



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

PŘÍPRAVEK PRO SNÍMÁNÍ MUZEJNÍ SBÍRKY

DEVICE FOR SCANNING THE MUSEUM COLLECTION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Lukáš Drapač

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Peter Honec, Ph.D.

BRNO 2022



Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Lukáš Drapač

ID: 209364

Ročník: 3

Akademický rok: 2021/22

NÁZEV TÉMATU:

Přípravek pro snímání muzejní sbírky

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a realizovat přípravek pro digitalizaci muzejní sbírky kraslic a vytvoření nástroje pro prohlížení digitálního archivu.

1. Seznamte se s principem plošných a řádkových kamer.
2. Navrhněte vhodné hardwarové vybavení pracoviště pro snímání kraslic (kamera, osvětlení, uchycení, rotace) s využitím web-kamery.
3. Vytvořte takové pracoviště a nasnímejte dodané vzorky.
4. Vytvořte aplikaci pro reprezentaci dat (3D prohlížečka).
5. Otestujte a zhodnoťte.

DOPORUČENÁ LITERATURA:

HLAVAC V., SONKA M., BOYLE R.: Image Processing, Analysis, and Machine Vision, ISBN 978-0495082521

Termín zadání: 7.2.2022

Termín odevzdání: 23.5.2022

Vedoucí práce: Ing. Peter Honec, Ph.D.

doc. Ing. Václav Jirsík, C.Sc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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

Bakalářská práce se zabývá návrhem a konstrukcí přípravku, který usnadní digitalizaci a vylepší způsob digitalizace muzejní sbírky kraslic, zejména z hlediska efektivity při digitalizaci velkého množství exponátů a prohlížení kraslic. Součástí práce je návrh přípravku, jehož tělo bylo vytisknuto na 3D tiskárně, a hardwarového vybavení. Pro ovládání přípravku a zobrazení 3D modelů kraslic byly vytvořeny samostatné programy v jazycích C++, C# a Java. Přípravek po spuštění pořídí snímky obvodu kraslice a zpracuje je. Výstupem jsou 3D modely kraslic, které je možné procházet a zobrazovat ze všech stran v galerii.

Klíčová slova

digitalizace, digitalizace muzejní sbírky, digitalizace kraslic, automatická digitalizace, 3D model kraslice, AForge.NET, Processing

Abstract

This bachelor thesis deals with designing and construction of the device, which should improve the digitalization of the museum collection of Easter eggs in terms of effectiveness and data presentation. This thesis presents design of the device, which was printed on a 3D printer, and the hardware components of the device. Programs for controlling the device and rendering 3D models of Easter eggs were written in C++, C# and Java. When started, the device takes pictures around the perimeter of the Easter egg and processes them. The results are 3D models, which can be viewed from Gallery in the desktop application.

Keywords

digitalization, digitalization of the museum collection, digitalization of Easter eggs, Easter egg 3D model, automated digitalization, AForge.NET, Processing

Bibliografická citace

DRAPAČ, Lukáš. *Přípravek pro snímání muzejní sbírky*. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/142665>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Peter Honec.

Prohlášení autora o původnosti díla

| | |
|-----------------------------------|--------------------------------------|
| Jméno a příjmení studenta: | Lukáš Drapač |
| VUT ID studenta: | 209364 |
| Typ práce: | Bakalářská práce |
| Akademický rok: | 2021/22 |
| Téma závěrečné práce: | Přípravek pro snímání muzejní sbírky |

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucího závěrečné 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 závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné 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.

V Brně dne: 23.5.2022

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Peteru Honcovi, PhD. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: 23.5.2022

podpis autora

Obsah

| | |
|---|-----------|
| SEZNAM OBRÁZKŮ | 14 |
| ÚVOD | 15 |
| 1. PRINCIP PLOŠNÝCH A ŘÁDKOVÝCH KAMER..... | 16 |
| 1.1 CCD..... | 17 |
| 1.2 CMOS SENZOR | 18 |
| 1.3 ELEKTRONICKÁ ZÁVĚRKA | 18 |
| 1.4 PLOŠNÉ KAMERY | 19 |
| 1.5 ŘÁDKOVÉ KAMERY | 20 |
| 2. KONSTRUKCE PŘÍPRAVKU..... | 21 |
| 2.1 KROKOVÝ MOTOR | 21 |
| 2.2 DRIVER A3967 | 22 |
| 2.3 WEBKAMERA | 23 |
| 2.4 SNÍMAČ VZDÁLENOSTI VL53L1X | 23 |
| 2.5 TĚLO PŘÍPRAVKU | 24 |
| 2.6 VZDÁLENOST WEBKAMERY OD KRASLICE | 25 |
| 2.7 OSVĚTLENÍ KRASLICE..... | 26 |
| 3. SOFTWAREVÁ ČÁST PŘÍPRAVKU | 27 |
| 3.1 KOMUNIKACE MEZI POČÍTAČEM A DESKOU ARDUINO | 27 |
| 3.2 UŽIVATELSKÉ ROZHŘANÍ | 28 |
| 3.3 OBSLUHA WEBKAMERY | 30 |
| 3.4 PRŮBĚH CYKLU SNÍMÁNÍ KRASLICE | 31 |
| 3.5 UKLÁDÁNÍ SNÍMKŮ | 32 |
| 3.6 ZPRACOVÁNÍ SNÍMKŮ METODOU SKLÁDÁNÍ OBDÉLNÍKŮ..... | 32 |
| 3.7 VÝSLEDEK ZPRACOVÁNÍ SNÍMKŮ METODOU SKLÁDÁNÍ OBDÉLNÍKŮ | 34 |
| 3.8 DALŠÍ MOŽNOSTI ZPRACOVÁNÍ SNÍMKŮ | 37 |
| 3.9 TVAR KRASLICE | 38 |
| 3.10 ZOBRAZENÍ VÝSLEDNÉHO MODELU | 38 |
| 3.11 SPOUŠTĚNÍ SCRIPTU S MODELEM KRASLICE..... | 40 |
| 3.12 POŽADAVKY NA VYBAVENOST UŽIVATELE | 40 |
| 4. POROVNÁNÍ ZPŮSOBŮ DIGITALIZACE SBÍRKY | 41 |
| 4.1 PŮVODNÍ ZPŮSOB DIGITALIZACE | 41 |
| 4.2 DIGITALIZACE POMOCÍ PŘÍPRAVKU | 42 |
| 5. ZÁVĚR..... | 44 |
| LITERATURA..... | 45 |
| SEZNAM PŘÍLOH..... | 46 |

SEZNAM OBRÁZKŮ

| | | |
|------|---|----|
| 1.1 | Fotoelektrický jev [1] | 16 |
| 1.2 | Bayerova maska [2]..... | 16 |
| 1.3 | Průchod světla mikroskopickou čočkou a filtry [3]..... | 17 |
| 1.4 | Odečítání hodnot náboje u CCD obrazového snímače [4] | 17 |
| 1.5 | CMOS snímač [3]..... | 18 |
| 1.6 | Porovnání globální (vlevo) a rolující (vpravo) závěrky [6]..... | 18 |
| 1.7 | Rolující závěrka [5]..... | 19 |
| 1.8 | Globální závěrka [5]..... | 19 |
| 1.9 | Princip řádkové kamery [7]..... | 20 |
| 2.1 | Blokový diagram přípravku..... | 21 |
| 2.2 | Driver A3967 [8]..... | 22 |
| 2.3 | Zapojení krokového motoru a driveru A3967 | 22 |
| 2.4 | Zapojení snímače vzdálenosti VL53L1X..... | 23 |
| 2.5 | Tělo přípravku..... | 24 |
| 2.6 | Sestava zajišťující rotaci | 25 |
| 2.7 | Snímek pořízený v krajní poloze 16,3 cm..... | 26 |
| 2.8 | Snímek pořízený v krajní poloze 32,3 cm..... | 26 |
| 3.1 | Čtení znaků z bufferu | 27 |
| 3.2 | Uživatelské rozhraní..... | 29 |
| 3.3 | Galerie 3D modelů kraslic..... | 30 |
| 3.4 | Vývojový diagram metody <code>startCycle()</code> | 31 |
| 3.5 | Vývojový diagram zpracování snímků..... | 32 |
| 3.6 | Ořezávaná oblast snímků..... | 33 |
| 3.7 | Porovnání 3D modelů a kraslice | 33 |
| 3.8 | Výsledek zpracování snímků..... | 34 |
| 3.9 | Výsledek při použití 50 snímků..... | 35 |
| 3.10 | Výsledek při použití 100 snímků..... | 35 |
| 3.11 | Výsledek při použití 160 snímků..... | 36 |
| 3.12 | Výsledek při použití 200 snímků..... | 36 |
| 3.13 | Snímek s poskládaný z poledníkových výsečí | 37 |
| 3.14 | Snímek s poskládaný z poledníkových výsečí | 37 |
| 3.15 | Textura zpracovaná metodou poledníkových výsečí..... | 38 |
| 3.16 | Zobrazení 3D modelu kraslice | 39 |
| 4.1 | Původní způsob digitalizace exponátů | 41 |
| 4.2 | 3D model kraslice..... | 42 |
| 4.3 | Výsledek digitalizace pomocí přípravku | 43 |

ÚVOD

V současné době spěje více a více oblastí k digitalizaci obsahu a tento trend se nevyhnul ani muzeím a jejich mnohdy rozsáhlým sbírkám. I když digitalizace muzejních sbírek spadá do dlouhodobé strategie státu, k dnešnímu dni je digitalizováno méně než 20% všech předmětů.

Moravské zemské muzeum spravuje početnou sbírku kraslic (přibližně deseti tisíc), které je nutné převést do digitální formy a archivovat, ale pořízení snímků jedné kraslice zabere velké množství času. Motivací této práce je vyrobit přípravek, který by digitalizaci sbírky co nejvíce zjednodušil a zároveň kvalitativně posunul léty zažitý proces prostého nafocení ze dvou stran, což je pro kraslici nevhodné. Nafocení jedné kraslice by pak mělo zabrat menší množství času, mělo by být co nejvíce automatizované a výsledkem by měla být digitalizovaná a reprezentovatelná data v takové formě, která odpovídá dnešním možnostem výpočetní techniky.

Hlavním cílem této práce je tedy vytvořit přípravek a ovládací program, který nasnímá kraslici ze všech stran a data automaticky uloží. Přípravek a program musí být jednoduchý na používání a ovládání, program navíc musí být volně šiřitelný.

V první kapitole jsou teoreticky popsány CCD a CMOS snímače a základní princip plošných a řádkových kamer.

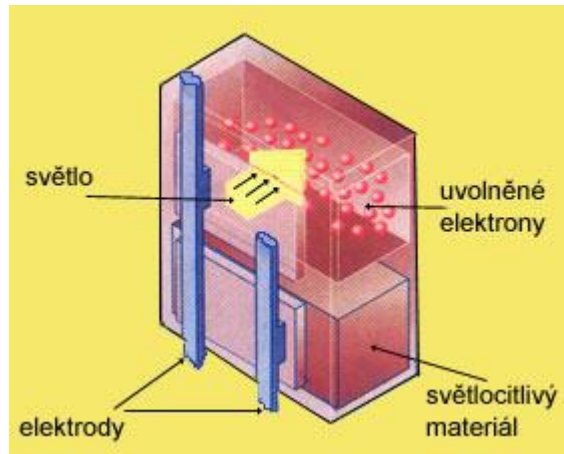
Druhá kapitola se zabývá návrhem hardwarové části přípravku. Je zde uvedeno zapojení desky Arduino, driveru A3967, snímače vzdálenosti VL53L1X a krokového motoru. V neposlední řadě je zde popsáno tělo přípravku, které bylo vytisknuto na 3D tiskárně.

Ve třetí kapitole jsou popsány tři programy. Prvním z nich je desktopová aplikace, pomocí které uživatel ovládá přípravek. Druhým z nich je program, který je nahrán v desce Arduino UNO a který skrze driver A3967 ovládá krokový motor a obsluhuje snímač vzdálenosti. Poslední je program, který slouží pro prezentaci 3D modelů kraslic. V kapitole je rozebrán i způsob komunikace mezi deskou Arduino a desktopovou aplikací a jsou zde uvedeny knihovny, které všechny programy používají.

Poslední kapitola porovnává původní způsob digitalizace, který je nutný dělat celý manuálně a digitalizaci s pomocí přípravku, který je z velké části automatizovaný a umožňuje prezentaci dat ve formě 3D modelu.

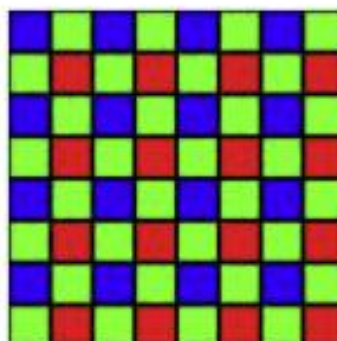
1. PRINCIP PLOŠNÝCH A ŘÁDKOVÝCH KAMER

V dnešní době se používají obrazové snímače na principu CCD a CMOS. Oba principy se liší způsobem, jakým je vyčítána hodnota na výstupu fotodiody. Základem obou snímačů je křemíková fotodioda. Při jejím osvětlení dojde k fotoelektrickému jevu a z krystalové mřížky jsou uvolněny volné elektrony, přičemž dochází ke vzniku náboje. Velikost takto vzniklého náboje je úměrná intenzitě dopadajícího světla na fotodiodu.[1]

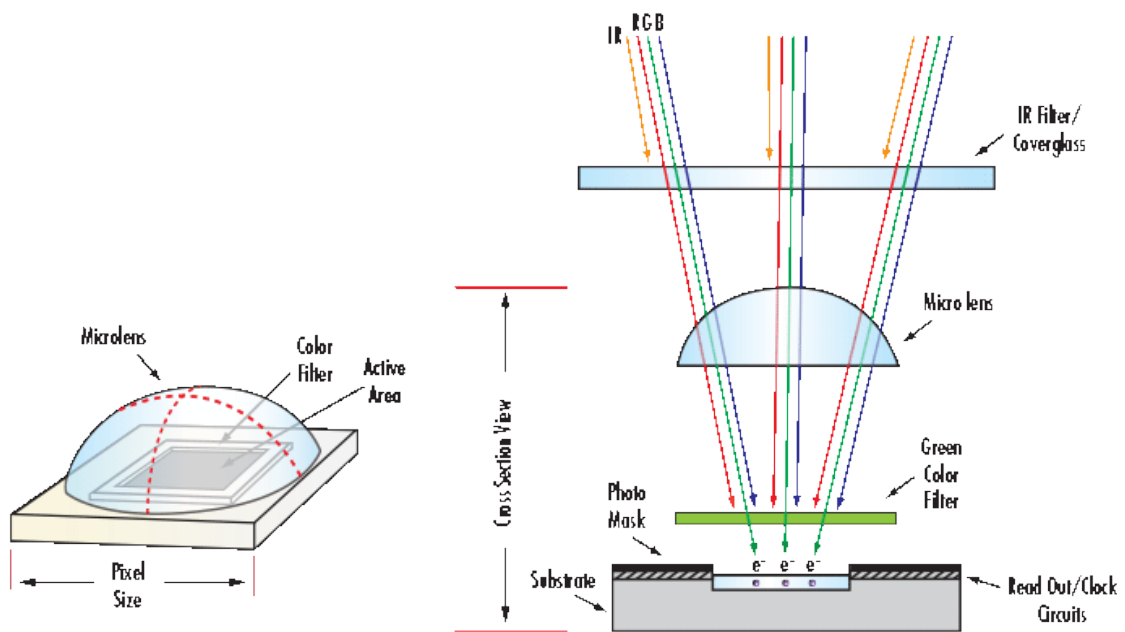


Obrázek 1.1 Fotoelektrický jev [1]

Fotodioda reaguje na intenzitu světla, nerozlišuje ale jeho barvu. Aby bylo možné získat barevný snímek je nutno použít buď 3-čipovou RGB kameru nebo použít čip s barevným kódováním, což je nejčastěji používaný způsob - nad snímacím čipem se nachází soustava mikroskopických čoček a Bayerova maska nebo subtraktivní maska. Dopadající světlo je usměrněno čočkou tak, aby na fotodiodu dopadalo skrze barevný RGB filtr, který propustí pouze záření o určité vlnové délce. Každý pixel pak ukazuje intenzitu dopadající červené, zelené nebo modré barvy.[2]



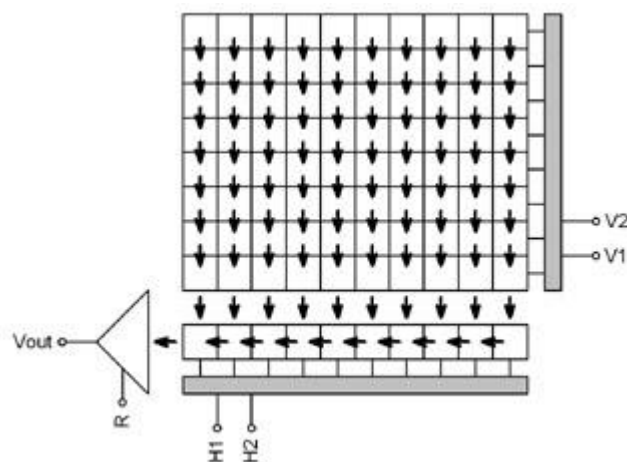
Obrázek 1.2 Bayerova maska [2]



Obrázek 1.3 Průchod světla mikroskopickou čočkou a filtry [3]

1.1 CCD

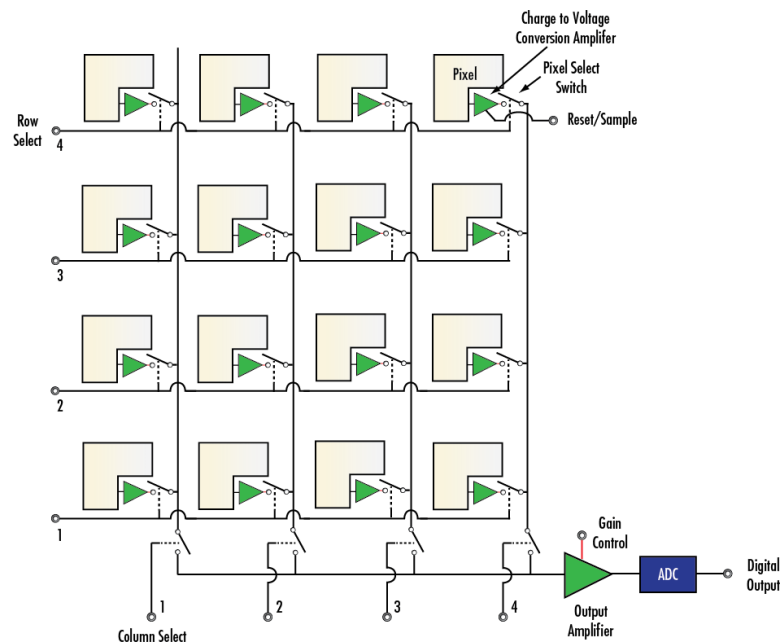
Při vyčítání nábojů z jednotlivých fotodiód pixelů, jsou náboje přesouvány po ploše čipu pomocí napěťové vlny generované budičem čipu. Přesouvání probíhá po řádcích směrem k horizontálnímu registru. Spodní řada nábojů je přesunuta do registru, kde je po jednom odečtena jejich velikost. Hodnota náboje je převedena nábojovým zesilovačem a pomocí AD převodníku převedena do digitální podoby. CCD snímače jsou používány v dražších fotoaparátech a kamerách. [4]



Obrázek 1.4 Odečítání hodnot náboje u CCD obrazového snímače [4]

1.2 CMOS senzor

U CMOS senzorů je elektronika potřebná k digitalizaci přímo součástí CMOS čipu. Hodnoty napětí jednotlivých pixelů jsou vyčítány řádek po řádku a postupně převedeny na digitální hodnotu za použití AD převodníku. CMOS snímače jsou používány především u levnějších kamer a fotoaparátů, i když dnes se kvalitativní parametry obou technologií stírají.[3], [5]



Obrázek 1.5 CMOS snímač [3]

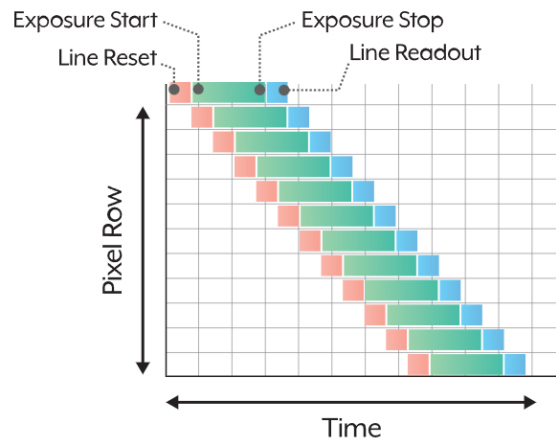
1.3 Elektronická závěrka

U většiny dnešních CMOS snímačů se expozice vůči světlu zapíná a vypíná po řádcích, což se nazývá rolující závěrka (rolling shutter). Každý řádek pixelů je tak světlu vystaven v trochu jinou dobu, což může u rychle se pohybujících objektů vést k deformaci výsledného obrazu. Jsou však CMOS senzory s globální závěrkou (global shutter), která tuto nevýhodu potlačuje.

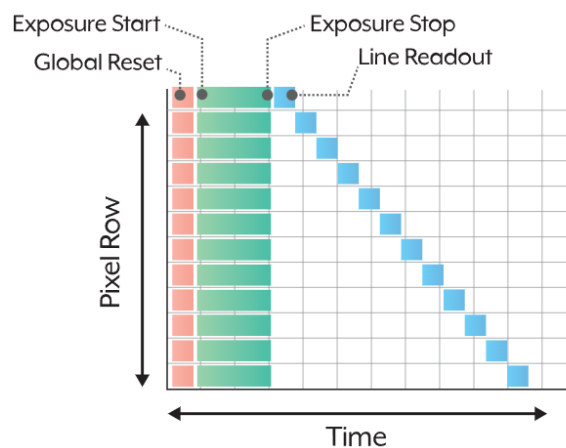
CCD snímače v principu pracují v režimu globální závěrky. Ta vystavuje všechny pixely světlu na stejnou dobu. [6]



Obrázek 1.6 Porovnání globální (vlevo) a rolující (vpravo) závěrky [6]



Obrázek 1.7 Rolující závěrka [5]



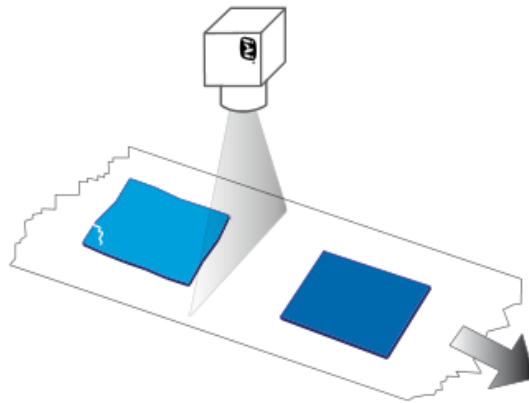
Obrázek 1.8 Globální závěrka [5]

1.4 Plošné kamery

Obecněji více známé plošné kamery mají pixely uspořádané do matice. Příklad takového uspořádání je na obrázku 1.2. Světlo dopadá na jednotlivé pixely s různou intenzitou a po vyčtení intenzity dopadajícího záření z každého pixelu je možné sestavit výsledný obraz.

1.5 Řádkové kamery

Snímač řádkové kamery obsahuje pouze jediný řádek pixelů. Principiálně funguje tak, že se před kamerou pohybuje snímáný předmět (např. dopravní pás s výrobky) a kamera po řádcích pořizuje snímky, ze kterých se pak skládá výsledný obraz. Tímto způsobem je možné snímat rychle se pohybující objekty s vysokou přesností, přičemž je nutné zajistit kontinuální rychlost snímaného objektu. [7]

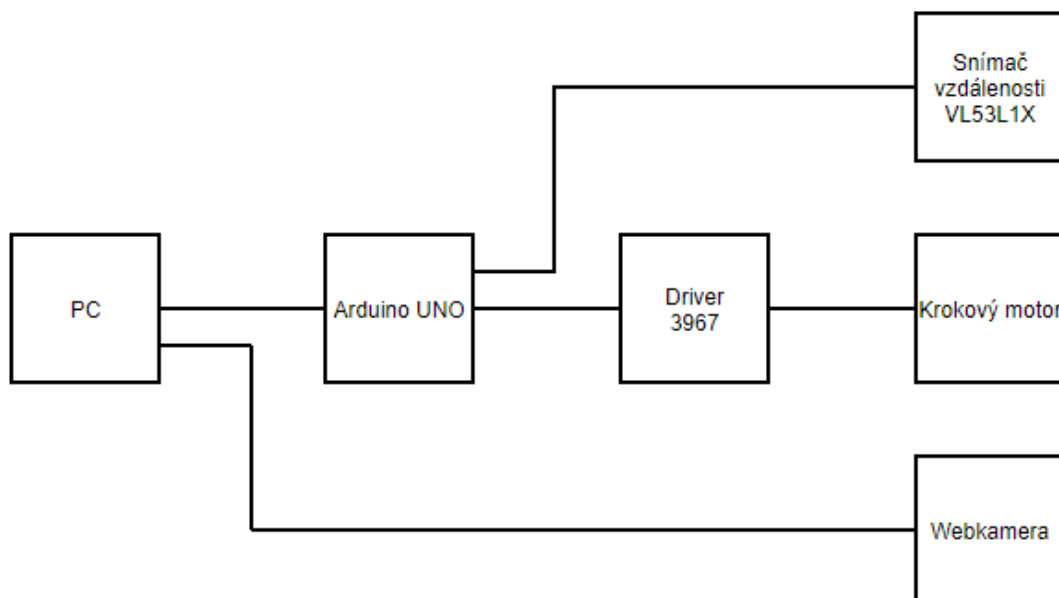


Obrázek 1.9 Princip řádkové kamery [7]

2. KONSTRUKCE PŘÍPRAVKU

Jedním z důležitých požadavků na přípravek je, aby ho mohl uživatel snadno zapojit a obsluhovat.

Blokové schéma celého přípravku je zobrazeno na obrázku 2.1 a skládá se ze tří hlavních částí. Tyto části byly navrženy a vybrány s ohledem na vyhovující parametry, pořizovací cenu i otevřenost řešení. První z nich je počítač, který pomocí desktopové aplikace řídí Arduino a sbírá a zpracovává data z webkamery. Druhou hlavní částí je deska Arduino, která dle příkazů z počítače ovládá pomocí driveru krokový motor a měří výšku vložené kraslice. Poslední hlavní částí je webkamera, která pořizuje snímky kraslice. Všechny komponenty jsou důkladněji popsány v této kapitole. Na konci kapitoly je navíc popsáno i tělo přípravku.



Obrázek 2.1 Blokový diagram přípravku

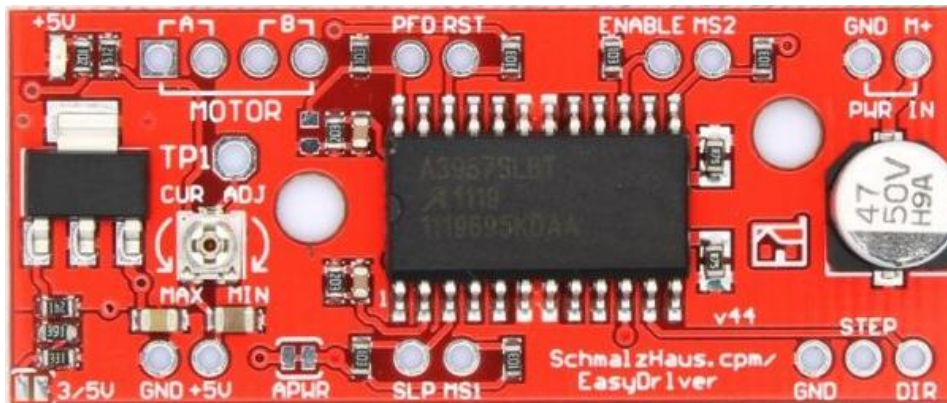
2.1 Krokový motor

Při pořizování snímků je nutné kraslici otočit o předem definovaný úhel a kraslici vyfotit. Celý proces se pak opakuje tak dlouho, dokud není kraslice nafocena ze všech stran. Právě proto byl zvolen pro otáčení kraslice krokový motor. Je pomocí něj možné vejce otáčet s dostatečnou přesností a bez nutnosti zpětné vazby.

Jako krokový motor byl zvolen Microcon Nema 17. Jedná se o menší motor používaný ve 3D tiskárnách. Jelikož motor nebude nijak zvláště zatěžován, tak nejdůležitějšími parametry byly jmenovitý proud a velikost kroku. Jmenovitý proud motoru je 1,5 A a velikost kroku (bez použití mikrokrokování) je 1,8°. Motor je zapojen sériově a je ovládán driverem A3967.

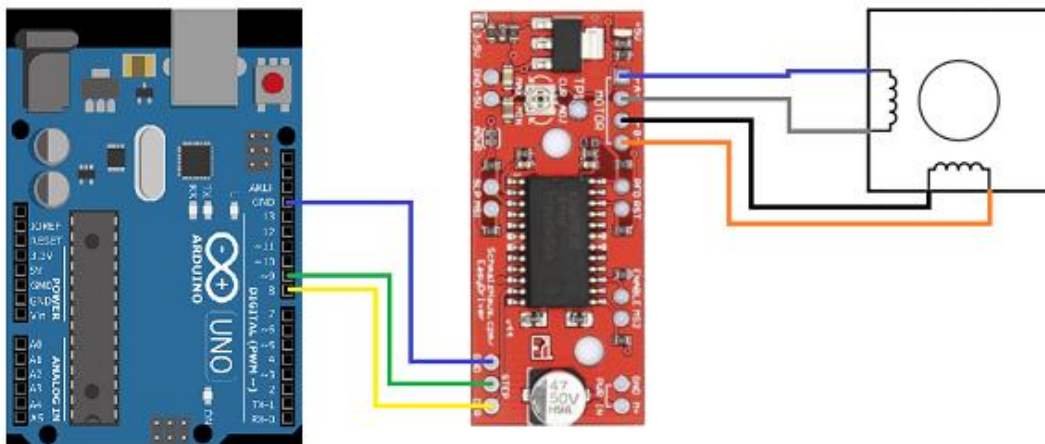
2.2 Driver A3967

Tento driver byl zvolen hlavně kvůli jeho jednoduchosti. Je navržen pro 5V digitální logiku, takže je snadné ho ovládat pomocí desky Arduino UNO. Piny desky Arduino jsou připojeny na vstupy *GND*, *STEP* a *DIR*. Driver reaguje na náběžné hrany na vstupu *STEP* a při každé takové změně dojde k otočení motoru o jeden krok. Velikost kroku pak závisí na stavu napětí na vstupech *MS1* a *MS2*, ve výchozím stavu je však driver nastaven na 1/8 kroku. Pomocí stavu na vstupu *DIR* se pak nastavuje směr otáčení.



Obrázek 2.2 Driver A3967 [8]

Cívky motoru jsou připojeny k pinům *MOTOR A* a *B*. Maximální proud na výstupu je 750 mA, je ale možné proud omezit pomocí potenciometru *CUR ADJ*. Driver je napájen 12V adaptérem. Aby se předešlo zničení, na driveru je nainstalován pasivní chladič a teplovodivá pasta.



Obrázek 2.3 Zapojení krokového motoru a driveru A3967

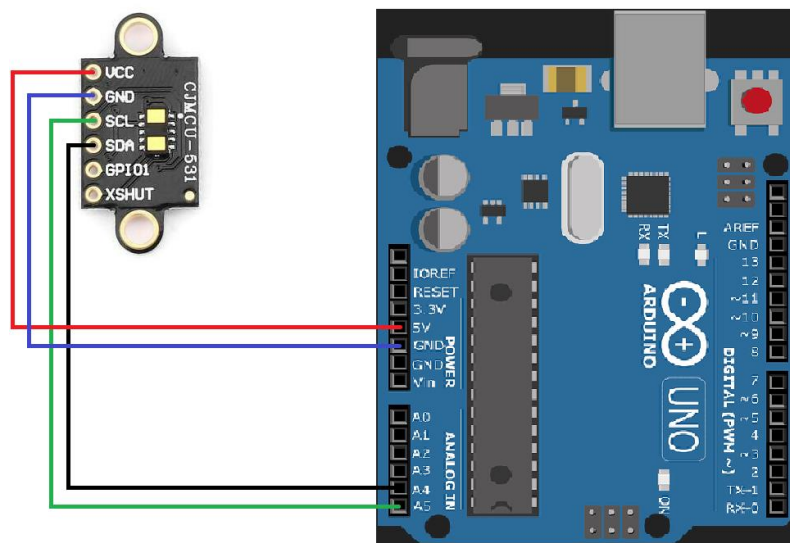
2.3 Webkamera

Jako webkamera byla zvolena Genius QCam 6000, při jejímž výběru bylo důležitých několik parametrů. Prvním a nejdůležitějším parametrem je rozlišení, které by mělo být alespoň Full HD. Druhým důležitým parametrem byl způsob, jakým je možné kameru připevnit k přípravku.

2.4 Snímač vzdálenosti VL53L1X

Aby bylo možné zpracovat snímky kraslic různých velikostí, je nutné znát jejich výšku. Pro měření výšky kraslic je využit laserový senzor vzdálenosti VL53L1X, který měří vzdálenost metodou ToF (Time-of-Flight – doba letu). Při výběru snímače bylo omezujícím parametrem jeho umístění na přípravku. Snímač je připevněn na vrcholu rotačního členu a měří vzdálenost uvnitř válce, což vylučuje například ultrazvukové snímače. Maximální rozsah snímače (v závislosti na okolním osvětlení a nastavení) je 400 cm a rozlišení 1 mm, což vyhovuje požadavkům.

Pin SCL, který je na desce Arduino připojen na pin A5, slouží k přenášení hodinového signálu a synchronizaci přenosu dat. Pin SDA je připojen k pinu A4 a slouží k přenosu dat mezi Arduinem a snímačem vzdálenosti. [9] Nastavení, komunikaci a samotné měření vzdálenosti zajišťuje knihovna *SparkFun_VL53L1X.h*.



Obrázek 2.4 Zapojení snímače vzdálenosti VL53L1X

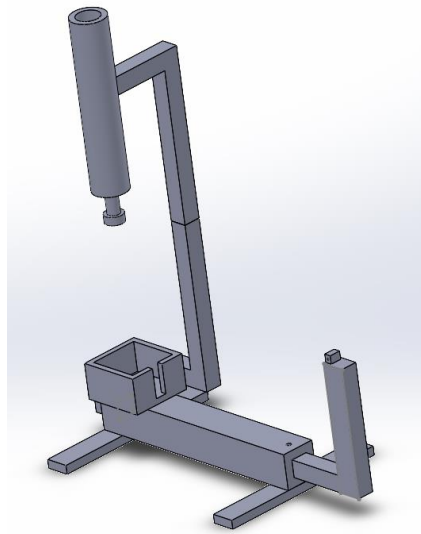
Při připojení Arduina k počítači dojde k automatickému změření výšky kraslice a tato hodnota se nastaví jako offset. Při startu cyklu pro pořízení snímků vždy dojde k opětovnému změření vzdálenosti. Vzdálenost je následně použita při zpracování snímků.

2.5 Tělo přípravku

Celé tělo přípravku je vytištěno na 3D tiskárnách Prusa i3 MK3 a Creality Ender 3. Použitými materiály bylo PETG a PLA. Při modelování jednotlivých částí byl nejvíce omezujícím faktorem velikost tiskové plochy 3D tiskárny. Kvůli tomuto omezení je přípravek složen z několika jednotlivých dílů, které jsou k sobě přilepeny, nebo mají drážky a dají se do sebe zasunout.

Krokový motor je umístěn v krabici na levé straně přípravku. Na hřídeli krokového motoru je nasazen držák, na který je kraslice posazena. Stejný držák je i v protilehlé části přípravku a přidržuje horní část kraslice. Na druhé straně přípravku, na posuvném dílu ve tvaru L, je uchycena webkamera. K přípravku je upevněna dvěma šroubky a uchycení je navrženo tak, aby kameru nebylo možné naklánět. Pod webkamerou je umístěn LED pásek, který osvětluje snímaný exponát.

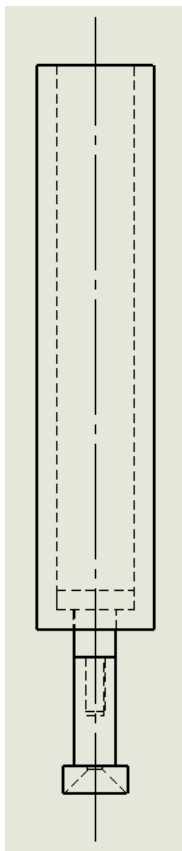
Do přípravku je možné upevnit kraslice o výšce od 4,1 cm do 7,5 cm, přičemž největší a nejmenší kraslice, které byly zapůjčeny Moravským zemským muzeem, byly vysoké 6,1 cm a 5,1 cm.



Obrázek 2.5 Tělo přípravku

Část přípravku, která drží kraslici, se skládá z několika dílů. Prvním z nich je válec upevněný na rameni v horní části přípravku. Na vrchní části válce je přilepen menší váleček, který drží snímač vzdálenosti. Ve válci je umístěný rotační element, který je pevně spojen s držákem kraslice, a který přidržuje vrchní část kraslice. Na vrcholu rotačního členu je zároveň nalepen bílá páska, která zlepšuje odraz laserového paprsku zpět ke snímači vzdálenosti. Celá soustava zajišťuje, že se držák může volně otáčet a zároveň je možné ho snadno pozvednout.

Mezi kraslicí a držákem při snímání občas došlo k prokluzu a kraslice se neposunula o celý počet kroků. Kraslice se zároveň během snímání mírně nakláněla, což vedlo k neuspokojivému výsledku. Oba držáky kraslice jsou proto pokryty gumou.



Obrázek 2.6 Sestava zajišťující rotaci

2.6 Vzdálenost webkamery od kraslice

Část přípravku držící webkameru je posuvná, takže je možné nastavit vzdálenost kamery od kraslice v rozmezí od 16,3 cm až po 32,3 cm. Ideální vzdálenost je však mezi 20 cm a 25 cm, takže je rameno v této vzdálenosti zafixováno šroubem. V této vzdálenosti je kraslice na snímku nejvíce ostrá a zároveň dostatečně velká.

Snímky pořízené z krajních poloh jsou na obrázku 2.7 a na obrázku 2.8. Jak je vidět na druhém z obrázků, snímek pořízený ze vzdálené krajní polohy je sice ostrý, ale nejsou dobře viditelné detaily. Naopak snímek pořízený z krajní polohy blízké exponátu je málo ostrý.



Obrázek 2.7 Snímek pořízený v krajní poloze 16,3 cm



Obrázek 2.8 Snímek pořízený v krajní poloze 32,3 cm

2.7 Osvětlení kraslice

Osvětlení kraslice je zajištěno LED páskem, který je připevněn pod webkamerou. Aby bylo zajištěno co nejvíce rovnoměrné osvětlení povrchu kraslice, LED pásek je pod difuzorem, který světlo rovnoměrně rozptýlí. Napájení je zajištěno ze stejného adaptéru, který napájí driver A3967. Barva LED pásku je denní bílá a příkon je 9,6W/m.

3. SOFTWAREVÁ ČÁST PŘÍPRAVKU

Softwarová část přípravku se skládá ze tří programů. Prvním z nich je program v C++, který je nahrán v desce Arduino UNO. Druhou částí je desktopová aplikace v jazyce C#, která běží na počítači a která ovládá přípravek. Poslední je Java program, který zobrazuje výsledný model kraslice.

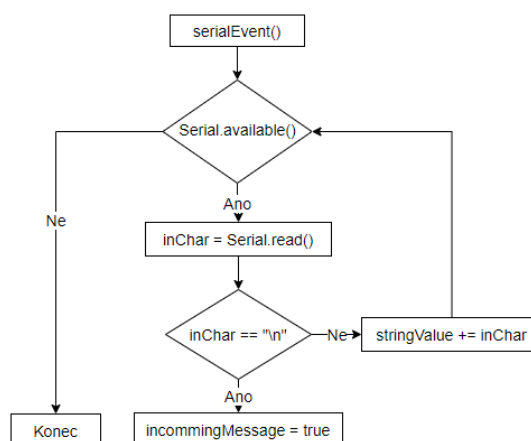
Jazyk C# byl vybrán hlavně ze dvou důvodů. Prvním z nich je fakt, že výsledný program musí být volně šiřitelný. Druhým důvodem je že díky .NET frameworku je možné pohodlně vytvořit desktopovou aplikaci s uživatelským rozhraním.

3.1 Komunikace mezi počítačem a deskou Arduino

Komunikace mezi počítačem a deskou Arduino je uskutečněna pomocí sériového přenosu (UART). Na straně Arduina byla k tomuto účelu použita základní knihovna `Serial`. Na straně desktopové aplikace byla ke komunikaci použita knihovna `System.IO.Ports`, která je také jednou ze základních knihoven C#. Metody, které slouží k ovládní Arduina jsou umístěny ve třídě `ArduinoCommunication`.

Než začne komunikace mezi deskou Arduino a programem, jsou na obou stranách nastaveny parametry komunikace. Po stisku tlačítka *Připojit* dojde k otevření připojení a program posílá první zprávu k ověření funkčnosti komunikace. Pokud desktopová aplikace od Arduina obdrží zpětnou vazbu, komunikace byla na obou stranách úspěšně otevřena. V případě, že aplikace neobdrží zprávu o úspěšném připojení do 5 vteřin, je čekání na zprávu přerušeno a je zobrazena chybová hláška.

Když deska Arduino obdrží zprávu, je vyvolán event `serialEvent()`. Následně funkce `Serial.read()` vrací jednotlivé znaky z bufferu a ukládá je do řetězce znaků. Pokud narazí na znak nového řádku, tento znak se již do řetězce nepřidává, zpráva je kompletní a je čtení ukončeno.



Obrázek 3.1 Čtení znaků z bufferu

Způsob čtení z bufferu na straně počítače je o něco snadnější, jelikož je možné využít metodu `System.IO.Ports.ReadLine()`, která vrátí celý jeden řádek z bufferu.

Při komunikaci jsou používány tři zprávy, a to příkazy START, CLK00 a MEASR. Příkaz START je použit vždy po vzájemném připojení a slouží spíše jako demonstrace toho, že komunikace mezi počítačem a deskou Arduino započala v pořádku. Příkaz CLK00 slouží pro otočení ve směru hodinových ručiček. Jelikož není potřeba kraslicí otáčet na obě strany, je příkaz CLK00 dostačující. Příkaz MEASR slouží jako požadavek pro změření výšky vložené kraslice.

Formát zprávy je ve tvaru XXXXX/YYY. První částí je jeden z výše zmíněných příkazů. Ve druhé části je uveden parametr příkazu. Aby bylo parsování co nejjednodušší, obě části mají pevně danou délku. Pokud tedy chceme například motor pootočit o 90°, zpráva bude ve tvaru CLK00/400. Jako zpětnou vazbu pak Arduino odešle zprávu XXXXX/YYY/DONE (v tomto případě CLK00/400/DONE).

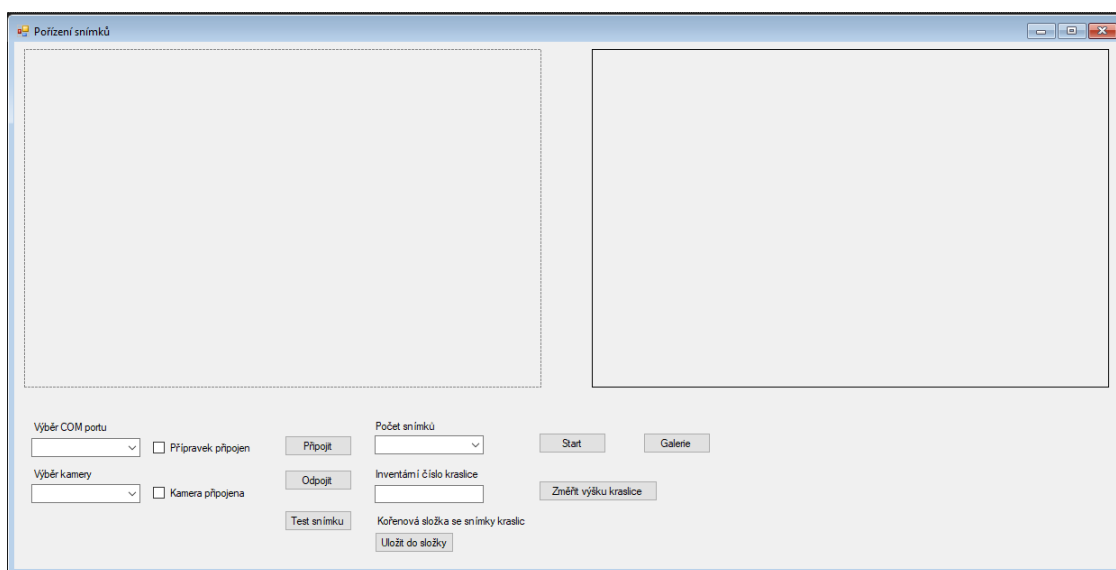
Trochu jinak je tomu u příkazu MEASR. Jelikož pro měření není potřeba žádný další parametr, ve druhé části příkazu může být uvedeno cokoli (musí ale být dodržena délka tří znaků). Arduino pak vrací zprávu v již definovaném formátu s tím rozdílem, že ve druhé části zprávy je uvedena naměřená vzdálenost v milimetrech.

3.2 Uživatelské rozhraní

Uživatelské rozhraní je rozděleno do dvou hlavních částí. Ve vrchní části jsou dvě okna, z nichž levé slouží pro zobrazení pořízených snímků v průběhu digitalizace kraslice a pravé slouží k promítání obrazu kamery. Po nasnímání kraslice a zpracování snímků je výsledek zobrazen v levém okně.

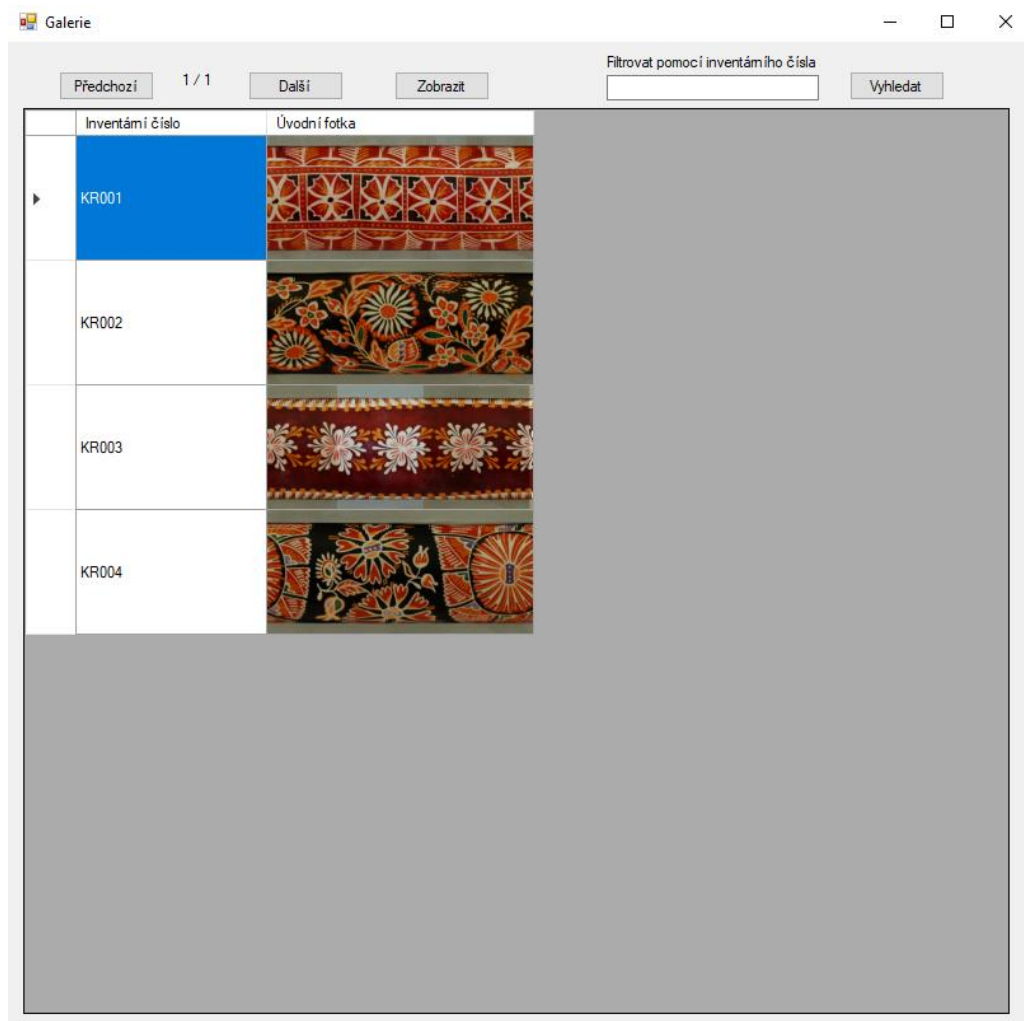
Ve spodní části jsou pak nástroje k ovládání přípravku a ke správě pořízených snímků. Všechny ovládací nástroje budou detailněji popsány dále v průběhu kapitoly.

Aby byl program co nejjednodušší na používání, většina tlačítek a rolovacích lišt je po spuštění neaktivních. Aktivují se postupně s tím, jak uživatel nastavuje jednotlivé parametry. Uživatel pak nemůže např. spustit snímání, aniž by zadal inventární číslo kraslice.



Obrázek 3.2 Uživatelské rozhraní

Kliknutím na tlačítko Galerie dojde k otevření nového okna, ve kterém je možné procházet již digitalizované kraslice. Okno je opět rozděleno na dvě části, ve vrchní části jsou tlačítka pro procházení mezi stránkami a pole pro filtrování kraslic podle inventárního čísla. Označením řádku a kliknutím na tlačítko Zobrazit dojde k otevření modelu kraslice. Ve spodní části jsou v řádcích zobrazeny již digitalizované kraslice. V řádku je uvedeno pouze inventární číslo a zpracovaný snímek kraslice. Digitalizované kraslice jsou zobrazeny po stránkách, přičemž na každé stránce je zobrazeno 50 kraslic. Tímto způsobem jsou zobrazeny kvůli omezené paměti a době načítání. Program totiž udržuje jednotlivé řádky načtené v operační paměti po celou dobu, kdy jsou na stránce zobrazeny. Jelikož má být aplikace použita pro procházení až tisíců záznamů, bylo by udržení všech snímků v operační paměti nemožné. Se stoupajícím počtem záznamů navíc stoupá i doba potřebná k jejich načtení.



Obrázek 3.3 Galerie 3D modelů kraslic

3.3 Obsluha webkamery

K obsluze kamery je použit framework AForge.NET, jehož součástí jsou mimo jiné i knihovny pro snadnější obsluhu webkamer. Výhodou frameworku je i to, že je k němu několik vzorových projektů pro různá použití.

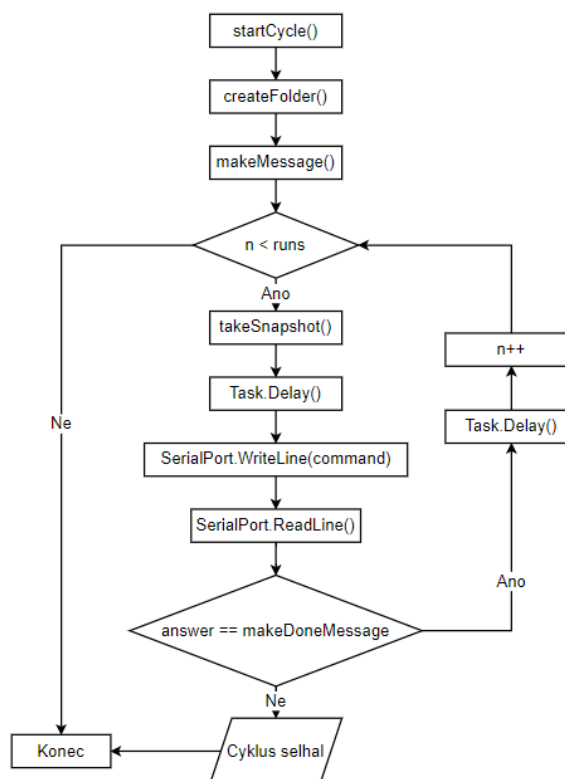
Před započítím komunikace je nejdříve nutné vybrat kameru, se kterou má program navázat spojení. Ihned po zapnutí programu je tedy inicializována třída `FilterInfoCollection(FilterCategory)`, která podle zvoleného parametru kategorie najde a uloží všechny připojené periferie, které odpovídají filtru. Všechny takto nalezené periferie jsou vypsané v rolovací liště s názvem Výběr kamery. Po tom, co uživatel vybere vhodnou kameru a stiskne tlačítko Připojit, dojde k nastavení rozlišení zvolené webkamery na velikost 1920×1080 a spustí se promítání v pravém horním okně uživatelského rozhraní.

Při vytváření snímku je zavolána metoda `takeSnapshot()`, která vytvoří kopii posledního snímku přijatého z webkamery. Tento snímek je zobrazen v levém horním okně a pokud byl vytvořen během cyklu snímání kraslice, je uložen do příslušné složky.

3.4 Průběh cyklu snímání kraslice

Po nastavení všech parametrů je umožněno uživateli spustit cyklus snímání kraslice. Po stisknutí tlačítka je zavolána metoda `startCycle()`. Nejdříve dojde ke změření výšky kraslice. Poté, pokud zatím neexistuje složka se zadaným inventárním číslem, je vytvořena složka pomocí metody `createFolder()`. Následně je pomocí metody `makeMessage()` vytvořen příkaz k ovládání motoru. Ve `for()` cyklu je pak nejdříve pořízen snímek a Arduino je metodou `SerialPort.WriteLine(command)` zaslán příkaz k pootočení krokového motoru. Přijatá zpětná vazba je porovnána s návratovou hodnotou metody `makeDoneMessage()`. `for()` cyklus se opakuje tak dlouho, dokud není pořízen požadovaný počet snímků.

Metoda `Task.Delay()` je použita proto, že během cyklu někdy docházelo k zaseknutí obrazu kamery. To způsobovala metoda `SerialPort.ReadLine()`, jelikož vždy čeká, než z bufferu načte celý řádek. Během této doby však program ignoruje přicházející eventy s novými snímky webkamery.



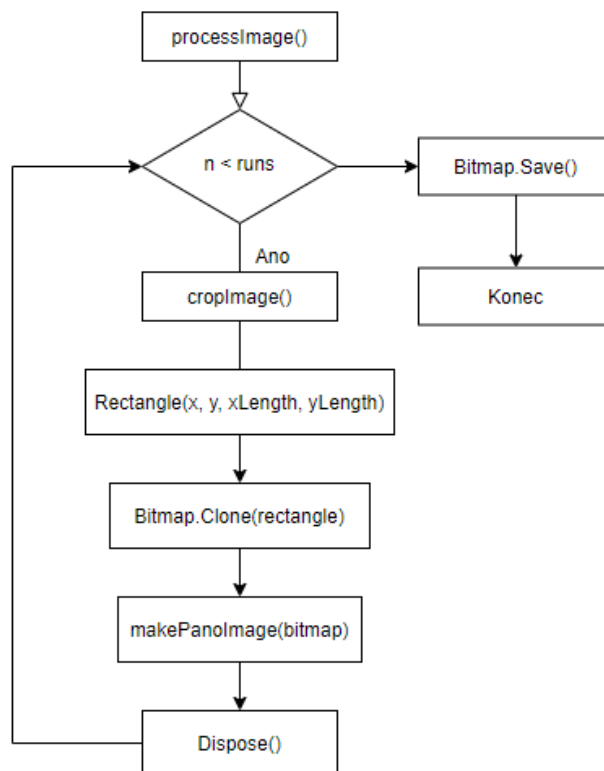
Obrázek 3.4 Vývojový diagram metody `startCycle()`

3.5 Ukládání snímků

Před započítím snímání kraslice je nutné zapsat do textového pole inventární číslo kraslice. Podle zadaného inventárního čísla bude pojmenována složka, ve které se budou pořízené snímky kraslice uchovávat. Podobně budou pojmenovány i jednotlivé snímky s tím rozdílem, že na konci jména je i pořadové číslo snímku (např. *KR001_1*). V případě, že složka se stejným názvem již existuje, budou nové snímky ukládány do již existující složky a dojde k nahrazení starých snímků novými.

3.6 Zpracování snímků metodou skládání obdélníků

V desktopové aplikaci je pro zpracování snímků implementována metoda skládáním obdélníků. Po pořízení a uložení poslední fotky, následuje proces zpracování snímků. Pro zpracování snímků a vytvoření panoramatické fotografie slouží třída `ProcessImage`. Ze složky je nahrán snímek a v závislosti na výšce kraslice je oříznut. Z každého snímku je pomocí metody `Bitmap.Clone()` vyříznut obdélník o zadaném rozměru a oříznutá část snímku je následně vložena do bitmapy. Vyříznuté obdélníky jsou skládány vedle sebe bez dalších úprav. Následně je zavolána metoda `Dispose()`, která uvolní paměť naposledy načtené bitmapy. Stejným způsobem jsou postupně zpracovány všechny snímky a výsledek je uložen do složky s původními snímky.



Obrázek 3.5 Vývojový diagram zpracování snímků

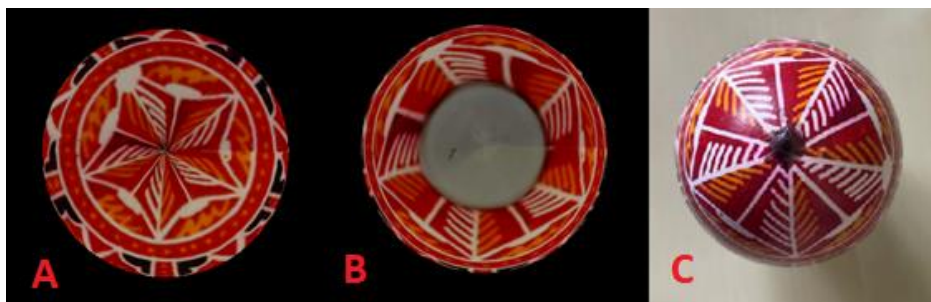
Způsob zpracování skládáním obdélníků byl zvolen nejdříve pouze pro testování kvůli jeho jednoduchosti, jelikož stačí z každého snímku vybrat oblast ve tvaru obdélníku a tu vložit do nové bitmapy. Od určitého počtu snímků je výsledek dostatečně kvalitní a přechody mezi obdélníky jsou i v blízkosti pólů málo patrné.



Obrázek 3.6 Ořezávaná oblast snímků

Ze snímku je záměrně vyříznuta i oblast kousek nad a pod kraslicí. Důvodem je to, že při použití metody `Ellipsoid.texture(this, „path to image“)` (metoda definuje texturu renderovaného 3D modelu) je snímek roztažen až do pólů. Problém je ale v tom, že póly kraslice nejsou na snímku viditelné a snímek je tedy roztažen a model tak neodpovídá reálné kraslici.

Na obrázku 3.7 je porovnání 3D modelů s různě zpracovanými texturami a reálné kraslice. Textura A je ořezávána na rozmezí kraslice a držáku kraslice. Z porovnání s kraslicí (C) je patrné, že snímek je roztažen a nereprezentuje skutečný vzhled kraslice. Textura B byla oříznuta podle obrázku 3.6 a při porovnání se skutečnou kraslicí je patrné, že skutečný exponát reprezentuje mnohem přesněji. Šedá oblast je během pořizování snímků skryta pod držákem kraslice.



Obrázek 3.7 Porovnání 3D modelů a kraslice



Obrázek 3.8 Výsledek zpracování snímků

3.7 Výsledek zpracování snímků metodou skládání obdélníků

Nevýhodou zvolené metody skládání výsledné textury je bezesporu to, že aby výsledek vypadal uspokojivě, musí být pro jeho vytvoření použito velké množství snímků. Jelikož je snímaným objektem vejce, počet potřebných snímků pro uspokojivý výsledek dosahuje dvou set snímků.

Na následujících obrázcích 3.9-3.12 jsou ukázky textur složených z 50, 100, 160 a 200 snímků. Z ukázek je patrné, že vzory ve střední části kraslice na sebe dobře navazují již při 50 snímcích, naopak blízko pólů jsou dobře viditelné přechody mezi jednotlivými obdélníky a vzory tak nedávají skoro žádný smysl. Při 100 snímcích už se tato situace zlepšuje, ale v blízkosti pólů jsou vzory stále silně nekonzistentní. Při 160 snímcích už jsou obdélníky dostatečně úzké a u pólů jsou dostatečně konzistentní. Nejvíce uspokojivých výsledků je logicky dosaženo při použití největšího množství snímků. Při použití 200 snímků jsou už i u pólů hrany obdélníků málo patrné.



Obrázek 3.9 Výsledek při použití 50 snímků



Obrázek 3.10 Výsledek při použití 100 snímků



Obrázek 3.11 Výsledek při použití 160 snímků



Obrázek 3.12 Výsledek při použití 200 snímků

Pokud ale porovnáme texturu složenou ze 160 snímků a 200 snímků na 3D modelu, rozdíly jsou minimální. Pokud bychom jako jeden z parametrů brali i čas nutný pro pořízení snímků, nejlepší možností by byla textura složená ze 160 snímků. Pořízení 160 snímků zabere téměř 45 vteřin méně, což je při digitalizaci několika tisíc kraslic velký rozdíl.

Uživatel si v desktopové aplikaci může sám zvolit, kolik snímků chce pro vytvoření textury použít v závislosti na požadované kvalitě a dostupném čase.

3.8 Další možnosti zpracování snímku

Zpracování snímků skládáním obdélníků je funkční a její výsledky jsou pro účely bakalářské práce dostačující. Jejím problémem je ale již zmíněná náročnost na vysoký počet vstupních snímků. Tento nedostatek by bylo možné vyřešit, pokud by byly jednotlivé obdélníky ještě před vložením do finálního snímku dále zpracovány.

V každém obdélníku je ponechána pouze poledníková výseč, u zbylých pixelů je změněna barva pixelu na černou. Šířka oblasti, která zůstane beze změny se směrem k poledníkům zmenšuje.



Obrázek 3.13 Snímek s poskládaný z poledníkových výsečí

Následuje opětovné přiřazení RGB hodnoty začerněným pixelům. Všem začerněným pixelům v jednom řádku je přiřazena RGB hodnota nejbližšího nezměněného pixelu v řádku.



Obrázek 3.14 Snímek s poskládaný z poledníkových výsečí

Výsledek na obrázku 3.14 je složen ze 45 snímků. Při porovnání výsledku zpracování snímků metodou poledníkových výsečí (obrázek 3.14) a skládáním obdélníků (obrázek 3.12) je vidět, že metoda skládání poledníkových výsečí vyžaduje přibližně čtvrtinový počet snímků pro dostatečně uspokojivý výsledek. Výhodou této metody tedy je, že nevyžaduje velké množství vstupních snímků a umožňuje tak větší úsporu času. Naopak nevýhodou je o něco složitější implementace.



Obrázek 3.15 Textura zpracovaná metodou poledníkových výsečí

3.9 Tvar kraslice

Popsat tvar vejce není snadný úkol a stejně složité by bylo takovýto objekt zobrazit. Proto je při zobrazení 3D modelu použit elipsoid, který je tvarově ze všech běžně používaných prostorových objektů vejci nejbližší. V prostředí Processing je pro zobrazení a práci s elipsoidem využita knihovna `Shapes3D`.

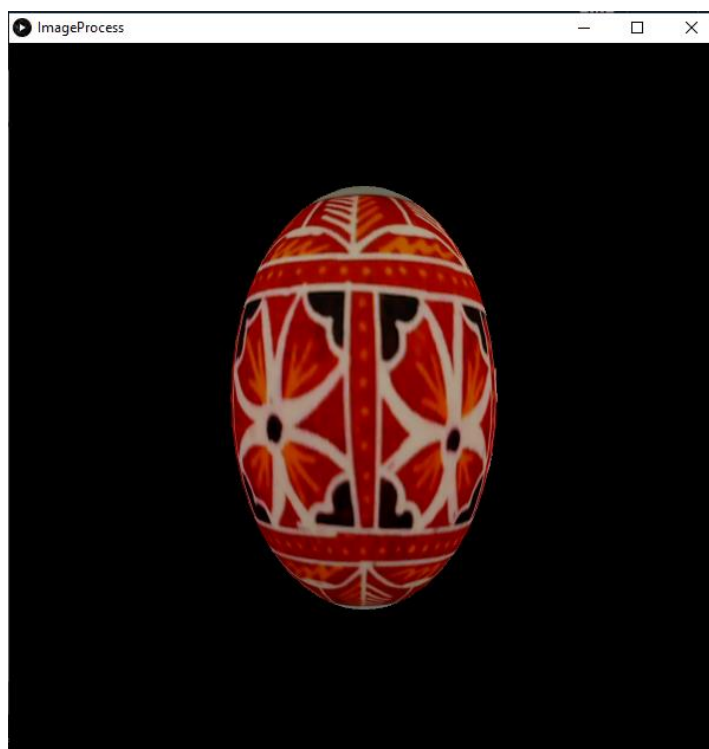
3.10 Zobrazení výsledného modelu

Vizualizace výsledného 3D modelu je vytvořena ve vývojovém prostředí Processing za použití grafické knihovny Processing [10]. Vývojové prostředí umožňuje vytvářet aplikace v jazycích Java (Processing), Python (Processing.py) a JavaScript (p5.js) pro webové aplikace. Pro řešení bakalářské práce byla zvolena po krátkém testování varianta jazyku Java. Processing má vlastní jednoduché vývojové prostředí, kde je možné vyvíjet a testovat vlastní aplikace. Pro vytvoření programu bylo využito IDE Processing 3.5.4.

Jednoduché tvary a prostorové objekty je možné v prostředí Processing vytvořit a zobrazit velmi efektivně. Základní knihovna ale neobsahuje žádnou metodu pro vytvoření elipsoidu, proto je ve skriptu použita knihovna `Shapes3D`, která obsahuje metody pro zobrazení dalších, méně obvyklých prostorových objektů.

Při spuštění skriptu je jako první zavolána metoda `settings()`, ve které se kontroluje, jestli parametr `args` obsahuje cestu k textuře jako vstupní argument. Pokud obsahuje argument, je uložen do proměnné a je volána metoda `size(width, height, P3D)`, která nastaví velikost plátna. Poslední parametr metody definuje, že vše v programu bude renderováno jako 3D objekty. Dalším krokem je vytvoření instance elipsoidu konstruktorem `new Ellipsoid(radX, radY, radZ, nbrSegs, nbrSlices)`. První tři parametry udávají rozměr elipsoidu a poslední dva argumenty definují počet segmentů v osách XZ a v ose Y. Následně je volána metoda `Ellipsoid.texture(this, „path to image“)`, která definuje texturu renderovaného objektu. Posledním krokem je zobrazení objektu použitím metody `Ellipsoid.draw()`. [11], [12] Součástí skriptu je i metoda `mouseDragged()`. Pokud uživatel drží levé tlačítko myši a pohybuje kurzorem v ose x , kraslice se otáčí kolem osy y .

Tento způsob prezentace dat uživateli umožňuje 3D pohled na kraslici, otáčení kraslice a zobrazení více kraslic ve více oknech.



Obrázek 3.16 Zobrazení 3D modelu kraslice

3.11 Spouštění scriptu s modelem kraslice

Napsaný script je možné spustit dvěma způsoby. Prvním je spuštění pomocí vývojového prostředí Processing. Druhým způsobem je spouštění scriptu pomocí příkazového řádku. Tento způsob umožňuje spouštění z jiné aplikace a pomocí příkazu je navíc možné i předávat parametry (v tomto případě cesta ke snímku, který má být použit jako textura). Tento způsob umožňuje spustit script vícekrát, pokaždé pro jinou kraslici.

Spuštění scriptu v příkazovém řádku je možné pomocí následujícího příkazu skládajícího se za čtyř částí:

```
processing-java --sketch=„cesta ke složce se skriptem“ --run „parametr“
```

Příkaz `processing-java` spouští program, který zajišťuje zpracování scriptu. Druhý příkaz `--sketch` specifikuje cestu ke scriptu, který má být spuštěn a příkaz `--run` script zkompiluje a spustí. Vše, co je napsáno za příkazem `--run` jsou parametry, které jsou použity ve scriptu. V tomto případě je jako jediný parametr předávána cesta ke snímku, který bude použit jako textura.

Framework .NET podporuje spouštění jiných aplikací a programů s použitím příkazové řádky. Nejdříve je nutné definovat hodnoty, které budou použity při spuštění procesu. Nejdůležitějšími jsou `ProcessStartInfo.FileName`, která je uchovává cestu k `processing-java.exe` aplikaci. Další je `ProcessStartInfo.Arguments`, která uchovává zbytek příkazů (`--sketch`, `--run` a `parametr`). Dále je nutné nastavit `ProcessStartInfo.CreateNoWindow` na hodnotu `true`, aby se aplikace při spuštění nespouštěla vždy v novém okně. Po nastavení všech hodnot je možné spustit aplikaci spustit metodou `Process.Start(process)`.

3.12 Požadavky na vybavenost uživatele

Vytvořená desktopová aplikace v jazyce C# už v sobě obsahuje všechny potřebné knihovny, se kterými pracuje. Pro její fungování tak není potřeba instalovat žádný software. Samotná aplikace je ve formátu `.exe`, takže ani ji není potřeba instalovat. Pro správnou funkci programu zajišťujícího zobrazení 3D modelů kraslic je nutné nainstalovat vývojové prostředí Processing. Návod na jeho instalaci je součástí přílohy.

V případě přípravku je nutné, aby měl uživatel k dispozici dva volné USB sloty na svém počítači 230V zásuvku.

4. POROVNÁNÍ ZPŮSOBŮ DIGITALIZACE SBÍRKY

4.1 Původní způsob digitalizace

Původně byl pro digitalizaci sbírky použit fotoaparát a exponát byl manuálně otáčen. Výhodou tohoto postupu je, že snímky byly pořízeny kvalitním fotoaparátem, takže jsou ostré a jsou viditelné i malé detaily. Navíc je možné pořídit snímky ze všech stran, takže i vrchní a spodní části kraslice. Největší nevýhodou je, že digitalizace jednoho exponátu zabere velké množství času. Kvůli nasvícení u některých kraslic dochází k tomu, že se přední část exponátu leskne. To je vidět zejména na obrázku 4.1.

Při digitalizaci sbírky je navíc nutné, aby na fotce byla kromě exponátu umístěna i barevná kalibrační tabulka s pravítkem a evidenční číslo exponátu. Kvůli tomu, že jsou snímky pořizovány pod úhlem, není možné přesně určit velikost kraslice.

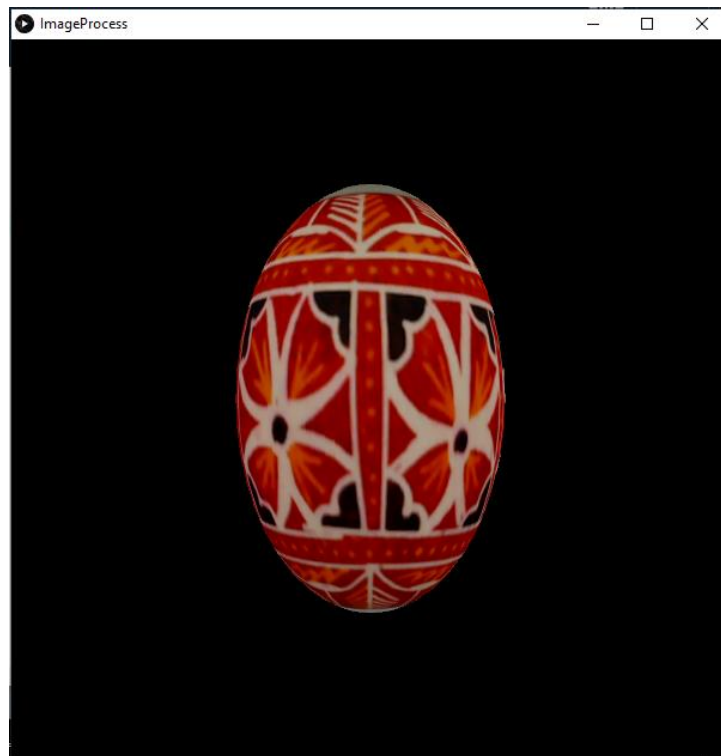


Obrázek 4.1 Původní způsob digitalizace exponátů

4.2 Digitalizace pomocí přípravku

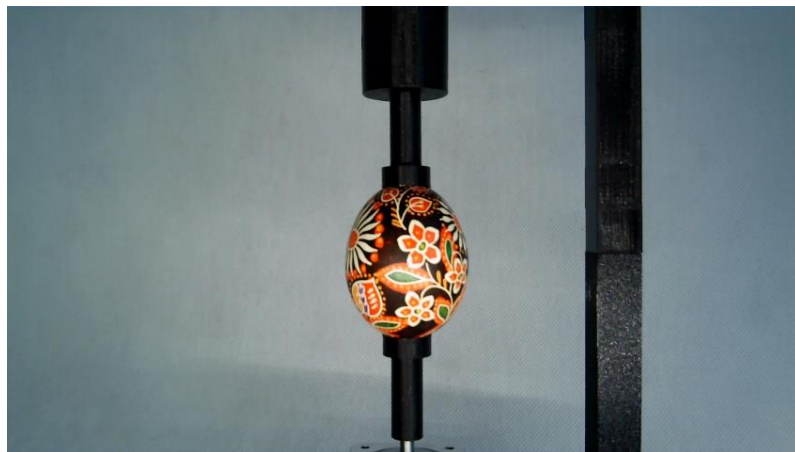
Přípravek společně s desktopovou aplikací uživateli umožňuje jednoduše a pohodlně digitalizovat kraslice. Největší výhodou je, že pořízení snímků jedné kraslice je jednoduché, a i když může zabrat pár minut (pořízení 50 snímků trvá přibližně minutu), uživateli stačí vložit kraslici do přípravku a spustit snímání. Celý proces, včetně zpracování snímků je proveden automaticky. Není nutné ani přehrávat snímky z paměťové karty do počítače, protože jsou snímky automaticky rozděleny podle inventárního čísla do složek v kořenovém adresáři.

Další nespornou výhodou je to, že ze snímků je poskládána výsledná fotografie, která může být použita jako textura při zobrazení 3D modelu. Všechny digitalizované kraslice, u kterých byla textura vytvořena, je možné procházet v galerii. Je možné je filtrovat podle inventárního čísla a zobrazit model libovolné kraslice. Modelů si uživatel může zobrazit větší množství zároveň a s každou kraslicí může uživatel otáčet a prohlédnout si ji ze všech stran.



Obrázek 4.2 3D model kraslice

Naopak nevýhodou digitalizace s použitím přípravku je, že snímky jsou pořizovány pomocí webkamery, takže se kvalitou nemohou rovnat snímkům pořízeným kvalitním fotoaparátem. Dalším mínusem je, že kraslice jsou foceny pouze z přední strany. Detaily na pólech kraslice jsou tedy buď hůře viditelné, nebo jsou úplně zakryté držáky.



Obrázek 4.3 Výsledek digitalizace pomocí přípravku

5. ZÁVĚR

Cílem bakalářské práce bylo navrhnout a sestrojít přípravek, který by umožnil snadnější a hlavně kvalitnější digitalizaci sbírky kraslic, kterou spravuje Moravské zemské muzeum.

Ve druhé kapitole je popsán prototyp hardwarového řešení přípravku. Samotný přípravek je vybaven vším, co uživatel k pořízení snímků bude potřebovat. Je možné ho snadno zapojit a používat. Tělo přípravku, které bylo vytisknuto na 3D tiskárně, umožňuje snadné upevnění i vyjmutí kraslice. Slabším místem přípravku je spoj ramene a těla přípravku. Rameno je mírně vychýleno, což má za následek občasný mírný prokluz kraslice během krokování. Tento nedostatek z větší části eliminuje gumový povrch na držácích kraslice.

Ve třetí kapitole je popsána funkce desktopové aplikace a programu, který je nahrán v desce Arduino UNO. Oba programy společně umožňují ovládání krokového motoru, snímače vzdálenosti, webkamery a automatické pořizování snímků. Desktopová aplikace automaticky zpracovává snímky, umožňuje uživateli procházet data digitalizovaných kraslic v galerii a zobrazovat jejich 3D model. Dále byl vytvořen program, který dává uživateli možnost zobrazit a manipulovat s 3D modely kraslic.

Přípravek společně s programovým vybavením uživateli nabízí způsob jak jednoduše a automatizovaně digitalizovat sbírku kraslic. U již digitalizovaných kraslic může uživatel zobrazit 3D model se kterým může manipulovat. Prohlížení 3D modelů je v porovnání s prezentací dat za pomoci fotografií pohodlnější a dává ucelenější pohled na kraslici.

Výsledkem bakalářské práce je funkční přípravek a programové vybavení, které je možné využít k digitalizaci sbírky kraslic. Dalším výstupem jsou snímky a 3D modely kraslic, vypůjčené od Moravského zemského muzea, pořízené pomocí přípravku.

LITERATURA

- [1] „Jak funguje CCD čidlo", *FotoAparát.cz*. <https://www.fotoaparar.cz/clanek/233/jak-funguje-ccd-cidlo-5021/> (viděno 18. prosinec 2021).
- [2] J. Reichl a M. Všetička, „Encyklopedie fyziky", 2006.
<http://fyzika.jreichl.com/main.article/view/1533-bayerova-masko> (viděno 2. leden 2022).
- [3] „Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications". <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-camera-sensors-for-machine-vision-applications/> (viděno 18. prosinec 2021).
- [4] „Úvod do techniky CCD čipů". <https://www.gxccd.com/art?id=303&lang=405> (viděno 18. prosinec 2021).
- [5] „Introduction to Image Sensors | LUCID Vision Labs". <https://thinklucid.com/tech-briefs/understanding-digital-image-sensors/> (viděno 2. leden 2022).
- [6] „Mechanická, nebo elektronická závěrka? Jaké jsou mezi nimi rozdíly a kterou vybrat", *Milujeme fotografii – vše o digitální fotografii*.
<https://www.milujemefotografii.cz/mechanicka-nebo-elektronicka-zaverka-jake-jsou-mezi-nimi-rozdily-a-ktou-vybrat> (viděno 2. leden 2022).
- [7] „ATEsystem - Řádkové kamery - Typy a technologie".
<https://eshop.atesystem.cz/clanek/187/dil-1-radkove-kamery--typy-a-technologie> (viděno 2. leden 2022).
- [8] navody.arduino-shop.cz, „Driver krokových motorů A3967 | Návod Drátek".
<https://navody.drátek.cz/navody-k-produktum/driver-krokovych-motoru-a3967.html> (viděno 14. prosinec 2021).
- [9] „I2C tutorial". <https://www.robot-electronics.co.uk/i2c-tutorial> (viděno 15. duben 2022).
- [10] „Welcome to Processing!", *Processing*. <https://processing.org/> (viděno 24. duben 2022).
- [11] „Reference", *Processing*. <https://processing.org/reference/> (viděno 25. duben 2022).
- [12] „Shapes 3D: Main Page". <http://lagers.org.uk/s3d4p/ref/index.html> (viděno 25. duben 2022).

SEZNAM PŘÍLOH

| | |
|--|----|
| PŘÍLOHA A - ELEKTRONICKÉ PŘÍLOHY | 47 |
|--|----|

Příloha A - Elektronické přílohy

V příloze jsou zdrojové kódy pro Arduino a desktopovou aplikaci. Dále obsahuje všechny modely těla přípravku a jeho sestavu. Součástí jsou i snímky kraslic nasnímaných pomocí přípravku.

| | | |
|---|------------------------------|--|
| — | Modely..... | Modely těla přípravku |
| — | —Sestava..... | Sestava přípravku |
| — | Programy-BP..... | Zdrojové kódy |
| — | —Arduino_Controller..... | Desktopová aplikace |
| — | —ImageProcess..... | Program pro zobrazení 3D modelů kraslic |
| — | —Arduino_Controller_EXE..... | Desktopová aplikace .exe |
| — | —Stepper_Controller..... | Program pro Arduino UNO |
| — | Snímky_Kraslic..... | Snímky kraslic pořízené s pomocí přípravku, zpracované snímky pro zobrazení 3D modelu |
| — | —KR001 | |
| — | —KR002 | |
| — | —KR003 | |
| — | —KR004 | |
| — | —KR005 | |
| — | —KR006 | |