2.2 nastaven vzorkovací čas na 16 hodinových cyklů. Jako referenční napětí pro horní mez bylo zvoleno napájecí napětí a pro spodní mez zem. Převod hodnoty každé osy probíhá v režimu jednorázový převod jednoho kanálu.

4.2.6 Konfigurace GPS modulu

GPS modul je inicializován tak, aby skrze kanál A zasílal věty VTG a GGA s frekvencí 0,5 Hz. Součástí každé odesílané věty je kontrolní součet. Přenosová rychlost je ponechána na výchozí hodnotě 4800 baudů. Datový rámec je tvořen start bitem, osmi datovými bity a jedním stop bitem.

4.2.7 Komunikace s GPS modulem

Komunikaci s GPS modulem bylo nutné realizovat pomocí knihovny od Johna H. Daviese, která byla představena v části 2.1.6. K tomuto řešení bylo nutné přistoupit, jelikož volně dostupné piny mikrokontroléru nelze vnitřně namapovat do modulu USCI, pomocí kterého by bylo možné komunikaci realizovat.

Knihovna je založena na využití časovače Timer_A a jeho kanálů nula a jedna spolu s jejich přerušovacími vektory. Kanál nula, jehož výstup je vyveden na výstupní pin P2.2, slouží pro odesílání. Vstupem pro kanál jedna, který slouží pro příjem, je vstupní pin P2.3. Hodnoty těchto pinů slouží jako datové signály, které jsou u rozhraní UART popisovány jako signál TX a RX.

Ke komunikaci se zbytkem programu potřebuje knihovna využít nějaký jiný přerušovací vektor s alespoň dvěma příznaky přerušování, které lze softwarově nastavit. Ty poté mají naprosto stejnou funkci jako příznaky UCAnRXIFG a UCAnTXIFG u modulu USCI, který byl představen v části 2.1.5. Tato aplikace k tomuto účelu využívá přerušovací vektor pro port P2 a jeho příznaky P2IGF_bit.P6 (UCAnRXIFG) a P2IGF_bit.P7 (UCAnTXIFG). Tyto dva piny není nutné nastavovat. Vstupním hodinovým signálem do časovače je signál SMCLK, jehož frekvence je uvnitř modulu upravena na hodnotu 1 Mhz.

4.2.8 Indikace a změna stavu zařízení

K indikaci stavu zařízení slouží dvě svítivé diody, které jsou součástí desky s mikrokontrolérem. Tyto svítivé diody jsou připojeny na piny P1.0 a P1.1, které jsou nakonfigurovány jako výstupní. Stav svítivých diod v závislosti na stavu zařízení byl uveden v rámci popisu jednotlivých stavů.

Samotné ovládání svítivých diod je realizováno zápisem příslušných hodnot do registru P1OUT. O něco složitější je realizace blikání svítivé diody. Ta spočívá ve využití kanálu dva časovače Timer_A v režimu porovnání. Pro dosažení nízké frekvence blikání je registr TACCR2 inicializován na maximální hodnotu. Samotná změna stavu svítivé diody je realizována v rámci obslužné rutiny kanálu dva časovače Timer_A.

Ke změně stavu zařízení slouží tlačítko, které je vyvedeno na pin P1.2. Ten je nakonfigurován jako vstupní pin citlivý na sestupnou hranu. Zároveň je na tento pin připojen vnitřní pull up rezistor, který definuje logickou hodnotu pinu při neseprnutém stavu.

4.3 Obsluhy přerušování

Veškerá činnost, která souvisí s řízením chodu systému, je prováděna v rámci obslužných rutin jednotlivých přerušování. Z pohledu řízení chodu systému jsou nejzajímavější obslužné rutiny pro porty P1 a P2, které jsou v této části popsány velmi zjednodušeně. Jejich podrobnější popis je vložen v příloze této práce.

4.3.1 Obsluha přerušení pro port P1

Tato obsluha slouží pro zpracování požadavku uživatele na změnu stavu zařízení pomocí tlačítka. Pro odstranění mechanických zákmitů při sepnutí tlačítka je přidána na začátek obsluhy přerušení zpoždovací smyčka. Po ní je na základě aktuálního stavu zvolen nový stav zařízení. Například při přechodu z pasivního do aktivního stavu je provedena inicializace datového úložiště, zahájení záznamu dat a samotná změna stavu.

4.3.2 Obsluha přerušení pro port P2

Již bylo zmíněno, že komunikace s GPS modulem je založena na využití přerušovacího vektoru pro port P2. Konkrétně jsou uvnitř obslužné rutiny rozlišovány příznaky P2IGF_bit.P7 a P2IGF_bit.P6. První zmíněný příznak slouží pro indikaci dokončení odeslání jednoho bajtu skrze sériovou linku do GPS modulu. Druhý naopak značí přijetí jednoho bajtu z GPS modulu. Příznaky tedy mají obdobnou funkci jako příznaky UCAnTXIFG a UCAnRXIFG z modulu USCI. Samotné dekodování a zpracování vstupních a výstupních NMEA vět je realizováno čistě v rámci těchto přerušovacích rutin. V aktivním stavu je sběr jednotlivých leteckých dat taktován příchozími větami z GPS modulu. Každá sada dat vzniká zpracováním hodnot z věty GGA, věty VTG a akcelerometru. Na závěr jsou data opatřena kontrolním součtem a zařízení znovu čeká na další příchozí sekvenci vět GGA a VTG. Více o formátu ukládaných dat je v části 4.4.

4.3.3 Obsluha přerušení pro Timer_A

Obsluha přerušení je rozdělena do vektoru přerušení pro kanál nula a vektoru přerušení pro kanály jedna a dva. Samotným obslužným rutinám se nebudu věnovat příliš dopodrobna, jelikož jsou implementovány v rámci knihovny pro realizace rozhraní UART.

V krátkosti však uvedu, že přerušovací vektor pro kanál nula obsahuje vše potřebné pro realizaci odeslání bajtu skrze sériovou linku. Druhý přerušovací vektor je společný pro kanál jedna a dva. Samotná knihovna využívá pro odeslání bajtu skrze sériovou linku pouze kanál jedna. Blikání zelené LED, které slouží pro indikaci počátečního stavu, je realizováno pouze pomocí negace hodnoty výstupního pinu P1.1 v rámci obslužné rutiny pro kanál dva.

Tento způsob řešení komunikace s GPS modulem a blikání svítivé diody je výhodný v tom, že je možné kdykoliv povolením nebo zakázáním přerušení pro daný kanál časovače konkrétní službu úplně zakázat. U kanálu nula, který slouží pro odeslání, je zbytečné obsluhu přerušení speciálně zakazovat a stačí ji pouze povolit v rámci inicializace. U kanálu jedna je situace ovšem složitější. Příjem jednotlivých bajtů uvnitř hlavního programu je zakazován s přerušením pro pin P2.6. To ovšem znamená, že hlavní program pouze není informován o přijetí jednoho bajtu. Obslužná rutina pro kanál jedna je pak stále zbytečně volána, což má za následek zvyšující se spotřebu celého zařízení. Z tohoto důvodu je příjem vět z GPS modulu povolován/zakazován výhradně pomocí bitu TACCTL1_bit.CCIE, který povoluje/zakazuje přerušení pro kanál jedna. Stejně je tomu i u kanálu dva, jehož činnost je opět povolována/zakazována výhradně pomocí bitu TACCTL2_bit.CCIE.

4.3.4 Obsluha přerušení pro ADC10 a USCI

Obě obsluhy přerušení pouze provádí přechod z režimu LPM0 do režimu AM. Princip práce s oběma moduly je totiž téměř totožný. V obou případech je v programu provedena

inicializace operace a pozastavení vykonávání kódu přechodem do režimu LPM0. V rámci obslužné rutiny je vykonávání kódu přechodem do již zmíněného režimu AM opět spuštěno.

4.4 Způsob zpracování a ukládání dat do paměti

Jednou z věcí, na kterou se bylo nutné zaměřit, byl způsob ukládní dat do paměti. Nutnost použití kódování již byla vysvětlena v části 3.3. Každá sada dat je tvořena: GPS souřadnicemi ve tvaru ddmm.mmmm, nadmořskou výškou a rychlostí jako desetinné číslo ve tvaru n.n a náklony v osách x a y jako celé číslo. Ukládat je možné po slovech či bajtech. S hodnotami rychlosti a výšky je pracováno jako s celými čísly, která vznikla vynásobením původních hodnot deseti.

4.4.1 Kódování

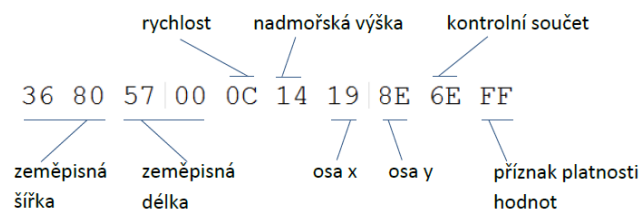
Bez využití kódování potřebujeme k uložení hodnot osm slov. S využitím delta kódování se lze dostat za ideálního stavu na hodnotu čtyř slov. Faktorem, který efektivnost kódování ovlivňuje, jsou velikosti rozdílů hodnot jednotlivých veličin dvou po sobě jdoucích bodů. Předpokladem je, že ve většině případů by měly být rozdíly jednotlivých hodnot veličin takové, aby je bylo možné uložit pomocí bajtu v přímém kódu. Tato hodnota je navíc ponížena o dvě hodnoty, které jsou vyhrazeny pro speciální účely. Veškerá tato omezení stanovují maximální hodnotu kladného a záporného přírůstku na hodnotu 126 (pro výšku a rychlost 12.6). Znaky s hodnotou 127 a 255 jsou vyhrazeny. Dále zařízení musí umět zaznamenávat i hodnoty vyšších přírůstků. Ty jsou poté ukládány pomocí slova v přímém kódu a jednoho bajtu, který slouží jako příznak. Je pravdou, že tyto vysoké hodnoty přírůstků velmi výrazně snižují efektivnost celého kódování. Tento způsob je využit pro ukládání hodnot rychlosti, nadmořské výšky a náklonů v obou osách. Při nemožnosti stanovit hodnotu rychlosti a nadmořské výšky jsou uloženy znaky s hodnotou 127.

U GPS souřadnic je vždy ukládán pouze rozdíl desetinné části minut pomocí jednoho slova v přímém kódu. Je totiž vyloučeno, aby mezi jednotlivými body došlo k přírůstku, který je větší než jeden stupeň. V přepočtu totiž vychází, že jedna minuta zeměpisné délky zhruba odpovídá vzdálenosti 1,854 km. U zeměpisné šířky se jedná o 1,189 km. Při rychlosti 500 km/hod, což je hodnota značně převyšující cestovní rychlost sportovních letadel, urazíme za dvě vteřiny 0,277 km. Jedná se tedy o přibližně čtvrtinu minuty u zeměpisné šířky a o šestinu minuty zeměpisné délky. Neplatné hodnoty GPS souřadnic jsou ukládány jako čtyři znaky s hodnotou 127.

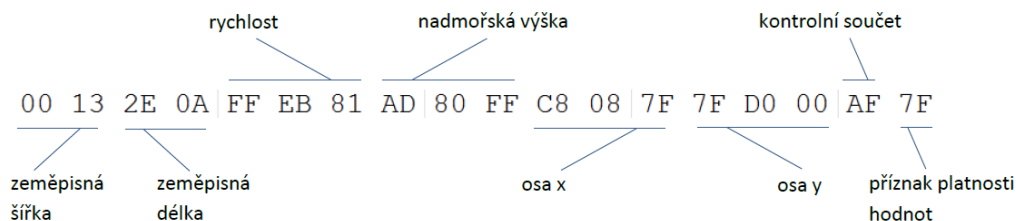
Příklad nejlepšího a nejhorší vzorku dat je na obrázcích 4.4 a 4.5. V ideálním stavu, který je na prvním obrázku, stačí pro uložení pět slov. Za těchto podmínek je možné uložit zhruba 90 minut letových dat. U nejhoršího vzorku dat je nutné počítat s devíti slovy, která zkracují dobu záznamu na zhruba 50 minut. Poslední dva bajty slouží pro uložení kontrolního součtu a příznaku platnosti dat.

4.4.2 Kontrolní součet

V předchozím textu bylo zmíněno, že je vhodné jednotlivá data doplnit o kontrolní součet tak, aby bylo možné ověřit správnost zápisu dat do paměti. V tomto případě jsou počítány dva kontrolní součty pro každou sadu dat zvlášť. První kontrolní součet je počítán ze všech hodnot dané sady, které mají být uloženy do paměti. Zároveň je tento kontrolní součet ukládán v rámci každé sady dat jako předposlední bajt. Druhý kontrolní součet je



Obrázek 4.4: Ukázka formátu nejlepšího vzorku dat v paměti



Obrázek 4.5: Ukázka formátu nejhoršího vzorku dat v paměti s chybou zápisu

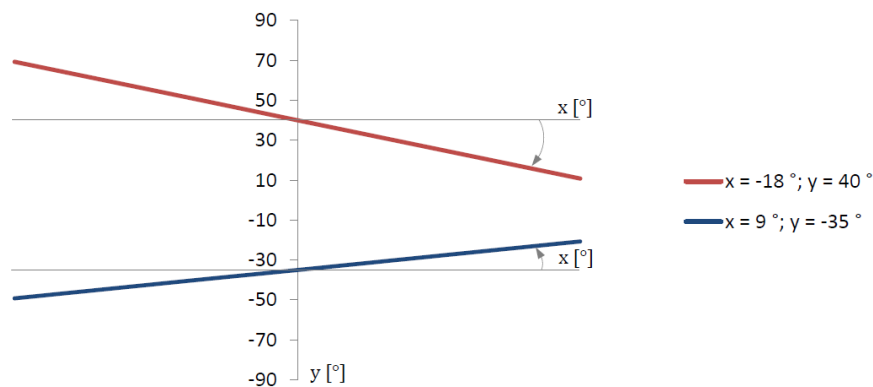
počítán z již uložených hodnot dané sady. Na základě porovnání obou kontrolních součtů je jako poslední bajt sady uložen znak s hodnotou 127 nebo 255. První znak slouží pro zneplatnění hodnot v případě, kdy hodnoty kontrolních součtů jsou rozdílné. Druhý pro potvrzení platnosti hodnot sady.

4.5 Výstup

Proces importu zaznamenaných leteckých dat ze zařízení do počítače lze rozdělit na dvě fáze: migraci obsahu datového úložiště a převod těchto dat do lidsky čitelného formátu. Samotná migrace spočívá v logování aktivity na virtuálním sériovém portu na straně počítače a současném odesílání obsahu datového úložiště skrze komunikační rozhraní UART na straně mikrokontroléru. Takto zachycená data jsou následně převedena do lidsky čitelného formátu pomocí skriptu, který je nad nimi spuštěn.

4.5.1 Formát hodnot snímaných veličin

GPS souřadnice jsou reprezentovány ve formátu ddmm.mmmm (d - stupně, m - minuty). Nadmořská výška a rychlost jsou vyjádřeny jako desetinné číslo ve formátu x.x. Náklony letadla v osách x a y jsou vyjádřeny ve stupních jako celá čísla. Hodnota osy y odpovídá hodnotě úhlu stoupání (kladné hodnoty osy y) nebo úhlu klesání (záporné hodnoty osy y). Hodnoty osy x odpovídají hodnotám příčného náklonu letadla. Kladné hodnoty osy x značí, že levé křídlo se nachází níže než pravé. Záporné hodnoty osy x naopak říkají, že pravé křídlo je níže než levé. Příklad závislosti hodnot úhlu v obou osách z pohledu pilota na pozici letadla je na obrázku 4.6.



Obrázek 4.6: Závislost pozice letadla z pohledu pilota na velikostech úhlů v osách x a y

4.5.2 Migrace zaznamenaných dat

Již bylo zmíněno, že migrace je založena na logování virtuálního sériového portu, který vznikne po správném nainstalování ovladače MSP430 Application UART od Texas Instruments [18]. Další nutnou podmínkou je připojení USB programátoru/debuggeru eZ430-RF spolu s cílovou deskou eZ430-RF2500T do jednoho z volných USB portů v počítači.

Po zahájení logování portu je nutné zařízení uvést do stavu plná paměť a z něj poté pomocí stisku tlačítka zahájit přenos obsahu datového úložiště. Odesílání je indikováno rozsvícenými svítivými diodami na desce s mikrokontrolérem. Po ukončení odesílání zařízení přechází do počátečního stavu, který je indikován blikáním zelené LED.

Nástroje pro logování sériového portu

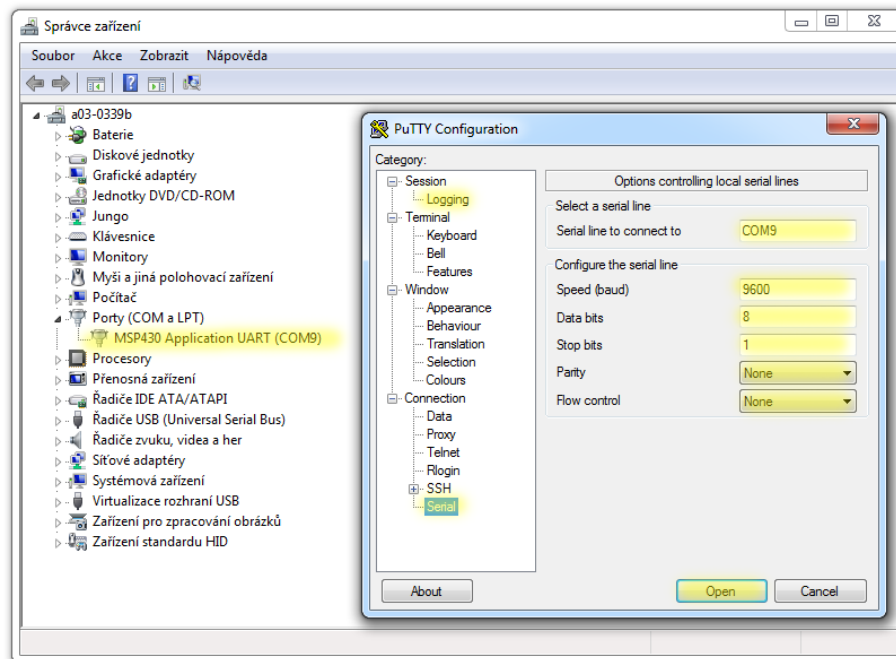
Samotných nástrojů pro logování portů jistě existuje více, než je v této části zmíněno. Osobně mám odzkoušené nástroje Putty, HyperTerminal a RS232 Data Logger. Všechny tyto zmíněné nástroje mají společné to, že jsou nabízeny bezplatně a jsou plně kompatibilní s 64 bitovým operačním systémem Windows 7, na kterém zároveň probíhalo veškeré testování.

Nástroj Putty

Většinu testování jsem prováděl pomocí nástroje Putty verze 0.62, který jako jediný považuji za řádně otestovaný. Využití jiného nástroje sebou nese povinnost modifikovat skript pro rekonstrukci dat, který očekává na začátku vstupního souboru hlavičku, která je automaticky vkládána nástrojem Putty.

Konfigurace nástroje Putty je na obrázku 4.7. Ta se skládá z nastavení identifikátoru sériového portu, datového rámce a výstupního souboru. Identifikátor sériového portu je možné zjistit například pomocí správce zařízení u operačního systému Windows 7. Datový rámec je tvořen osmi datovými bity a jedním stop bitem. Komunikační rychlost je 9600 baudů. Řízení toku dat je nutné vypnout. Jako výstupní soubor je možné ponechat výchozí nastavení. Alternativně lze, při zachování typu souboru, použít libovolný název, který je poté nutné zadat jako parametr skriptu.

Po dokončení nastavení je možné otevřít sériový port, čímž současně dojde k zahájení jeho logování do výstupního souboru. Pro ukončení logování stačí pouze nástroj zavřít.



Obrázek 4.7: Konfigurace nástroje Putty pro logování virtuálního sériového portu

4.5.3 Převod zaznamenaných dat do lidsky čitelného formátu

Soubor, který byl vytvořen pomocí nástroje Putty, je vzhledem k použitému kódování v této podobě nečitelný. Pro rekonstrukci samotných dat jsem vytvořil skript v jazyce Python verze 3.3.0, který ze vstupního binárního souboru vytvoří soubor typu KML. Alternativně je možné využít ladící mód, jehož výstupem je soubor typu TXT.

Použití skriptu

Pro použití skriptu je nutné mít nainstalovaný jazyk Python verze 3.3.0 nebo novější. Použití skriptu je na obrázku 4.8. Výchozím výstupním formátem je formát KML. Jméno výstupního souboru je možné zadat pomocí parametru nebo ponechat implicitní název. Jméno vstupního souboru je možné opět zadat nebo ponechat výchozí hodnotu, která je současně výchozí hodnotou názvu výstupního souboru nástroje Putty. Ladící výpis ve formátu TXT je vybrán při zadání přepínače *debug*.

Soubor typu KML

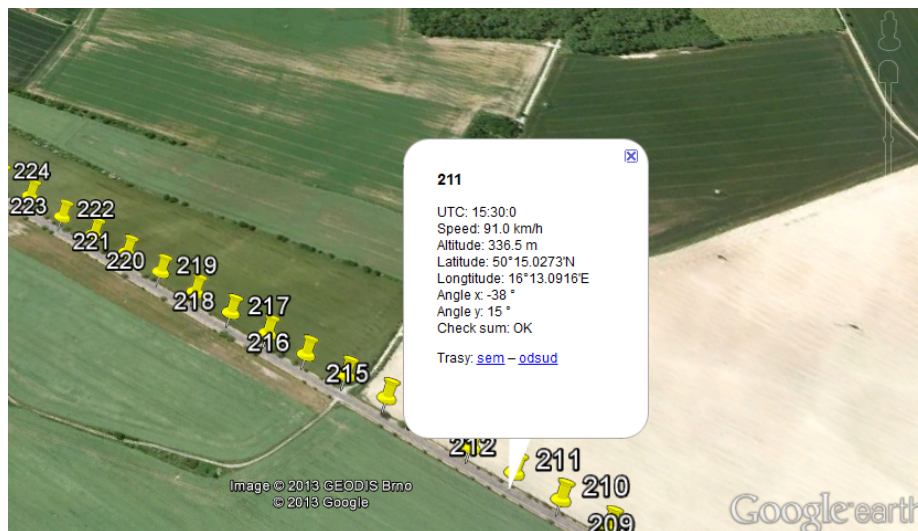
KML je označení typu souboru, využívající gramatiku souboru XML, který je určen pro popis geografických útvarů. Jedním z možných prohlížečů je aplikace Google Earth, která umožňuje interaktivní prohlížení zaznamenaných dat. Ukázka vizualizace výstupu z tohoto systému pomocí Google Earth je na obrázku 4.9.


```

C:\Users\Adam\Desktop\skriptBP>python bp.py -h
Description:
    Script for reconstruction of the flight data. Output format KML or TXT
    for debug mode.
Synopsis:
    bp.py [-h][--input=file_name][--output=file_name][--debug]
Options:
    -h                print help
    --input=file_name input file name
                    default: putty.log
    --output=file_name output file name
                    default: FDR.UTC-hh-mm-ss.kml
    --debug           debug mode

```

Obrázek 4.8: Parametry skriptu pro rekonstrukci dat



Obrázek 4.9: Vizualizace výstupu pomocí Google Earth

Každý soubor je tvořen množinou bodů, které jsou pojmenovány pomocí sekvenčního čísla bodu. Ke každému bodu je možné v aplikaci Google Earth rozkliknutím zobrazit kartu s detaily, která obsahuje hodnoty snímaných dat v daném bodě. Místa, ve kterých nebylo možné stanovit GPS souřadnice, jsou v souboru vynechány.

Jedním z údajů, který je v kartě obsažen, je kontrolní součet. V případě, že kontrolní součet v daném bodě není v pořádku, je pro snadnější orientaci do názvu bodu přidán symbol '**'.

Soubor typu TXT

Pro ladící účely je použit výstupní formát TXT. Obsah souboru je tvořen konečnou množinou záznamů, kdy každý záznam odpovídá jednomu bodu. Příklad jednoho záznamu je na obrázku 4.10.

```
pointer > 8693
pointer offset > 26
latitude min diff > 193
longtitude min diff > 210
altititude diff > 1.2 m
speed diff > 0.4 km/h
axis x diff > 30°
axis y diff > -2°
16°19.6301'E; 50°4.1265'N; 320.0 m; 67.4 km/h; -24 °; 18 °; OK
```

Obrázek 4.10: Ukázka záznamu ladícího výpisu

Kromě hodnot jednotlivých veličin v daném bodě ve formátu: zeměpisná délka; zeměpisná šířka; nadmořská výška; rychlost vůči zemi; náklon v ose x; náklon v ose y; stav kontrolního součtu, je každý záznam opatřen hlavičkou. Ta je tvořena hodnotami přírůstků vůči předchozímu bodu, hodnotou ukazatele a offsetu ve vstupním binárním souboru.

4.6 Testování

Jedním z cílů práce je otestovat zařízení v cílovém prostředí, pro které je vyvíjeno. V tomto případě se jedná o poněkud finančně náročný způsob testování, jelikož náklady na jednu letovou hodinu se u sportovních letadel pohybují okolo 2500 Kč. Vzhledem k tomuto faktu bylo nutné část testování provádět v jiném prostředí, které by bylo finančně méně náročné. V tomto případě se nabízela možnost testování pomocí osobního automobilu. Finální testování již ovšem probíhalo na palubě ultralehkého letadla. Veškeré pokročilé testy včetně popisu jsou uloženy na přiloženém CD.

4.6.1 Stacionární testování

Jednou z doposud nezmíněných metod testování, která byla použita především v počáteční fázi vývoje aplikace, je stacionární testování. Tento způsob testování byl použit pro ověření funkce jednotlivých částí systému, jako je práce s pamětí, citlivost akcelerometru, komunikace s jednotlivými moduly a stabilita v závislosti na okolních teplotách.

Z parametrů jednotlivých komponentů zařízení vyplývá, že maximální operační rozsah teploty by měl být od -40 do 85 °C. V rámci testů bylo dosaženo nejnižší okolní teploty -15 °C. Naopak nejvyšší hodnotou okolní teploty bylo 40 °C. V obou zmíněných případech byl chod zařízení bezchybný.

4.6.2 Testování v osobním automobilu

Testování s využitím osobního automobilu sloužilo k ověření činnosti systému jako celku. Mimo jiné byla ověřována schopnost vypořádání se s dočasnou ztrátou GPS signálu, validita výstupních hodnot z GPS modulu a stabilita komunikace s GPS modulem v závislosti na různých přenosových rychlostech.

Výhodou tohoto prostředí je možnost testování téměř v libovolný čas a na libovolném místě za několikanásobně nižší náklady vůči testování na palubě ultralehkého letadla. Právě náklady a časová náročnost na testování v cílovém prostředí byly klíčové. Celková doba trvání pokročilých testů, které byly provedeny pomocí automobilu, byla přibližně 8,5 hodin. Tato doba odpovídá 15 300 bodů, jejichž hodnoty byly zaznamenány a analyzovány.

Přesnost GPS souřadnic

Jedním z klíčových parametrů, který byl testován, byla přesnost GPS souřadnic. Výrobce použitého GPS modulu udává, že by měla být deset metrů. První typ testu probíhal logováním GPS souřadnic z konstantní pozice. Příklad výsledku testu, který je tvořen 2657 body, je na obrázku 4.11.



Obrázek 4.11: Test přesnosti GPS souřadnic z konstantní pozice

Bílá kružnice vyznačuje pásmo, ve kterém by se měly nacházet veškeré body. Jedná se o kružnici s poloměrem deset metrů (výrobce udávaná přesnost) a středem v místě, kde bylo umístěno zařízení. Skutečný rozptyl zaznamenaných GPS souřadnic je vyznačen pomocí světle modré kružnice, jejíž poloměr je 14 metrů. Ve druhé fázi byla testována přesnost GPS souřadnic za pohybu. Příklad jednoho z mnoha výstupů je na obrázku 4.12.

Na základě popisovaných testů lze říci, že přesnost GPS modulu je pro tuto aplikaci zcela dostačující. Drobné problémy nastávají pouze v případě, kdy je zařízení po delší časový úsek bez pohybu. Maximální chyba však i v tomto případě byla 14 metrů. Navíc absolutní většina bodů bez problému splňovala výrobcem stanovenou přesnost.

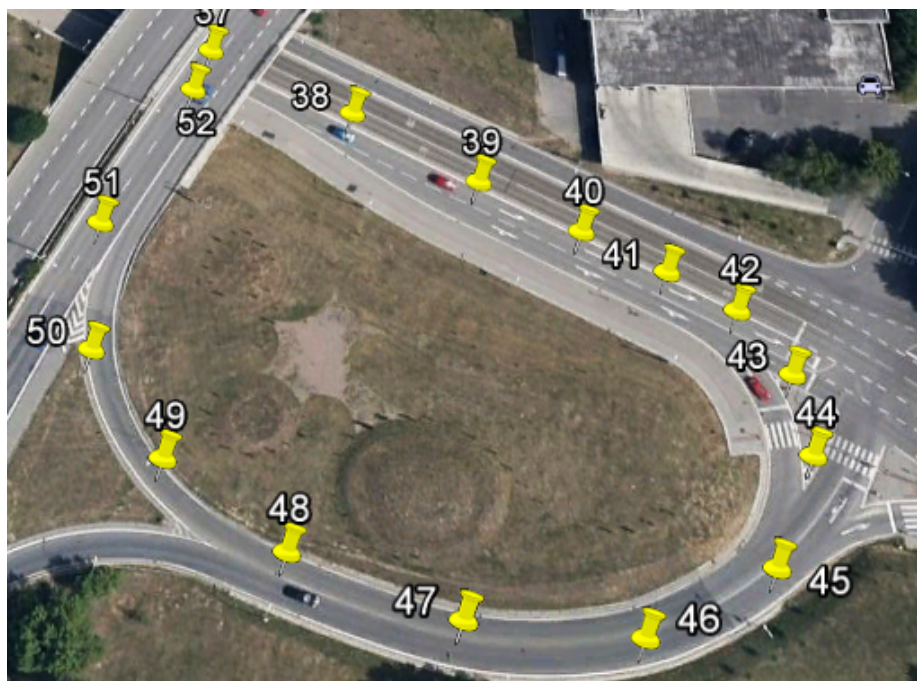
Dočasná ztráta GPS signálu

Další důležitou věcí, kterou bylo nutné otestovat, byla schopnost vypořádat se s dočasnou ztrátou GPS signálu. Po tuto dobu by zařízení mělo ukládat do paměti příznaky neplatných hodnot z GPS modulu v daných bodech, které by poté neměly být v rámci rekonstrukce dat zahrnuty do výstupního souboru typu KML. Příklad výstupu, který byl poznamenán dočasnou ztrátou signálu, je na obrázku 4.13.

Hodnoty rychlosti a spolehlivost zpětného kanálu UART

Ověření hodnot rychlosti bylo realizováno porovnáním výstupní hodnoty z GPS modulu s aktuální hodnotou digitálního tachometru vozidla. Hodnota rychlosti byla zasílána v reálném čase z mikrokontroléru pomocí zpětného kanálu UART do počítače. Díky tomu bylo možné otestovat i tuto komunikaci, která je využívána pro migraci obsahu datového úložiště.

S vlastními hodnotami rychlosti problém nebyl. Problémy se ovšem ukázaly u jednoho z nástrojů, který byl využit pro logování portu. Konkrétně jsem měl problémy s nástrojem



Obrázek 4.12: Test přesnosti GPS souřadnic za pohybu

HyperTerminál, který zdvojoval hodnoty některých bajtů ve výstupním souboru. S nástrojem Putty, kterým byl HyperTerminál nahrazen, žádné problémy za celou dobu testování nevznikly.

Paměťové operace

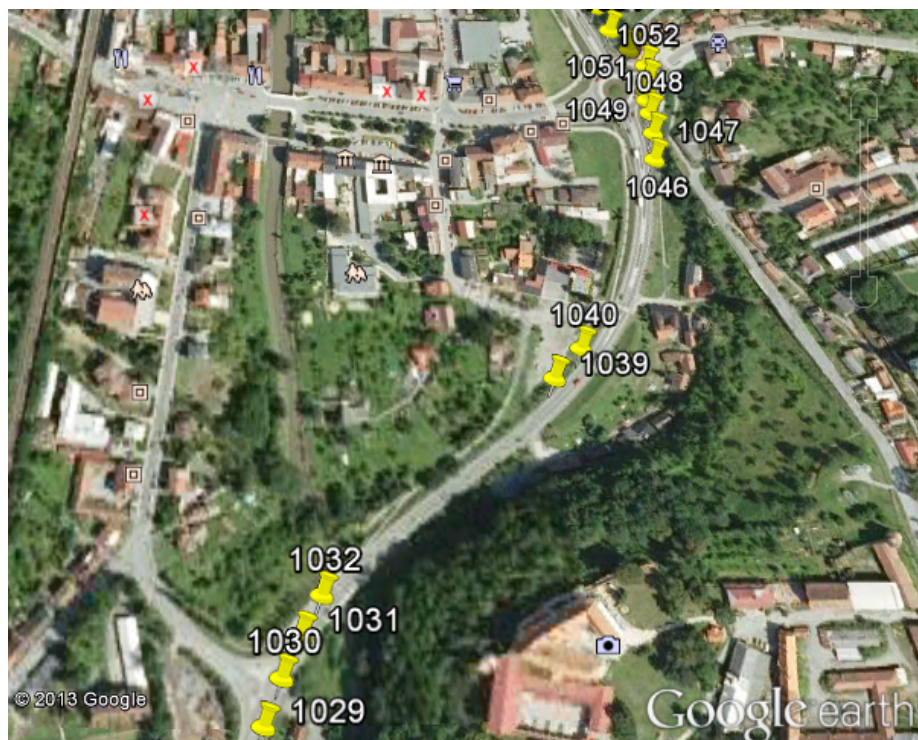
V rámci většiny testů byla současně testována i práce s pamětí, na kterou byly všechny hodnoty zapisovány a následně kontrolovány pomocí kontrolních součtů. Zápis do paměti nebyl bezchybný. Jednalo se však o ojedinělé případy, které v konečném součtu tvořily minimální procento z celkového počtu zápisů do paměti. Chybný zápis do paměti nastal v případě, kdy je hodnota příznaku platnosti hodnot sady rovna 127. Zároveň v tomto případě bude pro daný bod neplatný kontrolní součet.

4.6.3 Testování na palubě ultralehkého letadla

Celkově byly provedeny čtyři testovací lety z Kotvrdovického letiště, které se nachází zhruba 40 km severovýchodně od Brna. První dva lety odhalily chyby v implementaci, kterou jsem následně odstranil. Poslední dva lety již považuji za úspěšné. V době prvního a třetího testovacího letu bylo zařízení napájeno z baterií. Během zbylých dvou letů bylo zařízení napájeno pomocí adaptéru na 12 V.

První let

V rámci prvního letu byla odhalena stejná chyba v ukládání vysokých hodnot přírůstků nadmořské výšky a rychlosti. Vzhledem k použitému kódování dat jsou bohužel všechny následující hodnoty poznamenány touto chybou. Trajektorie letu byla téměř v pořádku.



Obrázek 4.13: Ověření činnosti systému s krátkodobými výpadky GPS signálu

Se samotným příjmem a přesností GPS signálu nebyl téměř žádný problém. Drobné výpadky signálu nastaly zejména při přeletu Brna, kde ovšem nastalo prudké zhoršení počasí. V nejhorším případě trval výpadek 30 sekund. Důležité bylo, že následující hodnoty GPS souřadnic nebyly těmito výpadky poznamenány. Vzhledem k problémům, které nastaly v ukládání, bylo možné zaznamenat pouze 41 minut letu, které jsou tvořeny 1229 body. Trajektorie tohoto letu je na obrázku 4.14.

Druhý let

Druhý a zároveň poslední let, během kterého byla odhalena chyba v implementaci. Konkrétně se jednalo o chybu v dekódování věty typu GGA, ze které jsou vyčítány hodnoty GPS souřadnic a nadmořské výšky. Tato chyba zapříčinila chybné označení většiny vět GGA za nevalidní. Z tohoto důvodu se podařilo rekonstruovat trajektorii pouze poslední fáze letu, která zahrnovala návrat na letiště, přiblížení a samotné přistání. Mimo jiné má vzniklá chyba za následek posunutí GPS souřadnic. Hodnoty rychlosti a úhlů v osách x a y jsou zaznamenány z celého letu. U nich žádný problém nenastal. Hodnoty rychlosti v době přistání odpovídají realitě. Na základě toho lze říci, že předchozí chyba byla odstraněna. Chyba v dekódování GGA věty byla velmi rychle nalezena a odstraněna. Následující dva lety měly toto tvrzení ověřit. Trajektorie poslední fáze letu je na obrázku 4.15. Celková doba trvání letu byla 33 minut, během kterých bylo zaznamenáno 999 bodů.

Třetí let

Tento let byl odlišný od všech ostatních tím, že byl uskutečněn pomocí ultralehkého vrtulníku namísto ultralehkého letadla. Během letu nedošlo k žádné další chybě a sesbíraná



Obrázek 4.14: Trajektorie prvního pokusného letu



Obrázek 4.15: Trajektorie druhého pokusného letu

data lze považovat za validní. Pomocí kontrolních součtů bylo zjištěno, že u pár bodů došlo k chybě při migraci zaznamenaných dat do paměti. Tyto neplatné hodnoty mají za následek drobné nepřesnosti v hodnotách nadmořské výšky v době přistání. Hovoříme zde ovšem o chybě do 30 metrů. Je velice pravděpodobné, že opakované provedení migrace obsahu datového úložiště by tuto chybu odstranilo. Toto tvrzení ovšem nebylo možné ověřit, jelikož obsah datového úložiště byl v době zjištění již smazán. I tak považuji test za úspěšný. Chyba totiž nezvnikla na straně zařízení, o čemž se lze přesvědčit pohledem na příznaky platnosti jednotlivých sad v přijatém souboru. To znamená, že hodnoty byly do paměti uloženy nejspíše v pořádku. Ukázka trajektorie letu, který trval 29 minut, je na obrázku 4.16. Trajektorie letu je tvořena 872 body. Podíl vysokých hodnot přírůstků byl 1,6 %.

Čtvrtý let

Poslední testovací let byl proveden opět pomocí ultralehkého letadla. Kontrolní součty všech sad letových dat tohoto letu jsou v pořádku. To znamená, že nedošlo k chybě zápisu do

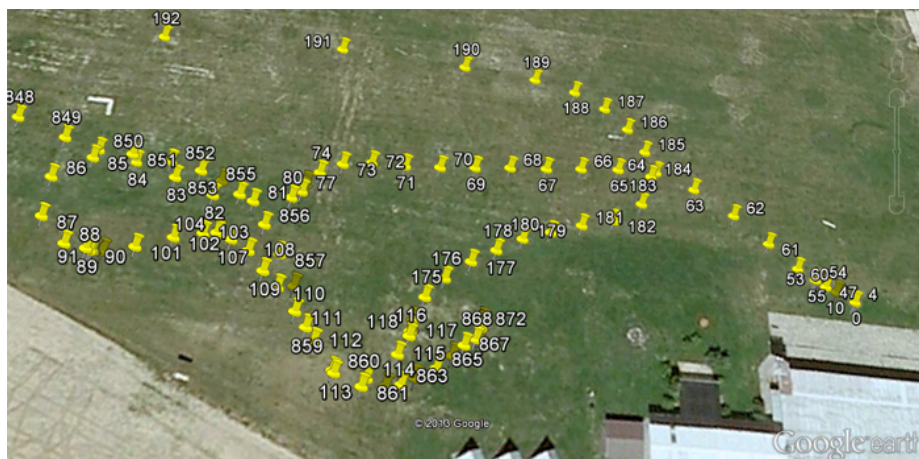


Obrázek 4.16: Trajektorie třetího pokusného letu

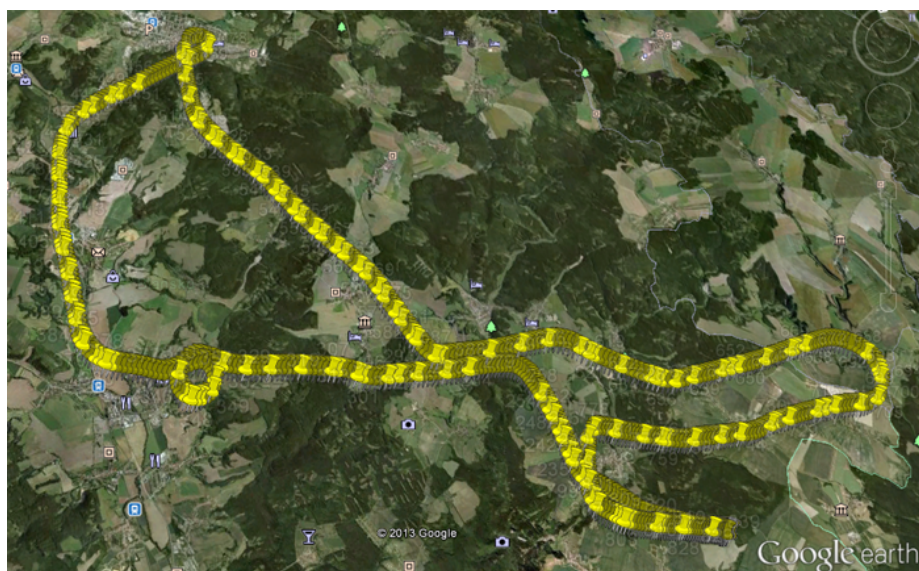
paměti nebo chybě při migraci obsahu datového úložiště. Výsledkem je, že GPS souřadnice v době přistání jsou téměř identické jako v době startu. Přesnost GPS je tedy naprosto vyhovující. Hodnoty rychlosti v době přistání a rolování taktéž odpovídají skutečnosti. Hodnoty nadmořské výšky v době vzletu a přistání se liší v řádech metrů, což je vzhledem k přesnosti použitého GPS modulu v pořádku. Hodnoty náklonů v obou osách jsou taktéž validní. Trajektorie v době přistání a vzletu je na obrázku 4.17. Trajektorie před vzletem a po přistání je na obrázku 4.18 a celková trajektorie letu je na obrázku 4.19. Doba trvání letu byla necelých 30 minut, během kterých bylo sesbíráno 878 bodů. Podíl vysokých hodnot přírůstků byl 0,25 %.



Obrázek 4.17: Trajektorie čtvrtého pokusného letu v době přistání a vzletu



Obrázek 4.18: Trajektorie čtvrtého pokusného letu v době rolování



Obrázek 4.19: Trajektorie čtvrtého pokusného letu

4.7 Cenové aspekty zařízení

Jedním z klíčových parametrů navrhovaného zařízení byla jeho cena. V tomto projektu je nejdůležitější pořizovací cena hardwaru. Kity eZ430-RF2500 a eZ430-C9 lze pořídit od společnosti Mouser Electronics za 2 232 Kč vč. DPH. GPS modul LR9540G a aktivní GPS anténu Mounting 32 má v nabídce společnost SECTRON s.r.o. za 891 Kč vč. DPH. Posledním komponentem je Autoadaptér pro nabíječky ipc-1 a ipc-11, které lze zakoupit přímo od výrobce, společnosti FK technics spol. s r.o., za 229 Kč vč. DPH. Celková cena za použitý hardware je tedy 3 352 Kč vč. DPH.

Jedná se ovšem o cenu, která zahrnuje vývojové kity mikrokontroléru a akcelerometru. Společnost Mouser Electronics nabízí i kit eZ430-RF2500T, který obsahuje stejný bateriový modul a stejnou cílovou desku jako kit eZ430-RF2500. Kit eZ430-RF2500T tedy obsahuje vše potřebné pro fungování navrženého zařízení. Tedy s výjimkou samotného naprogramování mikrokontroléru. Stejně tak je možné zakoupit samotný akcelerometr u té samé

společnosti. Cena za tyto dvě komponenty poté klesne z 2 232 Kč vč. DPH na 964 Kč vč. DPH. Po této záměně je ovšem nutné počítat s nutností vytvoření jednoduchého plošného spoje pro akcelerometr a zpětný kanál rozhraní UART.

Kapitola 5

Závěr

Cílem práce bylo vytvořit zařízení pro sběr leteckých dat, které bude postaveno nad mikropočítačem. Na základě výsledků testování na palubě ultralehkého letadla a vrtulníku mohu říci, že tento cíl byl splněn. K tomu bylo nezbytné nastudování problematiky mikropočítačů, možností snímání leteckých dat a principů vytváření vestavěných systémů. Na základě těchto poznatků jsem poté mohl vytvořit návrh zařízení, které by dokázalo zaznamenávat GPS souřadnice, nadmořskou výšku, rychlost a náklony letadla v čase. Tento návrh jsem zrealizoval pomocí komponent, které jsou v současné době v praxi běžně používány. Následně jsem prováděl testování celého systému, které vyvrcholilo testováním v cílovém prostředí na palubě ultralehkého letadla a vrtulníku. Dosažené výsledky a možnosti navrženého řešení jsem konzultoval s vedoucím mé bakalářské práce, který je současně aktivním pilotem a za jehož pomoci jsem mohl otestovat zařízení v cílovém prostředí.

Navržené zařízení umožňuje snímat letecká data až po dobu jedné a půl hodiny. Tato doba je dána koncepcí navrhovaného řešení. Výhodou představeného řešení je nezávislost na cílovém prostředí. Ta byla ověřena v průběhu testování, které bylo mimo jiné realizováno na palubě dvou odlišných ultralehkých letadel a jednoho ultralehkého vrtulníku. Zároveň lze říci, že se jedná o finančně dostupné řešení. Celková cena totiž nepřesáhla částku tří a půl tisíc korun včetně DPH. Značnou část přitom tvořily vývojové prostředky, které je možné opakovaně využít.

Největším přínosem pro mě bylo seznámení se s návrhem a možnostmi využití vestavěných systémů, kterým bych se rád věnoval i nadále v magisterském studiu a budoucí profesi. Mimo to mi tato práce umožnila aktivní účast na testování na palubě ultralehkého letadla, což byl jeden z důvodů výběru tohoto tématu.

V této práci plánuji pokračovat i nadále. V současné době je vytvořeno zařízení, na kterém bylo možné ověřit možnosti využití mikropočítače na palubě ultralehkého letadla pro sběr leteckých dat. Na něm je možné přímo či nepřímo dále pracovat a vytvořit tak komplexnější zařízení. Zda bude sloužit jako bezpečnostní nebo informační prvek, je už otázkou poptávky. Osobně bych upřednostnil vývoj zařízení pro informační účely, které by dokázalo snímat větší množství dat. Nevylučuji ovšem ani druhou variantu, jelikož principiálně se jedná o velmi podobná zařízení. Na pokračování vývoje tohoto druhu zařízení jsem předběžně domluven s mým vedoucím bakalářské práce profesorem Pavlem Zemčíkem.

Literatura

- [1] BIERL, L.: *MSP430 Family Mixed-Signal Microcontroller*. January 2000, [Online], [cit. 2013-04-01].
URL <http://www.ti.com/lit/an/slaa024/slaa024.pdf>
- [2] DAVIES, J. H.: *MSP430 Microcontroller Basics*. Elsevier Science, 2008, ISBN 978-0-7506-8276-3.
- [3] eZ430-RF2500 Development Tool. April 2009, [Online], [cit. 2013-03-23].
URL <http://www.ti.com/lit/ug/slau227e/slau227e.pdf>
- [4] GASPAR, P. D.; SANTO, A. E.; RIBEIRO, B.; aj.: Device Systems, Operating Modes and General Purpose Input/Output. 2009, [Online], [cit. 2013-03-10].
URL <http://ares.ee.nchu.edu.tw/Course.files/epr100/lecture4.pdf>
- [5] Götthans, T.: GPS Receiver. Technická zpráva, Ústav radioelektroniky, FEKT VUT, 2009, [Online], [cit. 2013-05-01].
URL <http://www.urel.feec.vutbr.cz/MIA/2009/gotthans/index.html>
- [6] Krčmář, P.: Neo FreeRunner: co se skrývá uvnitř - Root.cz. October 2008.
URL <http://www.root.cz/clanky/neo-freerunner-co-se-skrывa-uvnitř/>
- [7] KXTC9 Series Accelerometers and Inclinometers. November 2010, [Online], [cit. 2013-03-19].
URL
<http://www.kionix.com/sites/default/files/KXTC9%20Product%20Summary.pdf>
- [8] KXTC9-2050 Specifications. June 2010, [Online], [cit. 2013-03-19].
URL http://www.kionix.com/sites/default/files/KXTC9-2050%20Specifications%20Rev%201_0.pdf
- [9] LEADTEK GPS MODULE LR9540G. April 2008, [Online], Verze 0.5, [cit. 2013-03-20].
URL ftp://ftp.leadtek.com.tw/gps/9540/9540_9805ST_v0.5_041408.pdf
- [10] LEADTEK GPS MODULE LR9805ST. May 2008, [Online], Verze 0.9, [cit. 2013-03-20].
URL ftp://ftp.leadtek.com/gps/9805st/LR9805ST_V0.9_052808.pdf
- [11] LITOVSKY, G.: Beginning Microcontrollers with the MSP430. [Online], Verze 0.4, [cit. 2013-03-20].
URL http://www.glitovsky.com/Tutorialv0_4.pdf

- [12] MSP430™ Ultra-Low-Power Microcontrollers. 2012, [Online], [cit. 2013-04-02].
URL <http://www.ti.com/lit/sg/slab034v/slab034v.pdf>
- [13] MSP430x2xx Family. January 2012, [Online], [cit. 2013-03-03].
URL <http://www.ti.com/lit/ug/slau144i/slau144i.pdf>
- [14] MUEHLHOFER, A.: *MSP430 Flash Self-Programming Technique*. November 2000, [Online], [cit. 2013-03-13].
URL <http://www.gaw.ru/pdf/TI/app/msp430/slaa103.pdf>
- [15] NMEA Reference Manual. December 2007, [Online], Verze 2.1, [cit. 2013-03-25].
URL <http://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual-Rev2.1-Dec07.pdf>
- [16] SMITH, S. W.: *The scientist and engineer's guide to digital signal processing*, kapitola Data Compression. San Diego: California Technical Publishing, druhé vydání, 1999, ISBN 0-9660176-6-8, s. 481–502.
- [17] WESTLUND, L.: Using the Timer_A UART Library. July 2007, [Online], [cit. 2013-04-24].
URL <http://www.ti.com/lit/an/slaa307a/slaa307a.pdf>
- [18] WWW stránky: EZ430 Backchannel UART Configuration - Texas Instruments Embedded Processors Wiki. [Online], [cit. 2013-04-04].
URL http://processors.wiki.ti.com/index.php/EZ430_Backchannel_UART_Configuration
- [19] WWW stránky: Flight Data Monitoring - Appareo Systems. [Online], [cit. 2013-03-25].
URL <http://www.appareo.com/cat/primarymenu/products/alerts-flight-data-monitoring/>
- [20] WWW stránky: Index of /home/X34SES/prednasky. [Online], [cit. 2013-03-19].
URL <http://www.micro.feld.cvut.cz/home/X34SES/prednasky/>
- [21] WWW stránky: Kionix KX022 Series MEMS Motion Sensing Acceleromete. [Online], [cit. 2013-03-19].
URL <http://www.kionix.com/accelerometers/kx022>
- [22] WWW stránky: MSP430 Microcontroller — 16 bit Microcontroller — 16 bit MCU - Getting Started - TI.com. [Online], [cit. 2013-03-20].
URL http://www.ti.com/llds/ti/microcontroller/16-bit_msp430/getting-started.page/
- [23] WWW stránky: MSP430 Microcontroller — 16 bit Microcontroller — 16 bit MCU - Tools & Software - TI.com. [Online], [cit. 2013-03-23].
URL www.ti.com/ez430
- [24] WWW stránky: MTVP. [Online], [cit. 2013-03-19].
URL <http://www.umel.feec.vutbr.cz/MTVP/>
- [25] WWW stránky: NMEA. [Online], [cit. 2013-03-25].
URL http://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp

- [26] WWW stránky: Wi-Flight - Products. [Online], [cit. 2013-03-25].
URL <https://www.wi-flight.net/products.html>
- [27] WWW stránky: Basic Clock Module, Oscillator and Clock Generator. September 1998, [Online], [cit. 2013-03-03].
URL <http://www.datasheetarchive.com/SLAU023-datasheet.html>
- [28] WWW stránky: Using the Timer_A UART Library (Rev. A), slaa307a - TI.com. August 2007, [Online], [cit. 2013-04-28].
URL <http://www.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=slaa307a>
- [29] Zákon č. 49/1997 Sb., o civilním letectví a o změně a doplnění zákona č. 455/1991 Sb., o živnostenském podnikání (živnostenský zákon), ve znění pozdějších předpisů. In: Sbírka zákonů. 28. 3. 1997, částka 142. [Online], [cit. 2013-04-30].
URL
http://www.mdcr.cz/NR/rdonlyres/FDF949DE-E2BC-4504-BB16-ACF47EC82706/0/49_1997_konsolidovane_zneni_cz_duben_2012.pdf

Seznam příloh

A Obsah CD	61
B Obsluha přerušení portu P1	62
C Obsluha přerušení portu P2	63
C.1 Příznak P2IGF_bit.P7	63
C.2 Příznak P2IGF_bit.P6	63

Příloha A

Obsah CD

dokumentace: Zdrojové soubory tohoto textu psaného v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

projektIAR: Projek pro vývojové prostředí IAR Embedded Workbench verze 5.51 se zdrojovými soubory systému.

data\bp.py: Skript v jazyce Python verze 3.3.0 pro rekonstrukci dat.

data\testy.txt: Popis přiložených testů.

data\vstup: Adresář se zdrojovými soubory z testování.

data\vystup: Adresář se soubory KML a TXT z testování.

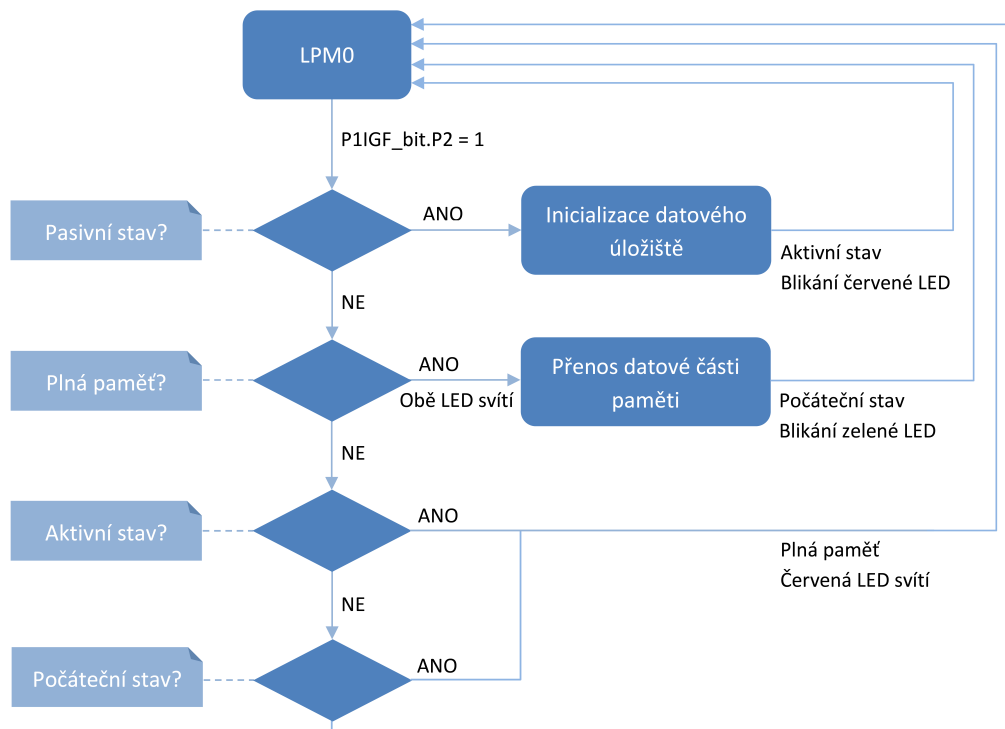
projekt.pdf: Technická zpráva.

readme.txt: Obsah CD.

Příloha B

Obsluha přerušení portu P1

V rámci této obsluhy přerušení, jejíž funkční schéma je na obrázku B.1, je prováděna změna stavu zařízení na základě stisku tlačítka uživatelem. Ten může zahájit sběr leteckých dat přechodem z pasivního stavu, který je indikován svícením zelené LED, do aktivního stavu. Před samotným zahájením sběru dat je provedena inicializace datového úložiště. Sběr leteckých dat je ukončen automaticky při zaplnění paměti nebo manuálně stiskem tlačítka uživatelem v aktivním nebo počátečním stavu. Samotná migrace obsahu datového úložiště je zahájena stiskem tlačítka ve stavu plná paměť. Po dobu přenosu obsahu datového úložiště svítí červená i zelená LED. Po dokončení přenosu obsahu datového úložiště zařízení přechází do počátečního stavu, který je indikován blikáním zelené LED.



Obrázek B.1: Funkční schéma obsluhy přerušení pro port P1

Příloha C

Obsluha přerušeni portu P2

Funkční schéma obsluhy přerušeni je na obrázku [C.1](#). V rámci obsluhy přerušeni je řízena veškerá komunikace knihovny pro realizace sériové komunikace pomocí časovače `Timer_A` a hlavního programu. Jsou rozlišovány příznaky `P2IGF_bit.P7` a `P2IGF_bit.P6`.

C.1 Příznak `P2IGF_bit.P7`

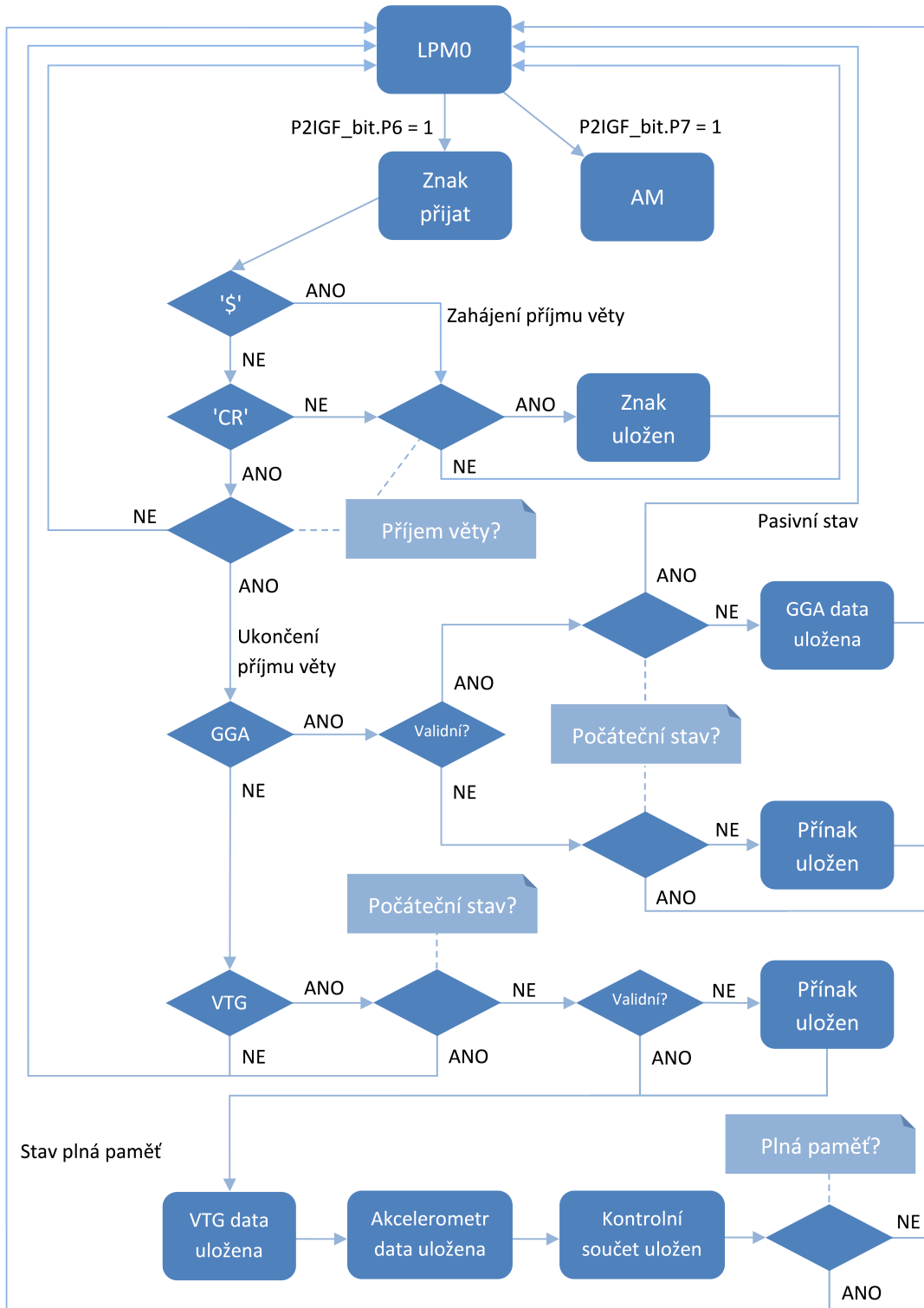
Významově je tento příznak shodný s příznakem `UCAnTXIFG`, který nalezneme u modulu `USCI` mikrokontrolérů z rodiny `MSP430`. Odesílání bajtů po sériové lince do GPS modulu pomocí knihovny totiž spočívá ve vložení odesílané hodnoty do příslušné proměnné, která slouží jako náhrada posuvného registru, a přechodu do režimu `LPM0`. Tím je vykonávání kódu pozastaveno až do doby, než je dokončeno odesílání bajtu, který byl vložen do proměnné. Samotné pokračování ve vykonávání kódu je realizováno přechodem do režimu `AM` z režimu `LPM0` uvnitř obslužné rutiny. Z pohledu implementace je tedy použití knihovny stejné jako použití modulu `USCI` v režimu `UART`, který je v tomto systému taktéž využíván.

C.2 Příznak `P2IGF_bit.P6`

Tento příznak slouží pro indikaci přijetí jednoho bajtu skrze sériovou linku od GPS modulu. Významově je opět totožný s příznakem modulu `USCI`. Konkrétně s příznakem `UCAnRXIFG`. Samotný příznak `P2IGF_bit.P6` je nastavován v rámci obslužné rutiny časovače `Timer_A` po přijetí celého datového rámce a vložení přijatého bajtu do příslušné proměnné, která opět supluje posuvný registr.

V rámci obslužné rutiny jsou s nastaveným příznakem příjmu věty jednotlivé přijaté znaky ukládány do pomocného bufferu. Příznak je nastavován při přijetí znaku '\$' a naopak smazán při přijetí znaku 'CR', kterým je každá NMEA věta ukončena. Současně s rozpoznáním znaku 'CR' je přijatá věta dekodována. Akceptovány jsou pouze věty `GGA` a `VTG`. Žádné jiné by ani po správné konfiguraci GPS modulu neměly být přijaty. U každé věty je kontrolována validnost pomocí jednoduchého kontrolního součtu, který je součástí každé NMEA věty. Pomocné funkce pro výpočet kontrolního součtu byly převzaty z projektu `Control of airplane model servo system through TI MSP430 micro-controller`, který mi poskytl vedoucí bakalářské práce. Samotný výpočet kontrolního součtu je mimo jiné definován uvnitř standardu `NMEA 0183`. Mimo kontrolní součet je také kontrolována délka samotné přijaté věty. GPS modul totiž pracuje tak, že stále zasílá nakonfigurované NMEA věty bez ohledu na sílu GPS signálu. Je tedy čistě v režii příjemce, aby provedl kontrolu

a vyhodnocení stavu GPS přijímače. Příliš krátká věta značí, že úroveň signálu GPS přijímače není dostatečná a nebylo možné vypočítat jednotlivé položky dané věty. Tohoto principu je využito při přechodu z počátečního stavu do pasivního při přijetí validní GGA věty. Pasivní stav totiž značí připravenost systému pro zahájení sběru leteckých dat. Nevalidní věta GGA je v počátečním stavu systému naopak zahazována. Druhým stavem, ve kterém může být přijata věty GGA, je aktivní stav. Při přijetí validní věty GGA v tomto stavu jsou uloženy hodnoty GPS souřadnic a nadmořské výšky do paměti. U nevalidní věty jsou uloženy příznaky neplatných dat. Věta VTG může být přijata v pasivním nebo aktivním stavu. Akceptována je pouze v aktivním. Stejně jako v případě věty GGA je u věty VTG stejným způsobem kontrolována validnost. Při přijetí validní věty je uložena hodnota rychlosti do paměti. U nevalidní věty je naopak uložen příznak neplatnosti. V obou případech následuje zpracování a uložení aktuálních hodnot výstupu akcelerometru. Poté je proveden přepočítání kontrolního součtu z uložených hodnot dané sady. Na základě porovnání právě vypočítané hodnoty s hodnotou, která je počítána z hodnot, které měly být uloženy v rámci této sady dat, je uložen příznak platnosti nebo neplatnosti hodnot dané sady do paměti. Na závěr obslužné rutiny je provedena kontrola zbývající kapacity datového úložiště. Ta by měla být větší nebo rovna velikosti nejhoršího vzorku dat, kterým může být tvořena sada hodnot. Při nedostatku místa přechází zařízení do stavu plná paměť.



Obrázek C.1: Funkční schéma obsluhy přerušování pro port P2