



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

STROVÉ VIDĚNÍ PRO ROBOTIZOVANÉ VÝROBNÍ SYSTÉMY

MACHINE VISION FOR ROBOTIC MANUFACTURING SYSTEMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Strach

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Vetiška, Ph.D.

BRNO 2022

Zadání diplomové práce

Ústav: Ústav výrobních strojů, systémů a robotiky
Student: **Bc. Petr Strach**
Studijní program: Strojní inženýrství
Studijní obor: Výrobní stroje, systémy a roboty
Vedoucí práce: **Ing. Jan Vetiška, Ph.D.**
Akademický rok: 2021/22

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Strojové vidění pro robotizované výrobní systémy

Stručná charakteristika problematiky úkolu:

Současná doba klade velký důraz na 100% kontrolu produktů. Vizuální kontrola je základem při kontrole vyrobených produktů. Provádění vizuální kontroly je velmi jednodušná a únavná činnost, kterou je možné nahradit strojovým viděním. Práce se zaměřuje na vizuální kontrolu produktů a navádění robotu pro automatickou manipulaci s kontrolovanými produkty.

Cíle diplomové práce:

Rešerše dané problematiky
Tvorba modelové úlohy
Systémový rozbor
Reálné sestavení a testování modelové úlohy

Seznam doporučené literatury:

SICILIANO, Bruno a Oussama. KHATIB. Springer handbook of robotics. Berlin: Springer, c2008. ISBN 978-3-540-23957-4
HORNBERG, Alexander. Handbook of Machine Vision. 1. Weinheim: WILEY-VCH Verlag GmbH & Co. KGaA, 2006. ISBN 978-3-527-40584-8
NOF, S. Y. Springer Handbook of Automation. Springer, 2009. 1812 s. ISBN 978-3-540-78830-0

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.

ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.

děkan fakulty

ABSTRAKT

Diplomová práce se zabývá strojovým viděním ve spojitosti s robotizovanými výrobními systémy. Teoretická část se věnuje hardwaru strojového vidění, principy a charakteristikami strojového vidění, typickými úlohami ve spojitosti se strojovým viděním, naváděním robotu pomocí kamery, logickými úlohami pro zpracování obrazu v programu TwinCAT. Praktická část se zaměřuje na návrh robotického pracoviště s kamerou pro kontrolu natočení součásti. Toto pracoviště bylo nejprve zprovozněno virtuálně a poté i otestováno na reálném pracovišti.

ABSTRACT

The diploma thesis deals with machine vision in connection with robotic production systems. The theoretical part deals with machine vision hardware, principles and characteristics of machine vision, typical tasks in connection with machine vision, robot guidance using a camera, logical tasks for image processing in the TwinCAT program. The practical part focuses on the design of a robotic workplace with a camera to control the rotation of the part. This workplace was first put into operation virtually and then tested at a real workplace.

KLÍČOVÁ SLOVA

Strojové vidění, robot, navádění robotu, vizuální kontrola, kontrola kvality, průmyslové kamery, zpracování obrazu, automatizace.

KEY WORDS

Machine vision, robot, robot guidance, visual inspection, quality control, industrial cameras, image processing, automation.

BIBLIOGRAFICKÁ CITACE

STRACH, Petr. *Strojové vidění pro robotizované výrobní systémy*. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/139457>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Jan Vetiška.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že jsem diplomovou práci na téma návrh zařízení pro značení bužírek vypracoval samostatně s použitím odborné literatury a pramenů, uvedených v seznamu, který tvoří přílohu této práce.

V Brně dne 20.5.2022

.....
Bc. Petr Strach

PODĚKOVÁNÍ

Touto cestou bych rád poděkoval panu Ing. Janovi Vetiškovi Ph.D., za cenné rady a připomínky během vypracování a za poskytnutí hardwaru k otestování stroje. Dále bych chtěl poděkovat rodině a přítelkyni za podporu během studia.

OBSAH

1	ÚVOD.....	17
2	Motivace	18
3	Hardware systému strojového vidění	19
	3.1 Osvětlovací soustava.....	19
	3.1.1 Analýza vlastností snímaného objektu.....	20
	3.1.2 Stanovení potřebných vlastností použitého světla	20
	3.1.3 Stanovení vhodné geometrie osvětlení.....	21
	3.1.4 Výběr vhodného typu osvětlovače	23
	3.1.5 Provozní režimy osvětlení.....	26
	3.2 Kamera.....	26
	3.2.1 Typ obrazu.....	27
	3.2.2 Typ detektoru	28
	3.2.3 Černobílá a barevná kamera	29
	3.3 Objektiv	30
	3.3.1 Důležité parametry pro volbu objektivu.....	31
	3.3.2 Typy objektivů	32
	3.3.3 Vady objektivů	34
4	Princip a charakteristika strojového vidění	36
	4.1 Co je strojové vidění	36
	4.2 Princip strojového vidění	36
	4.3 Kategorie systémů strojového vidění.....	37
	4.3.1 Kamerový senzor (<i>vision sensor</i>).....	37
	4.3.2 Inteligentní kamera (<i>smart camera</i>).....	37
	4.3.3 PC/PLC systémy (<i>PC/PLC vision</i>).....	38
	4.3.4 Zákaznické systémy (<i>custom vision</i>).....	38
5	Typické úlohy pro strojové vidění v průmyslové praxi.....	39
	5.1 Nalezení povrchových vad.....	39
	5.2 Nalezení polohy a navádění robotu.....	39
	5.3 Měření a kontrola tolerancí.....	39
	5.4 Identifikace barev	39
	5.5 Čtení kódů a textů	39
	5.6 Bin picking.....	40
6	Robot a strojové vidění.....	41
	6.1 Systém strojového vidění OnRobotEyes	41
	6.2 Sjednocení systému souřadnic robotu a obrazu	41
7	Logické řízení a strojové vidění v jednotném prostředí TwinCAT Vision	42
	7.1 Architektura	42
	7.2 Zpracování obrazu v PLC	43
8	Komunikační protokoly	44
	8.1 OPC server	44
	8.2 EtherNET/IP	45

8.3	EtherCAT Device protocol.....	45
8.4	EtherCAT Automation protocol.....	45
8.5	PROFINET.....	46
9	Představení praktické části projekt.....	47
9.1	Měřená součást.....	47
9.2	Systémový rozbor.....	47
10	Virtuální zprovoznění pracoviště.....	49
10.1	AUTODESK Inventor Professional.....	49
10.1.1	Držák pro kameru.....	49
10.1.2	Měřicí stanice.....	49
10.1.3	Vedení pro kostku.....	50
10.2	Robot Studio.....	51
10.2.1	Pracovní objekty a body manipulace.....	51
10.2.2	Signály robotu.....	51
10.2.3	Deklarování proměnných.....	52
10.2.4	Popis programu MAIN.....	52
10.3	TwinCAT PLC.....	53
10.3.1	Program MAIN a prostředí HMI.....	53
10.3.2	Program POU_Automat.....	56
10.3.3	Program POU_Servis.....	56
10.3.4	Program POU_Errors.....	56
10.4	TwinCAT Vision.....	57
10.4.1	Postup přípravy snímků pro virtuální zprovoznění.....	57
10.4.2	Nastavení File Source.....	58
10.4.3	Adresování.....	58
10.4.4	Popis programu POU_Kamera.....	59
10.5	Vytvoření OPC serveru v projektu.....	61
10.5.1	OPC UA na straně TwinCAT.....	61
10.5.2	Připojení k OPC UA pomocí RobotStudia.....	62
10.5.3	Kontrola komunikace virtuálního prostředí.....	63
11	Reálné zprovoznění.....	64
11.1	Hardware navržené robotické linky.....	64
11.1.1	Optika.....	64
11.1.2	Součásti pro manipulaci s měřenou součástí.....	66
11.2	Postup reálného zprovoznění pracoviště.....	67
11.2.1	Připojení kamery k PLC.....	67
11.2.2	Připojení I/O karty k PLC.....	68
11.2.3	Robot.....	70
12	ZHODNOCENÍ A DISKUZE.....	71
13	ZÁVĚR.....	72
14	SEZNAM POUŽITÝCH ZDROJŮ.....	73
15	SEZNAM ZKRATEK.....	76
16	SEZNAM OBRÁZKŮ.....	77
17	SEZNAM TABULEK.....	79

18	SEZNAM PŘÍLOH	80
19	PŘÍLOHY	81

1 ÚVOD

Diplomová práce je zaměřena na strojové vidění v robotizovaných výrobních pracovištích. Nejprve se věnuje hardwaru, kde jsou charakterizovány jednotlivé výhody a nevýhody jednotlivých typů komponent. Následuje popis a principy strojového vidění, rozdělení do jednotlivých skupin kategorií a každá z těchto skupin byla charakterizována. Další část je věnována úlohám strojového vidění jako takového. Jsou zde popsány jednotlivé operace, které se v praxi nejčastěji používají, na které navazují možnosti propojení souřadného systému kamery a průmyslového robotu. Rešerše pokračuje základními informacemi o možnostech vytvoření řízení strojového vidění v prostředí TwinCAT, kde je nejprve popsáno, pro jaké operace je tento program vhodný, poté objasněna základní architektura prostředí a charakterizovány možnosti zpracování obrazu, jako vytvoření algoritmů, nebo použití na vstup obrázky uložené přímo v počítači anebo možnosti připojení kamery. Následuje část vybraných komunikačních protokolů, které se v praxi nejčastěji používají s popisem, jakým směrem se snaží postupovat. Je zde i charakterizováno pár vybraných, jako například OPC server, který se použije poté i ve virtuálním zprovoznění. Poslední část rešerše se věnuje představení zadání praktické části, ve které se nachází zadání i systémový rozbor, ve kterém je zamýšlí nad pracoviště z hlediska optického komunikační problému a možnosti odebírání komponent z pásu.

Práce pokračuje návrhem virtuálního pracoviště s kamerovým systémem pro kontrolu otvoru v měřené komponentě. Tato část začíná návrhem jednotlivých komponent, které bylo potřeba k vytisknout na 3D tiskárně. Následovalo vytvoření virtuálního modelu v prostředí Robot studio, kde jsou charakterizovány použité proměnné a signály pro komunikaci s PLC a ovládání gripperu a popsán algoritmus, který robot řídí. V předposlední části je vytvoření řídicího programu pro PLC, který se skládá z pěti podprogramů. Hlavním z nich je program MAIN, který ovládá HMI a volá i ostatní programy. Každá z těchto částí programů je podrobněji popsána, Začíná se částí pro popis jednotlivých obrazovek pro obsluhu, následuje program POU_Automat, který řídí POU_Automat, který řídí automatický cyklus stroje, POU_Servis, který má dvě části, pro nastavení hodnot pro kameru a druhá pro nastartování a spuštění motorů a algoritmu robotu, POU_Errors a poslední nejdůležitější program POU_Kamera, který je věnován zprovoznění visionu, ve kterém je napsán kód pro vyčtení snímku a jeho vyhodnocení. Poslední částí virtuálního zprovoznění je navázání komunikace mezi použitými programy pomocí OPC serveru.

Na závěr této práce je popsáno, jak probíhalo testování programů na reálné sestavě, použité komponenty pro připojení kamery a její nastavení, připojení I/O karty, export programu z Robot Studia a nahrát do robotu, zaměření přesných pozic operace a samotné testování úlohy.

2 Motivace

V dnešní době důraz na kontrolu čím dál tím větší a postupem času se stále zvyšuje. Základním prvkem je vizuální kontrola, měření rozměrů a podobné. To je však velice jednotvárná a únavná činnost, která vede k unavení obsluhy a její následné nepozornosti vedoucí k provedení chyby. Takovou práci je možné nahradit strojovým viděním. Strojové vidění je levné velmi dobře dostupné a tím pádem má i rychlou finanční návratnost. Lze jej udělat variabilním provedení a může tak jedna kamera provádět vizuální kontroly několika typů produktů.

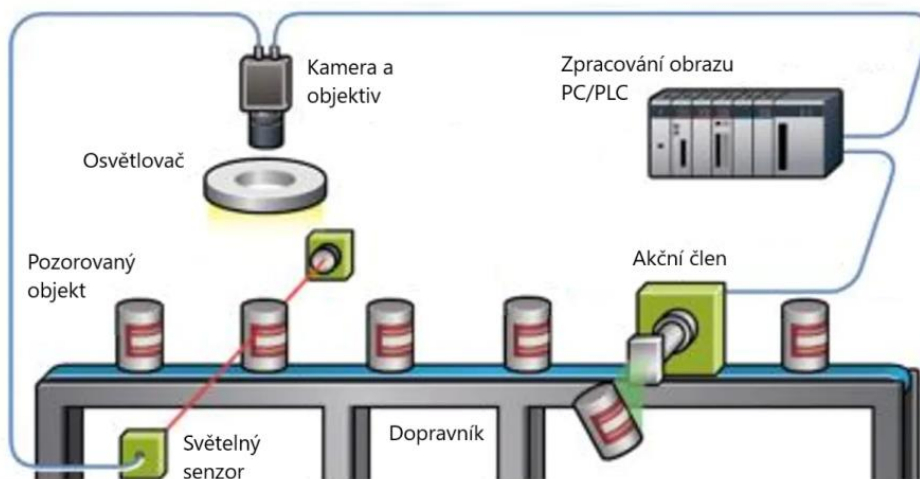
3 Hardware systému strojového vidění

Kapitola je zaměřena na základní hardware strojového vidění, popisu vlastností a doporučení pro jakou aplikaci jsou vhodné.

Zpracování obrazu v počítači vzniklo v sedmdesátých letech minulého století. Již v této době byla výpočetní technika na úrovni, která umožňovala zpracovávat velkého objemu dat, které nesou obrazovou informaci. Vznikl tak nový obor s názvem *computer vision* neboli počítačové vidění. Dnes označuje obecně systémy, které automaticky pracují na základě zpracování obrazů získaných z kamery. Předmětem počítačového vidění může být v základu prakticky cokoliv, například lidské oči, postava, dopravní situace, a jiné. Mezi předměty počítačového vidění patří také proces výroby, pro který je dnes používán *machine vision* neboli strojové vidění. [1]

Strojové vidění je v dnešní době používáno mimo jiné i v průmyslové automatizaci. V praxi se využívá stále častěji, a to především pro bezkontaktní a velice rychlou kontrolu kvality výrobků, počítání objektů, ke kontrole sestav či čtení textů nebo nejrůznějších kódů uvedených na výrobcích.

Na Obrázek 1 níže je vyobrazen příklad klasického použití strojového vidění pro kontrolu kvality výrobků. Pozorované objekty se pohybují po dopravníkovém páse. Jakmile jsou snímány světelným senzorem, kamerový systém (složený z kamery, objektivu a osvětlení) vytvoří snímek. Vytvořený snímek je poté odeslán do řídicího systému (PC/PLC) ke zpracování na základě předdefinovaných algoritmů a dochází k vyhodnocení, zda je OK/NOK. Pomocí akčního členu je pak případně odstraněn z pracovního procesu.



Obrázek 1) Příklad strojového vidění [2]

3.1 Osvětlovací soustava

Osvětlovače jsou stejně důležitou součástí strojového vidění jako kamera a inspekční software. Z určitého hlediska a nadsázkou můžeme říct, že strojové vidění je v podstatě měření kontrastu světla. Úkolem osvětlení je snaha dosáhnout co největšího kontrastu částí objektu, na které se zaměřujeme (zájmová část), oproti těm, které nejsou pro měření zajímavé (rušivé části). V neposlední řadě je také důležité, aby byla minimalizována změna vlivu v okolí na reprezentaci objektu jasovým obrazem kamery, aby rozhodující program vykazoval opakovatelné neměnné výsledky. [1]

Kontrastu mezi zájmovou a rušivou částí lze docílit dvěma způsoby. Buď využijeme jejich rozdílné absorpce, nebo rozdílu jasu, který vznikne vhodně zvoleným směrovým osvětlením. Oba způsoby začínají návrhem osvětlovací soustavy analýzou ovlivnění objektu a světla. Z výsledků této analýzy lze určit jak a čím snímáný objekt osvětlit a jak eliminovat případné rušivé vlivy. Celý postup lze shrnout do několika bodů a to:

analýza vlastností snímaného objektu vzhledem k úloze strojového vidění,
 analýza potřebných vlastností použitého světla,
 stanovení vhodné geometrie osvětlení,
 výběr vhodného osvětlovače,
 eliminace rušivých vlivů. [2]

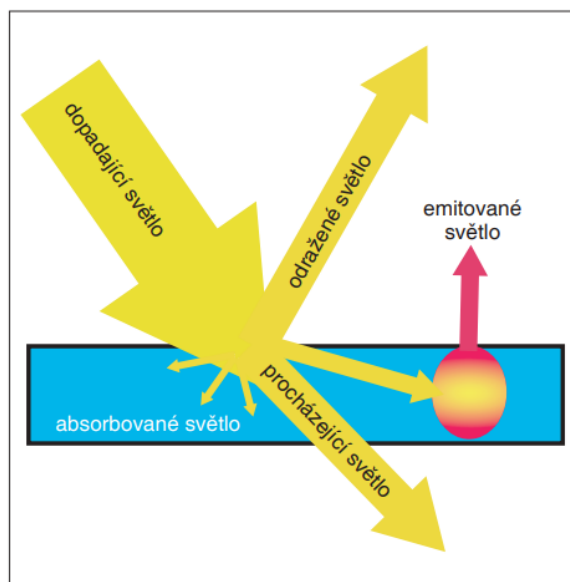
3.1.1 Analýza vlastností snímaného objektu

Analýza vlastností snímaného objektu vzhledem k úloze strojového vidění se zaměřuje především na tyto vlastnosti objektu:

- velikost absorpce nebo odrazivosti světla v zájmových místech objektu, která určuje kontrasty a lesky (souvisí rovněž s barvou objektu),
- struktura povrchu v zájmových místech objektu, kterou je někdy nutné zvýraznit, jindy potlačit,
- členitost objektu (výstupky, otvory atd.), vyžadující různé způsoby vytvoření či potlačení kontrastu v obraze,
- tvar a velikost objektu pro určení velikosti a vzdálenosti osvětlovače,
- průsvitnost a průhlednost částí objektu, které mohou být využity nebo musí být potlačeny,
- pozadí objektu, které může působit rušivě,
- možná změna polohy nebo případný pohyb objektu. [3]

3.1.2 Stanovení potřebných vlastností použitého světla

Pro určení nutných vlastností světla je potřeba analyzovat, jak se bude použité světlo chovat v interakci se snímaným objektem a jak bude poté reagovat na snímač obrazu kamery. Důležité působení světla na objekt. Světlo, které dopadá na sledovaný objekt se podle zákonů optiky rozdělí na část, která se odrazí, která projde a část se absorbuje (viz. Obrázek 2) Odráž světla). Ve strojním vidění může být využita každá z těchto částí. Interakce mezi objektem a světlem závisí na barvě objektu a vlnové délce světla, na které také i na citlivosti kamery. [3]



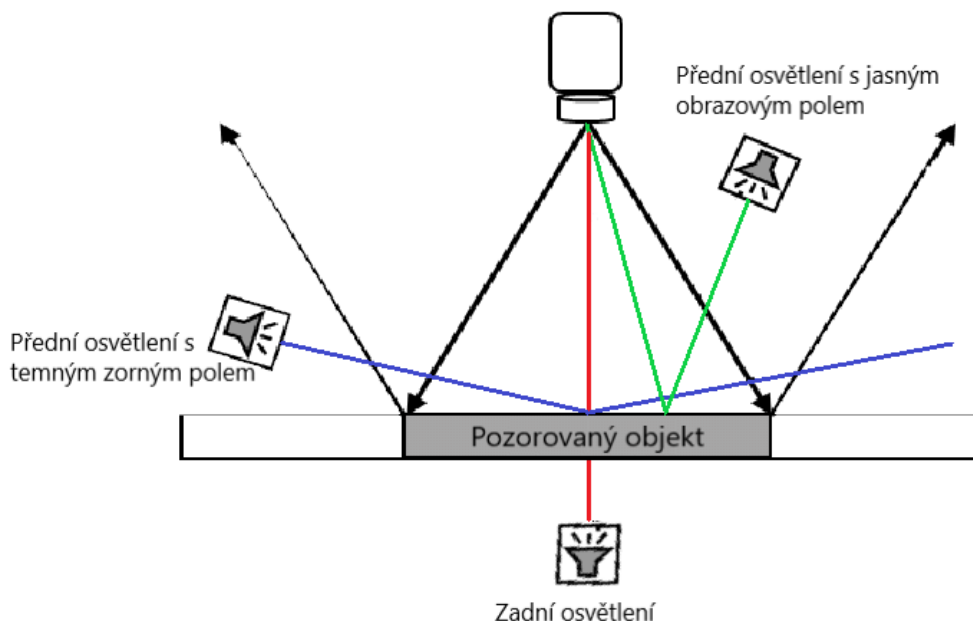
Obrázek 2) Odráž světla

Ve strojovém vidění se nejčastěji používá kombinace osvětlovače s červeným světlem a černobílou kamerou. Jejich kombinace je z hlediska spektrální citlivosti kamer výhodná, avšak v mnoha případech nemusí mít optimální výsledek, protože kontrasty na objektu se mohou výrazně lišit od kontrastu na bílém světle, viditelných i lidským okem. [3]

Pro realizaci zmíněných způsobů osvětlení je na trhu velký výběr osvětlovačů. V oboru strojového vidění se již výhradně využívají LED diody. Důvody je jejich poměrně vysoká účinnost, snadná regulovatelnost světelného toku, vysoká životnost, jsou vytvořeny pro nízké napětí a možnosti využití pulzního režimu. Další výhodou LED diod jsou rozměry, díky nimž lze snadno vytvářet osvětlení nejrůznějších typů, tvarů a velikostí. Široké spektrum využití je obrovskou výhodou pro integrátory, kteří mohou každou aplikaci řešit individuálně na míru potřeb snímání. [4]

3.1.3 Stanovení vhodné geometrie osvětlení

Vlivem různého způsobu vytvoření kontrastu mezi zájmovou a rušivou částí bylo zavedeno názvosloví pro určité geometrie osvětlení (viz také obrázek 3). Tyto typy jsou celkem tři, přední osvětlení s jasným zorným polem (*bright-field lighting*), přední osvětlení s tmavým zorným polem (*dark-field lighting*) a zadní světlo (*back lighting*).

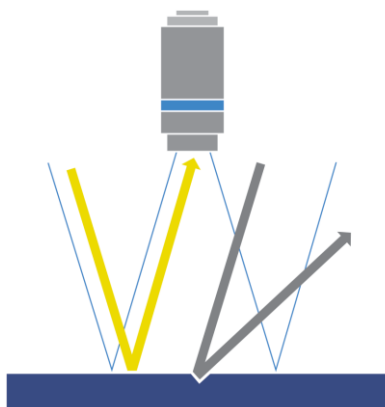


Obrázek 3) Geometrie osvětlení [5]

Geometrie osvětlení je dána složením svazku světla. Složení reálného svazku světla leží mezi dvěma extrémami, a to paprsky rovnoběžné, kdy se světlo pohybuje s rovnoběžnými paprsky a difuzní, kdy prochází paprsky rovnoměrně rozmístěny do okolí ve všech směrech. Proto reálný paprsek dopadá na snímání objektu pod určitým rozmezím úhlů. Obecně tak platí, že rozptýlené světlo zmenšuje kontrasty způsobené členitostí objektu a strukturou povrchu. Naopak směrové světlo se volí v moment, kdy potřebujeme-li kontrastně zobrazit členění objektu. [3]

Přední osvětlení s jasným obrazovým polem

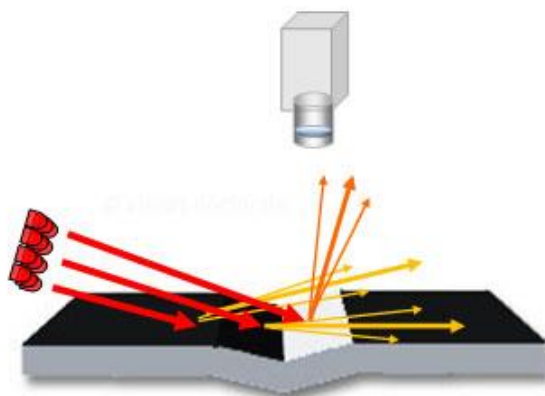
Jde o jednoduchý způsob, kdy osvětlovač osvětluje objekt tak, aby se světlo odrážející o pozorovaný objekt odráželo do objektivu na obrazový snímač. Osvětlení se většinou používá ve spojení s rozptýleným světlem pro vytvoření kontrastů na základě rozdílné absorpce. Méně často se používá směrové světlo, které je vhodné na lesklé povrchy. Vytváří tak kontrastní obraz na světlém poli. [3]



Obrázek 4) Přední osvětlení s jasným obrazovým polem [6]

Přední osvětlení s tmným zorným polem

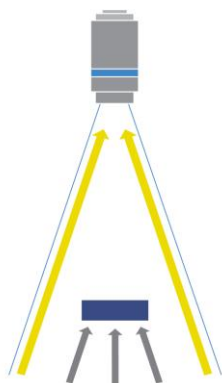
V naprosté většině případů využívá směrové světlo, aby členité části snímaného objektu odráželi světlo na objektiv snímače obrazu, zatímco rovné a lesklé části s paprsky světla odrazí mimo objektiv. Tímto způsobem tvoří obraz o vysokém kontrastu mezi členitostmi povrchu v tmném poli. [3]



Obrázek 5) Přední osvětlení s tmným zorným polem [7]

Zadní osvětlení

Zadní osvětlení se používá k zobrazení obrysů objektů a průchozích otvorů, typicky pro měření rozměrů. Používá se v případě, kdy je nutné vytvořit kontrastní obrys objektu před průhlednou lesklou přepážkou. Dominantně používá rozptýlené světlo. [3]



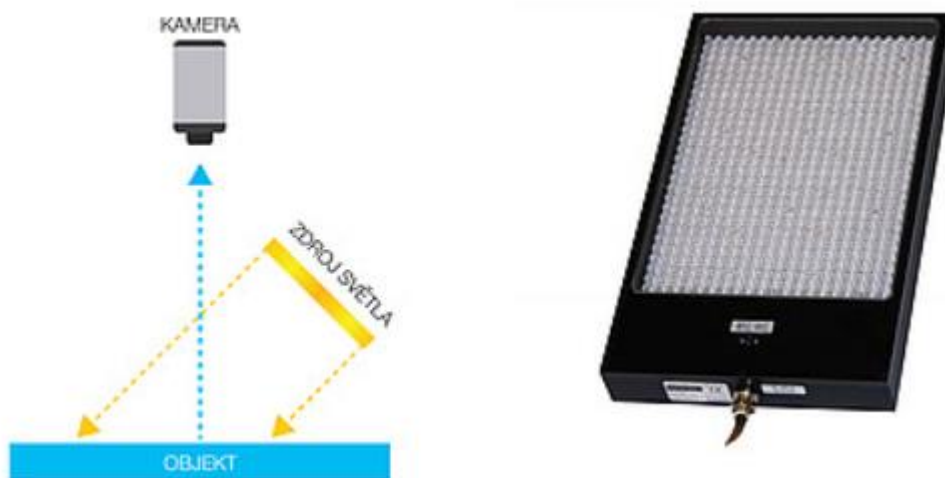
Obrázek 6) Zadní osvětlení [6]

3.1.4 Výběr vhodného typu osvětlovače

Pro realizaci výše zmíněných způsobů osvětlení je na trhu velký výběr osvětlovačů. V oboru strojového vidění se v dnešní době používají výhradně už jenom LED diody. Důvody proč se používají výhradně jsou jejich malé rozměry, poměrně vysoká účinnost, snadná regulovatelnost světelného toku, vysoká životnost, stačí si s nízkým napětím a možností využití pulzního režimu. Další výhodou LED diod je, že díky svým rozměrům lze snadno vytvářet osvětlení nejrůznějších typů, tvarů a velikostí. Tato možnost širokého spektra použití je obrovskou výhodou pro integrátory, kteří mohou poté každou aplikaci řešit individuálně na míru potřeb snímání. [4]

Plošná osvětlovací pole

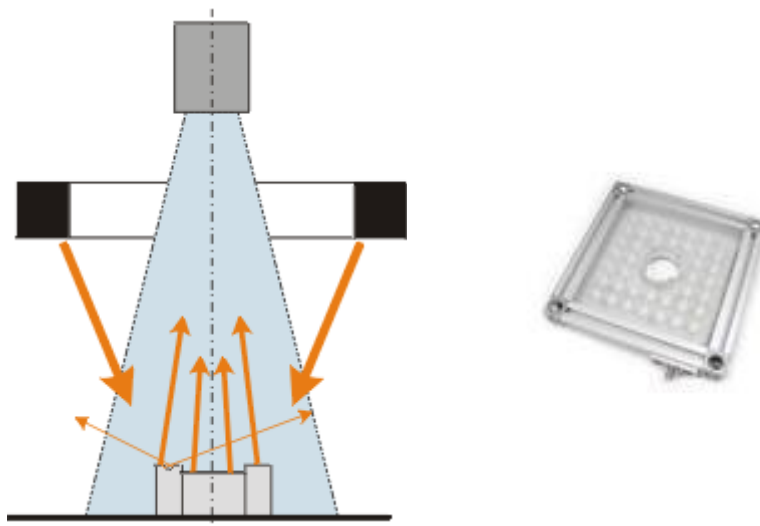
Plošná osvětlovací pole jsou znázorněna na Obrázek 7 níže. Většinou využívá směrového světla pro osvětlení s temným zorným polem. V náročnějších operacích mohou nahradit i nákladnější osvětlení. Menší a speciální pole se využívají jako pomocná světla, nebo jako zdroje pro složitější operace. [3]



Obrázek 7) Plošné osvětlení [8]

Kruhové osvětlovače

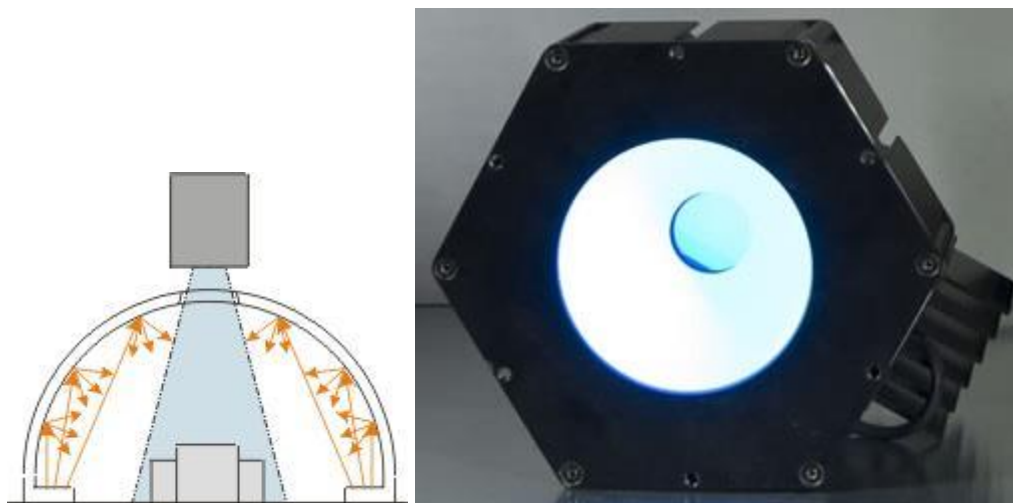
Kruhový osvětlovač vyzářuje více rozptýleného světla směřujícího od objektivu obrazového snímače. Prakticky jde o osvětlovač pro osvětlení s jasným obrazovým polem a rozptýleným světlem. Nemá však zcela ideální vlastnosti, obzvláště v provedení jednoduchého kruhu pomocí LED diod, které často výrobci umísťují okolo objektivu kamery. Příklad kruhového osvětlovače je znázorněn na Obrázek 8. [3]



Obrázek 8) Kruhové osvětlení [9]

Kopulové osvětlovače

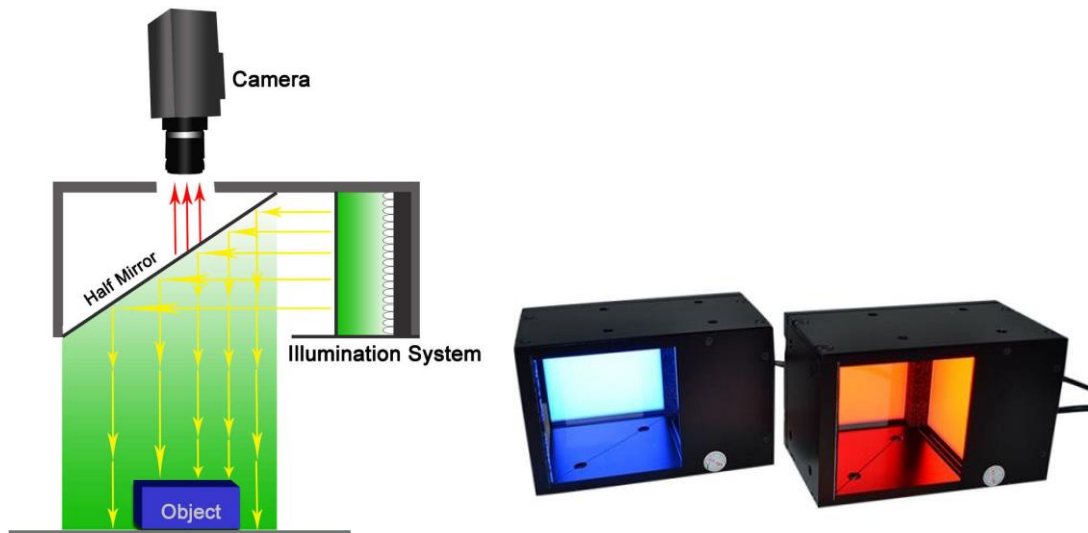
Kopulové osvětlovače (Obrázek 9) se používají pro osvětlení ze všech stran, jedná se totiž o osvětlení s jasným obrazovým polem, kdy je nezbytně nutné osvětlit objekt světlem skutečně rozptýleným. K takovému rozptýlení světla se tvoří difúzním odrazem na vnitřní ploše polokoule, vzniká tak osvětlení, kterému se přezdívá bez stínové osvětlení (*shadowless*). Typická úloha pro použití tohoto osvětlení je při čtení tištěných nápisů, čtení kódů a podobně. [3]



Obrázek 9) Kopulové osvětlení [9]

Osvětlovače DOAL

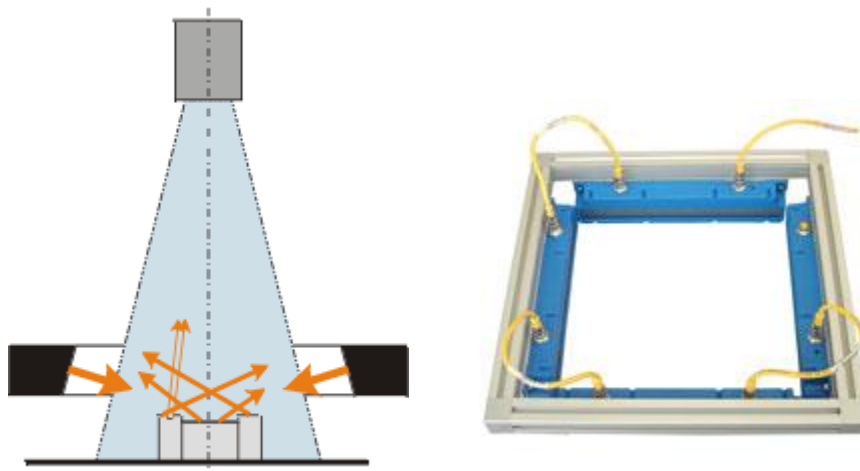
Osvětlovač DOAL (Obrázek 10), z angličtiny Diffused ON Axis Lighting poskytuje specifické osvětlení s jasným obrazovým polem a rozptýleným světlem, které je díky difuzoru rozptýleno ještě lépe. Díky konstrukci světlo dopadá rovnoměrně z celé plochy nad objektem, tedy i z prostoru objektivu kamery, a to pomocí polopropustného zrcadla. Nevýhodou je vyšší cena a omezené zorné pole. [3]



Obrázek 10) DOAL osvětlení [10]

Osvětlovače dark-field

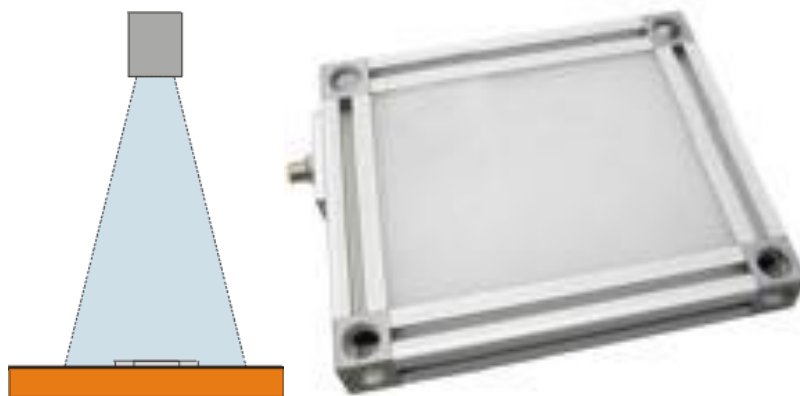
Jak už z názvu vyplývá, poskytují přední osvětlení s temným zorným polem, čehož je docíleno pomocí směrového světla, téměř kolmého k ose objektivu kamery. Díky této konstrukci je typickým využitím osvětlovače zdůraznění gravírovaných nebo laserem vypálených znaků či nápisů. Konstrukce tohoto osvětlovače je nejčastěji tvořena kruhem LED diod s úzkou vyzařovací charakteristikou, které svítí téměř v rovině do jeho středu. [3]



Obrázek 11) Osvětlení dark-field [9]

Osvětlovače backlight

Zadní rozptýlené světlo neboli backlight (obrázek 12), je konstruován jako pole led diod s difuzorem (předřazeným velkoplošným rozptylovačem). Jedná se o nejčastěji používaný typ zadního osvětlení, který se velmi často uplatňuje pro měření obrysu součásti. Důležité je vzít v úvahu, že dané měření vyžaduje zcela homogenní zadní světlo. Dalším častým příkladem je úloha, kdy se zajišťuje poloha nebo rozpoznání součásti v průhledném pouzdru. [3]



Obrázek 12) Osvětlení back light [9]

3.1.5 Provozní režimy osvětlení

Řízení osvětlovačů

V operacích strojového vidění nemusí být osvětlovač nijak řízen a může jen kontinuálně svítit. V jiných operacích však může být potřeba zapínat světlo pouze v okamžik expozice, například kvůli rušení okolních kamer, či životnosti osvětlení. Většina výrobců osvětlení pro strojní vidění dodává i řídicí systém k ovládání pomocí digitálních vstupů a výstupů. Několik nejčastějších typů režimů osvětlení je uvedeno níže. [11]

Pulzní provozní režim

V tomto případě se využívá pulzního napětí o nominální hodnotě. V aplikacích strojového vidění se daný způsob řízení používá spíše k prodloužení životnosti LED diod i přesto, že led diody mohou svítit i v kontinuálním režimu. Kontinuální řízení je ve většině případů nutné, aby světlo svítilo jen po dobu vytvoření snímku. V praxi to může například vypadat tak, že výrobek dojde na místo snímání, pomocí detektoru se určí přítomnost objektu a řídicí systém spustí světlo jen na určitou chvíli, než kamera vytvoří snímek, poté osvětlení znovu vypne. Výhodou je prodloužení životnosti, zmenšení spotřeby energie a snížení množství vytvořeného tepla. [11]

Strobvací provozní režim

Jedná se o režim velice podobný výše zmíněnému pulznímu, navíc však využívá krátkodobé přetížení LED diod, kterým je docíleno větší intenzity osvětlení. Díky tomu můžeme dosáhnout ekonomičtějšího osvětlení dané operace, za použití levnějšího osvětlení, které bude krátkodobě přetíženo. Je třeba, aby byl interval přetížení velmi krátký, v opačném případě by mohlo dojít k nenávratnému poškození. V úvahu musíme vzít také velikost tepla, která při přetížení vzniká. Systémy po přetížení svítí na nominální napětí nebo jsou úplně vypnuté, z důvodu ochlazení. [11]

Kontinuální provozní režim

Na rozdíl od výše uvedených způsobů řízení je kontinuální režim trvale sepnutý. Tento způsob je využíván zejména v operacích, kde je frekvence snímání velice častá až nepřetržitá, například při kontrole kvality kontinuálního materiálu. Pro tyto aplikace platí, že je objekt neustále přítomen, a proto světlo svítí trvale. [11]

3.2 Kamera

Kamera je pravděpodobně nejdůležitější částí strojového vidění. Jde o zařízení zaznamenávající obraz, který následně dokáže uložit či přenést. Světlo odražené od objektů je přeneseno pomocí objektivu a optického systému kamery na plochu světelného snímače, ten poté převádí informaci obsahující intenzitu a frekvenci.

Druhů kamer existuje mnoho, pro správný výběr je důležité vzít v potaz velké množství kritérií. Jde například o typ detektoru, snímač (černobílý či barevný), velikost, rozlišení čipu,

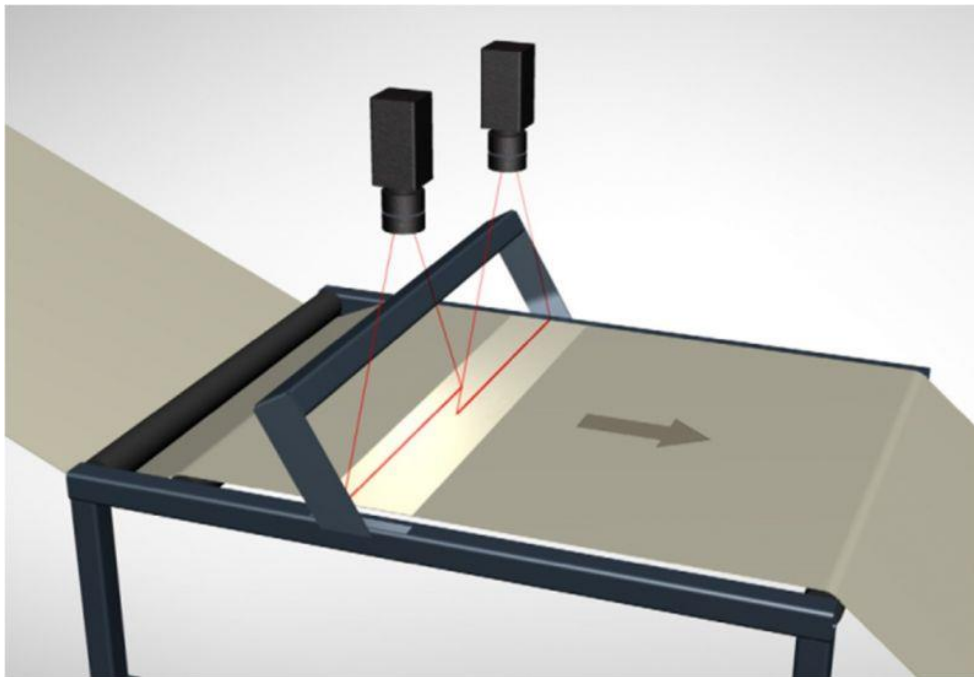
typ obrazu či připojení ke kameře. Připojení existuje dvojího typu, analogové a digitální rozhraní. Dnes se využívá výhradně digitální rozhraní, mezi něj patří Ethernet/IP, USB nebo Firmware. [12]

3.2.1 Typ obrazu

1D Kamerový systém – řádkové kamery

Tyto kamery pracují podobně jako kancelářský skener. Obsahují čipy s jedním nebo více řádky složených z obrazových bodů. Zatímco u plošných kamer se udává rozlišení jako horizontální krát vertikální počet pixelů, u řádkových kamer se uvádí pouze počet bodů v jedné řadě. Pro snímání celého objektu je třeba jím pohybovat vzhledem ke kameře. Řádkové kamery se tak s výhodou využívají v případech, kdy je potřeba snímat kontinuálně pohybující se materiál, či výrobky na běžícím pásu. Typickými úlohami pro použití řádkové kamery jsou kontrola kvality tisku, detekce kazů a vad materiálu odvíjející se z role, detekce otřepů ocelových lan, textilních vláken a jiné. [13]

Na rozdíl od plošných kamer mají řádkové kamery vyšší citlivost a dokážou snímat v řádů desítek či stovek kHz. Rozlišení kamer se pohybuje okolo 1kPx až 12kPx, je tak možné dosáhnout obrazu o extrémně vysokém rozlišení. Horizontální rozlišení je dáno vlastním rozlišením kamery, případně přednastavenou oblastí zájmu. Vertikální rozlišení závisí na nastavením počtu řádků, rychlosti pohybu a rychlosti expozice. S řízením kamery tak mohou vzniknout problémy, pokud lze nastavit konstantní rychlosti objektů, lze nastavit pevnou frekvenci snímání, pokud však konstantní rychlosti nedosáhneme, je možnost tuto frekvenci řídit pomocí digitálních vstupů. Toto snímání je však zpravidla velice obtížné na správné seřízení. [13]



Obrázek 13) řádkové kamery [14]

2D Kamerový systém – plošné kamery

Plošné kamery jsou nejvíce využívaným typem v praxi. Jedná se o systém, který na rozdíl od řádkových kamer, snímá obraz ve dvou osách (2D). Rozlišení těchto typů kamer je udáváno v horizontálním krát vertikálním počtu pixelů. Nevýhodou oproti řádkovým kamerám je, že nemají tak vysokou snímací frekvenci, na druhou stranu jsou méně komplikované. Plošné kamery jsou využívány např. při měření rozměrů součástí, detekci svaru, součástí apod

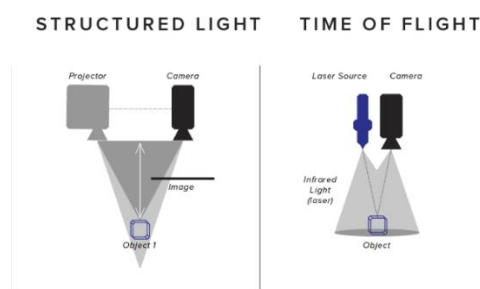


Obrázek 14) Plošné kamery [15]

2,5D kamerový systém – plošné kamery s měřením hloubky obrazu

2,5D kamera je schopna snímání 2D obrazu, ale také hloubky obrazu, dokáže detekovat předměty různé tloušťky, i naskládané na sebe. Pracovní vzdálenost detekce hloubky obrazu je určena minimální a maximální hranicí, kde je možno měřit (například čtyři sta až tisíc milimetrů). Nevýhoda 2,5D kamery je vysoká pořizovací cena, tak i pomalá snímací frekvence (snímek). Doba zpracování obrazu je většinou v rozmezí od půl s do jedné vteřiny. Přesnost měření detekce hloubky obrazu je v přesnostech jednotek milimetrů. [16]

Způsob měření hloubky obrazu jsou uvedeny na Obrázek 15 níže.



Obrázek 15) 2,5 kamerové systémy [17]

3.2.2 Typ detektoru

CCD

CCD je typicky charakterizován vlastností dynamického rozsahu, který uvádí rozsah mezi nejjasnější bílou a černou, kterou je senzor schopen rozlišit. Jako senzor tohoto fotonu se používá fotodioda. Při dopadu fotonu na měřicí plochu pixelů jsou generovány fotony, které poté obsahují elektrický náboj. Ten je poté převeden do hodnoty proudu nebo napětí. Šum, který vznikne parazitním spodním nábojem se projevuje tak, že sejmutý tmavší bod se jeví jako světlejší. Tento šum je ovlivněn jeho fyzikálními vlastnostmi a okolní zvýšenou teplotou. [18]

Výhodami CCD detektoru jsou vyšší citlivost (použití při horším osvětlení scény), maximální rozlišení, lepší poměr mezi šumem a signálem. Naopak nevýhody jsou pak vysoká pořizovací cena dána specifickou výrobou, větší rozměry a nemožnost integrace elektroniky přímo na čip. [18]

CMOS

CMOS je v dnešní době méně používaný. Oproti CCD nepřesouvá elektrický náboj, ale přímo jej převádí na elektrické napětí. Snímače jsou založeny na principu pixelového aktivního snímače. Díky tomu, že elektronika na pixelové úrovni je tvořena třemi až čtyřmi

tranzistory, je převedena informace rovnou na napětí, které je poté zesíleno. Dosáhneme tak rozlišení tmavé a světlé části v jednom snímku. Výhodou je možnost využít snímání pouze pixelů, které nás zajímají, což vede ke zrychlení a zvýšení flexibility aplikace. Na druhou stranu tento systém není vhodný tam, kde potřebujeme vysokou kvalitu obrazu, protože systémy CMOS ji nedosahují. V aplikacích, kde je potřeba velice kvalitní obraz, převládá použití CCD snímače. [18]

Lze tedy říct, že CMOS mají výhodnější vlastnosti jako vysoká odolnost vůči šumu, nižší energetická náročnost, méně nákladné na výrobu díky jednoduché konstrukci, menší rozměry, vyšší snímatelná frekvence a vyšší dynamický rozsah. Nevýhodou těchto snímačů jsou špatná citlivost, nutnost mít velice dobře osvětlený prostor, menší kvalita obrazu. Přestože je tento senzor všeobecně méně využívaný, právě pro strojové vidění je aplikace nejčastější. [19]

3.2.3 Černobílá a barevná kamera

Černobílé kamery

Černobílé kamery zpracovávají obraz do matice čísel, kde je každý pixel zobrazen jako hodnota 0–255 (8bit systém), dle množství světla, které na snímač světla dopadá. Černá barva je reprezentována hodnotou 0, a bílá barva hodnotou 255 (pro 8bit systém). Kamery nejčastěji využívají 8bitový systém, ale existují i typy 10bit, 12bit a podobné.

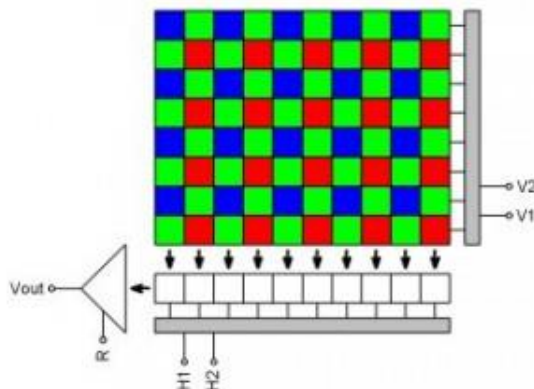
Černobílá kamera je ve strojovém vidění nejrozšířenější, používá se ve všech aplikacích, kdy není potřeba rozlišit barvu snímaného objektu. Výhodou použití černobílé kamery je, že na stejné rozlišení obrazu má třikrát méně dat pro uložení do paměti a proces zpracování obrazu se tak významně urychluje, protože je zpracováno třikrát méně dat.

Ve strojovém vidění je černobílá kamera kombinována s červeným, nebo zeleným osvětlením. Tyto kombinace se používají, protože okolní světlo tolik neovlivňuje změnu osvětlení obrazu, dále je vhodná vzhledem k spektrální citlivosti většiny kamer. Nevýhodou tak je, že kontrast vznikající červeným světlem nemusí být tak jasně odlišný, jak při použití bílého světla. [3]

Barevné kamery

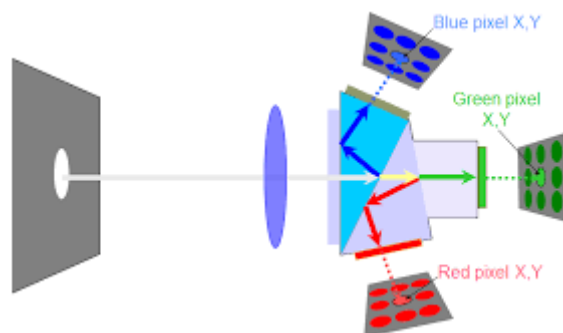
U barevných snímačů obrazu je oproti černobílým kamerám potřeba na jeden bod více světlo-citlivých buněk. Barevné kamery ve strojovém vidění jsou v zásadě tří typů. Jedná se o jednočipové provedení, tříčipové provedení nebo sekvenční snímání s barevnými filtry.

Principem jednočipového provedení je, že dopadá světlo na jeden čip, který má barevnou mozaiku tvořenou ze čtveřice bodů, jeden je červený, jeden modrý a dva body jsou zelené. Takovému rozlišení barev se říká Bayerův filtr. Přináší nejen nízkou rozlišovací schopnost, ale i vzájemný posuv obrazu v jednotlivých barevných kanálech. Příklad barevné mozaiky jednočipového provedení je zobrazen na Obrázek 16. [20]



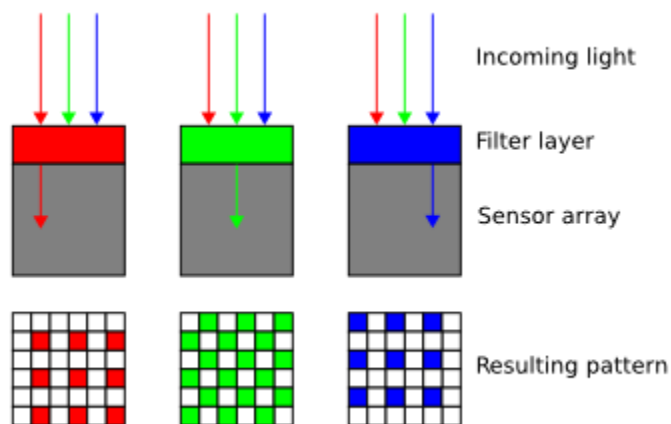
Obrázek 16) Bayerova maska [21]

Tříčipové provedení je založené na lomu světla a jeho průchodnosti. Světlo procházející objektivem je pomocí optického hranolu rozloženo na obraz modrý, červený a zelený. Každá z těchto barev je snímána černobílým senzorem světla a celkový obraz je pak složen ze všech tří obrazů do jednoho. Schéma je uvedené na Obrázek 17. [20]



Obrázek 17) Tříčipové provedení [22]

Poslední z možností principu barevných kamer je použití sekvenčního snímání a barevným filtrem. Zmíněný způsob je velice podobný tříčipovému provedení, s tím rozdílem, že se skládá z jednoho snímače černobílého světla a barevným filtrem, který se natáčí. Světlo prochází přes filtr tak, že projde poprvé pouze červené světlo, snímáno senzorem a obraz je uložen do paměti. Poté se filtr pootočí na další polohu a prochází pouze zelené světlo, které je rovněž uloženo do paměti, obdobný postup jak s modrým světlem. Výsledný obraz je pak složen z třech předešlých. Jedná se o způsob, který se dnes využívá již velmi málo, a to z důvodů dlouhé expozice, kdy se musí zajistit, aby se snímáný předmět nepohnul. Schéma popsaneho provedení je uvedeno na Obrázek 18. [20]



Obrázek 18) Sekvenční snímání [20]

3.3 Objektiv

Další z nejdůležitějších prvků strojového vidění je objektiv. Volba správného typu objektivu a jeho zorného pole je jedním z neméně důležitějších rozhodnutí při návrhu systému strojového vidění. K nejběžnějšímu typu objektivů patří obraz do plochy s tzv. perspektivní projekcí. Uživatelé to tak nutí při návrhu vlastností projektivního zobrazování třírozměrné scény do dvourozměrné plochy povrchu snímacího senzoru. Zorné pole objektivu je tak pohledovým komolým kuzelem. Plocha obrazového senzoru má tvar obdélníku, tento kužel dále redukuje na jehlan. Vrchol jehlanu se nazývá ohnisko projekce. Ke značné ztrátě informací dochází při převodu obrazu scény uvnitř pohledového jehlanu do obrazové plochy. Každá polopřímka procházející ohniskem projekce je v obrazové ploše reprezentována jediným bodem. [23]

Za předpokladu teoreticky dokonalého objektivu s lineárním převodem úhlu je nutné se vyrovnat se zkresleném obrazu vlivem perspektivní chyby. Představme si například obraz temných teček na světlém pozadí s konstantním rozestupem. K jeho dosažení předpokladu přesné projekce v promítaném obraze, musí být předloha snímána z vnitřního povrchu kulové plochy. Při rovinné předloze se v promítnutém obraze tečky budou od sebe oddalovat v závislosti na jejich vzdálenosti od optické osy. Ztráta prostorové informace během perspektivní projekce komplikuje například měření rozměrů trojrozměrných těles. Bez předešlých znalostí tvarů snímaného předmětu nelze tyto ztráty dodatečně eliminovat. Je potřebná vysoká úroveň porozumění obrazu, protože i když známe tvar předmětu, vyžaduje korekce projektivních chyb identifikaci objektů programových vybavení pro strojové vidění. [23]

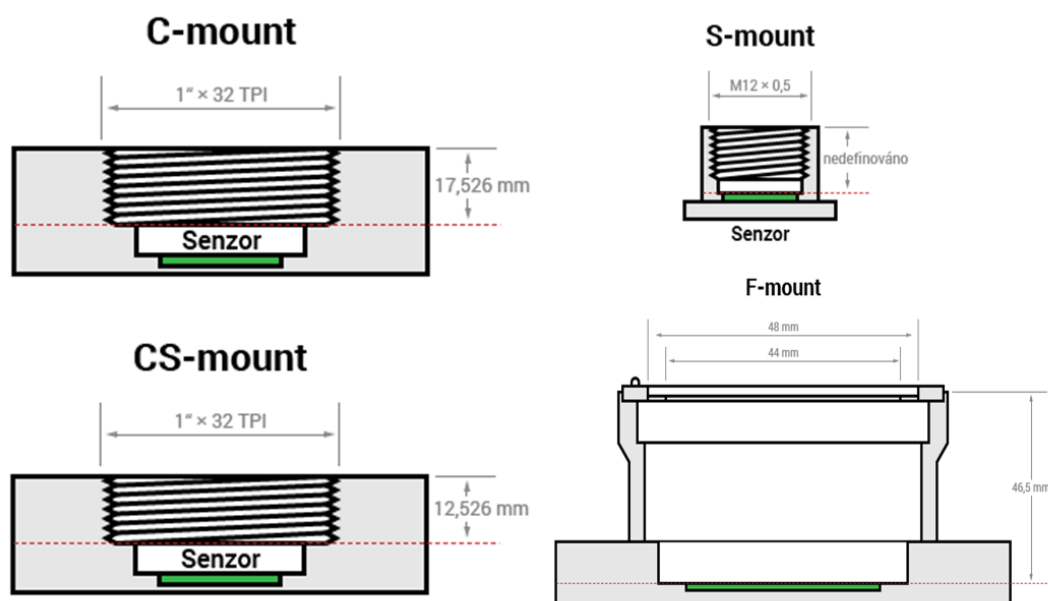
3.3.1 Důležité parametry pro volbu objektivu

Světlost je bezrozměrná veličina udávající množství světla, které je schopen přenést objektiv kamery. Je označována písmenem f a číselnou hodnotu například 18-55 mm $f3-5$, což znamená že objektiv propustí třetinu až pětinu světla v úhlu záběru podle posunutí ohniska objektivu. Ideální objektiv dosahuje hodnoty 1, tedy propustí veškeré světlo. Většinou se hodnoty pohybují od $f/1,2$ do $f/6,3$, objevují se však i výjimky speciálních objektivů kdy se pohybuje i na hodnotě 0,95 propustnosti světla. Platí pravidlo, že čím menší clonové číslo, tím menší hloubka ostrosti. [24]

Hloubka ostrosti je hodnota udávající rozsah vzdálenosti, kdy dokáže zaměřit obraz ostře. Čím vyšší hodnota hloubky ostrosti, tím ostřejší snímky na větší vzdálenosti můžeme pořizovat. Vysoké hodnoty se používají spíše při focení krajin a podobné, naopak velice nízké hodnoty využívají spíše mikroskopy, kde se hodnoty pohybují na úrovni desetin milimetrů. [25]

Ohnisková vzdálenost. Čím větší hodnota ohniskové vzdálenosti, tím nižší hloubky ostrosti dosáhneme. Vzniká tak rozostřené pozadí, například pro objektivy hodnot 100 a více milimetrů. V případě strojového vidění jsou nejčastěji využívány objektivy s menší hodnotou ohniskové vzdálenosti obvykle, mez deseti až dvaceti milimetry. Snímek je poté velice ostrý. [25]

Závit objektivu je důležitý hlavně proto, aby byl stejný typ montážního závitu na objektivu i na kameře. Neliší se jen velikostí, ale také nastavením objektivu do potřebné vzdálenosti od roviny senzoru. Nejmenším závitem je upevnění s názvem S-Mount, který je používán převážně pro vestavbu. Nejběžnějším typem závitu objektivu pro strojové vidění je typ C-Mount, na trhu nejrozšířenější. Obdobným typem je i CS-Mount, který má stejný závit jako C-Mount s rozdílem, že CS-Mount má kratší závit o pět milimetrů. Objektiv C-Mount lze tedy připevnit na kameru se závitem CS-Mount při použití redukčního kroužku o tloušťce pět milimetrů. Opačně to však nefunguje, protože by kamera nezaostřila. Mezi další typ patří F-Mount, který uvedla na trh firma Nikon. Tento typ se používá s velkoformátovými senzory a řádkovou kamerou. Výhodou tohoto typu oproti ostatním závítům je bajonetový spoj. Jako alternativa typu F-Mount se používají závity M42, M58x0,75 nebo 72x1, jedná se o velice robustní typ závitu. Příklady jsou uvedeny níže na Obrázek 19. [26]



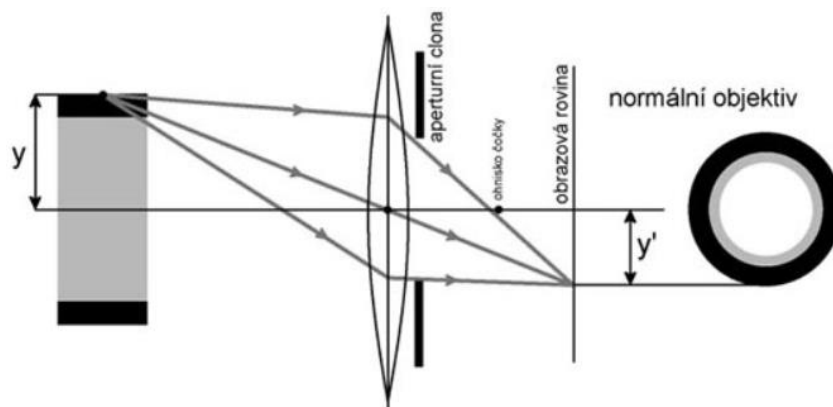
Obrázek 19) Montážní závity objektivů [26]

IF nebo RF korekci používá jen určitá část výrobců. Jedná se o objektiv s funkcí tzv. vnitřním nebo zadním ostření. Neprobíhá tak klasické ostření, při kterém se pohybuje celým objektivem, ale pouze pohybem několika čoček uvnitř. Díky tomuto typu ostření nedochází k otáčení přední části objektivu, ani ke změně jeho rozměrů, a především je zmíněný způsob ostření daleko rychlejší.

3.3.2 Typy objektivů

Standardní objektivy

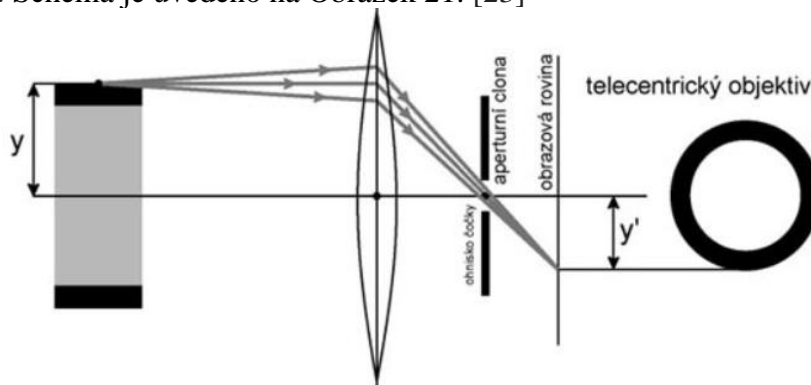
Označují se také jako normální objektivy. Mají ohniskovou vzdálenost podobnou lidskému oku, a tím pádem uhel zorného pole, které je téměř totožný jako lidské oko. Schéma standardního objektivu je uvedeno na Obrázek 20. Standardní objektivy se využívají tam, kde nahrazujeme lidskou obsluhu. Zatímco normální objektiv je zacloněn před hlavním obrazovým bodem, což umožňuje vstup paprsků z celého pohledového kužele. Vlivem směru paprsků pozorujeme jak průřez, tak i vnitřní strany stěn. Nevýhodou je změna velikosti tvaru v závislosti na vzdálenosti objektivu od snímané součásti. Vzhledem k této vlastnosti nebývá objektiv až na výjimky u strojního vidění při měření používáný. [27]



Obrázek 20) Standardní objektiv [28]

Telecentrické objektivy

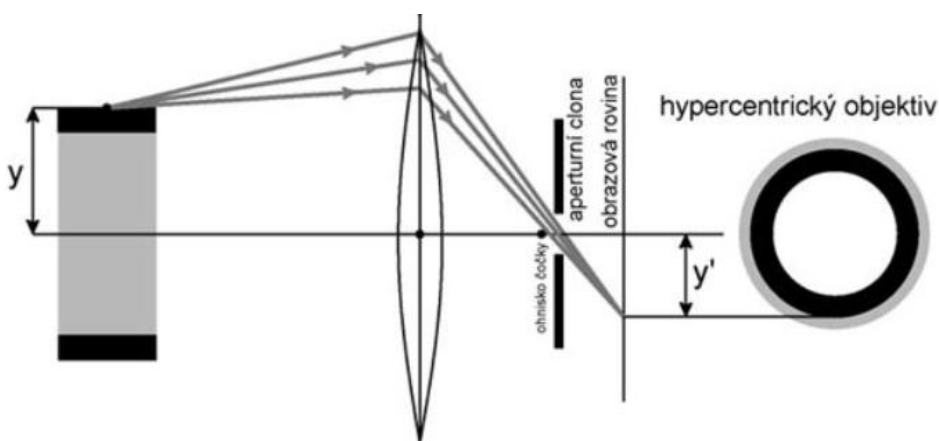
Mimo standardní objektivy s perspektivní projekcí existují také speciální objektivy s ortografickou projekcí. Zobrazují scénu s kolmou paralelní projekcí, nikoliv však s ohniskovou perspektivou jako objektivy standardní. Díky zobrazení s kolmou paralelní projekcí jsou zobrazené velikosti předmětu vždy stejné, nezávisle na jejich vzdálenosti (tzv. délkojevná projekce). Telecentrické objektivy s ortografickou projekcí mají velice jednoduchý princip. Pomocí aperturní clony umístěné v rovině hlavního obrazového bodu se odstíní veškeré paprsky přicházející z jiných směrů než rovnoběžně s optickou osou. Při tomto typu projekce ale musí být velikost vstupní plochy objektivu shodná s plochou snímané scény. Proto jsou telecentrické objektivy velice rozměrné a drahé, přesto ale pro měření nejpoužívanější. Schéma je uvedeno na Obrázek 21. [23]



Obrázek 21) Telecentrický objektiv [28]

Hypermetrický objektiv

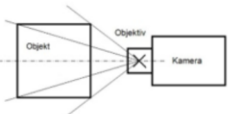
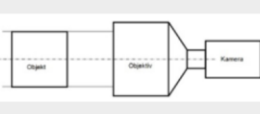
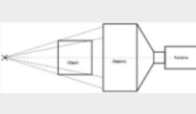
Tento objektiv je velice podobný Telecentrickému s tím rozdílem, že posuneme aperturní clonu mezi obrazový hlavní bod a obrazovou rovinu. Dostáváme tak pohledový kužel obráceně a s rostoucí vzdáleností od objektivu se zužuje. Aperturní clona je umístěna mezi hlavním obrazovým bodem a obrazovou rovinou, vzdálenější předměty jsou pak zobrazovány jako větší. Na rozdíl od telecentrického, který díky rovnoběžnosti paprsků vidí přímo průřez, hypercentrický objektiv vidí navíc vnější stěnu. Schéma popsaného objektivu je uvedeno na Obrázek 22. [12]



Obrázek 22) Hypermetrický objektiv [28]

Porovnání výše uvedených typů objektivů

Na výše uvedených obrázcích projekce uvedených tří typů objektivů (standardní – Obrázek 20, Telecentrický – Obrázek 21, hypermetrický – Obrázek 22) pohledů v ose trubky lze rozpoznat jednotlivé rozdíly mezi objektivy. Standardní objektiv snímek znázorňuje tak, že vidíme čelo a vnitřní stranu trubky. Telecentrický objektiv znázorní pouze čelo trubky a hypercentrický objektiv nezvykle znázorní čelo i vnější povrch trubky. Detailnější rozdíly mezi jednotlivými typy objektivů jsou popsány v Tabulka 1 níže. [28]

Druh objektivu	Standardní	Telecentrický	Hypercentrický
Princip			
Pozice středu perspektivy	Mezi kamerou a analyzovaným objektem	V nekonečnu	Za analyzovaným objektem
Způsob zobrazení objektů	V dálce malé, blízko velké	Stejná velikost bez ohledu na vzdálenost	Malé blízko, velké v dálce
Délka objektivu	malá	Odpovídá pracovní vzdálenosti	Větší než pracovní vzdálenost
Průměr objektivu	malý	Větší než analyzovaný objekt	Mnohem větší než analyzovaný objekt
Pracovní vzdálenost	$> 0 \dots \infty$	typicky < 300 mm	Velmi malá

Tabulka 1) Porovnání objektivů [29]

3.3.3 Vady objektivů

Všechny optické sestavy mají řadu zkreslení a vad, způsobených nedokonalostí soustavy, lomem světla a jeho vlnovým charakterem. Seznam nejčastějších vad a informace, jak jim zabránit, nebo je alespoň zmírnit je uvedený níže:

Perspektivní zkreslení obrazu – dochází k němu vždy při použití standardních objektivů vždy. V aplikaci strojového vidění, kdy je potřeba přesného měření, nebo navádění robotu způsobí vadu, která může diametrálně změnit výsledky. Jestliže, ale budeme měřit barevnost, pouhou přítomnost objektu, nebo počet objektů, může být vada škodlivá. Pro zabránění vady je vhodné použít telecentrických objektivů, které odstíní paprsky z jiných směrů, než je rovnoběžný s optikou. [29]

Optické vady – platí pouze pro monochromatické světlo a paprsky v těsné blízkosti optické osy, avšak vzdalováním se od osy se začínají projevovat složité úkazy, které snímek deformují. Bohužel k těmto vadám dochází u všech objektivů, lze je pouze zmenšovat ať už více či méně. Tyto vady jsou například vady sférické, astigmatické, asymetrické, chromatické, zkreslení soudkovité, poduškovité či vlnovité. [29]

Vinětace – projevuje se při zcela odcloněném objektivu, lze ji však snížit, a to přivíráním clonového kroužku do určité hodnoty. Jedná se o vadu, kdy je rovnoměrné světlo směrem od středu obrazu ke krajům. [29]

Centrování – projevuje se sníženou ostrost snímku v rozích obrazu. Je způsobena podobně jak výše uvedena chyba, a to plným odcloněním objektivu. Stejně jako u vinětace se zavíráním clonění tato chyba zmenšuje. [29]

Reflexy – jedná se o nechtěné odlesky světla od členů optické soustavy, jako vnitřní části tubusu objektivu, clonovém kroužku a dalších součástech uvnitř objektivu. Těmito odrazy vznikají na snímcích nejrůznější světelné skvrny a závoje. Možnost ovlivnění chyb návrhatelem je použití co nejmenšího počtu optických členů, tedy objektivu s pevným ohniskem, který je k této vadě méně náchylný. K zamezení vady dochází nejčastěji již u

výrobce použitím antireflexní vrstvy na optických členech a speciální povrchové zpracování jednotlivých konstrukčních prvků objektivu. [29]

Chronická vada – projevuje se v důsledku rozdílného indexu lomu každé barvy ve světelném spektru. Vada je viditelná pouze na velice ostrých snímcích a kontrastních hranách objektů, které jsou mírně rozmazané vlivem zobrazení jako kontura v barevném spektru. Vada se dá ovlivnit pouze ze strany výrobce, a to korigováním použití různých materiálů při výrobě optických členů o různém indexu lomu světla. [29]

4 Princip a charakteristika strojového vidění

Obvyklým postupem při řešení aplikace ve strojovém vidění je nejprve co nejpodrobněji specifikovat požadavky řešené aplikace. Poté vybereme vhodné komponenty, testujeme aplikaci a vytváříme tzv. scény a algoritmy pro zpracování obrazu. Nakonec dochází k finálnímu testování a zprovoznění. [1]

4.1 Co je strojové vidění

Zpracování obrazu v počítači vzniklo v sedmdesátých letech minulého století. Již v této době byla výpočetní technika na úrovni, kdy umožňovala zpracovávat data ve velkém objemu nesoucí obrazovou informaci. Vznikl tak nový obor, který se od této doby nazývá *computer vision* neboli počítačové vidění. Zmíněný název v dnešní době označuje obecně systémy, které automaticky pracují na základě zpracování obrazů získaných z kamery. Předmětem počítačového vidění může být v základu prakticky cokoli, například lidské oči, lidská postava, dopravní situace a jiné. Patří sem také proces výroby, pro který je dnes používán *machine vision* neboli strojové vidění. [1]

Strojové vidění je v dnešní době používáno v průmyslové automatizaci. V praxi se využívá stále častěji, a to především pro bezkontaktní a velice rychlou vizuální kontrolu kvality výrobků, počítání objektů, ke kontrole sestav či čtení textů a nejrůznějších kódů uvedených na výrobcích.

4.2 Princip strojového vidění

Strojové vidění je velice podobné lidskému. Na rozdíl od lidského oka zachycuje obraz zkoumaného objektu kamera. Systém záznam z kamery vyhodnotí pomocí předepsaného algoritmu a na základě vyhodnocení provede akci. Z pohledu technického popisu je nejjednodušší hledět na systém strojového vidění jako na obecný senzor.

Snímaný objekt je ozařován pomocí osvětlovače. Osvětlovací soustava musí být navržena tak, aby záření vytvořilo na snímacím prvku kamery jasový dvojrozměrný obraz. Důležité je, aby byla v obraze obsažena informace o sledovaném objektu, kterou je zapotřebí znát. Podmínka se zdá být triviální, je však problémem spousty úloh. Nejdůležitější je *know-how* integrátora a jeho znalosti. [1]

Dvojrozměrný jasový obraz se vytváří na citlivém prvku kamery. V kameře s polovodičovým snímacím prvkem jsou měrnými veličinami velikosti napětí nebo proudů, které jsou úměrné míře dopadajícího světla. U černobílých kamer tyto veličiny nesou informaci na údaj o kombinovaném jasu barevných složek. U barevných mohou nést i informaci o barvě snímaného předmětu. Protože jediným efektivním způsobem zhodnocení obrazu je v dnešní praxi jediná možnost a to digitalizovat obraz. To znamená převést analogový signál do digitálního. Digitální informace je pak v počítači jednoduše zpracována pomocí algoritmů, aby byla získána požadovaná informace o objektu. Tomuto procesu se říká *image processing*. Takto obdržená informace se poté po komunikační sběrnici posílá do okolí. Komunikační rozhraní je určeno k připojení zařízení, díky kterým lze zařízení nastavovat, programovat a udržovat. Takové zařízení se označuje jako MMI (*Man-Machine-Interface*), jako takové se většinou využívá počítač typu PC. [1]

Typické pro strojové vidění je, že výměna informací systému a okolí funguje jako zpětná vazba při řízení výrobního procesu. Systém tak získá nezbytné vstupní údaje, nejčastěji to povel k pořízení snímku v okamžiku, kdy je sledovaný objekt právě ve vhodné poloze, případně prostoru. Podle výsledku vyhodnocení obrazu systém vykoná zásah do procesu pomocí akčního členu. Nejčastěji vyřazení vadného kusu nebo jeho indikaci. Akční zásahy může provést jak samotný digitální výstup systému strojového vidění, nebo je informace předána řídicímu systému, který potom provede zásah sám. [1]

4.3 Kategorie systémů strojového vidění

Zařízení strojového vidění lze sestavit několika způsoby. Je možné použít nejrůznější techniku, části systému lze kombinovat a integrovat. V průběhu vývoje se systém ustálil na čtyři všeobecně uznávané kategorie. Jedná se o kamerový senzor, inteligentní kameru, PC systém a zákaznický systém. Charakteristické znaky jednotlivých kamer jsou uvedeny v Tabulka 2. Je potřeba podotknout, že způsoby těchto systémů se překrývají a rozvojem se navzájem sblíží. Komerční výroba a dodavatelé hardwaru pro strojové vidění nemohou stavět systémy individuálně, ale snaží se mít kamery s pokrytím co nejvíce úloh. Detailnější popis jednotlivých systému je popsán v odstavcích níže. [1]

Kamerový senzor <i>Vision sensor</i>	Kompaktnost, zjednodušený hardware, omezené softwarové nástroje, omezené I/O, omezené komunikační možnosti, omezená třída použití, nižší cena
Inteligentní kamera <i>Smart camera</i>	Kompaktnost, výkonný mikroprocesor, univerzální softwarové nástroje, více I/O, vyšší komunikační možnosti, široké možnosti použití, vyšší cena
PC/PLC systém <i>PC/PLC system</i>	Modularita, možnost více kamer, výkonný hardware PC/PLC, možnosti rozšíření, univerzální software nástroje, široké komunikační možnosti, široké možnosti využití, vyšší cena
Zákaznické systémy <i>Custom vision</i>	Nepatří do žádné z předchozích tříd, úplná podřízenost požadavkům zákazníka, speciální použití, vysoká cena

Tabulka 2) Kategorie strojového vidění [1]

4.3.1 Kamerový senzor (*vision sensor*)

Typicky nejjednodušší systém strojového vidění. Je nejčastěji celý integrován do jednoho pouzdra a velice omezeně má oddělenou kameru. Počet digitálních vstupů a výstupů má omezen a mají už předem definované funkce, například port pro řízení obrazu, nebo výstup OK či NOK a podobné. Komunikační rozhraní má pouze pro připojení zařízení pro nastavení kamery v softwaru, který je dodaný dodavatelem kamery. Zřídka se nastavuje kamera pouze z vestavěné klávesnice (používáno spíše v dřívější době). Konstrukce kamerového senzoru včetně výkonu procesoru a výrobních nákladů je podřízena účelu vytvořit systém strojového vidění pro velmi snadné použití. [1]

4.3.2 Inteligentní kamera (*smart camera*)

Jedná se o systém velice podobný kamerovému senzoru (*vision sensor*) popsáný v kapitole výše. Charakteristickým znakem inteligentní kamery je její kompatibilita s použitím výkonného čipu. Program se ukládá do paměti typu flash, která bývá většinou vestavěný, ale má i možnost použít externí karty MM nebo SD karty. Jako vstupy a výstupy jsou často uplatněny digitální bitové rozhraní, které se volitelně využívají jako vstupy, nebo výstupy. Tyto vstupy a výstupy odpovídají standardům používaných v průmyslu, tedy do třiceti voltů. V některých případech vstupy a výstupy úplně chybí a je potřeba použít adaptér, který lze připojit ke komunikačnímu rozhraní. [1]

Komunikační rozhraní se používá k několika funkcím. Nejčastějším připojením je Ethernet, souží jak pro hlavní funkci, která je určena k připojení zařízení pro programování funkcí kamery. V současné době je jako zařízení pro programování kamery používají v drtivě většině v PC s nainstalovaným programem dodaným výrobcem kamery. Komunikační rozhraní poté může být použito pro přenášení dat do nadřazeného PLC, nebo jiného z řídicích systémů, dále umožňují komunikaci mezi několika dalšími kamerami. Mimo obvyklé připojení Ethernetu bývá vybaven i pomocným sériovým rozhraním RS-232, nebo RS-485. Důvodem je možnost připojení k většině existujících PLC. Průmyslové sběrnice typu Profibus, Modbus a podobné se u těchto kamer většinou neobjevují, avšak tuto komunikaci umožňují převodníky z Ethernetu. [1]

4.3.3 PC/PLC systémy (*PC/PLC vision*)

PC/PLC systém je odlišný od výše uvedených hlavně tím, že je modulární. Vyhodnocení pořízeného obrazu se provádí na straně vyhodnocovací jednotky. Kamera je samostatná a může jich být použito i několik. Nejčastěji jsou kamery připojeny za použití vysokorychlostního Ethernetu. Výpočetní výkon a velikosti paměti jsou dány použitým PC/PLC. Co se týče digitálních vstupů, jsou určeny kartami připojenému k PLC. Díky tomu, že kamera pouze posílá obraz a veškeré operace jsou řešeny v rámci PLC, nabízí nám neomezené možnosti použití vstupů a výstupu, komunikací a podobné funkce co trh nabízí. Protože se inteligentní kamery blíží výpočetnímu výkonu PLC, používá se systémem méně. Nejčastěji se tento způsob řeší pouze ve větších aplikacích, kdy je potřeba většího množství kamer, případně přenosu obrazu do HMI panelu. [1]

4.3.4 Zákaznické systémy (*custom vision*)

Zákaznické systémy jsou ojedinělé, jedná se o systém stavěný na míru pro případ, kdy nelze použít ani jednu z výše uvedených variant. Patří sem například úlohy vyžadující vysokou rychlost zpracování, speciální algoritmy nebo nestandardní kamery. Provedení zákaznického systému odpovídá danému účelu. Nejčastěji se k realizaci používají rychlé signálové procesory nebo hradlová pole se softwarem vytvořeným na míru. [1]

5 Typické úlohy pro strojové vidění v průmyslové praxi

Mezi nejčastější aplikace strojového vidění patří kontrola produktu, aby byla zaručena jeho naprostá bezchybnost při výstupu produktu z výrobního procesu. Daný proces je tedy posloupnost výrobních operací. Existuje mnoho aplikací pro kontrolu kvality, všechny však předpokládají naprosto bezchybné provedení jakékoliv, nebo alespoň většiny operací tak, aby byl výsledný produkt bezchybný. Právě strojové vidění umožňuje tuto podmínku splnit. Nejtypičtějším použitím strojového vidění ve výrobě je kontrola správnosti provedení výrobní operace, bin picking, nebo nalezení polohy součásti, případně jejich měření. Jednotlivé operace, kdy je strojové vidění použito v podkapitolách níže. [30]

5.1 Nalezení povrchových vad

Daná úloha je typická pro fázi vstupní a výstupní kontroly materiálu, výrobků a podobně. Může se jednat o identifikaci poškození povrchu materiálu, kvalitu nátěrů a jejich vad, kontrola povrchových úprav, potisků a jiné. Mimo vstupní a výstupní kontrolu můžeme kontrolovat kvalitu povrchu materiálu i v průběhu výrobních operací. [31]

5.2 Nalezení polohy a navádění robotu

Používá se při nakládání komponentů na pás, které následně odebírá robot. Součásti totiž mají mírně rozdílnou polohu a orientaci, ty je potřeba ji změřit. Další typickým použitím je případ, kdy jede několik součástí na páse a jsou následně pomocí několika robotů přeskládány na jinou pozici, případně dalším uplatněním je přesné robotické svařování, kde nelze umístit součást na přesnou pozici. Tato operace se v průmyslové praxi moc často nepoužívá, protože je jednodušší a levnější vyřešit přesné polohování součástí. [30]

5.3 Měření a kontrola tolerancí

V dnešní době, kdy je důraz na stoprocentní kontrolu vyšší než kdy jindy, je daný směr strojového vidění nejvíce v pokroku. Také však patří mezi ty, které jsou nejvíce náročné na výpočetní výkon procesoru. Měřit pomocí strojového vidění je možné nejen ve 2D, ale lze snímat a porovnávat i 3D objekty, pomocí tzv. 3D -scannerů. Výhodou měření pomocí strojového vidění je jednoduchost systému, rychlost a zároveň přesnost. [30]

5.4 Identifikace barev

Tato aplikace se vyskytuje velice ojediněle. Jedná se o úlohu, kdy je třeba rozpoznávat barvy při vstupní nebo výstupní kontrole, může jít např. o jeden ze znaků při identifikaci typu materiálu. V potravinářství může barva potraviny signalizovat sníženou kvalitu produktu vstupujícího do operace vyskladnění. [31]

5.5 Čtení kódů a textů

Čtení a rozpoznávání kódu jsou málokdy samostatnou úlohou strojového vidění. Pokud však nastane úloha, kdy je cílem strojového vidění pouze číst nějaký kód, nebo text je lepší použít specializované čtečky. Pro čtení kódů a textu se ve strojovém vidění používá například pro třídění komponent, nebo zápisu komponent do skladového systému.

Kódy se většinou čtou v první a druhé fázi výrobní operace, kdy je zjištěno, zda je k operaci dodán správný díl na správném místě. Čtení je rovněž typické pro poslední operaci, pro identifikaci dílů, nebo například pro čtení popisku s výrobním číslem a datem expedice. [31]

5.6 Bin picking

Jde o operaci, kdy se z krabice odebírají součásti, jež jsou nahodile otočeny od sebe. Strojové vidění se je snaží analyzovat a zjistit, jak jsou natočené a předat tuto informaci robotu, aby věděl, jak nejlépe předmět uchopit. Bin picking je nejčastěji používán v kombinaci s vibračním stolem. V případě, že není nalezena jediná součást tak, abychom ji byli schopni odebrat, spustí se motor a vibrační stolu se komponenty pootočí. Bin picking se používá hlavně pro manipulaci s drobnými součástmi jako třeba elektrické čipy. [31]

6 Robot a strojové vidění

Ve výrobních systémech i přes všechnu možnou automatizaci co je dnes nabízena je daleko levnější nakládat produkty z krabice, nebo produkty nakládat nahodile, protože pomocí lidského nakládání nebude nikdy dosaženo přesné polohy. Náhodné nakládání materiálu ovšem neusnadní problém, jak se na první pohled může zdát, ale spíše je posune na další úsek výrobní linky, kdy je potřeba s tímto materiálem manipulovat. To poté vyžaduje buď drahou pracovní sílu, systémy pro přesné polohování, nebo singulátory, jako například vibrační zásobníky. Pro robot je poté poměrně složité odebrat takový náhodně orientovaný produkt z krabice. [32]

Je to tedy úkol relativně obtížný, avšak neřešitelný. Spousta firem používá pro *bin picking* (neboli česky „výběr z bedny“, nebo „výběr náhodně orientovaného objektu“), 3D průmyslovou kameru. Robot odebírá nahromaděné díly z krabice a podle předem naprogramovaného kódu, díky kterému orientaci a natočení již známe. Součásti mohou mít různé tvary a velikosti, takto lze odebírat z bedny nahodile orientované polotovary a poté je přesně zakládat například do obráběcího centra. Odebírá je jeden po druhém a přesnost detekce se většinou pohybuje okolo dvou milimetrů, záleží na velikosti bedny, kde se polotovary nachází. [32]

6.1 Systém strojového vidění OnRobotEyes

Jedná se o systém, který je určený pro montáž na rameno robotu, který umožňuje pomocí jednoho snímku zkalibrovat a následně podle barev a tvarů rozpoznávat objekty určené ke třídění a uchopovacích úloh jako například operace pick and place, při zakládání polotovarů do výrobního stroje a podobné. Kameru je možné umístit jak na rameno robotu, tak na pevné rameno konstrukce stroje. [16]

Systémy OnRobotEyes používá 2,5D kameru, díky které je tedy schopný snímat, jak obraz, tak i hloubku obrazu metodou Active IR Stereo. Je tak schopný detekovat různé tloušťky předmětů, nebo předměty naskládané na sobě. Pracovní vzdálenost těchto kamer se pohybuje v rozmezí čtyři sta až tisíc milimetrů, obvyklá doba zpracování obrazu se pohybuje okolo půl vteřiny s přesností do dvou milimetrů. [16]

Jednou z výhod tohoto systému je jednoduché programování kamery a tím i rychlé zavedení do provozu. Tento systém je kompatibilní se všemi předními výrobci jak průmyslových, tak kolaborativních robotů. Jedná se tedy o jednoduchý systém strojového vidění, kde se předpokládá s častými změnami výroby a požadavkem flexibility robotizovaného pracoviště. Systém však nemusí být spojen jen s robotem, ale lze jej použít i pro méně náročné operace automatické kontroly kvality. [16]

6.2 Sjednocení systému souřadnic robotu a obrazu

Prímý vstup o poloze souřadného systému zjištěných zpracováním obrazu, do řídicího systému robotu vyžaduje, aby byl systém souřadnic odkud jsou informace předávány identické se systémem souřadnic, které používá robot. V podstatě však nelze dosáhnout toho abychom orientovali kameru tak aby její souřadnice byly totožné s těmi robotu. Obraz je natočen, zkeslen perspektivou, chybou objektivu a dalšími rušivými jevy. [33]

Pro kalibraci kamery ať už pro podmínky měření či sjednocení souřadnic se dnes využívá kalibrace pomocí jednoho či více kalibračních štítků. Na těchto štítcích je nejčastěji zobrazena šachovnice, případně pole kruhu, ať už symetrických, nebo nesymetrických. Kalibrovat kameru můžeme i v průběhu operace automaticky, nebo manuálně za pomoci obsluhy. Nejčastěji je někde v zorném poli obsažen souřadný systém, ke kterému je odeslána informace. Kalibrace nemusí být vždy jen standartní šachovnicí, či pole s kruhy, lze si vytvořit vlastní a, nebo u některých výrobců lze použít i 3D model. [33]

7 Logické řízení a strojové vidění v jednotném prostředí TwinCAT Vision

Software společnosti Beckhoff dovoluje kombinovat do jediného systému s logickým řízením, řízením polohy a pohybu, vizualizací, řízením robotů a bezpečnostními funkcemi. Pomocí softwaru TwinCAT Vision umožňuje aplikace strojového vidění. Díky tomuto softwaru se kamery konfiguruji podobně jako ostatní průmyslové sběrnice nebo jiné pohybové osy. Zpracování obrazu se provádí pomocí algoritmů, které pochází z programovacího jazyka IEC 61131-3. Výhodou těchto algoritmů je, že jsou zpracovávány ve stejném real-time jako systému zpracování PLC programu. [34]

PLC programátor v průmyslové automatizaci má kontrolu nad řídicím programem, pohony, bezpečnostními funkcemi, pracuje s digitálními i analogovými vstupy a výstupy, dělá také s mnoha dalšími funkcemi. Každá z těchto úloh je však součástí jednoho konfiguračního nástroje, jedné řídicí platformy. Tvoří je tak v běžných programovacích jazycích pro PLC. Avšak je jedna oblast průmyslové automatizace, která je standardně od těch dříve zmíněných typicky oddělena. Je tedy většinou řešena externě, nejčastěji používá svoje vlastní programovací jazyky a konfigurační nástroje pro naladění. Jedná se tedy o oblast strojového vidění a zpracování obrazu. TwinCAT vision kombinuj tyto světy do jednoho. Kamery se konfiguruji ve stejném nástroji jako průmyslové sběrnice nebo pohybové osy. K programování algoritmů pro zpracování obrazu se používají programovací jazyky PLC a tyto algoritmy pracují ve stejném real-time systému jako zpracování programu PLC. [34]

7.1 Architektura

Veškeré řídicí systémy firmy Beckhoff jsou založeny na platformě PC, díky tomu mají běžně rozhraní s gigabitovým internetem. Na ethernetovém rozhraní je právě založen přenosový standart GigE Vision pro spolehlivou a rychlou komunikaci, pro přenos obrazových dat. TwinCAT Vision driver pro tento účel využívá reálného času pro přenos dat z ethernetu přímo do paměti řídicího systému. Nejen díky podpoře GigE standardu je možnost využití velkého množství různých druhů kamer, které jsou na trhu k dostání od různých výrobců. [34]

Nejdříve je potřeba zajistit spojení kamery s hardwarem Beckhoffu. Poté typicky následuje konfigurace kamery. Tu ulehčuje formát GenAPI, který definuje jednotlivý popis konfiguračních dat u kamer s rozhraním GigE. Pro práci s konfiguračními daty je ve vývojovém prostředí TwinCAT určen tzv. TwinCAT Vision Configuration Assistant. Ten umožňuje uživateli, jak přehledně zobrazit, tak i změnit všechny parametry kamery, které jsou pro nastavení kamery nutné. Změny konfigurace, jako nastavení oblasti zájmu, nastavení doby osvětlení a podobné změny konfigurace, lze nastavit rychle a jednoduše. Vliv změn na nastavení konfigurace kamery také lze zobrazit online přímo v prostředí TwinCAT. [34]

Vývojové prostředí nabízí i jiné průvodce kalibrace kamery, například kalibrace, kdy je cílem odstranit geometrické zkreslení obrazu. Součástí tohoto průvodce je také určení parametrů popisující vztah mezi souřadného systému kamery a reálného světa. Na základu tohoto vztahu lze vztáhnout pozice z obrazu k pozicím reálného světa, ale i rozměry, které jsou udávány v jednotkách pixelů lze převést na jednotky běžného metrického systému. Mimo perspektivní zkreslení jsou brána v úvahu i nelineární zkreslení projevující se jako viditelné deformace v obraze. [34]

Kalibrace kamery může probíhat nasnímáním jednoho, či více kalibračních vzorů. Vzory mohou být snímány přímo ve vývojovém prostředí, nebo je možnost importu již existujících obrázků, například z informačního systému Beckhoff. Po výběru kalibračního vzoru je potřeba už jen spustit automatický výpočet kalibračních parametrů. Mimo jiné vzory, než jsou běžné, jako například šachovnice, nebo pole s kružnicemi nebo kruhy je možnost použít i jakýkoliv uživatelem vytvořenými 2D nebo 3D vzory. Kalibrace kamery také nemusí

probíhat manuálně za přítomnosti obsluhy, ale může probíhat například i pravidelně automaticky přímo z PLC. [34]

7.2 Zpracování obrazu v PLC

Obrazová data se ukládají přímo do paměti PLC. Obrazová data lze získat několika způsoby, ve virtuální přípravě lze předpřipravit program tak, že nahrajeme již vytvořený obraz, který nahrajeme do PLC a voláme jej pomocí triggeru a funkčního bloku FB_VN_FileSourceControl. Případně pokud se jedná přímo o vyčítání obrazu z kamery je potřeba, aby kamera byla ve stavu, kdy zaznamenává obraz, popřípadě dostává signál k vytvoření snímku (trigger), který snímání obrazu řídí. K tomu je určen funkční blok PLC FB_VN_GevCameraControl. Odeslání signálu trigger zajišťuje výstupní terminál EL2262, ten má podporu odesílání přesného řízení výstupního signálu. Díky této funkci dokážeme odesílat signál trigger odesílat v řádu jednotek milisekund. Poněvadž toto probíhá v jednom real-time systému s přesnou časovou synchronizací, tak lze z kamery časovat obrazy tak že na sebe navazují. Většina kamer má i vlastní digitální výstupy, které odesílají v návaznosti s předem definovanými akcemi, jako je například oznámení počátku snímání obrazu a podobně. Vstupní digitální terminály společnosti Beckhoff dokážou tyto signály odchytnout na straně řídicího systému, a proto mohou být využity k dodatečné synchronizaci. [34]

Pro zpracování obrazových dat je v prostředí TwinCAT Vision dostupná PLC knihovna s názvem TC2_Vison, která obsahuje spoustu dostupných algoritmů. Při předzpracování obrazu lze upravit obraz, například měnit velikost obrazu, převést snímek do barevného, nebo černobílého formátu, případně potlačit, nebo zvýraznit vlastnosti filtrů. Dále je také možné provést binarizaci pomocí prahování k nalezení kontur v obraze. Z těchto nalezených kontur lze na základě jejich vlastností vyhodnotit, které vhodné pro rozpoznávání a měření objektů. Díky kalibraci kamery je možné převést body popisující kontury zpět na souřadnice reálného systému, jako polohu, rozměry a podobné. TwinCAT Vision běží jako real-time prostředí, takže je nutné použít tzv. WatchDogy, které hlídají délku cyklu. Tato funkce je zavedena z důvodu, že když například projde kolem stanice s kamerou obsluha a vytvoří stín, nebo nějak posvítí do prostoru kamerou nějakým světlem, může vytvořit spousty nových kontur, které bude muset PLC vyhodnotit a cyklus by trval opravdu dlouho, případně by hrozilo seknutí programu. I při předčasném ukončení výpočtu neztrácíme veškerá data a nepočítáme znovu od začátku, máme k dispozici alespoň částečná data. Dále není problém rozdělit funkce na zpracování obrazu mezi jádra pomocí tzv. job-task. TwinCAT dokáže optimalizovat využití všech možností co dnešní obvykle více jádrové procesy dokáží. [34]

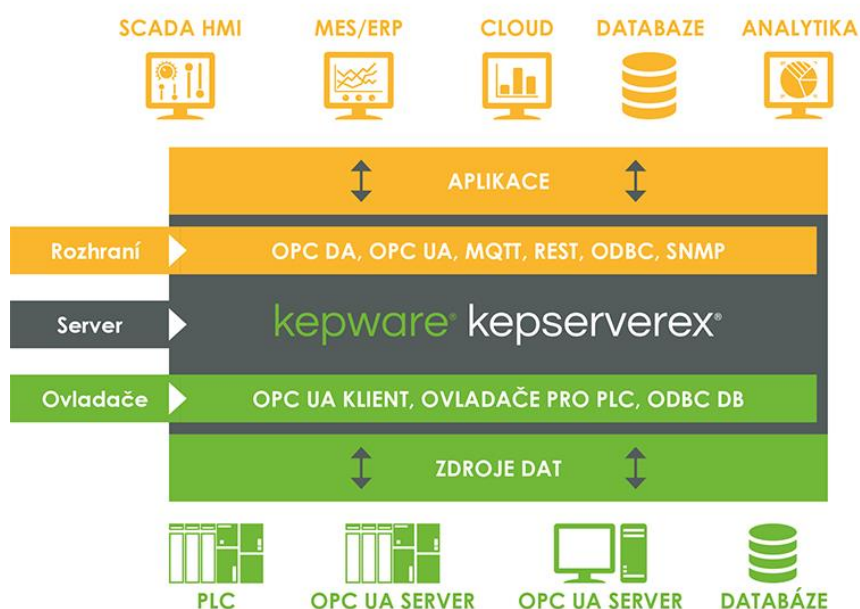
Co se týče analýzy a vizualizace výsledků máme možnost pracovat jak s binární formě, tak i s obrázkem. Jednotlivé mezivýsledky lze ukládat a do obrázků lze libovolně zapisovat. Je možné přidávat text, tvary (jako čtverec, elipsy a podobné), obarvovat hrany kontur různými barvami, například pro rozlišení špatně zobrazeného dílu apod. Uživatel je v podstatě omezen pouze okraji obrázku. Tyto obrázky lze prohlížet vývojář přímo ve vývojovém prostředí TwinCAT, v záložce *ADS Image Watch* a lze je také zobrazit koncovému uživateli přes vizualizaci v TwinCAT HMI. [34]

8 Komunikační protokoly

Komunikačních protokolů se objevují spousty. Od těch starších jako je PROFIBUS, CAN, Modbus a další se dnes používají ve stále menším množství. Snaha v průmyslu je vše standardizovat, pokud možno používat co nejvíce podobné komunikační systémy. Z tohoto důvodu se v dnešní době stále více používají protokoly, které využívají ethernetového propojení. Mezi takové patří PROFINET, OPC/OPC UA, EtherNET/IP, nebo Modbus TCP. Několik z těchto systému je zběžně popsáno níže.

8.1 OPC server

Server OPC slouží pro výměnu dat mezi různými průmyslovými systémy. Je to univerzální komunikační platforma. V průmyslu se používá pro sdílení dat do cloudového úložiště, nebo jako komunikační protokol pro spojení programů virtuálního zprovoznění. OPC servery mají několik standardů jako je OPC Security/Common Definitions/Complex data, APC Data Access, OPC Alarms and Events, OPC Historical DA, OPC XML DA, OPC Commands, OPC Batch, OPC Data Exchanger. Dva základní typy jsou popsány v podkapitolách níže. [35]



Obrázek 23) OPC server

OPC DA

Celým názvem je OLE Process Control Data Access. Jedná se o jeden z nejpoužívanějších standardů OPC. Zaměřuje se n-spojité charakter datové komunikace mezi koncovými zařízeními jako je PLC, DSC a klientskými aplikacemi jako jsou uživatelská rozhraní nebo datové rozhraní HMI/SCADA. Tento způsob komunikace se zabývá pouze komunikací v reálném čase. [35]

OPC UA

Aby bylo možné v budoucnu zvládnout veškeré požadavky průmyslu, musí být průmyslové sítě schopny vyhovět stále rostoucím nárokům. Společnosti budou potřebovat konvergentní sítě fungující na principu od zařízení k zařízení (end-to-end) včetně zabezpečení, které jsou flexibilně škálovatelné, dynamicky přizpůsobivé a schopné pojmout velké množství uživatelů-přirozeně, s nejrychlejší možnou dobou odezvy. OPC UA je klíčovou součástí. Otevřený neutrální komunikační standard umožňuje bezproblémovou

komunikaci s aplikacemi třetích stran a lze jej flexibilně přizpůsobit tak, aby splňoval specifické požadavky. [35]

Díky svým sémantickým funkcím OPC UA podporuje více než jen přenos dat. Obsahuje také unikátní informační model. Ověřené bezpečnostní mechanismy, jako je autentizace, autorizace a šifrování, zaručují bezpečné připojení. Pro OPC UA jsou také připravovány mechanismy podpory PROFI-safe komunikace, které umožňují standardizovanou bezpečnostní komunikaci mezi stroji související s bezpečností strojů. OPC UA lze také integrovat do existujících sítí průmyslového ethernetu a může tak běžet na stávající infrastruktuře PROFINET bez snížení výkonu. [35]

Výrobci zařízení integrují OPC UA server do svých produktů čím dál častěji. Běžně už to jsou třeba PLC, CNC systémy, pohony, routery apod. V OPC UA serveru najdeme často už předpřipravené hodnoty, které můžeme ihned začít číst OPC UA klientem.

Velkou výhodou OPC UA je skutečnost, že pro svůj provoz už nepotřebuje DCOM knihovny společnosti Microsoft, a díky tomu lze OPC UA provozovat i na jiných operačních systémech, jako je iOS, Android, Linux a další. Je tedy mnohem univerzálnější a netrávíte hodiny nastavováním OPC komunikace mezi počítači. Komunikace je také robustnější a bezpečnější, integrované funkce pro autorizaci a šifrování dat zvyšují bezpečnost dat na maximum. [35]

8.2 EtherNET/IP

Protokol TCP a uživatelský datový protokol (UDP) jsou základními komunikačními protokoly internetu. EtherNet / IP využívá TCP port pro tzv. *Explicitní zasílání zpráv*. To znamená, že systém odešle data klientovi v reakci na konkrétní požadavek. EtherNet/IP využívá TCP / IP protokol, který explicitně spravuje propojení mezi klientem a serverem. Část TCP, která je základem sítě TCP / IP, pomáhá fragmentovat datové pakety tak, aby datové zprávy dosáhly svého cíle. Část IP se zabývá pouze pakety. TCP umožňuje dvěma hostitelům navázat spojení a realizovat výměnu dat. TCP zaručuje včasné doručení dat a také, že pakety budou doručeny v pořadí, v jakém byly odeslány. [36]

EtherNet / IP využívá port UDP pro *implicitní zasílání zpráv* – systémová komunikace odesílaná z přednastavených paměťových míst do kontroléru nebo jiného klienta v předem naplánovaném intervalu. Taková komunikace je mnohem rychlejší než explicitní zasílání zpráv a jednosměrný datový přenos UDP (bez ověřování příjmů) zjednodušuje cyklické aktualizace systému. [36]

8.3 EtherCAT Device protocol

Protokol EtherCAT se svým unikátním funkčním principem – zpracováním datagramů za chodu – se za posledních deset let stal jedním z nejvíce rozšířených komunikačních standardů mezi průmyslovými sběrnici založenými na Ethernetu. Protokol EtherCAT, také nazývaný EtherCAT Device Protocol – EDP, je určen pro „hard real-time“ komunikaci mezi řídicím systémem a proměnným počtem zařízení, jako jsou vzdálené vstupy a výstupy, servoměniče, proporcionální ventily nebo komunikační brány (gateway), k další sběrnici. Sběrnice EtherCAT s protokolem EDP podporuje velmi krátkou dobu cyklu (až v řádu desítek mikrosekund), maximálně efektivně využívá šířku pásma a umožňuje dosáhnout vysokého stupně synchronizace mezi zařízeními propojenými v síti (jitter je v řádu stovek nanosekund). [37]

8.4 EtherCAT Automation protocol

Protokol EAP je zaměřen na komunikaci mezi řídicími systémy. Umožňuje jejich autonomní chod, často při použití různě dlouhých řídicích cyklů. V tomto případě není optimální komunikace typu master-slave, kterou používá sběrnice EtherCAT s protokolem EDP. Místo toho EAP přebírá komunikační model publisher-subscriber (metoda Pushed Data

Exchange). Každé zařízení komunikující protokolem EAP, které bude poskytovat data, je nadefinováno jako publisher a posílá data do sítě. Jedno či několik zařízení nadefinovaných jako subscriber může zachytit ethernetový paket odeslaný zařízením stanoveným jako publisher a přijatá data použít. Data mohou být publikována cyklicky (s periodou odpovídající době cyklu zařízení publisher nebo jejímu násobku), popř. se data odešlou až po změně jejich hodnoty. Způsob odesílání dat lze definovat jako unicast, kdy je k jednotce publisher určeno jedno konkrétní zařízení subscriber, nebo jako multicast, kdy je nadefinována konkrétní skupina zařízení subscriber, nebo broadcast, kdy publisher odesílá data do celé sítě. [37]

Publisher odesílá data společně s dalšími šestnácti bity, jejichž číselná hodnota je aktualizována při odesílání. Zařízení, které data přijímá (subscriber), může dané číslo přečíst a zkontrolovat, zda se neztratil některý z telegramů. Navíc každý příjemce (subscriber) může použít vlastní pomocný šestnáctibitový registr, aktualizovaný v každém cyklu PLC nově příchozími daty, a odečítat dobu od poslední aktualizace dat. Díky těmto diagnostickým proměnným mohou zařízení komunikující protokolem EAP kontrolovat vstupní data na úrovni komunikace v reálném čase a správně reagovat na chyby v komunikaci. [37]

8.5 PROFINET

Komunikace PROFINET a PROFIBUS jsou deterministické. To umožňuje podporu automatizačních systémů s předem definovanými strukturami vstupů/výstupů a jejich přesné načasování. PROFINET může také zajišťovat izochronní výměnu dat v reálném čase (IRT). IRT v podstatě využívá velmi přesné časové hodiny PROFINETu, aby upřednostňoval průchod některých typů datového provozu. IRT vyniká v náročných aplikacích, jako je řízení pohybu a aplikace, které vyžadují deterministický provoz než provoz v reálném čase. Při výměně dat v reálném čase jsou doby cyklu sběrnice kratší než deset milisekund. Naproti tomu k výměně dat IRT dochází během několika desítek mikrosekund. [36]

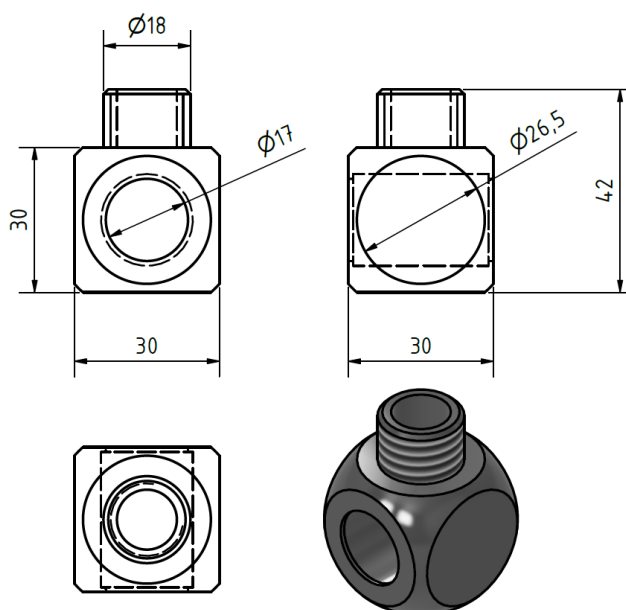
Například pomocí protokolu PROFINET v procesu balení a označování se přenáší data během plnění lahví. Přenos je tak rychlý, že je schopen zajistit, aby lahve byly naplněny na správnou úroveň za méně než sekundu. PROFINET může také detekovat, kvantifikovat a upozorňovat operátory na jakékoli anomálie v procesu plnění a v případě havárie okamžitě vypnout plnicí procesy. [36]

9 Představení praktické části projekt

Zadáním praktického úkolu bylo, že po dopravníku přijíždí jednotlivé součástky, které jsou nahodilého pozicování a natočení tak, že jsou na dopravník umístěny závitovou částí směrem vzhůru. Tyto součásti jsou odebrány z dopravníku odkud je odložen na měřicí stanici, kde je zjištěna pozice otočení. V poslední části odložíme součást, u které již známe natočení do zakladače tak, že jsou otvory v kostce vždy stejně orientovány.

9.1 Měřená součást

Jedná se o kostku tvaru krychle o rozměrech třicet milimetrů, na kterém je válcová část se metrickým závitem M18. V krychli je průchozí otvor který je ve středu jedné ze stěn krychle a zároveň je kolmý na osu závitu v horní části. Hrany všech rohu jsou zaobleny. Na obrázku 24 je výkres se základními rozměry této součásti a je zde i vyobrazen 3D pohled součásti.



Obrázek 24) Měřená součást

9.2 Systémový rozbor

Optické hledisko

Pro snímání kamerou je potřeba správně zvolit typ kamery a připravit scénu tak, abychom dosáhli co nejlepšího výsledku. To znamená vytvořit obraz, který bude mít vysoký kontrast stran, které nás pro snímání zajímají. Nejdůležitější věcí, abychom toho dosáhli je zvolit správné osvětlení. Poněvadž budeme v programu dále zjišťovat pomocí kamery, zdali je natočená součást otvorem ke kameře, nebo ne, je pro nás prioritní dosáhnout co nejlepšího rozlišení právě v této části. To nám však ovlivňuje také okolní osvětlení, které nám do scény může přivést stíny a osvětlit části, které bychom chtěli mít neosvětlené.

Druhá neméně podstatná část je zvolení správného typu kamery a objektivu. Kamery jsou v základu dvojího typu, a to černobílá a barevná. Ve většině případů se volí černobílá kamera, barevnou volím jen v případě, kdy nedokážeme docílit dobrého kontrastu pomocí barevné kamery.

Poslední důležitá součást optické sestavy je volba objektivu, výběrem těchto objektivů je popsána v rešeršní části (kap. 3.3 Objektiv). Pro tento úkol bude nejspíš při volbě objektivu zhodnotit typ závitu na kameře, nejspíše by bylo voleno standartního objektivu, protože je ekonomicky nejvýhodnější a jeho parametry jsou více než dostačující.

Komunikace

Ve výrobním systému, kde se nachází více systémů je důležité naplánovat si který systém bude řídit celý proces. Pro jeho výběr je ideální, pokud by to byl systém, který je spojený s co nejvíce systémy. Dalším důležitým parametrem je zohlednění, jakou rychlostí je potřeba komunikovat a jaké jsou možnosti komunikace.

Vedení kostky

Volbou vedení kostky na dopravníkové páse se odvíjí od taktu vkládání kostek, množství kostek vložených najednou a další provozní podmínky. Základním rozdělením nakládání na pás je, zda vkládá kostky obsluha, nebo stroj. Špatnou volbou vedení kostky může vést k zaplnění dopravníkového pásu do stavu kdy může dojít k nečekaným stavům jako zaplnění dopravníku do takového stavu, že se kostky zastaví ještě před najetím do přesné pozice.

Pokud ukládá obsluha, je vhodné systém navrhnout tak, že naloží několik kostek, které po dopravníku putují na páse do míst manipulace robotu. Jak dojedete k místu manipulace PLC zastaví pás, robot najede s kamerou nad dopravník a nasnímá plochu dopravníku. Pomocí tohoto snímku zhodnotí pozice a natočení n kostek, které se na páse právě vyskytují. Tyto informace poté postupně posílá robotu a ten je odebírá.

Druhou možností je, že jsou kostky na dopravník vkládány pomocí manipulátoru nebo stroje. U tohoto systému se předpokládá že bude kostky na dopravník vkládat v určitém taktu a bude se tak nacházet na dopravníku vždy přesně daný počet a pravidelně se přidávají další množství komponent po předem stanoveném čase. Takto vytvořený systém poté může mít pouze polohování mechanickým skluzem, který navede součást do přesného místa, kde by mohl být pouze snímač přítomnosti, zda je robot na místě či nikoliv.

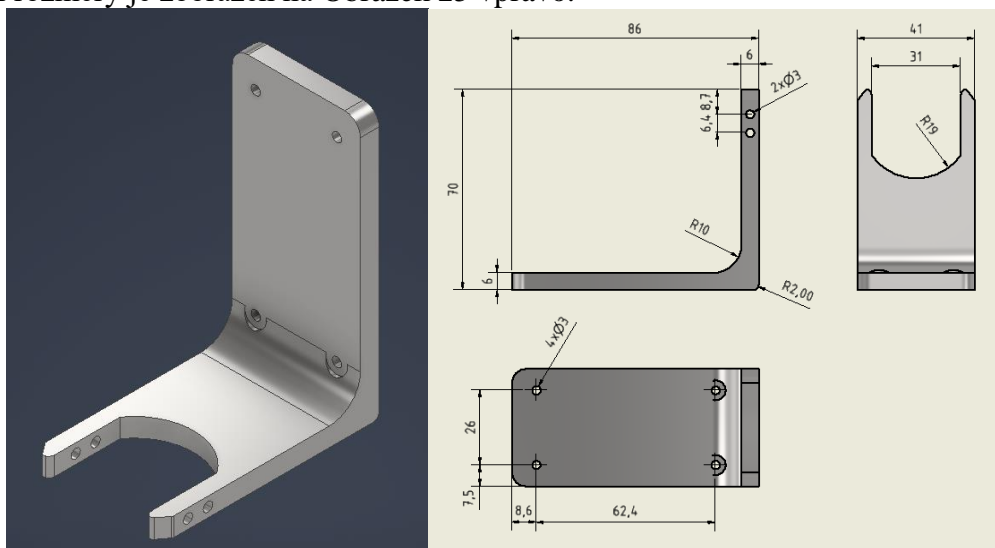
10 Virtuální zprovoznění pracoviště

Tato kapitola se věnuje postupu virtuálního zprovoznění navrženého robotického pracoviště. První část byla věnována vytvoření komponent, které poté budou potřeba při vytváření virtuálního modelu, k tomu byl použit program Autodesk Inventor Professional (viz kap. 9.1). V dalším kroku bylo připraveno pracoviště v Programu RobotStudio (viz kap. 9.2) a pracoviště bylo virtuálně zprovozněno a vytvořen program pro robot. V poslední části byl připraven program v prostředí TwinCAT od firmy Beckhoff (viz kap 9.3). V tomto kroku byl připraven program MAIN a algoritmus pro zpracování obrazu fotografií, které byly pro virtuální přípravu vytvořeny pomocí telefonu předem. V posledním kroku byli připraveny i obrazovky pro obsluhu. Následně bylo pracoviště virtuálně zprovozněno za pomoci OPC serveru a otestována funkčnost, pro použití v reálném pracovišti.

10.1 AUTODESK Inventor Professional

10.1.1 Držák pro kameru

V programu Autodesk Inventor Professional bylo zapotřebí nakreslit držák pro kameru, díky kterému bude poté připevněn na robot. Držák je tvaru písmene L, je to z toho důvodu, že na jedné straně je uchycení k robotu, které je kolmé na osu Z koncové části. Druhá strana držáku je na ploše, která je rovnoběžná s osou Z efektoru. Na této straně jsou otvory pro šrouby k uchycení kamery. Tento držák pro kameru byl poté vytisknut na 3D tiskárně. Pohled do Autodesk Inventor s držákem kamery je na Obrázek 25 vlevo. Výkres držáku se základními rozměry je zobrazen na Obrázek 25 vpravo.

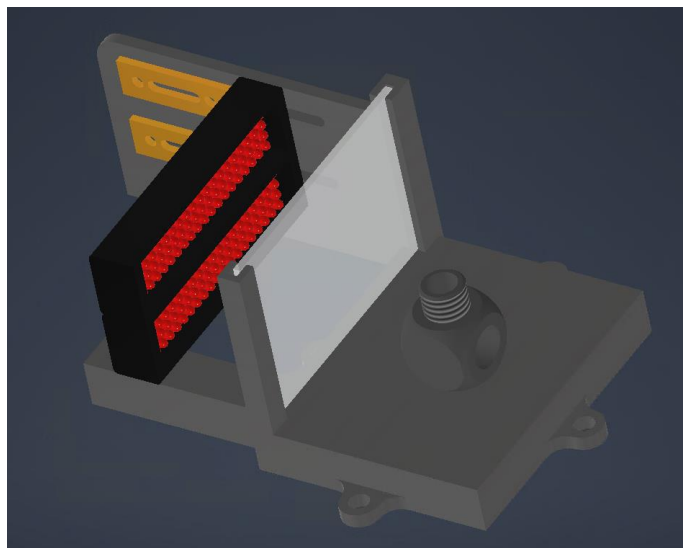


Obrázek 25) Držák pro kameru

10.1.2 Měřicí stanice

Pro místo, kde potřebuji měřit jsem si zvolil pracoviště s osvětlením Back-light, jak vyplývá z řešeršní části, pro co nejdokonalejší osvětlení průchozích otvorů a venkovních hran součásti. Osvětlení bylo tedy vytvořeno za pomoci dvou LED svítilen, které jsou v zadní části stanice, a difuzoru, který slouží pro rozložení světla na celou plochu difuzoru. Pro správné seřízení vzdálenosti světla je zde vedení, na kterém jsou uchyceny tak, aby se dala nastavit jejich vzdálenost od difuzoru. Pokud by byl zdroj světla příliš blízko byla by snímána kontura osvětlené části mezi světly, Pokud by byl zdroj světla příliš daleko od difuzoru mohl by být nedostatečně osvětlen.

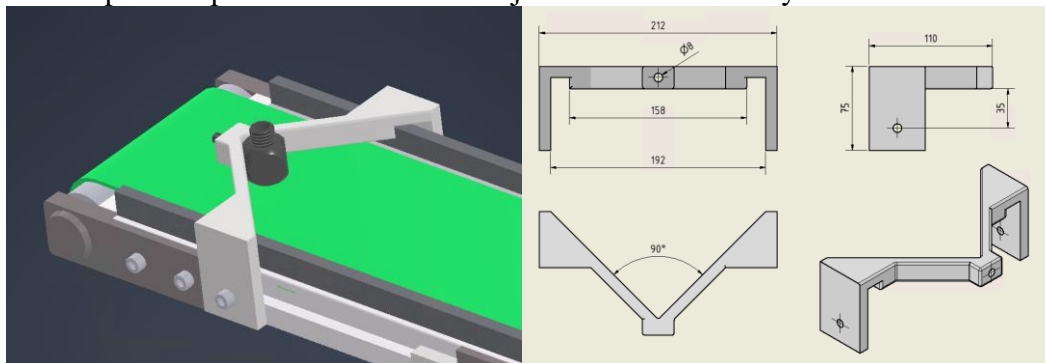
Na této stanici jsou čtyři úchyty pomoci, kterých ji lze ukotvit na stůl na pevné místo v několika pozicích. Náhled na virtuální sestavu měřicí stanice, včetně měřené součásti je zobrazen na Obrázek 26.



Obrázek 26) Měřicí stanice

10.1.3 Vedení pro kostku

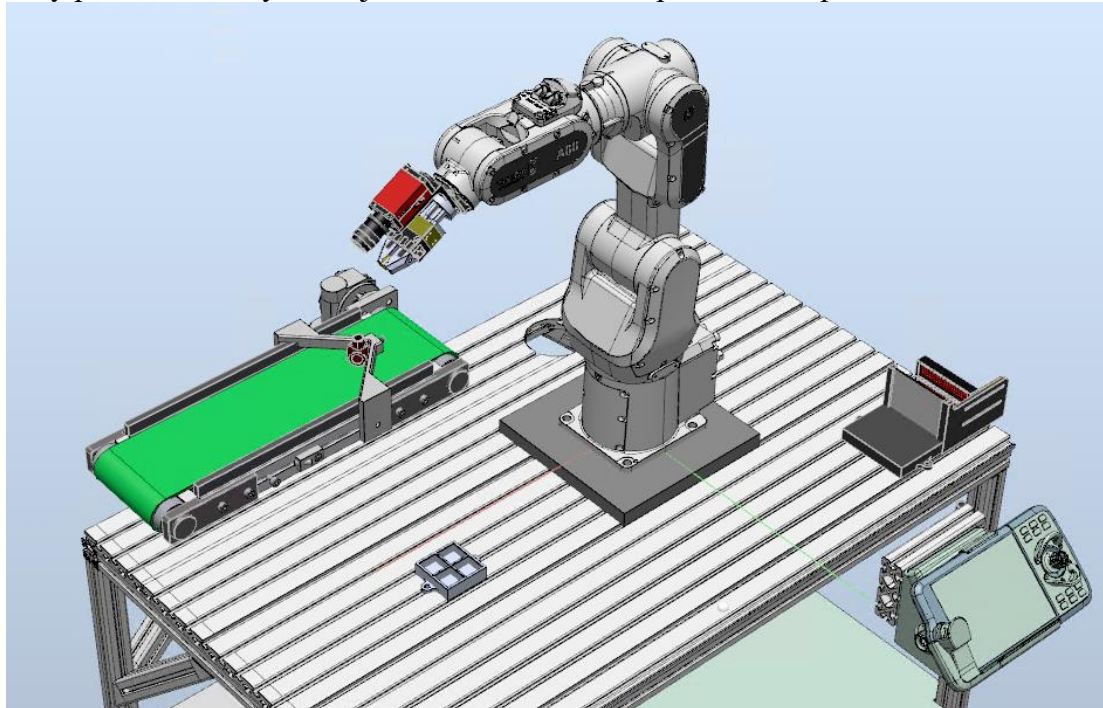
Vedení kostky na páse bylo zvoleno pomocí vedení ve tvaru V. úhel mezi jednotlivými rameny svírá devadesát stupňů. Je to z důvodu přesného polohování součásti. Součást přijede nahodile orientována v náhodné části pásu a je díky vedení dopravena na přesnou polohu, odkud je poté robotem odebírána. Vedení je uchyceno po stranách pomocí dvou šroubů a rovněž leží svojí horní částí na stranách pro vedení. V zadní části je předpřipravený otvor pro našroubování indukčního senzoru pro snímání, zda je měřená součást přítomna a sedí v přesné pozici pro odebíráání. Na Obrázek 27 vlevo je znázorněn model vedení na modelu dopravníkového pásu. V pravé části Obrázek 27 jsou základní rozměry vedení.



Obrázek 27) Vedení kostky na dopravníku

10.2 Robot Studio

Tato část se věnuje virtuálnímu zprovoznění pracoviště v programu Robot Studio od firmy ABB. V tomto programu bylo připraveno virtuální model pracoviště. Vloženy součásti, které byly buď naimportovány, nebo nakresleny v programu AUTODESK Inventor Professional. Poté byli zaměřeny souřadné systém WorkObjectů do nich byli vloženy body jako cíle manipulace. Následně byli napsány logické operace pro manipulace se součásti, které jsou pojmenovány path_00 až Path_40. Pak byl sepsán program MAIN, který celý algoritmus řídí. V poslední části přípravy virtuálního modelu byla spuštěna simulaci a nastaveny poslední detaily. Níže je zobrazeno navržené pracoviště v prostředí Robot Studio.



Obrázek 28) RobotStudio pracoviště

10.2.1 Pracovní objekty a body manipulace

Díky správně připraveným pracovním objektům lze snadno a rychle reálně zprovoznit robotizované pracoviště, protože stačí zaměřit méně pracovních bodů. Ostatní body jsou odsazeny o předem stanovenou hodnotu pozice a natočení podle virtuální přípravy.

V tomto projektu bylo použito celkem tři pracovních objektů, a to pro každé zařízení na stole. Prvním z nich je WobjConveyor, které je použito jako místo pro odběr komponenty, která přijede na dopravníkovém páse, kde je poté přesně polohována. Dalším je WobjMericiStanice, která je určena pro zaměření bodů k odkládání a vkládání měřené komponenty na měřicí stanici a polohu robotu pro snímání kamerou. Posledním pracovním objektem je WobjZakladac, který má úlohu jednoduchého zaměření pozic pro odkládání do zásobníku.

10.2.2 Signály robotu

Dále musela být připravena komunikaci mezi PLC a robotem. Zejména kolik bude potřeba vstupů a výstupů a jakého typu. Program je připravený pro komunikaci pomocí několika bajtů, díky kterým komunikuje s PLC a několika bity, díky kterými řídí jednotlivá zařízení, jako například gripper. Komunikace s PLC je tedy tvořena dvěma vstupními a dvěma výstupními bajty. Nultý bajt je používán jako číslo pro komunikaci, kde každé číslo má speciální sdělení. Druhý bajt je rozdělen do několika bitů, které jsou rozděleny na jednotlivé digitální informace, například nultý bit prvního bytu je digitální signál, zda jsou bezpečnostní okruhy robotu v pořádku, dále první bit znamená, zda jsou spuštěné motory

robotu a tak dále. Použité signály jsou popsány v Tabulka 3) Signály robotu níže, případně v Příloha 51.

Název signálu	Typ signálu	Popis signálu
diGrippClosed	Digitální vstup	Informace o uzavření gripperu
diGrippOpened	Digitální vstup	Informace o otevření gripperu
doGrippClose	Digitální výstup	Povel pro uzavření gripperu
doGrippOpen	Digitální výstup	Povel pro otevření gripperu
CommunicationBYTE_0	Byte vstup	Byte pro řízení komunikace mezi PLC a robotem číslem
CommunicationBYTE_0_Q	Byte výstup	Byte pro řízení komunikace mezi PLC a robotem číslem

Tabulka 3) Signály robotu

10.2.3 Deklarování proměnných

Tato podkapitola se věnuje popisu proměnných, které byly použity v programu pro RAPID program k řízení robotu. Proměnné lze rozdělit na dvě části, pro lokalizaci bodů pro pohyb robotu a proměnné použité k řízení pracovního procesu. V projektu je tedy použito celkem pět konstant pro polohu robotu označené jako typu rotarget. Dále jsou to proměnné typu num (number neboli číslo), tyto proměnné jsou popsány v tabulce níže. Zápis těchto proměnných je vyobrazen v Příloha 5.

Název proměnné	Popis proměnné
OffX	Tzv. offset pozice pro krokování při vkládání do zásobníku v ose X.
OffY	Tzv. offset pozice pro krokování při vkládání do zásobníku v ose Y.
PosX	Pro zápis, do jakého pole v zásobníku je právě vkládán výrobek v ose X.
PosY	Pro zápis, do jakého pole v zásobníku je právě vkládán výrobek v ose Y.
PosFinalX	Celkový počet pozic zásobníku v ose X.
PosFinalY	Celkový počet pozic zásobníku v ose Y.

Tabulka 4) Proměnné Robotu

10.2.4 Popis programu MAIN

Program MAIN funguje jako stavový automat, který pro přepínání do jednotlivých stavů využívá vstupní signál „CommunicationBYTE_0“. Tento signál obdrží od PLC a přepne do jednotlivého stavu. Těchto stavů je zde několik, pokud jsme ve stavu nula, robot pouze čeká až dostane nějakou informaci o tom, co má udělat. Pokud budeme ve stavu pět, robot vjede do domovské pozice a pošle o tom informaci PLC. V moment, kdy robot bude ve stavu deset, nejprve pošle signál o tom, že začíná operaci, v tomto případě patnáct (začínám odebírat komponentu). Poté začne s manipulacemi a po dokončení pohybů pošle signál zpět o tom, že manipulaci dokončil. Podobně jsou poté i řešeny ostatní povely stavového automatu, tedy, že nejprve pošle signál, že operaci začal a poté i signál že manipulaci ukončil. Podrobný náhled do kódu pro řízení robotu naleznete v Příloha 51.

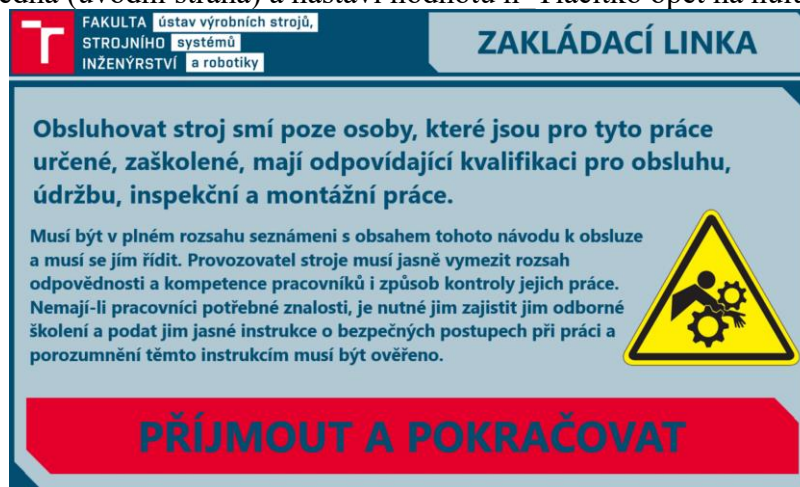
10.3 TwinCAT PLC

V prostředí TwinCAT bylo vytvořeno prostředí pro vytvoření řídicích algoritmů a programů pro řízení. Tento projekt je rozložen do čtyřech menších programů. Hlavní z nich je program MAIN, který celý proces řídí. Dalšími jsou programy POU_Automat, který slouží pro chod ve stavu automatického zakládání, program POU_Servis, tento program má dvě části, jedna slouží pro nastavení parametrů kamery, druhá ke spuštění robotu. Předposlední program je POU_Errors, kde jsou popsány jednotlivé chybové stavy. Posledním z programu je POU_Kamera, tento program slouží k nahrání obrazu z kamery a jeho vyhodnocení. Všechny více zmíněné programy jsou v následujících podkapitolách podrobněji popsány, výjimkou je program POU_Kamera, který je popsán v TwinCAT Vision

10.3.1 Program MAIN a prostředí HMI

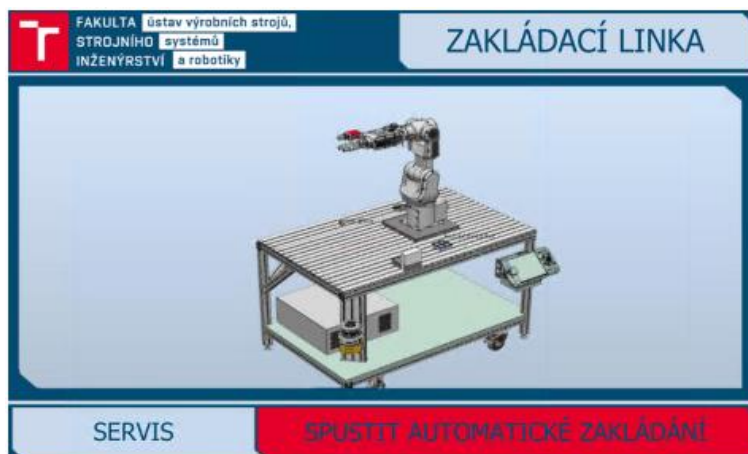
Je program, který je volán jako hlavní a od kterého se odvíjí veškeré řízení. Je složen jako stavový automat, který podle stavu (hodnota proměnné $n_Obrazovka$), zobrazuje právě danou obrazovku. Každý tento stav obsahuje hlavní seznam if podmínek, které reprezentují jednotlivá tlačítka na obrazovce, pokud je tedy $n_Tlačitko$ rovno nule, není spuštěné zmáčknuté žádné tlačítko. Každé další číslo uložené do proměnné $n_Tlacitko$ znamená splnění podmínek, který if podmínka obsahuje. Tento program byl vyexportován a je k zobrazení v Příloha 15.

Celý tento program ve stavu, kdy je $n_Obrazovka$ rovna nule a je tedy ve stavu nula. Na panelu HMI je zobrazena Upozornující hláška o tom, že tento stroj může obsluhovat pouze školená obsluha. Zde je pouze jedna možnost, a to pouze tlačítko „Přijmout a pokračovat“. Po stlačení tohoto tlačítka se přepne $n_Tlačitko$ do stavu 1, kde podle programu vykoná přechod do $n_Obrazovka$ jedna (úvodní strana) a nastaví hodnotu $n_Tlačitko$ opět na nulu.



Obrázek 29) Obrazovka školená obsluha

Ve stavu $n_Obrazovka$ rovno jedné je zobrazena úvodní stránka (Obrázek 30), kde se v prostřední části nachází obrázek pracoviště a ve spodní části této obrazovky se nachází dvě tlačítka, pro přechod do servisního stavu a pro přechod do stavu automatického zakládání.



Obrázek 30) Obrazovka úvodní stránka

Obrazovka servis kamera (Obrázek 31) slouží pro nastavení parametrů vyhodnocení obrázku kamery. Na obrazovce servis kamera si můžeme všimnout v levé části obrázek z kamery, který je zhotoven pomocí kamery a dále zpracován. Na obrázku si můžeme všimnout zelené kružnice, která značí hrany nalezené vnitřní kružnice a dvou červených kružnic, které doporučí parametry pro vyhodnocení maximálního a minimálního průměru. V pravé části je zobrazeno několik hodnot, které značí jednotlivé parametry, prvním z nich je odhadovaná pozice středu, tento parametr značí střed nalezené součásti, není vykreslen do obrázku je pouze orientační pro nastavení hodnoty „Přepis hodnot středu GVL“, kde je snaha posunout tento bod dovnitř otvoru součásti. Tato hodnota je znázorněna pomocí modrého bodu. Pokud se podaří umístit modrý bod do vnitřní strany kruhu, zakreslí se nám na obrazovce hranice vnitřního průměru a pomocí algoritmu dopočítány i hodnoty pro minimální a maximální poloměr hledaného otvoru. Hodnoty těchto kružnic jsou zapsány v pravé části uprostřed a jsou rovněž zakresleny červenou barvou do obrázku.



Obrázek 31) Obrazovka servis kamera

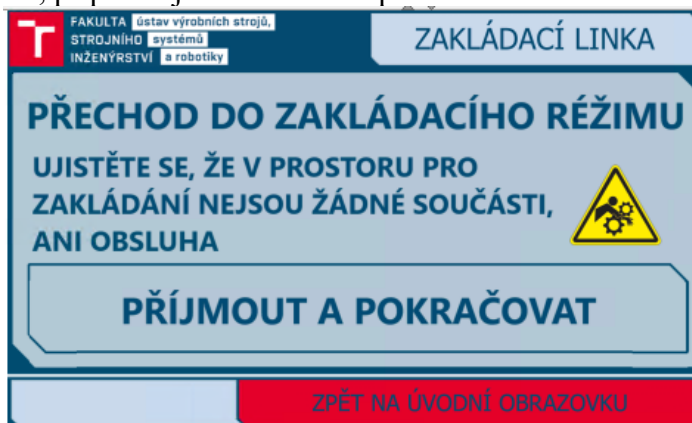
Obrazovka servis robot (Obrázek 32) slouží pro vyčtení základních hodnot zda jsou zapnuté motory, je robot v automatu a ostatní důležité stavy. V pravé části se nachází tlačítka pro spínání jednotlivých povelů pro robot. Obrazovka rovněž ve spodní části obsahuje tlačítka pro přepnutí na servis kamery, případně zpět na úvodní obrazovku. Program MAIN

vyhodnocuje, jaké je stlačené tlačítko a pokud žádné, nastaví hodnotu pro stav, že se provádí servis robotu a spustí program POU_Servis.



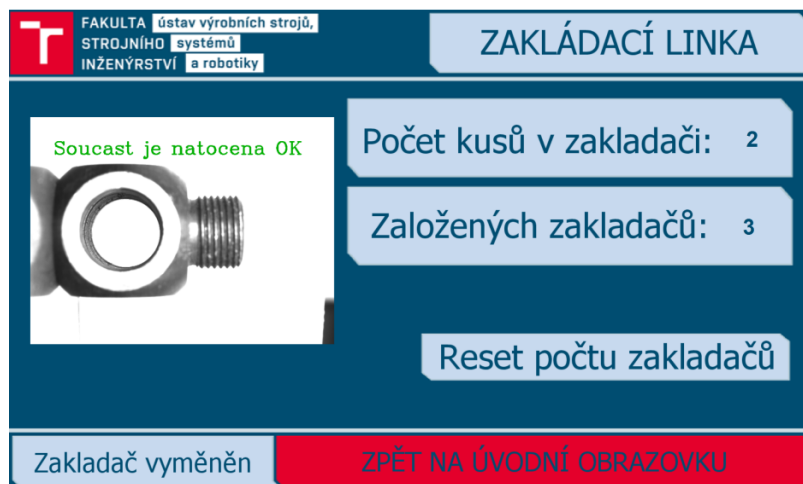
Obrázek 32) obrazovka servis robot

Při spuštění automatického režimu na obrazovce vyskočí potřebné potvrzení přechodu do automatického režimu, případně je zde i tlačítko pro návrat.



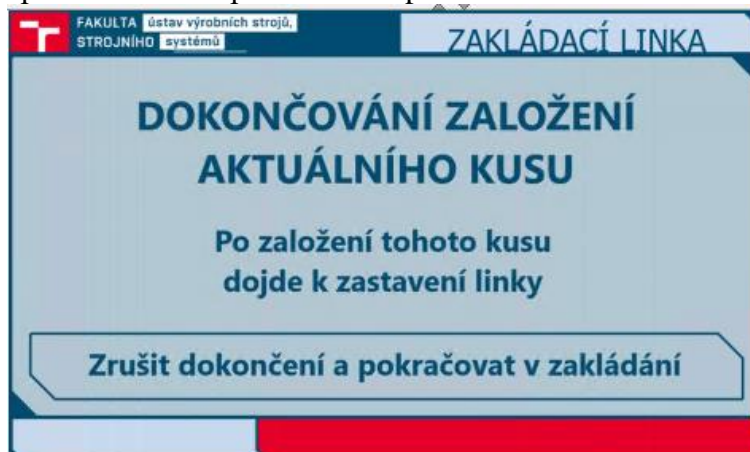
Obrázek 33) Obrazovka přechod automat

Při spuštění zakládání se objeví obrazovka Automat (Obrázek 34), kde v levé části obrazovky se objeví snímky z jednotlivých měření doplněné o text, zda je součást natočena správně, nesprávně, nebo že byla součást nenalezena. V pravé části se objevují hodnoty počtu kusů v zakladači a počty kusů již plně naložených zakladačů. Tato obrazovka obsahuje tři tlačítka, jedno pro resetování celkového počtu hotových zásobníků, tlačítko pro návrat do úvodní stránky a tlačítko které reprezentuje signál o výměně zakladače, při založení všech pozic zakladače.



Obrázek 34) Obrazovka Automat

Při návratu z automatického režimu se objeví hláška, že je potřeba počkat v založení posledního kusu, tento režim však může být stlačen omylem a je zde i tlačítko pro zrušení ukončení zakládání a pracoviště bude pokračovat v práci.



Obrázek 35) Obrazovka ukončení zakládání

10.3.2 Program POU_Automat

Tento program je volaný v MAIN, kdy je nastaven v automatického režimu. Program se skládá ze stavového automatu, který je složen z pěti částí. V prvním stav (10), je rozdělen do dvou IF podmínek, kdy jedna z podmínek vyhodnocuje, zda je komponenta na dopravníkovém páse a není povel pro stop linky. Pokud ano, dá povel pro robot, aby odložil součást na měřicí stanici, pokud ale není dá povel pro najetí do domovské pozice, druhá if podmínka je, pokud dostanu informaci od robotu, že provádí manipulaci přejdu do stavu 20, kde čekám na signál pro start měření, když přijde informace, že se může začít spustíme stav 30, kde na začátku rozsvítím osvětlení scény, vynuluje vyhodnocení z minulého měření a nastaví proměnnou pro spuštění *POU_Kamera* pro automatický režim a spustí program *POU_Kamera*, který vytvoří a vyhodnotí snímek. Po vyhodnocení vypne osvětlení. Zdali vyhodnotil součást jako špatně natočenou pošle signál o pootočení součásti a přepne dostavu 50, kde se čeká na odezvu o provádění natočení a čeká na spuštění nového snímání, kdy opět přepne do stav 30. Pokud však byli nalezeny správné rozměry otvoru, dá povel pro založení součásti do zásobníku a přepne do stavu 40, kdy čeká, zda robot začal manipulace, poté co tuto informaci dostane přepne zpět do stavu 10, který je popsán na začátku tohoto odstavce.

10.3.3 Program POU_Servis

V první části programu *POU_Servis* je věnován nastavení kamery. Je zde pouze časovač typu *TON*, který cykluje stále dokola a zapisuje tak povely pro kameru, aby vytvořila nový snímek, který je poté vyhodnocen. Délka tohoto cyklu byla nastavena na jednu vteřinu, obraz kamery i s přepočtem se obnovují na kameře každou vteřinu.

Druhá část se skládá z dalších dvou if podmínek, pokud je hodnota od tohoto tlačítka nastavena na *TRUE*, dej ji do hodnoty *FALSE* a spusť to funkci podmínky. V druhé části servis robot jsou podmínky pro resetování výstupu z první části servisu pro robot.

10.3.4 Program POU_Errors

Jedná se o stavový automat, kde argumentem pro změnu stavu je proměnná z *GVL*. Tento stavový automat v každém stavu zapisuje název chyby do proměnné *ErrorNazev* a popis dané chyby do *ErrorPopis*. Tyto proměnné i se číslem chyby (*ErrorID*) je propsán do obrazovky s názvem *Obrazovka Errors*. Výchozím stavem je stav nula, který je název a popis nevyplněný, každý další stav poté znamená jednotlivé chyby.

10.4 TwinCAT Vision

TwinCAT nemá v základním sestavení Vision. Toto rozšíření je potřeba doinstalovat. Pro správnou funkci Vision je potřeba také doplnit knihovny. To se provádí ve stromě projektu pod záložkou references. Na tuto složku stačí kliknout pravým tlačítkem a přidat knihovnu, která nese název TC3_Vision. Pokud se chceme podívat co tato knihovna obsahuje, můžeme ji lze v prostředí TwinCAT rovnou otevřít a prohlédnout si ji. Dalším krokem je potřeba si zobrazit Vision, který bývá při vytvoření nového projektu zneviditelněn. Po zobrazení je zde potřeba ještě vytvořit aplikace, díky kterým lze rozdělit kamerové systémy do několika skupin a mít tak mezi nimi dobrý přehled. Dalším krokem je vytvoření FileSource device, nebo GigE camera device, které jsou popsány níže.

10.4.1 Postup přípravy snímků pro virtuální zprovoznění

TwinCAT vision má možnost vyčítání nejen obrazu pomocí kamery (GigECamera), ale zvládne i použít obrázky, které má uložené v paměti. Ty volá pomocí funkčního bloku (fb_VN_FileSource). Nové verze TwinCAT mají pro tyto funkce funkci společnou. Jedná o funkční blok fb_VN_SimpleCamera. Tento blok tedy dokáže, jak vyčítat kamery z paměti, tak i z kamery.

Před vytvořením programu pro snímání obrazu byla potřeba vytvořit snímky, pomocí kterých, by byl program vytvořen. To lze v části pro virtuální zprovoznění dělat několika způsoby. Jedním z nich je vytvoření virtuálního modelu, vložení modelu do programu, kde se dají nastavit různé typy osvětlení (jako například herní engine, nebo jiné programy pro virtuální realitu, kde se snažíme napodobit skutečný stav) a poté vytvořením snímku obrazovky počítače. Dostaneme obraz, který pak v programu na úpravu fotografií upravíme na rozlišení kamery, formátu obrazu a podobné. Další možnost, která se více blíží reálnému zprovoznění je, pokud máme hotový kus, který máme za úkol vyhodnocovat, nebo alespoň jeho kopii vytvořenou pomocí 3D tisku. Když máme tuto součást, můžeme se pokusit nasimulovat prostředí, které se bude podobat tomu, co bude vytvořené podle navrženého projektu. Ke snímání takového obrazu lze použít jak průmyslovou kameru, která bude poté použita ke snímání i ve výsledném pracovišti, tak i mobilní telefon, kde pomocí fotoaparátu vytvoříme snímek, který bude poté také potřeba upravit, stejně jako v případě vytvoření snímku obrazovky.



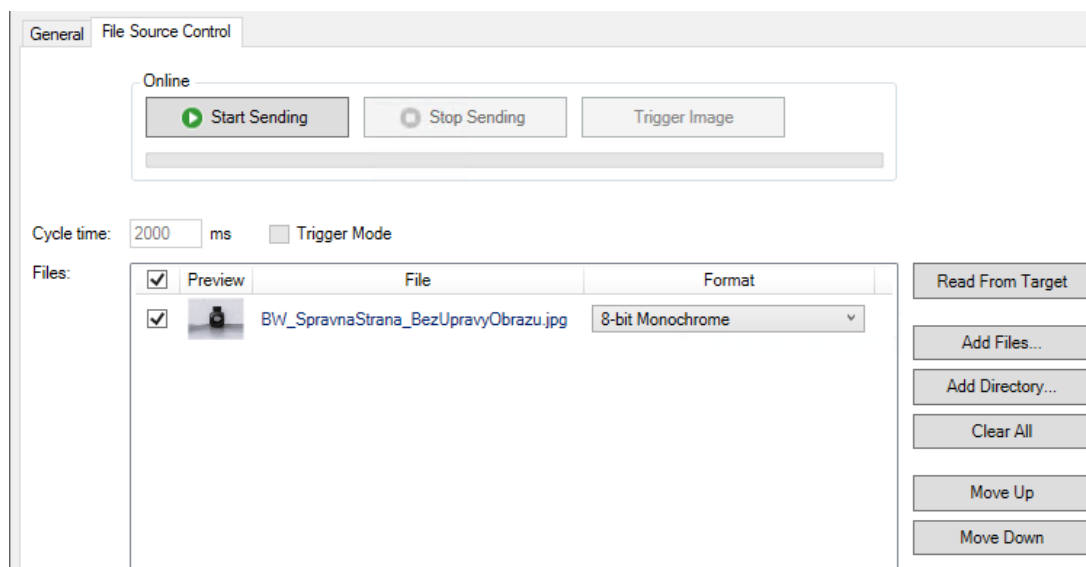
Obrázek 36) TwinCAT připravený obraz

10.4.2 Nastavení File Source

File source je vhodná pro předpřipravení programu na základu několika vytvořených snímků, ať už kamerou, telefonem, nebo snímkem obrazovky virtuálního modelu. Tyto obrázky jsou poté uloženy v průmyslovém počítači. Pozor, volání obrázků z paměti je pomalejší, než nahrávat obraz z kamery.

Do námi vytvořené „*aplication 1*“ z minulého kroku byla přidán nový item vybrali jsme typ „*File Source*“, který byl pojmenován „*FileSource1*“. Po vytvoření bylo potřeba ve *File Source Control* nastavit jaké obrázky budeme pomocí něj načítat a jakým způsobem. Nejprve tedy byli zvoleny obrázky. To se dá provést pomocí tlačítek „*Add Files*“, „*Add Directory*“ nebo pomocí tzv. *Drag and drop* (přetažením obrázku). Obrázky lze vkládat jakéhokoliv typu a lze i případně upravit formát, na černobílí 8-bit anebo barevný. Pro nasimulování prostředí jako by byla k systému připojena kamera bylo zvoleno *trigger mode*, na jejímž principu bude poté volán snímek z kamery. Toho je dosaženo zaškrtnutím volby *trigger mode*, pokud by zaškrtnutý nebyl, znamenalo by to kontinuální předávání snímků do PLC. Struktura tohoto nastavení pro tento projekt je zobrazena na Obrázek 37.

Pro ukázkou připojení kamery k PLC je popsána v kapitole **11.2.2 Připojení kamery k PLC**.



Obrázek 37) TwinCAT File Source

10.4.3 Adresování

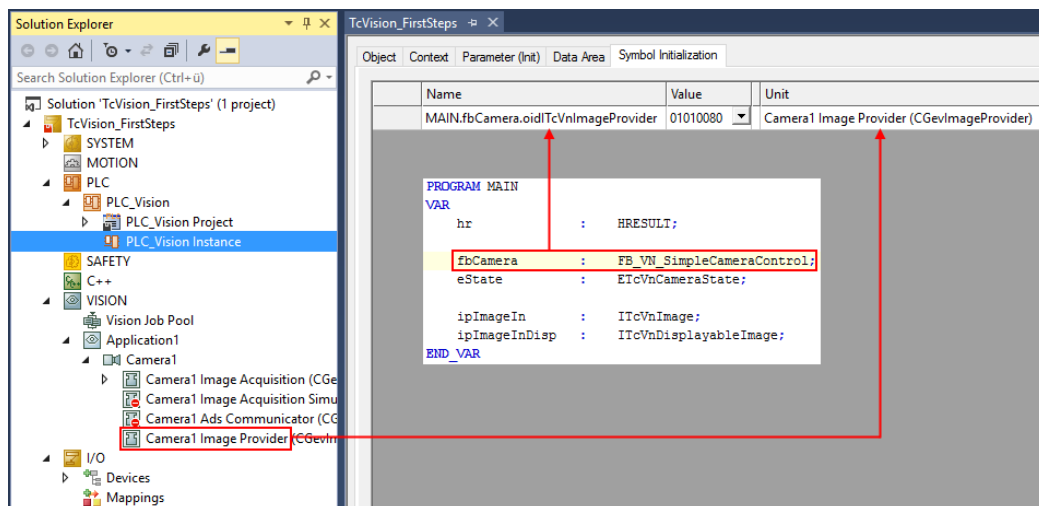
Pro zpracování obrazu v PLC je potřeba adresovat se na zvolený zdroj, odkud bude funkční blok obraz volat. Tyto funkční bloky jsou třech typů:

FB_VN_GevCameraControl, který je vhodný pro propojení pouze s kamerou.

FB_VN_FileSourceControl, ten jde spojit pouze s obrazem uloženým v počítači.

FB_VN_SimplecameraControl je novější systém proti dvěma předešlým, kdy jej lze propojit s oběma možnostmi zdroje obrazu. Jeho hlavní výhodou je, že stačí napsat pouze jeden program a při reálném uvedení do provozu jej adresovat na druhé zařízení, což nebylo dříve možné, protože předešlé dva způsoby používali odlišné funkční bloky.

Adresování se provádí ve stavu, kdy máme definovanou proměnnou, která je právě ze tří uvedených typů a vytvořené zařízení typu *GigE Camera*, nebo *File Source*. Samotné adresování poté probíhá v *PLC instance* ve stromě projektu, který otevřeme a přepojíme okno na *Symbol Initialization*. Cesta k tomuto adresování je zobrazena na Obrázek 10 Obrázek 38. Po adresování je potřeba obnovit konfiguraci, aby proběhl její zápis. Resetování konfigurace je potřeba provádět při každé změně nastavení *PLC_Vision_Instance*.



Obrázek 38) TwinCAT VISION instance

Při adresování kamery můžeme rovnou zkontrolovat jaký je stav kamery, zda je vše OK a mimo jiné je zde i možnost se podívat na živý obraz vystupující z kamery.

10.4.4 Popis programu POU_Kamera

Proměnné

Seznam a popis použitých pro část zpracování obrazu, je obsažen v Tabulka 5) Proměnné POU Kamera.

Proměnná	Typ proměnné	Popis proměnné
fbCamera	FB_VN_SimpleCameraControl	funkční blok pro vyčítání obrazu, je potřeba jej definovat v PLC Instance => symbol inicialization
eState	ETcVnCameraState	proměnná pro stavový automat kamery
Hr	HRESULT	slouží pro zaznamenání erroru (bez chybového procesu), pokud je 0 je vše ok, když ne, poslední trojčíslí popisuje chybu
hrFunc	HRESULT	hResault pro elipsu, stejné jako výše
ipImageIn	ITcVnImage	Proměnná, do které se načítají data (pro vstupní obraz)
ipImageRes	ITcVnImage	Proměnná, do které se načítají data (pro upravovaný obraz)
ipImageResDisp	ITcVnDisplayableImage	Vykreslení ipImageRes do ADS Image Watch, nebo HMI
stEllipse	TcVnRotatedRectangle	neznámá do které vrátí údaje o elipse, lze je pak vyčítat například stEllipse.aCenter pro pozici středu
Gvl.aCenter	TcVnPoint2_REAL	Očekávaný střed elipsy
Gvl.fRadius	REAL	Poloměr kolem středu pro hledání hran
ipContourPoints	ITcVnContainer	sub pixelový přesný obrys, pokud není vyžadován nastavit na 0
aColorRed	TcVnVector4_LREAL	červená barva
aColorGreen	TcVnVector4_LREAL	zelená barva
aColorBlue	TcVnVector4_LREAL	modrá barva
sText	STRING	Text pro přepis do obrázku

Tabulka 5) Proměnné POU_Kamera

Stavový automat kamery a získávání obrazu

Skládá se ze několika stavů. Celý program začíná ve stavu TCVN_CS_INITAL, ve kterém probíhá inicializace kamery, v tomto programu je spuštěna funkce začni získávat obraz z kamery. Po tomto kroku se spustí část TCVN_CS_ACQUIRING, který popsán níže, ve zkratce se vyhodnocuje obraz a po zpracování je tento stav přepnut do stavu TCVN_CS_STOPACQUISITION, ve kterém je hodnoceno, zda má proběhnout triggování kamery a vytvoření tak dalšího snímku, který přepne stav pro získání a vyhodnocení obrazu. Program tak koluje v této smyčce. Stav TCVN_CS_ERROR je zde pro důvod, že pokud nastane chyba celý funkční blok se resetuje na výchozí pozici, odkud začínal po spuštění.

Zpracování obrazu

Tato část je vložena do stavu TCVN_CS_ACQUIRING, kdy je funkční blok ve stavu získávání obrazu. Na začátku této části se nejprve spustí funkce pro nahraní obrazu. Následuje kontrola, zda při načítání obrazu nenastal nějaký problém a byl již nějaký obraz načten. Poté následuje část rozdělená do dvou stavů vyhodnocení, zda je snímek porovnáván pro servis, či vyhodnocovací účely v automatickém režimu.

V části pro servis kamery jsou dvě funkce pro hledání elipsy. Jedna slouží pro měření otvoru a druhá je určena ke hledání středu součásti, jako orientační body, kam nastavit výchozí hodnoty. Rozdíl mezi těmito funkcemi je ten že měření otvoru se hledá elipsa nejbližší středu a pro lokalizaci bodu je tomu obráceně. Dalším rozdílem je také poloměr hledání. Kdy lokalizování součásti je nastaven tak, aby zabral celý snímek, funkce pro hledání je nastavena v GVL a lze ji také měnit přímo ve vizualizaci. Po nalezení elipsy následuje vytvoření výstupního obrázku z toho vstupního a převedení jej do barevného formátu. Dále jsou vykresleny nalezené parametry. Vypočítání parametrů pro nastavení maximální a minimální kružnice pro vyhodnocení, které jsou poté vykresleny na obrazovku i vypsány jejich hodnoty. Obraz je poté uložen do paměti počítače i jako vizuální obraz, který jsme poté schopní zobrazovat v *ADS Image IN*.

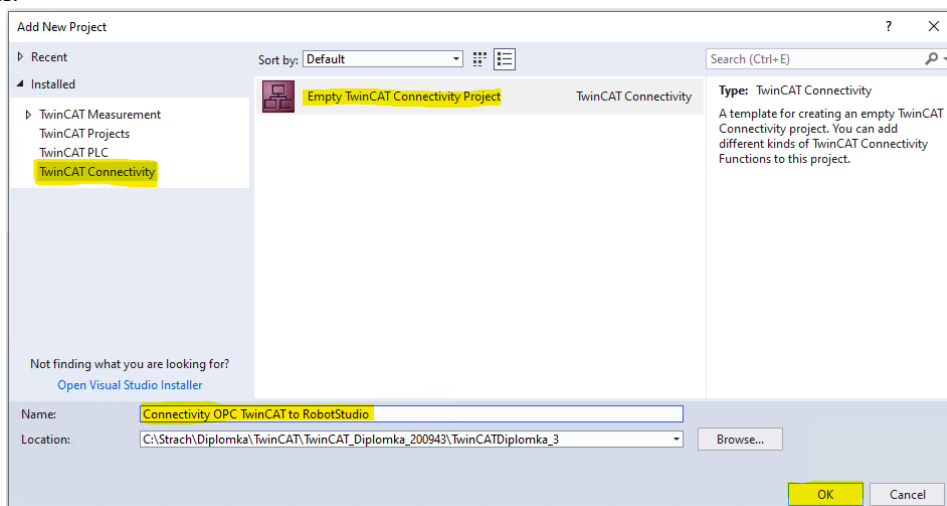
V části pro automatické režimu se nejprve vyhodnotí obraz, aby se zjistili parametry elipsy, které jsou uloženy do proměnné stEllipse. Dalším krokem je vytvoření kopie obrázku, která je převedena do barevného obrazu. Následuje vyhodnocení, tedy zda byla nalezena elipsa, která splňuje parametry správně natočené součásti, nebo zda nebyl, zde se určuje další podmínka, zda nalezené parametry byli nalezeny alespoň nějaké parametry, což značí o špatné orientaci součásti, nebo jsou parametry nulové, což nastane v případě odstranění součásti z prostoru scény. Každý ze zmíněných důvodu hodnocení, vykreslí do obrazu text a přepne stavový automat do specifického režimu. Následuje opět uložení obrázku a pokud uložení proběhlo úspěšně přepíše proměnnou pro verzi obrázku, díky které se tak ve vizualizaci přepíše. Obraz je znovu načten ve vizualizaci a pomocí funkce na konci programu je i zobrazen jako vizuální obraz, který jsme poté schopní zobrazovat v okně *ADS Image IN*.

10.5 Vytvoření OPC serveru v projektu

Zde je popis, jak bylo vytvořeno spojení mezi programy RobotStudio a TwinCAT.

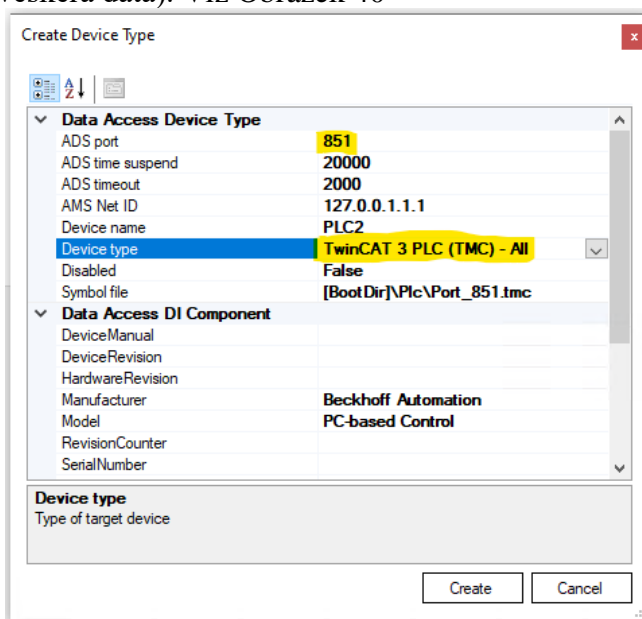
10.5.1 OPC UA na straně TwinCAT

TwinCAT umožňuje komunikaci pomocí OPC serveru vytvořit několika způsoby. Prvním a nejdůležitějším bodem je si aktivovat licenci TF6120 – TC3 OPC-DA. Jedním z nich je vložení projektu OPC serveru, který byl zvolen pro připojení v tomto projektu. Nejdříve jsme tedy vytvořili nový projekt, vybrali jsme ze záložky *TwinCAT Connectivity* typ *Empty TwinCAT Connectivity Projekt*, projekt pojmenovali a vybrali umístění projektu, kam se projekt uložil.



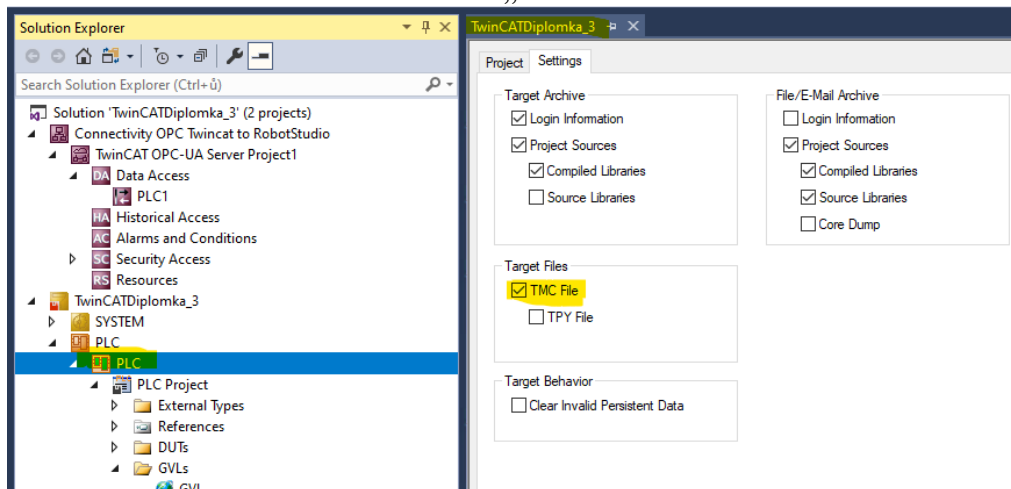
Obrázek 39) TwinCAT OPC Vytvoření projektu

Do vytvořeného projektu jsme vložili nový TwinCAT OPC-UA Server Projekt, který byl pojmenován „*Connectivity OPC TwinCAT to RobotStudio*“. Vytvořili jsme ve složce nový Device Type, zde se nastavují parametry pro komunikaci, jako je ADS port, ID sítě pro komunikaci, název pro komunikačního zařízení a typ zařízení (které informace a jakého typu bude odesíláno). Z těchto parametrů byl nastaven port komunikace, na 851 (tovární port twinCAT) a typ komunikace na TwinCAT3PLC(TMC)-all, (komunikace pomocí TMC spouborů a sdílet na síť veškerá data). Viz Obrázek 40



Obrázek 40) vytvoření OPC device

Tímto by, jsme měli nastavený OPC server, ale je zde ještě potřeba v našem PLC umožnit mu komunikovat pomocí TMC souborů, to se provede tak, že si otevřeme v našem projektu PLC a v nastavení odškrtneme možnost „TMC File“ viz. Obrázek 41.



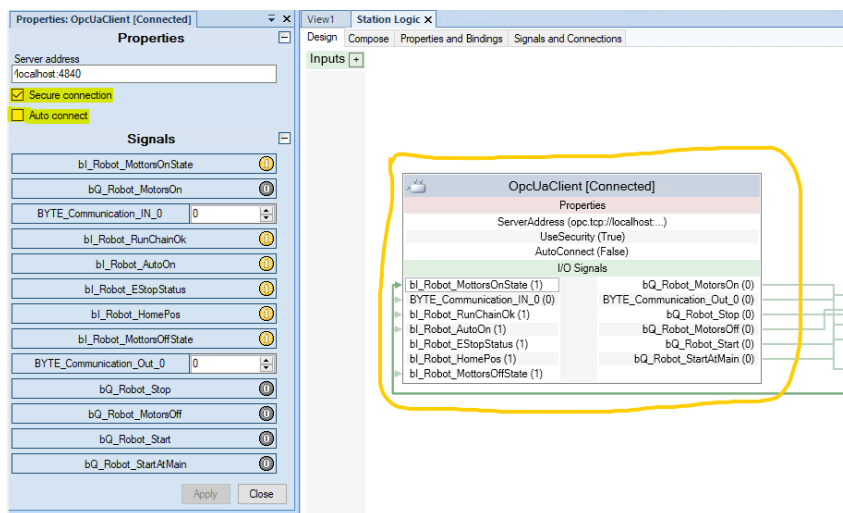
Obrázek 41) Povolení komunikace pomocí TMC File

Poté je potřeba aktivovat konfiguraci a připojit se k OPC serveru. Kde je dobré ihned po připojení aktivovat konfiguraci i OPC serveru, aby se nastavili i ostatní parametry.

10.5.2 Připojení k OPC UA pomocí RobotStudia

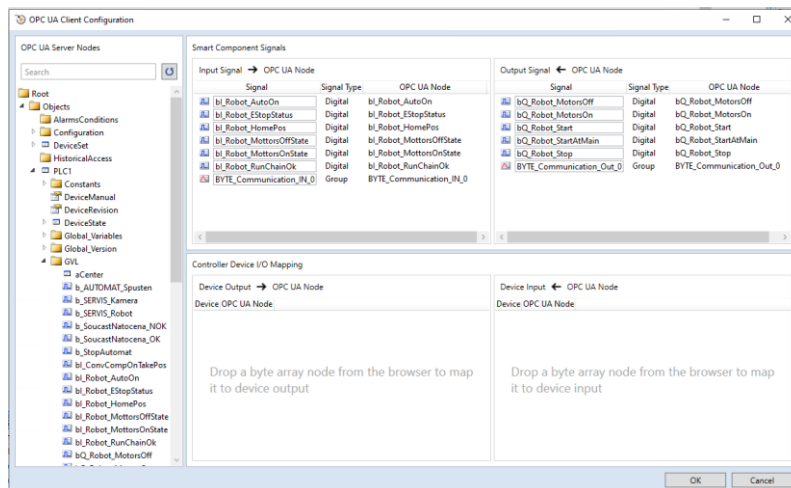
V prostředí RobotStudio se připojuje k OPC serveru pomocí Smart komponenty. Je tedy potřeba otevřít logiku stanice. Vložit zde novou Smart komponentu, s názvem OpcUaClient. Jako je uvedeno na

Obrázek 42 vlevo. V pravé části tohoto obrázku je její nastavení, bylo zvoleno vypnout automatické připojení, a to z důvodu změn v programu se při režimu automatického připojení hodnoty komunikace v RobotStudiosu nepřepisovali. Je zde ještě *Service connection*, který byl ponechán sepnutý.



Obrázek 42) RobotStudio OPC

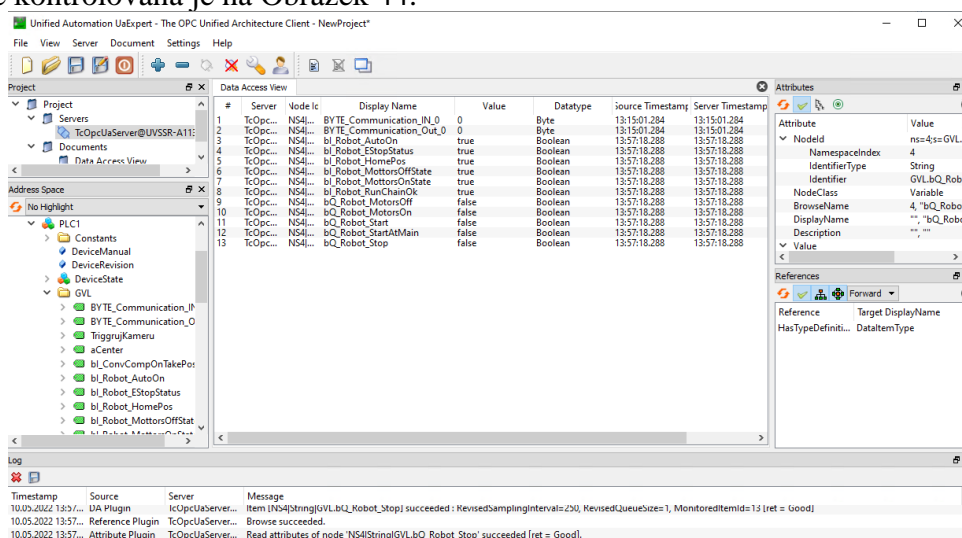
Dále bylo potřeba ve Smart komponentě nastavit, které signály má vyčítat, a které naopak zapisovat do OPC serveru. To se provádí konfigurací Smart komponenty, po otevření konfigurace. Z pravé části byli vybrány signály ze souboru GVL (global variable list) ty, které byli důležité pro komunikaci mezi řídicím systémem PLC a řídicím systémem robotu.



Obrázek 43) OPC UA Client Configuration

10.5.3 Kontrola komunikace virtuálního prostředí

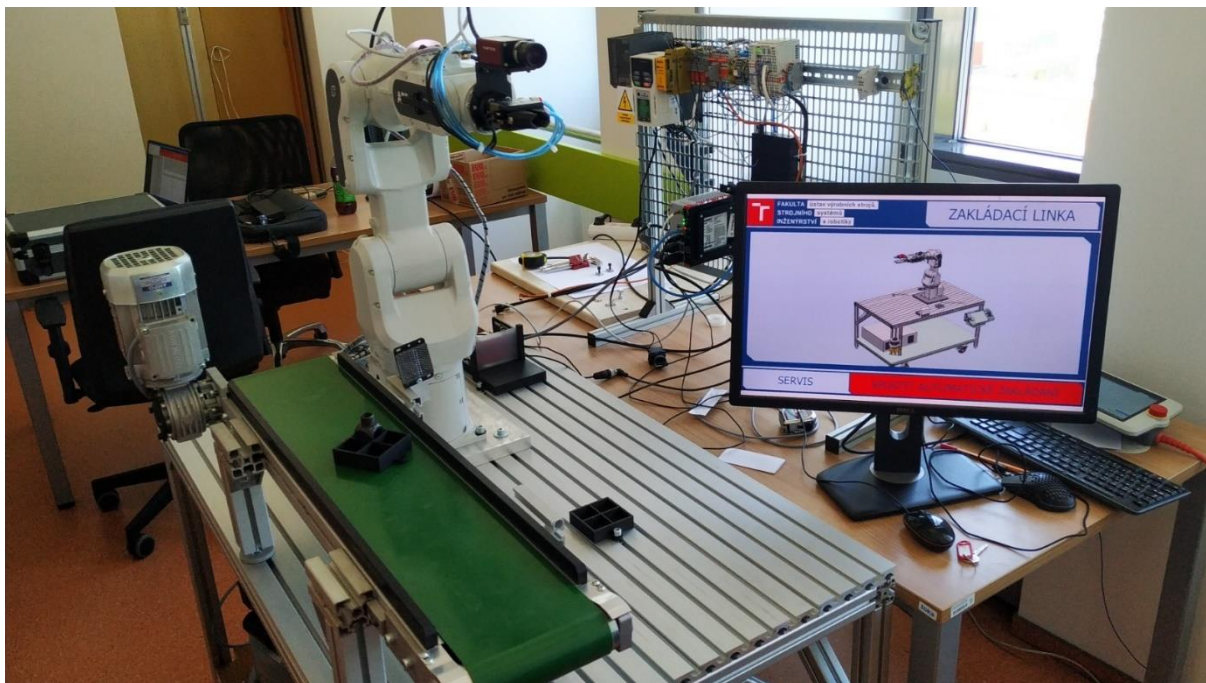
Kontrola komunikace probíhala na platformě *Unified Automation UaExpert*. Tento program slouží pouze pro vizualizaci a kontrolu přenášených dat. Tento program měl tedy pouze kontrolní funkci. Připojením k serveru bylo dosaženo tak, že jsme zde vytvořili projekt a zapsali do něj adresu našeho OPC. Po navázání spojení bylo ve výběrovém stromu v pravé části přetaženy signály, které bylo nutno kontrolovat. Náhled do programu, kdy byla komunikace kontrolována je na Obrázek 44.



Obrázek 44) Unified Automation UaExpert

11 Reálné zprovoznění

Kapitola reálné zprovoznění je věnována v první části popisu komponent použitých v robotickém pracovišti, kde je i zdůvodněný výběr. Další část je věnována potřebným změnám ve virtuálním modelu, připojením kamery a ostatní potřebné úkony ke zprovoznění pracoviště.



Obrázek 45) Reálné pracoviště

11.1 Hardware navržené robotické linky

Použité komponenty byly vybrány z dostupných zdrojů ústavu, ostatní komponenty bylo potřeba vytisknout na 3D tiskárně.

11.1.1 Optika

Kamera

Bylo voleno mezi dvěma typy kamery, a to mezi kamerou s barevným obrazem a kamerou s černobílým snímáním, která měla nižší rozlišení. Obě tyto kamery byly osazeny standartními objektivy. Pro úlohu vytvoření obrazu při režimu osvětlení back-light je dosaženo vysokého kontrastu, tím pádem odpadl důvod vybírat barevnou kameru. Dalším rozhodujícím faktorem bylo, jaká potřebná míra rozlišení kamery. Poněvadž operace nakonec směřovala směrem měření otvoru díry, bylo potřeba zvolit dostatečné rozlišení u kterého se zjistilo že i tento parametr je dostatečný u obou kamer. Nakonec byla vybrána černobílá kamera, a to z důvodu menšího množství dat pro uložení, a tím zkrácena i doba vyhodnocení obrazu, která se zkrátila na polovinu.

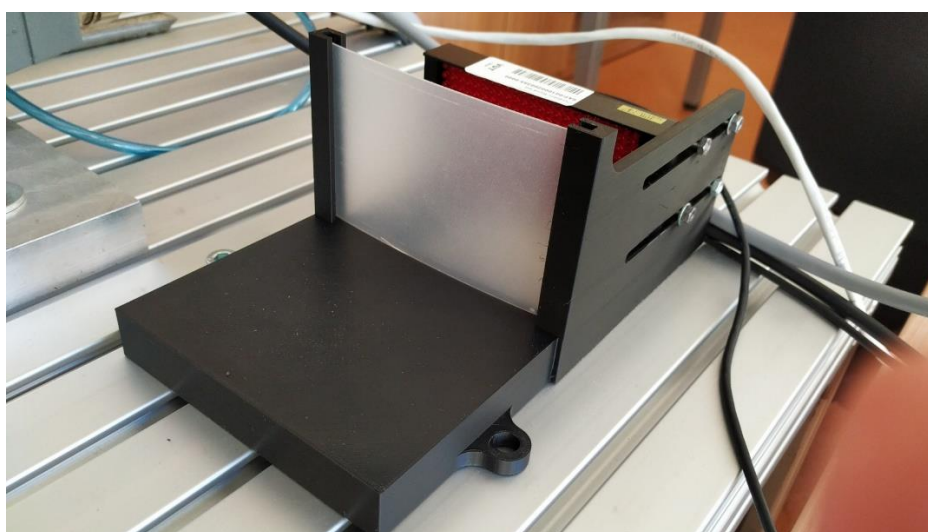


Obrázek 46) Použitá kamera

Měřicí stanice

Dalším důležitým faktorem pro správné vytvoření snímku je volba osvětlení. Zde se volilo mezi předním osvětlením s jasným obrazem, přední osvětlení s tmavým obrazem a osvětlením zadním. Při osvětlení z přední části objektu došlo ke zviditelnění obou hran přední části, tomu bylo snaha zabránit, a to z důvodu, aby byl otvor buď zobrazen anebo nikoliv. K tomu, jak vychází z rešeršní části je nejlepší použít osvětlení ze zadní části. V takovém osvětlení je zvýrazněn otvor a nehrozí zde možnost, že bude zobrazena i okolní hrana se kterou může být zaměněna.

V této stanici bylo použito pro osvětlení dvě LED plošné osvětlení. To osvětlovalo difuzor, který byl mezi zdrojem osvětlení a měřenou součástí. Byl zde i požadavek nastavení vzdálenosti zdroje osvětlení od difuzoru. Toho bylo dosaženo pomocí vodících otvorů v zadní části, ve kterých byla světla osazena. LED osvětlení má červenou barvu, což černobílá kamera ovšem vidí jako jakýkoliv jiné světlo, navíc literatura vádí, že červené světlo díky své frekvenci kmitání je lépe zachyceno snímači světla, které se používají v kamerových systémech.

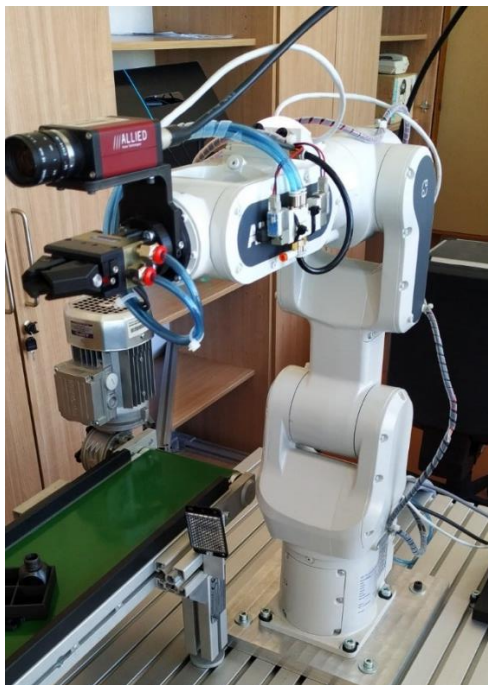


Obrázek 47) Použitá měřicí stanice

11.1.2 Součásti pro manipulaci s měřenou součástí

Robot

Robot pro tento projekt nebyl zvolen, ale byl předem stanoven, jedná se o kolaborativní robot firmy ABB, a to SWIFTI CRB 1100. Tento robot má šest os, nosnost čtyři kilogramy a na horní rameno lze uložit další půl kilogram. Kdyby bylo pro tento projekt mít možnost výběru nejspíš by se zvolil průmyslový robot, protože v pracovišti tohoto třízení se obsluha pohybovat nebude.



Obrázek 48) Robot ABB SWIFTI CRB 1100

Dopravník a vedení kostky

Zvolený dopravník je řízený pomocí motorového měniče, které lze propojit a ovládat pomocí BOOL signálů přímo z PLC. Měnič má možnost, jak řídit směr posouvání dopravníku (vlivem napájení tří fází), tak i jeho rychlost (frekvence výstupu). Tento způsob ovládání však nebyl pro tento projekt použit a je zde předpoklad že dopravník bude neustále v provozu a nebude zastavovat.

Pro jednoduchost reálného zprovoznění bylo na dopravníku místo odběru s vedením tvaru písmene V, jak tomu je ve virtuálním zprovoznění bylo nahrazeno odkládacím místem a byl poslán BOOL signál do PLC aby povolovalo robotu odběr komponenty.



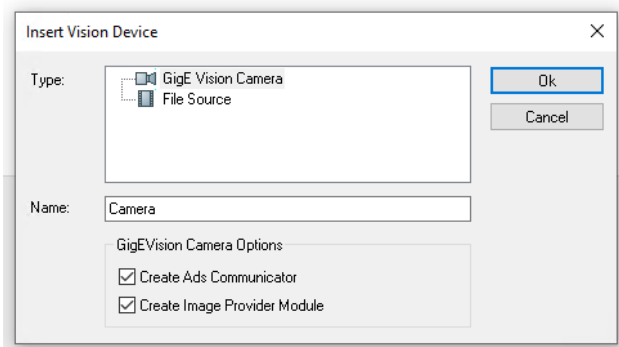
Obrázek 49) Použitý dopravník

11.2 Postup reálného zprovoznění pracoviště

K dispozici už byl plně funkční virtuální model. Postup jeho vytvoření je věnován v kapitole 10. K vytvoření reálného pracoviště byla potřeba ještě několika málo věcí, kterým jsou věnovány následující podkapitoly, ve kterých je řešeno například přenos programu z RobotStudia do reálného robotu, vytvoření komunikace mezi robotem a PLC, navázání komunikace s kamerou a její nastavení apod.

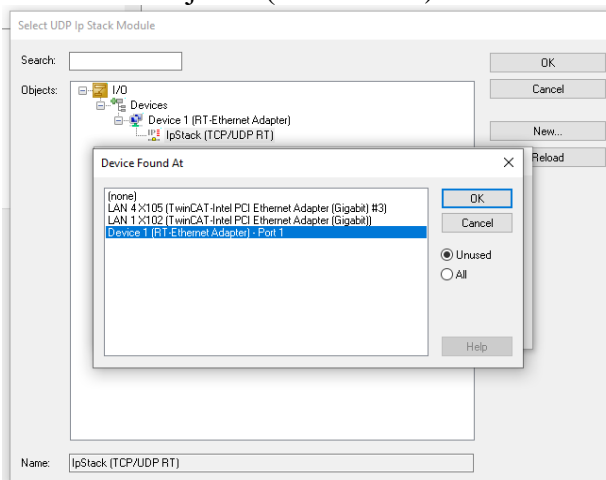
11.2.1 Připojení kamery k PLC

Pro upravování obrazů přímo z kamery je potřeba připojit kameru ke kameře a navázat s ní komunikace. To se provádí tak, že do vytvořené aplikace 1 ve stromě projekt, přidáme ještě zařízení, kde zvolíme typ *GigE Camera*, kameru pojmenujeme a nyní oproti vytváření *File Source* přibýlo možnosti zvolení *Create Ads Communication* (pokud je zaškrtnutý snímá kamera plnou rychlosti a kontinuálně) a *Create image Provider Module* (je poskytovatel rozhraní pro komunikaci s PLC, ve většině případů vyžadován).

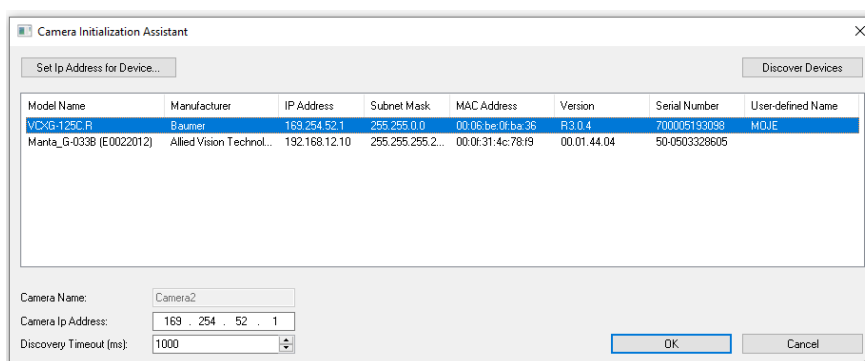


Obrázek 50) TwinCAT GigE Camera

Po vytvoření zařízení je potřeba se ke kameře připojit. Nyní je potřeba se propojit kameru s průmyslovým počítačem pomocí ethernetového kabelu, pokud jsme tak neučinili již dříve. Po spojení těchto dvou systému je potřeba kameru správně adresovat. To probíhá tak, že zvolíme, v jakém portu je kamera připojena v *Select UDP Ip Stack Module* vybereme možnost *New*. Poté zvolíme opět *New*, zde již vybereme přesný port, na kterém je kamera připojena (Obrázek 51). Dalším krokem je potřeba zapsat IP adresu kamery. Zde jsou dvě možnosti buď napíšeme IP adresu kamery ručně, nebo je zde možnost se ji pokusit vyčíst ze systému a poté zvolit o které zařízení se jedná (Obrázek 52).

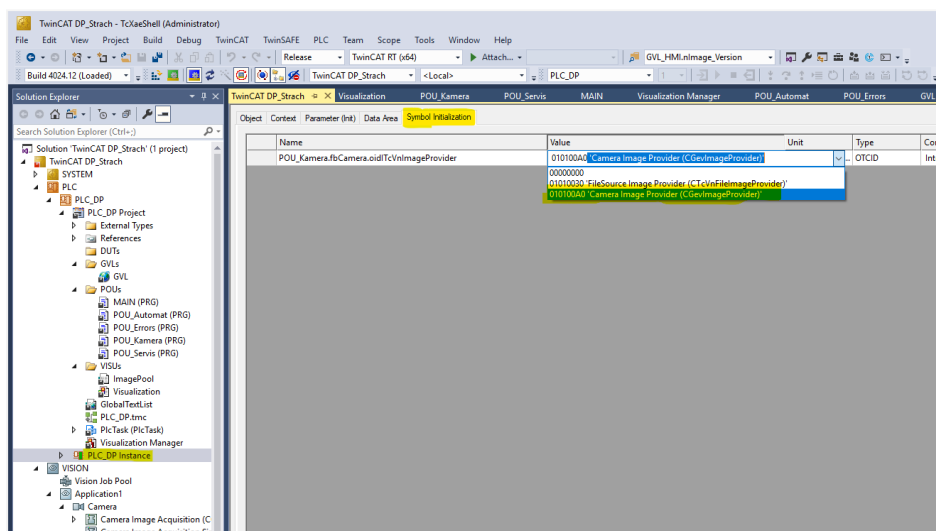


Obrázek 51) LAN připojení kamery



Obrázek 52) TwinCAT IP Camera

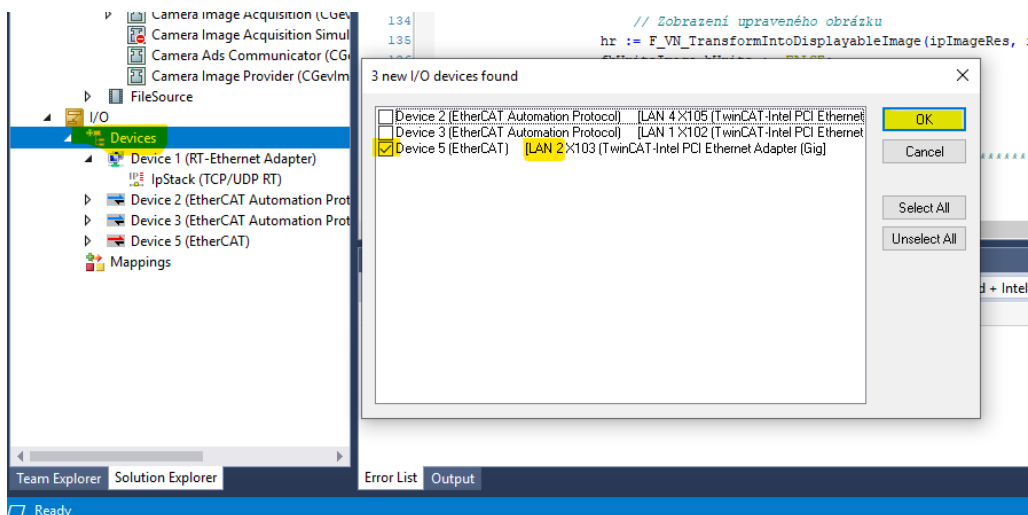
Posledním krokem je vytvoření *FB_VN_Simplecamera* z programu *POU_Kamera* k naší nově vytvořené kameře. To se provádí v *PLC_Instance*, který se nachází ve stromě projektu. Po otevření této karty je potřeba se přepnout do *Symbol Inicialization*, kde se nachází tabulka, ve které se nasměrují funkční bloky pro přijímání obrazu se zdroji obrázku (*FileSource* nebo *GigE kamera*).



Obrázek 53) Symbol inicialization

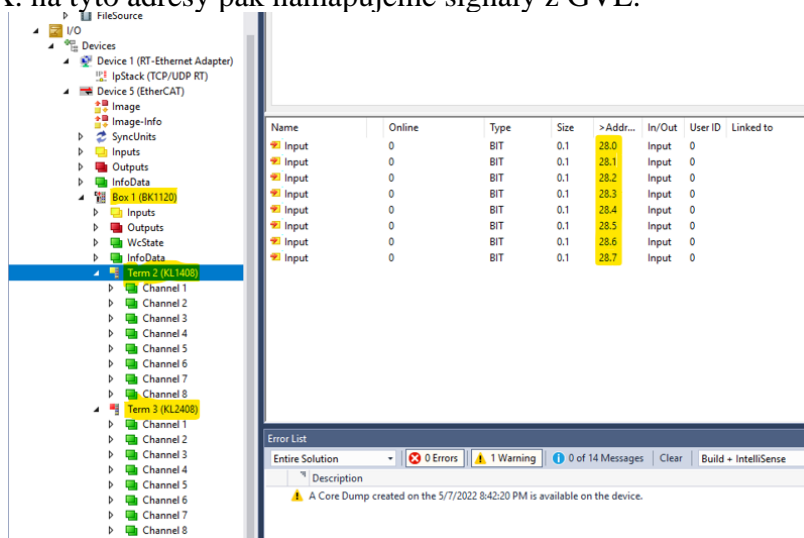
11.2.2 Připojení I/O karty k PLC

Pro možnost ovládání I/O je potřeba propojit pomocí ethernetového kabelu průmyslový počítač s kartou BK1120. Po připojení je nejjednodušším propojením je způsob připojením pomocí skenu připojených zařízení, to se provádí pomocí kliknutím pravého tlačítka na *I/O devices* a zvolíme možnost *Scan*. Objeví se okno, ve kterém zvolíme, které zařízení chceme připojit.



Obrázek 54) I/O připojení

Nyní je potřeba ještě namapovat signály. Pro zjištění adres jednotlivých I/O signálů. Otevřeme *Device5 > Box 1 > Term 2*. Po otevření okna můžeme vidět jednotlivé adresy, jak je tomu na obrázku X. na tyto adresy pak namapujeme signály z GVL.



Obrázek 55) Adresy I/O signálů.

11.2.3 Robot

Export programu a nahrání do robotu

Před samotným exportem bylo program ještě upravit, nejprve bylo potřeba přesunout data o pracovních objektech a informace o nástroji do modulu 1, který se poté samostatně uložil.

Samotný export probíhal pomocí RobotStudia, kdy byl otevřen virtuální model, počítač se připojil pomocí ethernetového kabelu k počítači. Následně vyhledán virtuální kontrolér. Po připojení k reálnému robotu bylo potřeba požádat o povolení úprav programu, které bylo potvrzeno na TeachPendantu, aby byl ovládán robot pouze z jednoho místa. Do prostředí RAPID byl poté nahrán námi vytvořený modul. Poněvadž jsme nahráli pouze modul, muselo být provedeno vytvoření a mapování signálů.

Zaměřování pozic bodů robotu

Dalším krokem bylo zaměření přesných pozic bodů manipulace, toho dosáhnout lze několika způsoby. Základem tohoto zaměřování je najetí do přesné pozice pomocí funkce jog. Po najetí do přesné pozice zvolíme na panelu vzhledem ke kterému souřadnému systému chceme zobrazit hodnoty. Ty se nám zobrazí v levé spodní části. Nyní je můžeme přepsat v RobotStudia, nebo najít tuto pozici v programu, který je nahráný v kontroléru. Po označení této pozice se nám v pravé části zobrazí možnost *Update position*. Jako je tomu na Obrázek 56.



Obrázek 56) Teach pendantt ABB

Odkoušení programu

Po nastavení všech potřebných parametrů uvedené výše začala fáze testování, kdy byl program otestován nejdříve v deseti procentní rychlosti bez komponenty pro manipulaci. Poté otestován znovu, tentokrát i s komponentou manipulace. Následovali další testování, u kterých byla postupně zvyšována rychlost každým cyklem o deset procent, dokud nebylo dosažené plné rychlosti.

12 ZHODNOCENÍ A DISKUZE

Výsledků zadání bylo dosaženo ale jako každé jiné zadání je i zde spousta možností, jak by se dalo toto pracoviště dotáhnout do úplného konce. Kdyby toto pracoviště mělo být zavedeno do praxe, musela by být provedena kontrola bezpečnosti pracoviště, a jiné další záležitosti. Dalším krokem by bylo kalibrovat kameru a když už máme rozměr v pixelech, pomocí kalibrace bychom jej dosáhli převést na milimetry a kontrolovat nejen natočení, ale i přesný rozměr. Taková kalibrace se provádí pomocí kalibračního vzoru, ať už stanovený výrobcem nebo je možnost vytvořit i vlastní, případně pokud známe přesný rozměr otvoru alespoň jedné komponenty může proběhnout kalibrace právě na pomoci ní.

Vedení na dopravníkovém páse bylo zvoleno tohoto typu, protože bylo časově méně náročné, podařilo se rozpracovat i variantu, kdy by na dopravníku bylo n součástek, ale z časových důvodů nebylo dokončeno a je zde uvedena varianta s jednodušším vedením a polohováním. Při zmíněné variantě by obsluha naložila n kostek na dopravníkový pás, součásti dojedou na místo odběru, PLC zastaví pás a vytvoří se snímek, díky kterému se vyhodnotí pozice a natočení jednotlivých komponent. Aby toho bylo dosaženo, byla by potřeba nejprve zkalibrovat kameru tak abychom z ní byli schopni dostat informaci v milimetrech, a ne pixelech, jak je tomu v praktické části tohoto úkolu. Dalším krokem postupu by bylo potřeba sjednotit souřadný systém kamery a robotu. Mohl by však vznikat problém s opakovatelností pozice a bylo by lepší provádět kalibraci automaticky, například mít kalibrační vzor součásti stolu a pravidelně ji po n krocích obnovit.

13 ZÁVĚR

Diplomová práce byla zaměřena na strojové vidění v robotizovaných pracovištích. V rešeršní část byla věnována nejprve hardwarem strojového vidění, kde byly popsány jednotlivé komponenty jako osvětlení scény, kamera a objektiv. U každé z těchto komponent byli charakterizovány jednotlivé typy a popisy výhod a nevýhod jednotlivých typů při výběru do praxe. Další částí rešerše bylo popis principu strojového vidění a rozdělení jednotlivých kamerových senzorů do kategorií jakým způsobem fungují. Následovaly typické úlohy pro strojové vidění, jako nalezení povrchových vad, bin picking a jiné. Každá z těchto operací byla nastíněna a byl vysvětlen její význam. Poté byli charakterizovány jednotlivé možnosti spojení kamery s průmyslovým robotem. V této kapitole byla věnována hlavně pozornost, jaké jsou možnosti sjednocení souřadných systémů robotu a kamery a možnosti jejich zaměřování. Následovali základní informace o možnostech vytvoření řízení strojového vidění v prostředí TwinCAT, kde bylo nejprve popsáno, pro jaké operace je tento program vhodný, poté objasněna základní architektura prostředí, jaké používá standardy, a nakonec charakterizovány možnosti zpracování obrazu, jako vytvoření algoritmů, nebo použití na vstup obrázky uložené přímo v počítači anebo možnosti připojení kamery. Následoval vybraných komunikačních protokolů, jaká je snaha průmyslové komunikace, popsány vybrané komunikační protokoly. Nejprve to byl OPC server, který byl použitý v části virtuálního zprovoznění, poté následovali i jiné které jsou v praxi nejčastěji používány. Další částí bylo přestavení zadání praktické části, za kterým následoval popis měřené komponenty, za kterým poté následoval systémový rozbor, který se zamýšlel nad pracovištěm z hledisek optického, komunikačního problému a možnosti odebírání kostek z dopravníkového pásu.

V návrhu virtuálního zprovoznění pracoviště s kamerovým systémem, nejprve je část věnována návrhu jednotlivých komponent i s odůvodněním tvaru a principu jejich funkce, které poté byli vytištěny na 3D tiskárně. Druhá část je věnována vytvoření virtuálního modelu pracoviště, popisy vybraných proměnných a signálů robotu, zrychlený přehled programu vlastního stavového automatu, který řídí pohyby robotu. Třetí část je věnována Vytvoření hlavního řídicího programu pro nadřazené PLC, které řídí celý systém. Nejprve byl popsán program MAIN společně s obrazovkami pro možnosti ovládání obsluhou. Následoval program POU_Automat, který řídí automatický cyklus stroje, POU_Servis, který má dvě části, pro nastavení hodnot pro kameru a druhá pro nastartování a spuštění programu robotu, POU_Errors a poslední nejdůležitější program POU_Kamera, která byla věnována zprovoznění visionu, kde bylo popsáno vytvoření snímku pro přípravu programu před testováním na reálné sestavě. Dále bylo objasněn algoritmus pro zpracování a vyhodnocení. Poslední částí virtuálního zprovoznění bylo navázání komunikace mezi použitými programy pomocí OPC serveru.

Poslední část byla věnována testováním na reálné sestavě virtuálního modelu, kde byly popsány použité komponenty, připojení kamery k PLC a její nastavení, připojení I/O karty, export programu do robotu, zaměření bodů operace a navázání komunikace.

14 SEZNAM POUŽITÝCH ZDROJŮ

- AUTOMA. *AUTOMA: Strojové vidění I: Principy a charakteristiky* [online].
 [1] 2008, **2008**(01), [cit. 2022-05-09]. Dostupné z: https://automa.cz/cz/casopis-clanky/strojove-videni-i-principy-a-charakteristiky-2008_01_36550_5518/
- Vision system design. In: *Vision system design* [online]. online: Factory
 [2] Automation, 2012, celý dokument [cit. 2022-05-09]. Dostupné z: <https://www.vision-systems.com/factory/article/16751731/webcast-10-things-to-consider-when-designing-a-machine-vision-system>
- Strojové vidění IV: Osvětlovače. *AUTOMA* [online]. 2008, **2008**(04), [cit. 2022-
 [3] 05-09]. Dostupné z: https://automa.cz/cz/casopis-clanky/strojove-videni-iv-osvetlovace-2008_04_36988_5853/
- Vlastnosti světla. *Odbornecasopisy* [online]. 2008, **2008**(1), [cit. 2022-05-14].
 [4] Dostupné z: <http://www.odbornecasopisy.cz/>
- Positions of bright-field and dark-field Lights. In: *ResearchGate* [online]. online:
 [5] ResearchGate, 2016 [cit. 2022-05-14]. Dostupné z: https://www.researchgate.net/figure/Positions-of-bright-field-and-dark-field-Lights-based-upon-23_fig1_303785367
- Bright field, front light illumination. In: *OTPO engineering* [online]. online:
 [6] OTPO engineering, 2012 [cit. 2022-05-14]. Dostupné z: <https://www.opto-e.com/basics/bright-field-front-light-illumination>
- Dark field illumination. In: *VisonDoctor* [online]. online: VisonDoctor, 2015 [cit.
 [7] 2022-05-14]. Dostupné z: <https://www.vision-doctor.com/en/illumination-techniques/dark-field-illumination.html>
- Plošné osvětlovače. In: *Analýza obrazu* [online]. online: Analýza obrazu, 2018
 [8] [cit. 2022-05-14]. Dostupné z: <http://www.analyza-obrazu.cz/osvetlovace/plosne/>
- Osvětlovače pro strojové vidění. *W-Technika* [online]. 2016, **2016**(12), [cit. 2022-
 [9] 05-14]. Dostupné z: <https://www.w-technika.cz/osvetlovace-pro-strojove-videni/>
- Machine Vision LED coaxial light. In: *Made-in-china* [online]. 2019: MIC, 2019
 [10] [cit. 2022-05-14]. Dostupné z: <https://m.made-in-china.com/product/Machine-Vision-LED-Coaxial-Light-722956215.html>
- Lighting. *OptoEngineering* [online]. 2016, **2016**(2), [cit. 2022-05-14]. Dostupné z:
 [11] <https://www.opto-e.com/basics/lighting>
- Strojové vidění – několik úskalí návrhu systémů. *Mii* [online]. 2010, **2010**(4), [cit.
 [12] 2022-05-14]. Dostupné z: <https://www.mii.cz/art?id=558&lang=405>
- Řádkové kamery - úvod do problematiky. *ELCOM* [online]. 2016, **2016**(6), [cit.
 [13] 2022-05-14]. Dostupné z: <https://www.prumyslove-kamery.cz/clanky-a-aktuality/clanky?pg=443>
- Řádkové kamery - obrázek. In: <https://www.uvbtechnik.cz/automaticka-detekce-povrchovych-vad> [online]. online: UVB Technik, 2017 [cit. 2022-05-14]. Dostupné z:
 [14] <https://www.uvbtechnik.cz/automaticka-detekce-povrchovych-vad>
- Cameras. In: *Edmundoptics* [online]. online: Edmund Optics, 2016 [cit. 2022-05-
 [15] 14]. Dostupné z: <https://www.edmundoptics.co.uk/knowledge-center/application-notes/imaging/camera-types-and-interfaces-for-machine-vision-applications/>
- Systém strojového vidění OnRobot Eyes. *AUTOMA* [online]. 2020, **2020**(11), [cit.
 [16] 2022-05-14]. Dostupné z: <https://automa.cz/cz/casopis-clanky/system-strojoveho-videni->

onrobot-eyes-2020_11_0_13255/

2,5D Cameras system. In: *Gushiciku* [online]. online: gushiciku, 2018 [cit. 2022-05-14]. Dostupné z: <https://www.gushiciku.cn/pl/2ThP>

CCD - základní princip. *Fyzika.jreichl* [online]. 2009, **2009**(10), [cit. 2022-05-14]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/748-ccd-zakladni-princip>

Obrazové snímače CCD vs. CMOS. *Netcam* [online]. 2008, **2008**(03), [cit. 2022-05-14]. Dostupné z: <https://netcam.cz/encyklopedie-ip-zabezpeceni-obrazove-snimace-ccd-cmos.php>

Bayerova maska. *Diit.cz* [online]. 2013, **2013**(2) [cit. 2022-05-14]. Dostupné z: <https://diit.cz/clanek/>

Bayerova maska. In: *Fotoalestomcik* [online]. online: fotoalestomcik, 2014 [cit. 2022-05-14]. Dostupné z: <https://fotoalestomcik.estranky.cz/clanky/bayerova-maskafotopojmy-fotonazvy-foto-pojmy-terminologie-nazvoslovi.html>

Vliv CCD snímače a optiky na kvalitu obrazu. *Visionx* [online]. 2015, **2015**(2), [cit. 2022-05-14]. Dostupné z: <https://docplayer.cz/15640061-Atesystem-s-r-o-kamery-pro-prumyslove-aplikace-vliv-ccd-snimace-a-optiky-na-kvalitu-obrazu-www-visionx-cz.html>

Strojové vidění – několik úskalí návrhu systémů. *AUTOMA* [online]. 2010, **2010**(04), [cit. 2022-05-14]. Dostupné z: https://automa.cz/cz/casopis-clanky/strojove-videni-nekolik-uskali-navrhu-systemu-2010_04_40946_5759/

SVĚTELNOST OBJEKTIVU, PROČ JE DŮLEŽITÁ?. *FOOLAB CEWE blog* [online]. 2019, **2019**(2), [cit. 2022-05-14]. Dostupné z: <https://www.fotolab.cz/blog/svetelnost-objektivu-proc-je-dulezita/>

Hloubka ostrosti. *Megapixel* [online]. 2008, **2008**(1), [cit. 2022-05-14]. Dostupné z: <https://www.megapixel.cz/videoporadna-hloubka-ostrosti>

Známé velikosti montážních závitů. *KINALI.CZ* [online]. 2019, **2019**(01), [cit. 2022-05-14]. Dostupné z: <http://www.prumysloveobjektivy.cz/pages/view/zname-velikosti-montaznich-zavitu>

Technologien. *Vision-control.com* [online]. 2019, **2019**(55), [cit. 2022-05-14]. Dostupné z: <https://www.vision-control.com/technologien/>

Základní principy strojového vidění - 3 díl. *ElektroPrůmysl.cz* [online]. 2011, **2011**(8), [cit. 2022-05-14]. Dostupné z: <https://www.elektroprumysl.cz/automatizace/zakladni-principy-strojoveho-videni-3-dil>

Objektivy pro strojové vidění. *W-Technika* [online]. 2010, **2010**(2), [cit. 2022-05-14]. Dostupné z: <https://www.w-technika.cz/objektivy-pro-strojove-videni/>

Pilotní aplikace. *Br-automation.com* [online]. 2012, **2012**(2), [cit. 2022-05-14]. Dostupné z: <https://www.br-automation.com/cs/produkty/strojove-videni/pilotni-aplikace/>

Strojové vidění II: Úlohy, nástroje a algoritmy. *AUTOMA* [online]. 2008, **2008**(02), [cit. 2022-05-09]. Dostupné z: https://automa.cz/cz/casopis-clanky/strojove-videni-ii-ulohy-nastroje-a-algoritmy-2008_02_36676_5521/

Dáváme robotům oči. *AUTOMA* [online]. 2021, **2021**(06), [cit. 2022-05-14]. Dostupné z: https://automa.cz/cz/casopis-clanky/davame-robotum-oci-2021_06_0_13557/

Robot a strojové vidění. *AUTOMA* [online]. 2009, **2009**(03), [cit. 2022-05-14]. Dostupné z: https://automa.cz/cz/casopis-clanky/robot-a-strojove-videni-2009_03_38714_6086/

- Logické řízení a strojové vidění v jednotném prostředí TwinCAT Vision.
- [34] *AUTOMA* [online]. 2018, **2018**(12), [cit. 2022-05-14]. Dostupné z: https://automa.cz/cz/casopis-clanky/logicke-rizeni-a-strojove-videni-v-jednotnem-prostredi-twincat-vision-2018_12_0_11891/
- OPC UA VÁM ZJEDNODUŠÍ ŽIVOT, ALE... *FOXON* [online]. 2019, **2019**(9), [cit. 2022-05-14]. Dostupné z: https://foxon.cz/blog/ostatni/454-opc-ua-vam-zjednodusi-zivot-ale?gclid=EAiaIQobChMIrOS057vK9wIVm4ODBx22JwppEAAYASABEgKqpD_BwE
- EtherNet / IP versus PROFINET. *Automatizace.hw.cz* [online]. 2020, **2020**(3), [cit. 2022-05-14]. Dostupné z: <https://automatizace.hw.cz/ethernet-ip-versus-profinet.html>
- EtherCAT Automation Protocol. *AUTOMA* [online]. 2017, **2017**(2), [cit. 2022-05-14]. Dostupné z: https://automa.cz/cz/casopis-clanky/ethercat-automation-protocol-2017_02_0_9811/

15 SEZNAM ZKRATEK

1D	jedno dimenzový
2,5D	dvou a půl dimenzový
2D	dvou dimenzový
3D	tří dimenzový
ADS	Adaptive Damping Systém
BOOL	Bolean
CCD	Charge-coupled device
CMOS	Complementary Metal–Oxide–Semiconductor
DOAL	Diffused On Axis Lighting
GigE	Gigabit Ethernet
GVL	Gobal variable list
HMI	Human Mechine Interface
HW	Hard-Ware
I/O	Input / Output
kHz	kilo hertz
kPx	kilo pixel
LED	Light-Emitting Diode
Max	Maximum
Min	Minimum
MMI	Men machine interface
NOK	No Ok
OK	Ok
PC	Programing computer
PLC	Program logic controler
SD	Secure digital
SW	Software
TON	Od delay time
ÚVSSR	Ústav výrobních stroju, systémů a roboty
VUT	Vysoké učení technické

16 SEZNAM OBRÁZKŮ

Obrázek 1) Příklad strojového vidění [2].....	19
Obrázek 2) Odraz světla.....	20
Obrázek 3) Geometrie osvětlení [5].....	21
Obrázek 4) Přední osvětlení s jasným obrazovým polem [6]	22
Obrázek 5) Přední osvětlení s tmným zorným polem [7]	22
Obrázek 6) Zadní osvětlení [6].....	22
Obrázek 7) Plošné osvětlení [8]	23
Obrázek 8) Kruhové osvětlení [9].....	24
Obrázek 9) Kopulové osvětlení [9].....	24
Obrázek 10) DOAL osvětlení [10].....	25
Obrázek 11) Osvětlení dark-field [9]	25
Obrázek 12) Osvětlení back light [9]	26
Obrázek 13) řádkové kamery [14]	27
Obrázek 14) Plošné kamery [15].....	28
Obrázek 15) 2,5 kamerové systémy [17]	28
Obrázek 16) Bayerova maska [21].....	29
Obrázek 17) Tříčipové provedení [22].....	30
Obrázek 18) Sekvenční snímání [20].....	30
Obrázek 19) Montážní závity objektivů [26]	32
Obrázek 20) Standartní objektiv [28].....	32
Obrázek 21) Telecentrický objektiv [28]	33
Obrázek 22) Hypermetrický objektiv [28].....	33
Obrázek 23) OPC server	44
Obrázek 24) Měřená součást.....	47
Obrázek 25) Držák pro kameru.....	49
Obrázek 26) Měřicí stanice	50
Obrázek 27) Vedení kostky na dopravníku.....	50
Obrázek 28) RobotStudio pracoviště	51
Obrázek 29) Obrazovka školená obsluha.....	53
Obrázek 30) Obrazovka úvodní stránka.....	54
Obrázek 31) Obrazovka servis kamera	54
Obrázek 32) obrazovka servis robot	55
Obrázek 33) Obrazovka přechod automat.....	55
Obrázek 34) Obrazovka Automat	55
Obrázek 35) Obrazovka ukončení zakládání	56
Obrázek 36) TwinCAT připravený obraz	57
Obrázek 37) TwinCAT File Source	58
Obrázek 38) TwinCAT VISION instance.....	59
Obrázek 39) TwinCAT OPC Vytvoření projektu	61
Obrázek 40) vytvoření OPC device	61
Obrázek 41) Povolení komunikace pomocí TMC File	62
Obrázek 42) RobotStudio OPC.....	62
Obrázek 43) OPC UA Client Configuration	63
Obrázek 44) Unified Automation UaExpert	63
Obrázek 45) Reálné pracoviště	64
Obrázek 46) Použitá kamera	65
Obrázek 47) Použitá měřicí stanice.....	65
Obrázek 48) Robot ABB SWIFTI CRB 1100.....	66

Obrázek 49) Použitý dopravník	66
Obrázek 50) TwinCAT GigE Camera	67
Obrázek 51) LAN připojení kamery	67
Obrázek 52) TwinCAT IP Camera	68
Obrázek 53) Symbol inicialization	68
Obrázek 54) I/O připojení.....	69
Obrázek 55) Adresy I/O signálů.....	69
Obrázek 56) Teach pendatnt ABB.....	70

17 SEZNAM TABULEK

Tabulka 1) Porovnání objektivů [29]	34
Tabulka 2) Kategorie strojového vidění [1]	37
Tabulka 3) Signály robotu.....	52
Tabulka 4) Proměnné Robotu	52
Tabulka 5) Proměnné POU_Kamera	59

18 SEZNAM PŘÍLOH

Příloha 1 RobotStudio MAIN.....	81
Příloha 2 TwinCAT POU GVL	85
Příloha 3 TwinCAT POU_MAIN.....	86
Příloha 4 TwinCAT POU_Automat	89
Příloha 5 TwinCAT POU_Servis	92
Příloha 6 TwinCAT POU_Errors	94
Příloha 7 TwinCAT POU_Kamera.....	95

19 PŘÍLOHY

Príloha 1 RobotStudio MAIN

```

MODULE Module1

    CONST robtarget Zakl_Place:=[[53,19,28],[0,-0.707106781,0.707106781,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Conv_Take:=[[15.832,15.828,16.19],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST jointtarget JointTarget_1:=[[0,0,0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget MericiSt_Mereni:=[[75,69.5,15],[0.5,-0.5,-0.5,-0.5],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget MericiSt_Manipulation:=[[75.000608686,69.50019751,44.046747919],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

    VAR num OffX := -34;
    VAR num OffY := 34;

    VAR num PosX := 0;
    VAR num PosY := 0;

    VAR num PosFinalX :=2;
    VAR num PosFinalY :=2;

|*****

PROC main()

    TEST CommunicationBYTE_0

    CASE 0:

    CASE 5:
        IF doRobotAtHomePos = 0 THEN
            Path_home;
            SetDO doRobotAtHomePos, 1;
        ENDIF

    CASE 10:
        IF NOT CommunicationBYTE_0_Q = 15 THEN
            SetGO CommunicationBYTE_0_Q, 10;

            Path_10;

            Path_20;

            SetGO CommunicationBYTE_0_Q, 15;
        ENDIF

    CASE 15:

    CASE 40:
        IF NOT CommunicationBYTE_0_Q = 40 THEN

```

```

SetGO CommunicationBYTE_0_Q, 40;
IF Posx = PosfinalX THEN                ! offsetovani X
    PosX := 0;
    PosY := PosY + 1;

    IF PosY = PosfinalY THEN            !Offsetovani Y
        PosX := 0;
        PosY := 0;
    ENDIF
ENDIF
Path_40;
PosX := PosX +1;                        !Offset pozic v Zásobníku
ENDIF

CASE 50:
IF NOT CommunicationBYTE_0_Q = 15 THEN
    SetGO CommunicationBYTE_0_Q, 50;
    Path_30;
    Path_20;
    SetGO CommunicationBYTE_0_Q, 15;
ENDIF
ENDTEST

IF CommunicationBYTE_0 <> 5 THEN
    SetDO doRobotAtHomePos, 0;
ENDIF

ENDPROC

PROC Path_10()                          ! Manipulace z pásu na měřicí stanici
    !Otevření Gripperu
    PulseDO\PLength:=1,doGrippOpen;
    WaitDI diGrippOpened,1;
    !najetí na pozici odběru
    MoveJ offs(Conv_Take,0,0,100),v1000,fine,toolMPG50_1\WObj:=WobjConveyorr;
    MoveL Conv_Take,v100,fine,toolMPG50_1\WObj:=WobjConveyorr;
    PulseDO\PLength:=1,doGrippClose;
    WaitDI diGrippClosed,1;
    !Najetí na místo položení v měřicí stanici
    MoveL offs(Conv_Take,0,0,100),v100,fine,toolMPG50_1\WObj:=WobjConveyorr;
    MoveJ offs(MericiSt_Manipulation,0,0,100),v1000,fine,toolMPG50_1\WObj:=WobjMericiStanice;
    MoveL MericiSt_Manipulation,v100,fine,toolMPG50_1\WObj:=WobjMericiStanice;
    PulseDO\PLength:=1,doGrippOpen;
    WaitDI diGrippOpened,1;

    !Odjetí nad místo položení
    MoveJ offs(MericiSt_Manipulation,0,0,100),v100,fine,toolMPG50_1\WObj:=WobjMericiStanice;
ENDPROC

PROC Path_20()                          !Pozice pro měření kamerou

```

```

MoveJ MericiSt_Mereni,v1000,fine,MyNewKamera\WObj:=WobjMericiStanice;

ENDPROC

PROC Path_30()                                !Otáčení měřené součásti v měřicí stanici
!najetí nad soucast
MoveJ offs(MericiSt_Manipulation,0,0,100),v1000,fine,toolMPG50_1\WObj:=WobjMericiStanice;

!odebrání obrobku do gripperu
MoveL MericiSt_Manipulation,v100,fine,toolMPG50_1\WObj:=WobjMericiStanice;
PulseD0\PLength:=1,dogrippClose;
WaitDI diGrippClosed,1;

!Vyjetí nahoru a otočení součásti
MoveL offs(MericiSt_Manipulation,0,0,100),v100,fine,toolMPG50_1\WObj:=WobjMericiStanice;
MoveL offs(RelTool(MericiSt_Manipulation,0,0,0,\Rz:=90),0,0,100),v100,fine,toolMPG50_1
\WObj:=WobjMericiStanice;

!Založení součásti zpět
MoveL RelTool(MericiSt_Manipulation,0,0,0,\Rz:=90),v100,fine,toolMPG50_1
\WObj:=WobjMericiStanice;
PulseD0\PLength:=1,dogrippOpen;
WaitDI diGrippOpened,1;

!Vyjetí nad součást
MoveL offs(RelTool(MericiSt_Manipulation,0,0,0,\Rz:=90),0,0,100),v100,fine,toolMPG50_1
\WObj:=WobjMericiStanice;
ENDPROC

PROC Path_40()                                !Manipulace součásti z měřicí stanice do kastlíku

!Odebrání OK pozicované součásti
MoveJ offs(MericiSt_Manipulation,0,0,100),v1000,fine,toolMPG50_1\WObj:=WobjMericiStanice;
MoveL MericiSt_Manipulation,v100,fine,toolMPG50_1\WObj:=WobjMericiStanice;
PulseD0\PLength:=1,dogrippClose;
WaitDI diGrippClosed,1;
MoveJ offs(MericiSt_Manipulation,0,0,100),v1000,fine,toolMPG50_1\WObj:=WobjMericiStanice;

!Odložení do Zakladače
MoveJ offs(Zakl_Place,OffX*PosX,OffY*PosY,100),v1000,fine,toolMPG50_1\WObj:=WobjZakladac;
MoveL offs(Zakl_Place,OffX*PosX,OffY*PosY,0),v100,fine,toolMPG50_1\WObj:=WobjZakladac;
PulseD0\PLength:=1,dogrippOpen;
WaitDI diGrippOpened,1;
MoveL offs(Zakl_Place,OffX*PosX,OffY*PosY,100),v100,fine,toolMPG50_1\WObj:=WobjZakladac;

ENDPROC

PROC Path_home()                              !Najetí do HOME pozice
MoveAbsJ JointTarget_1,v1000,z100,toolMPG50_1\WObj:=WobjConveyorr;

```

```
ENDPROC
PROC Path_00()                                !Funkce pro přepis pozic bodů, v MAIN nepožítá, když použiji ↵
offset, nepřepisují se pozice
  MoveJ Conv_Take,v1000,z100,My_Mechanism_1\WObj:=WobjConveyorr;
  MoveJ Zakl_Place,v1000,z100,My_Mechanism_1\WObj:=WobjZakladac;
  MoveJ MericiSt_Mereni,v1000,z100,MyNewKamera\WObj:=WobjMericiStanice;
  MoveJ MericiSt_Manipulation,v1000,z100,MyNewKamera\WObj:=WobjMericiStanice;
ENDPROC
PROC Path_15()
  MoveJ MericiSt_Mereni,v1000,fine,MyNewKamera\WObj:=WobjMericiStanice;
ENDPROC
ENDMODULE
```

Příloha 2 TwinCAT POU GVL

Global Variable List: GVL

```

1  {attribute 'qualified_only'}
2  VAR_GLOBAL
3      //ROBOT
4      //Komunikace
5      //BYTE
6          BYTE_Communication_IN_0      :   BYTE ;
7          BYTE_Communication_Out_0     :   BYTE ;
8      // BOOL
9      //INPUTS
10         bI_Robot_AutoOn              :   BOOL ;
11         bI_Robot_EStopStatus         :   BOOL ;
12         bI_Robot_MottorsOffState     :   BOOL ;
13         bI_Robot_MottorsOnState     :   BOOL ;
14         bI_Robot_RunChainOk         :   BOOL ;
15         bI_Robot_HomePos             :   BOOL ;
16         //OUTPUTS
17         bQ_Robot_MotorsOn            :   BOOL ;
18         bQ_Robot_MotorsOff          :   BOOL ;
19         bQ_Robot_Start               :   BOOL ;
20         bQ_Robot_Stop                :   BOOL ;
21         bQ_Robot_StartAtMain        :   BOOL ;
22
23         //Komunikace
24         //Osvětlení
25         bQ_Light1                    AT %Q* :   BOOL ;
26         bQ_Light2                    AT %Q* :   BOOL ;
27         //SENSORS
28         //INPUTS
29         bI_ConvCompOnTakePos         :   BOOL ;
30         //HMI
31         b_StopAutomat :   BOOL ;
32         n_VerzeObrazku :   BYTE ;
33         //VISION
34         b_SERVIS_Robot :   BOOL ;
35         b_SERVIS_Kamera :   BOOL ;
36         b_AUTOMAT_Spusten :   BOOL ;
37         //Trigger kameru
38         b_TriggrujKameru :   BOOL ;
39         //Input parametry
40         aCenter : TcVnPoint2_REAL := [ 190 , 235 ] ; //
41         předpokládaný střed kružnice
42         //Pou Vyhodnocení Automat
43         fMinRaduisVyhodnoceni :   REAL := 50 ;
44         fMaxRaduisVyhodnoceni :   REAL := 100 ;
45         //NatoceniSOučásti
46         b_Soucastnatocena_NOK :   BOOL ;
47         b_Soucastnatocena_OK :   BOOL ;
48         //Errors
49         bErrors :   BOOL ;
50         nErrorID :   INT ;
51     END_VAR

```

Příloha 3 TwinCAT POU_MAIN

POU: MAIN

```

1  PROGRAM MAIN
2  VAR
3      n_Obrazovka      : INT := 0 ;
4      n_Tlaciditko    : INT := 0 ;
5
6      n_obrazek        : INT ;
7
8
9  END_VAR
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```

```

1  CASE n_Obrazovka OF
2  //*****
3      0 : //Safety Start
4          IF n_Tlaciditko = 1 THEN //Přijmou a pokračovat
5              n_Obrazovka := 1 ;
6              n_Tlaciditko := 0 ;
7          END_IF
8  //*****
9      1 : //Úvod
10         IF n_Tlaciditko = 1 THEN //Automatický provoz
11             n_Obrazovka := 4 ;
12             n_Tlaciditko := 0 ;
13         ELSIF n_Tlaciditko = 2 THEN //Servis
14             n_Obrazovka := 2 ;
15             n_Tlaciditko := 0 ;
16         END_IF
17 //*****
18         2 : //Servis Kamera
19             gvl.bQ_Light1 := TRUE ; gvl.bQ_Light2 := TRUE ; //Zapnutí světel
20             IF n_Tlaciditko = 0 THEN //Zůstávám? pokud ano, zůstáváme
21                 v tomto stavu zjistí údaje z kamery
22                 gvl.b_SERVIS_Kamera := TRUE ;
23                 POU_Kamera ( ) ;
24                 POU_Servis ( ) ;
25             ELSE
26                 gvl.bQ_Light1 := FALSE ; gvl.bQ_Light2 := FALSE ; //Zapnutí
27                 světel
28                 gvl.b_SERVIS_Kamera := FALSE ;
29             END_IF
30         IF n_Tlaciditko = 1 THEN //Servis Robotu
31             n_Obrazovka := 3 ;
32             n_Tlaciditko := 0 ;
33         ELSIF n_Tlaciditko = 2 THEN //Zpět na úvodní obrazovku
34             n_Obrazovka := 1 ;
35             n_Tlaciditko := 0 ;
36         END_IF

```

POU: MAIN

```

36
37
38
39 //*****
40 3 : //ServisRobot
41 //Zůstávám, nebo ne
42 //Ano
43 IF n_Tlacidko = 0 THEN
44   gvl.b_SERVIS_Robot := TRUE ;
45   POU_Servis () ;
46 ELSE
47   //Ne, jdu na jiný stav
48   gvl.b_SERVIS_Robot := FALSE ;
49 END_IF
50 IF n_Tlacidko = 1 THEN //Servis Kamera
51   n_Obrazovka := 2 ;
52   n_Tlacidko := 0 ;
53 ELSIF n_Tlacidko = 2 THEN //Zpět na úvodní obrazovku
54   n_Obrazovka := 1 ;
55   n_Tlacidko := 0 ;
56 END_IF
57 //*****
58 4 : //Přechod Automat
59 //Přijmout a pokračovat
60 IF n_Tlacidko = 1 THEN
61   n_Obrazovka := 5 ;
62   n_Tlacidko := 0 ;
63 ELSIF n_Tlacidko = 2 THEN //Zpět na úvodní obrazovku
64   n_Obrazovka := 1 ;
65   n_Tlacidko := 0 ;
66 END_IF
67 //*****
68 5 : //Automat
69 //Zpět na úvodní obrazovku
70 gvl.b_StopAutomat := TRUE ; // zastav v moment kdy dokončí robot
71 celou operaci
72   n_Obrazovka := 6 ;
73   n_Tlacidko := 0 ;
74 END_IF
75 IF gvl.bErrors THEN
76   n_Obrazovka := 8 ;
77 END_IF
78 POU_Automat () ;
79 //*****
80 6 : //Dokončení zakládání
81 //Zrušit dokončení zakládání
82   n_Obrazovka := 5 ;
83   n_Tlacidko := 0 ;

```

POU: MAIN

```

82         gvl.b_StopAutomat := FALSE ;
83     END_IF
84
85     POU_Automat () ;                               //Pokračuj v programu
86
87     IF gvl.bI_Robot_HomePos THEN //Robot : "Najel jsem do homePos ačkáám na
rozказы"
88         n_Obrazovka      := 1 ;
89         gvl.b_StopAutomat := FALSE ;
90     END_IF
91     //*****
92     7 :                                           //E-Stop
93
94     IF n_Tlacitko = 1 THEN //
95         n_Obrazovka      := 1 ;
96         n_Tlacitko       := 0 ;
97     ELSIF n_Tlacitko = 2 THEN //
98         n_Obrazovka      := 1 ;
99         n_Tlacitko       := 0 ;
100    ELSIF n_Tlacitko = 3 THEN //
101        n_Obrazovka      := 1 ;
102        n_Tlacitko       := 0 ;
103    END_IF
104    //*****
105    8 :                                           //Errors
106    IF n_Tlacitko = 1 THEN //Vše je OK, pokračuj
107        n_Obrazovka      := 1 ;
108        n_Tlacitko       := 0 ;
109        gvl.bErrors      := FALSE ;
110        gvl.nErrorID     := 0 ;
111    END_IF
112    POU_Errors () ;
113 END_CASE

```


Příloha 4 TwinCAT POU_Automat

POU: POU_Automat

```

1  PROGRAM POU_Automat
2  VAR
3      n_StateAutomat      : INT := 10;           //Proměnná pro stavový
automát
4      //Pocitani DO HMI
5      nPocetKusuVZakladaci : BYTE;             //počítání aktuálních
kusů v zásobníku
6      nPocetHotovychZakladacu : INT;          // počítání hotových
zakladaču
7      //Errors
8      nPocetOtoceni       : BYTE;             //počítání natočení
9      bPritomnostSoucasti : BOOL;            //zjistení přítomnosti
součásti
10     bZakladacvymenen : BOOL;
11     bVynulujPocetZakladacu : BOOL;
12 END_VAR
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

POU: POU_Automat

```

27         n_StateAutomat := 40 ;
28         nPocetOtoceni := 0 ;
29     END_IF
30     IF gvl.b_Soucastnatocena_NOK THEN //Součást je natočená špatně
31         gvl.BYTE_Communication_Out_0 := 50 ; //PLC: ""Součást je NOK,
otoč ji"
32         n_StateAutomat := 50 ;
33         nPocetOtoceni := nPocetOtoceni + 1 ;
34     END_IF
35     IF gvl.bErrors = TRUE THEN
36         gvl.nErrorID := 1 ;
37     END_IF
38     IF gvl.bErrors OR gvl.b_Soucastnatocena_NOK OR gvl.
b_Soucastnatocena_OK THEN
39         gvl.bQ_Light1 := FALSE ; gvl.bQ_Light2 := FALSE ;
40     END_IF
41     //*****
42     40 :
43     IF gvl.BYTE_Communication_In_0 = 40 THEN //Robot: "Spouštím
manipulaci pro odložení součásti"
44         gvl.BYTE_Communication_Out_0 := 15 ; //PLC: "Ok, požívám
se jestli je zde další prvek"
45         nPocetKusuVZakladaci := nPocetKusuVZakladaci + 1 ;
46         IF nPocetKusuVZakladaci = 4 THEN
47             n_StateAutomat := 45 ;
48         ELSE
49             n_StateAutomat := 10 ;
50         END_IF
51     END_IF
52     //*****
53     45 :
54     IF bZakladacvymenen THEN
55         nPocetKusuVZakladaci := 0 ;
56         bZakladacvymenen := FALSE ;
57         NPocetHotovychZakladacu := NPocetHotovychZakladacu + 1 ;
58         n_StateAutomat := 10 ;
59     END_IF
60     //*****
61     50 :
62     IF gvl.BYTE_Communication_In_0 = 50 THEN //Robot: "Spouštím
manipulaci pro otočení součásti, a najedu zpět na snímání"
63         gvl.BYTE_Communication_Out_0 := 15 ; //PLC: "Ok, počkám,
než mi opět dáš povel pro snímání"
64         n_StateAutomat := 20 ;
65     END_IF
66
67 END_CASE
68
69 IF bVynulujPocetZakladacu THEN
70     bVynulujPocetZakladacu := FALSE ;
71     NPocetHotovychZakladacu := 0 ;

```

POU: POU_Automat

```
72     END_IF  
73  
74  
75  
76  
77  
78
```

Příloha 5 TwinCAT POU_Servis

POU: POU_Servis

```

1  PROGRAM POU_Servis
2  VAR
3      //HMI_Servis_Kamera
4      b_HMI_PrepisHodnoty      : BOOL ;
5      Timer_ServisKamera      : TON ;
6      //HMI_Servis_Robot
7      //bool z hmi
8      b_HMI_MotorsON          : BOOL ;
9      b_HMI_MotorsOFF         : BOOL ;
10     b_HMI_Start              : BOOL ;
11     b_HMI_Stop               : BOOL ;
12     b_HMI_StartAtMain        : BOOL ;
13
14     //timer for outputs reset
15     Timer_RobotStart         : TON ;
16     Timer_RobotStop          : TON ;
17     Timer_RobotStartAtMain   : TON ;
18 END_VAR
19
20
21 //***** Servis kamera *****
22 IF gvl.b_SERVIS_Kamera THEN
23 // vytvoření dalšího snímku po intervalu 1s
24 Timer_ServisKamera ( IN := TRUE , PT := T#1S );
25 IF Timer_ServisKamera.Q THEN
26 gvl.b_TriggerujKameru := TRUE ;
27 Timer_ServisKamera ( IN := FALSE , PT := T#1S );
28 END_IF
29 //Tlačítka
30
31 END_IF
32
33 //***** Servis ROBOT *****
34 IF gvl.b_SERVIS_Robot THEN
35 //SET Outputs TREU
36 IF b_HMI_MotorsON THEN
37 b_HMI_MotorsON := FALSE ;
38 gvl.bQ_Robot_MotorsOn := TRUE ;
39 END_IF
40 IF b_HMI_MotorsOFF THEN
41 b_HMI_MotorsOFF := FALSE ;
42 gvl.bQ_Robot_MotorsOff := TRUE ;
43 END_IF
44 IF b_HMI_Start THEN
45 Timer_RobotStart ( IN := TRUE , PT := T#2S );
46 b_HMI_Start := FALSE ;
47 gvl.bQ_Robot_Start := TRUE ;
48 END_IF
49 IF b_HMI_Stop THEN
50 Timer_RobotStop ( IN := TRUE , PT := T#2S );

```

POU: POU_Servis

```

31         b_HMI_Stop := FALSE ;
32         gvl.bQ_Robot_Stop := TRUE ;
33     END_IF
34     IF b_HMI_StartAtMain THEN
35         Timer_RobotStartAtMain ( IN := TRUE , PT := T#2S ) ;
36         b_HMI_StartAtMain := FALSE ;
37         gvl.bQ_Robot_StartAtMain := TRUE ;
38     END_IF
39 END_IF
40
41 //SET Outputs FALSE
42 IF gvl.bI_Robot_MottorsOnState THEN
43     gvl.bQ_Robot_MotorsOn := FALSE ;
44 END_IF
45 IF gvl.bI_Robot_MottorsOffState THEN
46     gvl.bQ_Robot_MotorsOff := FALSE ;
47 END_IF
48 IF Timer_RobotStart.Q THEN
49     gvl.bQ_Robot_Start := TRUE ;
50     Timer_RobotStart.IN := FALSE ;
51 END_IF
52 IF Timer_RobotStop.Q THEN
53     gvl.bQ_Robot_Stop := TRUE ;
54     Timer_RobotStop.IN := FALSE ;
55 END_IF
56 IF Timer_RobotStartAtMain.Q THEN
57     gvl.bQ_Robot_StartAtMain := TRUE ;
58     Timer_RobotStartAtMain.IN := FALSE ;
59 END_IF
60
61

```

Příloha 6 TwinCAT POU_Errors

POU: POU_Errors

```
1 PROGRAM POU_Errors
2 VAR
3     nStateError : INT ;
4     ErrorPopis : STRING (255) ;
5     ErrorNazev : STRING (255) ;
6 END_VAR
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Příloha 7 TwinCAT POU_Kamera

POU: POU_Kamera

```

1  PROGRAM POU_Kamera
2  VAR
3  // Camera
4  fbCamera          : FB_VN_SimpleCameraControl ; //funkční blok pro
vyčítání obrazu, je potřeba jej definovat v PLC Instance => symbol
inicialization
5  eState           : ETcVnCameraState ; //proměnná pro
stavový automat kamery
6  // hResults
7  hr               : HRESULT ; //slouží pro zaznamenání erroru
(bezchybového procesu), pokud je 0 je vše ok, když ne, poslední trojčíslí
popisuje chybu
8  hrFunc           : HRESULT ; //hResult pro elipsu, stejně jako více
9  //proměnné pro obrázky
10 ipImageIn        : ITcVnImage ; //Proměnná do které se
načítají data (pro vstupní obraz)
11 //ipImageInDisp : ITcVnDisplayableImage; //Vykreslení ipImageIn
do ADS Image Watch, nebo HMI
12 ipImageRes       : ITcVnImage ; //Proměnná do které se
načítají data (pro upravovaný obraz)
13 ipImageResDisp   : ITcVnDisplayableImage ; //Vykreslení ipImageRes do
ADS Image Watch, nebo HMI
14
15 //Výstupy
16 stEllipse        : TcVnRotatedRectangle ; //neznámá do které vrátí
udaje o elipse, lze je pak vyčítat například stEllipse.aCenter pro pozici
středu
17 stEllipse2       : TcVnRotatedRectangle ;
18 ipContourPoints : ITcVnContainer ; // subpixelový přesný obrys,
pokud není vyžadován nastavit na 0
19 // proměnné pro vykreslení
20 //barvy          používá osmibytové proměnné
RGB(hodnoty barvy 0-255)
21 aColorRed        : TcVnVector4_LREAL := [ 200 , 0 , 0 ] ;
//červená barva
22 aColorGreen      : TcVnVector4_LREAL := [ 0 , 175 , 0 ] ;
//zelená barva
23 aColorBlue       : TcVnVector4_LREAL := [ 0 , 0 , 200 ] ;
//modrá barva
24 //ukládání obrázku pro vizualizaci
25 fbWriteImage     : FB_VN_WriteImage ;
26 //Algoritmus pro přepočítání na doporučené hodnoty
27 n_HMI_Radius     : REAL ;
28 n_HMI_RadiusMinimum : REAL ;
29 n_HMI_RadiusMaximum : REAL ;
30 END_VAR
31

```

```

1  eState := fbCamera.GetState () ;
2  CASE eState OF

```

POU: POU_Kamera

```

3          TCVN_CS_INITIAL , TCVN_CS_INITIALIZING , TCVN_CS_INITIALIZED ,
TCVN_CS_OPENING , TCVN_CS_OPENED , TCVN_CS_STARTACQUISITION :
4          hr := fbCamera . StartAcquisition ( ) ;
5          TCVN_CS_STOPACQUISITION :
6          IF gvl . b_TriggrujKameru THEN
7              hr := fbCamera . StopAcquisition ( ) ;
8              hr := fbCamera . StartAcquisition ( ) ;
9              gvl . b_TriggrujKameru := FALSE ;
10         END_IF
11         TCVN_CS_ERROR :           //Pokud se zaznamená error
12         hr := fbCamera . Reset ( ) ;           //Restartuj se
13
14         TCVN_CS_ACQUIRING :           //pokud je
získávání obrázku aktivní
15
16         hr := fbCamera . GetCurrentImage ( ipImageIn ) ;           //nahrej
obrázek do proměnné ipImageIn
17         IF SUCCEEDED ( hr ) AND ipImageIn <> 0 THEN           //pokud
nenastal žádný error(hr), a je načtený obrázek(velikost je rozdílná od 0
18
19         //*****SERVIS*****
20         IF gvl . b_SERVIS_Kamera THEN           //Pokude je
spouštěný servis
21             //Lokalizace vnitřního průměru
22             hrFunc := F_VN_LocateEllipseExp (
                //Lokalizování Elipsy
23                 ipSrcImage := ipImageIn ,
                //Zdrojový obrázek, z kterého se vychází
24                 stEllipse := stEllipse ,
                //Neznámá do které vrátí udaje o elipse, lze je pak vyčítat
                například stEllipse.aCenter pro pozici středu
25                 aCenter := gvl . aCenter ,
                //Očekávaný střed elipsy
26                 fSearchRadius := 250 ,
                //poloměr kolem středu pro hledání hran
27                 eEdgeDirection := TCVN_ED_LIGHT_TO_DARK
,           // specifikace jak má hledat           // := součást je tmavá, na světlém
pozadí
28                 fMinStrength := 10 ,
                //Specifikace minimální pevnosti(rozdílu hran)
29                 nMaxThickness := 3 ,
                //maximální tloušťka okraje ve kterém má hledat
30                 bInvertSearchDirection := FALSE ,
                // pokud je True, hledá od vnějšku směrem ke středu,
pokud False naopak
31                 fMinSearchRadius := 10 ,
                // poloměr od středu který má přeskočit a nehledat v
něm
32                 nSubpixelsIterations := 5 ,

```


POU: POU_Kamera

```

// maximální počet iterací pro optimalizaci
parametrů
33     nSearchLines      := 40 ,
// body kružnice, podm.: min 8, dělitelné 4, čím víc
bodů, tím delší čas
34     fApproxPrecision := 0.001 ,
// přesnost aproximace
35     eAlgorithm       :=
TCVN_EDA_INTERPOLATION , //výběr algoritmu pro detekci hran
36     ipContourPoints  := ipContourPoints ,
//vrátí subpixelový přesný obrys, pokud není
vyžadován nastavit na 0, pokud je vyžadován ipContourPoints
37     hrPrev           := hr ) ;
// kontroluje bezchybnost jednotlivých kroků
38     // Lokalizace vnějšího průměru
39     hrFunc := F_VN_LocateEllipseExp (
//Lokalizování Elipsy
40     ipSrcImage       := ipImageIn ,
//Zdrojový obrázek, z kterého se vychází
41     stEllipse         := stEllipse2 ,
//Neznámá do které vrátí údaje o elipse, lze je pak vyčítat
například stEllipse.aCenter pro pozici středu
42     aCenter           := gvl . aCenter ,
//Očekávaný střed elipsy
43     fSearchRadius     := 250 ,
//poloměr kolem středu pro hledání hran
44     eEdgeDirection    := TCVN_ED_LIGHT_TO_DARK
, // specifikace jak má hledat // := součást je tmavá, na světlém
pozadí
45     fMinStrength      := 10 ,
//Specifikace minimální pevnosti(rozdílu hran)
46     nMaxThickness     := 3 ,
//maximální tloušťka okraje ve kterém má hledat
47     bInvertSearchDirection := TRUE ,
// pokud je True, hledá od vnějšku směrem ke středu,
pokud False naopak
48     fMinSearchRadius := 10 ,
// poloměr od středu který má přeskočit a nehledat v
něm
49     nSubpixelsIterations := 5 ,
// maximální počet iterací pro optimalizaci parametrů
50     nSearchLines      := 40 ,
// body kružnice, podm.: min 8, dělitelné 4, čím víc
bodů, tím delší čas
51     fApproxPrecision := 0.001 ,
// přesnost aproximace
52     eAlgorithm       :=
TCVN_EDA_INTERPOLATION , //výběr algoritmu pro detekci hran
53     ipContourPoints  := ipContourPoints ,
//vrátí subpixelový přesný obrys, pokud není
vyžadován nastavit na 0, pokud je vyžadován ipContourPoints

```

POU: POU_Kamera

```

54             hrPrev             := hr );
                    // kontroluje bezchybnost jednotlivých kroků
55             // ZAKRESLENÍ VÝSLEDKŮ DO VIZUALIZACE
56             //odkaz pro co vše se dá vykreslit
                (https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/11084167691.html?id=
57             // preved obrázek z černobílé do barevné
58             hr := F_VN_ConvertColorSpace ( ipImageIn , ipImageRes ,
TCVN_CST_GRAY_TO_RGB , hr );
59             // zakreslení nalezených hodnot
60             hr := F_VN_DrawPoint ( REAL_TO_UDINT ( stEllipse . aCenter [ 0
] ) , REAL_TO_UDINT ( stEllipse . aCenter [ 1 ] ) , ipImageRes , TCVN_DS_PLUS ,
aColorGreen , hr );
61             hr := F_VN_DrawEllipse ( stEllipse , ipImageRes ,
aColorGreen , 2 , hr );
62             hr := F_VN_DrawCircle ( REAL_TO_UDINT ( gvl . aCenter [ 0 ] ) ,
REAL_TO_UDINT ( gvl . aCenter [ 1 ] ) , 2 , ipImageRes , aColorBlue
, 2 , hr );
63             //Výpočty
64             n_HMI_RadiusMinimum := ( MIN ( stEllipse . stSize . fHeight ,
stEllipse . stSize . fWidth ) / 2 ) * 0.75 ;
65             n_HMI_RadiusMaximum := ( MAX ( stEllipse . stSize . fHeight ,
stEllipse . stSize . fWidth ) / 2 ) * 1.5 ;
66             // Zakreslení kružnic pro min,max
67             hr := F_VN_DrawCircle ( REAL_TO_UDINT ( stEllipse . aCenter [ 0
] ) , REAL_TO_UDINT ( stEllipse . aCenter [ 1 ] ) , REAL_TO_UDINT (
n_HMI_RadiusMinimum ) , ipImageRes , aColorRed , 2 , hr );
68             hr := F_VN_DrawCircle ( REAL_TO_UDINT ( stEllipse . aCenter [ 0
] - 10 ) , REAL_TO_UDINT ( stEllipse . aCenter [ 1 ] ) ,
REAL_TO_UDINT ( n_HMI_RadiusMaximum ) , ipImageRes , aColorRed ,
2 , hr );
69             // uložení obrázku pro vyčtení na vizualizaci
70             fbWriteImage (
71                 ipImage := ipImageRes ,
72                 sFilePath :=
' C:\TwinCAT\3.1\Boot\Plc\Port_851\Visu\40_Servis.bmp' ,
73                 bWrite := TRUE
74             );
75             IF NOT ( fbWriteImage . bError AND fbWriteImage . bBusy )
THEN
76                 gvl . n_VerzeObrazku := gvl . n_VerzeObrazku + 1 ;
77                 fbWriteImage ( sFilePath := '' , bWrite :=
FALSE );
78             END_IF
79
80             gvl . b_SERVIS_Kamera := FALSE ;
81             // Zobrazení upraveného obrázku
82             hr := F_VN_TransformIntoDisplayableImage ( ipImageRes ,
ipImageResDisp , S_OK );
83             //ukončení Acquisition
84             hr := fbCamera . StopAcquisition ( ) ;
85

```

POU: POU_Kamera

```

86         END_IF
87         //*****
AUTOMAT*****
88         IF gvl . b_AUTOMAT_Spusten THEN                                //Pokud je
spuštěný automat
89             stEllipse . stSize . fHeight := 0 ;
90
91             hrFunc := F_VN_LocateEllipseExp (
92                 //Lokalizování Elipsy
93                 ipSrcImage := ipImageIn ,
//Zdrojový obrázek, z kterého se vychází
94                 stEllipse := stEllipse ,
//Neznámá do které vrátí údaje o elipse, lze je pak vyčítat
například stEllipse.aCenter pro pozici středu
95                 aCenter := gvl . aCenter ,
//Očekávaný střed elipsy
96                 fSearchRadius := 250 ,
//poloměr kolem středu pro hledání hran
97
98                 eEdgeDirection := TCVN_ED_LIGHT_TO_DARK
, // specifikace jak má hledat // := součást je tmavá, na světlém
pozadí
99                 fMinStrength := 10 ,
//Specifikace minimální pevnosti(rozdílu hran)
100                nMaxThickness := 3 ,
//maximální tloušťka okraje ve kterém má hledat
101                bInvertSearchDirection := FALSE ,
// pokud je True, hledá od vnějšku směrem ke středu,
pokud False naopak
102                fMinSearchRadius := 10 ,
// poloměr od středu který má přeskočit a nehledat v
něm
103                nSubpixelsIterations := 5 ,
// maximální počet iterací pro optimalizaci parametrů
104                nSearchLines := 40 ,
// body kružnice, podm.: min 8, dělitelné 4, čím víc
bodů, tím delší čas
105                fApproxPrecision := 0.001 ,
// přesnost aproximace
106                eAlgorithm :=
TCVN_EDA_INTERPOLATION , //výběr algoritmu pro detekci hran
107                ipContourPoints := 0 ,
//vrátí subpixelový přesný obrys, pokud není vyžadován
nastavit na 0, pokud je vyžadován ipContourPoints
108                hrPrev := hr ) ;
// kontroluje bezchybnost jednotlivých kroků
109                // preved obrázek z černobílé do barevné
hr := F_VN_ConvertColorSpace ( ipImageIn , ipImageRes ,
TCVN_CST_GRAY_TO_RGB , hr ) ;
110                IF ( gvl . fMaxRaduisVyhodnoceni > ( MAX ( stEllipse .
stSize . fHeight / 2 , stEllipse . stSize . fWidth / 2 ) ) ) AND ( ( MIN ( stEllipse .

```

POU: POU_Kamera

```

stSize.fHeight / 2, stEllipse.stSize.fWidth / 2)) > gvl.
fMinRaduisVyhodnoceni) THEN
110     gvl.b_SoucastNatocena_OK := TRUE;
111     F_VN_PutTextExp ('Soucast je natocena OK',
ipImageRes, 50, 80, TCVN_FT_HERSHEY_COMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE, hr);
112     ELSE
113         IF stEllipse.stSize.fHeight <> 0 THEN
114             gvl.b_SoucastNatocena_NOK := TRUE;
115             F_VN_PutTextExp ('Soucast je natocena NOK',
ipImageRes, 50, 80, TCVN_FT_HERSHEY_COMPLEX, 1.3, aColorRed, 2,
TCVN_LT_8_CONNECTED, FALSE, hr);
116         ELSE
117             gvl.bErrors := TRUE;
118             F_VN_PutTextExp ('Soucast nerozpoznana',
ipImageRes, 50, 80, TCVN_FT_HERSHEY_COMPLEX, 1.3, aColorRed, 2,
TCVN_LT_8_CONNECTED, FALSE, hr);
119         END_IF
120     END_IF
121
122     // uložení obrázku pro vyčtení na vizualizaci
123     fbWriteImage (
124         ipImage      := ipImageRes,
125         sFilePath    :=
'C:\TwinCAT\3.1\Boot\Plc\Port_851\Visu\41_Automat.PNG',
126         bWrite       := TRUE
127     );
128     IF NOT (fbWriteImage.bError AND fbWriteImage.bBusy)
129         THEN
130             gvl.n_VerzeObrazku := gvl.n_VerzeObrazku + 1;
131             fbWriteImage (sFilePath := '', bWrite :=
FALSE );
132         END_IF
133
134     // Zobrazení upraveného obrázku
135     hr := F_VN_TransformIntoDisplayableImage (ipImageRes,
ipImageResDisp, S_OK);
136     fbWriteImage.bWrite := FALSE;
137     gvl.b_AUTOMAT_Spusten := FALSE;
138     //ukončení Acquisition
139     hr := fbCamera.StopAcquisition ();
140     END_IF
141
142     //*****
143     END_IF ;
END_CASE ;

```