



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

**SOFTWARE PRO OVLÁDÁNÍ DSLR S VYUŽITÍM V
MIKROSKOPII**

SOFTWARE FOR DSLR CONTROL FOR USE IN MICROSCOPY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jakub Mokrejš

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vratislav Čmiel, Ph.D.

BRNO 2023

Diplomová práce

magisterský navazující studijní program **Bioinženýrství**

Ústav biomedicínského inženýrství

Student: Bc. Jakub Mokrejš

ID: 211297

Ročník: 2

Akademický rok: 2022/23

NÁZEV TÉMATU:

Software pro ovládání DSLR s využitím v mikroskopii

POKYNY PRO VYPRACOVÁNÍ:

1) Nastudujte, jakými parametry a funkcemi disponuje běžná kamera používaná v optické mikroskopii pro zobrazování ve světlém poli a porovnejte její vlastnosti s DSLR. 2) Seznamte se s uživatelským software používaným k obsluze mikroskopické kamery a popište základní funkce a možnosti, které nabízí. 3) Navrhněte programové vybavení s grafickým uživatelským rozhraním pro obsluhu DSLR. Programové vybavení musí zahrnovat minimálně funkce nastavení parametrů expozice, snímání základních a multi expozičních snímků s možností ukládání dat, zobrazení živého náhledu a zpracování HDR. 4) Realizujte navržené programové vybavení. 5) Zajistěte komunikaci s DSLR. K tomuto účelu využijte vývojové SW prostředí na PC a nebo vhodné vývojové mikroprocesorové zařízení (např. vývojový kit typu Raspberry Pi, ESP32) s dostupnými knihovnamy (např. gPhoto2). 6) Realizujte technické a softwarové řešení, které umožní efektivní ovládání DSLR jako mikroskopické kamery, základní zpracování, zobrazení a ukládání mikroskopických snímků. 7) Ověřte funkčnost realizovaného technického řešení a výsledky diskutujte.

DOPORUČENÁ LITERATURA:

- [1] OGAMA, T.: What to Consider When Choosing a Microscope Camera [online]. Dostupné online z: <https://www.olympus-lifescience.com/en/resources/white-papers/choosing-a-microscope-camera/>
- [2] KEAL, L.: What is Scientific Imaging Quality? [online]. Dostupné z: <https://www.photometrics.com/wpcontent/uploads/2021/01/Scientific-Image-Quality-A3-19-11-2020.pdf>

Termín zadání: 6.2.2023

Termín odevzdání: 22.5.2023

Vedoucí práce: Ing. Vratislav Čmiel, Ph.D.

doc. Ing. Radim Kolář, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor semestrální práce nesmí při vytváření semestrální práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato diplomová práce se nejprve zabývá rešerší kamer využívaných v mikroskopii, popisem jejich vlastností a rozdílem mezi mikroskopickými a DSLR kamerami. Dále jsou popsány programy využívané mikroskopickými kamerami pro ovládání přes počítač. Práce poté definuje technologie, které se využívají pro vývoj software a představuje návrh programového vybavení software. Praktická část práce nejprve popisuje technologie využitě pro vývoj software společně s vývojovým diagramem. V další části práce jsou popsány funkce, které software využívá ke komunikaci s DSLR kamerou a je představeno grafické uživatelské rozhraní software. Závěr praktické části se věnuje testování samotného software, kde zároveň představuje různé způsoby jeho ovládání. Výsledky testování jsou poté zhodnoceny v diskusi.

Abstract

This master's thesis initially deals with a review of cameras used in microscopy, describing their properties and the difference between microscopic and DSLR cameras. Furthermore, it describes the programs used by microscopic cameras for control via a computer. The thesis then defines the technologies used for software development and presents a design of the software's program equipment. The practical part of the thesis first describes the technologies used for software development along with a development diagram. The next section describes the functions that the software uses to communicate with the DSLR camera and introduces the software's graphical user interface. The conclusion of the practical part is devoted to testing the software itself, where it also presents various ways of its control. The test results are then evaluated in the discussion.

Klíčová slova

CCD, CMOS, sensor, DSLR, software, GUI, vývoj software, Python, ovládání kamery, gPhoto2, PyCharm, funkce, mikroskopie

Keywords

CCD, CMOS, sensor, DSLR, software, GUI, software development, Python, camera control, gPhoto2, PyCharm, functions, microscopy

Bibliografická citace

MOKREJŠ, Jakub. *Software pro ovládání DSLR s využitím v mikroskopii*. Brno, 2023. 60 stran. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/150880>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství. Vedoucí práce Vratislav Čmiel.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	<i>Jakub Mokrejš</i>
VUT ID studenta:	<i>211297</i>
Typ práce:	<i>Diplomová práce</i>
Akademický rok:	<i>2022/23</i>
Téma závěrečné práce:	<i>Software pro ovládání DSLR s využitím v mikroskopii</i>

Prohlašuji, že svou semestrální práci jsem vypracoval samostatně pod vedením vedoucího semestrální práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené semestrální práce dále prohlašuji, že v souvislosti s vytvořením této semestrální práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

**Jakub
Mokrejš**

V Brně dne: 22. května 2023

podpis autora

Obsah

SEZNAM OBRÁZKŮ	7
SEZNAM TABULEK.....	8
ÚVOD	9
1. KAMERY POUŽÍVANÉ V MIKROSKOPII.....	10
1.1 VÝHODY VYUŽITÍ KAMER V MIKROSKOPII	10
1.2 CCD A CMOS OBRAZOVÉ SENSORY PRO MIKROSKOPII.....	10
1.3 PARAMETRY OVLIVŇUJÍCÍ KVALITU OBRAZU	11
1.4 REÁLNÉ PARAMETRY MIKROSKOPICKÝCH A DSLR KAMER	15
2. SOFTWARE PRO OVLÁDÁNÍ KAMER.....	18
2.1 ŽIVÉ ZOBRAZENÍ, MOŽNOSTI SNÍMÁNÍ A ULOŽENÍ SNÍMKU	18
2.2 NASTAVENÍ	19
2.3 AKVIZICE	19
2.4 ZPRACOVÁNÍ OBRAZU	20
2.5 ANALÝZA	20
3. TECHNOLOGIE PRO VÝVOJ SOFTWARE.....	22
3.1 BACK-END.....	22
3.1.1 <i>Python</i>	22
3.1.2 <i>DigiCam</i>	22
3.1.3 <i>gPhoto2</i>	22
3.2 FRONT-END	22
3.2.1 <i>PyQt5</i>	23
3.2.2 <i>Tkinter</i>	23
3.2.3 <i>Kivy</i>	23
3.2.4 <i>WxPython</i>	23
3.2.5 <i>Figma</i>	23
3.3 VÝVOJOVÉ PROSTŘEDÍ PYCHARM.....	24
4. NÁVRH PROGRAMOVÉHO VYBAVENÍ.....	25
4.1 GRAFICKÉ UŽIVATELSKÉ ROZHRAŇÍ SOFTWARE	25
4.1.1 <i>Blok pro volbu kamery a místa uložení snímku</i>	25
4.1.2 <i>Blok pro nastavení parametrů pro snímání</i>	25
4.1.3 <i>Blok pro pojmenování snímku a spouštění snímání</i>	26
5. PRAKTICKÁ ČÁST.....	27
5.1 POPIS SOFTWARE.....	27
5.1.1 <i>Ovládání kamery pomocí gphoto2 a Python</i>	27
5.2 VÝVOJOVÝ DIAGRAM SOFTWARE	27
5.3 FUNKCE PRO KOMUNIKACI S KAMEROU.....	28
5.3.1 <i>Funkce pro detekci kamery detect_camera()</i>	30
5.3.2 <i>Funkce pro získání hodnot konkrétního nastavení get_settings()</i>	30
5.3.3 <i>Funkce pro zjištění aktuálního nastavení kamery current_settings()</i>	31
5.3.4 <i>Funkce pro nastavení konkrétní hodnoty DSLR kamery set_settings()</i>	32
5.3.5 <i>Funkce pro pořízení snímku take_picture()</i>	32

5.3.6	<i>Funkce k zobrazení živého zobrazení live_view()</i>	32
5.4	GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ SOFTWARE	33
6.	TESTOVÁNÍ	35
6.1	PŘÍPRAVA PROSTŘEDÍ PRO SPUŠTĚNÍ SOFTWARE	35
6.2	DEFINICE POSTUPU TESTOVÁNÍ	36
6.3	MOŽNOSTI OVLÁDÁNÍ SOFTWARE	37
6.3.1	<i>Ovládání přes počítač s operačním systémem Linux</i>	38
6.3.2	<i>Ovládání přes Raspberry Pi</i>	43
6.3.3	<i>Vzdálené ovládání přes Raspberry Pi</i>	46
7.	DISKUSE	50
8.	ZÁVĚR	53
	LITERATURA	54
	SEZNAM SYMBOLŮ A ZKRATEK	57
	SEZNAM PŘÍLOH	58

SEZNAM OBRÁZKŮ

1	Schéma znázorňující rozdíl mezi designem předního a zadního osvětlení senzorů. Převzato a upraveno z [5].	12
2	Křivky spektrální odezvy ukazující kvantovou účinnost v závislosti na vlnové délce dopadajícího fotonu pro řadu kamerových senzorů sCMOS. Převzato a upraveno z [5].	13
3	Kvantová účinnost jednotlivých barevných kanálů kamery Kiralux CS505CU. Převzato a upraveno z [5].	16
4	Leica LAS X Software pro mikroskopickou kameru. Převzato z [11].	19
5	Program LAS EZ pro mikroskopickou kameru. Převzato z [13].	21
6	Ukázka vývojového prostředí PyCharm. Převzato z [23].	24
7	Návrh rozložení grafického uživatelského rozhraní.	26
8	Diagram základního fungování aplikace.	28
9	Ukázka grafického uživatelského rozhraní.	34
10	Okno, které se zobrazí v případě správně nastaveného prostředí.	36
11	Postup testování software.	37
12	Blokové schéma komunikace s DSLR kamerou.	38
13	Výběr úložiště fotografií s vypsáním do GUI.	39
14	Ukázka vypsání obsahu složky po akvizici snímku.	39
15	Vyskakovací okno živého zobrazení.	40
16	Ukázka vyskakovacího okna pro pořízení HDR snímku a nastavení hodnot – Linux PC.	41
17	Snímky s odlišnou hodnotou expozice pořízené pro zpracování HDR fotografie.	41
18	Výsledný HDR snímek.	42
19	Nastavení kamery pro snímání s časovačem a obsah složky.	43
20	Ukázka GUI na mikropočítači Raspberry Pi.	44
21	Nastavení kamery před sekvenčním snímáním.	45
22	Snímky pořízené sekvenčním snímáním.	45
23	Snímek pořízený módem „Night Portrait“.	46
24	Povolení VNC na Raspberry Pi.	47
25	Výpis v příkazovém řádku po zadání příkazu „ifconfig“. Obrázek byl barevně invertován.	47
26	Ukázka nastavení parametrů v programu VNC Viewer.	48
27	Ukázka zřízeného přenosu souborů.	48
28	Proces přenosu souborů pomocí VNC Viewer.	49
29	Dialogové okno s chybovou hláškou.	51

SEZNAM TABULEK

1	Vybrané parametry dvou zvolených mikroskopických kamer.	16
2	Parametry dvou vybraných DSLR kamer.....	17
1.3	Technické specifikace počítače využívaného pro testování	38
1.4	Porovnání časů úkonů pro testované způsoby ovládání	50

ÚVOD

Mikroskopie v dnešní době využívá kamery každý den. Díky pokroku technologie obsluha již může pouze vkládat vzorek pod mikroskop a nastavení mikroskopu řídit přes počítač. Zobrazení vzorku je zajištěno díky kamerám, které jsou buď zabudovány nebo připevněny k mikroskopu. Dedikované mikroskopické kamery dodávané výrobcem jako Andor Technology nebo Thorlabs poskytují ke kameře ovládací software, který umožňuje rychlou změnu parametrů, živý náhled a pořizování snímků nebo videa. Kromě dedikovaných mikroskopických kamer lze využít pro snímání vzorku například mobilní telefon nebo DSLR kameru. Tyto zařízení je však potřeba ovládat přímo a ne vzdáleně, což může mít negativní vliv jak na výsledný obraz, tak na samotný mikroskop.

Využití DSLR nebo speciálních mikroskopických kamer má svoje výhody i nevýhody, které se zakládají především na vlastnostech těchto kamer a jejich parametrech. Jednou z nevýhod DSLR kamer je nutnost manuálního nastavování parametrů kamery přímo na těle. Umožnění vzdáleného ovládání DSLR kamery by tak obsluhu nejen zjednodušilo, ale také zefektivnilo práci.

Tématem diplomové práce je Software pro ovládání DSLR s využitím v mikroskopii. V teoretická část práce se zabývá kamerami, které se využívají v mikroskopii, popisuje jejich základní vlastnosti a reálné parametry které ovlivňují výslednou kvalitu obrazu. Dále porovnává parametry mikroskopických a DSLR kamera popisuje zásadní funkce software od různých výrobců, které se používají pro ovládání mikroskopických kamer. V dalších kapitolách práce definuje technologie, které lze využít k vývoji software pro ovládání DSLR kamer a popisuje návrh grafického uživatelského zobrazení, s funkcemi, kterými by měl být tento software vybaven.

Praktická část práce definuje technologie, se kterými se bude pracovat při vývoji a představuje vývojový diagram software. Na tomto základě jsou realizovány funkce pro ovládání software, které využívají definovaných technologií pro komunikaci s DSLR kamerou. Po realizaci grafického uživatelského rozhraní software je provedeno ověření funkčnosti pomocí testování, které zahrnuje tři možné metody pro ovládání software. Výsledky a pozorování z testování jsou uvedeny v diskuzi.

Práce je rozdělena na sedm kapitol. První čtyři kapitoly spadají do teoretické části práce. Pátá kapitola začíná praktickou část a poslední sedmá kapitola diskutuje výsledky praktické části práce.

1. KAMERY POUŽÍVANÉ V MIKROSKOPII

Kamery se v mikroskopii využívají pro zobrazování, nahrávání, zaznamenávání a mohou být také efektivně využity k výuce a živým ukázkám. Díky novým technologiím jsou nyní kamery téměř v každém zařízení, mikroskopie však pracuje s velkým zvětšením, a tak ne všechny kamery jsou pro toto zobrazení vhodné. Některé typy mikroskopů již obsahují zabudované kamery, ostatní typy využívají různé způsoby připojení externí kamery.

Jedním ze základních typů je okulárová kamera, kterou lze připojit přímo k okuláru mikroskopu a přes USB k počítači. Další možností je mikroskopická kamera se speciálním snímačem, která oproti okulárové kameře disponuje větší kvalitou obrazu díky více megapixelům. Pro použití s mikroskopem však tyto kamery vyžadují speciální c-mount adaptér. K mikroskopu lze díky držáku použít i mobilní telefon, což vzhledem k tomu, že nyní již každý jeden vlastní, popsat jako nejjednodušší způsob pořízení snímku z mikroskopu. Posledním typem je digitální zrcadlová kamera (DSLR), která umožňuje zachycení videa i fotek v té největší kvalitě. Pro snímání na mikroskopu však vyžaduje hned dva adaptéry – c-mount adaptér pro mikroskop a specifický adaptér pro kameru.

1.1 Výhody využití kamer v mikroskopii

Mimo zobrazování a zaznamenávání, které je pro mikroskopii zásadní mají kamery i jiné výhody. Jednou z nich je to, že napomáhají od namáhání očí, které se projevuje při dlouhodobém pozorování přes okulár. Dlouhé pozorování přes okulár vede také k naklánění se nad mikroskop a tím i k zhoršenému postoji. Připojením kamery k mikroskopu je tak umožněno pozorování přes displej mobilu nebo počítače což dokáže ulevit od namáhání očí nebo držení špatného postoje a ergonomiky [1].

1.2 CCD a CMOS obrazové sensory pro mikroskopii

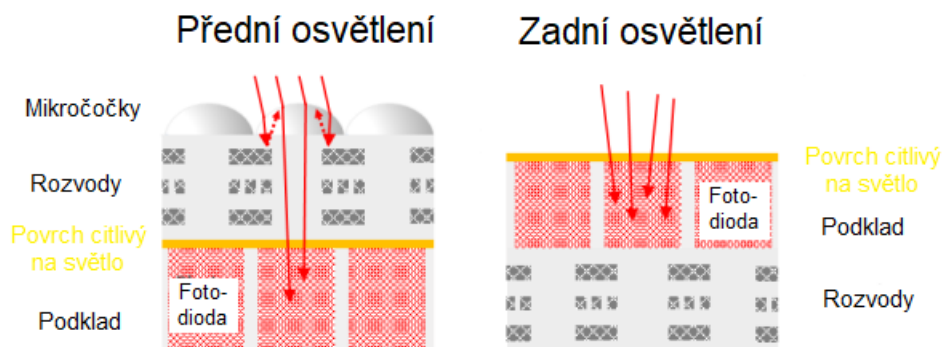
Vysoká kvalita obrazu je pro hodnocení detailů v mikroskopii nezbytná. Kamery v dnešní době využívají dva typy sensorů, které k tomuto dopomáhají: CCD senzor (Charge-Coupled Device – senzor obsahující vázané náboje) a CMOS senzor (Complementary Metal-Oxide Semiconductor). Tyto senzory pracují s principem fotoelektrického jevu, který lze zjednodušeně popsat jako převod světelné energie na elektrickou. Sensor složený z kondenzátorů poskládaných do matice po vystavení světlu začne na kondenzátorech hromadit náboj. Čím více světla na snímač dopadne, tím více náboje se naakumuluje. Výhody a nevýhody těchto dvou technologií tak vychází ze způsobu, jak zachází se signálem, který se naakumuluje v daném pixelu. Technologie

CCD postupně po řádcích zpracuje náboje v každém pixelu a převede je na analogový signál. Tento signál se mimo sensor zesílí a pomocí A/D převodníku se jeho hodnota převede do digitální podoby a uloží do paměti. Oproti technologii CCD dokáže CMOS provést převod na analogovou hodnotu přímo v každém pixelu na sensoru. Poté je tato hodnota poslána odděleně do A/D převodníku, kde je převedena na digitální hodnotu. Hlavní výhodou CMOS technologie tak je rychlejší převod hodnot a díky tomu i větší snímkovací frekvence [2]. Oba sensory jsou monochromatické, což znamená, že získaný obraz je černo-bílý. Pro získání barevného obrazu lze použít filtry, přes které na snímače dopadá světlo. Další možností je využití průhledného miniaturního filtru, který je uložen v mozaikovém vzoru přes pixely.

Další důležitou vlastností, která rozlišuje tyto dva sensory je jejich energetická náročnost. CCD sensory jsou složeny z analogových součástek, které jsou nejen dražší, ale také spotřebují až stokrát větší energii, než je tomu u sensorů CMOS. Toto automaticky znamená nejen větší spotřebu, ale také možné problémy s přehříváním kamery, což může ovlivnit jak kvalitu obrazu, tak zvýšené náklady na provoz kamery. Technologie CMOS navíc oproti CCD umožňuje velmi krátkou dobu expozice i v módu sekvenčního snímání. CMOS sensory obsahují navíc oproti CCD konverzní elektroniku, díky které má tento sensor více funkcí. Tyto obvody navíc ale mohou způsobovat šum v podobě pruhů, nebo jiných vzorů. CCD sensory byly vyvinuty speciálně pro kamery a jsou využívány již více jak třicet let. Ve srovnání s CMOS sensory měly dříve CCD sensory více výhod, jako světelnou citlivost nebo menší šum [3]. Tyto rozdíly se díky vývoji CMOS technologie smazaly a dnes se je již vývoj CCD sensorů téměř zastaven. Všechny odvětví průmyslu tak již přechází na sensory s technologií CMOS, které již předčily kvality sensorů s technologií CCD a budou se dále vyvíjet.

1.3 Parametry ovlivňující kvalitu obrazu

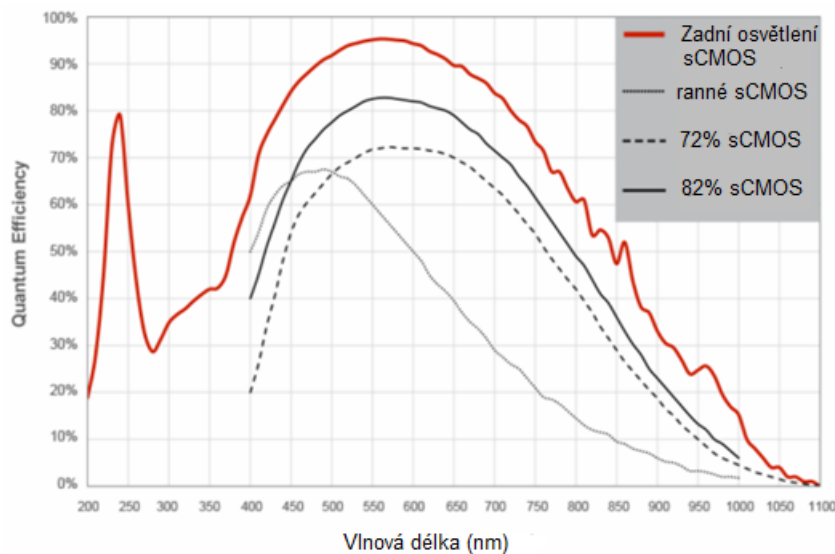
Jedním ze základních faktorů, které ovlivňují výslednou kvalitu obrazu je schopnost sběru světla (light collection). Následující vlastnosti udávají účinnost sběru světla. První z nich je kvantová účinnost QE (Quantum efficiency), která měří, jak efektivně je zobrazovacího zařízení schopno přeměnit dopadené fotony na elektrony. Hodnota QE je nejvíce ovlivněna tím, zda je využito designu předního osvětlení sensorů, nebo zadního osvětlení sensorů. Design předního osvětlení je sice jednodušší na výrobu, ale je schopen ztratit až 25 % světla. Tyto ztráty jsou dány hlavně tím, že mezi světlo citlivou křemíkovou vrstvou a mikročočkami pro usměrnění fotonů dále leží elektronika a kabeláž pro ovládaní sensorů viz. Obrázek 1. U designu zadního osvětlení je sensor převrácený což umožní fotonům se dostat přímo na světlo citlivý křemík bez nutnosti usměrnění mikročočkami. Pro dosažení nejlepší hodnoty QE je zapotřebí aby byl u tohoto designu ztenčen křemík na světlo citlivou oblast o tloušťce $\sim 1,1 \mu\text{m}$.



Obrázek 1 Schéma znázorňující rozdíl mezi designem předního a zadního osvětlení senzorů. Převzato a upraveno z [5].

V dnešní době se hodnoty QE u nejmodernějších vědeckých kamer, které využívají designu zadního osvětlení pohybuje okolo 95 %. Tato hodnota je ale závislá na vlnové délce detekovaného světla. Obrázek 2 ukazuje jak v červené a fialové oblasti viditelného spektra má snímač nižší hodnoty QE, zejména pokud jde o kamery s designem předního osvětlení sensorů. Pokud má tedy zařízení hodnotu kvantové účinnosti 1 (100 %), znamená to, že dokáže přeměnit všechny dopadené fotony na elektrony. Vysoká hodnota kvantové účinnosti je tak pro kvalitu výsledného obrazu velmi důležitá, protože čím více fotonů dokáže sensor nashromáždit, tím větší může být kvalita obrazu [4].

Další důležitou vlastností při určování účinnosti sběru světla je velikost pixelu neboli rozteč pixelů. Tento pojem určuje rozměr jednoho pixelu na sensoru. S narůstajícími rozměry pixelu narůstá i počet fotonů, který je jeden pixel schopný nashromáždit. Zdvojnásobení velikosti pixelu vede až k 4x většímu počtu zachycených fotonů, z čehož vyplývá, že zvětšení pixelu může být velmi účinná metoda, jak vylepšit citlivost kamery. Citlivost kamery není však udávána jen pomocí velikostí pixelu, i když má na samotnou citlivost větší vliv, ale také svojí hodnotou kvantové účinnosti [5]. Autoři [6] uvádí, že změnou velikosti pixelu z 5,5 μm na 6,5 μm naroste citlivost asi o 40 %. Pokud však změníme hodnotu kvantové účinnosti z 75 % na 90 % pozorujeme vzrůst citlivost o 20 %. Pokud tedy není rozlišení omezeno optickým systémem, může zvětšení velikosti pixelu negativně ovlivnit kvalitu rozlišení.



Obrázek 2 Křivky spektrální odezvy ukazující kvantovou účinnost v závislosti na vlnové délce dopadajícího fotonu pro řadu kamerových senzorů sCMOS. Převzato a upraveno z [5].

Poslední vlastností, která ovlivňuje schopnost sběru světla je „full well capacity“, což se dá přeložit jako maximální kapacita pixelu (nádrže, studny). Tato vlastnost definuje to, jaké množství náboje může jednotlivý pixel pojmout před tím, než saturuje. Kamery využívané ve výzkumu používají přesnější termín „linear full well capacity“. Tento termín referuje na skutečnost, že když se kapacita pixelu blíží maximu, tak přítomnost tolika fotoelektronů vede k vyloučení náboje, který ovlivňuje schopnost pixelu sbírat další fotoelektrony, což ovlivňuje linearitu. Ztráta linearit je pro použití ve výzkumu nepřijatelná. Velká maximální kapacita pixelu je důležitou vlastností u zobrazovacích technik, které vyžadují velký dynamický rozsah nebo pro zlepšení poměru signálu k šumu u technik s vysokou úrovní osvětlení. Naopak pro techniku jako je fluorescenční mikroskopie, která vyžaduje nízkou úroveň osvětlení není maximální kapacita studny důležitou vlastností [5].

Poměr signálu k šumu (SNR) je podle autorů [6] zásadním kvantitativním ukazatelem kvality obrazu, který zahrnuje jak sběr signálu, tak i šumovou výkonnost kamery. Vysoká hodnota SNR ukazuje, že obraz je bez šumu, který způsobuje zkreslení a artefakty. Čím vyšší je SNR, tím lepší je kvalita snímku a s tím se zlepšuje schopnost vidět požadované výsledky. Za dostatečného světla je vysoký poměr signálu k šumu lehce dosažitelný, ale v praxi se projevují technické i fyzické nedostatky. Zásadní fyzická limitace je způsobena kvůli statistické chybě v počtu elektronů generovaných na

čipu senzoru. Tato statistická chyba je určena jasnou vzorku a výše zmiňovanou citlivostí fotoaparátu. Technické limitace jsou způsobeny především temným proudem (dark current), který teče přes světlo citlivé zařízení, když do něj aktivně nevstupují fotony což se u moderních kamer projeví pouze pokud je expozice delší jak 2 sekundy. Dále se technické limitace týkají čtecího šumu (readout noise) a citlivosti, které jsou naopak od temného proudu klíčovými faktory pro zlepšení SNR při krátkém expozičním času.

Snímkovací frekvence udává počet zachycených snímků za sekundu a je důležitou vlastností pro plynulé živé zobrazování. Vysoko-rychlostní rozhraní jako je USB 3.0 společně s technologií CMOS umožnili většině kamerám snímkovací frekvenci větší než 30 snímků za sekundu (fps – frames per second). V mnoha případech jako je například zobrazování rychlých biologických jevů nebo výpočetní zobrazování včetně super rozlišení založeného na zpracování obrazu je však zapotřebí větší snímkovací frekvence než 30 fps [6]. Technologie CMOS má díky designu paralelního snímání výhodu oproti CCD což jí zaručuje větší maximální snímkovací frekvenci. Obě technologie však s větší snímkovací frekvencí zkracují dobu expozice díky čemuž se zmenšuje poměr signálu k šumu a s tím i kvalita obrazu [7]. Jako řešení pro zmenšenou hodnotu šumu a kvality obrazu se využívá různých technik pro zpracování obrazu.

Další důležitý aspekt, který ovlivňuje kvalitu obrazu je dynamický rozsah kamery, který udává kolik odlišných a samostatných úrovní šedi je kamera schopna detekovat a uložit. Dá se tedy říci, že dynamický rozsah detekuje nejnižší možnou úroveň světla, kterou je sensor schopný zachytit a kolik fotonů zachytí předtím než saturuje. Díky velkému rozdílu mezi těmito hodnotami je jednodušší rozeznat jemné rozdíly ve světle, které přichází z různé oblasti vzorku. S velkým dynamickým rozsahem kamery tak lze lépe detekovat malé rozdíly ve stínování v rámci vzorku. Hodnota dynamického rozsahu se udává jako maximální kapacita pixelu dělena čtecím šumem (read noise) a uvádí se často jako poměr – 3000:1 znamená, rozsah od 1 (černá) po 3000 (bílá). Při různých snímkovacích frekvencích se mění i dynamický rozsah, takže je nutné při výběru kamery zvážit i potřebnou snímkovací frekvenci pro aplikaci a jak ovlivňuje dynamický rozsah kamery.

Pro kvalitu výsledného obrazu je dále důležitý i zvolený režim a čas uzávěrky senzoru. Autoři [7] zmiňují globální režim a rolovací režim. Tyto režimy mají velký vliv na kvalitu obrazu, pokud se snímáný objekt pohybuje. V režimu rolovací uzávěrky jsou pixely vystaveny světlu po řádcích, což může mít za následek, že objekt mezi zachycením jednotlivých řádků změnil svou polohu. To pak ve výsledném obraze způsobí prostorové zkreslení. Výhodou je pak zejména nižší šum při čtení, díky menšímu počtu elektronických komponent. Režim globální uzávěrky umí odkrýt pixely ve stejnou dobu a předejít tak zkreslení obrazu při pohybu objektu. Zatímco sensory

s technologií CMOS jsou dostupné jak v režimu globální i rolovací uzávěrky, CCD sensory umí pouze globální režim.

Zpracování obrazu je používáno pro některé aplikace k tomu, aby překonalo optická a fyzická omezení. Například techniku obrazu s rozšířeným ohniskem (EFI) lze použít zejména pomocí stereomikroskopů k získání silného vzorku v jednom snímku. Další technikou může být zobrazování s vysokým dynamickým rozsahem (HDR), které se díky své schopnosti zachytit reflexní vzorky hojně využívá pro průmyslovou kontrolu. Jiná technika jako je například automatické více snímkové průměrování se zaměřuje na zlepšení SNR v živém fluorescentním snímání, zatímco se udržuje vysoký počet snímků [6].

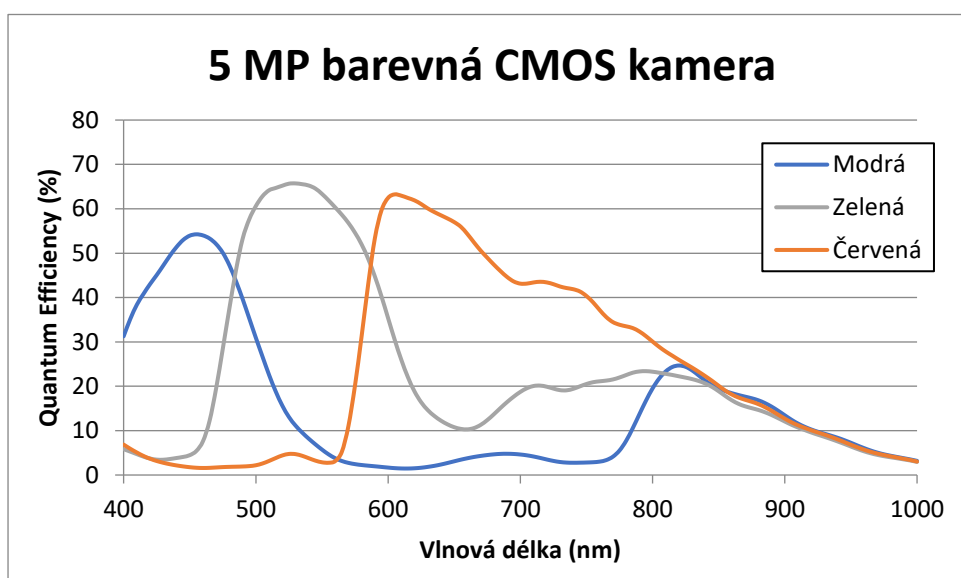
Kvalita obrazu ve fluorescentním snímání je závislá na správném nalezení rovnováhy mezi určitými vlastnostmi senzoru společně s vlastností samotné kamery a oblastí využití ve vědě, medicíně nebo průmyslu. Podle autorů [7] vyžaduje fluorescentní snímání vysoké hodnoty citlivosti, SNR a QE, vysokou kapacitu pixelu a zejména široký dynamický rozsah. Tyto vlastnosti nejlépe splňuje monochromatický CMOS sensor. Tyto sensory nevyužívají barevné filtry na senzoru a díky tomu tak k senzoru dopadne více světla. Proto je zásadní velká citlivost senzoru společně s vysokou kapacitou pixelu a širokým dynamickým rozsahem. Zajištění kvalitního fluorescentního snímku je tak závislé na výše uvedených vlastnostech.

1.4 Reálné parametry mikroskopických a DSLR kamer

Vybrané reálné parametry mikroskopických kamer od výrobců Andor a Kiralux jsou v tabulce 1. Oba výrobci používají sensor s technologií CMOS, přičemž kamera Andor Zyla 5.5 specifikuje využití předního osvětlení. Kamera Kiralux má 2x menší sensor, což se projevuje i na velikosti pixelu a kapacitě pixelu. Malý čtecí šum u kamery Andor Zyla dovoluje kameře pracovat s vyšším dynamickým rozsahem oproti druhé porovnávané kameře. Obě kamery mají podobnou hodnotu QE. Výrobce Kiralux poskytuje ke kameře podrobný graf, který lze vidět na Obrázku 3. Tento graf znázorňuje hodnoty QE jednotlivých barevných kanálů v závislosti na vlnové délce. Z grafu je tak vidět, že největší hodnotu QE má zelený kanál při vlnové délce kolem 535 nm. Červený kanál má největší hodnotu QE 62 % při vlnové délce 610 nm a nejmenší hodnotu má modrý kanál, který se dostane pouze na hodnotu 54 % QE při vlnové délce kolem 470 nm.

Tabulka 1 Vybrané parametry dvou zvolených mikroskopických kamer.

Model	Andor Zyla 5.5	Kiralux CS505CU
Typ sensoru	sCMOS s předním osvětlením	barevný CMOS
Velikost sensoru	16,6 x 14 mm	8,4 x 7,1 mm
Velikost pixelu	6,5 μm	3,45 μm
Snímkovací frekvence	100 fps	53,2 fps
Rozlišení	2560 x 2160 (5.5 megapixelů)	2448 x 2048 (5 megapixelů)
Maximální QE	64 %	65 %
Hloubka studny pixelu	30 000 e-	> 10 000 e-
Čtecí šum	0,9 e-	<2,1 e-
Dynamický rozsah	33 000:1	Až 71 dB



Obrázek 3 Kvantová účinnost jednotlivých barevných kanálů kamery Kiralux CS505CU. Převzato a upraveno z [5]

Reálné parametry dvou vybraných DSLR kamer jsou vypsány v tabulce 2. Oba sensory kamer také využívají technologii CMOS a mají podobnou velikost. DSLR kamery disponují velkým rozlišením a vysokým počtem megapixelů. Autoři [8] zmiňují, že v mikroskopii není rozlišení kamery až tak důležité. Mezi kamerou s 22 a 36 megapixely je pouze nepatrný rozdíl v rozlišení, protože rozlišení je limitováno čočkou mikroskopu. Důležitějším faktorem pro lepší zobrazení při větším přiblížení je podle [9] vysoký počet pixelů, díky čemuž lze reprodukovat více detailů.

Tabulka 2 Parametry dvou vybraných DSLR kamer

Model	Canon EOS R	Nikon D800
Typ sensoru	CMOS	CMOS
Velikost sensoru	36 x 24 mm	35,9 x 24 mm
Velikost pixelu	5,36 μm	4,88 μm
Rozsah ISO	100–40000	100–6400
Snímkovací frekvence	30 fps	30 fps
Rozlišení	6720 x 4480 (30,3 megapixelů)	7360 x 4912 (36.3 megapixelů)
Maximální QE	~ 50 %	~ 55 %
Hloubka studny pixelu	64 000 e-	48 000 e-

Hlavním rozdílem mezi mikroskopickými a DSLR kamerami je tak především velikost sensoru. Díky tomu tak mohou DSLR kamery nabídnout i větší počet pixelů a s tím spojené větší rozlišení. Vlastnosti DSLR a mikroskopických kamer ukazují, že velikost sensoru DSLR kamer je dvakrát větší, zatímco hodnoty rozlišení DSLR kamer několikanásobně převyšují hodnoty u mikroskopických kamer. Jako další výhody DSLR kamer autoři [9] uvádí i široký dynamický rozsah nebo barevnou hloubku, která je u DSLR kamery typicky alespoň 14 bitů na barevný kanál což je 42 bitů na barevný pixel. Mikroskopické kamery však nabízejí větší hodnoty QE. U porovnávaných mikroskopických kamer se tato hodnota pohybuje okolo 65 %, zatímco u DSLR kamer je okolo 50 %. Hodnota QE je velmi důležitá hlavně z toho důvodu, že dokáže ovlivnit kvalitu výsledného obrazu. Další vlastností, kterou mikroskopické kamery převyšují DSLR kamery je snímkovací frekvence. Tato vlastnost je důležitá pro plynulost živého zobrazení. Autoři [10] uvádí, že na snímkovací frekvenci 30 fps jsou již dnes lidé zvyklí jako na standart. Hodnoty 60 fps nebo 120 fps jsou považovány jako vysoké snímkovací frekvence. Výpočetní zobrazování nebo záznam dynamických dějů vyžadují vysokou snímkovací frekvenci kamery. Porovnávaná kamera Zyla 5.5 nabízí snímkovací frekvenci 100 fps a hodnotu QE 64 %, což je oproti oběma porovnávaným DSLR kamerám výrazný rozdíl. Zyla 5.5 by díky těmto vlastnostem byla vhodnou volbou pro záznam dynamických dějů. DSLR kamery mají oproti kameře Zyla 5.5 nižší snímkovací frekvenci, ale díky několikanásobně většímu rozlišení a velikosti sensoru tak umožňují zachytit více světla a produkovat snímky s vysokým rozlišením.

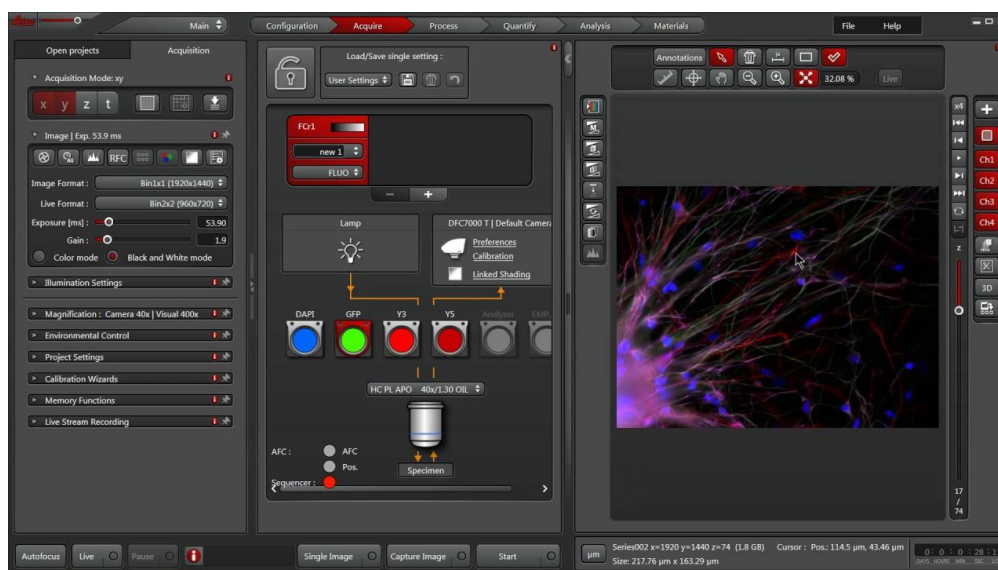
Díky nízké snímkovací frekvenci tak není vhodné využívat DSLR kamery pro záznam dynamických dějů. Hlavní předností DSLR kamer je již zmiňované vysoké rozlišení a s tím spojená vysoká kvalita snímků, což lze využít zejména při snímání s potřebou vysokých detailů.

2. SOFTWARE PRO OVLÁDÁNÍ KAMER

Zásadním rozdílem mezi mikroskopickými a DSLR kamerami je jejich samotné ovládání. Mikroskopické kamery jsou dodávány se softwarem, který umožňuje nastavování parametrů vzdáleně pomocí počítače. Po prvním zapojení a umístění kamery tak není potřeba s kamerou manipulovat. Toto však neplatí pro DSLR kamery, které jsou připojeny k počítači a využívány jako mikroskopické kamery. Pro změnu parametrů totiž vyžadují manuální změnu přímo na těle kamery, což je velmi nepraktické a může při tom dojít k narušení nastavení jak kamery, tak mikroskopu. Následující podkapitoly porovnávají programy LAS X, LAS EZ a ThorCam, které jsou určeny pro ovládání mikroskopických kamer, popisují jejich funkce a jednotlivé rozložení prvků.

2.1 Živé zobrazení, možnosti snímání a uložení snímku

U všech výše zmiňovaných programů zabírá největší plochu obrazovky okno pro živé zobrazení snímaného objektu. Díky tomu tak lze zhodnotit správné nastavení kamery a pozici snímaného objektu. Programy LAS X [11] a ThorCam [12] nabízí oproti LAS EZ funkci zastavení živého zobrazení. Samozřejmostí je pak možnost znovu obnovit zastavené živé zobrazení. Tyto dvě funkce jsou reprezentovány v programu LAS X jako dvě oddělená tlačítka, jak lze vidět v levém dolním rohu na Obrázku 4. Nahrávání videa je další funkcí, kterou uživatelé ocení zejména pro zachycení pohybujících se objektů pod mikroskopem. Kromě zobrazení, zastavení a nahrání živého náhledu je nezbytné obraz zachytit a uložit jako obrázek. Zachycením je míněno samotné vyfotografování objektu s parametry, které jsou nastaveny. Pro uložení je důležité, aby si uživatel mohl vybrat kam přesně chce zachycený obrázek uložit a jak bude pojmenován. LAS X umožňuje uživateli procházet uložené obrázky přes speciální kartu "Open projects" a LAS EZ pomocí karty „Browse“, díky níž lze také zobrazovat informace nejen o samotných snímcích, ale také jaké bylo nastavení kamery v době zachycení snímku.



Obrázek 4 Leica LAS X Software pro mikroskopickou kameru. Převzato z [11]

2.2 Nastavení

Tato karta (v programu jako „Configuration“) umožňuje uživateli nastavení samotné kamery a mikroskopu, ale také všech hardware komponent, připojených k němu. Mezi tyto komponenty se dále počítá objektiv mikroskopu nebo stolku na kterém je umístěn snímáný objekt. Déle se v této kartě dá nastavit výše zmiňovaná cesta ukládání snímků. Ve stejné kartě nazvané „Memory“ se dá dále zjistit zbývající místo na disku nebo název projektu. U programu LAS EZ se karta nastavení dá najít pod názvem „Preferences“, což nabízí jiné možnosti oproti LAS X. V této kartě je umožněno měnit například cestu uložení snímku, jméno, které mu bude přiřazeno nebo jeho formát. Další možností je nastavení toho, co má program provést po zachycení snímku nebo s jakým zvětšením bude mikroskop pracovat.

2.3 Akvizice

Jak již bylo výše zmíněno, tak živé zobrazení umožňuje uživateli hodnotit kvalitu výsledného obrazu. Programy tak obsahují panel, který umožňuje změnu určitých parametrů. Tyto parametry jsou pojmenovány různě podle zvoleného programu. Program Las X má panel pro změnu parametrů kamery umístěn v levé části obrazovky pod názvem „Acquisition“ (Akvizice). Je zde možné nastavit mód akvizice, parametry snímku jako formát, expozice, zisk, černobílý nebo barevný mód a také formát v jakém

chceme, aby bylo živé zobrazování. Programy tak nabízí volbu různých hodnot rozlišení a expozice. Rozlišení je závislé na typu kamery a jejím maximálním

dosažitelném rozlišení. Velké hodnoty rozlišení se hodí pro snímání vyžadující více detailů. Typické hodnoty rozlišení jsou 1920x1080 nebo 512x512. Podle autora [5] může volba vhodné expozice zabrat určitý čas a je často určována metodou pokus omyl. Toto platí obzvláště pro vzorky, které mají nerovnoměrný signál z důvodu nerovnoměrného zbarvení. Volba expozice tak záleží i na tom co chceme na snímku vidět. Lze také zmínit nastavení přiblížení kamery. U programu LAS EZ se panel pro nastavení parametrů kamery nachází také na levé straně, ale rozložení prvků je jiné, jak lze vidět na Obrázku 5. Nastavení parametrů kamery je rozděleno do několika skupin, které lze rozbalit a sbalit. Oproti programu LAS X má LAS EZ nastavení formátu jak živého zobrazení, tak snímku svoji vlastní skupinu, kde lze tyto parametry měnit. Program LAS X pracuje dále s kartou „Acquire“, která umožňuje změnit pokročilejší parametry kamery nebo objektivu mikroskopu. Je zde možné rychlé uložení nebo načtení nastavení pro snímání. Toto je vhodné k tomu, aby obsluha nemusela manuálně nastavovat parametry, ale rovnou zkoušet různé předdefinované nastavení kamery. Dalším nastavením zde je změna parametrů pro různé kanály pro fluorescentní snímání. Řešení programu ThorCam je jiné. Pro nastavení parametrů kamery je potřeba v horním menu programu zvolit položku „Settings“, která otevře nové okno, ve kterém je možné změnit všechny již zmiňované parametry kamery. ThorCam neumožňuje změnu nastavení formátu živého zobrazení.

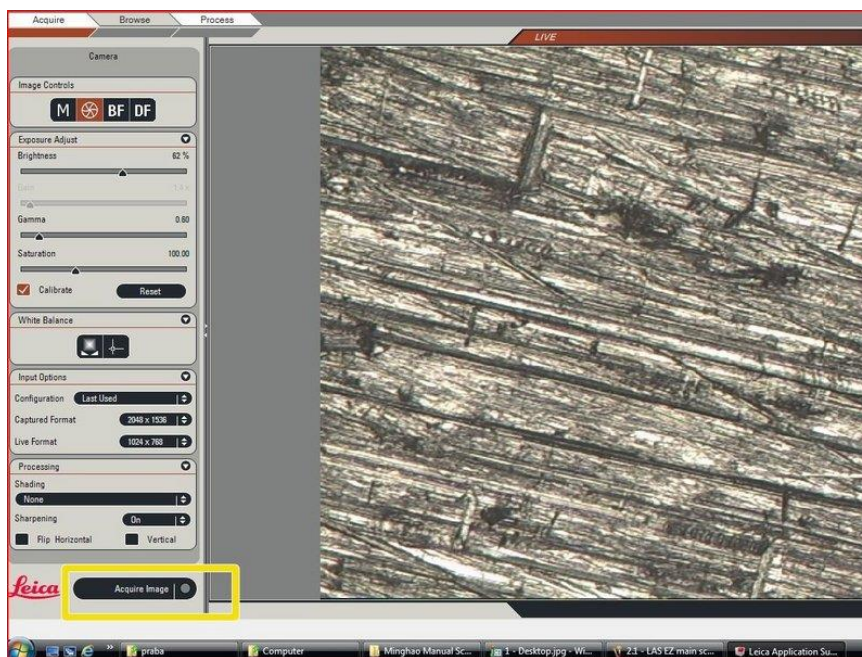
2.4 Zpracování obrazu

LAS X i LAS EZ nabízí kartu s názvem „Process“, která upravuje parametry pořízeného snímku a zpracovává ho. LAS EZ umožňuje uživateli změnit například název snímku nebo přidat popis. Dále je zde možné změnit jas, kontrast, barvy, orientaci nebo snímek oříznout. Program LAS X nabízí větší škálu možností nejen pro zpracování obrazu. Na Obrázku 4 lze v pravé části vidět okno s živým náhledem, které obsahuje různé funkce pro zpracování obrazu. Samotná karta „Process“ pak nabízí pokročilé funkce zpracování jako je 2D nebo 3D konvoluce, odstranění šumu, segmentace nebo oddělení barev. Tyto funkce mají schopnost se projevit na živém náhledu. Parametry tak mohou být měněny podle potřeby uživatele a po správném nastavení parametrů je pořízen výsledný snímek.

2.5 Analýza

Kartou s názvem „Analysis“ disponuje pouze program LAS X. Tato karta umožňuje nastavit automatizovaný pracovní postup. Můžeme zde definovat standardizovaný proces, který zadáme do „pipeline“, což umožní efektivní procházení dat a kompletní analýzu. Tato karta také nabízí analýzu jak 2D tak 3D snímků, což

umožňuje obsluhu provádět měření pomocí přístupu založeného na prahové binární masce u barevných nebo monochromatických snímcích.



Obrázek 5 Program LAS EZ pro mikroskopickou kameru. Převzato z [13]

3. TECHNOLOGIE PRO VÝVOJ SOFTWARE

Tato kapitola popisuje technologie, které budou využity k vytvoření softwaru pro obsluhu DSLR kamer. Technologie jsou rozděleny do dvou částí podle toho, zda se jedná o technologii, která se stará o klientskou vizuální stránku nebo o logickou stránku softwaru.

3.1 Back-end

Back-end neboli serverová strana aplikace je vše co uživatel používající aplikaci nevidí. Zajišťuje fungování dynamických prvků aplikace, a hlavně má na starost logiku celé aplikace. Back-end softwaru pro ovládání DSLR kamer bude vytvořen v jazyku Python. Pro komunikaci se nabízí využití buď knihovny DigiCam nebo gPhoto2.

3.1.1 Python

Python je objektově-orientovaný, vysokoúrovňový programovací jazyk. Je velmi využíván zejména pro rychlý vývoj aplikací, jako skriptovací jazyk nebo jako spojovací jazyk pro připojení komponent dohromady. Syntaxe toho to jazyku podporuje čitelnost a je tak díky tomu méně náročný na údržbu. Díky tomu, že jazyk Python není třeba kompilovat se cyklus pro úpravu-test-ladění výrazně zrychlí [14].

3.1.2 DigiCam

Jedná se o knihovnu vytvořenou v jazyce Python, která umožňuje ovládat DSLR kamey s využitím open source programu DigiCamControl. Knihovna umožňuje po připojení kamery k počítači a nastavení parametrů pořízení snímku. Kameru tak ovládat pomocí skriptu, vytvořeného v jazyce Python [15].

3.1.3 gPhoto2

Tato knihovna je komplexní rozhraní Pythonu na libgphoto2. Díky tomu, že je toto rozhraní vytvořeno pomocí SWING, tak poskytuje přímý přístup k téměř všem funkcím libgphoto2. libgphoto2 je základní knihovna, která je navržena tak, že umožňuje přístup k digitálnímu fotoaparátu přes externí programy [16].

3.2 Front-end

Pojmem front-end je myšlena klientská strana aplikace. Vše, co uživatel vidí po spuštění aplikace (tj. tlačítka, obrázky, grafika, animace a další) je považováno za front-end. Jiný název pro front-end může být tedy „klientská strana“ aplikace která má na starost hlavně vzhled aplikace a propojení se serverovou neboli back-end částí aplikace.

Front-end využívá různé nástroje, jazyky, frameworky a knihovny k navržení a dosažení designu a rozložení aplikace [17]. Navrhovaná aplikace bude využívat Python jako jazyk pro back-end i front-end. Python využívá různé GUI rámce pro tvoření uživatelského rozhraní. Níže jsou zmíněny nejpoužívanější rámce pro vývoj GUI, a nakonec i nástroj využívaný pro tvorbu designu a rozložení softwaru.

3.2.1 PyQt5

Tento balíček je postaven na Qt rámci, což je multiplatformní rámec, který se využívá pro vytvoření velkého množství aplikací pro různé platformy. Pro Python obsahuje podrobnou sadu vazeb na základě nejnovější verze aplikačního rámce Qt5. PyQt je stejně jako Qt multiplatformní, což umožňuje vytvářet aplikace pro platformy jako Windows, Mac, Linux, Android a další. Vytváření GUI je díky modulům QtGUI a QtDesigner snadné a nabízí spoustu vizuálních prvků, které lze implementovat. Jednou z nevýhod tohoto rámce je, že pro jeho použití vyžaduje zakoupení komerční licence.

3.2.2 Tkinter

Velkou většinou Python vývojářů je označován za nejlepší GUI rámec. Tkinter většinu vizuálních prvků nazývá widgetem. Každý widget pak nabízí jinou míru toho, jak je možné ho upravit. Tkinter je obzvláště výhodný pro vývoj desktopových aplikací. Další výhodou je, že pro jeho používání není potřeba nic instalovat, Tkinter je totiž součástí nejnovějších verzí Pythonu.

3.2.3 Kivy

Kivy je další multiplatformní knihovna Pythonu, která je navržena pro rychlý vývoj aplikací se složitým uživatelským rozhraním. Oproti PyQt5 se jedná o open-source knihovnu, což znamená, že je bezplatná. Jednou vytvořený kód může být použit na více platformech. Nejběžnějším použitím této knihovny lze vidět v aplikacích pro Android a iOS. Dalším rozšířenou implementací lze vidět v uživatelských rozhraních na zařízeních Linux, Windows, Mac OS nebo Raspberry Pi.

3.2.4 WxPython

Multiplatformní GUI toolkit, který umožňuje vytvoření nativních robustních a funkčních GUI jednoduchým a snadným způsobem. Dále aplikaci nepřidává žádné další režie, což umožňuje funkčnost knihovny bez jakékoliv zátěže. WxPython je nejen multiplatformní, ale také open source, takže nepředstavuje žádné další náklady. Pro využití vyžaduje snadnou instalaci pomocí příkazu pip [18][19].

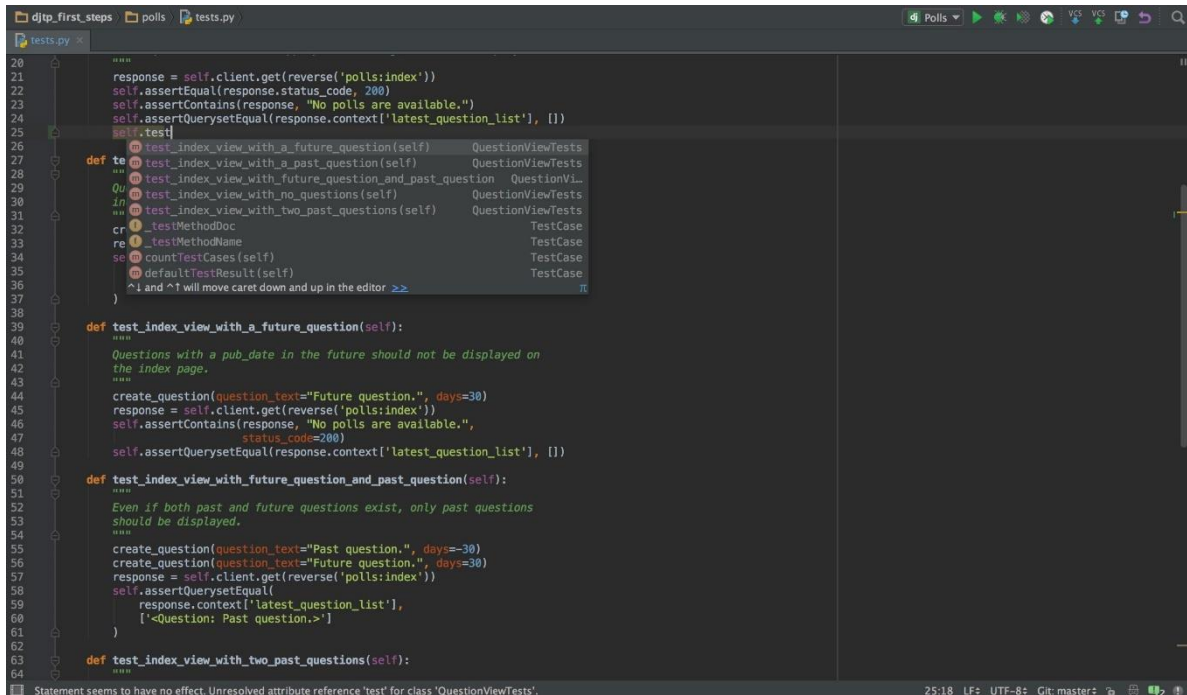
3.2.5 Figma

Jedná se o bezplatný cloud-based nástroj určený pro design. Jednou z největších výhod Figmy je, že se dá použít na jakémkoliv zařízení, které má nainstalovaný webový

prohlížeč. Figma umožňuje spolupráci více lidí na jednom projektu v reálném čase. Je tak možné nejen upravovat v reálném čase, ale také prezentovat vytvořený design. Možnost vkládat komentáře do designů je další funkcí, která usnadňuje spolupráci v týmu. Figma lze používat i v off-line režimu, který ale neumožňuje náhled ostatním členům týmu v reálném čase [20].

3.3 Vývojové prostředí PyCharm

PyCharm je integrované vývojové prostředí (IDE) pro programovací jazyk Python. Každé IDE se skládá z editoru, do kterého se píše kód a kompilátoru, který slouží ke kompilaci programů, což umožňuje efektivní vývoj software. Toto vývojové prostředí dokáže navrhnout relevantní klíčová slova a odlišovat barvou nejen třídy a metody, jak můžete vidět na Obrázku 6, ale také poskytuje odlišné barvy pro správná a nesprávná klíčová slova, což usnadňuje uživateli práci.



```
20
21 response = self.client.get(reverse('polls:index'))
22 self.assertEqual(response.status_code, 200)
23 self.assertContains(response, "No polls are available.")
24 self.assertQuerysetEqual(response.context['latest_question_list'], [])
25 self.test
26
27 def test_index_view_with_a_future_question(self):
28     """
29     test_index_view_with_a_past_question(self)
30     test_index_view_with_future_question_and_past_question
31     test_index_view_with_no_questions(self)
32     test_index_view_with_two_past_questions(self)
33     """
34     cr = testMethodDoc
35     re = testMethodName
36     se = CountTestCases(self)
37     defaultTestResult(self)
38     TestCase
39
40 def test_index_view_with_a_future_question(self):
41     """
42     Questions with a pub_date in the future should not be displayed on
43     the index page.
44     """
45     create_question(question_text="Future question.", days=30)
46     response = self.client.get(reverse('polls:index'))
47     self.assertContains(response, "No polls are available.",
48                        status_code=200)
49     self.assertQuerysetEqual(response.context['latest_question_list'], [])
50
51 def test_index_view_with_future_question_and_past_question(self):
52     """
53     Even if both past and future questions exist, only past questions
54     should be displayed.
55     """
56     create_question(question_text="Past question.", days=-30)
57     create_question(question_text="Future question.", days=30)
58     response = self.client.get(reverse('polls:index'))
59     self.assertQuerysetEqual(
60         response.context['latest_question_list'],
61         ['<Question: Past question.>']
62     )
63
64 def test_index_view_with_two_past_questions(self):
```

Obrázek 6 Ukázka vývojového prostředí PyCharm. Převzato z [23]

4. NÁVRH PROGRAMOVÉHO VYBAVENÍ

Následující kapitola, definuje grafické uživatelské rozhraní a představuje návrh rozložení prvků softwaru pro obsluhu DSLR kamer. Rozložení prvků je rozděleno na tři bloky, které jsou v podkapitolách popsány. Kapitola tak slouží jako nástroj, ze kterého lze dále vycházet v praktické části.

4.1 Grafické uživatelské rozhraní softwaru

Grafické uživatelské rozhraní, často také zkráceně jako GUI je část softwaru, se kterou uživatel přímo interaguje. Zkratka GUI se často zaměňuje s UI, které označuje uživatelské rozhraní. Jako příklad GUI lze použít webovou stránku, která obsahuje grafické prvky, díky kterým uživatel interaguje se stránkou. Mezi UI se tak řadí například textové rozhraní příkazového řádku (CLI), které neobsahuje žádné grafické prvky a je tak mnohem méně uživatelsky přívětivé [21].

Na Obrázku 7 lze vidět návrh rozložení prvků softwaru pro obsluhu DSLR kamer. Rozložení je rozděleno do tří bloků, které spolu navzájem spolupracují.

4.1.1 Blok pro volbu kamery a místa uložení snímku

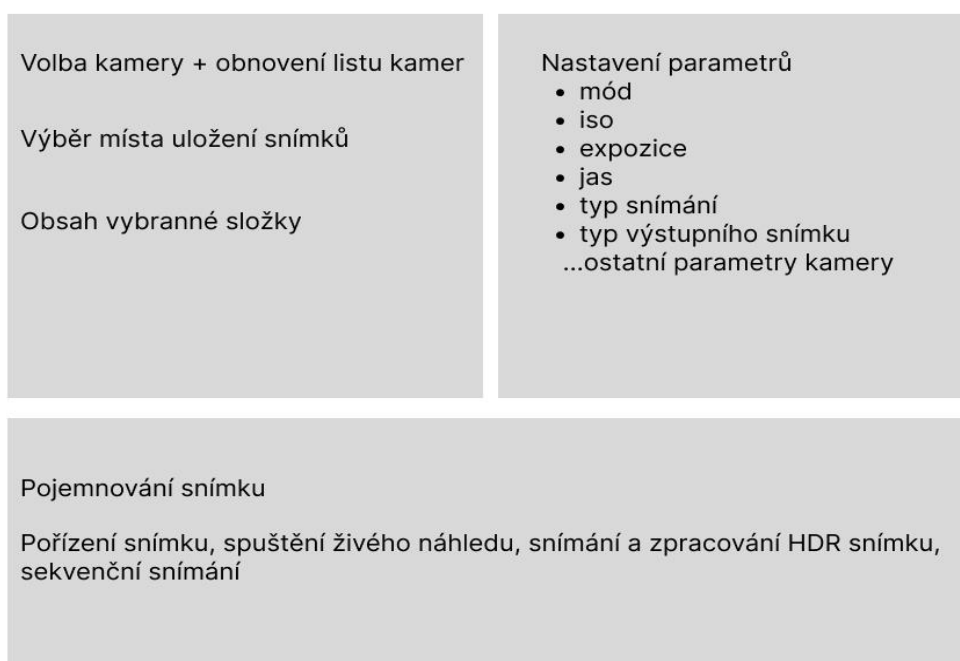
Hlavní úkol bloku, který se nachází v levém horním rohu je načtení připojené kamery. Toto je důležité zejména pokud má uživatel připojeno více kamer a potřebuje rozlišit se kterou momentálně komunikuje. Dále si v tomto bloku bude moct uživatel vybrat kam bude chtít pořízené fotografie ukládat. Tato funkce bude nejspíš reprezentována tlačítkem, které zobrazí dialogové okno pro výběr cesty do určité složky. Po zvolení by se měla aktuálně vybraná cesta uživateli ukázat z důvodu přehlednosti. Jako další by se po zvolení cesty měl vypsát aktuální obsah složky. Konkrétně tedy všechny obrázky, které jsou ve zvolené složce, což usnadní uživateli práci při hledání pořízených fotografií. Tento seznam by se měl po každém snímání aktualizovat, aby měl uživatel přehled o názvech a počtu pořízených snímků.

4.1.2 Blok pro nastavení parametrů pro snímání

Druhý blok, který je umístěn v pravém horním rohu je určen k nastavování parametrů kamery. Umístění tohoto bloku na pravé straně se liší od výše zmíněných programů LAS X a LAS EZ hlavně z důvodu přehlednosti a návaznosti. Možnost změny parametrů přímo bez proklikávání do nových oken programu je také jednou z výhod tohoto návrhu. Blok umožňuje nastavit ISO, expozici, jas, typ snímání a také možnost přepnout do automatického módu nastavování parametrů.

4.1.3 Blok pro pojmenování snímku a spouštění snímání

Posledním blok je umístěn ve spodní části a jeho primárním účelem je spouštění snímání a určení názvu pořízeného snímku. Změna názvu snímku je užitečnou funkcí hlavně pro přehlednost, kdy si uživatel může změnit název pořizovaného snímku v závislosti na tom, s jakým vzorkem a při jakém nastavení kamery pracuje. Tento blok by měl dále obsahovat tlačítka pro spuštění snímání, živého náhledu, akvizici a zpracování HDR snímku a v neposlední řadě také tlačítko pro sekvenční snímání, které pořídí sekvenci tří snímků v řadě.



Obrázek 7 Návrh rozložení grafického uživatelského rozhraní.

5. PRAKTICKÁ ČÁST

Praktická část se zaměřuje nejen na implementaci navrženého řešení software a zajištění komunikace s DSLR, ale také na výběr vhodných knihoven a zařízení. V této části je prezentováno softwarové řešení pro umožnění ovládní DSLR jako mikroskopické kamery.

5.1 Popis software

Celý software je napsán v programovacím jazyce Python a je rozdělen na dva hlavní a jeden vedlejší skript, které budou popsány v dalších podkapitolách. Jako vývojové prostředí byl zvolen program PyCharm pro jeho přehlednost a navigaci v něm. Pro komunikaci mezi DSLR kamerou a PC, bylo využito knihovny libgphoto2, konkrétně příkazového řádku gphoto2, který navazuje komunikaci s kamerou a dokáže měnit hodnoty nastavení.

5.1.1 Ovládní kamery pomocí gphoto2 a Python

Pro ovládní kamery je využito knihovny libgphoto2. Tato knihovna disponuje příkazovým řádkem gphoto2, který umožňuje ovládní fotoaparátu, získávání informací o fotoaparátu, jeho nastavení a mnoho další funkcionality [24]. Pro komunikaci s kamerou pomocí příkazového řádku se využívá následující syntaxe:

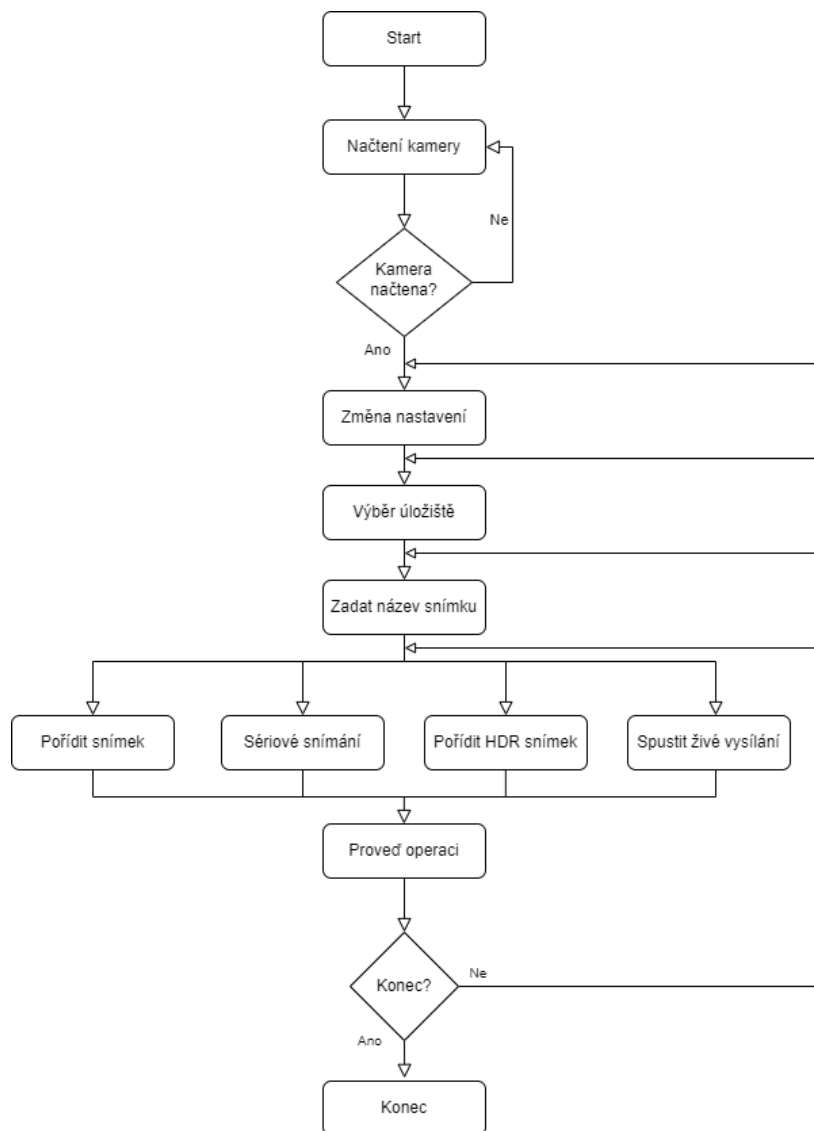
```
„gphoto2 --list-cameras“
```

Tento konkrétní příklad vypíše do příkazového řádku všechny připojené kamery a porty na kterých jsou kamery připojeny. Pro účely vývoje software se využilo programovacího jazyka Python a knihovny subprocess, která umožňuje spouštět procesy a číst jejich výstupy. Skript cameraControl.py využívá funkce ``subprocess.run()`` a ``subprocess.check_output()``.

5.2 Vývojový diagram software

Pro následující vývoj software je důležité určit základní funkčnost, kterou lze definovat pomocí vývojového diagram na Obrázku 8. Prvním blokem tohoto diagramu je načtení kamery. V případě, že nelze kameru načíst se tento blok opakuje, dokud není detekce kamery úspěšná. Dalším blokem je nastavení parametrů kamery jako je například hodnota ISO, vyvážení bíle nebo rychlost uzávěrky. V dalším kroku je nutné zvolit destinaci pro ukládání pořizovaných fotografií a zadat řetězec, pro identifikaci fotografie v destinaci. Jakmile jsou nastaveny parametry kamery, úložiště a název snímku, tak by si měl uživatel vybrat ze tří typů snímání nebo spuštění živého náhledu. Lze tedy pořídít samostatný snímek, spustit sériové snímání, které pořídí tři fotografie v

úzkém sledu za sebou, nebo vyvolat HDR snímání. Software po provedení jedné z možností uloží výsledné fotografie a uživatel může znovu měnit jak hodnoty nastavení DSLR kamery, tak i úložiště a název fotografie.



Obrázek 8 Diagram základního fungování aplikace

5.3 Funkce pro komunikaci s kamerou

Základní funkce pro detekci a ovládání kamery jsou uloženy ve skriptu cameraControl.py a stejnojmenné třídě cameraControl. Tato třída obsahuje tři základní typy funkcí, první má za úkol získat hodnoty ISO, vyvážení bílé barvy, clony, rychlosti závěrky, formátu výsledného obrazu a módu zaostření společně s módem kamery. Další

zjišťuje aktuální hodnotu zvoleného nastavení a poslední funkce nastavuje hodnotu vybraného nastavení na novou. Tyto základní funkce mají název `get_settings`, `current_settings`, `set_settings`. Další funkce, které tato třída definuje jsou `detect_camera`, která má za úkol detekovat připojené kamery a funkce `take_picture`, která definuje název, pořízeného snímku, cestu, kam se pořízený snímek uloží a poté pořídí snímek s aktuálním nastavením.

Následující podkapitoly mají za úkol popsat funkce, které se ve třídě `cameraControl` nejvíce využívají. Tyto funkce využívají pro bezchybný průběh a komunikaci několik knihoven, které nebyly zmíněny v kapitole 3. Mezi tyto knihovny patří:

Subprocess

Tato knihovna má za úkol spouštět nové procesy, připojit se k jejich vstupů a výstupům a také získání jejich návratových kódů.

sys

Knihovna `sys` poskytuje přístup k proměnným a funkcím, které mají vztah k interpreteru jazyka Python. Obsahuje nástroje, pro interakci se systémem na nízké úrovni a zároveň získává informaci o aktuálně běžícím interpreteru a jeho momentální konfiguraci.

Os

`Os` poskytuje rozhraní, které pracuje s operačním systémem a umožňuje tak s ním interagovat na úrovni souborů, adresářů a procesů. S touto knihovnou tak lze číst, upravovat i přesouvat soubory.

Cv2

`Cv2` neboli `OpenCV` je knihovna pro zpracování obrazu a počítačového vidění. Obsahuje přes 2500 optimalizovaných algoritmů pro zpracování obrazu a strojového učení.

datetime

Knihovna `datetime` poskytuje třídy pro manipulaci s daty a časem. Dále také umožňuje efektivní extrakci atributů pro výstupní formátování a manipulaci s daty a daty.

NumPy

`NumPy` poskytuje třídy, které mají za úkol práci s vektory, maticemi a vícerozměrnými poli.

5.3.1 Funkce pro detekci kamer `detect_camera()`

Funkce `detect_camera` spouští příkaz, který vrátí názvy a čísla portů na kterém je každá kamera připojena. Dále je separuje do proměnných `camera_names` a `camera_ports`, které jsou poté navráceny.

```
def detect_camera(self):

    cmd = [ 'gphoto2', '-suto-detect' ]
    output = subprocess.check_output(cmd)
    lines = output.splitlines()

    camera_lines = []

    for line in range(2, len(lines)):
        camera_lines.append(lines[line])

    camera_names = []
    camera_ports = []

    for line in camera_lines:
        parts = line.split()
        name = ' '.join(parts[:-1])
        port = parts[-1]

        camera_names.append(name)
        camera_ports.append(port)

    return camera_names, camera_ports
```

5.3.2 Funkce pro získání hodnot konkrétního nastavení `get_settings()`

Níže uvedená funkce získává hodnotu uživatelem zvoleného nastavení z připojené DSLR kamery. Funkce přijímá argument `port`, který definuje port, na kterém je zvolená kamera připojena a argument `settings`, který je definován jako řetězec. Dále je definován list s názvem `cmd`, který obsahuje jednotlivé příkazy, potřebné pro zadání do příkazového řádku. Řetězec `--get-config=` společně s řetězcem `settings` definují, jaké konkrétní hodnoty nastavení chceme dostat jako výstup. Výstup z příkazového řádku je uložen do proměnné `result` přes knihovnu `subprocess` a její funkci `check_output`, která přijímá proměnnou `cmd` jako argument. Prázdný list s názvem `settings_values` je definován před `for` cyklusem, který má za úkol extrahovat hodnoty ISO vrácené do proměnné `result`. V tomto `for` cyklu je ošetřeno, aby i hodnoty, které obsahují více než dvě slova byly správně uloženy.

V posledním kroku funkce vrací list hodnot vybraného nastavení připojené DSLR kamery.

```
def get_settings(self, port, settings = str):

    cmd = [ 'gphoto2', '--port', port, '--get-config=', settings]
    result = subprocess.check_output(cmd)
    settings_values = []

    for line in result.decode('utf-8').split('\n'):
        if line.startswith('Choice:'):
            parts = line.split()
            temp = []

            for part in range(2, len(parts)):
                temp.append(parts[part])
            temp = ' '.join(temp)
            settings_values.append(temp)

    return settings_values
```

5.3.3 Funkce pro zjištění aktuálního nastavení kamery current_settings()

Tato funkce využívá stejnou strukturu jako funkce get-config s tím rozdílem, že návratová proměnná je textový řetězec.

```
def current_settings(self, port, settings = str):

    cmd = [ 'gphoto2', '--port', port, '--get-config=', settings]
    result = subprocess.check_output(cmd)
    current_settings = ''

    for line in result.decode('utf-8').split('\n'):
        if line.startswith('Current:'):
            parts = line.split()
            temp = []

            for part in range(1, len(parts)):
                temp.append(parts[part])
            temp = ' '.join(temp)
            current_settings = temp

    return current_settings
```

5.3.4 Funkce pro nastavení konkrétní hodnoty DSLR kamery set_settings()

Pro změnu specifického nastavení kamery se využívá níže uvedená funkce, která bere jako argumenty řetězce `settings` a `settings_value`. Tyto řetězce jsou poté vloženy do proměnné `cmd`, která formuje příkaz pro příkazový řádek a poté ho spustí.

```
def set_settings(self, port, settings = str, settings_value = str):  
  
    cmd = [ 'gphoto2', '--port',port, '--set-config=' ,settings +'='+  
           settings_value]  
  
    subprocess.run(cmd)
```

5.3.5 Funkce pro pořízení snímku take_picture()

Pořízení snímku je definováno ve funkci `take_picture`, která se nachází níže. Jako argumenty přijímá dva řetězce `path` a `photo_name`. Tyto proměnné mají za úkol definovat cestu pro uložení snímku a název snímku. Funkce dále zpracuje jméno snímku kvůli jeho unikátnosti, vytvoří příkaz s `'--capture-image-and-download'` a spustí proces pro pořízení a stažení snímku.

```
def take_picture(self, path=str, photo_name=str):  
  
    date = datetime.datetime.now()  
  
    now = [str(date.day), str(date.month), str(date.microsecond)]  
  
    now = '_'.join(now)  
  
    cmd = [ 'gphoto2', '--capture-image-and-download', '--filename',  
           path+'/'+photo_name+'_'+now]  
  
    subprocess.run(cmd)
```

5.3.6 Funkce k zobrazení živého zobrazení live_view()

`live_view` přijímá index kamery, který je dále využit ke kontrole připojení kamery. Dále funkce spustí čtení kamery a zobrazí živý náhled z vybrané kamery.

```
def live_view(self, camera_num):  
    cap = cv.VideoCapture(camera_num)  
  
    if not cap.isOpened():
```

```

        raise IOError(„Cannot open webcam“)

while True:
    ret, frame = cap.read()
    if frame is None:
        print(„Image not found, Check camera index or camera
            connection.“)
        break

    frame = cv.resize(frame, None, fx=0.2, fy=0.2,
        interpolation=cv.INTER_AREA)
    cv.imshow(„Live View“, frame)

    c = cv.waitKey(1)
    if c == 27:
        break

cap.release()
cv.destroyAllWindows()

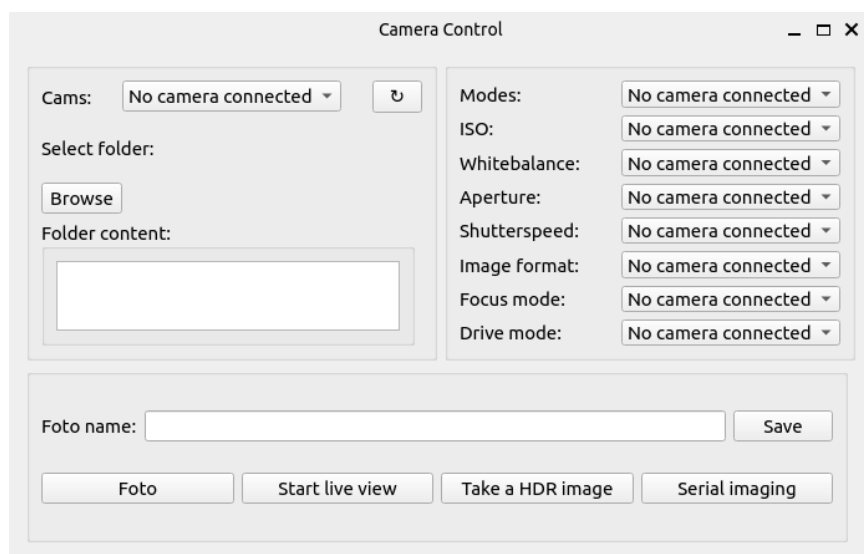
```

5.4 Grafické uživatelské rozhraní software

GUI bylo tvořeno podle návrhu, který byl definován v kapitole 4. K vývoji software bylo využito knihovny PyQt5. Obrázek 9 ukazuje GUI software, které je rozděleno do tří bloků. První blok, který se nachází v levém horním rohu slouží k načtení připojených kamer, zvolení cesty, kam se mají uložit pořízené snímky a vypsání obsahu zvolené složky.

Druhý blok, který leží v pravém horním rohu slouží ke změně hodnot nastavení kamery. Je zde využito rozbalovacích seznamů, které uchovávají jednotlivé možnosti konkrétního nastavení.

Spodní blok obsahuje textové pole s tlačítkem, které slouží k zadání a uložení názvu snímku. Dále jsou v tomto bloku tlačítka pro pořízení snímku, spuštění živého vysílání, pořízení HDR snímku a spuštění sériového snímání.



Obrázek 9 Ukázka grafického uživatelského rozhraní

6. TESTOVÁNÍ

Testování je nezbytnou částí při vývoji software. Zajišťuje, že testovaný software splňuje všechny požadované specifikace a funguje podle očekávání. Testování je dále nezbytné pro odhalení chyb a problémů, ale také přispívá ke zlepšení kvality produktu díky identifikaci oblastí, které mohou být vylepšeny. Bez správného testování by mohl uživatel přijít do styku se software, který není funkční a nesplňuje tak očekávání.

Důležitým aspektem pro uživatele je vědět na jakém zařízení může software spustit, jaké k tomu potřebuje další periférie a nástroje. Při testování je tak důležité zohlednit všechny zařízení na kterých může být software spuštěn a všechny způsoby, jak může být software ovládán.

Tato kapitola se nejprve zaměřuje na specifikaci potřebných nástrojů a aplikací, které uživatel potřebuje pro spuštění programu. Poté je definován konkrétní postup testování software pro správnou funkčnost, načež je navázáno samotným testováním software na různých zařízeních s různými způsoby ovládání, které jsou dostupné pro tato zařízení. Pro každý způsob ovládání a zařízení budou zmíněny požadavky, které jsou ke spuštění software vyžadovány. Testování tak bude zaměřeno nejen na funkčnost software, ale také na časy, které vyžaduje software k jednotlivým úkonům jako je pořízení fotografie, změna konkrétní hodnoty nastavení nebo módu a také načtení kamery. Časy změn jsou důležité kvůli uplatnění software v mikroskopii. Díky tomuto tak bude možné specifikovat pro jaké pozorování se software hodí nejlépe.

6.1 Příprava prostředí pro spuštění software

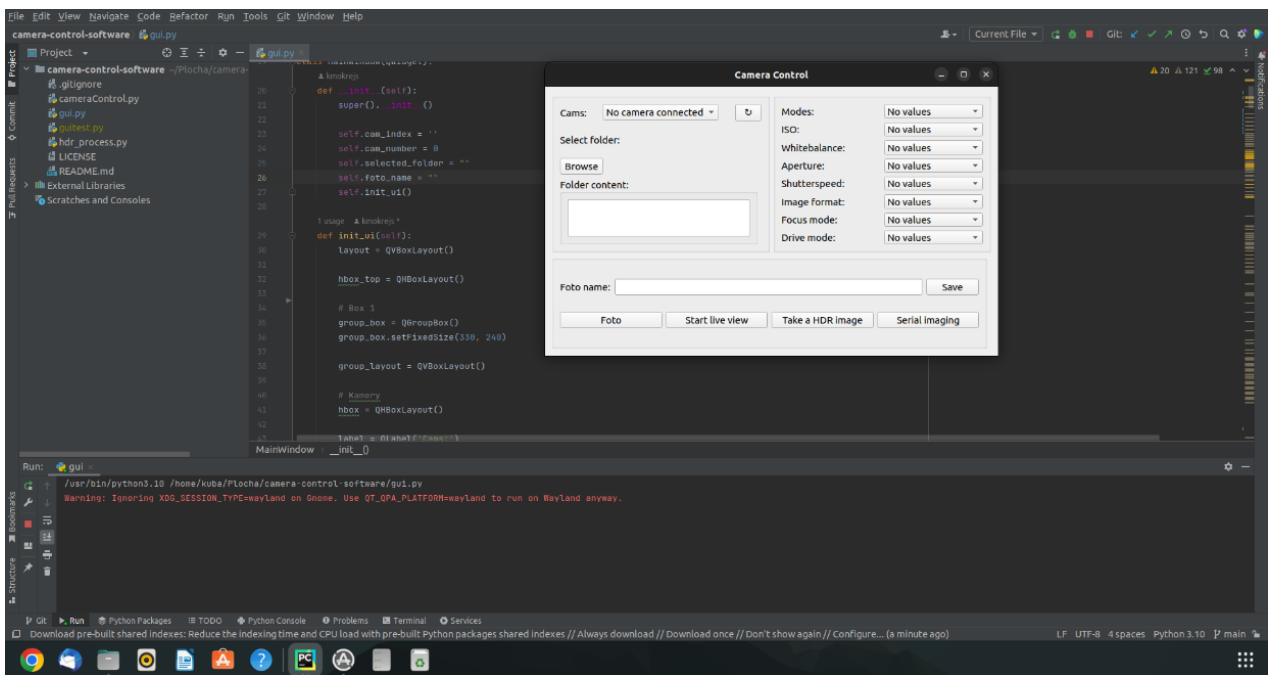
Před testováním software na jakémkoliv zařízení je potřeba zajistit, aby společně se složkou s projektem byly na zařízení nainstalovány i potřebné knihovny. Tyto knihovny jsou definovány v kapitolách 3 a 5.3.

Uživatel může získat složku s projektem z přílohy této práce nebo lze naklonovat projekt z repositáře na stránce Github [25]. Pro získání knihoven potřebných ke spuštění software lze na operačním systému Linux využít příkazového řádku a příkazu „`sudo apt-get install`“. Pro instalaci knihovny OpenCV tak bude příkaz vypadat následovně:

```
„sudo apt-get install python3-opencv“
```

Další důležitou částí pro spuštění software je instalace vývojového prostředí. Pro vývoj tohoto software bylo využito vývojového prostředí PyCharm, lze však využít i jiné používané vývojové prostředí jako je Visual Studio Code nebo Vim.

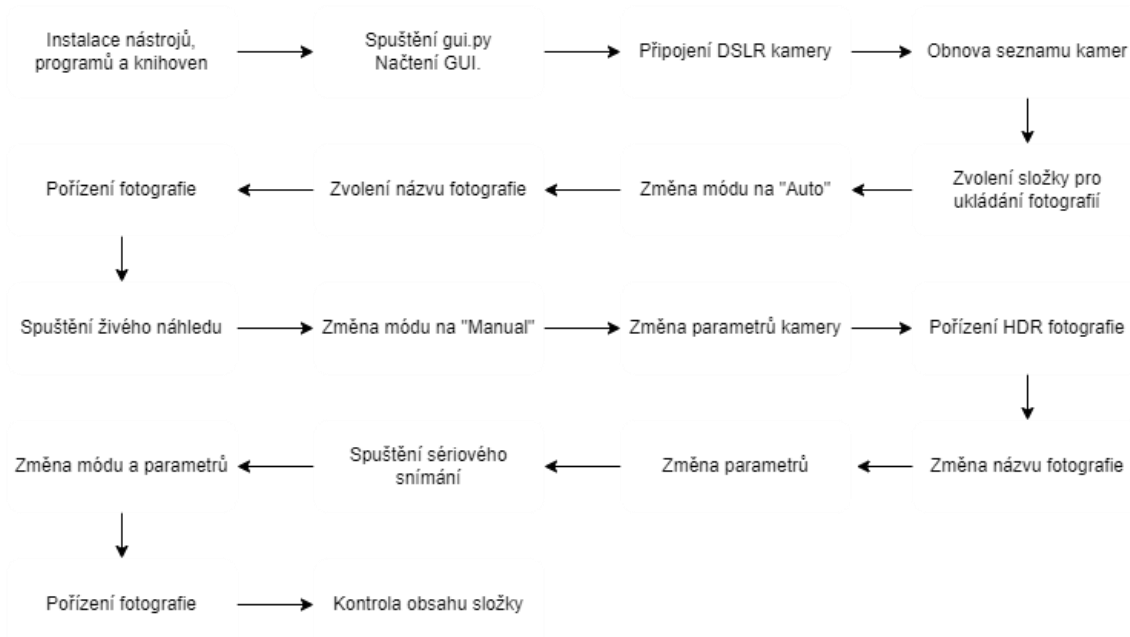
Pro otestování správně stažených a nainstalovaných knihoven a projektu se uživatel může navigovat do složky projektu, kde buď pomocí zvoleného vývojového prostředí nebo příkazového řádku spustí skript s názvem „gui.py“. Po spuštění by se uživateli mělo zobrazit okno jako na Obrázku 10, které značí, že je vše správně nainstalováno a připraveno k použití.



Obrázek 10 Okno, které se zobrazí v případě správně nastaveného prostředí

6.2 Definice postupu testování

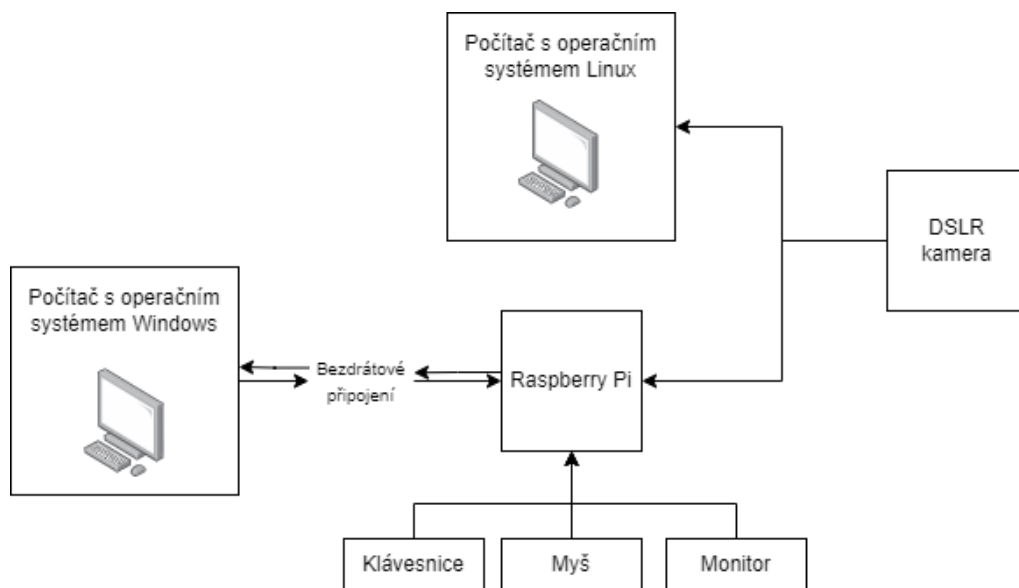
Testování bude opakováno podle počtu metod ovládání, a tak je nutné zajistit stejný postup. K definici postupu testování slouží diagram na Obrázku 11. Tento postup byl zvolen s ohledem na otestování všech funkcionalit software a s ohledem na potřebu uživatele otestovat mnoho nastavení kamery pro pořízení jednoho správného snímku. Je využito i spuštění živého náhledu ke kontrole správného pozičního nastavení kamery a potřebu zachytit několik snímků za sebou pomocí sériového snímání.



Obrázek 11 Postup testování software

6.3 Možnosti ovládání software

Pro ovládání software se nabízí více způsobů. V této podkapitole jsou popsány tři možnosti, jak je možné software ovládat. Blokové schéma na Obrázku 12 představuje možnosti ovládání software a komunikace mezi kamerou a zařízením na kterém je spuštěno. Dvě možnosti představují přímou komunikaci mezi kamerou a software buď pomocí mikropočítače Raspberry Pi nebo obyčejného počítače, na kterém je nainstalován operační systém Linux. Déle lze Raspberry Pi využít jako prostředníka, který umožňuje bezdrátové připojení a ovládání mikropočítače se software přes počítač s operačním systémem Windows.



Obrázek 12 Blokové schéma komunikace s DSLR kamerou

6.3.1 Ovládání přes počítač s operačním systémem Linux

Jako první testovaný způsob ovládání bylo zvoleno ovládání přes počítač, který využívá operační systém Linux. Pro testování byl využit počítač Acer Aspire V3-372-754K s distribucí operačního systému Linux s názvem Ubuntu.

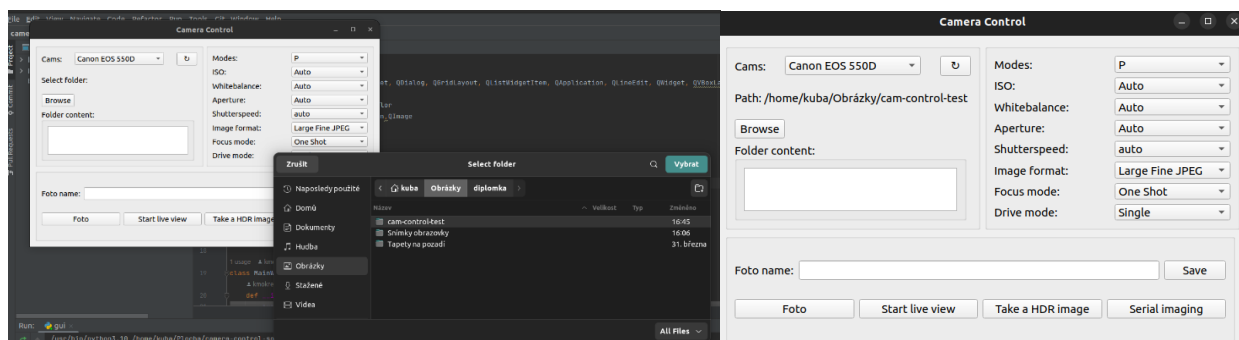
Tabulka 1.3 Technické specifikace počítače využívaného pro testování

Model	Acer Aspire V3-372-754K
Procesor	Intel Core i7 6500U (2.5/3.1GHz 2jádra/4vlákna)
Velikost operační paměti	8 GB
Grafika	Intel HD Graphics 520
Kapacita disku	266 GB SSD
Rozhraní	HDMI, USB 2.0, USB 3.0, Wi-Fi, Bluetooth
Operační systém	Linux - Ubuntu

Prvním krokem při testování bylo podle výše definovaného postupu stažení a nainstalování všech potřebných programů, knihoven a nástrojů. Jako vývojové prostředí zde bylo využito programu PyCharm. K získání složky projektu byl projekt naklonován z repositáře na stránce Github [25]. Ověření funkčnosti všech stažených a nainstalovaných knihoven, programů a projektu proběhlo podle popisu v kapitole 6.1

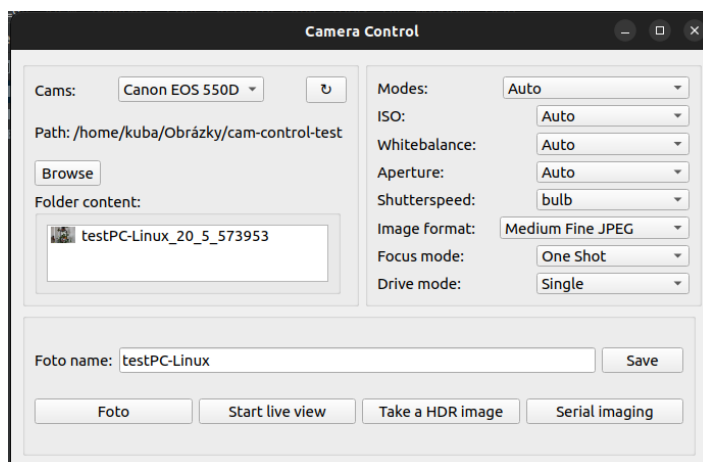
Následovalo propojení DSLR kamery s počítačem pomocí USB kabelu. Po propojení bylo ověřeno načtení kamery kliknutím na tlačítko "U", které má za úkol obnovení seznamu kamer. Načtení kamery trvalo zhruba šedesát pět vteřin. Po této době

se načety hodnoty do bloku s aktuálními parametry fotoaparátu. Poté co se zvolila cesta, kam se fotografie budou ukládat, přes tlačítko „Browse“. Řetězec s cestou do zvolené složky se zobrazil nad samotným tlačítkem v sekci „Path:“. Obsah složky se ukazoval jako prázdný, jelikož ve složce zatím nebyly uloženy žádné fotografie.



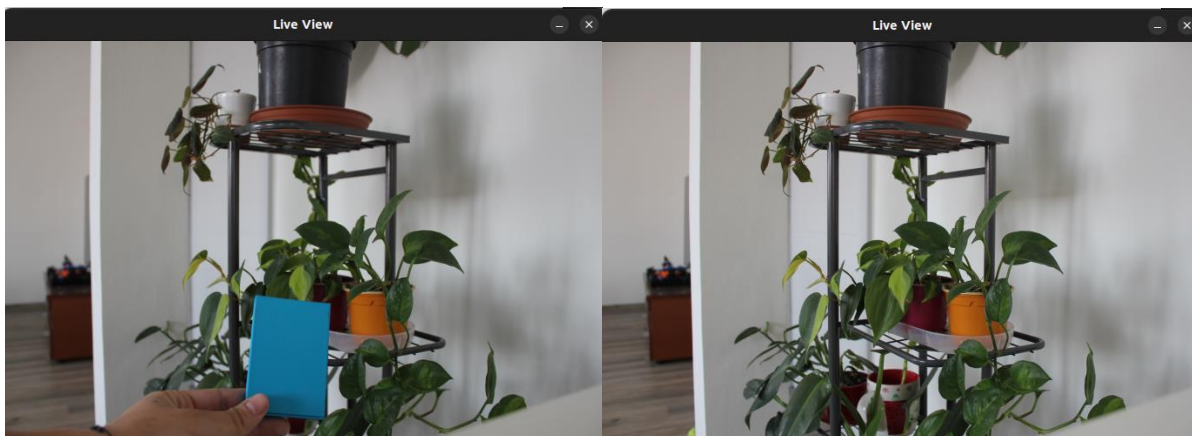
Obrázek 13 Výběr úložiště fotografií s vypsáním do GUI

V dalším kroku se změnil mód kamery z módu „P“ na „Auto“. Tento proces trval devět vteřin. Pro pojmenování snímku se zadal do pole „Foto name:“ řetězec „testPC-Linux“ a kliknutím na políčko „Save“ se potvrdila změna názvu fotografie. Pořízení fotografie stisknutím na tlačítko „Foto“ trvalo méně než dvě vteřiny, po kterých se objevil v sekci „Folder content“ řádek s malým náhledem fotografie a názvem, jak lze vidět na Obrázku 14, ke kterému byl přiřazen řetězec, s aktuálním datem a časem pořízení kvůli eliminaci vzniku fotografie se stejným názvem.



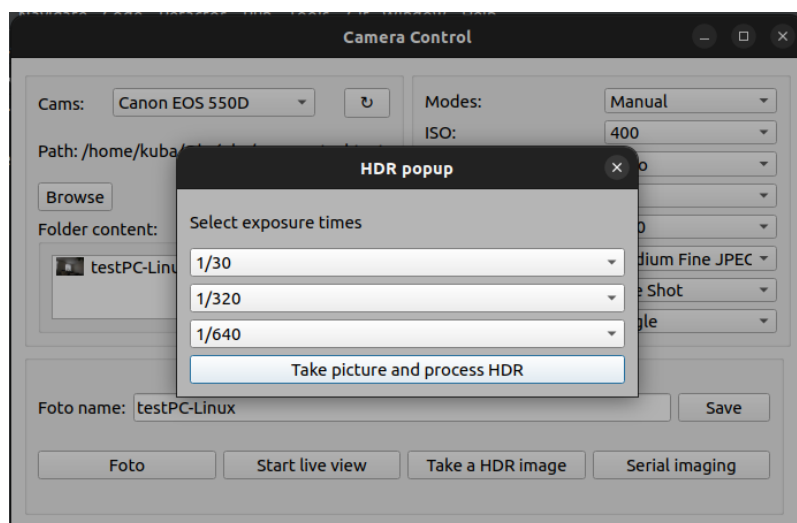
Obrázek 14 Ukázka vypsání obsahu složky po akvizici snímku

Další testovanou funkcí bylo spuštění živého náhledu přes tlačítko „Start live view“. Po zmáčknutí tlačítka a prodlevě asi tři vteřiny spustí kamera sekvenční snímání, jehož výstup je zobrazen po dalších dvou vteřinách v novém vyskakovacím okně. Vyskakovací okno s živým zobrazením lze ukončit stisknutím klávesy „esc“. Výstup živého zobrazení je zpožděn asi o dvě vteřiny a rychlost zobrazování je jeden snímek za sekundu. Na Obrázku 15 lze vidět, jak vypadá vyskakovací okno živého zobrazení.



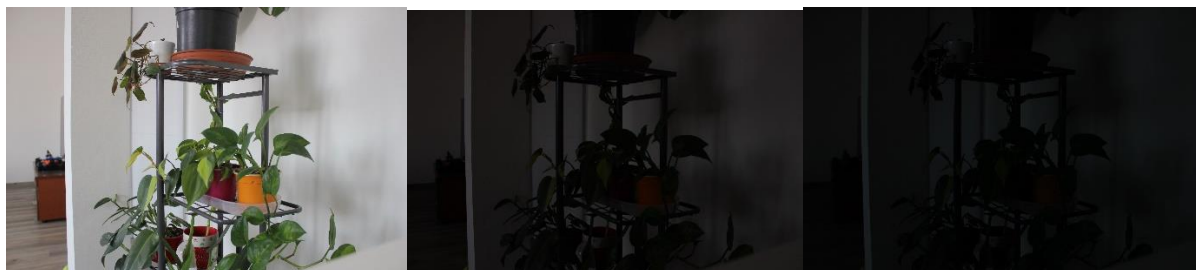
Obrázek 15 Vyskakovací okno živého zobrazení

Pro otestování funkce pořízení HDR fotografie bylo potřeba přepnout mód kamery na „Manual“, který umožňuje změnu parametru rychlosti uzávěrky – shutter speed. Proces změny módu kamery z „Auto“ na „Manual“ trval dvacet pět vteřin. Rozdílné délky změn módů kamery jsou způsobeny počtem hodnot nastavení, které je potřeba načíst. Po změně módu, nastavení parametrů, kontrole nastavení parametrů pomocí živého zobrazení a kliknutí na tlačítko „Take a HDR image“ se objevilo vyskakovací okno ve kterém byly zvoleny délky expozice jednotlivých snímků, ze kterých byl poté zpracován HDR obraz. Vyskakovací okno společně se zvolenými hodnotami expozice lze vidět níže na Obrázku 16.

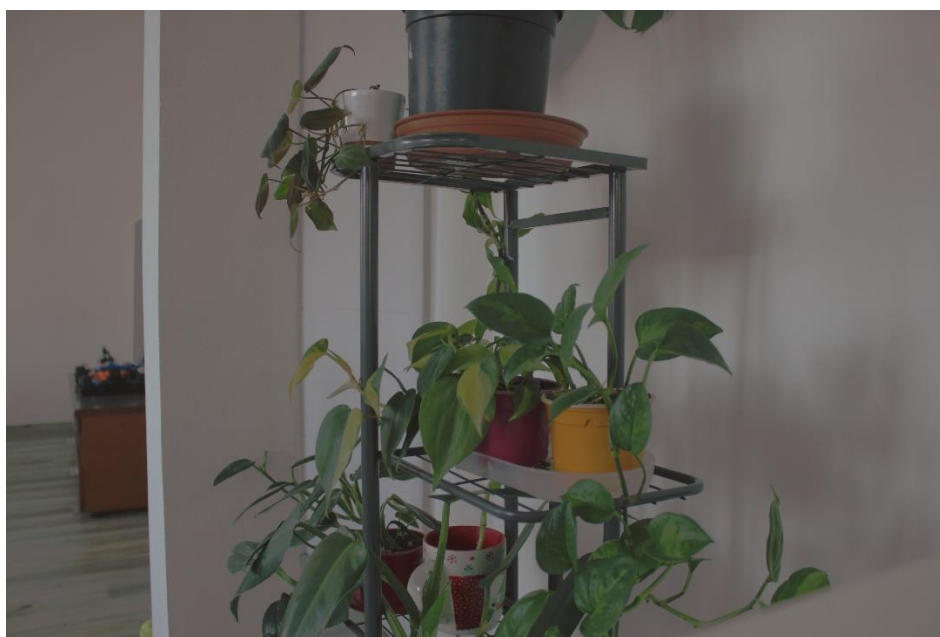


Obrázek 16 Ukázka vyskakovacího okna pro pořízení HDR snímku a nastavení hodnot – Linux PC

Akvizice snímků pro zpracování HDR trvala osm vteřin, po kterých byly další čtyři vteřiny snímky zpracovávány. Jednotlivé snímky byly uloženy do složky s názvem začínajícím na „hdr“ a pokračujícím hodnotou aktuálního času. Výsledný HDR snímek byl pak uložen do zvolené složky pro ukládání fotografií pod stejným názvem jako složka.



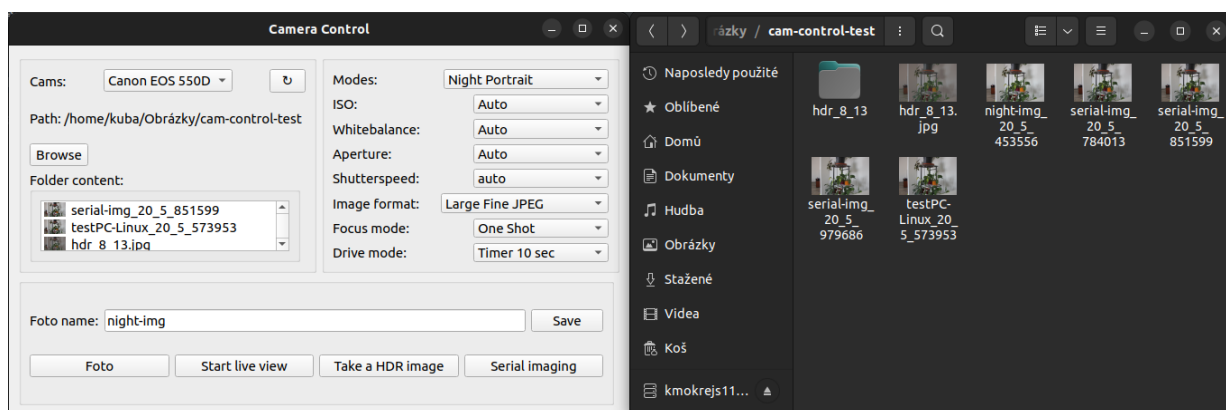
Obrázek 17 Snímky s odlišnou hodnotou expozice pořízené pro zpracování HDR fotografie.



Obrázek 18 Výsledný HDR snímek

Dalším krokem v testování bylo spuštění funkce pro sériového snímání pomocí tlačítka „Serial imaging“. Před spuštěním této funkce byl upraven název fotografie na „serial-img“. Dále se změnil režim snímání ze „Single“ na „Continuous“ a režim zaostření na „AI Focus“ pro lepší ostření při rychlém snímání. Proces sériového snímání trval devět vteřin, po kterých se aktualizovala sekce „Folder content“ a pořízené fotografie se zde zobrazily.

Poslední krok testování zahrnoval změnu módu na „Night Portrait“, nastavení časovače na deset sekund a změna názvu fotografie na „night-img“ pro přehlednost. Po uskutečnění těchto změn byla pořízena fotografie. Následně byla provedena kontrola pořízených fotografií v cílové složce. Složka s fotografiemi je přílohou této práce.



Obrázek 19 Nastavení kamery pro snímání s časovačem a obsah složky

6.3.2 Ovládání přes Raspberry Pi

Druhý testovaný způsob ovládání software zahrnoval využití mikropočítače Raspberry Pi. Pro testování bylo využito modelu Raspberry Pi 4 Model 4 Rev 1.1. Jako operační systém byl zvolen Raspberry Pi OS. Pro plnohodnotné využívání Raspberry Pi bylo potřeba připojit myš, klávesnici a monitor. K připojení klávesnice a myši byly využity dva ze čtyř USB portů a k připojení externího monitoru byl využit jeden z dvou mikro HDMI portů.

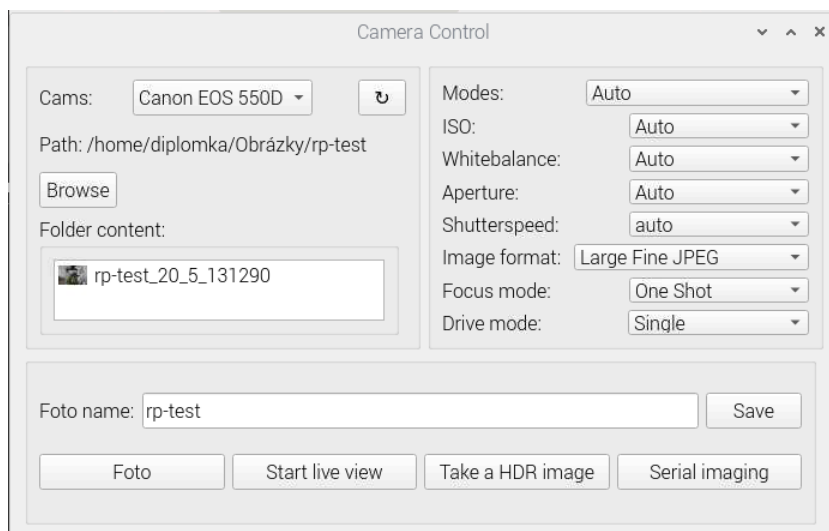
Tabulka 6.3 Technické specifikace používaného Raspberry Pi

Model	Raspberry Pi 4 Model 4 Rev 1.1
Procesor	Broadcom BCM2711 1.5 GHz quad-core
Operační paměť	4 GB
Kapacita paměti	16 GB microSD karta
Rozhraní	2x USB 2.0, 2x USB 3.0, Bluetooth, Wi-Fi, 2x microHDMI
Operační systém	Linux – Raspberry Pi OS

Po připojení periférií, instalaci operačního systému a knihoven byl z GitHubu naklonován repositář se složkou projektu. Ke spuštění skriptu bylo využito vývojového prostředí Visual Studio Code. Ověření správné instalace bylo provedeno spuštěním skriptu „gui.py“ a následným načtením GUI software.

Jakmile bylo GUI načteno, tak byla pomocí USB kabelu připojena DSLR kamera, a zmáčknuto tlačítko pro obnovu seznamu připojených kamer. Načtení kamery a hodnot nastavení trvalo necelé dvě minuty a patnáct vteřin. Jako další krok byla zvolena složka pro uložení pořízených fotografií, načej se vypsala cesta ke složce do sekce „Path:“. Změna módu z „P“ na Auto“ poté proběhla za deset vteřin. Jako název

snímku byl zvolen řetězec „rp-test“ a pomocí tlačítka „Save“ byl tento název uložen. Přes tlačítka „Foto“ byl pořízen testovací snímek, jehož akvizice trvala tři sekundy. Na Obrázku 20 je vyobrazen aktuální stav aplikace i s vypsáním názvu pořízeného snímku v sekci „Folder content:“.

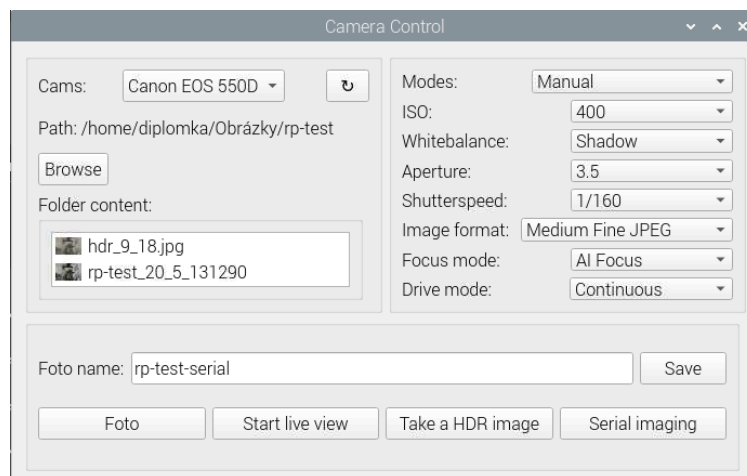


Obrázek 20 Ukázka GUI na mikropočítači Raspberry Pi

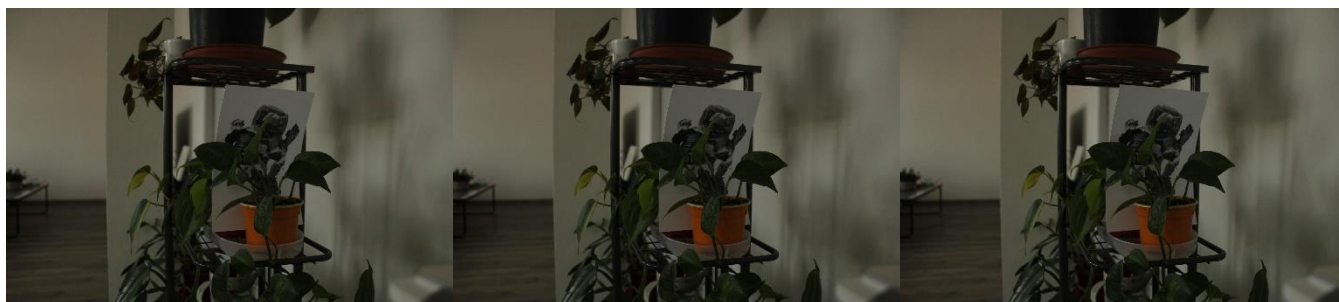
Po spuštění živého vysílání začala kamera sekvenčně snímat během dvou sekund, zatímco oknu s náhledem trvalo načtení deset sekund. Snímkovací frekvence byla zhruba 1 fps. Dalším krokem bylo opětovaná změna módu kamery na manuální režim ovládání což zabralo čtyřicet sekund. Následující změny nastavení kamery jako hodnota ISO a formát výstupní fotografie trvala každá mezi dvěma až třemi sekundami.

Vyskakovací okno, pro pořízení HDR snímku se otevřelo do dvou sekund. Po nastavení hodnot rychlosti uzávěrky a potvrzení pořízení snímku přes tlačítka „Take picture and process HDR“ trvala akvizice prvního snímku čtyři sekundy, zatímco pořízení dalších dvou snímků trvalo sedm sekund, v nichž se v destinaci zároveň vytvořila složka pro uložení pořízených snímků. Toto zpomalení při pořizování snímků může být způsobeno pomalejším procesem změny hodnot na fotoaparátu. Následné zpracování HDR snímku a uložení do zvolené destinace trvalo patnáct sekund.

Před otestováním funkce sériového snímání se změnil název fotografie na „rp-test-serial“ a hodnoty módu zaostření a módu snímání byly změněny na hodnoty „AI Focus“ a „Continuous“, Poté se přes tlačítka „Serial imaging“ spustila funkce pro sériové snímání, které trvalo dvanáct vteřin pořízení tří snímků v řadě. Pořízené fotografie byly ihned zobrazeny v sekci „Folder content:“



Obrázek 21 Nastavení kamery před sekvenčním snímáním



Obrázek 22 Snímky pořízené sekvenčním snímáním

Pro poslední krok byl změněn mód kamery stejně jako v předchozím případě na „Night Portrait“, což trvalo třicet vteřin, a časovač byl nastaven na deset sekund. Déle byl změněn název fotografie na „night-portrait“. Po stisknutí tlačítka „Foto“ a uplynutí deseti sekund byla pořízena fotografie, která se ihned uložila do složky. Kontrola obsahu složky zahrnovala zobrazení každého snímku a zhodnocení kvality. Pořízené fotografie z testování touto metodou lze také najít v příloze této práce.

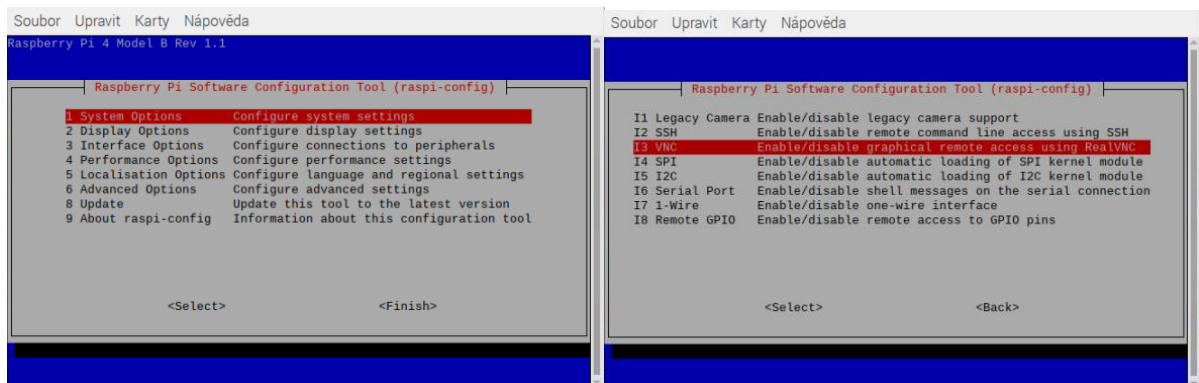


Obrázek 23 Snímek pořízený módem „Night Portrait“

6.3.3 Vzdálené ovládání přes Raspberry Pi

Poslední testovaná možnost ovládání software využívá Raspberry Pi jako prostředníka pro vzdálené ovládání což je naznačeno v grafu na Obrázku 12. Tato metoda ovládání dokáže eliminovat potřebu připojení periférií a uživatel tak může využívat software i na počítačích s operačním systémem Windows. Před plnohodnotným využitím vzdáleného ovládání je však nutné vytvořit připojení z počítače, který bude ovládat software k Raspberry Pi. Tato podkapitola se tak bude zabývat prvotním nastavením vzdáleného ovládání Raspberry Pi přes počítač s operačním systémem Windows.

První krok k vytvoření vzdáleného připojení zahrnoval povolení VNC (Virtual Network Computing) v nastavení Raspberry Pi. Povolení bylo provedeno pomocí příkazového řádku, kam se zadal příkaz `sudo raspi-config`, který spustil konfigurační nástroj. V tomto nástroji se poté navigovalo na řádek „Interference Control“, který po zvolení poskytl výčet všech možností propojení. Zde se zvolil řádek s VNC a potvrdilo se povolení. Tato změna nastavení se poté potvrdila pomocí tlačítka „Finish“.



Obrázek 24 Povolení VNC na Raspberry Pi

Nyní bylo potřeba zjistit IP adresu Raspberry Pi kvůli následnému připojení. IP adresa byla zjištěna přes příkazový řádek po zadání příkazu „ifconfig“. Adresa se nacházela v sekci „wlan0“ pod řetězcem „inet“. Adresu si bylo potřeba zaznamenat pro použití v dalším kroku.

```
Soubor Upravit Karty Nápověda
diplomka@raspberrypi:~$ ifconfig
eth0: flags=4099<AKTIVOVÁNO,VŠESMĚR,MULTICAST> mtu 1500
    ether dc:a6:32:49:ee:96 délka_odchozí_fronty 1000 (Ethernet)
    RX packetů 0 bajtů 0 (0,0 B)
    RX chyb 0 zahozeno 0 přetečení 0 rámců 0
    TX packetů 0 bajtů 0 (0,0 B)
    TX chyb 0 zahozeno 0 přetečení 0 přenos 0 kolizí 0

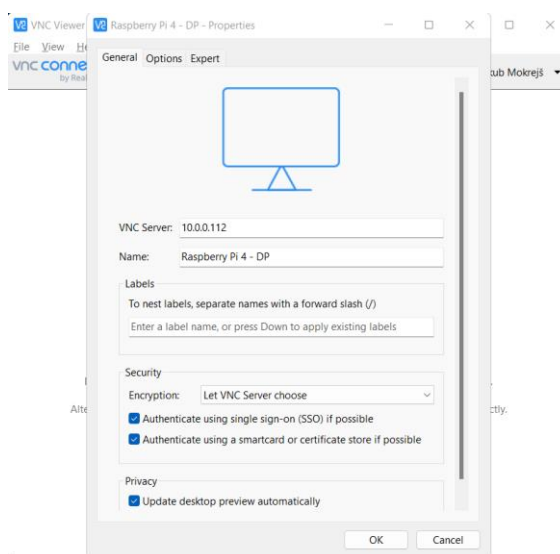
lo: flags=73<AKTIVOVÁNO,SMYČKA,BĚŽÍ> mtu 65536
    inet 127.0.0.1 síťová_maska 255.0.0.0
    inet6 ::1 délka_prefixu 128 scopeid 0x10<stroj>
    loop délka_odchozí_fronty 1000 (Místní smyčka)
    RX packetů 368 bajtů 1015894 (992,0 KiB)
    RX chyb 0 zahozeno 0 přetečení 0 rámců 0
    TX packetů 368 bajtů 1015894 (992,0 KiB)
    TX chyb 0 zahozeno 0 přetečení 0 přenos 0 kolizí 0

wlan0: flags=4163<AKTIVOVÁNO,VŠESMĚR,BĚŽÍ,MULTICAST> mtu 1500
    inet 10.0.0.112 síťová_maska 255.255.255.0 všesměr 10.0.0.255
    inet6 fe80::ff15:7e32:b909:de75 délka_prefixu 64 scopeid 0x20<linka>
    ether dc:a6:32:49:ee:97 délka_odchozí_fronty 1000 (Ethernet)
    RX packetů 61755 bajtů 56033153 (53,4 MiB)
    RX chyb 0 zahozeno 0 přetečení 0 rámců 0
    TX packetů 55446 bajtů 11807467 (11,2 MiB)
    TX chyb 0 zahozeno 0 přetečení 0 přenos 0 kolizí 0
```

Obrázek 25 Výpis v příkazovém řádku po zadání příkazu „ifconfig“. Obrázek byl barevně invertován

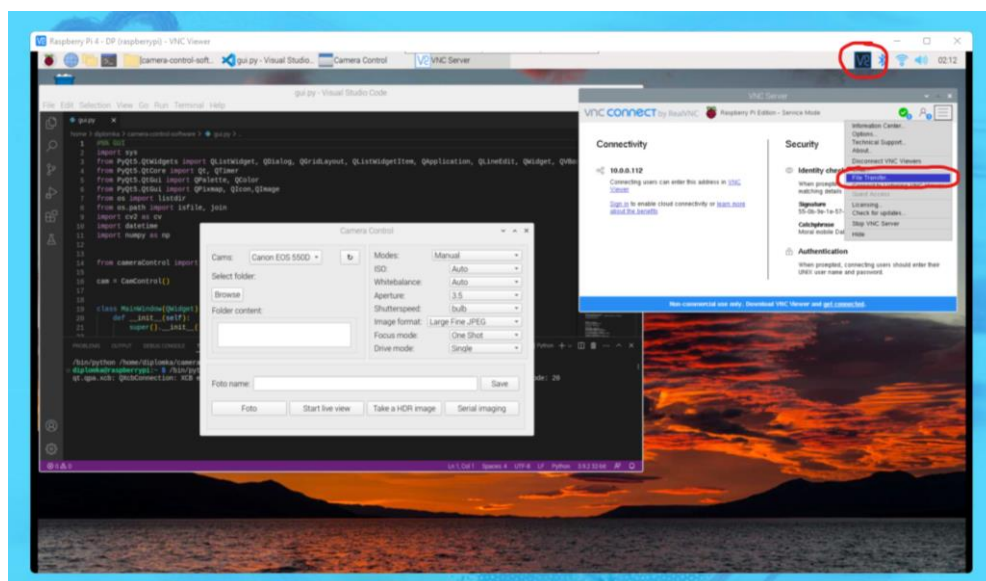
Pro nastavení vzdáleného ovládní bylo potřeba stáhnout a nainstalovat program VNC Viewer [26] na počítač, který bude Raspberry Pi ovládat, což obnášelo, založení účtu pomocí emailu. Po instalaci, spuštění a přihlášení uživatele bylo potřeba kliknout na záložku „File“ v levém horním rohu a z nabídky vybrat volbu „New connection“. Ve vyskakovacím okně se poté zadala IP adresa Raspberry Pi do políčka „VNC Server“ a

název připojení v poličku „Name“. Potvrzení proběhlo stiskem tlačítka „OK“ jak je znázorněno na Obrázku 26.



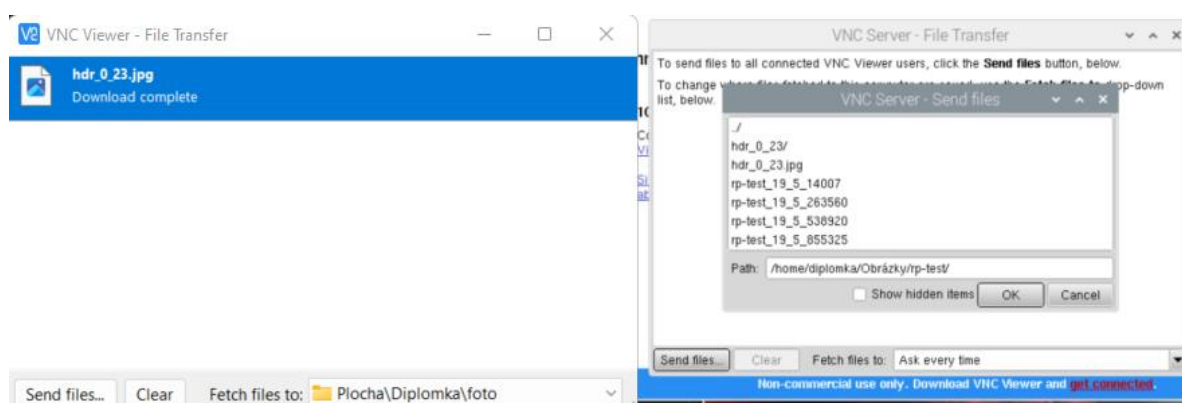
Obrázek 26 Ukázka nastavení parametrů v programu VNC Viewer

Po potvrzení nastavení bylo vytvořeno nové připojení, které se dvojným kliknutím spustilo. Před připojením k Raspberry Pi bylo ještě potřeba zadat jméno a heslo k mikropočítači. Po správném zadání těchto údajů, se zobrazila plocha Raspberry Pi, což indikovalo úspěšné připojení a možnost ovládat mikropočítač.



Obrázek 27 Ukázka zřízeného přenosu souborů

Program VNC Viewer dále umožňuje přenášení souborů z Raspberry Pi do počítače, který ho ovládá. Přenos souborů byl inicializován kliknutím na ikonu programu, která je zakroužkovaná na Obrázku 27, která vyvolala dialogové okno. Po zvolení sekce možností v pravém horním rohu dialogového okna a výběru „File Transfer“ se zobrazilo nové vyskakovací okno, které umožnilo výběr souboru, který chce uživatel odeslat. Zvolením souboru a kliknutím na tlačítko „Send files.“ se zvolený soubor odeslal do uživatelem zvolené složky na počítač, který ovládá Raspberry Pi. Dialogová okna, která se zobrazí při odesílání souboru jsou zobrazena na Obrázku 1.28.



Obrázek 28 Proces přenosu souborů pomocí VNC Viewer

Testování software pro vzdálené ovládání přes Raspberry Pi představovalo stejné časy jako při testování přímého propojení Raspberry Pi – DSLR kamera. Z tohoto důvodu nebylo provedeno plné testování funkčnosti software, ale pouze testování vzdáleného ovládání Raspberry Pi. Pro připojení byl využit počítač s operačním systémem Windows, protože je to operační systém, který využívá většina počítačů, a tak tento způsob představuje nejobecnější možnost vzdáleného připojení využívající Raspberry Pi.

7. DISKUSE

Z uživatelského hlediska je nejdůležitější jednoduchost použití každé testované metody ovládání. Metoda ovládání přes počítač s operačním systémem Linux se dá nazvat jako uživatelsky nejjednodušší, vzhledem k tomu, že stačí pouze propojit kameru s počítačem. U metody využívající Raspberry Pi je kromě první instalace operačního systému také nutné připojit periferie pro ovládání a zobrazení výstupu. U vzdáleného přístupu odpadá nutnost připojení periférií, avšak nastavení tohoto ovládání může být pro uživatele složité. Dalším aspektem u těchto metod je samotné propojení kamery s počítačem. Při propojení kabelem vzniká limitace, na jakou vzdálenost se může uživatel vzdálit od kamery. Tato limitace však neplatí pro vzdálené připojení, díky kterému může uživatel pohodlně kameru ovládat i z jiné místnosti. Raspberry Pi je také podstatně levnější než většina počítačů, a tak je tato metoda i dostupnější. Dostupnost je však vykompenzována nižším výkonem oproti běžným počítačům, což se může dále projevit i na funkčnosti software. Mezi další výhody Raspberry Pi patří jeho rozměry, které umožňují efektivní úsporu místa.

Mezi jedno z nejdůležitějších kritérií z pohledu mikroskopického snímání patří čas, který je potřeba pro změnu parametrů kamery a pořízení požadovaného snímku. Níže uvedená tabulka porovnává časy jednotlivých úkonů provedených při testování.

Tabulka 1.4 Porovnání časů úkonů pro testované způsoby ovládání

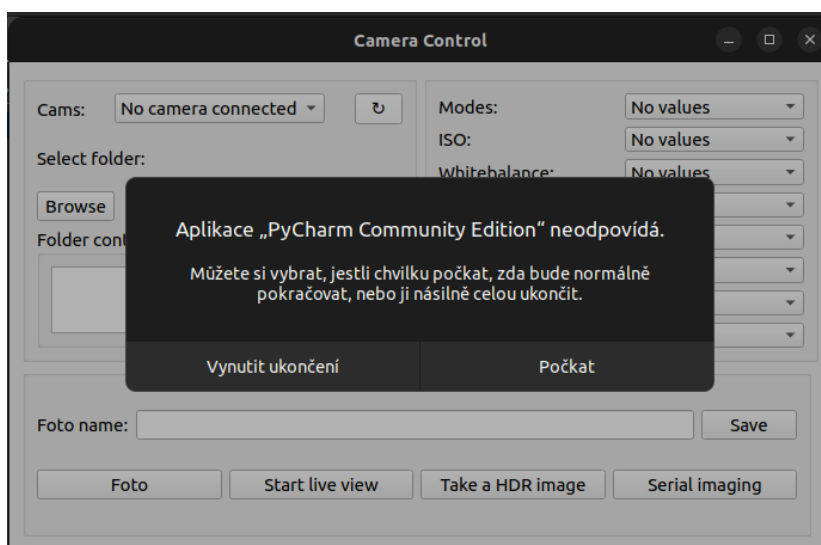
Způsob ovládání	Linux PC	Raspberry Pi 4
Načtení kamery	65 s	135 s
Změny módu	9–25 s	10–40 s
Akvizice snímku	2 s	3 s
Spuštění živého náhledu	5 s	12 s
Změna parametrů kamery	2 s	3 s
HDR snímání	12 s	32 s
Sekvenční snímání	9 s	12 s

Z Tabulky 1.4, lze vyčíst, že časy nastavování parametrů i akvizice snímků jsou u metody využívající Raspberry Pi až dvakrát větší. Tyto mohou být způsobeny především horšími hardwarovými specifikacemi u Raspberry Pi oproti počítači s operačním systémem Linux. Rozptyl časů při změnách módu kamery je způsoben především rozdílnými požadavky na načtení hodnot pro jednotlivé módy. To znamená, že například při načítání módu „Manual“ je potřeba načíst všechny parametry nastavení a jejich hodnoty, zatímco při načítání módu „Auto“ je většina parametrů již nastavena na automatickou hodnotu a uživatel nemá možnost měnit takový počet parametrů jako při manuálním ovládání.

Další pozorování založené na testování metod ovládání odhalilo, že spuštění živého náhledu a následná snímkovací frekvence je ovlivněna aktuálním nastavením kamery. Pokud má tedy kamera nastavenou delší expozici, tak se snímkovací frekvence podle tohoto parametru sníží. Naopak při rychlé expozici se snímkovací frekvence může zvýšit. Pro živý náhled je důležité zobrazení výstupu kamery s aktuálním nastavením což v tomto případě splňuje i za cenu snížené obnovovací frekvence.

Vzhledem k delším časům trvání některých operací a funkcí bych pro případný budoucí vývoj software navrhl přidání indikátoru probíhající operace. Tato indikace by mohla být realizována například pomocí vyskakovacího okna, které by obsahovalo zprávu pro uživatele o tom, že operace právě probíhá. Dalším řešením by pro tento problém bylo využitím točícího se kolečka. Tato funkce by se hodila pro operace jako je načtení kamery, změny módu kamery, spuštění živého náhledu, HDR snímání a také pro spuštění sekvenčního snímání. Při těchto operacích se v momentálním stavu software grafické uživatelské rozhraní zamrzne. To znamená, že například operace procesu zpracování HDR snímku je spuštěna na pozadí a uživatel po dobu této operace nemůže interagovat s GUI. Přidáním indikace probíhající operace by tak uživatel měl větší přehled o tom, že operace stále probíhá a přesně by věděl, kdy může znovu začít interagovat s GUI. Toto vylepšení by software udělalo více uživatelsky přívětivý.

V průběhu testování na počítači s operačním systémem Linux došlo k situaci, kdy se při průběhu operace na pozadí, konkrétně načtení kamery, objevilo dialogové okno s chybovou hláškou, že aplikace neodpovídá. Toto dialogové okno lze vidět níže na Obrázku 29.



Obrázek 29 Dialogové okno s chybovou hláškou

Dialogové okno po konci operace zmizelo a načtení kamery proběhlo úspěšně. Jednou z příčin tohoto problému může být snaha o interakci s GUI v době, kdy na pozadí probíhá operace, která načítá parametry fotoaparátu do software. Jedním z možných řešení eliminace tohoto chování by mohlo být již výše zmíněná implementace vyskakovacího okna nebo rotujícího kolečka znázorňující aktivní proces na pozadí. Toto by zabránilo uživateli v interakci s GUI a vyvolání tohoto dialogového okna.

Vzhledem k časovým odezvám jednotlivých operací a snímkovací frekvence živého náhledu se software hodí spíše pro pozorování pomalých jevů v mikroskopii jako je růst nebo migrace buněk. Při pozorování rychlejších dějů by mohly časy změny módu, pořízení snímků nebo nízká snímkovací frekvence způsobit, že se nepodaří určitý děj zachytit včas. Pro bližší určení využití software v mikroskopii by bylo nutné software vyzkoušet s DSLR kamerou připevněnou na mikroskopu.

8. ZÁVĚR

Cílem této práce bylo navrhnout a realizovat software pro ovládání DSLR kamery s využitím v mikroskopii.

První část práce se zabývá řešením. Tato část byla rozdělena do několika podkapitol, které se zabývali nejdříve samotným využitím kamer v mikroskopii, poté popisem tak základních parametrů kamer, parametrů, které ovlivňují kvalitu výsledného obrazu, a nakonec porovnáním reálných parametrů mikroskopických kamer s DSLR kamerami.

Dalším cílem bylo seznámení se se software, které se využívají k obsluze mikroskopických kamer, popsání funkcí, kterými disponují. V této části byly popsány funkce a styl rozložení prvků tří vybraných software. Tato část posloužila jako odrazový můstek pro následný návrh rozložení a definici funkcí software pro ovládání DSLR kamer.

Posledním cílem první části práce bylo navržení programového vybavení s grafickým uživatelským rozhraním pro obsluhu DSLR. Této části práce předcházela řešení technologií, které mohou být využity pro vývoj software. Zde se definovali technologie pro back-end a front-end, které umožňují vytvoření software pro ovládání DSLR kamer. Na základě těchto technologií byl vytvořen návrh grafického uživatelského rozhraní, který byl dále rozdělen do tří bloků. Funkce jednotlivých bloků pak byly popsány v jednotlivých podkapitolách.

V praktické část práce se nejprve definovaly technologie využívané k tvorbě software a poté byly tyto technologie využity k realizaci. K ovládání a komunikaci s kamerou je využito funkcí napsaných v programovacím jazyku Python, které využívají knihovnu gPhoto2. Realizace dále navázala tvorbou grafického uživatelského prostředí.

Pro testování bylo využito tří způsobů ovládání software. Jeden ze způsobů zahrnoval použití Raspberry Pi, další počítače s operačním systémem Linux a poslední využíval vzdálené připojení na Raspberry Pi přes počítač s operačním systémem Windows. Testována byla nejen celková funkčnost software, ale byly také měřeny časy potřebné k jednotlivým úkonům.

V navazující diskusi jsou jednotlivé způsoby ovládání a funkčnosti zhodnoceny společně s výsledky měření časů. Diskuse zmiňuje i problémy, které nastaly při testování a poskytuje možné řešení těchto problémů.

V budoucnu by tak bylo možné software rozšířit o funkcionalitu, která uživateli indikuje probíhající proces na pozadí a zajistí, aby uživatel nemohl interagovat se software.

LITERATURA

- [1] DEES, Justine. *What camera options are there for microscopes?. Joyful Microbe* [online]. Justine Dees, 2022, 25. května 2022 [cit. 2022-12-27]. Dostupné z: <https://joyfulmicrobe.com/camera-options-for-microscopes/>
- [2] KURODA, Takao. *Essential principles of image sensors*. Boca Raton: CRC Press, [2015]. ISBN 978-1-4822-2005-6.
- [3] *CCD CAMERAS VS CMOS CAMERAS. Meiji Techno* [online]. San Jose: Meiji Techno, c1964-2018 [cit. 2022-12-27]. Dostupné z: <https://meijitechno.com/ccd-cameras-vs-cmos-cameras/>
- [4] SEIDLEIN, Stefan. *Exploding the myths of digital microscopy: Is CCD always better than CMOS?. JENOPTIK* [online]. Jena: JENOPTIK, 2020, 13. února 2020 [cit. 2022-12-27]. Dostupné z: <https://www.jenoptik.com/news/blog/articles/2020/02/13/myths-of-digital-microscopy-ccd-better-than-cmos>
- [5] KEAL, Louis. *What is Scientific Imaging Quality?. Teledyne photometrics* [online]. Tucson: Teledyne photometrics, 2020, 19. listopadu 2020 [cit. 2022-12-27]. Dostupné z: <https://www.photometrics.com/wp-content/uploads/2021/01/Scientific-Image-Quality-A3-19-11-2020.pdf>
- [6] OGAMA, Takeo. *What to Consider When Choosing a Microscope Camera. EVIDENT* [online]. Tokyo: EVIDENT [cit. 2022-12-27]. Dostupné z: <https://www.olympus-lifescience.com/en/resources/white-papers/choosing-a-microscope-camera/>
- [7] *Cameras for Fluorescence Microscopy. Basler* [online]. Ahrensburg: Basler, c2022 [cit. 2022-12-27]. Dostupné z: <https://www.baslerweb.com/en/vision-campus/markets-and-applications/cameras-for-fluorescence-microscopy/>
- [8] *Practical test: Nikon D800. LMscope* [online]. Graz: MICRO TECH LAB, c1999-2022, 24. 9. 2012 [cit. 2022-12-27]. Dostupné z: https://www.lmscope.com/en/Nikon_D800_on_microscope_en.html
- [9] *With so many choices, it's hard to decide – which camera system is best suited to my microscopy application?. LMscope* [online]. Graz: MICRO TECH LAB, 2015, 14. 4. 2015 [cit. 2022-12-27]. Dostupné z: https://www.lmscope.com/en/camera_choice_microscopy_en.html
- [10] LIU, Yutao, Guangtao ZHAI, Debin ZHAO a Xianming LIU. *Frame Rate and Perceptual Quality for HD Video* [online]. *Advances in Multimedia Information Processing*, 2015, [cit. 2022-12-27]. Dostupné z: https://www.researchgate.net/publication/282967265_Frame_Rate_and_Perceptual_Quality_for_HD_Video
- [11] *Microscope Software Platform LAS X Life Science. Leica Microsystems* [online]. Wetzlar: Leica Microsystems, c2022 [cit. 2022-12-27]. Dostupné z: <https://www.leica-microsystems.com/products/microscope-software/p/leica-las-x-ls/>

- [12] *ThorCam. THORLABS* [online]. Newton: THORLABS, c2023, 2011 [cit. 2022-12-27]. Dostupné z: https://www.thorlabs.com/software_pages/ViewSoftwarePage.cfm?Code=ThorCam
- [13] HUA, Low Jun. *How to use LAS EZ. ProfCFYP* [online]. ProfCFYP, c2022 [cit. 2022-12-27]. Dostupné z: <https://profcfyp.dozuki.com/Guide/How+to+use+LAS+EZ/13>
- [14] *Python* [online]. Python, c2001-2022 [cit. 2022-12-27]. Dostupné z: <https://www.python.org/doc/essays/blurb/>
- [15] CASSADY, Jacob Taylor. *DigiCam. Github* [online]. 2014, 19 November 2019 [cit. 2022-12-27]. Dostupné z: <https://github.com/jtcass01/DigiCam>
- [16] EASTERBROOK, Jim. *Python-gphoto2. Github* [online]. 2014, 25 March 2014 [cit. 2022-12-27]. Dostupné z: <https://github.com/jim-easterbrook/python-gphoto2>
- [17] VERNEROVÁ, Sára. *Front end vs. Back end - jaký je mezi nimi rozdíl? APITREE* [online]. Praha: ApiTree, 2021, 13. 9. 2021 [cit. 2022-12-27]. Dostupné z: <https://www.apitree.cz/blog/front-end-vs-back-end-jaky-je-mezi-nimi-rozdil>
- [18] COSTA, Claire D. *Top 10 Python GUI Frameworks for Developers. Towards Data Science* [online]. Towards Data Science, 2020, 7 December 2020 [cit. 2022-12-27]. Dostupné z: <https://towardsdatascience.com/top-10-python-gui-frameworks-for-developers-adca32fbe6fc>
- [19] NEDERKOORN, Cordny. *Top 10 Python GUI Frameworks Compared. ActiveState* [online]. Vancouver: ActiveState, 2022, 9 August 2022 [cit. 2022-12-27]. Dostupné z: <https://www.activestate.com/blog/top-10-python-gui-frameworks-compared/>
- [20] KOPF, Ben. *The Power of Figma as a Design Tool. Toptal* [online]. New York: Toptal, c2010-2022 [cit. 2022-12-27]. Dostupné z: <https://www.toptal.com/designers/ui/figma-design-tool>
- [21] MING YOU, Ang, Ching SIEH YUAN, Francis TAM a Pua XUAN ZHAN. *GUI Programming. NUS computing* [online]. Singapore: NUS School of Computing, 2012, April 2012 [cit. 2022-12-27]. Dostupné z: <https://www.comp.nus.edu.sg/~seer/book/2e/Ch12.%20Graphical%20User%20Interfaces.pdf>
- [22] *DigiCamControl. SOURCEFORGE* [online]. San Diego: SOURCEFORGE, c2022, 2011 [cit. 2022-12-27]. Dostupné z: <https://sourceforge.net/projects/digicamcontrol/>
- [23] *JetBrains. PyCharm: the Python IDE for Professional Developers by JetBrains* [online]. Prague JetBrains c2023, 2023 [cit. 2023-05-20]. Dostupné z: <https://www.jetbrains.com/pycharm/>
- [24] *gPhoto – Command-Line Interface. gPhoto* [online]. gPhoto Project, c2023, 2023 [cit. 2023-05-18] Dostupné z: <http://www.gphoto.org/doc/manual/ref-gphoto2-cli.html>

- [25] KMOKREJS, Jakub Mokrejš. *Camera-control-software*. *Github* [online]. 2023 [cit. 2023-05-23]. Dostupné z: <https://github.com/kmokrejs/camera-control-software>
- [26] *RealVNC*. VNC Viewer | RealVNC [online]. Cambridge: RealVNC Ltd., c2023, 2023 [cit 2023-05-19]. Dostupné z: <https://www.realvnc.com/en/connect/download/viewer/>

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

DSLR	Fakulta elektrotechniky a komunikačních technologií
CCD	Vysoké učení technické v Brně
CMOS	Complementary Meta-Oxide-Semiconducto
A/D převodník	Analogově digitální převodník
SNR	Signal to Noise Ratio
FPS	frames per second
nm	nanometr
s	sekunda

SEZNAM PŘÍLOH

V příloženém souboru naleznete veškeré skripty potřebné ke spuštění software. Dále se zde nachází fotografie pořízené při testování software.

