



ČESKÉ  
VYSOKÉ  
UČENÍ  
TECHNICKÉ  
V PRAZE

**Fakulta elektrotechnická**  
**Katedra měření**

**Diplomová práce**

# **Experimentální kamera CMOS**

**Bc. Petr Douša**

**Květen 2017**

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





## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Petr Douša**

Studijní program: **Kybernetika a robotika**  
Obor: **Senzory a přístrojová technika**

Název tématu česky: **Experimentální kamera CMOS**

Název tématu anglicky: **Experimental CMOS Camera**

### Pokyny pro vypracování:

Navrhněte a realizujte experimentální kameru CMOS, která bude sloužit pro laboratorní experimenty a testování vlastností různých obrazových senzorů CMOS. Pro přenos obrazu do nadřazeného PC v reálném čase využijte hotovou desku s řadičem firmy Cypress s rozhraním USB ve verzi „High speed“. K této desce připojte různé varianty senzorů CMOS, které se budou testovat. Vytvořte aplikaci pro PC, která zajistí všechny funkce související s ovládáním senzorů i funkce související s přenosem dat do PC a jejich následným zobrazením v potřebné formě. S využitím této kamery sestavte pracoviště pro laboratorní experimenty studentů.

### Seznam odborné literatury:

- [1] Fischer, J.: Optoelektronické senzory a videometrie. Skripta ČVUT – FEL, Praha 2008
- [2] Cypress Semiconductor: EZ-USB technical reference manual, 2014

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

Datum zadání diplomové práce: 24. ledna 2017

Platnost zadání do<sup>1</sup>: 30. září 2018

Prof. Ing. Jan Holub, Ph.D.  
vedoucí katedry



Prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 24. 1. 2017

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





## **Čestné prohlášení autora práce**

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

V Praze dne: 26.05.2017

Podpis autora práce: .....





## Poděkování

Rád bych poděkoval všem, kteří mi pomohli k dokončení této diplomové práce, zejména pak vedoucímu mé diplomové práce doc. Ing. Janu Fische- rovi, CSc., který mě seznámil s problematikou a měl cenné připomínky a rady při řešení problémů souvisejících s diplomovou prací. Dále bych rád poděkoval rodině a blízkým za podporu a důvěru po celou dobu mého studia.





## Abstrakt

Tato diplomová práce se zabývá vyhotovením měřícího pracoviště pro laboratorní experimenty. Jedná o měřící pracoviště, které testuje obrazové snímače CMOS. Měřící stanoviště obsahuje počítač, osvětlovací zařízení, propojovací desku a senzory CMOS. Je zde šest různých druhů CMOS senzorů různých vlastností a rozlišení, nastavitelné pomocí sběrnice I<sup>2</sup>C. Propojovací deska od firmy Cypress poskytuje propojení senzorů s počítačem prostřednictvím USB.

CMOS senzory jsou osvětleny přes osvětlovací zařízení, jehož data budou přes propojovací desku Cypress přenášena do počítače, kde budou vyhodnocena a aplikace zpětně nastaví osvětlovací zařízení. Počítač je hlavním ovládacím prvkem této soustavy, propojovací deska slouží jako buffer osvětlovacího zařízení a pro osvětlení senzoru je použito osvětlovací zařízení.

Aplikace umožní měření šumu, linearity, citlivosti a zobrazení dalších parametrů a vlastností pro všechny použité CMOS senzory. Práce bude využívána studenty na Fakultě elektrotechnické, kteří se tak budou moci seznámit s CMOS senzory, změřit si jejich základní parametry, vyzkoušet různá jejich nastavení.

Klíčová slova: aplikace, CMOS senzory, kamera, řadič Cypress, KODAK, Micron, Aptina.

## Abstract

This thesis proposes making of measuring workplace for laboratory experiments. This workplace is focusing on testing CMOS image sensors. Workplace consists of computer, lighting device, controller and CMOS sensors. There is at least six different CMOS sensors of different properties and resolutions which are adjustable by I<sup>2</sup>C bus. Cypress controller manages connection between sensors and computer using USB.

CMOS sensors are lighted by lighting device and its data are send into computer through Cypress controller, there are evaluated and application then sets lighting device accordingly. Computer is primary controlling unit of workplace, controller is used as buffer of lighting device and for sensor illumination is used lighting device.

Application measures noise, linearity, sensitivity and display of other parameters and properties of all used CMOS sensors. Results of this thesis will be used by students at Faculty of Electrical Engineering which can use it to familiarize with CMOS sensors, measure its basic parameters, try its different settings.

Key words: application, CMOS sensors, video camera, Cypress controller, KODAK, Micron, Aptina.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Rozbor práce</b>	<b>3</b>
2.1	Metodika měření . . . . .	3
2.1.1	Popis světla . . . . .	3
2.1.2	Přenos světla . . . . .	3
2.1.3	Měření světla . . . . .	4
2.1.4	Světelný šum . . . . .	4
2.1.5	Popis linearity . . . . .	5
2.1.6	Popis citlivosti . . . . .	5
2.2	Měřící stanoviště . . . . .	5
2.2.1	Základní měření CMOS senzorů . . . . .	5
<b>3</b>	<b>Využitý hardware</b>	<b>7</b>
3.1	Použité CMOS senzory a jejich vlastnosti . . . . .	7
3.1.1	Obrazové vlastnosti CMOS senzorů . . . . .	8
3.1.1.1	Vlastnosti senzoru MT9M001 . . . . .	8
3.1.1.2	Vlastnosti senzoru MT9T031 . . . . .	8
3.1.1.3	Vlastnosti senzoru MT9V032 . . . . .	8
3.1.1.4	Vlastnosti senzoru MT9V034 . . . . .	8
3.1.1.5	Vlastnosti senzoru KAC-9638 . . . . .	9
3.1.1.6	Vlastnosti senzoru KAC-9618 . . . . .	9
3.1.2	Komunikační vlastnosti CMOS senzorů . . . . .	9
3.1.2.1	Komunikační vlastnosti senzorů MT9M001, MT9T031, MT9V032 a MT9V034 . . . . .	9
3.1.2.2	Komunikační vlastnosti senzorů KAC-9618 . . . . .	10
3.1.2.3	Komunikační vlastnosti senzorů KAC-9638 . . . . .	11
3.2	Komunikační rozhraní . . . . .	11
3.2.1	Základní vlastnosti I <sup>2</sup> C kompatibilní sériové komu- nikace . . . . .	11
3.2.2	Základní vlastnosti USB . . . . .	13
3.2.3	Popis paralelního rozhraní pro kamery . . . . .	14
3.3	Popis ovládací a komunikační desky Cypress . . . . .	15
3.3.1	Připojení komunikační desky Cypress . . . . .	15
3.4	Osvětlovač CMOS senzorů . . . . .	17
<b>4</b>	<b>Návrh experimentální sestavy pro kameru CMOS</b>	<b>19</b>
4.1	Programové vybavení pro měřící aplikaci . . . . .	20
4.1.1	Ovladače a knihovny pro USB . . . . .	20

4.1.2	Popis QT frameworku . . . . .	21
4.1.3	Použité kompilátory pro QT . . . . .	23
4.2	Program pro komunikační desku Cypress . . . . .	24
4.2.1	Nastavení komunikační desky Cypress . . . . .	24
4.2.2	Popis komunikace I <sup>2</sup> C na komunikační desce Cypress . . . . .	26
4.2.3	Implementace paralelního rozhraní pro kamery v desce Cypress . . . . .	27
4.3	Programová implementace USB komunikace . . . . .	29
4.3.1	Knihovna LibUsb a její použití . . . . .	30
4.3.2	Knihovna CyAPI a její použití . . . . .	31
4.3.3	Příkazy pro ovládání Cypress desky . . . . .	31
4.4	Počítačový program pro ovládání měření . . . . .	33
4.4.1	Připojení komunikační desky Cypress . . . . .	34
4.4.2	Implementace měření v počítačovém programu . . . . .	36
<b>5</b>	<b>Měřící pracoviště</b>	<b>39</b>
5.1	Uspořádání pracoviště a prvotní spuštění programu . . . . .	39
5.1.1	Spuštění aplikace a připojení CMOS senzoru . . . . .	41
5.1.2	Měření základních parametrů a změna registru . . . . .	42
5.1.2.1	Měření šumu a linearity . . . . .	43
5.2	Naměřené grafy a výsledky z měření . . . . .	44
<b>6</b>	<b>Závěr</b>	<b>49</b>
	<b>Literatura</b>	<b>51</b>

## Seznam obrázků

1	Časový diagram přenosu obrazových dat pro senzory MT9M0xx. Převzato z: (Aptina 2004) . . . . .	10
2	Komunikace po sběrnici I <sup>2</sup> C . . . . .	12
3	Čtení 8 a 16 bitů dat po sběrnici I <sup>2</sup> C . . . . .	13
4	Paralelní komunikace pro kamery . . . . .	14
5	Schéma ovládací desky Cypress převzato z diplomové práce Bc. Martina Zoubka (B. M. Zoubek 2011a) . . . . .	16
6	Sestavení měřicího pracoviště . . . . .	19
7	Nastavení FIFO paměti, převzato z Cypress datasheet (Cypress 2013) . . . . .	25
8	Srovnání čtení jednoho a dvou bajtů dat po protokolu I <sup>2</sup> C	26
9	Srovnání zápisu jednoho a dvou bajtů dat přes protokol I <sup>2</sup> C	27
10	Rozpoznání připojeného senzoru podle adresy . . . . .	27
11	Paralelní komunikace pro získání obrazových dat z CMOS senzoru . . . . .	28
12	Vývojový diagram pro propojovací desku Cypress . . . . .	29
13	Vývojový diagram pro připojení desky Cypress k počítači .	36
14	Vývojový diagram pro počítačový program . . . . .	37
15	Správce zařízení neznámé zařízení . . . . .	40
16	Instalace ovladačů . . . . .	41
17	Hlavní okno aplikace . . . . .	42
18	Výčet registrů . . . . .	42
19	Měření závislostí . . . . .	43
20	Postupné nastavení měřících parametrů . . . . .	44
21	Měření jasů s výběrem linky na senzoru KAC-9618 . . . . .	45
22	Měření šumu jednoho bodu na senzoru MT9V034 . . . . .	45
23	Postupné zvyšování jasů jednoho bodu na senzoru KAC-9618	46
24	Postupné zvyšování jasů jednoho bodu na senzoru MT9V034	47



# Kapitola 1

## Úvod

Tato práce se zabývá vývojem a realizací měřicího pracoviště pro laboratorní experimenty v oblasti videometrie. Vyvíjené měřicí pracoviště se zaměřuje na testování obrazových snímačů CMOS a zpravidla jej tvoří samotný senzor CMOS, elektronická komunikační deska a osobní počítače s příslušným softwarem.

Senzory CMOS poskytne vedoucí diplomové práce doc. Ing. Jan Fischer, CSc a jejich výběr není obsahem této práce. Spolupráci mezi senzorem a počítačem zajišťuje již existující komunikační deska s řadičem Cypress disponující integrovaným USB (Universal Serial Bus) rozhraním. Vyvíjený řídicí a vyhodnocovací software bude spuštěn na počítači s operačním systémem od společnosti Microsoft. Tento program bude použit přímo k laboratorním experimentům. Další součástí měřicího pracoviště je osvětlovací jednotka pro osvětlení CMOS senzorů, která bude také řízená vyvíjeným počítačovým programem.

V současné době existují v laboratoři již hotové přípravky na měření, ale neposkytují kompletní, unifikované a dostatečně pokročilé řešení pro potřeby laboratoře videometrie. Vyvíjená aplikace umožní měření šumu, linearity a zobrazení dalších parametrů a vlastností všech připojovaných CMOS senzorů. Program sám rozpozná daný senzor a umožní uživatelsky jednotnou práci se všemi podporovanými senzory. Zmíněné existující přípravky neumožňují výměnu zobrazovacích čipů v požadované míře (jednotlivé senzory jsou ovládány odlišnými programy různých autorů).

Tato práce navazuje na bakalářské a diplomové práce, které byly v minulosti vytvořeny na Katedře měření a využívaly shodně použitý řadič Cypress. Všechny zmíněné práce se zabývají připojením obrazových senzorů k počítači. Bakalářská práce pana Zoubka (M. Zoubek 2009) se zabývá připojením řádkové kamery, na svou práci pak navázal diplomovou prací na téma Určování polohy pomocí řádkových obrazů (B. M. Zoubek 2011b). Pan Zima se ve své práci (Zima 2009) zabýval zrychlením přenosu obrazových dat ze senzoru do počítače. Poslední práce, která navázala na toto téma, byla bakalářská práce pana Ondříčky (Ondříčka 2014), který se snažil použít sestavení CMOS senzoru a řadiče Cypress v Raspberry PI. Zkušenosti s řadičem Cypress se využijí a dále rozšíří v této práci. Poskytnuté zdrojové kódy k vybraným pracím umožní urychlit nahlédnutí do této problematiky a zkrátit dobu vývoje.

Cílem této práce je vyvinout a sestavit plnohodnotné měřicí pracoviště, které bude využívat minimálně 6 druhů CMOS senzorů. Tyto CMOS senzory budou poskytnuty vedoucím práce. Senzory jsou připojeny k řadiči Cypress CY7C68013A. Jedná se o propojovací desku vyvinutou na Katedře měření, která poskytuje propojení CMOS senzorů s počítačem prostřednictvím USB.

Experimentální kamera CMOS bude využívána studenty na Fakultě elektrotechnické a

## 1.0 ÚVOD

budou se moci seznámit s CMOS senzory, změřit si jejich základní parametry, vyzkoušet různá jejich nastavení. Po ukončení měření budou moci mít uložené grafy včetně dat, z kterých tyto data byla naměřena.



## Kapitola 2

# Rozbor práce

Začátek této práce je věnován teoretickému úvodu problematiky, zejména pak nastíněním problematiky měření. Toto téma obsahuje teorii světla, přenosu světla a jeho měření, a také teorii světelného šumu. Základy této teorie jsou nezbytné pro pochopení měření senzorů z hlediska linearit a šumu. Následně se práce zabývá rozboru měřicího pracoviště a všech jeho složek.

### 2.1 Metodika měření

Při měření CMOS senzorů je důležité světelné záření dopadající na senzor. Senzor takové záření převede na elektrický signál, který je dále zpracován AD převodníky a vyhodnocen. Z tohoto důvodu je podstatné rozumět problematice světelného záření a jeho aspektům.

#### 2.1.1 Popis světla

Světlo je z fyzikálního hlediska viditelná část elektromagnetického záření, tedy elektromagnetické vlnění v rozsahu přibližně od 390–760 nm. Tyto vlnové délky odpovídají spektru mezi infračerveným zářením, viditelným světlem a ultrafialovým zářením. Touto částí elektromagnetického záření se zabývá optika. Některé oblasti techniky zahrnují i oblast infračerveného a ultrafialového záření.

V zásadě rozlišujeme tři typy vzniku viditelného záření:

- záření vyvolané vysokou teplotou emitujícího povrchu
- záření elektrického výboje v párách kovů a v plynech
- luminiscence pevných látek

#### 2.1.2 Přenos světla

K přenosu světla dochází pomocí elementárních částic – fotonů. Z našeho pohledu je foton částicí způsobující elektromagnetickou interakci, zahrnuje elektromagnetické spektrum od radiových vln po gama záření, které je kvantováno na fotony a popsáno vlnovou délkou, frekvencí, energií a hybností. Celá tato oblast spadá do kvantové elektrodynamiky.

## 2.1 METODIKA MĚŘENÍ

### ■ 2.1.3 Měření světla

Mezi základní měření světla patří Fotometrie a Kalorimetrie. Fotometrie popisuje a řeší měření intenzity světla. Patří sem jas, světelný tok, svítivost a intenzita osvětlení.

a) Jas definujeme jako měřenou veličinu svítivosti. Jednotkou dle (SI) je nit (nt) – candela na čtvereční metr (cd/m<sup>2</sup>)

$$L = \frac{dI}{dS \cos \alpha}$$

b) Světelný tok  $\theta_\lambda$  se vztahuje k přenosu světla prostorem, vyjadřuje intenzitu zrakového vjemu normálního oka, vyvolaného energií světelného záření, které projde za jednotku času určitou plochou v prostoru, kterým se světlo šíří. Jednotkou světelného toku je (lumen). Definiční vztah záleží na tom, zda se jedná o světelný tok monochromatického záření s jednou vlnovou délkou, více vlnovými délkami, nebo zda se jedná o světelný tok představující spojité spektrum či jeho část, a zda posuzujeme i účinnost monochromatického zařízení.

c) Svítivost  $I$  bodového zdroje v daném směru lze definovat jako podíl světelného toku vyzářeného zdrojem v tomto směru do malého prostorového úhlu a velikosti tohoto prostorového úhlu:

$$I = \frac{\delta\Phi}{\delta\Omega}$$

Základní jednotkou je Kandela (cd).

d) Intenzita osvětlení  $E$  je definována jako světelný tok, který dopadá na jednotku plochy. Jednotkou intenzity osvětlení je lux (lx) a platí pro ní obecný vztah:

$$E = \frac{d\Phi}{dA}$$

Kolorimetrie se zabývá se barevným projevem vnímání světla. Do kolorimetrie patří koherence a polarizace.

a) Koherence je fázová shoda elektromagnetických vln ve světelném svazku.

b) Polarizace světla je příčné elektromagnetické vlnění, v němž vektor  $E$  kmitá k rovině kolmé na směr, kterým se vlnění šíří. Přirozené světlo není polarizované, pouze světlo, v němž vektor kmitá pouze v jedné rovině, je lineárně polarizované. Základními způsoby, jak dosáhnout polarizace nepolarizovaného světla, je polarizace odrazem a lomem, dvojlomem nebo pomocí polaroidů (polaroidy – speciální filtry sloužící k polarizaci světla).

### ■ 2.1.4 Světelný šum

Světelný šum znamená světelné znečištění jak z pohledu astronomického, tak z pohledu technického. Aby šum neovlivnil kvalitu obrazu, musí být poměr signálu k šumu dostatečně velký. Příkladem může být dopad fotonů na snímací prvek. Dopadem fotonu se vyzáří energie a světlocitlivá buňka vygeneruje velký elektrický signál. Elektrický signál může být ovlivněn rovněž jiným elektromagnetickým zářením (rádiové vlny, mikrovlny) nebo polovodičovými součástkami. Šum může vzniknout rovněž nerovnoměrností struktury, tepelnými kmity a podobně.

### ■ 2.1.5 Popis linearity

Linearita je vlastnost, kdy je výstupní veličina úměrná vstupní veličině, tedy citlivost je konstantní v určitém rozsahu vstupní veličiny. Například fotoelektrický snímač Fotometru je lineární jen v určitém rozsahu vstupní veličiny. Chyba linearity (nelinearity) je určena vztahem:

$$f = \frac{Y}{Y_{max}} \cdot \frac{X_{max}}{X} - 1$$

kde X je vstupní veličina, Y výstupní veličina při vstupní hodnotě X, X<sub>max</sub> vstupní hodnota při hodnotě Y<sub>max</sub>, Y<sub>max</sub> výstupní hodnota na horní hranici měřeného rozsahu. Chyba f, je obvykle maximální hodnota v rámci celého měřicího rozsahu.

### ■ 2.1.6 Popis citlivosti

Citlivost lidského oka nebo elektronického snímače na světlo je pojem, který vysvětluje, kolik světla musí dopadnout na snímač, aby byl schopný zaznamenat nějakou změnu. Pouze v případě, že bychom docílili absolutní tmy, by byla citlivost nulová, tak tomu ovšem v praxi není. Aby snímače i v podmínkách velmi slabého světla fungovaly, máme dvě možnosti, a to prodloužit expoziční dobu nebo zvýšit citlivost snímače. Prodloužení expozice znamená, že předměty se nesmí hýbat, jinak dochází k rozostření obrazu. Je-li citlivost snímače malá, zaznamená jen náznaky obrazu. Pokud je ještě menší, snímač zaznamená pouze šum. Naopak, je-li světla příliš a citlivost snímače je velká, výsledek bude podobný jako u nízké citlivosti. Pouze při správně zvolené citlivosti vzhledem k osvětlení bude snímač fungovat správně.

## ■ 2.2 Měřící stanoviště

Měřící stanoviště bude obsahovat CMOS senzory, propojovací desky, osvětlovací zařízení a počítač. CMOS sensor bude osvětlen přes osvětlovací zařízení, jehož data budou přes propojovací desku Cypress přenášena do počítače, kde budou vyhodnocena a aplikace zpětně nastaví osvětlovací zařízení. Počítač bude hlavním ovládacím prvkem této soustavy, propojovací deska bude sloužit jako buffer zařízení s kompatibilitou s CMOS senzory a pro osvětlení senzoru je použito osvětlovací zařízení. Toto pracoviště bude zobrazovat naměřená data, dále poskytne možnost ukládání zobrazených dat a měřit zejména linearitu a šum při různých nastaveních senzoru.

### ■ 2.2.1 Základní měření CMOS senzorů

Podstatnou částí měření CMOS senzorů je měření převodní charakteristiky, citlivosti a šumu. Převodní charakteristika je dále v práci zmiňována jako linearita. Měření linearity je závislost vstupní veličiny na výstupní obrazová data. Abychom docílili postupného

## 2.2 MĚŘÍCÍ STANOVIŠTĚ

zvyšování jasů v okolí CMOS senzoru, je vhodné využití osvětlovacího zařízení s počítačem kontrolovaným výstupem. Takový osvětlovač musí být co nejpřesnější, přesně definovaný

Citlivost senzoru je funkcí vlnové délky dopadajícího záření a vlastností senzoru. Obvykle je vyjádřena jako volt, případně ampér, na watt dopadajícího toku. Tato hodnota se měří jako elektrický výstup za optickým výstupem.

Původce šumu v CMOS senzorech může být například šum temného proudu, fotonový šum, kvantitativní šum a další. Šum temného proudu je zapříčiněn pohybem nábojů a atomů v elektronických vodičích. Fotonový šum vzniká přirozeným procesem náhodnou fluktuací toku dopadajících fotonů. Kvantitativní šum je pak nejvíce způsobován AD převodníky. Měření šumu se docílí porovnáním dostatečného množství obrazových dat, jež byla opatřena ve stejných podmínkách.

Osvětlovací zařízení by se mělo chovat konstantně za různých podmínek, být nastavitelné pomocí počítače, mít lineární charakteristiku a naopak by neměl mít PV modulaci, jež by mohla zkreslovat výsledky. Pro správně osvětlení CMOS senzorů je potřeba použít dostatečné množství jasů.

## Kapitola 3

# Využitý hardware

V této kapitole se věnuji prostředkům, které bude tato práce využívat, jaké budou mít parametry a jaké jsou jejich možnosti. Dále nastíním problematiku měření CMOS senzorů.

Nejprve rozeberu jednotlivé senzory CMOS, jejich parametry, možnosti nastavení, komunikaci a přesun obrazových dat. Je využito sedm různých senzorů s odlišnými parametry. Tyto CMOS senzory budou v této diplomové práci hlavním zkoumaným prvkem.

Použitá komunikační rozhraní budou v této kapitole popsána hlavně z hlediska použitých vlastností. Pro lepší popis těchto komunikačních rozhraní doporučuji podívat se do seznamu použité literatury, kde jsou popsána důkladněji. Použitá komunikační rozhraní jsou USB, I<sup>2</sup>C, paralelní rozhraní pro kamery a UART přes standard RS232 pro osvětlovací zařízení. Jelikož osvětlovací zařízení jen používám, nebudu tu UART rozhraní popisovat.

Na závěr kapitoly popíšu hlavní komunikační desku mezi počítačem a senzory. Tato deska obsahuje čip od firmy Cypress a poskytuje komunikační rozhraní I<sup>2</sup>C, paralelní rozhraní pro kamery a USB. Deska Cypress obsahuje FIFO paměť mezi paralelním rozhraním a USB pro lepší možnosti komunikace. Do této FIFO paměti budou uložena přijímací obrazová data ze senzoru.

### 3.1 Použité CMOS senzory a jejich vlastnosti

Senzory CMOS, které jsou využity v této práci, jsou: KAC-9618, KAC-9638, MT9M001, MT9T031, MT9V032 a MT9V034. Dva CMOS senzory jsou od firmy KODAK, další senzor vyrábí firma Micron a poslední tři senzory dodává firma Aptina. Každý ze senzorů má různé vlastnosti a rozlišení. Rozlišení se pohybuje od 648x488 px do 2048x1536 px s různou snímkovou frekvencí. Senzor MT9T031 je barevný senzor s Bayer maskou, ostatní CMOS senzory jsou černobílé. Registry těchto senzorů jsou 8-bitové, avšak data k těmto registrům jsou buď 8-bitová nebo 16-bitová podle použitého senzoru. Všechny senzory podporují komunikační rozhraní I<sup>2</sup>C a paralelní rozhraní pro kamery. Komunikace I<sup>2</sup>C se využívá pro nastavování parametrů senzoru a jiná nastavení. Paralelní rozhraní je používáno pro příjem obrazových dat CMOS. Senzory mají interní AD převodník i vlastní elektronikou závěrku nastavitelnou pomocí I<sup>2</sup>C. Nastavení CMOS senzoru se provádí zápisem do vnitřních registrů, kde lze nastavit například již zmíněnou závěrku, ROI, integrační čas a další nastavení.

## 3.1 POUŽITÉ CMOS SENZORY A JEJICH VLASTNOSTI

### ■ 3.1.1 Obrazové vlastnosti CMOS senzorů

V této části textu vytyčím hlavní parametry CMOS senzorů a jejich rozdíly. Dalšími zmíněnými parametry jsou závěrka, velikost snímané plochy a jakou firmou byly tyto senzory vyrobeny.

#### ■ 3.1.1.1 Vlastnosti senzoru MT9M001

Senzoru MT9M001 vyrábí firma Micron. Tento senzor má velikost snímané plochy  $35,43 \text{ mm}^2$  v černobílé variantě. Maximální rozlišení tohoto CMOS senzoru je  $1280 \times 1024$  při maximální obnovovací frekvenci 30 snímků za sekundu. Z tohoto rozlišení může vybrat libovolnou menší výseč pomocí nastavení registrů. Další možnost, kterou senzor podporuje, je přeskočení pravidelných řádků nebo sloupců, a to každý druhý, každý čtvrtý a každý osmý. Obsahuje rolling uzávěrku. Používá 10-bitový převodník ADC.

#### ■ 3.1.1.2 Vlastnosti senzoru MT9T031

Senzoru MT9T031 navrhla společnost Aptina. Velikost samotné snímané plochy je  $32,23 \text{ mm}^2$  v barevné variantě. Používá barevný RGB Bayer filtr. Obnovovací frekvence je 12 snímků za sekundu při maximálním rozlišení  $2048 \times 1536$ . Lze použít libovolný výřez a dosáhnout tím vyšší obnovovací frekvence. Dá se například dosáhnout 93 snímků za sekundu při rozlišení  $640 \times 480$ . Podporuje přeskočení určitých řádků a sloupců pomocí nastavení v registru. Obraz se dá otočit pomocí nastavení v registru jak po horizontální, tak i vertikální ose. Obsahuje rolling uzávěrku. Používá 10-bitový převodník ADC.

#### ■ 3.1.1.3 Vlastnosti senzoru MT9V032

Senzor MT9V032 vyrobila firma Aptina. Snímaná aktivní plocha je  $13 \text{ mm}^2$  v černobílé variantě. Maximální rozlišení  $752 \times 480 \text{ px}$  je možné snímat při maximální obnovovací frekvenci 60 snímků za sekundu. Přenášená velikost obrazových dat tohoto CMOS senzoru se může libovolně měnit do maximálního rozlišení. Umožňuje převrácení obrazu a přeskočení řádků i sloupců. Také umožňuje sloučení sousedních pixelů do jednoho a pomocí toho snížit rozlišení při zachování přibližně stejné citlivosti. Obsahuje global uzávěrku. Tato uzávěrka by měla být efektivnější při zpracování pohybujících se objektů před senzorem. Používá 10-bitový převodník ADC.

#### ■ 3.1.1.4 Vlastnosti senzoru MT9V034

Senzoru MT9V034 je velice podobný senzoru MT9V032 s tím rozdílem, že má vyšší citlivost v blízkosti infračerveného záření. Rozlišení, počet snímků za sekundu a změna rozlišení jsou stejné. Použitá elektronická uzávěrka je stejná jak u senzoru MT9V032.

## 3.1 POUŽITÉ CMOS SENZORY A JEJICH VLASTNOSTI

### ■ 3.1.1.5 Vlastnosti senzoru KAC-9638

Senzoru KAC-9638 je černobílý senzor firmy Kodak. Tento senzor má velikost snímané plochy  $47,85 \text{ mm}^2$ . Maximální rozlišení tohoto CMOS senzoru je  $1032 \times 1288$  při maximální obnovovací frekvenci 18 snímků za sekundu. Pomocí nastavení registru je možné nastavit jiné menší rozlišení, převrácení obrazu, přeskokování řádků a sloupců a další parametry. Obsahuje rolling uzavěrku. Používá 10-bitový převodník ADC.

### ■ 3.1.1.6 Vlastnosti senzoru KAC-9618

Poslední senzor, KAC-9618, je také od firmy Kodak a jedná se též o černobílý senzor. Velikost snímané plochy je  $17,8 \text{ mm}^2$  při rozlišení  $648 \times 488$  px. Snímková frekvence při maximálním rozlišení je 30 snímků za sekundu. Senzor umožňuje nastavit horizontální a vertikální dílčí vzorkování, a to 2:1 nebo 4:2. Obsahuje elektronickou rolling uzavěrku. Používá 10-bitový převodník ADC.

## ■ 3.1.2 Komunikační vlastnosti CMOS senzorů

Pro komunikaci se senzory je potřeba ze senzoru získat obrazová data a zajistit nastavení registrů. Komunikaci se senzory zajišťuje paralelní rozhraní pro přenos obrazových dat a I2C slouží pro nastavení registrů nebo jejich vyčtení.

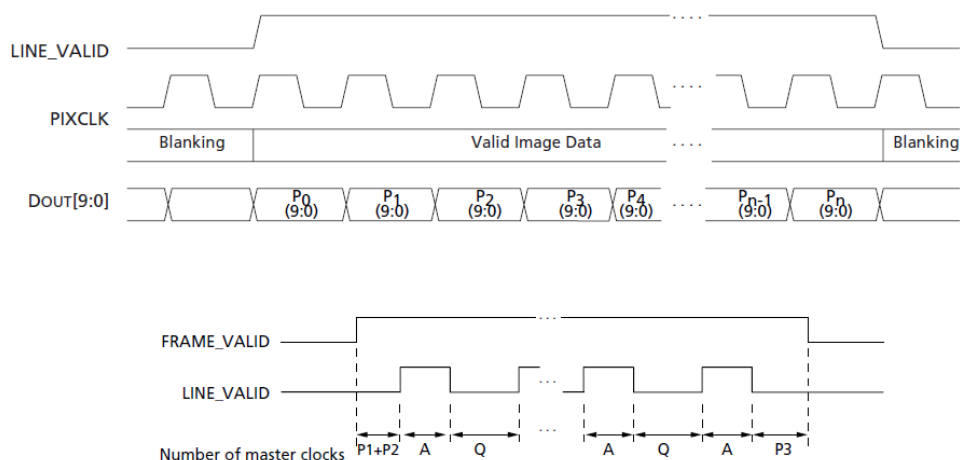
Komunikace přes I<sup>2</sup>C sběrnici je přesně definovaná pomocí specifikace, kterou nyní spravuje NXP Semiconductors. Časování sběrnice I<sup>2</sup>C je přesně popsáno, včetně adresování. Tento postup a postup komunikace na lince popíšu později.

Paralelní rozhraní pro kamery není přesně definované, každý senzor definuje tuto komunikaci jinak. Senzory MT9M001, MT9T031, MT9V032 a MT9V034 mají velice podobné vlastnosti komunikace. Senzory KAC-9618 a KAC-9638 používají mírně odlišné způsoby přesunu obrazových dat, proto budou popsány zvlášť.

### ■ 3.1.2.1 Komunikační vlastnosti senzorů MT9M001, MT9T031, MT9V032 a MT9V034

Senzory MT9M001, MT9T031, MT9V032 a MT9V034 přenášejí obrazová data podle časového diagramu 1 :

### 3.1 POUŽITÉ CMOS SENZORY A JEJICH VLASTNOSTI



Obrázek 1: Časový diagram přenosu obrazových dat pro senzory MT9M0xx. Převzato z: (Aptina 2004)

Jak lze sledovat z časového diagramu, jsou použity signály LINE\_VALID, FRAME\_VALID, PIXCLK a datové vodiče. Signály LINE\_VALID a FRAME\_VALID vyjadřují svoji platnost pomocí napěťové úrovně, kdežto PIXCLK je platný na náběžnou hranu signálu a datové vodiče jsou platné při spádové hraně signálu PIXCLK. Platný snímek probíhá nastavením signálu FRAME\_VALID. Poté následuje signál LINE\_VALID. Signál LINE\_VALID naznačuje právě přijímaný řádek senzoru. Mezi signálem LINE\_VALID a FRAME\_VALID existuje časová prodleva v podobě několika tiků signálu PIXCLK. Na obrázku je to označeno jako P<sub>1</sub>+P<sub>2</sub>. Tato prodleva mezi těmito signály se dá nastavit pomocí změny registrů senzoru. Každý vyčítaný řádek je potvrzován signálem LINE\_VALID. Každý vyčítaný pixel ze senzoru je potvrzovaný signálem PIXCLK. Po dokončení přenosu celého obrazu je shozen signál FRAME\_VALID a tím je naznačen konec přenosu celého obrazu.

#### 3.1.2.2 Komunikační vlastnosti senzorů KAC-9618

Senzor KAC-9618 podporuje paralelní rozhraní v řízeném i řídicím módu. V řízeném módu zařízení přijímá FRAME\_VALID a LINE\_VALID z ovládací desky a není generováno řadičem. V řídicím módu tento senzor generuje signály FRAME\_VALID a LINE\_VALID a PIXCLK. Řídicí mód může být nakonfigurován v level módu nebo v puls módu. V level módu jsou signály FRAME\_VALID a LINE\_VALID generovány, když jsou platná data. Paralelní rozhraní senzoru KAC-9618 jsou defaultně povolena a nastavena do puls módu. V puls módu je senzor nakonfigurován tak, že signály FRAME\_VALID a LINE\_VALID jsou potvrzovány na konci přenosu vytvořením změny napětí pro počet definovaných hodinových signálů.



### ■ 3.1.2.3 Komunikační vlastnosti senzorů KAC-9638

Senzor KAC-9638 má stejné komunikační možnosti jako senzor KAC-9618, ale tento senzor je nastaven defaultně v módu řízeném. Pro nastavení senzoru do módu řídicího je nutné nastavit registry pomocí sběrnice I<sup>2</sup>C.

## ■ 3.2 Komunikační rozhraní

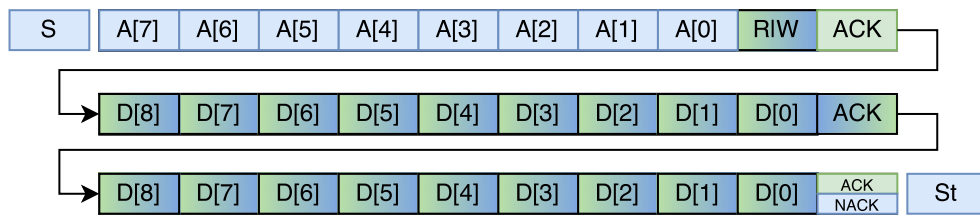
Hlavní komunikační rozhraní je paralelní rozhraní pro kamery a sériová sběrnice I<sup>2</sup>C. Sběrnice I<sup>2</sup>C je použita pro základní nastavení CMOS senzoru a vyčtení informací o tomto senzoru. Po sběrnici I<sup>2</sup>C lze u CMOS senzorů například nastavit ROI, dobu závěrky a mnoho dalších nastavení. Paralelní rozhraní pro kamery je u těchto senzorů CMOS použito pro přenos obrazových dat. Napájení senzorů je realizováno přímo z uvedené desky, která je napájena z USB portu. Deska tedy zajistí napájení a veškerou komunikaci se senzory a postará se o přenesení všech obrazových dat do počítače.

Další použité komunikační rozhraní je USB. Toto rozhraní je použité pro přenos obrazových dat do počítače, pro možnost nastavení CMOS senzoru, zahájení přenosu obrazových dat a pro nastavení přenosu obrazových dat. Přenos obrazových dat po USB musí být dostatečně rychlý. U senzoru s nejvyšším rozlišením (Aptina 2006) je potřeba přenosová rychlost v nejvyšším nastavení senzoru  $V * S * t = 2048 * 1536 * 12 = 37748736 B/s$ . Maximální rychlost USB 1.0 je 1.5 Mbit/s, USB 1.1 je 12 Mbit/s a USB 2.0 je 480 Mbit/s. Rychlost 480 Mbit/s je dostačující pro přenos všech obrazových dat při maximálním nastavení senzoru. Rychlost USB 2.0 je pojmenována jako High Speed. Počítač, na kterém bude spuštěna měřicí aplikace, je obsazen USB 2.0, proto rychlost High Speed bude využita pro přenos dat.

### ■ 3.2.1 Základní vlastnosti I<sup>2</sup>C kompatibilní sériové komunikace

Sériová sběrnice I<sup>2</sup>C je založena na dvoudrátové sběrnici, která je využívána především pro komunikaci s periferiemi. Tato sériová komunikace byla vyvinuta firmou Philips v roce 1982. V dnešní době se tato firma jmenuje NXP Semiconductors. Komunikace probíhá pomocí dvou drátů, kde jeden komunikační signál je hodinový (SCL) a druhý signál je datový (SDA). Komunikace využívá open-drain vodiče, kde jsou signály pomocí odporů přivedeny na logickou úroveň 1. Logická úroveň 0 je generována elektronickými zařízeními, které komunikují po I<sup>2</sup>C. Průběh komunikace můžeme sledovat na následujícím obrázku 2.

### 3.2 KOMUNIKAČNÍ ROZHRANÍ



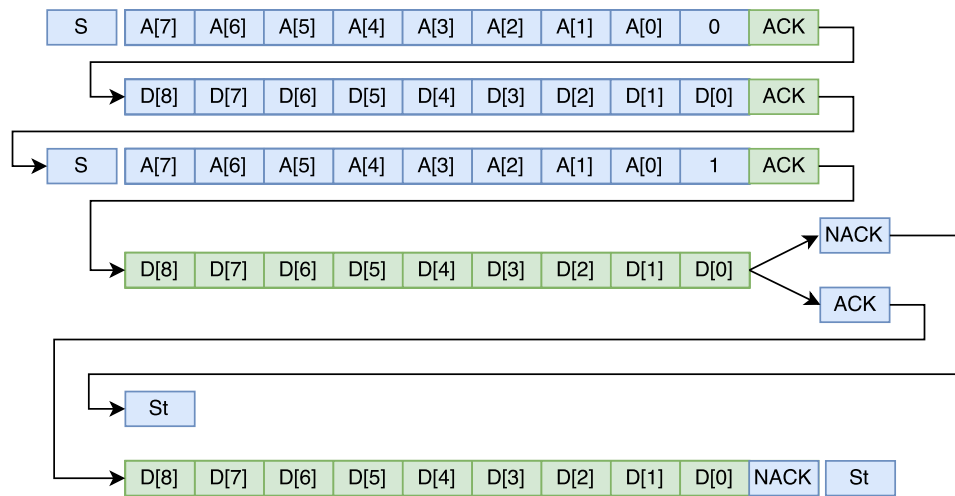
Obrázek 2: Komunikace po sběrnici I<sup>2</sup>C

Jak lze sledovat na obrázku, jedná se o základní komunikaci I<sup>2</sup>C. Komunikace je založena na principu řídicí-řízený (na obrázku znázorněno modrá-zelená), kde řídicí zahájí přenos start bitem (na obárku označeno písmenem S, změna datového signálu při hodinovém signálu v logické úrovni 1 na začátku přenosu). Po tomto start bitu se nachází adresa řízeného zařízení (A0-A7), která obsahuje 7 bitů. Existuje i adresování, které může mít více jak 7 bitů, ale to není použité v této diplomové práci. Poté je poslán bit pro čtení nebo zápis (R respektive W). Devátý bit je potvrzení (ACK), zda existuje řízené zařízení s touto adresou. Pokud není potvrzen tento devátý bit (NACK), žádné řízené zařízení by nemělo vykonávat odpověď na další části tohoto přenosu. V oficiální I<sup>2</sup>C specifikace je napsáno:

*“When SDA remains HIGH during this ninth clock pulse, this is defined as the Not Acknowledge signal. The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer. There are five conditions that lead to the generation of a NACK: 1. No receiver is present on the bus with the transmitted address so there is no device to respond with an acknowledge.”* (NXP 2014)

Tímto způsobem je možné rozpoznávat, zda je zařízení přítomno na sběrnici. Toho lze využít při identifikaci připojeného senzoru a jeho typu. Pokud senzor odpoví na požadovanou adresu, která je definována v datasheetu, je tento senzor rozpoznán.

Další vlastností, kterou využijí, je možnost přečíst 16-bitová data i přes komplikace, kdy senzor poskytuje pouze 8-bitová data. Průběh lze sledovat na následujícím obrázku 3. Jak lze sledovat, až do sekvence přečtení jednoho bajtu je komunikace totožná. Devátý bit, po prvních 8 adresových bitech dat přenosu, je při přenosu pouze osmi bitů dat nastaven na nepotvrzení (NACK) od řízeného a je ukončen přenos. Bity před nepotvrzením jsou stejné, jsou přečteny naprosto stejně jako u přenosu 16 bitů dat. Pro senzory, které nepodporují 16-bitový přenos, je pak nutné ignorovat poslední přenesený bajt dat.

Obrázek 3: Čtení 8 a 16 bitů dat po sběrnici I<sup>2</sup>C

### ■ 3.2.2 Základní vlastnosti USB

USB je zkratka Universal Serial Bus, což v překladu znamená univerzální řadiče sériové sběrnice. Toto komunikační rozhraní umožňuje například připojení myši, klávesnice, kamery, paměťového zařízení a mnoho dalšího k počítači. Specifikace USB byla vydána v roce 1995 a od té doby byla velice rozšířena. V dnešní době se nachází na většině domácích počítačů, včetně počítače, kde bude běžet měřicí úloha navržená v této diplomové práci.

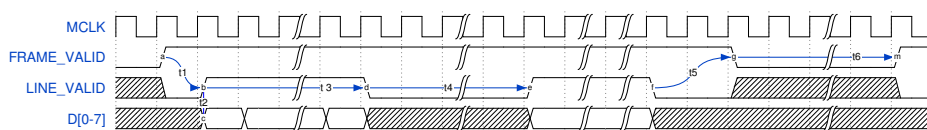
Zařízení USB se po připojení do počítače automaticky identifikuje a nastaví požadované parametry pro napájení i výběr potřebných ovladačů. Některá zařízení USB používají ovladače již integrované v operačním systému. Tyto ovladače se pak použijí s konkrétním USB zařízením. Pokud není ovladač obsažen v systému, je nutné naprogramovat ovladač a ten pak nainstalovat a použít.

Maximální teoretická rychlost USB 2.0 je 480 Mbit/s (60MB/s), což je dostačující rychlost pro přenos obrazových dat a pro nastavení CMOS senzoru. V komunikaci je jedno zařízení řídicím a toto zařízení inicializuje komunikaci a přenos dat. V tomto případě je řídicí zařízení počítač, kde bude běžet měřicí aplikace, a řízené zařízení je komunikační deska Cypress.

K přenosu využijí takzvaného bulk přenosů, který umožňuje spolehlivě přenést data přes rozhraní USB. Bulk přenos je jeden z typů přenosu po endpointech. Endpoint si lze představit jako logickou rouru komunikace, která představuje logickou vazbu dat přenášených po USB. Typ bulk přenos spolehlivě přenáší data po USB a při chybě automaticky opakuje přenos dat, ale nemá garantovanou přenosovou kapacitu linky. Proto při použití bulk přenosů je nutné vypojit ostatní USB zařízení z USB a zajistit aby toto zařízení bylo jediné připojené USB zařízení v počítači.

### 3.2.3 Popis paralelního rozhraní pro kamery

Komunikační paralelní rozhraní pro kamery není přesně specifikováno, proto zde popíšu paralelní rozhraní využívané senzory i samotnou Cypress komunikační deskou. Toto paralelní rozhraní slouží k přenosu obrazových dat, kde komunikační deska Cypress je poskytovatel hodinových impulsů hodin (MCLK) a přijímá signály FRAME\_VALID, LINE\_VALID, PIXCLK a paralelní datové signály D0 až D9. Senzory poskytují i jiné možnosti komunikace, jako je například LVDS, přijímání signálů FRAME\_VALID, LINE\_VALID místo jejich generování, nebo rozdílné nastavení FRAME\_VALID a LINE\_VALID, které nejsou kompatibilní s komunikační deskou Cypress.



Obrázek 4: Paralelní komunikace pro kamery

Všechny signály MCLK, FRAME\_VALID, LINE\_VALID, PIXCLK a datové signály D0 až D9 jsou napětově kompatibilní pro připojení senzoru ke komunikační desce Cypress. Po připojení sensorové desky ke komunikační desce Cypress a po správném nastavení senzoru lze zajistit, že časová komunikace bude prováděna podle obrázku 4. Jediný signál MCLK (Master Clock, hlavní hodiny) vede z komunikační desky Cypress do senzoru. Senzor přijímá signál MCLK a následně pomocí tohoto signálu generuje ostatní signály zpět do řídicí desky Cypress. Signál FRAME\_VALID naznačuje, kdy jsou přenášena obrazová data celého obrazu. Změnou signálu FRAME\_VALID do logické úrovně 1 začíná přenos snímku obrazových dat a ukončuje se přenos snímku, když se tento signál navrátí do logické úrovně 0. Mezi signálem FRAME\_VALID probíhá signál LINE\_VALID. Signál LINE\_VALID naznačuje, kdy je přenášeno validní řádku senzoru. Změnou signálu z logické úrovně 0 do logické úrovně 1 je naznačen přenos řádku obrazových dat. S logickou úrovní 0 signálu LINE\_VALID je ukončen přenos obrazových dat řádku. Mezi signálem LINE\_VALID probíhá signál PIXCLK s datovými signály D0 až D9. Signál PIXCLK svojí změnou úrovně napětí do logické úrovně 1 naznačuje, že na datových vodičích D0 až D9 jsou validní obrazová data pro daný pixel. Tento přenos se opakuje a validní obrazová data jsou spolehlivě přenášena do zařízení.

Signály LINE\_VALID, PIXCLK a datové signály D0 až D9 se mohou měnit i v průběhu, kdy je FRAME\_VALID na logické úrovni 0. V tomto případě nejsou přenášena validní obrazová data, mohou to být například tmavé pixely, které slouží pro vnitřní funkci senzoru CMOS. Jiné obrazové senzory umožňují tato data nepřenášet, signál LINE\_VALID je na logické úrovni 0. Časové prodlevy mezi FRAME\_VALID od logické úrovně 1 k logické úrovni signálu LINE\_VALID do logické úrovně 1 jsou u každého typu senzoru jiné, stejně tak i časové prodlevy mezi FRAME\_VALID a LINE\_VALID v logické úrovni 0. Proto je nutné nepřenášet data, kdy signál FRAME\_VALID je na logické úrovni 0 nebo kdy signál LINE\_VALID je na logické úrovni 0. Pro vyčtení černých pixelů využít nastavení CMOS

senzoru pokud to tento senzor podporuje.

## ■ 3.3 Popis ovládací a komunikační desky Cypress

Jako komunikační deska mezi počítačem a CMOS senzorem byl ze zadání určen čip CY7C68013A od firmy Cypress. Hlavní výhodou tohoto čipu je, že obsahuje sběrnici USB ve formě Hi-Speed. Tento čip obsahuje paralelní rozhraní pro příjem, I<sup>2</sup>C sběrnici, UART a mikroprocesorové jádro 8051.

Paralelní rozhraní je možné využít pro příjem obrazových dat ze senzoru CMOS, sběrnici I<sup>2</sup>C lze využít ke konfiguraci CMOS senzoru. Jádro 8051 pak může sloužit k rozhodování a konfiguraci jednotlivých periférií. Nahrávání programu pro jádro 8051 je možné z USB nebo z připojené paměti EEPROM.

Vývojovou desku vytvořil ve své diplomové práci Ing. Martin Zoubek. Hlavní součásti této desky jsou USB, řadič Cypress, paralelní rozhraní pro kamery, krystal, dělič frekvence, paměť EEPROM a LED dioda. Deska obsahuje další podpůrné elektronické součástky. Tyto desky jsem dostal v neosazeném tvaru. Pro tuto diplomovou práci jsem připravil 4 kusy, které jsem následně otestoval.

V této komunikační desce je připojena elektrotechnická součástka krystal k čipu Cypress, a proto je možné desku konfigurovat na pracovní frekvenci mikroprocesorového jádra 8051 na hodnotu 48 MHz. Jádro 8051 je navrženo pro zvládnutí za 4 takty jednu instrukci. Čip od firmy Cypress neobsahuje vnitřní paměť flash, ale program je uložen v 16KB paměti RAM, která může být nahrána přes USB rozhraní (Soft Configuration), nebo přes externí připojenou EEPROM paměť. Poté, co je program spuštěn, mohou být nakonfigurovány periférie, včetně periférie USB.

Periférie USB je připojena k 4kB FIFO paměti, která napomáhá k zajištění Hi-Speed USB přenosu. Tato FIFO paměť je připojena k paralelnímu rozhraní, které je dále připojeno k pinům tohoto mikrokontroléru.

Řadič Cypress dále obsahuje sběrnici I<sup>2</sup>C, a to pouze v módu řídicím. Tato sběrnice umožňuje komunikovat s I<sup>2</sup>C senzorem a to umožňuje nastavení CMOS senzoru. Periférie I<sup>2</sup>C umožňuje detekovat potvrzení odpovědi na správnou adresu řízeného zařízení, včetně detekování potvrzení při přenosu dat.

### ■ 3.3.1 Připojení komunikační desky Cypress

Schéma ovládací desky je zobrazeno na obrázku 5, jedná se o schéma vytvořené Ing. Martinem Zoubkem. Jak lze na tomto schématu sledovat, připojení senzoru je konektor pojmenovaný jako HEADER CONNECTOR MT9M001. Na tomto konektoru lze zapojit i ostatní senzory CMOS použité v této práci, dále je na konektoru připojeno napájení 3,3 V zem (GND), datové signály D0 až D7, hlavní hodiny (MCLK), FRAME\_VALID, LINE\_VALID, PIXCLK, I<sup>2</sup>C komunikace SDA a SDO, zapnutí senzoru (OE, open ena-



jsem použil takzvaný druhý kód. V jeho diplomové práci je napsáno ohledně rychlosti tohoto zákazu zápisu do FIFO paměti toto:

*”Druhý způsob je založen na tom, že pokud nějaká adresa neadresuje ve FIFO paměti žádný blok paměti, pak jí lze stejně jako signál SLCS využít k blokování zápisu do FIFO. Právě 2. verze řízení zapisovacího signálu FIFO paměti bude využita. Rychlost reakce interních řídicích obvodu FIFO paměti na změnu adresy je dostatečná.”* (B. M. Zoubek 2011b)

Díky těmto vlastnostem můžu bezproblémově používat tyto signály pro zapnutí a vypnutí FIFO paměti. Tyto vlastnosti spolu s psaním kódu v jazyce symbolických adres umožňuje dosáhnout vysoké rychlosti pro začátek přenosu obrazových dat ze senzoru do řadiče Cypress, kde změna signálu FRAME\_VALID z logické úrovně 0 do logické úrovně 1 musí algoritmus programově zasáhnout a provést požadované operace.

## ■ 3.4 Osvětlovač CMOS senzorů

Osvětlovací zařízení slouží k osvětlování CMOS senzorů, kde LED diody jsou před CMOS senzorem. Tento osvětlovač musí být nastavitelný ve svém osvětlovacím rozsahu. Senzor musí být ovladatelný přes počítač, a to pomocí programu. Tento osvětlovač bude hlavně použit při měření šumu a linearity.

Pro osvětlení CMOS senzorů jsem využil osvětlovací zařízení vyvinuté v mé bakalářské práci. Jeho hlavní předností je použití operačního zesilovače pro měření proudu procházejícím LED. Pomocí tohoto postupu je dosaženo skoro lineární změny nastavení osvětlení, výsledné nastavení osvětlení je stálé, není použita například PWM modulace, která by mohla zkreslit výsledek.

Parametry osvětlovacího zařízení jsou uvedeny v mé bakalářské práci, kde je napsáno:

*”Pro rovnoměrné ozáření CMOS senzoru bylo zvoleno po dohodě s vedoucím bakalářské práce 60 LED diod. Tyto LED diody jsou rozprostřeny na kruhovém základu o poloměru 52 mm. Proud osvětlovače byl zvolen 300 mA. Deska je rozdělena na dvě části, na osvětlovací a řídicí. Řídicí obvod je složen převážně z mikrokontroleru, stabilizátoru, převodníku napětí pro UARTa z nízkošumového Rail-to-Rail operačního zesilovače AD822BN, to pomáhá k nerozkmitání obvodu, které nesmí nastat, aby tento zdroj byl lineární a dále pak k úplnému uzavření tranzistoru. Ovládání tohoto senzoru může být zprostředkováno za pomoci sériového rozhraní UART nebo Virtual-COM port.”* (Douša 2014)

Jak z citace vychází, komunikace s tímto zařízením je pomocí Virtual-COM port. Po připojení USB osvětlovací desky do počítače je tento COM port zprovozněn a jsou načteny informace o ovladačích. Po připojení k tomuto Virtual-COM portu lze pak jednoduše posílat zprávy ve tvaru: Y0000X, kde číslo 0000 je v rozsahu 0000 (minimální osvětlení) až 2000 (maximální osvětlení). Tento osvětlovač je dostatečný pro osvětlení CMOS senzorů a umožní následné měření šumu a linearity.

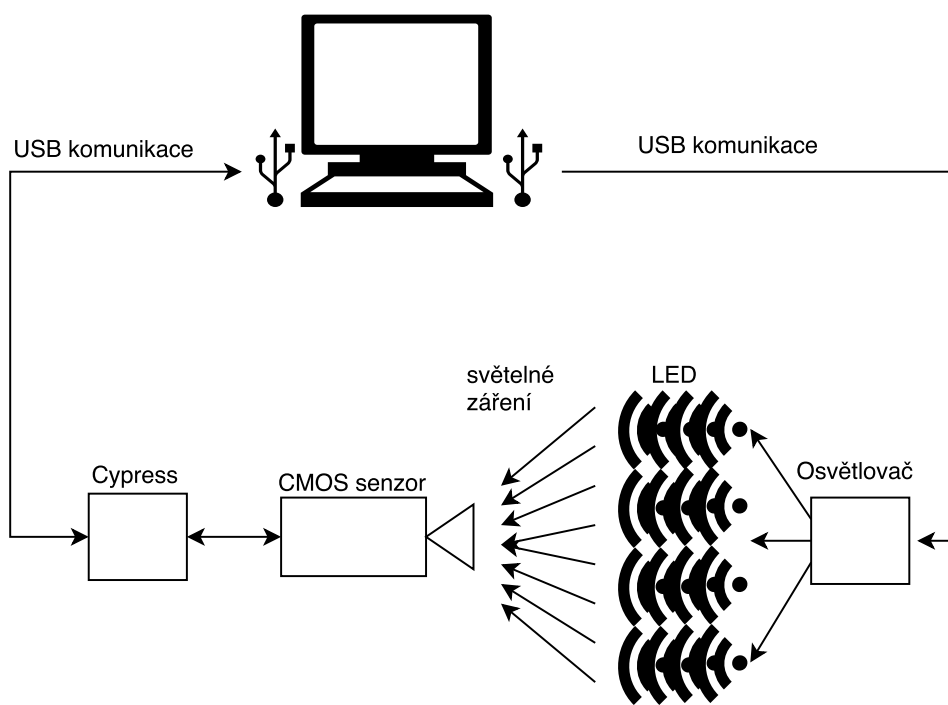




## Kapitola 4

# Návrh experimentální sestavy pro kameru CMOS

Z předešlé kapitoly vyplývá, že pracoviště bude složeno z výměnného senzoru CMOS připojeného k přenosové desce s řadičem od firmy Cypress připojeného k počítači pomocí USB. Na počítači bude běžet aplikace, která bude tento senzor ovládat a vyhodnocovat přijaté informace. K počítači bude nadále připojeno osvětlovací zařízení, které bude počítačovou aplikací ovládáno. Toto sestavení je zobrazeno na obrázku 6.



Obrázek 6: Sestavení měřícího pracoviště

V této části textu rozeberu hlavně program pro PC a pro komunikační desku Cypress. Dále popíšu důležité ukázky zdrojového kódu a to zejména pro přenos obrazových dat. Dále ukážu vývojové diagramy jak jsem zhotovil program pro komunikační desku Cypress a pro počítač. Nakonec nastíním problematiku instalování ovladačů.

### ■ 4.1 Programové vybavení pro měřící aplikaci

Z teorie vychází potřeba vytvoření programu pro řadič Cypress a programu pro PC. Pro zhotovení měřícího pracoviště je potřeba programovat ve více programových jazycích. Řídicí deska s řadičem Cypress je programována v jazyce symbolických odkazů a C. Programuje se ve vývojovém prostředí Keil pro mikrokontrolery kompatibilní s 8051. Vývojové prostředí Keil je podporováno firmou Cypress pro řadiče FX2. Je možné využít vývojového programu Visual studia od Microsoftu. Diplomové a bakalářské práce, na které navazují, tuto možnost nevyužily.

Programové vybavení pro ovládací program pro PC není ze zadání určeno. Program pro PC musí být kompatibilní s Windows 7, musí rychle zpracovávat veškeré přijaté informace a být kompatibilní s ovladači CyAPI, LibUSB a UART. Pro mé znalosti ovladače LibUSB a možnosti debugu v linuxovém prostředí využiji tohoto ovladače na začátku programování. Využiji k tomu bakalářskou práci pana Ondříčky, která využívá ovladač LibUSB pod Linuxem. Tento ovladač je kompatibilní s vývojovými jazyky Java, C, C++, C# a překladači gcc i Visual studio. Ovladač CyAPI je kompatibilní pouze s Visual studio překladačem a Borland Developer Studiem. Prostředí musí splňovat podmínku zkompilevanosti pomocí Visual studio překladačem. Ovladač i knihovna CyAPI byly využívány v předešlých diplomových i bakalářských pracích, na které navazují.

Vybral jsem vývojové prostředí QT Creator s QT framework. Toto prostředí umožňuje programovat grafickou aplikaci pro PC kompatibilní s operačními systémy Windows. Dále umožňuje jak využití překladače gcc, tak je možné využít překladače Visual studio. Také je kompatibilní s LibUSB a i s CyAPI. V QT frameworku je obsažena knihovna pro komunikaci s UART. Rychlost odezvy programů vytvořené v tomto QT frameworku je na vysoké úrovni, narozdíl od Javy. Framework QT dále poskytuje mechanismy pro snadné zacházení s vlákny, což je výhoda oproti C#.

#### ■ 4.1.1 Ovladače a knihovny pro USB

Pro USB komunikaci je možné využít LibUSB, které použil pan Matěj Ondříčka ve své diplomové práci (Ondříčka 2014). Knihovna CyAPI, která je dodávána s čipem Cypress, byla využita v bakalářských i diplomových pracích na katedře Měření pro účely Videometrie. Knihovny jsou potřeba pro ovládání USB zařízení a příjem dat po USB, zejména pro příjem dat z určitého endpointu a odesílání dat do zařízení.

Knihovna LibUSB je opensourcová knihovna pro ovládání zařízení USB. Tato knihovna se vyznačuje dobrou kompatibilitou s operačním systémem Linux a umožňuje použití této knihovny v systému Windows. V systému Windows je nutná instalace ovladačů, které nejsou digitálně podepsány, proto je instalace ovladačů mírně obtížnější. Knihovna obsahuje příkazy pro čtení a zápis dat do endpointu a vyčtení základních informací. Výhodou jsou velmi podrobné výpisy při připojování zařízení a vyčítání nebo zapisování dat.

Knihovna CyAPI je vyvinuta firmou Cypress. Obsahuje již digitálně podepsané ovladače. Tato knihovna poskytuje funkce pro připojení zařízení, vyčítání a zápis dat do endpointu a zobrazení informací o zařízení. Poskytuje všechny potřebné metody pro příjem a vysílání

dat k účelům přijímání obrazových dat a nastavení senzoru CMOS. Další výhodou je plná kompatibilita s čipem Cypress.

### ■ 4.1.2 Popis QT frameworku

Framework QT je neustále vyvíjeno firmou The Qt Company, která zajišťuje jak QT framework, tak i vývojové prostředí QT Creator. QT framework je licencován pod GPL, LGPL nebo komerční verzí. Tyto licence by měly být dostatečné pro účely této práce. Další výhodou tohoto prostředí je velice dobrá dokumentace na stránkách QT (Nord a Chambe-Eng 2017a). Instalaci QT provede program, který lze najít na stránkách QT. Program se postará o instalaci QT frameworku i potřebného kompilátoru i debugovacího programu pro Windows. Na ostatních platformách doporučuji pročíst dokumentaci (Nord a Chambe-Eng 2017b) a postupovat podle doporučeného postupu pro danou platformu.

Prostředí QT je kompatibilní s operačními systémy Windows, Linux, MacOS a dalšími. Pro kompilaci lze využít kompilátor gcc, Visual studio a další. Kompilátory jsou automaticky rozpoznány aplikací QT Creator a jsou následně použity. Program lze proto psát v jazyce C++. Při instalaci QT Creator pro Windows lze vybrat možnost instalace MinGW, která se postará o kompilaci projektu pomocí gcc.

Při vytvoření projektu vznikne soubor s příponou pro, ve kterém jsou uloženy hlavní vlastnosti projektu. V tomto souboru doporučuji přidat řádek pro použití nejnovějšího standardu C++, a to přidáním řádku:

```
CONFIG += c++11
```

Pro přidání již zkompileovaných knihoven do projektu je nutné přidat tyto soubory do souboru s příponou pro. Například pro přidání statických knihoven LibUSB zkompileovaných pro MinGW (Minimalist GNU for Windows, programovací vybavení pro Windows) je nutné přidat tyto soubory pomocí:

```
win32: LIBS += -L$$PWD/libUSB/MinGW32/static/  
win32: INCLUDEPATH += $$PWD/libUSB/MinGW32/static/  
win32: DEPENDPATH += $$PWD/libUSB/MinGW32/static/  
LIBS += -lusb-1.0
```

Předpona win32 naznačuje, že tyto soubory budou přidány pouze při kompilování pro 32-bitový systém Windows, `$$PWD/libUSB/MinGW32/static/` je cesta k statickým knihovnám umístěným v počítači. Příkazem `-lusb - 1.0` přidáme knihovny LibUSB do projektu. Po přidání knihoven lze pak snadno využívat příkazy z této knihovny a ovládat USB zařízení.

Přidání knihovny CyAPI lze pomocí následujících příkazů:

```
LIBS += -L$$PWD/lib/  
LIBS += -lCyAPI -llegacy_stdio_definitions -luser32
```

## 4.1 PROGRAMOVÉ VYBAVENÍ PRO MĚŘÍCÍ APLIKACI

Statické knihovny jsou uloženy v tomto případě ve složce `$$PWD/lib/`. Je nutné přidat knihovny `legacy_stdio_definitions` pro přidání knihoven pro kompatibilitu se staršími knihovnamy `stdio.h`. Knihovna `user32` slouží pro potřeby knihovny `CyAPI`. Po přidání těchto souborů a následně i souboru s definicemi lze využít tuto knihovnu v projektu.

Pro příjem videozáznamu využívám vlastní vlákno, které přijímá obrazová data a vyžádá si nová videodata. Ukázka kódu pro příjem dat:

```
if(... && getVideo(data,paket,paket,timeout)){
...
if(!(registry==1 || registry==2)){
    prikaz[0]=getVideoCommand;
    prikaz[1]=0x00;
    send(prikaz,1,0);
}
...
frames++;
emit renderedImage(vid, videoPripoj);
}
```

Tato data jsou přijímána metodou `getVideo` s parametry. Tato metoda přijme obrazová data a uloží je do pole s proměnnou `data`. Pokud jsou data úspěšně přijata, vyšle se příkaz pro přijetí nových obrazových dat. Příkaz pro přijetí obrazových dat se provádí pomocí metody `send` s příslušným příkazem. Tyto příkazy jsou popsány níže. Při požadavku na příjem obrazových dat nelze se senzorem komunikovat, dokud nejsou přijata zobrazovací data. Pokud se provede příkaz pro příjem obrazových dat, mohou se přenést všechna video data ze senzoru. Pokud by se tento příkaz zpozdil, není pak možné zaručit příjem všech dat ze senzoru, proto vlákno obsahující tento kus kódu musí běžet ve vysoké prioritě. Pokud chceme poslat jiný příkaz do senzoru, například pro přečtení nebo zapsání do registru, musí se tento příkaz přenést dříve, než se vyžádá nový snímek. Proto se tento příkaz neposílá vždy, ale jen tehdy, pokud nejsou vyžadována jiná data. Po přijetí obrazových dat a odeslání nového požadavku jsou tato data zpracována, a to příkazem `emitrenderedImage`, kde se následně v hlavní `main` smyčce tato data zpracují a zobrazí v aplikaci.

Metoda pro samotný příjem dat pomocí `LibUSB` je následující:

```
bool cypress_video_thread::getVideo(unsigned char * data,
int size,int expectedBytes,uint timeout)
{
    int err;
    int transffered_bytes = 0;
    err = libusb_bulk_transfer(dev_handle,0x82,
        data,size,&transffered_bytes,timeout);
    if(err || transffered_bytes < expectedBytes){
...
        return false;
    }
```

```

    }
    return true;
}

```

V tomto příkladu jsou data přijímána metodou `libusb_bulk_transfer`, která přijímá data z endpointu `0x82` velikosti `size` a maximální dobou `timeout`. Data jsou uložena do proměnné `data`. Pokud jsou data přijata v požadovaném čase, je navrácena hodnota úspěch, jinak je navrácen neúspěch.

Tato metoda přepsaná pro CyAPI je následující:

```

bool cypress_video_thread::getVideo(unsigned char * out_data_buf,
    long size,int expectedBytes,
    uint timeout)
{
...
    endpt->XferData(out_data_buf,size);
    transferrered_bytes=size;
...
    if(transferrered_bytes < expectedBytes){
...
        return false;
    }
    return true;
}

```

V tomto kusu zdrojového kódu je samotná přijímací metoda `endpt->XferData`. Timeout je v tomto případě nastaven dříve v programu a není využit přímo. Návrátová hodnota je stejná jako v předešlém případě.

Pro zobrazování přijatých obrazových dat využívám vlastní třídu, která dědí z QT frameworku z třídy `QLabel`. To mi umožňuje využít zobrazení přijatých dat jako `QImage`, který může `QLabel` zpracovat a zobrazit.

Pro zobrazování grafů využívám widget `QCustomPlot`. Je licencován pod GPL, je tedy možné jeho využití v tomto projektu. Umožňuje přibližování a oddalování grafů, export do různých typů souborů a mnoho dalších nastavení. Ve své práci jsem tuto knihovnu implementoval a využil pro zobrazení naměřených dat.

### ■ 4.1.3 Použité kompilátory pro QT

V operačním systému Windows lze využít různé kompilátory pro sestavení zdrojových kódů. QT Creator podporuje nejrozličnější kompilátory, a to včetně gcc a Visual studio.

Ve své diplomové práci jsem použil gcc překladač i překladač od Visual studia. Překladač gcc jsem využil pro kompilaci s ovladači LibUSB. Překladač od MinGW spolu s gcc se vyznačují skvělou podporou debugu v programu, kde lze program pozastavit na jakémkoliv místě a vyčítat všechny proměnné s velkou rychlostí. Je možné zastavovat program i na

## 4.2 PROGRAM PRO KOMUNIKAČNÍ DESKU CYPRESS

bodech, kde je špatná alokace paměti nebo volání neexistujících metod. Další jeho velkou výhodou je možnost využití knihovny LibUSB, která je snadno linkovatelná. Překladač MinGW je velice podobný kompilátoru použitému v Linuxu, proto mi usnadnil vývoj v systému Linux. MinGW lze nainstalovat přímo s instalací QT Creator.

Druhý překladač, který jsem použil, je překladač ve vývojovém prostředí Visual studio. Tento překladač jsem využil pro kompilaci s USB ovladačem CyAPI. Ovladač CyAPI není kompatibilní s gcc a tudíž ani s MinGW. Po instalaci Visual studio rozpozná program QT Creator jeho nainstalování a pomůže s nastavením pro jeho použití. Po instalaci a nakonfigurování tohoto překladače je možné využít ovladač CyAPI a komunikovat s USB zařízením pomocí tohoto ovladače.

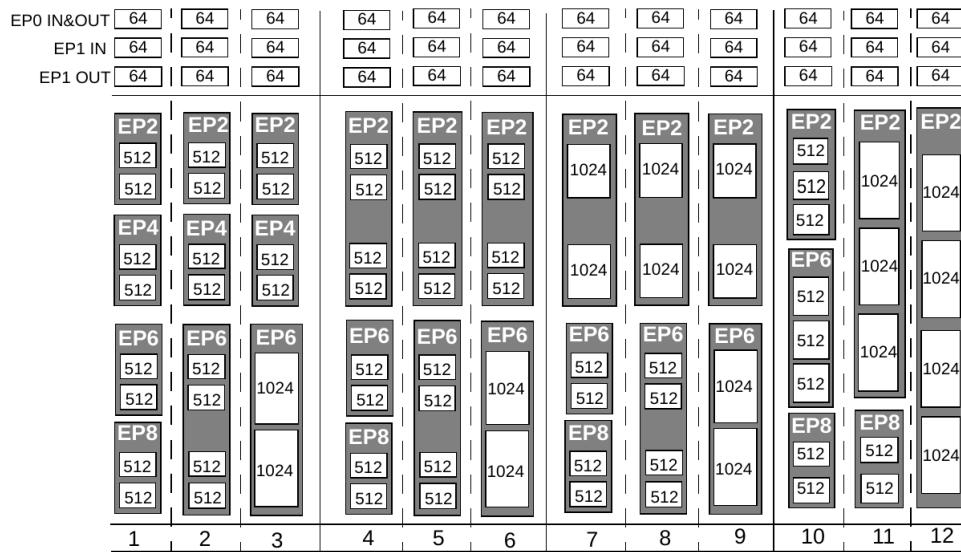
## ■ 4.2 Program pro komunikační desku Cypress

Program pro komunikační desku Cypress jsem převzal z diplomové práce pana Ondříčky (Ondříčka 2014), která implementovala jak program pro čip Cypress, tak i program pro Raspberry PI. Tento program byl vyvíjen pouze pro jeden senzor, a proto bylo nezbytné tento program zdokonalit a rozšířit. Rozšířil jsem ho o zápis jednoho bajtu dat po komunikační sběrnici I<sup>2</sup>C, o čtení dvou bajtů dat pomocí I<sup>2</sup>C, o možnost přijímání více typů obrazových dat a optimalizoval jsem jeho rychlost.

### ■ 4.2.1 Nastavení komunikační desky Cypress

Komunikační deska Cypress je nastavena tak, aby byly aktivní 3 endpointy s aktivním řídicím endpointem. Řídicí endpoint je plně řízený čipem Cypress a programový zásah do tohoto endpointu jsem neprogramoval. Další je využit endpoint číslo jedna, a to jak směrem z počítače, tak do počítače. Endpoint číslo 1 je v mém programu využit pro přenos I<sup>2</sup>C dat, řízení přenosu obrazových dat a pro požadavek na přenos snímku obrazových dat. Endpoint má v obou směrech vyhrazenou vlastní paměť pro odesílání. Obrazová data se přenáší po endpointu číslo 2, který je připojen k FIFO paměti a je nastaven v automatickém módu. Endpointy číslo 2, 4, 6 a 8 mají sdílenou FIFO paměť. Jelikož využívám jen jeden endpoint z této kategorie, mohu využít celou FIFO paměť pouze pro tento jeden určený endpoint. Velikost paketů odesílaných přes rozhraní USB jsem nastavil na 512 bajtů kvůli možnosti lepšího zaokrouhlení přijatých dat z CMOS senzoru. Počet odeslaných obrazových dat musí být násobek 512 a příjem obrazových dat musí s tímto problémem počítat. Nastavení FIFO paměti je znázorněno na obrázku 7 a to ve sloupci číslo 5.

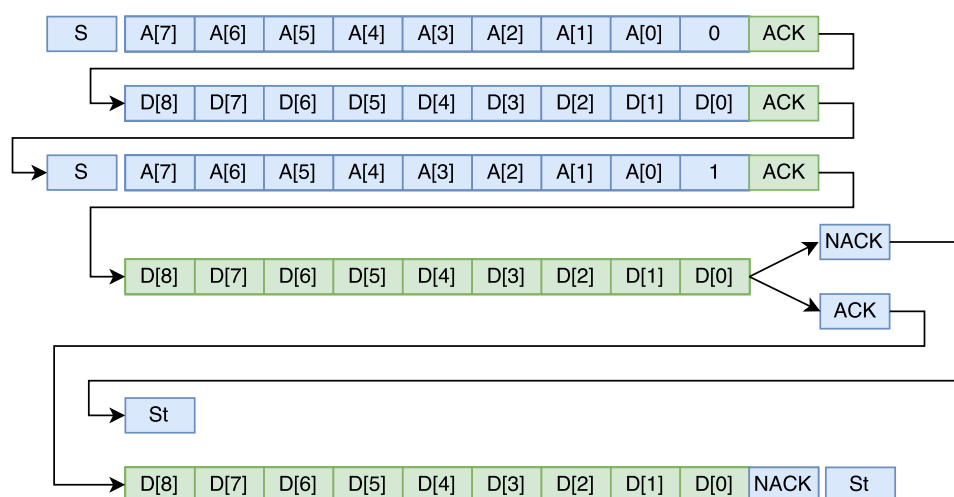
## 4.2 PROGRAM PRO KOMUNIKAČNÍ DESKU CYPRESS



Obrázek 7: Nastavení FIFO paměti, převzato z Cypress datasheet (Cypress 2013)

## 4.2.2 Popis komunikace I<sup>2</sup>C na komunikační desce Cypress

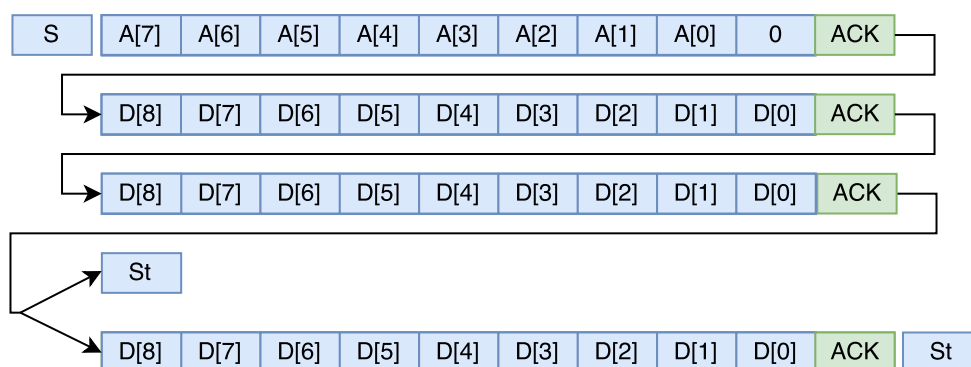
Pro nastavení registru na senzorech CMOS slouží komunikace I<sup>2</sup>C. Tato komunikace získává informace o CMOS senzoru a umožňuje nastavení ROI, uzávěrky, snížení rozlišení a mnoho dalších parametrů. Sensory podporují I<sup>2</sup>C komunikační rozhraní, ale u každého CMOS sensor je komunikace mírně odlišná. Přečtení dvou bajtů informací z I<sup>2</sup>C jsem vybral kvůli kompatibilitě jak se senzory, které obsahují jen jeden bajt registrů, tak i se senzory, které obsahují 16-bitové velikosti registrů. Pro čtení jen jednoho bajtu dat se přečtou bajty dva, ale hledaná informace je obsažena v prvním přijatém bajtu. Jak lze sledovat z obrázku 8, rozdíl při přečtení jednoho nebo dvou bajtů dat je pak v odpovědi ACK, případně NACK, od mikrokontroléru, data jsou však tou dobou již úspěšně přijata. Pokud sensor podporuje čtení pouze jednoho bajtu, data prvního bajtu jsou validní a při čtení dalšího bajtu může být přečteno cokoliv. Z tohoto důvodu musí být zbylá data ignorována.



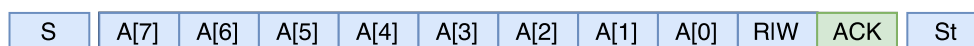
Obrázek 8: Srovnání čtení jednoho a dvou bajtů dat po protokolu I<sup>2</sup>C

Zápis dat do protokolu I<sup>2</sup>C jsem rozšířil v programu o zápis jednoho bajtu. Rozhodl jsem se tak proto, že některé senzory neumožňují zápis dvou bajtů po I<sup>2</sup>C, chování těchto sensorů pak může být nepředvídatelné. Na obrázku 9 je zobrazen zápis dvou bajtů přes I<sup>2</sup>C. Některé senzory obsahují takzvané *shadow* registry. Jedná se o registry, u kterých se hodnota zapsaná do protokolu I<sup>2</sup>C projeví až po předem definované události senzoru. Po experimentech prováděných s různým nastavením komunikace jsem zjistil, že u senzoru 3.1.1.4 je možné zapisovat do těchto *shadow* registrů pomocí I<sup>2</sup>C, a to konkrétně zápisem dvou bajtů dat místo jednoho. Toto chování není přesně definované v datasheetu, je tedy nutné chování při zápisu do těchto registrů předem vyzkoušet. U ostatních sensorů obsahujících *shadow* registry se mi do nich zapisovat nepodařilo.



Obrázek 9: Srovnání zápisu jednoho a dvou bajtů dat přes protokol I<sup>2</sup>C

Dalším problémem bylo zjištění a rozpoznání jednotlivých senzorů CMOS. Protokol I<sup>2</sup>C umožňuje adresování senzoru, kde každá přijatá adresa je zodpovězena pomocí ACK nebo NACK. Každý senzor odpoví ACK právě na svoji vlastní adresu. Tato komunikace je zobrazena na obrázku 10. Každý ze senzorů popsaných v kapitole 3.1.2 má definovanou jinou adresu pro komunikaci po sběrnici I<sup>2</sup>C. Těto vlastnosti využívám k rozpoznání senzorů. Odpověď na adresu jako rozpoznávací znak nestačí, protože některé senzory umožňují tuto adresu změnit. Toho lze docílit přečtením registru, který se nachází na adrese 0. Registry na adrese 0 obsahují ID senzoru nebo jeho revize. Toto platí u všech senzorů s výjimkou senzor 3.1.1.4. Tento senzor neposkytuje v registru s adresou 0 žádné informace, avšak požadované informace odesílá na I<sup>2</sup>C sběrnici. Každý senzor použitý v této práci tak lze nezaměnitelně identifikovat.



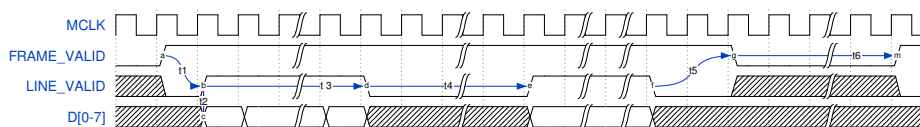
Obrázek 10: Rozpoznání připojeného senzoru podle adresy

### 4.2.3 Implementace paralelního rozhraní pro kamery v desce Cypress

Paralelní rozhraní pro kamery je použito pro přijímání obrazových dat ze senzorů CMOS. Obrazová data jsou hlavními měřenými daty, a proto jejich přenos musí být co nejspolehlivější. Obrazová data jsou přenášena paralelním rozhraním pro kameru hlavně z důvodu rychlého přenosu velkého počtu dat.

Rychlost přijímací smyčky pro obrazová data musí být co největší kvůli správnému zapnutí přijímacího bufferu pro zobrazovací data z CMOS senzoru, jak je znázorněno na obrázku 11. Program musí čekat na FRAME\_VALID od senzoru, následně program zapne FIFO paměť, a pak už se o zápis do FIFO paměti starají signály LINE\_VALID a PIX\_CLK. Mezi jednotlivými snímky se musí data přenést do počítače a počítač musí znovu požádat o nová obrazová data. Toto je zapříčiněno automatickou zprávou FIFO paměti, kterou jsem využil kvůli různorodosti senzorů.

## 4.2 PROGRAM PRO KOMUNIKAČNÍ DESKU CYPRESS



Obrázek 11: Paralelní komunikace pro získání obrazových dat z CMOS senzoru

Posledním zlepšením programu bylo napsání několika různých kódů přijímání obrazových dat z CMOS senzorů. Následující kus kódu přijímá obrazová data, který se u některých senzorů liší pouze odstraněním vybraných řádků. Konkrétně jde o zapnutí výstupu pro senzor a nečekání na zapnutí a vypnutí FIFO paměti od signálu LINE\_VALID.

```
FIFORESET = 0x02;
SYNCDELAY;
SEN_OE = inverted;

#pragma asm
  CLR FIFOADR1
  FRAME_VALID_1_6: JB CCD_PORT.0, FRAME_VALID_1_6
  FRAME_VALID_0_6: JNB CCD_PORT.0, FRAME_VALID_0_6

  LINE_VALID_6: JB CCD_PORT.1, LINE_VALID_6

  CLR FIFOADRO

  FRAME_VALID_6: JB CCD_PORT.0, FRAME_VALID_6

  LINE_VALID_1_6: JNB CCD_PORT.1, LINE_VALID_1_6
  LINE_VALID_2_6: JB CCD_PORT.1, LINE_VALID_2_6

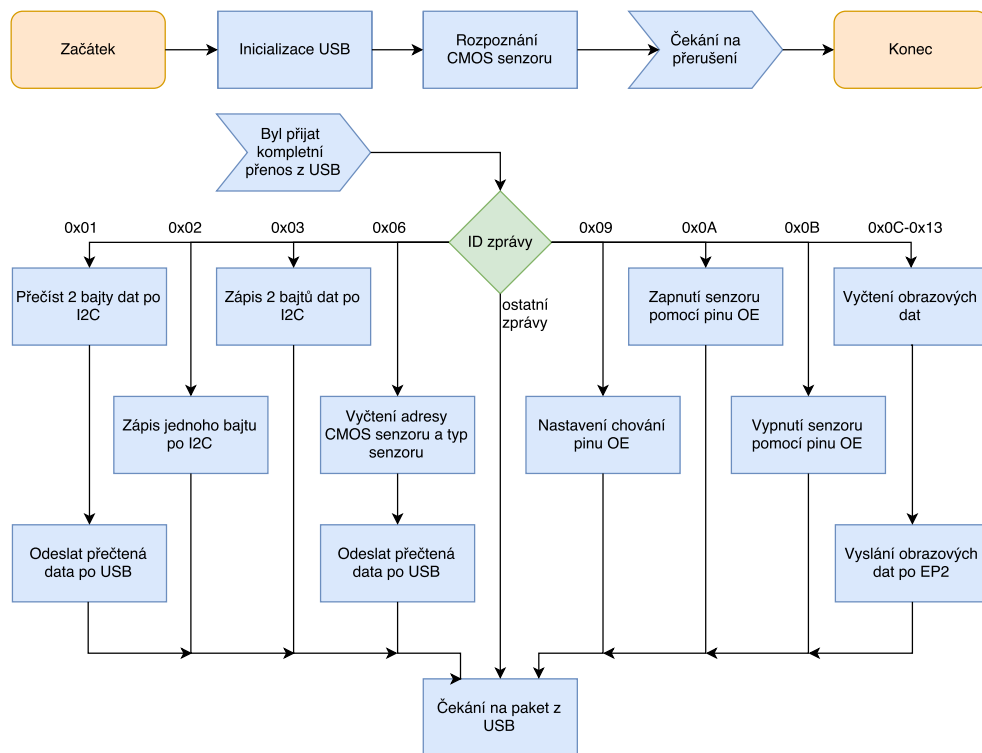
  FIFOADRO
  FIFOADR1
#pragma endasm
SEN_OE = inverted2;
```

Tento kus programu musí být co nejrychlejší, aby byl zajištěn spolehlivý přenos obrazu do počítače. Proto je většina tohoto kódu napsána v jazyce symbolických odkazů. Ze začátku je doporučeno použít příznak NACK pro všechny USB přenosy a dále pak resetovat FIFO paměti a povolit zpět přenosy po USB. Kvůli rychlosti jsem využil pouze reset FIFO paměti bez příznaku NACK pro USB přenosy. Poté využívám *SEN\_OE* pro zapnutí senzoru podle již dříve nakonfigurované proměnné *inverted*. Program dále čeká na FRAME\_VALID, který označuje příjem nového snímku přes paralelní rozhraní. Pro některé senzory je nutné vynechat první sloupec přijímaných dat, zejména u senzorů, kde se FRAME\_VALID a LINE\_VALID aktivují ve stejný čas. Pokud se FRAME\_VALID a LINE\_VALID aktivuje ve stejný čas, nestihne hlavní smyčka zapnout FIFO paměť a

### 4.3 PROGRAMOVÁ IMPLEMENTACE USB KOMUNIKACE

data jsou pak nekonzistentní. LINE\_VALID zajistí včasné zapnutí FIFO paměti. Příkaz *CLR FIFOADR0* aktivuje FIFO paměť. Po zapnutí FIFO paměti se čeká na dokončení snímku signálem LINE\_VALID do logické úrovně 0. Velikost dat ze senzoru musí být násobky 512 z důvodu kompatibility s USB přenosem. Čekání na LINE\_VALID má zajistit přenos dalších obrazových dat tak, aby byla splněna minimální násobnost 512. Tato násobnost není použita při některých příkazech pro příjem dat. Nakonec se vypne FIFO paměť příkazem *FIFOADR0* a dále je možné FIFO paměť přenést do počítače.

Vývojový diagram pro propojovací desku je na obrázku 12.



Obrázek 12: Vývojový diagram pro propojovací desku Cypress

## 4.3 Programová implementace USB komunikace

Pro využití USB komunikace je nutné zajistit přijímání dat z tohoto zařízení. Je možné využít již hotové komunikace se zařízeními jako jsou HID, Mass Storage, Video, Vendor Specific a mnoho dalších. Pro většinu existují ovladače přímo zajištěné operačním systémem, které po připojení nainstalují a zprovozní požadovaný ovladač. U USB zařízení jako je Vendor Specific je nutné nainstalovat svůj vlastní ovladač, ale poskytují výhodu v plné kontrole nastavení USB přenosu a jeho zpracování. Vybral jsem proto Vendor Specific,

## 4.3 PROGRAMOVÁ IMPLEMENTACE USB KOMUNIKACE

abych zajistil, že přenos bude přesně vyhovovat požadavkům přenosu obrazových dat, a také bude možné nastavit senzor do požadovaného stavu.

Je nutné zajistit vlastní ovladače pro specifikaci Vendor Specific a vytvořit vlastní komunikaci přes USB. Využil jsem proto LibUSB a CyAPI, které zajišťují instalaci ovladačů a poskytují rozhraní pro možnost přijímání dat z USB. LibUSB je knihovna pro operační systémy Windows, Linux, OS X a další. CyAPI je USB knihovna poskytována firmou Cypress pro jejich produkty. Výhodou CyUSB je, že poskytuje podepsané ovladače pro systémy Windows.

### ■ 4.3.1 Knihovna LibUsb a její použití

Knihovna LibUSB se stará o posílání a odesílání dat po endpointech pro USB. V Linuxu je LibUSB velice dobře implementována a komunikace s jakýmkoliv USB zařízením je bezproblémová. Ve Windows je nutná instalace nepodepsaných ovladačů. LibUSB poskytuje předpřipravé zkompileované knihovny pro snazší použití této knihovny. Po naimportování a nainstalování potřebných ovladačů lze nainicializovat USB zařízení pomocí níže uvedeného kódu. Je potřeba inicializovat samotnou knihovnu a pak už jen stačí vybrat správné zařízení pomocí *libusb\_get\_device\_list* a následně získat descriptor a připojit se k interface. Ukázka jak to může být provedeno:

```
err = libusb_init(NULL);
...
cnt = libusb_get_device_list(NULL, &devs);
device = devs[i];
...
err = libusb_open(device, &dev_handle);
...
err = libusb_get_device_descriptor(device, &desc);
...
libusb_detach_kernel_driver(dev_handle,0);
err = libusb_claim_interface (dev_handle, 0);
```

Jedná se pouze o zjednodušený kód, vlastní kód obsahuje navíc výběr zařízení, kontroly chyb a jejich ošetření. Po této inicializaci lze pak následně využívat například pro bulk přenos následující metodu:

```
err =
    libusb_bulk_transfer(dev_handle,0x01,
        in_data_buf,size,&transffered_bytes,1);
```

Prvním parametrem je zařízení, další parametry jsou číslo endpointu, pole přijatých nebo odeslaných dat a počet přijatých nebo odeslaných dat a jejich skutečné odeslání. Poslední parametr je timeout v milisekundách. Po zavolání této funkce se bude program snažit přečíst nebo zapsat data na určitý typ endpointu o velikosti *size*. Pokud se to nepovede do času určeného v timeout, je program přerušen a navracená hodnota je chyba.

Instalace samotných ovladačů není v Linuxu potřeba. V operačním systému Windows je potřeba nainstalovat požadované ovladače, například programem Zadig (Batard 2017). Tento program umožňuje nainstalovat nepodepsané ovladače určitému USB zařízení. Pojmem nepodepsány se rozumí ovladače, které nejsou podepsány firmou Microsoft, proto je nutné vypnout vynucené podepisování ovladačů. Toho lze docílit například například přidržet klávesy Shift při stisknutí tlačítka restartovat. Počítač se pak dostane do obrazovky pro výběr opravy Windows, kde lze vypnout podepisování ovladačů.

### ■ 4.3.2 Knihovna CyAPI a její použití

Knihovna CyAPI poskytuje podobné vlastnosti jako LibUSB, ale je plně vyvíjena firmou Cypress. Poskytuje jako LibUSB předkompilované knihovny, které stačí jen připojit k programu a lze je používat. Inicializaci lze provést následujícím kódem:

```
USBDevice = new CCyUSBDevice();
USBDevice->Open();
USBDevice->SetAltIntfc(1);
int eptCount = USBDevice->EndPointCount();
...
endpt = (CCyBulkEndPoint *) USBDevice->Endpoints[i];
endpt->Reset();
endpt->SetXferSize(size);
endpt->PktsPerFrame = size/512;
endpt->TimeOut = timeout;
```

První příkaz vytvoří a inicializuje USB knihovnu a další řádek otevře toto zařízení. Zvolí se první konfigurace a poté se stačí připojit k jednotlivým endpointům, které lze nastavit (primárně velikost přijatých a odeslaných dat, velikost paketů a timeout)

Poté už stačí zavolat metodu pro odesílání nebo přijímání dat:

```
endpt->XferData(in_data_buf, size);
```

Proměnná `in_data_buf` obsahuje buffer přijatých nebo odeslaných dat a `size` obsahuje velikost tohoto pole, metoda do ní navrátí počet přijatých nebo odeslaných dat.

### ■ 4.3.3 Příkazy pro ovládání Cypress desky

Ovládání desky Cypress pomocí komunikace USB a přenos obrazových dat využívá 16 příkazů pro příjem obrazových dat, nastavení senzoru, vyčtení informací nebo nastavení komunikace. Nejvíce příkazů je pro příjem obrazových dat.

Každý paket přijatý komunikační deskou Cypress je dekodován tak, že první bajt je požadovaný příkaz a další bajty jsou data pro tento příkaz. Pokud příkaz potřebuje odeslat data zpět do počítače, odešle tato data přes endpoint číslo 1 směrem do počítače.

### 4.3 PROGRAMOVÁ IMPLEMENTACE USB KOMUNIKACE

Příkaz 0x01 načte dva bajty dat přes protokol I<sup>2</sup>C. Registr, ze kterého budou čteny 2 bajty, je uložen jako první bajt přijatých dat po USB. Poté se odešlou 2 bajty dat do počítače. Pokud se přečtou neplatná data, je navrácena hodnota 255 v obou datech.

Příkaz 0x02 zapíše 1 bajt po sběrnici I<sup>2</sup>C. Registr, do kterého se budou data zapisovat, je uložen jako první bajt přijatých po USB, další parametr je bajt dat, který se má zapsat. V tomto příkazu nejsou odeslána žádná data zpět.

Příkaz 0x03 je obdoba příkazu 0x02, ale na místo jednoho bajtu se odesílají bajty dva.

Příkazy 0x04, 0x05 a 0x08 měly být použity pro časové a externí snímání, prozatím jsou tyto příkazy ve vývoji.

Příkaz 0x06 vyčte informace o adrese I<sup>2</sup>C a číslo rozpoznávaného senzoru. Program dokáže rozpoznat všechny CMOS senzory uvedené v této diplomové práci. Navrácená informace v přijatých datech pro PC, je v prvním bajtu adresa I<sup>2</sup>C a v druhém bajtu je číslo senzoru podle následujícího seznamu: 1 je senzor MT9M001, číslo 2 je senzor MT9T031, číslo 3 je senzor KAC9618, číslo 4 je senzor KAC9638, číslo 5 je senzor MT9V032 a nakonec číslo 6 je senzor MT9VO34. Pokud je přijato číslo 255, pak sběrnice I<sup>2</sup>Cc rozpoznala senzor, který není použit v této DP, lze s ním však komunikovat a nastavit požadované parametry. Při navrácení hodnoty 0 nebyl rozpoznán žádný senzor na sběrnici I<sup>2</sup>C, komunikace po I<sup>2</sup>C sběrnici se pak odesílá na adrese 0.

Příkaz 0x09 nastaví zapínací signál pro CMOS senzor (OE). Jedná se o pin, který povoluje přenos obrazových dat po paralelním rozhraní pro kamery. Tento pin může být aktivní jak v logické úrovni 0, tak i v logické úrovni 1. Tímto příkazem je možné nastavit požadovanou logiku pro příjem obrazových dat. Pokud přijatá data jsou rovna nule, je nastavena logika úrovně 0 při příjmu obrazových dat. Pokud jsou data nenulová, jsou obrazová data přenášena při logické úrovni 1 na pinu OE.

Příkaz 0x0A a příkaz 0x0B zapínají nebo vypínají pin OE, který dovoluje zapnutí pinu pro přijímání obrazových dat. Některé příkazy pro příjem obrazových dat spoléhají na zapnutí senzoru, čehož lze docílit právě pomocí příkazů 0x0A a 0x0B.

Příkazy 0x07, 0x0C až 0x13 jsou pro zahájení přenosu obrazových dat. Obrazová data se přenáší po endpointu číslo 0x02. Každý příkaz přeneše obrazová data v jiném tvaru. Obrazová data jsou přenášena od nultého až po nejvyšší pixel v raw hodnotě. K tomu účelu je využita 4kB vyrovnávací paměť pro rychlý přenos dat po USB. Hodnoty rozlišení nejsou přenášeny, a proto je nezbytné u každého přenosu znát samotné rozlišení.

Příkaz 0x07 a 0x0C přenáší obrazová data hned po prvním příznaku pinu FRAME\_VALID a končí přenos, když se příznak FRAME\_VALID dostane do logické úrovně 0. Obrazová data se při tom přenáší pomocí endpointu 0x02 do počítače. Pokud nejsou data průběžně čtena z toho endpointu, může nastat ztráta dat a přenos nemusí být kompletní. Proto je nutné, aby vlákno běželo s vysokou prioritou. Tento příkaz při každém požadavku zapne pin OE a po přenosu tento pin vypne podle nastavení pro určitý senzor.

Příkaz 0x0D začne přenášet obrazová data po endpointu 0x02 stejně jako příkaz 0x0C, ale s tím rozdílem, že se po příznaku FRAME\_VALID čeká ještě na jeden řádek přenosu dat. Tento přenos dat může být nápomocen při problému čekání mezi signály FRAME\_VALID a LINE\_VALID, pokud čas mezi těmito signály není dostatečně dlouhý. Tímto příkazem se přenesou data o jednom snímku, ale s vynechaným prvním řádkem dat. I v tomto případě

se senzor nejdříve zapne a po přenosu dat se pin OE vypne.

Příkaz 0x0E je stejný jako příkaz 0x0C s tím rozdílem, že senzor a pin OE se automaticky nezapíná. Této vlastnosti je dobré využít při rychlejším přenosu obrazových dat.

Příkaz 0x0F je stejný jako příkaz 0x0D, ale s tím rozdílem, že senzor a pin OE se automaticky nezapíná.

Příkaz 0x10 je stejný jako příkaz 0x0C, ale na konci přenosu se čeká po FRAME\_VALID ještě na přenos jednoho řádku dat pomocí LINE\_VALID. Toho lze využít, pokud přenesená data nejsou dělitelná číslem 512. Pokud data i nadále nejsou dělitelná 512, data se nepodaří přenést do počítače. V tomto příkazu je zapínán a vypínán senzor. Pokud není zapínání žádoucí, lze využít obdobného příkazu 0x12.

Příkaz 0x11 kombinuje příkazy 0x0D a 0x10. Senzor čeká na přenos až po přečtení jednoho řádku obrazových dat a je ukončen až po přečtení jednoho řádku obrazu dat po signálu FRAME\_VALID. Senzor se v tomto případě automaticky zapíná a vypíná. Pokud není zapínání senzoru žádoucí, lze využít obdobného příkazu 0x13.

## ■ 4.4 Počítačový program pro ovládání měření

Program pro počítač byl vytvořen ve frameworku QT. To umožňuje jednoduchou práci s vlákny, kterou jsem ve svém programu využil. Pro příjem obrazových dat, zpracování obrazových dat, ovládání osvětlovače nebo obnovení GUI jsou vytvořena vlastní vlákna, která napomáhají, aby hlavní zobrazovací vlákno nebylo přerušeno a GUI aplikace se nezašakávala. Pro co nejspolehlivější příjem má vlákno pro příjem obrazových dat nejvyšší prioritu, ostatní vlákna mají prioritu normální. Program umí zobrazit obrazová data, zahájit měření, zobrazit grafy a zobrazit základní informace.

Program je dělen na hlavní třídy MnohoMereni, MnohoGraph, MnohoVidei, Nastaveni, Graph a Video. Tyto třídy zajišťují fungování celého programu a umožňují zobrazení obrazových dat, zobrazení grafu a zobrazení grafiky pro měření a jeho možnost nastavení a mnoho dalšího. Třídy MnohoMereni, MnohoGraph a MnohoVidei umožňují v sobě ukládat vytvořené grafy, vytvořené obrázky z obrazových dat a uchovávat v sobě grafiku a instance měřících a ovládacích tříd. Třída Nastaveni uchovává potřebné informace o nastavení jednotlivých zařízení včetně měření a dále poskytuje možnosti nastavení pro jednotlivé instance měřících a ovládacích tříd.

Pro zobrazení grafů používám QCustomPlot, knihovnu pro framework QT, která je licencována pod GPL. Tato knihovna umožňuje zobrazení grafu, export, interakci pomocí myši, automatickou změnu hranic grafu a další funkce. Pro manipulaci s grafy jsem vytvořil vlastní třídu Graph, která uchovává instanci na QCustomPlot, data, jména os a další data. Tato třída umožňuje export grafu do tabulkového formátu nebo export grafu jako obrázku. Třída Graph se vkládá do třídy MnohoGraf, která spravuje instance třídy Graph, umožňuje zobrazení grafu ve vlastním okně a umožňuje počítat informace obsažené v grafech.

Pro zobrazení obrazových dat využívám třídu Video, která dědí z QLabel. Třída QLabel

## 4.4 POČÍTAČOVÝ PROGRAM PRO OVLÁDÁNÍ MĚŘENÍ

slouží pro zobrazení textu nebo obrazových dat. Třída Video rozšiřuje QLabel o prvky možnosti výběru obrazových dat, a to obdélníkové oblasti, přímky nebo bodu. Dále tato třída umožňuje zvětšení nebo zmenšení zobrazených obrazových dat. Třída Video se vkládá do třídy MnohoVidei, která poskytuje informace o zobrazených datech, umožňuje export do formátu Windows Bitmap a dále základní manipulaci s třídou Video.

Třída MnohoMereni uchovává informace o třídách, které mají vlastní GUI, poskytují připojení k zařízením nebo jakékoliv úpravy nastavení programu. Tyto třídy zprostředkovávají připojení řídicí desky Cypress, připojení osvětlovacího zařízení, možnost generovat video, měření šumu, základní přepočítání grafu a další. Třída MnohoMereni umožňuje pouze přístup k informacím k jednotlivým třídám a jejich vzájemným propojkám. Třída MnohoMereni se taky stará o zobrazení GUI k třídám, které je využijí. Toto zobrazení je pak vkládáno do pravé strany programu a umožňuje přidávat další potřebná měření, připojení dalších senzorů a další třídy, které využívají třídu MnohoMereni.

### ■ 4.4.1 Připojení komunikační desky Cypress

O připojení k desce Cypress se stará třída CypressVideoThread. Třída CypressVideoThread se stará o grafické rozhraní k připojení, nastavení senzoru a videa. Třída CypressVideoThread běží ve vlastním vlákně s nejvyšší prioritou, stará se o připojení k desce Cypress, vyčtení základních informací o senzoru, nastavení základních registrů a příjem obrazových dat. Dále se stará o konzistenci přijatých obrazových dat. Z diplomové práce Ing. Zoubka vychází, že přijímání obrazových dat ze senzoru není spolehlivé.

*“100% bezchybnosti přenosu snímku nelze dosáhnout, proto je nutné chybně přijaté snímky detekovat a neposkytovat je dále ke zpracování zobrazovacímu vlákně.”* (B. M. Zoubek 2011b)

Postupoval jsem podobným způsobem a všechna špatně přijatá data ignoruji, jsou zahozena a odpovídající paměť je přepsána novými obrazovými daty.

Po připojení komunikační desky Cypress do portu USB deska Cypress automaticky rozpozná typ připojeného senzoru a adresu I<sup>2</sup>C, na který tento senzor reaguje. Tuto informaci program vyčte při připojení ke komunikační desce jako první. Po tomto kroku jsou podle rozpoznání senzoru nastaveny registry senzoru tak, aby byla kompatibilní s přijímáním obrazových dat. Je také nastaven základní počet pixelů v řádku a ve sloupci. Pokud senzor nepodporuje základní přijímání obrazových dat, je nastaven specifický příkaz pro příjem obrazových dat. Po těchto krocích je zapnut senzor pro příjem obrazových dat.

V hlavní přijímací smyčce se nejdříve dynamicky vytvoří přijímací paměť ve tvaru jednorozměrného 8-bitového pole o velikosti přijímaných dat ze senzoru CMOS. Pokud je potřeba toto pole vytvořit znovu, je původní pole smazáno a dealokováno a je vytvořeno nové požadované pole. Po vyžádání od uživatele je toto pole znovu vytvořeno, a to při změně počtu přijatých obrazových dat. Vytvoří se automaticky ještě jedno pole, které má stejnou paměťovou velikost jako popisované pole, které využívám k mnohonásobnému buffering, abych zajistil konzistenci dat.

Po vytvoření přijímacího pole se kontroluje, zda uživatel vyžádal změnu registru nebo přečtení určitého registru, nebo jiné specifické změny příjmu obrazových dat. Pokud uživatel



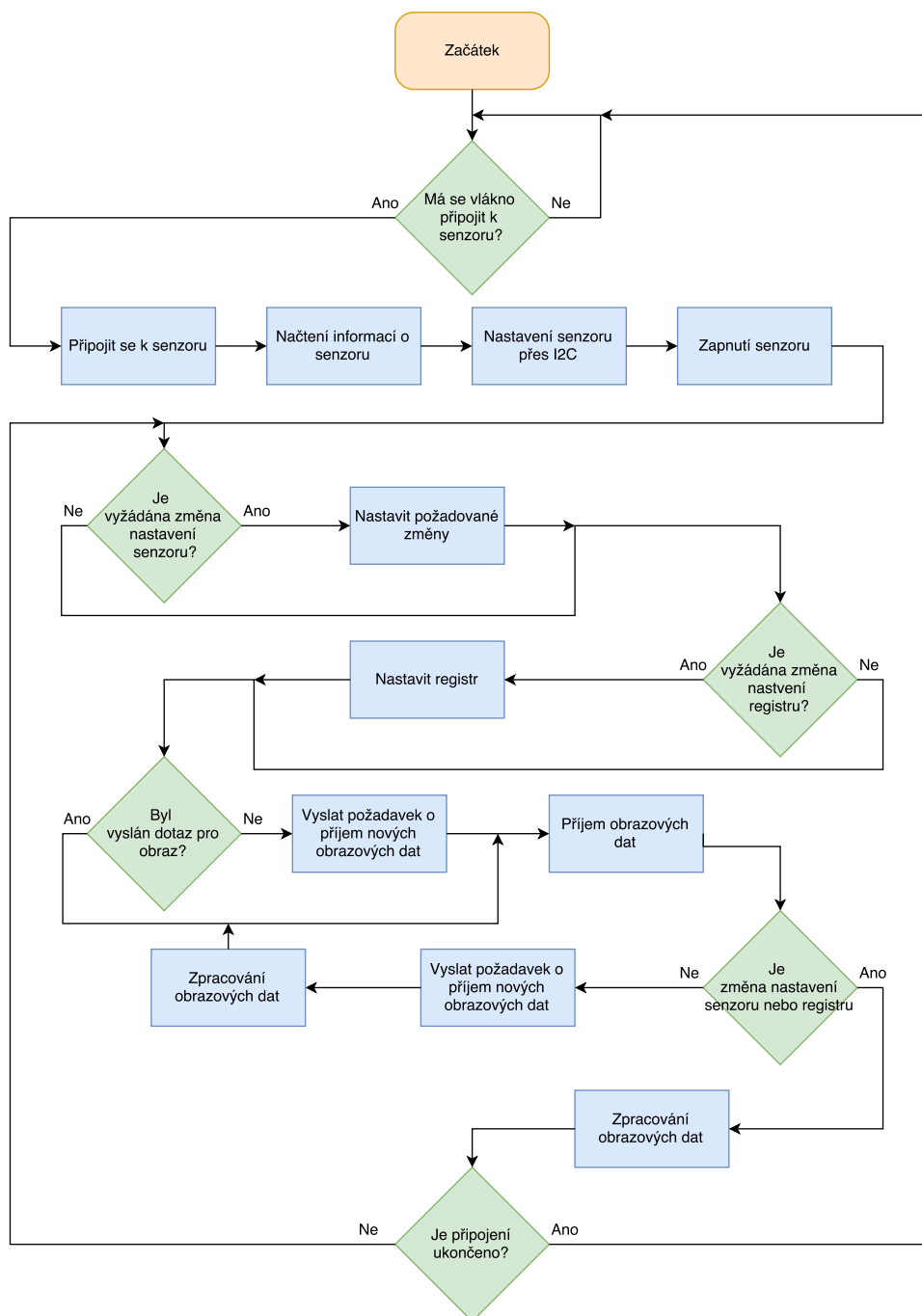
#### 4.4 POČÍTAČOVÝ PROGRAM PRO OVLÁDÁNÍ MĚŘENÍ

vyžádal určitou změnu, je tento krok proveden. Pokud uživatel nevyžádal žádnou informaci ani manipulaci s registry, program pokračuje dál.

Po tomto kroku jsou vyžádána obrazová data ze senzoru, pokud již tak nebylo provedeno. Dále jsou čtena obrazová data ze senzoru a jsou ukládána do přijímacího zobrazovacího bufferu. Po dokončení přenosu dat se zkontroluje velikost přijatých dat a pošle se požadavek na přenos nových obrazových dat. Tento požadavek se nevykoná, pokud uživatel zvolil změnu registru nebo jeho vyčtení. Až po tomto kroku jsou obrazová data zobrazena v GUI a jsou provedeny potřebné měřicí úkony. Hlavní přijímací smyčka se po tomto kroku znovu opakuje. V celém průběhu této smyčky se kontroluje, zda není potřeba tuto smyčku ukončit a odpojit se od zařízení.

Celý tento průběh můžeme sledovat na obrázku 13

#### 4.4 POČÍTAČOVÝ PROGRAM PRO OVLÁDÁNÍ MĚŘENÍ



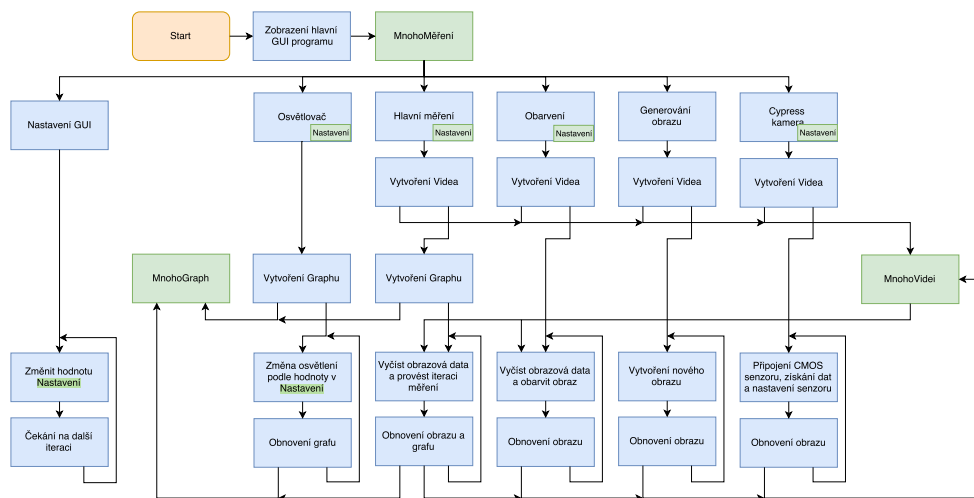
Obrázek 13: Vývojový diagram pro připojení desky Cypress k počítači

#### ■ 4.4.2 Implementace měření v počítačovém programu

Každé měření, které je prováděno programem, běží ve vlastním vláknu, aby nedocházelo ke zpomalení GUI. Výsledek měření může být zobrazen jako graf nebo jako nová zob-

#### 4.4 POČÍTAČOVÝ PROGRAM PRO OVLÁDÁNÍ MĚŘENÍ

rozená obrazová data. Měření může přijímat obrazová data, a to buďto celého obrazu, čtvercového výseče, přímky nebo pouze bodu. Tato data jsou zkopírována a analyzována pomocí měřicího algoritmu. Některé měřicí algoritmy vyžadují připojení externího zařízení, které algoritmus musí ovládat. Po příjmu nových obrazových dat je měřicím algoritmem tento stav zjištěn a zkopírují se potřebná zdrojová data. Tato data jsou analyzována, a pokud je potřeba, je nastaveno odpovídající zařízení na požadovanou hodnotu. Ovládání zařízení se provádí pomocí Nastavení, které zajistí požadované nastavení. Toto Nastavení umožňuje nastavit registry CMOS a také ovládat osvětlovač. Po tomto průběhu jsou data zobrazena v odpovídajícím formátu. Průběh je naznačen na následujícím obrázku 14.



Obrázek 14: Vývojový diagram pro počítačový program



## Kapitola 5

# Měřicí pracoviště

Pracoviště pro laboratorní experimenty se skládá ze senzorů CMOS, propojovací desky Cypress, počítače a osvětlovacího zařízení. Počítač je nutné nastavit a to z hlediska instalace ovladačů pro knihovnu LibUSB nebo CyAPI. V kapitole 5.1.2 pak představím měřící program pro získání dat z CMOS senzoru a jednotlivá měření. Jak bude znázorněno, program umožňuje spuštění nejrůznějších měření nejen těch, co tu budou popsána. Dále je možné ukládání zobrazeného obrazu nebo sekvenci obrazů, což umožňuje pozdější analýzu.

### 5.1 Uspořádání pracoviště a prvotní spuštění programu

Pracoviště obsahuje následující elektronické zařízení: laboratorní zdroj s výstupy  $U_{Logic} = 5\text{ V}$  a  $U_{Diody} = 15\text{ V}$ , osvětlovací zařízení, převodník RS232, propojovací USB mini-B kabel, propojovací desku Cypress a všechny měřené senzory CMOS a v poslední řadě počítač kompatibilní s měřící aplikací.

Tento laboratorní zdroj je použit pro napájení osvětlovacího zařízení. Musí být použit přesný laboratorní zdroj, aby nedocházelo ke zkreslení měření. Je navržena druhá varianta tohoto osvětlovacího zařízení v bakalářské práci, která stabilizuje napětí automaticky, ale toto elektronické zařízení nebylo zrealizováno. K osvětlovacímu zařízení je připojen převodník RS232 na USB, který je připojen dále do počítače. K instalaci ovladačů pro převodník RS232/USB postupujte podle návodu k tomuto převodníku.

Po připojení osvětlovacího zařízení je dobré toto zařízení vyzkoušet a to vysláním zprávy "Y2000X" po komunikační lince RS232 k osvětlovači. Po přijetí této zprávy pracuje osvětlovač v nejvyšším výkonu.

Dále je potřeba připojit komunikační desku Cypress. Aby bylo možné připojit komunikační desku, je nezbytné nejdříve připojit senzor CMOS ke komunikační desce. Připojování CMOS senzoru se musí provádět při vypnuté komunikační desce jinak by mohlo dojít k poškození čipu Cypress. Každá výměna CMOS senzoru musí proběhnout při vypnuté propojovací desce Cypress, při inicializaci se rozpozná typ připojeného senzoru a tuto informaci je pak možné vyčíst pomocí USB. K tomu je potřeba nainstalovat USB ovladače do počítače.

Instalace ovladačů CyUSB v systémech Windows lze provést bez vypnutí kontroly podpisu ovladačů. Aplikace je kompatibilní s podepsanými ovladači dodávanými firmou Cypress. Instalace ovladačů LibUSB je složitější, neboť je potřeba vypnout kontrolu podepsání

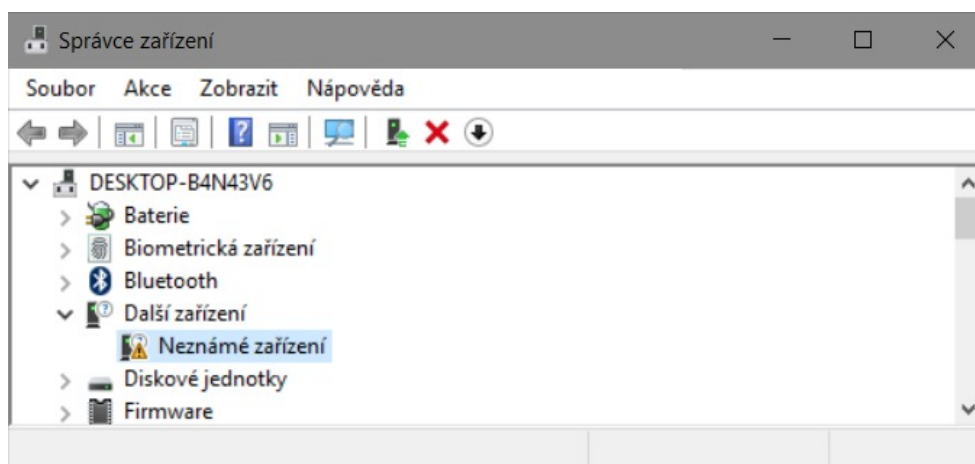
## 5.1 USPOŘÁDÁNÍ PRACOVNÍHO MÍSTA A PRVOTNÍ SPUŠTĚNÍ PROGRAMU

ovladačů. Zakázání podepsání ovladačů lze pomocí příkazů:

```
bcdedit /set loadoptions DISABLE_INTEGRITY_CHECKS  
bcdedit.exe /set NOINTEGRITYCHECKS ON
```

Tyto příkazy napsané do příkazové řádky vypne po restartu systému vynucení podepsání ovladačů a ovladače poskytnuty v příloze půjdou nainstalovat. Je možné využití programu [Batard 2017](#), který může pomoci s instalací potřebného ovladače LibUSB.

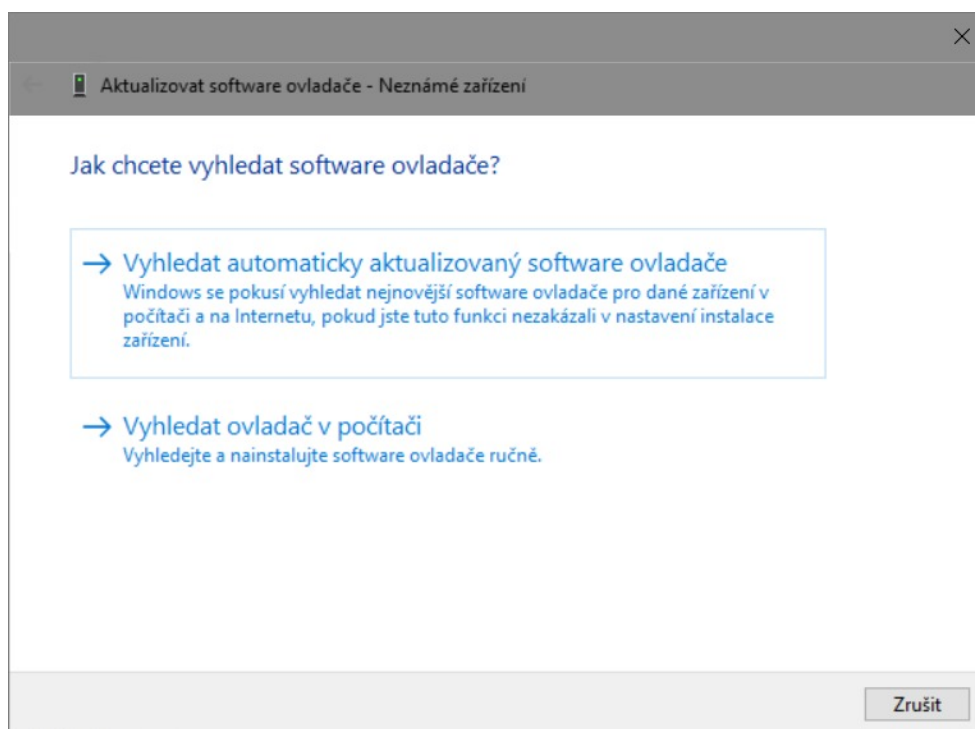
Při připojení desky Cypress se ve správci zařízení objeví neznámé zařízení jak je zobrazeno na obrázku 15.



Obrázek 15: Správce zařízení neznámé zařízení

Pro instalaci CyUSB ovladačů je nutné vybrat Vyhledat ovladač v počítači, jak je znázorněno na obrázku 16, automatické vyhledání ovladačů není možné použít. Po jednotlivých krocích lze pak sledovat úspěšné nainstalování ovladačů a je možné spustit měřicí aplikaci.

## 5.1 USPOŘÁDÁNÍ PRACOVNÍHO MÍSTĚ A PRVOTNÍ SPUŠTĚNÍ PROGRAMU



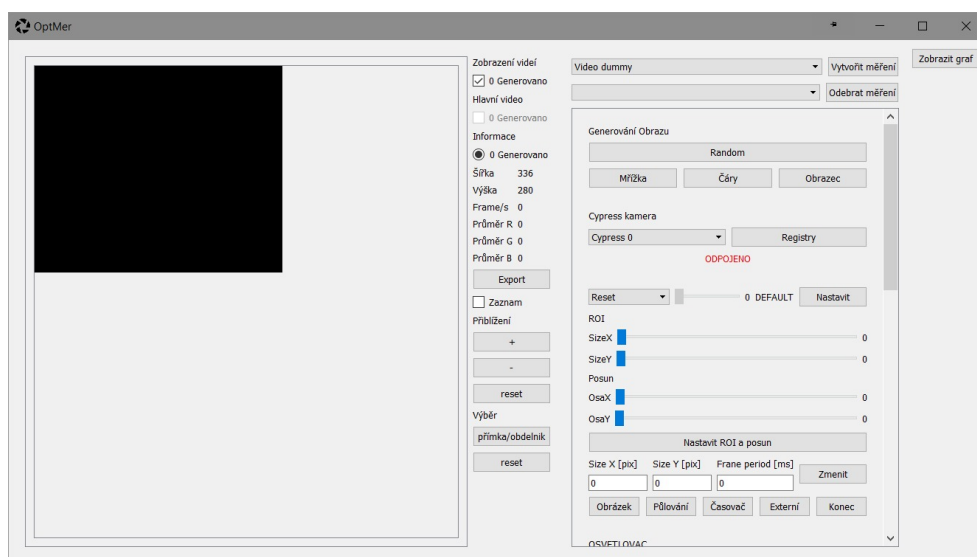
Obrázek 16: Instalace ovladačů

### ■ 5.1.1 Spuštění aplikace a připojení CMOS senzoru

Po spuštění měřicí aplikace se spustí hlavní okno a okno s grafy. Na hlavním oknu v levé části jsou zobrazeny přijatá nebo vygenerovaná obrazová data. Tyto obrazová data lze vypnout pomocí prostředního panelu, kde se také zobrazují základní informace o obrazových datech. V levé části je pak zobrazeny ovládací prvky pro připojení komunikační desky Cypress nebo pro zahájení měření. Toto okno lze sledovat na obrázku 17. Jak lze sledovat, při zapnutí aplikace je zobrazeno jedno okno s obrazovými daty, které patří ke generování obrazu. Pomocí tlačítek lze změnit zobrazená data a zkusit základní nastavení programu.

V levé části okna programu lze pak připojit ke komunikační desce Cypress, kde program provede automatické nastavení a zobrazí přijatá obrazová data v levé části programu. Také lze nastavit ROI a další parametry jako elektronickou závěrku nebo přeskočení zobrazených řádků a sloupců.

## 5.1 USPOŘÁDÁNÍ PRACOVNÍŠTĚ A PRVOTNÍ SPUŠTĚNÍ PROGRAMU



Obrázek 17: Hlavní okno aplikace

### 5.1.2 Měření základních parametrů a změna registru

Pro měření základních parametrů je potřeba umět načíst registry ze senzoru CMOS a umožnit jejich změnu. To se provede pomocí tlačítka Registry. Po zmáčknutí tlačítka se zobrazí okno, které bude mít prvním sloupci zobrazeno číslo registru, v druhém sloupci je zobrazena hodnota registru a ve třetím a čtvrtém sloupci lze měnit hodnotu registru. Třetí sloupec umožňuje změnu registru zápisem 16-bitového přenosu, čtvrtý sloupec umožňuje změnu registru zápisem 8-bitového přenosu dat. Oba zápisy dat jsou realizovány přes sběrnici I<sup>2</sup>C. Po zápisu dat se tento registr znovu načte a zobrazí se nová hodnota registru. Ukázka této tabulky je na obrázku 18.

	1	2	3	4
0 (0x0000)	4900 (0x1324)	-1	-1	
1 (0x0001)	1 (0x0001)	-1	-1	
2 (0x0002)	4 (0x0004)	-1	-1	
3 (0x0003)	480 (0x01E0)	-1	-1	
4 (0x0004)	752 (0x02F0)	-1	-1	
5 (0x0005)	94 (0x005E)	-1	-1	
6 (0x0006)	45 (0x002D)	-1	-1	
7 (0x0007)	380 (0x017C)	-1	-1	
8 (0x0008)	443 (0x01B3)	-1	-1	
9 (0x0009)	473 (0x01D9)	-1	-1	
10 (0x000A)	355 (0x0163)	-1	-1	
11 (0x000B)	400 (0x0190)	-1	-1	
12 (0x000C)	0 (0x0000)	-1	-1	
13 (0x000D)	768 (0x0300)	-1	-1	
14 (0x000E)	0 (0x0000)	-1	-1	
15 (0x000F)	256 (0x0100)	-1	-1	
16 (0x0010)	64 (0x0040)	-1	-1	
17 (0x0011)	32834 (0x8042)	-1	-1	
18 (0x0012)	34 (0x0022)	-1	-1	
19 (0x0013)	11570 (0x2D32)	-1	-1	
20 (0x0014)	3586 (0x0E02)	-1	-1	
21 (0x0015)	3634 (0x0E32)	-1	-1	
22 (0x0016)	10342 (0x2802)	-1	-1	
23 (0x0017)	19328 (0x4B30)	-1	-1	
24 (0x0018)	19328 (0x4B30)	-1	-1	
25 (0x0019)	10342 (0x2802)	-1	-1	

Obrázek 18: Výčet registrů

Některé senzory obsahují takzvané *shadow* registry, kde se změna registru projeví až

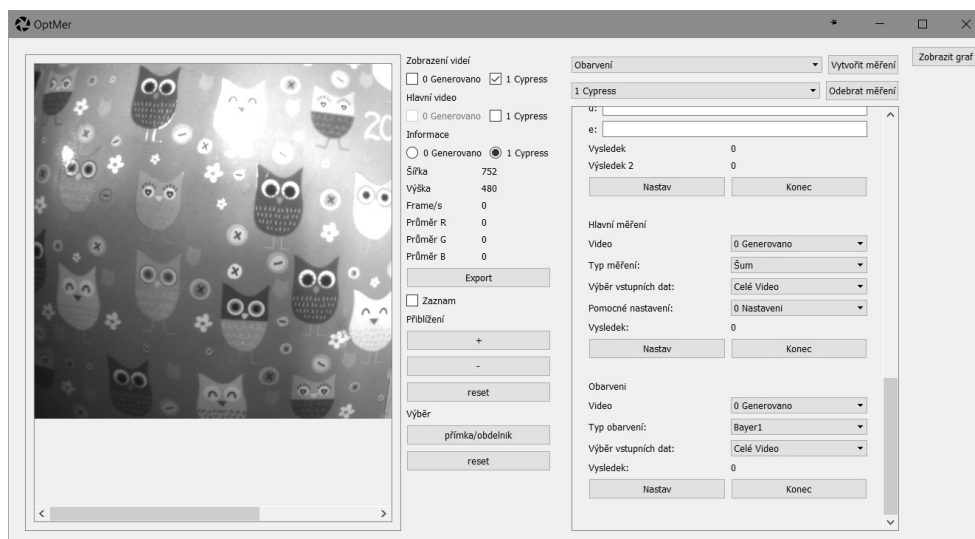


## 5.1 USPOŘÁDÁNÍ PRACOVNÍHO MÍSTA A PRVOTNÍ SPUŠTĚNÍ PROGRAMU

po předem definované události senzoru. Proto hodnota načtená ihned po zápisu může obsahovat ještě nezměněná data. Tuto vlastnost je nutné mít na mysli při využívání této tabulky.

### ■ 5.1.2.1 Měření šumu a linearity

Pro měření parametrů CMOS senzorů jsem vytvořil GUI Nastavení a GUI Hlavní měření. Tyto dva ovládací moduly jsou zobrazeny na obrázku 19, respektive na obrázku 20. Modul Nastavení slouží k postupnému nastavení registru, osvětlení, prahování, ROI, posunu obrazu CMOS pomocí registrů, nastavení měření a dalších možností. Lze použít při modulu Měření. Tento modul také podporuje postupnou změnu parametrů v závislosti na čase. Toho lze docílit pomocí řádku perioda opakování, který nastaví periodicky požadovanou hodnotu stanoveného času v milisekundách. Nastavovaná hodnota se může měnit o konstantní hodnotu nebo lineárně, kvadraticky, logaritmicky nebo lomenou funkcí. Vybraný výsledek je pak zobrazen ve spodní části modulu.



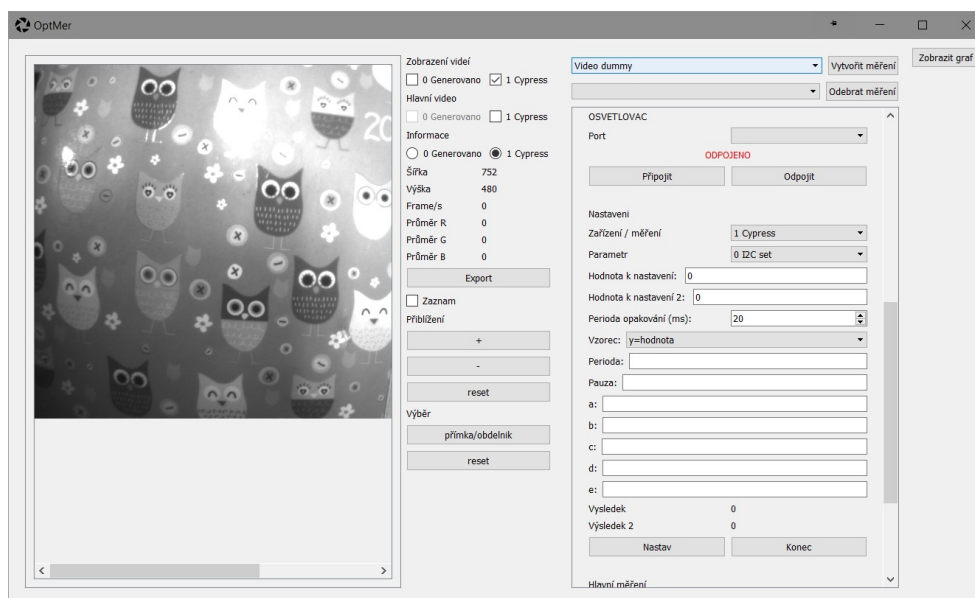
Obrázek 19: Měření závislostí

Modul měření je závislý na rychlosti počítače, typicky tedy nelze snímat celé video. Tento modul obsahuje měření šumu, linearity, jasu, histogram a obarvení. U všech typů měření lze nastavit výběr, pro které se toto měření má provést. Jmenovitě se jedná o tyto možnosti: celé video, bod, linka a obdélník. Vybraná zdrojová obrazová data mohou být zobrazena v levém okně hlavního programu. Výsledné grafy jsou zobrazeny ve vedlejším okně a pokud jsou výsledkem obrazová data, jsou zobrazena v levé části hlavního programu.

Měření šumu pouze převezme z výběru obrazových dat hodnoty a zobrazí je v časové závislosti v grafu. Měření linearity převezme z výběru obrazových dat hodnoty  $n$  vzorků ( $n$  je nastavitelné v modulu Nastavení), zobrazí jejich průměr v grafu a inkrementuje hodnotu (krok) v GUI Nastavení. Takto se provede pro  $m$  kroků, jež jsou také nastavitelné

## 5.2 NAMĚŘENÉ GRAFY A VÝSLEDKY Z MĚŘENÍ

v modulu Nastavení. Posledním možným měřením je měření jasu, které umožňuje zobrazení obrazových dat v grafu. Toho lze využít například při porovnání vedlejších pixelů.



Obrázek 20: Postupné nastavení měřících parametrů

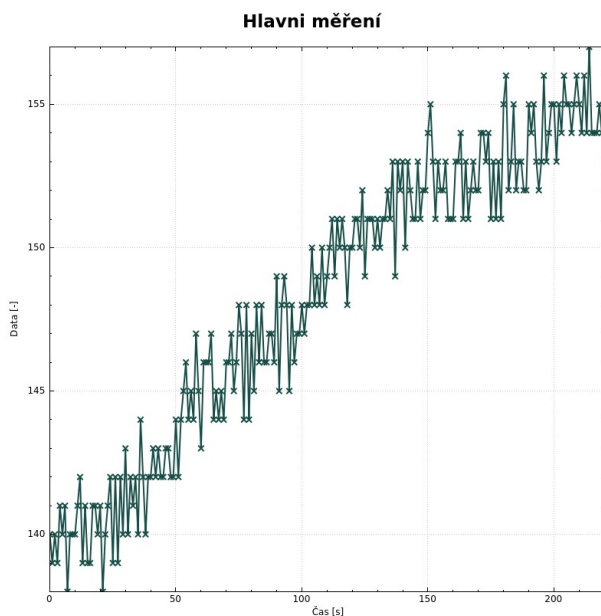
Program v neposlední řadě umožňuje obarvení černobílého obrazu pomocí bayer filtru. Toho lze využít pouze u některých senzorů, neboť ne všechny typy senzorů tuto vlastnost podporují. Pro potřeby uživatele je v programu možné i zobrazení histogramu. Ukázky naměřených dat jsou zobrazeny v následující kapitole.

## 5.2 Naměřené grafy a výsledky z měření

Pro ověření funkčnosti programu jsem změřil několik speciálních případů grafů vytvořených tímto programem. Při práci s grafy jsem využil měření linearity, šumu a jasu. Vyzkoušené obrazové senzory byly KAC-9618, KAC-9638 a MT9V034. U senzorů typu KAC byla měřena linearita a šum.

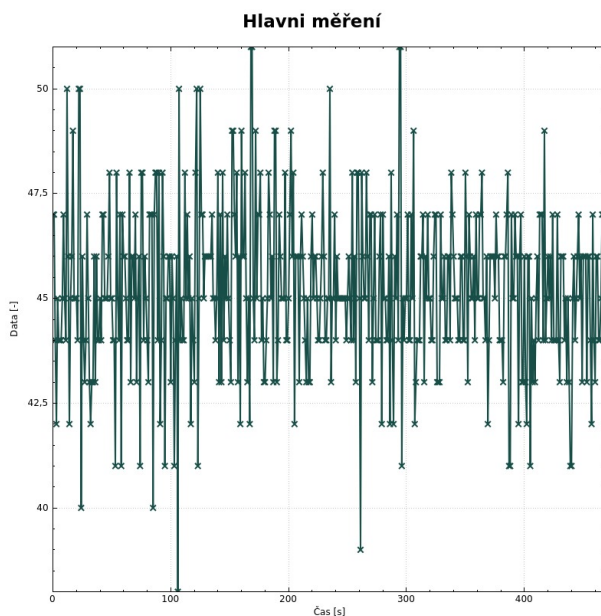
Na prvním obrázku 21 jsou znázorněna obrazová data senzoru KAC-9618 neboli jas s výběrem linky. Lze zde sledovat, že senzor nebyl osvětlen rovnoměrně a nabývá hodnot 138 až 157. Pokud by se před senzorem zobrazil Sayce graf (graf s postupným snižováním vzdáleností a šířky čar), mohli bychom dalšími výpočty změřit citlivost senzoru. Na tomto obrázku lze pozorovat, že osvětlení senzoru nebylo rovnoměrné ani za použití osvětlovacího zařízení. To může být zapříčiněno i nečistotami na objektivu.

## 5.2 NAMĚŘENÉ GRAFY A VÝSLEDKY Z MĚŘENÍ



Obrázek 21: Měření jasu s výběrem linky na senzoru KAC-9618

Ve druhém měření (obrázek 22) byl měřen šum jednoho bodu na senzoru MT9V034. Vzorek byl kvůli názornosti měřen na malém počtu dat. Měření probíhalo při stejném osvětlení senzoru. Lze sledovat, že i při stejném osvětlení senzoru CMOS se hodnota na jeho výstupu mění. Zde se jedná o rozsah přibližně 13, a to od 38 do 51.

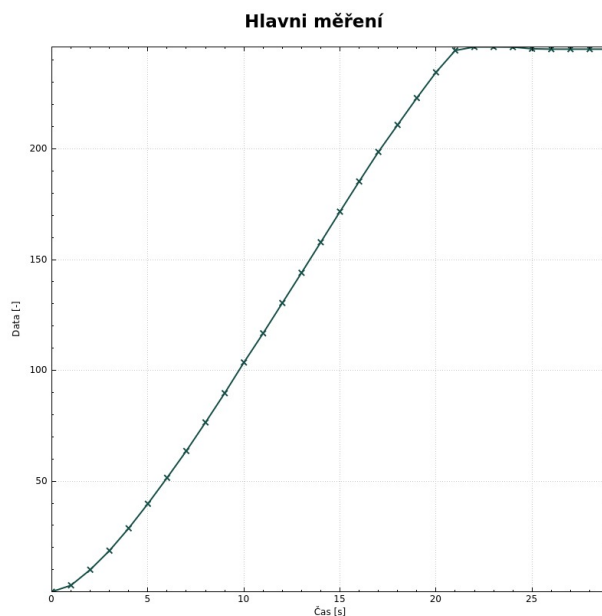


Obrázek 22: Měření šumu jednoho bodu na senzoru MT9V034

Na obrázku 23 lze pozorovat další měření, které spočívalo v postupném zvyšování jasu

## 5.2 NAMĚŘENÉ GRAFY A VÝSLEDKY Z MĚŘENÍ

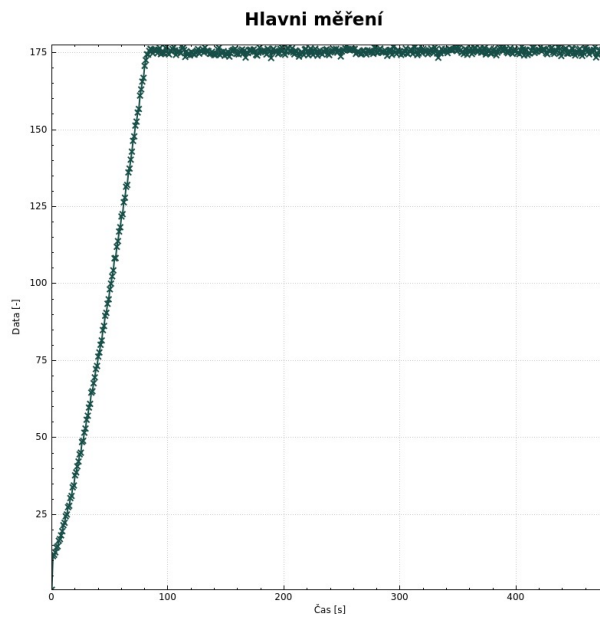
osvětlovacího zařízení a sbírání obrazových dat ze senzoru KAC-9618 z jednoho bodu. Šlo o průměr, který se skládal ze 20 vzorků. Měření bylo ukončeno po 30 krocích osvětlovacího zdroje. Tou dobou již senzor nabyl saturace.



Obrázek 23: Postupné zvyšování jasu jednoho bodu na senzoru KAC-9618

## 5.2 NAMĚŘENÉ GRAFY A VÝSLEDKY Z MĚŘENÍ

Poslední měření, viditelné na obrázku 24, probíhalo obdobně jako třetí měření, avšak byl použit senzor MT9V034. Ani při plném nastavení osvětlovacího zařízení se tento senzor nedostal do saturace. Z tohoto důvodu bylo měření vypnuto po 230 krocích, ve kterých se hodnota ustálila na 175. To je zapříčiněno tím, že je na senzoru zapnuto AGC (automatic gain control) a AEC (automatic exposure control).



Obrázek 24: Postupné zvyšování jasu jednoho bodu na senzoru MT9V034



## Kapitola 6

# Závěr

Tato práce se zabývala vyhotovením měřicího pracoviště pro laboratorní experimenty. Toto měřicí pracoviště by mělo sloužit k testování obrazových snímačů CMOS. Pracoviště se skládá ze senzorů CMOS, osvětlovacího zařízení, propojovací desky a počítače. Senzory CMOS byly poskytnuty vedoucím diplomové práce doc. Ing. Janem Fischerem, CSc. Byla využita komunikační deska s radičem Cypress, který využívá USB ke komunikaci s počítačem. K laboratorním experimentům je potřeba osvětlovací přípravek pro osvětlení CMOS senzorů, který se ovládá pomocí programu pro počítač. Existující hotové přípravky na měření byly rozšířeny o měření šumu, linearity a zobrazení dalších parametrů a vlastností.

Cílem práce tedy bylo vytvoření plnohodnotného měřicího pracoviště, které by využíval minimálně 6 druhů CMOS senzorů. Použity byly tyto CMOS senzory: KAC-9618, KAC-9638, MT9M001, MT9T031, MT9V032 a MT9V034. Dva CMOS senzory jsou od firmy KODAK, další senzor je od firmy Micron a poslední tři senzory jsou od firmy Aptina. Rozlišení se pohybuje od 648x488 px do 2048x1536 px s různou snímkovou frekvencí. Senzor MT9T031 je barevný senzor s Bayer maskou, ostatní CMOS senzory jsou černobílé. Všechny senzory lze nastavit pomocí sběrnice I<sup>2</sup>C zápisem do vnitřních registrů těchto senzorů.

Byla navržena a realizována experimentální kamera CMOS sloužící k laboratorním experimentům a testování obrazových vlastností CMOS. Na desce se nachází 6 různých druhů CMOS senzorů, jež byly úspěšně testovány. Byla vytvořena aplikace pro PC zajišťující veškeré funkce související s ovládáním senzorů, přenosem dat do PC a jejich zobrazením.

Výsledky diplomové práce budou využívány studenty na Fakultě elektrotechnické. Zde se budou moci seznámit s CMOS senzory, změřit si jejich základní parametry, vyzkoušet různá jejich nastavení.





## Literatura

- 007, Fyzika (2017). *Polarizace světla - Fyzika 007*. [online]. URL: <http://www.fyzika007.cz/optika/polarizace-svetla>.
- Aptina (2004). *Datasheet 1/2-Inch Megapixel CMOS Digital Image Sensor MT9M001*. [Datasheet].
- (2006). *Datasheet 1/2-Inch 3-Megapixel CMOS Digital Image Sensor MT9T031*. [Datasheet].
- (2008a). *Datasheet 1/3-Inch Wide-VGA CMOS Digital Image Sensor MT9V032*. [Datasheet].
- (2008b). *Datasheet 1/3-Inch Wide-VGA CMOS Digital Image Sensor MT9V034*. [Datasheet].
- Batard, Pete (2017). *Zadig - USB driver installation made easy*. [online]. URL: <http://zadig.akeo.ie/>.
- Cypress (2013). *EZ-USB® FX2LPTM USB Microcontroller High-Speed USB Peripheral Controller*. [Datasheet].
- Cypress, Semiconductor (2011). *Cypress CyAPI Programmer's Reference*. [online]. URL: <https://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwi2rre9wvrTAhXjD8AKHa8NA14QFggiMAA&url=http%3A%2F%2Fwww.cypress.com%2Ffile%2F53106&usq=AFQjCNHdUW1Uyuz1fNREQ2pmHEGfpg41dg&sig2=rp00exK4VgWF-ThFsSvjIA>.
- Douša, Petr (2014). *Experimentální sestava pro testování senzorů CMOS*. Praha: Bakalářská práce. České vysoké učení technické. Fakulta elektrotechnická.
- Fischer, Jan (2008). *Optoelektronické senzory a videometrie*. Skripta ČVUT - FEL. Praha.
- Kodak (2004a). *Datasheet KODAK KAC-9618 CMOS IMAGE SENSOR*. [Datasheet].
- (2004b). *Datasheet KODAK KAC-9638 CMOS IMAGE SENSOR*. [Datasheet].
- LibUSB (2017). *LibUSB*. [online]. URL: <http://libusb.info/>.
- NEC, Compaq DEC HP IBM Intel Microsoft a Nortel (2000). *Universal Serial Bus Specification*. [online]. URL: [http://www.usb.org/developers/docs/usb20\\_docs/usb\\_20\\_033017.zip](http://www.usb.org/developers/docs/usb20_docs/usb_20_033017.zip).
- Nord, Haavard a Eirik Chambe-Eng (2017a). *Qt — Cross-platform software development for embedded and desktop*. [online]. The Qt Company. URL: <https://www.qt.io/>.
- (2017b). *Qt Documentation*. [online]. The Qt Company. URL: <http://doc.qt.io/>.
- NXP, Semiconductors (2014). *I2C-bus specification and user manual*. [online]. URL: [http://cache.nxp.com/documents/user\\_manual/UM10204.pdf](http://cache.nxp.com/documents/user_manual/UM10204.pdf).
- Ondříčka, Matej (2014). *Rychlý vstup dat pre Raspberry PI*. Praha: Bakalářská práce. České vysoké učení technické. Fakulta elektrotechnická.

- Všetička, Jaroslav Reichl Martin (2017). *Fotometrické veličiny :: MEF*. [online]. URL: <http://fyzika.jreichl.com/main.article/view/535-fotometricke-veliciny>.
- Zima, Jiří (2009). *Rychlý přenos obrazu a dat s využitím EZ-USB*. Praha: Bakalářská práce. České vysoké učení technické. Fakulta elektrotechnická.
- Zoubek, Bc. Martin (2011a). *Driving Board for CMOS Image Sensor*. [scheme].
- (2011b). *Určování polohy pomocí řádkových obrazů*. Praha: Diplomová práce. České vysoké učení technické. Fakulta elektrotechnická.
- Zoubek, Martin (2009). *Optoelektronický senzor s řadičem EZ-USB*. Praha: Bakalářská práce. České vysoké učení technické. Fakulta elektrotechnická.