

# 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

## RASPBERRY PI ČTEČKA 1D/2D KÓDŮ

1D/2D CODE READER USING RASPBERRI PI

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Jakub Nevláčil

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Peter Honec, Ph.D.

BRNO 2020

# Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Jakub Nevláčil

**ID:** 198594

**Ročník:** 3

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## Raspberry Pi čtečka 1D/2D kódů

### POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navržení a vytvoření kamerové čtečky 1D a 2D kódů na platformě Raspberry PI. Aplikace by měla splňovat standardní průmyslové požadavky (zobrazení a uložení výsledku, komunikace s PLC...). Součástí práce je výběr vhodné kamery, optiky a osvětlení. Předpokládaný použitý programovací jazyk je Python s knihovnami OpenCV.

1. Seznamte se s řetězcem strojového vidění, čtením datových kódů a aplikací strojového vidění v průmyslu.
2. Seznamte se s vývojem aplikací v jazyce Python na platformě Raspberry PI a knihovnou OpenCV.
3. Vyberte vhodný kamerový modul, optiku a osvětlení.
4. Navrhněte mechaniku modulu s využitím 3D tisku.
5. Navrhněte a vytvořte obslužnou aplikaci s definovanými požadavky (zobrazení a ukládání výsledků, GUI, DIO,...) event. servisní WEB aplikaci.
6. Zhodnoťte řešení (omezení, rychlost, vylepšení,...)

### DOPORUČENÁ LITERATURA:

ASHWIN PAJANKAR: Raspberry Pi Computer Vision Programming, ISBN 978-1784398286

HALFACREE GARET, UPTON EBEN: Raspberry Pi: User Guide 3rd Edition, ISBN 978-80-251-4819-8

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

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 8.6.2020

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

**doc. Ing. Václav Jirsík, CSc.**  
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**

Cílem této bakalářské práce je vytvoření obslužné aplikace pro čtečku 1D/2D čárových kódů na platformě Raspberry Pi. Pro tuto aplikaci je dále vytvořeno jak grafické, tak i webové rozhraní. Aplikace je schopna přečíst nasnímaný 1D/2D čárový kód, jeho obsah zobrazit v uživatelském rozhraní a následně odeslat do PLC. K naprogramování aplikace je použit jazyk Python. Ke čtení čárových kódů je použita knihovna OpenCV a další knihovny, které knihovnu OpenCV využívají. Předmětem této bakalářské práce je dále také návrh mechaniky modulu čtečky včetně výběru kamery a osvětlení.

## **Klíčová slova**

Raspberry Pi, Python, OpenCV, čárový kód, Flask, Tkinter

## **Abstract**

The aim of this bachelor's thesis is to create a service application for a 1D / 2D barcode reader on the Raspberry Pi platform. Both a graphical and a web interface were also created for this application. The application is able to read the captured 1D/2D barcode, display its content in the user interface and then send the content to the PLC. This application was programmed in Python programming language. The OpenCV library and other libraries that use the OpenCV library are used to read the barcodes. The next subject of this bachelor's thesis is also the design of the mechanics of the barcode reader module, including the selection of the camera and lighting.

## **Keywords**

Raspberry Pi, Python, OpenCV, barcode, Flask, Tkinter

## **Bibliografická citace:**

NEVLÁČIL, Jakub. *Raspberry PI čtečka 1D/2D kódů* [online]. Brno, 2020 [cit. 2020-06-03]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/127115>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Peter Honec.



## **Prohlášení**

Prohlašuji, že svou bakalářskou práci na téma Raspberry PI čtečka 1D/2D kódů jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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: **8. června 2020**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské práce panu Ing. Peteru Honcovi, Ph.D. a konzultantovi Ing. Pavlu Popovskému za cenné rady a odbornou pomoc při zpracování mé bakalářské práce.

V Brně dne: **8. června 2020**

.....  
podpis autora

# Obsah

1	Úvod .....	1
2	Strojové vidění.....	2
2.1	Princip strojového vidění .....	2
2.2	Typické úlohy pro strojové vidění v průmyslu .....	3
2.2.1	Nalezení povrchových vad .....	4
2.2.2	Rozpoznání, nalezení polohy a počítání .....	4
2.2.3	Měření a kontrola tolerancí.....	5
2.2.4	Identifikace barev.....	6
2.2.5	Čtení kódů a textů.....	6
2.3	Snímání obrazu.....	7
2.3.1	Senzor obrazu .....	7
2.4	Osvětlení .....	7
2.4.1	Typ zdroje světla.....	7
2.4.2	Variety provedení zdrojů světla .....	8
2.4.3	Vlnová délka osvětlení .....	9
2.5	Objektiv .....	10
2.5.1	Parametry objektivu .....	10
3	Čárové kódy.....	11
3.1	1D kódy .....	11
3.2	2D kódy .....	12
4	Programovací jazyk.....	13
4.1	Python.....	13
4.1.1	Python a Raspberry Pi.....	13
4.2	Knihovna OpenCV.....	14
5	Výběr komponent čtečky .....	15
5.1	Raspberry Pi.....	15
5.2	Výběr kamerového modulu .....	15
5.3	Výběr osvětlení.....	16
5.4	Napájení .....	17
5.5	Řízení osvětlení .....	18
5.6	Zvuková signalizace.....	19
6	Návrh mechaniky modulu .....	20
7	obslužná aplikace .....	24
7.1	Knihovny pro detekci a čtení čárových kódů.....	24
7.2	Odesílání dat do PLC .....	26
7.2.1	Knihovna socket.....	26

7.3	Knihovna Tkinter.....	27
7.4	Výsledná podoba obslužné aplikace.....	28
7.4.1	Single mode a Freerun mode .....	29
7.5	Tvorba webové aplikace .....	30
7.5.1	Flask.....	30
7.5.2	JavaScript.....	30
7.6	Výsledná podoba webové aplikace .....	30
7.7	Instalace knihoven .....	32
7.8	Uvedení kamery do provozu.....	33
7.9	Spuštění aplikací .....	33
8	Závěr .....	35
	Literatura.....	37
	Seznam symbolů, veličin a zkratk .....	39
	Seznam příloh.....	40

## Seznam obrázků

Obr. 2.1 Obecné uspořádání systému strojového vidění[1] .....	3
Obr. 2.2 Příklad nalezení povrchové vady: přerušení housenky těsnícího tmelu[2] .....	4
Obr. 2.3 Příklad úlohy rozpoznání: špatně zalisovaný vodič v konektoru[2] .....	5
Obr. 2.4 Příklad úlohy počítání: kontrola naplnění blistru s léky[2] .....	5
Obr. 2.5 Příklad úlohy kontroly tolerancí: měření závitů na kulovém čepu[2].....	6
Obr. 2.6 Příklad úlohy čtení textu: kontrola data výroby nebo expirace[2] .....	6
Obr. 2.7 Princip zadního osvětlení[4] .....	8
Obr. 2.8 Osvětlovač s koaxiálním difuzním světlem[4] .....	8
Obr. 2.9 Kopulový difuzní osvětlovač[4] .....	8
Obr. 2.10 Princip dark-field osvětlovače[5].....	9
Obr. 2.11 Princip bright-field osvětlovače[5] .....	9
Obr. 2.12 Kruh barev[4] .....	9
Obr. 2.13 Závislost ohniskové vzdálenosti na zorném úhlu[6] .....	10
Obr. 2.14 Princip clony[6] .....	10
Obr. 3.1 Příklad 1D čárového kódu typu EAN [10].....	12
Obr. 3.2 Příklad 2D kódu typu QR [10] .....	12
Obr. 5.1 Jednodeskový počítač Raspberry Pi 4 [11] .....	15
Obr. 5.2 Arducam 16Mpx IMX298 MIPI Color Camera Module [12] .....	16
Obr. 5.3 Použitá výkonová LED dioda .....	17
Obr. 5.4 Raspberry Pi GPIO sběrnice [14] .....	18
Obr. 5.5 Schéma zapojení tranzistoru s LED diodami.....	19
Obr. 6.1 Návrh mechaniky modulu-pohled zepředu .....	20
Obr. 6.2 Spodní patro modulu s připevněným Raspberry Pi .....	21
Obr. 6.3 Horní patro krabičky s rozmístěnou elektronikou.....	22
Obr. 6.4 Víko krabičky s kapsou pro ventilátor .....	23
Obr. 7.1 Ukázka čtení čárového kódu typu EAN13 knihovnou pyzbar.....	25
Obr. 7.2 Ukázka čtení čárového kódu typu Datamatrix knihovnou pylibdmtx .....	25
Obr. 7.3 Ukázka přijímání dat v PLC.....	27
Obr. 7.4 Výsledná podoba obslužné aplikace.....	29
Obr. 7.5 Záložka Home webové aplikace.....	31
Obr. 7.6 Záložka Settings webové aplikace .....	32

# 1 ÚVOD

Pod pojmem čárový kód si většina lidí si asi představí klasický jednorozměrný čárový kód, se kterým se denně setkáváme na obalech zboží v obchodech. Tyto čárové kódy jsou schopné nést pouze krátký řetězec složený z čísel, případně kombinaci čísel a písmen, který na základě databáze daný výrobek jednoznačně identifikuje. Tyto kódy nám značně usnadňují a urychlují každodenní život.

Postupem času nám ale kapacita jednorozměrných čárových kódů přestala stačit, a tak vznikly čárové kódy, které kódují informaci ve dvou směrech. Tyto kódy mohou nést nejen řetězec alfanumerických znaků stejně jako jednorozměrné kódy, ale jejich kapacita jim dovoluje uchovávat i text. Tento text může obsahovat například instrukce k použití, emailovou adresu nebo i odkaz na internetovou stránku.

Dvourozměrné kódy jsou hojně využívány především v průmyslu díky své schopnosti přenášet větší objem dat.

Tato bakalářská práce vznikla na základě požadavku od firmy TE Connectivity Kuřim, který zahrnoval návrh, sestavení a naprogramování univerzální průmyslové čtečky 1D/2D kódů. Výsledkem této práce by měl být funkční modul čtečky s Raspberry Pi, kamerou a osvětlením. Tato čtečka by dále měla být schopna číst 1D/2D kódy, hodnotu těchto kódů prostřednictvím webové aplikace zobrazit a následně odeslat do PLC.

## 2 STROJOVÉ VIDĚNÍ

Historie zpracování obrazu v počítači se začala psát v sedmdesátých letech dvacátého století, kdy již tehdejší výpočetní technika umožnila zpracovávat tak velké objemy dat, které nesou obrazovou informaci. Vznikl nový obor, pro který se vžilo označení počítačové vidění (computer vision). Tento název nyní označuje obecně systémy, které pracují automaticky na základě informací získaných ze zpracování obrazu z kamery. Objektem počítačového vidění může být v podstatě cokoli, např. dopravní situace nebo lidská tvář či lidská činnost. A rovněž i proces výroby. Pro počítačové vidění využívané v průmyslové výrobě je dnes obvykle volen termín strojové vidění (machine vision).

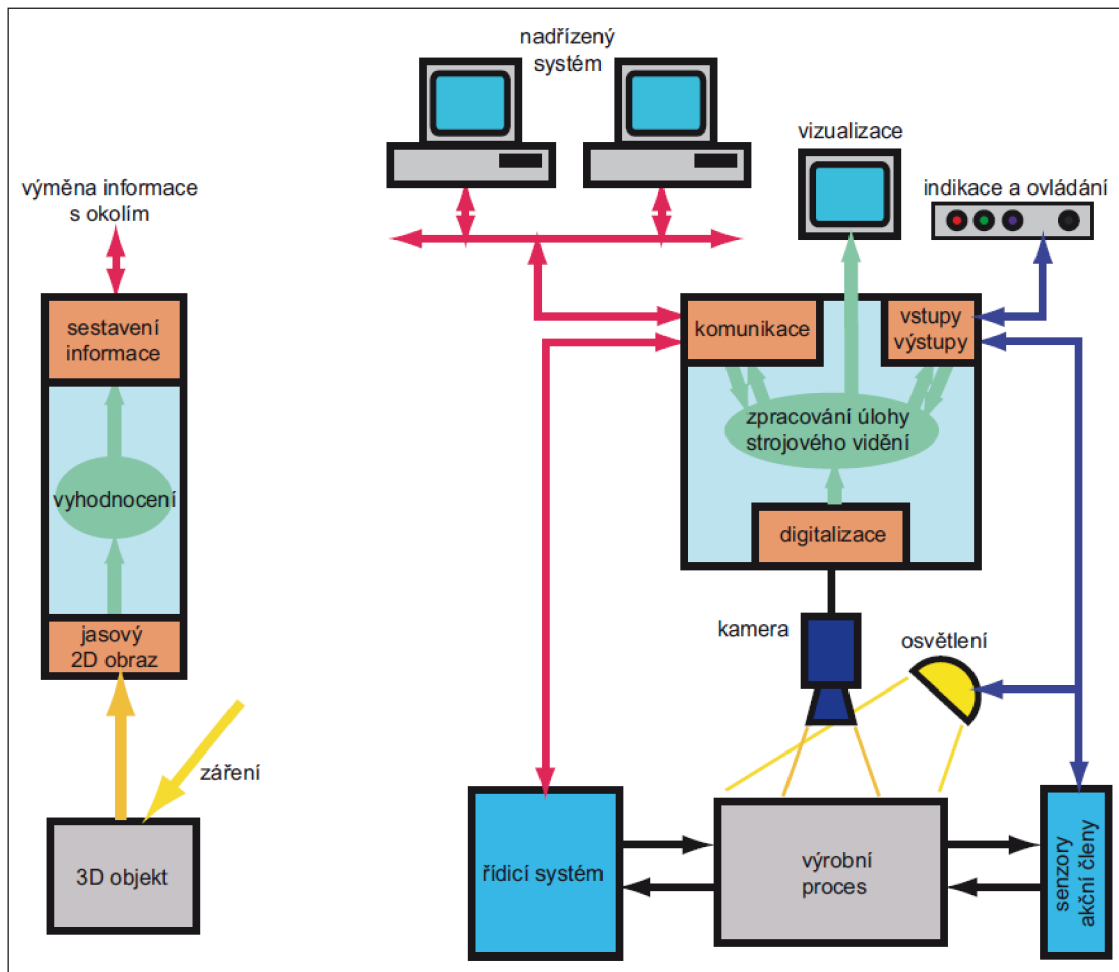
Strojové vidění je v současné době považováno za využití počítačového vidění v průmyslové automatizaci. Je charakterizováno vazbou na výrobní proces a orientací na typické úlohy spojené s řízením výrobního procesu. K těmto úlohám patří hlavně vizuální inspekce předepsaných viditelných parametrů, počítání objektů, hledání defektů apod. [1]

### 2.1 Princip strojového vidění

Strojové vidění „přistupuje“ k danému úkolu podobně jako člověk. Stejně jako lidské oko zachytí i kamera obraz zkoumaného předmětu, systém jej vyhodnotí podle předepsaného algoritmu a provede akci na základě výsledku vyhodnocení. Pro účely technického popisu je nesnadnější hledět na systém strojového vidění jako na obecný senzor, jak ukazuje obr. 2.1 vlevo.

Sledovaný objekt, většinou trojrozměrný je ozařován zdrojem záření. Objekt musí být schopen toto záření odrazit tak, aby odrazžené záření vytvořilo na snímacím prvku senzoru jasový dvojrozměrný obraz. Důležité je, aby v tomto obraze byla obsažena informace, kterou je zapotřebí o sledovaném objektu znát.

Obraz vytvořený na snímacím prvku je převeden na vhodnou měronosnou veličinu a vyhodnocen tak, aby z něj byla získána požadovaná informace o sledovaném objektu. Ta je přes vhodné rozhraní předána ze senzoru do okolí. Obvyklou konstrukci takového senzoru ukazuje obr. 2.1 vpravo. [1]



Obr. 2.1 Obecné uspořádání systému strojového vidění[1]

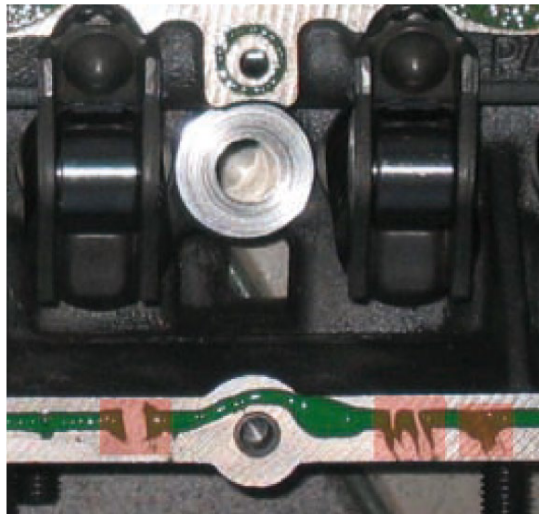
## 2.2 Typické úlohy pro strojové vidění v průmyslu

Strojové vidění většinou řeší kontrolu produktů tak, aby byla zaručena jejich naprostá bezchybnost při výstupu z výrobního procesu. Výrobní proces lze definovat jako posloupnost výrobních operací. Existuje mnoho teoretických prací, které se zabývají vlivem organizace výrobního procesu na kvalitu výroby, vždy však obsahují podmínku bezchybného provedení alespoň některých výrobních operací k tomu, aby byl vyroben bezchybný výrobek. Právě strojové vidění umožňuje tuto podmínku splnit. Nejtypičtějším použitím strojového vidění ve výrobě je tedy kontrola správnosti provedení výrobní operace. [2]



### 2.2.1 Nalezení povrchových vad

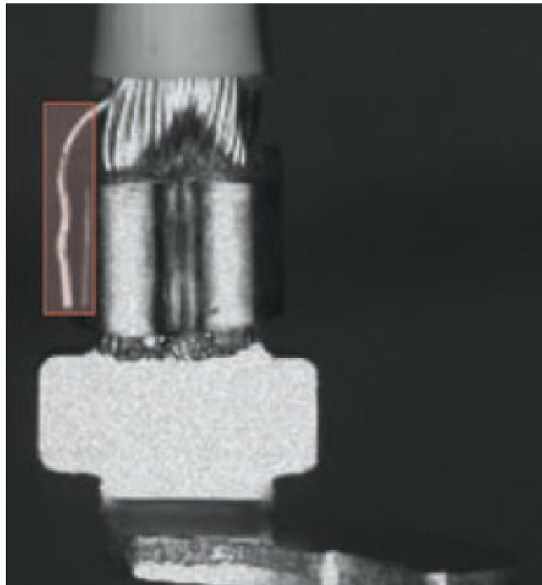
Tato úloha je typická pro fázi shromažďování materiálu k operaci provázené vstupní kontrolou. Může jít o identifikaci poškozených povrchů, vad nátěrů či povrchových úprav, neúplných potisků atd. Je-li nátěr sám výrobní operací, uplatní se tento typ úlohy i ve fázi vyzvednutí výsledného produktu. Na obr. 2.2 je ukázán příklad špatně nanesené housenky těsnícího tmelu. [2]



Obr. 2.2 Příklad nalezení povrchové vady: přerušení housenky těsnícího tmelu[2]

### 2.2.2 Rozpoznání, nalezení polohy a počítání

Tato úloha může mít mnoho podob. Nejčastěji se v ní zjišťuje, je-li správný objekt na správném místě. Například před provedením operace zalisování je nutné se přesvědčit, zda jsou k zalisování připraveny všechny díly ve správné poloze. Totéž je třeba udělat po skončení operaci – chybu v zalisovaném konektoru ukazuje obr. 2.3. Zmíněná úloha se rovněž vyskytuje při vstupní kontrole. Může jít o identifikaci deformované části, otřepů po stříhání, nevyvrtaného otvoru nebo poškozené hrany. Počítání se uplatní při kontrole naplněné přepravky nebo blistru (obr. 2.4). [2]



Obr. 2.3 Příklad úlohy rozpoznání: špatně zalisovaný vodič v konektoru[2]



Obr. 2.4 Příklad úlohy počítání: kontrola naplnění blistru s léky[2]

### 2.2.3 Měření a kontrola tolerancí

Úloha měření prostupuje celou výrobní operací. Měří se tolerance vstupních dílů, poloha dílů před operací i tolerance hotového výrobku. Na obr. 2.5 je příklad kontroly tolerancí kulového čepu. [2]



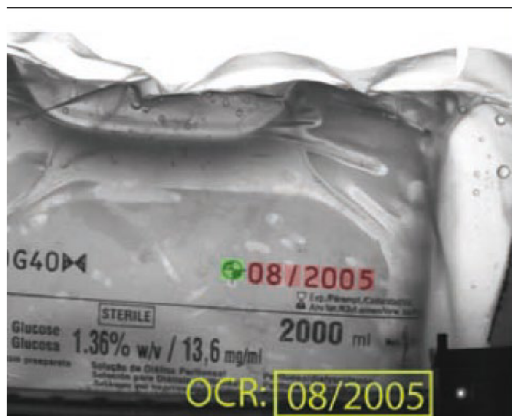
Obr. 2.5 Příklad úlohy kontroly tolerancí:  
měření závitů na kulovém čepu[2]

## 2.2.4 Identifikace barev

Barvy je třeba rozpoznávat při vstupní nebo výstupní kontrole, může jít např. o jeden ze znaků při identifikaci typu materiálu. V potravinářství může barva potravin signalizovat sníženou kvalitu produktu vstupujícího do operace vyskladnění. [2]

## 2.2.5 Čtení kódů a textů

Kódy, ať již čárové nebo maticové, se čtou hlavně v první a druhé fázi výrobní operace, kdy je zjišťováno, zda je k operaci připraven správný díl na správném místě. Čtení nebo verifikace kódů jsou málokdy samostatnou úlohou strojového vidění ve výrobní operaci, ale je-li tomu tak, je výhodnější svěřit ji specializované čtečce. Vykonná-li však systém strojového vidění v dané operaci více úloh, je výhodné použít ho i pro čtení kódů a textů. Příklad čtení je ukázán na obr. 2.6. [2]



Obr. 2.6 Příklad úlohy čtení textu:  
kontrola data výroby nebo expirace[2]

## 2.3 Snímání obrazu

Abychom mohli obraz zpracovávat a vyhodnocovat, je nejdříve nutné požadovaný objekt správně nasnímat. K nasnímání obrazu je zapotřebí snímač, osvětlení a objektiv.

### 2.3.1 Senzor obrazu

Senzory obrazu složí především k převodu snímané scény na digitální signál.

Při výběru kamery máme na výběr ze dvou základních technologií snímačů: CCD (Charge Coupled Devices) nebo CMOS (Complementary Metal-Oxide Semiconductor). Hlavní předností CCD senzorů je jejich vysoká kvalita obrazu. Oproti CMOS senzorům jsou ale dražší, mají vyšší spotřebu a menší rychlost sekvenčního snímání.

Dále můžeme senzor obrazu vybírat podle jeho struktury. Podle struktury se senzory dělí na plošné, což jsou klasické senzory používané ve fotoaparátech, a řádkové. Řádková kamera obsahuje snímač, který má v principu jeden jediný řádek obrazových bodů (pixelů). Kamera pracuje v podstatě jako kancelářský skener. Pod kamerou se lineárně pohybuje snímaný objekt nebo pás a kamera jej snímá řádek po řádku a skládá dohromady obraz, který posílá ke zpracování do nadřazeného vision systému. Toto uspořádání umožňuje snímání pohyblivých materiálů vysokou rychlostí s poměrně velkými detaily. [3] Z toho plyne, že plošné kamery jsou vhodné především pro snímání nehybných nebo pomalu se pohybujících objektů.

Nakonec si zvolíme, zda chceme kameru barevnou nebo černobílou. Černobílá kamera na rozdíl od barevné kamery přenáší menší objem dat. Proto se v průmyslu často používá právě černobílá kamera.

## 2.4 Osvětlení

Výběr osvětlení může být pro danou aplikaci počítačového vidění stěžejní. Abychom mohli navrhnout systém osvětlení musíme nejprve znát v jakém prostředí se měřený objekt bude nacházet, typ jeho povrchu a jeho barvu. Navržené osvětlení by pak mělo splňovat tyto tři podmínky: [4]

1. Maximalizovat kontrast bodů našeho zájmu
2. Minimalizovat kontrast okolí
3. Odstranit okolní rušivé vlivy

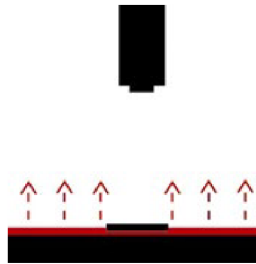
### 2.4.1 Typ zdroje světla

Typicky používané zdroje světla v aplikacích strojového vidění jsou zářivky, halogenové žárovky, LED, rtuťové výbojky a xenonové výbojky.

Zářivky, halogenové žárovky a LED se nejčastěji používají pro aplikace s malou až střední náročností měření. Rtuťové výbojky, xenonové výbojky a vysokotlaké sodíkové výbojky jsou typicky používané pro osvětlování velkých ploch nebo v místech, kde je potřeba velmi jasný zdroj světla. Rtuťové výbojky jsou dále často využívány v mikroskopii. [4]

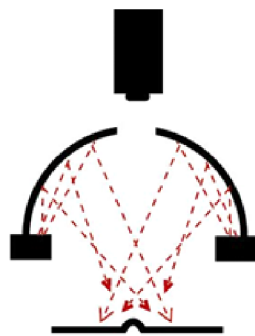
## 2.4.2 Varianty provedení zdrojů světla

Zadní osvětlení se nejběžněji používá k detekci děr, ke zjištění umístění a orientace objektů a ke kontrole tvaru objektu. [4]

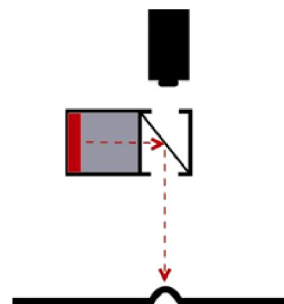


Obr. 2.7 Princip zadního osvětlení[4]

Difuzní osvětlení se uplatňuje především při osvětlování reflexních ploch nebo ploch se smíšenou reflexivitou. Difuzní osvětlovače můžeme dále rozdělit na dva typy- kopulový difuzní osvětlovač a osvětlovač s koaxiálním difuzním světlem. [4]

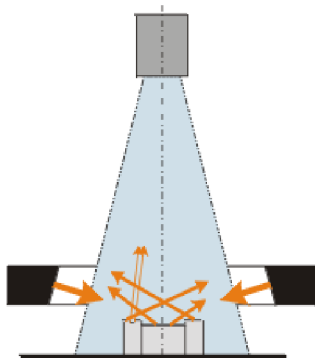


Obr. 2.9 Kopulový difuzní osvětlovač[4]



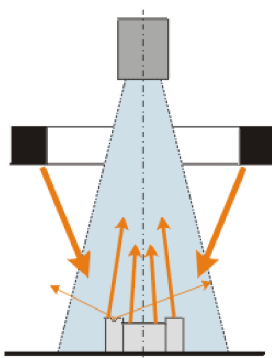
Obr. 2.8 Osvětlovač s koaxiálním difuzním světlem[4]

Osvětlovače dark – field (temné pole) jsou vhodné především pro detekci prasklin, vrypů a jiných vad u transparentních materiálů. [5]



Obr. 2.10 Princip dark-field osvětlovače[5]

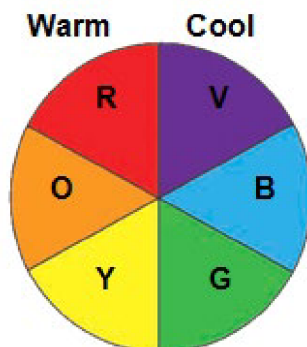
Osvětlovače bright-field (jasné pole) jsou vhodné pro snímání předmětů malých rozměrů. [5]



Obr. 2.11 Princip bright-field osvětlovače[5]

### 2.4.3 Vlnová délka osvětlení

Při výběru vlnové délky osvětlení se řídíme tzv. kruhem barev, který je znázorněn na obr. 3.6. Pokud má tedy náš objekt např. červenou barvu, tak k jeho osvětlení použijeme protější barvu z tohoto kruhu. Tím docílíme toho, že červená barva bude zelené světlo pohlcovat a dojde tedy k zvýšení kontrastu.



Obr. 2.12 Kruh barev[4]

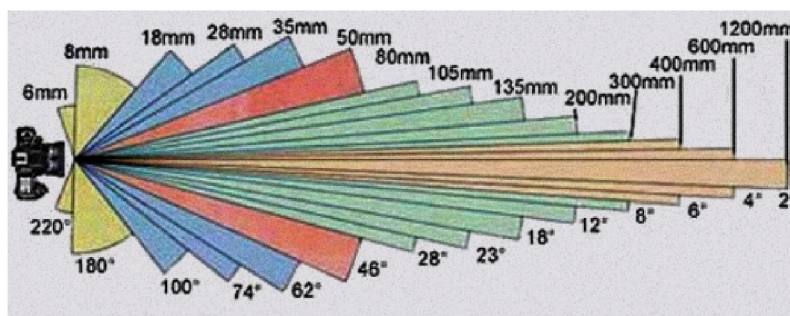
## 2.5 Objektiv

Objektivy mají za úkol soustředit světlo odrážející se od pozorovaného objektu na senzor kamery. Skládají se ze soustavy čoček a clony tvořené několika lamelami.

### 2.5.1 Parametry objektivu

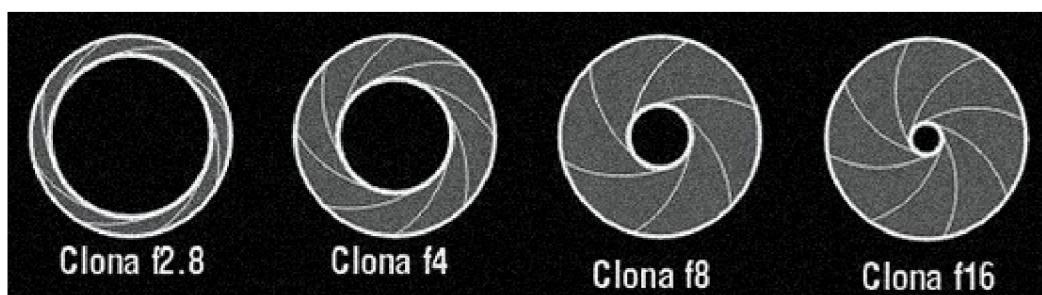
Jedním z nejdůležitějších parametrů objektivu je ohnisková vzdálenost. Ohnisková vzdálenost objektivu je vzdálenost středu optické soustavy od roviny (senzor kamery), na které dokáže optická soustava ostře vykreslit snímanou scénu.[6]

S ohniskovou vzdáleností souvisí také zorný úhel zachycené scény. Zorný úhel je určen poměrem velikosti snímače a ohniskové vzdálenosti. Čím je tedy ohnisková vzdálenost menší, tím větší je zorný úhel.



Obr. 2.13 Závislost ohniskové vzdálenosti na zorném úhlu[6]

Dalším parametrem objektivu je světelnost objektivu. Světelnost popisuje schopnost objektivu propouštět světlo a značí se písmenem  $f$  společně s číselným údajem, např.  $f/2,8$ . Čím nižší číslo, tím více světla je objektiv schopen propustit a tím kratší časy potřebujete ke správné expozici snímku, což je důležité např. při sledování rychlých dějů na výrobní lince. [7]



Obr. 2.14 Princip clony[6]



## 3 ČÁROVÉ KÓDY

S čárovými kódy se setkáváme především na obalech zboží v obchodech, poštovních zásilkách nebo v průmyslu. Čárové kódy nejčastěji obsahují data potřebná k identifikování dané položky. Například na pokladnách v obchodech slouží čárové kódy pouze k identifikaci položek pomocí příslušné databáze. Čárové kódy dnes ale mohou obsahovat i další dodatečné informace. Čárové kódy se dělí podle počtu směrů, ve kterých jsou data uložena na 1D a 2D kódy.

### 3.1 1D kódy

Symbol čárového kódu se skládá z určitého počtu čar a mezilehlých mezer. Před a za symbolem musí být klidová zóna-prázdné místo určité šířky bez jakéhokoliv potisku. Symbol začíná znakem start, pak následují vlastní data s případným kontrolním součtem a na konci je znak stop.[8]

1D čárové kódy se běžně vyskytují na spotřebním zboží a bývají numerické a některé z nich i alfa-numerické (např. Code128 zvládne všechny znaky z tabulky ASCII 128). Tyto kódy využívají ke kódování řadu černých linek a bílých prostorů mezi nimi s proměnnou šířkou.

1D čárové kódy jsou vhodné pro zakódování ID produktů nebo krátkého tagu, ale to je asi tak vše, co od nich můžeme očekávat, protože řetězce, které mají více než 25-30 znaků, tvoří dlouhý čárový kód, který je pro běžné využití nepraktický (zabírá mnoho místa nebo je potřeba jej načítat z větší výšky, což nemusí každá čtečka zvládnout). Nicméně i přes jejich malou datovou kapacitu jsou tyto čárové kódy jedny z nejpoužívanějších díky jejich jednoduché tvorbě a jednoduchému čtení.

Největšího smyslu nabývají 1D čárové kódy až po připojení k databázi. Možné příklady využití 1D čárového kódu:

- Skenování čárového kódu artiklu na pokladně v supermarketu. Po načtení kódu, nejčastěji ID položky, se na kase zobrazí artikl a jeho cena, která je načtena z databáze. Tyto systémy čárových kódů jsou nezbytností pro velké maloobchodníky a napomáhají zvýšit přesnost zásob a šetřit čas.
- Skenování čárového kódu operace z výrobní průvodky, kterou vygeneroval informační systém pro řízení výroby. Po načtení čárového kódu operace může pracovník zaznamenat data z výroby přímo ke konkrétní operaci, např. spotřebovaný čas nebo počet vyrobených kusů.
- U velkoobchodníka může jít o řízení skladu pomocí čárových kódů, kterými identifikuje zboží, velikost balení tohoto zboží (různý čárový kód pro balení po 100 ks, pro paletu apod.) a pozici, na které je toto zboží umístěno. [9]



## 3.2 2D kódy

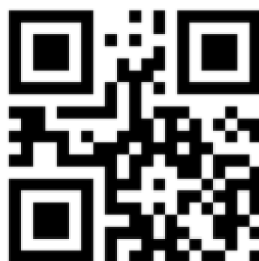
2D kódy, jako je např. QR Code nebo PDF417, používají vzory čtverců, šestiúhelníků, bodů a dalších tvarů pro zakódování dat. Vzhledem k tomu, že 2D kód používá tyto matice, tak by se už neměl nazývat čárovým kódem, jelikož je toto označení matoucí. Vzhledem k jejich struktuře mohou 2D kódy obsahovat více dat než 1D čárové kódy (až 2000 znaků), zatímco se objevují ve fyzicky stále menším formátu. Data jsou zakódována na základě horizontálního i vertikálního uspořádání vzoru, a proto jsou čteny ve dvou rozměrech. Adresa Vašich webových stránek a kontaktní informace nyní mohou být reprezentovány černobílým čtvercem o rozměrech 2 × 2 cm.

Největšího smyslu nabývají 2D kódy v případech, kdy je nutné přenášet větší množství dat bez připojení k databázi. Možné příklady využití 2D kódu:

- QR kód je využíván pro úhradu plateb na fakturách přes mobilní bankovní aplikace
- 2D kód PDF417 je využíván v leteckém průmyslu jako statická databáze, jelikož dokáže nést až 1800 znaků. Například na letence může kód obsahovat údaje o letu.
- V marketingu je 2D kód (QR code) využíván pro sdělování informací a podpoře prodeje – po načtení kódu je zákazník přesměrován na webovou stránku společnosti se slevovou akcí. Výhodou je možnost skenovat QR code v podstatě jakýmkoliv telefonem vybaveným fotoaparát a jednoduchým dekódovacím software.
- 2D kód se v podstatě hodí kamkoliv, kde jsou vysoké nároky na úsporu místa (malý předmět, potřeba malého štítku) nebo je potřeba zakódovat více informací. Příkladem může být tisk štítku o malých rozměrech s 2D kódem, který obsahuje informace o zařízení, datu výroby, informace o vlastníkovi, nebo doporučených intervalů servisní prohlídky.[9]



Obr. 3.1 Příklad 1D čárového kódu typu EAN [10]



Obr. 3.2 Příklad 2D kódu typu QR [10]

## 4 PROGRAMOVACÍ JAZYK

V zadání práce je jako podmínka použit programovací jazyk Python s knihovnou pro počítačové vidění OpenCV. V této kapitole si tedy tyto dva pojmy více přiblížíme.

### 4.1 Python

Python je vysokoúrovňový skriptovací programovací jazyk, který v roce 1991 navrhl Guido van Rossum. Nabízí dynamickou kontrolu datových typů a podporuje různá programovací paradigmaty, včetně objektově orientovaného, imperativního, procedurálního nebo funkcionálního. V roce 2018 vzrostla jeho popularita a zařadil se mezi nejoblíbenější jazyky. V řadě různých žebříčků dosahuje jedno z prvních třech míst, výjimkou nebývají první místa.

Python je vyvíjen jako open source projekt, který zdarma nabízí instalační balíky pro většinu běžných platform (Unix, MS Windows, macOS, Android); ve většině distribucí systému GNU/Linux je Python součástí základní instalace. [15]

Jazyk Python se postupem času průběžně vyvíjí a dnes existují již tři verze tohoto jazyka, které spolu ale nejsou kompatibilní, Python 1, Python 2 a Python 3. Jelikož Python 1 se již dlouhou dobu nepoužívá a podpora pro Python 2 skončila 1. ledna 2020, aktuální verze je tedy Python 3. Tyto verze se od sebe liší například použitými knihovnami nebo rozdílnou syntaxí.

Jazyk python se doporučuje především pro začátečníky, jakožto vstup do světa programování, a to zejména díky své jednoduché syntaxi. Je ale hojně využíván i při tvorbě rozsáhlých projektů zabývajících se strojovým učením, zpracováním velkého množství dat nebo vývojem webů.

#### 4.1.1 Python a Raspberry Pi

Jak již bylo řečeno, programovací jazyk python je podporován na většině běžných platform. To platí i pro Raspbian, což je operační systém primárně určený pro počítače Raspberry Pi, který je založen na Debianové distribuci Linuxu.

Operační systém Raspbian je dodáván s již předinstalovaným balíčkem jazyka Python, není jej tedy třeba dodatečně instalovat. Předinstalovaná je jak verze Python 3, tak i verze Python 2 a to především z důvodu kompatibility se staršími knihovnami. Bude ale nutné doinstalovat některé doplňující balíčky, k čemuž slouží nástroj pip a příkaz „pip install“.

Nástroj pip funguje jako správce balíčků programovacího jazyka python. Pomocí tohoto nástroje můžeme velice snadno instalovat doplňující balíčky jazyka Python. Instalace balíčku se provádí v konzoli zadáním příkazu „pip install“

následovaného jménem požadovaného balíčku. Pip si následně automaticky stáhne balíček z jeho hlavního repozitáře nazývaného PyPI a nainstaluje jej.

## 4.2 Knihovna OpenCV

OpenCV (Open Computer Vision) je open source sada knihoven pro práci s počítačovým viděním (computer vision) v reálném čase. S knihovnou lze vyvíjet aplikace pro rozpoznávání gest, tváří, objektů atp. Je tedy využívána zejména v oblasti robotiky a pro implementaci aplikací z oblasti experimentálních rozhraní pro styk člověka s počítačem (Human-Computer Interaction) aj. S OpenCV je spojena firma Intel, která implementuje jeho podporu přímo do hardwaru – využívá technologii Intel Integrated Performance Primitives, která poskytuje vysoký výkon pro nízkoúrovňové rutiny pro zvuk, video, rozpoznávání řeči, kódování, dekodování nebo kryptografii.

Samotná knihovna OpenCV je velmi robustní. Obsahuje nástroje pro kompletní analýzu obrazu (filtrování obrazu, detekce objektů, analýzu pohybu), transformace obrazu, práci s videem, jednoduchou práci s okny a uživatelským rozhraním, kalibraci kamery, podporu sterea (zapojení více kamer) apod. Výhodou je multiplatformnost, framework existuje ve verzích pro GNU/Linux, Microsoft Windows i Mac OS X a to i pro 64b systémy.[16]

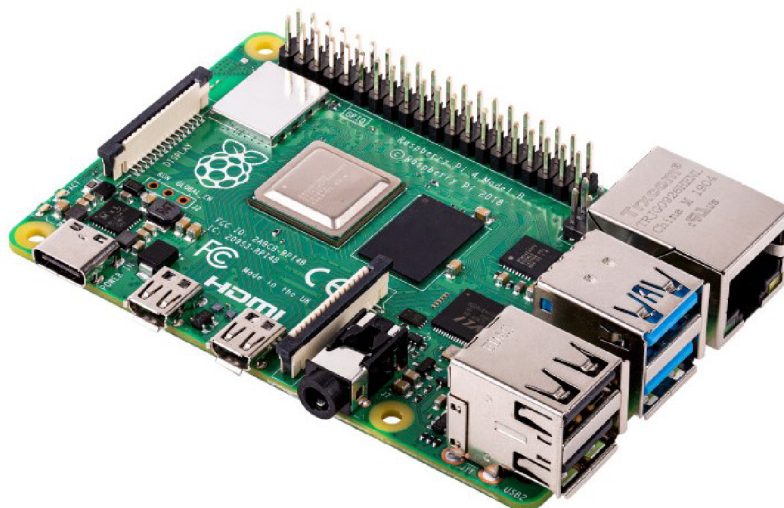
Knihovna OpenCV je především určena pro programovací jazyky C a C++. Je ale možné ji používat i ve spojení s jazykem Python, Java, MATLAB a dalšími.

## 5 VÝBĚR KOMPONENT ČTEČKY

### 5.1 Raspberry Pi

Jako výpočetní modul čtečky má být použit počítač Raspberry Pi. Raspberry Pi je jednočipový počítač, který je srovnatelný se (slabším) stolním počítačem. Obsahuje vývod pro monitor (HDMI), přes USB je možné připojit klávesnici a myš. Vyvinuto bylo již několik generací tohoto počítače, které se liší výkonem a zamýšleným použitím. Použitý mikroprocesor je z rodiny ARM, takže je srovnatelný s běžným smartphonem. Na počítači Raspberry Pi je možné provozovat různé distribuce Linuxu, RISC OS, jakož i Microsoft Windows 10 IoT Core.[11]

Od roku 2012 vyšlo již několik verzí tohoto počítače. Jako poslední a také nejvýkonnější vyšla v roce 2019 verze Raspberry Pi 4 Model B. V tomto projektu tedy bude použita tato verze se 4GB RAM[13]



Obr. 5.1 Jednodeskový počítač Raspberry Pi 4 [11]

### 5.2 Výběr kamerového modulu

Pro tuto práci byl vybrán kamerový modul Arducam 16Mpx IMX298 MIPI Color Camera Module z důvodu bezproblémové kompatibility s platformou Raspberry Pi. Tento kamerový modul disponuje 1/28" CMOS senzorem Sony IMX298 o rozlišení 16 Mpx. Jednou z dalších vlastností tohoto kamerového modulu je například možnost manuálního nastavení expoziční doby nebo možnost programovatelného ostření. Obě tyto funkce budou při tomto projektu užitečné. Aby bylo možné tento kamerový modul ovládat a využívat jeho funkce, je nutné použít knihovnu

poskytovanou výrobcem této kamery. Tato knihovna bude obsažena v příloze pod názvem arducam\_mipicamera.py

Při porovnání s ostatními kamerovými moduly pro platformu Raspberry Pi se tento kamerový modul vyznačuje zejména vyšším rozlišením, vyšší snímkovací frekvencí a také programovatelným ostřením.



Obr. 5.2 Arducam 16Mpx IMX298 MIPI Color Camera Module [12]

### 5.3 Výběr osvětlení

Tato čtečka bude určena především pro ruční skenování 1D/2D čárových kódů. Dá se tedy předpokládat, že kvalita pořízených snímků 1D/2D čárových kódů bude silně ovlivněna mírou otřesů a pohyblivostí čtečky při skenování. Tyto faktory jsou závislé na operátorovi čtečky. Z procesu skenování je tedy nutné co nejvíce eliminovat lidský faktor. Toho je možné dosáhnout vyžitím funkce manuálního nastavení expoziční doby kamery.

Expoziční doba kamery je časový interval, po který je senzor kamery vystaven světlu. Tento časový interval ovlivňuje jak jas výsledného snímku, tak i jeho ostrost. Čím tedy bude expoziční doba kratší, tím méně světla dopadne na

snímač a výsledný snímek pak bude tmavší. Zároveň ale snímač za tuto dobu zachytí méně pohybu a výsledný snímek tak bude ostřejší. Expoziční dobu je tedy třeba nastavit takovou, aby měl snímek dostatečný jas a zároveň byl dostatečně ostrý pro spolehlivé přečtení čárového kódu. Tento problém je možné vyřešit volbou dostatečně výkonného osvětlení.

Při výběru osvětlení je nutné dbát také na to, že se čtečkou bude manipulováno ručně. Je tedy žádoucí, aby osvětlení bylo co nejvíce kompaktní a lehké. Jako osvětlení tedy byly zvoleny 4 výkonové bílé LED diody. Každá dioda má výkon 1W a světelný tok 140lm. Požadavky na napájení každé z diod jsou 3,2V a 350mA.



**Obr. 5.3 Použitá výkonová LED dioda**

## 5.4 Napájení

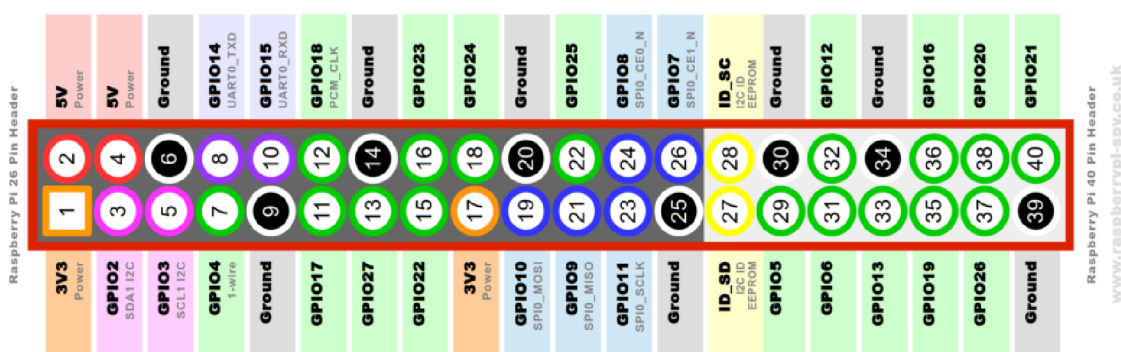
Pro napájení jednodeskového počítače Raspberry Pi 4 je primárně určen konektor USB-C. Tento způsob napájení je ale pro naše účely nevhodný, jelikož by napájecí konektor musel vyčnívat mimo tělo samotné čtečky do prostoru a při časté manipulaci se čtečkou by se tak mohl nečekaně odpojit nebo dokonce urazit. K účelům napájení počítače Raspberry Pi tak bude nejvhodnější využít možnost napájení skrze GPIO sběrnici. Jak je vidět z obr. 5.4, tak pro napájení slouží piny č. 2 a 6.

Dále je třeba zamyslet se nad energetickou náročností celé čtečky včetně LED diod. Výrobce počítače Raspberry Pi 4 doporučuje k napájení použít zdroj o proudovém výkonu alespoň 2,5A. Pokud by byly osvětlovací diody zapojeny paralelně, byl by jejich celkový proudový odběr 1,4A. Celkový proudový odběr by



tedy byl 3,9 A. Běžné síťové adaptéry o napětí 5 V mají nejvyšší proudový výkon 4 A. Bude tedy vhodnější pro napájení použít síťový zdroj o vyšším napětí a ke snížení napětí pak použít DC-DC step-down regulátor.

Pro napájení počítače Raspberry Pi byl tedy zvolen DC-DC step-down regulátor LM2596 s maximálním výstupním proudem 3A, který je pro napájení počítače dostatečný. Pro napájení Led diod byl pak zvolen DC-DC step-down regulátor LM2596 s možností omezení výstupního proudu. Oba tyto regulátory pak budou paralelně připojeny na síťový adaptér o výstupním napětí 12V. Výhodou použití těchto regulátorů je možnost použít síťový adaptér o výstupním napětí až 40V.



Obr. 5.4 Raspberry Pi GPIO sběrnice [14]

## 5.5 Řízení osvětlení

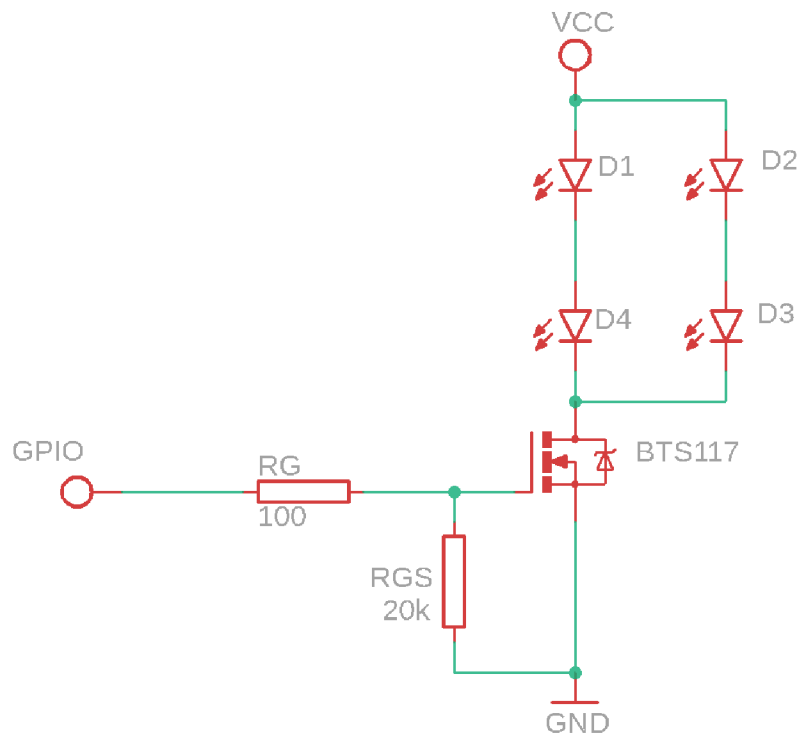
Dále je nutné navrhnout ovládání a spínání LED diod. K tomuto účelu budou použity piny sběrnice GPIO, ale jelikož řídicí piny sběrnice GPIO nejsou schopny napájet tak velkou zátěž, jako jsou čtyři 1W LED diody, bude potřeba použít tranzistor, který nám poslouží jako výkonový spínač. Při výběru tranzistoru je nutné dbát na to, aby jeho prahové napětí  $U_{GS}$  bylo menší, než je napětí řídicích pinů na sběrnici GPIO, což je 3,3V. A dále musí mít proud  $I_D$  větší, než je celkový odběr LED diod. Vybraný tranzistor BTS117 TO220 s napětím  $U_{GS} = 2,2V$  a proudem  $I_D = 7A$  obě tyto podmínky splňuje. Schéma zapojení tranzistoru je na obr. 5.5.

Ze schématu je vidět, že na GATE tranzistoru je připojen rezistor  $R_G = 100\Omega$ . Tento rezistor slouží k eliminaci oscilací způsobených kapacitami tranzistoru a induktancí přívodního drátu. Rezistor  $R_{GS}$  pak slouží k vybití vnitřního náboje tranzistoru při jeho vypnutí. Jelikož čtečka bude napájena ze zdroje o napětí 12V a každá dioda má vstupní napětí 3,2V, je nutné zapojit LED diody do dvou paralelních větví, jako je to znázorněno na obr. 5.5. Na napájecím regulátoru je poté třeba nastavit výstupní napětí kolem 6,4V a proudové omezení na 700mA. Vstup, který je na obr. 5.5 označený jako GPIO pak připojíme na jeden z řídicích pinů GPIO sběrnice.

Ve zdrojovém kódu je pro tento vstup definován pin č.11. Výstup VCC připojíme na + napájecího step-down regulátoru a výstup GND propojíme s - napájecího step-down regulátoru a také s jednou se zemí na GPIO sběrnici.

## 5.6 Zvuková signalizace

Pro zvukovou signalizaci bude použit bzučák se vstupním napětím 3,3 až 5V. Je možné vybírat ze dvou druhů bzučáků – aktivní a pasivní. Aby pasivní bzučák vydával zvuk, je třeba na jeho vstup přivést obdélníkový průběh napětí. Výsledný tón pak bude závislý na frekvenci přivedeného obdélníkového napětí. Pro účely čtečky ale postačí bzučák aktivní, který již má oscilátor generující obdélníkové napětí zabudovaný v sobě. Aby bzučák vydával zvuk stačí tak na jeho vstup přivést pouze stejnosměrné napětí. Nevýhodou aktivního bzučáku oproti pasivnímu je, že výsledný tón je závislý na frekvenci vnitřního oscilátoru a nelze tak vstupním signálem změnit.

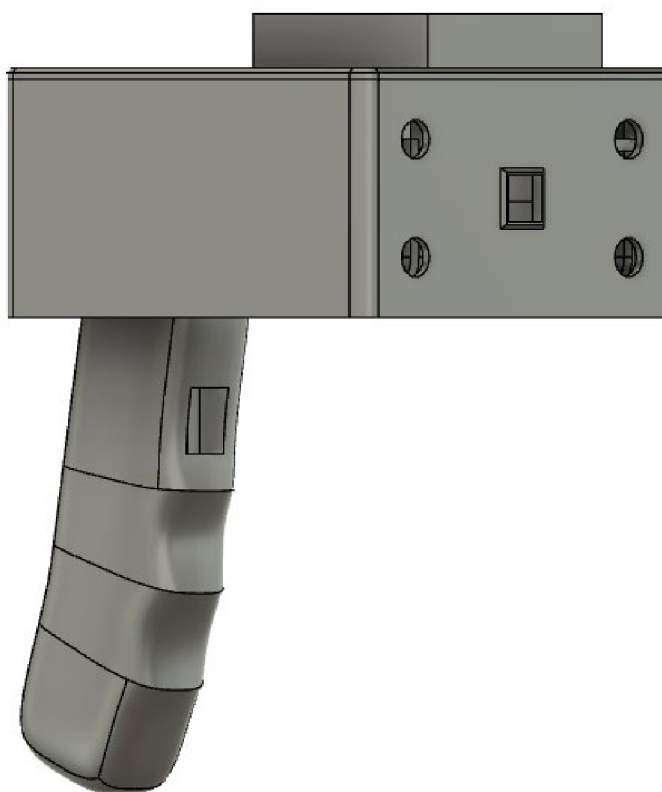


Obr. 5.5 Schéma zapojení tranzistoru s LED diodami



## 6 NÁVRH MECHANIKY MODULU

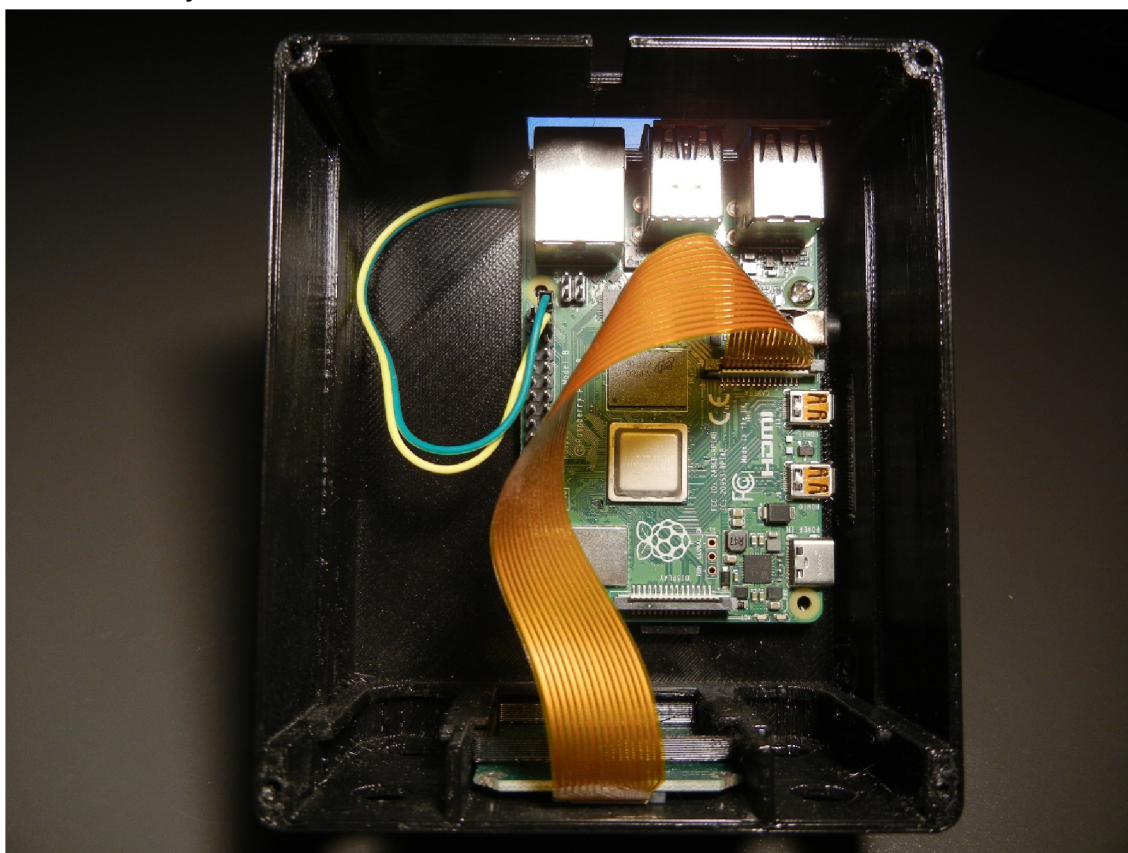
Nyní když máme všechny součástky pohromadě je potřeba je uložit do nějaké úhledné a funkční krabičky. Za tímto účelem byla v programu Fusion 360 vymodelována sestava krabičky, rukojeti a víka. Sestava je znázorněna na obr. 6.1. Krabička s rukojetí bude vytištěna na FDM 3D tiskárně z materiálu PET-G, který zajišťuje dostatečnou pevnost a také tepelnou odolnost výtisků.



Obr. 6.1 Návrh mechaniky čtečky - pohled zepředu

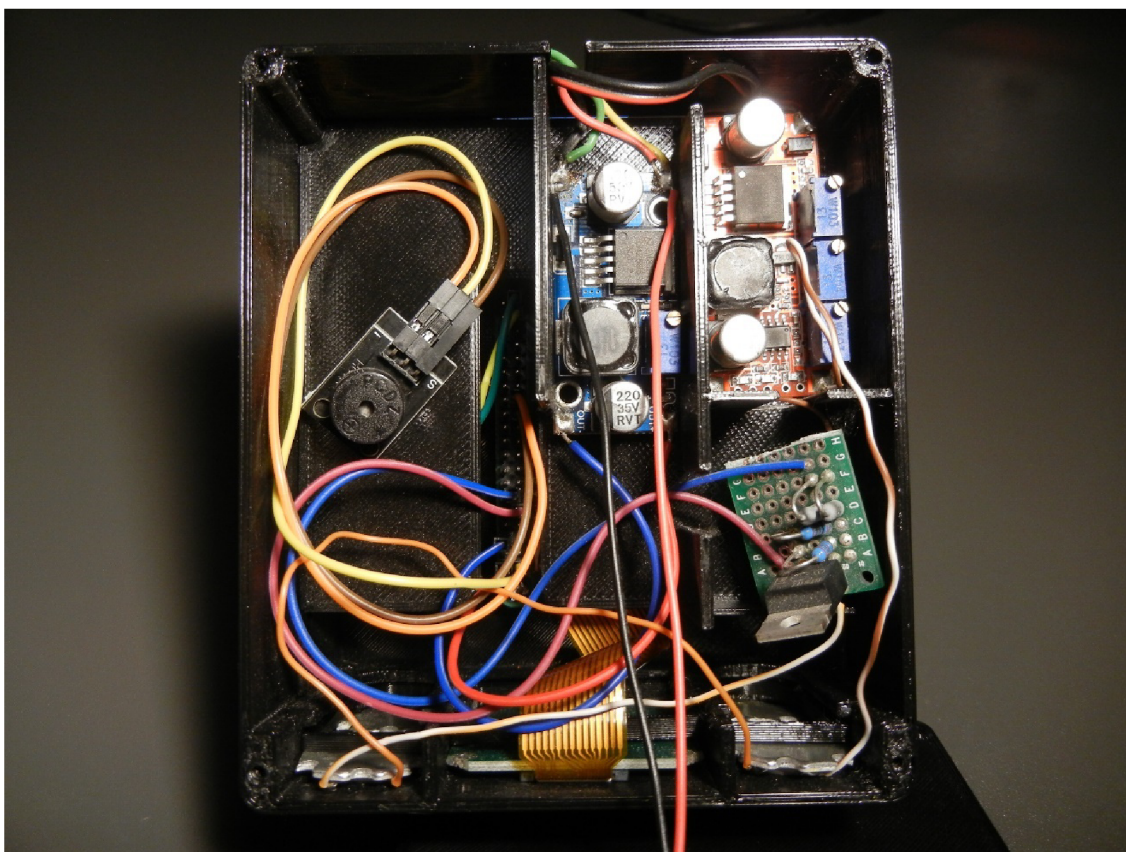
Sestava se skládá z rukojeti s výřezem pro tlačítko, které bude plnit funkci spouště samotné čtečky, a krabičky rozdělené na dvě patra pro uložení počítače Raspberry Pi a ostatní elektroniky. Rukojeť a krabička pro Raspberry Pi jsou spojeny dvěma vruty se zapuštěnou hlavou o rozměrech 4x12 mm. Přívodní vodiče od tlačítka zasazeného do rukojeti prochází tunelem uvnitř rukojeti a jsou vyvedeny do vnitřního prostoru krabičky. Na obr. 6.2 můžeme vidět již upevněný počítač Raspberry Pi s připojenými konektory tlačítka spouště. Jeden vodič vedoucí od tlačítka připojíme na jeden z ovladatelných pinů GPIO sběrnice. Ve zdrojovém kódu je pro tento vývod definován pin č. 37. Druhý vodič vedoucí od tlačítka připojíme na jednu ze zemí na GPIO sběrnici. Dále zde můžeme vidět kamerový modul, který je

vsunut do výřezu a upevněn vymežovací podložkou. Raspberry Pi je ke krabičce připevněno šrouby o rozměrech M2,6x6 mm, které jsou zašroubovány do otvorů k tomu určených.



**Obr. 6.2 Spodní patro modulu s připevněným Raspberry Pi**

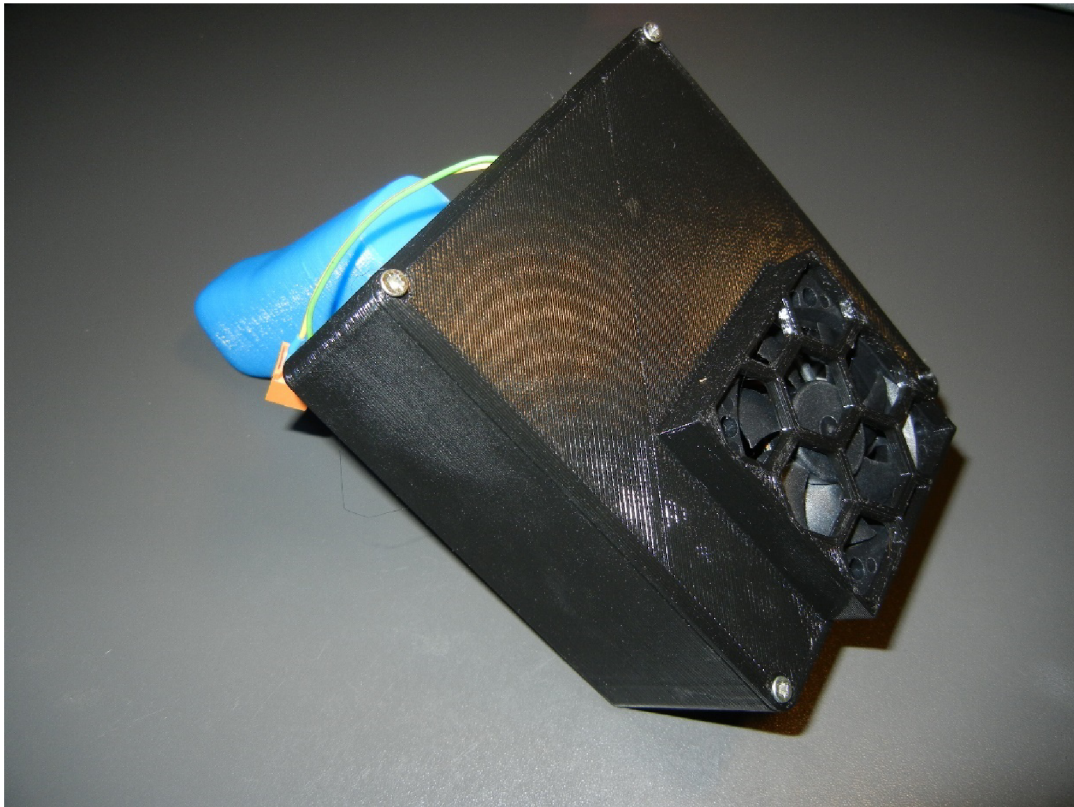
Na obr. 6.3 můžeme vidět již rozmístěnou a zapojenou doplňující elektroniku k Raspberry Pi. V horní části obrázku jsou umístěny oba step-down regulátory propojeny paralelně a konektor pro napájení síťovým adaptérem je vyveden zadní částí krabičky ven. V pravé dolní části krabičky je umístěn obvod s tranzistorem pro spínání LED diod, který je připojený ke step-down regulátoru s proudovým omezením a ke GPIO sběrnici. LED diody jsou stejně jako kamerový modul vsunuty do otvorů a zajištěny vymežovacími podložkami. V levé části je umístěn signalizační bzučák a vedle něj pak můžeme vidět výřez pro přístup ke sběrnici GPIO. Bzučák má tři vstupy. Dva pro napájení a jeden pro řídicí signál. Vstupní napětí bzučáku je 3,3 až 5V, je tedy možné jej na GPIO sběrnici připojit jak na 3,3V pin tak na 5V pin. Vstup pro řídicí signál připojíme na jeden z ovladatelných pinů na GPIO sběrnici. Ve zdrojovém kódu je k tomuto účelu definován pin č. 23.



**Obr. 6.3 Horní patro krabičky s rozmístěnou elektronikou**

Jako poslední krok zbývá vyřešit chlazení. Tento krok sice není úplně nezbytný, nicméně ničemu tak neuškodíme. Raspberry Pi 4 má oproti předchozím verzím nejvyšší výkon, ale také se oproti předchozím verzím nejvíce zahřívá. Umístěním Raspberry Pi 4 do téměř uzavřeného prostoru by mohlo docházet k přehřívání a procesor by snížil svou frekvenci, čímž by se snížil jeho výkon. Tento jev se nazývá thermal throttling. Aby se tomuto jevu předešlo byla na víko krabičky vymodelována kapsa pro ventilátor o rozměrech 50x50x10mm. Studený vzduch tak bude vstupovat otvorem ve víku a teplý vzduch bude vystupovat výřezy kolem konektorů Raspberry Pi v dolní části krabičky. Víko je připevněno vruty o rozměrech 2,5x12mm. Použitý ventilátor bude mít vstupní napětí 12 V, aby jej bylo možné připojit přímo k výstupnímu napětí použitého síťového adaptéru.





**Obr. 6.4 Víko krabičky s kapsou pro ventilátor**

## 7 OBSLUŽNÁ APLIKACE

Před vytvořením obslužné aplikace je třeba definovat co od ní očekáváme. Nejprve potřebujeme stisknutím tlačítka pořídit snímek požadovaného 1D nebo 2D čárového kódu a z tohoto snímku následně získat informaci uloženou v čárovém kódu. V případě, že se povede informaci z čárového kódu přečíst, oznámíme tuto událost zvukovým signálem a informaci následně zobrazíme a odešleme do PLC. Pro tyto funkcionality je potřeba následně vytvořit grafické rozhraní a webovou aplikaci. Abychom ale byli schopni tyto úlohy provést, musíme se nejdříve seznámit s knihovny, které nám tyto úkony umožní.

### 7.1 Knihovny pro detekci a čtení čárových kódů

Mezi nejpoužívanější knihovny pro detekci čárových kódů v jazyce Python patří knihovny pyzbar, pylibdmtx a zxing.

Knihovna pyzbar je určena pro čtení 1D čárových kódů a navíc podporuje ještě QR kódy. Podporované typy čárových kódů jsou: EAN-13/UPC-A, UPC-E, EAN-8, Code 128, Code 39, Interleaved 2 of 5, Codabar a QR kódy. Při testování této knihovny se projevila především její rychlost, kdy programová smyčka, která zahrnovala snímání snímku a detekci čárového kódu touto knihovnou trvala necelou desetinu vteřiny.

Knihovna pylibdmtx podporuje pouze jeden typ 2D čárových kódů a to je Datamatrix (datamatic). Detekce 2D čárového kódu touto knihovnou může být opět velice rychlá pokud použijeme parametr timeout. Tento parametr přinutí program vyskočit z funkce po skončení nastaveného časového intervalu, jestliže nebylo možné v pořízeném snímku detekovat datamatici.

Knihovna zxing oproti předchozím dvěma knihovnám podporuje o několik typů čárových kódů více než předchozí dvě knihovny, ale nalezení a dekodování čárového kódu touto knihovnou trvalo i více než 1,5 vteřiny. Z tohoto důvodu se tato knihovna jeví jako nevhodná.

V tomto projektu tedy pro dekodování 1D/2D kódů budeme používat knihovny pyzbar a libdmtx, které podporují naprostou většinu běžně používaných typů kódů. Na obrázcích č. 7.1 a 7.2 pak můžeme vidět ukázky skenovaných 1D/2D kódů těmito knihovny. Čárové kódy jsou v obrázku ohraničeny bílým obdélníkem a nad nimi je pak uveden obsah daného kódu. Obsah datamatice na obrázku č. 7.2 se do snímku již nevešel, jelikož byl příliš dlouhý.



**Obr. 7.1 Ukázka čtení čárového kódu typu EAN13  
knihovnou pyzbar**



**Obr. 7.2 Ukázka čtení čárového kódu typu Datamatrix  
knihovnou pylibdmtx**

## 7.2 Odesílání dat do PLC

Pro odesílání dat získaných z čárových kódů využijeme protokoly TCP/IP. TCP/IP (Transmission Control Protocol over Internet Protocol) v sobě shromažďuje protokoly pro spojování jednotlivých počítačů (serverů) do vzájemně provázaných počítačových sítí. Podle tohoto standardního protokolu síťové vrstvy TCP/IP komunikuje i celý Internet.

Samotné TCP je protokolem transportní vrstvy, který se stará o převod převáděné zprávy na paket a zpět. IP je protokolem vrstvy síťové, která slouží pro směrování a adresování.

Architektura TCP/IP je založena na možnosti snadného připojení k síti nezávisle na technologii.

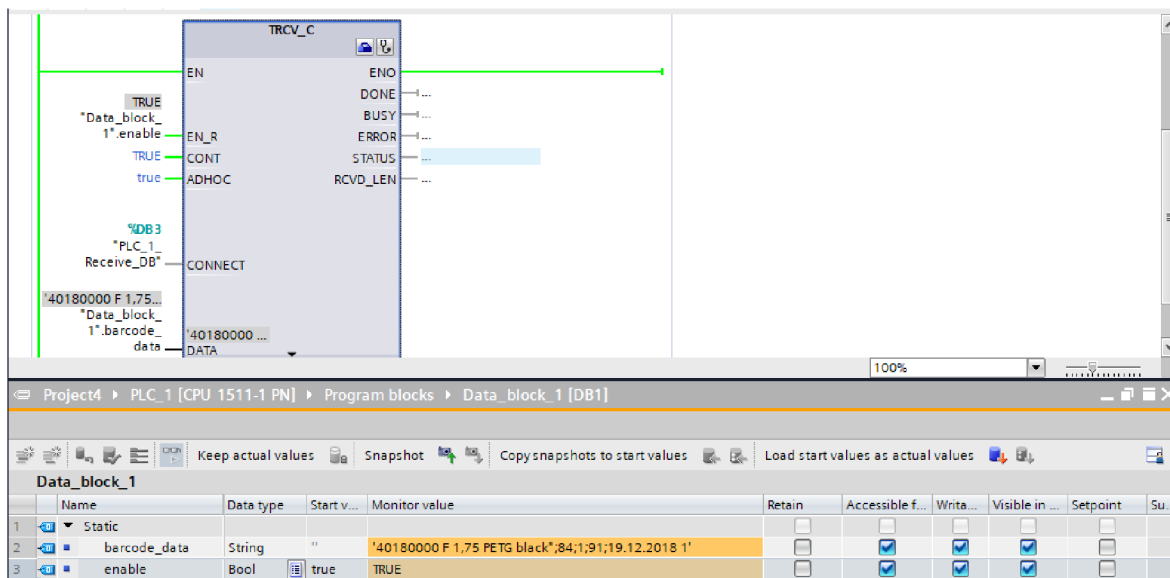
Důraz je zde kladen především na neobyčejně rychlý přenos cílových dat. Jedná z nevýhod je ale to, že data jsou nešifrovaná, čehož může snadno využít osoba hackera, který má nekalé úmysly.[17]

### 7.2.1 Knihovna socket

Výrazem socket označuje koncový bod připojení přes počítačovou síť. Socket je tvořen IP adresou a číslem portu. Komunikace probíhá mezi dvěma koncovými zařízeními (serverem a klientem). Aby se klient mohl připojit k serveru a odeslat zprávu, musí znát IP adresu serveru a port, na kterém server naslouchá. Pokud má klient tyto informace k dispozici, může se zkusit připojit k serveru. Připojením k serveru předává klient informace serveru o své IP adrese a portu. Server si poté vytvoří nový socket, do kterého vloží informace o IP adrese a portu klienta ale i informace o své IP adrese a portu. Na straně klienta se tento socket vytvoří také a komunikace pak může probíhat oběma směry.

Tento typ komunikace zajišťuje knihovna socket. Tato knihovna je součástí balíku Python a není ji tedy potřeba instalovat. Odesílání dat do PLC bylo otestováno v programu TIA Portal V14 na PLC simulovaném programem PLCSIM Advanced V3.0. Přijímání dat probíhalo za pomoci jednoduše nastaveného bloku TRCV\_C.

Na obrázku 7.3 můžeme v horní části vidět blok TRCV\_C který obstarává příjem dat odesílaných ze čtečky prostřednictvím knihovny socket. V dolní části pak můžeme vidět proměnnou barcode\_data, která obsahuje přijatá data.



Obr. 7.3 Ukázka přijímání dat v PLC

## 7.3 Knihovna Tkinter

Tkinter je knihovna jazyku Python, která slouží pro navrhování a tvorbu grafických uživatelských rozhraní. Obsahuje sadu widgetů (komponent), díky kterým je programátor schopen vytvářet aplikace běžící v oknech. Pojmem widgety jsou označovány základní grafické prvky, jako jsou tlačítka, popisky, rámečky, plátna, zaškrtačovací políčka, textová pole atd.

Grafická rozhraní vytvořena knihovnou Tkinter nevynikají futuristickým vzhledem, na druhou stranu Tkinter vyniká svou jednoduchostí a snadnou implementací. Navíc je tato knihovna již obsažena v balíčku jazyka Python, takže ji není třeba instalovat.

Při vývoji aplikací v tomto prostředí je ale potřeba mít na mysli, že grafické prvky se musejí neustále obnovovat, a tedy hlavní vlákno programu nesmí vstoupit do žádné nekonečné smyčky. Pokud by se tak stalo, aplikace by vyhodila výjimku. Tento problém tedy bylo nutné vyřešit za pomoci vláknového programování a knihovny threading.



## 7.4 Výsledná podoba obslužné aplikace

Na obrázku 7.4 můžeme vidět výslednou podobu obslužné aplikace vytvořené pomocí knihovny Tkinter. Čtení 1D/2D čárových kódů může probíhat ve dvou režimech, „Single mode“ a „Freerun mode“. Tyto režimy je možné přepínat tlačítky vlevo nahoře v liště s ovládacími prvky a je nutné si předtím, než začneme skenovat jeden zvolit. Mezi režimy je samozřejmě možné kdykoli přepínat.

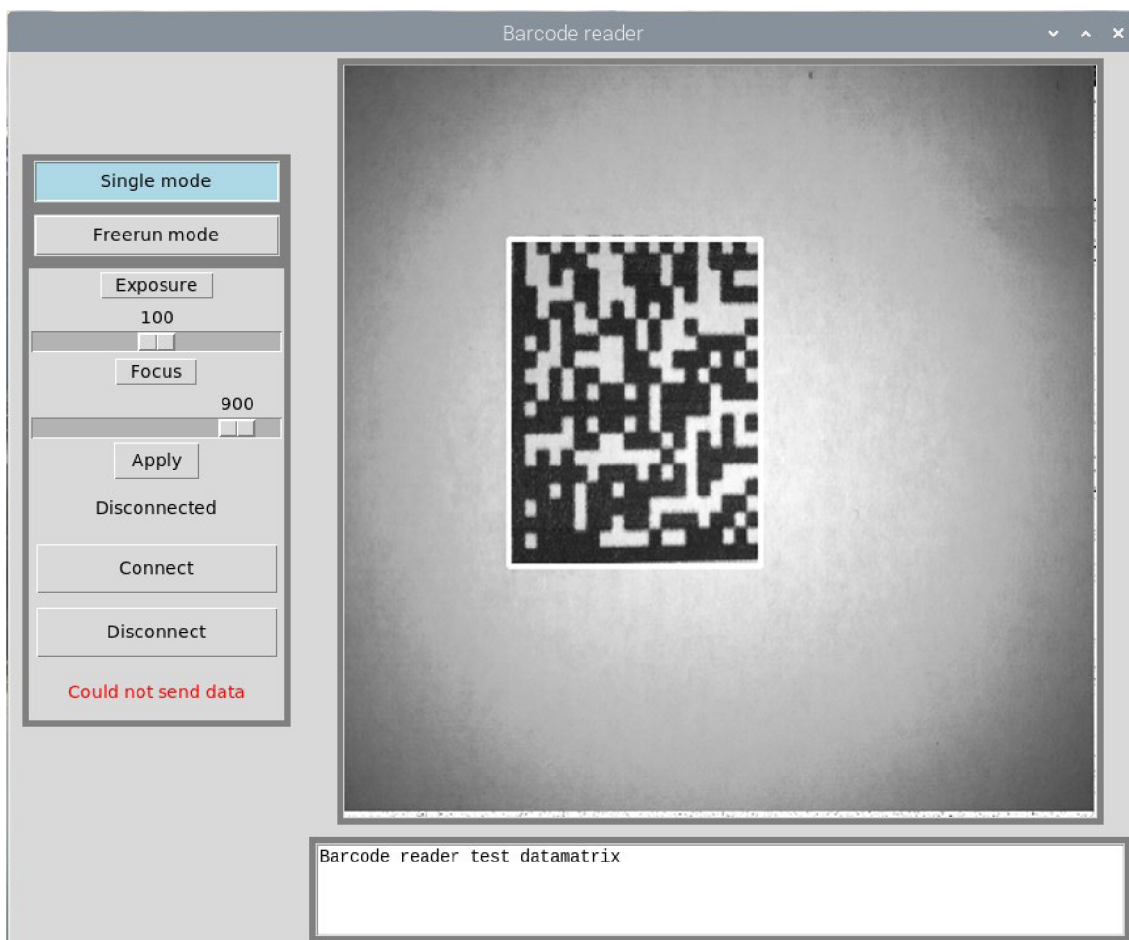
Pod tlačítka pro výběr režimu se nacházejí dva slidery pro nastavení expozičního času a ostření. Potvrzení hodnot se následně provádí tlačítkem „Apply“. Tyto slidery jsou defaultně nastaveny na optimální hodnoty pro snímání standardních velikostí 1D/2D čárových kódů a není je potřeba měnit. Jejich změna by byla nutná pouze v případě, kdy bychom chtěli číst čárové kódy nadstandardních velikostí z větší vzdálenosti.

Dále se v ovládací liště nacházejí tlačítka „Connect“ a „Disconnect“. Tyto tlačítka slouží pro připojení a odpojení od serveru, na který se budou odesílat data získaná čtením čárových kódů. Jestliže je čtečka k serveru připojena, bude nad tlačítky zobrazen nápis „Connected“. Naopak pokud čtečka není připojena, bude zde zobrazen nápis „Disconnected“. Pokud se data z nějakého důvodu nepodařilo odeslat, zobrazí se pod tlačítky červený nápis „Could not send data“. Adresu a port serveru je nutné nastavit přepsání proměnných „self.address“ a „self.port“, které se nachází v horní části souboru se zdrojovým kódem „barcode\_reader\_GUI.py“.

Napravo od panelu s ovládacími prvky můžeme vidět panel, ve kterém se po detekci čárového kódu bude zobrazovat snímek s tímto čárovým kódem. Detekovaný čárový kód bude vždy ohraničen obdélníkem. Po každém skenování, při kterém bude nalezen čárový kód, bude snímek automaticky nahrazen tímto novým snímkem.

Poslední prvek v tomto grafickém uživatelském rozhraní je pole, ve kterém se budou zobrazovat data obsažená ve snímaném čárovém kódu. Také toto pole se bude automaticky obnovovat se snímanými čárovými kódy.

Aplikaci je možné ukončit standardně křížkem v pravém horním rohu. Tento křížek ale také zabezpečuje korektní ukončení aplikace. Po jeho zmáčknutí se tedy okno nejen že uzavře, ale také se spustí funkce, která vypne kamerový modul a uvolní alokovanou paměť.



Obr. 7.4 Výsledná podoba obslužné aplikace

### 7.4.1 Single mode a Freerun mode

Jak již bylo zmíněno, aplikace může pracovat ve dvou režimech, „Single mode“ a „Freerun mode“.

Pokud je aktivován režim „Single mode“, čtečka čeká na stisknutí tlačítka. Po stisknutí tlačítka se rozsvítí LED diody, čtečka začne pořizovat snímky a hledat v nich čárové kódy. Jakmile čtečka čárový kód nalezne, LED diody se zhasnou, čtečka přestane pořizovat snímky, v grafickém uživatelském rozhraní se zobrazí snímek s nalezeným čárovým kódem společně s daty získanými z čárového kódu, zazní zvukový signál a pokud je to možné, odešlou se data z čárového kódu do PLC. Jestliže čtečka žádný čárový kód do pěti vteřin od stisknutí tlačítka nenalezne, proces snímání se sám ukončí a čtečka čeká na další stisknutí tlačítka.

Režim „Freerun mode“ je stejně jako „Single mode“ aktivován tlačítkem. Po jeho stisknutí se stejně jako u režimu „Single mode“ rozsvítí LED diody a čtečka začne pořizovat snímky čárových kódů. Rozdíl oproti režimu „Single mode“ je v tom, že po nalezení čárového kódu se proces snímání neukončí, ale čtečka pokračuje

v hledání dalšího čárového kódu. Každý rozpoznáný čárový kód i jeho obsah se zobrazí v grafickém uživatelském rozhraní, tato skutečnost se oznámí zvukovým signálem a data se odešlou do PLC. Proces snímání se ukončí až po opětovném stisknutí tlačítka. Aby nedocházelo k několikanásobnému přečtení jednoho a téhož čárového kódu, čtečka si průběžně ukládá do paměti pět naposledy přečtených kódů. Pokud tedy čtečka narazí na jeden z těchto kódů uložených v její paměti, kód ignoruje.

## 7.5 Tvorba webové aplikace

Kromě tvorby grafického uživatelského rozhraní je součástí zadání také tvorba webové aplikace. K tvorbě této webové aplikace byl použit webový framework Flask, jazyk HTML a na některé funkcionality byl použit i JavaScript.

### 7.5.1 Flask

Mezi nejznámější webové frameworky pro jazyk Python patří Django, Flask nebo Pyramid. Zvolil jsem tedy webový framework Flask, jelikož je nejrychlejší na pochopení, a tudíž je vhodný pro začátečníky. I přesto je ale možné v tomto frameworku vytvářet komplexní webové aplikace.

### 7.5.2 JavaScript

JavaScript je objektově orientovaný programovací jazyk, využívaný při tvorbě webových stránek. Na rozdíl od serverových programovacích jazyků (například PHP) sloužících ke generování kódu samotné stránky, JavaScript běží na straně klienta, tedy v prohlížeči, až po stažení do vašeho počítače.

JavaScript se používá především pro vytváření interaktivních webových stránek. Příkladem použití mohou být nejrůznější kontroly správného vyplnění formulářů, obrázky měnící se po přejetí myší, rozbalovací menu atd.[18]

V této webové aplikaci byl JavaScript využit především pro funkcionality některých tlačítek, sliderů a vypisování dat či zpráv pro uživatele.

## 7.6 Výsledná podoba webové aplikace

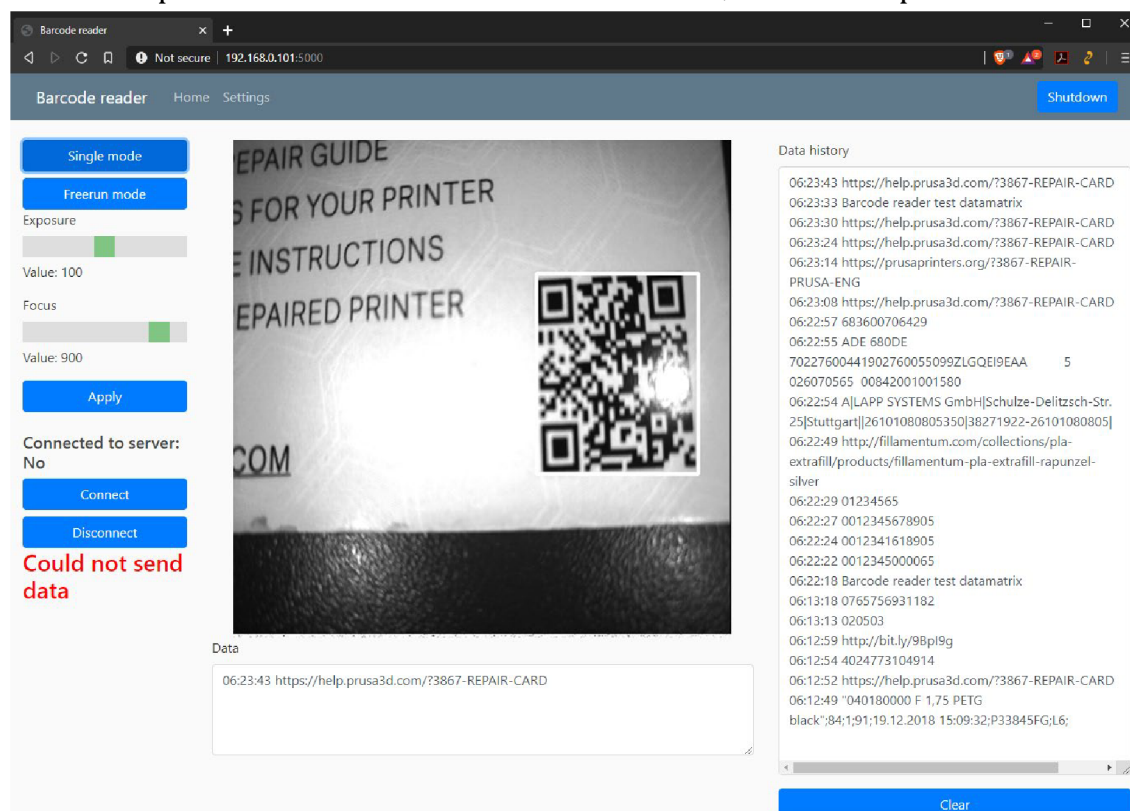
Na obrázcích 7.5 a 7.6 můžeme vidět finální podobu webové aplikace. Tato webová aplikace funguje téměř identicky jako již výše popsána obslužná aplikace, pouze je rozšířena o několik dalších funkcionalit. V této webové aplikaci je nyní možné přepínat mezi dvěma záložkami, „Home“ a „Settings“.

Záložka „Home“ má podobné rozložení jako grafické rozhraní obslužné aplikace. Opět je zde v levé části panel s ovládacími prvky, kde je možné přepínat

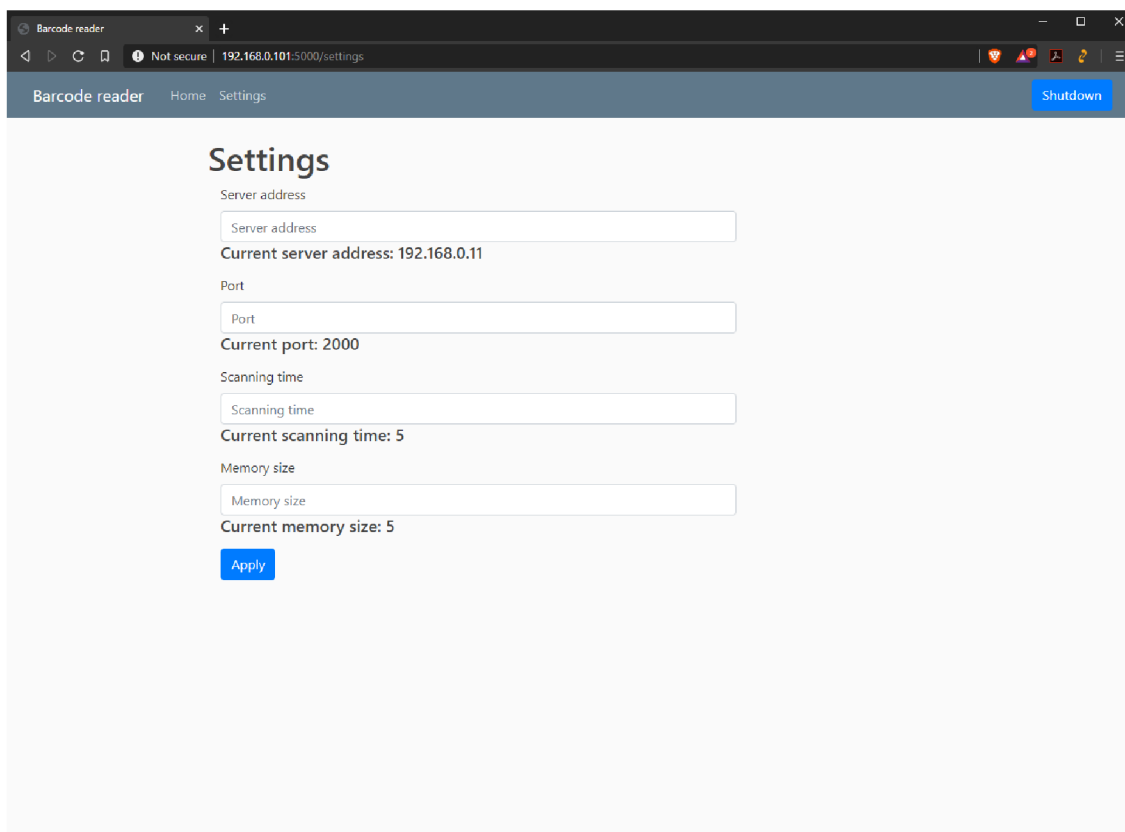
mezi režimy „Single mode“ a „Freerun mode“, nastavovat hodnoty expozičního času a ostření nebo se připojit, či odpojit od PLC. Ve středu stránky se nachází okno, kde se zobrazují snímky čárových kódů. Snímek v tomto okně se po každém úspěšně přečteném čárovém kódu automaticky aktualizuje bez toho, aniž by se aktualizovala celá stránka. Pod tímto oknem se opět nachází pole, ve kterém se zobrazují data z přečtených čárových kódů. Nově se ale u těchto dat vypíše i čas, kdy byl čárový kód přečten. Na pravé straně stránky se nachází okno, do kterého se vypisuje historie přečtených čárových kódů. Nové čárové kódy se do tohoto okna vkládají vždy seshora. Nejnovější čárový kód se tedy nachází vždy nahoře. Pod tímto oknem se pak nachází tlačítko „Clear“ pro smazání této historie.

V záložce „Settings“ je možné měnit některá nastavení, jako je IP adresa, port PLC, délka časového intervalu při skenování v režimu „Single mode“ nebo počet čárových kódů, které si čtečka pamatuje v režimu „Freerun mode“. Pokud chceme některou z těchto hodnot změnit, stačí zadat novou hodnotu do příslušného pole a stisknout tlačítko „Apply“. Všechny tyto hodnoty se ukládají do dokumentu typu JSON a při každém startu se z tohoto dokumentu zase načítají. Zvolenou hodnotu tedy stačí změnit pouze jednou a aplikace si ji při každém dalším startu bude pamatovat, dokud tuto hodnotu nepřepíšeme hodnotou novou.

V pravém horním rohu se pak nachází tlačítko „Shutdown“, které slouží ke korektnímu ukončení aplikace. Po jeho stisknutí se vypne kamera, uvolní se alokovaná paměť a ukončí se běh webového serveru, na kterém aplikace běží.



Obr. 7.5 Záložka Home webové aplikace



Obr. 7.6 Záložka Settings webové aplikace

## 7.7 Instalace knihoven

Aby bylo možné obě výše popsané aplikace spustit, je nezbytné do operačního systému Raspbian nainstalovat knihovny použité ve zdrojových kódech těchto aplikací. Bývá dobrým zvykem tyto dodatečné knihovny instalovat do virtuálního prostředí. To nainstalujeme a nastavíme příkazy:

```
pip3 install virtualenv
mkvirtualenv barcode_reader
workon barcode_reader
```

Pokud se předchozí příkazy provedly správně, nacházíme se nyní ve virtuálním prostředí s názvem `barcode_reader`. Nyní můžeme nainstalovat potřebné knihovny. Názvy těchto knihoven se nachází v souboru `requirements.txt` v příloze. Jejich instalaci provedeme příkazem:

```
pip3 install -r requirements.txt
```

Po zadání tohoto příkazu by se měly knihovny automaticky stáhnout a nainstalovat. Bohužel po nedávném přechodu ze staré verze Python 2 na Python 3 nebyla ještě knihovna `v4l2` uzpůsobena pro Python 3. Je tedy potřeba kód souboru této knihovny lehce pozměnit. Upravená verze souboru se nachází v příloze pod názvem `v4l2.py` a

stačí tímto upraveným souborem přepsat starý soubor v adresáři s knihovnamy námi vytvořeného virtuálního prostředí.

Další problém se nachází v knihovně `pylibdmtx`, ve které se vyskytuje chyba, která znemožňuje použití parametru `timeout`. Opravený soubor se taktéž nachází v příloze pod názvem `wrapper.py` a opět stačí pokud tímto souborem přepíšeme původní soubor v adresáři s knihovnamy námi vytvořeného virtuálního prostředí.

Jako poslední zbývá nainstalovat knihovnu `OpenCV`. Tu je možné také nainstalovat nástrojem `pip`. Tento způsob instalace se ale nedoporučuje, jelikož se ne vždy povede. Je tedy nutné tuto knihovnu zkompileovat a nainstalovat ručně. Návod k instalaci je možné vyhledat na internetu, kde je těchto návodů spousta.

## 7.8 Uvedení kamery do provozu

Pro správnou funkci kamerového modulu `Arducam 16Mpx IMX298 MIPI Color Camera Module` je třeba nejprve provést několik změn v nastavení. Nejprve je nutné povolit použití kamery a `i2c` sběrnice v nastavení `Raspberry Pi`. Do tohoto nastavení se dostaneme příkazem:

```
sudo raspi-config
```

V záložce „`Interfacing Options`“ pak povolíme použití kamery a `i2c` sběrnice.

V druhém kroku musíme zvětšit velikost paměti dostupné pro GPU jednotku. To provedeme přepsáním hodnoty `gpu_mem` v souboru `config.txt` do kterého se dostaneme příkazem:

```
nano /boot/config.txt
```

V tomto souboru pak najdeme řádek, na kterém je uvedeno `gpu_mem=xx` a přepíšeme jej na `gpu_mem=180`. Nyní je možné kameru používat.

## 7.9 Spuštění aplikací

Pokud máme nainstalovány všechny potřebné knihovny, můžeme aplikace spustit. Aby to ale bylo možné, musíme se nacházet v námi dříve vytvořeném virtuálním prostředí, do kterého jsme knihovny nainstalovali. Do tohoto prostředí vstoupíme příkazem:

```
workon barcode_reader
```

Obslužnou aplikaci s grafickým uživatelským rozhraním pak spustíme příkazem:

```
python3 barcode_reader_GUI.py
```

Pro spuštění webové aplikace musíme nejdříve nastavit proměnnou prostředí `FLASK_APP`, to provedeme příkazem:

```
export FLASK_APP=flask_barcode_reader.py
```

Webovou aplikaci pak spustíme příkazem:

```
flask run --host=0.0.0.0
```

K webové aplikaci se pak můžeme z počítače, který se nachází ve stejné síti jako čtečka, připojit zadáním IP adresy čtečky a portu 5000 do prohlížeče, např.  
192.168.0.192:5000

## 8 ZÁVĚR

Cílem této práce bylo vytvoření aplikace na platformě Raspberry Pi, která by byla schopna číst 1D/2D kódy, obsah těchto kódů následně zobrazit a odeslat do PLC. Pro tuto aplikaci bylo dále nutné vytvořit grafické rozhraní a také webovou aplikaci. Před naprogramováním aplikace bylo ale nejprve nutné navrhnout mechaniku modulu a také vybrat vhodný kamerový modul a osvětlení.

Návrh mechaniky čtečky se odvíjel od výběru samotného kamerového modulu a osvětlení. Jako kamerový modul byl zvolen Arducam 16Mpx IMX298 MIPI Color Camera Module, který oproti většině ostatních modulů určených pro platformu Raspberry Pi disponuje vyšším rozlišením a vyšší snímkovací frekvencí, která může výrazně ovlivňovat rychlost čtení čárových kódů. Mezi další výhody této kamery patří programovatelné ostření. Většina ostatních kamer určených pro Raspberry Pi disponuje pouze ostřením manuálním nebo pevně nastaveným.

Při volbě osvětlení bylo nutné dbát na dostatečnou intenzitu světla, aby bylo možné využívat kratších expozičních časů pro dostatečně ostré snímky nerozmazané pohybem jak samotné čtečky, tak také pohybem samotného výrobku či čárového kódu. Zároveň bylo ale potřeba, aby osvětlení bylo dostatečně kompaktní. Pro tento účel tedy bylo vhodné použít čtyři 1W výkonové LED diody.

V kapitole č. 6 je popsána mechanika modulu čtečky, která byla navržena tak, aby do modulu bylo možné umístit všechny potřebné elektronické součástky a modul byl zároveň dostatečně kompaktní. Pro modul byl navržen i systém chlazení, který zabezpečuje bezproblémový dlouhodobý provoz čtečky.

Nyní bylo možné vytvořit samotnou obslužnou aplikaci čtečky. Výsledná aplikace je schopna číst 1D čárové kódy typu EAN-13/UPC-A, UPC-E, EAN-8, Code 128, Code 39, Interleaved 2 of 5, Codabar a 2D čárové kódy typu QR code a Datamatrix. Tento výčet typů čárových kódů zahrnuje většinu běžně používaných typů 1D/2D čárových kódů. Firma TE Connectivity kladla důraz především na čárové kódy typu Datamatrix, jelikož z 90% používají právě tento typ kódů. Aplikace dále dokáže obsah přečteného čárového kódu zobrazit prostřednictvím grafického nebo webového rozhraní a následně obsah čárového kódu odeslat do PLC.

Rychlost skenování čárových kódů je výrazně závislá na cviku operátora čtečky. Pokud je čtečka umístěna ve správné vzdálenosti a celý snímaný čárový kód se nachází v zorném poli čtečky, je možné kód přečíst již z prvního, nebo druhého snímku. Pořízení a analýza jednoho snímku trvá přibližně 2-3 desetin sekundy. To znamená, že přečtení jednoho čárového kódu je možné přibližně za 3-5 desetin sekundy za předpokladu, že se skenovaný čárový kód nachází ve snímku celý a je dostatečně zaostřen. Při testování čtečky bylo s trochou cviku možné tohoto výsledku dosáhnout ve většině případů.



Na konci kapitoly 7 je pak uveden postup pro instalaci knihoven potřebných pro spuštění. Je zde také uveden postup pro uvedení kamery do provozu a také postup pro spuštění samotného grafického nebo webového rozhraní.

Původním plánem zadání bylo vytvoření nepřenosného modulu, který by měl pevné napájení a který by pouze ověřil vhodnost použitého hardware a software pro čtení 1D/2D kódů. Po konzultaci se školitelem a konzultantem jsem navrhnul, aby tato čtečka byla pojata jako přenosné zařízení. Nad rámec zadání tedy bylo nutné vyřešit napájení, mechaniku a ergonomii celého modulu.

# Literatura

- [1] Strojové vidění I: Principy a charakteristiky. *Automa*. 2008, **2008**(1), 42.
- [2] Strojové vidění II: Úlohy, nástroje a algoritmy. *Automa*. 2008, **2008**(2), 54-55.
- [3] GREŠ, Tomáš. DÍL 1: ŘÁDKOVÉ KAMERY – TYPY A TECHNOLOGIE. *Ate system* [online]. [cit. 2019-01-03]. Dostupné z: <http://kamery.atesystem.cz/know-how/line-scan-velky-pruvodce-radkovymi-kamerami/dil-1-radkove-kamery-typy-a-technologie/>
- [4] A Practical Guide to Machine Vision Lighting. *NATIONAL INSTRUMENTS* [online]. 30 Ledna 2017 [cit. 2019-01-03]. Dostupné z: <http://www.ni.com/white-paper/6901/en/>
- [5] Osvětlovače pro strojové vidění. *W-Technika* [online]. [cit. 2019-01-03]. Dostupné z: <http://www.prumyslova-kamera.cz/content/15-osvetlovace>
- [6] ŠEVČÍK, Zbyněk. Objektiv – oko fotoaparátu – díl první. *DtPko* [online]. 27. ledna 2012 [cit. 2019-01-03]. Dostupné z: <http://www.dtpko.cz/clanky/objektiv-oko-fotoaparatu-dil-prvni/>
- [7] Objektivy pro strojové vidění. *W-Technika* [online]. [cit. 2019-01-03]. Dostupné z: <http://www.prumyslova-kamera.cz/content/21-objektivy>
- [8] Čárové kódy (teorie). *Gaben* [online]. [cit. 2019-01-03]. Dostupné z: <http://www.gaben.cz/cz/faq/carove-kody-teorie>
- [9] Rozdíl mezi 1D (typicky EAN) a 2D (typicky QR) čárovým kódem. In: *Pdaflores.cz* [online]. [cit. 2020-06-03]. Dostupné z: <https://pdaflores.cz/cs/blog/rozdil-mezi-1d-typicky-ean-a-2d-typicky-qr-carovym-kodem-b14.html>
- [10] Rozdíl mezi 1D a 2D čárovými kódy. *Www.prumyslove-tablety.cz* [online]. [cit. 2020-06-03]. Dostupné z: <http://www.prumyslove-tablety.cz/1d-vs-2d-carove-kody/>
- [11] Raspberry Pi 4 Model B - 4GB RAM. In: *Rpishop.cz* [online]. [cit. 2020-06-03]. Dostupné z: <https://rpishop.cz/raspberry-pi-4b/1598-raspberry-pi-4-model-b-4gb-ram-765756931182.html>
- [12] Arducam IMX298 MIPI 16MP Color Camera Module for Raspberry Pi 4/3B+/3. In: *Www.arducam.com* [online]. [cit. 2020-06-01]. Dostupné z: <https://www.arducam.com/product/16mp-imx298-color-mipi-camera-module-raspberry-pi/>
- [13] Raspberry Pi. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2020 [cit. 2020-06-02]. Dostupné z: [https://cs.wikipedia.org/wiki/Raspberry\\_Pi#cite\\_note-2](https://cs.wikipedia.org/wiki/Raspberry_Pi#cite_note-2)

- [14] Simple Guide to the Raspberry Pi GPIO Header. In: *Www.raspberrypi-spy.co.uk* [online]. [cit. 2020-06-03]. Dostupné z: <https://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>
- [15] Python. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2020 [cit. 2020-06-03]. Dostupné z: <https://cs.wikipedia.org/wiki/Python>
- [16] Obecné informace o knihovně. In: *Is.mendelu.cz* [online]. [cit. 2020-06-03]. Dostupné z: [https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=25212](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=25212)
- [17] Co je TCP/IP. In: *Www.sprava-site.eu* [online]. [cit. 2020-06-04]. Dostupné z: <https://www.sprava-site.eu/tcp-ip/>
- [18] JavaScript. In: *Www.adaptic.cz* [online]. [cit. 2020-06-05]. Dostupné z: <https://www.adaptic.cz/znalosti/slovnicek/javascript/https://www.adaptic.cz/znalosti/slovnicek/javascript/>
- [19] NEVLÁČIL, Jakub. *Raspberry PI čtečka 1D/2D kódů*. Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/115166>. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Peter Honec.

# Seznam symbolů, veličin a zkratek

GPIO	-	General-purpose input/output
HDMI	-	High-Definition Multimedia Interface
LED	-	Light-Emitting Diode
TCP	-	Transmission Control Protocol
IP	-	Internet Protocol
PLC	-	Programmable Logic Controller
GUI	-	Graphical User Interface
HTML	-	Hypertext Markup Language
Pip	-	Pip installs packages

# Seznam příloh

Příloha 1 – Mechanika modulu

Příloha 2 – Upravené knihovny

Příloha 3 - Zdrojové kódy