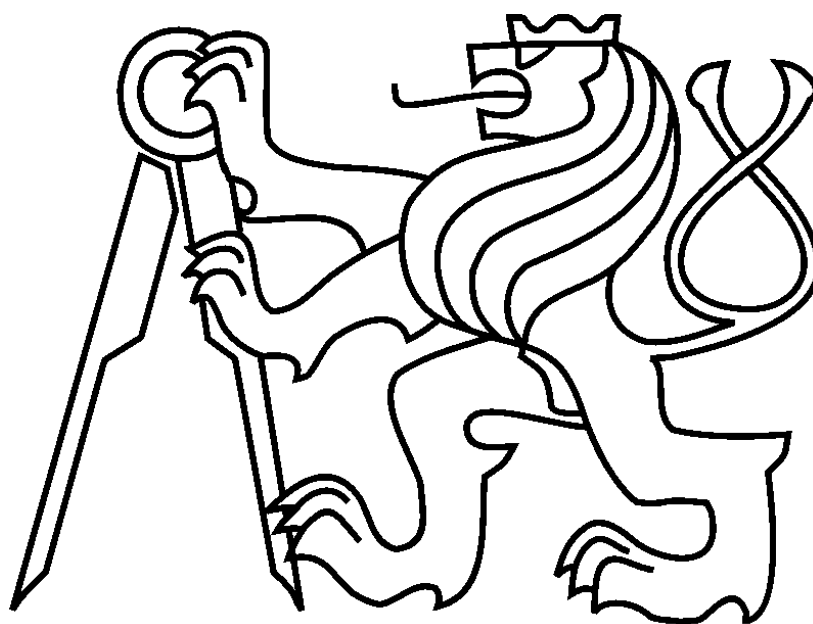


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ

KATEDRA MĚŘENÍ



DIPLOMOVÁ PRÁCE
MNOHOŘÁDKOVÝ VIDEOSENZOR

Praha, 2007

Autor:
Vedoucí práce:

Vojtěch Nádvorník
Ing. Jan Fischer, CSc.

ZADÁNÍ DIPLOMOVÉ PRÁCE

ABSTRAKT

Tato diplomová práce se zabývá návrhem a realizací mnohořádkového videosenzoru pro základní bezdotyková měření rozměrů a polohy. Měření je prováděno na základě zpracování řádkové obrazové informace. Jako zdroj obrazové informace je použit CMOS senzor. Pro účely polohování videosenzoru je přenášén získaný obraz v reálném čase prostřednictvím rozhraní USB 2.0 na nadřazený počítač, který slouží jako zobrazovací monitor.

ABSTRACT

This diploma thesis deals with a development of multi line videosensor for basic contactless size and position measurements. Measurement is based on multi line video information processing. CMOS image sensor is used as a video signal source. Positioning of videosensor is performed using transfer digitalized image to PC in real-time through USB 2.0 interface.

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne

Podpis

OBSAH

1 Úvod a cíl práce	1
1.1 Úvod do řešené problematiky	1
1.1.1 Řádkový (jednorozměrný) a plošný (dvourozměrný) snímač.....	2
1.1.2 Typy snímačů (CDD a CMOS snímač).....	2
1.1.3 Současné videometrické metody pro měření rozměru	3
1.1.4 Víceřádkový videosenzor pro měření rozměru	7
1.1.5 Motivace pro realizaci víceřádkového senzoru	10
1.2 Cíl práce	10
2 Rozbor řešeného úkolu	12
2.1 Aplikační příklady.....	12
2.1.1 Aplikační příklad - kontrola odizolování vodičů.....	12
2.1.2 Aplikační příklad - kontrola lahví na dopravníku.....	13
2.2 Analýza požadavků na videosenzor.....	14
2.2.1 Konfigurační režim	14
2.2.2 Režim měření	18
2.2.3 Vstupy a výstupy videosenzoru - technické požadavky	21
2.3 Problematika „ON-LINE“ zpracování obrazových dat	23
2.3.1 Hranová detekce.....	24
2.3.2 Prostředky pro zpracování obrazových dat	26
2.4 Zvolené řešení videosenzoru	28
2.4.1 Ideové blokové schéma	28
2.4.2 Stavební kameny	29
3 Popis realizace videosenzoru.....	31
3.1 Blokové schéma videosenzoru.....	31
3.2 Seznam významných signálů, jejich významy a propojení	33
3.3 Popis elektrického zapojení.....	34
3.3.1 Napájení systému	34
3.3.2 Multiplexovaný UART	34
3.4 Přenos obrazu do nadřazeného PC.....	35
3.5 Hranový detektor	39
3.5.1 Hranový detektor - komparační metoda.....	39
3.5.2 Hranový detektor - diferenční metoda.....	43
3.5.3 Srovnání metod z hlediska implementace v CPLD.....	49
3.5.4 Parametrizace bitové šířky diferenčního detektoru	51
3.6 Metody měření pozice hrany	51
3.6.1 Zachycení hrany s využitím CPLD.....	52
3.6.2 Zachycení hrany pomocí „capture“ registru.....	53
3.7 Detekce měřicího řádku.....	54
3.8 Detekce objektu v obraze - preprocessing hran	56
3.8.1 Příkaz FIT1	57
3.8.2 Příkaz FIT2	57
3.8.3 Příkaz FIT3	58
4 Výsledky práce, měření parametrů videosenzoru	59
4.1 Provedená měření.....	59
4.1.1 Zachycení hrany - měření odběru dvou blízkých hran.....	59
4.1.2 Převodní charakteristika videosenzoru při měření polohy objektu	60
4.2 Zhodnocení výsledků a funkčnosti.....	62
4.2.1 Detekce a zachycení hrany	62
4.2.2 Detekce objektu.....	62

4.2.3 Dokumentace zdrojových kódů pro mikroprocesor ARM7.....	64
5 Chyby , náměty a poznámky.....	66
5.1 Seznam chyb a navržené způsoby řešení.....	66
5.2 Náměty pro další vývoj videosenzoru	66
5.3 Shrnutí poznatků a „triků“ získaných v rámci vývoje.....	67
6 Závěr.....	68
7 Seznam obrázků.....	71
8 Seznam tabulek	72
9 Seznam zkratk a cizích slov	73
10 Seznam použité literatury	74
11 Seznam příloh.....	75
12 Příloha A.....	i
13 Příloha B.....	viii
14 Příloha C.....	xv
15 Příloha D.....	xvii
16 Příloha E.....	xx
17 Příloha E.....	xxiii

1 ÚVOD A CÍL PRÁCE

Otázka po smysluplnosti času věnovaném činnosti provází lidskou společnost generacemi. Je to klíčový faktor ovlivňující motivaci, sílu překonávat problémy a touhu dorazit k cíli. Rád bych také tuto práci začal rozvahou nad motivem, který vede k vývoji mnohořádkového videosenzoru.

Prakticky ve všech průmyslových odvětvích se dnes setkáváme s potřebou určení rozměru, polohy a dalších parametrů objektu. Měření těchto veličin můžeme rozdělit na dvě hlavní skupiny: kontaktní a bezkontaktní metody měření. Hlavními výhodami bezkontaktních metod je neovlivňování měřeného objektu, možnost měření pohybujících se objektů, možnost měření objektů tam, kde by hrozilo zničení senzoru (např. vysoká teplota).

Videometrie je obor zabývající se měřením fyzikálních veličin z obrazu. Název pochází z latinských slov „video“ (vidět) a „metria“ (měření). Pod pojmem „videosenzor pro měření polohy a rozměru“ pak budeme rozumět senzor (funkční prvek tvořící vstupní blok měřicího řetězce), který je schopen opticky změřit požadovanou veličinu. Výstup videosenzoru pak může být přímo navázán na systém, pro který je videosenzor vstupem měřené veličiny, bez nutnosti účasti nadřazeného systému (např. PC). Měřenou veličinou pak může být přítomnost, poloha, rozměr objektu a další.

Tento pohled pak nabízí široké aplikační možnosti, protože zapadá do zažitého způsobu řízení, regulace a měření a směřuje k použití videosenzoru jako autonomního prvku – senzoru, který je vstupem pro řídicí systém.

1.1 ÚVOD DO ŘEŠENÉ PROBLEMATIKY

Velmi důležitým parametrem, který je klíčovým kritériem pro aplikaci senzoru v praxi, je „rychlost odezvy“ (reakční doba senzoru na změnu v jeho obrazovém poli). Tento parametr je rozhodující, protože přímo ovlivňuje dynamiku řízení procesu. Chceme-li řídit proces vysokou dynamikou (např. pohybující se objekty), je nutné použít videosenzor s nízkou reakční dobou. Tato doba je přímo úměrná výpočetnímu výkonu jádra videosenzoru, ale také množství obrazové informace, které se rozhodneme pro měření zpracovávat.

1.1.1 ŘÁDKOVÝ (JEDNOROZMĚRNÝ) A PLOŠNÝ (DVOUROZMĚRNÝ) SNÍMAČ

Pod pojmem jednořádkový snímač budeme rozumět snímač s jediným fotocitlivým řádkem, který se skládá z mnoha obrazových elementů (pixelů). Obvyklý počet se pohybuje od 256 až do řádově 10 000 pixelů. Počet pixelů určuje rozlišovací schopnost senzoru a tím i přesnost měření. Rozměry jednotlivých pixelů se pohybují v řádu jednotek a desítek μm .

Silnou stránkou jednořádkových snímačů je vysoká opakovací frekvence měření¹. Protože se jedná o jednořádkový snímač, bude opakovací frekvence měření odpovídat řádkové frekvenci² (řádově tak opakovací frekvence měření dosahuje desítek kHz).

Plošný snímač obrazu se skládá z obdélníkové matice fotocitlivých elementů. Rozlišení těchto snímačů se pohybuje v širokých mezích (až jednotky milionů pixelů). Snímková frekvence³ se pohybuje v oblasti desítek až stovek Hz. Ovšem v tomto případě (jedná se zpracování velkého množství dat) již není zanedbatelný čas potřebný k zpracování obrazové informace a získání finálního výsledku.

Plošný snímač tedy nabízí široké možnosti zpracování obrazové informace a složitějšího vyhodnocení. Cenou za to ovšem bude vyšší reakční doba.

1.1.2 TYPY SNÍMAČŮ (CDD A CMOS SNÍMAČ)

Pod pojmem snímač budeme rozumět fotoelektrický převodník složený z mnoha fotodetektorů (fotoelementů), které jsou integrované na jednom polovodičovém čipu. Snímání obrazu je pak prováděno čtením informace z jednotlivých fotoelementů - tím je získána informace o snímané scéně. V současné době se na trhu vyskytují dva základní typy snímačů: CCD snímač a CMOS snímač.

Jako fotodetektory CCD snímače se používají MOS kapacity nebo fotodiody s PN přechodem. Dopadající fotony způsobí akumulování náboje ve fotodetektoru, který je po skončení daného časového intervalu převeden pomocí CCD

¹ Opakovací frekvence měření - odpovídá době potřebné k vyčtení všech pixelů

² Řádková frekvence - odpovídá době potřebné k vyčtení jednoho řádku (u jednořádkových snímačů odpovídá snímkové frekvenci)

³ Snímková frekvence - odpovídá počtu snímků vyčtených ze snímače za jednu sekundu

posuvného registru na výstup snímače. Zde je náboj převeden na napětí. Pokud je signál dále zpracováván digitálně, je tedy potřeba provést jeho digitalizaci pomocí AD převodníku.

Integrované CMOS optoelektrické snímače využívají dnes vyspělé CMOS technologie. Jako fotoelement je použita fotodioda pracující ve třetím kvadrantu. U CMOS snímače jsou fotoelementy uspořádány do maticové struktury (jedná se tedy o plošný snímač) propojené vertikálními a horizontálními vodiči podobně jako u polovodičových paměťových obvodů. Díky CMOS technologii vysoké integrace mohou být na čipu integrovány generátory řídicích impulzů, zesilovače, AD převodníky.

Výhodou CMOS snímačů je také nízké napájecí napětí, což umožňuje snadnou integraci. Oproti CCD snímačům mají ovšem větší šum.

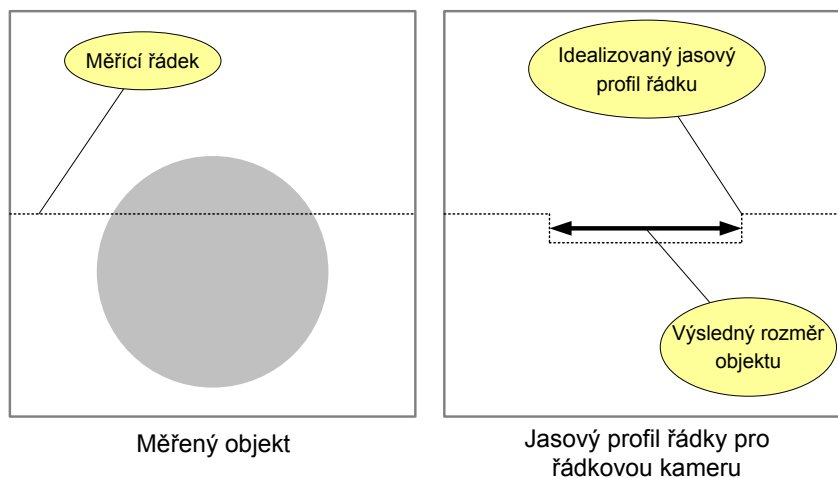
1.1.3 SOUČASNÉ VIDEOMETRICKÉ METODY PRO MĚŘENÍ ROZMĚRU

Zadání této práce je navrhnout videosenzor pro měření polohy a rozměru objektů. Zkusme tedy zhodnotit současné videometrické systémy na trhu při použití pro měření rozměru, kdy měřítkem nám bude reakční doba a potřebný výpočetní výkon (resp. tedy cena).

1.1.3.1 ŘÁDKOVÁ KAMERA + JEDNOŘÁDKOVÉ ZPRACOVÁNÍ OBRAZU

Řádkovou kamerou budeme rozumět systém s jednořádkovým snímačem a optickou projekční soustavou. Získáváme tedy obrazový jednořádkový jasový profil měřené scény. Tento jasový profil je dále zpracován tak, že jsou v něm detekovány hrany, jejichž vzdálenost určuje rozměr objektu.

Jak již bylo řečeno v popisu jednořádkového snímače, výhodou je vysoká řádková frekvence, které je dosaženo výrazně nižšímu objemu zpracovávaných obrazových dat oproti plošným snímačům. Daní za nízkou reakční dobu senzoru pak ovšem bude nízká informační hodnota, kterou je možné z jednořádkové obrazové informace získat. Situaci je možné dobře ilustrovat na příkladu měření průměru kruhového objektu (obr. 1.1).

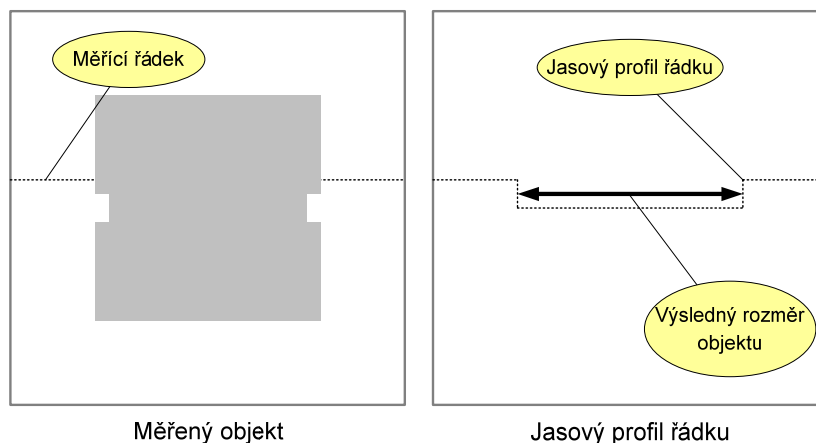


Obr. 1.1 Měření rozměru kruhového objektu řádkovou kamerou⁴

Jestliže nebudeme schopni zajistit definovanou polohu objektu v okamžiku měření, nejsme z jediného řádku schopni určit pozici, ve které byl objekt změřen. Ve výsledku tedy získaný rozměr nemá žádnou vypovídací hodnotu o průměru objektu.

Dalším problémem řádkových kamer je komplikované polohování, protože s pouze jediným řádkem je výrazně obtížnější zaměření kamery na požadované místo. Situaci ilustruje obr. 1.2.

⁴ Zobrazená scéna na obr. 1.1 odpovídá zobrazení objektu po promítnutí optickou projekční soustavou na snímač. Protože je použit jednořádkový snímač, získáváme pouze jediný jasový profil řádku v místě fotocitlivých elementů. Stejný způsob zobrazení bude použit i u následujících obrázků podobného charakteru.



Obr. 1.2 Polohování řádkové kamery

Úkolem v uvedeném příkladě na obr. 1.2 je určit rozměr zúženého místa na profilu objektu. Je tedy nutné zaměřit kameru tak, aby měřicí řádek řádkové kamery byl nejlépe uprostřed zúžené oblasti. U řádkové kamery je však zaměření na toto zúžené místo komplikované, protože není možné zobrazit celý objekt a poté vybrat místo měření. Jediným možným způsobem, jak zjistit, kam je kamera zaměřena, je jasový profil změřené řádky.

Řádkové kamery nalézají široké uplatnění v aplikacích pro měření rozměru a ve scannerech. Výrobou těchto systémů se zabývají např. firmy Basler [6], Schäfter + Kirchhoff [10], Dalsa [12].

1.1.3.2 SYSTÉM PC + FRAMEGRABBER

Jedná se o kameru s plošným snímačem obrazu, kde framegrabber digitalizuje analogový video signál z kamery. Získaná obrazová data jsou dále zpracovávána ve výkonném PC pomocí specializovaného software.

Analogickým řešením k analogové kameře s framegraberem je v současnosti velmi často používaná sestava digitální kamera připojená k PC (používaná rozhraní: Fire wire, Ethernet, USB 2.0).

Současné programové balíky pro zpracování obrazových dat nabízejí široké možnosti využití a velké množství metod zpracování obrazu. Pro příklad uvedeme několik z mnoha nabízených metod:

- Rozpoznávání objektů a třídění do tříd
- Měření plochy objektů
- Kontrola povrchu

- Rozpoznávání textu (OCR⁵)
- Rozpoznání vzoru (Pattern match)

Vzhledem k tomu, že je v kameře (ať analogové či digitální) použit plošný snímač, bude dosaženo v porovnání se systémem „řádková kamera + jednořádkové zpracování obrazu“ výrazně nižší obnovovací frekvenci měření.

Dále je třeba uvažovat i čas potřebný na zpracování pomocí PC, protože se jedná o operace prováděné nad velkým množstvím dat. Pokud tedy budeme tento systém hodnotit z pohledu použití jako videosenzor pro měření rozměru, můžeme konstatovat, že reakční doba narůstá do řádu stovek milisekund až sekund, a tím se tento systém stává nepoužitelný pro řízení procesů s vyšší dynamikou, kde je požadována nízká reakční doba senzoru.

1.1.3.3 INTELIGENTNÍ KAMERA

Pod pojmem „inteligentní kamera“ rozumíme videometrický systém s opět plošným snímačem obrazu. Zpracování obrazových dat je ovšem prováděno přímo v kameře. Kamera je tedy autonomní a nepotřebuje ke své činnosti nadřazené PC a tím se více přibližuje potřebám průmyslu.

Rozdílem oproti systému „PC+ framegrabber“ je to, že kamera obsahuje interface, který umožňuje její přímé připojení do řídicího systému (dnes nejčastěji programovatelné automaty PLC). Tím se velmi přibližuje klasickému pojetí senzoru v řídicích systémech. Zpracování obrazových dat je opět časově náročné a klade velké výpočetní nároky na systém. Prováděno je pomocí zákaznických obvodů (ASIC), častěji však hradlových polí či signálových procesorů.

Výrobou těchto systémů se zabývají např. firmy Basler [6] a Keyence [12].

Výpočetní prostředky nasazované v těchto systémech se svým vysokým výkonem vyrovnávají systémům, kde jsou obrazová data zpracovávána v PC. Standardem se dnes již stávají i náročné metody jako porovnání vzoru s předchozí rotací a změnou měřítko.

1.1.3.4 SHRNUÍ

Inteligentní kamera i systém „framegrabber + PC“ nabízí kvalitní prostředky pro zpracování obrazu. Jedná se o velmi dobře a široce konfigurovatelný vi-

⁵ OCR - Optical Character Recognition

deometrický systém, který umožňuje mnoho různorodých metod zpracování. Potřebný výpočetní výkon se samozřejmě promítá v ceně. Vzhledem k použití plošného snímače obrazu se snímková frekvence pohybuje v řádu desítek hertzů. Můžeme tedy shrnout, že tyto systémy pro měření rozměru samozřejmě vyhoví, ale díky jejich univerzálnosti nebude jejich výkonnost využita pro jednoduché úlohy, jako je měření rozměru, efektivně a reakční doba bude díky nízké snímkové frekvenci vyšší.

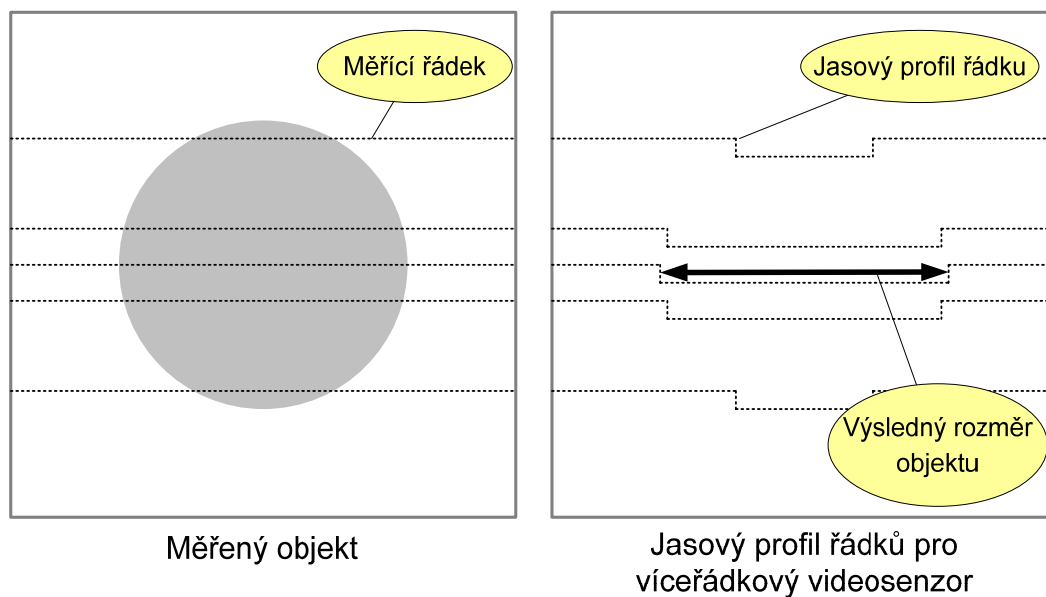
Videometrický systém „řádková kamera + jednořádkové zpracování obrazu“ je v porovnání s předchozími systémy naopak primárně používán pro měření rozměru. Metody použité na zpracování jednořádkových obrazových dat jsou výrazně méně výpočetně náročné a snímková frekvence díky použití jednořádkového senzoru dosahuje řádově desítek kHz. Nevýhodou tohoto řešení pro měření rozměru a polohy je nemožnost ověření, zda je naměřená veličina měřena na správném objektu, resp. na správném místě objektu, protože z jediného obrazového řádku je problematické toto ověření provést. Dalším problémem tohoto řešení je obtížné polohování kamery.

Jestliže tedy shrneme uvedená fakta, ukazuje se, že současná řešení nabízejí dvě krajní varianty, ovšem chybí tu jakási střední cesta pro úlohy, kde požadujeme prosté měření rozměru, ovšem rádi bychom využili část výhod, které nabízejí plošné snímače.

1.1.4 VÍCEŘÁDKOVÝ VIDEOSENZOR PRO MĚŘENÍ ROZMĚRU

Víceřádkový videosenzor pro měření rozměru můžeme chápat jako systém s plošným snímačem obrazu, z kterého se rozhodneme zpracovávat pouze vybrané řádky. Řádky jsou vybrány tak, abychom získali obrazovou informaci, která nám umožní určit požadovanou měřenou veličinu, a zároveň byla zajištěna spolehlivá detekce objektu⁶. Tyto řádky pak budou nejčastěji lineárně rozmístěny po měřeném objektu, s možností hustšího pokrytí oblasti měřeného rozměru (viz obr. 1.3).

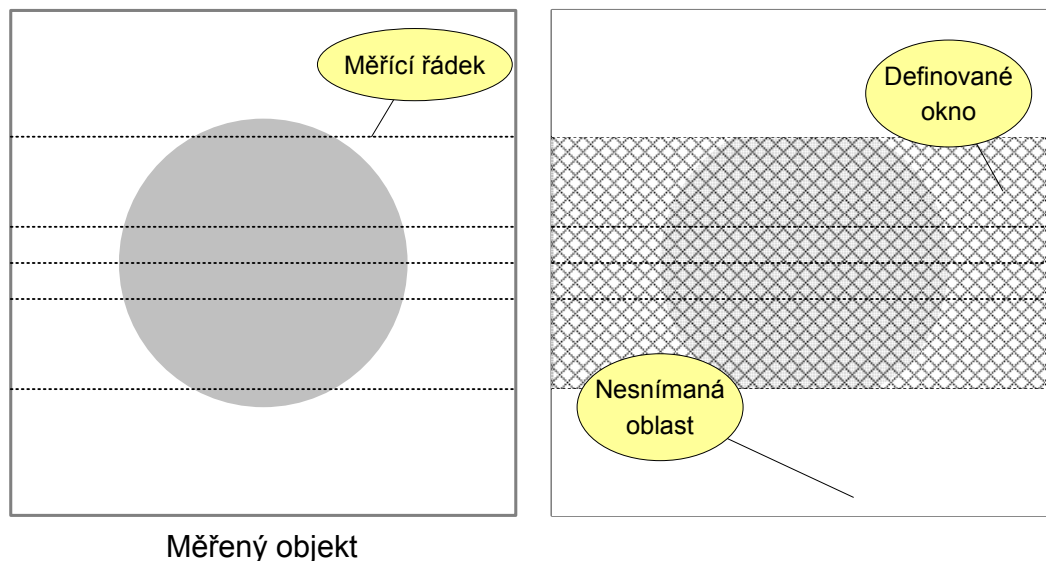
⁶ Spolehlivost detekce objektu – jedná se o změřený požadovaný rozměr a zároveň ověření, zda je měřeno na správném objektu a na správném místě objektu.



Obr. 1.3 Měření rozměru kruhového objektu víceřádkovým videosenzorem

Analyzujme nyní vlastnosti tohoto řešení:

- Výrazně větší spolehlivost detekce objektu (je zde velmi jasná vzájemná vazba mezi jasovými profily jednotlivých řádků). Tak je možné ověřit, že dané měření je provedeno na požadovaném objektu.
- Řádkové zpracování obrazových dat: jestliže snížíme požadavky na univerzálnost systému a zaměříme se na zpracování po řádcích (data jsou tedy analyzována pouze v horizontálním směru), nabízí se díky velmi specifickému zadání možnost použití značně efektivních metod. Díky tomu jsou opět výrazně sníženy výpočetní nároky na systém. Jedná se tedy o zúžení oboru úloh s cílem dosáhnout vyšší efektivity zpracování.
- Reakční doba senzoru: jestliže se rozhodneme použít rozložení řádků dle příkladu uvedeného na obr. 1.3, mohli bychom přepokládat, že dosáhneme 5x nižší reakční doby než v případě použití jednořádkového senzoru o stejném rozlišení. Ovšem běžně dostupné obrazové CMOS senzory neumožňují konfigurovat senzor tak, aby poskytoval pouze data z vybraných pěti řádků. Možná je konfigurace tzv. okna (windowing).



Obr. 1.4 Windowing

Pro příklad z obr. 1.3 bychom okno v horizontálním směru nastavili na šířku senzoru. Ve vertikálním směru by okno začínalo s prvním horním řádkem a končilo s posledním. Budeme-li chtít určit snímkovou frekvenci pro tyto parametry: $f_{\text{PIX}}=40\text{MHz}$, rozlišení senzoru: 648×488 , pozice prvního řádku: 300, pozice posledního řádku 350, výpočet provedeme následovně:

$$\text{snímková frekvence} = \frac{\text{pixel frekvence}}{H * \text{pocet radku v okne}} = \frac{12\text{MHz}}{648 * 25} = 740\text{Hz} \quad (1.1)$$

Pro plošný snímač obrazu získáme:

$$\text{snímková frekvence} = \frac{\text{pixel frekvence}}{H * V} = \frac{12\text{MHz}}{648 * 488} = 38\text{Hz} \quad (1.2)$$

Z výsledků je patrný rozdíl v možnosti použití senzoru pro řízení procesů s vyšší dynamikou.

- Možnost přejít k režimu „plošný snímač obrazu“: pro realizaci víceřádkového videosenzoru je fyzicky použit plošný obrazový senzor, proto je možné ve fázi nastavování optické soustavy a polohování přejít do režimu plošný snímač obrazu a tím uživateli umožnit zobrazení celé optické scény.
- Nižší výpočetní nároky na videosenzor: výpočetní zatížení videosenzoru roste s množstvím dat. Zpracování pouze několika obrazových řádků výrazně

sníží výpočetní nároky a zároveň nabízí nenáročné metody, které mohou pro měřicí účely poskytnout dostatečné a spolehlivé výsledky.

Víceřádkový videosenzor tedy nabízí nezanedbatelné množství dobrých vlastností, které dávají dobrou pozici pro použití v technické praxi. Jedná se o jakousi „zlatou střední cestu“ mezi řádkovým a plošným snímačem obrazu. Víceřádkový senzor kombinuje dobré vlastnosti krajních cest, na druhou stranu však není schopen konkurovat reakční době jednořádkových senzorů ani širokým možnostem zpracování obrazu pomocí plošných snímačů obrazu.

1.1.5 MOTIVACE PRO REALIZACI VÍCEŘÁDKOVÉHO SENZORU

V tomto odstavci stručně shrneme výhody a inovativnost tohoto řešení z předchozí kapitoly:

- Plošný snímač: pro polohování možnost zobrazení celé scény.
- Zvýšení snímkové frekvence: při měření je možné konfigurovat plošný snímač tak, aby snímal pouze oblast, která bude pokryta měřicími řádky.
- Snížení výpočetních nároků:
 - Řádkové zpracování dat.
 - Snížení objemu zpracovávaných obrazových dat.
- Zvýšení spolehlivosti detekce objektu oproti řádkové kameře.

1.2 CÍL PRÁCE

Cílem je navrhnout mnohořádkový videosenzor pro měření rozměru s orientací na řádkové zpracování obrazových dat. Vývoj jednoznačně směřuje k uplatnění v průmyslové automatizaci, snahou tedy je přiblížit se funkcí běžnému senzoru tak, aby se začlenění videosenzoru do řídicího systému (po skončení fáze konfigurace a polohování) nelišilo od běžného senzoru.

Následuje výčet atributů, k jejichž naplnění bude vývoj videosenzoru směřovat.

- Autonomnost: ve fázi měření videosenzor pracuje nezávisle na nadřazeném konfiguračním nástroji.
- Výstupy videosenzoru: snahou je volit výstupy takové, aby umožňovaly jednoduché začlenění do řídicího systému.
- Snadná konfigurace: důležitým prvkem pro uplatnění na trhu je, aby konfigurační nástroje byly tzv. „User Friendly“.

- V konfiguračním a polohovacím režimu bude možné přenášet celkový obraz na nadřazený počítač, což usnadní konfiguraci a polohování optické soustavy.
- Univerzálnost: snahou je, aby aplikační vybavení videosenzoru pokrývalo co největší část základních úloh měření polohy, rozměru.
- Minimalizace reakční doby videosenzoru.
- Návrh s ohledem na průmyslové prostředí.

2 ROZBOR ŘEŠENÉHO ÚKOLU

2.1 APLIKAČNÍ PŘÍKLADY

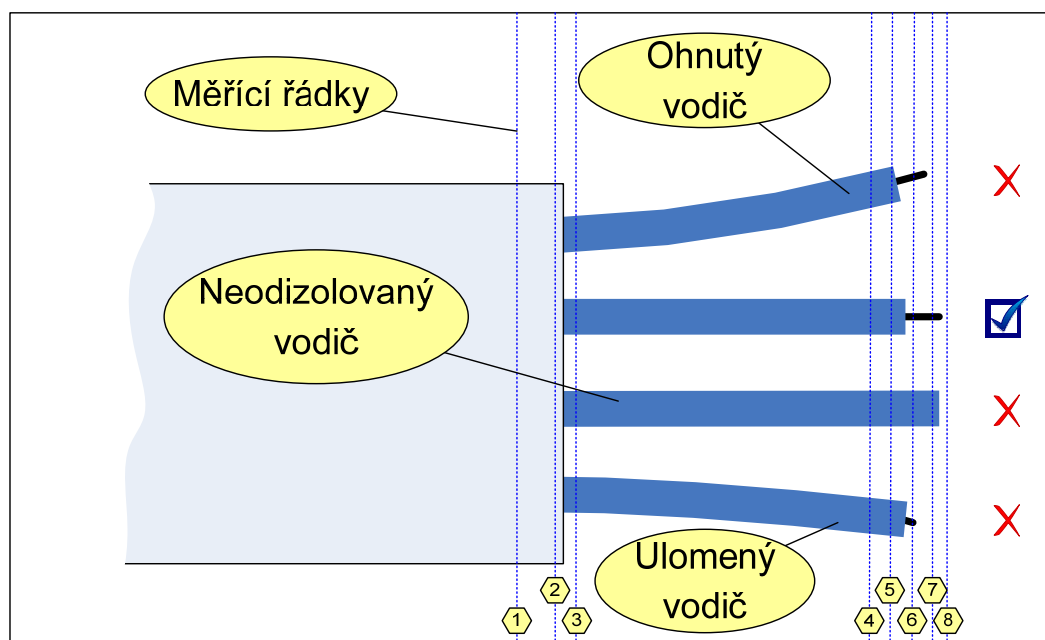
Následující tabulka (tab. 2.1) uvádí seznam základních typů měření, ke kterým je možné použít víceřádkový videosenzor.

STATICKÁ MĚŘENÍ	DYNAMICKÁ MĚŘENÍ
Přítomnost objektu	Rychlost objektu
Rozměr objektu	Změna rozměru
Poloha objektu	Změna polohy
Vzájemná poloha více objektů	Určení trajektorie
Měření úhlu natočení	

Tab. 2.1 Aplikační příklady měření videosenzorem

2.1.1 APLIKAČNÍ PŘÍKLAD – KONTROLA ODIZOLOVÁNÍ VODIČŮ

Na obr. 2.1 je zobrazen aplikační příklad použití víceřádkového videosenzoru při kontrole odizolování vodičů před nalisováním konektoru. Cílem této aplikace videosenzoru je zajistit kontrolu před nalisováním konektoru.



Obr. 2.1 Aplikační příklad – kontrola odizolování vodičů

Z uvedeného příkladu je zřejmé, že v této konfiguraci rozložení měřících řádků je možné víceřádkovým videosenzorem kontrolovat zároveň několik parametrů. Tímto způsobem je možné detekovat chyby vzniklé při odizolování kabelu a zamezit nalisování konektoru v případě, že proces odizolování nebyl

proveden bezchybně. Vyhodnocení jasových profilů měřících linek bude provedeno následujícím způsobem:

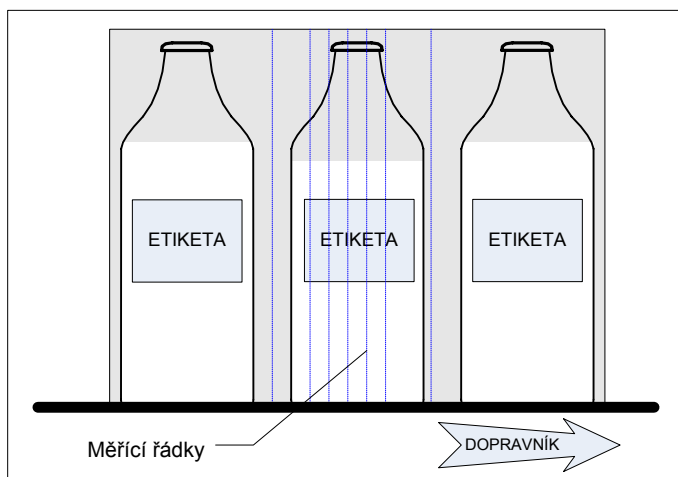
- Pomocí měřících řádků „2“ a „3“ je detekováno odizolování vnějšího pláště kabelu a tím i referenční rovina pro měření délky vodičů.
- Měřící řádky „5“ a „6“ detekují odizolování pláště jednotlivých vodičů.
- Měřící řádky „6“ a „7“ definují délku odizolovaného vodiče.
- Vzdáleností řádků „7“ a „8“ je určena tolerance celkové délky vodiče.

Tímto způsobem zpracování je tedy možné určit následující parametry:

- Správná délka odizolování.
- Přítomnost a správná poloha všech vodičů .
- Kontrola zkroucení.

Tato aplikace přesně zapadá do systému měření kvality výrobních řetězců v procesu řízení jakosti.

2.1.2 APLIKAČNÍ PŘÍKLAD – KONTROLA LAHVÍ NA DOPRAVNÍKU



Obr. 2.2 Aplikační příklad – kontrola lahví na dopravníku

Konfigurace na obr. 2.2 demonstruje použití víceřádkového videosenzoru na pásovém dopravníku. Měření umožňuje např.:

- Detekci množství kapaliny v láhvi.
- Kontrolu usazení uzávěru.
- Kontrolu etikety láhve.
- Počítání lahví na dopravníku.

Uvedené aplikační příklady mají velmi specifické rysy a vyžadují specializovaný software videosenzoru. Pokud však provedeme rozbor měření, zjistíme, že úlohy se skládají z charakteristických stavebních kamenů, které možné je zobecnit a tím dosáhnout jisté univerzálnosti videosenzoru. Tento rozbor provedeme v následující kapitole 2.2.

2.2 ANALÝZA POŽADAVKŮ NA VIDEOSENZOR

V této kapitole bude provedena analýza požadavků na videosenzor s ohledem na výčet jednotlivých atributů, které byly uvedeny v kapitole 1.2. Celou problematiku můžeme rozložit do dvou fází: „konfigurační režim“ a „režim měření“.

2.2.1 KONFIGURAČNÍ REŽIM

Videosenzor pracující v konfiguračním režimu uživateli umožní nastavit optickou soustavu, způsob lokalizace objektu, vlastnosti měření, jednotlivé parametry, které mají být změřeny, a chování výstupu senzoru v závislosti na změřených parametrech. Už z množství parametrů je zřejmé, že v případě nastavení senzoru se nejedná o jednoduchou proceduru, a proto bude snahou nabídnout uživateli nástroj, který ho celou procedurou provede a nabídne mu rychlé, průhledné a komfortní řešení. Takovým nástrojem je možnost uvedení senzoru do konfiguračního režimu, která spolu s příslušným programem na PC umožní přehledně provést tuto konfiguraci.

2.2.1.1 KONFIGURAČNÍ REŽIM - NASTAVENÍ OPTICKÉ SOUSTAVY

Jedná se o první krok při zprovoznění senzoru. Je třeba zajistit vhodné osvětlení snímaného objektu, nasměrování videosenzoru na objekt, nastavení videosenzoru tak, aby měřený rozměr byl ve směru řádku kamery, zaostření objektivu.

Je zřejmé, že nutnou podmínkou pro tuto fázi je možnost uživatele vidět celý obraz (ne tedy vybrané řádky určené pro měření), aby mohl kontrolovat celou scénu, nasměrovat správně zdroj světla, zaostřit. K tomuto účelu bude tedy po celou dobu konfigurace videosenzor připojen k PC, které zajistí vizualizaci a provede uživatele jednotlivými kroky. Vzhledem k tomu, že se předpokládá přenos celého obrazu nekomprimovaným způsobem, můžeme určit potřebný datový tok:

$$N_x \times N_y \times f_{FRAME} = 640 \times 480 \times 25 = 7,32 MBs^{-1} \quad (2.1)$$

Uvedená hodnota datového toku sice odpovídá 15% z maximální přenosové rychlosti USB 2.0 (HS), ale i přes to se jedná o značný datový tok, který klade nároky na výkon PC při zobrazení dat.

2.2.1.2 KONFIGURAČNÍ REŽIM - TRIGGER MĚŘENÍ

Jestliže porovnáme aplikační příklady „pásový dopravník“ (obr. 2.2) a „kontrolu odizolování vodičů“ (obr. 2.1), shledáme zásadní rozdíly v přístupu k vyvolání jednotlivého měření - tzv. „TRIGGER MĚŘENÍ.“

U optické kontroly očekáváme následující postup:

- Nadřazený automat provede odizolování, případně přesune odizolovaný kabel na definované místo v zorném poli videosenzoru.
- Automat vydá požadavek videosenzoru na kontrolu odizolování.
- Videosenzor provede jediný odměr a prezentuje výsledek měření (odizolování pořádku / chyba).
- Automat reaguje na výsledek měření.

U pásového dopravníku je situace složitější. Chceme-li například nakonfigurovat videosenzor pro kontrolu etiket, jsme postaveni před problém určení okamžiku, kdy odměr provést (optimální situace nastává ve chvíli, kdy je láhev uprostřed zorného pole videosenzoru). Úlohu můžeme tedy rozdělit následujícím způsobem:

- Detekce průchodu láhve zorným polem videosenzoru.
- Vyvolání odměru ve vhodném okamžiku + nastavení měřících řádek do správné aktuální pozice měřeného objektu.
- Prezentování výsledku.

V prvním případě „optické kontroly“ jsme tedy získali požadavek pro start měření od nadřazeného systému (tzv. „externí trigger“). V případě druhém je událost odměru vyvolána automaticky ve videosenzoru na základě měnící se scény. Pokusme se nyní na základě aplikačních příkladů shrnout možné způsoby vyvolání odměru:

NÁZEV TRIGGERU	POPIS
Externí trigger	Odměr je proveden na základě žádosti z nadřazeného systému.
Periodický trigger	Odměr je proveden opakovaně s definovanou periodou.
Trigger objektem	Odměr je proveden jedenkrát pro každý objekt procházející scénou v optimální okamžik. Výstupem tohoto triggeru je zároveň také informace o aktuální poloze objektu, který trigger vyvolal (tzv. „LOKALIZACE“).
FREE RUN	Měření je prováděno opakovaně s maximální frekvencí – vhodné například pro regulaci polohy.
Trigger odměrem	Jestliže výsledek měření splňuje triggerovací podmínku (např. objekt není v pořádku), je např. provedeno nové detailnější měření, které má za cíl detekovat typ vady.

Tab. 2.2 Možné způsoby vyvolání odměru – TRIGGERy

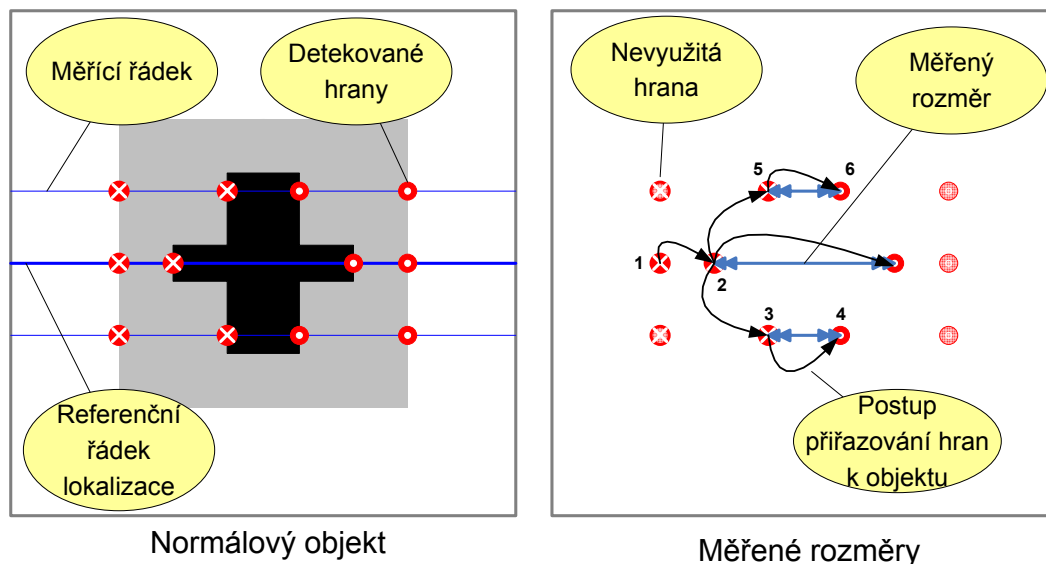
2.2.1.3 KONFIGURAČNÍ REŽIM – VLASTNOSTI MĚŘENÍ A JEDNOTLIVÉ PARAMETRY

Výstupem této fáze by měla být datová struktura, která bude obsahovat informace nutné pro změření objektu:

- Rozložení měřících řádků na lokalizovaném objektu: ne tedy absolutně vzhledem k zornému poli videosenzoru, ale relativně vztažené k lokalizovanému objektu.
- Nastavení hranového detektoru.
- Tolerance rozměrů normálového objektu.
- Měřené rozměry: definice způsobu určení hodnot, které budou vstupovat do bloku výstupů.
- Meze měřených parametrů pro validaci měřeného objektu (častým požadavkem na výstup měření je: „objekt v pořádku ANO/NE“).
- Referenční rozměr, který umožní přepočítání získaných rozměrů v pixelech do reálných jednotek.
- Rozměry „normálového objektu“: pro každý měřený parametr jsou uvedeny očekávané rozměry a polohy jednotlivých jasových profilů měřících řádků vůči sobě navzájem. Tím je definován „normálový objekt“.

Metodika detekce měřeného objektu na základě znalosti „normálového objektu“ je ilustrována na obr. 2.3. První referenční hrana je např. nalezena (přiřazena objektu) na základě znalosti rozměru šedého čtverce, tak je zajištěna lokalizace objektu v horizontálním směru. Od této první hrany je již možné nalézt hranu č.2, právě pomocí znalosti vzdálenosti hrany č.1 a č.2 definované v normálovém objektu. Obdobně je poté postupováno při hledání hran č.3 až č.6.

Výsledné rozměry objektu jsou získány jako rozdíl pozic hran přiřazených k objektu.



Obr. 2.3 Grafické znázornění normálového objektu

2.2.1.4 KONFIGURAČNÍ REŽIM – NASTAVENÍ CHOVÁNÍ VÝSTUPU SENZORU

Pro univerzálnost videosenzoru a jeho snadné začlenění do řídicího systému je vhodné, aby bylo možné konfigurovat chování výstupu senzoru v závislosti na změřených parametrech. Typy výstupů budou diskutovány v následující kapitole 2.2.3. Nyní definujme standardní úlohy, které by videosenzor měl být schopný reflektovat na svých výstupech:

- Binární výstup:
 - Měřený objekt prošel zorným polem kamery + čítání objektů.
 - Objekt je/není v pořádku (rozpoznává se pomocí tolerancí zadaných ke každému z měřených rozměrů).
 - Regulace polohy: (1) poloha v pořádku, (2) akční zásah doleva, (3) akční zásah doprava.
 - Žádný objekt není detekován v zorném poli (např. výstup pro regulaci polohy není platný).
- Číselný výstup: z naměřených rozměrů určená veličina popisující objekt v reálných jednotkách.

- Logování naměřených hodnot: měřicí systém + sběr dat.

Obecně tedy můžeme nahlížet na tuto fázi konfigurace, jako na matici popisující způsob promítnutí naměřených výsledků výše uvedené výstupy. V dalším textu budeme tuto matici označovat jako „výstupní matice“.

2.2.2 REŽIM MĚŘENÍ

Po fázi konfigurace přichází okamžik připojení videosenzoru k řídicímu systému. V této fázi bude důležitá minimalizace reakční doby videosenzoru, která v konfiguračním režimu nebyla podstatná, spolehlivost detekce objektu a robustnost celého systému. Od tohoto okamžiku začíná být videosenzor nezávislý na konfiguračním PC a chová se jako běžný senzor. Poskytuje tedy naměřená data nadřazenému řídicímu systému takovým způsobem, jak bylo definováno v konfiguraci.

V předchozí kapitole 2.2.1 jsme definovali různé typy spuštění jednotlivého měření. Provedme nyní podobnou analýzu pro typy měření.

2.2.2.1 MĚŘENÍ - „SPOLEHLIVÝ ROZMĚR OBJEKTU“

Aplikačně nejčastější úlohou víceřádkového videosenzoru, a tedy základním stavením kamenem bude, „spolehlivé měření rozměru objektu“, případně „spolehlivé určení polohy objektu“.

Pojmem „spolehlivé“ rozumějme ne prosté určení rozměru či polohy, ale zároveň verifikování⁷ objektu. Jestliže tedy určí videosenzor polohu objektu a poskytne nadřazenému systému informaci o poloze, je nutné, aby bylo spolehlivě zaručeno, že se tato poskytnutá informace vztahuje opravdu k správnému objektu.

V tomto režimu měření je tedy videosenzor nakonfigurován na mnohořádkový a dle pozice objektu, která je vstupem do měření, jsou rozmístěny měřicí řádky po objektu. Měření můžeme rozdělit na následující kroky:

- Je proveden odměr.
- Z jasových profilů řádek jsou získány hrany.

⁷ Ověření pravosti

- Metodou „Best Fit“⁸ jsou ze získaných hran vybrány ty, které nejlépe odpovídají „normálovému objektu“ (obr. 2.3).
- Dle verifikačních podmínek je určena pravost objektu.
- V pozitivním případě jsou získané výsledky aplikovány na „výstupní matici“.

Uvedli jsme, že vstupem do měření je pozice objektu, která se použije pro rozmístění řádků. Otázka získání této pozice bude diskutována v následující kapitole.

2.2.2.2 MĚŘENÍ - „LOKACE OBJEKTU“

V nezanedbatelném množství aplikací není pevně definována poloha měřeného objektu vůči zornému poli videosenzoru (obr. 2.2 Aplikační příklad – kontrola lahví na dopravníku). Není tedy možné, bez znalosti okamžité polohy objektu, určit řádek, ve kterém se má měřit. Je tedy nutné určit způsob, kterým bude prováděna lokalizace objektu.

Pro model pásového dopravníku je optimální, pokud mohou být měřící řádky ve směru pohybu pásu. Na vzorku měřeného objektu pak uživatel nastaví polohu lokalizačního řádku a meze, v nichž se může měřený rozměr pohybovat.

Zobecníme-li celou situaci, jedná se o porovnání profilu jednoho, případně dvou, změřených řádků s normálovými řádky získanými při konfiguraci – tím je maximalizována opakovací měřící frekvence pro lokalizaci. Pokud je splněna lokalizační podmínka, je vyvolán trigger příslušné události.

Pro proces lokalizace objektu je klíčová minimální reakční doba a jednoznačnost určení polohy objektu. V této fázi bude tedy „plošný snímač obrazu“ konfigurován jako jednořádkový (případně dvouřádkový, kde řádky nebudou příliš vzdáleny) senzor. Zanedbáme-li režii na zpracování dat, můžeme pro lokalizaci (pro běžný videosenzor) určit pro orientaci opakovací frekvenci měření:

$$f_{LOKALIZE} = \frac{f_{PIX}}{N_x} = \frac{12MHz}{1024} = 11,7kHz \quad (2.2)$$

⁸ Metoda „Best Fit“ – metoda kompenzace odchylek. Hrany jsou vybrány takovým způsobem, aby suma absolutních hodnot odchylek byla minimální.

Důležitá je správná volba způsobu lokalizace. Optimální je zvolit specifický (nejčastěji maximální) rozměr objektu včetně jeho tolerancí. Lokalizátor by měl být také schopný detekovat fáze (1) příchod, (2) prezence a (3) odchod objektu z lokalizačního řádku, aby bylo možné správně detekovat průchod objektu přes zorné pole videosenzoru (tím bude zaručeno správné čítání objektů).

2.2.2.3 MĚŘENÍ - „RYCHLÁ REGULACE POLOHY“

U tohoto režimu měření je důležitá reakční doba videosenzoru. Podobně jako v režimu měření „lokalizace“ bude plošný snímač konfigurován jako jedno či dvouřádkový. Pro spolehlivost měření je tedy nezbytný dostatek kontrastu pro správné nalezení hran využitých pro regulaci.

2.2.2.4 MĚŘENÍ - „SPOLEHLIVÁ POLOHA OBJEKTU“

Jako protiklad k měření typu „rychlá regulace polohy“ uvedme regulaci na spolehlivou polohu objektu. Stejně jako v rychlé regulaci dochází k automatickému cyklickému spouštění měření (TRIGGER FREE RUN). Při každém měření ovšem dojde k verifikaci detekovaného objektu, čímž je zajištěna vyšší spolehlivost regulace. Zvýšení spolehlivosti je samozřejmě na úkor opakovací frekvence měření, protože je nezbytné konfigurovat videosenzor jako víceřádkový a zároveň stoupají výpočetní nároky.

V následující tabulce (tab. 2.1) jsou tedy shrnuty identifikované základní přístupy k měření pomocí víceřádkového videosenzoru.

NÁZEV MĚŘENÍ	CMOS ⁹	POPIS
Lokalizace	SL	Jednořádkové, případně dvouřádkové, měření s hranovou detekcí, které bude schopné detekovat a vyhodnocovat přítomnost, průchod objektů a čítání objektů a bude základním triggerem pro víceřádkové „měření rozměru objektu“. TRIGGER: free run
Rychlá regulace polohy	SL	Jednořádkové, případně dvouřádkové, měření s hranovou detekcí s cílem dosáhnout maximální měřicí frekvence. TRIGGER: free run
Spolehlivý rozměr objektu	ML	Víceřádkové měření s hranovou detekcí určené k spolehlivému určení rozměrů objektu. TRIGGER: odměrem (LOKALIZACE)
Spolehlivá poloha objektu	ML	Víceřádkové měření s hranovou detekcí určené k spolehlivému určení polohy objektu. TRIGGER: odměrem (LOKALIZACE)

Tab. 2.3 Základní typy měření pomocí víceřádkového videosenzoru

2.2.3 VSTUPY A VÝSTUPY VIDEOSENZORU – TECHNICKÉ POŽADAVKY

Analýza vstupů a výstupů videosenzoru byla provedena v předchozích kapitolách 2.2.1.2 a 2.2.1.4. Nyní požadavky na vstupy a výstupy doplníme z pohledu hardwarového. Primární kritérium by tedy mělo směřovat k snadnému začlenění do standardních průmyslových řídicích systémů. Budeme tedy vycházet z standardních řešení, které podobné kamerové systémy na trhu používají. Příkladem je [6] a [7]. V této kapitole uvedeme charakteristické parametry pro jednotlivé typy vstupů / výstupů u řešení, s kterými se dnes běžně setkáváme na trhu.

2.2.3.1 DIGITÁLNÍ VSTUPY/VÝSTUPY

Jak již bylo uvedeno, jako digitální vstup videosenzoru bude nejčastěji „externí trigger“ a výstupy budou signalizovat události týkající se detekce průchodu a prezenze objektu a regulace polohy. Současným dominantním řešením průmyslové automatizace jsou PLC¹⁰. Při definování technických požadavků budeme tedy vycházet ze zažitých technických standardů u PLC.

- VSTUPY: galvanické oddělení, jedna společná zem.
 - Definovaná úroveň log. 0: napětí na vstupu 0 ... 2V.

⁹ Způsob konfigurace CMOS senzoru: SL – jednořádkový, ML – víceřádkový, FF – plošný snímač obrazu

¹⁰ PLC – Programmable Logic Controller

- Definovaná úroveň log. 1: napětí na vstupu 8 ... 30V.
- Maximální vstupní proud: 10mA.
- VÝSTUPY: standardním řešením jsou releové či galvanicky oddělené tranzistorové výstupy:
 - Maximální zatěžovací proud: 0,2A.
 - Odolnost proti zkratu.

2.2.3.2 PŘIPOJENÍ NA PRŮMYSLOVOU SBĚRNICI

Současným trendem u inteligentních kamer je i možnost připojení na průmyslové sběrnice. Na trhu se nejčastěji objevují tato řešení: LAN 10/100/1000 MBit, Profibus a DeviceNet. Připojení na průmyslovou sběrnici pak umožňuje začlenit videosenzor do rozsáhlejších řídicích systémů a sběr dat z videosenzoru nadřazeným systémem.

2.2.3.3 PŘIPOJENÍ K PC

Hlavním využitím připojení videosenzoru k PC je konfigurace videosenzoru. Tato problematika je diskutována v kapitole 2.2.1.

Jako další využití připojení videosenzoru k PC uvedme následující funkce:

- Vyčtení a analýza videosenzorem logovaných dat.
- Vyčtení a analýza chyb měření.

Standardním řešením připojení k PC je dnes USB 2.0.

2.2.3.4 POLOHOVÁNÍ¹¹ A VIZUALIZACE

Problematika polohování byla diskutována v kapitole 2.2.1.1. Standardním v průmyslu používaným řešením je monitorový výstup, případně rozhraní USB 2.0, a zobrazení obrazu pomocí PC.

Pod pojmem „vizualizace měření“ si můžeme představit jakýsi diagnostický režim, kdy videosenzor pracuje autonomně na PC (automaticky bez účasti PC tedy provádí jednotlivá měření). PC je zde využito pro prezentování průběhu a výsledků měření. Tento vizualizační režim umožňuje uživateli po skončení fáze konfigurace ověřit funkci videosenzoru.

¹¹ Polohování – možnost pozorování snímaného obrazu

Pro tyto účely bude vhodné použít rozhraní USB 2.0.

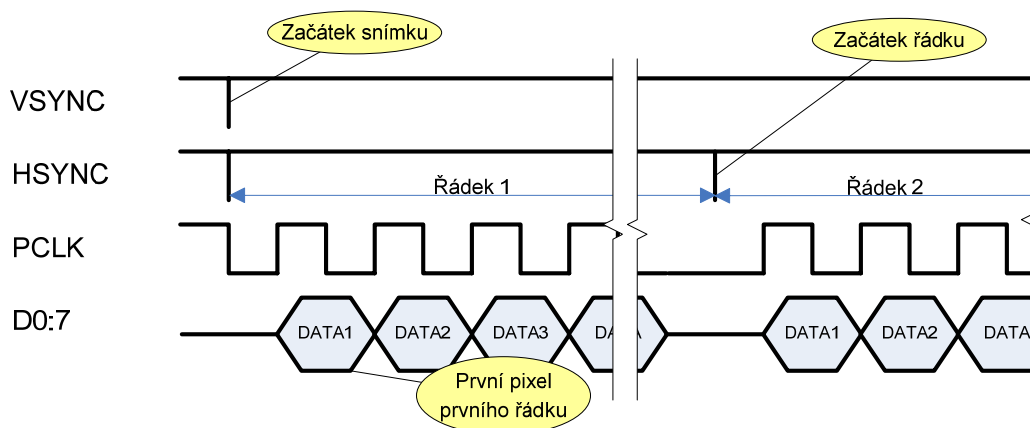
2.3 PROBLEMATIKA „ON-LINE“ ZPRACOVÁNÍ OBRAZOVÝCH DAT

Doposavad jsme se zaměřovali na analýzu z pohledu CMOS senzoru a žádným způsobem jsme v našich úvahách nezohledňovali časové a výpočetní nároky na zpracování obrazových dat, která získáme z CMOS senzoru.

Optimálním zpracováním obrazových dat budeme rozumět zpracování, které nebude zanášet do měření zpoždění, a poskytne výsledek současně s posledním pixelem zpracovávaného bloku dat. Tato varianta je samozřejmě nereálná.

On-line zpracováním obrazových dat budeme rozumět zpracování, kdy tok dat nemusí být z důvodu zpracování zastaven, a zároveň jsou zpracována všechna data. Výsledek je pak poskytnut s jistým zpožděním od posledního pixelu zpracovávaného bloku obrazových dat. Naším cílem je pak tedy minimalizovat toto zpoždění.

Analýzu on-line zpracování začneme základním úvodem k časovému průběhu obrazových toků dat a jejich synchronizaci. Na obr. 2.4 je ilustrován časový průběh výstupních signálů CMOS senzoru pro první dva řádky obrazových dat.



Obr. 2.4 Časový průběh důležitých signálů CMOS senzoru

Signál VSYNC vzestupnou hranou signalizuje začátek snímku (tzn. příchod prvního řádku). Signál HSYNC vzestupnou hranou signalizuje začátek řádku. Dále pak následují jednotlivá datová slova¹² odpovídající informaci vyčtené

¹² Pro CMOS senzor standardně 12 bitů.

z odpovídajících fotoelementů. Data jsou platná se sestupnou hranou signálu PCLK.

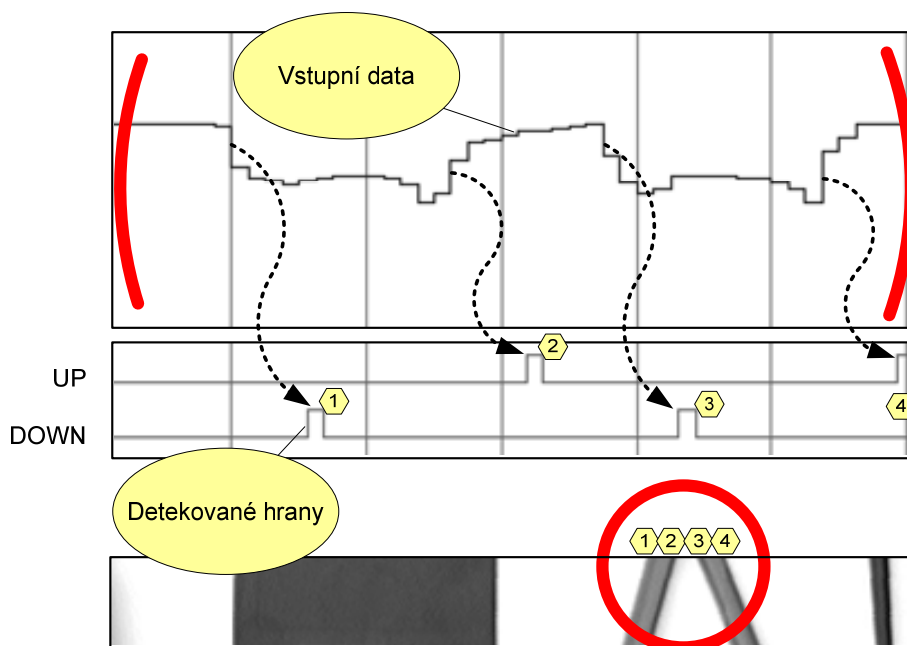
V našem případě víceřádkového videosenzoru budeme uvažovat čtyři základní fáze zpracování obrazových dat víceřádkovým videosenzorem:

- Načtení obrazových dat.
- Detekce hran v jasovém profilu řádku (viz obr. 1.3).
- Zpracování detekovaných hran: selekce správných hran, které náleží objektu a výpočet rozdílu jejich pozic – získání rozměru.
- Prezentování získaných rozměrů na výstupy videosenzoru.

V následující kapitole 2.3.2 bude následovat základní rozdělení možností zpracování digitálního signálu a zhodnocení jejich efektivity s ohledem na čtyři na výše zmíněné fáze zpracování obrazových dat.

2.3.1 HRANOVÁ DETEKCE

V kapitole 1.2 jsme definovali zaměření videosenzoru na „měření rozměru s orientací na řádkové zpracování obrazových dat“. Základní metoda pro měření rozměru při řádkovém zpracování dat je „hranová detekce“. Pojem hrana budeme při zpracování obrazových dat chápat jako výrazný přechod dvou sousedních oblastí v obrazu, kde každá oblast má v nasnímaném obrazu jinou jasovou hodnotu. Tímto způsobem je poté možné detekovat rozměry objektů, které mají odlišnou barvu od pozadí obrazu. Princip hranové detekce v jasovém profilu obrazového řádku je zřejmý z obr. 2.5.



Obr. 2.5 Princip hranové detekce

Výřez jasového profilu první řádky je zobrazen v horní části obr. 2.5. Šedé pásy na bílém pozadí odpovídají nižší hodnotě v jasovém profilu. Detekované hrany jsou zobrazeny v střední části obr. 2.5 (hrany jsou zobrazeny s časovým zpožděním, které je dáno metodou zpracování užitou v simulátoru). Ve spodní části obr. 2.5 je zobrazen obrázek použitý pro simulaci.

Jestliže budeme chtít detekovat hranu šedého pásu (rozhraní šedá - bílá), zjistíme, že hrana odpovídá výraznému přírůstku, resp. poklesu v jasovém profilu řádku. Jestliže tedy chceme detekovat hrany objektů v daném měřicím řádku, je třeba detekovat vhodným způsobem tyto změny v jasovém profilu.

Z takto získaných hran¹³ je již možné určit rozměr objektu a polohu - v našem případě šedého pásu. Hrana „1“ označuje pozici začátku prvního šedého pásu. Tato hrana je sestupná. Další vzestupná hrana bude označovat konec

¹³ Získané hrany - pozice hrany od počátku řádku (počet pixelů od začátku řádku)

objektu. V našem případě se jedná o hranu „2“. Rozdílem¹⁴ těchto hran získáváme šířku šedého pásu.

Výhodou této metody zpracování obrazu je „komprimace zpracovávaných obrazových dat“. Jestliže se rozhodneme dále již zpracovávat pouze detekované hrany, dochází k významnému zmenšení objemu dat, protože místo jednotlivých pixelů celého řádku již po hranové detekci pracujeme jen s pozicemi několika hran, které byly v řádku detekovány.

2.3.2 PROSTŘEDKY PRO ZPRACOVÁNÍ OBRAZOVÝCH DAT

V této kapitole provedeme analýzu zpracování obrazových dat pomocí mikrokontroléru a programovatelného logického obvodu. Analýza bude zaměřena na fáze zpracování obrazových dat definované v kapitole 2.3.

2.3.2.1 ZPRACOVÁNÍ OBRAZOVÝCH DAT POMOCÍ MIKROKONTROLÉRU

Z výše zmíněných fází zpracování obrazových dat bude pro matematické zpracování získaných hran vhodným výkonným prostředkem mikroprocesor. Zároveň široká nabídka mikroprocesoru periférií nabízí jednoduchý způsob prezentování výsledků a navázání na nadřazený systém. Základní problém ovšem nastává v prvním bodě - načtení obrazových dat a detekování hran. Běžný mikroprocesor (např. řady 8051, ARM7, ATMEGA) nemá obvodové prostředky pro načítání 8bitových dat s frekvencí desítek MHz. Vzniká tak problém synchronizace načítání dat mikroprocesorem s datovým tokem. Metody¹⁵, které tyto problémy řeší na procesoru ARM7, jsou popsány v literatuře [1].

Nevýhodou tohoto řešení je, že mikroprocesor není schopen on-line zpracování, protože je při načítání obrazových dat do paměti plně vytížen. Pokud budeme uvažovat víceřádkový videosenzor, můžou být data zpracována v průběhu řádků, které se do procesoru nenačítají. Pokud se ovšem bude jednat o režim, kdy je videosenzor konfigurován jako jednořádkový, bude nutné minimálně každý druhý řádek obrazová data nenačítat a v získaném čase zpracovat data z předchozího řádku.

¹⁴ Rozdíl hran – rozdíl pozic dvou hran (vzdálenost těchto dvou hran od sebe v pixelech)

¹⁵ Jedná se o synchronizaci založenou na konstantní reakční době při přesušení od řídicího signálu HSYNC a načítání dat pomocí lineárního kódu.

Příkladem řešení tohoto problému může být využití funkce 2DDME procesoru Balckfin firmy Analog Device. Tato funkce umožní automaticky přenést paralelní obrazová data z PPI rozhraní DMA kanálem do paměti bez účasti výpočetního jádra procesoru.

2.3.2.2 ZPRACOVÁNÍ OBRAZOVÝCH DAT POMOCÍ PROGRAMOVATELNÉHO LOGICKÉHO OBVODU

Programovatelné logické obvody - PLD¹⁶ - obsahují programovatelné kombinační a synchronní sekvenční obvody. V současné chvíli jsou na trhu dostupné dva základní typy PLD. CPLD¹⁷ obsahující až 500 makrobuněk¹⁸ a FPGA¹⁹ obsahující 30000 makrobuněk.

Synchronní sekvenční logické obvody umožňují řešit synchronní načtení dat oproti mikroprocesoru jednoduchým způsobem. Další výhodou PLD je, že veškeré zpracování probíhá, na rozdíl od mikroprocesoru, paralelně, čímž je možné výrazně snížit zpoždění mezi okamžikem získání obrazových dat a dokončením zpracování.

Jak CPLD tak i FPGA není ovšem primárně zaměřeno na přímý interfacing (neobsahuje periferie jako mikroprocesor). Předpokládá se tedy použití ve spolupráci s dalším podpůrným obvodem.

Shrnutí následuje v tab. 2.4.

	SYN. DATA	MATEMATIKA	INTERFACE	CENA
Mikroprocesor	☹☹	☹	☹☹	\$5
Signálový procesor	☹	☹☹	☹☹	\$20
CPLD	☹	☹☹	☹	\$5
FPGA	☹☹	☹	☹	\$20

Tab. 2.4 Srovnání možností pro zpracování dig. signálu

Vzhledem k tomu, že je videosenzor koncipován jako low-cost aplikace, nejvhodnějším řešením bude mikroprocesor v kombinaci s CPLD, přičemž CPLD bude využito jako hranový detektor a výsledky tohoto zpracování budou dále již zpracovány v mikroprocesoru.

¹⁶ PLD - Programmable Logic Device

¹⁷ CPLD - Complex Programmable Logic Device

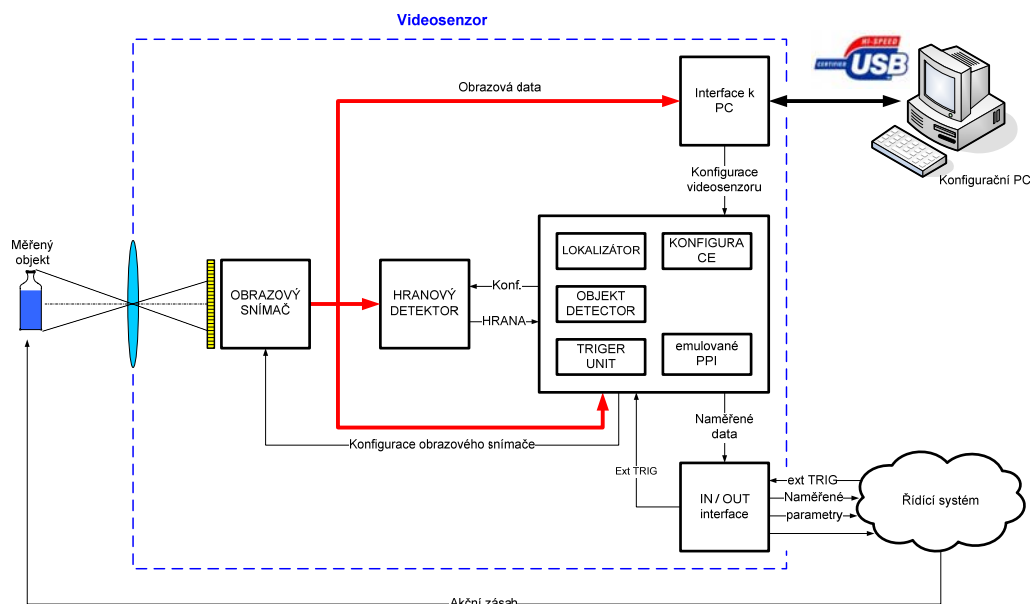
¹⁸ Makrobunčka - základní logický element

¹⁹ FPGA - Field Programmable Gate Array

2.4 ZVOLENÉ ŘEŠENÍ VIDEOSENZORU

Na základě analýzy provedené v kapitole 2 bylo navrženo následující uspořádání vnitřní struktury videosenzoru:

2.4.1 IDEOVÉ BLOKOVÉ SCHÉMA



Obr. 2.6 Principiální blokové schéma videosenzoru

Principiální blokové schéma (obr. 2.6) odráží požadavky definované v předchozích kapitolách 2.2 a 2.3. Vzhledem k zadání diplomové práce a jejímu navázání na [1] byly zvoleny pro dosud obecně uváděné HW prvky následující obvody:

- Obrazový snímač: CMOS senzor KAC-9638
- Mikroprocesor: LPC2148 – řada ARM7
- Programovatelné logické pole využité pro hranovou detekci: XC9572XL
- USB interface k PC: USB řadič řady CY7C68013A-100AXC

Obrazová data získaná ze CMOS senzoru jsou pro režim konfigurace (polohování kamery) odesílána přímo přes řadič USB do konfiguračního PC, na jehož monitoru bude zobrazen. Konfigurační data pro celý videosenzor jsou zaslána z PC do mikroprocesoru, kde jsou uložena. Konfigurace jednotlivých prvků (CMOS senzor, hranový detektor) videosenzoru zajišťuje mikroprocesor.

V režimu měření hranový detektor neustále vyhledává hrany v obrazových datech. Pokud se jedná o řádek, který se má zpracovávat, jsou pozice hrany načí-

tány mikroprocesorem do paměti. Následně dojde k pracování získaných hran a výsledky jsou prezentovány na výstupy.

Sběrnice obrazových dat z CMOS senzoru je také přivedena na bránu mikroprocesoru - tak je možné do paměti mikropočítače načíst obrazovou informaci z vybraných řádků a zpracovávat ji složitějšími způsoby než pouze hranovou detekcí.

2.4.2 STAVEBNÍ KAMENY

Vzhledem k tomu, že vlastnosti jednotlivých HW komponent byly dostatečně diskutovány v [1] a [8], uvedu zde jen základní parametry.

2.4.2.1 ŘÍDÍCÍ PROCESOROVÁ JEDNOTKA

Mikroprocesor LPC2148 (výrobce firma Philips Semiconductors). Základem tohoto procesoru je 32 bitové jádro ARM7. Maximální výpočetní výkon je až 60 MIPS.

- 512 kB FLASH paměti programu.
- 40 kB on-chip static RAM paměť.
- ISP/IAP rozhraní, JTAG rozhraní.
- USB 2.0 Full speed (2 kB RAM pro endpointy).
- 2x 10bit AD převodníky, 1x 10bit DA převodník.
- 2x 32bit čítače/časovače a PWM jednotka.
- RTC – obvod reálného času.
- 2x UART, 2x I2C, SPI – sériová rozhraní.
- napájení procesoru 3,0 až 3,6 V.
- 5V tolerantní vstupy.

2.4.2.2 OBRAZOVÝ SENZOR

CMOS obrazový senzor KAC-9638 (výrobce firma Kodak).

- Velikost aktivní části: 1288(H) x 1032(V) pixelů.
- Velikost pixelu: 6,0 μm x 6,0 μm .
- Výstup: 8 a 10 bitů.
- Dynamický rozsah: 55dB.
- Napájecí napětí: 3,0V \pm 10%.
- Konfigurace: I2C rozhraní.

2.4.2.3 USB ŘADIČ

USB mikrokontrolér CY7C68013A (výrobce firma Cypress)

- Řadič sběrnice USB 2.0.
 - 4 programovatelné endpointy.
 - BULK, INTERRUPT, ISOCHRONOUS datové přenosy.
- Mikrokontrolér 8051
- 4x FIFO (8/16bit).
- GPIF (General Programmable Interface).
- 16kB RAM.
- Napájení procesoru 3,3 V.
- 5V tolerantní vstupy.
- I2C kontrolér: umožňuje boot z EEPROM.

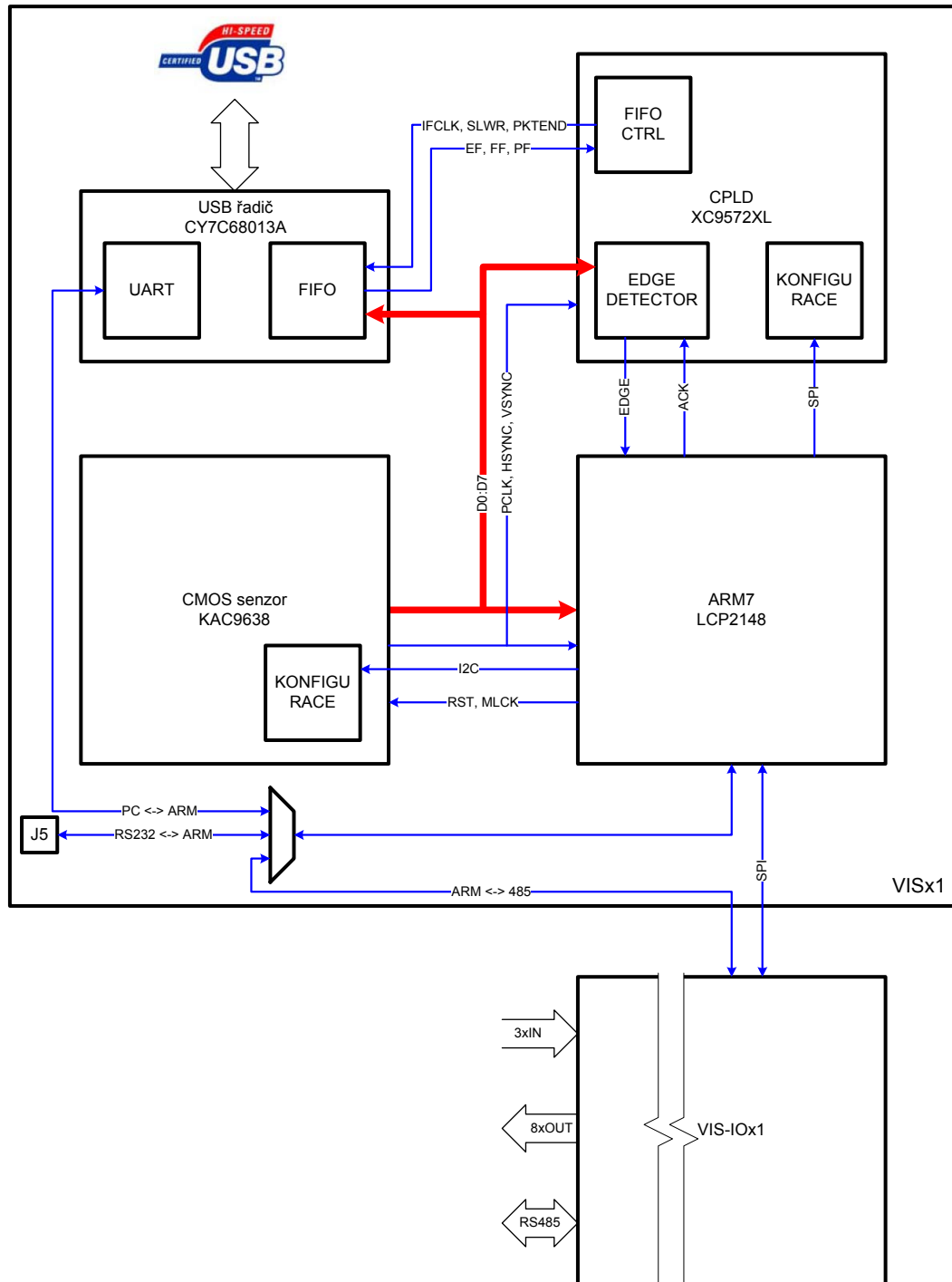
2.4.2.4 CPLD - XC9572XL

CPLD XC9572XL je produktem firmy Xilinx

- 72 makrobuněk, 1,600 využitelných hradel.
- Napájení procesoru 3,3 V.
- 5V tolerantní vstupy.
- In-system JTAG programování.
- 5ns „pin to pin“ zpoždění logického signálu.
- 3x globální clk.

3 POPIS REALIZACE VIDEOSENZORU

3.1 BLOKOVÉ SCHÉMA VIDEOSENZORU



Obr. 3.1 Blokové elektrické schéma videosenzoru

Pro konstrukci videosenzoru bylo zvoleno modulární uspořádání. Celý videosenzor se skládá z tří modulů. CMOS obrazový snímač je umístěn na samo-

statné desce plošného spoje. Tato deska je propojena konektorem s procesorovou deskou, která je nazvána VISx1.

Procesorová deska obsahuje mikroprocesor, CPLD a řadič USB. Tyto tři prvky tvoří výpočetní jádro celého systému a zajišťují zpracování obrazových dat. CPLD je použito jako hranový detektor (pro detekci hran v jasových profilech jednotlivých řádcích obrazové informace) a pro řízení přenosu obrazu do PC.

Třetím modulem je deska vstupů - výstupů VIS-IOx1. Tato deska je spojena přes konektor s procesorovou deskou. Deska vstupů obsahuje tři vstupy, osm výstupů a budič sběrnice RS485. Všechny tyto periferie jsou galvanicky oddělené. Ovládání periferií zajišťuje mikroprocesor přes rozhraní SPI.

Podrobný popis jednotlivých modulů bude následovat v následujících kapitolách.

3.2 SEZNAM VÝZNAMÝCH SIGNÁLŮ, JEJICH VÝZNAMY A PROPOJENÍ

SIGNÁL	CMOS	ARM7	CPLD	CY7C68013A	POPIS
	ZDROJ SIGNÁLŮ ²⁰				
SIGNÁLY TÝKAJÍCÍ SE CMOS SENZORU					
D0:D7	O	I	I	I	Obrazová data ²¹
ARM_MCLK	I	O	I		Master clock, hodinový signál generovaný pro CMOS senzor (12-27 MHz)
CMOS_PCLK	O	I	I		Synchronní hodinový signál s D0:D7
CMOS_HSYNC	O	I	I		Synchronizační řádkový signál
CMOS_VSYNC	O	I	I		Synchronizační snímkový signál
RESETB_CMOS	I	O			Reset CMOS senzoru
SIGNÁLY TÝKAJÍCÍ SE USB ŘADIČE CY7C68013A					
IFCLK_FIFO			O	I	Hodinový signál pro zápis do FIFO paměti USB řadiče, signál vzniká hradlováním PLCK
SLWR_FIFO			O	I	Povolení zápisu do FIFO paměti
PKTEND_FIFO			O	I	Signál způsobí odeslání USB paketu
FIFO_FLAGA, FIFO_FLAGB, FIFO_FLAGC FIFO_FLAGD			I	O	Flagy reprezentující stav FIFO USB řadiče. FLAGC = EF, FLAGB = FF
SIGNÁLY TÝKAJÍCÍ SE HRANOVÉHO DETEKTORU					
\CPLD_E_UP, \CPLD_E_DN		I	O		Příznak detekované hrany
\ARM_UP_ACK, \ARM_DN_ACK		O	I		Potvrzení přijetí detekované hrany
SIGNÁLY TÝKAJÍCÍ SE KOMUNIKACÍ					
ARM_SDA, ARM_SCL	I	O			I2C konfigurace CMOS senzoru
MOSI_CPLD, SCK_CPLD		O	I		SPI konfigurace CPLD
SCK_ARM, MISO_ARM, MOSI_ARM, SHIFT_EN, SHIFT_LOAD		O			SPI komunikace s deskou VIS-IOx1
TXD_ARM, RXD_ARM, \ARM_485DE, \USB_UART_RQ, USB_TXD, 485_TXD, RS232_TXD,					Multiplexovaná UART komunikace

Tab. 3.1 Seznam významných signálů videosenzoru

V tabulce tab. 3.1 jsou uvedeny názvy významných signálů. Názvy uvedené v této tabulce se shodují s názvy použitými v elektrickém schématu videosenzo-

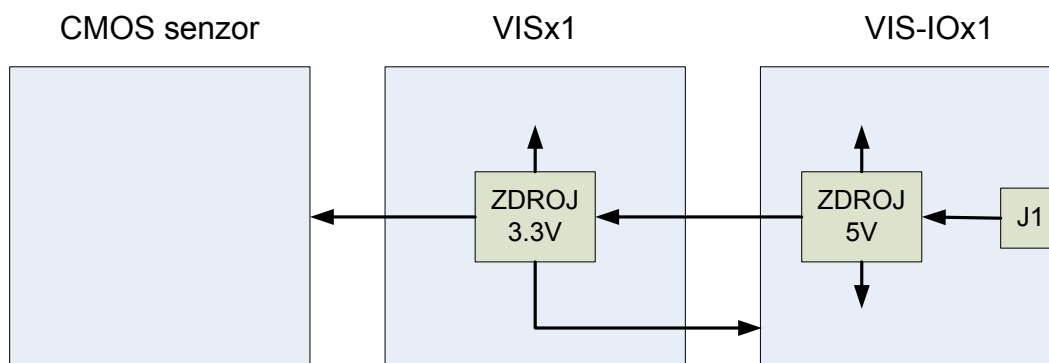
²⁰ I – signál je do obvodu přiveden jako vstup, O – obvod je zdrojem signálu

²¹ Signál D6 je vedený přes CPLD, protože je multiplexován se signálem RS232_RTS

ru (Příloha A) a v kódu mikroprocesoru. Z tabulky je zároveň zřejmé připojení jednotlivých signálů na jednotlivé obvody

3.3 POPIS ELEKTRICKÉHO ZAPOJENÍ

3.3.1 NAPÁJENÍ SYSTÉMU



Obr. 3.2 Schématické znázornění struktury napájení jednotlivých částí

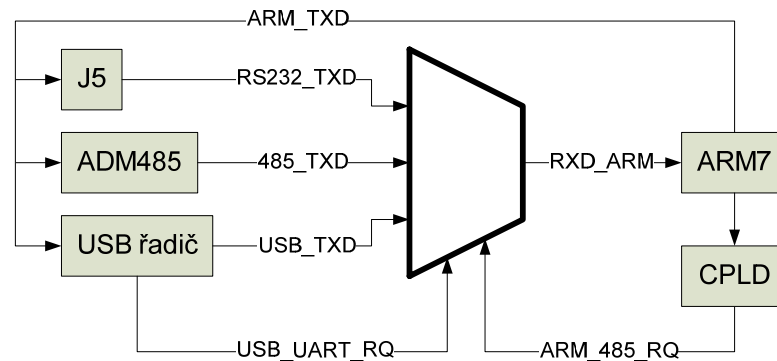
Struktura napájení celého systému videosenzoru je znázorněna na obr. 3.2. Deska VIS-IOx1 obsahuje spínaný zdroj (5V) s obvodem LM2596, který pracuje jako snižující měnič. Požadované vstupní napětí videosenzoru je tedy určeno obvodem LM2596 (6.5V – 40V). Tento zdroj je dimenzovaný na zatěžovací proud 0,8A. Omezujícím prvkem je tlumivka L1 (220uH/0.9A, Würth Elektronik).

Z desky VIS-IOx1 je napájena procesorová deska VISx1, která obsahuje lineární stabilizátor na 3.3V, protože se předpokládá při vývoji samostatné použití desky VISx1. Deska CMOS senzoru a logické obvody desky VIS-IOx1 navázané na VISx1 jsou napájené z 3.3V.

3.3.2 MULTIPLEXOVANÝ UART

Pro komunikace mezi mikroprocesorem, řadičem USB a budičem linky RS485 byl zvolen sériový asynchronní přenos. Zároveň je tato sériová linka využívána k IAP programování mikroprocesoru. Mikroprocesor LCP2148 sice obsahuje dvě periferie UART, bohužel však při návrhu bylo nezbytné použít piny asociované k periférii UART1 pro jiné účely, a tudíž bylo nezbytné vyřešit komunikaci mezi jednotlivými obvody na jediné sdílené sériové lince.

Bylo zvoleno řešení, kdy jsou k periférii UART0 pomocí multiplexeru 74LV4052 přepínány jednotlivé obvody. Způsob přepínání je zřejmý z obr. 3.3.



Obr. 3.3 Multiplexovaný UART – blokové schéma

Po zapnutí napájení je zajištěno, že signály USB_UART_RQ a ARM_485_RQ ovládající multiplexer jsou v logické úrovni „1“, čímž je umožněno IAP programování mikroprocesoru (po resetu a jsou všechny piny mikroprocesoru i USB řadiče konfigurovány jako vstupy). Po startu aplikace²² v mikroprocesoru a USB řadiči je přepínání multiplexoru řízeno mikroprocesorem a řadičem USB.

B - USB ŘADIČ	A - ARM	
0	0	USB_TXD
0	1	485_TXD
1	0	RS232_TXD
1	1	RS232_TXD

Tab. 3.2 Pravdivostní tabulka multiplexeru UARTu

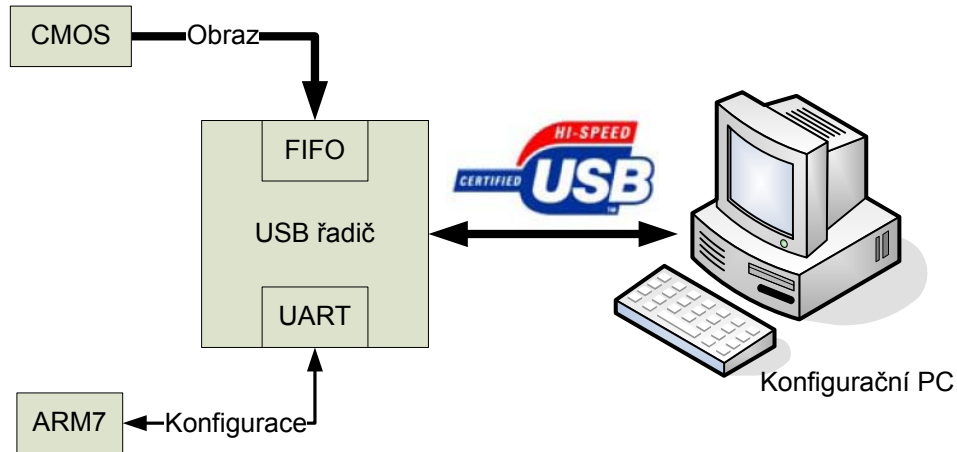
Způsob přepínání multiplexeru je zřejmý z pravdivostní tabulky tab. 3.2.

3.4 PŘENOS OBRAZU DO NADŘAZENÉHO PC

Použitý způsob přenosu obrazu z videosenzoru do PC přes rozhraní USB 2.0 vychází z výsledků [8]. Obrazová data jsou přenášena přes rozhraní USB s využitím USB řadiče CY7C68013A, který obsahuje rozhraní FIFO paměti pro zápis a čtení přenášených dat. Navržené zapojení umožňuje přenášet přes FIFO rozhraní obrazová data z videosenzoru do PC. Opačný směr (z PC do videosenzoru) není v navrženém zapojení přes FIFO rozhraní umožněn. Pro tento účel se využívá

²² Startem aplikace rozumějme start jádra mikroprocesoru a započítí vykonávání programového kódu.

multiplexovaný UART popsany v kapitole 3.3.2. Popsany způsob datových toků mezi videosenzorem a PC ilustruje obr. 3.7.



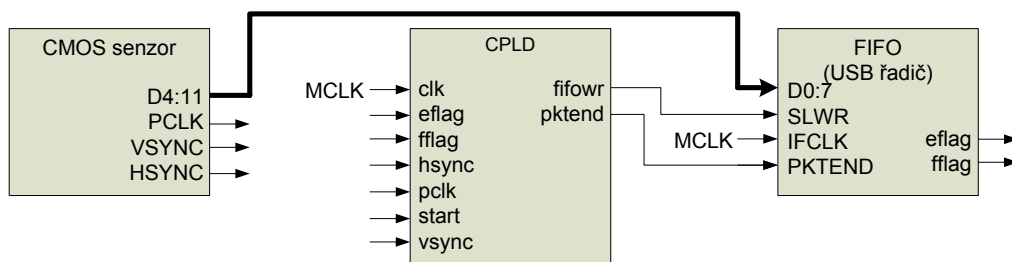
Obr. 3.4 Blokové schéma – datový tok mezi videosenzorem a PC

Pro řízení zápisu obrazových dat do FIFO paměti jsou využívány následující signály (význam signálů je uveden pro asynchronní režim, v kterém je paměť používána):

- D0:7 : datové vodiče FIFO paměti.
- SLWR: signál pro zápis slova do FIFO paměti.
- IFCLK: hodinový signál (s tímto signálem je vzorkován vstup SLWR).
- PKTEND: signál, který způsobí okamžité odeslání paketu.

Zápis do FIFO paměti je řízen pomocí CPLD. Na obr. 3.5 je znázorněno propojení mezi CMOS senzorem, VHDL modulem²³ v CPLD, které řídí zápis do FIFO paměti, a FIFO paměti. Řídící algoritmus použitý v CPLD spočívá v hradlování signálu „PCLK“ na výstupní signál „fifowr“ a generování signálu „pktend“.

²³ VHDL modul – uzavřený blok s definovaným interface

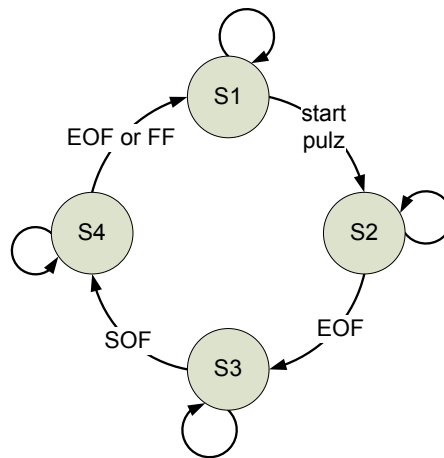


Obr. 3.5 Blokové schéma - řízení přenosu obrazu do PC

Obrazová data se do PC přenášejí jako celé snímky v jednom USB paketu a zároveň je zajištěno, že začátek paketu bude odpovídat začátku snímku. Tímto způsobem je zajištěna synchronizace při zobrazení obrazu na PC. Způsob řízení zápisu do FIFO paměti je znázorněn stavovým automatem na obr. 3.6.

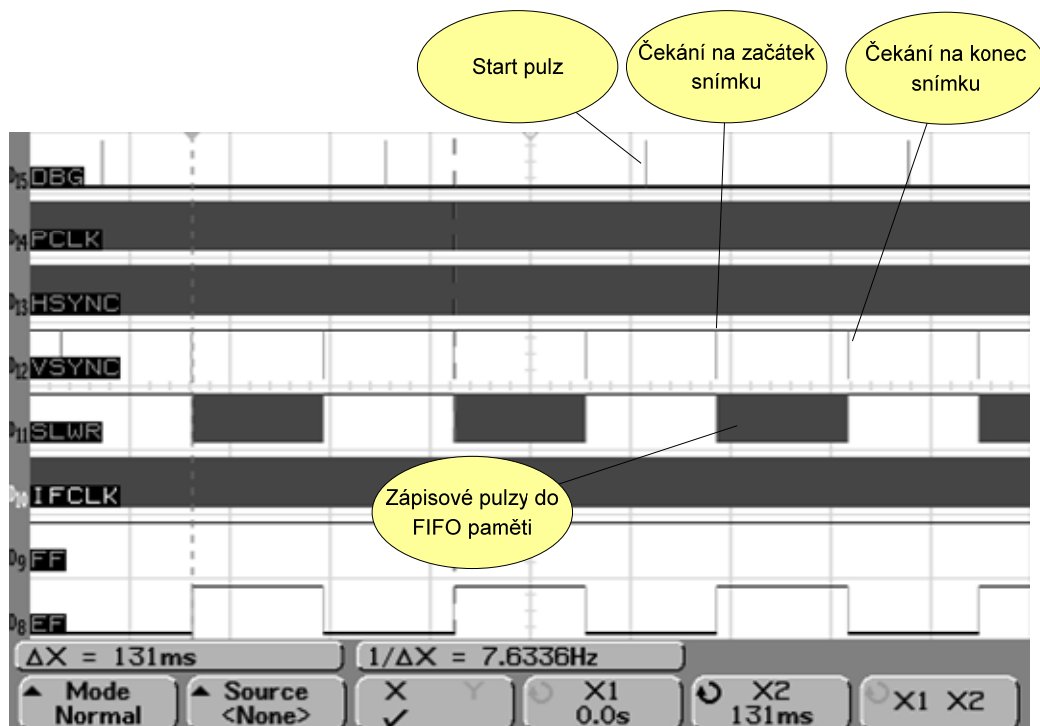
- Stav „S1“: základní stav po resetu – čekání na start impuls.
- Stav „S2“: automat čeká na dokončení právě probíhajícího snímku (čekání na splnění podmínky VSYNC = 0).
- Stav „S3“: automat čeká na začátek snímku (tento snímek bude přenesen).
- Stav „S4“: s tomto stavu je na hradlovaný výstup CPLD „slwr“ propuštěn signál „PCLK“. Tento stav je ukončen koncem snímku, nebo naplněním FIFO paměti (detekováno signálem „fflag“). Při přechodu do stavu „S1“ je vygenerován „pktend“ pulz.

K naplnění FIFO paměti dojde v případě, že PC včas nevyzvedne připravená data. V tomto případě dojde ke ztrátě dat v přenášeném snímku a je zbytečné pokračovat v přenosu dat, protože zároveň dojde i k ztrátě synchronizace. Tento, pouze částečně, přenesený snímek je v PC detekován a není zobrazen.



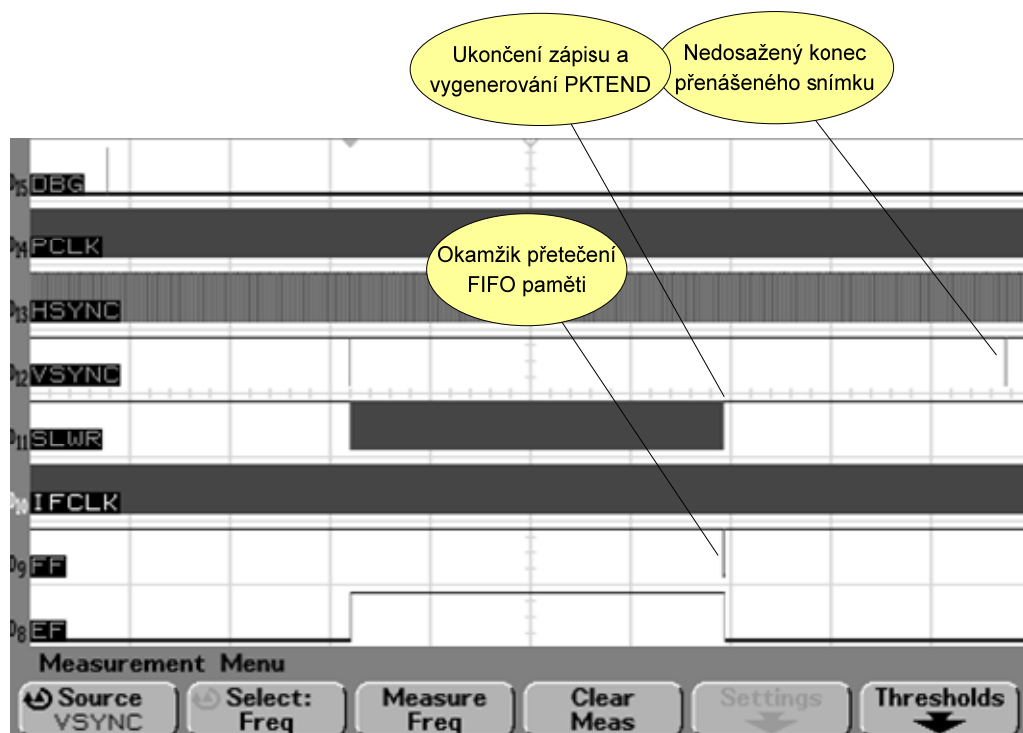
Obr. 3.6 Stavový automat pro řízení zápisu do FIFO paměti

Následující snímek z osciloskopu (obr. 3.7) zobrazuje zapsání čtyř snímků do FIFO paměti USB řadiče. Sled signálů odpovídá výše uvedenému popisu stavového automatu. Stav „S3“ není ze snímku patrný, protože v daném zobrazení je interval, po který je signál VSYC aktivní, zobrazen pouze jako jediný pulz.



Obr. 3.7 Snímek z osciloskopu – přenos čtyř snímků do PC

Dále pro úplnost uvedeme snímek při situaci, kdy dojde k přeplnění FIFO paměti (obr. 3.8). Přeplnění paměti je detekováno příznakovým signálem FF (full flag), který při stavu „S4“ společně se signálem VSYNC testován. Pokud je detekováno přeplnění, je vygenerován signál PKTEND a stavový automat přechází do stavu „S1“.



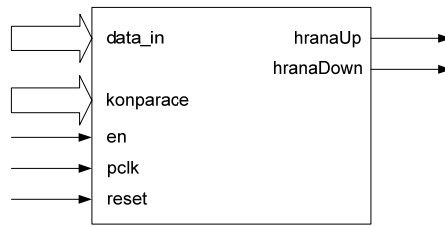
Obr. 3.8 Snímek z osciloskopu – přeplnění FIFO paměti

3.5 HRANOVÝ DETEKTOR

V této kapitole bude popsán realizovaný způsob detekce hran v obrazových datech (D0:D7) komparační a diferenční metodou. Pro funkci hranového detektoru je ve videosenzoru použit obvod CPLD – popis obvodu byl uveden v kapitole 2.4.2.4. Detekce hran se provádí v jednotlivých řádcích obrazových dat.

3.5.1 HRANOVÝ DETEKTOR – KOMPARAČNÍ METODA

Komparační metoda detekce hran je nejjednodušší možností, jak získat hrany při řádkovém zpracování obrazových dat.



Obr. 3.9 RTL schéma modulu hranového detektoru

Do modulu hranového detektoru vstupuje signál obrazových dat (`data_in`) a komparační hodnota. Tato komparační hodnota je do CPLD nahrána přes sériový interface (signály `MOSI_CPLD`, `SCK_CPLD`) z mikroprocesoru. Vstup „en“ je využit pro povolení výstupních signálů modulu. Na vstup „pclk“ je přiveden synchronizační hodinový signál obrazových dat. A konečně na vstup „reset“ je přiveden řádkový synchronizační signál `HSYNC`, který resetuje celý modul hranového detektoru na začátku každého řádku obrazových dat. Výstupem modulu hranového detektoru jsou signály „hranaUp“ a „hranaDown“, které prezentují detekované příznaky příslušné hrany v obrazových datech.

Následující část VHDL kódů hranového detektoru definuje výše uvedený interface a dále zde jsou definovány signály, které budou použity v jednotlivých procesech při zpracování dat v hranovém detektoru.

Signál „GF_0“ je příznak vzniklý porovnáním vstupních dat a komparace (hodnota „1“ odpovídá situaci kdy `data_in > komparace`). Signál „GF_1“ je o jednu periodu hodinového signálu „pclk“ zpožděný signál „GF_0“. Toto zpoždění je na HW úrovni realizováno klopným obvodem typu „D“.

```

entity edge_detector is
  Port ( data_in : in  STD_LOGIC_VECTOR (7 downto 0);
        Komparace : in  STD_LOGIC_VECTOR (7 downto 0);
        pclk : in  STD_LOGIC;
        reset : in  STD_LOGIC;
        en : in  STD_LOGIC;
        hranaUp : out  STD_LOGIC;
        hranaDown : out  STD_LOGIC);
end edge_detector;
-----
architecture Behavioral of edge_detector is

  signal GF_0, GF_1: std_logic;
  signal EdgeUp, EdgeDown: std_logic;

begin

  hranaUp <= EdgeUp;
  hranaDown <= EdgeDown;

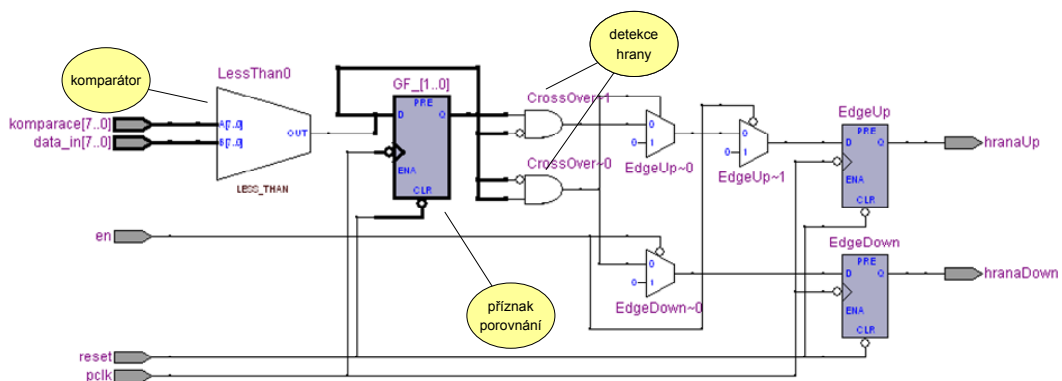
```

Hranový detektor – VHDL kód – definice interface

Komparační hranový detektor pracuje následujícím způsobem:

- S každou periodou hodinového signálu „pclk“ porovnají signály „data_in“ a „komparace“. Příznak (výsledek) porovnání je uložen v signálu „GF_0“.
- Ze signálu „GF_0“ je vytvářen výše popsáním způsobem zpožděný příznak porovnání „GF_1“.
- Porovnáním „GF_0“ „GF_1“ je provedena detekce sestupné či vzestupné hrany.

Popsaná funkce komparačního hranového detektoru je zřejmá z RTL schématu na obr. 3.10.



Obr. 3.10 RTL schéma komparačního hranového detektoru

Následuje VHDL kód popisující procesy „komp“ a „CrossOver“, které realizují uvedenou funkci komparačního hranového detektoru.

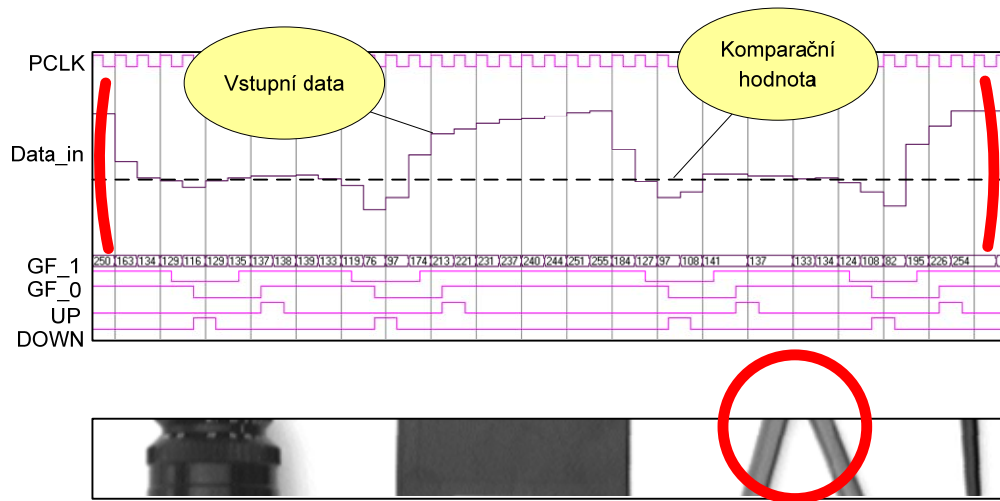
```

komp: PROCESS (pclk, reset)
begin
    if reset = '0' then
        GF_0 <= '0';
        GF_1 <= '0';
    elsif falling_edge(pclk) then
        GF_1 <= GF_0;
        if unsigned(data_in) > unsigned(komparace)
            then GF_0 <= '1';
            else GF_0 <= '0';
        end if;
    end if;
END PROCESS;
-----
CrossOver: PROCESS (pclk, reset)
begin
    if reset = '0' then
        EdgeUp <= '0';
        EdgeDown <= '0';
    elsif falling_edge(pclk) then
        if (EN = '0') then
            EdgeUp <= '0';
            EdgeDown <= '0';
        elsif GF_0 = '0' and GF_1 = '1' then -- sestupna hrana
            EdgeUp <= '0';
            EdgeDown <= '1';
        elsif GF_0 = '1' and GF_1 = '0' then -- vzestupna hrana
            EdgeUp <= '1';
            EdgeDown <= '0';
        else
            EdgeUp <= '0';
            EdgeDown <= '0';
        end if;
    end if;
END PROCESS;

```

Hranový detektor – VHDL kód – proces komparace a detekce hrany

Ukázka simulace výše popsaného komparačního detektoru je zobrazena na obr. 3.11. Jako vstupní data detektoru byl použit první řádek testovacího obrázku (v dolní části obr. 3.11). Komparační úroveň byla nastavena na hodnotu 128. Tato hodnota byla zvolena záměrně pro demonstraci problematiky nastavení komparační úrovně.



Obr. 3.11 Komparační hranový detektor – ukázka simulace

3.5.1.1 VLASTNOSTI METODY KOMPARAČNÍ DETEKCE

Způsob určení komparační hodnoty je klíčový pro celou funkci detektoru. Pokud bychom například pro naši simulaci zvolili komparační hodnotu 150, získali bychom, na rozdíl od současné situace, pouze dvě sestupné a dvě vzestupné hrany. Tím vzniká otázka, jak správně zvolit komparační hodnotu pro daný obrazový signál a jak zajistit její korekci v průběhu měření, pokud například dojde k změně osvětlení scény.

Správné nastavení komparační úrovně pro daná obrazová data je možné určit z histogramu těchto dat. Nevýhodou ovšem je, že pokud budeme nuceni korigovat hodnotu komparační úrovně v průběhu měření, jednoduchost komparační detekce bude bohužel kvitována výrazně matematicky složitějšími algoritmy pro nastavení komparační úrovně.

3.5.2 HRANOVÝ DETEKTOR – DIFERENČNÍ METODA

Diferenční metoda detekce hran vychází z hledání maxima první diference z obrazových dat. Maximum je detekováno z průchodu druhé diference nulou.

Interface VHDL modulu diferenčního detektoru byl zvolen stejný jako u komparačního detektoru, což umožní jednoduchým způsobem zaměnit použití jednotlivých detektorů v VHDL projektu. Následující část VHDL kódů hranového detektoru definuje výše uvedený interface (obr. 3.9) a dále zde jsou definovány signály, které budou použity v jednotlivých procesech při zpracování dat v hranovém detektoru.

- DATA1 a DATA2 jsou zpožděné signály vstupních dat pro vytvoření první diference.
- DIFF1_0 a DIFF1_1 jsou signály první diference a první zpožděné diference.
- DIFF1_ZF0 a DIFF1_ZF1 jsou příznakové signály detekující nulovost první diference.
- DIFF2_SF0 a DIFF2_SF1 jsou příznakové signály detekující znaménko druhé diference.
- Signály EdgeUp a EdgeDown jsou přiřazeny na výstupy modulu.

```

entity edge_detector is
    generic(n: natural := 8);
    Port ( data_in : in  STD_LOGIC_VECTOR (7 downto 0);
          pclk : in  STD_LOGIC;
          reset : in  STD_LOGIC;
          komparace : in  STD_LOGIC_VECTOR (7 downto 0);
          en : in  STD_LOGIC;
          hranaUp : out  STD_LOGIC;
          hranaDown : out  STD_LOGIC);
end edge_detector;

-----

architecture Behavioral of edge_detector is
    signal DATA1, DATA2: STD_LOGIC_VECTOR (n-1 downto 0);
    signal DIFF1_0, DIFF1_1: STD_LOGIC_VECTOR (n downto 0);
    signal DIFF2_SF0, DIFF2_SF1, DIFF1_ZF0, DIFF1_ZF1: std_logic;
    signal EdgeUp, EdgeDown: std_logic;

begin

    hranaUp <= EdgeUp;
    hranaDown <= EdgeDown;

```

Hranový detektor – VHDL kód – definice interface

3.5.2.1 PROCES PRVNÍ A DRUHÉ DIFERENCE A PRAHOVÁNÍ

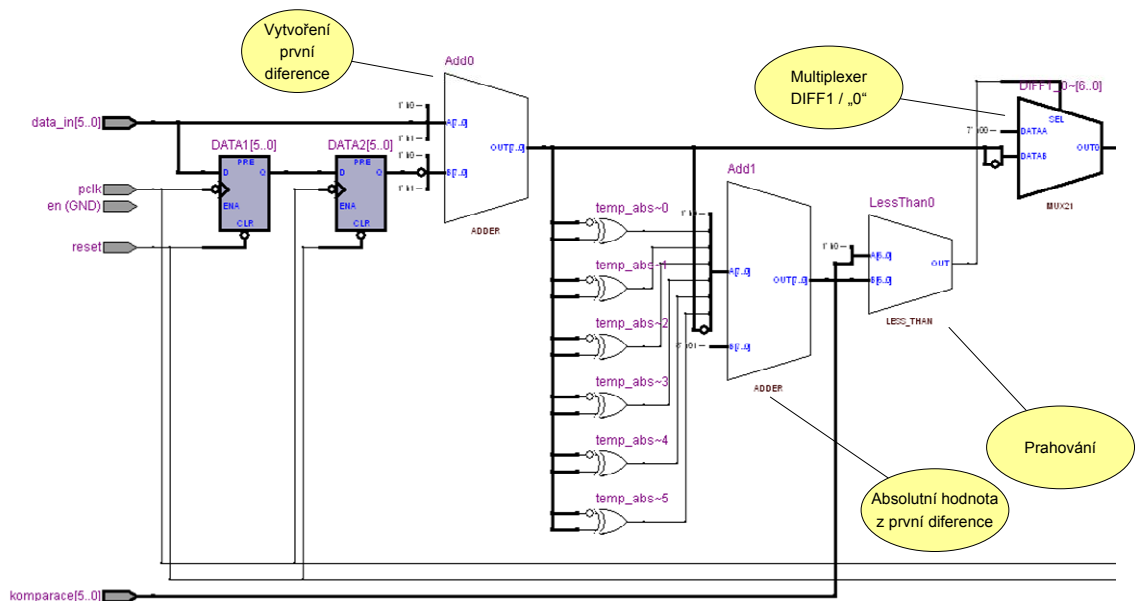
Hranový detektor je realizován následujícími kroky:

- První diference: je vytvořena z vstupních dat (použitý operátor: $[-1 \ 0 \ 1]$ ²⁴).
- Prahování: pokud je získaná první diference menší než vstup „komparace“ je první diference nahrazena nulovou hodnotou.

²⁴ Operátor $[-1 \ 0 \ 1]$ – diference není vytvářena ze sousedních prvků, ale z dvou prvků o vzdálenosti 2. Tím je získána vyšší hodnota první diference

- Vytvoření příznaků ZF první diference: tyto příznaky jsou v závěrečné fázi využity pro blokování signálu EdgeUp a EdgeDown pro odstranění falešných hran.
- Vytvoření znaménkového příznaku SF druhé diference: (pokud je druhá diference záporná SF = 1). Tyto signály jsou v závěrečné fázi použity pro detekci maxima první diference.

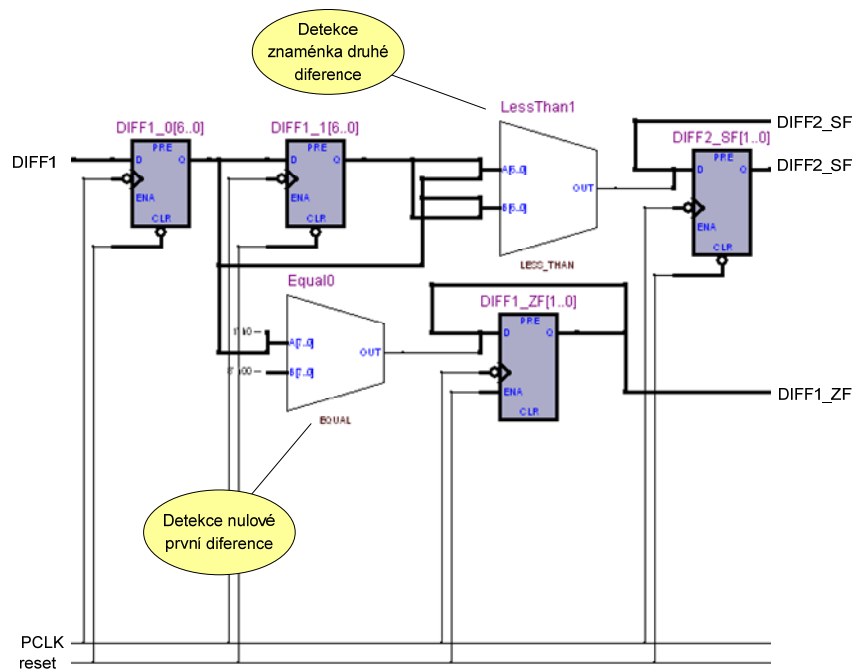
Způsob vytvoření první diference a prahování je znázorněn na obr. 3.12. Rozdíl vstupních dat (`data_in`) a signálu zpožděných dat (`DATA2`) je vytvářen sčítačkou „Add0“. Dále je pak podmíněnou bitovou inverzí a sčítačkou „Add1“ vytvořena absolutní hodnota rozdílu. Komparátor „LessThan0“ porovnává absolutní hodnotu rozdílu a hodnotu signálu „komparace“. Tento komparátor pak řídí multiplexer „MUX2“, který vytváří prahovanou první diferenci.



Obr. 3.12 RTL schéma vytvoření první diference a prahování

Z následujícího obrázku obr. 3.13 je zřejmý způsob vytvoření příznakových signálů DIFF2_SF²⁵ a DIFF1_ZF. Příznak DIFF1_ZF (příznak nulovosti difference) je vytvořen pomocí komparátoru „Equal0“.

Příznak DIFF2_SF (znaménko druhé difference) je vytvořeno pomocí komparátoru „LessThan1“, na jehož vstupy jsou přivedeny signály DIFF1_0 a zpožděný DIFF1_1.



Obr. 3.13 RTL schéma vytvoření příznaků DIFF2_SF a DIFF1_ZF

²⁵ Název signálu DIFF2_SF je použit pro zjednodušení pro sdruženou dvojici DIFF2_SF0 a DIFF2_SF1. Stejně bude použito v následujícím textu pro další signály.

```

DIFF: PROCESS (pclk, reset)-- vytvoreni prvni diference oprahovani a pote druhe diference
    variable temp,temp_abs: STD_LOGIC_VECTOR (n downto 0);
    variable prah: Boolean;
begin
    if reset = '0' then
        DATA2 <= (others => '0');
        DATA1 <= (others => '0');
        DIFF1_0 <= (others => '0');
        DIFF1_1 <= (others => '0');
        DIFF2_SF0 <= '0';
        DIFF2_SF1 <= '0';
    elsif falling_edge(pclk) then
        DATA2 <= DATA1;
        DATA1 <= data_in;
        DIFF1_1 <= DIFF1_0;
        -- prvni diference pred prahovanim (operator [-1 0 1])
        temp := ('0' & unsigned(data_in)) - ('0' & unsigned(DATA2));
        temp_abs := abs(signed(temp)); -- absolutni hodnota pro prahovani
        if(temp_abs>komparace)
            then prah := true;
            else prah := false;
        end if;
        if (prah)          -- prahovani
            then DIFF1_0 <= temp;
            else DIFF1_0 <= (others => '0');
        end if;
        -- vytvoreni znamenkoveho priznaku druhe diference
        DIFF2_SF1 <= DIFF2_SF0;
        if signed(DIFF1_0) < signed(DIFF1_1)
            then DIFF2_SF0 <= '1';
            else DIFF2_SF0 <= '0';
        end if;
        -- vytvoreni Zero priznaku od signalu DIFF1_0
        DIFF1_ZF1 <= DIFF1_ZF0;
        if (DIFF1_0 = 0)
            then DIFF1_ZF0 <= '1';
            else DIFF1_ZF0 <= '0';
        end if;
    end if;
END PROCESS;

```

Hranový detektor – VHDL kód – proces „DIFF“

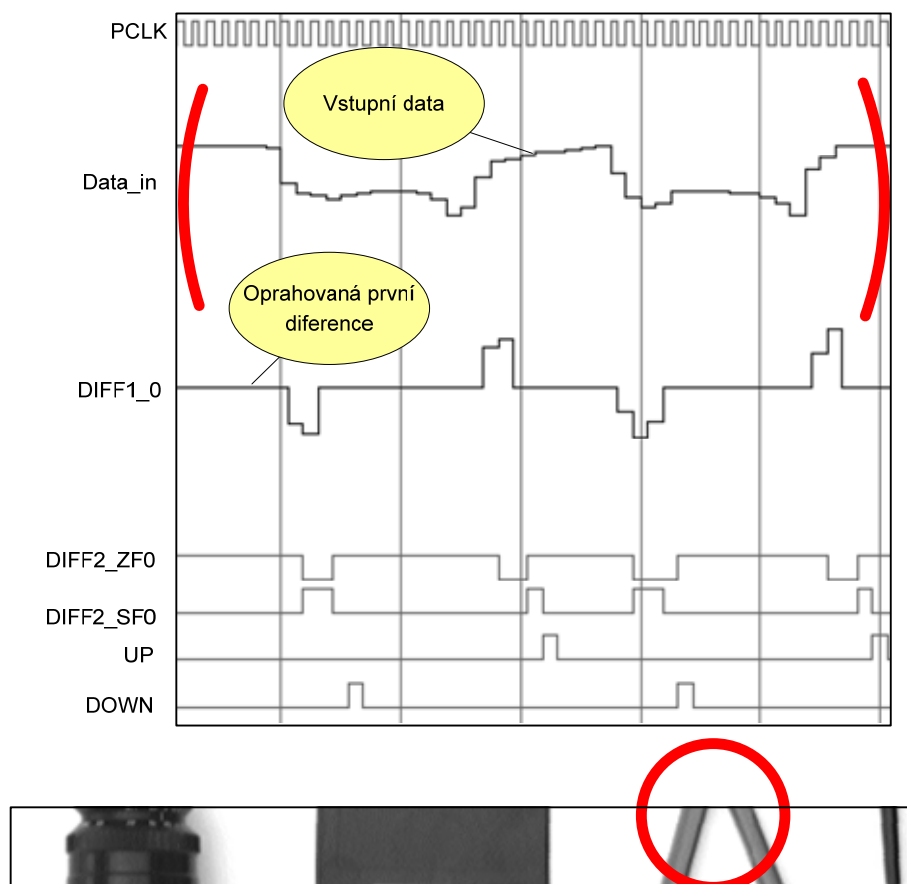
3.5.2.2 PROCES „ZEROCROSSOVER“ – DETEKCE MAXIMA PRVNÍ DIFERENCE

Pro detekci maxima první diference je v modulu hranového detektoru napsán proces „ZeroCrossOver“. Proces detekuje změnu znaménka druhé diference (přechod druhé diference nulou). Zároveň je využit signál DIFF1_0, který je využit k blokování vygenerování falešné hrany v případě, že dojde ke změně znaménka při změně druhé diference z nuly na jakoukoli zápornou hodnotu.

```
ZeroCrossOver: PROCESS (pclk, reset)
begin
    if reset = '0' then
        EdgeUp <= '0';
        EdgeDown <= '0';
    elsif falling_edge(pclk) then
        if DIFF1_ZF1 = '1' then
            EdgeUp <= '0';
            EdgeDown <= '0';
        elsif DIFF2_SF1 = '1' and DIFF2_SF0 = '0' then -- sestupna hrana
            EdgeUp <= '0';
            EdgeDown <= '1';
        elsif DIFF2_SF1 = '0' and DIFF2_SF0 = '1' then -- vzestupna hrana
            EdgeUp <= '1';
            EdgeDown <= '0';
        else
            EdgeUp <= '0';
            EdgeDown <= '0';
        end if;
    end if;
END PROCESS;
```

Hranový detektor – VHDL kód – proces „ZeroCrossOver“

Ukázka simulace výše popsaného diferenčního detektoru je zobrazena na obr. 3.14. Jako vstupní data detektoru byl použit první řádek z testovacího obrázku (v dolní části obr. 3.14). Jako prahovací hodnota byla v simulaci nastavena hodnota 32. Z časového průběhu simulace je dobře patrný výše popsaný princip hranové detekce diferenční metodou.



Obr. 3.14 Diferenční hranový detektor – ukázka simulace

3.5.2.3 VLASTNOSTI METODY DIFERENČNÍ DETEKCE

- Odolnost proti změně osvětlení v měřeném obraze, které může být např. způsobena vinětací objektivu nebo nerovnoměrným nasvícením scény.
- Odolnost proti změně osvětlení v čase.
- Schopnost detektoru rozlišit více jasových úrovní (např. bílá, šedá a černá). Pro tento účel by při použití komparační metody musely být použity dva detektory, kdy první detektor by měl komparační úroveň nastavenou na jasové rozhraní bílá-šedá a druhý na rozhraní šedá-černá.

3.5.3 SROVNÁNÍ METOD Z HLEDISKA IMPLEMENTACE V CPLD

Vlastnosti detekce obou uvedených metod byly diskutovány v kapitolách 3.5.1.1 a 3.5.2.3. Nyní provedeme srovnání z hlediska implementace detekčních metod v CPLD.

V tab. 3.3 jsou uvedeny parametry jednotlivých realizací hranového detektoru získané po „fitaci²⁶“. Zpoždění je časový interval mezi detekovanou hranou a reakcí výstupu hranového detektoru. Uvedená hodnota odpovídá počtu pixelů mezi hranou a reakcí výstupu. Pojem makrobuňka byl vysvětlen v kapitole 2.3.2.2. Číslo uvedené v tabulce odpovídá počtu potřebných makrobuněk v CPLD k realizaci daného VHDL kódu.. P-term („product term“) je součinnový logický výraz, který je v CPLD realizován programovatelnou AND maticí. Číslo uvedené v tabulce odpovídá počtu řádků AND matice potřebných pro implementaci VHDL kódu do CPLD.

	ZPOŽDĚNÍ	MAKROBUŇKY	P-TERMY
Komparační hranový detektor	1	4	25
Dif. hranový detektor - 8 bit	4	82	608
Dif. hranový detektor - 7 bit ²⁷		75	572
Dif. hranový detektor - 6 bit		56	471

Tab. 3.3 Parametry navržených hranových detektorů získané z „fitace“

Zpoždění realizovaných detektorů je různé, což ovšem nečiní problém, protože jak pro vzestupnou, tak i pro sestupnou hranu je zpoždění pro daný detektor shodné, a tak je možné zpoždění na úrovni mikroprocesoru korigovat odečtením dané hodnoty.

Srovnáme-li oba detektory z hlediska makrobuněk a p-termů, je zřejmé, že komparační detektor má výrazně nižší nároky na implementaci v CPLD (ve smyslu využití HW prostředků CPLD). Implementace komparačního detektoru do obvodu XC9572 je tedy bezproblémová a je zde dostatek prostoru (HW prostředků) pro realizaci dalších pomocných funkcí. Z výsledků „fitace“ diferenčního detektoru je zřejmé, že implementace plného 8bitového detektoru v obvodu XC9572 (72 makrobuněk /360 p-termů) není reálná. Tato situace byla známa již v době výběru obvodu CPLD, přesto z cenových a rozměrových důvodů bylo rozhodnuto o použití obvodu XC9572.

²⁶ „Fitace“ – proces přiřazení logickým funkcím konkrétních HW prostředků CPLD

²⁷ Změna bitové šířky hranového detektoru bude vysvětlena v následující kapitole 3.5.4

3.5.4 PARAMETRIZACE BITOVÉ ŠÍŘKY DIFERENČNÍHO DETEKTORU

Pro snížení nároků diferenčního detektoru na implementaci v CPLD byl VHDL modul detektoru parametrizován z hlediska bitové šířky rozdílu první difference, vytvoření absolutní hodnoty první difference a prahování. Tímto způsobem je možné snížit nároky detektoru na implementaci v CPLD. Vliv snížení bitové šířky je zřejmý z tab. 3.3. Redukce bitové šířky pod hodnotu 5 je nevhodné z hlediska snížení rozlišení prahovací úrovně 32 hodnot.

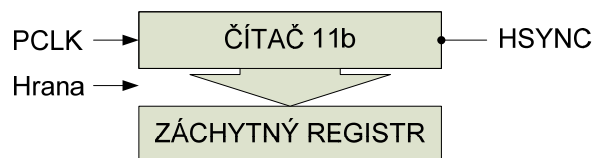
Z tohoto důvodu bylo další potřebné snížení nároků hranového detektoru dosaženo pomocí připojení fixní vstupní prahovací hodnoty, což vede ke zjednodušení komparátoru „LessThan0“. Tím bohužel přicházíme o možnost automaticky korigovat prahovací úroveň během procesu měření na základě získaného počtu detekovaných hran v snímku.

Pomocí výše uvedených zjednodušení hranového detektoru se podařilo realizovat pomocí obvodu XC9572 hranový diferenční detektor, který požadovaným způsobem detekuje hrany v reálných obrazových datech, jež jsou generována CMOS snímačem.

3.6 METODY MĚŘENÍ POZICE HRANY

V předchozí kapitole byl popsán realizovaný jednotný interface hranového detektoru (obr. 3.9). Nyní se budeme zabývat problematikou zachycení pozice (vzdálenost hrany od začátku řádku v pixelech) detekované hrany a její uložení v paměti mikroprocesoru, kde má být nadále zpracovávána.

Protože se principiálně jedná o měření časového intervalu od začátku řádku do vygenerované hrany, je možné využít metody používané v technické praxi pro měření času a frekvence.

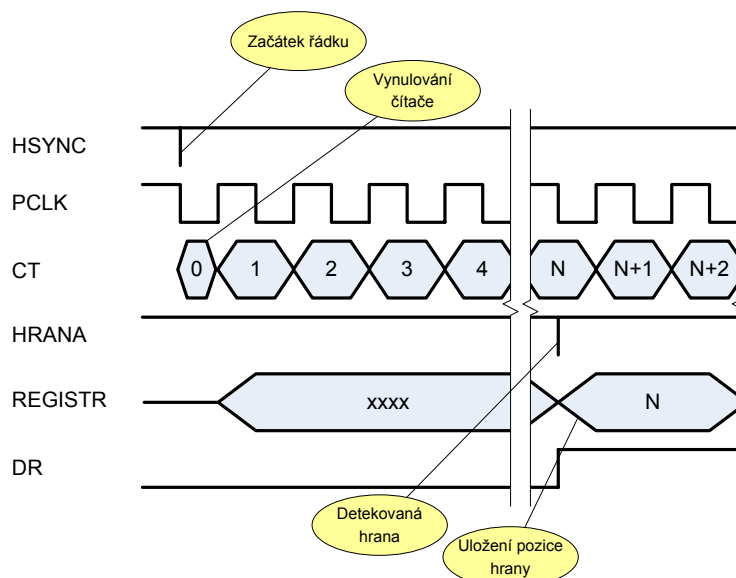


Obr. 3.15 Zachycení hrany pomocí čítače a záchytného registru

V nejjednodušším případě bychom tedy mohli použít například jedenáctibitový čítač (rozlišení CMOS snímače KAC9638 v horizontálním směru je 1288) signálu PCLK, který bude zastaven pomocí signálu „hranaUp“ nebo „hranaDn“.

Tímto způsobem bychom ovšem mohli v každém řádku uložit pouze jednu hranu. Jestliže doplníme čítač záchytným registrem dle obr. 3.15 a využijeme signál „hranaUp“, resp. „hranaDn“, pro přesání obsahu registru hodnotou v čítači, získáváme způsob měření pozice hrany, kde čítač může běžet neustále, a tak je možné detekovat více hran v jednom řádku. Čítač by byl nulován na začátku každého řádku signálem HSYNC.

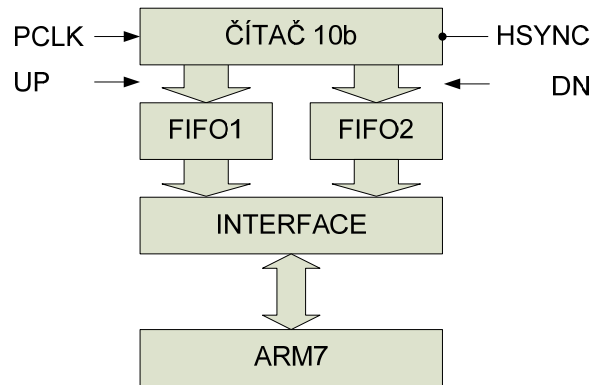
Po zapsání hodnoty do záchytného registru by přes zatím nedefinovaný interface (DR – data ready) mikroprocesor vyzvednul zachycenou hranu a až poté by bylo možné zachycení další hrany. Otázka rozlišení vzestupné a sestupné hrany bude diskutována později. Časový průběh popsané metody je znázorněna na obr. 3.16.



Obr. 3.16 Časový průběh signálů při zachycení hrany registrem

3.6.1 ZACHYCENÍ HRANY S VYUŽITÍM CPLD

Vzhledem k tomu, že hranový detektor je realizován v CPLD, bylo snahou implementovat obvody pro zachycení hrany také do CPLD. V rámci ověřování funkce zapojení videosenzoru byl pomocí dvou obvodů XC9572 realizován následující obvod (obr. 3.17) umožňující zachycení hrany, kdy první CPLD bylo využito jako hranový detektor a druhé jako obvod pro zachycení hrany a interface pro mikroprocesor.



Obr. 3.17 Obvod pro zachycení hrany s FIFO pamětí

Realizovaný obvod pro zachycení hrany obsahoval oddělenou FIFO paměť pro vzestupnou i sestupnou hranu. Velikost FIFO paměti byla 2x10bitů, což umožňovalo zachytit dvě vzestupné a dvě sestupné hrany. Motivací pro implementaci FIFO paměti byla snaha zachytit hrany, které následují po sobě ve velmi krátkých časových intervalech, a reakční doba mikroprocesoru by byla nedostačující. Realizovaný interface byl 8 bitový, kdy se pozice hrany přenášela v dvou krocích. V první fázi bylo přeneseno spodních 8 bitů pozice hrany a do druhého slova byla do horních nevyužitých bitů přidána informace o typu (vzestupná / sestupná) hrany.

Vzhledem k tomu, že se nepodařilo implementovat výše popsaný obvod pro zachycení hrany společně s hranovým detektorem v jediném obvodu XC9572, byl pro finální řešení videosenzoru zvolen následující způsob zachycení hrany.

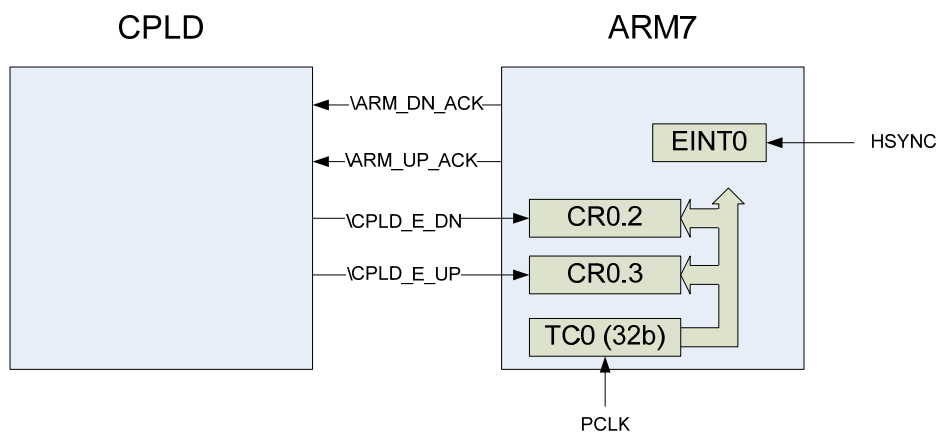
3.6.2 ZACHYCENÍ HRANY POMOCÍ „CAPTURE“ REGISTRU

Použitý mikroprocesor LCP2148 obsahuje dva 32 bitové čítače (lze samozřejmě využít i jako časovač), kde ke každému z čítačů jsou přidruženy čtyři 32bitové „capture“ registry. Čítač je tedy možné nakonfigurovat tak, aby při definované události („capture event“) došlo k naplnění daného „capture“ registru aktuální hodnotou čítače. Toto naplnění je vykonáno nezávisle na běhu programu, čímž je vyloučeno zpoždění, které by způsobilo problémy při využití přerušovacího systému.

Zdrojem „capture event“ může být:

- Vzestupná hrana na definovaném pinu.
- Sestupná hrana na definovaném pinu.

Dále je možné konfigurovat přerušovací systém tak, aby po naplnění „capture“ registru bylo vygenerováno přerušení, v kterém je možné vyzvednout hodnotu uloženou v registru.



Obr. 3.18 Způsob využití „capture“ registru pro zachycení hrany

Realizovaný způsob zachycení hrany je znázorněn na obr. 3.18. Na vstup čítače TC0 je přiveden signál PCLK. Nulování čítače je prováděno na konci¹ každého řádku obrazových dat externím přerušením EINT0 navázaného na sestupnou hranu signálu HSYNC. „Capture“ registry jsou plněny sestupnou hranou signálů \CPLD_E_UP, resp. \CPLD_E_DN. Signály \ARM_UP_ACK \ARM_DN_ACK jsou potvrzením mikroprocesoru o vyzvednutí pozice hrany. Tímto způsobem je zajištěno, že nedojde k přepsání „capture“ registru novou hranou během časového intervalu potřebného k vyvolání přerušení a vykonání obslužné rutiny.

3.7 DETEKCE MĚŘÍCIHO ŘÁDKU

V této kapitole bude popsán realizovaný způsob detekce měřícího řádku (tedy řádku, v kterém budou ukládány nalezené hrany). Tato detekce je prováděna na úrovni mikroprocesoru. Hranový detektor realizovaný v CPLD pracuje neu-

¹ Nulování čítače pixelů na konci řádku předpokládá konfiguraci CMOS snímače do režimu „Data Ready Mode“, kdy je signál PCLK generován pouze s platnými pixely. Po resetu je snímač konfigurován v režimu „Free Running Mode“.

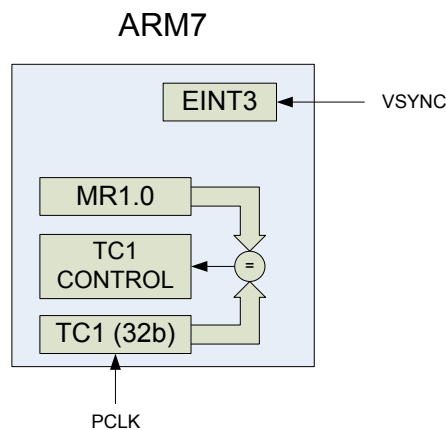
stále a mikroprocesor ukládá hrany pouze v řádcích, které jsou vybrány pro měření.

Pro účel detekce měřícího řádku je použit čítač TC1, na jehož vstup je přiveden signál HSYNC. Čítač TC1 tedy pracuje jako čítač řádků. Jeho nulování je prováděno na konci každého snímku externím přerušením EINT3 navázaného na sestupnou hranu signálu VSYNC.

Další funkcí čítačů u použitého mikroprocesoru LPC2148 umožňují „match“ registry (každý čítač obsahuje čtyři). Hodnota „match“ registru je porovnávána s hodnotou odpovídajícího čítače - viz obr. 3.19. Jestliže dojde k jejich rovnosti, mikroprocesor je schopný reagovat následujícími způsoby:

- Vygenerování přerušení.
- Resetování (vynulování) odpovídajícího čítače.
- Zastavení odpovídajícího čítače.
- Nastavení odpovídajícího výstupního pinu definovaným způsobem.

Pro účely detekce měřeného řádku je při rovnosti „match“ registru s TC1 generováno přerušení, v jehož obslužné rutině je povoleno ukládání detekovaných hran.

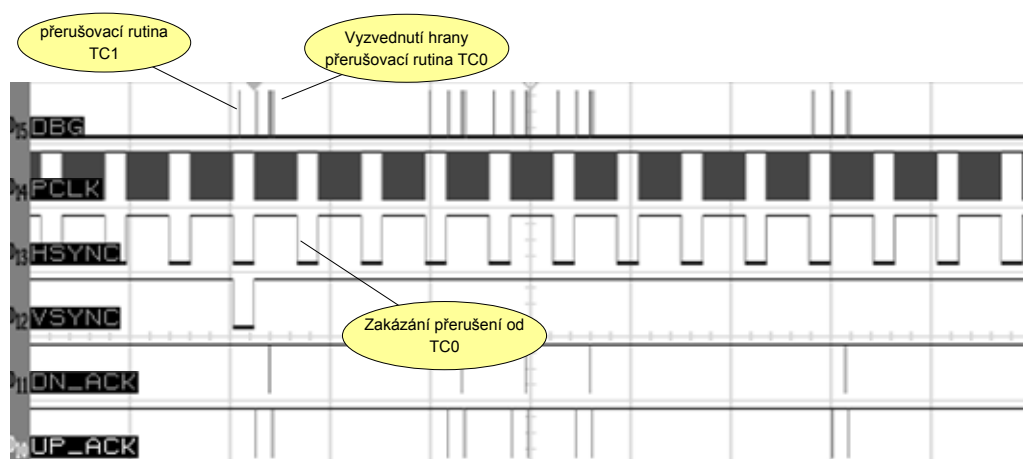


Obr. 3.19 Detekce měřícího řádku

Z následujícím snímkem z osciloskopu (obr. 3.20) je patrný způsob detekce měřících řádků. V uvedeném příkladě jsou jako měřící linky nastaveny tyto řádky: 1, 4, 5, 6 a 10. Pomocí ladícího pinu DBG (signál v první řádce) je zobrazeno vykonání obslužné rutiny čítačů TC1 a TC2.

S příchodem sestupné hrany VSYNC dojde k vynulování čítače řádků TC1 a naplnění „match“ registru hodnotou prvního měřeného řádku sníženou o jedna. V našem případě je tedy „match“ registr naplněn hodnotou nula, čímž automaticky dojde k vyvolání přerušovací rutiny čítače řádků TC1. V této rutině je povoleno přerušování čítače pixelů TC0. Během prvního řádku je tedy při události detekované hrany vyvolána obslužná rutina čítače pixelů TC0 a pozice detekované hrany je uložena.

Se sestupnou hranou HSYNC (konec řádku) je vyvoláno přerušování¹, které zakáže přerušování od čítače pixelů TC0, a proto v druhém řádku nejsou detekované hrany ukládány. Během prvního volání přerušovací rutiny čítače řádků TC1 byl také naplněn „match“ registr novou hodnotou (v našem případě hodnotou 3). Po skončení třetí řádky tedy dojde k druhému vyvolání přerušovací rutiny čítače řádků TC1 a celá situace se opět opakuje.



Obr. 3.20 Snímek z osciloskopu – detekce měřícího řádku

3.8 DETEKCE OBJEKTU V OBRAZE – PREPROCESSING HRAN

V této kapitole bude popsán způsob detekce objektu v obraze z pozic naměřených hran na základě znalosti očekávaných rozměrů objektu. Ty jsou definovány v „normálovém objektu“. Způsob definice „normálového objektu“ a popis způsobu přiřazení hran k objektu, včetně ilustračního obrázku (obr. 2.3), je uveden v kapitole 2.2.1.3. Detekce objektu spočívá ve vyhledání nejvhodnějších hran

¹ Toto přerušování není pro zvýšení přehlednosti pomocí DBG pinu zobrazeno

a jejich přiřazení k objektu. Toto přiřazení je provedeno na základě znalosti rozměrů „normálového objektu“ metodou „best fit“.

Protože bylo cílem navrhnout způsob detekce objektu, který bude možné konfigurovat z aplikace v nadřazeném PC bez nutnosti změny firmwaru v mikroprocesoru, byl zvolen přístup popisu detekce objektu pomocí tzv. „fit“ příkazů. Tyto příkazy jsou vykonávány během detekce objektu sekvenčně (s definovaným pořadím vykonání příkazů). Výsledkem „fit“ příkazu je nalezená hrana, resp. dvojice či trojice hran, které nejlépe odpovídají kritériu popsanému v parametrech příkazu. Počet „fit“ příkazů je omezena na 256. Jedná se tedy o vytvoření jakéhosi programovacího jazyka, který popisuje způsob nalezení objektu v obraze. Tímto způsobem je umožněno měnit vlastnosti chování videosenzoru bez nutnosti změny firmwaru mikroprocesoru.

3.8.1 PŘÍKAZ FIT1

Tento příkaz vyhledá hranu s pozicí definovanou v kritériu příkazu. Vzdálenost hrany je absolutní, tedy vztažená k prvnímu pixelu snímače. Pro použití příkazu je třeba nastavit následující parametry:

- Očekávaná vzdálenost hrany od prvního pixelu.
- Řádek, v kterém bude provedeno hledání.
- Typ hledané hrany (sestupná, vzestupná).
- Kritérium pro verifikaci: Tímto parametrem je možné určit toleranci pro akceptaci výsledku příkazu.

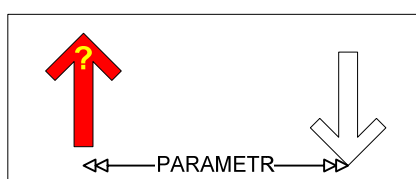


Obr. 3.21 Příkaz FIT1

3.8.2 PŘÍKAZ FIT2

Tento příkaz vyhledá hranu, jejíž vzdálenost od jiné referenční bude nejlépe odpovídat hodnotě definované v parametru příkazu. Hrany (hledaná a referenční) nemusí být ze stejného řádku. Pro použití příkazu je třeba nastavit následující parametry:

- Odkaz na referenční hranu: Hrana nalezená jedním z předchozích příkazů.
- Očekávaná horizontální vzdálenost hledané hrany od referenční hrany. Pokud chceme hledat hranu, která je nalevo od referenční hrany, zadáme záporné číslo.
- Řádek, v kterém bude provedeno hledání.
- Typ hledané hrany (sestupná, vzestupná).
- Kritérium pro verifikaci: Tímto parametrem je možné určit toleranci pro akceptaci výsledku příkazu.



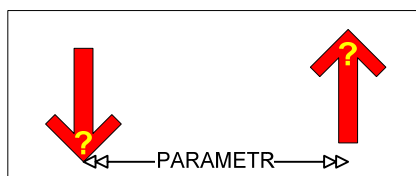
Obr. 3.22 Příkaz FIT2

3.8.3 PŘÍKAZ FIT3

Výstupem tohoto příkazu jsou dvě hrany, jejichž horizontální vzdálenost nejlépe odpovídá hodnotě definované v parametru. Hrany nemusí být ze stejného řádku, je však nezbytné oba řádky definovat.

Pro použití příkazu je třeba nastavit následující parametry:

- Očekávaná horizontální vzdálenost hledaných hran.
- Řádek pro první hranu, řádek pro druhou hranu.
- Typy hrany (sestupná, vzestupná) - mohou být různé.
- Kritérium pro verifikaci: Tímto parametrem je možné určit toleranci pro akceptaci výsledku příkazu FIT3.



Obr. 3.23 Příkaz FIT3

4 VÝSLEDKY PRÁCE, MĚŘENÍ PARAMETRŮ VIDEOSENZORU

4.1 PROVEDENÁ MĚŘENÍ

4.1.1 ZACHYCENÍ HRANY – MĚŘENÍ ODBĚRU DVOU BLÍZKÝCH HRAN

Cílem měření bylo ověřit schopnost metody popsané v kapitole 3.6.2, zachytit dvě blízké hrany a určit nejmenší možnou vzdálenost mezi dvěma hranami aby byly mikroprocesorem korektně uloženy do datové paměti RAM. Uložení hran do paměti mikroprocesoru je prováděno pomocí obslužné rutiny čítače TC0. Test byl proveden pro rutinu napsanou v jazyku C obsahující následující úkony:

- Test typu přijaté hrany.
- Uložení hrany do datového pole v paměti mikroprocesoru.
- Inkrementace pointeru do datového pole + kontrola mezí.
- Potvrzení mikroprocesoru signálem ARM_DN_ACK resp. ARM_UP_ACK.
- Test, zda nepřišla během doby zpracování rutiny zároveň hrana opačné polarity – v pozitivním případě následuje její vyzvednutí.
- Obsluha přerušovacího systému.

Měření bylo provedeno pomocí testovacího VHDL modulu, kterým byl nahrazen hranový detektor za následujících podmínek:

- PCLK: 5 MHz.
- Přerušování od čítače TC0 nastaveno jako FIQ (má tedy nejvyšší prioritu).
- Frekvence jádra mikroprocesoru: 60MHz.

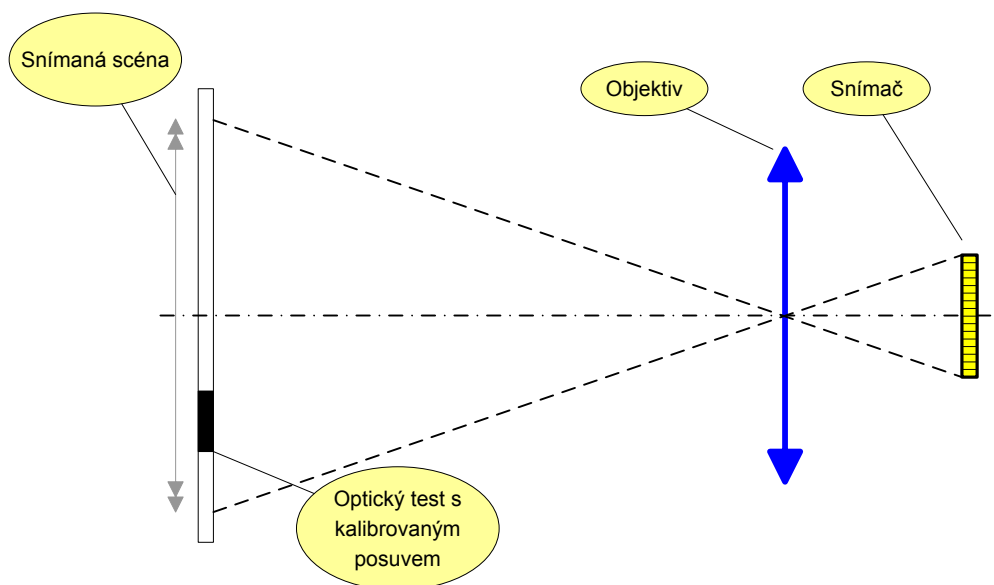
4.1.1.1 VÝSLEDEK MĚŘENÍ

- Minimální vzdálenost dvou hran byla změřena s výsledkem: 15 pixelů pro korektní načtení do paměti, čemuž odpovídá časový interval 3 μ s.

Z tohoto časového intervalu zabere 1 μ s reakce mikroprocesoru na přerušování, proto pokud během obsluhy přerušování přijde třetí hrana, obslužná rutina automaticky zopakuje načtení nové hrany, čímž se ušetří procesorový čas nutný na vyvolání a obsluhu přerušování. Minimální vzdálenost dvou hran pro korektní odběr by bylo možno zmenšit pomocí optimalizace rutiny v jazyku symbolických adres a dále pomocí využití třetího (volného) „capture“ registru.

4.1.2 PŘEVODNÍ CHARAKTERISTIKA VIDEOSENZORU PŘI MĚŘENÍ POLOHY OBJEKTU

Cílem měření bylo určit chybu měření polohy objektu pomocí realizovaného videosenzoru. Pro měření bylo použito následující uspořádání měřicí soustavy (obr. 4.1):



Obr. 4.1 Uspořádání měřicí soustavy

Měření bylo provedeno za následujících podmínek:

- Objektiv: Tevideon 1.4/25.
- Vzdálenost videosenzoru od optického testu: 800mm.
- Kalibrovaný posuv testu v snímané scéně: Rozlišení polohy testu - 1 μ m.
- Rozsah měření: 10 pozic optického testu na v rozsahu 225mm - odpovídá zornému poli videosenzoru.
- Šířka optického testu: 10mm.
- Rozložení měřících řádků: 20 měřících řádků lineárně rozložených na oblasti 40mm.
- Hranový detektor: Diferenciální, bitová šířka 6, prahovací úroveň 64.

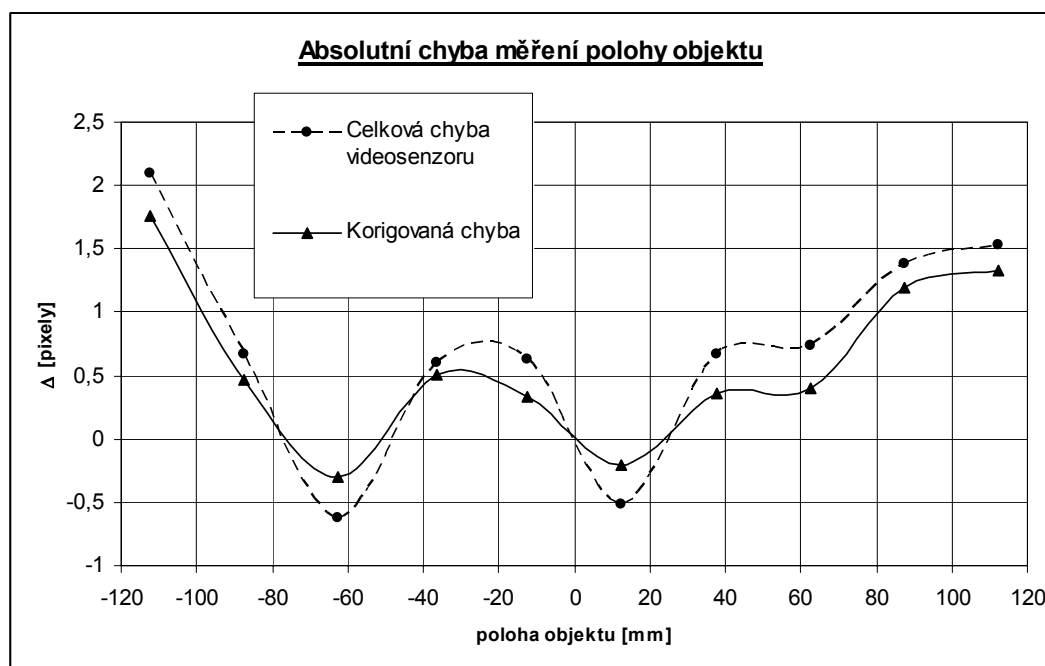
4.1.2.1 METODA ZPRACOVÁNÍ NAMĚŘENÝCH DAT

- Aritmetickým průměrem vzestupné a sestupné hrany byl získán střed optického testu. Tato hodnota byla dále použita pro určení chyby měření.
- Pro jednotlivé polohy optického testu bylo provedeno proložení přímkou.

- Z těchto desíti získaných proložení byl určen sklon testu, který byl použit k výpočtu pravé hodnoty měření v jednotlivých bodech měření.
- Celková chyba měření polohy videosenzorem byla získána jako rozdíl vypočtené pravé hodnoty měření a naměřených hodnot v jednotlivých bodech měření.
- Chyba měření polohy objektu způsobená vlivem hranového detektoru byla určena z proložení přímkou pro jednotlivé polohy testu.
- Následně byla od celkové chyby videosenzoru odečtena chyba hranového detektoru. Touto korekcí jsme získali chybu, která zahrnuje ostatní vlivy, jako jsou šumy snímače, kvantování, geometrie zobrazovací soustavy a další.

4.1.2.2 VÝSLEDEK MĚŘENÍ

- Chyba měření polohy objektu: 2,1 pixelu, čemuž odpovídá 0,16% z rozsahu.
Rozložení hodnoty chyby měření na měřeném rozsahu je zobrazen v grafu (obr. 4.2).



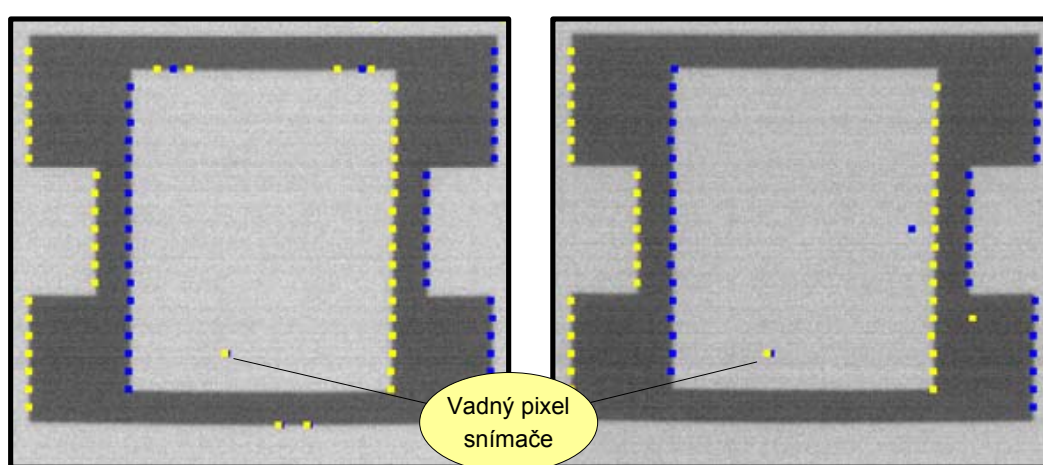
Obr. 4.2 Chyba měření polohy objektu

Z grafu je patrný vliv chyby způsobené geometrií zobrazovací soustavy v okrajích snímané scény.

4.2 ZHODNOCENÍ VÝSLEDKŮ A FUNKČNOSTI

4.2.1 DETEKCE A ZACHYCENÍ HRANY

Implementované hranové detektory mají vlastnosti shodné s výsledky získanými z simulace na obrazových datech a nic nebrání jejich použití na reálných obrazových datech v praxi. Při dostatečném nasvícení kontrastní scény je možné pro diferenční detektor použít prahovací hodnotu 64. V ostatních případech byla během vývoje videosenzoru používána prahovací hodnota 32. Výhodou navržených detektorů je možnost jejich implementace do obvodu XC9572.



Obr. 4.3 Ukázka výsledku hranové detekce (komparační a diferenční detektor)

Realizovaný způsob zachycení hrany pomocí „capture registru“ umožňuje zachycení vzestupné i sestupné hrany, jejich rozlišení a oddělené uložení pozic v paměti mikroprocesoru. Užití dvou „capture registrů“ zajišťuje možnost zachycení vzestupné i sestupné hrany s nulovou požadovanou vzdáleností mezi hranami, tím je umožněna detekce úzkých předmětů. Pro dvě po sobě příchozí hrany stejného typu je minimální časový interval pro korektní odběr obou hran $3\mu\text{s}$ (viz. kapitola 4.1.1).

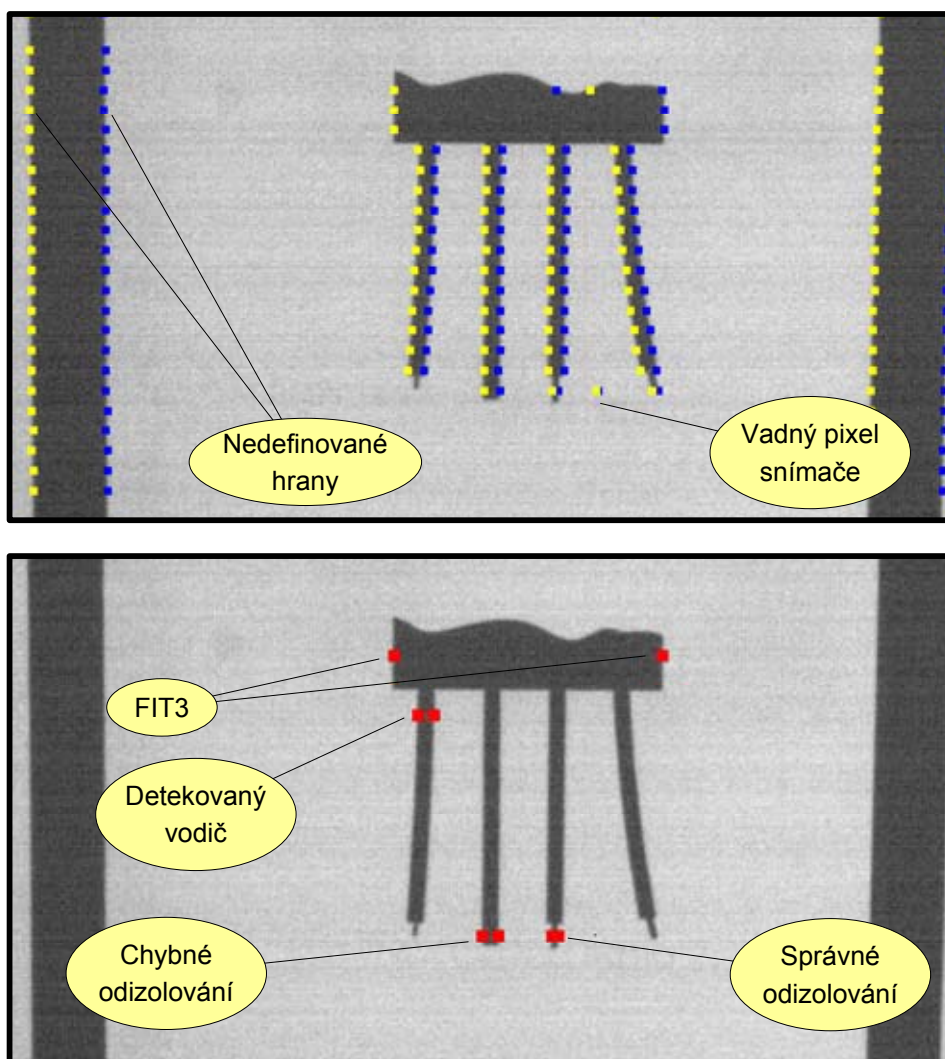
Pozice hran jsou v paměti mikroprocesoru uloženy (vzestupně seřazeny) v datovém poli společném pro všechny měřící řádky. Ukazatele na bloky hran příslušné jednomu měřicímu řádku jsou uloženy v tabulce měřících řádků, což umožňuje efektivní prohledávání.

4.2.2 DETEKCE OBJEKTU

Navržený způsob detekce objektu v obraze na základě znalosti „normálního objektu“ byl implementován a vyzkoušen na reálných obrazových datech. Při

správném zadání parametrů „normálového objektu“ algoritmus pracuje bezchybně. Na obr. 4.4 je zobrazena ukázka detekce objektu (jako optický test byl použit obrázek z aplikačního příkladu - kapitola 2.1.1).

Jako první byl použit příkaz FIT3, který detekoval plášť kabelu - tak byla získána dvojice hran. Levá hrana z této dvojice byla dále použita jako referenční pro detekci vodiče a kontrolu odizolování. Z výsledků je zřejmé, že je metodu možné použít i pro popis složitějších objektů.



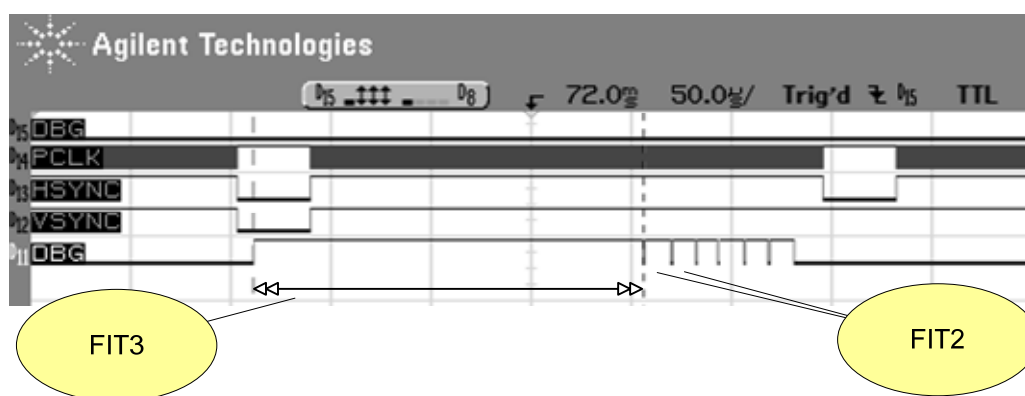
Obr. 4.4 Ukázka výsledku detekce objektu

4.2.2.1 VÝPOČETNÍ NÁROČNOST ALGORITMŮ PRO DETEKCI OBJEKTU

Implementované algoritmy v „fit“ příkazech 1 až 3 jsou závislé na počtu detekovaných hran, které je nutné při hledání hrany přiřazované k objektu otesto-

vat. Měření výpočetní náročnosti bylo provedeno pro scénu, která je po detekci hran redukována na průměrně 30 hran v měřícím řádku. Měřením byly získány následující výsledky:

- Příkaz FIT1: 13 μ s.
- Příkaz FIT2: 13,2 μ s.
- Příkaz FIT3: 195 μ s, při zpracování nejsou načítány nové hrany z „capture“ registru.
- Příkaz FIT3: 270 μ s, při zpracování jsou načítány nové hrany z „capture“ registru.



Obr. 4.5 Snímek z osciloskopu – měření délky trvání „fit“ příkazů

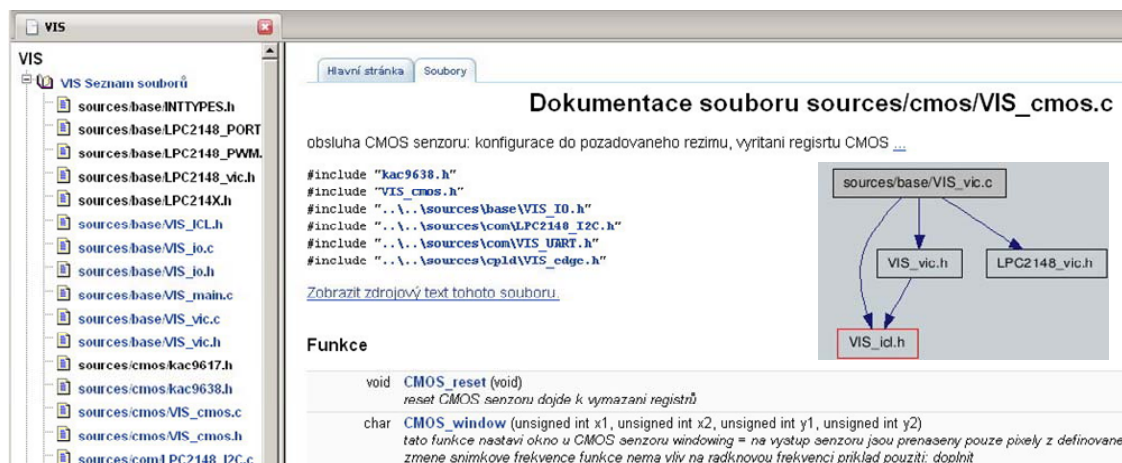
Naměřené časy potřebné k detekci objektu ukazují, že pro dané podmínky (průměrně 30 hran na řádek, horizontální rozlišení snímače 1280, $f_{PCLK} = 5\text{MHz}$) je možné provést detekci objektu během času potřebného k načtení hran z jednoho řádku. Tato skutečnost ukazuje na možnost použití algoritmu pro on-line měření, kdy bude videosenzor konfigurován jako rychlá řádková měřící kamera. Pokud bude videosenzor konfigurován jako víceřádkový, kdy alespoň některé řádky nebudou měřící, je možné použít i výpočetně náročnější algoritmy jako FIT3. Uvedené algoritmy nejsou optimalizované, proto se opět nabízí možnost implementace v jazyce symbolických adres.

4.2.3 DOKUMENTACE ZDROJOVÝCH KÓDŮ PRO MIKROPROCESOR ARM7

Jedním z osobních cílů při tvorbě firmware pro mikroprocesor ARM7 bylo již v průběhu psaní kódu tvořit kvalitně komentovaný přehledně, rozčleněný projekt. Ten je vytvořen v jazyce C s následujícími zásadami:

- Projekt je tvořen modulárním způsobem.

- Každá funkce je dokumentovaná syntaxí dokumentačního systému DOXYGEN. U složitějších funkcí je uveden příklad použití.
- Terminálové výpisy a ladící kódy jsou psány pomocí podmíněného překladu.



Obr. 4.6 Ukázka výstupu HTML dokumentace z systému DOXYGEN

5 CHYBY, NÁMĚTY A POZNÁMKY

5.1 SEZNAM CHYB A NAVRŽENÉ ZPŮSOBY ŘEŠENÍ

Následuje seznam chyb (tab. 5.1) vzniklých při návrhu desky plošných spojů s navrženými způsoby řešení.

CY7C68013A (U4) - PIN 27	
Popis chyby	Jedná se o rezervovaný pin, který má být dle dokumentace uzemněn. V současném návrhu desky je nezapojen.
Řešení v současné revizi	Propojeno propojkou kapacitor C8.
Návrh řešení pro další revizi	Oprava DPS.
EEPROM (U5) - PIN 7	
Popis chyby	Jedná se o pin WP (write protect). V současném návrhu je připojen na VCC, tudíž není povolen zápis do paměti.
Řešení v současné revizi	Odpojení pinu – paměť obsahuje interní pull down.
Návrh řešení pro další revizi	Oprava DPS.
SIGNÁL DTR (J5) - PIN 2	
Popis chyby	Chybějící pull-up.
Řešení v současné revizi	Doplňen přímým naletováním na rezistory R30 a R31.
Návrh řešení pro další revizi	Oprava DPS.
SIGNÁL RTS (J5) - PIN 3	
Popis chyby	Chybějící pull-up.
Řešení v současné revizi	Signál je veden přes CPLD, kde je provedena korekce chyby.
Návrh řešení pro další revizi	Oprava DPS.
D0:7 (J9)	
Popis chyby	Datové vodiče jsou připojeny spodní část datové sběrnice CMOS senzoru.
Řešení v současné revizi	Přemapování výstupů CMOS (pro 10bit výstup KAC9638 je důsledek chyby redukování sběrnice na 6bit).
Návrh řešení pro další revizi	Oprava DPS: D0:7(VIS) – D9:2(CMOS).
MULTIPLEXOR (U1) – ŘÍDÍCÍ VSTUPY A,B	
Doporučená úprava	Prohodit signály na vstupech A,B.
SIGNÁLY/CPLD_E_UP A/CPLD_E_DN	
Doporučená úprava	Umožnit připojení logického analyzátoru.

Tab. 5.1 Seznam chyb a návrhy řešení

5.2 NÁMĚTY PRO DALŠÍ VÝVOJ VIDEOSENZORU

Následuje seznam nerealizovaných námětů pro rozvoj SW mikroprocesoru, které se mi z časových důvodů nepodařilo realizovat. :

- „Interrupt nesting“: upravit obslužnou rutinu UART0, aby mohla být přerušena přerušováními s vyšší prioritou. Tato úprava je vhodným řešením při spuštění kontinuálního měření, které zamezí výpadkům měření v případě komunikace po rozhraní UART0.
- Implementovat do CPLD modul, který ošetří potvrzování přijetí hrany signály ARM_DN_ACK a ARM_UP_ACK.

- Implementovat „fit“ příkaz pro nalezení dvou hran o dané vzdálenosti, kdy mezi nimi není žádná jiná hrana.
- Implementovat možnost uložení konfigurace do FLASH paměti mikroprocesoru.
- Implementovat SPI rozhraní a modul pro ovládání desky VIS-IOx1.
- Implementovat do USB řadiče kontrolu CRC přijatých odpovědí z mikroprocesoru.

5.3 SHRNUÍ POZNATKŮ A „TRIKŮ“ ZÍSKANÝCH V RÁMCI VÝVOJE

- LPC FLASH UTILITY: K IAP programování mikroprocesoru byl použit nástroj LPC FLASH UTILITY. Při použití verzí 2.2.1 a 2.2.2 lze procesor LPC2148 naprogramovat do velikosti kódu 10kB. Při překročení této meze je třeba použít verzi 2.2.3.
- Senzor KAC9638, registr OPCTRL, bit 1: V dokumentaci je bit označen jako rezervovaný pro výrobní testování a měl by být nastaven do log. „1“. Nastavení do log. „0“ způsobí zablokování zápisu do „double buffered registrů“.
- Použití dlouhého (20mm) „header“ konektoru J9 osazeného v DPS takovým způsobem, aby piny vyčnívaly na obou stranách desky, umožnilo komfortní připojení logického analyzátoru současně s CMOS.
- Programování CPLD pomocí programátoru PRESTO: K programování jsem využíval aplikaci „JTAG Player for PRESTO“. SVF soubor se vytvoří v prostředí „Xilinx ISE“ pomocí přesměrování výstupu při programování do SVF souboru.
- Generování RTL schémat: RTL schémata VHDL modulů jsou vygenerována pomocí aplikace „Quartus II 7.1sp1 Web Edition“, která je produktem společnosti Altera.

6 ZÁVĚR

Cílem diplomové práce bylo navrhnout a realizovat mnohořádkový videosenzor pro základní bezdotyková měření rozměrů a polohy, kde měření je prováděno na základě zpracování řádkové obrazové informace. Pro videosenzor bylo definováno následující zadání:

- Zdroj obrazu: Snímač CMOS.
- Řídící jednotka: Mikroprocesor řady ARM7.
- Předzpracování obrazu (detekce hran) pomocí programovatelného logického obvodu - CPLD.
- Přenos obrazu na nadřazený počítač prostřednictvím rozhraní USB 2.0 pro účely polohování.
- Možnost konfigurace videosenzoru jako rychlou měřicí řádkovou kameru.

Při řešení diplomové práce bylo dosaženo následujících výsledků:

Na základě analýzy problematiky víceřádkového videosenzoru pro bezdotykové měření rozměru a polohy byl navržen modulární systém, který se skládá z následujících modulů: desky CMOS snímače, procesorové desky a desky vstupů-výstupů. Deska snímače obsahuje CMOS senzor KAC9638, který je zdrojem obrazu pro videosenzor. Tento modul je propojen s procesorovou deskou, která je jádrem celé diplomové práce. Deska obsahuje řídicí mikroprocesor LPC2148, CPLD XC9572 a USB řadič CY7C68013A. Deska vstupů-výstupů je konektorem připojena k procesorové desce a její ovládání zajišťuje mikroprocesor.

Realizovaný HW umožňuje následující funkce:

Pro účely polohování videosenzoru je obraz přenášen prostřednictvím USB řadiče na nadřazený počítač, který je použit jako zobrazovací monitor. Realizovaný software na PC umožňuje vedle zobrazení obrazu konfigurovat prostřednictvím řídicího mikroprocesoru CMOS snímač. Tak je možné měnit rozlišení videosenzoru.

Prvním krokem při zpracování obrazu za účelem měření rozměru je hranová detekce, která se provádí ve vybraných řádcích obrazových dat. Měřicí řádky je opět možné vybrat pomocí aplikace z nadřazeného počítače. Hranová detekce je

prováděna pomocí CPLD, do kterého byly implementovány následující dvě metody hranové detekce: komparační a diferenční.

Dalším blokem navázaným na hranový detektor je „capture“ jednotka mikroprocesoru, která zajišťuje zachycení pozice hrany a její uložení v paměti mikroprocesoru pro účely dalšího zpracování. Protože hranový detektor provádí svoji činnost neustále, je v mikroprocesoru pomocí čítače řádků zajištěno načtení detekovaných hran pouze pro měřicí řádky.

Posledním krokem při zpracování detekovaných hran je detekce objektu v obraze. Detekcí rozumíme výběr významných hran, které budou použity pro výpočet výsledného rozměru. Výběr hran je proveden na základě předchozí znalosti rozměrů měřeného objektu, jenž je definován pomocí sekvence příkazů popisujících detekci objektu - konfigurace je opět provedena z nadřazeného PC. Tímto způsobem je dosaženo univerzality videosenzoru bez nutnosti změny firmware mikroprocesoru. Výsledné rozměry a poloha měřeného objektu jsou vypočteny z pozic vybraných hran.

Realizovaný videosenzor má následující parametry:

- Chyba měření polohy objektu realizovaným videosenzorem je 0,16% z rozsahu.
- Možnost zachycení vzestupné i sestupné hrany s nulovou požadovanou vzdáleností mezi hranami.
- Pro dvě po sobě přichozí hrany stejného typu je minimální časový interval pro korektní odběr obou hran 3 μ s.
- Časová náročnost implementovaných algoritmů umožňuje použít videosenzor jako rychlou měřicí řádkovou kameru. Pro úlohu určení vzdálenosti dvou pruhů na optickém testu je možné dosáhnout následujících výsledků: opakovací frekvencí měření 3kHz, zpoždění výsledku měření 300 μ s. Tyto uvedené hodnoty odpovídají rozlišení snímače 1280 pixelů.

Cíle stanovené v úvodu této diplomové práce považuji za splněné. Výsledky práce ukazují, že po dokončení vývoje videosenzoru bude možné použít zařízení v praxi.

Na tomto místě bych rád poděkoval Ing. Janu Fischerovi, CSc. za vedení, odborné konzultace a korekci tohoto textu.

Dále bych rád poděkoval:

Tatíkovi za to, že rozhodl, že se budu zabývat elektrotechnikou a přihlásil mne na Střední průmyslovou školu elektrotechnickou v Pardubicích. Dále pak za pečlivé vypilování krabičky na videosenzor.

Mamince za nezměrnou podporu a lásku během celého mého mládí a studia, zvláště pak za všechny výborné svačiny na cesty do školy.

Všem svým sourozencům Marušce, Vítkovi a Pavlíkovi za krásné rodinné prostředí, taktí nerušení při psaní tohoto textu a za všechny laskominy donášené do mého pokoje.

Přátelům ze společenství FrMol a roverského kmene Olo Jump za přátelství a podporu po všechna ta léta.

Středoškolskému učiteli Ing. Jiřímu Nobilisovi za uvedení do nového vnímání elektrotechniky a za jeho nezapomenutelný pedagogický přístup s barevnými křídami.

7 SEZNAM OBRÁZKŮ

Obr. 1.1 Měření rozměru kruhového objektu řádkovou kamerou	4
Obr. 1.2 Polohování řádkové kamery	5
Obr. 1.3 Měření rozměru kruhového objektu víceřádkovým videosenzorem.....	8
Obr. 1.4 Windowing	9
Obr. 2.1 Aplikační příklad – kontrola odizolování vodičů	12
Obr. 2.2 Aplikační příklad – kontrola lahví na dopravníku	13
Obr. 2.3 Grafické znázornění normálového objektu	17
Obr. 2.4 Časový průběh důležitých signálů CMOS senzoru	23
Obr. 2.5 Princip hranové detekce.....	25
Obr. 2.6 Principiální blokové schéma videosenzoru.....	28
Obr. 3.1 Blokové elektrické schéma videosenzoru	31
Obr. 3.2 Schématické znázornění struktury napájení jednotlivých částí	34
Obr. 3.3 Multiplexovaný UART – blokové schéma.....	35
Obr. 3.4 Blokové schéma – datový tok mezi videosenzorem a PC	36
Obr. 3.5 Blokové schéma - řízení přenosu obrazu do PC.....	37
Obr. 3.6 Stavový automat pro řízení zápisu do FIFO paměti.....	38
Obr. 3.7 Snímek z osciloskopu – přenos čtyř snímků do PC	38
Obr. 3.8 Snímek z osciloskopu – přeplnění FIFO paměti	39
Obr. 3.9 RTL schéma modulu hranového detektoru	40
Obr. 3.10 RTL schéma komparačního hranového detektoru	41
Obr. 3.11 Komparační hranový detektor – ukázka simulace.....	43
Obr. 3.12 RTL schéma vytvoření první diference a prahování.....	45
Obr. 3.13 RTL schéma vytvoření příznaků DIFF2_SF a DIFF1_ZF.....	46
Obr. 3.14 Diferenční hranový detektor – ukázka simulace	49
Obr. 3.15 Zachycení hrany pomocí čítače a záchytného registru.....	51
Obr. 3.16 Časový průběh signálů při zachycení hrany registrem.....	52
Obr. 3.17 Obvod pro zachycení hrany s FIFO pamětí.....	53
Obr. 3.18 Způsob využití „capture“ registru pro zachycení hrany	54
Obr. 3.19 Detekce měřicího řádku	55
Obr. 3.20 Snímek z osciloskopu – detekce měřicího řádku.....	56
Obr. 3.21 Příkaz FIT1	57
Obr. 3.22 Příkaz FIT2	58
Obr. 3.23 Příkaz FIT3	58
Obr. 4.1 Uspořádání měřicí soustavy	60
Obr. 4.2 Chyba měření polohy objektu	61
Obr. 4.3 Ukázka výsledku hranové detekce (komparační a diferenční detektor) 62	
Obr. 4.4 Ukázka výsledku detekce objektu	63
Obr. 4.5 Snímek z osciloskopu – měření délky trvání „fit“ příkazů	64
Obr. 4.6 Ukázka výstupu HTML dokumentace z systému DOXYGEN	65

8 SEZNAM TABULEK

Tab. 2.1 Aplikační příklady měření videosenzorem	12
Tab. 2.2 Možné způsoby vyvolání odměru - TRIGGERy	16
Tab. 2.3 Základní typy měření pomocí víceřádkového videosenzoru	21
Tab. 2.4 Srovnání možností pro zpracování dig. signálu	27
Tab. 3.1 Seznam významných signálů videosenzoru	33
Tab. 3.2 Pravdivostní tabulka multiplexeru UARTu	35
Tab. 3.3 Parametry navržených hranových detektorů získané z „fitace“	50
Tab. 5.1 Seznam chyb a návrhy řešení	66

9 SEZNAM ZKRATEK A CIZÍCH SLOV

AREA SCAN CAMERA - kamera s plošným snímačem

AREA SENSOR -plošný snímač

ASIC - (Application Specific Integrated Circuit) zakázkový integrovaný obvod

CPLD - (Complex Programmable Logic Device) programovatelný logický obvod s konfigurací v paměti EEPROM

FIFO - (First In First Out) typ dvoubránové paměti RAM

FPGA - (Field Programmable Gate Array) programovatelné hradlové pole s konfigurací v paměti RAM

HDL - (Hardware Description Language) jazyk pro popis hardwaru

ISP/IAP - (In System Programming / In Application Programming)

JTAG - (Joint Test Action Group) port pro diagnostiku a programování logických obvodů

LINE SCAN CAMERA - řádková kamera

LINE SENSOR - řádkový snímač

OCR - (Optical Character Recognition) optické rozpoznávání znaků

Pattern match - rozpoznání vzoru v obraze

PIXEL - obrazový bod

PLC - (Programmable Logic Controller) programovatelný automat

PLD - (Programmable Logic Device) programovatelný logický obvod

VHDL - (Very high speed integrated circuits HDL)

10 SEZNAM POUŽITÉ LITERATURY

- [1] Dobruský, J.: *Kompaktní videosenzor*. Diplomová práce ČVUT-FEL, Praha 2007
- [2] Česák, P.: *Autonomní videoprocessor pro zpracování videosignálu a řízení osvětlovačů*. Diplomová práce ČVUT-FEL, Praha 2005
- [3] Fischer, J.: *Optoelektronické senzory a videometrie*. Skripta ČVUT, Praha 2002
- [4] UM10139-LPC2148 *User Manual*. Philips, 2005
- [5] KAC9617LongSpec *User Manual rev. 1.5*. KODAK, 2005
- [6] Webové stránky společnosti Basler AG,
URL: <http://www.baslerweb.com/index.html>
- [7] Webové stránky společnosti Cognex,
URL: <http://www.cognex.com/>
- [8] Souček, P.: *Použití rozhraní USB2.0 pro rychlý přenos obrazu*. Bakalářská práce ČVUT-FEL, Praha 2007
- [9] KAC9638LongSpec *User Manual rev. 1.9*. KODAK, 2004
- [10] Webové stránky společnosti Schäfter + Kirchhoff,
URL: <http://www.sukhamburg.de/>
- [11] Webové stránky společnosti Dalsa,
URL: <http://mv.dalsa.com/>
- [12] Webové stránky společnosti Keyence,
URL: <http://www.keyence.com/>

11 SEZNAM PŘÍLOH

12 Příloha A	i
Deska plošného spoje procesorové desky VISx1 (TOP).....	v
Deska plošného spoje procesorové desky VISx1 (BOT).....	v
Osazovací výkres procesorové desky VISx1 (TOP).....	vi
Kusovník desky VISx1	vii
13 Příloha B	viii
Deska plošného spoje procesorové desky VISx1 (TOP).....	xii
Deska plošného spoje procesorové desky VISx1 (BOT).....	xii
Osazovací výkres procesorové desky VISx1 (TOP).....	xiii
Kusovník desky VIS-IOx1	xiv
14 Příloha C	xv
Termo snímek DPS VIS-IOx1	xvi
15 Příloha D	xvii
Mechanický model videosenzoru – VISx1	xviii
Mechanický model videosenzoru – VIS-IOx1	xviii
Mechanický model videosenzoru – kompletní sestava.....	xix
16 Příloha E	xx
Fotodokumentace - procesorová deska VISx1.....	xxi
Fotodokumentace - deska vstupů – výstupů VIS-IOx1	xxi
Fotodokumentace - videosenzor – kompletní sestava	xxii
Fotodokumentace - videosenzor v krabičce.....	xxii
17 Příloha F	xxiii

12 PŘÍLOHA A

SCHÉMA, NÁVRH DPS A ROZPISKA SOUČÁSTEK PROCESOROVÉ DESKY VIDEOSENZORU

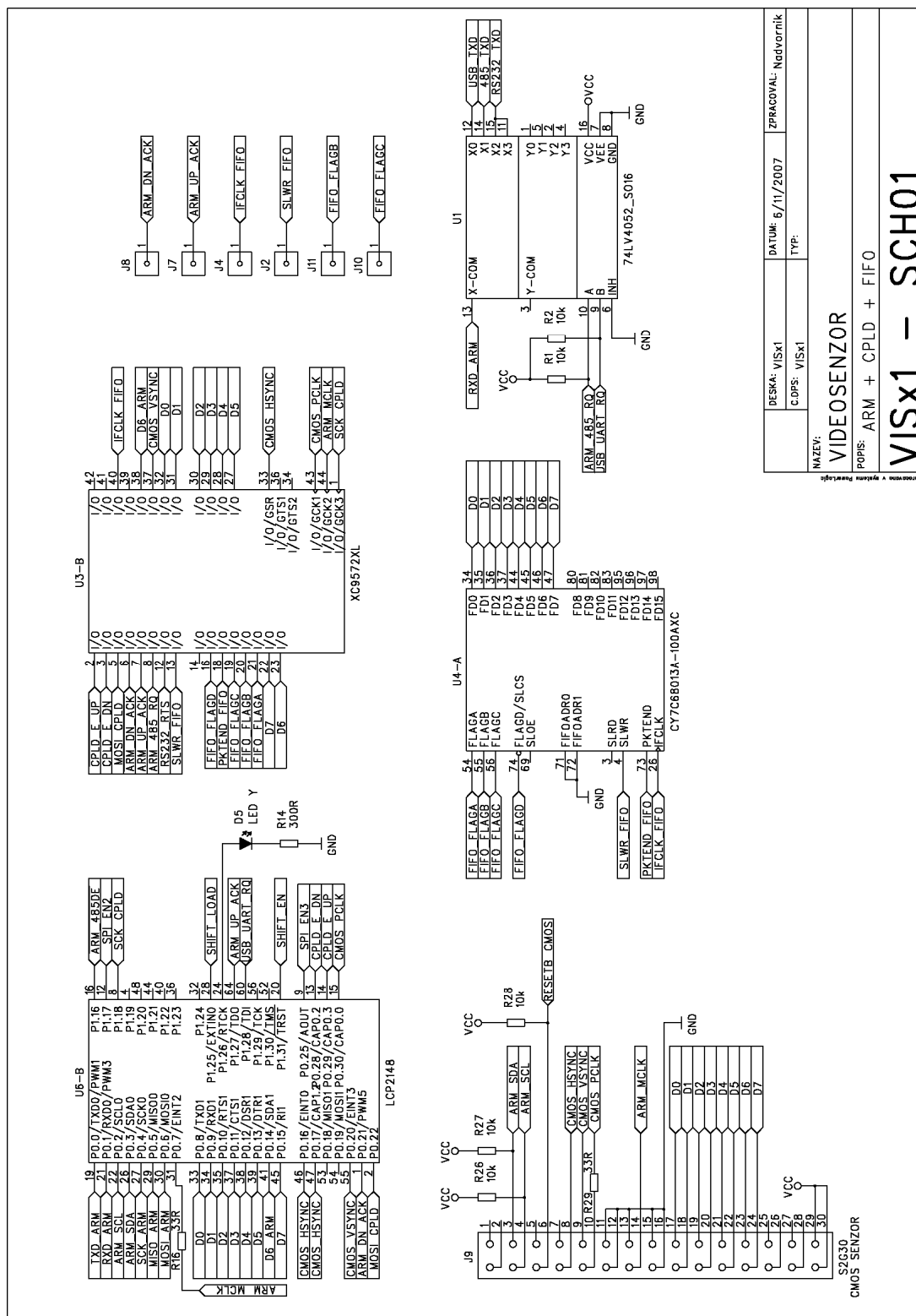


Schéma VISx1 - list 1/3

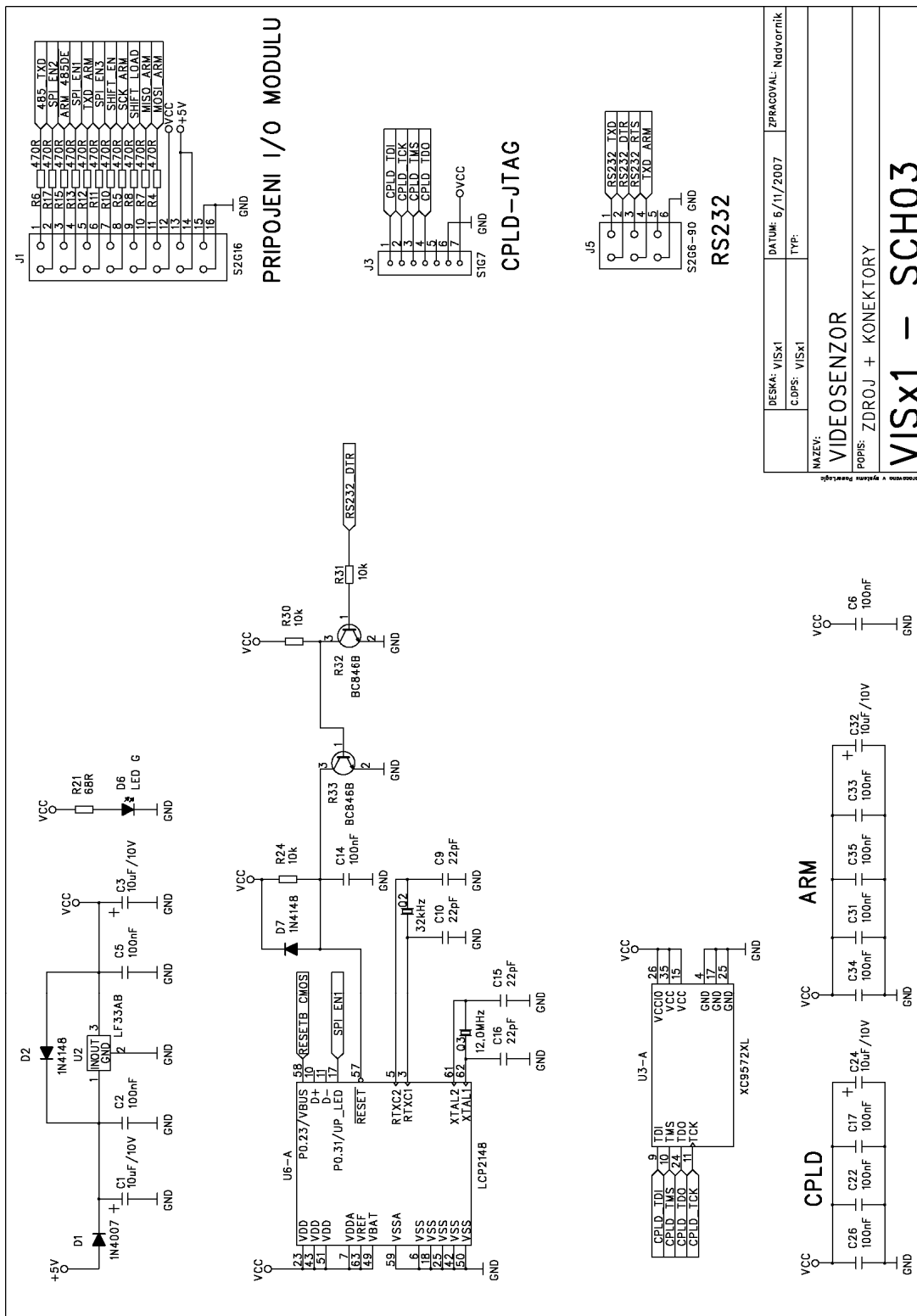
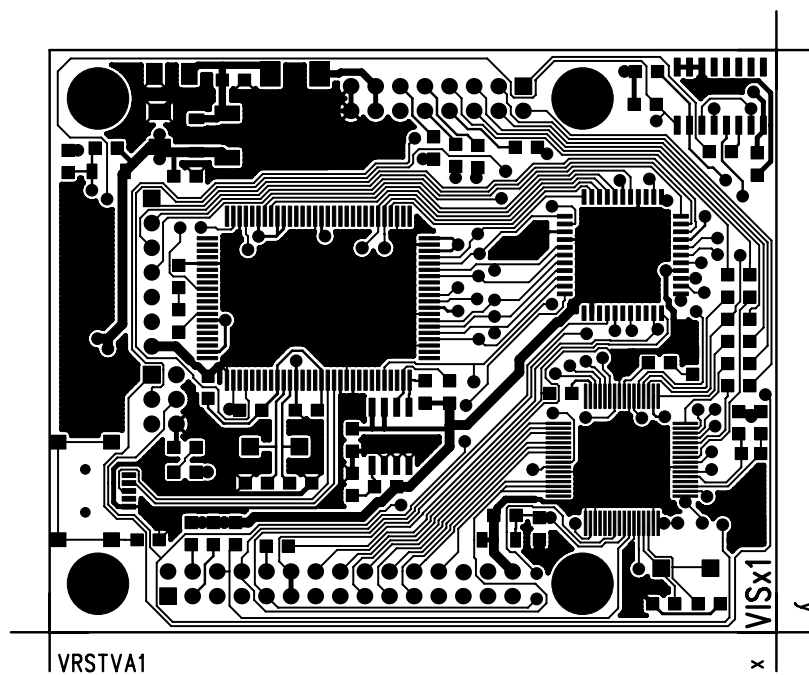
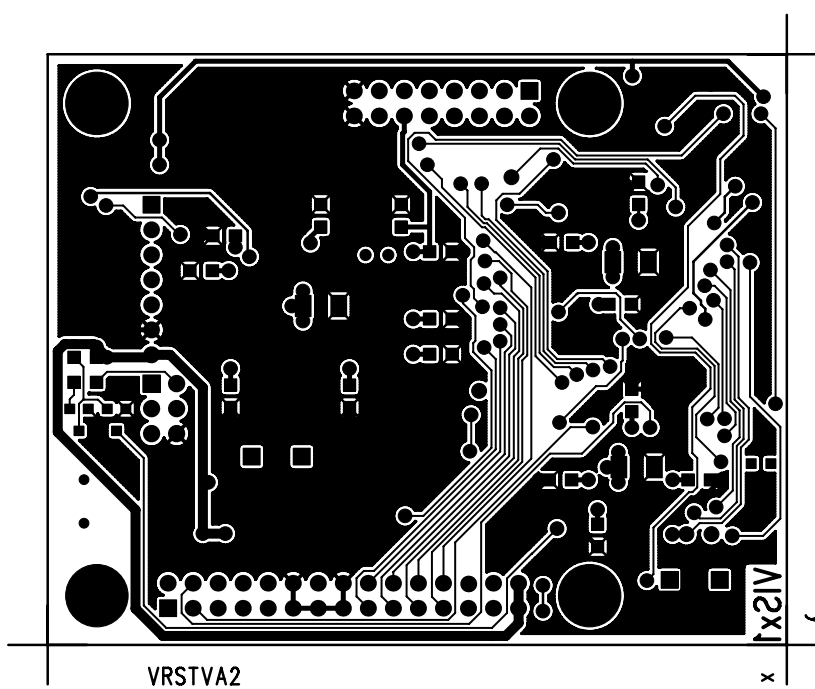


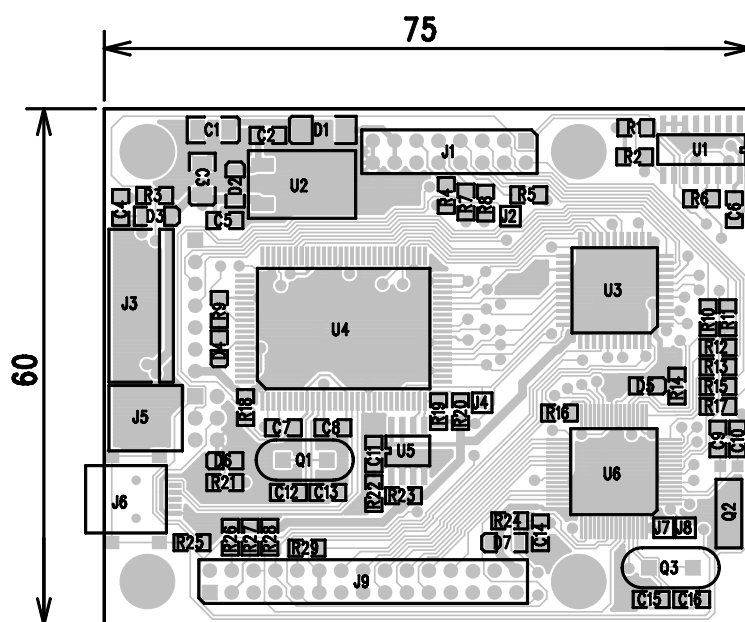
Schéma VISx1 - list 3/3



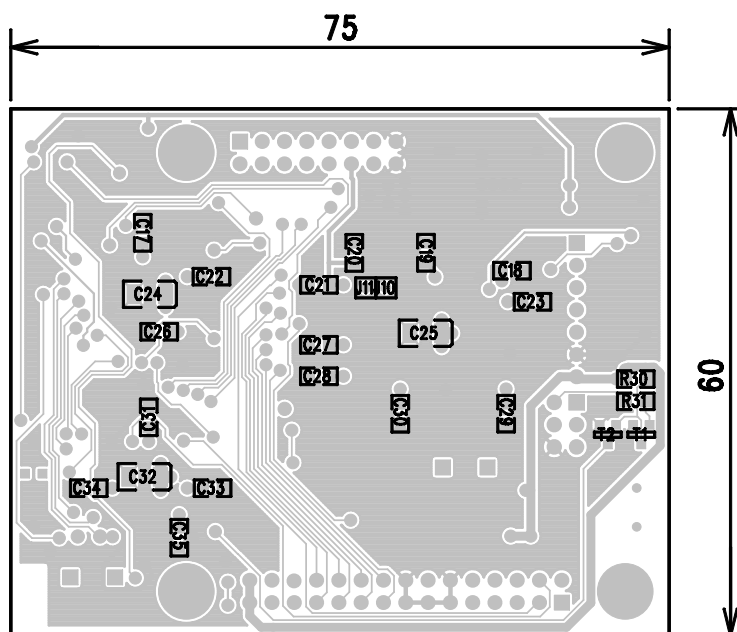
Deska plošného spoje procesorové desky VISx1 (TOP)



Deska plošného spoje procesorové desky VISx1 (BOT)



Osazovací výkres procesorové desky VISx1 (TOP)



Osazovací výkres procesorové desky VISx1 (BOT)

ROZPISKA SOUČÁSTEK				Číslo dokumentu:	
				01	
Název jednotky:		VISx1	Název DPS:		VISx1
			Č.v.osazení:		
		CELKOVÁ	Datum vzniku:		8.12.2007
	JEDNOTKA:	VISx1.01			Č.dok.
Pol.	Reference	Hodnota	Kód	Počet	Osaz.
1	*DPS	VISx1	VISx1	1	
2	C1 C3 C24 C25 C32	10uF/10V	CT 10uF/10V SMD-B	5	
3	C2 C4 C5 C6 C7 C8 C11 C14 C17 C18 C19 C20 C21 C22 C23 C26 C27 C28 C29 C30 C31 C33 C34 C35	100nF	C0805 100nF X7R 63V	24	
4	C9 C10 C12 C13 C15 C16	22pF	C0805 22pF X7R 63V	6	
5	D1	1N4007	1N4007 MELF	1	
6	D2 D3 D7	1N4148	LL4148 SOD80	3	
7	D4 D5	LED Y	LED 0805/20mA Y	2	
8	D6	LED G	LED 0805/20mA G	1	
9	J1	S2G16	S2G16	1	
10	J2 J4 J7 J8 J10 J11	S1G1	S1G1	6	
11	J3	S1G7	S1G7-90	1	
12	J5	S2G6-90	KON_S2G6-90	1	
13	J6	MINUSB	KON_MINIUSB	1	
14	J9	S2G30	S2G30	1	
15	Q1	24,0MHz	PKS-J 24,0000MHz HC49/U3	1	
16	Q2	32kHz	PKJ-S 32,768kHz KX38	1	
17	Q3	12,0MHz	PKS-J 12,0000MHz HC49/U3	1	
18	R1 R2 R3 R22 R23 R24 R25 R26 R27 R28 R30 R31	10k	R0805 10k 5%	12	
19	R4 R5 R6 R7 R8 R10 R11 R12 R13 R15 R17	470R	R0805 470R 5%	11	
20	R9 R14	300R	R0805 300R 5%	2	
21	R16 R29	33R	R0805 33R 5%	2	
22	R18	10R	R0805 10R 5%	1	
23	R19 R20	2k2	R0805 2k2 5%	2	
24	R21	68R	R0805 68R 5%	1	
25	R32 R33	BC846B	BC846B SOT23	2	
26	U1	74LV4052_SO16	USMD_74LV4052_SO16	1	
27	U2	LF33AB	LF33AB_DPAK	1	
28	U3	XC9572XL	XC9572XL	1	
29	U4	CY7C68013A-100AXC	CY7C68013A-100AXC	1	
30	U5	24CXX	24CXX_SO8	1	
31	U6	LCP2148	LCP2148	1	

Kusovník desky VISx1

13 PŘÍLOHA B

SCHÉMA NÁVRH DPS A ROZPISKA SOUŠÁSTEK DESKY VSTUPŮ-VÝSTUPŮ VIDEOSENZORU

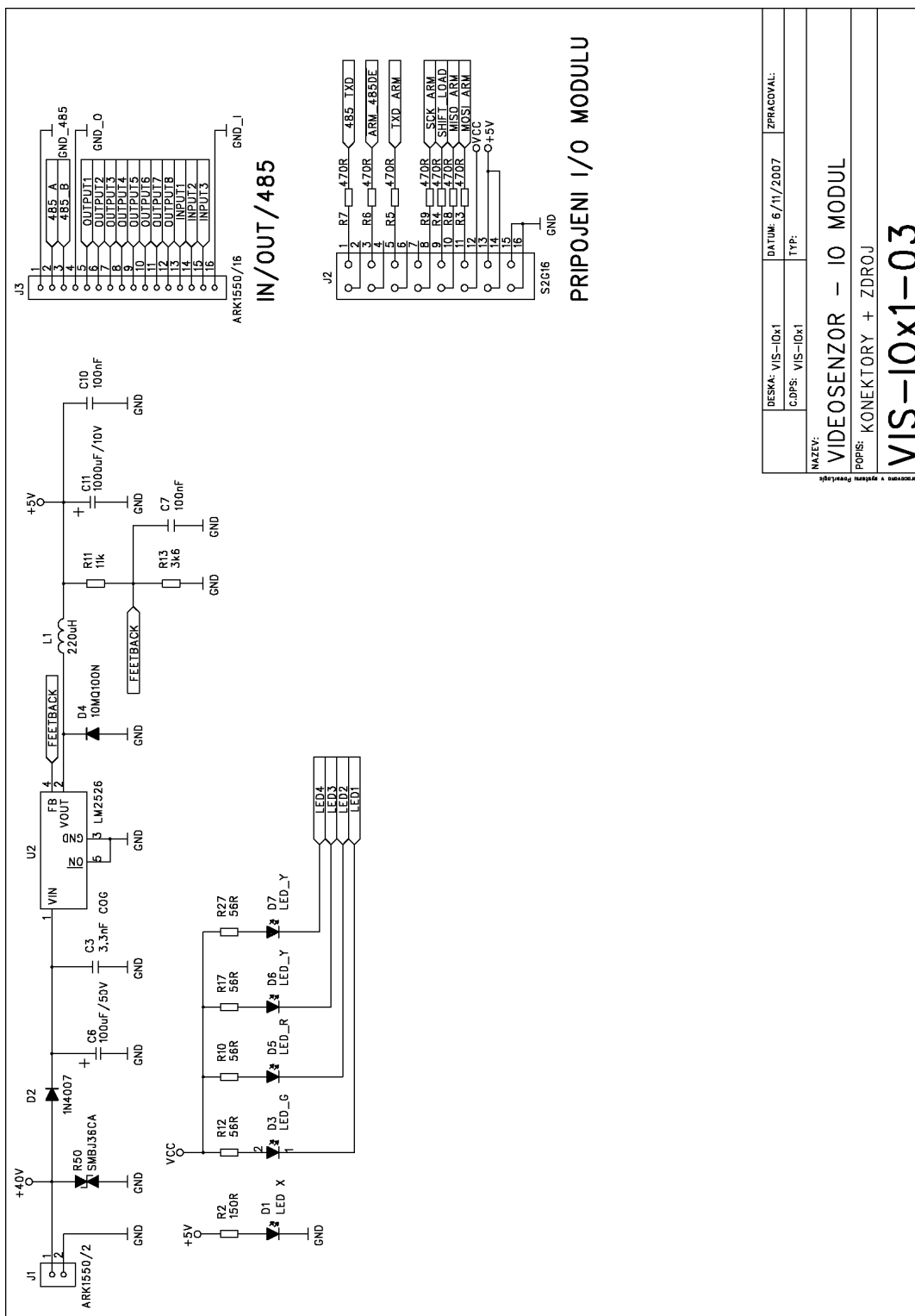


Schéma VIS-IOx1 - list 2/3

DESKA: VIS-IOx1	DATUM: 6/11/2007	ZPRACOVAL:
C.DPS: VIS-IOx1	TPP:	
NAZEV: VIDEOSENZOR - IO MODUL		
POPIS: KONEKTORY + ZDROJ		
VIS-IOx1-03		

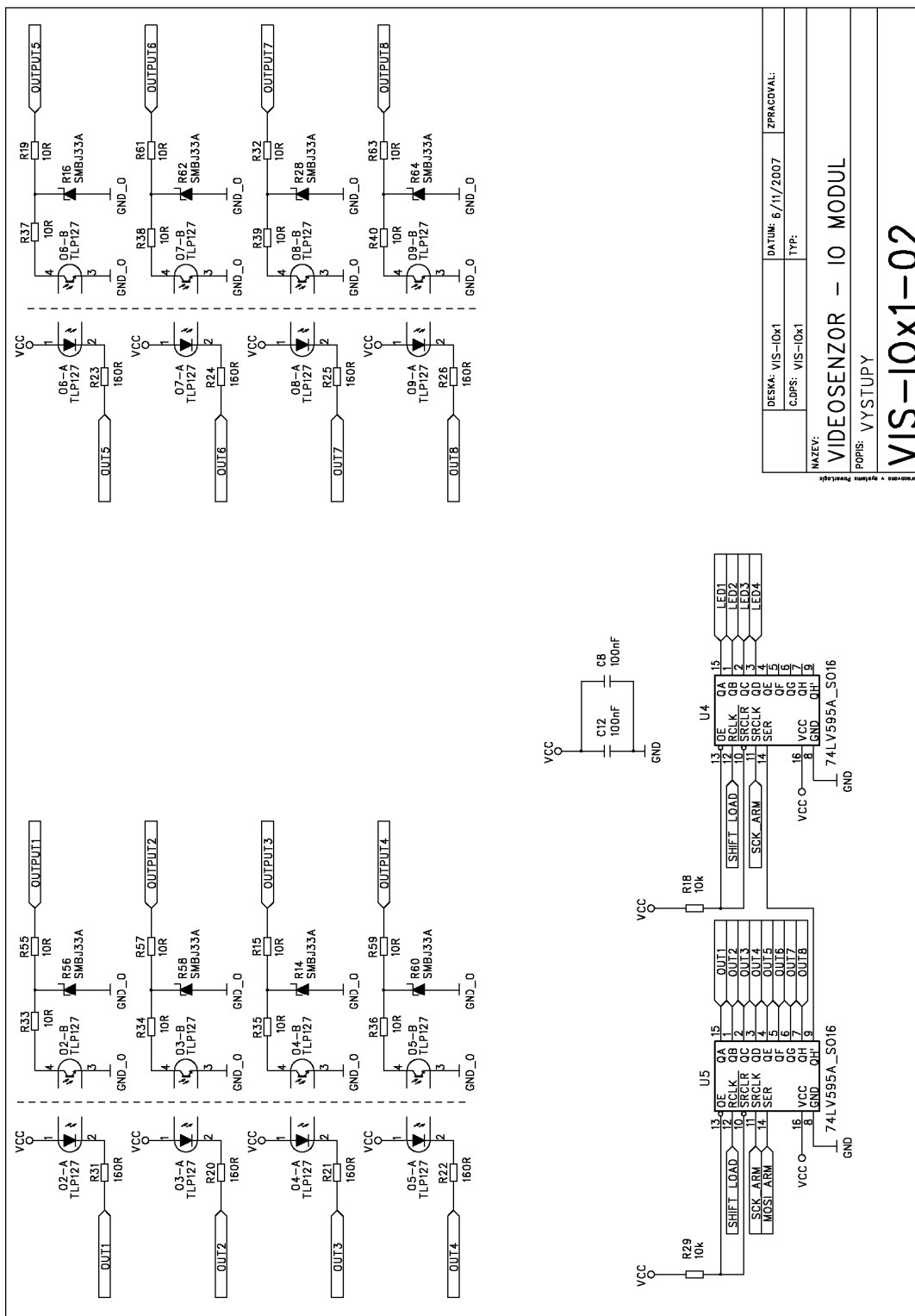
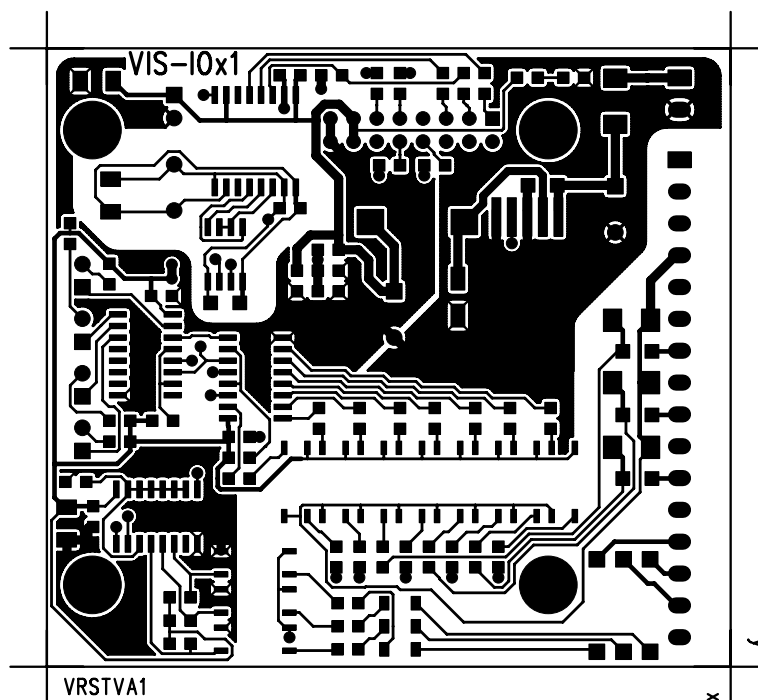
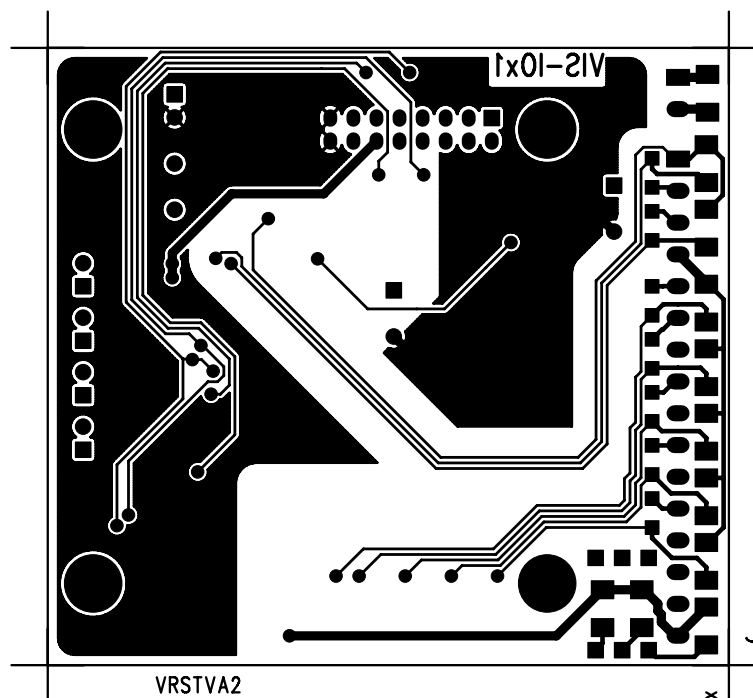


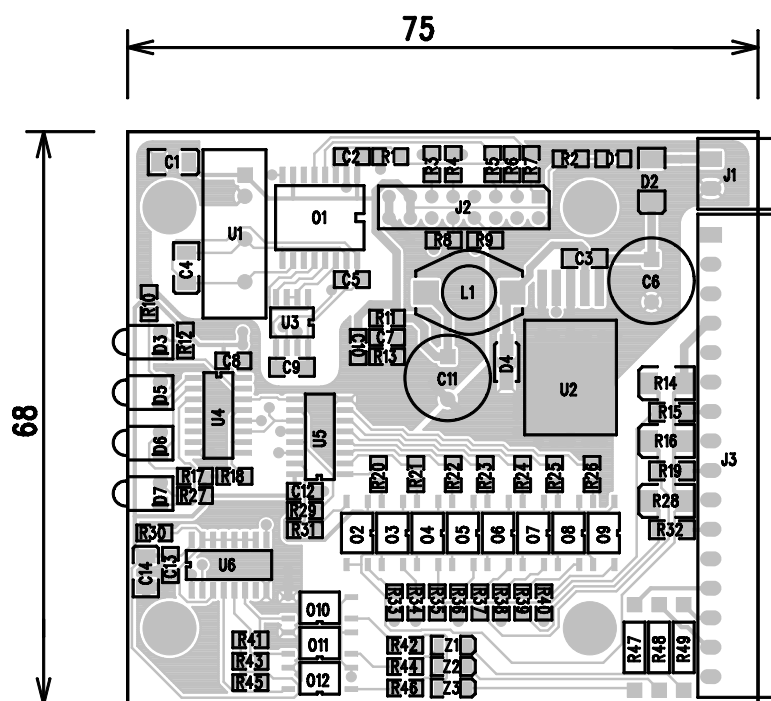
Schéma VIS-IOx1 - list 3/3



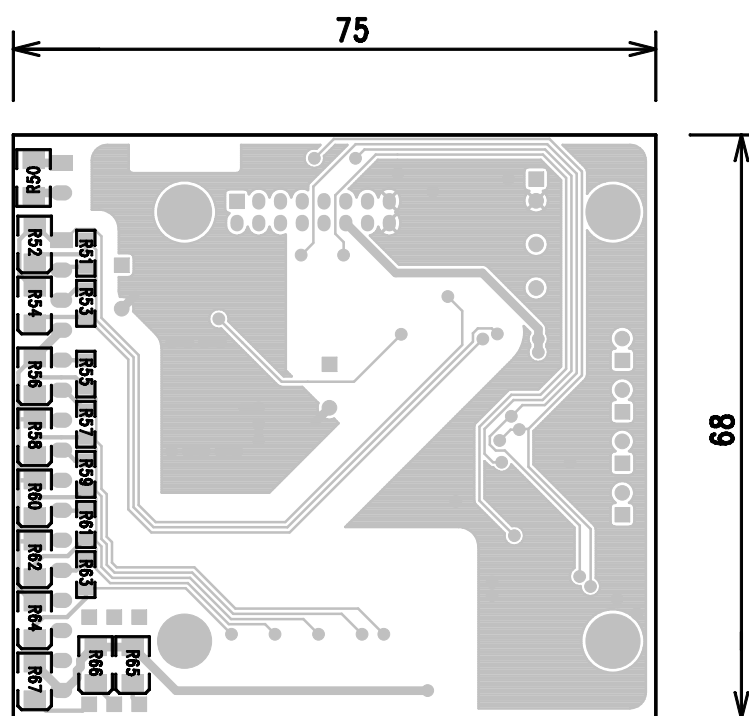
Deska plošného spoje procesorové desky VISx1 (TOP)



Deska plošného spoje procesorové desky VISx1 (BOT)



Osazovací výkres procesorové desky VISx1 (TOP)



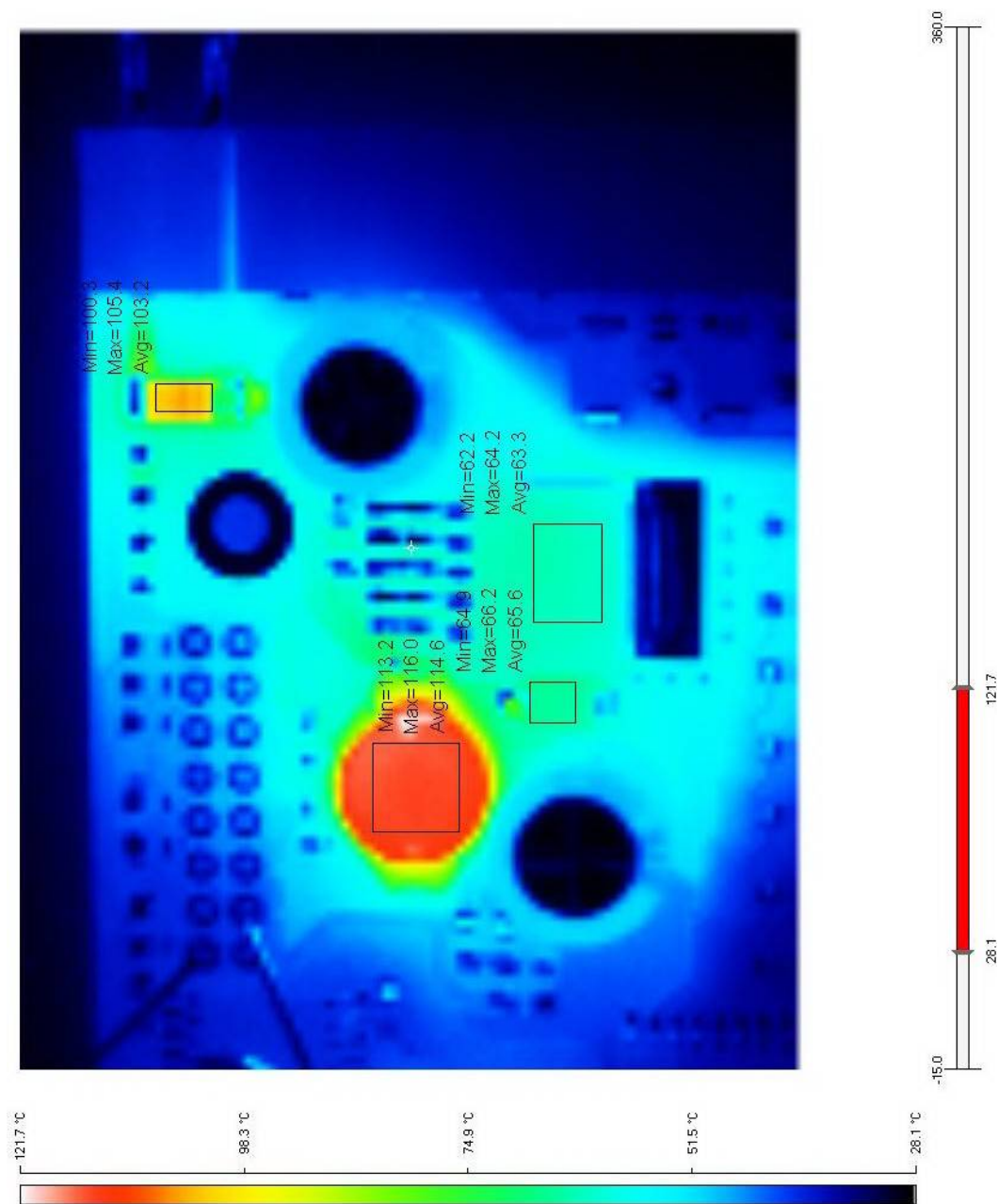
Osazovací výkres procesorové desky VISx1 (BOT)

ROZPISKA SOUČÁSTEK				Číslo dokumentu:	
Název jednotky:		VIS-IOx1	Název DPS:	VIS-IOx1	
			Č.v.osazení:		
		CELKOVÁ	Datum vzniku:	8.12.2007	
	JEDNOTKA:	VISx1.01			C.dok.
Pol.	Reference	Hodnota	Kód	Počet	Osaz.
1	*DPS	VIS-IOx1	VIS-IOx1	1	
2	C1 C4 C14	10uF/10V	CT 10uF/10V SMD-B	3	
3	C2 C5 C7 C8 C10 C12 C13	100nF	C0805 100nF X7R 63V	7	
4	C3	3,3nF COG	C1206 3,3nF COG 100V	1	
5	C6	100uF/50V	CE 100uF/50VIM-H	1	
6	C9	100nF	C1206 100nF X7R 63V	1	
7	C11	1000uF/10V	CE_1000uF/10VIM	1	
8	D1	LED X	LED 0805/20mA X	1	
9	D2	1N4007	1N4007 MELF	1	
10	D3 D5 D6 D7	LED_G	LED_H16	4	
11	D4	10MQ100N	10MQ100N SMA	1	
12	J1	ARK1550/2	KON_ARK1550/2	1	
13	J2	S2G16	S2G16	1	
14	J3	ARK1550/16	KON_ARK1550/16	1	
15	L1	220uH	L 220uH/0.9A SMD 12,7x10mm	1	
16	O1	ADuM1301	ADuM1301ARW SO16/400	1	
17	O2 O3 O4 O5 O6 O7 O8 O9 O10 O11 O12	TLP127	OSMD_TLP127	11	
18	R1 R18 R29 R30 R41 R43 R45	10K	R0805 10K 5%	7	
19	R2	150R	R0805 150R 5%	1	
20	R3 R4 R5 R6 R7 R8 R9	470R	R0805 470R 5%	7	
21	R10 R12 R17 R27	56R	R0805 56R 5%	4	
22	R11	11k	R0805 11k 5%	1	
23	R13	3k6	R0805 3k6 5%	1	
24	R14 R16 R28 R56 R58 R60 R62 R64	SMBJ33A	SMBJ33A	8	
25	R15 R19 R32 R51 R53 R55 R57 R59 R61 R63	10R	R1206 10R 5%	10	
26	R20 R21 R22 R23 R24 R25 R26 R31	160R	R0805 160R 5%	8	
27	R33 R34 R35 R36 R37 R38 R39 R40	10R	R0805 10R 5%	8	
28	R42 R44 R46	330R	R0805 330R 5%	3	
29	R47 R48 R49	1k5	RR 1k5/0.6W	3	
30	R50	SMBJ36CA	SMBJ36CA	1	
31	R52 R54	SMBJ7A	SMBJ7A	2	
32	R65 R66 R67	SMBJ8A	SMBJ8A	3	
33	U1	CHS10505	U_CHS10505	1	
34	U2	LM2526	LM2526	1	
35	U3	ADM485	65176B SO8	1	
36	U4 U5	74LV595A_SO16	USMD_74LV595A_SO16	2	
37	U6	74LV165A_SO16	USMD_74LV165A_SO16	1	
38	Z1 Z2 Z3	2.7V	BZV55C-2,7V SOD80	3	

Kusovník desky VIS-IOx1

14 PŘÍLOHA C

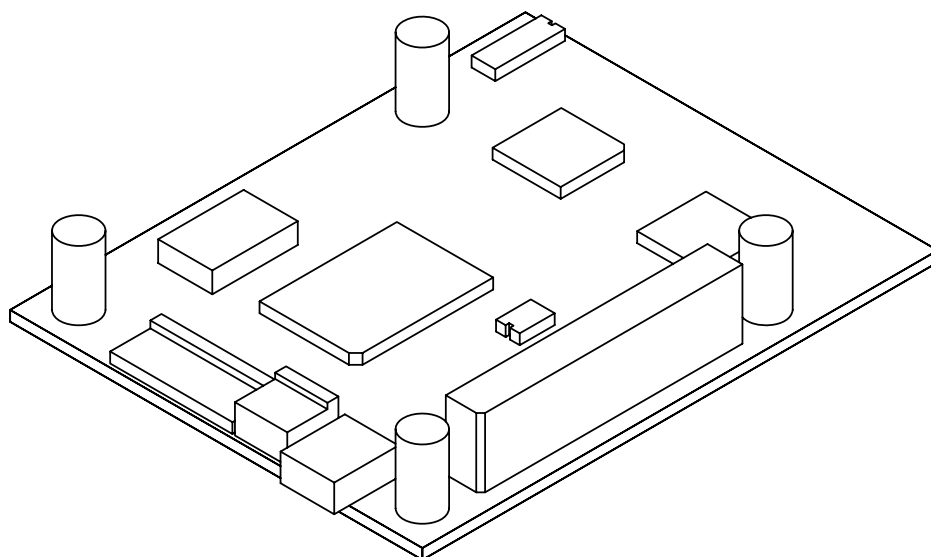
TERMO SNÍMEK SPÍNANÉHO NAPÁJECÍHO ZDROJE DESKY VIS-IOX1



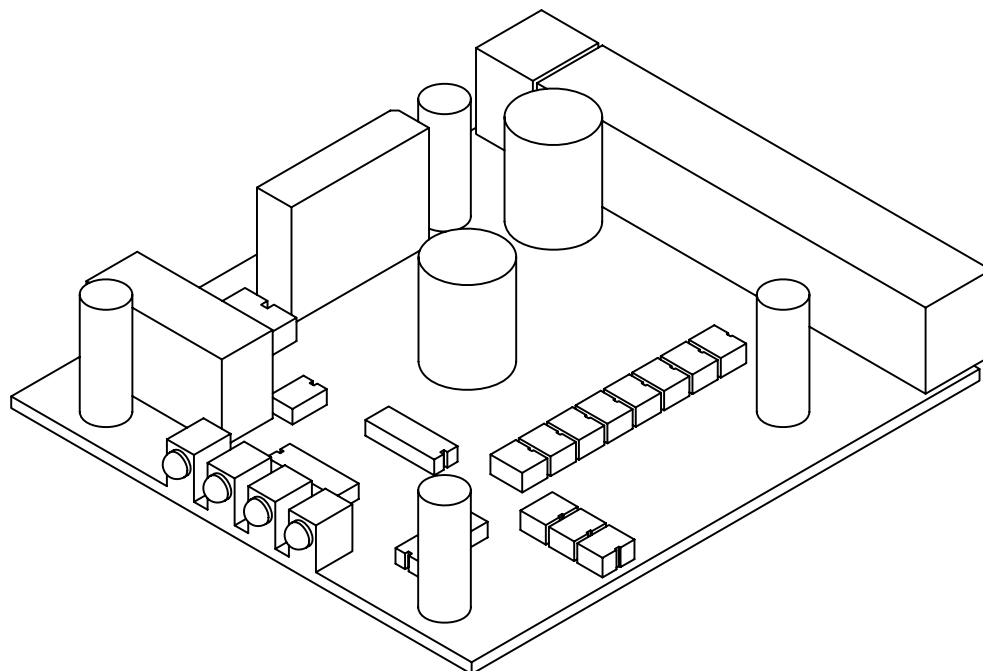
Termo snímek DPS VIS-IOx1

15 PŘÍLOHA D

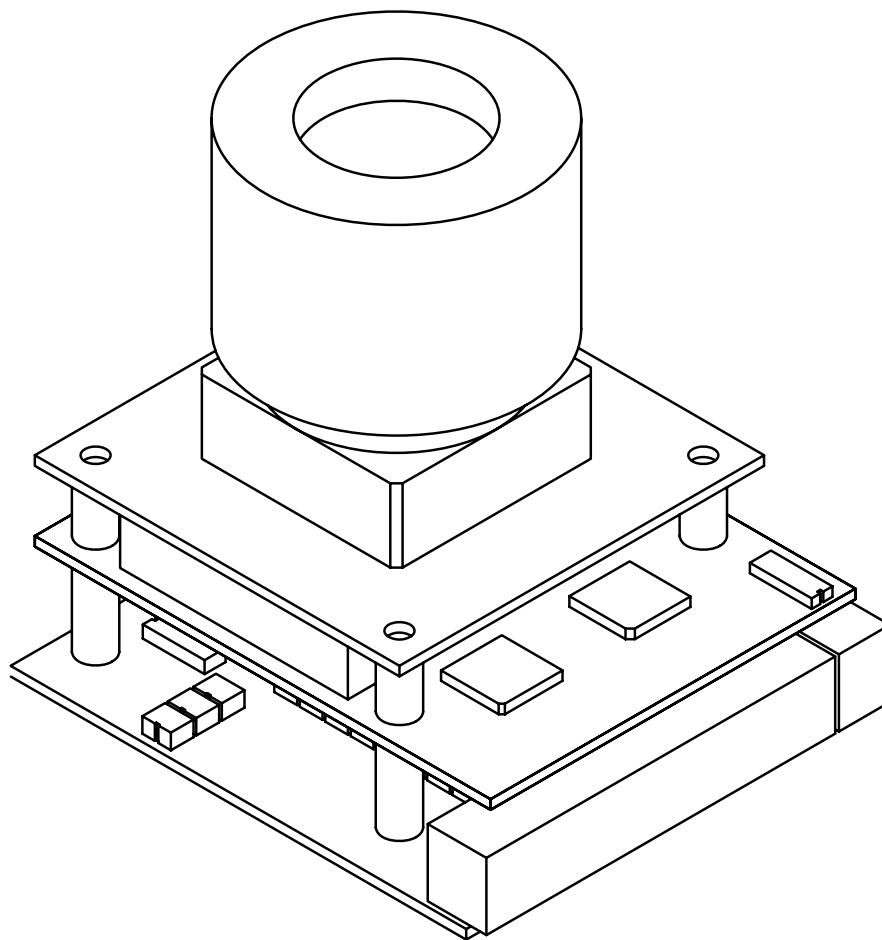
MECHANICKÝ MODEL VIDEOSENZORU



Mechanický model videosenzoru – VISx1



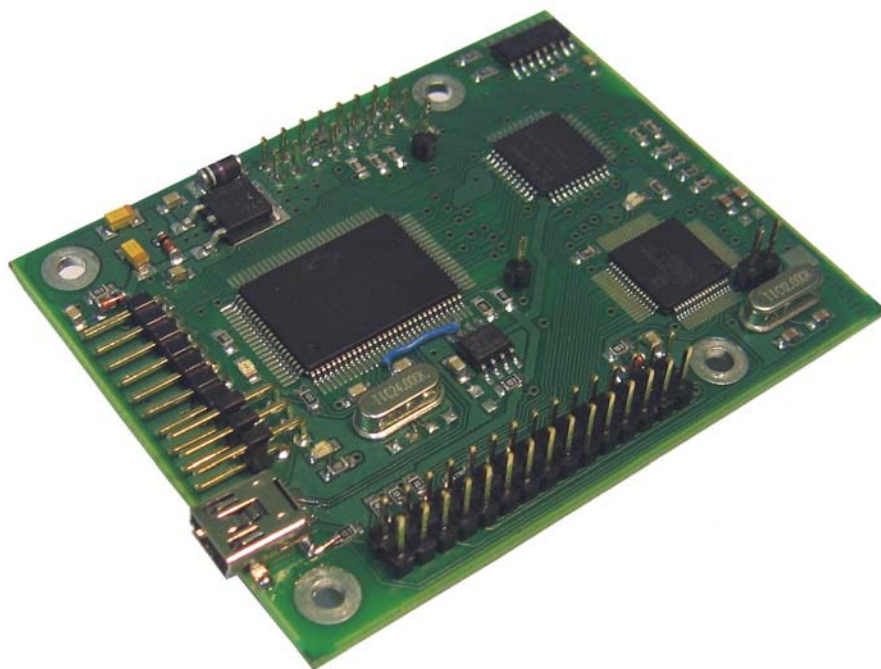
Mechanický model videosenzoru – VIS-IOx1



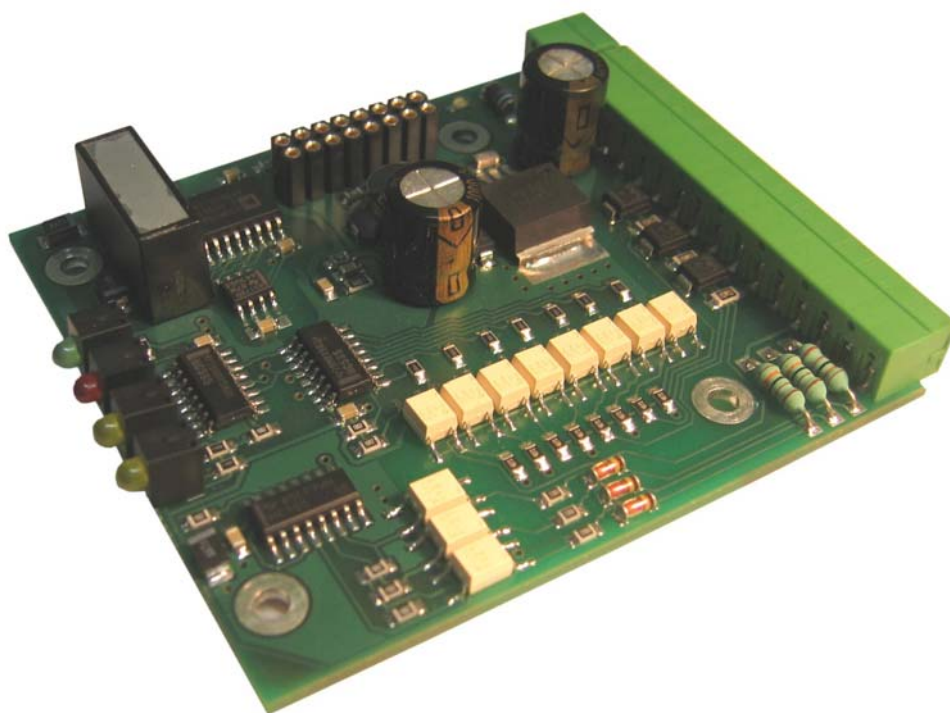
Mechanický model videosenzoru – kompletní sestava

16 PŘÍLOHA E

FOTODOKUMENTACE



Fotodokumentace - procesorová deska VISx1



Fotodokumentace - deska vstupů - výstupů VIS-IOx1



Fotodokumentace - videosenzor – kompletní sestava



Fotodokumentace - videosenzor v krabičce

17 PŘÍLOHA F

OBSAH PŘILOŽENÉHO CD

- Diplomová práce v pdf formátu.
- Zdrojové kódy - firmware procesoru.
- VHDL kódy - firmware pro CPLD.
- Zdrojové kódy - firmware USB řadiče.
- Zdrojové kódy řídicí aplikace na nadřazeném PC.
- Elektronickou verzi schémat, navržených DPS, rozpisek.
- Dokumentace k procesoru LPC2148.
- Základní nástroje pro vývoj, ukázkové zdrojové kódy a ovladače pro práci s mikroprocesorem a USB řadičem.