



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra měření

Snímač polohy využívající obrazové informace

Image Based Position Sensor

Diplomová práce

Praha, leden 2011

Autor: Petr Bělík

Vedoucí práce: doc. Ing. Jan Fischer, CSc.

Studijní program: Kybernetika a měření

Studijní obor: Měřicí a přístrojové systémy



ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Petr Bělík**

Obor: **Kybernetika a měření – blok Měřicí a přístrojové systémy**

Název tématu česky: **Snímač polohy využívající obrazové informace**

Název tématu anglicky: **Image Based Position Sensor**

Zásady pro vypracování:

Navrhněte a realizujte metody zpracování obrazu scény snímané senzorem CMOS, které budou poskytovat informaci o poloze, využitelnou pro regulaci polohy objektů. Orientujte se také na možnost průběžného zpracování digitalizovaného videosignálu signálovým procesorem ADSP BF-532 tak, aby se informace o poloze a orientaci objektu (ve 2D a ve 3D) získala s minimálním zpožděním. Navrhněte snímač polohy, který bude obsahovat minimální sestavu obrazového senzoru CMOS a signálového procesoru řady ADSP-BF pro specializované úkoly určování polohy objektů a její regulaci. Navržené metody implementujte a prakticky ověřte.

Seznam odborné literatury:

- [1] Fischer, J.: Optoelektronické senzory a videometrie. Skripta ČVUT, Praha 2002.
- [2] Heijden F.: Image Based Measurement Systems. Hoboken, NJ, Wiley, 1994.
- [3] Analog Devices: ADSP BF53x User's manual

Vedoucí diplomové práce: Ing. Jan Fischer, CSc.

Datum zadání diplomové práce: 26. listopadu 2009

Platnost zadání do¹: 10. června 2011

Prof. Ing. Pavel Ripka, CSc.
vedoucí katedry



Doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 26. 11. 2009

¹ Platnost zadání je omezena na dobu tří následujících semestrů.

Anotace

Diplomová práce se zabývá vývojem snímače polohy s CMOS obrazovým senzorem a signálovým procesorem Blackfin ADSP BF-532. Hlavními rysy zařízení jsou minimální obvodová konfigurace, průběžné zpracování obrazové informace ze senzoru a rychlé měření polohy objektu pro využití v nejrůznějších úlohách regulace. Snímač polohy je realizován na plošném spoji se stejnými rozměry jako má deska obrazového senzoru a je opatřen konektorem pro vzájemné modulární propojení těchto dvou zařízení. Dále disponuje řadičem High-Speed USB 2.0 pro přenos obrazu, komunikaci s PC a programování bootovací paměti procesoru. Kromě rozhraní USB je snímač polohy vybaven ještě procesorovými rozhraními UART, SPORT, JTAG a GPIO, která jsou vyvedena na konektory a mohou být použita pro mnohé aplikace v měřicí technice. Nedílnou součástí práce je software pro spolupráci s PC a vlastní implementace algoritmů.

Annotation

This thesis deals with the development of position measuring element with CMOS image sensor and DSP Blackfin ADSP BF-532. The main features of this device are minimal circuit configuration, On-The-Fly image processing and rapid position measurement for use in various regulation tasks. The position sensor is implemented on the PCB with the same format as the CMOS sensor board and it has a connector for modular connection of these two devices. The sensing device has also a High-Speed USB 2.0 controller for image transmission, communication with PC and programming processor booting memory. Besides USB interface, there are also UART, SPORT, JTAG and GPIOs, which are available on connectors for the next various usage in the applied measurement engineering. An integral part of this thesis is software for cooperation with PC and the appropriate implementation of algorithms.

Čestné prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Tato práce vznikla v laboratoři videometrie, katedry měření ČVUT - FEL v Praze pod vedením doc. Ing. Jana Fischera, CSc. Navazuje též na výzkum v rámci MSM6840770015 - "Výzkum metod a systémů pro měření fyzikálních veličin a zpracování naměřených dat", jehož některé poznatky a výstupy v oblasti optoelektronických senzorů také využívá.

V Praze dne

.....

Podpis autora práce

Poděkování

Děkuji především doc. Ing. Janu Fischerovi, CSc. za aktivní vedení práce, mnoho přínosných konzultací a snahu o okamžité řešení vzniklých problémů. Dále děkuji doktorandům laboratoře videometrie katedry měření Ing. Ondreji Pribulovi, Ing. Janu Šedivému a Ing. Janu Pohankovi za cenné rady během vývoje zařízení. Upřímné poděkování patří také mým rodičům za pochopení a stoprocentní podporu během celého studia a mým dalším blízkým za pomoc s korekcí chyb v tomto textu.

Obsah

1 Úvod.....	9
1.1 Měření polohy	9
1.2 Použitelnost obrazových sensorů CMOS pro snímače polohy	10
1.3 Princip funkce CMOS sensorů.....	10
1.4 Určení polohy objektu z obrazových dat.....	12
1.5 Princip funkce PSD sensorů jejich využití k měření polohy	13
1.6 Určení polohy objektu z obrazových dat signálovým procesorem	15
1.7 Cíl práce a představa o výsledné podobě snímače polohy	16
2 Postup vývoje vlastního hardware.....	18
2.1 Shrnutí požadavků na snímač polohy z hardwarového hlediska.....	18
2.2 Volba USB řadiče.....	19
2.2.1 FT245R s programováním EEPROM v režimu Bit Bang.....	21
2.2.1.1 Ovládání asynchronního Bit Bang módu USB řadiče FT245R na straně PC pod operačním systémem Windows	23
2.2.1.2 Obslužný program řadiče FT245R pro generování programovacích průběhů na SPI sběrnici	26
2.2.1.3 Zhodnocení a uplatnění programátoru EEPROM 25LC256 v módu Bit Bang u obvodu FT245R	29
2.2.2 FT232R s programováním EEPROM v režimu Bit Bang.....	30
2.2.3 Obvod FT2232H ve funkci programátoru EEPROM a USB řadiče pro rychlý přenos obrazových dat	31
2.2.3.1 Popis některých konfigurovatelných režimů řadiče FT2232H a možnost jejich uplatnění v zapojení snímače polohy	32
2.2.3.2 Výsledná koncepce zapojení řadiče FT2232H pro přenos dat z procesoru BF-53x a programování bootovací paměti EEPROM.....	34
2.2.3.3 Programování EEPROM obvodem FT2232H v režimu MPSSE SPI.....	35
2.2.3.4 Přenos dat USB řadičem FT2232H mezi PC a BF-53x.....	38
2.2.3.5 Zhodnocení vlastností programátoru EEPROM s FT2232H v režimu MPSSE SPI a paralelního 8-bitového přenosu dat.....	40

2.3	Processorová část snímače polohy.....	43
2.3.1	Stručný přehled vlastností procesorů Blackfin řady BF-53x.....	43
2.3.2	Propojení procesoru BF-53x s obrazovým senzorem.....	44
2.3.2.1	Komunikace I ² C	45
2.3.2.2	Řídicí signály.....	47
2.3.2.3	Signály pro přenos obrazových dat	47
2.3.2.4	Synchronizační signály	48
2.3.2.5	Hodinové signály.....	49
2.3.2.6	Rozvod napájecího napětí	49
2.3.3	Periferní obvody procesoru	50
2.3.3.1	Hodinový systém.....	50
2.3.3.2	Resetovací obvod	51
2.3.3.3	Bootování procesoru a paměť EEPROM	52
2.3.4	Uživatelská rozhraní snímače polohy	54
2.3.4.1	UART, časovač, sériová linka RS-232/485.....	54
2.3.4.2	Rozhraní GPIO (Programmable Flags)	55
2.3.4.3	Sériový port SPORT.....	57
2.3.4.4	Ladicí rozhraní JTAG.....	58
2.3.5	Napájecí systém.....	59
3	Softwarová část vývoje.....	61
3.1	Přenos obrazu ze senzoru do PC	61
3.1.1	Nastavení GPIO linek a konfigurace jednotky PLL.....	61
3.1.2	Rychlý přenos velkého objemu dat z procesoru do USB řadiče	63
3.1.2.1	Maximalizace zapisovací rychlosti procesoru do řadiče FT2232H	65
3.1.2.2	Zpomalení toku dat z obrazového senzoru.....	68
3.1.2.3	Kontinuální vyčítání dat z USB zařízení na straně PC a jejich následné zpracování	70
3.1.3	Konfigurace PPI rozhraní a DMA kanálu pro přenos dat z CMOS senzoru do vnitřní paměti	72
3.1.4	Softwarová implementace I ² C rozhraní a nastavení registrů CMOS senzoru	74

3.2	Hledání těžiště světelné stopy	76
3.2.1	Výpočetní metoda určení těžiště světelné stopy z obrazových dat	76
3.2.2	Rozbor algoritmu pro výpočet těžiště jedné světelné stopy pouze s vnitřní pamětí SRAM	77
3.2.3	Situace při více světelných stopách v obraze	79
3.2.4	Nastavení obrazového senzoru, přenos informace o poloze objektu do PC a grafické vyobrazení v demonstrační aplikaci	80
4	Závěr.....	82
4.1	Shrnutí dosažených výsledků	82
4.1.1	Měření polohy světelné stopy	83
4.1.2	Přenos obrazu ze senzoru v plném rozlišení do počítače	84
4.1.3	Návrh plošného spoje	84
4.2	Perspektiva a možnosti využití snímače polohy	85
4.2.1	Integrace zařízení do prostředí LabView a packetová komunikace mezi snímačem polohy a PC.....	85
4.2.2	Další aplikovatelnost zařízení v úlohách měřicí techniky.....	86
5	Seznam obrázků a tabulek	87
6	Seznam odborné literatury	89
7	Přílohy	93
7.1	Příloha A: Obsah přiloženého CD.....	93
7.2	Příloha B: Fotodokumentace	95
7.3	Příloha C: Výkresy DPS (60 × 60 mm) a osazovací předlohy.....	98
7.4	Příloha D: Soupiska součástek	100
7.5	Příloha E: Výkres výsledného schématu zapojení snímače polohy na listu A3.....	101

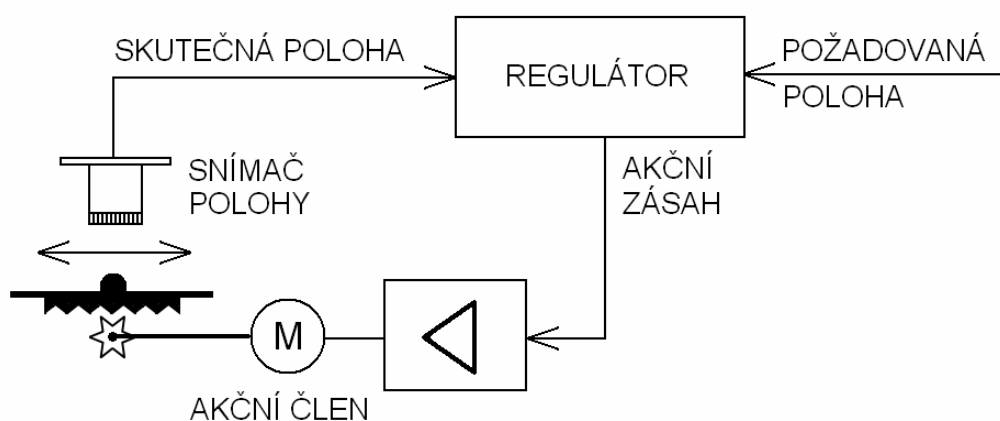
1 Úvod

1.1 Měření polohy

Přesné a rychlé měření polohy určitých objektů je v současné době požadováno v mnoha odvětvích lidské činnosti. Jestliže je poloha zaznamenávána v čase, pak je možné polohu analyzovat jako pohyb, čehož se využívá například v:

- *průmyslu* – řízení strojů, robotů, nárazové testy automobilů, sledování stopy,
- *zdravotnictví* – prevence úrazů, rehabilitace, předoperační vyšetření, virtuální operace,
- *fyzioterapie* – analýza chůze, hodnocení výkonu, trénování, efektivita rehabilitace,
- *psychologie* – interakce ve virtuální realitě, studie chování lidí a zvířat,
- *animace obrazu* – speciální efekty, oživení videoher, kreslené filmy, choreografie a další, viz [1].

Oborově nejbližší příklad využití informace o poloze může být v automatizaci, kdy regulační smyčku tvoří snímač polohy, regulátor a akční člen (polohovací mechanismus), viz obr. 1.1. Poloha objektu je v tomto případě měřena snímačem polohy, a pokud se liší od polohy požadované, polohovací mechanismus ji začne okamžitě regulovat. Na tomto příkladě se také vysvětluje požadavek na rychlost poskytování informace o poloze – vliv velkého zpoždění informace o poloze od změny polohy objektu by způsoboval velkou setrvačnost při regulaci.



Obr. 1.1 Schematické znázornění úlohy snímače polohy v regulační smyčce

V současnosti existuje velká řada snímačů polohy, které fungují na nejrůznějších fyzikálních principech. Použití vhodného typu snímače vždy záleží na konkrétní aplikaci. Pro ilustraci, snímače polohy lze zjednodušeně rozdělit podle principu na *odporové*, *indukčnostní*, *kapacitní* a *optoelektronické*. Bližší rozbor jednotlivých typů je vysvětlen v [2]. Tato práce se zaměřuje pouze na úzkou část optoelektronických snímačů polohy, konkrétně na snímače polohy

s obrazovými senzory CMOS, doplněných vhodným objektivem. Více je později zmíněno v kapitolách 1.2 a 1.3. Optoelektronické snímače patří do skupiny bezdotykových snímačů, tzn. měřený systém není snímačem prakticky ovlivňován a odpadá potřeba kontaktního spojení systému a snímače, což je v některých případech požadavek zcela zásadní.

Polohu lze určovat jednak v 2D rovině s jedním snímačem, ale i v 3D prostoru pomocí snímačů dvou, které se spolu s měřeným objektem nastaví do určité geometrické konfigurace, z níž pak metody pro určení souřadnic vycházejí. Podrobněji pojednává o určování souřadnic polohy v prostoru a o stereovidění např. [3].

1.2 Použitelnost obrazových senzorů CMOS pro snímače polohy

Za poslední desetiletí došlo k mohutnému nárůstu rozvoje obrazových senzorů CMOS a dnes je možné se s nimi setkat prakticky ve všech oblastech. V podstatě lze CMOS senzory rozdělit do dvou základních kategorií:

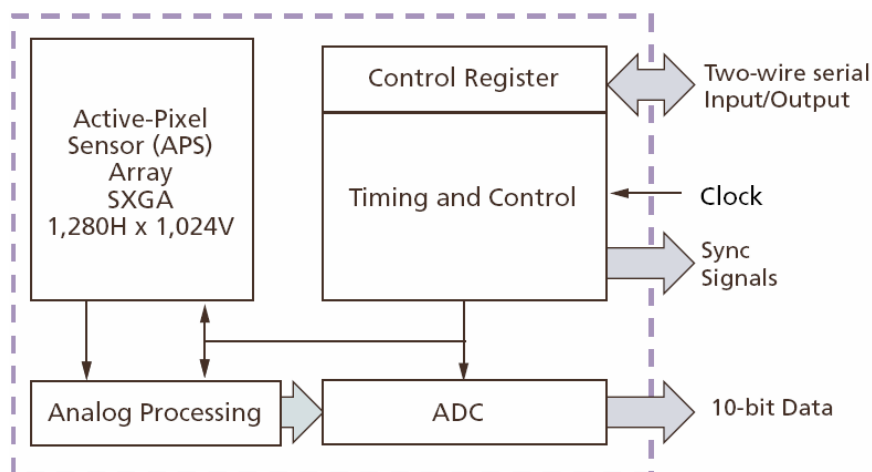
- „*low-end*“ (senzory masově používané ve spotřební elektronice jako jsou mobilní telefony, webkamery, osobní automobily, čtečky čárových kódů, zabezpečovací technika, hračky) a
- „*high-end*“ (velmi kvalitní senzory s vysokým rozlišením, které mají uplatnění např. ve fotoaparátech, HD kamerách, lékařství, průmyslu, strojovém vidění atd.).

Se vzrůstajícím uplatněním CMOS senzorů na trhu a s jejich masovou výrobou dochází zákonitě k poklesu jejich pořizovací ceny, což platí u obou výše uvedených kategorií [4].

Relativně nízká cena a dostupnost CMOS obrazových senzorů vedla k myšlence použití při vývoji snímače polohy, kdy z původního určení – *senzoru obrazu* – má vzniknout *senzor polohy*. Mělo tak dojít k „degradaci“ obrazového senzoru k principiálně blízkému PSD senzoru. Oba dva principy jsou uvedeny v následujících kapitolách.

1.3 Princip funkce CMOS senzorů

Princip plošného obrazového CMOS senzoru je vysvětlen podle blokového schématu senzoru *Micron* MT9M001 na obr. 1.2. Vnitřní bloková zapojení CMOS senzorů různých výrobců jsou pochopitelně částečně odlišná (možnost nastavení děličky hodin, typ elektronické závěrky, RGB obraz, množství konfigurovatelných parametrů, atd.), avšak základní princip zůstává stejný.



Obr. 1.2 Blokové schéma plošného obrazového senzoru CMOS (Micron MT9M001) ¹

Získání obrazové informace začíná v poli fotoelektrických elementů, což je uspořádaná matice buněk, sestávajících z MOS tranzistorů a fotodiody. Před expozicí jsou všechny buňky elektricky nabity (parazitní kapacity závěrně polarizovaných PN přechodů fotodiody). Světelná dávka, která poté dopadá na příslušnou fotodiodu v buňce, způsobí takové vybití, jaké je přímo úměrné intenzitě a době osvětlení. Podrobně je princip funkce fotoelementů vysvětlen v [6]. Pole fotoelementů lze přirovnat k dynamické paměti RAM, tzn. že pro vyčtení informace na jednom řádku se musí konkrétní řádek adresovat, a poté je příslušný sloupec se zesilovačem vybrán a přiveden na výstup. Toto má za úkol blok analogového zpracování. Zesílený signál, který nese informaci o velikosti náboje na příslušném fotoelementu je digitalizován analogově číslicovým převodníkem, odkud je n -bitovou sběrnicí připojen na výstup.

Pro časování veškerých vnitřních obvodů je zapotřebí hodinový (obdélníkový) signál, který je zpravidla přiváděn na samostatný pin. Slouží tedy k časování vyčítání jednotlivých fotoelementů, hradlování A/Č převodníku, generování vertikálních a horizontálních synchronizačních impulzů a pro časování I²C komunikace, která probíhá mezi senzorem a obsluhujícím mikrokontrolérem za účelem nastavení vnitřních řídicích registrů senzoru. Pomocí řídicích registrů senzoru lze nakonfigurovat jeho nejrůznější parametry, např. počet pixelů na řádku, počet řádků, délku řádkových a snímkových zatemňovacích impulzů, délka doby elektronické závěrky, módy vyčítání obrazových dat a další.

¹ Převzato z katalogového listu MT9M001[5].

1.4 Určení polohy objektu z obrazových dat

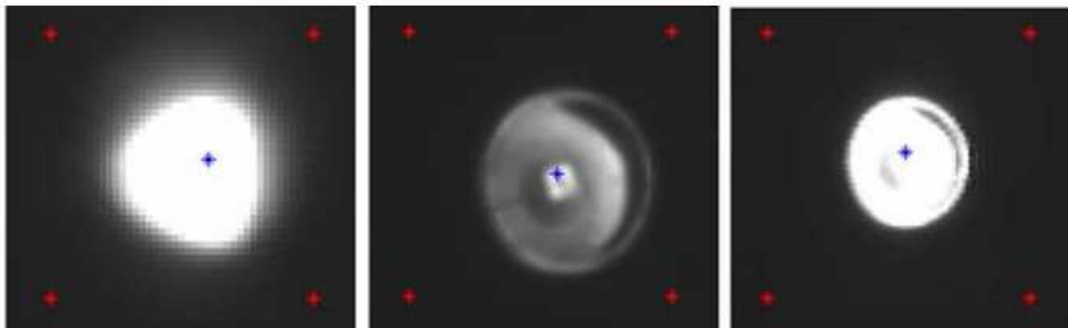
Obraz objektu z CMOS senzoru musí mít takovou strukturu, aby bylo možné strojovým viděním jednoznačně určit, ve kterém místě se zkoumaný objekt nachází. Poloha objektu, která má být měřena, musí být tedy nejdříve definována na nějakém bodě objektu, který bude brán jako vztažný. Může to být bod ve středu, na okraji nebo kdekoliv jinde. Referenční bod může být jeden nebo jich může být více, což umožňuje měřit např. úhel natočení objektu vzhledem k nějakému směru.

Aby byl tento bod dobře odlišitelný od ostatních součástí zkoumaného objektu, musí být jeho jas mnohonásobně vyšší. Clona objektivu před obrazovým senzorem se potom nastaví tak, aby na snímku byla viditelná pouze zářivá značka, zatímco její okolí zůstalo tmavé, viz obr. 1.3.

V praxi se používají dva způsoby označení referenčních bodů měřeného objektu:

- aktivními značkami – *Active Marker System* (světelné “body“, např. LED),
- pasivními značkami – *Passive Marker System* (reflexní odrazné plošky, na které je svíceno ze samostatného světelného zdroje).

Kromě značek (markerů) se může poloha určovat také podle projekce intenzivní světelné stopy (soustředěného svazku paprsků soustavou čoček, laser apod.).



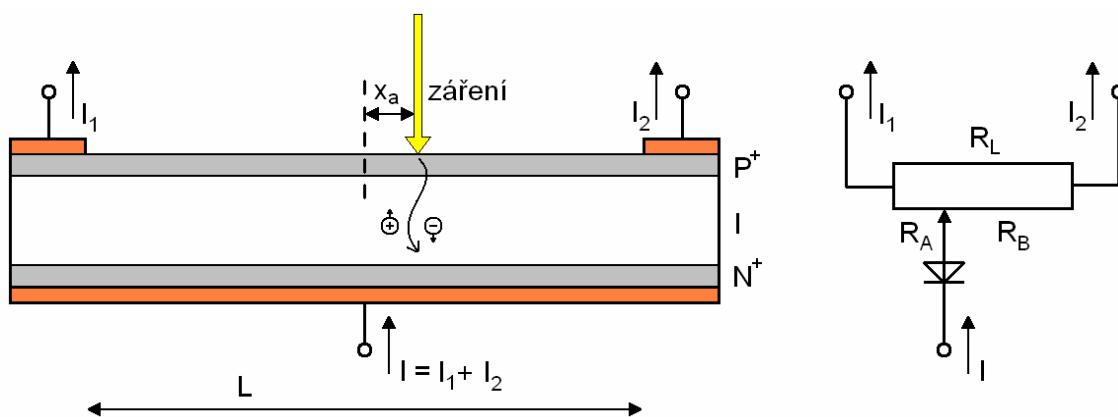
Obr. 1.3 Příklady světelných stop LED a vypočítaná těžiště při různě nastaveném objektivu

Příklady toho, jak „vidí“ obrazový senzor po seřízení clony a zaostření objektivu aktivní značku LED, jsou na obr. 1.3. Tvar světelné stopy není zpravidla kruhový a intenzita jasové plochy nebývá homogenní. Další deformaci tvaru stopy způsobuje pohyb objektu během získávání obrazu – zachycení snímku. Aby bylo možné určit jednoznačně střed takovéto světelné stopy, musí se hledat její těžiště, které je na obrázku v jednotlivých případech vyznačeno modrými body. Více o výpočtu těžiště světelné stopy procesorem a o implementaci algoritmu v programovacím jazyce C pojednává kapitola 3.2.

1.5 Princip funkce PSD sensorů jejich využití k měření polohy

Kromě obrazových sensorů CMOS lze zjišťovat polohu světelné stopy také pomocí tzv. PSD sensorů. Zkratka PSD vznikla z anglických slov *Position Sensitive photo – Detectors*, tedy polohově citlivé detektory. Strukturální vyobrazení a náhradní schéma jednorozměrného senzoru jsou na obr. 1.4. Pro dvojrozměrný případ je princip totožný, ovšem řádkové provedení je pro představu lépe názorné. Konstrukčně se PSD senzory blíží PIN diodě, tzn. obsahují horní silně dotovanou tenkou vrstvu P⁺, spodní vrstvu N⁺ a prostřední intrinsickou vrstvu I. Na horní vrstvě jsou připevněny po stranách dva kontakty, na spodní je jeden celoplošný.

Měrný odpor horní vrstvy je homogenně rozložen po celé délce vrstvy, což si lze představit jako odporovou dráhu (R_L) potenciometru v náhradním schématu. Tento imaginární potenciometr je zapojen jako dělič proudu *I* na proudy *I*₁ a *I*₂ v obráceném poměru drah R_A a R_B. Odbočka potenciometru je nastavována úzkým svazkem světelného paprsku, který dopadá do určitého místa horní odporové vrstvy P⁺. Dopadajícím zářením dochází ve vrstvě I ke generování páru elektron – díra a podle orientace polarizace přechodů PI a NI dojde k posuvu elektronů k vrstvě N a děr k vrstvě P. Průchodem proudu horní vrstvou P v tomto místě dojde k rozdělení odporu dráhy R_L na R_A a R_B.



Obr. 1.4 Princip řádkového PSD senzoru a jeho náhradní schéma²

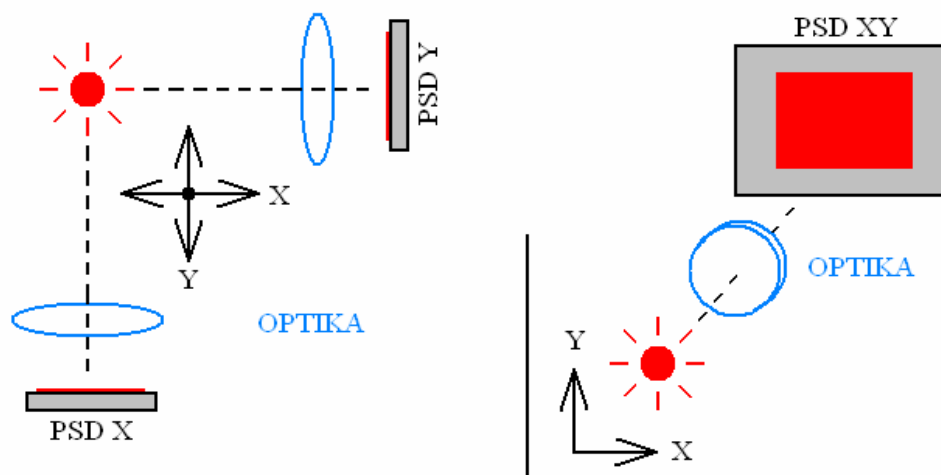
Určení polohy dopadu světelné stopy se provádí poměrově ze znalosti velikosti proudů *I*₁ a *I*₂. Lze tedy odvodit, že vzdálenost *x_a* od středu odporové dráhy, jejíž celková dráha je *L*, se vypočítá ze vztahu

$$\frac{I_2 - I_1}{I_2 + I_1} = \frac{2x_a}{L}. \quad (1.1)$$

² Převzato z [6].

Dopadá-li světelný paprsek do více míst na vrstvě P, pak velikosti proudů I_1 a I_2 budou dány superpozicí fotoproudů. Jestliže dojde k homogennímu osvětlení vrstvy po celé délce, tak I_1 a I_2 budou stejné, viz literatura [2] a [6].

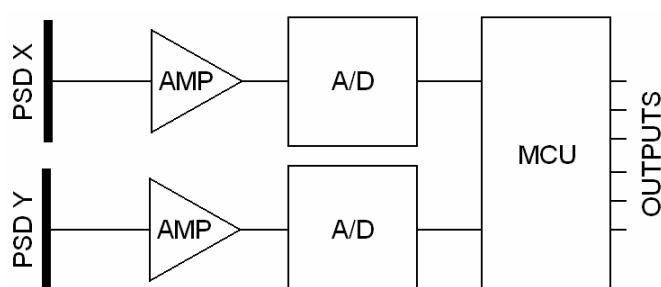
Způsob, jak umístit dva řádkové nebo jeden plošný PSD senzor pro měření polohy světelné stopy v rovině je znázorněn na obr. 1.5.



Obr. 1.5 Určení polohy světelné stopy pomocí dvou řádkových nebo jednoho plošného PSD senzoru

V případě použití dvou řádkových PSD senzorů musí být osy obou prvků navzájem kolmé a před každým senzorem musí být navíc umístěna vhodná optická soustava, která světelný paprsek soustředí do měřicí dráhy senzoru. Výhodou této konfigurace je nižší cena řádkových PSD senzorů, nevýhodou pak jejich složité mechanické umístění v okolí pohybujícího se objektu. Tento nedostatek lze však obejít pomocí jednoho plošného senzoru s jednou optickou soustavou. Osa senzoru je pak kolmá na rovinu pohybu objektu, podobně jako u obrazového CMOS senzoru.

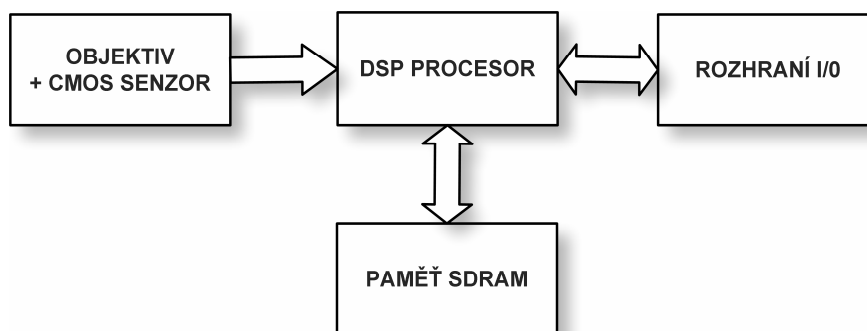
Jelikož mají signály z PSD senzorů analogový charakter, je nutné u obou typů (1D i 2D) provést digitalizaci, která je v době současné elektrotechniky nezbytná, viz obr. 1.6. Obvod pro úpravu signálů z PSD senzorů musí navíc obsahovat analogové zesilovače a typicky i vlastní mikrořadiče pro číslicové zpracování, což celý měřicí řetězec prodražuje, viz dále.



Obr. 1.6 Typické zpracování analogových signálů z PSD senzorů

1.6 Určení polohy objektu z obrazových dat signálovým procesorem

Tato kapitola se zabývá dvěma přístupy ke zpracování informace z obrazového CMOS senzoru. První způsob spočívá ve zpětném procesorovém zpracování dat z externí paměti (*Image Post Processing*), zatímco druhá metoda je založena na průběžném zpracování obrazových dat v omezené vnitřní paměti signálového procesoru (*On The Fly Image Processing*).



Obr. 1.7 Blokové schéma snímače polohy se signálovým procesorem a externí SDRAM

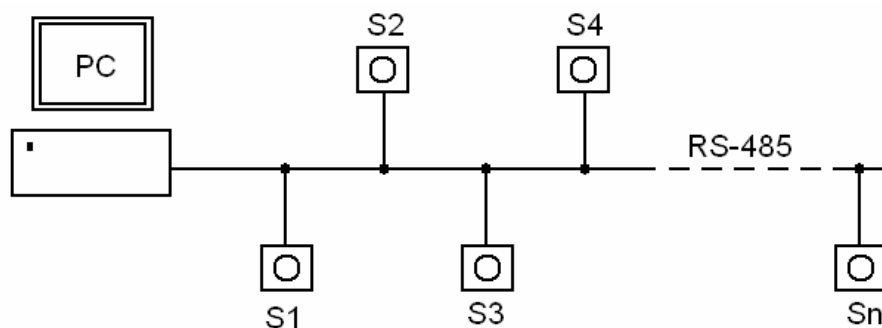
První, dříve ověřený přístup, znázorňuje blokové schéma na obr. 1.7. Jedná se o tzv. inteligentní kameru (*Smart Camera*), kdy je obraz ze senzoru ukládán přes procesor DMA kanálem do externí paměti typu SDRAM, odkud je následně zpracováván. Inteligentní kamera umožňuje vykonat nad uloženými daty nejrůznější metody zpracování obrazu, jako jsou např. hledání těžiště světelné stopy, prahování obrazu, hranovou detekci, vyhledání vzoru v obraze a další, viz [23]. Pokud je navíc kamera vybavena rychlým USB řadičem, pak lze snadno přenášet obrazové videosekvence do PC. Nevýhodou této koncepce s pamětí SDRAM je vyšší spotřeba elektrické energie a nutnost rozměrnějšího plošného spoje.

Proto je snaha tyto nedostatky odstranit, tzn. redukovat obvodové zapojení na minimální sestavu CMOS senzoru a signálového procesoru tak, aby bylo možné měřit polohu světelné stopy průběžně z dat, která budou v čase přicházet z obrazového senzoru a bude je možné ukládat pouze do omezené vnitřní paměti SRAM procesoru. Kromě snížené spotřeby je dalším požadavkem na snímač polohy tohoto typu také měření v reálném čase pro účely regulace a autonomní funkčnost bez nutnosti připojení PC.

1.7 Cíl práce a představa o výsledné podobě snímače polohy

Realizace snímače polohy inteligentní kamerou není výhodná z důvodů uvedených v předchozí kapitole (spotřeba, rozměry, cena). Ani použití PSD senzorů není optimální, a to zejména proto, že neumožňují měřit polohy více světelných stop. Dále také kvůli nutnosti doplnění o obvody pro zpracování a digitalizaci analogových signálů, což zvyšuje jejich cenu, která je pak srovnatelná s kombinací (dnes již relativně levného) CMOS senzoru a signálového procesoru. Hlavním cílem práce je tedy **navrhnout a realizovat rychlý snímač polohy tohoto typu v minimální možné obvodové konfiguraci na desce plošného spoje o rozměrech sensorové desky, s konektorem pro vzájemné propojení**, viz závěrečná podoba ve fotodokumentaci na obr. 7.3. Vedlejší požadavky na zařízení, které nemají takovou prioritu, jsou např. přenos obrazových dat do PC přes USB řadič, vyvedení některých rozhraní procesoru na konektory a možnost implementace některých dalších metod zpracování obrazu, které by byly zvládnutelné pouze s vnitřní SRAM pamětí.

Jak již bylo uvedeno, navrhovaný snímač polohy musí být schopný samostatné činnosti v regulační smyčce bez nutného připojení PC (kromě nastavování v servisním módu) a také by mělo být možné jeho připojení do sensorové sítě např. se sběrnici RS-485, viz obr. 1.8.



Obr. 1.8 Sensorová síť ze snímačů polohy S1 - Sn

Síť senzorů je vhodná např. tehdy, když je požadováno vícenásobné měření polohy a každý snímač se může chovat jako kamera s vnitřním monitoringem, tzn. sama hlídá polohu světelné stopy do té doby, než dojde k nějaké události, a poté začne přenášet obraz. Spouštěcí událostí může být např. vychýlení polohy objektu z dovolených mezí, skoková změna polohy, nestabilní poloha (chvění objektu) apod. Přenášený obraz nemusí být v plném rozlišení, podstatný je náhled (monitoring) měřené scény, což lze již při rychlosti 10 Mb/s na sběrnici RS-485 uskutečnit. Více je o možnostech přenosu obrazu komunikačními kanály s malou kapacitou pojednááno v kapitole 3.1.2. Tímto způsobem lze tedy realizovat distribuované zpracování obrazu, kdy každý snímač provádí samostatně předzpracování (*preprocessing*) své měřené scény a obraz je

přenášen až po nějaké události. Kromě toho může být snímač také periodicky požádán o obraz z řídicího PC, kam jsou následně snímky ukládány.

Hlavním důvodem upřednostnění signálového procesoru před hradlovým polem je jednoduché nahrání obslužného programu z bootovací EEPROM paměti a snadná rekonfigurovatelnost obrazového CMOS senzoru, zatímco obvod FPGA by musel být ještě doplněn o externí paměť a pomocný mikrokontrolér, což by vyžadovalo větší obvodovou složitost. Dalším důvodem výběru DSP procesorů řady BF-53x je přímá podpora připojení obrazového senzoru na rozhraní PPI a snadný přenos obrazových dat DMA kanálem do vnitřní paměti.

Aby bylo možné snadné a rychlé nahrání obslužného programu z PC do bootovací paměti procesoru, měl by snímač polohy disponovat USB řadičem, který by (kromě dalších funkcí) tuto vlastnost programátoru zastával. V EEPROM by pak mohl být v určité oblasti nahrán zavaděč (*loader*), který by se po restartu procesoru nahrál do jeho vnitřní paměti. Dále by procesor nahrával z příslušných sektorů EEPROM pouze ty algoritmy, jež by byly právě potřebné pro vykonání zvolené metody zpracování, čímž by se také dále snížilo zaplnění malé paměti SRAM, viz kapitola 2.3.3.3.

2 Postup vývoje vlastního hardware

2.1 Shrnutí požadavků na snímač polohy z hardwarového hlediska

Celkové pojetí zadání této práce ilustruje obr. 2.1. Finální zařízení musí být schopné měřit polohu jedné a více světelných stop na zkoumaném objektu a musí umožňovat autonomní měření, kdy informaci o poloze předává přes standardní rozhraní (např. RS-232, RS-485, SPI, I²C apod.). Dále musí být snímač polohy připojitelný k PC přes USB pro získání náhledového obrazu a pro konfiguraci nejrůznějších parametrů. Důležitým požadavkem je připojitelnost sensorových desek s CMOS senzory a objektivy, které na katedře měření vznikly v předchozích letech a mají definované rozměry plošného spoje a rozložení signálů na propojovacím konektoru. Rozměry desky snímače polohy se musí shodovat s rozměry sensorové desky, tedy 60 × 60 mm, a po sesazení spolu budou tvořit kompaktní celek.

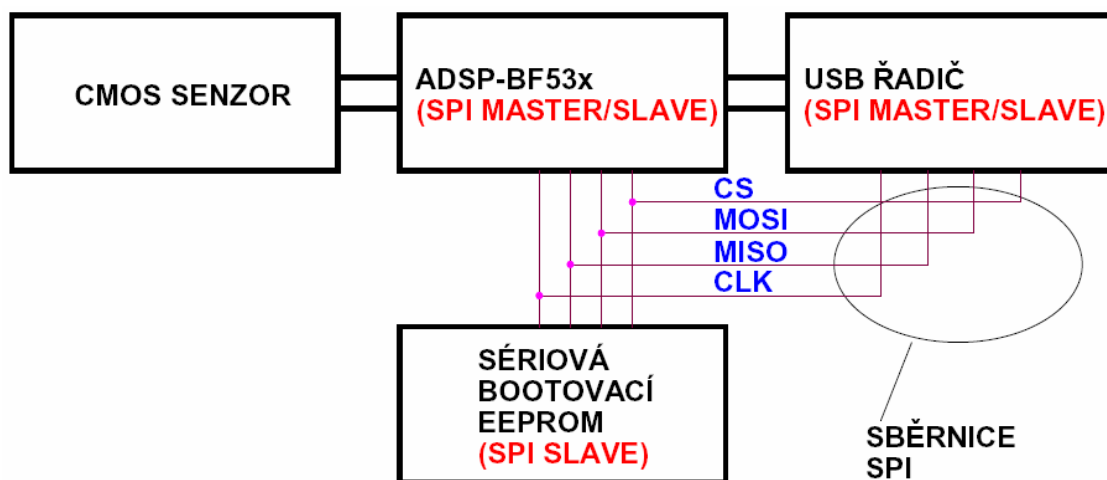


Obr. 2.1 Blokové schéma měření polohy obrazovým senzorem

Vlastní snímač polohy bude sestávat v minimální konfiguraci ze signálového procesoru ADSP BF-53x, bootovací paměti EEPROM, USB řadiče, měniče napájecího napětí pro potřebné úrovně, konektorů signálových rozhraní (dle možností velikosti DPS) a doplňkových obvodů (reset, tvarování hodinových pulzů, ochranných odporů, indikačních LED). Podrobný postup vývoje snímače polohy, vzniklé problémy a odůvodnění konkrétního výsledného zapojení budou popsány v následujících kapitolách.

2.2 Volba USB řadiče

Prvním úkolem práce bylo vyřešit způsob programování bootovací paměti procesoru, do kterého se z této paměti po resetu nahraje řídicí program. Dřívější způsoby programování buď vyžadovaly vyjmutí paměti z patice a naprogramování v externím programátoru, nebo jeho připojení k SPI rozhraní na plošném spoji, viz např. [7]. Jelikož tato procedura není při ladění programu příliš časově efektivní, hledalo se jiné řešení. Procesory ADSP BF-53x sice disponují rozhraním JTAG, se kterým by bylo ladění během vývoje software pohodlné, ovšem dostupnost emulátoru není vždy z důvodu vyšší ceny samozřejmostí.



Obr. 2.2 Využití USB řadiče k programování EEPROM

Na obr. 2.2 je blokové schéma snímače polohy se znázorněním SPI sběrnice, na níž jsou připojeny ADSP BF-53x, EEPROM a USB řadič. V blokovém schématu jsou dále vyznačeny uvažované režimy připojení jednotek ke sběrnici, tj. MASTER, SLAVE. Procesor je nakonfigurován tak, že po připojení napájecího napětí (nebo po resetu) začne bootovat z EEPROM, čili v ten okamžik se vždy připojí ke sběrnici. Aby nedošlo ke kolizi mezi procesorem a USB řadičem, musí být USB řadič po celou dobu od sběrnice odpojen, tzn. musí mít budiče ve třetím stavu. Přistupovat ke sběrnici smí pouze tehdy, když je procesor „držen“ ve stavu reset.

Pro možnost přímého programování EEPROM vyplývají z výše uvedeného tyto požadavky na USB řadič:

- SPI rozhraní (HW/SW), přes které půjde nahrát z PC program, tj. signály (CS, MOSI, MISO, CLK),
- trvalé odpojení od sběrnice s výjimkou vlastního programování EEPROM,
- možnost přímého resetu procesoru (optimálně).

Další úkol, který měl USB řadič splňovat, byl přenos obrázku ze senzoru do PC, aby bylo zjištěné, co snímač „vidí“, když je nainstalován nad zkoumaným objektem, tzn. aby uživatel na monitoru viděl, jak je třeba zaostřit objektiv a nastavit clonu, pokud to připojená sensorová deska umožňuje. V ideálním případě, při dostatečném datovém toku přes USB řadič, by mohl být snímač polohy využíván jako kamera. Na tomto místě je nutné připomenout, že rychlý přenos obrazu do PC nebyl primárním úkolem, ale pouze rozšiřující vlastností snímače polohy. Tím hlavním požadavkem bylo rychlé měření polohy a malé rozměry plošného spoje. O možnostech přenosu náhledu měřené scény přes komunikační kanál s malou kapacitou a bez uložení do externí paměti SDRAM se zmiňuje kapitola 3.1.2.

Po vytyčení hlavních požadavků následoval výběr konkrétního obvodu, přičemž v době rozhodování se byly na katedře dostupné tyto obvody:

- *FTDI Chip – FT245R: USB FIFO,*
- *FTDI Chip – FT232R: USB UART,*
- *FTDI Chip – FT2232H: Hi-Speed USB 2.0 - Dual UART/FIFO Converter,*
- *Cypress Semiconductor – CY7C680: EZ-USB FX2LP™ USB Microcontroller, High Speed USB Peripheral Controller.*

Shrnutí nejdůležitějších parametrů, které byly důležité pro optimální výběr, je uvedeno v tab. 2.1. Bližší popis možného zapojení konkrétního USB řadiče do obvodu bude vysvětlen v dalším textu.

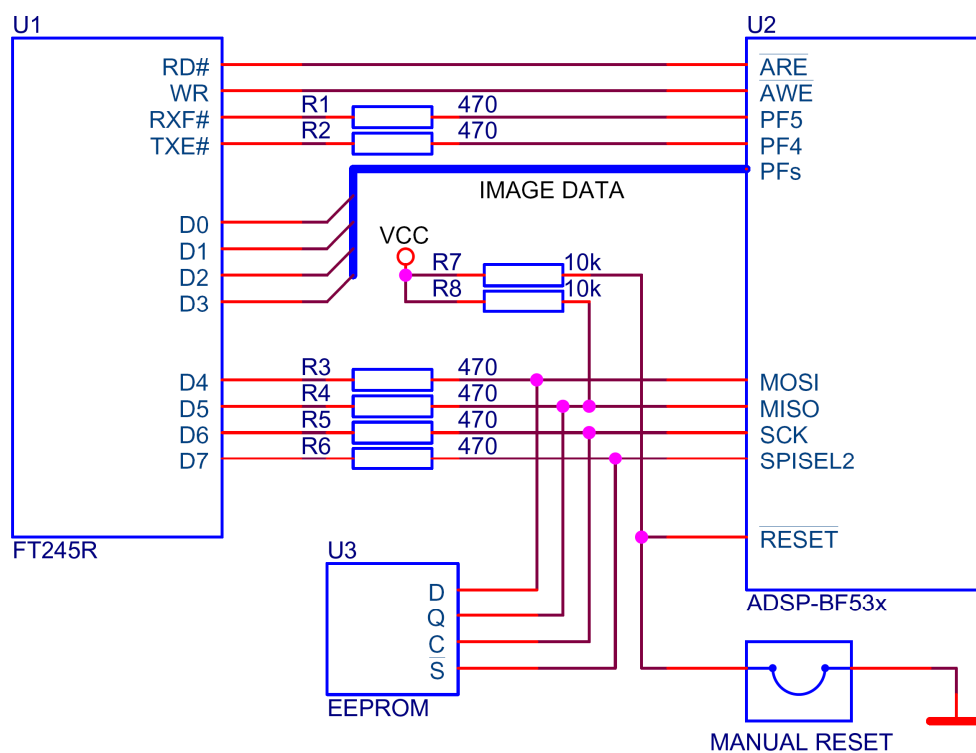
	FT245R	FT232R	FT2232H	CY7C6801
realizace SPI	Bit Bang: D4, 5, 6, 7	Bit Bang: CBUS0, 1, 2, 3	ADBUS0–3: SK, DO, DI, CS	PORT D: PD0, 1, 2, 3
přenos obrazu	Parallel FIFO: D0–3, RD#, WR, RXF#, TXE#	UART: RXD, TXD, handshaking	BDBUS0–7: D0–7, handshaking	Slave FIFO: FD[0–7], handshaking
max. přenosová rychlost	500 kB/s	300 kB/s	8 MB/s	46 MB/s ³
kompatibilita USB	Full Speed	Full Speed	High Speed	High Speed
pouzdro	SSOP – 28	SSOP – 28	LQFP – 64	SSOP – 56

Tab. 2.1 Přehled nejpodstatnějších parametrů dostupných USB řadičů

³ Teoretická hodnota, které v praxi nelze dosáhnout. Více se o dosažených rychlostech tohoto řadiče zmiňuje např. [12], kapitola 4.5.

2.2.1 FT245R s programováním EEPROM v režimu Bit Bang

Obvod FT245R je obousměrný převodník USB na paralelní FIFO. Pro nastavení nejrůznějších parametrů využívá interní integrovanou 128 B EEPROM, což je spolu s typem pouzdra SSOP – 28 výhodné z hlediska úspory místa na DPS. Uvnitř obvodu je také integrován 12 MHz oscilátor, takže odpadá potřeba připojení externího krystalu s keramickými kondenzátory. Velikost paměti Rx Bufferu (USB → FIFO) je 128 B, velikost Tx Bufferu (FIFO → USB) je 256 B. Přenosová rychlost může dosahovat až 1 MB/s, ovšem v případě zapojení dle obr. 2.3 je možná rychlost pouze poloviční, neboť pro přenos obrazu se využívají jen signálové linky D0 - D3. Mezi další vlastnosti patří možnost přepnutí obvodu do Bit Bang režimu, který popisuje následující odstavec. Více informací o tomto obvodu se nachází v [9].



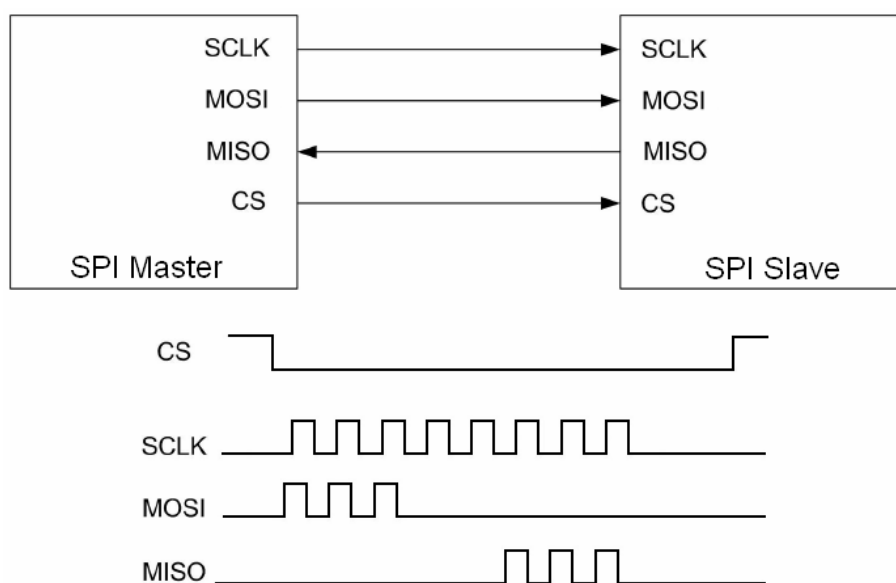
Obr. 2.3 Propojení FT245R, paměti EEPROM a BF-53x

Výrobce integrovaných obvodů FTDI Chip implementuje v současné době do všech svých USB řadičů tzv. Bit Bang mód⁴. Přepnutí obvodu do tohoto módu se provádí buď programově, nebo pomocí manuální konfigurace v interní/externí EEPROM paměti USB řadiče. Režim Bit Bang umožňuje pomocí PC přímý zápis (a čtení) na některé signálové vývody tak, že se chovají jako GPIO (*General Purpose Input/Output*) u nejrůznějších mikroprocesorů a mikrokontrolérů. Jinými slovy to znamená, že např. datové linky D0 - D7 řadiče FT245R mohou být programově přepnuty do Bit Bang módu a okamžitě se z nich stane 8-bitový paralelní IO port řízený z PC.

⁴ Informace o nabízených produktech jsou dostupné online (11/2010): <http://www.ftdichip.com/Products/ICs.htm>

Toho lze využít k nejrůznějším účelům, např. k ovládání svítivých diod, relé, spínacích obvodů, displejů, čtení z maticových klávesnic apod. Použití Bit Bang módu k programování bootovací paměti EEPROM je zjednodušeně znázorněno ve schématu na obr. 2.3.

Ke sběrnici SPI, která je mezi procesorem a jeho bootovací pamětí, byla připojena další jednotka – řadič FT245R, který generuje na signálových vodičích D4 - D7 takové průběhy, pomocí nichž je možné naprogramovat nebo vyčíst EEPROM, viz kapitola 2.2.1.2.



Obr. 2.4 Propojení jednotek na SPI sběrnici a časové průběhy signálů⁵

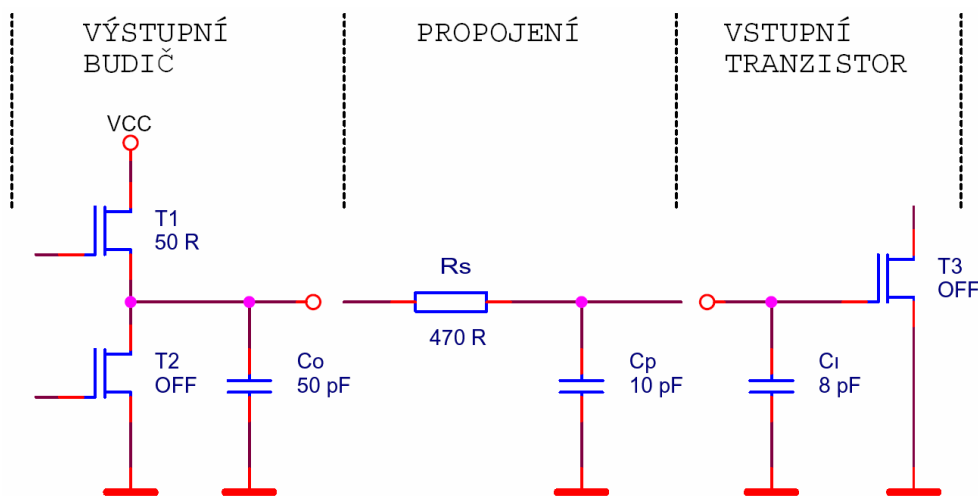
Obecný princip propojení dvou jednotek na sběrnici SPI připomíná obr. 2.4. V okamžiku zápisu na sběrnici musí být pouze jedna jednotka typu MASTER, viz dále. Signály na sběrnici mají následující význam:

- Chip Select (CS) – výběrový signál od MASTER,
- Serial Clock (SCLK) – hodinový signál od MASTER,
- Master Out Slave In (MOSI) – výstup sériových dat od MASTER,
- Master In Slave Out (MISO) – vstup sériových dat od SLAVE.

V sérii s datovými linkami jsou ochranné odpory, které v případě kolize na sběrnici zabrání vzájemnému zničení všech obvodů. Teoreticky by k tomu mohlo dojít tehdy, pokud by procesor nebyl ve stavu reset a USB řadič by se pokoušel o naprogramování paměti. Reset procesoru před začátkem programování se v tomto zapojení musí vyvolat manuálně propojkou nebo tlačítkem.

⁵ Převzato a upraveno z aplikační poznámky AN_114 pro FT2232H [13].

Pozn.: Pro reset procesoru by sice bylo možné vyhradit další linku paralelní FIFO sběrnice, ale kvůli dalšímu snížení počtu signálových vodičů pro přenos obrazu toto není výhodné.



Obr. 2.5 Sériový odpor s parazitními kapacitami na signálové cestě mezi USB řadičem a procesorem

Vložením ochranných sériových odporů dojde ke zpomalení dynamiky a k zaoblení náběžných i spádových hran obdélníkového průběhu. Po překreslení obvodu do schématu na obr. 2.5 je zřejmé, že výsledná časová konstanta bude složena z odporu sepnutého tranzistoru výstupního budiče R_{T1ON} , ochranného odporu R_S , kapacity výstupní C_O , kapacity spoje (propojení) C_P a kapacity vstupní C_I , tedy

$$\tau = (R_{T1ON} + R_S) \cdot (C_O + C_P + C_I). \quad (2.1)$$

Po dosazení hodnot ze schématu vychází, že $\tau = 35,4$ ns. Při maximální přenosové rychlosti 1 MB/s může docházet ke změnám signálu cca po 1 μ s, takže ochranný odpor nezpůsobí takovou degradaci obdélníkového průběhu, že by nebylo možné data správně přenést.

2.2.1.1 Ovládání asynchronního Bit Bang módu USB řadiče FT245R na straně PC pod operačním systémem Windows

Výrobce FTDI poskytuje volně ke stažení ovladače pro různé operační systémy (Windows, Linux, MAC) a podporu pro populární programovací jazyky jako např. C++, C#, Delphi, LabView, Visual Basic a další. Distribuované ovladače lze rozdělit na následující typy [15]:

- *D2XX Direct Driver* – pro Linux, MAC OS X (ver. 10.4 a vyšší), Windows CE (ver. 4.2 a vyšší). Tento ovladač může být nainstalovaný v operačním systému pouze samostatně

v jednom okamžiku. Využívá přímý přístup k připojenému USB zařízení pomocí funkcí z knihovny FTD2XX.DLL⁶,

- *VCP Virtual COM Port* – emulace klasického sériového portu, kdy se k FTDI řadiči přistupuje tak, jako by to byl další COMx port v PC,
- *CDM Combined Driver Model* – pouze pro Windows, je možné nainstalovat oba dva typy driverů (*D2XX*, *VCP*) najednou, ovšem v jednom okamžiku lze využívat pouze jeden z nich.

V této práci je veškeré programové ovládání řadičů FTDI soustředěno na OS Windows, jazyk C/C++ a ovladače *D2XX*. Každá aplikace, ve které je využíván *D2XX* driver musí mít ve svém adresáři soubory *FTD2XX.LIB* a *FTD2XX.H*, přičemž soubor *FTD2XX.LIB* musí být v daném projektu přidán k seznamu knihoven pro linker a *FTD2XX.H* vložen do zdrojového kódu pomocí příkazu **#include** "ftd2xx.h". Následující přehledový souhrn se věnuje *D2XX* funkcím, které se přímo využívají při programování v asynchronním Bit Bang módu.

- **FT_STATUS FT_Open** (*int iDevice*, **FT_HANDLE** **ftHandle*), kde
 - *iDevice* je index zařízení, které má být otevřeno (0, pokud je připojeno jediné),
 - *ftHandle* je ukazatel na proměnnou typu **FT_HANDLE**. Handler je nezbytný pro další manipulaci s otevřeným zařízením.

Otevírá zařízení s odpovídajícím Device ID. Návrátová hodnota **FT_STATUS** indikuje úspěšnost otevření jednotky (**FT_OK**), popř. nějaký chybový kód [15]. U dalších funkcí má tato hodnota totožný význam, proto nebude v následujícím textu opakovaně uváděna.

- **FT_STATUS FT_SetBitMode** (**FT_HANDLE** *ftHandle*, **UCHAR** *ucMask*, **UCHAR** *ucMode*), kde
 - *ftHandle* je handler zařízení,
 - *ucMask* je 8-bitová hodnota, nastavující masku Bit módu, tzn. 0 = vstup, 1 = výstup na odpovídajícím pinu,
 - *ucMode* je hodnota módu. Pro přepnutí FT245R do asynchronního módu se nastaví hodnota 0x01, hodnota 0x00 způsobí reset módu. Kompletní přehled všech módů je v [15].

Funkce přepne USB řadič do módu, ve kterém má být provozován.

⁶ Ovladače jsou dostupné k volnému stažení na <http://www.ftdichip.com/Drivers/D2XX.htm>.

- **FT_STATUS FT_GetBitmode** (**FT_HANDLE** *ftHandle*, **PUCHAR** *pucMode*), kde
 - *ftHandle* je handler zařízení,
 - *pucMode* je ukazatel na unsigned char, ve kterém je uložena okamžitá hodnota pinů sběrnice.

Na hodnotu uloženou na příslušnou adresu v paměti nemá vliv, zda se jedná o datové linky vstupní nebo výstupní, proto je na uživateli, aby získanou informaci správně vyhodnotil.

- **FT_STATUS FT_SetBaudRate** (**FT_HANDLE** *ftHandle*, **DWORD** *dwBaudRate*), kde
 - *ftHandle* je handler zařízení,
 - *dwBaudRate* je hodnota Baud Rate.

Rychlost přenosu dat je určena touto funkcí, přičemž časování pro asynchronní Bit Bang mód odpovídá ve skutečnosti $16 \times \text{Baud Rate}$. To znamená, že např. pro hodnotu 9600 Baud budou data přenášena rychlostí $(16 \times 9600) = 153600$ B/s, což činí 6,5 μs na každý bajt.

- **FT_STATUS FT_Write** (**FT_HANDLE** *ftHandle*, **LPVOID** *lpBuffer*, **DWORD** *dwBytesToWrite*, **LPDWORD** *lpdwBytesWritten*), kde
 - *ftHandle* je handler zařízení,
 - *lpBuffer* je ukazatel na pole bajtů s daty, která se mají zapsat na sběrnici,
 - *dwBytesToWrite* je počet zapisovaných bajtů,
 - *lpdwBytesWritten* je ukazatel na proměnnou typu DWORD, kam se uloží počet bajtů, které se na sběrnici zapsaly.

Funkce zapíše odpovídající slovo nebo posloupnost slov na výstupní piny sběrnice.

- **FT_STATUS FT_Read** (**FT_HANDLE** *ftHandle*, **LPVOID** *lpBuffer*, **DWORD** *dwBytesToRead*, **LPDWORD** *lpdwBytesReturned*), kde
 - *ftHandle* je handler zařízení,
 - *lpBuffer* je ukazatel na Buffer, kam se mají přijatá data ze zařízení uložit,
 - *dwBytesToRead* je počet bajtů, kolik se má ze zařízení vyčíst,
 - *lpdwBytesReturned* je ukazatel na proměnnou typu DWORD, kam se uloží počet bajtů, které byly ze sběrnice vyčteny.

Funkce přečte stavy pinů (vstupních i výstupních) a uloží je na určené paměťové místo. V případě požadavku na vícenásobné čtení jsou data na sběrnici vzorkována stejně jako při zápisu podle nastavení funkcí *FT_SetBaudRate*.

- **FT_STATUS FT_Close** (**FT_HANDLE** *ftHandle*), kde
 - *ftHandle* je handler zařízení.

Funkce se používá ke korektnímu uzavření otevřeného zařízení *ftHandle*.

2.2.1.2 Obslužný program řadiče FT245R pro generování programovacích průběhů na SPI sběrnici

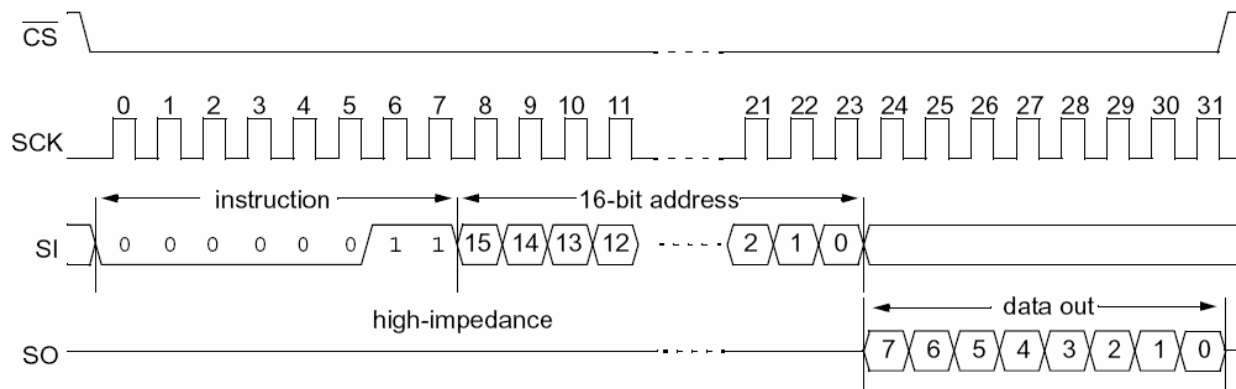
USB řadič FT245R umožňuje přepnutí do *asynchronního* a *synchronního* Bit Bang módu [14]. Na počátku vývoje programátoru EEPROM byl testován pouze *asynchronní* mód, vysvětlení bude uvedeno později. V asynchronním módu lze datové linky D0 - D7 libovolně konfigurovat jako vstupy nebo výstupy. Rychlost zápisu/čtení vychází z nastavení Baud Rate generátoru. Po zápisu dat se na výstupních linkách objeví zapisované slovo, které je na nich ovšem drženo až do chvíle, kdy je přepsáno jinými daty. Přehled nezbytných funkcí pro softwarové ovládání Bit Bang módu a možnosti instalovatelných ovladačů pro PC byly uvedeny v kapitole 2.2.1.1.

Prvním typem EEPROM paměti, která měla být v režimu Bit Bang naprogramována, byla paměť o velikosti 256 kbit - 25LC256 od výrobce *Microchip*. Jedná se o paměť s organizací 32768×8 bit, velikostí stránky 64 B a maximální frekvencí 10 MHz pro hodinový signál. Podrobný popis lze najít v katalogovém listu [16]. Programátor měl zvládat tyto základní operace:

- vyčtení celé paměti a uložení dat do souboru,
- vyčtení n bajtů z paměti od zvolené adresy a jejich uložení do souboru,
- zápis libovolného souboru dat do paměti od zvolené adresy.

Podarilo se realizovat všechny výše uvedené funkce, ovšem výsledná rychlost, zejména při čtení, nebyla dostatečná. Zdůvodnění této skutečnosti je obsaženo v následujících odstavcích. Podstata celého programátoru je založena na rozkladu průběhů jednotlivých signálů do posloupnosti stavů, které budou postupně zapisovány (čteny) na sběrnici pomocí elementárních funkcí **FT_Write**, **FT_Read** a **FT_GetBitMode**. Celá aplikace je naprogramována genericky ve smyslu možnosti přiřazení signálů \overline{CS} , SCK, SI, SO na libovolné datové linky D0 - D7, což může být později užitečné např. při návrhu plošného spoje. Obsluha programátoru je založena na konzolovém přístupu, tzn., že veškeré funkcionality jsou spouštěny z příkazové řádky, např.

- **ftdiprog.exe** *-w testprg.ldr* zapíše celý soubor *testprg.ldr* do EEPROM od adresy 0x0000
- **ftdiprog.exe** *-r unknown.txt* vyčte obsah celé EEPROM a uloží jej do souboru *unknown.txt*
- **ftdiprog.exe** *-r origprg.ldr 14278* vyčte 14278 B EEPROM od adresy 0x0000 a uloží jej do souboru *origprg.ldr*



Obr. 2.6 Signály na SPI sběrnici pro vyčtení bloku dat z paměti EEPROM 25LC256⁷

Aby bylo možné vyčíst obsah 25LC256, je zapotřebí vygenerovat na SPI sběrnici patřičné průběhy, viz obr. 2.6, což vyžaduje kombinovaný zápis a čtení. Signál \overline{CS} (*Chip Select*) slouží k výběru obvodu paměti a je aktivní v log. 0 po celou dobu vyčítání dat. Hodinový signál SCK (*Serial Clock Input*) slouží k hradlování událostí na vodičích SI a SO, přesněji řečeno s každou náběžnou hranou hodinového signálu dojde ke vzorkování hodnoty vstupního pinu SI a k posunutí sériových dat na výstupním pinu SO. Na vstup SI (*Serial Data Input*) musí přijít nejprve instrukce pro čtení, tj. READ - 0000 0011, poté 16-bitová adresa (od nejvíce po nejméně významný bit), od jakého místa v paměti má být vyčítání obsahu započato. Signál SO (*Serial Data Output*) poskytuje sled vyčítaných dat, která se na tomto pinu začnou po odeslání posledního adresového bitu objevovat s každým hodinovým impulzem.

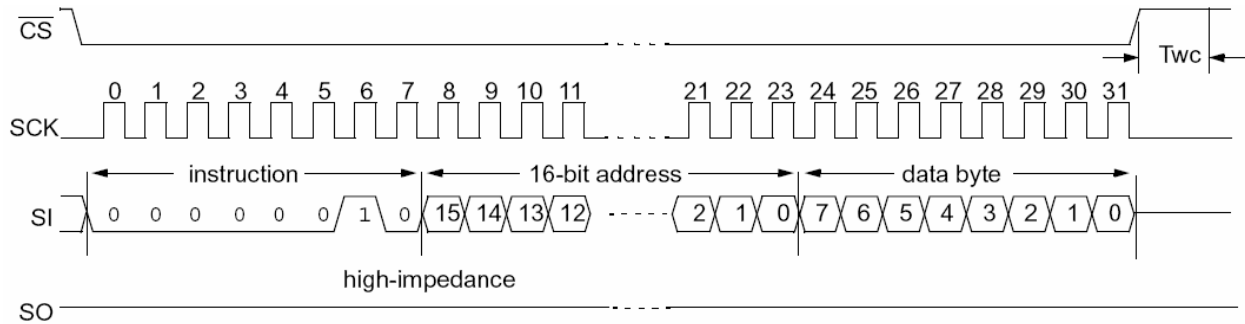
Z hlediska aplikačního programu tedy proces čtení nejprve kopíruje průběhy signálů na SPI sběrnici, tj. posloupnost: spádová hrana \overline{CS} , zápis instrukce READ na SI a zápis 16-bitové adresy na SI. Od tohoto okamžiku by mělo následovat vybavování dat na pin SO, což ovšem znamená nutnost přepínání funkcí **FT_Write** a **FT_Read**, tzn. sled těchto elementárních kroků:

- v 1. kroku se zapíše slovo s log. 0 na bitu SCK,
- ve 2. kroku následuje zápis slova s log. 1 na bitu SCK,
- ve 3. kroku je čteno slovo, z něhož je nemaskován pouze bit SO, který je posunut na 7. pozici zásobníku 1. vyčteného bajtu,
- ve 4. kroku se zapíše slovo s log. 0 na bitu SCK.

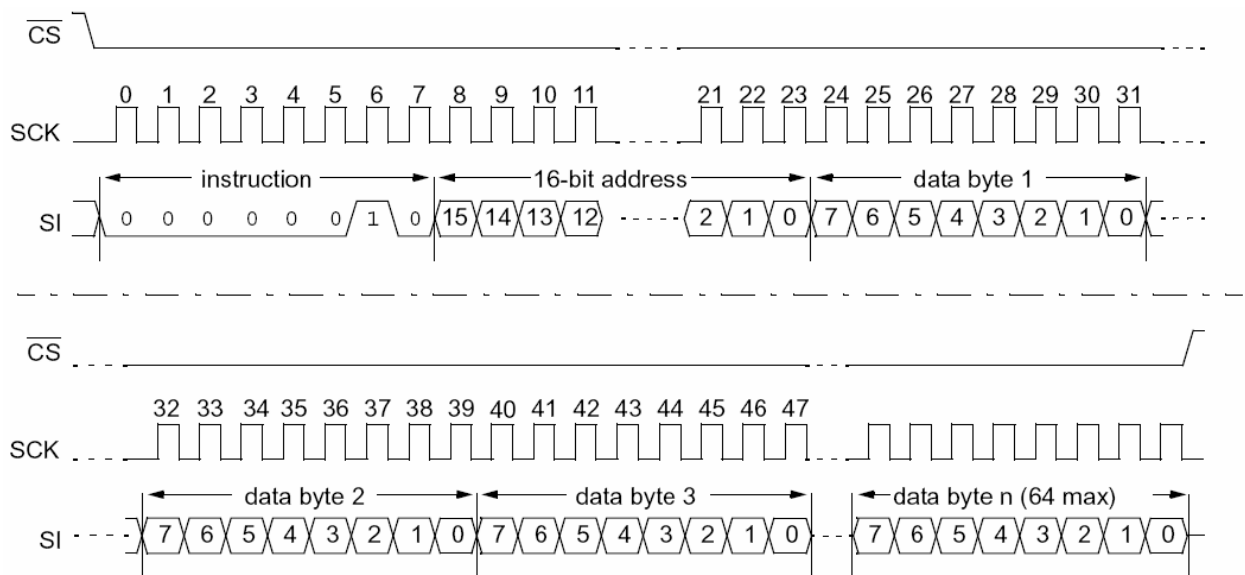
Tato procedura se opakuje pro všech zbývajících bitů v 8-bitovém slově, které se postupně uloží celé do zásobníku. Tímto způsobem se vyčte požadovaný obsah paměti a pole vyčtených slov je uloženo do souboru.

⁷ Převzato z katalogového listu 25AA256/25LC256 [16].

Zápis dat ze souboru do paměti EEPROM lze uskutečnit dvěma způsoby. Prvním je zápis po bajtech, kdy se zapíše kód instrukce WRITE – 0000 0010, 16-bitová adresa a osm bitů jednoho slova, viz obr. 2.7. Druhým způsobem je zápis po 64 B stránkách, tj. kód instrukce WRITE, 16-bitová adresa a 64 B zapisovaných dat, po kterých je nutné pro další zápis celý postup zopakovat s adresou posunutou o velikost stránky, viz obr. 2.8.



Obr. 2.7 Signály na SPI sběrnici pro zápis jednoho bajtu do paměti EEPROM 25LC256⁸



Obr. 2.8 Signály na SPI sběrnici pro zápis celé 64 B stránky do paměti EEPROM 25LC256⁸

V aplikačním programu se proces zápisu do EEPROM liší jen částečně. Stejně jako při čtení musí být nejprve na SPI sběrnici poslána data s kódem instrukce a počáteční adresa. Dále však již nedochází k přepínání mezi funkcemi **FT_Write** a **FT_Read**, neboť se pouze zapisuje datový bit stream, sestavený z posloupnosti bajtů zapisovaného souboru. Při zápisu celé stránky do paměti bylo zjištěno, že generování průběhů po jednotlivých slovech, sestavených z bitů \overline{CS} , SCK, SI a SO, je časově náročné. Jinými slovy, pro zápis 64 B stránky musela být 180 × volána funkce **FT_Write** (*ftHandle, currentByte, 1, lpdwBytesWritten*). Možnost, jak obejít tento

⁸ Převzato z katalogového listu 25AA256/25LC256 [16].

způsob, spočívala v přípravě celých průběhů do pole operační paměti počítače a následný zápis celého bloku najednou, tj. příkazem **FT_Write** (*ftHandle, memoryData, 180, lpdwBytesWritten*). Výsledky obou procesů čtení/zápis paměti EEPROM 25LC256 shrnuje následující kapitola.

2.2.1.3 Zhodnocení a uplatnění programátoru EEPROM 25LC256 v módu Bit Bang u obvodu FT245R

Jak již bylo řečeno, rychlost zapisování a zejména vyčítání se ukázala být manuálním Bit Bangem neúnosně malá. Z důvodu střídavého volání příkazů pro zápis (generování hodinového pulzu na pinu SCK) a čtení (vyčítání posunutého bitu na pinu SO) docházelo k časovým prodlevám, takže k vyčtení celého obsahu paměti, tj. 32768 B, bylo zapotřebí přibližně 1680 s času, tedy 25,6 ms pro jeden stav na sběrnici SPI. Zaplnění celé paměti EEPROM daty ze souboru při zápisu po stránkách a hromadném zápisu bloku dat z operační paměti počítače trvalo při nastavené Baud Rate = 256000 Bd přibližně 35, tedy cca 380 μ s na jeden stav sběrnice.

Výše uvedené parametry naznačují, že použití asynchronního Bit Bang módu k zápisu a hlavně vyčítání obsahu EEPROM paměti není z důvodu velkých nároků na čas ve snímači polohy uplatnitelné, ovšem využití obvodu FT245R je postačující k programování nejrůznějších jednočipových mikrokontrolérů a malých sériových pamětí s SPI rozhraním. V praxi totiž často není délka přeloženého kódu tak velká, aby jeho nahrání do paměti zabralo více času. Navíc je pro uživatele potřebné především zapisování dat do paměti, vyčítání jejího obsahu se lze vzdát. Obvod FT245R je snadno dostupný a relativně levný, takže se přímo nabízí budoucí využití programátoru ve školních úlohách s mikroprocesory. Kromě jiného lze uplatnit obvod také při doplňkové komunikaci s PC, což se dříve řešilo zpravidla pomocí sériové linky RS-232. Pro budoucí využití a případné rozšíření se nachází příloze A zdrojové kódy plně funkčního programátoru.

Od dalšího vývoje snímače polohy s programátorem EEPROM v Bit Bang módu se tedy upustilo ze dvou důvodů:

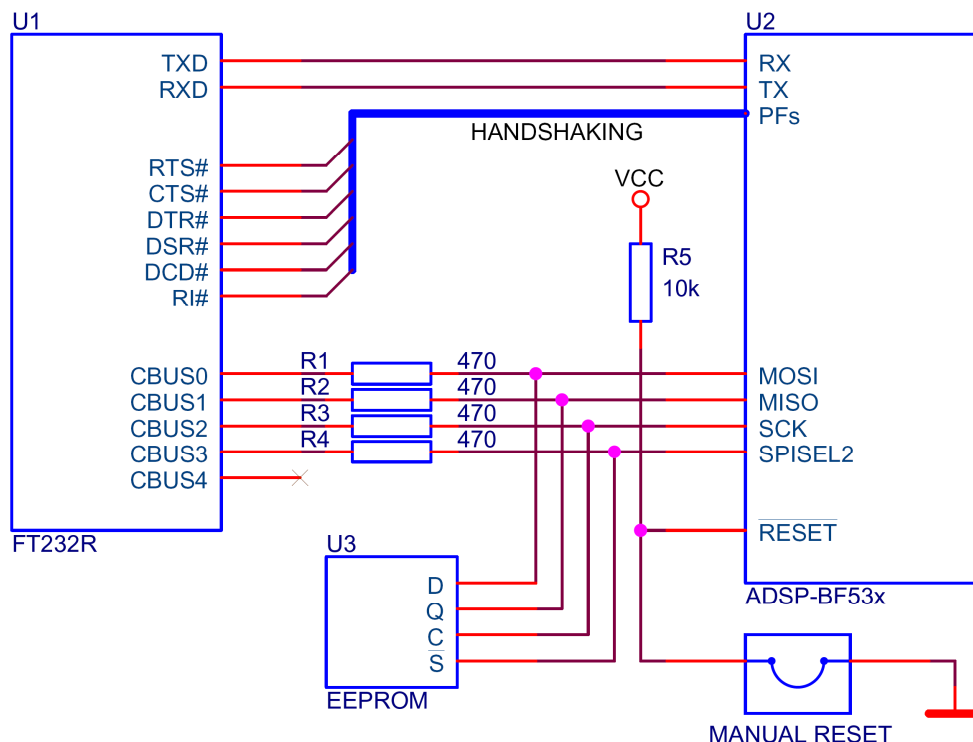
- pomalá rychlost SPI programování bootovací paměti EEPROM,
- nízká přenosová rychlost pro přenos obrazu ze senzoru od PC.

V následující kapitole je jen naznačeno možné řešení s Bit Bang programováním, ale další vývoj práce směřoval k řadiči FT2232H, jemuž se věnuje kapitola 2.2.3.

2.2.2 FT232R s programováním EEPROM v režimu Bit Bang

Obvod FT232R je rozhraním mezi USB a asynchronním sériovým přenosem s plnou podporou HW/SW handshakingu. Lze nastavit standardní parametry jako např. 7 nebo 8 datových bitů, 1 nebo 2 stop bity a lichou, sudou, mark, space nebo žádnou paritu. Přenosová rychlost sériové komunikace je nastavitelná v rozpětí 300 Bd – 3 MBd. Velikosti bufferů jsou 256 B na přijímací a 128 B na vysílací straně. Řadič využívá stejně jako FT245R interní integrovanou 128 B EEPROM, která slouží k uložení nejrůznějších parametrů obvodu, mezi něž patří např. také nakonfigurování linek CBUS jako zdroje výstupního obdélníkového hodinového signálu s kmitočty 48, 24, 12 a 6 MHz pro nejrůznější mikrokontroléry nebo FPGA. Další vlastnosti tohoto obvodu uvádí katalogový list [8].

Propojení FT232R s BF-53x mělo vycházet ze schématu podle obr. 2.9. Přenášení dat náhledového obrazu ze senzoru do USB řadiče mělo být uskutečňováno asynchronním sériovým přenosem po linkách RX, TX. Dále měly být využívány signály handshakingu pro řízení přenosu, avšak z důvodu nepokračování ve vývoji s tímto obvodem není handshaking ve schématu na straně procesoru detailně rozkreslen.



Obr. 2.9 Propojení FT232R, paměti EEPROM a BF-53x

Programování paměti EEPROM mělo být původně uskutečňováno v tzv. *CBUS Bit Bang módu*, což je Bit Bang mód, který má pouze FT232R. Pro aktivaci tohoto režimu na linkách CBUS0 - CBUS3 je nutné nastavení ve vnitřní konfigurační paměti. Obdobně jako v zapojení dle obr. 2.3 jsou vloženy v sérii s linkami ochranné odpory, aby nedošlo v případě kolize na sběrnici

SPI ke zničení některého ze zúčastněných obvodů. Stejně jako u FT245R měly být použity pro přístup k USB zařízení pod OS WINDOWS ovladače D2XX a příslušné knihovny.

Pro zápis/čtení se v CBUS Bit Bang módu využívají pouze funkce **FT_SetBitMode** a **FT_GetBitMode**. Funkce **FT_GetBitMode**, která již byla popsána v kapitole 2.2.1.1, se liší pouze v parametru *ucMode*, který je sice rovněž ukazatelem na unsigned char, ovšem platná hodnota je obsažena pouze ve spodních 4 bitech. Funkce **FT_SetBitMode** se liší od užití při asynchronním Bit Bang módu FT245R více, proto je zde připojen podrobnější popis:

- **FT_STATUS FT_SetBitMode** (**FT_HANDLE** *ftHandle*, **UCHAR** *ucMask*, **UCHAR** *ucMode*), kde
 - *ftHandle* je handler zařízení,
 - *ucMask* je 8-bitová hodnota, jejíž vyšší čtyři bity nastaví vlastnost linek CBUS3-CBUS0 (0 = vstup, 1 = výstup) a spodní čtyři bity nastaví na příslušné lince log. hodnotu 0 nebo 1 v případě, že dotyčná linka je nastavená jako výstupní,
 - *ucMode* je hodnota módu. Pro přepnutí FT232R do CBUS módu musí být zapsána hodnota 0x20.

Odůvodnění, proč ustoupit od dalšího vývoje SPI programátoru EEPROM v CBUS Bit Bang módu, byť i pomalého pro jednočipové mikrokontroléry, přináší citace [14] z kapitoly 1.4: *Jelikož funkce SetBitMode a GetBitMode umožňují odeslat nebo přijmout pouze jeden bajt, je tato verze Bit Bangu mnohem pomalejší než u asynchronního a synchronního typu, kdy jsou data přenášena pomocí funkcí FT_Write a FT_Read. Přenosová rychlost je totiž v CBUS módu limitována USB rámcí.*

2.2.3 Obvod FT2232H ve funkci programátoru EEPROM a USB řadiče pro rychlý přenos obrazových dat

Po zkušenostech s programováním USB řadičů FT245R a FT232R pomocí funkcí z D2XX knihoven bylo praktičtější dát přednost obvodu od stejného výrobce, tedy FT2232H, místo posledního zvažovaného obvodu CY7C6801. Pokračování ve vývoji s obvodem FT2232H, který byl v pouzdře LQFP – 64 s sebou totiž neslo riziko, že se nepodaří umístit všechny součástky na plošný spoj omezených rozměrů, tj. 60 × 60 mm. Tento obvod potřebuje ke své funkci na rozdíl od prvních dvou řadičů také externí konfigurační paměť EEPROM a krystal s keramickými kondenzátory jako zdroj hodinového signálu. V počátcích vývoje, kdy ještě nebyl plošný spoj navržen, byl obvod naletován na univerzální desce pro pouzdra typu LQFP – 64 a propojen s kontaktním nepájivým polem, vývojovým kitem s BF-532 [7] a ostatními součástkami, viz fotodokumentace v příloze B.

Obvod FT2232H vychází z FTDI produktů 5. generace a patří do třídy vysokorychlostních High Speed řadičů USB 2.0. V jednom pouzdře se nacházejí dvě nezávislá USB zařízení, tedy dva kanály A a B, pro něž lze nakonfigurovat následující režimy: *Async Serial (RS232)*, *Async 245 FIFO*, *Sync 245 FIFO*, *Async Bit Bang*, *Sync Bit Bang*, *MPSSE (Multi-Protocol Synchronous Serial Engine)*, *Fast Serial Interface*, *CPU FIFO Style*, *Host Bus Emulation*. Význam některých režimů je zjevný, u těch zbývajících jsou potřebné informace k dohledání v katalogovém listě [10]. Protože tento obvod nabízí více režimů, podle kterých by bylo možné části zapojení pro programování EEPROM a přenos dat z procesoru do PC realizovat, věnuje se následující kapitola stručnému odůvodnění výsledné volby.

2.2.3.1 Popis některých konfigurovatelných režimů řadiče FT2232H a možnost jejich uplatnění v zapojení snímače polohy

Při rozhodování o konečné podobě zapojení USB řadiče ve snímači polohy přicházely do úvahy režimy *Async Serial (RS232)*, *Async 245 FIFO*, *Sync 245 FIFO*, *MPSSE* a *CPU FIFO Style*. Postupnou rozvahou se zužoval výběr, až se nakonec ukázala optimální kombinace kanálu A v režimu *MPSSE – SPI* pro programování paměti EEPROM a kanálu B v režimu *Async 245 FIFO* pro přenos dat z procesoru do PC. Následující přehled se zabývá principy a vlastnostmi uvažovaných módů.

- **Async Serial (RS232)** – je vhodný pro asynchronní sériový přenos s plně podporovaným HW/SW handshakingem. Nejvyšší přenosová rychlost je 12 MBaud. Tento kanál měl být původně vyhrazen pro přenos dat jako v zapojení dle obr. 2.9, ovšem paralelní přenos byl výhodnější z důvodu vyšší rychlosti.
- **Async 245 FIFO** – obdoba rozhraní 245 FIFO s osmi datovými signály A(B)DBUS0 – A(B)DBUS7, tzv. „Busy Flagy“ RXF#, TXE#, a signály pro vyvolání zápisu/čtení WR#, RD#. Zapojení je stejné jako na obr. 2.3 s tím rozdílem, že pro přenos dat z procesoru je využíváno všech osmi datových linek. Rychlost přenosu v tomto režimu může dosahovat až 8 MB/s, proto byl zvolen v jednom kanálu jako režim pro přenos obrazových dat.
- **Sync 245 FIFO** – tento režim, jehož přenosová rychlost činí více než 25 MB/s, by byl ideální pro rychlý přenos velkého objemu obrazových dat. Jeho nevýhodou je, že v tomto režimu je nedostupný druhý kanál B potřebný k programování EEPROM. Datové i řídicí signály jsou shodné jako u *Async 245 FIFO*, navíc však obsahuje ještě OE# a CLKOUT. Signál CLKOUT je výstup 60 MHz hodin, podle kterých musí být veškerý přenos v tomto režimu synchronizován, což je další důvod, proč není v zapojení snímače polohy vhodný – sensorové desky typu *Micron*, které měly být používány, dovolují nejrychlejší taktovací

signál 48 MHz, takže by v tomto případě muselo být navíc zapojení doplněno o děličku kmitočtu.

- **MPSSE** – u obou kanálů A i B lze nastavit v tomto režimu nezávisle na sobě mnoho průmyslových standardů jako jsou JTAG, I²C, SPI nebo implementovat vlastní sběrniceový protokol. V zapojení snímače polohy byl použit mód SPI k programování bootovací EEPROM, tj. signály SK, DO, DI a CS. Rychlost SPI může být až 30 Mbit/s, což s rezervou překračuje mezní kmitočty uvažovaných sériových EEPROM pamětí. V režimu MPSSE lze využívat kromě SPI sběrnice zároveň i ostatní piny příslušného kanálu, které jsou nakonfigurovány jako GPIO, viz kapitola 2.2.1. To je u snímače polohy výhodné např. k tomu, že pomocí jedné linky GPIO může být procesor držen ve stavu reset po dobu programování EEPROM a nemusí být tento úkon prováděn manuálně uživatelem.
- **CPU FIFO Style** – princip této konfigurace se podobá Async 245 FIFO s tím rozdílem, že nejsou dostupné signály RXF#, TXE#, ale jsou zde navíc výběrový signál CS# a adresní signál A0. Vysvětlení tohoto režimu se opírá o tab. 2.2, ze které je patrné, že příslušný kanál USB řadiče se chová jako vstupně/výstupní brána, která je namapovaná do určitého adresního prostoru. Pro zjištění stavu zařízení je však nezbytný tzv. *Polling* (dotazování), z čehož vyplývá, že před každým zápisem/čtením musí být testováno, zda je v příslušné paměti FIFO volné místo pro zápis/jsou dostupná data k vyčtení. Tato procedura ovšem natolik zaměstnávala procesor, že se pro přenos obrazu ukázal výhodnější režim Async 245 FIFO.

CS#	A0	RD#	WR#
1	X	X	X
0	0	čtení dat z pipe	zápis dat do pipe
0	1	vyčtení stavu	okamžitý zápis

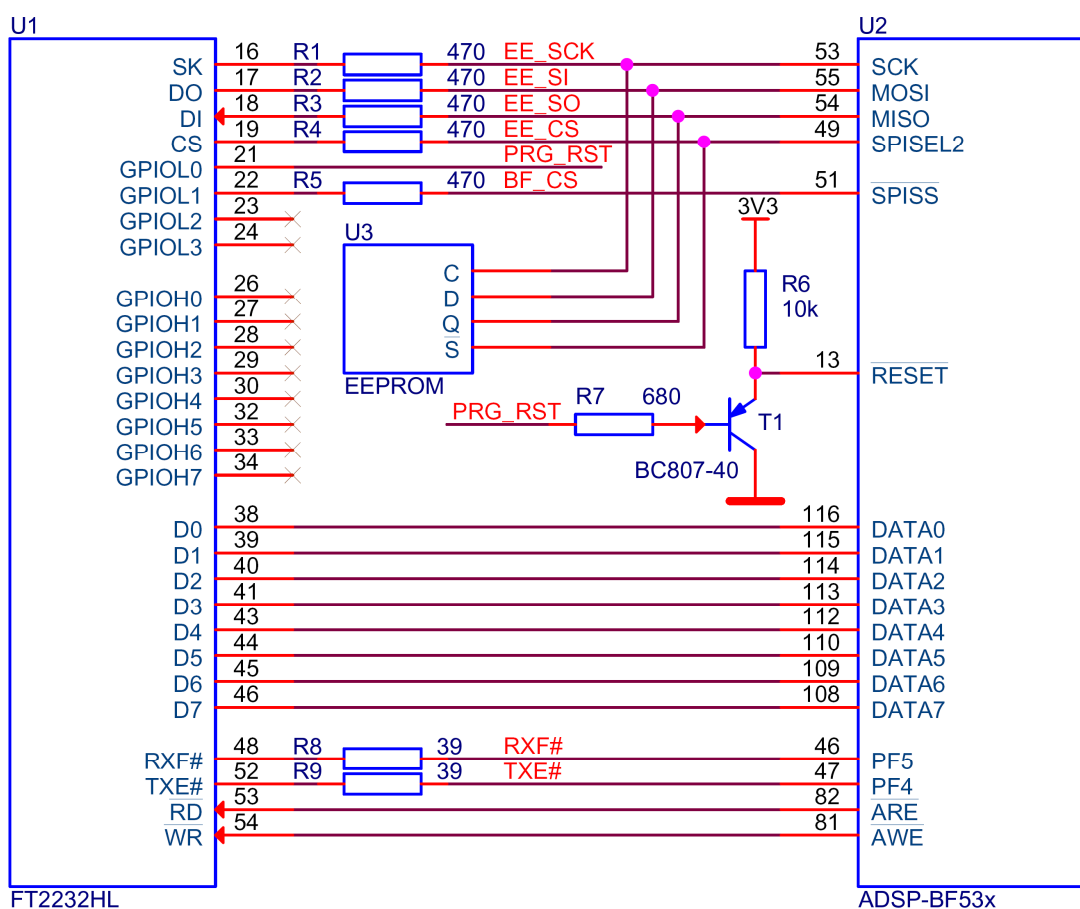
Tab. 2.2 Výběr operace v režimu CPU FIFO Style ⁹

⁹ Převzato z katalogového listu FT2232H [10].

2.2.3.2 Výsledná koncepce zapojení řadiče FT2232H pro přenos dat z procesoru BF-53x a programování bootovací paměti EEPROM

Z výše uvedených rozborů režimů, které obvod FT2232H nabízí, byl vybrán *MPSSE SPI* k programování EEPROM a *Async 245 FIFO* pro přenos dat mezi procesorem a PC. Principiální zapojení je naznačeno na obr. 2.10. Pro přechod do režimu *Async 245 FIFO* musí být provedena konfigurace v externí paměti EEPROM, která není ve schématu u obvodu FT2232H zakreslena. Možnosti tohoto zapojení jsou následující:

- SPI kanál (SK, DO, DI, CS) pro programování bootovací EEPROM,
- doplňková SPI komunikace mezi FT2232H (Master) a BF-53x (Slave) pomocí linek SK, DO, DI, GPIOL1,
- řízený reset BF-53x pomocí linky GPIOL0 během procesu programování paměti (ve zjednodušeném schématu naznačen tranzistorem T₁, ve skutečném pak doplněn obvodem ADM708TAR),
- další linky GPIOL2, GPIOL3 a GPIOH0 - GPIOH7 pro libovolné využití,
- přenos obrazu a dat po paralelní 8-bitové sběrnici D0 - D7.



Obr. 2.10 Propojení FT2232H, paměti EEPROM a BF-53x

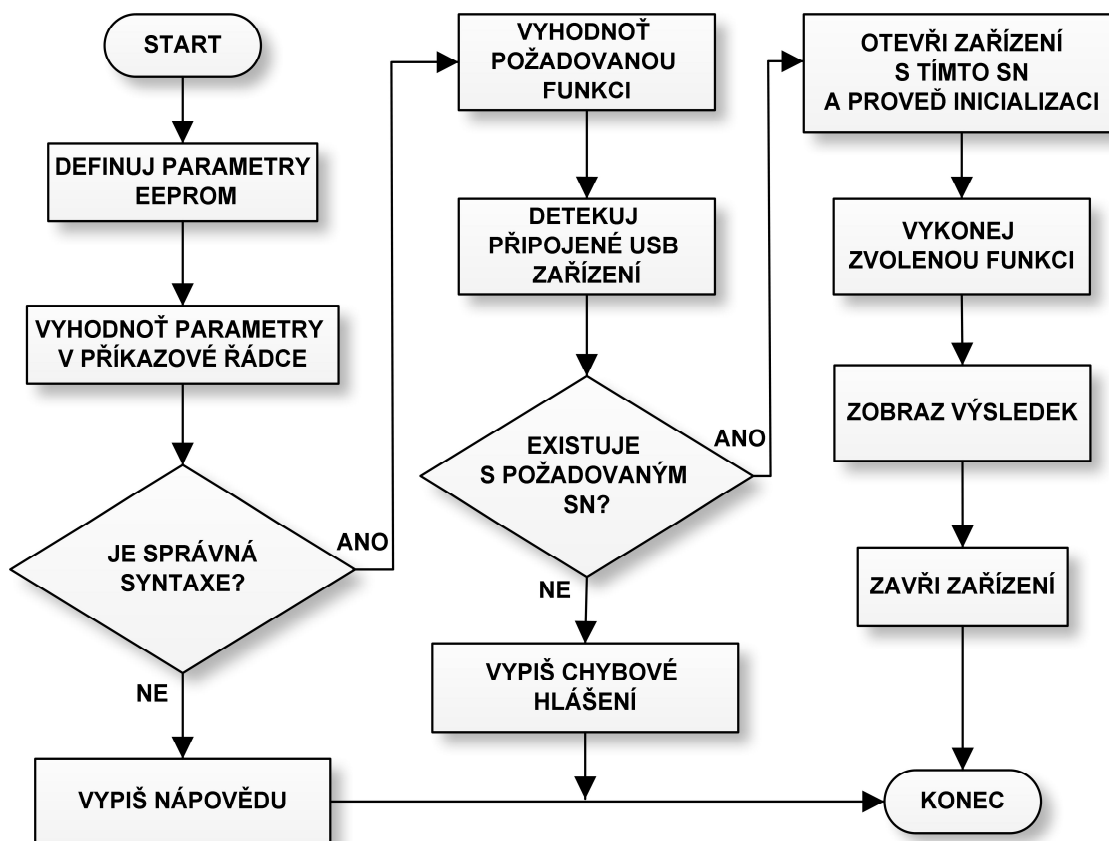
V zapojení jsou dále zakresleny ochranné sériové odpory, jejichž význam je stejný jako v kapitole 2.2.1, tedy $R_1 - R_4$ slouží k ochraně při konfliktu na SPI sběrnici, R_5 je zapojen proto, že ke kolizi by mohlo dojít tehdy, kdyby byl pin 51 procesoru v režimu výstupního flagu a zároveň by byl prováděn select na SPI signálu GPIOL1 u FT2232H, který přistupuje ke sběrnici pouze jako master. Odpory R_8 a R_9 mají obdobný význam – flagy PF4 a PF5 by rovněž mohly být omylem nastaveny jako výstupní, signály RXF# a TXE# u FT2232H jsou výstupní v tomto režimu vždy. Hodnota odporů 39Ω je z toho důvodu, že změny těchto signálů jsou podstatně rychlejší než na SPI sběrnici (až 10 MHz), proto musí být zajištěna i menší časová konstanta, tj. po dosažení do (2.1) vychází $\tau = 6,1 \text{ ns}$. Tato hodnota odporů je zároveň vhodná k potlačení odrazů na vedení.

2.2.3.3 Programování EEPROM obvodem FT2232H v režimu MPSSE SPI

Obvod FT2232H má v režimu MPSSE přímou podporu SPI standardu, který umožňuje sériovou komunikaci s ostatními zařízeními. Tento USB řadič je možné provozovat pouze jako zařízení master, což ovšem není v tomto případě tak závažný nedostatek, protože k programování EEPROM musí být FT2232H master a pro komunikaci s BF-53x lze tento procesor naprogramovat jako zařízení SPI slave. Důležitým požadavkem na HW počítače je, že musí podporovat High Speed USB 2.0, neboť nižší verze standardu neumožňují přistupovat k některým funkcím SPI DLL¹⁰. Podobně jako knihovny D2XX funkcí, které byly popsány v kapitole 2.2.1.1, musí být do projektu zvoleného programovacího jazyka přilinkována knihovna *FTCSPI.LIB* s hlavičkovým souborem *FTCSPI.H*. Podrobný popis SPI funkcí, které tato knihovna pro obvody FT2232H, FT2232D a FT4232H nabízí, lze dohledat v dokumentu [17].

Výsledná aplikace je použitelná jako konzolový programátor SPI EEPROM paměti a její zdrojové kódy v jazyce C++ se nachází v příloze A. Je postavena na ukázkovém příkladu výrobce FTDI ke komunikaci FT2232H se sériovou SPI EEPROM Microchip 93LC56, viz [18]. Vlastní podstata programátoru je naznačena ve vývojovém diagramu dle obr. 2.11. Následující text obsahuje stručné vysvětlení některých bloků.

¹⁰ Ovladače SPI DLL pro OS Windows se nacházejí na stránkách výrobce (11/2010): <http://www.ftdichip.com/Support/SoftwareExamples/MPSSE/FTCSPI/FTCSPI.zip>



Obr. 2.11 Vývojový diagram programátoru EEPROM s FT2232H

Nastavení parametrů EEPROM – po spuštění programu se nejprve načtou parametry z konfiguračního souboru, tj. typ EEPROM, velikost paměti v bajtech, velikost sektoru v bajtech pro sektorové mazání, velikost stránky v bajtech, rychlost SPI komunikace v MHz a kódy instrukcí paměti, tzn. např. READ, WRITE, WRDI, WREN apod.

Pozn.: Programátor byl vyvíjen a testován na dvou druzích EEPROM pamětí, konkrétně na dříve zmíněné 25LC256 [16] a na 8 Mbit paměti M25P80 od výrobce STMicroelectronic [19], což je paměť s velikostí stránky 256 B, celkovým i sektorovým mazáním a maximální frekvencí 25 MHz pro hodinový signál. Další popis programátoru je zaměřen na M25P80, neboť ta má rozdíl od 25LC256 zmíněné sektorové a celkové mazání.

Vyhodnocení parametrů příkazové řádky – úkolem je rozpoznání argumentů na příkazové řádce, tzn. zjištění požadované funkce, názvu vstupního/výstupního souboru, počtu bajtů, popř. adresu. Při nesprávném zápisu dojde k vypsání nápovědy se syntaxí a ukončení programu. Syntaxe je obdobná jako u programátoru s FT245R v Bit Bang módu a programátor byl navíc rozšířen o tyto funkcionality:

- **m25p80_prog.exe -be** provede kompletní smazání celé paměti, tzn. zapíše na všechny bity hodnotu log. 1,
- **m25p80_prog.exe -se addr** smaže obsah paměti od adresy *addr*.

Detekce, otevření a inicializace připojeného USB zařízení – program provede diagnostiku všech USB zařízení od výrobce FTDI, které jsou připojeny k PC, a najde-li řadič FT2232H s kanálem, jehož sériové číslo je shodné s definovaným, pak dojde k jeho otevření. Následuje posloupnost elementárních operací jako jsou reset otevřeného zařízení, inicializace MPSSE, vyčištění bufferů, nastavení SPI frekvence, konfigurace SPI signálů, signálu pro reset a ostatních GPIO linek.

Vykonání zvolené funkce – nejprve je aktivován na pinu GPIOL0 reset procesoru, aby byla uvolněna SPI sběrnice, a poté je spuštěn algoritmus, který provádí zápis, čtení nebo mazání paměti. Proběhne-li spuštěná operace korektně, dojde po jejím ukončení k zobrazení doby trvání operace (tato funkce přetrvala z období programování v Bit Bang módu, kdy hrál tento údaj rozhodující roli) a vypíše se počet zapsaných/vyčtených/smazaných bajtů. Následuje zavření USB zařízení a program je ukončen.

Pro představu, jakým způsobem se řadič FT2232H v režimu MPSSE SPI programuje, bude proveden rozbor knihovní funkce **SPI_WriteHiSpeedDevice**, která se vyskytuje na mnoha místech kódu programátoru. Spolu s touto funkcí jsou významné i další funkce jako např. **SPI_ReadHiSpeedDevice**, **SPI_SetHiSpeedDeviceGPIOs**, které jsou podrobně zdokumentované v [17]. Generování signálů protokolu SPI tímto řadičem se zásadně liší od způsobu v módu Bit Bang, který byl testován v počátcích vývoje s obvodem FT245R. Proto je v tab. 2.3 uveden přehled a význam všech argumentů výše uvedené funkce pro High Speed zápis.

Argument	Význam
FTC_HANDLE <i>ftHandle</i>	Handler FTDI Hi-Speed zařízení.
PFTC_INIT_CONDITION <i>pWriteStartCondition</i>	Ukazatel na strukturu, která obsahuje počáteční stavy (H/L) signálů SK, DO a CS.
BOOL <i>bClockOutDataBitsMSBFirst</i>	Sled sériových dat z DO. TRUE = první bit MSB, FALSE = první bit LSB
BOOL <i>bClockOutDataBitsPosEdge</i>	Hradlování datových bitů podle hodinového signálu. TRUE = náběžná hrana, FALSE = spádová hrana SK.
DWORD <i>dwNumControlBitsToWrite</i>	Počet řídicích bitů, které budou zapsány na externí zařízení. Platný rozsah je 2 – 2040, tedy 2040 bit odpovídá 255 B.

DWORD <i>dwNumControlBytesToWrite</i>	Počet bajtů v bufferu s řídicími daty pro zápis, platný rozsah je 1 – 255 B.
BOOL <i>bWriteDataBits</i>	Povolení (TRUE)/zakázání (FALSE) zápisu datových bajtů na externí zařízení.
DWORD <i>dwNumDataBitsToWrite</i>	Počet datových bitů, které budou zapsány na externí zařízení. Platný rozsah je 2 – 524280, tedy 524280 bit odpovídá 64 kB.
PWriteDataByteBuffer <i>pWriteDataBuffer</i>	Ukazatel na buffer, který obsahuje data pro zápis na externí zařízení.
DWORD <i>dwNumDataBytesToWrite</i>	Počet bajtů v datovém bufferu, které mají být zapsány. Platný rozsah je 1 – 65355 B.
PFTC_WAIT_DATA_WRITE <i>pWaitDataWriteComplete</i>	Ukazatel na strukturu, v níž je nastaveno, má-li FT2232D čekat před návratem z procedury na zápis všech dat na externí zařízení.
PPTH_HIGHER_OUTPUT_PINS <i>pHighPinsWriteActiveStates</i>	Ukazatel na strukturu, ve které je nastaveno, zda piny GPIOH0 – GPIOH7 budou během zápisu také využívány.

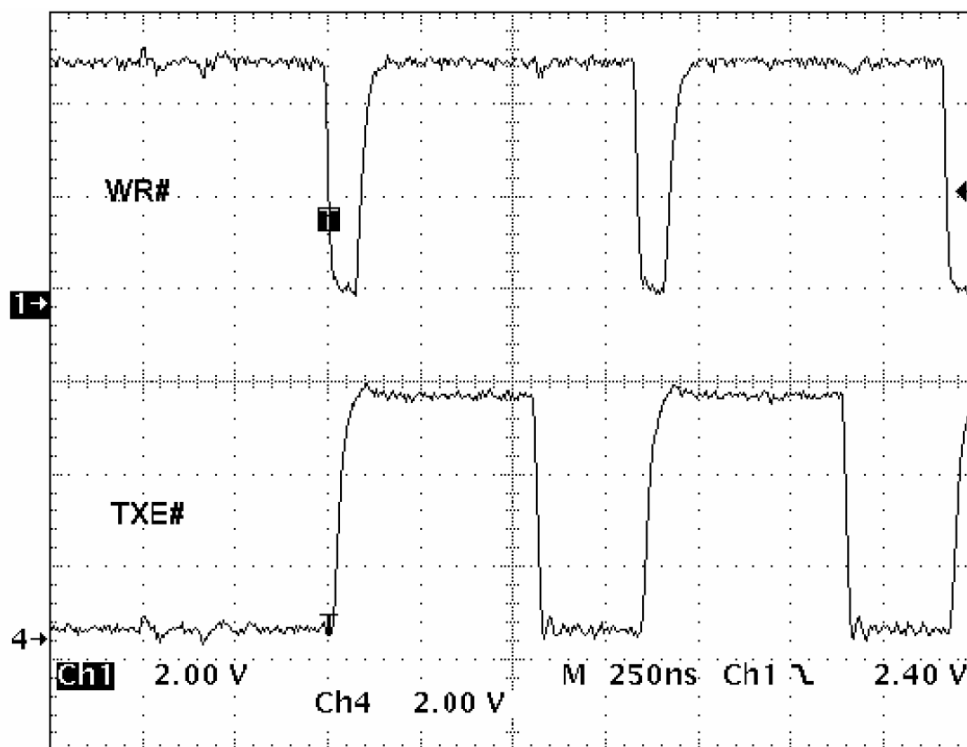
Tab. 2.3 Přehled a význam argumentů funkce *SPI_WriteHiSpeedDevice*¹¹

2.2.3.4 Přenos dat USB řadičem FT2232H mezi PC a BF-53x

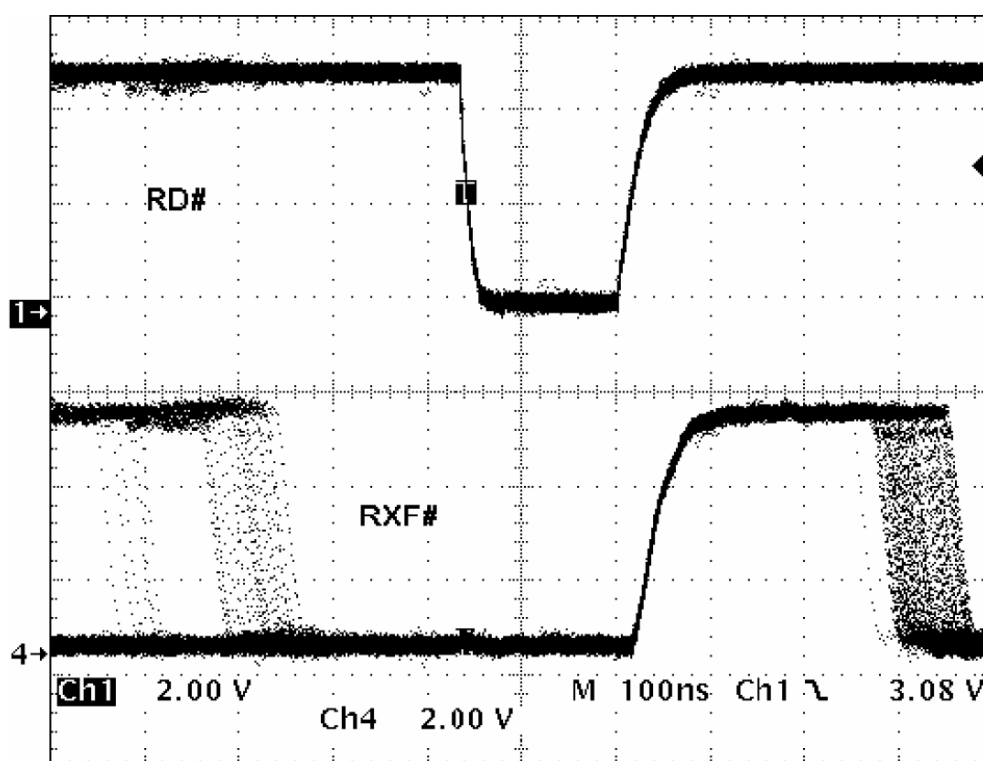
Jak již bylo uvedeno v kapitole 2.2.3.2, pro přenos dat mezi procesorem a USB řadičem byl zvolen 8-bitový paralelní přístup v režimu *Async 245 FIFO*, viz zapojení na obr. 2.10. Obvod FT2232H má shodnou velikost FIFO pamětí na přijímací i vysílací straně, konkrétně 4096 B. Pro vysvětlení principu obousměrného přenosu budou v dalším textu popsána obě rozhraní obvodu, tj. *PC ↔ USB* a *245 FIFO ↔ BF-53x*.

Rozhraní 245 FIFO ↔ BF-53x – u FT2232H jsou dva výstupní signály s označením TXE#, RXF#, které slouží k indikaci stavu vysílacího a přijímacího FIFO bufferu. Překlad popisu signálů od výrobce je spíše zavádějící, neboť používá označení *Transmitter Empty* – vysílač prázdný pro signál TXE# a *Receiver Full* – přijímač plný pro signál RXF#. Příhodnější by pravděpodobně bylo označení *Transmitter Ready* a *Receiver Ready*. Na obr. 2.12 a obr. 2.13 jsou průběhy těchto signálů od výrobce, které toto tvrzení dokazují. V průbězích jsou zároveň zobrazeny signály pro vykonání zápisu jednoho bajtu – WR# a vyčtení jednoho bajtu RD#.

¹¹ Překlad z aplikační poznámky AN_111 pro FTCSPI.DLL [17].



Obr. 2.12 Průběhy signálů WR# a TXE# na rozhraní Async 245 FIFO¹²



Obr. 2.13 Průběhy signálů RD# a RXF# na rozhraní Async 245 FIFO¹²

¹² Převzato z aplikační poznámky AN232B-06 pro FT245BM [20], průběhy jsou stejné jako u obvodu FT2232H, odlišnost je pouze ve velikosti časové základny, neboť FT2232H umožňuje přenášet data v režimu Async 245 FIFO rychlostí až 8 MB/s.

Z obr. 2.12 je patrné, že zápis dat z procesoru do USB řadiče je vykonán se spádovou hranou signálu WR# a že po každém zápisu dojde k nastavení signálu TXE# na určitou dobu do log. 1, což jinými slovy znamená, že po tuto dobu není možné zapisovat do obvodu další data, ačkoliv počet dříve zapsaných bajtů nepřekračuje kapacitu FIFO bufferu. Tím pádem není označení *Full* namístě. Tímto jevem se obvod odlišuje od tradiční FIFO paměti, kam lze při dodržení prodlevy t_{WR} (*Write Recovery Time*) opakovaně zapisovat až do zaplnění celé paměti, kdy je teprve příznak *Full* nastaven [21]. Nutnost testování signálů TXE# a RXF# je nepříjemná z důvodu zpomalení algoritmu zápisu/čtení v procesoru. Více se rychlým zápisem a čtením paralelních dat procesorem BF-53x zabývá kapitola 3.1.2.

Na obr. 2.13 je zachycena situace při čtení dat z USB řadiče. Číst data je možné, pokud se ve FIFO paměti nacházejí, což indikuje log. 0 signálu RXF#. Vyčtení jednoho bajtu proběhne při spádové hraně signálu RD#, po kterém následuje dočasný přechod RXF# do log. 1, kdy je další čtení zakázáno, ačkoliv tato informace žádným způsobem nevypovídá o dostupnosti dat ve FIFO bufferu kromě toho, že pokud se v Rx bufferu žádná data nenacházejí, je signál RXF# na hodnotě log. 1 trvale. Obdobně je tomu u Tx bufferu, kdy zůstane signál TXE# také v hodnotě log. 1 až do té doby, než jsou data z USB řadiče na straně PC vyčtena.

Rozhraní PC ↔ USB – ze strany PC se s daty v režimu Async 245 FIFO manipuluje standardně prostřednictvím D2XX funkcí, které byly popsány v kapitole 2.2.1.1. Kromě funkcí **FT_Open**, **FT_Close**, **FT_Write** a **FT_Read** je často používána funkce **FT_GetStatus**, která vrací počet bajtů v přijímacím a vysílacím FIFO bufferu. Tato informace je užitečná k řízení algoritmu obslužného programu, např. při rychlém vyčítání obrazových dat. Detailnější popis této problematiky poskytuje kapitola 3.1.2.3.

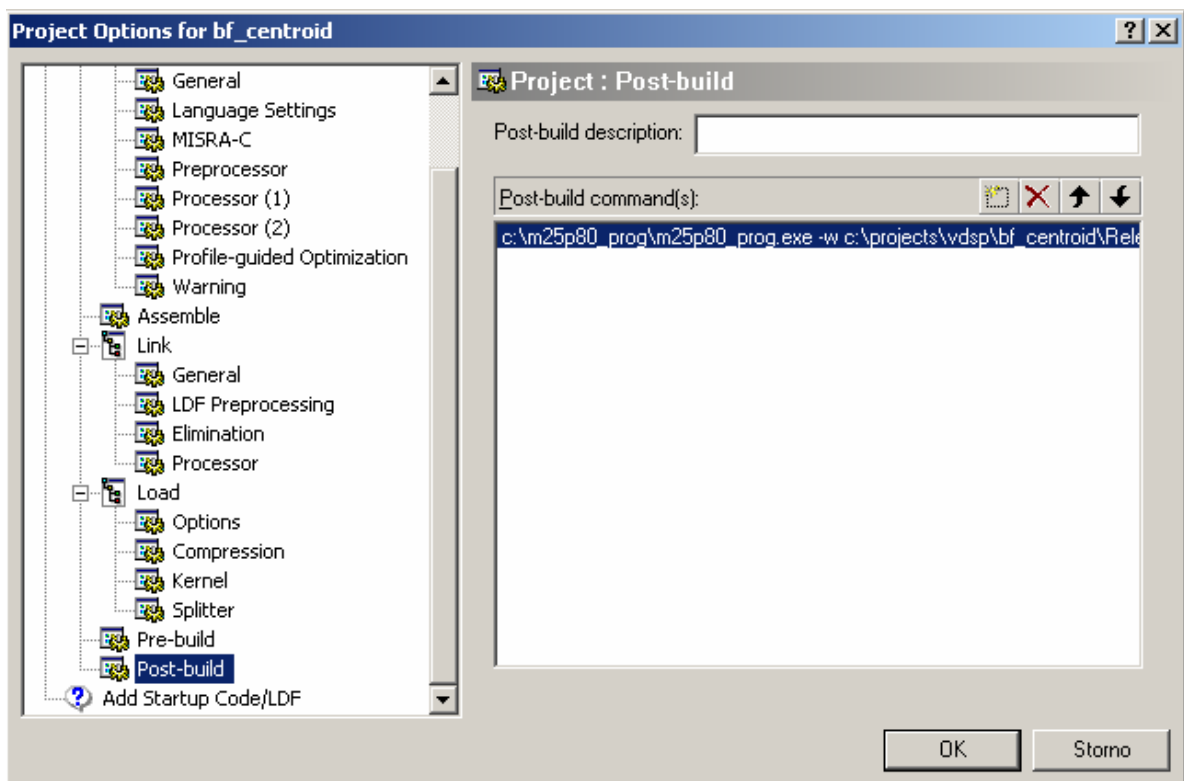
2.2.3.5 Zhodnocení vlastností programátoru EEPROM s FT2232H v režimu MPSSE SPI a paralelního 8-bitového přenosu dat

S řadičem FT2232H se podařilo realizovat plně funkční programátor bootovací EEPROM paměti pro procesor BF-53x. Rychlost zápisu a vyčítání EEPROM je dostatečná, srovnání s předchozím způsobem v Bit Bang módu a obvodu FT245R je v tab. 2.4. Naměřené hodnoty jsou platné pro přenos 32768 B dat oběma směry mezi USB řadičem a pamětí 25LC256.

	zápis 32768 B	vyčtení 32768 B
FT245R + Bit Bang	35 s	1680 s
FT2232H + MPSSE SPI	3,063 s	0,281 s

Tab. 2.4 Časy potřebné k zápisu/vyčtení celé EEPROM 25LC256

Programátor byl napsán genericky, tzn. že drobnou úpravou ve zdrojovém kódu lze realizovat programovací aplikaci pro libovolný typ sériové EEPROM paměti s SPI rozhraním. Během vlastního vývoje HW byly naprogramovány konzolové aplikace pro paměti *Microchip* 25LC256 a *STMicroelectronic* M25P80. Protože paměť M25P80 umožňuje i sektorové a celkové mazání, byly do kódu přidány funkcionality pro tyto operace. Jelikož je pinem GPIO0 řízen také reset procesoru, je možné celý proces naprogramování EEPROM zautomatizovat a integrovat např. do vývojového prostředí *VisualDSP++*, které je uzpůsobené ke tvorbě zdrojových kódů pro procesory *Blackfin*. Způsob, jak nastavit vývojové prostředí, přibližuje obr. 2.14. Ve vlastnostech projektu (ALT+F7) se nachází položka *Post-build*, ve které lze nastavit posloupnost příkazů, které se mají vykonat po překladu zdrojových kódů. Výstupem překladu je mimo jiných souborů i soubor s příponou *.ldr*, což je takzvaný loader, který má být nahrán do příslušného procesoru při bootování. Parametry projektu se nastavují zvlášť pro *Debug* a pro *Release*, což je nutné vědět při zadávání cesty k vytvořenému *.ldr* souboru. Po překladu tedy mohou být vykonány *Post-build* příkazy, tedy i naprogramování EEPROM nově přeloženým kódem.



Obr. 2.14 Dialogové okno prostředí *VisualDSP++* pro nastavení *Post-build* příkazů

Pozn.: Parametry souboru loaderu je nutné nastavit v *Project : Load : Options* tak, že *Boot Mode = SPI*, *Boot Format = Binary* a *Output Width = 8-bit*.

Příklad: Necht' aplikace programátoru se nachází v adresáři **c:\m25p80_prog** a projekt **bf_centroid.dpj** v adresáři **c:\projects\vdsp\bf_centroid**. Pak Post-build příkaz pro zápis loaderu do EEPROM bude následující:

c:\m25p80_prog\m25p80_prog.exe -w c:\projects\vdsp\bf_centroid\Release\bf_centroid.ldr

Výhodou tohoto nastavení je, že při ladění programu pro procesor nemusí být po každém překladu otevírána aplikace programátoru, zadávána cesta ke změněnému loaderu a následné programování paměti. V prostředí VisualDSP++ stačí pouze vyvolat kompilaci projektu (zdrojového kódu) a zbylé operace (programování EEPROM a reset procesoru) se vykonají automaticky, což při absenci JTAGu může ušetřit mnoho času.

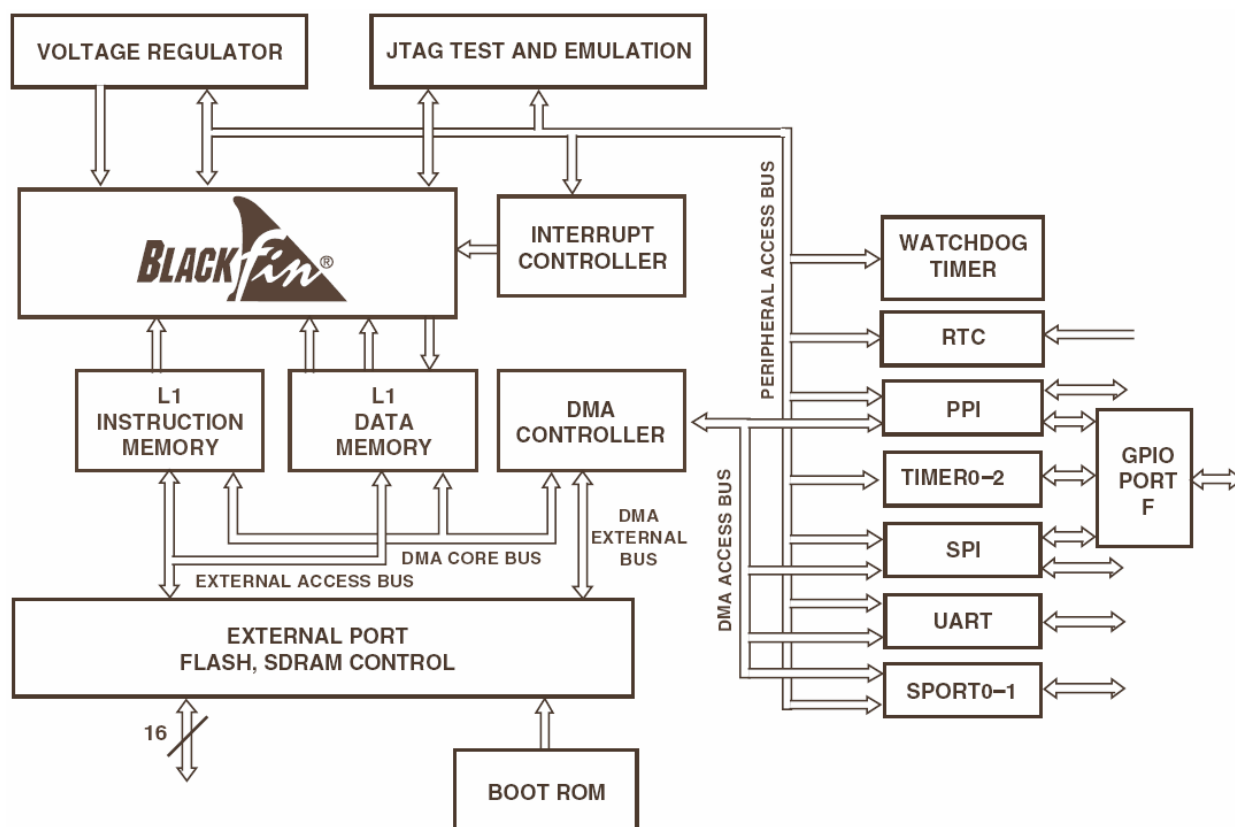
Druhý požadavek na USB řadič, tj. co nejrychlejší přenos dat mezi procesorem a PC byl splněn tak, že maximální zaručená rychlost činí 8 MB/s. Tato hodnota je již dostatečná pro přenos celého snímku z CMOS senzoru v plném rozlišení (např. 1280 × 1024) bez použití externí paměti. Více se způsobem přenosu obrazu zabývá kapitola 3.1.2. Způsob komunikace mezi PC a procesorem přes 8-bitové paralelní rozhraní a paměti FIFO obvodu FT2232H je založen na packetovém principu, který blíže popisuje kapitola 4.2.1.

Závěrem lze říci, že oba požadavky na USB řadič byly splněny, takže již nebylo zapotřebí provádět další experimenty s řadičem *Cypress Semiconductor* CY7C680. Mezi nevýhody řešení s FT2232H patřila především velikost pouzdra a potřeba dalších externích součástí, takže výsledné schéma zapojení přinášelo riziko, že se nepodaří navrhnout plošný spoj s omezenými rozměry 60 × 60 mm, viz kapitola 4.1.3.

2.3 Procesorová část snímače polohy

2.3.1 Stručný přehled vlastností procesorů Blackfin řady BF-53x

Procesory Blackfin s označením BF531/2/3 byly již mnohokrát popsány v nejrůznější literatuře např. [7], [22], [23]. Proto budou v této kapitole shrnuty jen základní rysy. V dalším textu pak bude uveden bližší popis jen těch částí, které se nějakým způsobem vztahují k uplatnění v realizovaném snímači polohy. Na obr. 2.15 je znázorněno blokové schéma procesorů této řady.



Obr. 2.15 Funkční blokové schéma procesorů Blackfin ADSP BF-53x¹³

V jádře procesoru jsou dvě 16-bitové násobičky, dva 40-bitové akumulátory, dvě 40-bitové aritmeticko-logické jednotky, čtyři video aritmeticko-logické jednotky a jeden 40-bitový posuvný registr. Instrukční model je postaven na redukované sadě RISC. Vnitřní paměť o velikosti až 148 kB (u BF-533), řadič pro vnější paměti typu SDRAM, SRAM, FLASH a ROM, možnost bootování z vnější SPI paměti. Z periférií je třeba zmínit PPI (*Parallel Peripheral Interface*) s podporou video formátů dat ITU-R 656, dva duální kanály pro plně duplexní synchronní sériovou komunikaci s podporou osmi stereo I²S kanálů, dva DMA řadiče

¹³ Převzato z katalogového listu ADSP-BF531/ADSP-BF532/ADSP-BF533 [22].

pro mezipaměťový přesun dat a osm DMA řadičů pro další periferie (SPORT, SPI, PPI, UART). Mezi další periferie patří SPI rozhraní se sedmi selektovacími signály v režimu master, tři 32-bitové čítače/časovače s podporou PWM, hodiny reálného času a watchdog časovač, jeden 32-bitový časovač jádra, až 16 pinů GPIO, rozhraní UART s podporou IrDA, debugovací rozhraní JTAG a jednotka PLL s násobičkou vstupního hodinového signálu, viz [22].

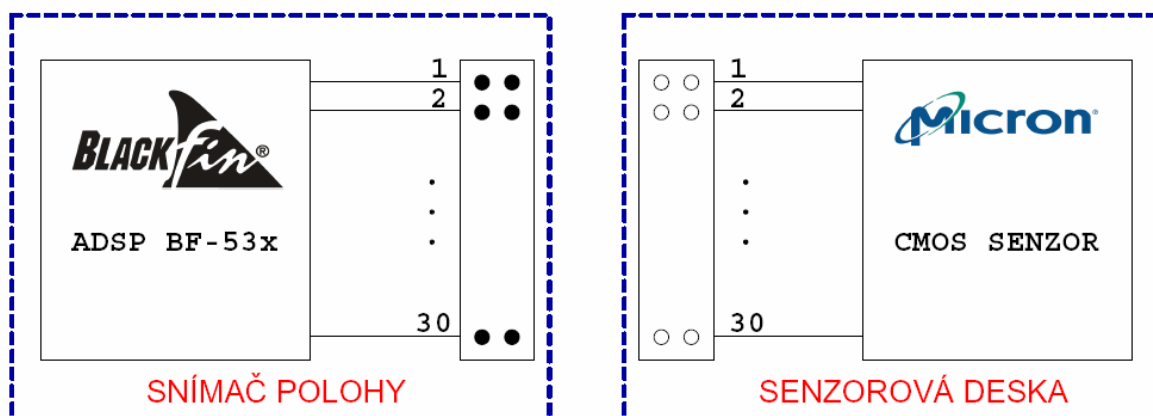
Výše uvedené rysy jsou pro všechny tři typy procesorů BF-53x společné. Mají i stejná pouzdra (160-CSPBGA, 169-PBGA a 176-LQFP) se shodným rozmístěním pinů. Jediným rozdílem je rychlost a velikost vnitřní paměti. Procesory BF-531 a BF-532 mají početní výkon 800 MMACs a mezní kmitočet hodin jádra CCLK 400 MHz. U BF-533 výkon dosahuje až 1200 MMACs a CCLK 600 MHz v pouzdře BGA. Srovnání velikosti různých typů paměti u těchto procesorů ukazuje tab. 2.5.

Typ paměti	BF-531	BF-532	BF-533
L1 Instr. SRAM/Cache	16 kB	16 kB	16 kB
L1 Instr. SRAM	16 kB	32 kB	64 kB
L1 Data SRAM/Cache	16 kB	32 kB	32 kB
L1 Data SRAM	–	–	32 kB
L1 Scratchpad	4 kB	4 kB	4 kB
L1 Celkem	52 kB	84 kB	148 kB
L3 Boot ROM	1 kB	1 kB	1 kB

Tab. 2.5 Přehled velikostí pamětí procesorů ADSP BF-53x

2.3.2 Propojení procesoru BF-53x s obrazovým senzorem

Důležitým úkolem bylo navázání signálů obrazových senzorů *Micron* MT9M001 aMT9V032 na procesor a ostatní obvody snímače polohy. Sensorové desky mají katedrou standardizované rozměry a rozložení pinů na propojovacím konektoru, čemuž musel být snímač polohy přizpůsoben, viz obr. 2.16.



Obr. 2.16 Znáznornění propojení procesoru BF-53x se signály sensorové desky

Piny obrazového CMOS senzoru MT9M001 lze přibližně rozdělit do těchto skupin¹⁴:

- komunikace I²C (SCLK, SDATA),
- řídicí signály (RESET#, OE#, TRIGGER, STANDBY),
- signály pro přenos obrazových dat (D0 - D9),
- synchronizační signály (LINE_VALID, FRAME_VALID, STROBE),
- hodinové signály (CLKIN, PIXCLK),
- rozvod napájecího napětí (VDD, DGND, VAA, AGND, VAAPIX).

Přehled o využitých signálech a způsobu propojení konektoru senzoru s procesorem BF-53x je popsán v tab. 2.6. V dalších odstavcích bude pojednáno o vlastnostech tohoto zapojení z hlediska procesoru i obrazového senzoru.

číslo pinu konektoru	signál na konektoru	pin na procesoru nebo signál v obvodu
1	STROBE	-
2	OE#	-
3	SDA	PF10
4	SCLK	PF9
5	SNAPSHOT	PF11
6	RESET#	RESET
7	PWDN	-
8	HSYNC	PPI_FS1
9	VSYNC	PPI_FS2
10	PCLK	PPI_CLK
14	MCLK	MCLK
11, 12, 13, 15, 16	GND	GND
19 - 20	D0 - D1	-
21 - 28	D2 - D9	PPI0 - PPI7
29, 30	VDD	3V3

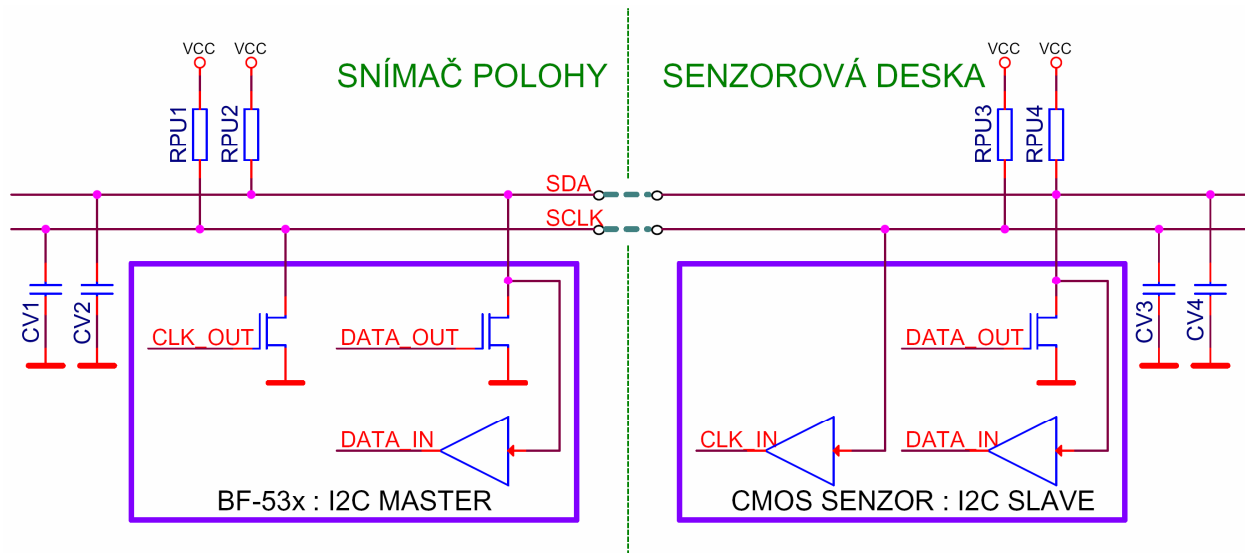
Tab. 2.6 Propojení konektoru senzorové desky s procesorem a celým obvodem

2.3.2.1 Komunikace I²C

Jedná se o obousměrný sériový přenos dat mezi procesorem a senzorem. Fyzickou vrstvu tvoří dvojvodičová sběrnice se signály SDA (sériová data) a SCL (sériové hodiny). Obě linky sběrnice musí být opatřeny *pull-up* odpory, aby byla na sběrnici v klidovém stavu úroveň log. 1. Hodnota těchto odporů by měla být taková, aby nedocházelo k neúnosnému zaoblení

¹⁴ U senzorové desky s MT9V032 jsou na konektor vyvedeny téměř identické signály, proto zde nebudou z důvodu zachování přehlednosti uváděny.

obdélníkového signálu a zároveň neprotékal příliš velký proud tranzistorovými budiči s otevřeným kolektorem. Každý *pull-up* odpor společně s kapacitou spoje sběrnice představuje integrační článek, jehož časová konstanta určuje mezní kmitočet, při kterém bude ještě možné komunikaci uskutečnit. Situace v navrhovaném propojení procesoru a obrazového senzoru je znázorněna na obr. 2.17.



Obr. 2.17 Sběrnice I²C mezi procesorem a CMOS senzorem

Každý z kondenzátorů C_{V1} - C_{V4} představuje parazitní kapacitu, na které se podílí kapacita spoje a kapacita příslušného I²C budiče. Odporů R_{PU3} a R_{PU4} je sice možné na sensorových deskách osadit, ovšem zpravidla na plošném spoji osazeny nejsou. Z toho důvodu bylo nutné počítat raději i s odporů R_{PU1} a R_{PU2} na desce snímače polohy, které by pro tento účel mohly sloužit i tehdy, když by k sensorovému konektoru bylo připojené jiné zařízení s I²C komunikací. Hodnota těchto odporů byla stanovena podle hodnot ve schématu sensorových desek, tj. 1500 Ω . Pokud by se stalo, že by odpory byly osazeny na obou deskách, nedošlo by přesto k přetížení budičů, neboť při $V_{CC} = 3,3$ V a paralelní kombinaci např. $R_{PU} = (R_{PU1} \parallel R_{PU3}) = 750 \Omega$ v sérii s odporem přechodu budiče Drain-Source v sepnutém stavu, kdy $R_{DS} = 50 \Omega$, tedy $I_{DS} = V_{CC} / (R_{PU} + R_{DS}) = 3,3 / 800 = 4,1$ mA.

Procesory Blackfin řady BF-53x v sobě nemají hardwarovou implementaci rozhraní I²C, proto muselo být I²C realizováno softwarově na GPIO linkách (PF9 a PF10), viz kapitola 3.1.4. Více informací o univerzálních vstupně/výstupních linkách GPIO procesorů BF-53x přináší kapitola 2.3.4.2.

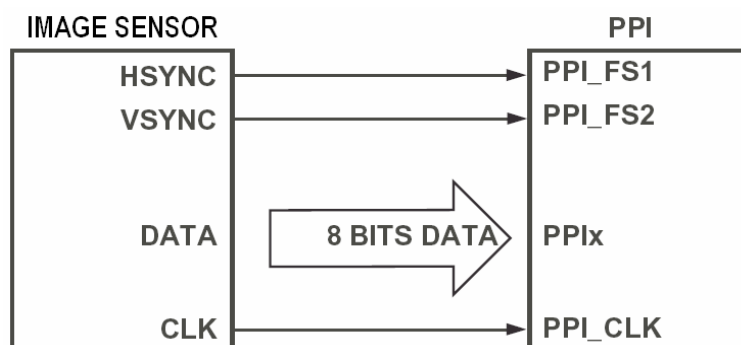
2.3.2.2 Řídicí signály

Z těchto signálů je na konektoru CMOS senzoru propojen pouze RESET# a TRIGGER (SNAPSHOT). **Signál RESET#** je veden na výstup resetovacího obvodu ADM708TAR a je společný i pro procesor, takže v okamžiku resetu procesoru dochází vždy i k resetu obrazového senzoru, viz kapitola 2.3.3.2. **Signál TRIGGER** v módu Snapshot slouží k pořízení jednoho snímku. Po přepnutí do tohoto módu nejsou generovány signály FRAME_VALID a LINE_VALID, pouze výstup hodin PIXCLK. Jakmile dojde ke změně do log. 1 na pinu TRIGGER, je zachycen jeden snímek a obrazová data se začnou přenášet na výstup. Zbývající signály nebylo z funkčního hlediska nutné zapojovat, prioritou byla co největší ušetření GPIO linek. **Signál OE#** je trvale na hodnotě log. 0, v opečném případě by se využíval k přepínání signálů D0 - D9, FRAME_VALID, LINE_VALID, PIXCLK a STROBE do třetího stavu. **Signál STANDBY** vypne analogové obvody, čímž se aktivuje režim nízké spotřeby.

2.3.2.3 Signály pro přenos obrazových dat

Paralelní přenos obrazových dat do procesoru zajišťuje tzv. PPI (*Parallel Peripheral Interface*). Toto rozhraní umožňuje poloduplexní obousměrný přenos dat, je vhodné pro připojení nejrůznější periferní zařízení, A/D a D/A převodníků, video enkodérů/dekodérů a podporuje operace standardu ITU-656. Rozhraní sestává z 16 datových bitů PPI0 – PPI15, hodinového vstupu PPI_CLK a tří signálů pro snímkovou synchronizaci, tj. PPI_FS1–3. Rozhraní PPI lze provozovat ve třech režimech, tzv. *General-Purpose Modes* [22]:

- *vstupní mód* – signály snímkové synchronizace i paralelní data vstupují do PPI,
- *Frame Capture mód* – signály snímkové synchronizace jsou výstupy, ale paralelní data vstupy PPI,
- *výstupní mód* – signály snímkové synchronizace i paralelní data jsou výstupy z PPI.



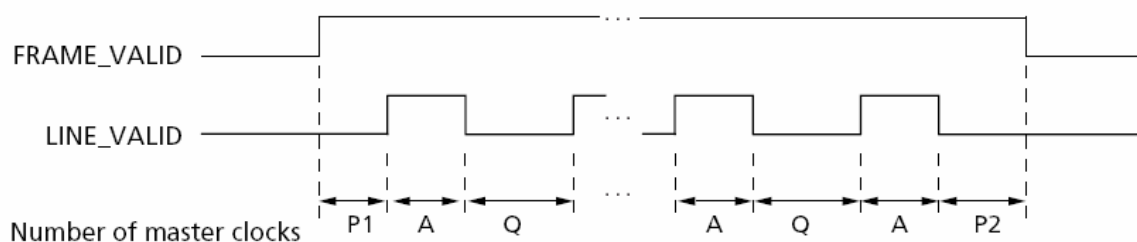
Obr. 2.18 Konfigurace PPI rozhraní pro přenos obrazových dat z CMOS senzoru¹⁵

¹⁵ Převzato a upraveno z hardwarového manuálu ADSP-BF533 [24].

V zapojení snímače polohy je pro přenos obrazových dat z CMOS senzoru využíván výhradně *vstupní mód* v konfiguraci patrné na obr. 2.18. Obrazová data jsou přenášena osmi PPI linkami PPI0 - PPI7. V této šířce datového slova je zaručena největší propustnost dat rozhraním, neboť dva 8-bitové vzorky mohou být spojeny do jednoho 16-bitového slova, čímž se docílí většího využití DMA kanálu, který je schopen do vnitřní paměti SRAM přenést právě 16 bitů s každým taktem systémových hodin SCLK. Z hlediska obrazového senzoru jsou přenášeny signály D2 - D9, ačkoliv senzor poskytuje 10 bitů jasové informace na každý pixel. Je to z toho důvodu, že pro metody zpracování obrazové informace za účelem určení polohy objektu není nutné mít k dispozici tolik kvantizačních úrovní na rozdíl od rychlého přesunu dat do paměti. Navíc by došlo k obsazení dalších dvou GPIO linek, které jsou se zmiňovanými vstupy PPI8,9 sdílené.

2.3.2.4 Synchronizační signály

Tyto signály mají význam pro řízení PPI, tzn. spouštějí nebo zastavují přenos dat ze senzoru, ačkoliv hodinový signál PIXCLK je na vstupu přítomný neustále. **Signál LINE_VALID** je aktivní přesně po dobu vyčítání jednoho řádku, **signál FRAME_VALID** pak po dobu jednoho snímku. Jinými slovy, signály slouží pro řádkovou a snímkovou synchronizaci, tj. např. pro odpočítání n řádků od začátku snímku, přeskokování m řádků apod. Zároveň platí, že oba tyto signály musí být aktivní, aby byla přenášena obrazová data rozhraním PPI. Pro názornost jsou průběhy těchto signálů na obr. 2.19.



Obr. 2.19 Průběhy signálů snímkové a řádkové synchronizace senzorů Micron¹⁶

Znaky A, P1, P2 a Q vyjadřují délku příslušného intervalu v počtu period řídicího hodinového signálu MCLK. **A** je doba aktivního výstupu dat, **P1** je zatemněná oblast na začátku snímku, **P2** je zatemněná oblast na konci snímku a **Q** je zatemněná oblast na konci každého řádku s výjimkou posledního. Platí, že $Q = P1 + P2$. Velikosti intervalů A a Q lze programově měnit zápisem do příslušných registrů obrazového senzoru. Toho se také využívá pro zpomalení vyčítání obrazových dat za účelem kontinuálního přenosu přes USB řadič do PC bez použití

¹⁶ Převzato z katalogového listu MT9M001 [5]. Průběhy a označení těchto signálů je stejné i u MT9V032

externí paměti SDRAM, viz kapitola 3.1.2.2. **Signál STROBE** je výstupní a slouží k indikaci doby integrace všech řádků. Ve snímači polohy však využíván není.

2.3.2.5 Hodinové signály

Obrazové senzory vyžadují pro svou funkci připojení pravoúhlého hodinového signálu, ze kterého jsou řízeny veškeré procesy. Senzor MT9M001 toleruje kmitočty v rozsahu 1 – 48 MHz. **Signál CLKIN** je hodinovým vstupem senzoru a je přiváděn přes tvarovač z krystalového oscilátoru, kterým je generován hodinový signál pro procesor. Uvnitř procesoru je jednotka fázového závěsu PLL, kterou lze násobit kmitočet oscilátoru v širokém rozsahu, což je výhodné pro optimální nastavení SCLK (*System Clock*) a CCLK (*Core Clock*). U obou výše zmíněných obrazových sensorů Micron však jakékoliv násobení nebo dělení kmitočtu CLKIN chybí, proto je vždy kmitočet výstupního signálu PIXCLK stejný. **Signál PIXCLK** je připojen na signál PPI_CLK a slouží ke vzorkování dat na sběrnici PPI. Výstup hodin ze senzoru je možné potlačit signálem OE# (*Output Enable*). Při návrhu snímače polohy tedy byla hodnota krystalu volena s ohledem na obrazový senzor, viz kapitola 3.1.2.2. Obrazové senzory mají odlišnou aktivní hranu signálu PIXCLK, což je nutné zohlednit při nastavování konfiguračních registrů PPI procesoru, jinak bude obraz ze senzoru obsahovat chyby. Více je uvedeno v kapitole 3.1.3.

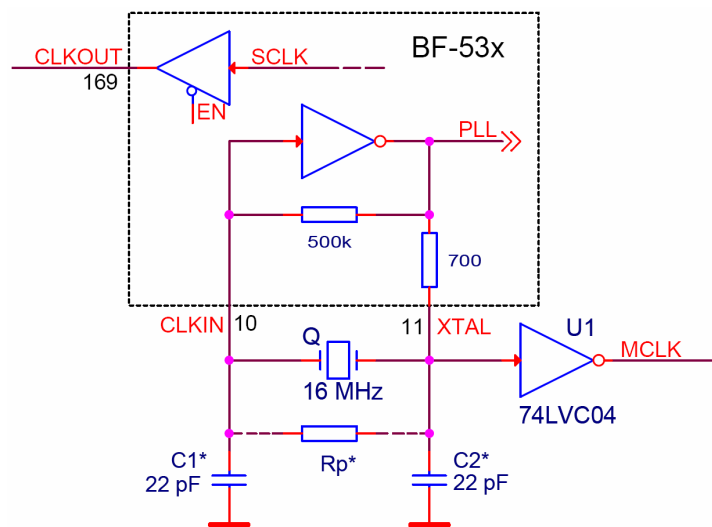
2.3.2.6 Rozvod napájecího napětí

Napájecí napětí 3,3 V je na konektor přiváděno z hlavního stabilizujícího měniče, který se také nachází v zapojení snímače polohy. Na sensorové desce se toto napětí odděluje přes tlumivku na napětí jádra VDD a napětí pro analogovou část VAA, jimž přísluší země DGND a AGND. K analogové části je rovněž připojeno VAAPIX, což je napájení pro matici pixelů. Při návrhu stabilizujícího měniče (kapitola 2.3.5) bylo nutné počítat s rezervou pro mezní proudy obrazových sensorů, které činí $I_{max} = 144$ mA pro MT9M001 a $I_{max} = 200$ mA pro MT9V032.

2.3.3 Periferní obvody procesoru

2.3.3.1 Hodinový systém

Procesory BF-53x mají vnitřní obvod, k němuž stačí připojit externí krystal Q a kondenzátory C_1 , C_2 , viz obr. 2.20. Hodnota těchto kondenzátorů byla volena experimentálně, neboť záleží na mnoha faktorech, které popisuje [22]. Zpětnovazební odpor R_P není třeba osazovat, protože je již obsažen ve vnitřním zapojení procesoru. Z výstupu XTAL je sinusový průběh veden do tvarovacího hradla U_1 , které má za úkol dodávat obdélníkový hodinový signál MCLK pro CMOS senzor. Nutno připomenout, že v používaných obrazových senzorech *Micron*, není žádná dělička hodinového signálu, proto jsou obrazová data vysouvána podle MCLK. Zvolená hodnota 16 MHz u krystalu Q je kompromisem dvou požadavků: zpomalení toku obrazových dat za účelem spojitého přesunu snímku do USB řadiče a co největší počet snímků za sekundu pro rychlé měření polohy světelné stopy. S kmitočtem 16 MHz byl u MT9M001 dosažen přenos obrazu v plném rozlišení 1280×1024 pixelů rychlostí 8 MB/s, tj. cca 6 snímků/s a minimální doba vyčtení jednoho snímku ze senzoru v plném rozlišení byla 100 ms.

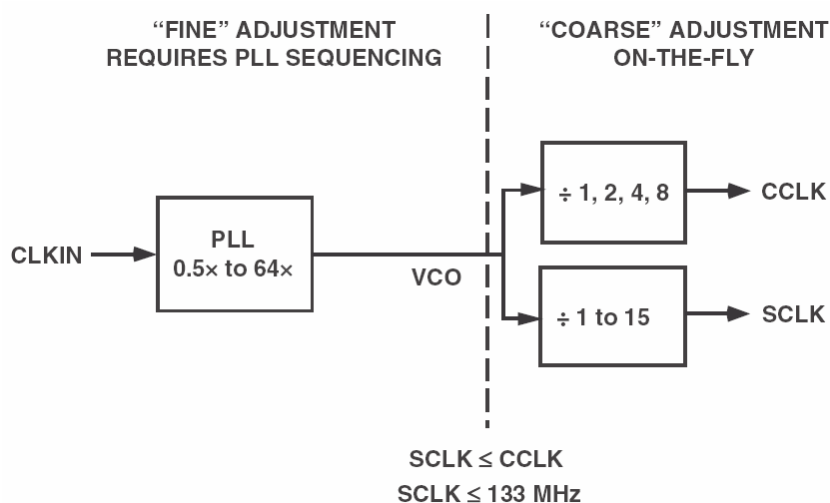


Obr. 2.20 Obvod pro generování hodinových signálů BF-53x¹⁷

Možnosti nastavení hodinových signálů uvnitř procesoru SCLK (hodiny periferií) a CCLK (hodiny jádra) nabízí jednotka PLL, viz obr. 2.21. V té lze signál CLKIN násobit faktorem 0,5 až 64, ale nesmí být překročen mezní kmitočet VCO (*Voltage Controlled Oscillator*). Následuje systém děliček, kterým se nastaví kmitočty SCLK a CCLK na požadovanou hodnotu. Platí, že $f_{VCOMAX} = f_{CCLK}$, přičemž pro procesor BF-533 v pouzdře LQFP-176 je $f_{CCLKMAX} = 400$ MHz a $f_{SCLKMAX} = 133$ MHz. Teoretické zvýšení těchto kmitočtů přetaktováním sice možné je (za

¹⁷ Převzato a upraveno z katalogového listu ADSP-BF531/ADSP-BF532/ADSP-BF533 [22].

dodržení jistých podmínek), ovšem během vývoje snímače polohy žádné experimenty tohoto typu prováděny nebyly.



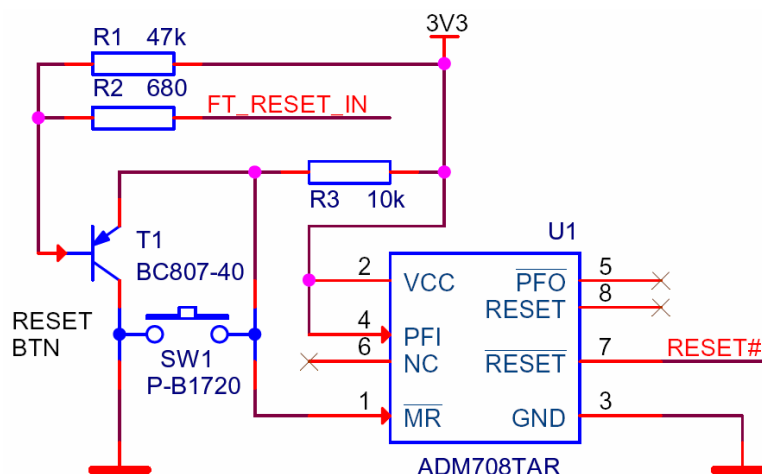
Obr. 2.21 PLL systém procesoru BF-53x¹⁸

Procesory BF-53x mají jeden pin vyhrazen jako výstup hodinového signálu CLKOUT, viz obr. 2.20. Tento signál je shodný s SCLK, lze jej tedy využít k taktování nejrůznějších externích obvodů, např. paměti SDRAM. Pro účely generování hodinového signálu pro obrazový senzor však vhodný není, neboť časování ostatních periférií by bylo přímo závislé na jeho okamžitém režimu (rychlosti vyčítání obrazových dat).

2.3.3.2 Resetovací obvod

Po přivedení napájecího napětí na procesor je jeho pin $\overline{\text{RESET}}$ držen v aktivní úrovni ještě po dobu $3500 \times t_{CLKIN}$ ($t_{CLKIN} = 62,5$ ns, tj. perioda hodinového signálu) od ustálení napájecích napětí V_{DDINT} a V_{DDEXT} , poté následuje bootování procesoru ze sériové paměti EEPROM (platí pro mód BMODE = 11, viz kapitola 2.3.3.3). Po výše uvedené době je možné procesor BF-53x resetovat přivedením log. 0 na jeho pin $\overline{\text{RESET}}$ minimálně po dobu $11 \times t_{CLKIN}$ [22]. Ke generování korektního resetovacího pulzu je v zapojení snímače polohy použit obvod ADM708TAR, viz obr. 2.22.

¹⁸ Převzato a upraveno z katalogového listu ADSP-BF531/ADSP-BF532/ADSP-BF533 [22].



Obr. 2.22 Schéma zapojení resetovacího obvodu pro BF-53x a obrazový senzor

Tento obvod má navzájem inverzní výstupní signály $\overline{\text{RESET}}$ a RESET , přičemž je schopný generovat resetovací pulz právě tehdy, když dojde:

- k zapínání a vypínání napájecího zdroje a při zakolísání napájecího napětí,
- k detekování napětí menšího než 1,25 V na vstupu PFI (*Power Fail Input*),
- k aktivní log. úrovni na vstupu $\overline{\text{MR}}$ (*Manual Reset*) s tolerovaným odskokem kontaktů.

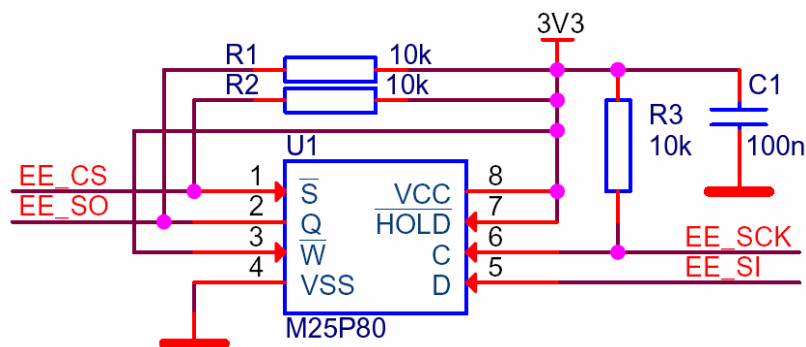
Podrobné parametry všech průběhů a úrovní jsou uvedeny v katalogovém listu obvodu [25]. Na vstup $\overline{\text{MR}}$ je připojeno tlačítko, aby bylo možné procesor kdykoliv resetovat, což je praktické při použití desky snímače polohy i pro jiné účely než zpracovávání obrazové informace z CMOS senzoru. K tlačítku je zároveň připojen paralelně tranzistor T_1 , který vykonává stejnou funkci, pokud je na jeho bázi přivedena přes odpor R_2 úroveň log. 0, což zprostředkovává signál FT_RESET_IN z GPIO0 linky USB řadiče FT2232H. Pro připomenutí, tento signál je generován z FT2232H po celou dobu programování paměti EEPROM, aby byl procesor BF-53x ve stavu resetu, aby měl budiče sběrnice SPI ve třetím stavu a aby po naprogramování této paměti provedl nabootování nově uloženého loaderu.

2.3.3.3 Bootování procesoru a paměť EEPROM

Procesory Blackfin řady BF-53x nemají paměť programu umístěnou na čipu jako některé mikrokontroléry, ale mají v sobě zabudovaný bootovací systém, který musí vykonávaný program nahrát z externí paměti. Výhoda této koncepce spočívá v tom, že velikost programu není omezena kapacitou vnitřní paměti, a je tak možné procesorem zpracovávat i větší množství programových dat. Režimy bootování BF-53x jsou určeny kombinací logických úrovní na vstupních pinech BMODE0,1 a jsou následující [26]:

- BMODE 00: bootování není povoleno, program je vykonáván přímo z externí 16-bitové paměti,
- BMODE 01: bootování z 8/16-bitové flash/PROM paměti,
- BMODE 10: bootování z hostitelského řadiče přes SPI v režimu slave,
- BMODE 11: bootování z 8/16/24-bity adresovatelné SPI paměti v režimu master.

Pro snímač polohy je použitelný pouze poslední způsob bootování, tj. nahrání programu do vnitřní paměti z externí sériové EEPROM, protože paralelní flash paměti ani další procesor by nebyly z hlediska rozměrů DPS a minimální obvodové konfigurace žádoucí.



Obr. 2.23 Připojení 1 MB paměti EEPROM M25P80 na sběrnici SPI

Na obr. 2.23 je znázorněno připojení paměti M25P80 na signály SPI sběrnice. Tato paměť byla již zmíněna v kapitole 2.2.3.3 a její podrobný popis poskytuje katalogový list [19]. Způsob propojení SPI sběrnice mezi USB řadičem, procesorem a EEPROM pamětí je pro připomenutí na obr. 2.10. Pro správnou funkčnost a tvarování signálů na sběrnici byly do obvodu přidány tzv. *pull-up* odpory R_1 , R_2 a R_3 . Jejich hodnota byla volena experimentálně tak, aby byla bezchybná komunikace i s ochrannými sériovými odpory $470\ \Omega$ u FT2232H v zapojení dle obr. 2.10. zaručena. Paměť M25P80 má kromě SPI signálů \bar{S} (*Chip Select*), C (*Clock*), D (*Data In*), Q (*Data Out*) ještě vstupní piny $\overline{\text{HOLD}}$ (*Hold*) a \bar{W} (*Write Protect*). Pokud je signál $\overline{\text{HOLD}}$ uveden do aktivní log. úrovně a na výběrovém pinu \bar{S} se objeví spádová hrana, dojde k přerušení veškeré sériové komunikace, aktivní zůstává pouze výběrový vodič \bar{S} . Ochrana zápisu a mazání je aktivní při úrovni log. 0 na pinu \bar{W} , přičemž se vztahuje pouze na určité bloky paměti, které se musí označit příslušnými bity BP0, BP1 a BP2 (*Block Protect*) ve stavovém registru. V obvodu snímače polohy nejsou funkce *Hold* ani *Write Protect* využívány, proto jsou na oba vstupní piny přivedeny úrovně log. 1.

Plošný spoj pod pamětí EEPROM byl navržen tak, aby bylo dobře připájitelné vývodové pouzdro M25P80 (SO8) a zároveň byl kompatibilní i pro bezvývodové pouzdro paměti M25P32 (MLP8) s plošným pinem na spodní části pouzdra, který je vnitřně propojen s V_{SS} . Paměť

M25P32 [27] má 4 × větší kapacitu než M25P80 (4 MB) a je uvažováno o jejím použití v budoucích aplikacích s deskou snímače polohy. Tato koncepce procesoru s velkou bootovací pamětí předpokládá, že obsah paměti bude rozdělen na sektory, do nichž se nahrají různé části programu mající odlišné funkcionality. Například v prvním sektoru bude inicializační (hlavní) program, v druhém program pro hledání těžiště světelné stopy, ve třetím program pro hledání hran, ve čtvrtém program pro měření plochy a podobně v dalších sektorech. Protože by se všechny naprogramované funkce nevešly do vnitřní paměti procesoru, nebo by zabíraly zbytečně mnoho místa, byl by do ní po resetu nahrán pouze inicializační program a další části by byly nahrávány podle aktuálního aplikačního využití. Více informací o tomto způsobu bootování přináší dokument [26], kapitola *Multi-Application Management* na str. 7.

2.3.4 Uživatelská rozhraní snímače polohy

2.3.4.1 UART, časovač, sériová linka RS–232/485

Procesory BF–53x jsou vybaveny obvody pro plně duplexní asynchronní sériovou komunikaci známou pod označením UART. Mezi základní parametry přenosu patří: počet datových bitů (5 - 8), počet stop bitů (1 nebo 2), parita (sudá, lichá, žádná), rychlost přenosu (až 8,3 MBaud). Sériová linka zahrnuje přijímací Rx a vysílací Tx kanál s datovými registry, pro něž je čtení nebo zapisování řízeno procesorem a obsluha události na přijímací straně prováděna např. přerušením nebo s minimální účastí procesoru, tj. pomocí DMA kanálů. Piny na procesoru RX a TX jsou přivedeny přes ochranné odpory 470 Ω na 4–pinový konektor J₂ s rozložením podle tab. 2.7.

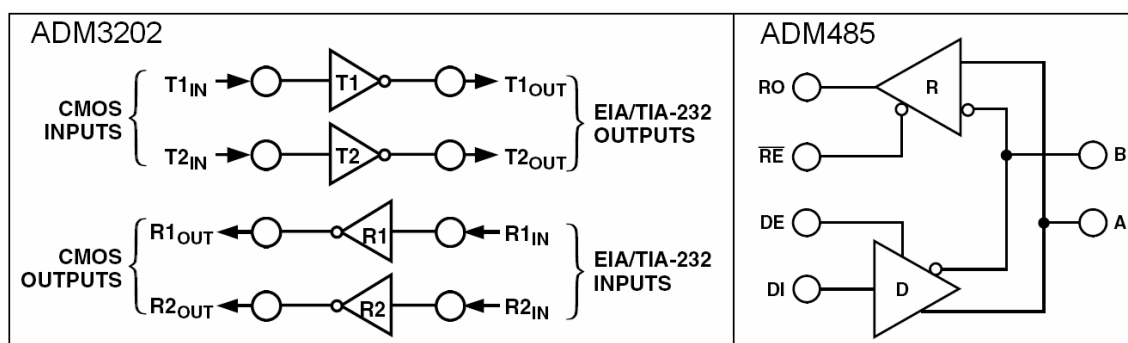
číslo pinu	1	2	3	4
signál	UART_TX	TMR0	UART_RX	GND

Tab. 2.7 Rozložení pinů konektoru J₂–UART na plošném spoji

Na pinu 2 konektoru J₂ je vyveden signál TMR0, který přímo nesouvisí se sériovou komunikací. Pin s tímto označením se však nachází v bezprostřední blízkosti signálů RX a TX, takže bylo výhodné jej vyvést také. TMR0 je vstupně/výstupní pin prvního ze tří identických časovačů, které procesory BF–53x obsahují, má tedy využití v mnoha dalších zapojeních. Časovače lze provozovat ve vstupních režimech WDT_CAP (*Pulse Width Count and Capture*), EXT_CLK (*External Event*) a výstupním režimu PWM_OUT (*Pulse Width Modulation*). Bližší popis časovačů a jejich režimů obsahuje [24].

Pro připojení snímače polohy k PC přes sériovou linku RS–232 musí být použit převodník napět'ových úrovní, např. ADM3202 [28]. Tento obvod není v zapojení z důvodu snahy o úsporu místa na DPS integrován, ale je možné jej připojit na dalším rozšiřujícím modulu. Obvod

ADM3202 je dvoukanálový, viz obr. 2.24, případné využití druhého kanálu by se nabízelo vytvořením dalšího UARTu softwarově na některých vyvedených linkách GPIO, tj. PFX.



Obr. 2.24 Funkční bloková schémata budičů RS-232/485¹⁹

Nevýhodou RS-232 je tzv. *point-to-point* komunikace, což prakticky znamená, že k jednomu PC by mohl být připojen pouze jeden snímač polohy, navíc délka sběrnice dosahuje pro vyšší přenosové rychlosti pouze několik jednotek metrů. Praktičtější je rozhraní RS-485, které podporuje *multipoint* komunikaci Master/Slave. Pro budoucí využití snímače polohy v sensorové síti by každý snímač mohl být připojen na sběrnici jako Slave a řídicí počítač jako Master. Sběrnici tvoří kroucený pár vodičů a její délka může dosahovat až 1200 m. Bez opakovací lze na sběrnici připojit až 32 zařízení, což značně přesahuje počet portů v běžných USB Hubech, takže ani USB rozhraní snímačů polohy s několikametrovými propojovacími kabely pro multisenzorovou síť vhodná nejsou. Sběrnice RS-485 je sice poloduplexní, ale rychlost komunikace může dosahovat až 10 Mb/s, je tedy dostatečná i pro přenos vícenásobné informace o poloze. U obvodu ADM485 jsou v blokovém schématu na obr. 2.24 znázorněny povolovací vstupy DE (*Driver Enable*) a \overline{RE} (*Receiver Enable*), jimiž se budič a přijímač připojují ke sběrnici a musí být řízeny z volných linek PFX.

2.3.4.2 Rozhraní GPIO (Programmable Flags)

Pro nejrůznější aplikační uplatnění jsou procesory BF-53x vybaveny 16-bitovým portem F, což jsou navzájem nezávislé piny PFX (*Programmable Flags*), a každý pin je možné v příslušných registrech nastavit buď jako vstupní nebo jako výstupní. U každého vstupního pinu je nastavitelné samostatné přerušení s libovolnou prioritou. Přerušení lze vyvolat úrovní, náběžnou hranou, spádovou hranou nebo oběma hranami [24]. Snahou vývoje snímače polohy bylo i vyvedení co největšího počtu PF na konektory, aby byly dostupné pro další použití. Všechny GPIO linky však sdílí příslušné piny procesoru i pro ostatní funkcionality (rozhraní

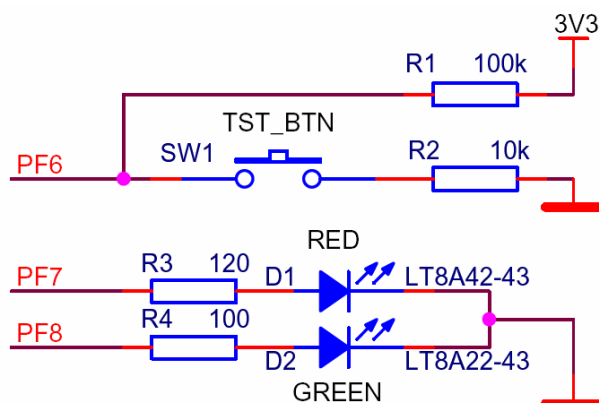
¹⁹ Převzato a upraveno z katalogových listů ADM3202/ADM3222/ADM1385 [28] a ADM485 [29].

PPI, výběrový vodič SPI sběrnice, časovač), takže po obsazení některých linek signály obrazového senzoru a řadiče FT2232H zbylo pouze 5 flagů, které byly vyvedeny na konektor J₄, viz tab. 2.8. V tabulce jsou uvedeny zároveň alternativní funkcionality, které má procesor na příslušném pinu.

číslo pinu	signál	další funkcionality	sdílení
1	PF7	SPISEL7 / PPI12	LED_R
2	PF8	PPI11	LED_G
3	PF6	SPISEL6 / PPI13	BTN
4	3V3	–	–
5	PF3	SPISEL3 / PPI_FS3	–
6	NC	–	–
7	PF1	SPISEL7 / TACLK	–
8	GND	–	–

Tab. 2.8 Propojení pinů na konektoru J₄ s piny GPIO

V některých případech je užitečné mít na desce vyvedeny nějaké vstupní a výstupní prvky pro elementární ovládací operace, což bylo u snímače polohy vyřešeno testovacím tlačítkem a dvěma indikačními LED, viz obr. 2.25. Tlačítko je připojeno na flag PF6, kde při stisknutí uvede vstupní pin do úrovně log. 0, při uvolnění zůstává v log. 1. Pokud by byl tento pin nastaven jako výstupní do log. 1 a došlo by ke stisknutí, nedojde díky ochrannému odporu R₂ k proudovému přetížení.



Obr. 2.25 Připojení testovacího tlačítka a indikačních LED na GPIO linky

Červená a zelená LED jsou buzeny přes předřadné odpory R₃ a R₄ proti zemi úrovní log. 1. Hodnoty odporů byly stanoveny tak, aby LED vykazovaly stejnou svítivost a proudová zátěž každého pinu nepřekročila 10 mA. U těchto dvou flagů je však nutné uvážit kvůli nelineární zátěži zkreslení obdélníkového průběhu, které by mohlo ovlivnit externí zařízení připojené na dotyčné linky. Proto je vhodné zejména při vyšších kmitočtech indikační LED vyřadit odletováním nebo změnou odporů R₃ a R₄.

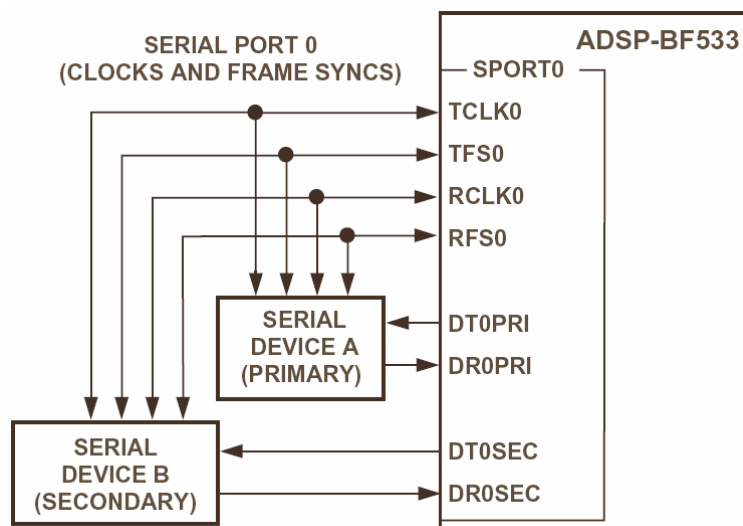
2.3.4.3 Sériový port SPORT

Mezi další rysy DSP procesorů Blackfin BF-53x patří implementace dvou sériových portů SPORT0,1. Na plošném spoji byl z důvodu nedostatku místa vyveden na konektor J₃ pouze SPORT0, jehož rozložení pinů s popisem signálů znázorňuje tab. 2.9. Rozhraní SPORT je určeno výhradně pro synchronní sériový přenos dat s možností plně duplexní komunikace.

číslo pinu	signál	popis
1	DR0SEC	Receive Data Secondary
2	TSCLK0	Transmit Clock
3	DR0PRI	Receive Data Primary
4	3V3	Power Supply 3,3V
5	RSCLK0	Receive Clock
6	RFS0	Receive Frame Sync
7	TFS0	Transmit Frame Sync
8	DT0PRI	Transmit Data Primary
9	DT0SEC	Transmit Data Secondary
10	GND	Ground

Tab. 2.9 Rozložení signálů SPORT0 na pinech konektoru J₃

Rozhraní SPORT zahrnuje jednu skupinu pinů pro vysílání (primární/sekundární data, hodiny, rámcovou synchronizaci) a druhou skupinu pro příjem s totožným označením. Na obr. 2.26 je znázorněn příklad možného připojení dvou sériových synchronních zařízení na SPORT0.



Obr. 2.26 Připojení synchronních zařízení A a B k sériovému portu SPORT0²⁰

K dalším vlastnostem SPORTu patří přenosová rychlost 66 nebo 133 Mbit/s (při současném využití sekundárního kanálu Rx/Tx), nastavitelnost slov o délce 3–32 bitů,

²⁰ Převzato a upraveno z hardwarového manuálu BF-533 [24].

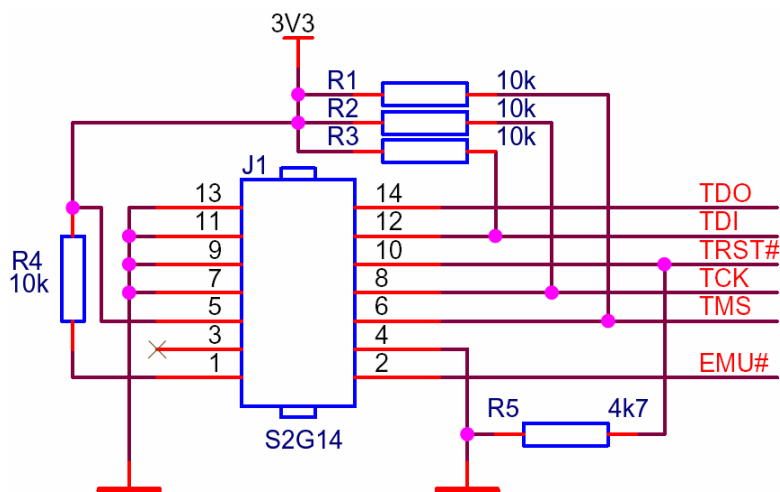
programovatelná vnitřní/vnější hodinová a rámcová synchronizace, podpora vícekanálových rozhraní TDM (*Time Divison Multiplex*) pro síťovou komunikaci nebo podpora protokolu I²S. Sériový port lze uplatnit pro připojení nejrůznějších audiokodeků, dalších procesorů, programovatelných polí FPGA apod.

2.3.4.4 Ladicí rozhraní JTAG

JTAG je velmi praktickým nástrojem při vývoji software pro procesory BF-53x, neboť umožňuje plnou emulaci procesoru na zapájené desce (tzv. *in-circuit*), což znamená, že je možné provádět krokování programu, sledovat a modifikovat proměnné, obsah paměti a registrů na reálném hardwaru. Na desce snímače polohy proto byly vyvedeny signály JTAGu na 14-pinový konektor podle standardu IEEE 1149.1, viz tab. 2.10. Signály TDI, TCK a TMS jsou doplněny *pull-up* odpory, signál $\overline{\text{TRST}}$ odporem *pull-down*, přičemž schéma na obr. 2.27 je převzato z ověřeného funkčního zapojení podle vývojové desky EZ-KIT Lite.

číslo pinu	signál	popis
1	3V3_10K	3,3 V With 10 k Ω
2	EMU#	Emulation Flag
3	NC	Not Connected
4, 7, 9, 11, 13	GND	Ground
5	3V3	Power Supply 3,3V
6	TMS	Test Mode Select
8	TCK	Test Clock
10	TRST#	Test Reset
12	TDI	Test Data Input
14	TDO	Test Data Out

Tab. 2.10 Rozložení a popis pinů na konektoru JTAG

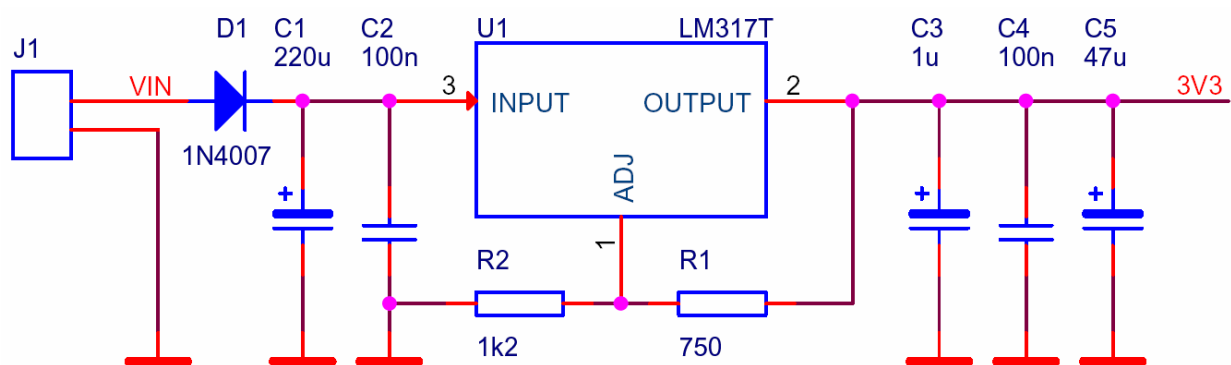


Obr. 2.27 Schéma zapojení JTAG konektoru

Testování funkčnosti JTAGu bylo prováděno na osazené desce snímače polohy pomocí emulátoru ICE-100B od Analog Devices, který byl katedrou měření nově zakoupen. Tento emulátor umožňuje vykonávat veškeré ladicí operace, které byly uvedeny na začátku kapitoly. Před integrací do vývojového prostředí VisualDSP++ bylo nutné nejprve nainstalovat patřičný update, poté proběhla instalace vlastního hardware emulátoru již standardním způsobem. Doporučený postup celé instalace je podrobně popsán v dokumentu [30].

2.3.5 Napájecí systém

Pro obvody snímače polohy bylo nutné zajistit dvě úrovně napájecího napětí – 3,3 V a 1,2 V. Napájecí napětí ze zdroje 3,3 V vyžadoval procesor BF-53x (V_{DDEXT}), řadič FT2232H, obrazový senzor, obě paměti EEPROM, resetovací obvod ADM708TAR i invertor 74LVC04. Napájení 1,2 V pak mělo být určeno pouze pro jádro procesoru. Bylo zřejmé, že proudové zatížení u příslušných napěťových hladin bude mít široký rozsah a bude záležet na okamžitém výpočetním výkonu procesoru, USB řadiče a dalších periférií, u nichž ani nemohl být odběr předem známý. Z toho důvodu bylo třeba počítat s jistou proudovou rezervou. Ve snaze o nepřekročení požadovaných rozměrů DPS snímače polohy (60×60 mm), nemohl být v konečném zapojení integrován žádný složitý spínaný měnič s indukčností. Zároveň byla zamítnuta myšlenka, že by se na vstupní svorky přivádělo přímo napětí 3,3 V (byť stabilizované), z něž by se odvozovalo napětí 1,2 V pro jádro. Výsledná koncepce pak byla realizována ze dvou kaskádně zapojených lineárních stabilizátorů LM317T, kdy návaznost napěťových hladin byla $V_{IN} \rightarrow 3,3 \text{ V} \rightarrow 1,2 \text{ V}$.



Obr. 2.28 Schéma zapojení lineárního měniče pro napětí 3,3 V s LM317T

Od původního záměru použít stabilizátory v SMD pouzdře muselo být upuštěno, neboť plochy potřebné k odvodu ztrátové tepelné energie plošný spoj omezené velikosti nenabízel. Oba dva prvky jsou tedy v pouzdře TO-220 s chladičskou plochou u okraje desky pro případnou montáž na chladič nebo stěnu montážní krabice. K parametrům chladiče je třeba přistupovat s ohledem

na velikost vstupního napětí a proudový odběr 3,3 V větve, z níž je odebírán i proud pro jádro procesoru přes 1,2 V stabilizátor. Vstupní napětí smí dosahovat až 40 V, ovšem při této úrovni je již ztrátový výkon značný a provoz neekonomický, proto je optimální napájet snímač polohy z 6 V zdroje. Mezní proud LM317T činí 1,5 A, takže pro případné napájení externích 3,3 V zařízení z některých konektorů je proudová rezerva dostatečná. Stabilizátor 1,2 V při provozu v pokojových teplotách chlazen být nemusí, protože vzniklé teplo odvede samotné pouzdro součástky.

Na obr. 2.28 je uvedeno zapojení stabilizátoru pro 3,3 V se zpětnovazebním odporovým děličem R_1 a R_2 , který určuje velikost výstupního napětí podle vztahu

$$V_{OUT} = 1,25 \cdot \left(1 + \frac{R_2}{R_1} \right) + I_{ADJ} \cdot R_2, \quad (2.2)$$

kde I_{ADJ} je proud zpětnovazebního pinu ADJ a činí 50 až 100 μ A. Stabilizátor 1,2 V je svým vstupem připojen na 3,3 V a od zapojení na obr. 2.28 se liší tak, že odpor R_1 není zapojen a R_2 má hodnotu 0 Ω , resp. pin ADJ je připojen přímo na zem. Po dosazení do rovnice (2.2) v tomto případě vyjde, že $V_{OUT} = 1,25$ V, což je s uvažováním úbytků na spojích při zatížení a tolerančním intervalu napájení jádra V_{DDINT} přípustné.

Napájecí systém tvoří kromě měničů napětí také rozsáhlá síť blokovacích keramických kondenzátorů, které jsou osazeny v těsné blízkosti jednotlivých napájecích pinů integrovaných obvodů. K vyhlazení zvlnění jsou na některých pinech doplněny filtrační tantalové kondenzátory. Proti přepólování celého obvodu snímače polohy slouží ochranná dioda D_1 , ovšem za cenu úbytku cca 0,6 V na PN přechodu v propustném směru. Pro indikaci připojení napájecího napětí V_{IN} slouží LED, která je napájena přes sériový odpor z větve 3,3 V. Kompletní schéma zapojení snímače polohy se skutečnými referencemi a hodnotami součástek obsahuje příloha E.

3 Softwarová část vývoje

3.1 Přenos obrazu ze senzoru do PC

Před samotným popisem způsobu přenosu obrazu z CMOS senzoru, připojeného ke snímači polohy přes USB řadič do PC, je vhodné provést stručný rozbor tohoto úkolu z hlediska všech zúčastněných hardwarových částí. V první řadě je nutné uvést programovou konfiguraci některých registrů procesoru potřebných k inicializaci GPIO linek (definování vstupních a výstupních pinů) a PLL jednotky. Další krok představuje popis rozhraní BF-53x s FT2232H a odesílání velkého objemu dat do PC. Následně lze přistoupit ke konfiguraci PPI rozhraní procesoru pro připojení obrazového senzoru a nastavení DMA kanálu. Před samotným přenosem obrazových dat musí být provedeno také nastavení registrů CMOS senzoru, což vyžaduje softwarovou implementaci I²C sběrnice pomocí programovatelných flagů. Nakonec je zapotřebí, aby byl rychlý tok dat na straně PC neprodleně vyčítán z USB zařízení a nedocházelo ke ztrátě dat na straně procesoru, který nemá dostatek paměti na uchování více než několika řádků snímku. Vyčtená data jsou v PC ukládána do paměti nebo do souboru, a poté vykreslena na monitor.

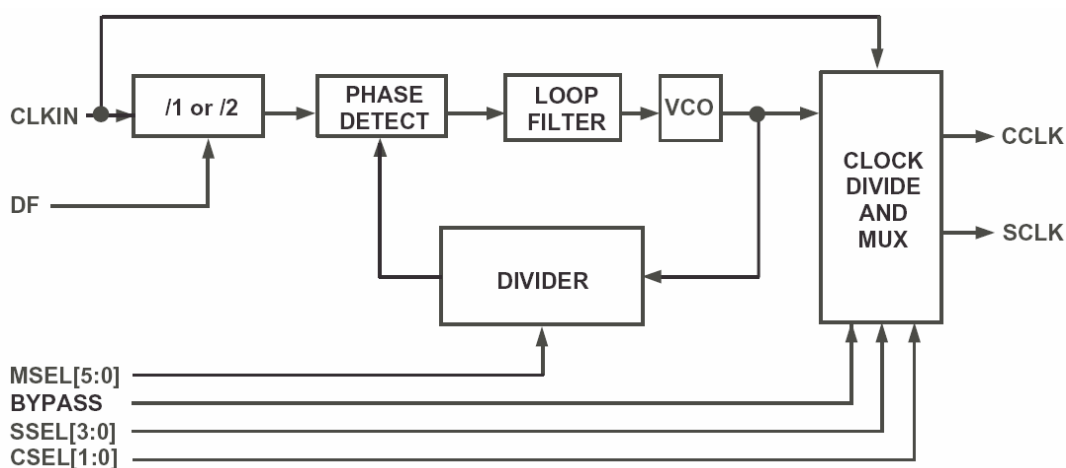
3.1.1 Nastavení GPIO linek a konfigurace jednotky PLL

Po připojení napájecího napětí procesoru nebo po jeho resetu musí dojít k inicializaci programovatelných flagů, tzn. musí být definováno, které GPIO linky budou vstupní a které výstupní. Pro nastavení směru linek slouží registr **FIO_DIR** (*Flag Direction*), kdy zápisem log. 1 na některý bit v šesnásobitovém registru dojde k nastavení příslušného flagu jako výstupu a log. 0 jako vstupu. Po resetu jsou všechny bity v log. 0, tj. všechny linky vstupní. Aby procesor začal vyhodnocovat (číst) stavy na vstupních linkách, musí být připojen vstupní buffer na příslušném flagu, což se provede nastavením log. 1 v registru **FIO_INEN** (*Flag Input Enable*). Příklad, jak nastavit flagy PF9, PF11 jako výstupy a PF4, PF5 jako vstupy uvádí následující kód.

```
/* Nastavení výstupních flagů */
*pFIO_DIR |= (PF9 | PF11);
/* Nastavení vstupních flagů */
*pFIO_DIR &= ~(PF4 | PF5);
*pFIO_INEN |= (PF4 | PF5);
```

*Pozn.: Názvy registrů a některých bitů jsou definovány pro procesory BF-53x pomocí maker v souboru **defBF532.h**, takže zápis zdrojových kódů není nutné provádět pomocí adres registrů a hexadecimálních hodnot. Ve vlastním zdrojovém kódu není zápis do registrů prováděn pomocí názvů flagů PFx, ale přes makra, která přiřadí příslušnému flagu přímý název signálu podle schématu, např. přiřazení S_SCLK flagu PF9 provede příkaz **#define S_SCLK PF9** v konfiguračním hlavičkovém souboru **config.h**. Tento soubor slouží k nastavení nejrůznějších parametrů a případnému předefinování signálů na GPIO linkách.*

Pro nastavení systémových hodin (SCLK) a hodin jádra (CCLK) musí být provedena konfigurace PLL jednotky a děliček, viz obr. 3.1.



Obr. 3.1 Systém pro generování hodinových signálů SCLK a CCLK²¹

V registru **PLL_CTL** (*PLL Control*) se nastavují tyto bity:

- **DF** (*Divide Frequency*) – nastavením bitu dojde k aktivaci děličky CLKIN/2 na vstupu jednotky PLL,
- **MSEL[5:0]** (*Multiplier Select*) – neboli zpětnovazební dělička, kterou se řídí kmitočet VCO (*Voltage Contolled Ocillator*), tj. kolikrát je násobený hodinový signál CLKIN za vstupní děličkou. Velikost násobnosti podle hodnoty bitů MSEL udává tab. 3.1.

Hodnota MSEL	0	1	2	$N = 3-62$	63
Násobnost	$64 \times$	$1 \times$	$2 \times$	$N \times$	$63 \times$

Tab. 3.1 Násobnost výstupního kmitočtu VCO vůči kmitočtu na vstupu PLL

- **BYPASS** – nastavením tohoto bitu bude PLL jednotka překlenována a kmitočty SCLK a CCLK budou odpovídat CLKIN.

²¹ Převzato z hardwarového manuálu BF-533 [24].

Dalším registrem, který určuje dělicí faktor kmitočtu napěťově řízeného oscilátoru pro odvození hodin SCLK a CCLK, je **PLL_DIV** (*PLL Divide*) s těmito nastavitelnými bity:

- **CSEL[1:0]** (*Core Select*) – platí, že $CCLK = VCO / x$, kde $x = \{1, 2, 4, 8\}$ podle bitů CSEL,
- **SSEL[3:0]** (*System Select*) – obdobně pro $SCLK = VCO / x$, kde $x = 1-15$ určeno bity SSEL, hodnota 0 není povolena.

Pro nastavení snímače polohy k hledání těžiště světelné stopy nebo rychlý přenos obrazových dat do PC musel být zajištěn maximální výkon (bez přetaktování), tzn. s nastavením SCLK = 133 MHz a CCLK = 400 MHz. S kmitočtem CLKIN = 16 MHz, tj. s připojeným 16 MHz krystalem, bude vypadat úsek zdrojového kódu takto:

```
/* MSEL = 25, DF = 0, BYPASS = 0, VCO = 25*CLKIN = 400 MHz */
*pPLL_CTL = 0x3200;
/* CSEL = 0 (CCLK = VCO = 400 MHz), SSEL = 3 (SCLK = CCLK/3 = 133 MHz) */
*pPLL_DIV = 0x0003;
```

Výhoda výše uvedeného hodinového systému procesorů BF-53x spočívá v tom, že po nastavení děliček pro SCLK a CCLK v registru PLL_DIV, dojde k přenastavení okamžitě, bez nutnosti předchozího uvádění procesoru do režimu *idle*.

3.1.2 Rychlý přenos velkého objemu dat z procesoru do USB řadiče

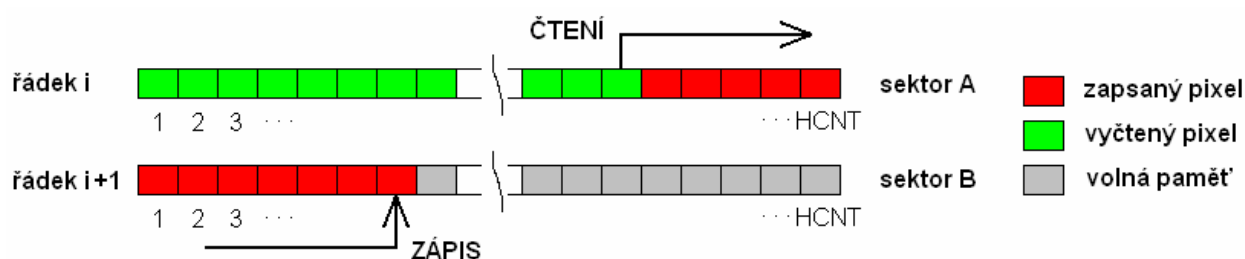
V této kapitole je popsán způsob, jak přenést snímek v plném rozlišení (např. 1,25 MB u MT9M001) ze senzoru do počítače bez použití externí paměti. Dřívější přístupy byly takové, že nejprve se snímek dostal vysokou rychlostí DMA kanálem do vnější SDRAM paměti, a potom byl nejvyšší možnou rychlostí použitého USB řadiče (nebo jiným PC rozhraním) přenesen do počítače. Měl-li však být realizován snímač polohy v minimální konfiguraci bez vnější paměti, musela pro přenos obrazu vystačit omezená vnitřní paměť procesoru, viz srovnání typů BF-53x v tab. 2.5. S vnitřní SRAM přicházejí v úvahu tyto metody, jak přenést obraz v nějaké podobě do PC:

Redukovaný obraz ve skip módu – obrazový senzor je možné nakonfigurovat tak, aby na svou datovou sběrnici posílal pouze určité pixely, ostatní „přeskočil“. Vynechávání pixelů vede ke snížení datového toku, ale i kvality obrazu, což ovšem v některých případech pro pouhé získání náhledu nemusí být překážkou. Dalšího snížení objemu dat lze dosáhnout výřezem segmentu ve snímku (např. střed) a omezení pozornosti na oblast zájmu, na tzv. ROI (*Region of Interest*). Tato obrazová data z jednoho snímku jsou přenesena do SRAM, odkud mohou být poslána dále přes USB řadič do PC.

Složení obrazu – sestává v plném rozlišení z několika snímků, kdy se do SRAM postupně ukládají n -řádkové pruhy, které se vzápětí přenášejí do PC. Pořadí pruhů závisí na implementovaném algoritmu a toto pořadí musí být známé při skládání obrazu na straně počítače. Jinými slovy to znamená, že od okamžiku spouště se v 1. snímku přenesou např. řádky 1–4, 240–244, 480–484, ..., ve 2. snímku 5–9, 245–249, 485–489 a tak dále, dokud nejsou odeslány všechny pruhy ze snímku.

Složení redukováného obrazu – je kombinací předchozích dvou metod, kdy se zvolí únosný *skip* mód, dostatečně velká oblast zájmu, a poté jsou pruhy tohoto redukováného obrazu postupně přeneseny do PC, kdy výsledný obraz je pochopitelně sestaven z nižšího počtu snímků.

Obraz v plném rozlišení – umožňuje přenést rychlý USB řadič s využitím SRAM jako vyrovnávací paměti. Právě tato metoda je používána ve snímači polohy s řadičem FT2232H. Princip spočívá v tom, že obrazová data jsou po řádcích ukládána DMA kanálem do vnitřní paměti a současně jsou v těsném závěsu čtena procesorem a odesílána na datovou sběrnici, ke které je připojen USB řadič, z něhož musí být data kontinuálně vyčítána, protože se nemohou déle hromadit v SRAM procesoru ani ve 4096 B paměti FIFO FT2232H.



Obr. 3.2 Paměť SRAM ve funkci vyrovnávacího bufferu

Situaci vystihuje ilustrace na obr. 3.2. Ve vnitřní paměti jsou alokovány dva sektory A a B, jejichž velikost v bajtech je stejná jako počet pixelů jednoho řádku obrazového senzoru ($HCNT = 1280$ pix u *MT9M001*). Do těchto sektorů se střídavě zapisují data z jednotlivých řádků a vzápětí jsou čtena procesorem a odesílána USB řadičem do PC. Pořadí je tedy takové, že 1. ř → sektor A, 2. ř → sektor B, 3. ř → sektor A, 4. ř → sektor B, ..., $(VCNT-1)$. ř → sektor A, $VCNT$. ř → sektor B. Data na straně PC nemusí být vždy z USB řadiče nepřetržitě vyčítána, občas může dojít k prodlevě vyvolané operačním systémem přepnutím kontextu pro proces s vyšší prioritou. Důležité ale je, aby tato prodleva nepřekročila dobu, za kterou by se zaplnila TX FIFO paměť USB řadiče a oba sektory paměti SRAM obrazovými daty ze senzoru, která se na jeho datové sběrnici vysouvají stále konstantní rychlostí a není možné je během jednoho

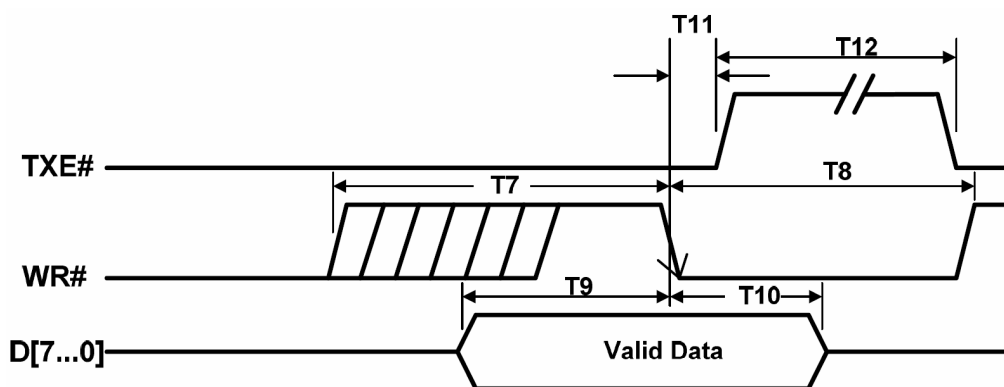
snímku pozastavit. Z výše uvedeného pak vyplývají pro přenos obrazu z jednoho snímku v plném rozlišení následující požadavky:

- maximalizace přenosové rychlosti řadiče FT2232H (dosaženo více než 8 MB/s),
- co největší zpomalení toku dat výstupu obrazového senzoru, aby nepřetekl vyrovnávací buffer paměti SRAM,
- okamžité a spojitě vyčítání dat z FT2232H na straně PC.

Pokud jsou současně všechny tři požadavky splněny, je možné tímto způsobem obraz ze senzoru v plném rozlišení do PC přenést. Na tomto místě je však třeba připomenout, že pro snímač polohy není tato funkcionality prioritní, je to pouze doplněk pro rychlejší a přesnější nastavení objektivu při získávání náhledu z měřené scény. V dalších kapitolách bude proveden popis jednotlivých bodů podrobněji.

3.1.2.1 Maximalizace zapisovací rychlosti procesoru do řadiče FT2232H

Způsob hardwarového propojení procesoru s USB řadičem FT2232H byl uveden na obr. 2.10. Z hlediska procesoru je k řadiči přístupováno jako k paralelní 8-bitové datové sběrnici asynchronní paměti se signály \overline{AWE} (*Asynchronous Memory Write Enable*) a \overline{ARE} (*Asynchronous Memory Read Enable*). Výběrové signály $\overline{AMS3-0}$ (*Asynchronous Memory Selects*) nejsou použity, procesor tedy provádí zápis (čtení) s každou adresací paměťového prostoru 0x20000000 – 0x204FFFFFF, viz [24], tab 17-3. To ovšem není podstatné, neboť datová sběrnice již není sdílena s žádným jiným adresovatelným obvodem. Důležité je správné nastavení časování pro R/W operace, tzn. aby bylo dodrženo časování řídicích signálů FT2232H potřebných pro korektní zápis/čtení z FIFO paměti. Průběhy signálů pro zápis dat do USB řadiče s okótováním jednotlivých intervalů jsou uvedeny na obr. 3.3. a význam signálů byl popsán v kapitole 2.2.3.4. Pro připomenutí – TXE# je výstup FT2232H, který indikuje připravenost řadiče pro zápis dat a musí být před každým zápisem testován, signál WR# je vstup FT2232H a s každou jeho spádovou hranou dochází k vzorkování dat na sběrnici D[7...0] s jejich následným zápisem do vnitřní FIFO paměti.



Obr. 3.3 Zapisovací signály u FT2232H v režimu Async 245 FIFO ²²

Před zapisováním (a čtením) dat na datovou sběrnici musí být v procesoru provedeno nastavení registrů pro řadič externí paměti EBIU (*External Bus Interface Unit*), ve kterých se především udává, kolik cyklů SCLK budou trvat děje během R/W operací na datové sběrnici a řídicích signálech. Při nastavení systémových hodin SCLK na mezní hodnotu 133 MHz vychází délka periody jednoho cyklu přibližně 7,5 ns. Z tohoto údaje se snadno vypočítá, kolik je potřebných cyklů na dodržení správného časování pro zápis podle doporučení výrobce, viz tab. 3.2. Parametry pro čtení jsou obdobné, proto zde nejsou uváděny a lze je nalézt v [10].

Interval	Popis	Min [ns]	Max [ns]	Reg. CTL	Cykly SCLK
T7	Délka neaktivního WR# signálu před zápisem	10	–	–	–
T8	Doba aktivního WR# pulzu	50	–	WAT	10
T9	Přítomnost dat před zápisem	20	–	ST	3
T10	Trvání dat po zápisu	10	–	–	–
T11	Reakční doba signálu TXE# po zápisu	10	25	–	–
T12	Doba zakázaného zápisu při neaktivním TXE#	49	84	–	–

Tab. 3.2 Časování signálů pro zápis do FT2232H

Aktivace přístupu do externí asynchronní paměti musí být nejprve provedena v registru **EBIU_AMGCTL** (*Asynchronous Memory Global Control Register*), kde se pomocí trojice bitů **AMBEN[2:0]** (*Asynchronous Memory Bank Enable*) zapnou požadované paměťové banky 0 - 3. Pro nastavení jednotlivých intervalů časování slouží registry **AMBCTL0,1** (*Asynchronous Memory Bank Control Register*), jejichž bity, které jsou pro konfiguraci časování FT2232H podstatné, mají např. pro banku 0 následující význam:

- **B0WAT** (*Bank0 Write Access Time*) – aktivní doba signálu $\overline{\text{AWE}}$, platný rozsah 1 - 15 cyklů,

²² Převzato z katalogového listu FT2232H [12].

- **B0RAT** (*Bank0 Read Access Time*) – aktivní doba signálu \overline{ARE} , platný rozsah 1 - 15 cyklů,
- **B0HT** (*Bank0 Hold Time*) – doba mezi náběžnou hranou \overline{AWE} nebo \overline{ARE} a náběžnou hranou $\overline{AMS0}$, platný rozsah 0 - 3 cykly,
- **B0ST** (*Bank0 Setup Time*) – doba mezi spádovou hranou $\overline{AMS0}$ a spádovou hranou \overline{AWE} nebo \overline{ARE} , platný rozsah 1 - 4 cykly,
- **B0TT** (*Bank0 Transition Time*) – doba vložená za čtecí přístup do banky 0 a před zapisovacím přístupem do téže banky nebo před čtecím přístupem do banky jiné, platný rozsah 1 - 4 cykly.

Inicializace registrů řadiče externí asynchronní paměti se vykoná po restartu procesoru společně s dalšími konfiguračními procesy jako např. nastavení jednotky PLL, viz kapitola 3.1.1. V následujícím kódu jsou nastaveny registry tak, aby bylo k řadiči přistupováno nejvyšší možnou rychlostí, kdy ještě nejsou překročeny parametry časování z tab. 3.2.

```

/* Zapnutí paměťové banky 0 */
*pEBIU_AMGCTL = AMBEN_B0;
/* Pro SLK = 133 MHz budou nastaveny tyto intervaly: */
/* Write Access Time = 75 ns, Read Access Time = 52,6 ns, */
/* Hold Time = 0 ns, Setup Time = 22,6 ns, Transition Time = 7,5 ns */
*pEBIU_AMBCTL0 = B0WAT_10 | B0RAT_7 | B0HT_0 | B0ST_3 | B0TT_1;

```

V tomto okamžiku, kdy byla konfigurace jednotky EBIU dokončena, bylo již možné přistoupit k samotnému zápisu/čtení dat z externí paměti, neboli řadiče FT2232H. Řadič byl adresován na počátek adresního prostoru banky 0, což se provedlo pomocí tohoto zápisu:

```

/* Globální proměnné */
uint32_t FT_Addr = 0x20000000; /* Adresa FT2232H */
volatile uint8_t *FT_Data; /* Ukazatel na datovou sběrnici */
/* Přiřazení adresy (ve funkci main), na které budou probíhat R/W operace */
/* při manipulaci s proměnou FT_Data. */
FT_Data = (volatile uint8_t *) FT_Addr;

```

Jak již bylo uvedeno v kapitole 2.2.3.4, před samotným zápisem nebo čtením každého bajtu z FT2232H musely být testovány příznaky TXE# a RXF#, které byly připojeny ke vstupním GPIO linkám PF4 a PF5. Testování výskytu dat v Rx bufferu při čtení nebylo kritické, neboť směrem z PC do BF-53x nebylo zapotřebí přenášet vysokorychlostně velký objem dat. Horší

situace nastala při zápisu do FT2232H, protože periodické zjišťování logické úrovně na příslušném flagu znamenalo výrazné zpomalení algoritmu, takže v počátcích vývoje nebylo možné dosáhnout ani mezní frekvence (8 MHz) zapisování na datovou sběrnici USB řadiče. Byly prováděny i experimenty se zápisem bez testování příznaku TXE#, kdy přenosová rychlost dosahovala s upraveným časováním až 10 MB/s, ovšem spolehlivost přenosu nebyla (a z principu ani nemohla být) zaručena.

Pozn.: V hardwarovém manuálu ADSP-BF533, viz [24], se uvádí v kapitole 14, podkapitole “Performance/Throughput“, že jsou-li programovatelné flagy nastaveny jako vstupy, pak při čtení těchto PFX pinů je zpoždění detekce změny jejich log. úrovně procesorem od okamžiku volání příkazu čtení 3 periody SCLK, tedy 22,6 ns při SCLK = 133 MHz. Pro režimy přerušování jsou tyto hodnoty ještě vyšší (4 p. SCLK pro detekci log. úrovně, 5 p. SCLK pro detekci hran), proto je nemělo smysl uvažovat.

Po nejruznějších obměnách zdrojového kódu a vypnutí optimalizace bylo nakonec dosaženo nejvyšší možné zaručené přenosové rychlosti řadiče FT2232H, tj. 8 MB/s. Výsledný úsek kódu pro zápis jednoho znaku na datovou sběrnici s testováním příznaku TXE# je tedy tento:

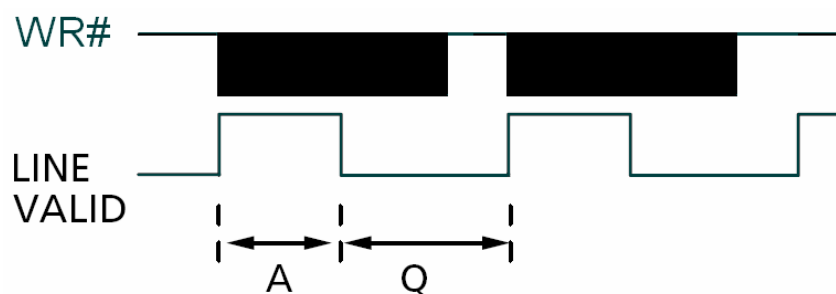
```
#define N_TXE PF4 /* Přiřazení flagu PF4 signálu TXE# */
#pragma optimize_off /* Zákaz optimalizace kódu překladačem */
/* Procedura pro zápis jednoho znaku ch na datovou sběrnici FT2232H */
void USB_PutChar(uint8_t ch) {
    while(*pFIO_FLAG_D & N_TXE); /* Způsob testování úrovně signálu TXE# */
    *FT_Data = ch; /* Zápis znaku ch na sběrnici (Tx buffer Ready) */
}
```

3.1.2.2 Zpomalení toku dat z obrazového senzoru

Má-li být přenesen do PC obraz v plném rozlišení bez použití externí paměti, musí být splněna základní podmínka, tj. že rychlost toku dat ze senzoru nebude větší než přenosová rychlost USB řadiče. Jak bylo uvedeno v předchozí kapitole, u FT2232H se podařilo dosáhnout rychlosti 8 MB/s, která je i výrobcem garantována. Tato rychlost je však podstatně nižší než ta, kterou jsou vysouvána data na sběrnici obrazového senzoru. Závisí pouze na kmitočtu hodinového signálu MCLK, neboť oba používané senzory Micron v sobě bohužel nemají žádnou vnitřní děličku, takže vždy platí, že PIXCLK = MCLK. Vytvoření externí děličky pomocí dalších součástek v obvodu snímače polohy již nepřicházelo v úvahu, protože by tím byl porušen

požadavek minimální možné obvodové konfigurace. Při použití krystalu 8 MHz by sice signál CLKIN ($f_{CLKIN} = f_{MCLK}$) bylo možné PLL jednotkou násobit, aby procesor pracoval na plném výkonu, ovšem snížil by se počet snímků přenesených za sekundu ze senzoru, což není žádoucí z hlediska rychlého hledání světelné stopy. Např. pro senzor *MT9M001* by s krystalem 8 MHz trvalo získání dat jednoho snímku v plném rozlišení cca 200 ms (5 snímků/s), ačkoliv je tento senzor schopný s doporučeným krystalem 48 MHz získat až 30 snímků/s. Jedinou možností tedy bylo zvolit kompromisní hodnotu krystalu (pokud možno dobře násobitelnou v PLL jednotce procesoru) a zpomalit výstup dat ze senzoru prodloužením zatemňovací doby za každým řádkem.

V kapitole 2.3.2.4 byly popsány řádkové a snímkové synchronizační průběhy, přičemž přiblížení vztahu signálů WR# a LINE_VALID je znázorněno na obr. 3.4. U použitých senzorů *Micron* lze prodlevu (*Q*) za aktivní dobu (*A*) vysouvání obrazových dat na sběrnici nastavit v širokém rozsahu příslušným registrem *Horizontal Blanking*. Zapisovací signál WR#, kterým se hradlují data z procesoru na sběrnici FT2232H, sestává v průběhu jednoho řádku z tolika pulzů, kolik jich je nastaveno v senzoru na řádek (proto nejsou na obrázku v tomto zvětšení rozlišitelné).



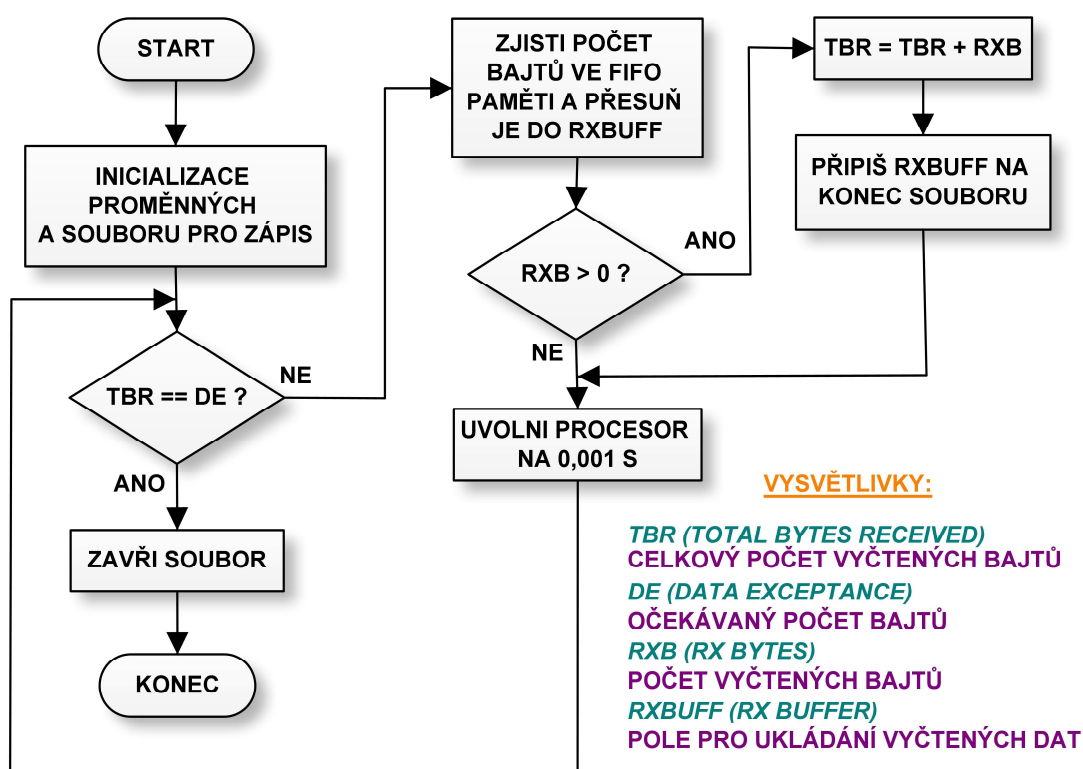
Obr. 3.4 Zapisovací signál WR# v průběhu celého řádku²³

Aby bylo možné přenést do PC celý snímek v plném rozlišení, musí být délka intervalu aktivního signálu WR# kratší než součet intervalů $A + Q$, což znamená, že přenos dat jednoho řádku musí být dokončen před koncem zatemňovacího intervalu Q . Tohoto bylo docíleno s krystalem 16 MHz, takže nejdelší perioda výstupu informace o poloze je 100 ms při plném rozlišení u senzoru *MT9M001*, viz kapitola 4.1.1. Kdyby délka intervalu aktivního signálu WR# nebyla kratší než $A + Q$, docházelo by k postupnému zaplňování vyrovnávacího bufferu v SRAM (kapitola 3.1.2) a po několika řádcích by začalo docházet k přepisování dat, která ještě nebyla do PC odeslána. Z toho důvodu je nutné nastavit registr *Horizontal Blanking* na maximální hodnotu.

²³ Převzato a upraveno z katalogového listu MT9M001 [5].

3.1.2.3 Kontinuální vyčítání dat z USB zařízení na straně PC a jejich následné zpracování

Jakmile se začne přenášet snímek z obrazového senzoru, musí být volný a zcela přístupný celý přenosový kanál ze senzoru až do PC, tzn. vnitřní SRAM procesoru a Tx FIFO paměť v řadiči USB, čehož lze dosáhnout pouze průběžným vyčítáním dat z FT2232H na straně PC, protože ani součet kapacit obou pamětí není takový, aby mohly pojmout celý snímek v plném rozlišení. Jelikož se jedná o poměrně velký objem dat (1,25 MB z *MT9M001*), který má být vysokorychlostně přenesen, musí být na straně PC softwarově zajištěno, aby nedocházelo během přenosu k prodlevám, během nichž by byl zaplněn veškerý volný paměťový prostor ve vyrovnávacím SRAM bufferu. K neúnosným prodlevám by mohlo dojít snadno, pokud by během doby přenosu jednoho celého snímku (100 ms u *MT9M001*) byl mnohokrát změněn kontext v důsledku plánování procesů operačního systému, tj. např. při současném otevření velkého počtu procesorově náročných úloh.



Obr. 3.5 Vývojový diagram pro vyčtení obrazových dat z USB řadiče

Způsob, jakým bylo kontinuální vyčítání obrazových dat z FT2232H provedeno, znázorňuje vývojový diagram algoritmu na obr. 3.5. Obrazová data jsou postupně ukládána do souboru na disku v PC tak, jak jsou posílána z procesoru do USB řadiče, tj. posloupnost bajtů, ve kterých je uložena hodnota jasu příslušného pixelu. Po inicializaci potřebných proměnných a otevření souboru je spuštěn cyklus, který trvá do té doby, než odpovídá celkový počet

vyčtených bajtů počtu očekávanému (např. 1310720 B u *MT9M001*). Poté je soubor s vyčtenými daty zavřen a proces končí. Při vlastním vyčítání je postupováno tak, že v každém cyklu je vyslán do FT2232H dotaz na počet bajtů, které jsou uloženy ve FIFO paměti, a následuje jejich okamžité vyčtení do pole RXBUFF v paměti počítače. Pokud je délka RXB pole RXBUFF nenulová, znamená to, že byla nějaká data vyčtena a uloží se na konec souboru, přičemž se o počet vyčtených bajtů RXB zvýší i počítadlo celkových přenesených dat TBR. V každém cyklu je důležitý moment, kdy se vlákno vzdá na 1 ms dobrovolně nároku na procesor (funkcí **Sleep** (**dwMilliseconds**) v knihovně **windows.h**), čímž částečně umožní ostatním procesům vykonat jejich činnost a není vláknu odebrán procesor operačním systémem „násilně“. Za dobu 1 ms ani nedojde k přetečení vyrovnávacích pamětí v procesoru a USB řadiči. Výrobce FTDI bohužel neuvádí, jakým způsobem fungují jeho ovladače D2XX pro USB zařízení pod OS Windows, takže je obtížné optimálně koordinovat součinnost PC a algoritmu odesílání dat procesorem. Ovšem ukázalo se, že výše uvedený způsob vede k žádanému cíli, tj. spolehlivému a opakovatelnému vyčítání celých snímků v plném rozlišení rychlostí 8 MB/s. Následuje ukázka jádra vyčítací funkce pro PC v jazyce C.

```
/* Inicializace proměnných */
unsigned int totalBytesReceived = 0;
char *filename = (char *)FILENAME;
FILE *file;
errno_t err;
/* Otevření souboru s názvem a cestou FILENAME */
err = fopen_s(&file, FILENAME, "wb");
while(totalBytesReceived < dataExpectance) { /* Vyčítací cyklus */
    Sleep(1); /* Vzdej se procesoru na 1 ms */
    /* Zjistí, kolik je ve FIFO pamětech FT2232H Rx a Tx dat */
    ftStatus = FT_GetStatus(ftHandle,&RxBytes,&TxBytes,&EventDWord);
    /* Přesuň dostupná data do RxBufferu */
    ftStatus = FT_Read(ftHandle,RxBuffer,RxBytes,&BytesReceived);
    if (BytesReceived > 0) { /* Byla nějaká data vyčtena? */
        /* Inkrementuj počítadlo celkově vyčtených dat */
        totalBytesReceived += BytesReceived;
        /* Zapiš nová data na konec souboru */
        result = fwrite(RxBuffer, 1, BytesReceived, file);
    }
}
fclose (file); /* Zavři soubor s uloženými obrazovými daty */
```

Z uloženého souboru mohou být data dále zpracovávána v jiných funkcích (např. vyobrazení snímku na monitor s podporou knihovny *OpenGL*). Celá funkce běží v samostatném vlákne a je spouštěna poté, co počítač vyšle směrem k procesoru požadavek na zachycení jednoho snímku. Sejmutí snímku je možné samozřejmě iniciovat i jiným způsobem, např. testovacím tlačítkem, ale před tímto úkonem musí být již vlákno připraveno na zahájení vyčítání z FT2232H.

3.1.3 Konfigurace PPI rozhraní a DMA kanálu pro přenos dat z CMOS senzoru do vnitřní paměti

V kapitole 2.3.2.3 bylo stručně popsáno PPI rozhraní a DMA přenos obrazových dat do vnitřní paměti SRAM procesoru. Na tomto místě bude provedena ukázka, jakým způsobem je třeba nakonfigurovat určité registry, aby tento přenos mohl být uskutečněn. Jelikož je zapotřebí nastavení mnoha registrů, bude podrobnější popis uveden v komentářích zdrojového kódu. Podstatou této konfigurace je, aby se po inicializaci přenášel vždy jen jeden celý řádek ze snímku na určité místo v paměti, které se za každým řádkem nastavuje a dochází tak ke střídání sektorů paměťového prostoru, viz obr. 3.2. K nastavování registrů jsou používány tyto procedury:

- **void DMA_PPI_Init(uint8_t sensorType)** – podle typu senzoru nastaví 2D DMA přenos (viz dále) o délce jednoho řádku s dvoubajtovým balením obrazových dat,
- **void DMA_LineInit(uint16_t *memory)** – slouží k nastavení adresy v paměti, kam bude zahájen DMA přenos,
- **void DMA_Enable(void)** – aktivace DMA kanálu,
- **void PPI_Enable(void)** – aktivace PPI přenosu,
- **void PPI_Disable(void)** – deaktivace PPI přenosu,
- **void WaitFor_DMA_TransferComplete(void)** – čekání na dokončení běžícího DMA přenosu.

Pro inicializaci DMA0 řadiče musí být nejprve nastaven registr **DMA0_PERIPHERAL_MAP**, ve kterém se určí, která periferie má být napojena na DMA kanál (PPI, UART, SPI, ...). Dále se v registrech **DMA0_?_MODIFY** nastaví velikost inkrementace adresy paměti mezi jednotlivými slovy na řádku a mezi posledním a prvním slovem dvou po sobě jdoucích řádků, což je hodnota 2 z důvodu balení (*Packing Mode*) dvou 8-bitových slov do jednoho 16-bitového. V registru **DMA0_?_COUNT** se stanoví počet slov v řádku a počet sloupců. Kombinací předchozích dvou registrů je možné vytvořit téměř libovolnou 2D ($M \times N$) organizaci ukládání dat do paměti, což je výhodné pro další zpracování. Registrem

DMA0_CONFIG se nastavují další atributy, jejichž význam je částečně uveden v komentářích kódu a podrobněji v hardwarovém manuálu [24]. Inicializační procedura pokračuje dále obdobným nastavením vlastností PPI přenosu, tj. počet sloupců **PPI_COUNT**, počet řádků **PPI_FRAME** a řízení PPI přenosu – **PPI_CONTROL**, kde je důležitý bit **POLC**. Tímto bitem se nastavuje, zda bude prováděno vzorkování dat ze senzoru na paralelní 8-bitové sběrnici náběžnou nebo spádovou hranou signálu PCLK. U dvou používaných senzorů *Micron* se tato hrana liší, takže při opomenutí tohoto faktu pak dochází k jisté chybovosti několika pixelů v obraze.

```

void DMA_PPI_Init(uint8_t sensorType) {
    /* Konfigurace kanálu DMA0 */
    *pDMA0_PERIPHERAL_MAP = 0x0000; /* Namapování DMA0 na PPI */
    *pDMA0_X_MODIFY = 0x0002; /* Inkrementace adresy o 2 bajty (Packing) */
    *pDMA0_Y_MODIFY = 0x0002; /* Inkrementace adresy o 2 bajty (Packing) */
    *pDMA0_X_COUNT = (uint16_t)(H_CNT/2); /* Poloviční počet 16-bit sloupců */
    *pDMA0_Y_COUNT = 0x0001; /* Jeden řádek */
    /* Vlastnosti: RESTART (vyčistění DMA FIFO před zahájením přenosu), */
    /* DMA2D (aktivace 2D DMA), WDSIZE_16 (16-bitový přenos), */
    /* WNR (směr toku dat – cílem je paměť) */
    *pDMA0_CONFIG = RESTART | DMA2D | WDSIZE_16 | WNR;

    /* Nastavení PPI přenosu */
    *pPPI_COUNT = H_CNT - 1; /* Počet sloupců - 1 */
    *pPPI_FRAME = 0; /* Počet řádků - 1 */
    if (sensorType == MT9M001) { /* Nastavení správné hrany vzorkování dat */
        /* Vzorkování dat spádovou hranou MCLK */
        /* Vlastnosti: PACK_EN (aktivace balení dvojice datových bajtů), */
        /* PORT_CFG (10 - dvě externí snímkové synchronizace), */
        /* XFR_TYPE (11 - vypnutí ITU-R 656 módu). */
        *pPPI_CONTROL = PACK_EN | (PORT_CFG & 0x0020) | XFR_TYPE | POLC;
    } else if (sensorType == MT9V032) {
        /* Vzorkování dat náběžnou hranou MCLK */
        *pPPI_CONTROL = PACK_EN | (PORT_CFG & 0x0020) | XFR_TYPE;
    }
}

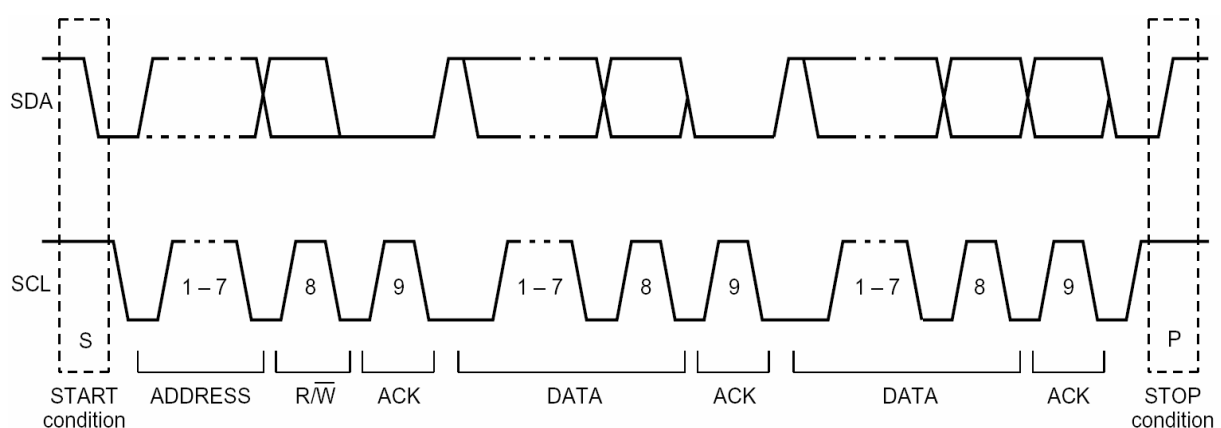
```

Další uvedené procedury obsahují ještě několik DMA/PPI registrů, které jsou ale již nastavovány průběžně během vlastního přenosu obrazových dat ze snímku. Do registru **DMA0_START_ADDR** se přiřazuje po skončení každého řádku ukazatel na adresu sektoru paměti, kam se budou ukládat data řádku následujícího. Zapnutí DMA0 kanálu se provádí nastavením bitu **DMAEN** v registru **DMA0_CONFIG** na začátku každého nového řádku po

konfiguraci počáteční adresy. Pro vyčkání na dokončení DMA přenosu příslušného řádku slouží stavový registr **DMA0_IRQ_STATUS**, ve kterém nastavený bit **DMA_DONE** indikuje kompletní přenos. U PPI řadiče se provádí zapnutí a vypnutí přenosu pouze na začátku a na konci snímku bitem **PORT_EN** v registru **PPI_CONTROL**. Veškerá řádková a snímková synchronizace probíhá automaticky a je řízena signály HSYNC (LINE_VALID) a VSYNC (FRAME_VALID) z obrazového senzoru, viz kapitola 2.3.2.4

3.1.4 Softwarová implementace I²C rozhraní a nastavení registrů CMOS senzoru

Jak již bylo uvedeno v kapitole 2.3.2.1, procesory řady BF-53x v sobě nemají hardwarovou podporu pro I²C komunikaci, musela být tedy emulována softwarově na dvou GPIO linkách, viz obr. 2.17. V zařízení snímače polohy slouží I²C komunikace pouze k výměně dat mezi procesorem (Master) a obrazovým senzorem (Slave), viz obr. 3.6. Princip této komunikace spočívá v tom, že Master stále generuje na vodiči SCL hodinový signál, přičemž po dobu vysílání zapisuje data na vodič SDA, po dobu příjmu z tohoto vodiče čte (zapisuje Slave). Vlastní přenos začíná tzv. START-stavem (spádová hrana SDA při SCL v log. 1) a končí STOP-stavem (náběžná hrana SDA při SCL v log. 1). Oba tyto stavy generuje jednotka Master. Po START-stavu následuje vyslání 7-bitové adresy, jednoho bitu čtení/zápis R/W a jednoho potvrzovacího bitu ACK, který je v log. 1 a jednotka Slave jej musí potvrdit připojením do log. 0. Poté již probíhá až do ukončení přenosu zápis nebo čtení sériových dat, která musí být také po každých osmi bitech potvrzována. Podrobnější popis I²C komunikace lze nalézt ve specifikaci této sběrnice [31].



Obr. 3.6 Kompletní datový přenos na I²C sběrnici²⁴

Softwarová implementace I²C pro BF-53x již byla na katedře měření vyvinuta a odladěna v rámci diplomové práce [23], proto byly po dohodě hotové knihovny pouze s mírnou modifikací

²⁴ Převzato ze specifikace I²C sběrnice [31].

převzaty. Knihovní funkce ovládají příslušné programovatelné flagy tak, že pokud má být signál na sběrnici v úrovni log. 1, přepne se dotýčný pin do vstupního režimu, kdy je ve stavu vysoké impedance, protože pokud by byl jako výstupní a nastaven procesorem do log. 1, došlo by sepnutím přechodu Drain-Source v budiči Slave zařízení ke kolizi (např. při potvrzování bitu ACK). Jestliže má být signál na sběrnici v úrovni log. 0, je příslušný pin nastaven jako výstup a programově uveden do této úrovně, viz [23]. Následující kód popisuje definovaná makra a hlavičky využívaných funkcí.

```
#define I2C_SDA_PIN PF10 /* data pin SDA = PF10 */
#define I2C_SCLK_PIN PF9 /* clock pin SCL = PF9 */

/* Inicializace, bitDurationTics = délka trvání jednoho bitu, závisí na CCLK */
void I2C_init(unsigned int bitDurationTics);
/* Zápis bajtu data do registru registerAddress zařízení deviceAddress */
void I2C_write(unsigned char deviceAddress, unsigned char registerAddress,
               unsigned char data);
/* Čtení bajtu z registru registerAddress zařízení deviceAddress */
unsigned char I2C_read(unsigned char deviceAddress,
                      unsigned char registerAddress);
```

Pomocí těchto funkcí se konfigurace obrazového senzoru provádí jednoduše pouhým zápisem do příslušného registru, jejichž přehled je uveden v katalogových listech *Micron*. Senzorová deska s MT9M001 má jednu I²C Slave adresu, u desky s MT9V032 lze nastavit tyto adresy čtyři (kombinací log. úrovní na vstupních pinech S_CTRL_ADRx). Adresu obrazového senzoru je nutné znát a nastavit v konfiguračním souboru příslušného programu. V budoucích verzích softwarového vybavení pro snímač polohy je plánována autodetekce připojeného obrazového senzoru, což se provede postupným testováním možných I²C adres sensorových desek a čtením registrů 0x00 (*Chip Version*). Se znalostí typu připojeného senzoru by pak byla možná i automatická konfigurace dalších parametrů, které by byly uloženy na vyhrazeném místě EEPROM paměti, tj. např. horizontální a vertikální rozlišení, polarita hodin PPI, velikosti zatemňovacích intervalů apod.

3.2 Hledání těžiště světelné stopy

Tato kapitola se věnuje hledání polohy objektu v rovině na základě určení těžiště světelné značky, která je na objektu připevněna. Z hlediska softwaru musí být provedeno jiné nastavení obrazového senzoru, PPI rozhraní a DMA kanálu. Následuje rozbor algoritmu průběžného zpracování obrazových dat a výpočet řádkových mezivýsledků. Konečným výstupem je informace o poloze objektu a jednoduchý program pro PC, který grafickým způsobem demonstrovuje okamžitou polohu objektu.

3.2.1 Výpočetní metoda určení těžiště světelné stopy z obrazových dat

Základní představa o měření polohy objektu spočívá v tom, že referenční bod objektu je opatřen světelně výraznou značkou (odrazka, svítivá dioda, projekce laserového paprsku apod.). V případě správného nastavení clony a zaostření objektivu sestává snímek monochromatického obrazového senzoru z tmavé plochy, ve které je na určitém místě shluk jasově výrazných bodů představujících světelnou stopu. Tvar světelné stopy nemusí být vždy pravidelný (kruhový) a může být při rychlém pohybu před objektivem rozmazán (analogie v televizním přenosu při zachycení reflektorů pohybující se kamerou). Proto se při měření polohy světelné stopy zavádí pojem **těžiště**, což je označení pro hmotný střed ve fyzice, ve videometrii by pak bylo pravděpodobně výstižnější synonymum **jasový střed**.

Obecné vzorce pro výpočet těžiště (x, y) světelné stopy vycházejí z mechaniky tuhého tělesa, kdy je pouze element hmotnosti tělesa nahrazen elementem jasu obrazu (snímku), tedy jasem jednoho pixelu. Pak platí tyto vztahy:

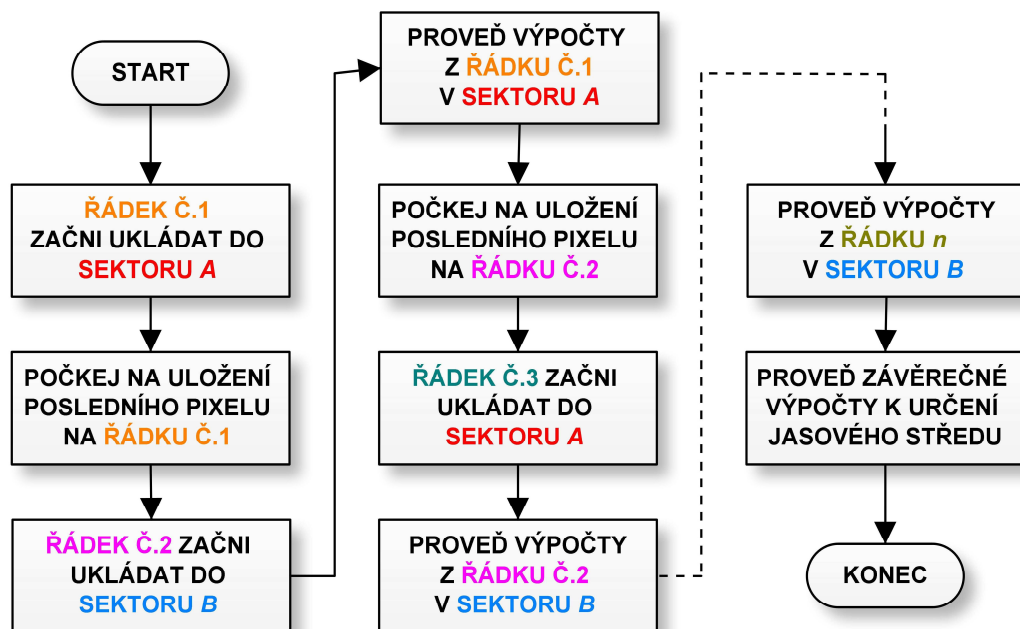
$$x = \frac{\sum_{i=0}^{xH} \sum_{j=0}^{yV} iI(i, j)}{S}, \quad y = \frac{\sum_{i=0}^{xH} \sum_{j=0}^{yV} jI(i, j)}{S}, \quad \text{kde} \quad (3.1)$$

$$S = \sum_{i=0}^{xH} \sum_{j=0}^{yV} I(i, j). \quad (3.2)$$

Funkce $I(i, j)$ zde znamená hodnotu jasu pixelu na pozici i, j . Hodnota xH odpovídá počtu pixelů na jednom řádku (horizontální rozlišení), yV je počet řádků (vertikální rozlišení). Proměnná S je součet všech jasových hodnot pixelů celého snímku. Výsledkem rovnic jsou desetinná čísla, která po zaokrouhlení vyjadřují pozici nejbližší jasovému středu světelné stopy.

3.2.2 Rozbor algoritmu pro výpočet těžiště jedné světelné stopy pouze s vnitřní pamětí SRAM

Výše uvedené vztahy pro výpočet těžiště jsou snadno algoritmizovatelné, pokud se obrazová data celého snímku nacházejí uložená v externí paměti a lze libovolně adresovat. V tomto případě stačí se dvěma vnořenými cykly vykonat nad všemi pixely příslušné sumační funkce, provést operace dělení a výsledek je určen. Před vlastním výpočtem se zpravidla vymezi oblast zájmu, ve které se světelná stopa ve snímku nachází, aby byla rychlost nalezení těžiště maximalizována. Tato operace je vykonávána buď výpočetně (prahováním), nebo je provedena uživatelem při konfiguraci měřicího zařízení.



Obr. 3.7 Algoritmus výpočtu těžiště světelné stopy při použití vnitřní paměti SRAM

K poněkud odlišné situaci dochází při průběžném zpracování obrazové informace s použitím omezené vnitřní paměti SRAM. Na obr. 3.7 je naznačen algoritmus, který provádí výpočet těžiště při využití dvou sektorů vnitřní paměti A a B, do nichž se vejdou právě dva po sobě jdoucí řádky. Na podobném principu ukládání dat do dvou sektorů je založen přenos obrazu celého snímku do PC, který byl popsán v kapitole 3.1.2. Zde musí být za dobu ukládání jednoho řádku do paměti provedeny všechny matematické operace nad řádkem předchozím, který již byl uložen. Délka DMA přenosu odpovídá vždy počtu pixelů jednoho řádku (přesněji řečeno polovině, protože je aktivován režim *Packing Mode*, tj. spojení dvou 8-bitových slov do jednoho 16-bitového). Po skončení každého řádku je provedena změna adresy (na druhý sektor), kam bude DMA řadič ukládat data z řádku následujícího po příchodu náběžné hrany signálu LINE_VALID, viz kapitola 2.3.2.4. Od tohoto okamžiku je činnost procesoru využita ke

zmiňovaným výpočetním operacím, do kterých je zahrnuto počítání čítelů ve vztazích (3.1) a součtu všech jasových hodnot S (3.2). Vlastní výpočty nad jedním řádkem lze v jazyce C zapsat tímto způsobem:

```

short int iX, iY; char iI unsigned int wX, wY, sumXY; /* Deklarace proměnných */
iY = 0; wX = 0; wY = 0; sumXY = 0; /* Inicializace počátečních hodnot */

/* ***** OPERACE NAD JEDNÍM ŘÁDKEM ***** */
for (iX = 0; iX < H_CNT; iX++) { /* Vykonej nad všemi pixely řádku iY */
    iI = memory[iX]; /* Načti do proměnné iI hodnotu jasu pixelu souřadnice iX, iY */
    if (iI > TRESHOLD) { /* Jestliže je tato hodnota jasu větší než práh, počítej */
        wX = wX + (iX + 1) * iI; /* Čítel vzorce pro výpočet těžiště v souřadnici x */
        wY = wY + (iY + 1) * iI; /* Čítel vzorce pro výpočet těžiště v souřadnici y */
        sumXY = sumXY + iI; /* Součet všech jasových hodnot */
    }
}
iY++; /* Inkrementuj řádkové počítadlo */
/* ***** */

```

Jinými slovy lze říci, že tento for–cyklus vykonává procesor nad každým řádkem uloženým v některém ze sektorů paměti SRAM, zatímco DMA kanálem je zároveň přenášen do paměti řádek následující. Hodnota jasu pixelu je uložena do proměnné **iI** a je v následující instrukci komparována s prahovou úrovní jasu danou konstantou **TRESHOLD**. Tímto krokem se vyloučí početní operace nad těmi pixely, jejichž jas nedosahuje požadované úrovně, a jsou tak považovány za černé body v obraze. Je-li podmínka prahování splněna, dojde k přičtení jasu jednoho elementu násobeného indexem polohy v daném směru ke globálním váhovým proměnným **wX** a **wY**, tj. je proveden jeden součet s inkrementovaným indexem i v čítelích vztahů (3.1). Zároveň se vykoná přičtení této hodnoty jasu k celkové jasové proměnné **sumXY** neboli S ve vztahu (3.2). Po dokončení těchto výpočtů nad posledním řádkem snímku následuje závěrečné vydělení váhových proměnných **wX** a **wY** proměnnou **sumXY** (pokud je větší než 0), což jsou výsledná těžiště celého snímku x a y .

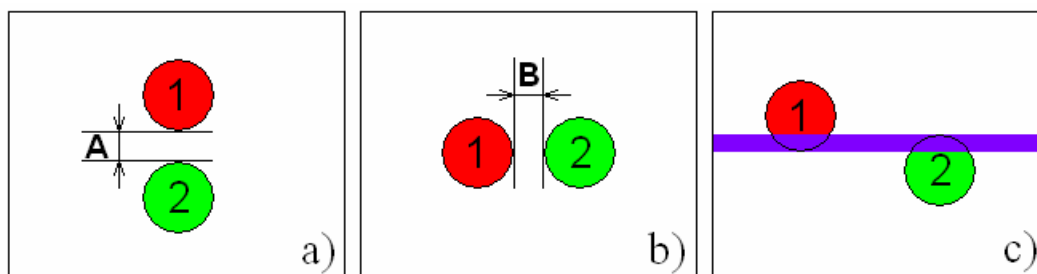
Rychlost tohoto algoritmu je závislá na velikosti světelné stopy, neboť čím více pixelů bude s hodnotou jasu vyšší než **TRESHOLD**, tím vícekrát bude muset být vykonána posloupnost početních operací uvnitř těla podmínky. Od určité velikosti světelné stopy by pak nebylo možné výpočty jednoho řádku stihnout za dobu trvání jednoho řádku, což by vedlo k situaci, kdy např. po dokončení operací nad prvním řádkem by byl již začátek druhého řádku ztracen (data ze senzoru vystupují nepřetržitě) a data do volného sektoru paměti by se začala ukládat až od řádku třetího, takže těžiště by bylo po zpracování yV řádků určeno chybně. Tuto

chybu je možné detekovat opatřením, kdy řadič DMA bude volat přerušení s každým dokončením přenosu dat jednoho řádku a v tomto přerušení bude nastavován příznak chybného výpočtu. Pokud však velikost světelné stopy nepřekročí povolenou mez, celý řádkový algoritmus proběhne ještě před dokončením DMA přenosu aktuálního řádku a zakáže přerušení, které bude opět nastaveno při inicializaci volného paměťového sektoru pro řádek následující.

Dalším úkolem vývoje snímače polohy bude optimalizace tohoto algoritmu po zjištění posloupnosti jednotlivých instrukcí při překladu z jazyka C do assembleru a způsobu proudového zpracování informace (*pipelining*). Příímým naprogramováním algoritmu v assembleru pak bude přesně známo instrukční zpracování v jednotlivých krocích, což nabízí výrazné možnosti zrychlení potřebných výpočtů.

3.2.3 Situace při více světelných stopách v obraze

V některých případech nemusí být měření polohy jedné světelné stopy dostačující a je žádoucí určit v jednom snímku polohy dvou a více stop. V takovém případě se již nebude určovat střed jasového rozložení celého snímku, ale výpočetní algoritmus bude muset být rozšířen o rozpoznání příslušnosti jasově výrazného pixelu k odpovídající světelné stopě, přičemž bude vykonávat početní operace pro každou stopu zvlášť. Vzájemná poloha dvou světelných stop bývá různá, proto je nutné uvážit, v jakých možných pozicích se budou během vlastního měření nacházet a podle toho použít vhodný algoritmus. Na obr. 3.8 jsou znázorněny typické polohy dvou idealizovaných kruhových světelných stop v obraze jednoho snímku. V následujících odstavcích jsou pak jednotlivé případy popsány a rozebrány principy algoritmů pro hledání těžišť bez použití externí paměti, tj. s počítáním z rychlé posloupnosti obrazových dat jednotlivých řádků. Pro tři a více stop by byl algoritmus rozšířen analogicky, pouze s nutným zpomalením toku obrazových dat kvůli větší výpočetní náročnosti.



Obr. 3.8 Různé možnosti vzájemné polohy dvou světelných stop v obraze

Světelné stopy pod sebou – pokud jsou po celou dobu měření dvě světelné stopy ve vzájemné poloze podle obr. 3.8 a), tj. mezi stopami pod sebou je vzdálenost A o velikosti alespoň jednoho tmavého řádku (stopy se nesmí dotknout a spojit v jednu), může být použit algoritmus, který

počítá dvě těžiště ve dvou horizontálních pruzích o šířkách m a n ($m + n + 1 =$ počet řádků na snímek), přičemž při výpočtu je uvažován doplňkový pruh vždy s jasem pod prahovou hodnotou jasu TRESHOLD (černá).

Světelné stopy vedle sebe – zde je poloha dvou stop obdobná (mezi nejkrajnějšími body stop je vzdálenost B alespoň jednoho tmavého pixelu), viz obr. 3.8 b). Ovšem výpočetní metoda je zcela odlišná, neboť v každém řádku musí být testována příslušnost výrazného pixelu ke správné stopě, tzn. po následování světlého pixelu v první stopě pixelem tmavým musí být na témže řádku zahrnut každý další světlý pixel k druhé stopě a na konci řádku musí přepnout výpočty opět nad první stopu.

Světelné stopy částečně vedle sebe – protože není v reálném obraze možné, aby na každém řádku byl vždy zastoupen alespoň jeden výrazný pixel z každé stopy jako je tomu v předchozím případě, je nutné uvažovat pouze s částečným výskytem dvou světelných stop na několika společných řádcích jako je tomu na obr. 3.8 c). Zde tedy musí být navíc správně určeno, ke které světelné stopě výrazný světelný bod přísluší, jestliže je v průběhu jednoho řádku ve skupině světlých bodů, mezi nimiž není žádný bod tmavý, který by indikoval výskyt druhé stopy, což je výpočetně náročné. Pokud by vzájemné polohy stop 1 a 2 byly v měřené scéně takové, že stopy by byly od sebe vždy v horizontálním směru vzdálené alespoň na jeden tmavý pixel, bylo by výhodnější otočit obrazový senzor o 90° a výsledné polohy po transpozici souřadnic určovat způsobem jako na obr. 3.8 a).

3.2.4 Nastavení obrazového senzoru, přenos informace o poloze objektu do PC a grafické vyobrazení v demonstrační aplikaci

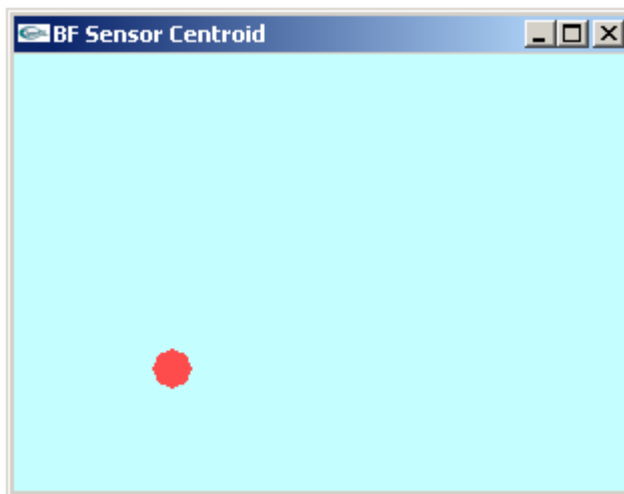
Nastavení obrazového senzoru v režimu měření polohy světelné stopy musí být takové, aby senzor poskytl co největší počet snímků za sekundu a měření tak bylo rychlé. Tím se v zásadě liší od nastavení při přenosu obrazu do PC, kdy naopak musí být tok dat maximálně zpomalen, viz kapitola 3.1.2.2. Zvyšovat snímkovou frekvenci lze u snímačů *Micron* těmito způsoby:

- snížením rozlišení přepnutím do tzv. *skip módu*,
- minimalizací horizontálního a vertikálního zatemňovacího intervalu,
- zvýšením hodinové frekvence pro senzor MCLK až do 48 MHz.

Použitelné je prakticky ale jen snížení rozlišení, protože registry horizontálního/vertikálního zatemnění jsou po resetu senzorů na minimálních hodnotách a zvýšení hodinové frekvence také není v zapojení snímače polohy myslitelné, jak již bylo též uvedeno v kapitole 3.1.2.2. Testované

obrazové senzory mohou pracovat v tzv. *skip módech*, což znamená, že jsou vynechány určité řádky a sloupce, takže je redukován počet pixelů na jeden snímek. Např. v *row, column skip módu* 2 jsou na datovou sběrnici posílány pixely 0, 1, 4, 5, 8, 9, ... v řádcích se stejnými čísly. Horizontální i vertikální rozlišení klesne na polovinu, takže počet aktivních pixelů jednoho snímku je čtvrtinový, tedy algoritmus vykoná výpočet těžiště za poměrně kratší dobu. Řádkové i sloupcové rozlišení lze libovolně kombinovat podle potřebné přesnosti algoritmu v daném směru (nižší přesnost určení těžiště na úkor vyšší snímkové rychlosti je zde zákonitá). Aby bylo zrychlení řádkové frekvence patrné, musí být při *column skip módu* zároveň v odpovídajícím poměru zkrácena doba integrace v registru *Shutter Width*.

Aby bylo možné ověřit funkčnost snímače polohy, bylo nutné nějakým způsobem zobrazovat souřadnice vypočítaného jasového středu světelné stopy. Nejjednodušší možnost představovala obrazovka počítače, do kterého byla s každým snímkem periodicky odesílána data s informací o souřadnicích x a y . Tato operace představovala odeslání osmi bajtů (**unsigned int x, y**) a jejich vyčtení na straně PC spolu s rychlým zobrazením. Výpis souřadnic byl opakovaně prováděn na jednom řádku konzolového okna, ovšem pro lepší představu o poloze těžiště světelné stopy bylo vhodné doplnit aplikaci o jednoduchý grafický výstup, jehož podoba je na obr. 3.9.



Obr. 3.9 Grafické okno demonstrační aplikace k znázornění polohy těžiště světelné stopy ve snímku

Po spuštění aplikace se otevře okno o rozměrech odpovídajících aktuálně nastaveného rozlišení obrazového senzoru. Pokud snímač polohy detekuje těžiště světelné stopy a odesílá souřadnice $x > 0$ a $y > 0$, je na těchto souřadnicích v grafickém okně zobrazena kruhová světelná stopa, která se při vzájemné změně polohy světelné značky na měřeném objektu a obrazového senzoru v okně pohybuje. Aby byl pohyb plynulý a spojitý, bylo třeba využít (jako při vykreslování obrazových dat) podpory grafické karty pomocí funkcí z knihovny standardu *OpenGL*.

4 Závěr

Úkolem této kapitoly je připomenutí cílů diplomové práce, uvedení úspěšnosti jejich splnění, shrnutí dosažených výsledků a nově objevených skutečností. Závěr práce také obsahuje popis některých zásadnějších problémů, vysvětlení jejich příčin a možné způsoby řešení. Ve druhé části kapitoly je pojednání o možnostech budoucího využití snímače polohy nejen z hlediska původního určení, tj. pro měření polohy, ale i z hlediska dalších rozšiřujících funkcí, které zařízení v důsledku navržené koncepce umožňuje.

4.1 Shrnutí dosažených výsledků

Cílem práce bylo navrhnout a realizovat snímač polohy s obrazovým senzorem CMOS a signálovým procesorem Blackfin BF-53x. Snímač polohy měl být v minimální obvodové konfiguraci bez externí paměti SDRAM pro ukládání obrazových dat a měl měřit polohu sledovaného objektu ve snímané scéně s minimálním zpožděním, což je hlavní požadavek pro účely regulace polohy.

Do výsledné podoby se nakonec přes různé menší i větší komplikace podařilo vyvinout zařízení, které výše uvedené požadavky splňuje. Hotový snímač polohy se nachází na oboustranně osazené desce plošného spoje, která díky svým rozměrům a propojovacímu rozhraní tvoří spolu se sensorovými deskami (které byly vyvinuty již dříve na katedře měření) kompaktní měřicí zařízení, viz obr. 7.3.

Mezi hlavní hardwarové rysy snímače polohy patří procesor BF-53x pro zpracování obrazových dat z CMOS senzoru, komunikace s PC přes High Speed USB řadič FT2232H a připojitelnost JTAG emulátoru procesoru na standardizovaný konektor. K dalším obvodovým rysům patří vyvedení procesorových rozhraní SPORT, UART a GPIO, která rozšiřují aplikovatelnost zařízení. Důležitou obvodovou součástí je SPI programátor bootovací paměti EEPROM procesoru, který je realizován také obvodem FT2232H.

Ze softwarového hlediska byla uskutečněna algoritmizace metody hledání těžiště světelné stopy z obrazových dat s využitím malé vnitřní SRAM paměti procesoru a rychlého poskytování vypočítaných výsledků s každým snímkem. Dalším programátorským úspěchem byl přenos obrazu z jednoho snímku v plném rozlišení CMOS senzoru přes procesor do USB řadiče a následně do PC, opět pouze s vnitřní SRAM ve funkci vyrovnávací paměti. Při této operaci bylo nutné vhodně koordinovat spolupráci PC s řadičem FT2232H, aby z něj byla data během celého přenosu bez prodlení vyčítána a nemohlo tak dojít k přepsání obrazových dat v SRAM, jelikož jejichž tok ze senzoru nebylo možné (ani žádoucí) po dobu jednoho snímku pozastavit.

4.1.1 Měření polohy světelné stopy

Hlavním požadavkem na snímač polohy bylo rychlé měření těžiště světelné stopy, což se podařilo s výsledky uvedenými v tab. 4.1. Výpočetní časy byly naměřeny logickým analyzátozem při velikosti světelné stopy, jejíž pixelová plocha zabírala přibližně 1 % z plochy v příslušném rozlišení snímku, tzn. že např. z 1310720 pixelů v rozlišení 1280×1024 bylo cca 13000 pixelů s jasem vyšším než prahová úroveň, takže s těmito body byly prováděny výpočetní operace, viz kapitola 3.2.2. Při této velikosti stopy jsou doba výpočetních operací jednoho řádku kratší než jeho aktivní doba, což je základní požadavek implementovaného algoritmu. Lze tedy očekávat, že od určité velikosti světelné stopy nebude algoritmus fungovat správně, což by bylo snadno indikovatelné např. vyvoláním přerušování od přenosu posledního řádkového pixelu DMA kanálem. Zde je nutno poznamenat, že snímková rychlost senzoru je dána rychlostí hodinového signálu, jehož kmitočet je v tomto případě 16 MHz a je určen kmitočtem externího krystalu oscilátoru procesoru, viz kapitola 2.3.3.1. Tento kmitočet lze zvýšit až na mezní hodnotu, jež je limitována nejslabším článkem ve dvojici procesor – senzor, tj. 40 MHz pro BF-53x, 48 MHz pro MT9M001 a 26,6 MHz pro MT9V032.

Rozlišení snímku	1280 × 1024	640 × 512	320 × 256	160 × 128
Výpočet těžiště jednoho snímku [ms]	100	29,7	9,9	5,3
Snímková rychlost [počet snímků/s]	10	33,7	101	188,7
Výpočetní operace nad jedním řádkem [us]	70,5	35,3	17,7	8,9
Aktivní doba jednoho řádku [us]	80	40	20	10

Tab. 4.1 Výpočet těžiště jedné světelné stopy se senzorem Micron MT9M001 v různých rozlišeních

Případná změna implementace algoritmu v jazyce C přílišné zrychlení výpočetních operací nad jedním řádkem nepřináší, neboť bude vždy záležet na překladači, jakým způsobem provede optimalizaci a transformaci zdrojových kódů do assembleru. Tato oblast zatím nebyla u procesorů BF-53x podrobněji prozkoumána, proto budou rychlé metody zpracování obrazu u těchto procesorů v budoucnu orientovány právě na jazyk assembler.

Paměťové nároky pro vykonání algoritmu nejsou velké, neboť ve vnitřní SRAM je alokována oblast pro dva řádky v plném rozlišení (což pro senzor MT9M001 činí pouze 2,5 kB), takže i s uvážením potřebného paměťového prostoru pro další proměnné je možné výše uvedenou metodu hledání těžiště jedné světelné stopy uskutečnit i na procesoru BF-531. Další

metody zpracování obrazu je však v budoucnu výhodnější testovat na procesorech BF-533, které mají velkou vnitřní paměť, viz přehled v tab. 2.5.

4.1.2 Přenos obrazu ze senzoru v plném rozlišení do počítače

Před samotným přenosem velkého objemu dat ze senzoru bylo nutné nejprve optimalizovat zapisovací algoritmus do USB řadiče FT2232H tak, aby mu byla předávána data z paměti maximální možnou rychlostí, která je garantována výrobcem, tj. 8 MB/s v režimu Async 245 FIFO. To se ukázalo být zpočátku poměrně obtížné, neboť nepříjemnou vlastností řadiče je skutečnost, že před každým zápisem do FIFO paměti musí být testována na pinu TXE# její připravenost, viz kapitola 2.2.3.4 a obr. 2.12. Toto testování totiž výrazně zpomaluje celý zapisovací cyklus, takže muselo být provedeno několik změn ve zdrojovém kódu, než se podařilo výše uvedené zapisovací rychlosti dosáhnout.

Zároveň bylo nutné při takto rychlém přenosu na straně PC zajistit, aby tok dat, který přicházel z procesoru do USB řadiče, byl neprodleně vyčítán a nedošlo k úplnému zaplnění FIFO paměti řadiče. V situaci, kdy je přenášen obraz jednoho snímku, totiž není možné přisun dat z obrazového senzoru pozastavit, takže by v případě větší prodlevy vyčítání dat počítačem došlo k přepsání obsahu ve vyrovnávacích bufferech vnitřní paměti SRAM procesoru.

Vyrovnávací paměť byla vytvořena ve dvou paměťových sektorech SRAM, která funguje tak, že do jednoho sektoru probíhá zápis obrazových dat aktuálního řádku, zatímco z druhého sektoru jsou data z předchozího řádku vyčítána a zapisována do USB řadiče. Každý paměťový sektor má tedy alokovanou velikost pouze jednoho řádku v plném rozlišení, takže je tato operace vykonatelná i s procesorem BF-531, jak již bylo popsáno v předchozí kapitole.

4.1.3 Návrh plošného spoje

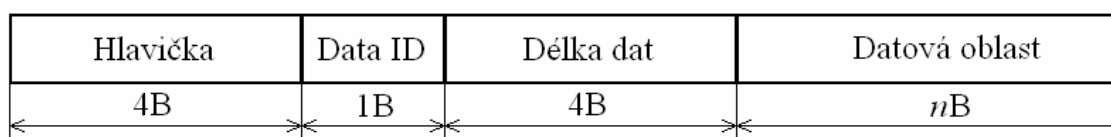
Velká část práce spočívala v návrhu plošného spoje, který musel být z důvodu omezených rozměrů (60 × 60 mm) osazen oboustranně kombinací klasických a SMD součástek. Pracný návrh se nakonec podařilo realizovat na dvouvrstvém spoji ve třídě přesnosti 5, tj. s minimální tloušťkou spoje 8 milů. Na vrchní straně spoje (TOP) se nachází procesor, USB řadič a další součástky, které svojí výškou příliš nezasahují do prostoru sensorové desky, neboť mezi oběma deskami je po vzájemném sesazení mezera pouze 9 mm. Na spodní straně (BOTTOM) jsou osazeny zbývající integrované obvody, konektory pro přístup k procesorovým rozhraním, tlačítka, konektor mini USB (typ B), konektor pro připojení napájecího napětí, filtrační kondenzátory a integrované stabilizátory v pouzdrech TO-220 na okraji desky, které je možné v případě potřeby přišroubovat na chladič. Kompletní fotodokumentace osazené desky a výkresy spojů s osazovacími předlohami se nacházejí v přílohách práce.

4.2 Perspektiva a možnosti využití snímače polohy

4.2.1 Integrace zařízení do prostředí LabView a packetová komunikace mezi snímačem polohy a PC

V současné době je na katedře měření realizován projekt v prostředí LabView, který má za úkol sjednotit přístup k nejrůznějším systémům s obrazovými senzory, které již byly v minulosti v rámci nejrůznějších úloh a absolventských prací vyvinuty. Znamená to, že bude naprogramována knihovna standardních dotazů, které bude možné pomocí této PC aplikace na příslušné komunikační rozhraní odesílat a bude očekávána odpověď ve formátu daného zařízení. Dotazy mohou být typu žádosti o obrazová data, test na šum použitého senzoru, odeslání souřadnic těžiště světelné stopy, obraz s detekovanými hranami apod., podle možností konkrétního kamerového systému. Zároveň bude zmiňovaná aplikace umožňovat konfiguraci nejrůznějších parametrů příslušného systému, případně parametrů samotného obrazového senzoru.

Aby byl požadavek na integraci snímače polohy do výše uvedeného projektu splnitelný, musí hlavní řídicí program procesoru obsahovat modul pro komunikaci s PC, ve kterém budou rozpoznány dotazy z PC aplikace a zároveň bude program procesoru schopný na tyto dotazy definovaným způsobem odpovídat. Jelikož již byla v minulých letech na katedře měření v rámci diplomové práce [23] naprogramována knihovna kvalitní packetové komunikace mezi procesorem BF-53x a PC, ukázalo se je pokračování na tomto typu komunikace výhodné i u snímače polohy. A to z důvodu, že některé moduly v laboratoři již packetovou komunikaci využívají, takže by ani pro snímač polohy nemuselo být v prostředí LabView implementováno další speciální komunikační rozhraní.



Obr. 4.1 Formát komunikačního packetu ²⁵

Packetová komunikace mezi procesorem a počítačem je založena na principu vzájemné výměny datových packetů, viz podrobný popis v diplomové práci [23]. Formát jednoho packetu je uveden na obr. 4.1, přičemž význam jednotlivých polí je následující:

- **hlavička** je tvořena posloupností čtyř bajtů jejichž výskyt ve vlastních datech je nejméně pravděpodobný a slouží k rozpoznání začátku packetu na straně příjemce dat,

²⁵ Upraveno z diplomové práce [23].

- **identifikátor dat** (*Data ID*) je jeden bajt, který definuje význam dat v packetu,
- **velikost přenášených dat** (*Délka dat*) jsou 4 bajty, ve kterých je uložena informace o počtu bajtů uložených v datové oblasti,
- **vlastní data** (*Datová oblast*) je posloupnost užitečných dat o délce definované v předchozím poli packetu.

4.2.2 Další aplikovatelnost zařízení v úlohách měřicí techniky

Kromě dalšího vývoje zařízení zejména v oblasti implementace metod průběžného zpracování obrazové informace do assembleru, nabízí deska snímače polohy využití jako vývojový kit procesorů BF-53x s těmito vyvedenými rozhraními:

- *SPORT* – s připojeným audiokodekem lze testovat nejrůznější metody zpracování audio signálů, generování nízkofrekvenčních průběhů,
- *UART* – využití pro jednoduchou komunikaci mezi dalšími procesorovými obvody, hradlovými poli nebo pro výše uvedené připojení k průmyslové sběrnici RS-485,
- *PFX* – 12 navzájem nezávislých vstupně/výstupních programovatelných flagů pro libovolné účely v číslicové a mikroprocesorové technice,
- *PPI* – možnost připojení k rychlému paralelnímu rozhraní o mezním vzorkovacím kmitočtu $f_{SCLK} / 2$ a šířce sběrnice až 14 bitů. Tímto rozhraním lze tedy jednoduše realizovat logický, analyzátor, generátor průběhů, číslicový osciloskop atd.
- *TMRO* – připojení k časovači, kterým je možné generovat PWM signál, měřit krátké časové intervaly apod.

Pro účely ladění a vývoje programů je deska zařízení opatřena také rozhraním *JTAG*, které bylo otestováno s emulátorem ICE-100B od výrobce *Analog Devices*. I při absenci emulátoru je možné snadné a rychlé programování bootovací EEPROM paměti pomocí USB řadiče přímo z vývojového prostředí VisualDSP++, viz kapitola 2.2.3.5. Využití zařízení snímače polohy jako vývojového kitu dále podporují dvě indikační LED a jedno testovací tlačítko, což bývají v počátcích práce s novým procesorem nejspolehlivější prvky k ověření funkčnosti nahraného programu.

5 Seznam obrázků a tabulek

Obr. 1.1	Schematické znázornění úlohy snímače polohy v regulační smyčce.....	9
Obr. 1.2	Blokové schéma plošného obrazového senzoru CMOS (Micron MT9M001)	11
Obr. 1.3	Příklady světelných stop LED a vypočítaná těžiště při různě nastaveném objektivu.....	12
Obr. 1.4	Princip řádkového PSD senzoru a jeho náhradní schéma	13
Obr. 1.5	Určení polohy světelné stopy pomocí dvou řádkových nebo jednoho plošného PSD senzoru	14
Obr. 1.6	Tické zpracování analogových signálů z PSD senzorů.....	14
Obr. 1.7	lokové schéma snímače polohy se signálovým procesorem a externí SDRAM..	15
Obr. 1.8	Senzorová síť ze snímačů polohy S1 - Sn.....	16
Obr. 2.1	Blokové schéma měření polohy obrazovým senzorem.....	18
Obr. 2.2	Využití USB řadiče k programování EEPROM.....	19
Obr. 2.3	Propojení FT245R, paměti EEPROM a BF-53x	21
Obr. 2.4	Propojení jednotek na SPI sběrnici a časové průběhy signálů	22
Obr. 2.5	Sériový odpor s parazitními kapacitami na signálové cestě mezi USB řadičem a procesorem.....	23
Obr. 2.6	Signály na SPI sběrnici pro vyčtení bloku dat z paměti EEPROM 25LC256	27
Obr. 2.7	Signály na SPI sběrnici pro zápis jednoho bajtu do paměti EEPROM 25LC256	28
Obr. 2.8	Signály na SPI sběrnici pro zápis celé 64 B stránky do paměti EEPROM 25LC256	28
Obr. 2.9	Propojení FT232R, paměti EEPROM a BF-53x	30
Obr. 2.10	Propojení FT2232H, paměti EEPROM a BF-53x	34
Obr. 2.11	Vývojový diagram programátoru EEPROM s FT2232H.....	36
Obr. 2.12	Průběhy signálů WR# a TXE# na rozhraní Async 245 FIFO	39
Obr. 2.13	Průběhy signálů RD# a RXF# na rozhraní Async 245 FIFO	39
Obr. 2.14	Dialogové okno prostředí VisualDSP++ pro nastavení Post-build příkazů.....	41
Obr. 2.15	Funkční blokové schéma procesorů Blackfin ADSP BF-53x	43
Obr. 2.16	Znázornění propojení procesoru BF-53x se signály sensorové desky	44
Obr. 2.17	Sběrnice I ² C mezi procesorem a CMOS senzorem.....	46
Obr. 2.18	Konfigurace PPI rozhraní pro přenos obrazových dat z CMOS senzoru	47
Obr. 2.19	Průběhy signálů snímkové a řádkové synchronizace senzorů Micron	48
Obr. 2.20	Obvod pro generování hodinových signálů BF-53x	50
Obr. 2.21	PLL systém procesoru BF-53x	51
Obr. 2.22	Schéma zapojení resetovacího obvodu pro BF-53x a obrazový senzor	52
Obr. 2.23	Připojení 1 MB paměti EEPROM M25P80 na sběrnici SPI.....	53
Obr. 2.24	Funkční bloková schémata budičů RS-232/485	55
Obr. 2.25	Připojení testovacího tlačítka a indikačních LED na GPIO linky.....	56
Obr. 2.26	Připojení synchronních zařízení A a B k sériovému portu SPORT0	57
Obr. 2.27	Schéma zapojení JTAG konektoru.....	58
Obr. 2.28	Schéma zapojení lineárního měniče pro napětí 3,3 V s LM317T	59
Obr. 3.1	Systém pro generování hodinových signálů SCLK a CCLK	62
Obr. 3.2	Paměť SRAM ve funkci vyrovnávacího bufferu.....	64
Obr. 3.3	Zapisovací signály u FT2232H v režimu Async 245 FIFO	66
Obr. 3.4	Zapisovací signál WR# v průběhu celého řádku	69
Obr. 3.5	Vývojový diagram pro vyčtení obrazových dat z USB řadiče	70
Obr. 3.6	Kompletní datový přenos na I ² C sběrnici	74
Obr. 3.7	Algoritmus výpočtu těžiště světelné stopy při použití vnitřní paměti SRAM	77
Obr. 3.8	Různé možnosti vzájemné polohy dvou světelných stop v obraze	79

Obr. 3.9	Grafické okno demonstrační aplikace k znázornění polohy těžiště světelné stopy ve snímku	81
Obr. 4.1	Formát komunikačního packetu	85
Obr. 7.1	Snímač polohy: osazená vrchní strana (TOP)	95
Obr. 7.2	Snímač polohy: osazená spodní strana (BOTTOM)	95
Obr. 7.3	Snímač polohy a sensorová deska: pohled zepředu	96
Obr. 7.4	Snímač polohy a sensorová deska: pohled zezadu.....	96
Obr. 7.5	Snímač polohy s připojeným JTAG emulátorem ICE-100B.....	97
Obr. 7.6	Počátky vývoje snímače polohy na nepájivém kontaktním poli	97
Obr. 7.7	Výkres plošného spoje – pohled shora (TOP).....	98
Obr. 7.8	Osazovací výkres pro vrchní stranu spoje (TOP).....	98
Obr. 7.9	Výkres plošného spoje – pohled zdola (BOTTOM)	99
Obr. 7.10	Osazovací výkres pro spodní stranu spoje (BOTTOM).....	99
Tab. 2.1	Přehled nejpodstatnějších parametrů dostupných USB řadičů.....	20
Tab. 2.2	Výběr operace v režimu CPU FIFO Style	33
Tab. 2.3	Přehled a význam argumentů funkce SPI_WriteHiSpeedDevice	38
Tab. 2.4	Časy potřebné k zápisu/vyčtení celé EEPROM 25LC256	40
Tab. 2.5	Přehled velikostí pamětí procesorů ADSP BF-53x	44
Tab. 2.6	Propojení konektoru sensorové desky s procesorem a celým obvodem	45
Tab. 2.7	Rozložení pinů konektoru J ₂ -UART na plošném spoji.....	54
Tab. 2.8	Propojení pinů na konektoru J ₄ s piny GPIO	56
Tab. 2.9	Rozložení signálů SPORT0 na pinech konektoru J ₃	57
Tab. 2.10	Rozložení a popis pinů na konektoru JTAG	58
Tab. 3.1	Násobnost výstupního kmitočtu VCO vůči kmitočtu na vstupu PLL.....	62
Tab. 3.2	Časování signálů pro zápis do FT2232H	66
Tab. 4.1	Výpočet těžiště jedné světelné stopy se senzorem Micron MT9M001 v různých rozlišeních.....	83
Tab. 7.1	Soupiska součástek pro snímač polohy	100

6 Seznam odborné literatury

- [1] KISER, John. *MOTION CAPTURE TECHNOLOGY: The Basics* [online]. [cit. 13. 11. 2010]. Dostupný z WWW:
<http://www.c-motion.com/corpfiles/Mocap%20Basics%20for%20Science.pdf>
- [2] ĎAĎO, Stanislav; KREIDL, Marcel. *SENZORY A MĚŘICÍ OBVODY*. 2. vydání. Praha : Vydavatelství ČVUT, 1996. 315 s. ISBN 80-01-02057-6.
- [3] PAVLIŠTA, Dan. *DIPLOMOVÁ PRÁCE: 3D videosensor* [online]. Praha ČVUT FEL, katedra řídicí techniky, 2009. [cit. 13. 11. 2010]. Dostupný z WWW:
http://support.dce.felk.cvut.cz/mediawiki/images/f/fa/Dp_2010_pavlista_dan.pdf
- [4] LAURENT, Robin; JEROME, Baron. *CMOS IMAGE SENSORS: Technologies & Markets, 2010 Report* [online]. [cit. 15. 11. 2010]. Dostupný z WWW:
http://www.imicronews.com/upload/Rapports/Yole_CMOS_Image_Sensor_2010_Report_SAMPLE.pdf
- [5] MICRON TECHNOLOGY, Inc. *KATALOGOVÝ LIST: MT9M001 – rev F* [online]. Boise 2006. [cit. 15. 11. 2010]. Dostupný z WWW:
<http://pdf1.alldatasheet.com/datasheet-pdf/view/115168/MICRON/MT9M001.html>
- [6] FISCHER, Jan. *OPTOELEKTRONICKÉ SENZORY A VIDEOMETRIE*. Praha : Vydavatelství ČVUT, 2002. 315 s. ISBN 80-01-02525-X.
- [7] PAVLÍČEK, Tomáš. *DIPLOMOVÁ PRÁCE: Metody průběžného zpracování obrazu a jejich implementace do signálového procesoru Blackfin ADSP-BF532* [online]. Praha ČVUT FEL, katedra měření, 2008. [cit. 15. 11. 2010]. Dostupný z WWW:
http://measure.feld.cvut.cz/cs/system/files/files/cs/vyuka/zaverecne_prace/DP_2008_Pavlic_ek_Tomas_locked.pdf
- [8] FTDI CHIP, Ltd. *KATALOGOVÝ LIST: FT232R – ver 2.07* [online]. Glasgow 2010. [cit. 22. 11. 2010]. Dostupný z WWW:
http://ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf
- [9] FTDI CHIP, Ltd. *KATALOGOVÝ LIST: FT245R – ver 2.10* [online]. Glasgow 2010. [cit. 22. 11. 2010]. Dostupný z WWW:
http://ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT245R.pdf
- [10] FTDI CHIP, Ltd. *KATALOGOVÝ LIST: FT2232H – ver 2.09* [online]. Glasgow 2010. [cit. 22. 11. 2010]. Dostupný z WWW:
http://ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT2232H.pdf

- [11] CYPRESS SEMICONDUCTOR CORPORATION. *KATALOGOVÝ LIST: CY7C6801xxA - rev *S* [online]. San Jose 2010. [cit. 22. 11. 2010]. Dostupný z WWW: <http://www.cypress.com/?docID=24267>
- [12] ZIMA, Jiří. *BAKALÁŘSKÁ PRÁCE: Rychlý přenos obrazu s využitím EZ-USB*. [online]. Praha ČVUT FEL, katedra měření, 2009. [cit. 22. 11. 2010]. Dostupný z WWW: http://measure.feld.cvut.cz/cs/system/files/files/cs/vyuka/zaverecne_prace/BP_2009_Zima_Jiri_locked.pdf
- [13] FTDI CHIP, Ltd. *APLIKAČNÍ POZNÁMKA AN_114: Interfacing FT2232H Hi-Speed Devices To SPI Bus – ver 1.0* [online]. Glasgow 2009. [cit. 25. 11. 2010]. Dostupný z WWW: http://www.ftdichip.com/Support/Documents/AppNotes/AN_114_FTDI_Hi_Speed_USB_To_SPI_Example.pdf
- [14] FTDI CHIP, Ltd. *APLIKAČNÍ POZNÁMKA AN232R-01: Bit Bang Modes for the FT232R and FT245R – ver 2.02* [online]. Glasgow 2010. [cit. 28. 11. 2010]. Dostupný z WWW: http://www.ftdichip.com/Support/Documents/AppNotes/AN_232R-01_Bit_Bang_Mode_Available_For_FT232R_and_Ft245R.pdf
- [15] FTDI CHIP, Ltd. *VÝVOJ APLIKAČNÍHO SOFTWARE: D2XX Programmer's Guide -- ver 1.01* [online]. Glasgow 2010. [cit. 28. 11. 2010]. Dostupný z WWW: http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX_Programmer%27s_Guide%28FT_000071%29.pdf
- [16] MICROCHIP TECHNOLOGY, Inc. *KATALOGOVÝ LIST: 25AA256/25LC256 – rev E* [online]. Arizona 2005. [cit. 28. 11. 2010]. Dostupný z WWW: <http://ww1.microchip.com/downloads/en/devicedoc/21822e.pdf>
- [17] FTDI CHIP, Ltd. *APLIKAČNÍ POZNÁMKA AN_111: Programmers Guide for High Speed FTCSPI DLL -- ver 1.1* [online]. Glasgow 2009. [cit. 04. 12. 2010]. Dostupný z WWW: http://www.ftdichip.com/Support/Documents/AppNotes/AN_111_Programmers_Guide_for_High_Speed_FTCSPI_DLL.pdf
- [18] FTDI CHIP, Ltd. *APLIKAČNÍ POZNÁMKA AN_114: Interfacing FT2232H Hi-Speed Devices To SPI Bus -- ver 1.0* [online]. Glasgow 2009. [cit. 04. 12. 2010]. Dostupný z WWW: http://www.ftdichip.com/Support/Documents/AppNotes/AN_114_FTDI_Hi_Speed_USB_To_SPI_Example.pdf
- [19] STMICROELECTRONIC. *KATALOGOVÝ LIST: M25P80 – rev 1.2, 2002* [online]. [cit. 04. 12. 2010]. Dostupný z WWW: http://www.datasheetcatalog.org/datasheets/228/227105_DS.pdf

- [20] FTDI CHIP, Ltd. *APLIKAČNÍ POZNÁMKA AN232B-06: Debugging FT232BM / FT245BM Designs – ver 1.1* [online]. Glasgow 2004. [cit. 05. 12. 2010]. Dostupný z WWW: http://www.ftdichip.com/Support/Documents/AppNotes/AN232B-06_11.pdf
- [21] VEDRAL, Josef; FISCHER, Jan. *ELEKTRONICKÉ OBVODY PRO MĚŘICÍ TECHNIKU*. Praha : Vydavatelství ČVUT, 2004. 340 s. ISBN 80-01-02966-2.
- [22] ANALOG DEVICES, Inc. *KATALOGOVÝ LIST: ADSP-BF531/ADSP-BF532/ADSP-BF533 – rev G* [online]. Norwood MA 2010. [cit. 6. 12. 2010]. Dostupný z WWW: http://www.analog.com/static/imported-files/data_sheets/ADSP-BF531_BF532_BF533.pdf
- [23] PRIBULA, Ondrej. *DIPLOMOVÁ PRÁCA: Vývojový modul pre spracovanie obrazu so signálovým procesorom ADSP-Blackfin*. [online]. Praha ČVUT FEL, katedra měření, 2007. [cit. 6. 12. 2010]. Dostupný z WWW: http://measure.feld.cvut.cz/cs/system/files/files/cs/vyuka/zaverecne_prace/DP_2007_Pribula_Ondrej_locked.pdf
- [24] ANALOG DEVICES, Inc. *KATALOGOVÝ LIST: ADSP-BF533 Blackfin Processor Hardware Reference – rev 3.4* [online]. Norwood MA 2009. [cit. 8. 12. 2010]. Dostupný z WWW: http://www.analog.com/static/imported-files/processor_manuals/bf533_hwr_Rev3.4.pdf
- [25] ANALOG DEVICES, Inc. *KATALOGOVÝ LIST: ADM706x/708x – rev C* [online]. Norwood MA 2008. [cit. 10. 12. 2010]. Dostupný z WWW: http://www.analog.com/static/imported-files/data_sheets/ADM706P_706R_706S_706T_708R_708S_708T.pdf
- [26] ANALOG DEVICES, Inc. *ENGINEER-TO-ENGINEER NOTE: EE240, ADSP-BF533 Blackfin Booting Process – rev 4* [online]. Norwood MA 2008. [cit. 11. 12. 2010]. Dostupný z WWW: http://www.analog.com/static/imported-files/application_notes/EE-240_Rev4.pdf
- [27] STMICROELECTRONIC. *KATALOGOVÝ LIST: M25P32 – rev 4.0*, 2004 [online]. [cit. 11. 12. 2010]. Dostupný z WWW: <http://www.acinonyx.tk/wp-content/uploads/2009/02/m25p32.pdf3>
- [28] ANALOG DEVICES, Inc. *KATALOGOVÝ LIST: ADM3202/ADM3222/ADM1385 – rev D* [online]. Norwood MA 2006. [cit. 11. 12. 2010]. Dostupný z WWW: http://www.analog.com/static/imported-files/data_sheets/ADM3202_3222_1385.pdf
- [29] ANALOG DEVICES, Inc. *KATALOGOVÝ LIST: ADM485 – rev F* [online]. Norwood MA 2008. [cit. 11. 12. 2010]. Dostupný z WWW: http://www.analog.com/static/imported-files/data_sheets/ADM485.pdf

- [30] ANALOG DEVICES, Inc. *UŽIVATELSKÁ PŘÍRUČKA: ICE-100B Emulator – rev 1.0*
[online]. Norwood MA 2009. [cit. 13. 12. 2010]. Dostupný z WWW:
http://www.analog.com/static/imported-files/emulator_manuals/ICE-100B_rev.1.0.pdf
- [31] NXP SEMICONDUCTORS. *SPECIFIKACE SBĚRNICE: The I2C-bus specification – ver 2.1*
[online]. Eindhoven 2000. [cit. 24. 12. 2010]. Dostupný z WWW:
http://www.nxp.com/acrobat_download2/literature/9398/39340011.pdf

7 Přílohy

7.1 Příloha A: Obsah příloženého CD

ZDROJOVÉ KÓDY PRO PC:

- **Programování EEPROM paměti**
 - programátor paměti 25LC256 s FT245R v režimu Bit Bang
 \SRC\PC\EEPROM_PROG\25LC256_BB\
 - programátor paměti 25LC256 s FT2232H v režimu MPSSE SPI
 \SRC\PC\EEPROM_PROG\25LC256_SPI\
 - programátor paměti M25P80 s FT2232H v režimu MPSSE SPI
 \SRC\PC\EEPROM_PROG\M25P80_SPI\
- **Přenos obrazu**
 - zobrazovač snímku z připojeného obrazového senzoru
 \SRC\PC\IMG_VIEW
- **Měření polohy**
 - zobrazovač polohy světelné stopy
 \SRC\PC\CENTROID_DEMO\

ZDROJOVÉ KÓDY PRO PROCESOR:

- řídicí program pro přenos obrazu a výpočet těžiště světelné stopy
 \SRC\BF\BF_CTRL\

KNIHOVNY:

- **Knihovny pro přístup k řadičům FTDI v OS WINDOWS**
 - soubory FTDI D2XX
 \LIB\PC\FTDI_D2XX\
 - soubory FTDI SPI DLL
 \LIB\PC\FTDI_CSPI\
- **Knihovny pro podporu grafického zobrazení OpenGL**
 - soubory GLUT
 \LIB\PC\OPENGL\

DIPLOMOVÁ PRÁCE:

- tato diplomová práce ve formátu pdf
 \DP\DP_2011_BELIK_PETR\

FOTODOKUMENTACE:

- fotodokumentace snímače polohy během vývoje a po dokončení
 \FOT\DOKUMENT\

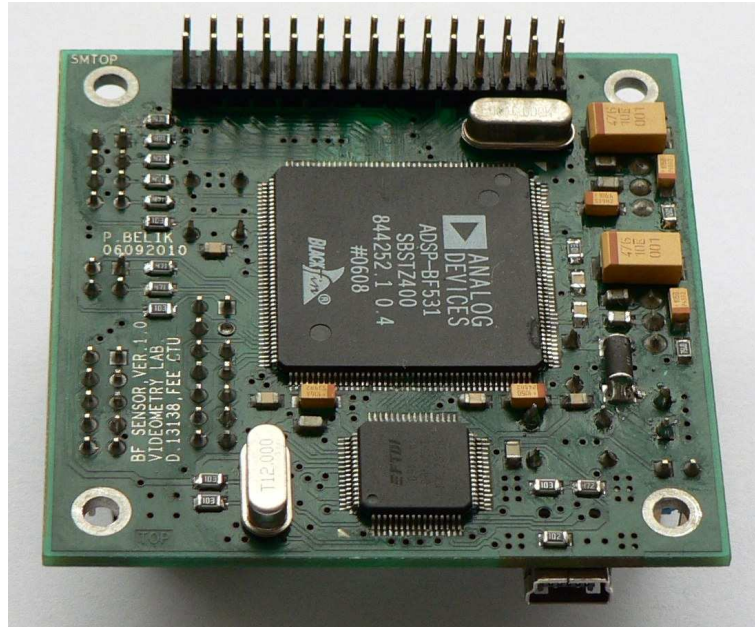
SCHÉMA ZAPOJENÍ:

- kompletní schéma zapojení ver. 2.7 ve formátu pdf
 \SRC\SCH\SCH_VER_27\

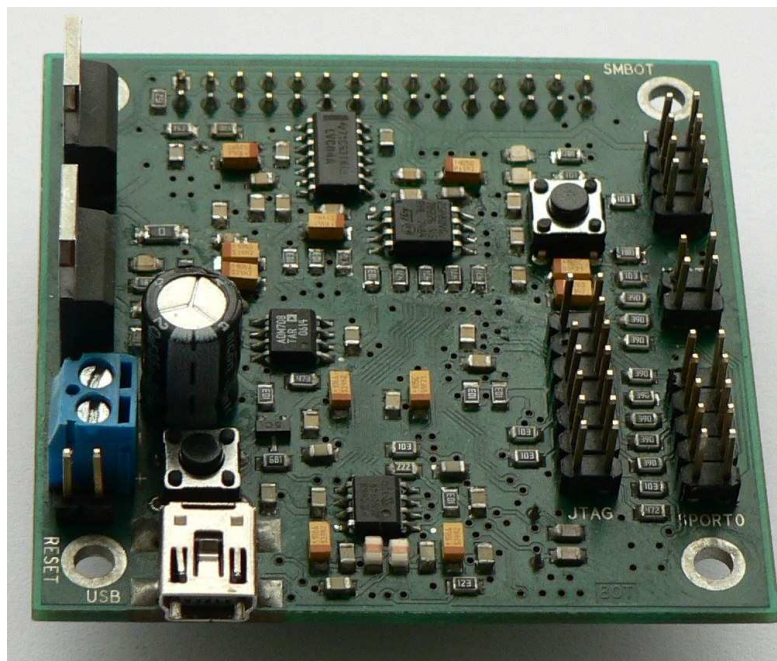
PLOŠNÝ SPOJ:

- výrobní data ver. 1.1
 \BRD\BRD_VER_11\VYROBA\
 - osazovací předlohy ver. 1.1 ve formátu jpg
 \BRD\BRD_VER_11\OSAZ\
 - výkresy plošného spoje ver. 1.1 ve formátu jpg
 \BRD\BRD_VER_11\DPS\
 - soupiska materiálu ver. 1.1 ve formátu xls
 \BRD\BRD_VER_11\BOM\

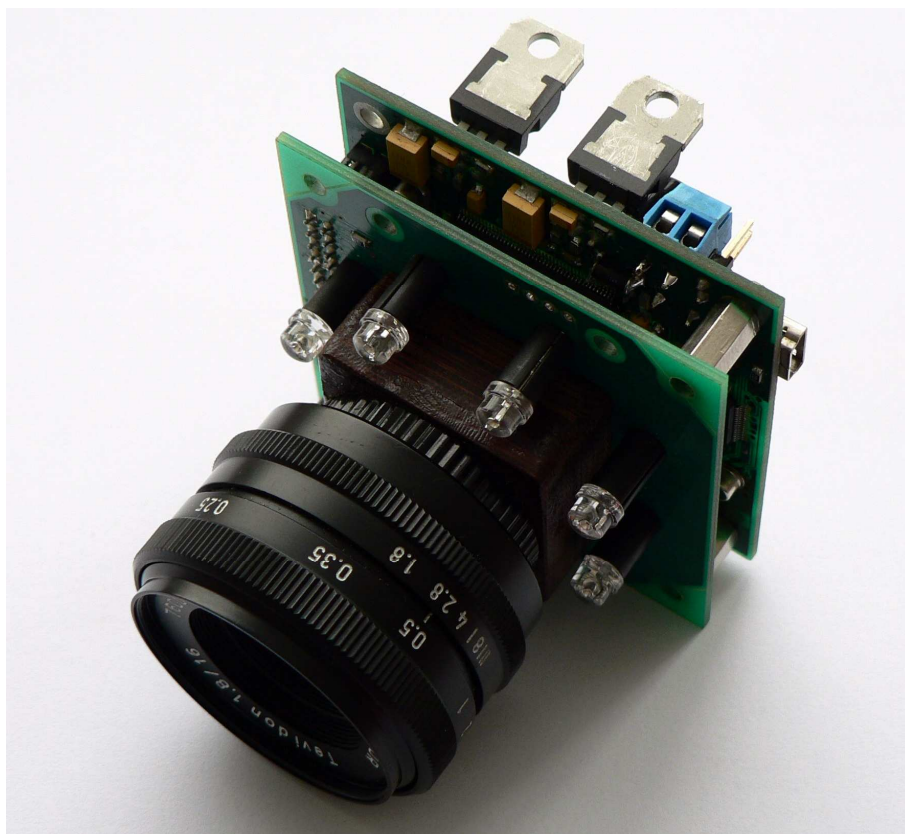
7.2 Příloha B: Fotodokumentace



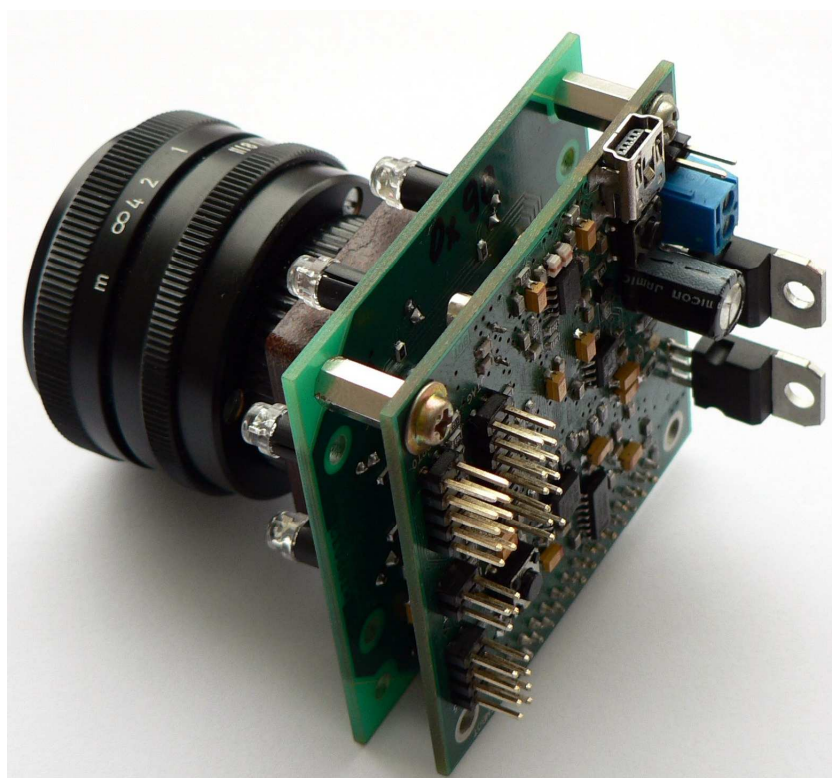
Obr. 7.1 Snímač polohy: osazená vrchní strana (TOP)



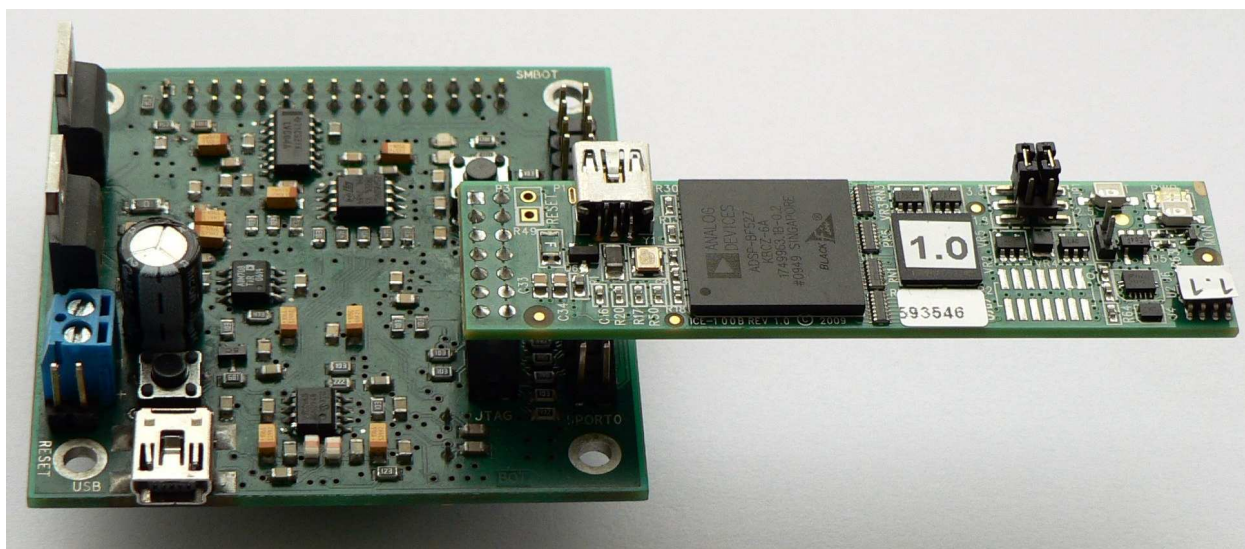
Obr. 7.2 Snímač polohy: osazená spodní strana (BOTTOM)



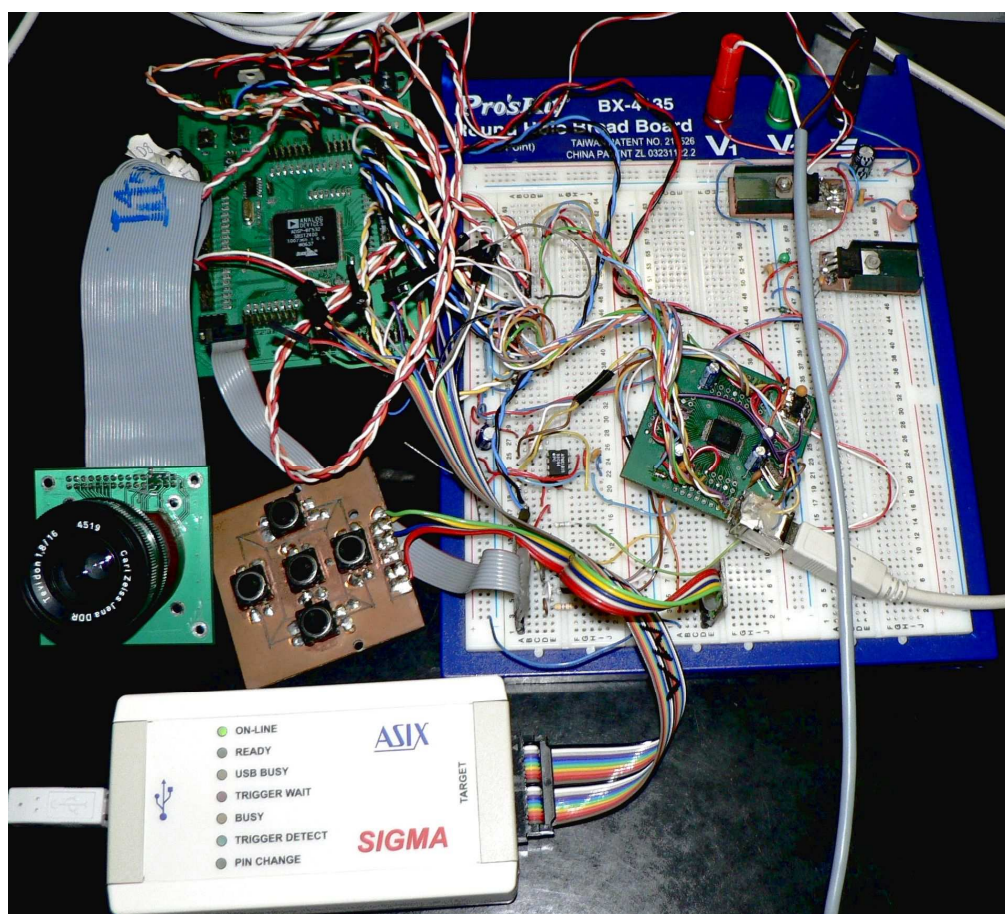
Obr. 7.3 Snímač polohy a senzorová deska: pohled zepředu



Obr. 7.4 Snímač polohy a senzorová deska: pohled zezadu

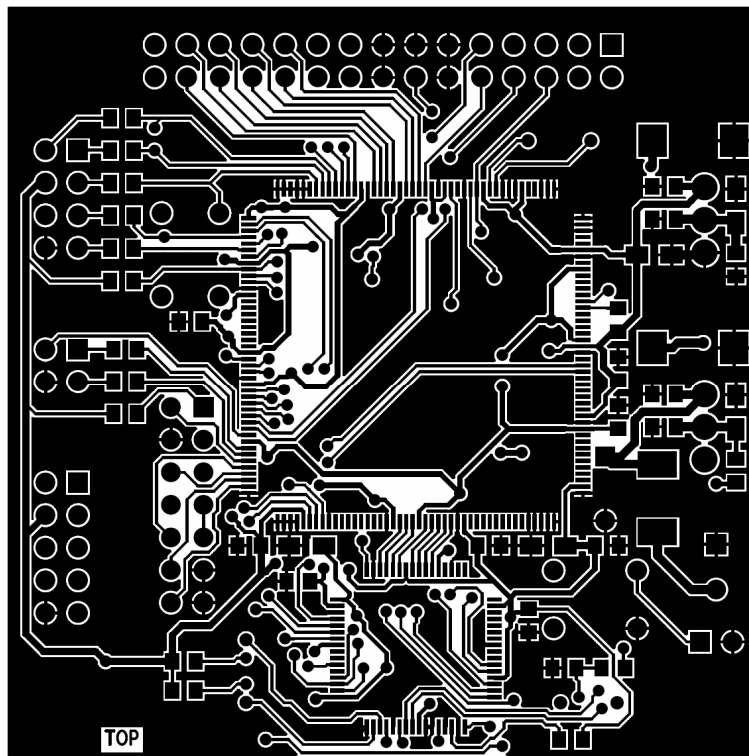


Obr. 7.5 Snímač polohy s připojeným JTAG emulátorem ICE-100B

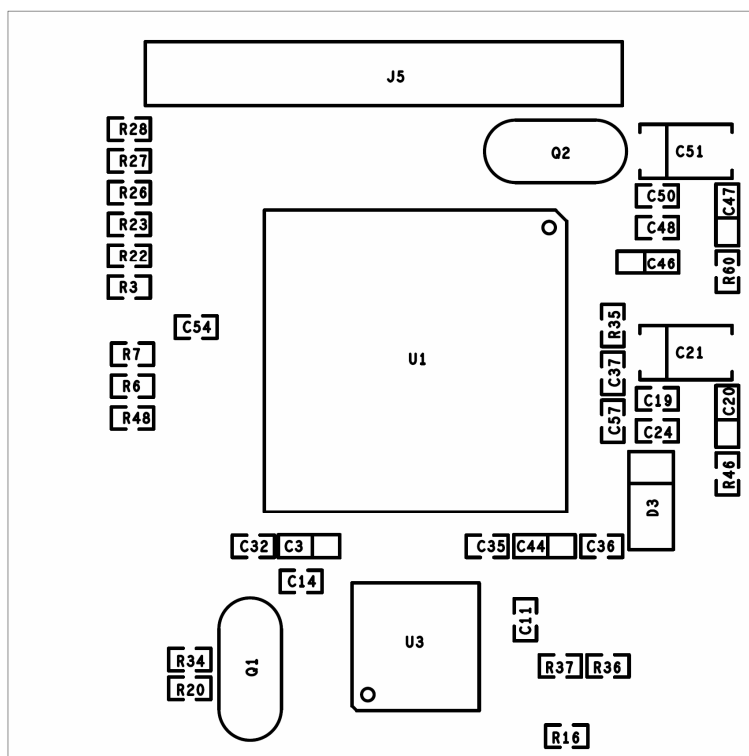


Obr. 7.6 Počátky vývoje snímače polohy na nepájivém kontaktním poli

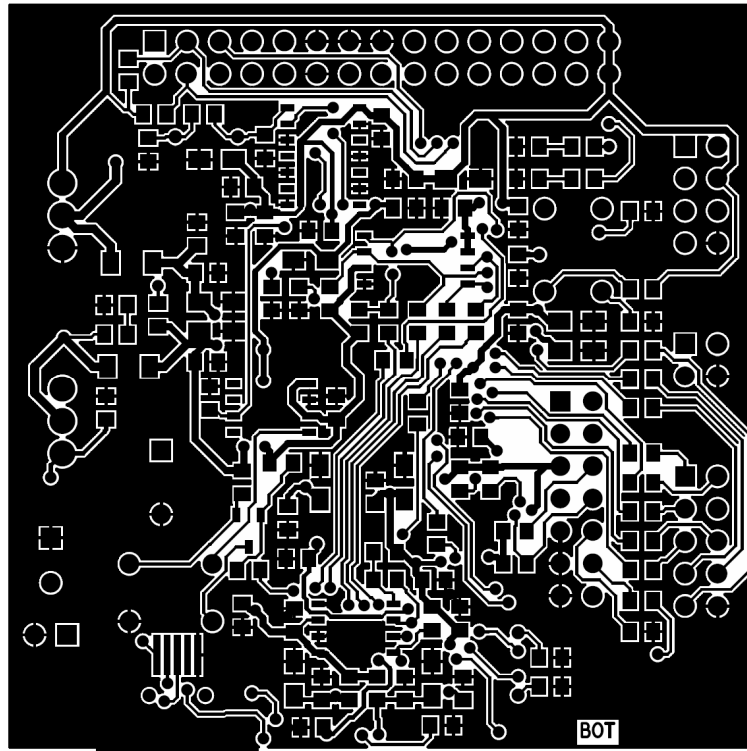
7.3 Příloha C: Výkresy DPS (60 × 60 mm) a osazovací předlohy



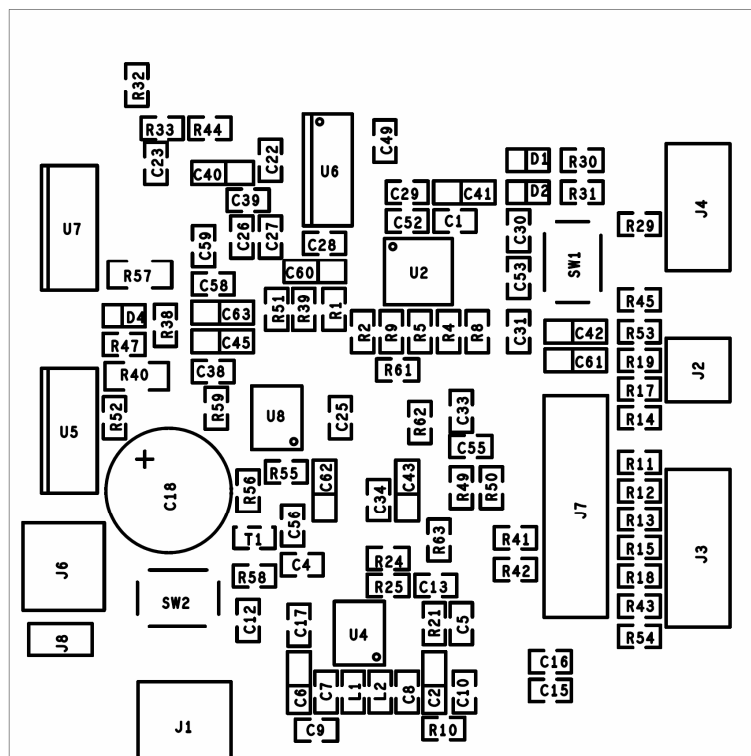
Obr. 7.7 Výkres plošného spoje – pohled shora (TOP)



Obr. 7.8 Osazovací výkres pro vrchní stranu spoje (TOP)



Obr. 7.9 Výkres plošného spoje – pohled zdola (BOTTOM)



Obr. 7.10 Osazovací výkres pro spodní stranu spoje (BOTTOM)

7.4 Příloha D: Soupiska součástek

Reference	Hodnota	Reference	Hodnota	Reference	Hodnota
R1	4k7	R53	10k	C42	1u
R2	10 k	R54	4k7	C43	1u
R3	10 k	R55	47k	C44	1u
R4	470	R56	10k	C45	1u
R5	470	R57	0R0	C46	10u
R6	470	R58	680	C47	1u
R7	470	R59	10k	C48	100n
R8	470	R60	120	C49	100n
R9	470	R61	39	C50	100n
R10	12k	R62	39	C51	47u
R11	39	R63	39	C52	10n
R12	39	C1	100n	C53	10n
R13	39	C2	10u	C54	10n
R14	39	C3	10u	C55	10n
R15	39	C4	100n	C56	10n
R16	1k	C5	100n	C57	10n
R17	39	C6	10u	C58	10n
R18	39	C7	100n	C59	10n
R19	39	C8	100n	C60	1u
R20	10k	C9	100n	C61	1u
R21	10k	C10	100n	C62	1u
R22	470	C11	100n	C63	1u
R23	470	C12	100n	U1	ADSP-BF533
R24	10k	C13	100n	U2	M25P80
R25	2k2	C14	100n	U3	FT2232HL
R26	470	C15	27p	U4	93LC46
R27	470	C16	27p	U5	LM317T
R28	470	C17	100n	U6	74LVC04/SO
R29	10k	C18	220u	U7	LM317T
R30	120	C19	100n	U8	ADM708TAR
R31	100	C20	1u	T1	BC807-40
R32	1k5	C21	47u	D1	LT8A42-43
R33	1k5	C22	22p	D2	LT8A22-43
R34	10k	C23	22p	D3	1N4007
R35	10k	C24	100n	D4	LT8A22-43
R36	4k7	C25	100n	L1	68nH
R37	10k	C26	10n	L2	68nH
R38	10k	C27	10n	Q1	12MHz
R39	10k	C28	10n	Q2	16MHz
R40	0R0	C29	10n	SW1	P-B1720
R41	10k	C30	10n	SW2	P-B1720
R42	10k	C31	10n	J1	USB B MINI
R43	10k	C32	10n	J2	S2Gxx
R44	100k	C33	10n	J3	S2Gxx
R45	100k	C34	10n	J4	S2Gxx
R46	750	C35	10n	J5	S2Gxx
R47	180	C36	10n	J6	ARK550/2EX
R48	10k	C37	10n	J7	S2Gxx
R49	10k	C38	10n	J8	S1Gxx
R50	10k	C39	10n		
R51	10k	C40	1u		
R52	1k2	C41	1u		

Tab. 7.1 Soupiska součástek pro snímač polohy

7.5 Příloha E: Výkres výsledného schématu zapojení snímače polohy na listu A3