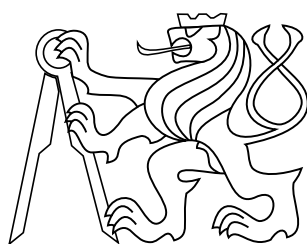


bakalářská práce

# **System sběru dat ze senzorů CCD s STM32Fx**

*Adam Berlinger*



Květen 2014

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

České vysoké učení technické v Praze  
Fakulta elektrotechnická, Katedra měření



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Adam Berlinger**

Studijní program: **Otevřená informatika**  
Obor: **Počítačové systémy**

Název tématu česky: **Systém sběru dat ze senzorů CCD s STM32Fx**

Název tématu anglicky: **STM32Fx Based DAQ System for CCD Sensors**

### Pokyny pro vypracování:

Navrhněte a realizujte jednotku řízení a sběru dat z optoelektronických senzorů CCD. Pro řízení optoelektronického řádkového senzoru CCD, digitalizaci jeho signálu a přenos do PC využijte mikrořadiče řady STM32Fx. Provéřte možnost využít pro řízení senzoru, záznam a zobrazení jeho signálu též modul STM32F429Disco. Vytvořte programové vybavení pro mikrořadič STM32Fx i aplikaci pro nadřazené PC. Aplikace má zajistit zobrazení signálu senzoru provozovaného v řádkovém režimu (1D) i v režimu typu scanner (2D).

### Seznam odborné literatury:

- [1] RM0090 reference Manual, STMicroelectronics, 2013; [www.st.com](http://www.st.com)
- [2] Zoubek, M.: Bakalářská práce, ČVUT-FEL, Praha 2009
- [3] Yiu J.: The definitive Guide to the ARM Cortex-M3. Elsevier 2007, ISBN: 978-0-7506-8534-4

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

Datum zadání bakalářské práce: 6. ledna 2014

Platnost zadání do<sup>1</sup>: 30. ledna 2015

Prof. Ing. Vladimír Haasz, CSc.  
vedoucí katedry



Prof. Ing. Pavel Ripka, CSc.  
děkan

<sup>1</sup> Platnost zadání je omezena na dobu dvou následujících semestrů.





## **Poděkování**

Tímto bych chtěl srdečně poděkovat doc. Ing. Janu Fischerovi, CSc. za všechny čas, který mi věnoval během konzultací, a odborné rady.

## **Prohlášení**

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

V praze dne .....

.....

## **Abstrakt**

Tato práce se zabývá využitím mikrořadičů řady STM32F3 a STM32F4 s jádrem ARM Cortex™ M4 pro řízení řádkového CCD senzoru, zpracování dat a komunikaci s počítačem přes sběrnici USB. Pro přenos data je využita aplikace Virtual COM port. Pro zapojení byly použity vývojové moduly STM32F3-Discovery a STM32F4-Disco. Modul STM32F429-Disco (označován také jako STM32F4-IDiscovery) obsahuje externí paměť SDRAM a displej LCD. To nám umožní rychle ukládat data jednotlivých řádků, poskládat je do obrázku a zobrazit na displeji. Výsledkem práce je firmware pro oba kity a jednoduchý zobrazovací software.

## **Klíčová slova**

senzor CCD; USB; Virtual COM port; ARM Cortex™ M4; STM32F3xx; STM32F4xx; ILX551a; Systém sběru dat

## **Abstract**

This work describes use of microcontrollers STM32F3 and STM32F4 series, containing ARM Cortex™ M4 core, for controlling CCD line sensor, processing readed data and sending them over USB interface. Virtual COM port application is used for data transfer. STM32F3-Discovery and STM32F429-Disco (also known as STM32F4-IDiscovery) development kits were used for creating functioning prototype. STM32F4-Disco kit contains external SDRAM and LCD display which enables to quickly load mutliplr line data, compose them into a picture and show the picture on LCD display. The result of this work is firmware for both kits and simple software for viewing measured data.

## **Keywords**

CCD sensor; USB; Virtual COM port; ARM Cortex™ M4; STM32F3xx; STM32F4xx; ILX551a; Data aquisition system

# Obsah

|  |           |
|--|-----------|
| <b>1. Úvod</b>   | <b>1</b>  |
| <b>2. Rozbor problematiky</b>  | <b>2</b>  |
| 2.1. Co je to senzor CCD   | 2         |
| 2.2. Parametry senzoru ILX551a                                       | 2         |
| 2.3. Parametry mikrořadiče STM32F3                                   | 3         |
| 2.4. Návrh řádkové kamery  | 4         |
| 2.5. Návrh PC programu   | 4         |
| 2.6. Využití modulu STM32F429Disco                                   | 4         |
| <b>3. Realizace řádkové kamery s řadičem STM32F3</b>                 | <b>5</b>  |
| 3.1. Možnosti řízení senzoru CCD pomocí řadiče STM32F3               | 5         |
| 3.1.1. Nastavení doby expozice                                       | 5         |
| 3.1.2. Generování řídicích signálů programově                        | 6         |
| 3.1.3. Generování řídicích signálů pomocí čítačů                     | 6         |
| 3.2. Zpracování výstupního signálu senzoru CCD pomocí řadiče STM32F3 | 9         |
| 3.2.1. Zpracování signálu v režimu bez Sample-Hold                   | 9         |
| 3.2.2. Zpracování signálu v režimu se Sample-Hold                    | 10        |
| 3.2.3. Zpracování signálu programem řadiče                           | 10        |
| 3.2.4. Detekce saturace CCD senzoru                                  | 10        |
| 3.3. Přenos dat po sběrnici USB                                      | 11        |
| 3.3.1. Popis aplikace virtual COM port                               | 11        |
| 3.3.2. Komunikace mezi zařízením a počítačem                         | 11        |
| 3.3.3. Ukončení přenosu v aplikaci Virtual COM port                  | 12        |
| 3.3.4. Realizace v programu mikrořadiče                              | 12        |
| 3.3.5. Měření rychlosti přenosu USB                                  | 12        |
| 3.4. Obvodové řešení řádkové kamery s řadičem STM32Fx                | 12        |
| 3.4.1. Rozvod napájení   | 13        |
| 3.4.2. Propojení řadiče STM32F3 se senzorem CCD                      | 13        |
| 3.4.3. Schéma zapojení   | 13        |
| 3.4.4. Umístění objektivu  | 15        |
| 3.5. Popis řídicího programu mikrořadiče STM32F3                     | 16        |
| 3.5.1. Nastavení periférií   | 16        |
| 3.5.2. Zpracování přerušení  | 17        |
| 3.5.3. Zpracování a odesílání dat přes USB                           | 17        |
| <b>4. Realizace řádkové kamery s modulem STM32F429Disco</b>          | <b>19</b> |
| 4.1. Řízení senzoru CCD pomocí řadiče STM32F4                        | 19        |
| 4.1.1. Generování řídicích pulsů s využitím režimu “center-aligned”  | 19        |
| 4.2. Zpracování signálu senzoru CCD pomocí mikrořadiče STM32F4       | 20        |
| 4.2.1. Zpracování v řádkovém režimu                                  | 20        |
| 4.2.2. Zpracování v režimu skener                                    | 20        |
| 4.2.3. Automatické spuštění skenování                                | 21        |
| 4.3. Řízení displeje LCD pomocí řadiče STM32F4                       | 22        |
| 4.3.1. Rasterizace signálu senzoru CCD                               | 22        |
| 4.3.2. Ovládání displeje LCD v režimu skener                         | 23        |



|           |  |           |
|-----------|--|-----------|
| 4.4.      | Komunikace po sběrnici USB s modulem STM32F429Disco . . . . .                        | 23        |
| 4.4.1.    | Přenos naskenovaného obrazu do PC . . . . .  | 24        |
| 4.4.2.    | Měření rychlosti přenosu USB . . . . .   | 24        |
| 4.5.      | Ovládání aplikace . . . . .  | 24        |
| 4.6.      | Popis řídicího programu modulu STM32F429Disco . . . . .                              | 24        |
| 4.6.1.    | Připojení externí paměti SDRAM . . . . .   | 24        |
| 4.6.2.    | Čtení stisku tlačítka . . . . .  | 24        |
| 4.7.      | Porovnání procesoru STM32F3 a STM32F4 . . . . .                                      | 26        |
| 4.8.      | Využití modulu STM32F407Discovery . . . . .  | 26        |
| 4.9.      | Obvodové řešení řádkové kamery s modulem STM32F429Disco na kontaktním poli . . . . . | 27        |
| 4.10.     | Kompaktní obvodové řešení řádkové kamery s modulem STM32F429Disco . . . . .          | 27        |
| <b>5.</b> | <b>Realizace zobrazovacího programu na PC</b>  | <b>29</b> |
| 5.1.      | Popis uživatelského rozhraní . . . . .   | 29        |
| 5.2.      | Připojení PC k zařízení . . . . .  | 29        |
| 5.3.      | Příjem dat přes USB . . . . .  | 29        |
| 5.4.      | Zobrazení dat v řádkovém režimu . . . . .  | 30        |
| 5.5.      | Zobrazení dat v režimu skener . . . . .  | 30        |
| <b>6.</b> | <b>Závěr</b>   | <b>33</b> |
|           | <b>Přílohy</b>   |           |
| <b>A.</b> | <b>Přiložené soubory na disku CD</b>   | <b>34</b> |
|           | <b>Literatura</b>  | <b>35</b> |

## Zkratky

Zde se nacházejí použité zkratky a jejich stručný význam/překlad

|       |  |
|-------|--|
| ARM   | Advanced RISC Machines — architektura procesorů/mikrořadičů a zároveň jméno firmy, která ji vyvíjí                   |
| CCD   | Charge-coupled device — zařízení pro přesouvání el. náboje, často používané v obrazových snímačích                   |
| GPIO  | General purpose input/output — vstupní/výstupní piny procesoru, jejichž logickou úroveň lze přímo programově ovládat |
| NVIC  | Nested-vector Interrupt Controller — řadič přerušení architektury ARM  |
| OC    | Output compare — jednotka čítače která porovnává aktuální hodnotu čítače s předem stanovenou konstantou              |
| PC    | Personal computer — osobní počítač   |
| PWM   | Pulse width modulation — pulzně šířková modulace   |
| RAM   | Random-access memory — operační paměť  |
| SDRAM | Synchronized dynamic random-access memory — druh operační paměti   |
| SIMD  | Single instruction multiple data — instrukce procesoru, které jsou schopné zpracovat více dat pomocí jedné instrukce |
| USB   | Universal serial bus — sériová sběrnice pro přenos dat   |

# 1. Úvod

Zařízení s řádkovými senzory CCD nejdou využít především v průmyslových aplikacích. Umožňují snímat velké množství snímků za sekundu a najdou uplatnění při monitorování výrobních linek, kde se výrobek pohybuje velikou rychlostí a je potřeba rychle zachytit jeho obraz a zkontrolovat, zda neobsahuje defekt. Řádkový senzor se také používá v cílové kamere, která zachytí dobíhající závodníky.

Další aplikací řádkového CCD senzoru jsou různé měřicí zařízení, která například zkoumají šířku objektu, nebo jeho vzdálenost od přístroje. Laserem se vyše paprsek světla a CCD senzor zachytí odkud se paprsek odrazil. Z toho se pak spočítá výsledná vzdálenost.

Dříve se podobná zařízení skládala, z několika součástí, kde každá součástka plnila jednu specifickou roli (viz. řešení Martina Zoubka [1]). Používaly se samostatné bloky pro generování řídicích pulsů, další pro převod dat z analogového signálu na číselný a jiné pro přenos dat do počítače nebo jiného zařízení.

V dnešní době, už se vyrábějí mikrořadiče, které obsahují celou řadu periférií a jejich využití může být velmi široké. Mikrořadiče podporují celou řadu komunikačních rozhraní (SPI, I<sup>2</sup>C, USART, USB) a také obsahují A/D převodníky, které umožňují zpracování analogového signálu. Také obsahují několik čítačů, které mohou být použity pro generování řídicích signálů.

Často nevyužijeme většinu možností a periférií, které nám mikrořadič nabízí a mohlo by se zdát, že zbytečně plýtváme hardwarovými prostředky. Na druhou stranu cena těchto mikrořadičů je v dnešní době poměrně nízká a jejich použití je často výhodnější, než skládat zařízení z jednotlivých komponent.

Pro realizaci je dobré zvolit takový mikroprocesor, který používá rozšířenou a dobře zdokumentovanou architekturu jádra, pro kterou jsou dostupné kvalitní vývojové nástroje. Nabízí se využití architektury ARM, která je v dnešní době hojně používaná v mobilních zařízeních, ale zároveň také nachází uplatnění ve vestavěných systémech. Architekturu ARM spravuje stejnojmenná firma, která také poskytuje vývojové nástroje. Téměř všichni velcí výrobci mikroprocesorů vyrábějí mikroprocesory s architekturou ARM, takže nejsme vázáni na jednoho výrobce a můžeme snáze naše aplikace přenášet na jiné mikroprocesory. To samozřejmě neplatí pro kód obsluhující periferie, které se liší mezi jednotlivými výrobci i mezi jednotlivými produkty daného výrobce.

Cílem této práce je navrhnout řádkovou kameru se senzorem CCD, která bude posílat zpracovaná data po sběrnici USB do počítače. Pro realizaci budeme chtít co nejvíce využít možnosti mikrořadiče STM32Fx od firmy STMicroelectronics. Na straně PC pak bude potřeba vyvinout aplikaci schopnou posílaná data zobrazit.

## 2. Rozbor problematiky

Pro realizaci řádkové kamery nejdříve musíme zjistit parametry senzoru CCD a mikrořadiče STM32Fx. Musíme zajistit vzájemnou kompatibilitu analogových a digitálních signálů. Případně použít takové součástky a zapojení, které tuto kompatibilitu zařídí. Dále musíme jednotlivým součástkám zajistit stabilní zdroj napětí.

### 2.1. Co je to senzor CCD

Řádkový senzor CCD (Charge-coupled device) je optický senzor, který je schopen zachytit dopadající světlo na úzký citlivý pruh a přeměnit ho na elektrický náboj.

Senzor CCD se skládá z několika částí [2]. První částí jsou jednotlivé optoelektronické prvky (fotodiody), které přeměňují dopadající světlo na náboj.

Dalším prvkem je posuvný registr CCD, který je schopný mezi svými buňkami přenášet náboj. Tato část nám umožní na jediném výstupním signálu číst velikost nábojů z jednotlivých fotodiód. Od toho se tyto senzory nazývají senzory CCD.

Mezi fotodiódami a posuvným registrem se ještě nachází brána, která zajišťuje přenos náboje z fotodiód do posuvného registru. Po uplynutí předem stanovené doby se na fotoelementech objeví dostatečný náboj a ten je pak během krátkého okamžiku převeden do posuvného registru. Z fotoelementů tak zmizí náboj reprezentující intenzitu světla a jsou tak připraveny přečíst další snímek. Časovému úseku mezi dvěma snímky říkáme doba expozice. Čím je doba expozice větší, tím světlejší bude výsledný obraz.

Pokud intenzita, dopadající na jeden fotoelement, přesáhne maximum náboje, jenž je schopen tento prvek udržet, dojde k přenosu náboje na ostatní fotoelementy. Fotoelement, tak již není schopen rozeznat vyšší intenzitu světla. Takovému stavu říkáme saturace. V některých aplikacích je dobré tento stav rozeznat, neboť výsledný videosignál pak neodpovídá realitě.

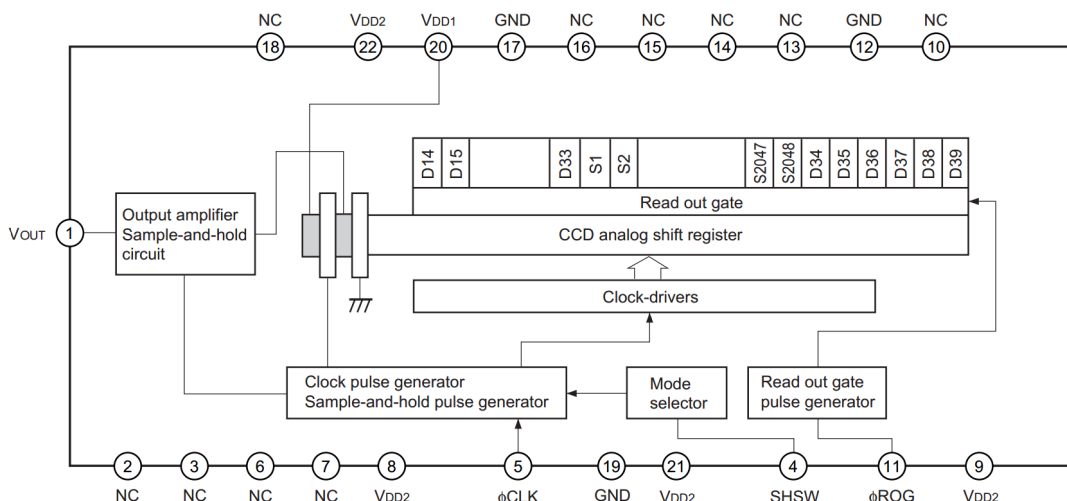
Na výstupu senzoru CCD se ještě nachází nábojový zesilovač, který upravuje výstupní signál.

### 2.2. Parametry senzoru ILX551a

Jako CCD senzor pro můj prototyp jsem zvolil Sony ILX551a. ILX551a je lineární černobílý řádkový CCD senzor, který obsahuje 2048 aktivních pixelů a 39 referenčních černých pixelů.

Senzor může pracovat ve dvou režimech:

- Normální režim — pro každý pixel se na výstupu se sestupnou hranou hodinového signálu objeví krátký resetovací pulz, poté referenční hodnota pixelu. S náběžnou hranou se objeví na výstupu hodnota pixelu.
- Režim “sample-hold” — s náběžnou hranou hodinového signálu se na výstupu objeví hodnota pixelu. Výstupní signál je oproti normálnímu režimu zpožděný o jeden pixel. Tento režim je méně přesný než normální režim a hodí se na aplikace, kde jsou omezené možnosti zpracování výstupního signálu.



Obr. 2.2.1. Vnitřní blokové schéma senzoru ILX551a [3]

Senzor potřebuje dvě napájecí napětí 9 V a 5 V. Na vstupu má tři digitální řídicí signály: CLK (hodinový signál), ROG (překlopení dat do posuvného registru) a SHSW (výběr režimu výstupního signálu). Senzor má jeden výstupní analogový signál.

## 2.3. Parametry mikrořadiče STM32F3

Pro řízení senzoru CCD musíme navrhnout takové obvodové řešení, které bude schopné generovat řídicí signály senzoru CCD, bude schopné výstupní signály zpracovat a uložit do paměti a nakonec tato data přenést dál do počítače ke zpracování.

V současné době se nabízí možnost využít mikrořadič STM32F303, který obsahuje většinu potřebných periférií pro řízení a zpracování. Řadič STM32F3 obsahuje GPIO (General-purpose input/output) piny, které je možné programově řídit. Dále obsahuje několik čítačů, které je možné použít ke generování PWM (pulse-width modulation) signálů. Řadič také obsahuje 4 12-bitové A/D převodníky, které je možné napojit na řadič DMA (direct-memory access), který je schopen ukládat snímky rychle do paměti. Nakonec čítač obsahuje periférie pro komunikaci s ostatními zařízeními. Čítač obsahuje full-speed USB, USART (universal synchronous asynchronous receiver transmitter), CAN (Controller area network) a I<sup>2</sup>C (Inter-integrated circuit interface). Tím se obvodové řešení řádkové kamery značně zjednoduší a většinu práce nám obstará mikrořadič a jeho periférie.

Mikrořadič je typicky napájen napětním  $U_{CC} = 3,3\text{ V}$  a na vstupech i výstupech používá 3,3 V CMOS logiku. Vyjímkou jsou analogové periférie (např. A/D a D/A převodníky). Maximální vstupní napětí většiny pinů je  $U_{MAX} = U_{CC} + 0,3\text{ V}$ . Mikrořadič také obsahuje piny, které jsou 5 V, případně 3,3 V tolerantní. U těchto pinů pak není maximální vstupní napětí závislé na napájecím napětí.

Mikrořadič může běžet na frekvenci až 72 MHz. To nám zajistí dostatečný výkon pro zpracování video signálu a obsluhu periférií.

## 2.4. Návrh řádkové kamery

Dále musíme senzoru CCD zajistit vhodné optické a světelné podmínky. K zařízení musíme být schopni připevnit objektiv. Musíme zařízení zkonstruovat tak, aby byl senzor ve správné vzdálenosti od objektivu a aby na senzor nedopadalo okolní světlo mimo objektiv.

Celé zařízení je třeba napájet ze stabilního zdroje napětí. Zdroj napětí 5 V nám může poskytnout sběrnice USB. Senzor ILX551a však potřebuje vedle napájení 5 V i napájení 9 V pro analogovou část. Použijeme proto externí napájecí zdroj 12 V a napěťové regulátory.

## 2.5. Návrh PC programu

Data z řádkové kamery chceme dále nějak zpracovat, případně je někam uložit. K tomuto účelu využijeme sběrnici USB a zařízení připojíme k počítači. Přes sběrnici USB tak budeme schopni data ze senzoru číst a dále zpracovávat. Abychom byli schopni data z řádkové kamery číst, musíme nejdříve navrhnout PC aplikaci, která bude se zařízením komunikovat.

## 2.6. Využití modulu STM32F429Disco

Zajímavé řešení se nabízí při použití modulu STM32F429Disco, který kromě procesoru STM32F4 obsahuje také externí paměť SDRAM a dotykový LCD displej.

Paměť SDRAM umožňuje rychle uložit několik po sobě jdoucích snímků přímo do paměti kamery. Z těchto snímků pak můžeme složit dvojrozměrnou fotografii. Řádková kamera tak může fungovat jako jednoduchý skener. LCD displej můžeme využít k zobrazování zpracovaného videosignálu. Na displeji také můžeme zobrazit námi naskenovaný obrázek. Řádková kamera tak může fungovat jako samostatné zařízení, bez nutnosti ji připojit k počítači.

## 3. Realizace řádkové kamery s řadičem STM32F3

Pro návrh prototypu řádkové kamery jsem použil vývojový modul STM32F3Discovery. Modul STM32F3Discovery obsahuje potřebné součástky pro správný chod procesoru a konektor USB, pomocí kterého můžeme ladit aplikace za běhu. Modul dále obsahuje další konektor USB, který můžeme použít pro naši konkrétní aplikaci.

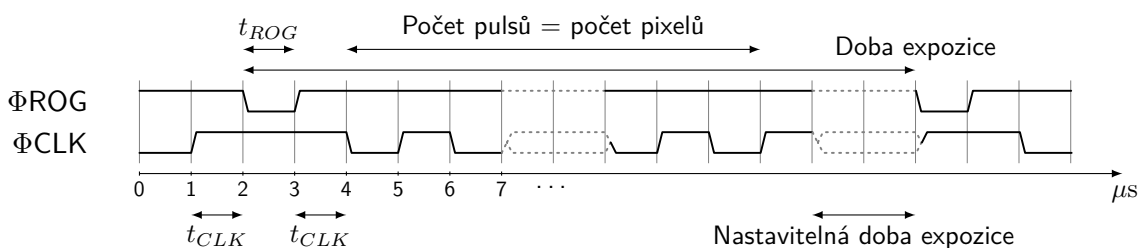
### 3.1. Možnosti řízení senzoru CCD pomocí řadiče STM32F3

Senzor CCD ILX551a pro svou funkci potřebuje tři digitální řídicí signály. Hodinový signál CLK, signál ROG, a signál SHSW.

Signál SHSW zapíná/vypíná režim sample-hold. Mění se jen zřídka a většinou je v předem nastavené úrovni. Režim “sample-hold” můžeme zapnout/vypnout změnou logické úrovně signálu. Logické je, že režim budeme měnit pouze až po skončení čtení videosignálu.

Hodinový signál CLK slouží k vyčítání video signálu jednotlivých pixelů ze senzoru a je vhodné, aby měl stálou frekvenci a střihu. Maximální frekvence signálu CLK je  $f_{MAX} = 5 \text{ MHz}$ .

Signál ROG slouží k převodu el. náboje mezi fotocitlivými prvky a posuvným registrem CCD. Při sestupné hraně signálu se náboj (data) přesune z fotocitlivých prvků do registru CCD. Signál ROG musí mít podle katalogu minimální délku  $t_{ROG} = 500 \text{ ns}$ , přičemž doporučená hodnota je  $t_{ROG} = 1 \mu\text{s}$ . Také je potřeba, aby během nízké úrovně signálu ROG byl hodinový signál v logické úrovni 1 a musí tak být s odstupem od setupné a náběžné hrany. Minimální odstup signálu CLK podle katalogu je  $t_{CLK} = 500 \text{ ns}$ , přičemž opět doporučená hodnota je  $t_{CLK} = 1 \mu\text{s}$ .



Obr. 3.1.1. Časový diagram řídicích signálů pro senzor ILX551a

#### 3.1.1. Nastavení doby expozice

Zařízení by mělo být schopno pracovat za různých světelných podmínek. Proto chceme měnit dobu expozice senzoru CCD. Čím temnější prostředí, tím delší dobu expozice musíme zvolit. Doba expozice je určena periodou ROG signálu. Pro zjednodušení programu budeme dobu expozice nastavovat pouze počtem hodinových pulsů navíc. Nevýhodou delší doby expozice je delší prodleva mezi jednotlivými snímky.

### 3. Realizace řádkové kamery s řadičem STM3F3

Počet hodinových pulsů mezi dvěma signály ROG musí být minimálně počet všech pixelů na senzoru (jinak bychom nemohli přečíst výstupní video signál senzoru). Maximální počet není omezen a umožňuje nám tak nastavit delší dobu expozice, přičemž nejkratší doba ( $t_{min}$ ) je dána frekvencí hodin ( $f_{CLK} \leq 5 \text{ MHz}$ ), počtem pixelů ( $N_p = 2048 + 39$ ) a délkou a odstupem ROG signálu. Stálým hodinovým signálem se navíc odvádí náboj z posuvného registru CCD a tím se snižuje šum na výstupu.

$$t_{min} = \frac{N_p}{f_{CLK}} - \frac{1}{2f_{CLK}} + t_{ROG} + 2 \times t_{CLK} \geq \frac{2087}{5 \text{ MHz}} - 0,2 \mu\text{s} + 1,5 \mu\text{s} = 418,7 \mu\text{s} \quad (3.1.1)$$

Nejdelší doba expozice je pak určena max. počtem hodinových pulsů ( $N_{MAX}$ ).

$$t_{max} = \frac{N_{MAX}}{f_{CLK}} - \frac{1}{2f_{CLK}} + t_{ROG} + 2 \times t_{CLK} \quad (3.1.2)$$

Dobu expozice bychom také mohli nastavit prodloužením intervalu ROG signálu, přičemž by hodinový signál byl neaktivní. V tom případě by ale došlo k zašumnění posuvného registru.

#### 3.1.2. Generování řídicích signálů programově

Nejjednodušší způsob jak generovat řídicí signály pro CCD senzor, je použití GPIO piny brány mikrořadiče a nastavovat logické úrovně přímo program procesoru. Pro měření času můžeme použít systémový čítač SysTick, který je součástí architektury ARM.

Součástí programu je pak smyčka, která kontroluje zda neuplynul příslušný časový interval a pokud ano, změní logickou úroveň řídicího signálu.

Výstupní piny musí být nastaveny ve výstupním režimu push-pull. Program je pak může řídit nastavení příslušného bitu ve výstupním registru příslušné brány.

Výhodou tohoto přístupu je jednoduchá implementace a snadná přenositelnost na jiný mikrořadič s architekturou ARM. Můžeme tedy snadno oživit senzor CCD a ověřit jeho funkčnost. Nevýhodou je poměrně velké zatížení mikrořadiče (potřeba provést operaci každou  $1 \mu\text{s}$ ). Navíc výsledný signál nemusí být pravidelný z hlediska času, protože během měření periody může přijít přerušení, které chod programu na chvíli zastaví a zpozdí tak výsledný signál. Dále musíme softwarově řídit i čtení z A/D převodníku.

#### 3.1.3. Generování řídicích signálů pomocí čítačů

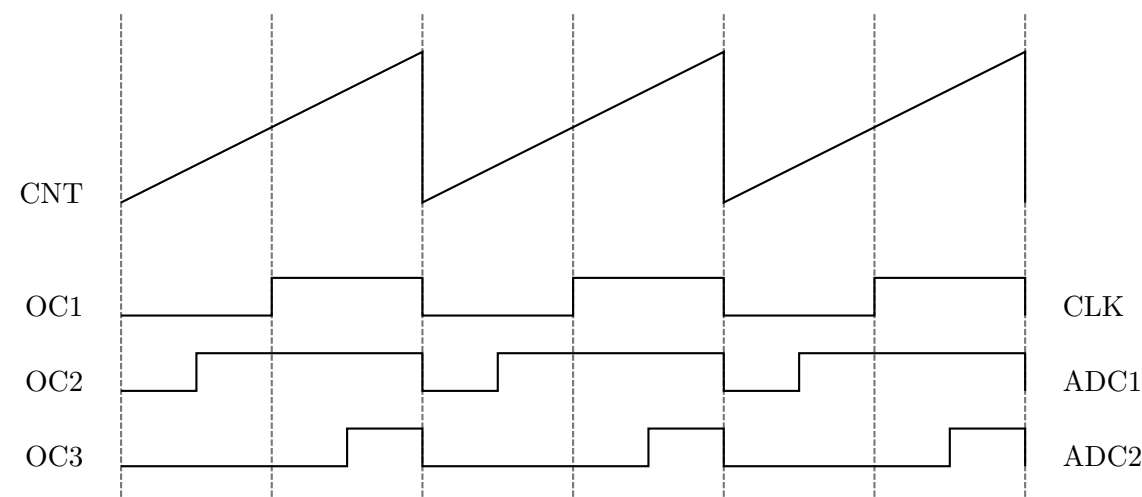
Další možností je použít čítače, které jsou součástí mikrořadiče STM32F3. Mikrořadič STM32F303 obsahuje celkem 10 čítačů, přičemž některé se můžou lišit pokročilými funkcemi, případně počtem OC (output-compare) kanálů.

Každý čítač obsahuje "Auto-reload" registr, který určuje maximální hodnotu čítače. V základním režimu čítač začíná v nule a postupně inkrementuje svou hodnotu (registr CNT), dokud nedosáhne hodnoty "Auto-reload" registru. Poté čítač vygeneruje "update event" a začne opět počítat od nuly. "Update event" může například vyvolat přerušení, případně spustit jiný čítač [4].

Čítač může běžet v režimu "one-pulse", kdy je činnost čítače zastavena, jakmile proběhne "update-event". Čítač pak může být znova spuštěn programově, případně pomocí jiného čítače (režim master-slave).

Každý čítač obsahuje několik OC kanálů. Každý OC kanál čítače má přiřazený registr obsahující hodnotu, která se porovnává s aktuální hodnotou registru čítače. Každý OC kanál má jeden výstup, který může být připojen na výstupní pin procesoru, případně





**Obr. 3.1.2.** Časový diagram chodu čítače *A* a jeho OC kanálů

může být interně připojen k jiným perifériím procesoru. OC kanál také může generovat přerušování procesoru. OC kanál lze nastavit do režimu PWM (pulse-width modulation), při kterém je výstup v aktivní úrovni, je-li hodnota registru OC kanálu menší než hodnota registru čítače.

Některé registry čítače (především “Auto-reload”) a OC kanálů mají ještě tzv. “Pre-load” registry, do kterých se zapíše nová konfigurace příslušného registru a po vyvolání “Update event” se hodnoty načtou do skutečné konfigurace. Toho můžeme využít například při změně doby expozice.

Čítače je možné nastavit do režimu master-slave, kdy náběžná hrana OC kanálu (případně jiná událost) jednoho čítače může zapnout, nebo restartovat čítač druhý, dále může výstup OC kanálu sloužit jako hodinový signál pro jiný čítač, případně může aktivní úroveň OC kanálu jiný čítač pozastavit (dokud není OC kanál v neaktivní úrovni). Režim master-slave podporuje i další nastavení, ale ty nebudeme pro generování řídicích signálů potřebovat.

Pro generování signálů jsem použil tři čítače (*A*, *B* a *C*) v cyklickém zapojení master-slave.

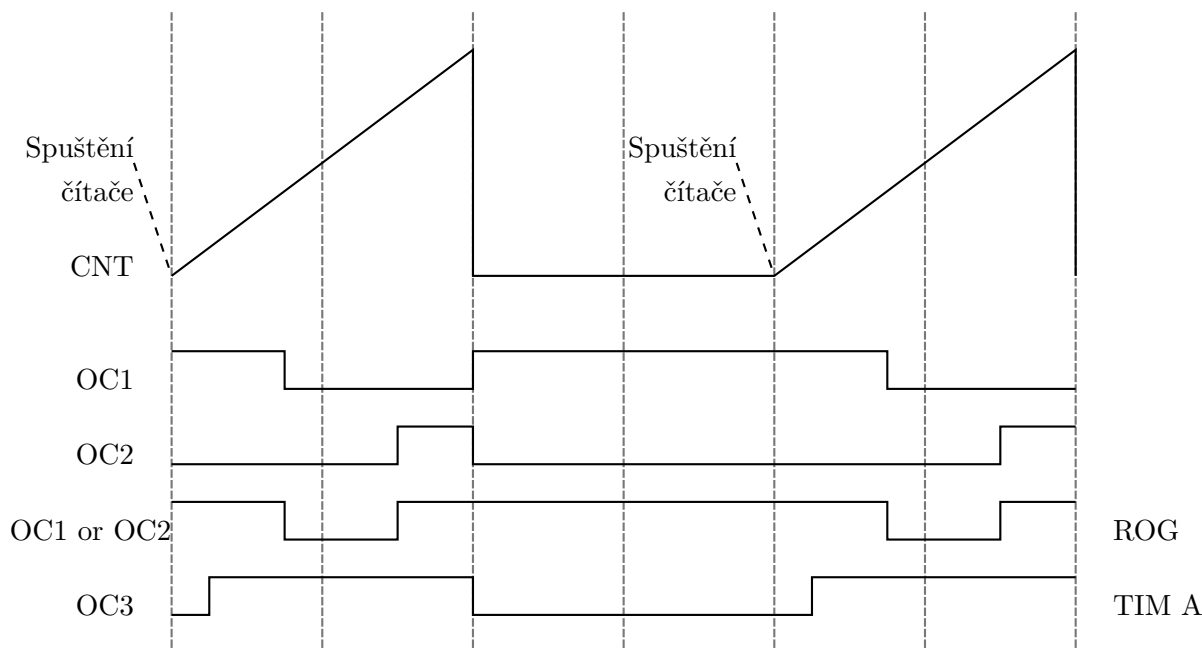
První čítač *A* zajišťuje generování hodinového signálu CLK a spouští A/D převodníky procesoru. K řízení signálu CLK slouží OC (Output-Compare) kanál, který je nastaven do režimu PWM a umožňuje nastavit střídu signálu. Ke spuštění A/D převodníků slouží náběžné hrany kanálů OC2 a OC3. Signál CLK je interně zapojen na vstup hodinového signálu čítače *B*.

Čítač *B* počítá počet hodinových pulzů signálu CLK a po přečtení předem stanoveného počtu (závisí na době expozice) aktivuje čítač *C* a vyvolá přerušování, které nastaví řadič DMA. Čítač používá dva OC kanály: jeden pro spuštění čítače *C* a druhý pro generování přerušování. Oba jsou nastaveny v režimu PWM.

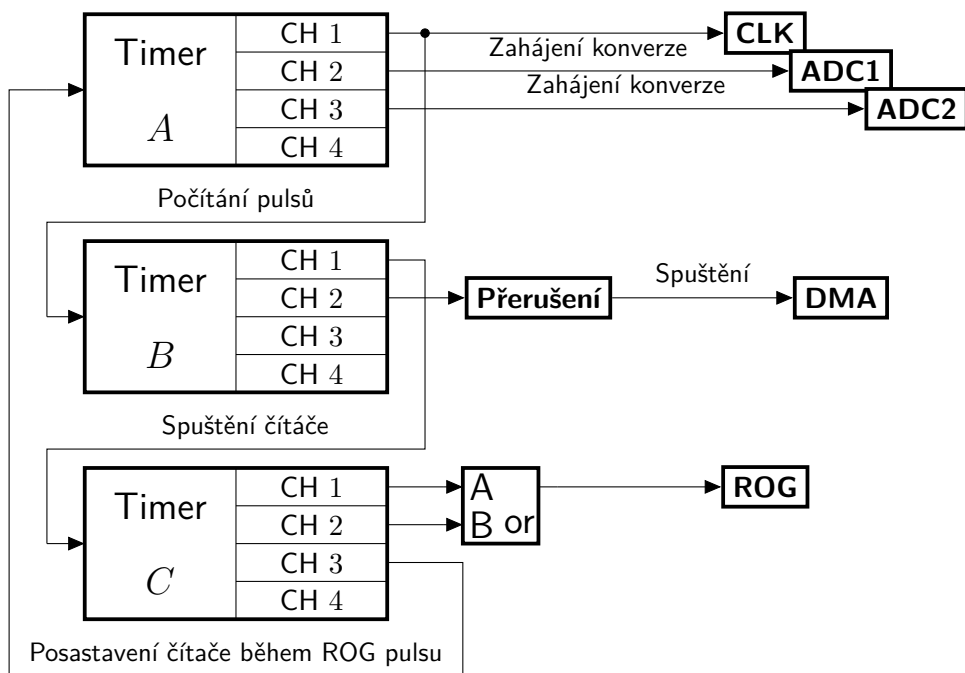
Poslední čítač *C* se stará o průběh signálu ROG. Po spuštění zastaví čítač *A* a zároveň s časovým odstupem generuje signál ROG. Čítač je nastaven v režimu “one-pulse”, tudíž se po přetečení čítacího registru čítač zastaví a je spuštěn až náběžnou hranou OC kanálu čítače *B*. Po zastavení čítače je čítač *A* opět zapnut. Čítač používá tři OC kanály. Dva slouží pro generování signálu ROG a jsou nastaveny v kombinovaném režimu (logický součet). Na výstupu se tak objeví logická 1, právě tehdy pokud jsou v logické 1 oba OC kanály. Třetí OC kanál slouží k pozastavení čítače *A*.

Konkrétní typy a čísla čítačů a OC kanálů závisí na tom, které výstupní piny chceme

### 3. Realizace řádkové kamery s řadičem STM3F3



**Obr. 3.1.3.** Časový diagram chodu čítače *C* a jeho OC kanálů (je-li hladni kanálu OC3 aktivní, pak je čítač *A* pozastaven)



**Obr. 3.1.4.** Zjednodušené schéma vnitřního zapojení čítačů pro generování řídicích signálů

použit, jaké čítače jdou spolu nastavit do režimu master-slave a jakými čítači/kanály lze spouštět spouštět A/D převodníky. Pro jsou aplikaci jsem použil čítače TIM2 (*A*), TIM3 (*B*) a TIM4 (*C*).

Aby se signály OC kanálů projevíly a na výstupních pinech procesoru, je potřeba příslušné piny nastavit jako výstup v režimu push-pull a nastavit alternativní funkci odpovídající příslušnému čítači.

Výhodou řízení pomocí čítačů je téměř nulové zatížení jádra mikrořadiče, časová spo-

lehlost signálu a automatické zahájení konverze A/D převodníku. Nevýhodou tohoto přístupu je poměrně složité nastavení jednotlivých čítačů, nutnost zkontrolovat dostupnost pinů mikrořadiče pro příslušné čítače a jejich kanály. Důležité je také zkontrolovat, kterými OC kanály lze spouštět konverzi A/D převodníku. Z toho vyplývá také špatná přenositelnost i mezi mikrořadiči řady STM32Fx, které mají sice téměř stejné čítače, ale napojení na výstupní piny a napojení na A/D převodníky se liší. Dalším problémem je zpoždění při spouštění/zastavení mezi jednotlivými čítači v nastavení master-slave, které je závislé na vstupních hodinách obou čítačů. Proto je dobré si správné fungování ověřit osciloskopem.

## 3.2. Zpracování výstupního signálu senzoru CCD pomocí řadiče STM32F3

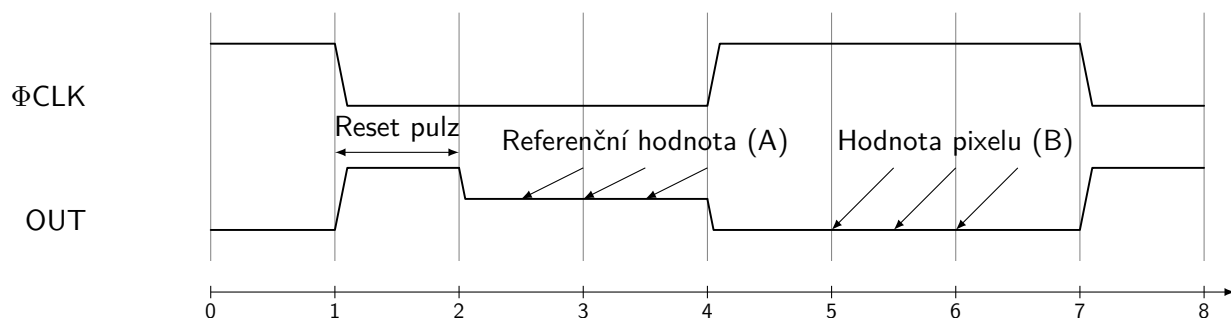
Mikroprocesor STM32F3 obsahuje celkem 4 12-bitové A/D převodníky, které můžeme použít pro digitalizaci signálu. A/D převodníky používají pro převod analogového signálu postupnou komparaci. Tyto převodníky jsou napojeny na řadič DMA, který nám umožňuje přečtené vzorky rychle a automaticky ukládat do paměti mikrořadiče.

Výstup video signál senzoru ILX551a může pracovat ve dvou režimech. V režimu bez sample-hold, se na výstupu senzoru při sestupné hraně hodinového signálu objeví krátký resetovací pulz a poté referenční hodnota pixelu. Při náběžné hraně se objeví hodnota pixelu vztážená k předchozí referenční úrovni.

V režimu sample-hold se s náběžnou hranou objeví na výstupu hodnota pixelu. Žádný reset pulz ani referenční hodnota se na výstupu neobjeví. Tento režim zjednodušuje zpracování video signálu, na druhou stranu tím přicházím o referenční hodnoty pixelů a tudíž i o přesnost. Režim je možno za běhu měnit z PC aplikace (viz. kapitola 3.3).

Převod analogového signálu A/D převodníkem je potřeba ve správný čas inicializovat. A/D převodník můžeme inicializovat buď programově (nastavením bitu v příslušném registru), nebo pomocí čítačů (viz sekce 3.1.3).

### 3.2.1. Zpracování signálu v režimu bez Sample-Hold



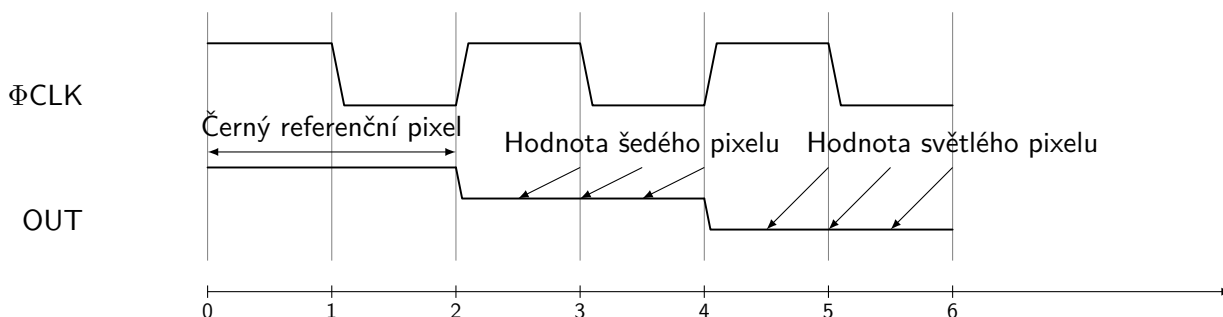
**Obr. 3.2.1.** Časový diagram výstupního signálu senzoru ILX551a v režimu bez sample-hold

Zpracování video signálu v režimu sample-hold probíhá tak, že se pro každý pixel přečte referenční hodnota (A) a pak hodnota pixelu (B). Rozdílem ( $A - B$ ) těchto hodnot dostaneme skutečnou hodnotu pixelu. Vzikne nám tak pozitivní video signál (čím větší hodnota, tím větší intenzita světla).

Vzorkování signálu se provádí ve dvou místech (viz. Graf 3.2.1). Celková intezita pixelu se získá rozdílem obou hodnot ( $I = A - B$ ). Vzorkování provádějí dva A/D

převodníky, oba napojené na řadič DMA. Jeden snímá referenční hodnoty (A), druhý snímá hodnoty pixelů (B). Výsledkem jsou tedy dvě pole bytů v paměti procesoru. Jedno pole obsahuje referenční hodnoty pixelů a druhé obsahuje hodnoty pixelů.

#### 3.2.2. Zpracování signálu v režimu se Sample-Hold



Obr. 3.2.2. Časový diagram výstupního signálu senzoru ILX551a v režimu se sample-hold

Zpracování signálu probíhá tak, že se nejdřív změří zakryté referenční pixely senzoru. Ty se pak použijí jako referenční hodnota pro zbylé pixely. Poté se přečtou hodnoty aktivních pixelů podobně jako v režimu bez sample-hold.

#### 3.2.3. Zpracování signálu programem řadiče

Zpracování signálu může probíhat jak na straně mikrořadiče, tak na straně PC. Na straně mikrořadiče je výhodné provést takové zpracování, které sníží počet posílaných dat do počítače. Proto před odesláním dat se odečte u každého pixelu referenční hodnota. Nemusíme tedy posílat referenční hodnoty a snížíme tak počet přenášených dat.

Dále ještě výslednou hodnotu vynásobíme koeficientem. Jelikož rozdíl  $A - B$  činí maximálně cca. 1,8 V a rozsah A/D převodníku je 3 V tak výslednou hodnotu vynásobíme koeficientem  $\times 1,6875$ . Rozlišení A/D převodníku je sice 12-bitů, ale vzhledem k tomu, že využíváme přibližně  $\frac{2}{3}$  rozsahu, pro výsledná data nám stačí rozlišení 8-bitů. Výsledkem zpracování je tak pole 2048 bytu reprezentující jednotlivé pixely, kde hodnota 0 značí černý pixel a 255 bílý pixel.

#### 3.2.4. Detekce saturace CCD senzoru

Při velké intenzitě dopadajícího světla se náboj zachycený na fotoelementech rozšiřuje do okolních fotoelementů. Přestože jsou referenční fotoelementy zastíněné, můžou se při saturaci jevit jako bílé. Pokud saturace senzoru CCD nastane, je dobré o tomto stavu informovat uživatele, aby případně světelné podmínky upravil.

Program mikrořadiče automaticky kontroluje, zda nedošlo k saturaci. Kontrola je důležitá především v režimu sample-hold, kde při saturaci se objeví náboj i na černých pixelech a po zpracování by se pixely jevily jako černé (přestože jsou v bílé).

Kontrola probíhá na referenčních pixelech senzoru. V normálním režimu se kontroluje rozdíl mezi referenční a skutečnou hodnotou jednoho referenčního pixelu. Pokud tento rozdíl přesáhne určitou mez, je detekována saturace. V režimu sample-hold se porovnává absolutní hodnota napětí na černém pixelu s předem stanovenou hodnotou. Pokud je absolutní hodnota menší, použije se jako referenční hodnota předem stanovená hodnota (tu jsem určil na základě referenční hodnoty, když se senzor CCD nenacházel v saturaci).

### 3.3. Přenos dat po sběrnici USB

Řadič STM32F3 obsahuje několik komunikačních rozhraní. Já jsem pro svoji aplikaci zvolil rozhraní USB, protože je dostupné na drtivé většině dnešních počítačů. Navíc modul STM32F3Discovery již obsahuje příslušný konektor. Řadič STM32F3 podporuje rozhraní USB v režimu “full-speed”.

Pro přenos dat přes sběrnici USB jsem použil aplikaci virtual COM port<sup>1</sup>, která na USB simuluje klasickou sériovou linku [5]. Výhodou tohoto přístupu je poměrně snadná implementace. Na straně PC můžeme použít knihovny pro práci se sériovou linkou. Navíc uživatel nemusí instalovat speciální ovladače, což obvykle vyžaduje administrátorská práva. Na straně mikrořadiče pak můžeme vycházet z ukázkového kódu programu, který firma STMicrotronics dodává k vývojovým modulům.

Nevýhodou tohoto přístupu je, že přenos dat (na straně PC) řídí ovladač, který nemá informaci o tom kolik dat má očekávat při čtení z portu. Navíc se data a řídicí požadavky přenášejí jedním kanálem a je třeba je od sebe nějak rozlišit.

#### 3.3.1. Popis aplikace virtual COM port

Aplikace virtual COM port, může využít pro přenos dat jak typ bulk, tak i typ isochronous. Výhodou typu bulk je, že máme zajištěn spolehlivý přenos. Nevýhodou je, že oproti ostatním přenosům má nejmenší prioritu a dostává prostor, když na sběrnici neprobíhá jiná komunikace.

Aplikace virtual COM port používá tři tzv. “endpointy” (koncové body). Jeden slouží k přijímání kontrolních příkazů (např. nastavení “baud rate”). Tyto příkazy můžeme ignorovat (používají se např. při převodu rozhraní RS-232 na rozhraní USB). Další dva “endpointy” slouží k přenosu dat. Jeden “endpoint” přenáší data do zařízení a druhý ze zařízení.

Já jsem použil přenos typu bulk. Pokud budeme předpokládat, že na sběrnici není připojené žádné jiné zařízení, pak můžeme využít až 90% kapacity sběrnice (10% je vyhrazeno pro systémovou komunikaci).

#### 3.3.2. Komunikace mezi zařízením a počítačem

Komunikace je podobná komunikaci po seriové lince. Počítač vyšle znak 0x03 a mikrořadič nejdříve vyšle znak 0x03 (0x83 v případě saturace) a pak sérii 2048 bytů, která odpovídá jednotlivým pixelům.

Nastavení doby expozice probíhá vysláním znaku 0x07 a jednoho bytu, který odpovídá době expozice v mikrosekundách. Vysláním znaku 0x10 počítač pomocí dalšího bytu nastaví příslušný režim (s/bez sample-hold).

| Znak | Význam dalších bytů                     | Popis                           |
|------|---|---------------------------------|
| 0x03 | -                                       | Vyžádá posláni řádku ze senzoru |
| 0x07 | Doba expozice v $\mu s$                 | Nastaví dobu expozice           |
| 0x10 | Režim (nenulový $\implies$ sample-hold) | Nastaví režim čtení             |

**Tabulka 3.3.1.** Příkazy posílané ze strany počítače přes Virtual COM port

<sup>1</sup> Specifikace je součástí “Communication Device Class” specifikace, která je součástí USB specifikace

### 3. Realizace řádkové kamery s řadičem STM3F3

|             |              |
|-------------|--------------|
| Úvodní znak | Data         |
| 0x03        | Pixely řádku |

**Tabulka 3.3.2.** Příkazy posílané ze strany řadiče přes Virtual COM port

#### 3.3.3. Ukončení přenosu v aplikaci Virtual COM port

Při použití přenosu typu bulk je čtení ze zařízení ukončeno [5]:

- Pokud bylo přečteno požadované množství dat
- Pokud byl přijat paket, který nevyužil maximální velikost paketu
- Pokud byl přijat prázdný paket

Vzhledem k tomu, že ovladač neví, kolik dat má přečíst a velikost přenášených dat je obvykle dělitelná velikostí paketů, je proto nutné nakonec poslat prázdný paket. Pokud bychom tak neučinili, ovladač by mohl vyčkávat na příjem dalších dat a nepředal by je pak komunikující aplikaci.

Naopak když budeme posílat velké bloky dat, je dobré využívat celé velikosti packetu. Tím řekneme PC ovladači, že na zařízení jsou dostupná data, která může přečíst.

#### 3.3.4. Realizace v programu mikrořadiče

První věc, kterou musíme provést na straně mikrořadiče, je správné nastavení popisu zařízení, které se bude předávat počítači. Zde je třeba také nastavit maximální velikost datových paketů. O předání těchto dat počítači se už postará ovladač USB.

Dále pak stačí z pomocných bufferů řadiče USB, přiřazených k jednotlivým “endpoin-tům”, číst příslušné požadavky a naopak při posílání data do těchto bufferů zapisovat.

#### 3.3.5. Měření rychlosti přenosu USB

Provedl jsem jednoduché měření rychlosti přenosu dat pomocí aplikace virtual COM port. Měření probíhalo za podmínky, kdy ke sběrnici USB nebylo připojeno žádné jiné zařízení a program mikrořadiče nevykonával, kromě posílání dat, žádnou jinou činnost. V ovladači USB jsem nastavil velikost paketů na 64 bytů, což je maximum pro přenos typu bulk v režimu “full-speed”. Tímto postupem jsem změřil přenosovou rychlost 800 kb/s.

Dále jsem měřil počet snímků za sekundu, které stihne poslat mikrořadič do zobrazovací aplikace. Měření probíhalo přímo v zobrazovací aplikaci. Naměřil jsem rychlost 14,2 snímků za sekundu, při době expozice 10 ms.

Možností jak zrychlit přenos by mohla být nějaká forma synchronizace mezi řádkovou kamerou a PC aplikací. Případně efektivnější kód na straně mikrořadiče.

### 3.4. Obvodové řešení řádkové kamery s řadičem STM32Fx

Obvodové řešení řádkové kamery musí řešit tři základní úkoly: propojit senzor CCD s řadičem, jednotlivé výstupy mikrořadiče a senzoru CCD upravit tak, aby vyhovovaly povoleným vstupním parametrům připojené součástky, a zajistit součástkám stabilní zdroj napětí.

Obvodové řešení se skládá ze dvou hlavních částí: z desky, na které se nachází CCD senzor společně se zesilovačem a převodem logických úrovní, a z vývojového modulu.

K zapojení mikrořadiče jsem použil vývojový modul STM32F3Discovery. Ten zajistí procesoru stabilní hodinový signál, napěťový regulátor z 5 V na 3,3 V a zapojení dvou

USB portů. Jeden slouží k programování a ladění aplikace, druhý použijeme na přenos dat do PC.

#### 3.4.1. Rozvod napájení

Jelikož CCD senzor vyžaduje dvě úrovně napájení (5 V a 9 V), musíme použít dva napěťové regulátory. Teoreticky bychom mohli pro napětí 5 V použít napájení z USB sběrnice, pak bychom ale museli řešit situaci, kdy je externí zdroj napětí odpojen a USB připojeno a naopak, kdy je USB připojeno a externí zdroj odpojen. Použijeme regulátory 7805, 7809 a externí zdroj napětí 12 V. K výstupu regulátorů připojíme blokovací kondenzátory.

Mezi napětím 5 V a 9 V je umístěna ochranná dioda, aby nenastala situace, že bude aktivní pouze 5 V napájení senzoru CCD, což by senzor mohlo poškodit. Dále je ochranná dioda u napájení externího zdroje, která má zabránit přepólování vstupního napětí. Nakonec je zde ještě zapojena LED dioda s odporem do série, která indikuje aktivní napájení.

#### 3.4.2. Propojení řadiče STM32F3 se senzorem CCD

Mikrořadič STM32F3 používá na výstupu 3,3 voltovou CMOS logiku, kterou musíme převést na 5-ti voltovou CMOS logiku pro senzor CCD. Pro převod použijeme obvod 74AHCT573, který funguje jako obvod latch a na vstupech je kompatibilní s logikou TTL. Navíc jeho vstupy jsou 5 V tolerantní, takže nedojde k poškození, pokud připojíme na napájení pouze mikrořadič.

Použité vstupní singály obvodu 74AHCT573 zapojíme s externím pull-down rezistorem. Tím definujeme logické úrovně vstupů v situaci, kdy není mikrořadič napájen, případně není jeho program inicializován. Piny “latch enable” připojíme na napájení (aktivní úroveň) a “output enable” připojíme na zem (aktivní úroveň). Nepoužité vstupní piny připojíme na zem. Nepoužité výstupní piny můžeme nechat nezapojené. Napájení obvodu 74AHCT zablokujeme kondenzátorem.

Pro připojení výstupu senzoru CCD použijeme PNP tranzistor (C556B W33). Do báze připojíme výstup senzoru, k emitoru připojíme napájecí napětí 9 V přes odpor 3 k $\Omega$  a kolektor připojíme na zem. Toto zapojení je doporučeno v katalogovém listu [3].

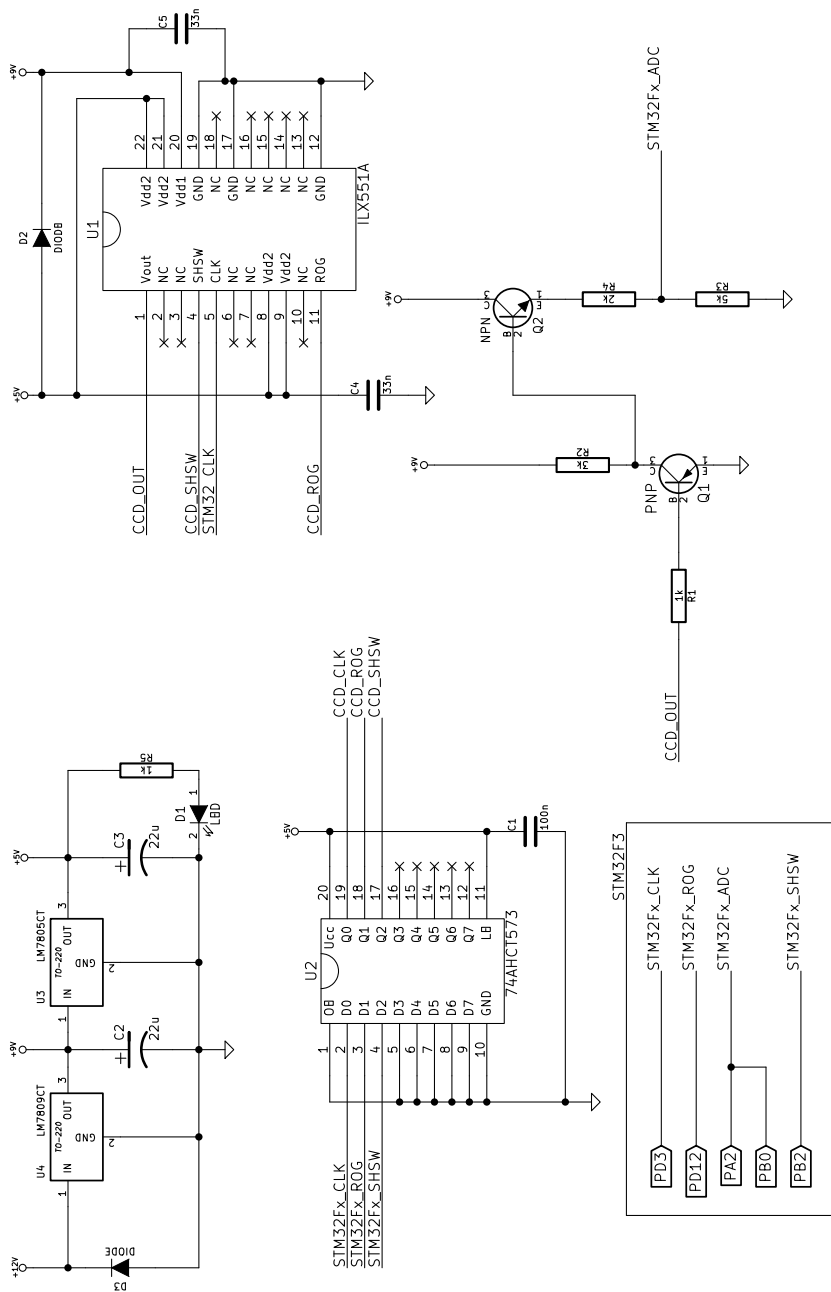
Výstupní napětí na emitoru PNP tranzistoru je stále velké, proto zapojíme NPN tranzistor (C546B W28). Bázi připojíme na emitor PNP tranzistoru, kolektor na napájení 9 V a k emitoru připojíme napěťový dělič z odporů 2 k $\Omega$  a 5 k $\Omega$ . Tím posuneme výstupní napětí o 0,7 V směrem k zemi a poté snížíme napěťovým děličem na  $\frac{2}{7}$ . Dostaneme tak výstupní napětí v rozsahu cca. 0,83 – 2,63 V.

#### 3.4.3. Schéma zapojení

Na obr. 3.4.1 se nachází schéma zapojení s modulem STM32F3Discovery. Na schématu není znázorněné zapojení procesoru v modulu STM32F3Discovery, protože modul obsahuje několik dalších součástí (akcelerometr, LED diody), které v aplikaci nejsou použity.

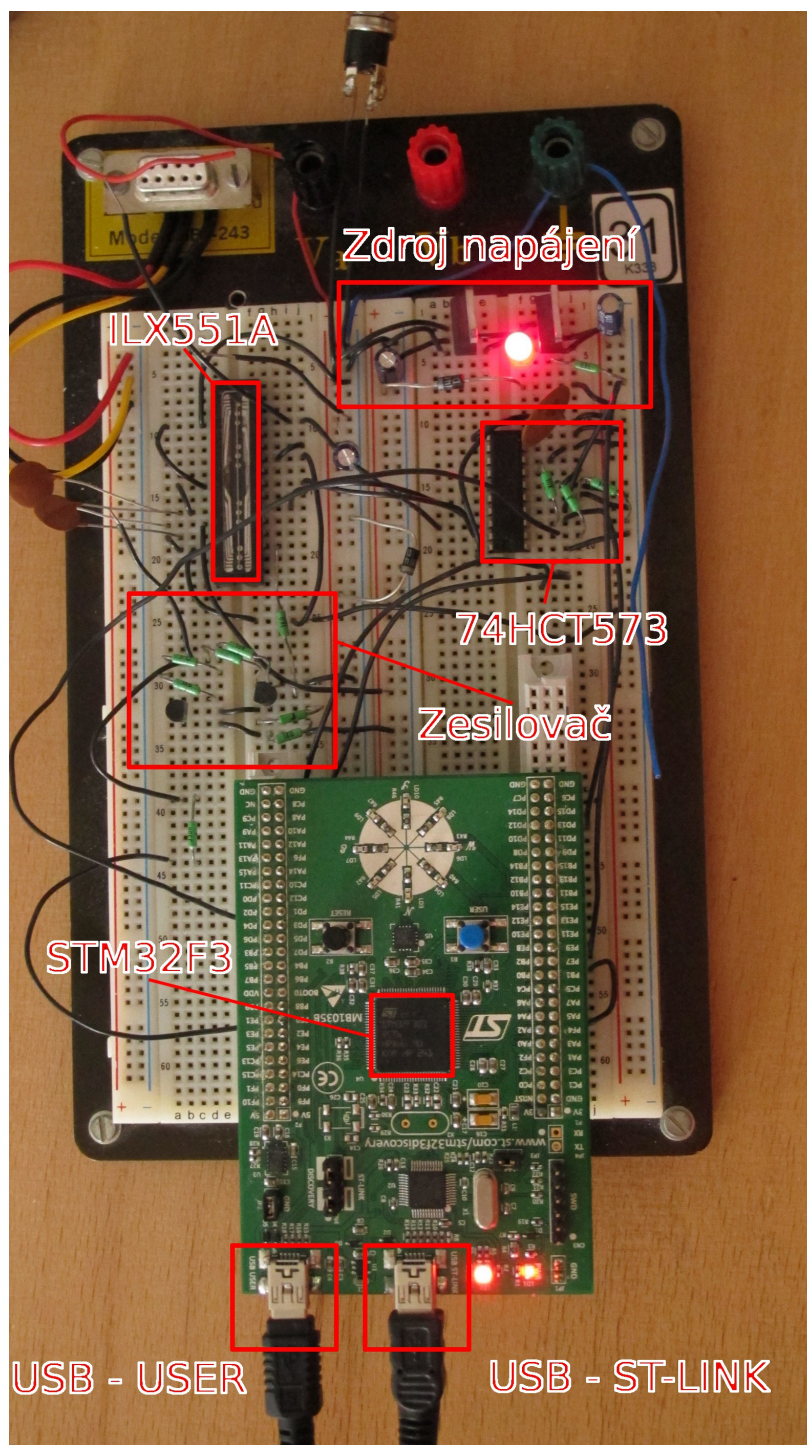
Na obr. 3.4.2 je vyfocená realizace na kontaktním poli. Nevýhodou tohoto prototypu byla nemožnost natočit senzor jinam než nahoru.

3. Realizace řádkové kamery s řadičem STM32F3



Obr. 3.4.1. Schéma zapojení senzoru CCD s modulem STM32F3Discovery





Obr. 3.4.2. Zapojení snímače s vývojovým kitem STM32F3-Discovery na kontaktním poli

### 3.4.4. Umístění objektivu

Pro umístění objektivu na kontaktní pole jsem využil staré kovové kostry počítačového zdroje, kterou jsem z obou stran zastínil kusem látky. Otvor v kostře jsem udělal v místě chlazení, kde se nacházejí průduchy. Kontaktní pole je pak potřeba něčím podložit aby senzor byl ve správné vzdálenosti od objektivu.



Obr. 3.4.3. Prozatímní umístění objektivu na kontaktním poli

## 3.5. Popis řídicího programu mikrořadiče STM32F3

Firmware pro řízení mikrořadiče jsem psal v jazyce C s použitím vývojového prostředí Keil  $\mu$ Vision<sup>2</sup>, které umožňuje i přehledné ladění aplikace. Nepoužil jsem žádný operační systém, ale využil jsem knihovny pro práci s perifériemi STM32Fx<sup>3</sup> a ovladač pro USB [6]. Dále jsem využil knihovnu “microlib” vývojového prostředí Keil, která se postará o základní inicializaci zásobníku a globálních proměnných.

Program obsahuje dva vstupní body. Funkci `SystemInit`, která inicializuje základní periférie, a funkci `main`. Nejdřív se zavolá funkce `SystemInit`, poté se inicializuje knihovna “microlib” a pak se teprve zavolá funkce `main`.

### 3.5.1. Nastavení periférií

První věc, kterou musí mikrořadič po spuštění provést, je nastavení externího hodinového signálu/kryystalu. V našem případě má externí signál frekvenci 8 MHz. V případě mikrořadiče STM32F3 dostaneme pomocí fázového závěsu interní hodinový signál  $8 \times 9 = 72$  MHz. Po změně hodinového signálu musíme změnit nastavení odezvy flash paměti v registru `FLASH_ACR`, jinak se další kód programu nenačte.

<sup>2</sup> Použil jsem volně dostupnou verzi, kde je omezená velikost výsledného programu 32 kb. Pro snížení velikosti programu je nutné zapnout optimalizaci kompilátoru

<sup>3</sup> V době psaní této práce byly dostupné nové nástroje a knihovny “STM32Cube”, ty ale zatím nepodporovaly řadu STM32F3xx. Proto jsem použil starší ovladače, které byly pro STM32F4 už označené jako zastaralé

Dále musíme povolit hodinové signály používaných periférií a nastavit správně dělič hodiny signálu. Důležité je správné nastavení hodin pro řadič USB (48 MHz), zbytek nastavení záleží na naší aplikaci a maximálních povolených hodnotách.

#### 3.5.2. Zpracování přerušení

Podstatná část činnosti programu je řízena přerušeními procesoru. Program zpracovává tato přerušení:

- Přerušení USB řadiče — o zpracování se stará ovladač USB
- Přerušení DMA řadiče — nastavuje se příznak, že data jsou přečtena a připravena k odeslání
- Přerušení čítače (viz. Graf 3.1.4) — zahájení dalšího čtení

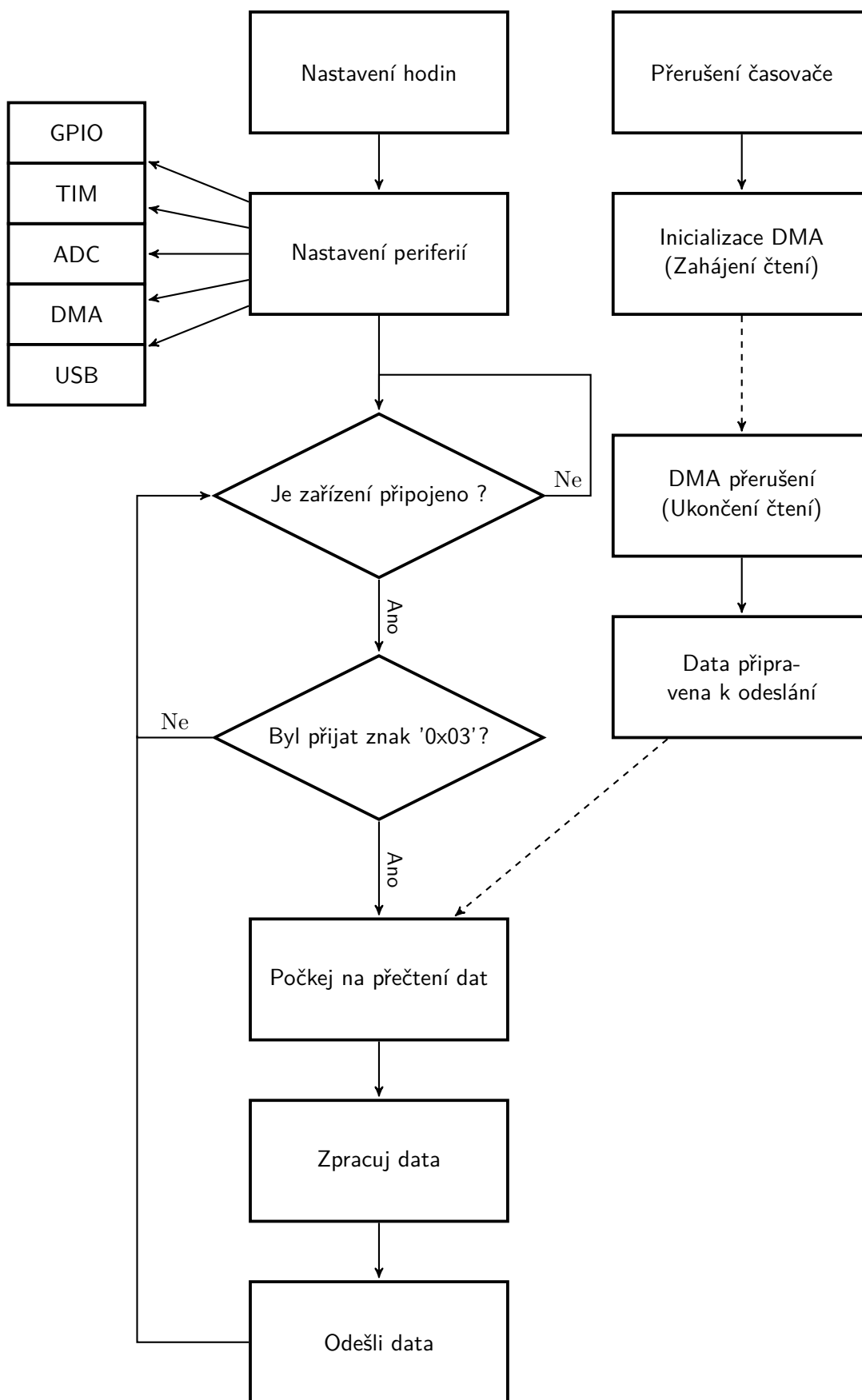
Pro zapnutí přerušení je potřeba povolit přerušení na příslušné periférii a povolit přerušení v nastavení řadiče NVIC (Nested-Vector Interrupt Controller), kde se dá nastavit i priorita jednotlivých přerušení.

Při zpracování přerušení je třeba zkontrolovat daný příznak v registru periférie, neboť některá přerušení jsou obsluhována stejnými funkcemi. Po zpracování přerušení je potřeba tento příznak vyčistit v registru periférie.

#### 3.5.3. Zpracování a odesílání dat přes USB

Před odesláním dat program mikrořadiče data zpracuje (viz. sekce 3.2.3). Pro výpočet intezity můžeme využít instrukce SIMD (Single instruction multiple data), pomocí kterých můžeme zpracovat 2 vzorky najednou (v případě 12-bitových vzorků), nebo až 4 vzorky najednou (v případě 8-bitových vzorků). Použití instrukce UQSUB16 (případně UQSUB8) navíc zabrání podtečení v případě, že referenční hodnota je menší než hodnota pixelu.

Jakmile jsou data zpracována, program je pošle pomocí ovladače USB do počítače. Důležité je po ukončení přenosu odeslat prázdný paket (viz. sekce 3.3.3).



Obr. 3.5.1. Digram chodu programu na mikrořadiči STM32F3

## 4. Realizace řádkové kamery s modulem STM32F429Disco

Další možností pro realizaci řádkové kamery je využití procesoru STM32F429. Vývojový modul STM32F429Disco nabízí kromě samotného procesoru také LCD displej o rozlišení 240x320 pixelů a paměť SDRAM o velikosti 8 Mb. Je tedy možné naměřené hodnoty přímo zobrazovat na LCD displeji a není potřeba připojovat počítač. Rozšíření paměti nám zase dovoluje rychle ukládat naměřené hodnoty v režimu skener. To nám umožňuje složit výsledný naskenovaný obraz přímo na straně zařízení.

Nevýhodou použití tohoto modulu je, že vstup A/D převodníku se nachází blízko řídicích signálů displeje a dochází tak k nežádoucím přeslechům. Dále je omezen počet výstupních pinů, které můžeme použít, neboť většina z nich je připojena k paměti SDRAM, nebo k displeji LCD.

### 4.1. Řízení senzoru CCD pomocí řadiče STM32F4

Generování řídicích signálů řadičem STM32F4 je velmi podobné jako generování řadičem STM32F3 (viz. kapitola 3.1). Opět můžeme generovat pulzy programově, nebo pomocí čítačů. Pro generování pomocí čítačů budeme opět potřebovat tři čítače *A*, *B* a *C* (viz. sekce 3.1.3).

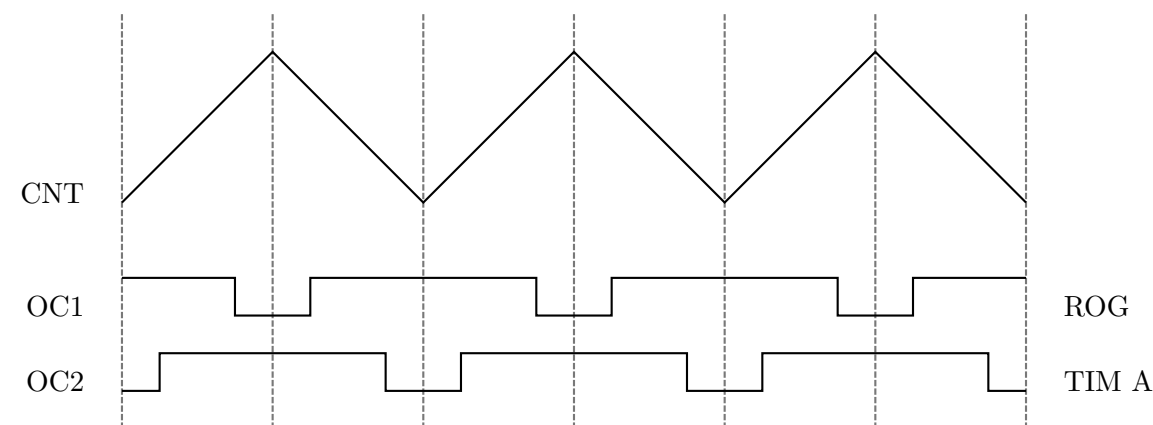
#### 4.1.1. Generování řídicích pulsů s využitím režimu “center-aligned”

Čítače řadiče STM32F4 jsou téměř stejné jako u řadiče STM32F3. Jediným podstatným rozdílem je, že čítače nepodporují kombinovaný PWM mód, který jsme použili na generování signálu ROG. Místo toho můžeme použít režim “center-aligned”. V tomto režimu čítač čítá nejdříve vzestupně a po dosažení hodnoty “Auto-reload” registru čítá sestupně k nule. Při dosažení nuly, nebo hodnoty “Auto-reload” registru, čítač generuje “update event”.

Zároveň potřebujeme, aby příslušný čítač běžel v režim “one-pulse”, ale aby se zastavil, až při návratu zpět do nuly. K vyřešení tohoto problému můžeme využít tzv. “repetition counter”, který nám umožní generovat pouze každý *n*-tý “update event”. Toto nastavení umožňují pouze čítače TIM1 a TIM8.

Při nastavování režimu PWM u čítačů TIM1 a TIM8, je, oproti jiným čítačům, nutné nastavit bit MOE (Main output enable) na 1, aby se výstupní signál objevil na pinu procesoru.

Pro generování řídicích pulsů jsem použil čítače TIM2 (čítač *A*), TIM4 (čítač *B*) a TIM1 (čítač *C*). Čítač TIM2 jsem zvolil proto, že je schopen spouštět oba A/D převodníky, a TIM1 proto, že má volné piny na modulu.



Obr. 4.1.1. Časový diagram chodu čítače  $C$  v režimu “center-aligned” a jeho OC kanálů

## 4.2. Zpracování signálu senzoru CCD pomocí mikrořadiče STM32F4

Zpracování signálu je opět podobné jako v kapitole 3.2. Nastavení A/D převodníku se podělekd liší, ale hlavní myšlenka zůstává stejná. A/D převodníky jsou opět připojeny na řadič DMA, který výsledky automaticky ukládá do paměti.

Jediným výraznějším rozdílem je, že se A/D převodníky zastaví ve chvíli, kdy DMA řadič již nepřijímá další data. Při zahájení čtení dalšího snímku je třeba vynulovat bit povolující DMA v registrech A/D převodníků a poté ho znova nastavit na 1.

### 4.2.1. Zpracování v řádkovém režimu

Při realizaci řádkové kamery s modulem STM32F429Disco se na vstupu A/D převodníku objevil šum způsobený řízením LCD displeje.

Vzhledem k tomu, že chceme zobrazovat aktuální data na displeji, nepřipadá jeho dočasné vypnutí v úvahu. Proto se výsledný signál průměruje ze 4 po sobě jdoucích měření.

Dalším možným způsobem filtrace je použití dolní propusti, kdy budeme průměrovat každý pixel s příslušným pixelem z předchozího snímku podle vzorce:

$$y_i = y_{i-1}a + x_i(1 - a) \quad (4.2.1)$$

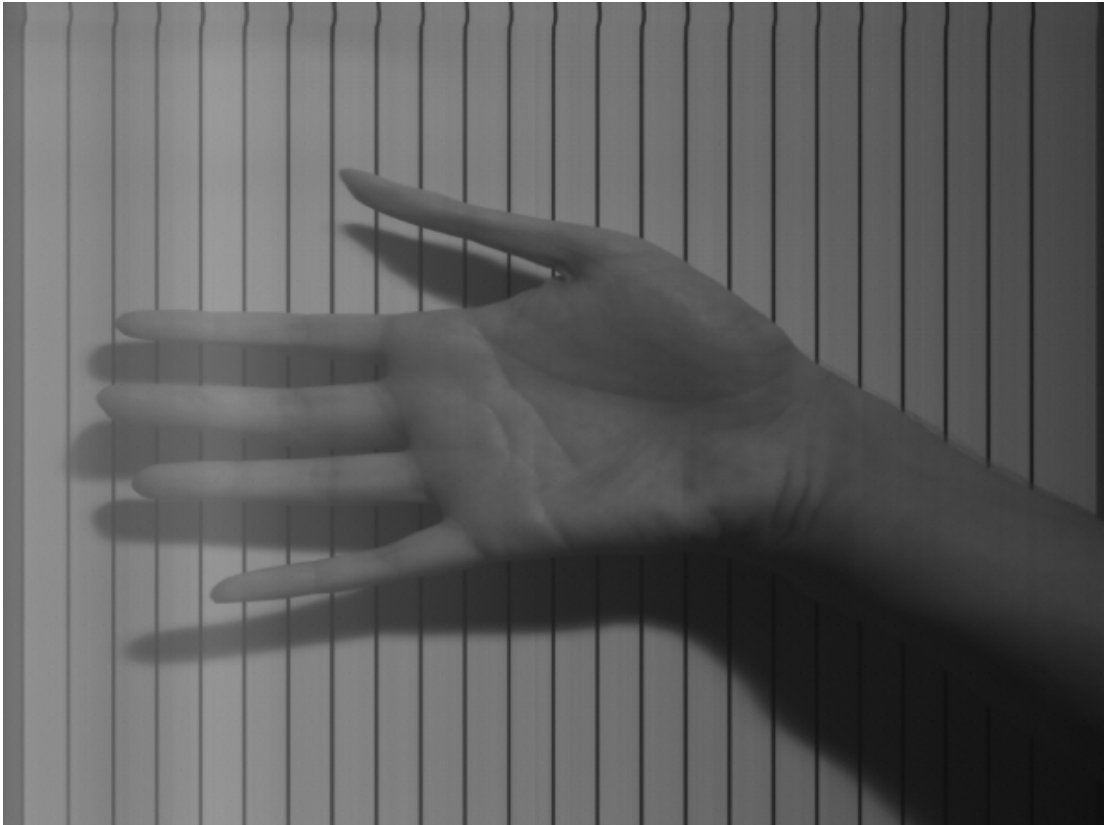
kde  $x_i$  je naměřená hodnota v čase  $i$ ,  $y_i$  je filtrovaná hodnota v čase  $i$ , a  $a \in \langle 0, 1 \rangle$ .

Výhodou tohoto přístupu je, že výsledný obraz je téměř neměnný při statické scéně (neprojevuje se šum). Nevýhodou je, že pak zpracovaný signál pomalu reaguje na rychlé změny mezi jednotlivými snímky.

### 4.2.2. Zpracování v režimu skener

Dalším výraznějším rozdílem je realizace režimu skener. Nastavení a řízení A/D převodníku je pořád stejné, ale změní se nastavení řadiče DMA. Ten je nejdříve vypnutý a čeká na zahájení skenování. Po zahájení skenování načítá jednotlivé nezpracované snímky do paměti SDRAM za sebe (bez zpracování).

Jakmile řádková kamera načte předem daný počet snímků, program přečtená data zpracuje (viz. sekce 3.2.3). Abychom mohli výsledný obrázek zobrazit na displeji, je třeba naskenovaný obrázek zmenšit na rozlišení  $320 \times 240$ .



**Obr. 4.2.1.** Naskenovaná ruka pohybující se nad senzorem CCD (rozlišení:  $640 \times 480$  pixelů)

Vzhledem k tomu, že nemáme dostatečně velkou paměť, abychom naskenovali obrázek ve stejném poměru stran jako je náš displej ( $2048 \times 1536$ ), bude výsledný obrázek v jednom směru zmenšen více než v druhém. To nám ale nevadí, protože míra zmenšení také záleží na rychlosti skenování. Navíc kdybychom skenovali 1536 řádků, museli bychom skenovaný objekt posouvat velmi pomalu, ale stále stejnou rychlostí.

První realizace fungovala tak, že se naskenuje 240 řádků a každý řádek se zkomprimuje na 320 pixelů. Poskládáním pak dostaneme výsledný obrázek  $320 \times 240$ . Ten pak stačí rovnou překopírovat do paměti displeje.

Další variantou bylo skenování v rozlišení  $640 \times 480$ , kde se obrázek zpracoval podobným způsobem. Při zobrazení na displeji se ale musel zmenšit průměrováním přes čtyři pixely. Výsledkem varianty je, že naskenovaný obrázek můžeme poslat do PC a tam už toto rozlišení dokážeme zobrazit.

Na obr. 4.2.1 vidíme výsledný obrázek skenování stažený z kamery. Snímek vznikl tak, že jsem posouval ruku nad senzorem. Na pozadí je vidět kus papíru, který sloužil k zaostření objektivu. Papír se však nepohyboval společně s rukou, proto jeho obraz neodpovídá skutečnosti.

### 4.2.3. Automatické spuštění skenování

Kromě spuštění režimu skener pomocí tlačítka se nabízí automatické spuštění při změně obrazu. Program porovná aktuální snímek s předchozím a pokud se obrazy liší, zahájí skenování.

Porovnání probíhá tak, že vybereme 64 pixelů z obou snímků a pokud jich více jak  $N$

liší s předchozím snímkem o více než  $I$ , tak se zahájí skenování. Pro program jsem zvolil hodnoty  $N = M = 8$ . Tento způsob detekce je velmi primitivní a ne příliš spolehlivý.

Další variantou by mohlo být sečtení druhých mocnin odchylek. Pokud by součet překročil určitou hranici, došlo by ke spuštění skenování.

Po ukončení skenování je po dobu 1 s funkce automatického spuštění vypnuta, aby nedošlo k nechtěnému spuštění.

### 4.3. Řízení displeje LCD pomocí řadiče STM32F4

Modul STM32F429Disco obsahuje barevný displej LCD o rozlišení  $240 \times 320$ . Mikrořadič STM32F4 obsahuje řadič pro obsluhu displeje LCD, který podporuje dvě obrazové vrstvy, ze kterých se složí výsledný obraz [7]. Řadič je napojen na paměť RAM, ze které přímo čte obrazová data jednotlivých vrstev.

Řadič podporuje několik formátů reprezentace obrazových dat. Já jsem ve své aplikaci použil 8-bitový černobílý formát. Vzhledem k tomu, že výstupní signál je také černobílý, nepřinášel by barevný formát významnou výhodu. Naopak tímto režimem ušetříme značnou část paměti. Takto obrazová data displeje zabírají 75 kb. V barevném režimu by data zabírala 3-krát více paměti (při zachování stejného barevného rozlišení pro černobílý rozsah).

Každá vrstva může mít jiné rozměry a jinou pozici na obrazovce. Já jsem pro jednoduchost použil pouze jednu vrstvu se stejnými rozměry jako má displej.

Obsluha displeje je v našem případě celkem jednoduchá. Nejdříve nastavíme řídicí registry displeje, podle parametrů konkrétního displeje. Zde můžeme vycházet z ukázkového programu dodávaného k modulu STM32F429Disco [6].

Dále inicializujeme vrstvu displeje (černobílý režim,  $240 \times 320$ ) a určíme oblast paměti odkud se budou interpretovat obrazová data. Dále musíme dbát na to, že displej je původně orientován na výšku. My ale chceme v aplikaci zobrazovat data na šířku. Dále tedy budeme předpokládat, že šířka je 320 a výška 240.

#### 4.3.1. Rasterizace signálu senzoru CCD

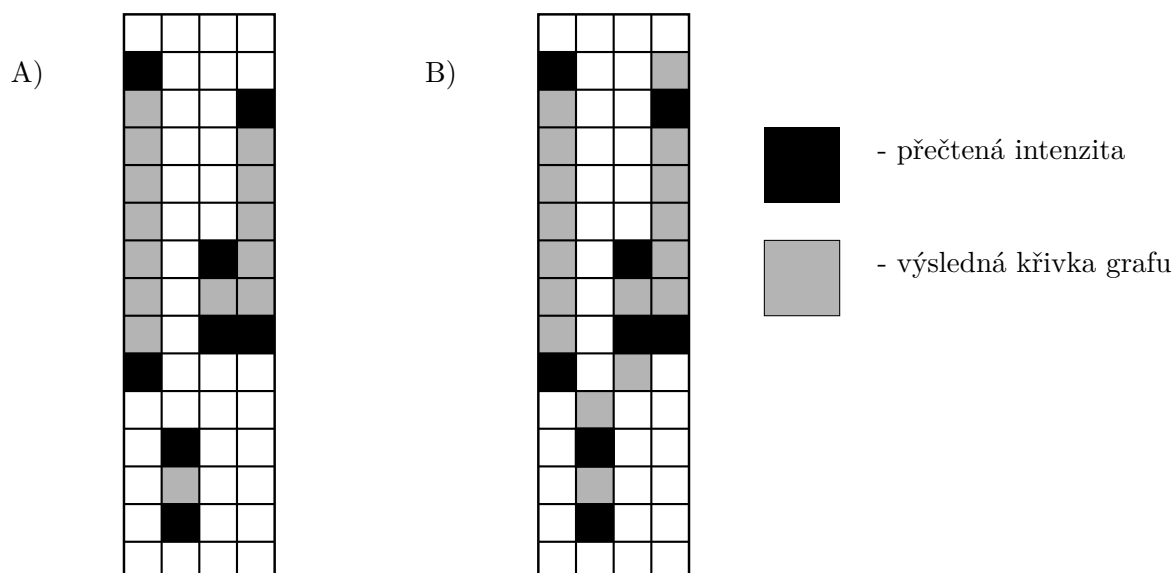
Jelikož chceme signál ze senzoru CCD zobrazit na LCD displeji, musíme ho nejdříve převést do nějaké grafické reprezentace. Nejjednodušší přístup je vytvořit ze signálu graf, kde na ose  $x$  bude pořadí pixelu a na ose  $y$  bude intenzita ozáření pixelu. Vzhledem k tomu, že senzor obsahuje 2048 pixelů a displej LCD má na šířku pouze 320, je nutné data nějak zkomprimovat. Na jeden sloupec na displeji LCD připadá několik pixelů (v průměru  $2048/320 = 6,4$ ).

Nejjednodušším řešením by bylo pro každý sloupec displeje LCD zprůměrovat příslušné hodnoty pixelů. Poté bychom vynesli bod na příslušné souřadnice  $x$  a  $y$ . Výsledek by však byl špatně čitelný a objekty menší než 6 pixelů by kvůli průměrování byly méně patrné.

Trochu složitějším, ale efektivnějším řešením je, že se pro každý sloupec displeje LCD vybere z příslušných pixelů minimum a maximum. V příslušném sloupci pak nakreslíme černou čáru mezi maximem a minimem (viz. Obr. 4.3.1 A). Docílíme tím toho, že i výchylka o velikosti jednoho pixelu bude na grafu jasně viditelná.

Dále je dobré, aby výsledný graf byl spojitý. Toho dosáhneme porovnáním maxima a minima s hodnotami maxima a minima z předchozího sloupce. Pokud je aktuální maximum menší než předchozí minimum, je za aktuální maximum dosazeno předchozí minimum. Pokud je aktuální minimum větší než předchozí maximum, je za aktuální minimum dosazeno předchozí maximum (viz. Obr. 4.3.1 B).





Obr. 4.3.1. Příklad rasterizace signálu senzoru CCD

Výhodou tohoto řešení je, že pokud máme úzké pruhy světlých pixelů (užší než poměr mezi pixely senzoru CCD a LCD displeje) na tmavém pozadí (nebo naopak), jsou tyto pruhy na grafu zřetelné.

Také bychom měli řešit úpravu dat v ose  $y$ , kdy hodnoty intenzity nabývají rozsahu 0 až 255, ale výška displeje je pouze 240 pixelů.

#### 4.3.2. Ovládání displeje LCD v režimu skener

Jak bylo dříve zmíněno, objevuje se na vstupu A/D převodníku šum způsobený řízením displeje LCD. Proto při zahájení skenování je vypnutá aktivní vrstva dokud není skenování ukončeno. Tak se alespoň omezí šum na videosignálu.

Po ukončení skenování se na displeji objeví výsledný obrázek. Zobrazení výsledného obrázku spočívá jen v překopírování dat do paměti přiřazené příslušné vrstvě LCD displeje. Je však nutné překonvertovat orientaci obrázku, protože displej je zapojen “na výšku”.

### 4.4. Komunikace po sběrnici USB s modulem STM32F429Disco

Mikrořadič STM32F4 obsahuje dva řadiče USB [7]. Jeden podporuje pouze režim “full-speed” (FS), druhý podporuje i režim “high-speed” (HS). V modulu STM32F429Disco je ke konektoru připojen řadič USB HS, ale využívá fyzickou vrstvu, která podporuje pouze režim “full-speed”.

Programové ovládání řadiče USB se značně liší od mikrořadiče STM32F3 a musí tak použít jiný ovladač. Výhodou je, že knihovna ovladače již obsahuje předpřipravené šablony pro základní třídy zařízení USB včetně Communications device class, pod které se řadí aplikace Virtual COM port.

Komunikace probíhá stejně jako v případě řadiče STM32F3 (viz. sekce 3.3.2), výjimkou je přenos naskenovaného obrazu. Další odlišnost je, že modul STM32F429Disco používá konektor micro-USB.

#### 4.4.1. Přenos naskenovaného obrazu do PC

Přenos obrazu do PC probíhá na základě žádosti ze strany PC. PC nejdříve vyšle přes COM port znak 0x13. Zařízení pak odpoví také znakem 0x13, poté následují 2 byty udávající šířku obrazu v pixelech, poté další 2 byty udávající výšku. Poté už následují obrazová data, kdy jeden byte reprezentuje jeden pixel.

#### 4.4.2. Měření rychlosti přenosu USB

Provedl jsem jednoduché měření počtu přečtených snímků za vteřinu na straně PC aplikace. Naměřil jsem rychlost 9,9 snímků za vteřinu. To je méně než v aplikaci s řadičem STM32F3. Menší rychlost je nejspíše způsobena použitím cyklického bufferu na straně ovladače USB. Ten pak občas odešle paket, který nevyužije maximální velikost paketu i když se v bufferu nacházejí další data. Toto se děje při přetečení bufferu.

### 4.5. Ovládání aplikace

Aplikaci je možné ovládat přes PC, jak je popsáno v kapitole 5, ale také pomocí tlačítka na modulu.

Po spuštění zařízení je aplikace v řádkovém režimu, kde na displej vykresluje graf právě snímaného řádku. Po stisknutí tlačítka se zařízení přepne do režimu skener, kde vyčkává na další stisk tlačítka, který zahájí skenování. Pokud je tlačítko znovu stisknuto během právě probíhajícího skenování, přepne se zařízení zpět do řádkového režimu.

Pro indikaci režimu slouží dvě LED diody. Pokud svítí červená dioda, je zařízení v režimu skener. Pokud svítí i zelená dioda, zařízení právě skenuje.

Po dokončení skenování se na displeji zařízení zobrazí naskenovaný obrázek. Během skenování je displej částečně vypnutý, aby nerušil vstup A/D převodníku.

### 4.6. Popis řídicího programu modulu STM32F429Disco

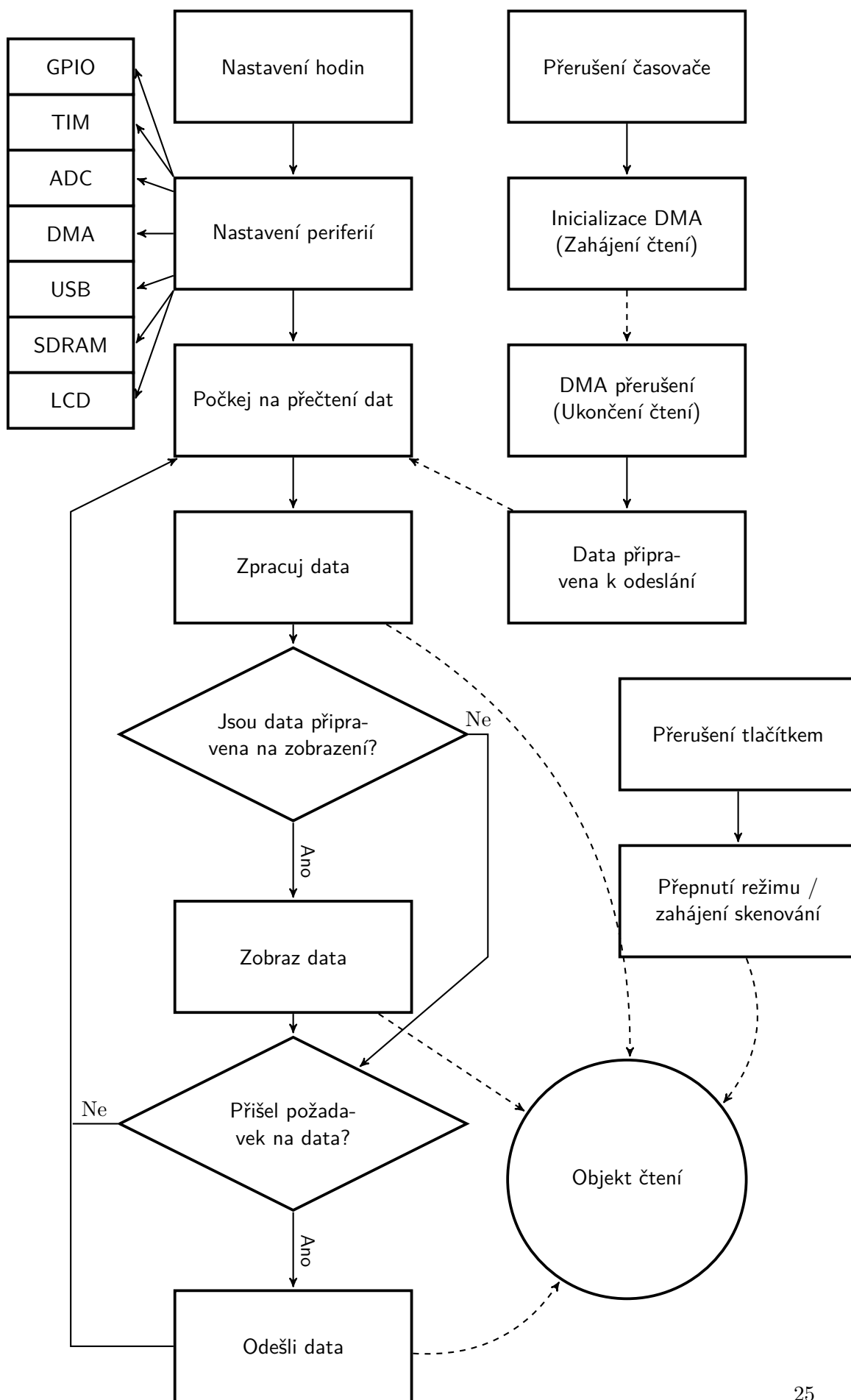
Některé části kódu jsou podobné řešení s procesorem STM32F3. Frekvenci hodinového signálu jádra jsem nastavil na 168 MHz. Zásadním rozdílem ve funkčnosti je skutečnost, že program nečeká na požadavek PC než signál zpracuje, ale zpracovává jej průběžně a zobrazuje na displeji.

#### 4.6.1. Připojení externí paměti SDRAM

Pro nastavení externí paměti SDRAM je třeba nastavit příslušné piny a konfigurovat registry řadiče FMC (Flexible memory controller). Řadič FMC pak namapuje paměť SDRAM do předem stanoveného paměťového prostoru (viz. reference manual).

#### 4.6.2. Čtení stisku tlačítka

Další odlišností je detekce stisku tlačítka, které je součástí modulu STM32F429Disco. Detekce stisku je řízena přerušením. Aby se zabránilo zákmitům tlačítka (tedy nechtěnému vícenásobnému stisku tlačítka), obsluha přerušování nejdříve zkontroluje, zda neproběhlo přerušování před 1 ms. Pokud ano, přerušování je ignorováno



Obr. 4.6.1. Digram chodu programu na kitu STM32F4-IDiscovery

## 4.7. Porovnání procesoru STM32F3 a STM32F4

Řada STM32F4 na rozdíl od STM32F3 nabízí širší spektrum dostupných periférií, zvláště co se týče komunikace (podporuje Ethernet High-speed USB apod.) a uživatelského rozhraní (připojení na LCD displej). Na druhou stranu je obsluha periférií složitější.

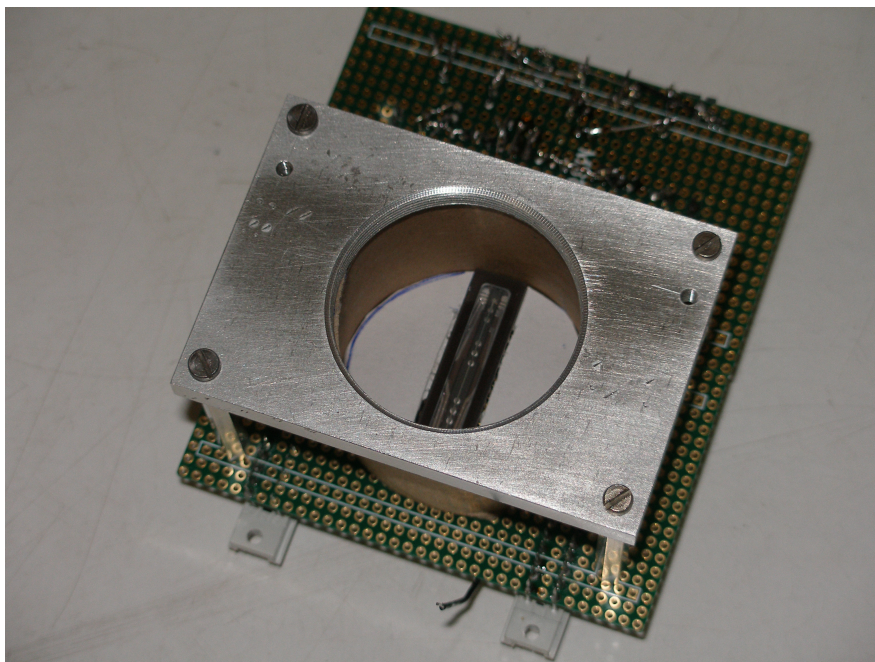
Například při napojení A/D převodníku na řadič DMA, mikrořadič F4 sleduje, zda nedošlo k přetečení DMA bufferu. Při každém přetečení (i když s ním počítáme), musíme nejdříve vypnout DMA na straně A/D převodníku a zase jej zapnout.

## 4.8. Využití modulu STM32F407Discovery

Při vývoji řádkové kamery jsem také zkoušel použít modul STM32F407Discovery. Ten obsahuje mikroprocesor STM32F407, který je velmi podobný STM32F429. Modul neobsahuje displej LCD ani paměť SDRAM.

Realizace řídicího programu je spíše podobná jako u realizace s mikrořadičem STM32F3, akorát je jiné ovládání periférií. Dalším rozdílem je zapojení USB, protože modul STM32F407 je připojen přes řadič USB FS (“full-speed”), zatímco STM32F429Disco je připojen přes USB HS (“high-speed”). Tento rozdíl však za nás vyřeší ovladač řadiče USB.

Pro generování řídicích pulsů jsem použil čítače TIM2 (čítač *A*), TIM4 (čítač *B*) a TIM8 (čítač *C*). Jako vstupy a výstupy jsem použil piny: PA2 (CLK), PB14 (ROG), PB2 (SH) a PB0 (ADC).



Obr. 4.10.1. Realizace stínění senzoru CCD pomocí kartonu a tvrdého papíru

## 4.9. Obvodové řešení řádkové kamery s modulem STM32F429Disco na kontaktním poli

Obvodové řešení jsem použil téměř stejné jako při realizaci kamery s procesorem STM32F3 na kontaktním poli. Pouze jsem použil jiné piny a v případě A/D převodníků použil společný pin, na který jsou napojeny oba převodníky. Také jsem zkusil na vstup A/D převodníku připojit paralelně kondenzátor (330 pF). Ten funguje jako dolní propust a částečně tak zmírnil šum způsobený displejem LCD. Přesto však byl šum stále patrný.

## 4.10. Kompaktní obvodové řešení řádkové kamery s modulem STM32F429Disco

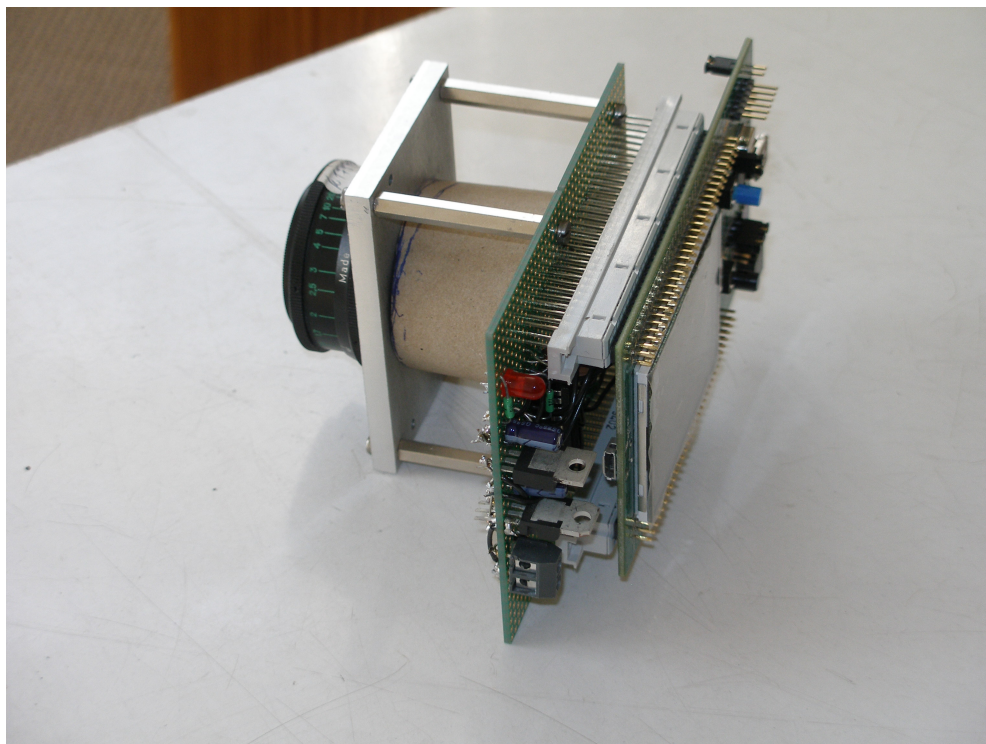
Dále jsem se pokusil o realizaci kompaktnějšího a přenositelnějšího modulu, který by demonstroval možnosti režimu skener. Zařízení se skládá ze dvou destiček, na které jsou připájeny součástky, a dva konektory, na které připevníme modul STM32F429Disco. Senzor CCD a obvod 74AHCT573 jsem nepájel přímo, ale použil jsem sokl. Všechny součástky jsou umístěné z jedné strany s výjimkou senzoru, který je umístěn z druhé strany. Snáze tak zajistíme stínění senzoru CCD.

Na obr. 4.10.1 je vidět realizace stínění senzoru CCD. Ze strany je senzor stíněn kartonovou rolí. Jelikož spodní destičky obsahují díry pro připájení součástek, je vhodné tyto díry zastínit. Já jsem použil kus tvrdého papíru, který jsem vložil mezi senzor CCD a sokl.

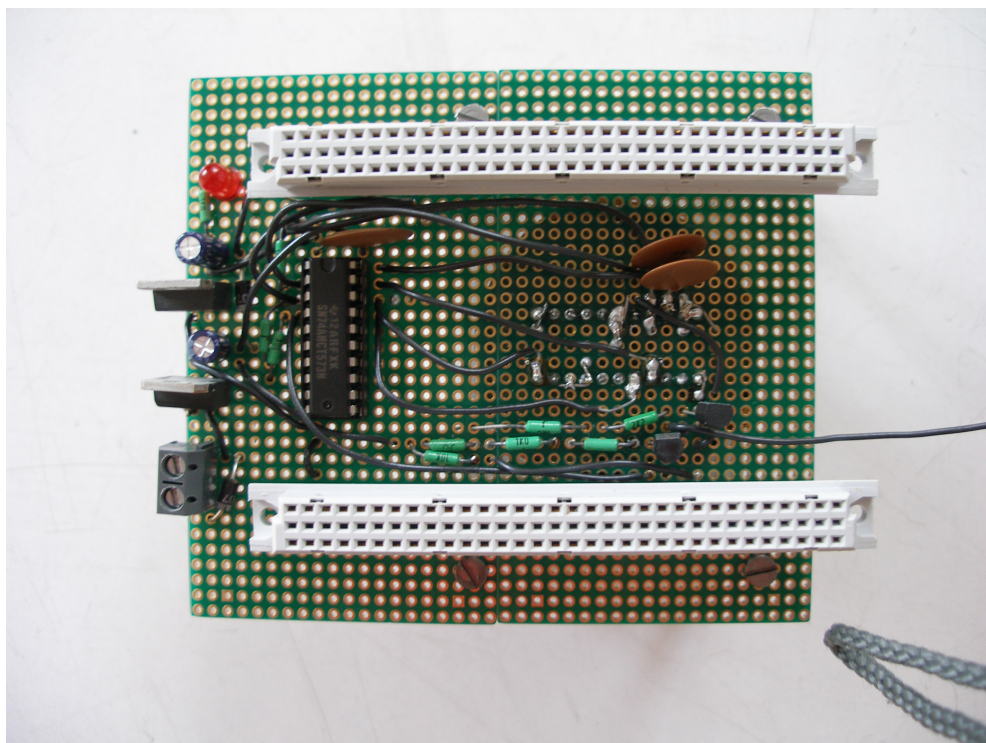
Na obr. 4.10.2 je vidět prototyp s připojeným modulem a připevněným objektivem. Na straně se nachází svorkovnice pro připojení napětového zdroje. Vedle jsou umístěny napěťové regulátory, které musejí být na kraji, protože by se nevešly mezi desku a vývojový modul.

Na obr. 4.10.3 je vidět rozložení jednotlivých součástek a jejich propojení.

4. Realizace řádkové kamery s modulem STM32F429Disco



**Obr. 4.10.2.** Kompaktní prototyp řádkové kamery s modulem STM32F429Disco



**Obr. 4.10.3.** Rozmístění a zapojení součástek v prototypu řádkové kamery

## 5. Realizace zobrazovacího programu na PC

Software pro zobrazování dat ze senzoru jsem napsal v jazyce C#. Tento jazyk jsem zvolil díky snadnému a rychlému vývoji. Další výhodou je snadný převod programu na jinou platformu díky projektu Mono<sup>1</sup>.

Program může ovládat jak kameru s procesorem STM32F3, tak i prototyp s modulem STM32F429Disco. Hlavní funkce programu jsou:

- Přečtení a zobrazení řádku ze senzoru
- Poskládání více řádků do bitmapy (režim skener)
- Přečtení naskenovaného obrázku ze zařízení (pouze s modulem STM32F4Disco)
- Uložení přečtených dat
- Nastavení doby expozice
- Zapnutí/vypnutí režimu “sample-hold”
- Upozornění uživatele, že je senzor v saturaci

### 5.1. Popis uživatelského rozhraní

Rozhraní aplikace se skládá se tří částí (viz Obr. 5.1.1):

**Horní panel** Zde uživatel může nastavit základní parametry senzoru: režim, dobu expozice a způsob čtení (opakovaně nebo jednorázově)

**Hlavní okno** Zde uživatel vidí zobrazená data, buď v řádkovém režimu, kde je vidět graf i reprezentace v odstínech šedi, nebo v režimu skener, kde uživatel vidí naskenovaný obrázek

**Dolní lišta** Zde uživatel vidí stav programu. V levém dolním rohu se vypisuje, zda je program připojen k zařízení. Dále se vypisuje stav čtení a upozornění, zda nedošlo při přenosu dat k chybě. A nakonec se zde ukazuje, zda nedošlo k saturaci senzoru.

### 5.2. Připojení PC k zařízení

První věc, kterou musí program po spuštění provést, je nalézt koncové zařízení a připojit se k příslušnému COM portu. Číslo COM portu zadá uživatel do příslušného pole. Po otevření portu program nastaví dobu expozice a zapne/vypne “sample-hold” podle příslušných hodnot v uživatelském rozhraní.

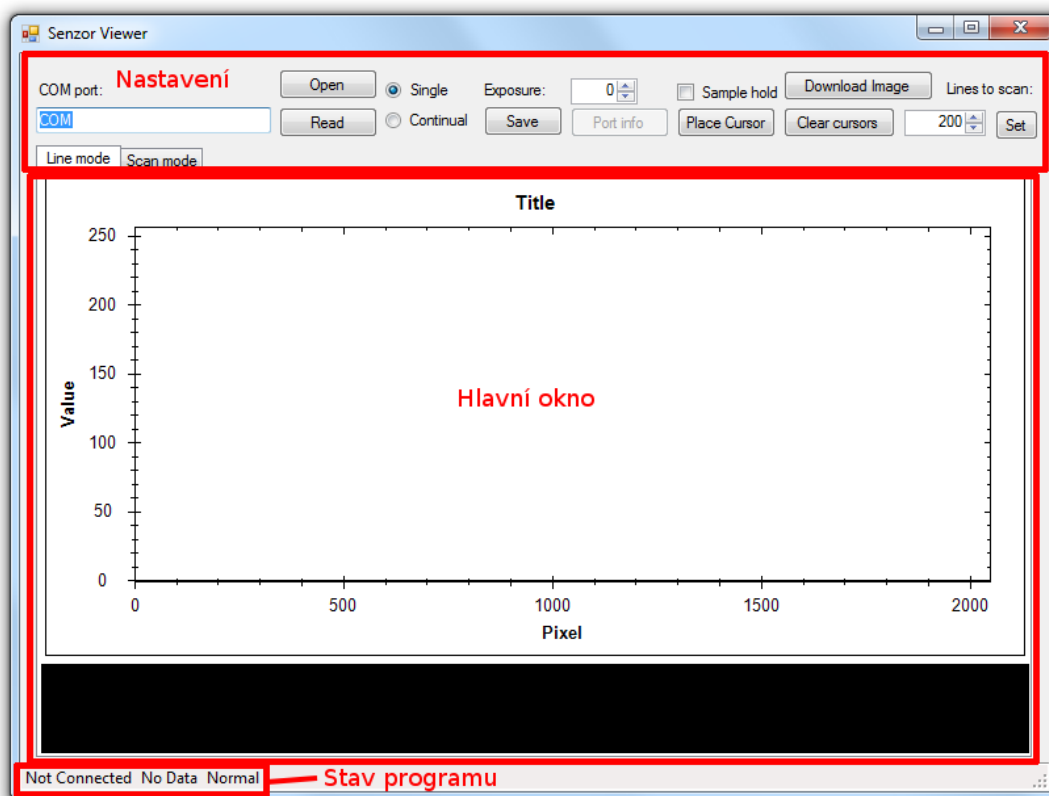
### 5.3. Příjem dat přes USB

Čtení řádku z USB je inicializováno stiskem tlačítka “Read” v horním panelu programu, nebo po dokončení předchozího čtení, je-li zapnutý režim skener případně je zapnuté stálé čtení (položka “Continual”).

---

<sup>1</sup><http://www.mono-project.com/>

## 5. Realizace zobrazovacího programu na PC



Obr. 5.1.1. Uživatelské rozhraní zobrazovacího programu

Před přečtením se nejdříve vyprázdní vstupní buffer sériové linky. Tím se zabrání čtení starých dat, které mohla v bufferu uvíznout např. po vypršení časového limitu při předchozím čtení.

Poté se pošle znak  $0x03$ , čímž se zahájí čtení. Pak program čeká na přijetí znaku  $0x03$  nebo  $0x83$  v případě saturace. Dále se přečte 2048 bytů reprezentující jednotlivé pixely. Nakonec program předá data příslušné zobrazovací komponentě.

### 5.4. Zobrazení dat v řádkovém režimu

Pro zobrazení v řádkovém režimu jsem použil komponentu Zed Graph<sup>2</sup>, která umožňuje snadno zobrazovat grafy, včetně přiblížení.

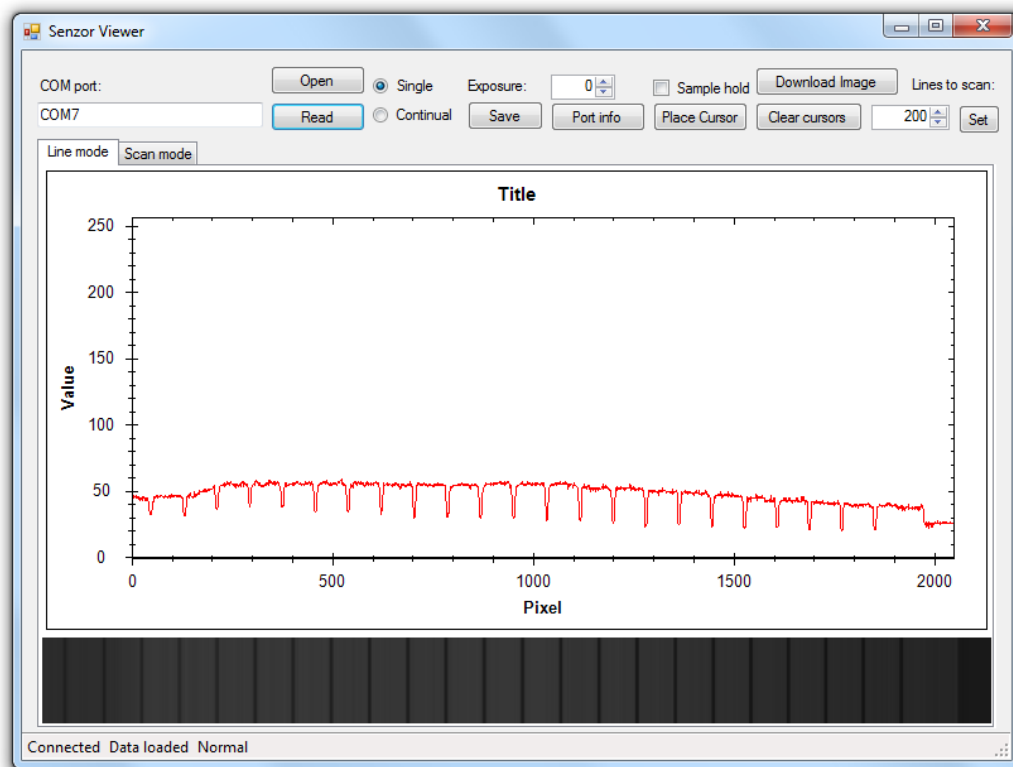
Pod grafem se nachází pruh, který reprezentuje obrazová data ze senzoru. Vzhledem k tomu, že běžné rozlišení monitoru má méně pixelů na šířku než 2048, musíme data zmenšit na požadované rozlišení.

### 5.5. Zobrazení dat v režimu skener

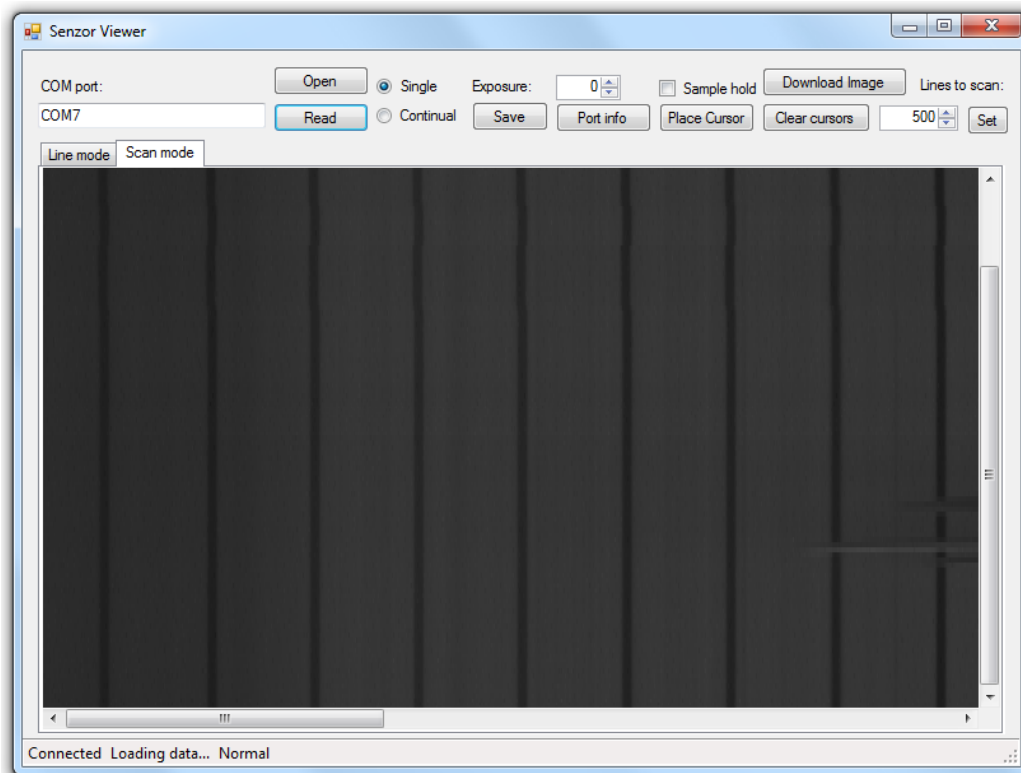
Zobrazení v režimu skener se skládá z jedné bitmapy, ve které se při skenování zezdola postupně objevují naskenované řádky. Je-li zaškrtnut políčko “Continous”, probíhá skenování stále dokola.

<sup>2</sup> <http://sourceforge.net/projects/zedgraph/>





Obr. 5.5.1. Zobrazení dat v řádkovém režimu PC aplikací



Obr. 5.5.2. Zobrazení dat v režimu skener PC aplikací

## 5. Realizace zobrazovacího programu na PC

V pravém horní rohu se dá nastavit výška skenovaného obrázku (počet řádků). Při změně se aktuálně naskenovaná data zahodí. Po dokončení skenování je možné uložit obrázek pomocí tlačítka “Save”.

## 6. Závěr

Záměrem této práce bylo prozkoumat možnosti řízení a sběru dat ze senzorů CCD s použitím procesor STM32F3 a zvážit využití modulu STM32F429Disco s displejem LCD a pamětí SDRAM v režimu skener.

Výsledkem této práce je realizace funkčního prototypu řádkové kamery se senzorem CCD a mikrořadičem STM32F3 na kontaktním poli. Také jsem na kontaktním poli realizoval funkční prototyp řádkové kamery s modulem STM32F429Disco. Zařízení umí pracovat v řádkovém režimu kdy zobrazuje obrazová data ve formě grafu na LCD displej.

Prototyp také podporuje režim skener, kdy rychle přečte několik po sobě jdoucích snímků. Po přečtení snímků jsou data zpracována do černobílého obrázku a zobrazena na LCD displeji. Pomocí PC aplikace je možné tento obrázek stáhnout do počítače přes sběrnici USB.

Dále jsem navrhl jednoduchou konstrukci prototypu přenosného zařízení, které by mohlo sloužit jako jednoduchá demonstrace možností řádkové CCD kamery.

Pro zařízení sloužící pouze pro sběr dat (bez vestavěného displeje) bych spíše zvolil řadič STM32F3, který je jednodušší na ovládání a zároveň je schopen zpracovávat výstupní signál senzoru. Mikrořadiče řady STM32F4 jsou pak vhodnější pro samostatné vestavěné systémy, případně pro aplikace v režimu skener, neboť umožňují připojit externí paměť.

Nakonec jsem vytvořil jednoduchou aplikaci umožňující číst a zobrazovat změřená data ze senzoru, případně stáhnout a uložit naskenovaný obrázek. Aplikace také umožňuje nastavit dobu expozice a zapnout/vypnout režim sample-hold senzoru CCD.

Možným rozšířením prototypu s modulem STM32F429Disco, by bylo využití dotykového displeje k nastavení doby expozice či přepínání režimu sample-hold. Případně přenos naskenovaného obrázku pomocí funkce “mass storage” přes USB.

## Příloha A.

### Přiložené soubory na disku CD

Přiložené CD obsahuje tento soubor s tímto textem. CD se skládá ze složek:

**Application** obsahuje zdrojové kódy aplikace pro zobrazení dat na PC

**Application\_Dist** obsahuje spustitelný kód aplikace pro zobrazení dat na PC pro platformu Windows

**Datasheets** obsahuje katalogové list součástek a referenční manuály mikrořadičů

**Projekt\_F3** obsahuje zdrojové kódy firmwaru pro mikrořadič STM32F3

**Projekt\_F4** obsahuje zdrojové kódy firmwaru pro mikrořadič STM32F407

**Projekt\_F5** obsahuje zdrojové kódy firmwaru pro modul STM32F429Disco

## Literatura

- [1] Martin Zoubek. *Bakalářská práce. Optoelektronický senzor s řadičem EZ-USB*. 2009.
- [2] KODAK COMPANY. *Kodak CCD Primer. CHARGE-COUPLED DEVICE (CCD) IMAGE SENSORS*. 2012. URL: <http://www.kodak.com/US/en/digital/pdf/ccdPrimerPart2.pdf> (cit. 27. 04. 2014).
- [3] Sony Corporation. *ILX551A datasheet*.
- [4] STMicroelectronics. *RM0316 Reference manual. STM32F302xx, STM32F303xx and STM32F313xx advanced ARM-based 32-bit MCUs*. 2013.
- [5] *Universal serial bus*. URL: <http://www.usb.org/> (cit. 27. 04. 2014).
- [6] *STMicroelectronics*. URL: <http://www.st.com/> (cit. 27. 04. 2014).
- [7] STMicroelectronics. *RM0090 Reference manual. STM32F405xx/07xx, STM32F415xx/17xx, STM32F442xxx and STM32F43xxx advanced ARM-based 32-bit MCUs*. 2014.