

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2025

Dominik Koláčný



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

REFEROVÁNÍ SCARA ROBOTA POMOCÍ KAMERY

HOMING OF SCARA ROBOT USING A CAMERA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Dominik Koláčný

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jakub Arm, Ph.D.

BRNO 2025

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Dominik Koláčný

ID: 247394

Ročník: 3

Akademický rok: 2024/25

NÁZEV TÉMATU:

Referování SCARA robota pomocí kamery

POKYNY PRO VYPRACOVÁNÍ:

Úkolem je navrhnout a integrovat modul, který bude umožňovat zreferování SCARA robotického manipulátoru pomocí kamery připevněné na jeho koncovém bodě. Je žádoucí použít standardní nebo průmyslovou kameru a implementovat obslužný SW do IPC, který ovládá robot, a to v prostředí Python a také pomocí vhodných TwinCAT3 pluginů.

1. Návrh referovacího modulu.
2. Rešerše a výběr kamery.
3. Realizace modulu v TwinCAT3.
4. Realizace modulu v Python.
5. Vyhodnocení vlastností modulu a dokumentace.

DOPORUČENÁ LITERATURA:

SMEJKAL, L.. PLC a automatizace 2. BEN, 2011. 208 s. ISBN: 978-80-7300-389-0.

Termín zadání: 10.2.2025

Termín odevzdání: 28.5.2025

Vedoucí práce: doc. Ing. Jakub Arm, Ph.D.

Ing. Miroslav Jirgl, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

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

Abstrakt

Tato bakalářská práce se zabývá návrhem a integrací modulu, který bude umožňovat zreferování SCARA robotického manipulátoru pomocí kamery připevněné na jeho koncovém bodě. Řešení částečně nahradí tradiční referování pomocí snímačů na osách robota systémem s detekcí referenčních značek pomocí zpracování obrazu. Bude proveden návrh referovacího modulu jak v softwarové, tak v hardwarové podobě. Dále bude vyhotovena rešerše pro výběr kamery. Součástí práce bude výběr vhodné kamery, návrh softwarového modulu v prostředí TwinCAT i v kombinaci s jazykem Python, návrh fyzického držáku kamery a rozmístění značek.

Klíčová slova

SCARA, kamera, zpracování obrazu, Python

Abstract

This bachelor's thesis focuses on the design and integration of a module that enables referencing of a SCARA robotic manipulator using a camera mounted on its end-effector. The solution will partially replace traditional referencing methods, which use sensors on the robot's axes, with a system that detects reference markers through image processing. The referencing module will be designed both in software and hardware form. The thesis will include the selection of an appropriate camera, the design of a software module in the TwinCAT environment in combination with the Python language, and the design of a physical camera mount and the layout of the markers.

Keywords

SCARA, camera image processing, Python

Bibliografická citace

KOLÁČNÝ, Dominik. *Referování SCARA robota pomocí kamery*. Online, bakalářská práce. Jakub ARM (vedoucí práce). Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2025. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/168189>. [cit. 2025-05-25].

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	Dominik Koláčný
VUT ID studenta:	247394
Typ práce:	Bakalářská práce
Akademický rok:	2024/25
Téma závěrečné práce:	Referování SCARA robota pomocí kamery

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

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

V Brně dne: 24. května 2025

podpis autora

Poděkování

Děkuji vedoucímu mé bakalářské práce doc. Ing. Jakubu Armovi Ph.D., který mi poskytoval odbornou pomoc při zpracovávání práce.

V Brně dne: 24. května 2025

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. VÝBĚR KAMERY.....	12
1.1 PARAMETRY KAMER.....	12
1.1.1 Ohnisková vzdálenost.....	12
1.1.2 Typ komunikačního rozhraní	13
1.1.3 Světelnost objektivu.....	13
1.1.4 Rozlišení.....	14
1.1.5 Snímkovací frekvence.....	14
1.2 PRŮMYSLOVÉ KAMERY.....	14
1.2.1 Výběr konkrétní kamery.....	15
1.2.2 Výběr objektivu	15
1.3 BECKHOFF KAMERY	16
1.4 IP KAMERY	17
1.5 WEBKAMERY.....	18
1.6 POROVNÁNÍ VYBRANÝCH KAMER.....	20
2. NÁVRH A IMPLEMENTACE REFEROVACÍHO MODULU V TWINCAT.....	21
2.1 NÁVRH SOFTWAREVÉHO MODULU V PROSTŘEDÍ TWINCAT.....	21
2.1.1 Struktura programu	21
2.1.2 Návrh homingové procedury manipulátoru Homing	22
2.1.3 Popis fungování funkčního bloku VISION	23
2.1.4 Hlavní programová smyčka MAIN.....	24
2.2 REALIZACE MODULU V TWINCAT3	24
2.2.1 Konfigurace VISION modulu	25
2.2.2 GigE Vision Camera	26
2.2.3 File Source.....	26
2.3 PRÁCE S NAČTENÝMI SNÍMKY V PROGRAMU	27
2.4 PROGRAM ZPRACOVÁVAJÍCÍ OBRAZ	28
2.4.1 F_VN_MatchTemplate	29
3. INTEGRACE PYTHON SKRIPTŮ PRO ZPRACOVÁNÍ OBRAZU A KOMUNIKACI	30
3.1 MAIN.....	30
3.2 READIMAGEBASLER	30
3.3 FINDMARK	30
3.3.1 Funkce HledaniRotaci.....	32
3.3.2 matchTemplate.....	32
3.4 FUNKCE ROZPOZNANI	35
3.5 DOHOMINGTEST	35
3.6 OFFSETCOMM	36
3.7 LOCComm.....	36
3.8 KONFIGURACE MODBUS KOMUNIKACE S PROSTŘEDÍM TWINCAT	37

4.	NÁVRH HARDWAROVÉHO ŘEŠENÍ	38
4.1	UMÍSTĚNÍ KAMERY	38
4.1.1	<i>Návrh držáku kamery</i>	<i>40</i>
4.2	REFERENČNÍ ZNAČKY	42
4.3	ROZMÍSTĚNÍ REFERENČNÍCH ZNAČEK	42
5.	IMPLEMENTACE	44
5.1	TESTOVÁNÍ.....	44
5.2	TESTOVÁNÍ ZNAČEK ZA RŮZNÝCH SVĚTELNÝCH PODMÍNEK	44
5.2.1	<i>Testování s maximálním osvětlením.....</i>	<i>44</i>
5.2.2	<i>Testování s minimálním osvětlením</i>	<i>45</i>
5.2.3	<i>Testování se změnou osvětlení</i>	<i>46</i>
5.3	POROVNÁNÍ SOFTWAREVÝCH ŘEŠENÍ.....	47
5.4	DALŠÍ MOŽNOSTI VYUŽITÍ	48
5.5	KOMPLIKACE PŘI IMPLEMENTACI NA HARDWARE.....	48
5.6	ÚČEL NAVRŽENÉHO SYSTÉMU	48
6.	ZÁVĚR.....	50
	SEZNAM SYMBOLŮ A ZKRATEK	53
	SEZNAM PŘÍLOH.....	54

SEZNAM OBRÁZKŮ

Obrázek 1.1 Ohnisková vzdálenost od čočky [3].....	12
Obrázek 1.2 Kamera řady Basler ace [10]	15
Obrázek 1.3 Objektiv C125-0818-5M [11].....	16
Obrázek 1.4 Kamera řady VCS2000 [12].....	17
Obrázek 1.5 Kamera VIGI C350 [14].....	18
Obrázek 1.6 Kamera C992 PRO HD [15].....	19
Obrázek 2.1 Souřadnicový systém robotu [16].....	21
Obrázek 2.2 Diagram stavového automatu homing procedury	23
Obrázek 2.3 Struktura hlavní programové smyčky MAIN	24
Obrázek 2.4 Vytvoření vision modulu	25
Obrázek 2.5 Přidání zdroje snímků.....	25
Obrázek 2.6 Výběr typu zdroje snímků	26
Obrázek 2.7 Kontrola správné konfigurace kamery	26
Obrázek 2.8 Načtení snímků při variantě File Source	27
Obrázek 3.1 Značka hledaná v první iteraci funkce findMark	30
Obrázek 3.2 Vývojový diagram funkce findMark	31
Obrázek 3.3 Načtený obraz z kamery	33
Obrázek 3.4 Mapa shod s obecnou referenční značkou	33
Obrázek 3.5 Výsledný obraz po nalezení maximální shody	34
Obrázek 3.6 Diagram funkce Rozpoznani	35
Obrázek 3.7 Diagram funkce OffsetComm	36
Obrázek 3.8 Tabulka s adresami modbus registrů [20].....	37
Obrázek 4.1 Uložení kamery na manipulátoru [21].....	38
Obrázek 4.2 Díry pro připevnění držáku kamery.....	39
Obrázek 4.3 Držák s chladičem pro kamery Sugar Cube [22]	39
Obrázek 4.4 Skutečné umístění kamery.....	40
Obrázek 4.5 Držák na kameru, pohled zepředu	41
Obrázek 4.6 Držák kamery, pohled z boku.....	41
Obrázek 4.7 Držák kamery, pohled shora.....	41
Obrázek 4.8 Držák kamery, perspektiva.....	41
Obrázek 4.9 Základ referenční značky	42
Obrázek 4.10 Příklady referenčních značek	42
Obrázek 4.11 Rozložení referenčních značek na pracovní ploše	42
Obrázek 4.12 Pracovní plocha manipulátoru Epson H55BN [21]	43
Obrázek 5.1 Výstupní obraz testování značky č. 5 na odlesky	45
Obrázek 5.2 Výstupní obraz testování značky č. 12 s minimálním osvětlením.....	46

SEZNAM TABULEK

Tabulka 1.1 Porovnání parametrů vybraných kamer	20
Tabulka 4.1 Rozměry pracovního prostoru [21]	43
Tabulka 5.1 Testování značky č. 12 se změnami osvětlení	46
Tabulka 5.2 Testování značky č. 23 se změnami osvětlení	47

ÚVOD

Cílem bakalářské práce je navrhnout a doplnit stávající referovací systém SCARA robotu, který pracuje s vestavěnými snímači na pohonech jednotlivých os, systémem využívajícím kamery snímající plochu pod kamerou, kde jsou umístěny značky pro referování, vybrat vhodnou průmyslovou kameru a implementovat obslužný software, a to v prostředí Python a také pomocí vhodných TwinCAT 3 pluginů. Tento program bude doplněn o homing proceduru využívající zpětnou vazbu z kamery.

Navržený kamerový referenční systém je koncipován ve dvou po sobě následujících fázích. V první fázi systém zajišťuje hrubé určení orientace robota po jeho inicializaci, což zahrnuje přiblížení výchozí polohy k předem definované referenční značce. Ve druhé fázi je realizováno jemné referování prostřednictvím přesného navedení robota na značku, čímž se dosahuje vysoké přesnosti konečné pozice. V rámci práce bude také ověřena spolehlivost referování pouze tímto kamerovým způsobem, bez využití klasických snímačů.

Práce bude členěna do několika částí, ve kterých bude nejprve provedena rešerše zaměřená na výběr kamery z kategorií průmyslových kamer, kamer Beckhoff, IP kamer a webkamer. U průmyslových kamer bude rovněž vybrán vhodný objektiv. Mezi kritéria výběru kamer budou patřit parametry, jako zorné pole kamery, její cena a typ komunikace.

Dále bude proveden softwarový návrh systému referování - nejprve kompletně v prostředí TwinCAT, následně kombinací programu částečně napsaného v jazyce Python a částečně v prostředí TwinCAT. Poté bude zpracován hardwarový návrh referovacího modulu, konkrétně umístění kamery a referenčních značek. Na závěr budou diskutovány výhody a nevýhody obou variant softwarových řešení.

Na modul TwinCAT Vision se zaměřuje práce paní Anny Furciniti. Práce dokazuje, že TwinCAT Vision zlepšuje automatizaci procesů díky zpracování obrazu v reálném čase, synchronizaci algoritmů a možnosti paralelních výpočtů na více jádrech. Klíčovým výsledkem je spolehlivá detekce objektů, která zvyšuje propustnost výrobních linek. Technologie minimalizuje zpoždění a zvyšuje efektivitu. [1]

Studie Camera-Assisted Calibration of SCARA Arms se zaměřuje na kamerou asistovanou kalibraci SCARA robotů s cílem zvýšit přesnost manipulace. Kalibrace zahrnovala opakované měření pozic robota pomocí kamery v různých konfiguracích. Bylo dosaženo maximální chyby na obrazové rovině pod 0,4 pixelu. Největší chyby se objevily v ose z, což je méně důležité pro běžné aplikace SCARA. Studie rovněž doporučuje optimální parametry a metody pro efektivní kalibraci a rozšíření pracovního prostoru robota. [2]

1. VÝBĚR KAMERY

Při výběru vhodné kamery pro modul sloužící k referenci (zreferování) SCARA robotu je třeba zohlednit následující hlediska: cena, ohnisková vzdálenost, komunikace a světelnost objektivu. Konkrétní typ kamery bude vybrán z několika základních kategorií: průmyslové kamery, kamery Beckhoff, IP kamery a “komerční“ webkamery.

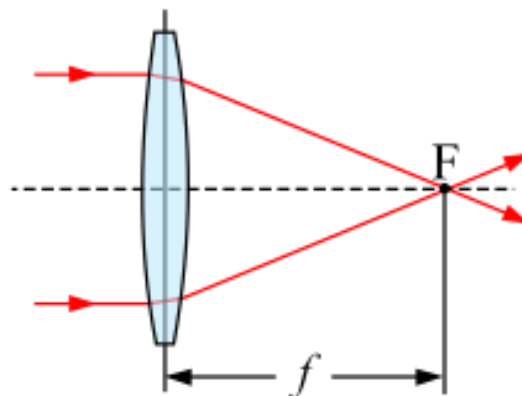
1.1 Parametry kamer

Aby bylo možné správně posoudit vhodnost jednotlivých typů kamer pro použití v systému referování SCARA robotu, je nutné porozumět klíčovým technickým parametrům, které ovlivňují výslednou funkčnost. Tyto parametry určují nejen kvalitu získaného obrazu, ale i možnosti integrace kamery do stávajícího systému, spolehlivost přenosu dat a celkový výkon celého modulu. Následující části se zaměřují na nejdůležitější z nich, které byly brány v úvahu při výběru konkrétního řešení.

Parametry, jež nejsou pro daný systém relevantní – jako například krytí (IP ochrana), provozní teplota nebo bitová hloubka senzoru – byly v rámci rešeršní části vynechány, neboť neovlivňují funkčnost systému v laboratorních podmínkách, ve kterých je zařízení provozováno.

1.1.1 Ohnisková vzdálenost

Ohnisková vzdálenost (f) je definována jako vzdálenost mezi optickým středem čočky a ohniskem, tedy bodem, ve kterém se protínají paprsky procházející čočkou (viz Obrázek 1.1).



Obrázek 1.1 Ohnisková vzdálenost od čočky [3]

Ohnisková vzdálenost je ovlivněna výběrem konkrétního objektivu kamery a přímo určuje velikost zorného pole kamery. Pro tuto aplikaci je vhodné použít objektiv s co nejmenší ohniskovou vzdáleností, aby bylo dosaženo co nejširšího zorného pole. Tím se docílí stavu, kdy bude na ploše pod kamerou potřeba co nejméně referenčních značek. Při

uvažované šířce snímané plochy 25 cm a pracovní vzdálenosti 50 cm lze potřebnou ohniskovou vzdálenost spočítat podle následující rovnice:

$$f = \frac{h \cdot WD}{hFoV} = \frac{4,22 \cdot 500}{250} = 8,44 \text{ mm} , \quad (1.1)$$

Kde:

- f je ohnisková vzdálenost
- WD je pracovní vzdálenost
- $hFoV$ je šířka zorného
- h je šířka senzoru kamery (v tomto případě 4,22 mm).

Tento výpočet se týká pouze kamer bez integrovaného objektivu. V této práci tedy výhradně průmyslové kamery, pro kterou nám z výpočtu vyplývá ohnisková vzdálenost 8,4mm nebo menší.

1.1.2 Typ komunikačního rozhraní

Různé typy kamer komunikují prostřednictvím různých komunikačních rozhraní. V oblasti počítačového vidění jsou běžně používána připojení skrze rozhraní jako Camera-Link, USB, FireWire, Gigabitový ethernet a další [4]. Pro tuto aplikaci je nejvhodnějším řešením použití ethernetového propojení GigE, jelikož představuje nejjednodušší způsob připojení k použitému Beckhoff IPC.

GigE Vision je průmyslový standard zavedený v roce 2006 pro přenos videa přes ethernetovou síť. Jeho hlavní výhodou je využití běžné síťové infrastruktury (Gigabit Ethernet), což umožňuje snadnou integraci bez potřeby speciálního hardwaru. Standard definuje protokoly pro detekci zařízení, konfiguraci, streamování a zpracování událostí. Díky UDP nabízí vysokou přenosovou rychlost a nízkou latenci. [5]

1.1.3 Světelnost objektivu

Světelnost objektivu vyjadřuje, jaké množství světla je objektiv schopen zachytit a soustředit fotocitlivý prvek [6]. Obvykle se udává pomocí clonového čísla ve tvaru f/x , kde nižší hodnota x značí vyšší světelnost, tedy schopnost objektivu propustit více světla. Objektivy s vyšší světelností jsou výhodné zejména při zhoršených světelných podmínkách, kdy umožňují kratší expoziční časy a lepší kvalitu obrazu bez nutnosti zvyšování citlivosti senzoru.

V případě použití kamery s vlastním nebo řízeným osvětlením scény (např. LED přisvětlení) však tento parametr částečně ztrácí na významu, jelikož lze světelné podmínky uměle optimalizovat. S ohledem na skutečnost, že navrhovaný systém bude provozován v laboratorních podmínkách s kontrolovaným osvětlením, není světelnost objektivu rozhodujícím faktorem při jeho výběru.

Použitý objektiv disponuje manuálně nastavitelnou clonou, což umožňuje jemně upravit množství procházejícího světla v závislosti na konkrétní scéně nebo požadované hloubce ostrosti.

1.1.4 Rozlišení

Rozlišení kamery představuje počet obrazových bodů (pixelů), které je senzor kamery schopen zaznamenat. Obvykle se udává jako součin počtu pixelů na šířku a na výšku (např. 1920×1080). Vyšší rozlišení znamená větší množství detailů zaznamenaných v obraze, což je zásadní pro úlohy vyžadující přesné vizuální rozpoznání.

V kontextu této práce, která se zabývá referenčním systémem pro navigaci robota, hraje rozlišení klíčovou roli. Kamera s vyšším rozlišením umožňuje přesnější detekci referenčních značek, a to i na větší vzdálenosti nebo za zhoršených světelných podmínkách. Vyšší hustota pixelů napomáhá lepší identifikaci tvarů, hran a kontrastních přechodů, což přispívá k vyšší robustnosti a přesnosti celého systému. Výběr kamery s adekvátním rozlišením je proto nezbytným předpokladem pro dosažení spolehlivé a opakovatelné funkce referenčního systému.

1.1.5 Snímkovací frekvence

Snímkovací frekvence udává, kolik snímků za sekundu je kamera schopna zachytit. Pro funkci referování není tento parametr příliš zásadní, neboť frekvence načítání snímků do programu je především ovlivněna rychlostí chodu samotného programu. Při referování se robot navíc pohybuje nízkou rychlostí, což rovněž poskytuje určitou toleranci vůči nižší snímkovací frekvenci.

1.2 Průmyslové kamery

Jsou navrženy pro fungování v průmyslu, poskytují potřebnou kvalitu, rychlost a odolnost. Využívají se především k sledování výroby, kontrolu kvality a měření. Podle typu zpracování obrazu je lze rozdělit do několika skupin jako například: Plošné kamery, Řádkové kamery, 3D kamery, nebo mikroskopové kamery. Pro moduly, jako je tento se používají kamery s plošným snímáním obrazu. Aby nedošlo k záměně s IP kamerami, mohou být průmyslové kamery označovány, jako GigE kamery [7], nicméně toto označení se používá především pro kamery užívající gigabitový ethernet.

Průmyslové kamery se vyznačují následujícími vlastnostmi:

- **Vysoká kvalita obrazu** – může dosahovat až 65 MP [8]
- **Velká přenosová rychlost** – některé technologie jsou závislé na vysoké rychlosti operací a proto je žádoucí, aby byla rychlost komunikace kamery a řídicího členu co nejrychlejší. Rychlosti přenosu informací se dělí podle použitého rozhraní (USB, GigE, FireWire, atd...). Přičemž přenosové rychlosti kamer s propojením na bázi ethernetu, vhodných pro tuto práci mohou dosahovat až 10 Gb/s [9].
- **Spolehlivost** – kamery se používají v průmyslu, což často znamená náročné prostředí s velkým rušením, vysoké, nebo naopak nízké teploty, vysoká vlhkost atd...

- **Vyšší cena** – ceny průmyslových kamer se pohybují v rozmezí od 3 000 Kč až do cen přesahujících 100 000 Kč. U těchto dražších kamer se však jedná o vysoce specializovaná zařízení.

1.2.1 Výběr konkrétní kamery

Na základě vyhodnocení parametrů, jako jsou cena, přenosová rychlost a kvalita obrazu, byla identifikována rozsáhlá množina kamer vhodných pro danou aplikaci, ze které je možné dále vybírat.

V případě výběru kamery od renomovanějšího výrobce lze jako vhodný model uvažovat **Basler ace Classic acA1920-25gm**.



Obrázek 1.2 Kamera řady Basler ace [10]

Kamera se vyznačuje následujícími specifiky:

- Cena: €239 (přibližně 6043,69 Kč dle kurzu 1 € = 25,29 Kč k datu 18.11.2024)
- Komunikace: GigE, Gigabitový Ethernet (1000 Mbit/s)
- Rozlišení: 1920 x 1080 pixelů (2MP)
 - jeden pixel odpovídá 2,2 μm x 2,2 μm
- Snímkovací frekvence: 25 FPS
- Monochromatický výstup
- Technický výkres kamery viz. Příloha A -Technický výkres kamery

Kamera je napájena skrze PoE systém (Power on Ethernet) a zpracování obrazu zajišťuje implementovaný CMOS MT9P031 senzor. Pro instalaci objektivu je kamera vybavena závitěm typu C-mount. [10]

1.2.2 Výběr objektivu

Vhodný objektiv pro danou aplikaci lze vybírat na základě několika kritérií, jako jsou cena, pracovní vzdálenost, kompatibilita s kamerou a ohnisková vzdálenost. Určení vhodné ohniskové vzdálenosti bylo podrobněji diskutováno v kapitole 1.1.1 Ohnisková vzdálenost.

Pro zjednodušení výběru bude uvažován objektiv od stejného výrobce jako zvolená kamera. Po zohlednění všech relevantních parametrů, jako jsou cena, typ závitového

upevnění a ohnisková vzdálenost, byl vybrán objektiv **Basler Lens C125-0818-5M**. Tento objektiv se vyznačuje následujícími specifiky:

- Cena: €139 (přibližně 3495,02 Kč dle kurzu 1 € = 25,14 Kč k datu 25.12.2024)
- Ohnisková vzdálenost: 8 mm
- Minimální pracovní vzdálenost: 0,1 m
- Optimální pracovní vzdálenost: 0,5 m
- Světelnost: f/1.8
- Typ závitu: C-mount
- Mechanické nastavení clony a ostření s možností aretace



Obrázek 1.3 Objektiv C125-0818-5M [11]

Výpočet horizontálního zorného pole:

$$hFoV = \frac{h \cdot WD}{f} = \frac{4,22 \cdot 500}{8} = 263,75 \text{ mm} , \quad (1.2)$$

Kde:

- h je horizontální velikost senzoru
- WD je pracovní vzdálenost
- f je ohnisková vzdálenost.

1.3 Beckhoff kamery

Společnost **Beckhoff** rozšířila své portfolio o řadu průmyslových kamer **VCS2000 Area Scan** s přenosovou rychlostí až 2,5 Gbit/s. Tyto kamery jsou navrženy pro náročné průmyslové aplikace v oblasti vizuální inspekce a automatizace. Mezi hlavní technické parametry těchto kamer patří:

- Kvalita obrazu – 2,3 MP až 24,6 MP
- Přenosová rychlost – až 2,5 GBit/s
- Rozlišení – 1920 x 1200 až 5328 x 4608 pixelů
- Snímkovací frekvence: 11 FPS až 167 FPS

Výběr konkrétního modelu z této řady je složitější především z důvodu absence veřejně dostupných cenových údajů, které představují důležitý faktor při rozhodování. Při nezohlednění ceny se jako technicky nejvhodnější varianta jeví model **VCS2020-1200**, jenž nabízí optimální poměr mezi snímkovací frekvencí a kvalitou obrazu.



Obrázek 1.4 Kamera řady VCS2000 [12]

Specifika kamery:

- Cena: Neznámá
- Komunikace: EtherCAT
- Rozlišení: 4128 x 3008 pixelů (12,4 MP)
 - jeden pixel odpovídá 2,74 μm x 2,74 μm
- Snímkovací frekvence: 23 FPS
- Monochromatický výstup

Kamera je napájena skrze EtherCAT P, zpracování obrazu zajišťuje CMOS senzor Sony IMX565 [12].

1.4 IP kamery

IP kamera (kamera komunikující prostřednictvím internetového protokolu) je typ digitální videokamery, který je funkčně podobný běžné webkameře. Na rozdíl od ní však komunikuje přes síťové rozhraní a disponuje vlastní IP adresou, díky čemuž se v síti chová jako samostatné zařízení (např. podobně jako tiskárna či skener).

Tyto kamery nejsou primárně určeny pro použití v průmyslových aplikacích, avšak jsou široce využívány v oblasti zabezpečovacích systémů a monitoringu [13]. Jejich hlavní výhodou oproti průmyslovým kamerám je výrazně nižší pořizovací cena, což může být rozhodujícím faktorem v méně náročných aplikacích.

V kontextu této práce by mohla být jako zástupce této kategorie uvažována například kamera **VIGI C350**, která nabízí vhodný poměr ceny a základních technických parametrů.



Obrázek 1.5 Kamera VIGI C350 [14]

Specifika kamery:

- Cena: 2559 Kč
- Komunikace: Ethernet
- Rozlišení: 1920 x 1080 pixelů (2 MP)
- Snímkovací frekvence: 20 FPS
- Barevný výstup

Kamera je napájena prostřednictvím technologie PoE (Power over Ethernet) a pro snímání obrazu využívá CMOS senzor. Je vybavena vestavěným objektivem s ohniskovou vzdáleností 6 mm, která je konfigurovatelná až na 2,8 mm. Díky integrovanému infračervenému přísvitu je kamera vhodná i pro provoz za zhoršených světelných podmínek nebo úplné tmy.

Vzhledem k tomu, že nejsou známy přesné rozměry senzoru, bude pro výpočet horizontálního zorného pole ($hFoV$) využit úhel záběru $\alpha = 56^\circ$ a pracovní vzdálenost $WD = 500$ mm [14]. Výpočet bude vycházet z trigonometrických vztahů, které umožňují přibližně určit šířku sledované scény v dané vzdálenosti.

$$hFoV = 2 * \tan\left(\frac{\alpha}{2}\right) * WD = 2 * \tan\left(\frac{56}{2}\right) * 500 = 531,7 \text{ mm} \quad (1.3)$$

Horizontální zorné pole je tedy dostačující.

1.5 Webkamery

Webové kamery, často označované jako webkamery, jsou digitální kamery běžně používané pro videohovory a základní snímání obrazu prostřednictvím osobních

počítačů. V oblasti robotiky a automatizace představují cenově nejdostupnější řešení, které je vhodné zejména pro méně náročné úlohy nebo pro počáteční fázi testování a vývoje.

Tyto kamery často disponují funkcemi, jako je automatická expozice, vyvážení bílé nebo automatické ostření. Oproti průmyslovým kamerám však zpravidla nenabízejí detailní možnosti manuální konfigurace, a to ani na úrovni hardwaru, ani prostřednictvím softwarového rozhraní. Jejich hlavními nevýhodami jsou nižší robustnost, omezená odolnost vůči průmyslovému prostředí a nižší dlouhodobá spolehlivost.

Vzhledem k široké nabídce webkamer na trhu lze pro účely srovnání uvažovat model od zavedeného výrobce. Jako zástupce této kategorie byla vybrána kamera **Logitech C922 Pro HD**, která poskytuje solidní obrazovou kvalitu za příznivou cenu a je běžně dostupná.



Obrázek 1.6 Kamera C922 PRO HD [15]

Specifika kamery:

- Cena: 3 499 Kč
- Komunikace: USB-A
- Rozlišení: 1920 x 1080 pixelů
- Snímkovací frekvence: 30 FPS
- Barevný výstup

Kamera je napájena přímo prostřednictvím rozhraní USB-A a disponuje funkcí automatického ostření (autofocus). Její diagonální zorné pole ($dFoV$) činí 78° . Na základě této hodnoty, známého poměru stran senzoru a rozlišení 1920×1080 pixelů, lze vypočítat horizontální zorné pole ($hFoV$), tedy šířku snímané scény ve vodorovném směru [15].

Nejprve byl určen úhel α , který je svírá mezi úhlopříčkou obrazu a jeho spodní hranou. Tento úhel odpovídá poměru výšky a šířky obrazu (16:9) a byl vypočten podle vztahu:

$$\alpha = \tan^{-1} \left(\frac{\text{Vertikální rozsah obrázku}}{\text{horizontální rozsah obrázku}} \right) = \tan^{-1} \left(\frac{1080}{1920} \right) = 29^\circ 21', \quad (1.4)$$

Následně byla vypočtena délka úhlopříčky snímané plochy v pracovní vzdálenosti $WD = 500 \text{ mm}$, kde γ představuje polovinu diagonálního zorného pole:

$$d = 2 * \tan(\gamma) * WD = 2 * \tan\left(\frac{78}{2}\right) * 500 = 809,78 \text{ mm} , \quad (1.5)$$

Horizontální zorné pole bylo získáno projekcí délky úhlopříčky do vodorovné roviny pomocí kosinové složky:

$$hFoV = \cos(\alpha) * d = \cos(29^\circ 21') * 809,78 = 705,84 \text{ mm} , \quad (1.6)$$

Z výsledku vyplývá, že kamera je schopna snímat scénu o šířce přibližně 706 mm ve vzdálenosti 500 mm od objektu, což je pro zamýšlenou aplikaci více než dostačující.

1.6 Porovnání vybraných kamer

Na základě předchozí analýzy byly vybrány čtyři kamery – vždy jedna z každé uvažované kategorie. Jejich základní parametry jsou uvedeny v následující tabulce:

Tabulka 1.1 Porovnání parametrů vybraných kamer

	Cena [Kč]	Komunikace	Rozlišení	hFoV [mm]
acA1920-25gm	6043,69	GigE	1920 x 1080	263,75
VCS2020-1200		EtherCAT	4128 x 3008	
VIGI C350	2559	Ethernet	1920 x 1080	531,7
C922 PRO	3 499	USB-A	1920 x 1080	705,84

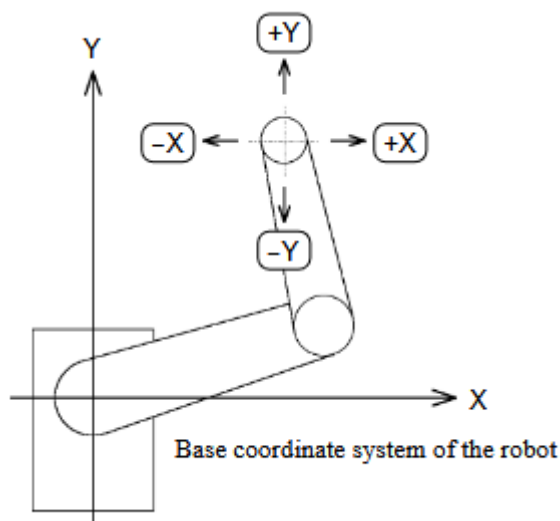
Z tabulky je patrné, že průmyslová kamera *acA1920-25gm* představuje nejdražší variantu a zároveň disponuje nejnižší hodnotou horizontálního zorného pole. Kamera *VCS2020-1200* od společnosti Beckhoff nabízí nejvyšší rozlišení ze všech porovnávaných modelů, avšak ostatní parametry této kamery nejsou v současnosti veřejně dostupné. IP kamera *VIGI C350* je cenově nejdostupnější, zatímco webkamera *C922 PRO* zaujímá druhé místo z hlediska ceny a zároveň dosahuje nejvyššího hodnoty horizontálního zorného pole. Její nevýhodou je však komunikace přes rozhraní USB-A, které není plně kompatibilní s VISION modulem v prostředí TwinCAT 3.

Pro danou aplikaci byla jako nejvhodnější zvolena průmyslová kamera *acA1920-25gm*, a to navzdory vyšší pořizovací ceně. Výběr byl proveden především s ohledem na kompatibilitu komunikačního rozhraní, mechanickou odolnost a provozní spolehlivost, což jsou klíčová kritéria při nasazení kamery v systému robotického manipulátoru.

2. NÁVRH A IMPLEMENTACE REFEROVACÍHO MODULU V TWINCAT

Návrh referovacího modulu lze rozdělit na návrh programové části a návrh fyzického umístění kamery. Při návrhu samotného povelování bylo vycházeno z diplomové práce pana Ing. Pavla Cvička, který se ve své práci zabýval řídicím systémem SCARA robotu.

Stávající kloubový systém řízení pohybu byl nahrazen lineárním systémem pohybu se souřadnicemi vztaženými k základně robotu.



Obrázek 2.1 Souřadnicový systém robotu [16]

2.1 Návrh softwarového modulu v prostředí TwinCAT

Při použití vhodných TwinCAT pluginů bylo využito VISION modulu přímo od společnosti Beckhoff, která vyvíjí vývojové prostředí TwinCAT, společně s vision knihovnami nabízejícími nespočet funkcí zaměřených na zpracování obrazu.

Modul VISION v prostředí TwinCAT umožňuje vyčítání obrazu z kamer komunikujících skrze GigE vision protokol, kamer přímo produkovaných společnostmi Beckhoff, anebo načítáním souborů přímo z uložení zařízení.

Pro zhotovení kompletního referovacího modulu byly vytvořeny dva propojené moduly a sice modul, který zpracovává obraz z kamery a vypočítává odchylku od středu referovací značky a modul, který obstarává samotné řízení směru a rychlosti pohybu robotu. Druhý modul je společný pro řešení s variantou, kdy je zpracování obrazu prováděno v pythonu.

2.1.1 Struktura programu

Program se skládá z podprogramů a funkčních bloků, z nichž některé byly převzaty z práce pana Ing. Pavla Cvička:

- Axis_Control
- Axis_n_Control
- CM_AI
- CM_AO
- Error_reset
- Extruder
- F_ControlPosition
- F_CheckReqPosition
- GCode_Interpreter
- Interpolation
- Kinematic_transformation

A z nově vytvořených programů určených pro realizaci a zpracování VISION instrukcí:

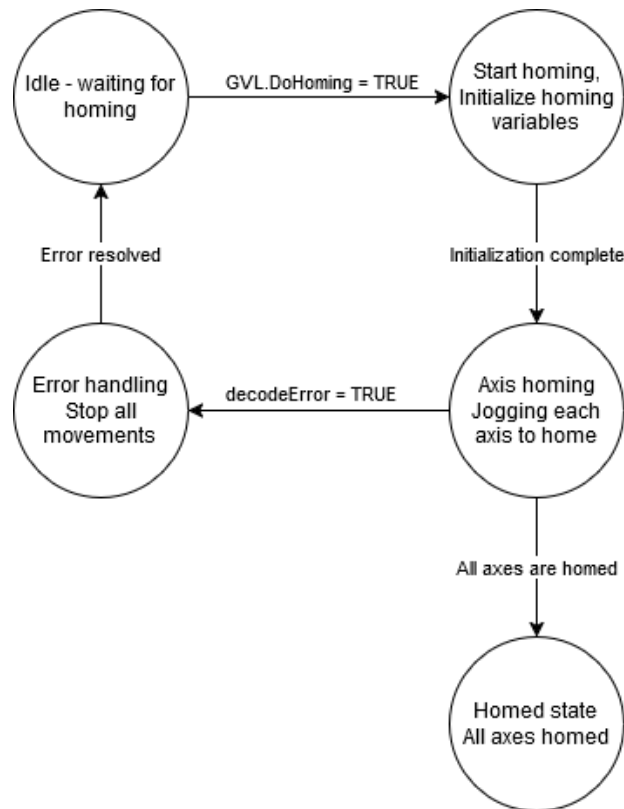
- Homing
- MAIN
- VISION

2.1.2 Návrh homingové procedury manipulátoru Homing

Při návrhu homingové procedury bylo vycházeno z již hotové diplomové práce pana Ing. Pavla Cvička. Nicméně v již vytvořené homing proceduře bylo používáno dat ze snímačů na servopohonech. Tato homing procedura byla realizovaná pomocí stavového automatu a byla přetvořena tím způsobem, že místo dat ze senzorů je použit feedback z kamery (offset v x-ové a y-ové ose od referenční značky).

Nejprve jsou nastaveny klíčové proměnné na inicializační hodnoty a jsou předány hodnoty odchylek od středu v y-ové a x-ové ose. Následně je zde ponechán homing poslední osy, která určuje výšku nad pracovní plochou. Realizace homing procedury této osy pomocí kamery by byl náročný a těžko proveditelný. Dále probíhá homing v x-ové ose a následně v y-ové ose.

Celková struktura homing procedury je znázorněna následujícím diagramem:



Obrázek 2.2 Diagram stavového automatu homing procedury

2.1.3 Popis fungování funkčního bloku VISION

Modul nejprve načte template snímek a uloží do jako proměnnou *ipTemplateImage* typu *ITcVnImage*. V průběhu programu v daném cyklu ověřuje, zda nenastal error (pokud ano, provede reset propojení s kamerou). Pokud je z kamery úspěšně přijat nový snímek, je uložen do proměnné typu *ITcVnImage*.

Následně je vypočten střed šablonového obrázku a střed aktuálního snímku z kamery. Po převedení obrazu do monochromatického spektra je na snímek aplikována funkce *F_VN_MatchTemplate* s použitím metody *TMM_CCORR_NORMED*, tedy normalizované korelace. Tato funkce vyhledává výskyt šablony v obraze pořízeném z kamery:

```
hr := F_VN_MatchTemplate(ipImageGray, ipTemplateImage, ipImageRes, hr);
```

Po aplikaci algoritmu je identifikován bod s největší mírou shody funkcí *F_VN_MaxPixelValue*. Ze souřadnic tohoto bodu, které jsou vráceny v proměnné *aPosition*, je následně vypočten offset vůči levému hornímu rohu snímku, a dále i odchylka středu nalezené značky od středu celého obrazu:

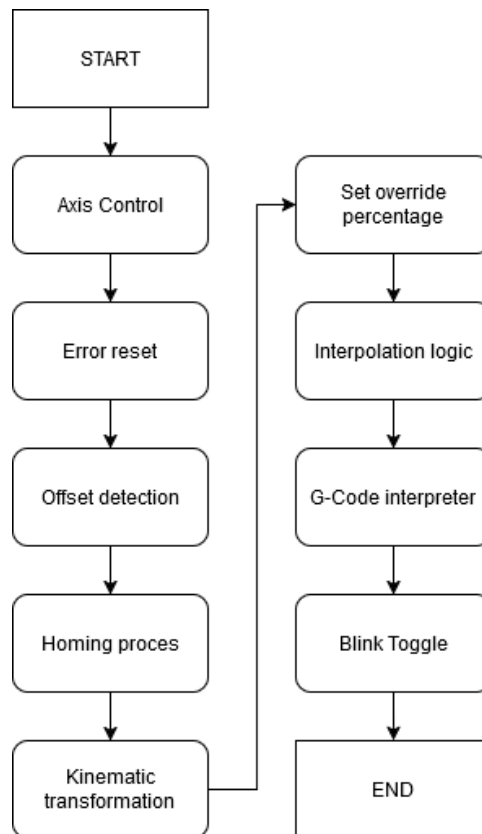
```
gvl.dist_from_center[1] :=
    UDINT_TO_INT(CenterT[0]) - UDINT_TO_INT(Center[0]);
gvl.dist_from_center[0] :=
    UDINT_TO_INT(CenterT[1]) - UDINT_TO_INT(Center[1]);
```

Jakmile se značka přiblíží ke středu kamery, je provedeno rozpoznání jejího konkrétního typu. Na základě tohoto rozpoznání je do globální proměnné přiřazena skutečná poloha koncového bodu vztažená k patě manipulátoru.

Podrobnější struktura programu VISION je popsána v kapitole 2.4 Program zpracovávající obraz. Tato kapitola se detailně zabývá popisem celého programu, který zpracovává obraz.

2.1.4 Hlavní programová smyčka MAIN

V hlavním programu jsou volány jednotlivé programy a funkce dle následujícího diagramu:



Obrázek 2.3 Struktura hlavní programové smyčky MAIN

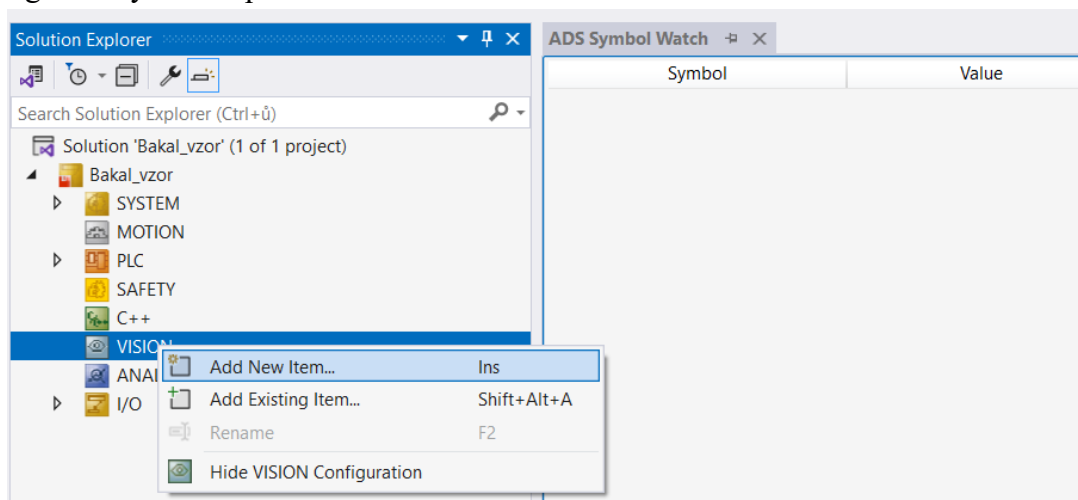
2.2 Realizace modulu v TwinCAT3

V prostředí TwinCAT je nejprve nutné založit nový projekt a následně vytvořit instanci řídicí jednotky (PLC). Poté je třeba nakonfigurovat VISION modul podle typu použitého zdroje obrazových dat. Po úspěšné konfiguraci modulu je přístupeno k implementaci řídicího programu, který zajišťuje zpracování obrazu.

Pro využití funkcionality zpracování obrazu musí být do projektu přidána a vložena knihovna *Tc3_Vision*, jež poskytuje potřebné bloky pro práci s obrazovými daty, jako jsou například detekce objektů, měření vzdáleností a vyhodnocování kontur.

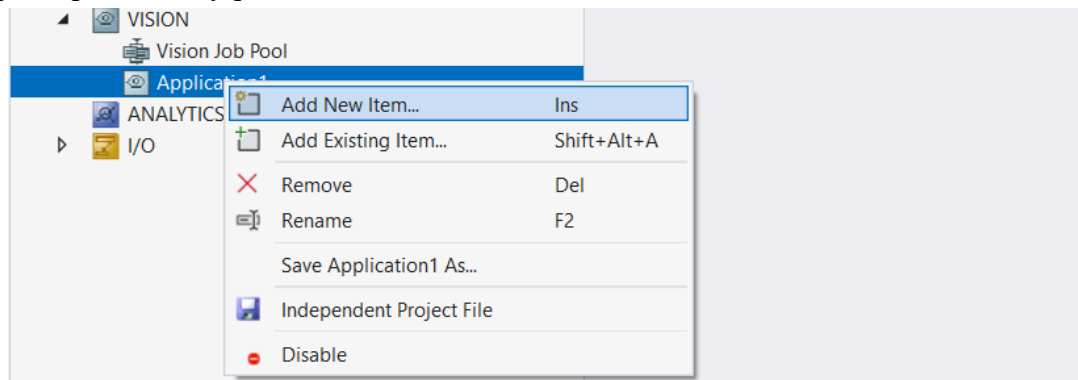
2.2.1 Konfigurace VISION modulu

V prostředí Solution Explorer je nutné kliknout pravým tlačítkem myši na složku *VISION* a z kontextového menu zvolit možnost **Add New Item...**. Tímto způsobem je vytvořen nový prvek, který bude následně použit pro definici vizuálních funkcí v rámci konfigurace systému zpracování obrazu.



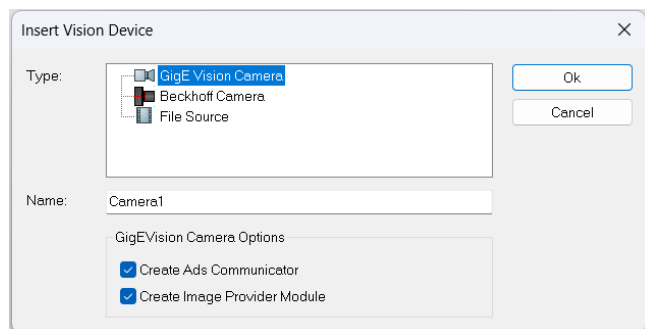
Obrázek 2.4 Vytvoření vision modulu

Následně je opět nutné kliknout pravým tlačítkem myši na položku *Application* a z dostupných možností zvolit **Add New Item...**. Tímto způsobem je do aplikační vrstvy projektu přidán nový prvek.



Obrázek 2.5 Přidání zdroje snímků

Po výběru položky **Add New Item...** se zobrazí dialogové okno, ve kterém je možné zvolit jednu ze tří dostupných variant pro zdroj obrazových dat. V rámci této práce byly dále využity a podrobněji analyzovány varianty GigE Vision Camera a File Source, které byly použity při praktické realizaci projektu.

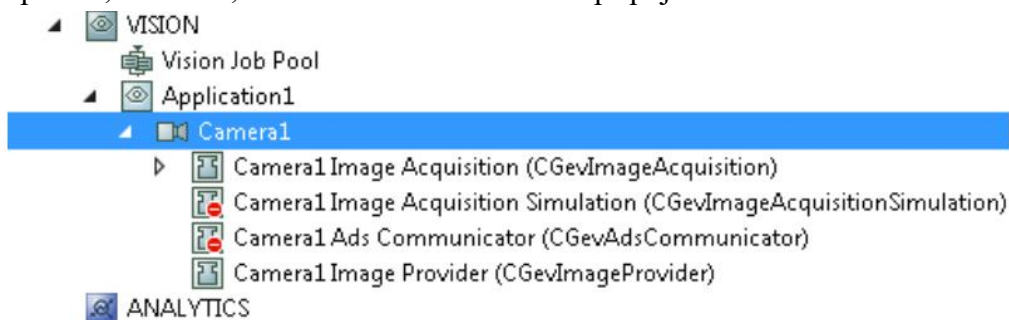


Obrázek 2.6 Výběr typu zdroje snímků

2.2.2 GigE Vision Camera

Pro zvolení možnosti GigE Vision Camera je nejprve nutné fyzicky připojit kameru k průmyslovému počítači (IPC) a následně v projektu nakonfigurovat její připojení prostřednictvím odpovídající IP adresy. Po úspěšném propojení by měla být kamera automaticky detekována a zobrazena v seznamu zařízení.

Dvojitým kliknutím na název kamery je otevřeno konfigurační okno, které obsahuje několik záložek umožňujících detailní nastavení parametrů kamery, jako jsou např. expozice, rozlišení, korekci zkreslení či síťové připojení.



Obrázek 2.7 Kontrola správné konfigurace kamery

V záložce General lze ověřit, zda bylo připojení kamery úspěšně navázáno a zda je zařízení ve stavu online. V této části by měly být rovněž zobrazeny základní identifikační údaje kamery, jako je IP adresa a MAC adresa.

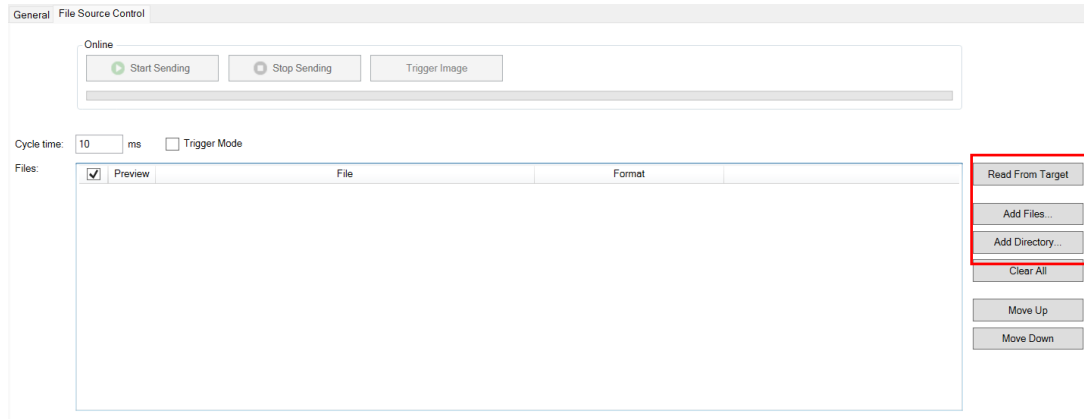
V záložce Configuration Assistant je možné zobrazit aktuální výstup z kamery, což slouží k rychlé vizuální kontrole správné funkce zařízení a ověření nastavení obrazového přenosu.

2.2.3 File Source

Po zvolení možnosti *File Source* je třeba specifikovat konkrétní soubory, případně adresář, ze kterého budou obrazová data načítána. Tato konfigurace se provádí v záložce *File Source Control*.

Pomocí tlačítek *Add Files* nebo *Add Directory* lze buď načíst jednotlivé obrazové snímky, nebo zvolit celý adresář, jehož obsah bude použit jako vstupní datová sada. Po

výběru požadovaných souborů je třeba stisknout tlačítko *Read From Target*, čímž jsou snímky nahrány do prostředí TwinCAT a připraveny k dalšímu zpracování.



Obrázek 2.8 Načtení snímků při variantě File Source

2.3 Práce s načtenými snímky v programu

Varianty File Source i GigE Vision Camera se v prostředí programu chovají shodně. Kamera, případně zdroj snímků, je v projektu reprezentován pomocí funkčního bloku *FB_VN_SimpleCameraControl*, který poskytuje základní funkcionalitu pro práci s obrazovými daty – ať již pocházejí z reálné kamery, nebo ze souborového vstupu.

Jednotlivé funkce tohoto bloku jsou implementovány ve formě metod, což znamená, že samotný blok nelze volat přímo jako proceduru. Místo toho je nutné využívat konkrétní metody, které slouží k řízení snímání, načítání snímků, ovládání přenosu a dalším operacím souvisejícím se zpracováním obrazu.

Metody ve *FB_VN_SimpleCameraControl*:

- *GetCurrentImage* – vrací aktuálně dostupný snímek (je-li k dispozici)
- *GetState* – vrací aktuální stav vnitřního stavového automatu kamery (Stavový automat je detailně popsán viz. Příloha C -Stavový automat kamery [17])
- *Reset* – uvede kameru do výchozího (inicializačního) stavu
- *StartAcquisition* – spustí akvizici snímků
- *StopAcquisition* – ukončí akvizici snímků
- *TriggerImage* – vyvolá pořízení dalšího snímku (softwarový trigger)

Pro získání snímku se využívá metoda *GetCurrentImage*, která uloží aktuální snímek do instance třídy typu *ITcVnImage*. V následující ukázce je tato instance pojmenována jako *ipImageIn*:

```
hr := fbCamera.GetCurrentImage(ipImageIn);
```

Každá metoda v knihovně *Tc3_Vision*, včetně metod funkčního bloku *FB_VN_SimpleCameraControl*, vrací návratovou hodnotu typu *HRESULT*. Ta je reprezentována jako osmimístné hexadecimální číslo a slouží především pro detekci chyb a monitorování stavu programu.

Třída *ITcVnImage* slouží k reprezentaci obrazových dat a umožňuje jejich další zpracování. Obsahuje rovněž metody pro získání informací o parametrech snímku, jako jsou jeho rozměry, formát pixelů a další vlastnosti.

Pro práci s obrázky je k dispozici velké množství funkcí. Zde jsou vybrané některé z nich:

- *F_VN_GetImageInfo* – vrátí strukturu *TcTvImageInfo*, ve které jsou uloženy informace o obrázku
- *F_VN_TransformIntoDisplayableImage* – z předaného obrázku vytvoří obrázek, který lze zobrazit v TwinCAT interface.
- *F_VN_CreateImage* – vytvoří kopii obrázku
- *F_VN_MatchTemplate* – aplikuje na obraz algoritmus hledání shody s template obrázkem

2.4 Program zpracovávající obraz

Nejprve jsou načteny template obrázky z uložení a je ošetřeno, že program se provede pouze, pokud je kamera připravená. Pokud ano, načte obraz a uloží ho do *ipImageIn* (třída typu *ITcVnImage*). Proměnné *eState* je přiřazen výstup z metody *GetState()*, použité na *fbCamera*, která je definovaná, jako *FB_VN_SimpleCameraControl*.

```
IF eState = TCVN_CS_ERROR THEN
    hr := fbCamera.Reset();
ELSIF eState < TCVN_CS_ACQUIRING THEN
    hr := fbCamera.StartAcquisition();
ELSIF eState = TCVN_CS_ACQUIRING THEN
    hr := fbCamera.GetCurrentImage(ipImageIn);
```

Dále je zjištěn střed template snímku a střed obrazu z kamery. Stačí zjistit střed pouze jednoho template vzoru, protože jejich jediná odlišnost je, že jeden je inverzí druhého. Jeho XY souřadnice jsou uloženy do polí o dvou prvcích. K zjištění parametrů je použita funkce *F_VN_GetImageInfo*.

```
hr:=F_VN_GetImageInfo(ipImage      := ipImageIn,
                     stImageInfo:= stUserImageInfo,
                     hrPrev      := hr);

Center[0] := stUserImageInfo.nHeight / 2;
Center[1] := stUserImageInfo.nWidth / 2;

hr:=F_VN_GetImageInfo(ipImage      := ipTemplateImage,
                     stImageInfo:= stUserImageInfoT,
                     hrPrev      := hr);

TOffset[0] := stUserImageInfoT.nHeight / 2;
TOffset[1] := stUserImageInfoT.nWidth / 2;
```

Jako další krok je snímek převeden do monochromatického spektra a je zavolána funkce hledání template obrázku. U funkce hledání shody implicitně zvolena metoda *TMM_CCORR_NORMED*.

```
hr := F_VN_ConvertColorSpace( ipSrcImage := ipImageIn,
                             ipDestImage := ipImageGray,
                             eTransform := TCVN_CST_RGBA_TO_GRAY,
                             hrPrev      := hr);
```

```
hr := F_VN_MatchTemplate(ipImageGray, ipTemplateImage, ipImageRes, hr);
```

Poté je použita funkce *F_VN_MaxPixelValue*, která vrátí jak maximální hodnotu shody, tak lokaci této shody.

```
hr := F_VN_MaxPixelValue(ipImageRes, aMaxValue, aPosition, hr);
```

Následně je vyhodnoceno, zda byl nalezen bod s dostatečnou shodou. Pokud se tak nestalo, je provedena rotace hledaného vzoru o 45° a o -45°. Jestliže stále nebyla nalezena shoda, hledání se provede znovu s invertovaným vzorem. V případě, že stále nebyla nalezena dostatečná shoda, je homing zastaven.

Po nalezení dostatečné shody je vypočten offset středu obrázku od středu značky.

```
CenterT[0] := aPosition[0] + TOffset[0];
CenterT[1] := aPosition[1] + TOffset[1];
```

```
gvl.dist_from_center[1] :=
    UDINT_TO_INT(CenterT[0]) - UDINT_TO_INT(Center[0]);
```

```
gvl.dist_from_center[0] :=
    UDINT_TO_INT(CenterT[1]) - UDINT_TO_INT(Center[1]);
```

Pokud je vzdálenost mezi středem obrazu a středem referenční značky v obou osách menší než 20 pixelů, provede se výřez oblasti rozpoznané značky z obrazu získaného z kamery. Současně jsou z uložení načteny všechny varianty referenčních značek.

Výřez je realizován pomocí funkce:

```
hr := F_VN_CopyImageRegion(ipImageGray, aPosition[0], aPosition[1],
stUserImageInfoT.nWidth, stUserImageInfoT.nHeight, ipCut, hr);
```

Po vytvoření výřezu následuje jeho porovnání se všemi variantami referenčních značek. Index značky, která vykazuje nejvyšší míru shody, je uložen do proměnné *nBestMatchIdx*.

Na základě hodnoty tohoto indexu je následně pomocí konstrukce CASE do globální proměnné *gvl.Location* zapsána skutečná pozice koncového bodu manipulátoru v souřadnicích x,y, které jsou vztažené k patě manipulátoru.

2.4.1 F_VN_MatchTemplate

Funkce *F_VN_MatchTemplate* provede porovnání template obrázku s každou pozicí na zkoumaném obrázku a uloží výsledek porovnání. Implicitně tato funkce pracuje s porovnávací metodou *TMM_CCORR_NORMED*, tedy s metodou normalizované korelace, která vrací hodnotu v rozmezí 0 až 1. Čím vyšší je hodnota na výstupu, tím vyšší je míra shody se vzorem.

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') * I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 * \sum_{x', y'} I(x + x', y + y')^2}} \quad 2.1$$

3. INTEGRACE PYTHON SKRIPTŮ PRO ZPRACOVÁNÍ OBRAZU A KOMUNIKACI

Při návrhu a realizaci modulu v pythonu bylo použito knihovny *OpenCV* zabývající-se zpracováním obrazu, *pymodbus* pro komunikaci s prostředím TwinCAT pomocí modbus protokolu a *pypylon* pro komunikaci s kamerou Basler.

Program se skládá z hlavní programové smyčky *main* a funkcí vykonávajících příslušné operace a činnosti.

V části, která je stále vykonávána v prostředí TwinCAT tedy nebudeme potřebovat funkční blok `Offset_visu`.

3.1 Main

Hlavní programový blok, ze kterého jsou volány jednotlivé funkce a podprogramy. Po inicializaci a splnění podmínky pro aktivní homing od PLC se zde spouští hlavní programová smyčka, v níž jsou volány funkce načítající a zpracovávající obraz a funkce obsluhující jak zpětnovazební komunikaci odchylky hlavy od značky, tak i komunikaci výsledné zjištěné polohy.

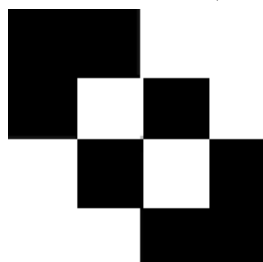
3.2 readImageBasler

Funkce zajišťuje komunikaci s kamerou, konkrétně navázání spojení, načtení a uložení aktuálního snímku.

Nejprve je vytvořena instance kamery, prostřednictvím které je ke kameře přistupováno. Následně je ověřeno, zda bylo spojení úspěšně navázáno. Poté je pomocí metody *GrabOne* z knihovny *pypylon* získán snímek a převeden do šedotónového formátu.

3.3 findMark

Jedná se o jednu z nejdůležitějších funkcí systému. Nejprve je hledán jednoduchý vzorový obrázek bez identifikační značky, který slouží obecnému určení pozice libovolné referenční značky na načteném snímku (Obrázek 3.1).



Obrázek 3.1 Značka hledaná v první iteraci funkce `findMark`

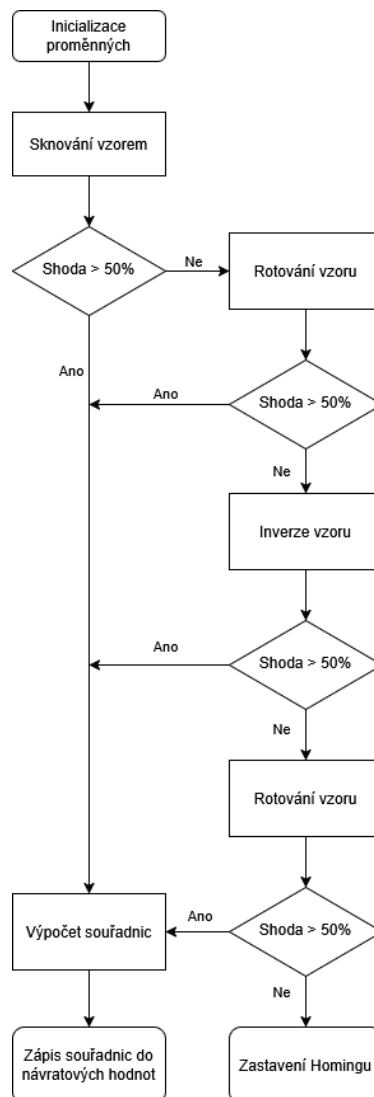
Tato značka se reálně nenachází nikde na pracovní ploše, ale její základní struktura je obsažena ve všech variantách reálných značek, které se na pracovní ploše vyskytují.

Následně je pomocí knihovny funkce *matchTemplate* z knihovny *OpenCV* prohledán celý obraz. Tato funkce vrací matici shod vzoru s jednotlivými oblastmi načteného snímku.

Pokud není nalezena dostatečná shoda, je zavolána funkce *HledaniRotaci*, která provede hledání s pootočeným referenčním obrázkem.

Pokud není ani v tomto případě dosaženo požadované míry shody (stanovené na 50 %), provede se stejná procedura ještě jednou s invertovanou značkou. Jestliže opět nebude nalezena shoda větší, než 50%, dojde k nastavení proměnné povolující homing na hodnotu *false* a tím k zastavení homingu.

V případě úspěšného nalezení shody jsou do návratových hodnot uloženy souřadnice nalezené pozice a odchylka od středu snímku.



Obrázek 3.2 Vývojový diagram funkce findMark

3.3.1 Funkce HledaniRotaci

Pomocí afinní transformace je značka nejprve otočena o 45° a celý snímek je znovu prohledán funkcí *matchTemplate*. Pokud není nalezena dostatečná shoda, je značka otočena o -45° vzhledem k původní poloze a vyhledání se opakuje. V případě, že ani jedno pootočení nedosáhne shody větší než 50%, je zpět do funkce *findMark* vrácena hodnota maximální nalezené shody.

Afinní transformace jsou realizovány pomocí transformačních matic, které vycházejí z rovnic:

$$x' = a_0 + a_1 * x + a_2 * y \quad (3.1)$$

$$y' = b_0 + b_1 * x + b_2 * y \quad (3.2)$$

Parametry a_0 a b_0 určují posun obrazu vůči výchozímu bodu se souřadnicemi (0, 0), tedy v levém horním rohu. Parametry a_1 , a_2 a b_1 , b_2 ovlivňují rotaci a případné zkosení obrazu kolem bodu na souřadnicích (0, 0). Tyto parametry se obvykle předávají do transformačních funkcí ve formě matice o rozměrech 2x3.

$$M = \begin{bmatrix} a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 \end{bmatrix} \quad (3.3)$$

Ačkoli pro čistou rotaci obrazu o definovaný úhel postačuje matice 2x2, knihovna *OpenCV* při rotaci umožňuje také změnu měřítka, což vyžaduje rozšíření transformační matice [18]. Modifikovaná rotační matice pak nabývá tvaru:

$$M = \begin{bmatrix} \alpha & \beta & (1 - \alpha) * C_x - \beta * C_y \\ -\beta & \alpha & \beta * C_x + (1 - \alpha) * C_y \end{bmatrix} \quad (3.4)$$

Kde:

- C_x je střed v ose x
- C_y je střed v ose y
- $\alpha = \text{měřítko} * \cos \theta$
- $\beta = \text{měřítko} * \sin \theta$

Takto sestavená transformační matice je předána jako argument funkci *wrapAffine* z knihovny *OpenCV*, která ji použije pro transformaci rotovaného obrázku podle matice M a specifikovaných rozměrů výstupního obrazu.

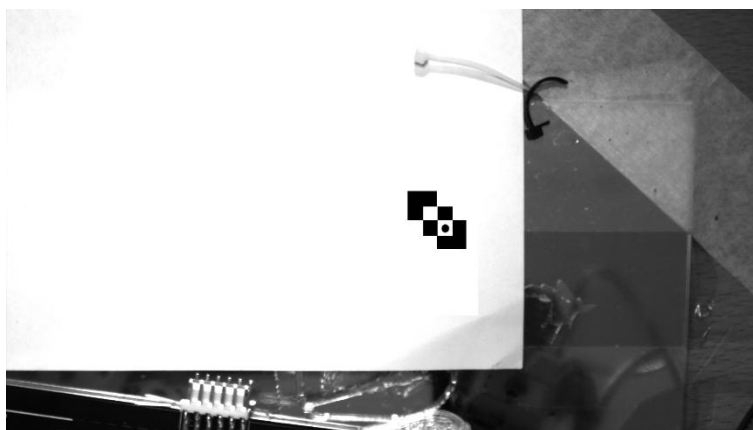
3.3.2 matchTemplate

Jedná se o funkci z knihovny *OpenCV*. Mimo vstupních parametrů zkoumaného snímku a hledaného vzorového obrázku je třeba určit, jakou metodou bude snímek zkoumán. Pro tuto aplikaci byla zvolena metoda normalizované korelace *TM_CCOEFF_NORMED*. Je třeba také zajistit, aby zkoumaný snímek měl stejný datový typ, jako vzorový obrázek.

Metoda funguje na základě skalárního součinu příslušných pixelů vzorového obrázku a aktuálního výřezu zkoumaného snímku a následného normování, jak definuje *OpenCV* [19].

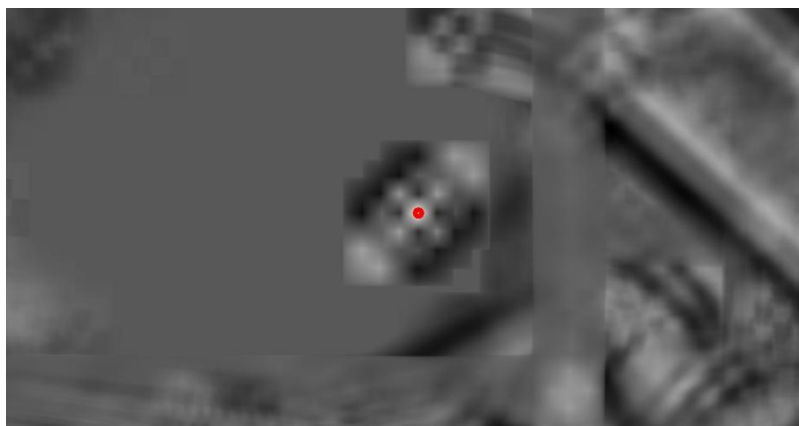
$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') * I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 * \sum_{x', y'} I'(x + x', y + y')^2}} \quad (3.5)$$

Jako výřez se volí okno o stejných rozměrech, jako má vzorový obrázek. Z tohoto důvodu musí být značka definovaná tak, aby její velikost na pořízeném snímku odpovídala rozměrům vzorových snímků uložených v paměti.



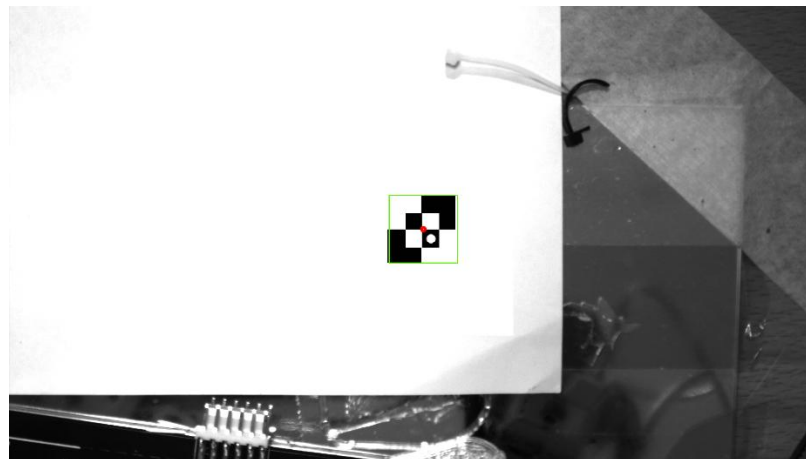
Obrázek 3.3 Načtený obraz z kamery

Po analýze obrazu funkcí *matchTemplate* vznikne mapa, jejíž rozměry jsou zmenšeny o šířku a výšku vzorového obrázku. Výsledkem je tedy takzvaná mapa shod. (viz. Obrázek 3.4). Jednotlivé prvky této dvourozměrné matice nabývají hodnot od -1 pro nejnižší shodu (toto odpovídá inverzi značky) až po 1, který implikuje maximální shodu. Růžovou tečkou je zvýrazněno místo s největší shodou.



Obrázek 3.4 Mapa shod s obecnou referenční značkou

Pro ověření správnosti byl do původního obrazu zakreslen zelený čtverec odpovídající poloze nalezené značky. Růžová tečka poté značí střed značky, tedy bod, vůči kterému se robot referuje.



Obrázek 3.5 Výsledný obraz po nalezení maximální shody

Ukázkový výstup z konzole v pythonu pro tento demonstrativní případ:

```
Maximální shoda: 94.8397%
Minimální shoda: -46.2576%
Střed obrazu: x: 1039 y: 585
Offset značky od středu obrazu: x: 143 y: 4
Souřadnice pro posun: [143.5, 4.0]
Identifikace
Šablona 1: podobnost = 1.0000
Šablona 2: podobnost = 0.9086
Šablona 3: podobnost = 0.9141
Šablona 4: podobnost = 0.9087
Šablona 5: podobnost = 0.9265
Šablona 6: podobnost = 0.9212
Šablona 7: podobnost = 0.9148
Šablona 8: podobnost = 0.9569
Šablona 9: podobnost = -0.9672
Šablona 10: podobnost = -0.8670
Šablona 11: podobnost = -0.8844
Šablona 12: podobnost = -0.8671
Šablona 13: podobnost = -0.8966
Šablona 14: podobnost = -0.8793
Šablona 15: podobnost = -0.8847
Šablona 16: podobnost = -0.9256
Šablona 17: podobnost = 0.9345
Šablona 18: podobnost = 0.9345
Šablona 19: podobnost = 0.9345
Šablona 20: podobnost = 0.9345
Šablona 21: podobnost = 0.9337
Šablona 22: podobnost = 0.9337
Šablona 23: podobnost = 0.9337
Šablona 24: podobnost = 0.9337
Šablona 25: podobnost = -0.9345
Šablona 26: podobnost = -0.9345
Šablona 27: podobnost = -0.9345
Šablona 28: podobnost = -0.9345
```

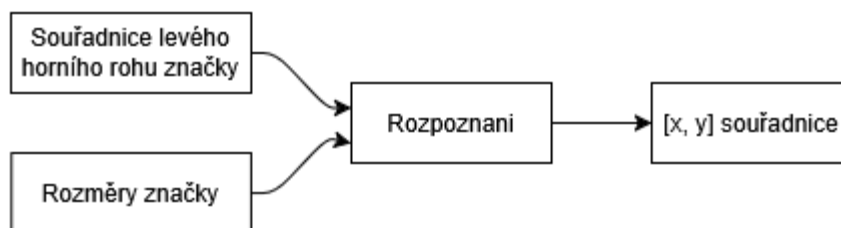
```
Šablona 29: podobnost = -0.9337
Šablona 30: podobnost = -0.9337
Šablona 31: podobnost = -0.9337
Šablona 32: podobnost = -0.9336
Nejlepší shoda: qr_1.png s podobností 100.00%
Byla zjištěna pozice: x: 1 y: 1
```

3.4 Funkce Rozpoznání

Druhá z klíčových funkcí sloužící k rozpoznání referenční značky umístěné na pracovní ploše a určení její pozice. Nejprve jsou načteny všechny varianty referenčních značek, které se mohou na snímané ploše vyskytovat. Jedná se o 32 předem připravených šablon ve formátu *.png*, které jsou uloženy v pracovním adresáři aplikace (viz Příloha D - Seznam použitých referenčních značek). Každá z těchto značek reprezentuje definovanou pozici na pracovní desce.

Následuje samotné rozpoznávání, které je realizováno metodou šablonového porovnávání pomocí funkce *cv.matchTemplate* z knihovny *OpenCV* (viz kapitola 3.3.2). Tato funkce vypočítává míru podobnosti mezi výřezem a každou z načtených šablon. Při každé iteraci se zaznamenává nejvyšší dosažená hodnota podobnosti (korelační koeficient) a identifikátor odpovídající šablony. Pokud žádná ze šablon nesplňuje minimální kritéria shody, funkce vrací výchozí hodnotu $[0, 0]$, což signalizuje neúspěšné rozpoznání.

V případě úspěšného nalezení shody je na základě identifikátoru nejlepší šablony přiřazena odpovídající logická pozice ve formátu $[x, y]$, která označuje konkrétní pozici značky na pracovní ploše. Tato informace je zásadní pro další navazující algoritmy systému. S těmito hodnotami přímo pracuje už jen funkce *LocComm*.



Obrázek 3.6 Diagram funkce Rozpoznání

3.5 DoHomingTest

Tato funkce slouží k navázání komunikace mezi Python skriptem a TwinCAT pomocí protokolu Modbus TCP. Jejím primárním úkolem je ověřit, zda je povolen režim *homing* (referencování) robotického systému, a to prostřednictvím čtení binárního registru na definované adrese.

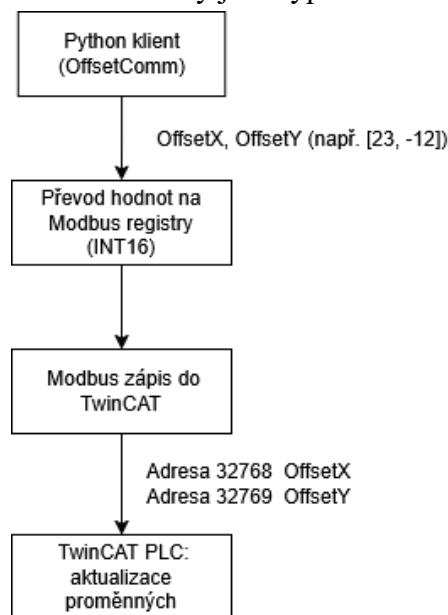
Funkce je definována jako asynchronní operace, což umožňuje efektivnější komunikaci s Modbus serverem. Po úspěšném navázání spojení s Modbus serverem je provedeno čtení stavu cívky (coil) na adrese 32768.

Jedná se tedy o klíčovou funkci pro synchronizaci hlavní programové smyčky main v python skriptu se stavem systému řízeného prostřednictvím TwinCAT. Díky ní je možné zajistit, že referencování bude spuštěno pouze tehdy, pokud to řídicí logika PLC výslovně povolí. To zvyšuje bezpečnost a robustnost celého řešení.

3.6 OffsetComm

Tato funkce slouží k navázání komunikace mezi Python skriptem a TwinCAT pomocí protokolu Modbus TCP stejně jako funkce *DoHomingTest*. Jejím primárním úkolem je zpětnovazební komunikace posunutí středu kamery od středu obrazu, a to prostřednictvím zápisu hodnot na holding registry na definované adrese.

Stejně jako u předchozí funkce se jedná o asynchronní operaci, která po úspěšném navázání spojení s Modbus serverem zapisuje hodnotu offsetu v ose x na adresu 32768 a offsetu osy z na adresu 32769. Obě hodnoty jsou typu INT16.



Obrázek 3.7 Diagram funkce OffsetComm

3.7 LocComm

Tato funkce stejně jako funkce OffsetComm slouží k asynchronní komunikaci mezi Python skriptem a TwinCAT skrze protokol Modbus TCP. Úkolem této funkce je na konci referování zapsat na holding registry skutečnou pozici koncového bodu manipulátoru.

Pozici v ose x zapíše na adresu 32770 a pozici v ose y na adresu 32771. Obě hodnoty jsou typu FLOAT32.

3.8 Konfigurace ModBus komunikace s prostředím TwinCAT

Nejprve je nutné mít nainstalovaný doplněk Modbus TCP z TwinCAT Package Manageru. Po spuštění nástroje *TcModbusSrvCfg.exe* je do konfiguračního okna třeba zadat IP adresu serveru (v tomto projektu byla použita lokální adresa *127.0.0.1*) a port (standartně *502*). Po stisku tlačítka *Import Configuration* je načten konfigurační soubor *Modbus_server_conf.xml*. Po potvrzení tlačítkem *Set configuration* dojde k přemapování výchozího rozsahu komunikačních adres na proměnné PLC. V PLC programu jsou tyto proměnné přístupné jako globální proměnné.

```
mb_Input_Coils      : ARRAY [0..255] OF BOOL;
mb_Output_Coils    : ARRAY [0..255] OF BOOL;
mb_Input_Registers : ARRAY [0..255] OF INT;
mb_Output_Registers : ARRAY [0..255] OF LREAL;
```

Při komunikaci z externí aplikace (z python programu) je nutné znát skutečné Modbus adresy, na které jsou jednotlivé registry mapovány. K tomu účelu byla využita referenční tabulka dostupná na webu Beckhoff Infosys (viz. Obrázek 3.8)

Modbus areas	Modbus address	ADS area	
Digital inputs	0x8000 - 0x80FF	Name of the variables in the PLC program	Data type
		GVL.mb_Input_Coils	ARRAY [0..255] OF BOOL
Digital outputs (coils)	0x8000 - 0x80FF	Name of the variables in the PLC program	Data type
		GVL.mb_Output_Coils	ARRAY [0..255] OF BOOL
Input registers	0x8000 - 0x80FF	Name of the variables in the PLC program	Data type
		GVL.mb_Input_Registers	ARRAY [0..255] OF WORD
Output registers	0x3000 - 0x5FFF	0x4020 - PLC memory area	0x0
	0x6000 - 0x7FFF	0x4040 - PLC data area	0x0
	0x8000 - 0x80FF	Name of the variables in the PLC program	Data type
		GVL.mb_Output_Registers	ARRAY [0..255] OF WORD

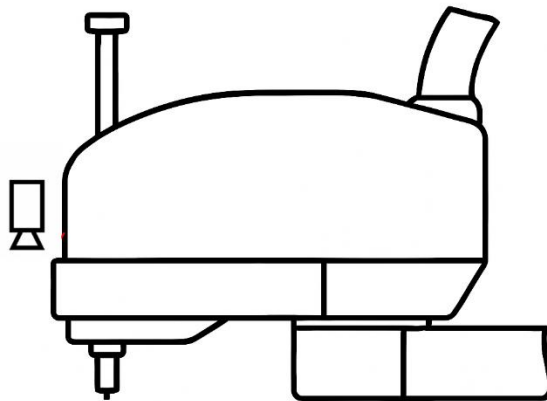
Obrázek 3.8 Tabulka s adresami modbus registrů [20]

V části realizované v pythonu komunikují přes modbus tři funkce.

- *DoHomingTest* – tato funkce po zavolání přečte binární hodnotu z coil registru na adrese 32768, což odpovídá nultému registru s hexadecimální adresou 0x8000 (viz. Obrázek 3.8).
- *OffsetComm* – funkce zapisuje hodnoty proměnných typu INT16 do holding registrů na adresách 32768 a 32769. Tyto adresy odpovídají registrům 0 a 1.
- *LocCom* – funkce se chová stejně jako funkce *OffsetComm*, pouze pracuje s adresami 32770 a 32771 a datovým typem FLOAT32.

4. NÁVRH HARDWAROVÉHO ŘEŠENÍ

Kamera je umístěna na navrženém držáku, který byl instalován do pozice zajišťující dostatečné zorné pole v přední části posledního horizontálního ramene. (tj. rameno mezi 2. a 3. osou). Návrhu tohoto držáku se dále věnuje kapitola 4.1.1. Toto uložení je schematicky znázorněno viz. Obrázek 4.1. Pracovní vzdálenost kamery od snímané plochy tedy činí 54,5 cm.



Obrázek 4.1 Uložení kamery na manipulátoru [21]

Jako další bylo třeba rozmístit po ploše pod robotem referenční značky. Proto bylo třeba nejprve vypočítat skutečnou velikost horizontálního a vertikálního zorného pole. K tomuto bylo využito již známých rovnic

$$hFoV = \frac{h \cdot WD}{f} = \frac{4,22 \cdot 545}{8} \approx 287,5 \text{ mm} , \quad (4.1)$$

Správnost výpočtu byla ověřena také experimentálním měřením pomocí svinovacího metru. Při měření byla zjištěna vzdálenost přibližně 29 cm, což odpovídá teoretickému výpočtu. S touto hodnotou se dále pracuje v kapitole 4.3, která se přímo zabývá rozmístěním značek.

4.1 Umístění kamery

V rámci projektu bylo třeba navrhnout vhodný držák kamery, který by zajišťoval optimální polohu a zorný úhel. Byla zvolena pozice využívající dva otvory se závitem M4, které se již nacházejí na posledním ramenu manipulátoru viz. Obrázek 4.1 a Obrázek 4.2.



Obrázek 4.2 Díry pro připevnění držáku kamery

Kamera je umístěna ve svém chladiči, který je k tomuto držáku připevněn.

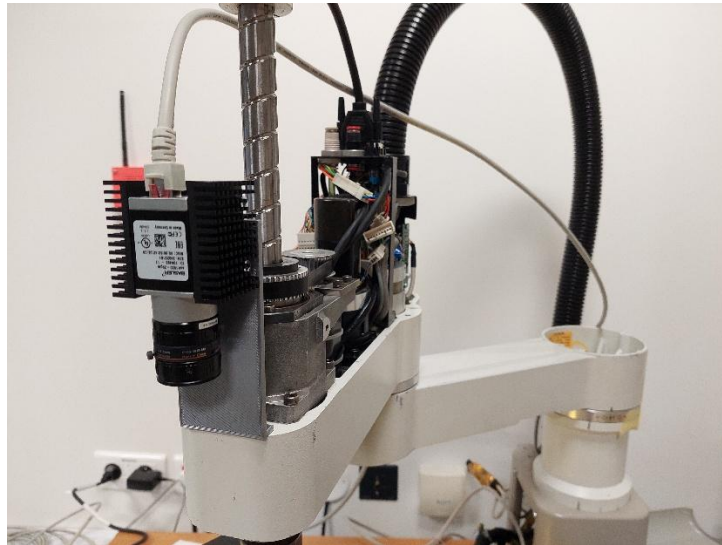


Obrázek 4.3 Držák s chladičem pro kamery Sugar Cube [22]

Kamera záměrně nebyla umístěna na koncový (pohyblivý) člen manipulátoru. Pakliže by kamera byla umístěna na koncový bod, operováním v z-ové ose by docházelo k otáčená kamery, což by přineslo několik problémů:

1. Přívodní kabel kamery by se obmotával kolem osy a mohlo by dojít k jeho poškození. Dále by zde mohl nastat problém se ztrátou propojení kvůli pohybu konektoru RJ45, u kterého bylo již při testech s kamerou zjištěno, že je citlivý na výpadky komunikace při manipulaci s kabelem.
2. Při otáčení obrazu kamery by docházelo k chybovým údajům při výpočtu odchylky od středu obrazu. Zároveň by mohl nastat stav, kdy by se kamera octla na opačné straně koncového členu, přičemž jeho poloha v XY souřadnicích by se nezměnila. V tomto případě by došlo k chybnému zreferování v osách x a y.

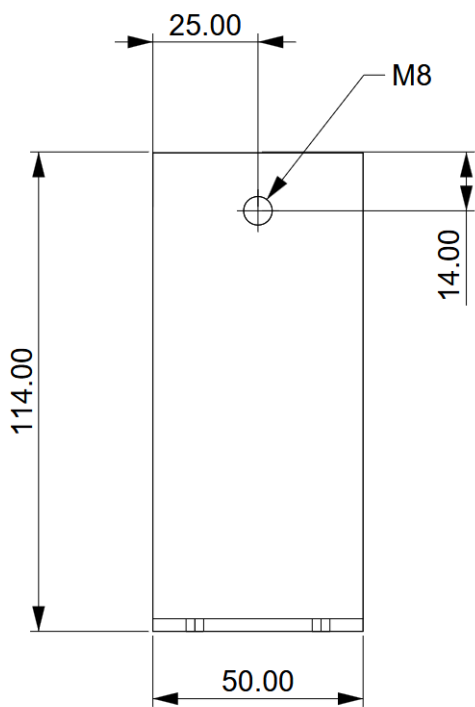
Kabel je následně veden nad krytem manipulátoru a dále je společně s dalšími vodiči veden pod desku, na které manipulátoru operuje. Zde je kabel přiveden do switchu připojeného k IPC, který poveluje manipulátor.



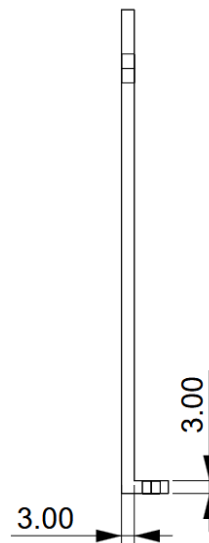
Obrázek 4.4 Skutečné umístění kamery

4.1.1 Návrh držáku kamery

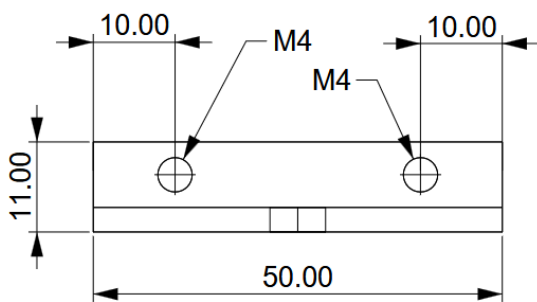
Držák byl navržen tak, aby jej bylo možné připevnit do otvorů se závitem M4, které se nacházejí v horní části konce posledního ramene. Kamera se k držáku upevňuje pomocí otvoru se závitem M8 umístěné na spodní straně chladiče kamery (viz Příloha B - Montážní výkresy chladiče kamery) [22].



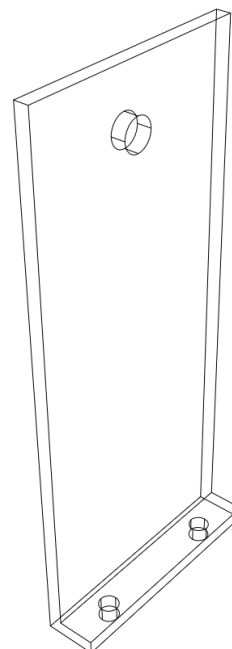
Obrázek 4.5 Držák na kameru, pohled zepředu



Obrázek 4.6 Držák kamery, pohled z boku



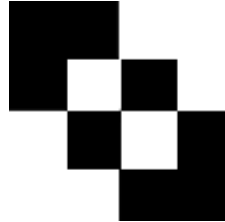
Obrázek 4.7 Držák kamery, pohled shora



Obrázek 4.8 Držák kamery, perspektiva

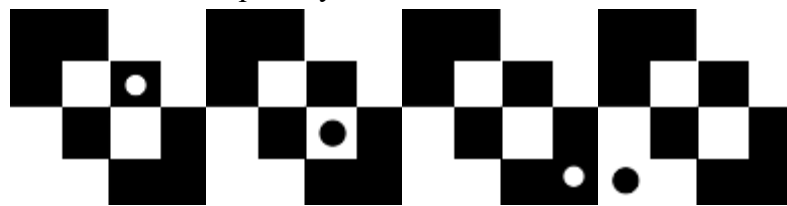
4.2 Referenční značky

Jako referenční značky bylo třeba zvolit nějakou jednoduchou variantu, aby byla značka snadno rozpoznatelná. Proto byla zvolena varianta jednoduché černobílé značky vytvořené kombinací dvou 2x2 navzájem invertovaných šachovnic viz. Obrázek 4.9. Tento vzor by měl zajistit spolehlivé odlišení značky od okolí.



Obrázek 4.9 Základ referenční značky

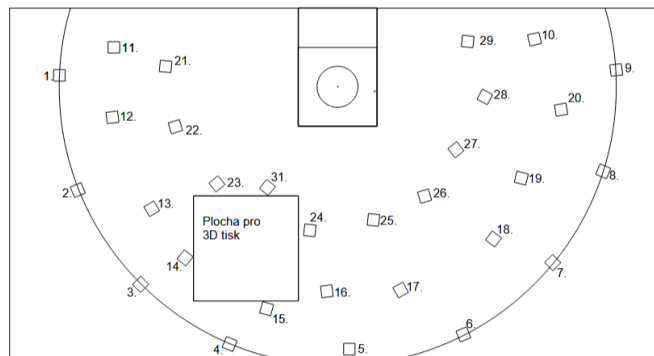
Jako identifikační značka, která určí, o který konkrétní bod pracovní plochy se jedná, byl zvolen kruh, který se nachází na jednobarevné části značky a má invertovanou barvu od svého okolí. Na obrázku níže je vyobrazeno pár příkladů těchto značek. Pro dosažení co největšího počtu značek bylo použito i invertovaných variant. Kompletní seznam značek viz. Příloha D -Seznam použitých referenčních značek.



Obrázek 4.10 Příklady referenčních značek

4.3 Rozmístění referenčních značek

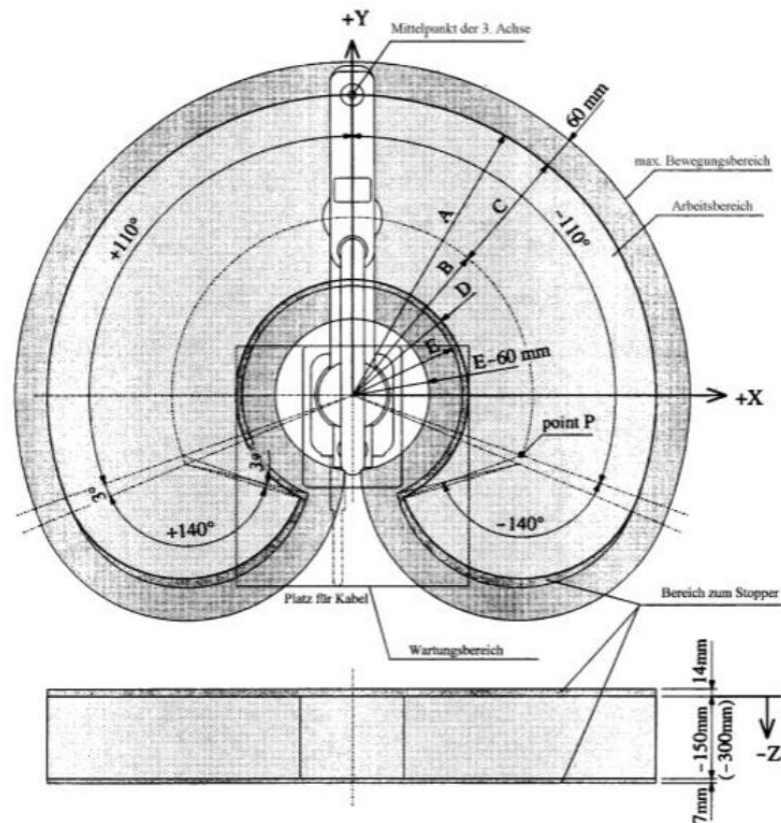
Pracovní plochu manipulátoru Epson H55BN tvoří kružnice o poloměru 550 mm (viz. Obrázek 4.12). Koncový bod, na kterém je připevněn držák s kamerou, opisuje kružnici o poloměru 610 mm. Připočtením vzdálenosti CMOS senzoru od chladiče (14,5 mm), tloušťky chladiče (10 mm) a tloušťky držáku (3 mm) získáme kruh o poloměru 637,5 mm, ve kterém se pohybuje samotná kamera.



Obrázek 4.11 Rozložení referenčních značek na pracovní ploše

Na této kružnici se nachází nejvzdálenější řada značek. První značka je umístěna ve vzdálenosti odpovídající polovině horizontálního zorného pole (*hFoV*), tedy přibližně 144 mm od okraje pracovní plochy. Jediná značka číslo 5 je mimo tuto kružnici, která v tomto bodě zasahuje těsně za pracovní plochu. Vzhledem k tomu, že rameno manipulátoru by se nemělo nikdy dostat mimo pracovní plochu (viz Obrázek 4.11), není uvažovaná situace, kdy by tento stav nastal. Následující rozestupy mezi značkami na kružnici jsou voleny jako polovina *hFoV* snižená o velikost značky, aby byl zajištěn částečný překryv pozorovaných oblastí.

Značky rozmístěné na kružnicích o menším poloměru jsou rozmístěny tak, aby v každém okamžiku kamera zaznamenala alespoň dvě značky. Jedinou výjimkou je oblast v pravé části plochy, kde se nachází podložka pro 3D tisk, na kterou nelze referenční značku umístit.



Obrázek 4.12 Pracovní plocha manipulátoru Epson H55BN [21]

Tabulka 4.1 Rozměry pracovního prostoru [21]

A	B	C	D	E
550 mm	325 mm	225 mm	211 mm	198 mm

5. IMPLEMENTACE

Po implementaci byl ověřen bezchybný přenos obrazových dat z kamery, detekce referenční značky a výpočet posunu (offsetu), který je prostřednictvím rozhraní Modbus TCP doručen řídicímu systému. Znaménka jednotlivých složek offsetu slouží jako vstupní signály pro stavový automat, jenž v reálném čase řídí pohyb manipulátoru.

V okamžiku, kdy odchylka mezi středem snímaného obrazu a detekovanou značkou klesne pod 20 pixelů, provede se finální identifikace značky a na jejím základě se předá přesná poloha koncového efektoru. Tento režim zajišťuje vysokou přesnost zaměření a stabilní provoz systému při lokalizaci a manipulaci s objekty v pracovním prostoru.

Díky optimálnímu rozmístění referenčních značek je navíc dosaženo maximálního a spolehlivého pokrytí sledované oblasti manipulátoru.

5.1 Testování

V rámci testování byly analyzovány různé scénáře poloh a orientací referenčních značek. Bylo zjištěno, že při natočení značky o více než přibližně 35° dochází ke zvýšení chybovosti detekce. Tento jev se projevuje poklesem hodnoty maximální úrovně shody, jak je popsáno v kapitole 3.3.2 matchTemplate.

Na základě těchto výsledků byl stanoven prahový limit pro rozhodnutí o úspěšné detekci značky, a to na úroveň 50 % shody. Tato hodnota představuje minimální akceptovatelnou hranici pro potvrzení, že byla značka detekována.

Z testů zaměřených na identifikaci konkrétní referenční značky dále vyplývá, že je nutné dosáhnout úrovně shody alespoň 75 %, aby bylo možné s dostatečnou mírou jistoty konstatovat, že se skutečně jedná o správnou značku.

5.2 Testování značek za různých světelných podmínek

Část testování se zaměřila na ověření funkčnosti systému za různých světelných podmínek. Měření byla realizována v laboratorním prostředí, kde byly využity jak stropní zářivky, tak přirozené denní světlo. Testovány byly následující scénáře:

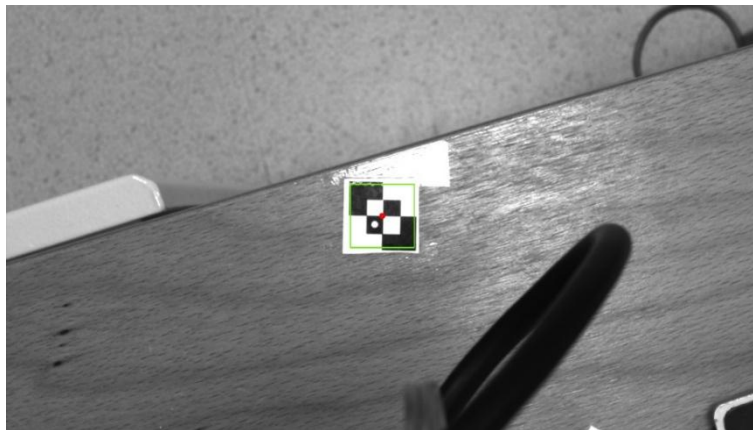
- **Maximální osvětlení** – všechna stropní svítidla byla zapnuta
- **Střední osvětlení** – zapnuté pouze jedno svítidlo v blízkosti stanice s manipulátorem
- **Přirozené osvětlení** – osvětlení pouze denním světlem z oken na protější straně laboratoře
- **Minimální osvětlení** – všechna svítidla vypnutá a zatažené žaluzie

5.2.1 Testování s maximálním osvětlením

Při **maximálním osvětlení** (tj. všechna svítidla zapnuta) bylo provedeno testování rozpoznávání značky č. 3. Ze série deseti pokusů byla značka správně identifikována ve

všech případech, s průměrnou mírou shody 90,52 %.

Následně byla testována značka č. 5, u níž se předpokládala zvýšená pravděpodobnost chyby v důsledku možných odlesků. Také v tomto případě bylo provedeno deset testů, přičemž každý skončil úspěšným rozpoznáním značky (viz. Obrázek 5.1), s průměrnou mírou shody 91,94 %.



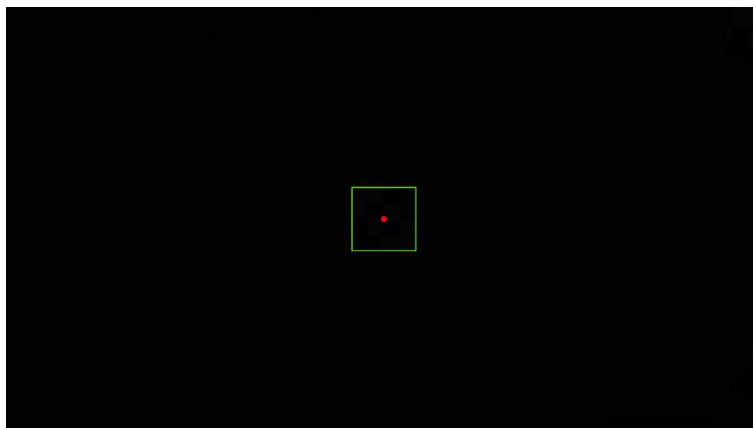
Obrázek 5.1 Výstupní obraz testování značky č. 5 na odlesky

Dále bylo při testování značky č. 11 zjištěno, že pokud se tato značka nachází přímo ve středu zorného pole kamery, dochází zároveň k detekci značky č. 1. Vzhledem ke způsobu zpracování obrazu a prioritnímu řazení značek v programu je v takovém případě jako první vyhodnocena značka č. 1, což vede k nechtěnému přesunu manipulátoru směrem k této značce. Pokud je však značka č. 1 v daném okamžiku zakryta (např. jiným objektem), je rozpoznání značky č. 11 provedeno správně.

Z důvodu potenciálních problémů s identifikací značek v situaci, kdy do zorného pole kamery zasahuje podložka pro 3D tisk, bylo provedeno testování rozpoznávání značky č. 23, která je umístěna v její těsné blízkosti. Během deseti po sobě jdoucích testů byla značka ve všech případech správně identifikována, s průměrnou mírou shody 92,77 %.

5.2.2 Testování s minimálním osvětlením

Při **minimálním osvětlení** (tj. všechna svítidla vypnuta a žaluzie zatažené) bylo testováno rozpoznávání značky č. 12. Ze série deseti pokusů byla značka ve všech případech správně identifikována (Obrázek 5.2), s průměrnou mírou shody 83,80 %.



Obrázek 5.2 Výstupní obraz testování značky č. 12 s minimálním osvětlením

5.2.3 Testování se změnou osvětlení

V závěrečné fázi testování byla analyzována situace, kdy dochází ke změnám světelných podmínek mezi jednotlivými testy.

Jako první byla testována značka č. 12. Nejprve byly provedeny tři testy za **středního osvětlení** (rozsvíceno jedno svítidlo v blízkosti stanice s manipulátorem). Mezi třetím a čtvrtým testem došlo k vypnutí umělého osvětlení, přičemž následující tři testy proběhly pouze za **přirozeného denního světla**. Nakonec, mezi šestým a sedmým testem, byla rozsvícena všechna svítidla, čímž bylo dosaženo **maximálního osvětlení**.

Tabulka 5.1 Testování značky č. 12 se změnami osvětlení

Test č.	Osvětlení	Úspěšná identifikace	Neúspěšná identifikace	Míra shody [%]
1	Střední	Ano		85,39
2	Střední	Ano		85,11
3	Střední	Ano		84,7
4	Přirozené	Ano		85,39
5	Přirozené	Ano		85,61
6	Přirozené	Ano		85,59
7	Maximální		Ne	
8	Maximální	Ano		80,39
9	Maximální	Ano		83,65
10	Maximální	Ano		84,38

Z provedeného testování vyplývá, že při náhlé změně světelných podmínek směrem k vyšší intenzitě osvětlení došlo v jednom případě k selhání detekce, kdy nebyla rozpoznána žádná značka (viz Tabulka 5.1). Tento jev může být způsoben softwarovým přizpůsobením expozice kamery, která pravděpodobně upravuje jas snímku na základě histogramu předchozího obrazu. Již v následujícím snímku však došlo k úspěšné identifikaci značky, což naznačuje, že systém je schopen se na novou světelnou situaci rychle adaptovat.

Dále byla testována značka č. 23 za podmínek postupné změny intenzity osvětlení. Nejprve byly provedeny tři testy za **středního osvětlení** (rozsvíceno jedno svítidlo poblíž stanice s manipulátorem). Následně, mezi třetím a čtvrtým testem, byla místnost co nejvíce zatemněna – všechna svítidla byla vypnuta a žaluzie zataženy – čímž byly vytvořeny podmínky **minimálního osvětlení**. Nakonec, mezi šestým a sedmým testem, byla rozsvícena všechna svítidla, čímž bylo dosaženo **maximální intenzity osvětlení**.

Tabulka 5.2 Testování značky č. 23 se změnami osvětlení

Test č.	Osvětlení	Úspěšná identifikace	Neúspěšná identifikace	Míra shody [%]
1	Střední	Ano		93,11
2	Střední	Ano		93,32
3	Střední	Ano		93,34
4	Minimální	Ano		83,58
5	Minimální	Ano		89,41
6	Minimální	Ano		89,5
7	Maximální		Ne	
8	Maximální		Ne	
9	Maximální	Ano		84,62
10	Maximální	Ano		90,49

Na základě výsledků uvedených v tabulce (Tabulka 5.2) lze konstatovat, že při náhlém přechodu z minimálního na maximální osvětlení (mezi testy č. 6 a 7) došlo ke dvěma po sobě jdoucím selháním identifikace – ve 7. a 8. testu nebyla detekována žádná značka. Tento jev lze pravděpodobně přičíst softwarové adaptaci expozice kamery, která upravuje jas obrazu na základě analýzy předchozích snímků (pravděpodobně pomocí histogramu). V 9. testu již systém opět správně značku rozpoznal.

Při přechodu mezi testem č. 3 a 4 došlo ke změně osvětlení ze střední na minimální úroveň. Tato změna sice nezpůsobila chybu v identifikaci značky, avšak z tabulky 5.1 je patrný pokles míry shody, což může naznačovat sníženou kvalitu obrazových dat při horších světelných podmínkách.

5.3 Porovnání softwarových řešení

V rámci této bakalářské práce byly navrženy a implementovány dvě varianty softwarového řešení. První varianta byla realizována výhradně v prostředí *TwinCAT* s využitím modulu *VISION*. Druhá varianta kombinuje program vytvořený v jazyce Python, který využívá knihovnu *OpenCV*, s programem vyvinutým v prostředí *TwinCAT*.

Obě varianty vykazují srovnatelné výsledky při zpracování obrazu, a to i přesto, že využívají odlišné metody porovnávání. Určité rozdíly mohou nastat z hlediska časové náročnosti – zejména u varianty s externím zpracováním obrazu v Pythonu, kde může být zpoždění způsobeno komunikací prostřednictvím protokolu *Modbus TCP*. Významnou

výhodou této kombinované varianty je však širší spektrum možností pro analýzu a zpracování obrazu, které knihovna *OpenCV* nabízí.

5.4 Další možnosti využití

Jeden z možných případů využití navrženého systému je navedení manipulátoru nad tiskovou podložku pro 3D tisk. Po detekci referenční značky získá řídicí systém informaci o aktuální poloze v souřadnicovém systému robota. Na základě toho mohou být vyhodnoceny souřadnice pro posun nad podložku pro tisk.

Další možné rozšíření systému představuje implementace pokročilých metod zpracování obrazu, jako je například rozpoznávání objektů pomocí metod strojového učení, zejména s využitím neuronových sítí. Tyto přístupy jsou zvláště vhodné v případech, kdy je požadováno rozlišení více typů objektů, nebo při práci ve složitějších scénách, kde dochází k částečnému překrývání objektů.

Implementace neuronových sítí by mohla významně přispět ke zvýšení robustnosti systému, především v podmínkách s proměnlivým osvětlením, složitým pozadím nebo výskytem rušivých prvků. Je však třeba zohlednit, že nasazení těchto metod vyžaduje předchozí fázi učení, během níž je síť trénována na reprezentativním souboru dat, který odpovídá reálnému provoznímu prostředí a rozpoznávaným objektům.

5.5 Komplikace při implementaci na hardware

Při finálních fázích projektu, kdy byl projekt nahráván do IPC došlo k zjištěným problémům se zprovozněním kinematiky jednotlivých pohonů v nejnovější verzi vývojového prostředí TwinCAT 4026, což zapříčinilo nesoulad v komunikaci.

Tento problém měl být vyřešen přeinstalací prostředí na verzi 4024, u které tyto komplikace neměly nastat, nicméně při odinstalaci verze 4026 pravděpodobně nebyla provedena úplná odinstalace všech komponent softwaru, což znemožňuje instalaci potřebných knihoven.

Systém byl tedy kompletně odsimulován na virtuálním PLC, kdy bylo dosaženo správného chodu stavového automatu, který ovládá manipulátor (tento automat byl již do jisté míry předpřipraven z práce Ing. Cvička). Byla tedy prokázána správná funkce chodu systému při implementaci na virtuální PLC.

Fyzicky byla ovšem otestována pouze část se skriptem napsaným v pythonu a následné čtení hodnot z modbusu v TwinCATu. Při pokusu o testování s pohybem manipulátoru došlo k výše popsaným komplikacím.

5.6 Účel navrženého systému

Tradiční metoda referování SCARA robotů vychází z použití mechanických dorazů a zpětné vazby z enkodérů. V této práci byl navržen alternativní přístup založený na

kamerové detekci referenčních značek umístěných v pracovním prostoru robota.

Kamerový systém přináší výraznou flexibilitu – výchozí pozice lze měnit pouhým přemístěním značek, bez nutnosti mechanických úprav. Umožňuje také rozšíření pracovního prostoru a snadné přizpůsobení různým scénářům, což je výhodné zejména v laboratorních a vývojových podmínkách.

Bezkontaktní princip referování eliminuje opotřebení mechanických částí a umožňuje referenci i při nepřesném zaparkování robota, pokud je značka stále v zorném poli kamery. Dále bylo pomocí vícero značek definováno více referenčních bodů, například pro různé pracovní režimy nebo nouzové návraty.

Systém poskytuje vizuální zpětnou vazbu o nalezení značek a míře shody, což usnadňuje diagnostiku a ladění. Může také sloužit jako doplňkový kontrolní mechanismus pro ověření přesnosti klasického homingu. Výhodou je jednoduchost při rozšiřování systému oproti mechanickým řešením.

6. ZÁVĚR

Tato bakalářská práce se zabývala návrhem a realizací systému referování SCARA robota pomocí kamery umístěné na jeho koncovém rameni. Hlavním cílem bylo navrhnout alternativní metodu, která by využívala zpracování obrazu ke zjištění polohy robota místo tradičních mechanických snímačů.

V teoretické části byly porovnány různé typy kamer s důrazem na jejich vhodnost pro použití v prostředí TwinCAT (viz kapitola číslo 1). Na základě této analýzy byla zvolena průmyslová kamera Basler ace acA1920-25gm a odpovídající objektiv. Následně byl rámci kapitoly č. 2 navržen a implementován softwarový modul pro zpracování obrazu přímo v prostředí TwinCAT pomocí VISION knihoven, a alternativně i pomocí skriptů v jazyce Python, které komunikují přes protokol Modbus TCP (viz kapitola 3).

Součástí kapitoly č. 4 byl i návrh držáku kamery a rozmístění referenčních značek tak, aby byla zajištěna spolehlivá detekce značek v rámci celého pracovního prostoru. Systém byl schopen rozpoznat referenční značky, vyhodnotit jejich polohu a provést referenci manipulátoru.

Z provedené implementace a následného testování, které je popsáno v kapitole č. 5.2, vyplývá, že navržený systém je schopen spolehlivě detekovat referenční značky a předávat výpočty posunu řídicímu systému s vysokou přesností. Byla potvrzena funkčnost za různých světelných podmínek i během jejich změn, přičemž krátkodobé selhání detekce bylo identifikováno jako důsledek adaptivního chování kamery.

Během implementace však došlo k nečekaným komplikacím při nahrávání projektu do IPC s nejnovější verzí prostředí TwinCAT 4026, které neumožnilo korektní zprovoznění kinematiky. Navzdory pokusům o přeinstalaci a návrat k verzi 4024 nebylo možné plně obnovit funkčnost systému na fyzickém PLC, a tak byl projekt otestován pouze na virtuálním PLC. Funkčnost stavového automatu i komunikace mezi Python skripty a prostředím TwinCAT byla ověřena, ale plnohodnotné otestování s pohybem manipulátoru na reálném zařízení se nepodařilo realizovat (viz kapitola 5.5).

Navržený systém přesto ukazuje potenciál pro praktické nasazení, a to zejména v případech, kdy není možné využívat klasické referenční snímače. Do budoucna by bylo vhodné zaměřit se na robustnější integraci s vývojovým prostředím, jakož i na zavedení pokročilejších algoritmů zpracování obrazu, například využívajících strojové učení pro detekci značek v proměnlivých podmínkách.

Literatura

- [1] FURCINITI, ANNA. *Robotic system with TwinCAT Vision: techniques for identifying and manipulating geometric objects on a conveyor belt*. Online, Diplomová práce, vedoucí Giovanni Boschetti. Padova: UNIVERSITÀ DEGLI STUDI DI PADOVA, 2024. Dostupné z: <https://hdl.handle.net/20.500.12608/74884>. [cit. 2024-12-30].
- [2] Camera-Assisted Calibration of SCARA Arms. Online. Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots. 1995, roč. 1995, č. 1, s. 507-512. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/525844>. [cit. 2024-12-30].
- [3] *Ohnisková vzdálenost*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2024. Dostupné z: https://cs.wikipedia.org/wiki/Ohniskov%C3%A1_vzd%C3%A1lenost. [cit. 2024-10-12].
- [4] TYLŠ, Ladislav. *Získávání dat z kamer*. Online, Diplomová práce, vedoucí Miloslav Richter. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky, 2009. Dostupné z: <http://hdl.handle.net/11012/1514>. [cit. 2024-10-12].
- [5] *GigE Vision*. Online. Balluff. 2025. Dostupné z: <https://www.balluff.com/cs-cz/gige-vision>. [cit. 2025-05-22].
- [6] ZMEŠKAL, Jiří. *Využití vývojových platforem pro výrobu kamery*. Diplomová práce, vedoucí Adámek, Milan. Zlín: Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky, Ústav elektroniky a měření, 2018. Dostupné z: <http://hdl.handle.net/10563/43446>. [cit. 2024-10-12].
- [7] *ATESystem. Průmyslová kamera – Industrial camera*. Online. Ostrava: ATESystem s.r.o., 2019. Dostupné z: <https://eshop.atesystem.cz/clanek/179/prumyslova-kamera-industrial-camera> [cit. 2024-10-01].
- [8] *Industrial cameras*. Online. Baumer - Passion for Sensors. 2024. Dostupné z: <https://www.baumer.com/ae/en/product-overview/industrial-cameras-image-processing/industrial-cameras/c/331>. [cit. 2024-10-07].
- [9] *Learn About GigE Vision Cameras*. Online. Phase1vision. 2024. Dostupné z: <https://www.phase1vision.com/resources/learn-about-gige-vision-cameras>. [cit. 2024-10-07].
- [10] *AcA1920-25gm | Basler AG*. Online. Basler. 2024. Dostupné z: <https://www.baslerweb.com/en/shop/aca1920-25gm/>. [cit. 2024-12-26].
- [11] *Basler Lens C125-0818-5M-P*. Online. Basler Product Documentation. 2024. Dostupné z: <https://docs.baslerweb.com/c125-0818-5m-p>. [cit. 2024-12-25].
- [12] *VCS2020-1200 | Area scan camera, 2.5 Gbit/s, Sony IMX565, monochrome, 12.4 MP*. Online. BECKHOFF new automation technology. 2024. Dostupné z:

- <https://www.beckhoff.com/cs-cz/products/vision/cameras/vcs2000-area-scan-cameras-2-5-gbit-s/vcs2020-1200.html> . [cit. 2024-12-26].
- [13] *IP camera*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: https://en.wikipedia.org/wiki/IP_camera. [cit. 2024-12-26].
- [14] *VIGI C350*. Online. VIGI by tp-link. 2024. Dostupné z: <https://www.tp-link.com/cz/business-networking/vigi-network-camera/vigi-c350/>. [cit. 2024-12-26].
- [15] *C922 Pro Stream 1080p Webcam + software Capture*. Online. Logitech. 2024. Dostupné z: <https://www.logitech.com/cs-cz/products/webcams/c922-pro-stream-webcam.960-001088.html?> . [cit. 2024-12-28].
- [16] EPSON. *USER'S MANUAL SRC-300/320 (REV.2)*. Epson Europe, Online. Dostupné z: https://download.epson-europe.com/robotics/share/E-Serie/USER%27S%20SRC-300_320%28REV.2%29.PDF [cit. 2025-05-04].
- [17] *Manual TwinCAT 3 / Vision - TF7000-TF7800_TC3_Vision_en*. Online. BECKHOFF new automation technology. 2024. Dostupné z: https://download.beckhoff.com/download/document/automation/twincat3/TF7000-TF7800_TC3_Vision_en.pdf#ID0EH1NO. [cit. 2024-12-27].
- [18] *OpenCV. Geometric Transformations of Images: Rotation*. Online. Dostupné z: https://docs.opencv.org/4.x/da/d6e/tutorial_py_geometric_transformations.html [cit. 2025-05-04].
- [19] *OpenCV. Image Processing: Object Detection – Enumeration Type Documentation*. Online. Dostupné z: https://docs.opencv.org/3.4/df/dfb/group_imgproc_object.html#gga3a7850640f1fe1f58fe91a2d7583695dac6677e2af5e0fae82cc5339bfaef5038 [cit. 2025-05-04].
- [20] BECKHOFF Automation. *Mapping between Modbus and ADS*. Online. Beckhoff Information System Dostupné z: <https://infosys.beckhoff.com/content/1033/tcmodbussrv/11379400715.html>
- [21] CVIČEK, Pavel. *Řídicí systém SCARA robotu* [online]. Brno, 2024 [cit. 2024-02-27]. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/159418> . Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Tomáš Benešl.
- [22] ATESystem. *Držák s chladičem pro kamery SugarCube* [online]. Ostrava: ATESystem s.r.o. Dostupné z: <https://www.atesystem.cz/produkt/drzak-s-chladicem-pro-kamery-sugarcube> [cit. 2025-04-16].

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

CMOS	Complementary metal-oxide-semiconductor
FPS	Frames per second
GigE	Gigabit Ethernet
IP	Internet protocol
IPC	Industrial PC
MAC adresa	Media Access Control address
MP	Megapixel
PLC	Programmable logic controller
PoE	Power over ethernet
RJ-45	Registered Jack-45
SCARA	Selective compliance assembly robotic arm
USB	Universal serial bus

Symboly:

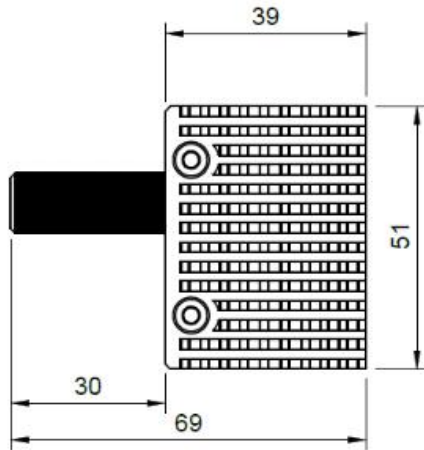
f	ohnisková vzdálenost	(mm)
WD	Pracovní vzdálenost	(mm)
hFoV	horizontální zorné pole	(mm)
dFoV	diagonální zorné pole	(mm)

SEZNAM PŘÍLOH

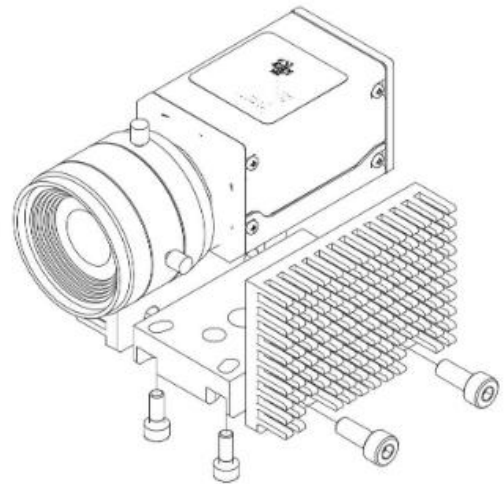
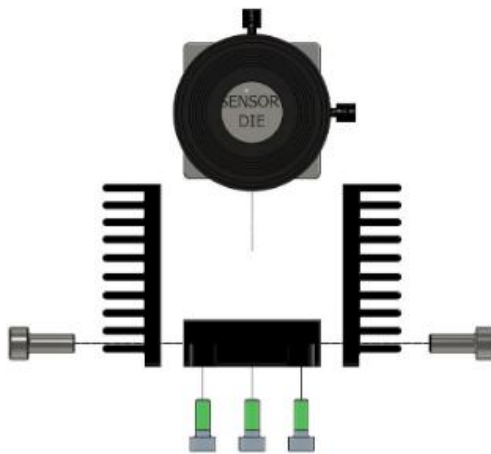
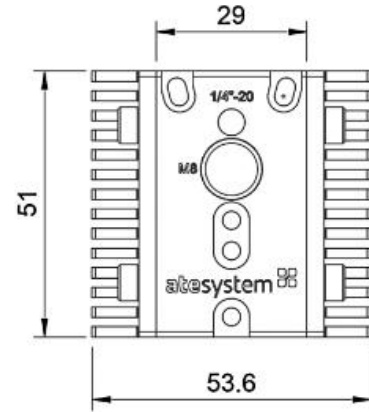
PŘÍLOHA A - TECHNICKÝ VÝKRES KAMERY	55
PŘÍLOHA B - MONTÁŽNÍ VÝKRESY CHLADIČE KAMERY.....	56
PŘÍLOHA C - STAVOVÝ AUTOMAT KAMERY	57
PŘÍLOHA D - SEZNAM POUŽITÝCH REFERENČNÍCH ZNAČEK	58

Příloha B - Montážní výkresy chladiče kamery

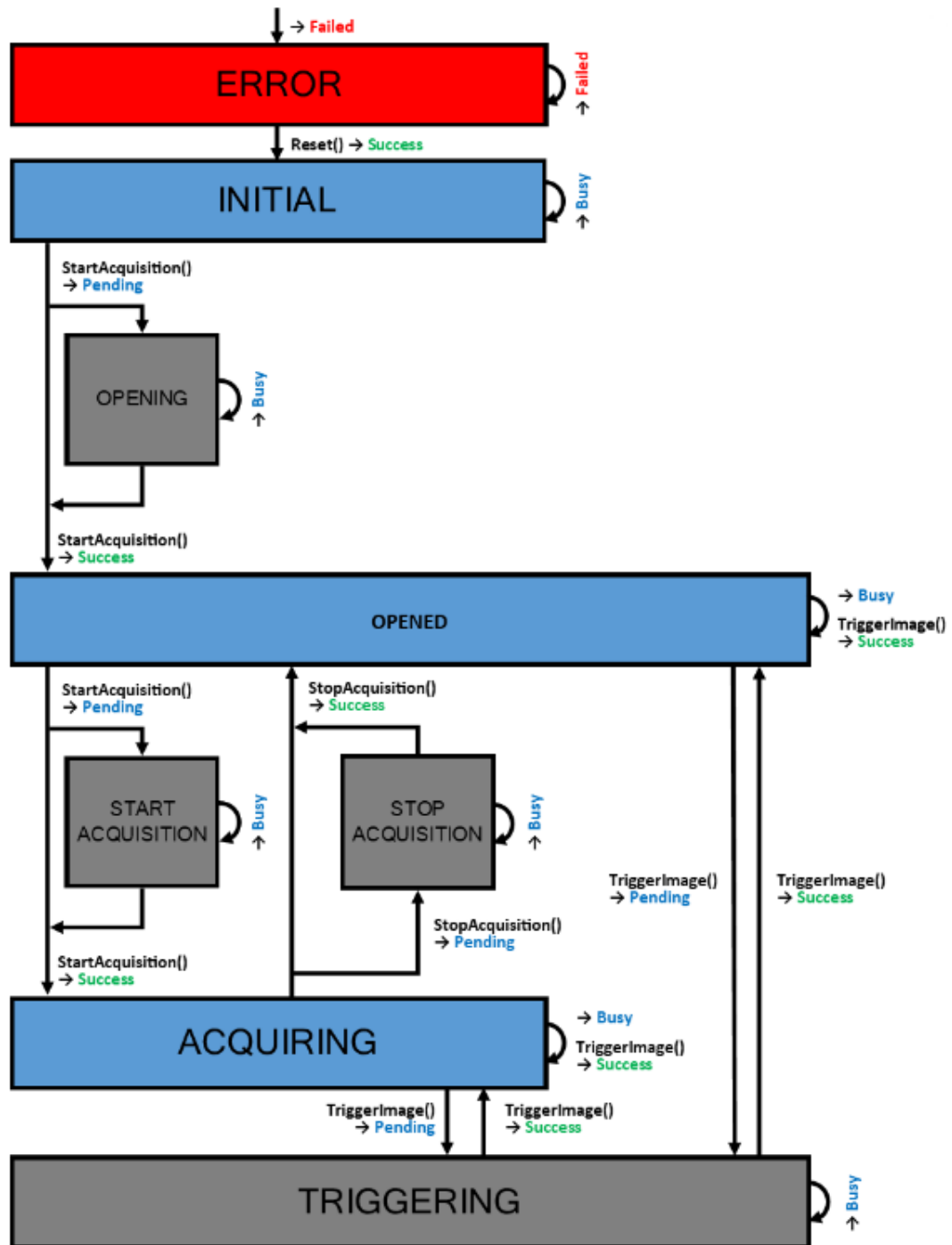
Pohled z boku:



Pohled zespodu:



Příloha C - Stavový automat kamery



Příloha D - Seznam použitých referenčních značek

