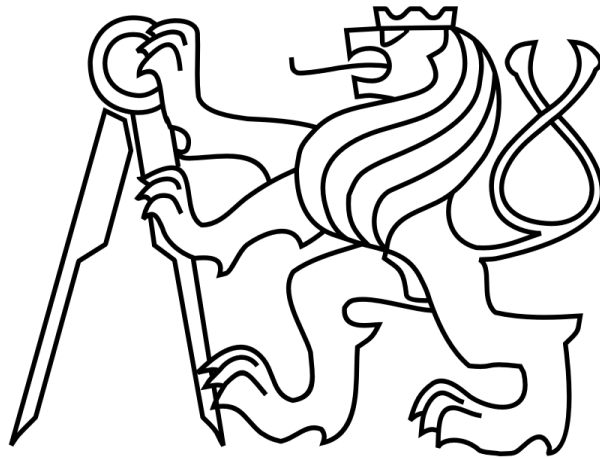


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

Ústav výrobních strojů a zařízení



Bakalářská práce

Detekce polohy, orientace a vlastností dílců pro aplikace pick and place

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Just** Jméno: **Dominik** Osobní číslo: **419604**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav výrobních strojů a zařízení**
Studijní program: **Teoretický základ strojního inženýrství**
Studijní obor: **bez oboru**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Detekce polohy, orientace a vlastností dílců pro aplikace pick and place

Název bakalářské práce anglicky:

Detection of the position, orientation and properties of parts for pick and place applications

Pokyny pro vypracování:

Popis tématu: Navrhněte systém strojového vidění pro manipulátor Fischertechnik s cylindrickou kinematickou strukturou. Osnova práce: Zpracujte přehled výrobců a systémů strojového vidění. Navrhněte vlastní systém strojového vidění na platformě National Instruments, tak aby poskytoval informaci o poloze a orientaci manipulovaných objektů v rovině a o jejich barvě. Systém by měl rozlišovat 3 tvary objektů a 3 barvy. Spuštění detekce bude v pevném časovém cyklu, nebo na základě vnějšího požadavku (např. přes digitální vstup). Řízení manipulátoru je řešeno na PLC Siemens Simatic S7-1500. Navrhněte způsob komunikace s tímto PLC. Navržený systém zprovozněte a otestujte (samostatně bez PLC - nad rámec BP).

Rozsah grafické části: 3D modely potřebných dílců, kompletní program s komentáři, uživatelský návod
Rozsah textové části: 40 - 60 stran.

Seznam doporučené literatury:

- [1] NI Vision: NI Vision Assistant Tutorial [online]. 372228M. Austin, Texas, USA, 2011 [cit. 2017-03-30]. Dostupné z: <http://www.ni.com/pdf/manuals/372228m.pdf>
- [2] DOERING, Ed. NI myRIO Vision Essentials Guide [online]. July 23, 2015. National Technology and Science Press, 2015 [cit. 2017-03-30]. Dostupné z: <http://www.ni.com/white-paper/52475/en/>
- [3] BERGER, Hans. Automating with SIMATIC S7-1500 Configuring, Programming, Motion Control and Security inside TIA Portal. Erlangen: PUBLICIS, 2013. ISBN 9783895784040.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

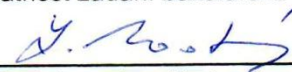
Ing. Lukáš Novotný Ph.D., ústav výrobních strojů a zařízení FS


Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **10.04.2017**

Termín odevzdání bakalářské práce: **10.07.2017**

Platnost zadání bakalářské práce: **30.09.2017**


Podpis vedoucí(ho) práce



Podpis vedoucí(ho) ústavu/katedry


Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

26.7.2017
Datum převzetí zadání


Podpis studenta

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a že jsem uvedl v příloženém seznamu veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací, vydaným ČVUT v Praze 1. 7. 2009.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne

.....
podpis

Poděkování

Úvodem bych chtěl poděkovat Ing. Lukáši Novotnému, Ph.D. za vedení této bakalářské práce a Ing. Ondřeji Dundáčkovi za technickou podporu. Dále bych chtěl poděkovat Kamilu Vrbíčkoví za podněty a podklady k řešení práce. Snad poděkovat bych měl všem, kdož mi byli radou, ba i přítelem, těm kterým jsem stál za úsměv vlídný, stěží by i vyšel člověk bez almužny, nespočtu těch co základy kladly, jiných podpírajíc je, by neupadly. Děkuji Zemi, co dává mléko a strdí, národu co se snad za to vděčen být stydí, že nebyl by lepší, toť musel bych lhát, a přec je mi drahý, a přec ho mám rád. Však největší dík lze vyslovit stěží, vždyť patří Jemu, jež vše v rukách drží, oblohu, Zemi, Slunce jak hvězdy. On při mně stál a bude i navždy.

Anotace

Jméno autora:	Dominik Just
Název BP:	Detekce polohy, orientace a vlastností dílců pro aplikace pick and place
Rozsah práce:	60 stran
Akad. rok vyhotovení:	2016/2017
Ústav:	Ústav výrobních strojů a zařízení
Vedoucí BP:	Ing. Novotný Lukáš, Ph.D.
Zadavatel tématu:	ČVUT FS, Ú12135
Využití:	Obohacení výuky ve školní laboratoři na stavebnici manipulačního robota o systém strojového vidění
Klíčová slova:	strojové vidění, řízení, detekce polohy, LabVIEW, vestavné zařízení NI myRIO, kamera;
Abstrakt:	Předložená bakalářská práce pojednává o tvorbě systému strojového vidění, který je aplikován na školního laboratorního robota s cílem obohatit výuku. Sestává se z rešerše na téma machine vision a popisu postupu zhotovení stojanu, kamery, testovacích dílců a inspekčního programu.

Annotation

Author:	Dominik Just
Title:	Detection of the position, orientation and properties of parts for pick and place applications
Extent:	60 pages
Academic Year:	2016/2017
Department:	Department of Production Machines and Equipment
Tutor:	Ing. Novotný Lukáš, Ph.D.
Submitter:	ČVUT FS, Ú12135
Application:	Extension of teaching on a robot manipulator in a school lab by machine vision system.
Keywords:	machine vision, control, detection of position, LabVIEW, NI myRIO embedded device, camera;
Abstract:	The bachelor thesis deals with the creation of the machine vision system, which is applied to the school laboratory robot in order to extend the teaching. It consists of a machine vision survey and a description of making a camera stand, test parts, and inspection program.

Obsah

1	Úvod.....	10
2	Rešerše - Strojové vidění.....	10
2.1	Princip strojového vidění.....	10
2.2	Základní čtyři druhy systémů strojového vidění.....	11
2.2.1	Kamerový senzor (vision sensor).....	11
2.2.2	Inteligentní kamera (smart camera).....	11
2.2.3	PC systémy (PC vision).....	12
2.2.4	Zákaznické systémy (custom vision).....	13
2.3	Software.....	13
2.3.1	Úlohy.....	13
2.3.2	Softwarové nástroje.....	16
2.4	Kamery a jejich části.....	18
2.4.1	Snímač obrazu.....	18
2.4.2	Elektrické obvody kamery.....	19
2.4.3	Optická soustava.....	20
2.4.4	Mechanické části kamery.....	21
2.5	Osvětlovače.....	21
2.5.1	Geometrie osvětlení.....	22
2.5.2	Druhy osvětlovačů.....	23
2.5.3	Řízení osvětlovačů.....	27
2.5.4	Eliminace rušivých vlivů.....	27
2.6	Přehled výrobců systémů strojového vidění.....	28
2.6.1	Komerční systém.....	28
2.7	Charakteristika zařízení použitého v naší úloze.....	29
2.7.1	Vestavné zařízení NI myRIO.....	29
2.7.2	Kamera Trust 1.3MPx.....	29
2.8	Přehled komunikačních rozhraní a protokolů.....	30
3	Vlastní řešení.....	31
3.1	Stojan na kameru.....	31
3.2	Dílce.....	33
4	Vytvoření inspekčního programu v prostředí LabVIEW.....	33
4.1	Požadavky na program.....	33
4.2	Rozvržení programu.....	35

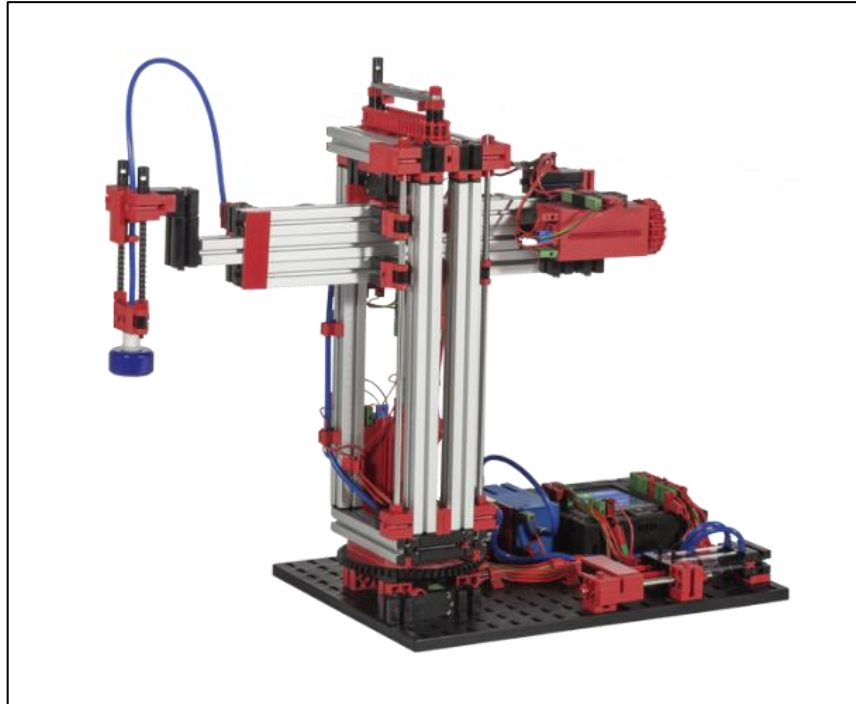
4.3	Vytvoření projektu	35
4.4	Získání obrazu z kamery	36
4.5	Vyhodnocení získaného obrazu ve Vision Assistant	37
4.5.1	Kalibrace	37
4.5.2	Binarizace	38
4.5.3	Převod na černobílý obraz	39
4.5.4	Zaplnění děr a odstranění malých objektů	39
4.5.5	Detekce tvaru orientace a polohy.....	39
4.6	Vložení Vision Assistant do smyčky	40
4.7	Připojení vstupů a výstupů	41
4.7.1	Vstupní obraz.....	41
4.7.2	Vytvoření pole vstupních hodnot pro binarizaci	42
4.7.3	Vystupující obraz	43
4.7.4	Nastavení detekce tvaru	43
4.7.5	Cesta ke složce se vzorovými tvary a kalibrací	44
4.7.6	Získání potřebných výsledků detekce z datového pole na výstupu	44
4.7.7	Rozdělení jednotlivých složek výsledků	45
4.7.8	Podmíněné ukončení For Loop smyčky	46
4.7.9	Vizualizace hodnot Score	46
4.8	Zpracování a přepočet dat.....	47
4.8.1	Vytvoření struktury přepínacího subdiagramu	48
4.8.2	Přepočet souřadnic x a y.....	48
4.8.3	Převod do cylindrických souřadnic	49
4.8.4	Nulové hodnoty natočení kruhu	49
4.8.5	Výsledné natočení	49
4.8.6	Zobrazení textových řetězců.....	50
4.9	Spuštění programu a jeho komunikace s okolím	51
4.9.1	Sdílené proměnné	51
4.9.2	Spuštění obrazové inspekce přes tlačítko na zařízení NI myRIO	52
4.9.3	Konečné propojení jednotlivých částí blokového diagramu	52
4.10	Tvorba uživatelského rozhraní.....	53
4.11	Nahrání programu na NI myRIO a nastavení automatického spuštění	54
4.12	Ověření funkčnosti	55
5	Závěr	56

Seznam použitých zkratk v angličtině

CCD	Charged Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
FPGA	Field Programmable Gate Array
HMI	Human Machine Interface
I/O	Input/Output
LED	Light-Emitting Diode
PC	Personal Computer
PLC	Programmable Logic Controller
ROI	Region of Interest
T/F	True/False
USB	Universal Serial Bus

1 Úvod

Cílem bakalářská práce je doplnit laboratorní stavebnici manipulačního programovatelného robota Vacuum Gripper Robot 24V od firmy fischertechnik (obrázek 1) kamerou, osvětlením a počítačem s programem tak, aby bylo možné zjišťovat a posílat informace o tvaru, barvě a poloze součástí v jeho manipulačním prostoru. Tomuto způsobu řízení se říká strojové vidění, proto o něm pojednává i rešerše.



Obrázek 1 Vacuum Gripper Robot 24V od firmy fischertechnik [3]

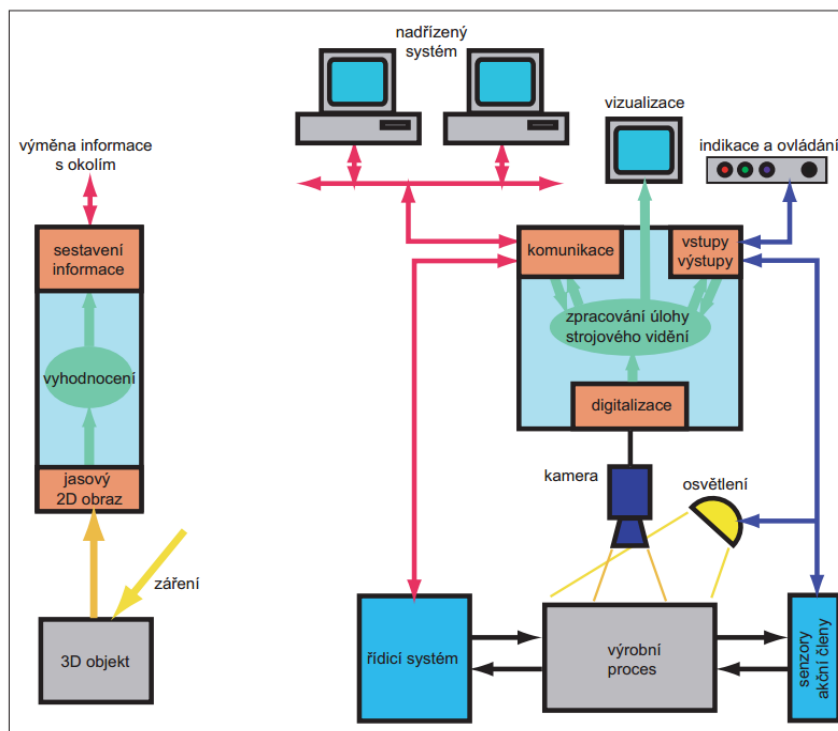
2 Rešerše - Strojové vidění

Rešerší by mělo být zmapováno jak strojové vidění obecně pracuje, rozčlenit základní druhy provedení těchto systémů, přiblížit strukturu softwaru a základní části hardwaru. U hardwaru půjde hlavně o kamery a osvětlovače, protože právě na těchto částech závisí, zda bude možné z pořízeného digitálního obrazu skutečnosti možné získat potřebné informace.

2.1 Princip strojového vidění

Podle [7]: Sledovaný objekt je většinou ozářen viditelným světlem pocházejícím z umělých zdrojů – osvětlovačů. Dvojměrný jasový obraz je vytvářen na snímacím prvku kamery. V kameře s polovodičovým snímacím prvkem jsou měronosnými veličinami náboje či proudy poskytované prvky obrazového snímače, které jsou citlivé na světlo. (Obr. 2 vlevo) Měronosné veličiny mohou nést informaci redukovanou na údaj o kombinovaném jasu barevných složek dopadajícího světla (černobílé kamery) nebo mohou nést i informaci o barvě zkoumaného předmětu (barevné kamery). Protože je potřeba zpracování informací počítačem, je nutné převést analogové měronosné veličiny do digitálního tvaru. Informace v digitální podobě je v počítači zpracována vhodnými algoritmy tak, aby byla získána požadovaná informace o objektu. Získaná informace má opět digitální podobu a předává se do okolí prostřednictvím digitálních výstupů nebo digitálního komunikačního rozhraní. Komunikační rozhraní je většinou určeno i k připojení zařízení, které umožňuje systém nastavovat,

programovat nebo udržovat. Toto zařízení se označuje HMI (Human-Machine-Interface). Někdy je v systému k dispozici i původní digitální obraz. Ten se využívá v případě, kdy je požadována nepřetržitá vizualizace sledovaného objektu. Pro strojové vidění je typické, že výměna informací systému s okolím funguje jako zpětná vazba při řízení výrobního procesu. Systém získává z procesu nezbytné vstupní údaje. Typicky to bývá povel k pořízení snímku v okamžiku, kdy je sledovaný objekt ve vhodné poloze. V závislosti na výsledku vyhodnocení obrazu systém obvykle vykoná akční zásah do procesu, nejčastěji indikaci nebo vyřazení vadného kusu. Zásah mohou zprostředkovat přímo digitální výstupy systému strojového vidění nebo je informace o výsledku vyhodnocení předána do řídicího systému, který provede zásah sám. (Obr. 2 vpravo)



Obrázek 2 Obecné uspořádání systému strojového vidění [7]

2.2 Základní čtyři druhy systémů strojového vidění

Podle [7] lze rozdělit systémy strojového vidění na tyto čtyři druhy:

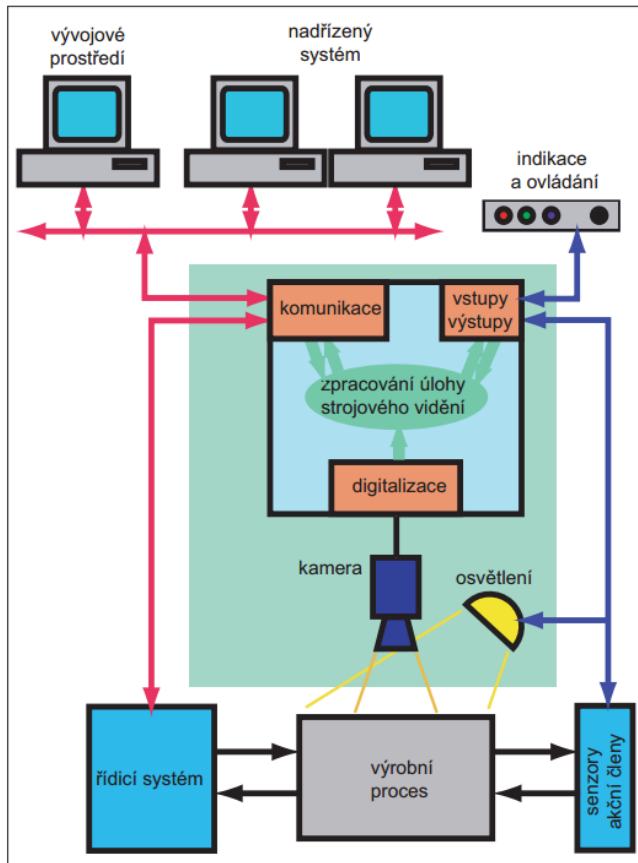
2.2.1 Kamerový senzor (vision sensor)

Celý systém včetně kamery a osvětlení bývá integrován do jednoho pouzdra. (obrázek 3) Počet digitálních vstupů a výstupů je omezen, a protože má kamerový senzor velmi omezené možnosti programování, mají vstupy a výstupy pevně přidělenou funkci. Používají se standardní tranzistorové výstupy PNP nebo NPN. Rychlé komunikační rozhraní (např. Ethernet) se většinou používá pouze k připojení k HMI pro nastavení parametrů úlohy. Někdy se parametry dokonce nastavují pouze z vestavěné klávesnice.

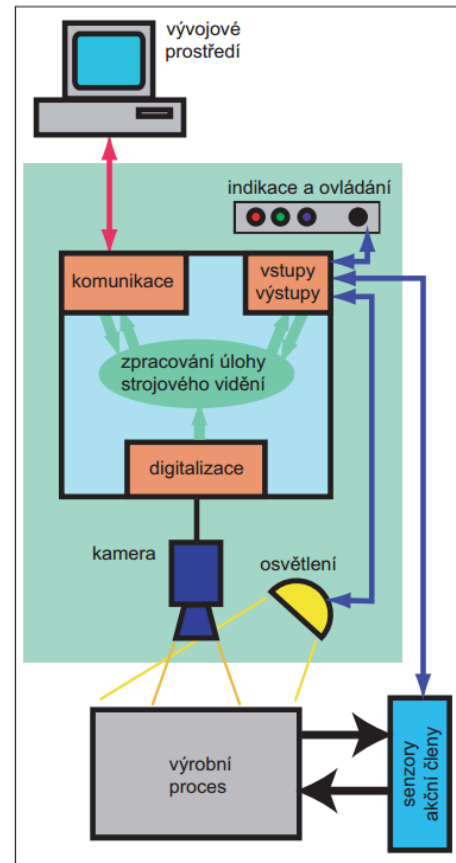
2.2.2 Inteligentní kamera (smart camera)

Inteligentní kamera je již kompletní systém strojového vidění. Jejími charakteristickými znaky jsou kompaktnost a použití výkonného mikropočítače jako vyhodnocovací jednotky. Jako digitální vstupy a výstupy se často uplatňují univerzální bitová rozhraní nastavitelná volitelně jako vstupy nebo

výstupy. Komunikační rozhraní plní u inteligentní kamery téměř vždy několik funkcí. Tou hlavní je většinou připojení k HMI, které umožňuje kameru programovat. Pro pohodlný vývoj úlohy je vhodné, aby HMI zobrazovalo obraz snímáný kamerou v reálném čase. Přenosová rychlost pak musí být poměrně vysoká, a proto se jako základní komunikační rozhraní nejčastěji používá Ethernet. Ten také může bez problémů přenášet data do nadřazeného řídicího systému nebo zprostředkovat komunikaci mezi několika kamerami. (Obr. 4)



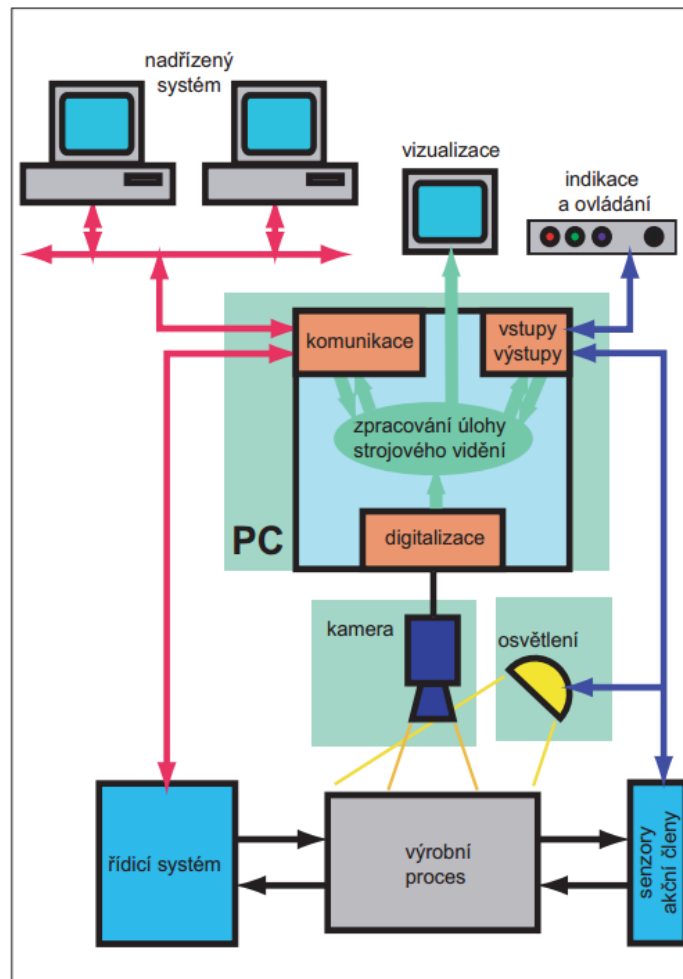
Obrázek 3 Blokové schéma inteligentní kamery [7]



Obrázek 4 Blokové schéma kamerového senzoru [7]

2.2.3 PC systémy (PC vision)

Pro PC systémy je charakteristické použití osobního počítače se standardním operačním systémem na místě vyhodnocovací jednotky. Dalším charakteristickým rysem je modularita. Kamera je samostatná a většina systémů umožňuje připojení i několika kamer. Kamery s analogovým výstupem se připojují prostřednictvím zásuvné karty digitalizátoru obrazu (frame grabber), kamery s digitálním výstupem přes speciální kamerové rozhraní (např. CameraLink) nebo s využitím vysokorychlostního Ethernetu. Výpočetní výkon a velikost paměti jsou dány použitým počítačem. Vstupy a výstupy i komunikační rozhraní se realizují způsobem běžným v technice PC, nejčastěji pomocí zásuvných karet. Předmětem dodávky systému PC tedy většinou bývá pouze sada softwaru, která obsahuje prostředí pro vývoj úlohy strojového vidění a run-time modul. Obvyklá je i dodávka karty digitalizátoru nebo rozhraní pro digitální kameru, pro které je program optimalizován.



Obrázek 5 Blokové schéma PC systému strojového vidění [7]

2.2.4 Zákaznické systémy (custom vision)

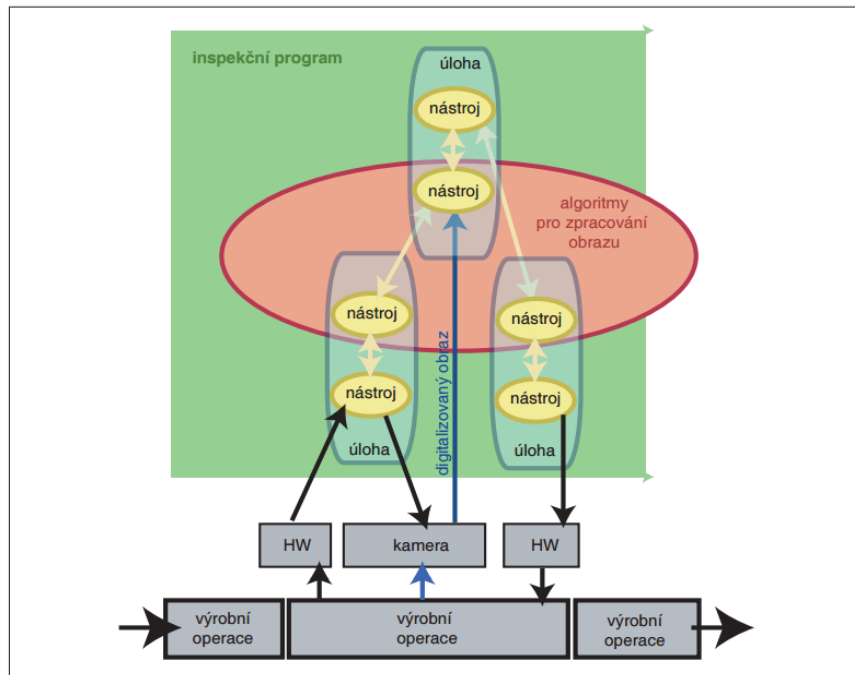
Do této kategorie patří systémy stavěné na míru v případě, kdy nelze použít žádnou ze zmíněných variant. Většinou jde o úlohy vyžadující vysokou rychlost zpracování, speciální algoritmy nebo nestandardní kamery. Nejčastěji se k realizaci používají rychlé signálové procesory nebo hradlová pole se softwarem vytvořeným na míru.

2.3 Software

Pro vytvoření inspekčního programu vyhodnocujícího obraz většinou stačí použití algoritmů řešící základní typy úloh, které se v aplikacích strojového vidění stále opakují, proto jsou vytvořeny vývojové prostředí, kde se pouze sestavují různé kombinace již hotových nástrojů. Tyto úlohy a nástroje se podle [8] dají specifikovat takto:

2.3.1 Úlohy

Komerčně není výhodné vyvíjet pro každou úlohu strojového vidění speciální program. Existuje jen omezený počet úloh, které se v různých variacích ve výrobě opakují a pro jejich řešení stačí určitý soubor již hotových nástrojů. Stejně se postupuje při vývoji řídicích systémů, které lze sestavit v programech z již vytvořených nástrojů. (obrázek 6)

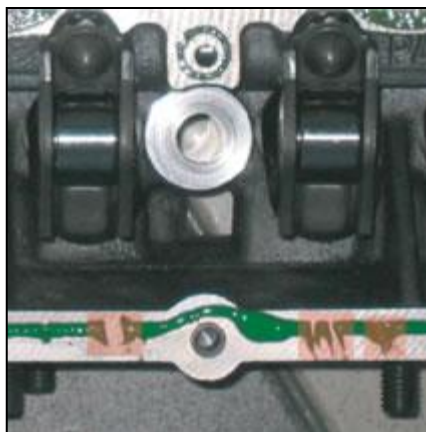


Obrázek 6 Vztah úlohy nástroje a algoritmu v inspekčním programu [8]

Strojové vidění většinou řeší kontrolu produktů tak, aby byla zaručena jejich naprostá bezchybnost při výstupu z výrobního procesu. Tyto kontrolní úlohy se opakují v různých modifikacích a lze je rozdělit podle [1] do těchto skupin:

2.3.1.1 Nalezení povrchových vad

Tato úloha je typická pro fázi shromažďování materiálu k operaci provázené vstupní kontrolou. Může jít o identifikaci poškozených povrchů, vad nátěrů či povrchových úprav, neúplných potisků atd. Je-li nátěr sám výrobní operací, uplatní se tento typ úlohy i ve fázi vyzvednutí výsledného produktu. Na obrázku 7 je ukázán příklad špatně nanesené housenky těsnicího tmelu.



Obrázek 7 Příklad úlohy nalezení povrchové vady: housenka těsnicího tmelu [1]

2.3.1.2 Rozpoznání, nalezení polohy a počítání

Tato úloha může mít mnoho podob. Nejčastěji se v ní zjišťuje, je-li správný objekt na správném místě. Například před provedením operace zalisování je nutné se přesvědčit, zda jsou k zalisování připraveny všechny díly ve správné poloze. Totéž je třeba udělat po skončené operaci. Zmíněná úloha se rovněž vyskytuje při vstupní kontrole. Může jít o identifikaci deformované části, otřepů po stříhání, nevyvrtaného otvoru nebo poškozené hrany. Počítání se uplatní při kontrole naplněné přepravky nebo blistru (obrázek 8).



Obrázek 8 Příklad úlohy počítání: kontrola naplnění blistru s léky [1]

2.3.1.3 Měření a kontrola tolerancí

Úloha měření prostupuje celou výrobní operací. Měří se tolerance vstupních dílů, poloha dílů před operací i tolerance hotového výrobku. Na obrázku 9 je příklad kontroly tolerancí kulového čepu.



Obrázek 9 Příklad úlohy kontroly tolerancí: měření závitu a kužele na kulovém čepu [1]

2.3.1.4 Identifikace barev

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



Obrázek 10 Příklad úlohy identifikace barev: třídění dílců podle barvy [1]

2.3.1.5 Čtení kódů a textů

Kódy, ať již čárové nebo maticové, se čtou hlavně v první a druhé fázi výrobní operace, kdy je zjišťováno, zda je k operaci připraven správný díl na správném místě. V průběhu operace někdy bývá vytvořen další kód, jenž se verifikuje v poslední fázi operace. Čtení nebo verifikace kódů jsou málokdy samostatnou úlohou strojového vidění ve výrobní operaci, ale je-li tomu tak, je výhodnější svěřit ji specializované čtečce. Vykonává-li však systém strojového vidění v dané operaci více úloh, je výhodné použít ho i pro čtení kódů a textů.



Obrázek 11 Příklad úlohy čtení textu: kontrola data výroby nebo expirace [1]

2.3.2 Softwarové nástroje

Nástroje pro řešení potřebných úloh si lze představit jako programové bloky, do kterých vstupují data, která jsou zpracovávána podle zadaných parametrů, a výsledek zpracování ve formě dat je předáván dál. Data vystupující z jednoho nástroje mohou vstupovat do dalšího jako jeho vstupy či parametry. Počet a výkonnost nástrojů a možnosti jejich propojení určují výkonnost a flexibilitu softwaru systému. Vývoj inspekčního programu obvykle probíhá tak, že uživatel použije nástroje na referenční (bezchybný) objekt a parametrizuje je tak, aby inspekci prošel jen bezchybný objekt. Poté se na základě výsledku teoretického výpočtu, simulace nebo použití skutečného objektu s vadami upravuje parametrizace nástrojů tak, aby byly spolehlivě identifikovány chyby, jejichž odhalení je požadováno. Součástí parametrizace bývají různé režimy „učení“ podle správného či chybného objektu. Při parametrizaci také bývá změna působení nástroje graficky znázorněna přímo v obraze objektu. Dále jsou k dispozici pomocné údaje, často v názorné grafické podobě.

Pro činnost systému strojového vidění jsou nutné nástroje pro tyto funkce:

2.3.2.1 Nástroje pro zpracování obrazu

Nejvýznamnějším znakem těchto nástrojů je, že jedním z jejich vstupů vždy bývá souvislá část obrazu, ve které je nástroj používán. Oblast se označuje zkratkou ROI (Region Of Interest). Software pro zpracování obrazů lze rozdělit na tři skupiny:

- filtry, které provádějí matematické operace s hodnotou jasu každého bodu digitalizovaného obrazu v ROI,
- nástroje pracující s tzv. hranami, což jsou rozhraní oblastí s rozdílným jasnem,
- nástroje pracující s rozložením jasu v ROI.

2.3.2.2 Nástroje pro měření v obrazu

Softwarové nástroje pro měření v obrazu pracují typicky s výstupy nástrojů pro zpracování obrazu. Většinou měří geometrické útvary, a to tím, že aproximují jejich hrany. Nejčastěji jde o měření délek, nebo ploch. Aby bylo možné stanovit délku ve skutečných délkových jednotkách, je nutné podrobit digitalizovaný obraz kalibraci. Problém při kalibraci spočívá v tom, že digitalizovaný obraz nemůže být lineárním zobrazením sledované scény. Když se převede trojrozměrná scéna na dvojrozměrný obraz, většinou vznikne zkreslení perspektivou. Použitá optická soustava dále vykazuje různé optické chyby a při digitalizaci se projeví chyba vzorkování.

2.3.2.3 Nástroje pro optické rozeznávání znaků a kódů

Nástroje pro optické rozeznávání znaků (OCR) musí zvládnout podstatně složitější úkol než komerční programy pro čtení tištěného textu. Tvar písmena bývá deformován vlivem nerovné podložky (čtení z obalů), musí být vyřešeno i čtení různých typů písma. Někdy lze použít nástroje pro vyhledání motivu na základě podobnosti.

2.3.2.4 Nástroje pro obsluhu hardwaru

Nástroje pro obsluhu hardwaru lze najít hlavně u kamerových sensorů a inteligentních kamer. U PC systémů je možné využít knihovny dodávané výrobcem příslušných periférií. Nástroje lze rozdělit do tří skupin pro tyto operace:

- práce se vstupy a výstupy,
- práce s komunikačním rozhraním,
- nastavení kamery.

2.3.2.5 Nástroje pro řízení postupu inspekce

Softwarové nástroje pro řízení postupu inspekce jsou součástí vývojového prostředí inspekčního programu. Umožňují větvit postup inspekce, volat podprogramy, zařazovat externí programy a kódy, obsluhovat interní čítače událostí, časové plánovače, spouštěcí podmínky a vykonávat další činnosti nezbytné pro chod inspekčního programu. U kamerových sensorů s omezenou flexibilitou vyhodnocení je těchto nástrojů k dispozici méně než u inteligentních kamer a softwaru pro PC systémy.

2.3.2.6 Nástroje pro zacházení s daty

Tyto nástroje se objevují jen u nejsložitějších programů pro inspekci. Umožňují např. transformovat geometrické útvary, měnit instance dat, sestavovat vlastní komunikační protokol.

2.4 Kamery a jejich části

Podle [9]: Kamerou se rozumí zařízení, které vytváří dvojrozměrný jasový obraz trojrozměrné reality a ten převádí na měronosné veličiny. Na příslušném rozhraní vysílá kamera tyto veličiny v dohodnuté podobě, buď jako analogový signál (videosignál), nebo digitální signál (digitalizovaný obraz). Kamera může být v systému pro strojové vidění přímo vestavěna, nebo pouze připojena (PC systém).

2.4.1 Snímač obrazu

Snímač obrazu je matice polovodičových buněk citlivých na světlo, v nichž se po dobu expozice akumuluje elektrický náboj úměrný osvětlení buňky. Typ a vlastnosti snímače obrazu jsou dány jednak technologií jeho výroby, jednak způsobem sběru náboje z buněk. Pro použití systémů strojového vidění v praxi většinou stačí znát základní vlastnosti dvou nejpoužívanějších snímačů obrazu.

2.4.1.1 CCD (Charged Coupled Device)

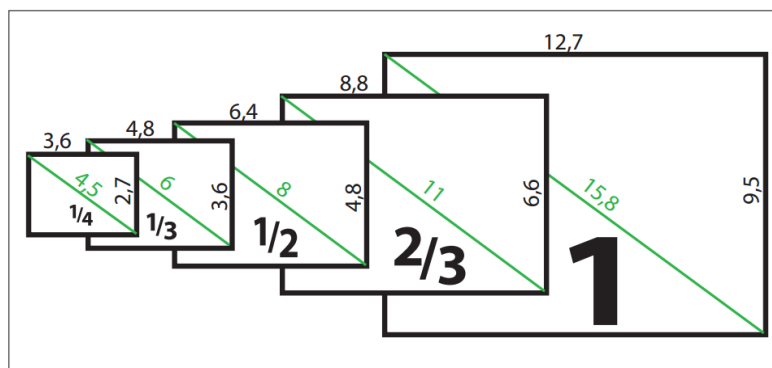
Snímače obrazu CCD vysouvá náboj akumulovaný ve světlocitlivých buňkách pomocí soustavy analogových posuvných registrů. Výstupem je tedy analogový signál, který řídící obvody kamery doplní potřebnou informací pro synchronizaci. Podle způsobu snímání řádků se kamery CCD dělí na kamery s prokládaným řádkováním (interlaced) a kamery pracující metodou progressive scan, kdy se vysouvá náboj akumulovaný v buňkách všech obrazových řádků v jednom okamžiku. Technologie CCD je poměrně nákladná a snímače potřebují k činnosti několik různých napájecích napětí. Navíc je nutné jejich analogový videosignál digitalizovat v následných obvodech, v současné době většinou vyrobených technologií CMOS.

2.4.1.2 CMOS (Complementary Metal Oxide Semiconductor).

Snímač CMOS má integrován snímač a A/D převodník na jednom čipu, tudíž poskytuje již digitalizovaný signál, což zjednodušuje konstrukci kamery. Odlišný způsob vybírání náboje z obrazových buněk také umožňuje vybírat obraz jen z části plochy snímače obrazu (tzv. subscan). Snímače CMOS však nejsou schopny poskytnout tak kvalitní obraz jako CCD. Proto se montují do levnějších kamer.

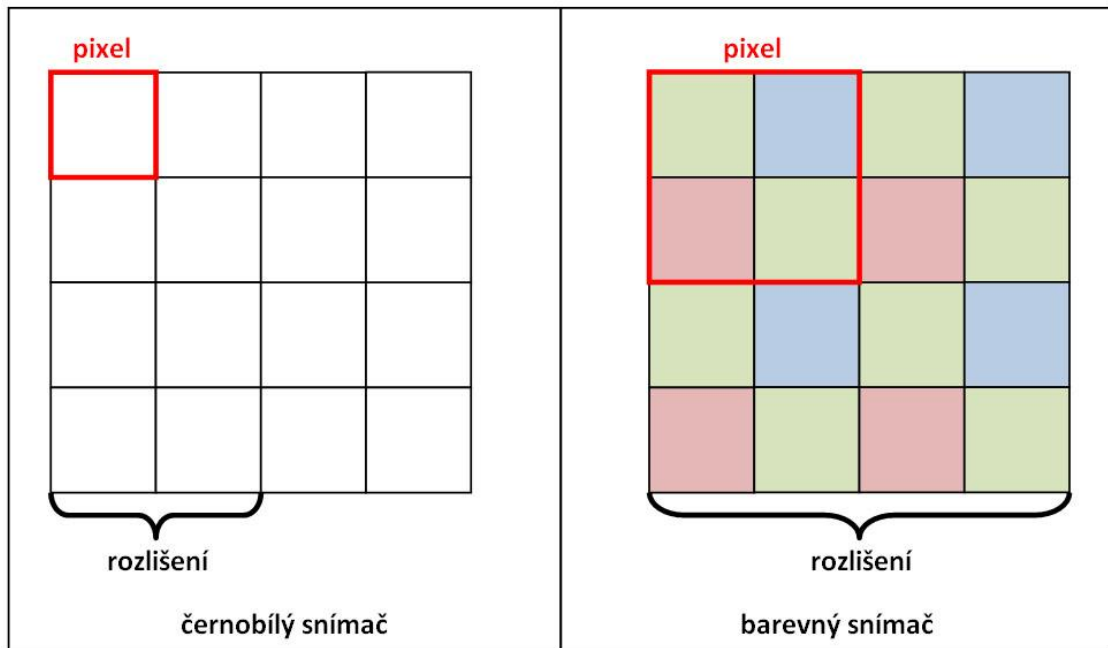
2.4.1.3 Velikost snímače

Velikost snímače obrazu se udává v palcích. Toto číslo však ve skutečnosti nevyjadřuje žádný rozměr snímače, jde o průměr skleněné trubice odpovídající ekvivalentní historické snímací elektronce. Skutečné rozměry snímací plochy nejpoužívanějších snímačů obrazu v milimetrech ukazuje obrázek. Standardní poměr stran snímačů je 4 : 3, jsou však i kamery se snímačem obrazu nestandardní velikosti.



Obrázek 12 Nejčastěji používané rozměry snímačů pro snímání obrazu [9]

Počet světlocitlivých buněk se udává v megapixelech (1 Mpx = 1 048 576 pixelů). Rozlišovací schopnost se u snímače pro černobílý obraz určí jako dvojnásobek rozměru světlocitlivé buňky. U snímače pro barevný obraz je potřeba více světlocitlivých buněk na jeden obrazový bod. (obrázek 13) Navíc, protože jsou buňky pro jednotlivé barvy na čipu vzájemně posunuty, jsou proti sobě posunuty i jednotlivé barevné složky digitalizovaného obrazu. Pro úlohy, kde je posun barevných složek zásadním problémem, existují kamery s optickou korekcí geometrického posunu barevných složek obrazu.



Obrázek 13 Znáornění pixelu a rozlišení na černobílém a barevném snímači

2.4.2 Elektrické obvody kamery

Elektronika kamery musí uživateli umožnit nastavit kameru tak, aby množství náboje akumulovaného ve světlocitlivých buňkách optimálně reprezentovalo informaci představenou dvourozměrným jasovým obrazem snímaného objektu. Inspekční software obvykle umožňuje:

- nastavit celkový jas (offset)
- nastavit kontrast či zesílení (gain)
- nastavit závěrku - dobu akumulace náboje (shutter)

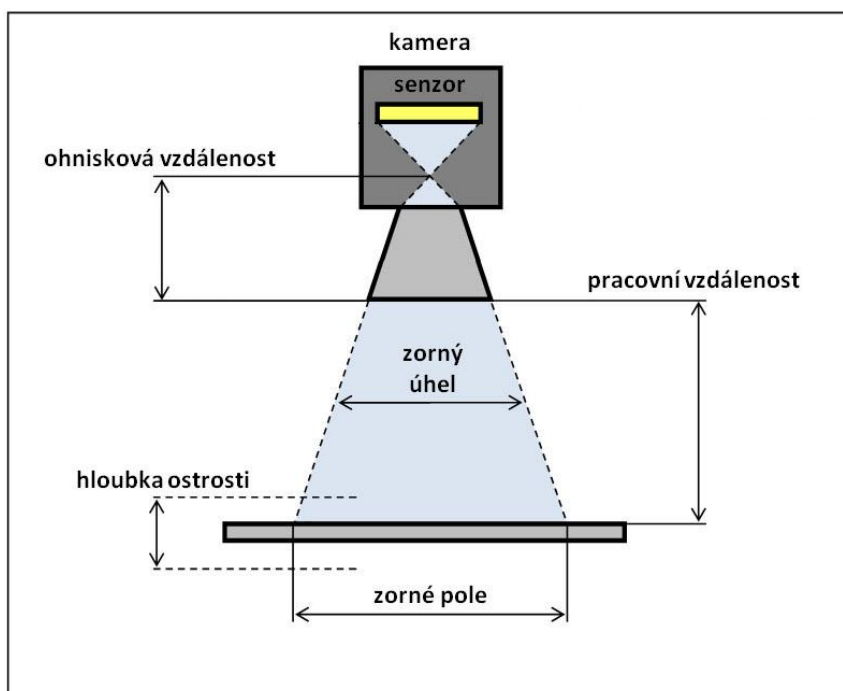
Kamera určená pro strojové vidění musí poskytovat možnost synchronizovat sejmутí snímku s časováním výrobní operace. Proto bývá ve spolupráci s inspekčním programem vybavena těmito několika režimy spouštění:

- volné pořizování řady snímků frekvencí danou dobou, za kterou je možné snímek pořídít a zobrazít v okně inspekčního programu - to se využívá při operacích zacílení kamery a ostření
- řada snímků pořizovaná v pravidelných intervalech s nastavitelnou frekvencí
- jednotlivě pořizované snímky na základě povelu přijatého komunikačním rozhraním nebo digitálním vstupem.

2.4.3 Optická soustava

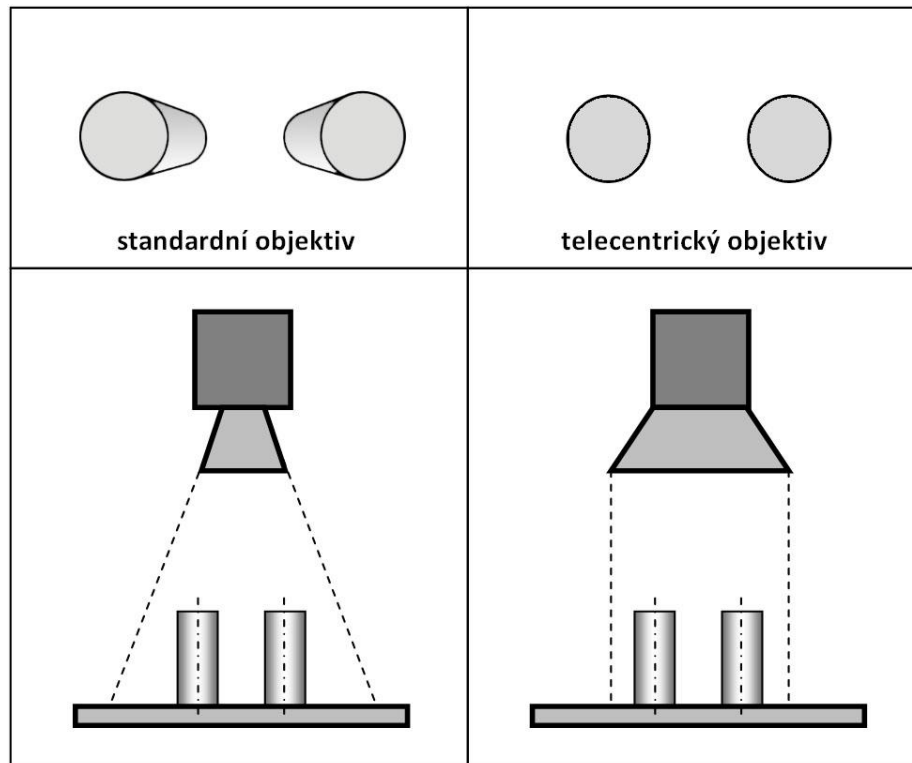
Optická soustava, kterou je ve většině případů kamerový objektiv a osvětlení, má za úkol vytvořit na snímači obrazu takový dvojrozměrný obraz třírozměrné skutečnosti, který obsahuje vyhodnotitelnou informaci potřebnou pro vyřešení úlohy strojového vidění. Optická soustava vytváří obraz, který by měl splňovat tyto parametry:

- dostatečné rozlišení
- vhodný kontrast
- dostatečnou hloubku ostrosti
- přijatelné geometrické zkreslení
- přijatelné perspektivní zkreslení



Obrázek 14 Snímání obrazu kamerou

Tyto parametry jsou ovlivňovány zaostřením a zacloněním objektivu. Vždy však platí, že na objektivy používané pro strojové vidění jsou kladeny mnohem větší požadavky než na objektivy používané v běžných kamerách. Týká se to hlavně rozlišení a kontrastu, které jsou v méně kvalitních objektivěch snižovány vinětací a nepřiměřenou vadou barvy nebo geometrického zkreslení způsobeného aberacemi. Pro strojové vidění jsou nevhodné také objektivy s proměnnou ohniskovou vzdáleností u nichž je již z konstrukčních důvodů nemožné dosáhnout stability parametrů v celém rozsahu nastavení ohniskové vzdálenosti. Zvláštní postavení mezi chybami optické soustavy má perspektivní zkreslení obrazu. Vždy se projeví při promítání standardním objektivem. Perspektivní zkreslení nemá tzv. **telecentrický** objektiv. Charakteristickou vlastností telecentrického objektivu je, že průměr jeho vstupní čočky je stejný jako úhlopříčka zorného pole. (obrázek 15) Pro měření ve větším zorném poli je zapotřebí rozměrný a nákladný objektiv se vstupní čočkou velkého průměru.



Obrázek 15 Zobrazení standardním a telecentrickým objektivem

2.4.4 Mechanické části kamery

Pouzdro a mechanické provedení kamery jsou důležité z hlediska prostředí, ve kterém bude kamera pracovat. Odolnost pouzdra kamery se udává IP kódem, který je definován v normě ČSN EN 60529. První číslice udává odolnost proti vniknutí cizích předmětů (prachu) a druhá udává odolnost proti vniknutí vody.

2.5 Osvětlovače

Podle [10]: Účelem osvětlení ve strojovém vidění je dosáhnout maximálního kontrastu těch částí objektu, které nás zajímají (zájmové části), oproti částem, které nejsou předmětem inspekce (rušivé části). Také je třeba minimalizovat vliv změn v okolí na reprezentaci objektu jasovým obrazem na snímači obrazu, aby inspekční program vykazoval opakovatelné výsledky. Kontrast zájmové a rušivé části objektu lze vytvářet v zásadě dvojnásobem:

- využitím jejich rozdílné absorpce (emise)
- využitím rozdílu jejich jasů, který vznikne vhodným směrovým osvětlením.

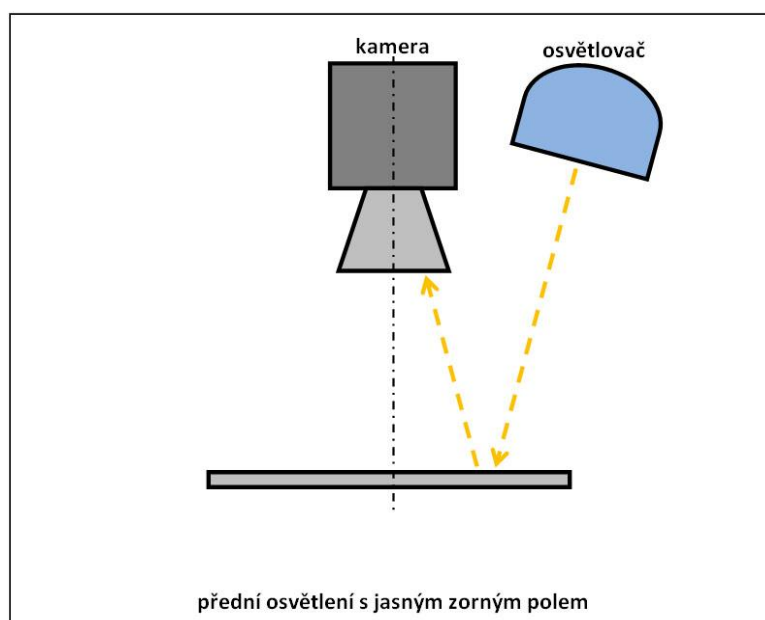
V obou případech začíná návrh optimální osvětlovací soustavy analýzou interakce objektu a světla. Na základě výsledků této analýzy je možné stanovit, jak a čím objekt osvětlit a jak eliminovat případné rušivé vlivy. Světlo, které dopadne z osvětlovače na sledovaný objekt, se podle zákonů optiky rozdělí na čtyři složky podle toho, zda se odrazí, způsobí v materiálu objektu emisi sekundárního záření, absorbují se, nebo projdou objektem. Ve strojovém vidění může být využita každá z těchto složek. Interakce světla s objektem závisí na barvě objektu a na barvě (vlnové délce) světla. Na vlnové délce světla závisí i množství akumulovaného náboje na snímači kamery. Ve strojovém vidění se nejčastěji používá kombinace černobílé kamery a osvětlovače s červeným světlem. Tato kombinace je výhodná z hlediska spektrální citlivosti většiny kamer.

2.5.1 Geometrie osvětlení

Geometrie osvětlení je dána složením svazku světla. Jsou-li paprsky světla rovnoběžné, jedná se o kolimované světlo. Jsou-li paprsky světla plně rozptýlené, jde o difuzní světlo. Složení svazku světla reálného osvětlovače svými vlastnostmi leží mezi těmito dvěma extrémy. Obecně platí, že rozptýlené světlo zmenšuje kontrasty způsobené členitostí objektu a strukturou povrchu a zvýrazňuje kontrasty způsobené absorpcí. Naopak, je-li třeba kontrastně zobrazit členění objektu, volí se světlo směřované. Pro různé způsoby vytvoření kontrastu zájmové a rušivé části objektu zavedli výrobci osvětlovačů pro používané geometrie osvětlení toto názvosloví:

2.5.1.1 Přední osvětlení s jasným zorným polem (*bright-field lighting*)

Osvětlovač osvětluje objekt tak, aby světlo od něj odražené dopadlo přes objektiv na obrazový snímač. (obrázek 16) Osvětlení se využívá ve spojení s rozptýleným světlem pro vytvoření kontrastů na základě rozdílné absorpce. Méně často se při předním osvětlení s jasným obrazovým polem používá i směřované světlo pro vytvoření kontrastu lesklých ploch.



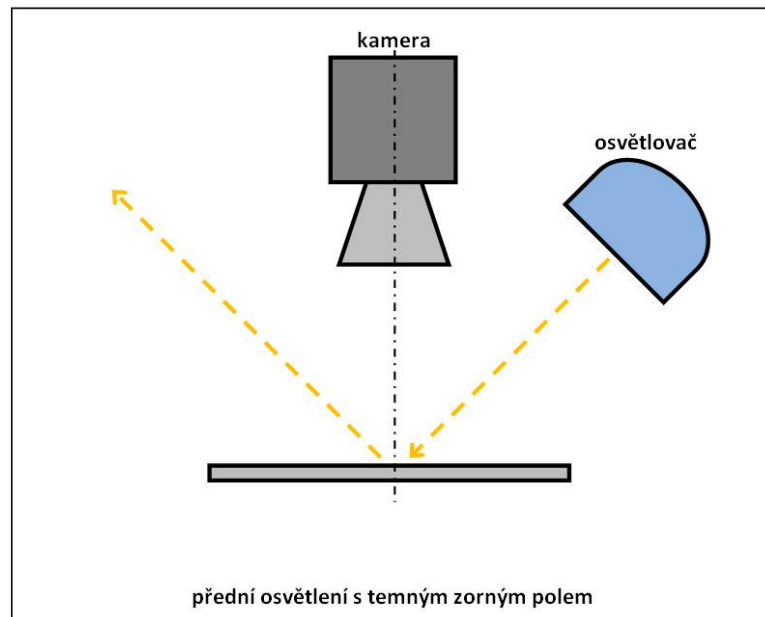
Obrázek 16 Schéma předního osvětlení s jasným zorným polem

2.5.1.2 Přední osvětlení s tmavým zorným polem (*dark-field lighting*)

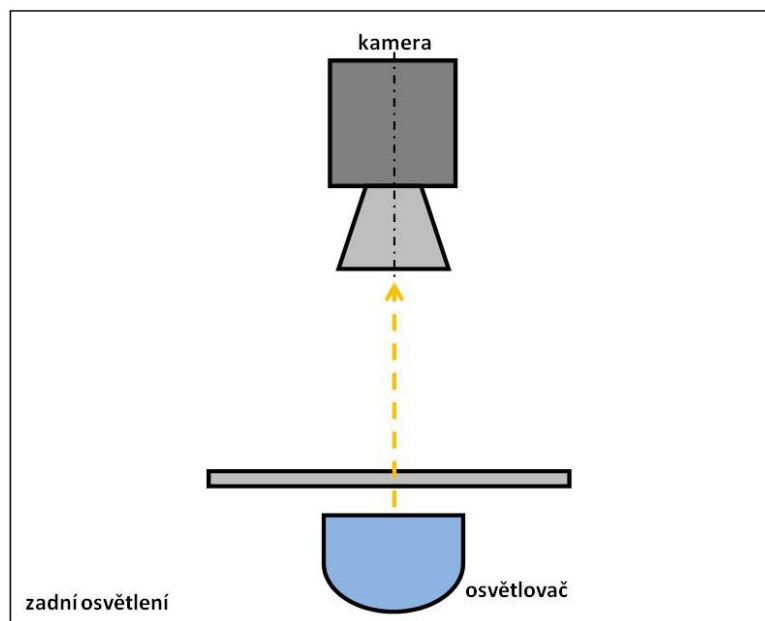
Využívá světlo směřované tak, aby členité části objektu odrážely světlo objektivem na snímač obrazu, zatímco povrch objektu odráží světlo mimo. (obrázek 17) Tím se vytvoří kontrastní obraz členitosti povrchu v tmavém poli (z této vlastnosti je odvozen i název).

2.5.1.3 Zadní osvětlení (*back lighting*)

Používá se k zobrazení obrysu objektu, typicky v úlohách měření rozměrů. (obrázek 18) Tento typ osvětlení je nutný, jestliže má být vytvořen kontrastní obrys objektu za průhlednou lesklou překážkou. Nejčastěji využívá rozptýlené světlo.



Obrázek 17 Schéma předního osvětlení s temným zorným polem



Obrázek 18 Schéma zadního osvětlení

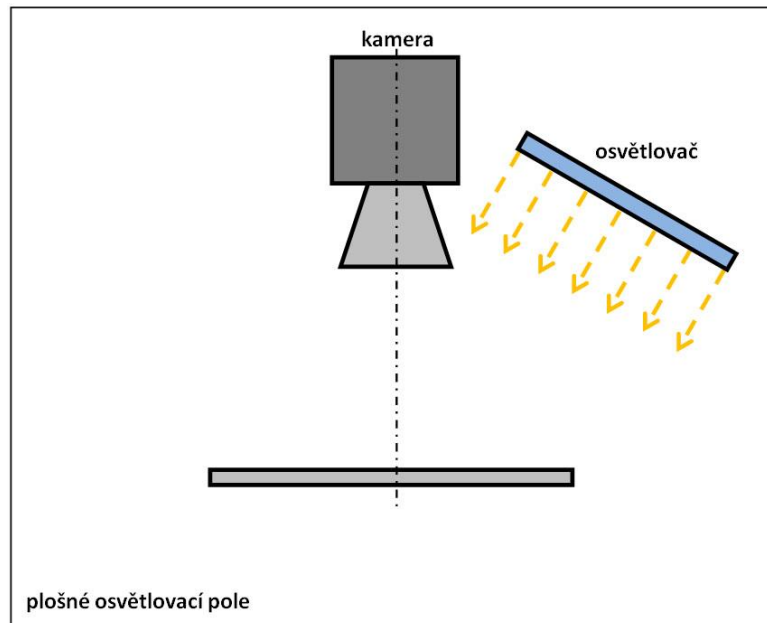
2.5.2 Druhy osvětlovačů

Téměř výhradně se používají osvětlovače s diodami LED, protože nepotřebují vysoké napětí a jejich světelný tok se snadno reguluje. Vykazují dlouhou životnost a vlastnosti jejich světla se mění jen málo. Většina výrobců dodává základní druhy osvětlovačů, které pokrývají používané geometrie osvětlení. Osvětlovače se dodávají v různých velikostech, s různou barvou světla a různým způsobem řízení (nepřetržité světlo, záblesky nebo stroboskopické osvětlení).

2.5.2.1 Plošná osvětlovací pole

Plošná osvětlovací pole většinou poskytují směrované světlo. (obrázek 19) Rozptýlení je dáno vyzařovacím úhlem použitých diod LED. V nenáročných úlohách mohou nahradit i nákladnější

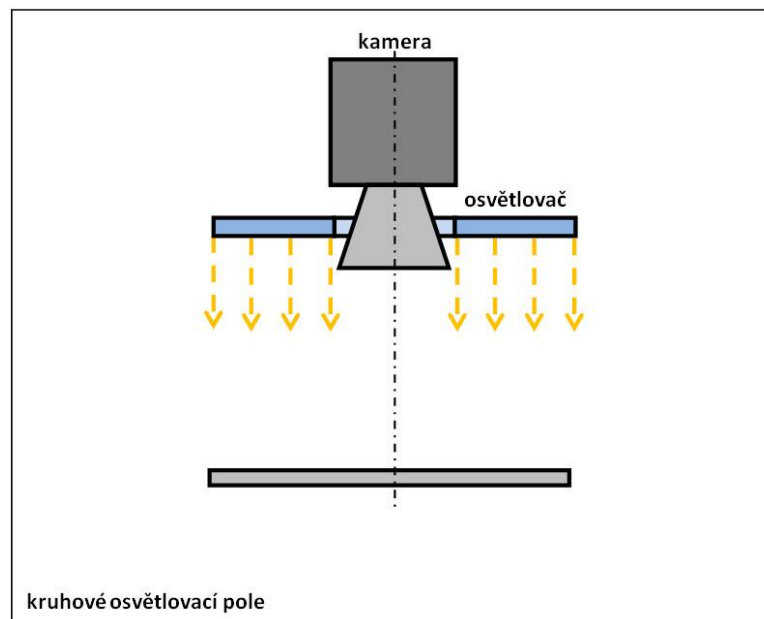
osvětlovače pro osvětlení s jasným obrazovým polem a rozptýleným světlem. Často se používají jako zdroje směrovaného světla pro osvětlení s temným zorným polem. Malá a speciální pole se využívají i jako pomocná světla nebo jako zdroje pro složitější osvětlovací soustavy.



Obrázek 19 Schéma plošného osvětlovacího pole

2.5.2.2 Kruhové osvětlovače

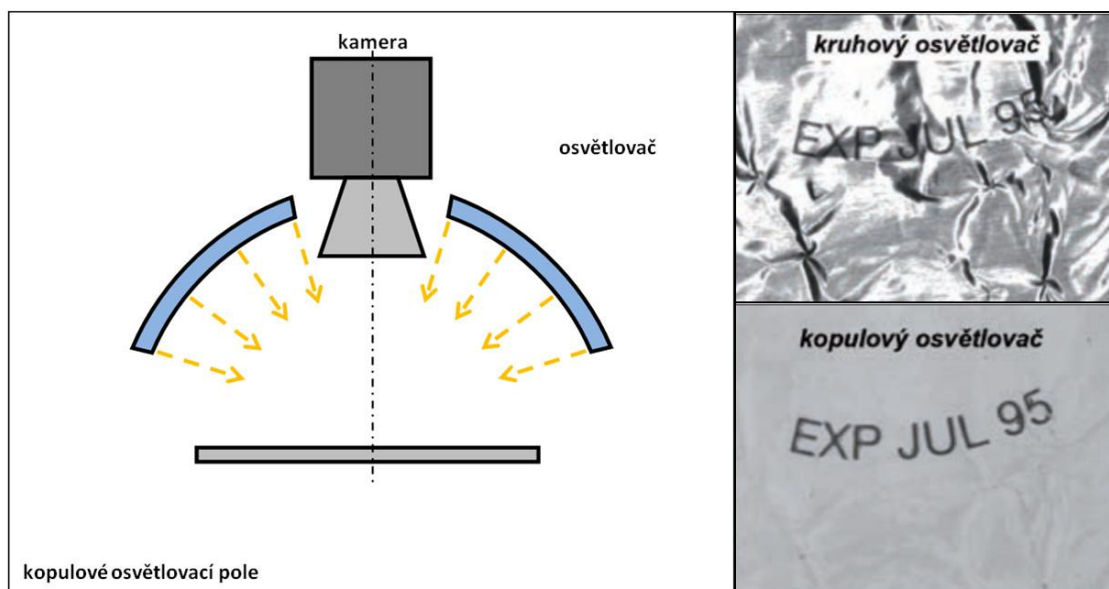
Kruhový osvětlovač poskytuje více rozptýlené světlo směřující od objektivu kamery. (obrázek 20) V principu jde o osvětlovač pro osvětlení s jasným obrazovým polem a rozptýleným světlem, nemá ovšem zcela ideální vlastnosti, zvláště v provedení jednoduchého kruhu LED, který výrobci někdy přidávají okolo objektivu kamery.



Obrázek 20 Schéma kruhového osvětlovacího pole

2.5.2.3 Kopulové osvětlovače

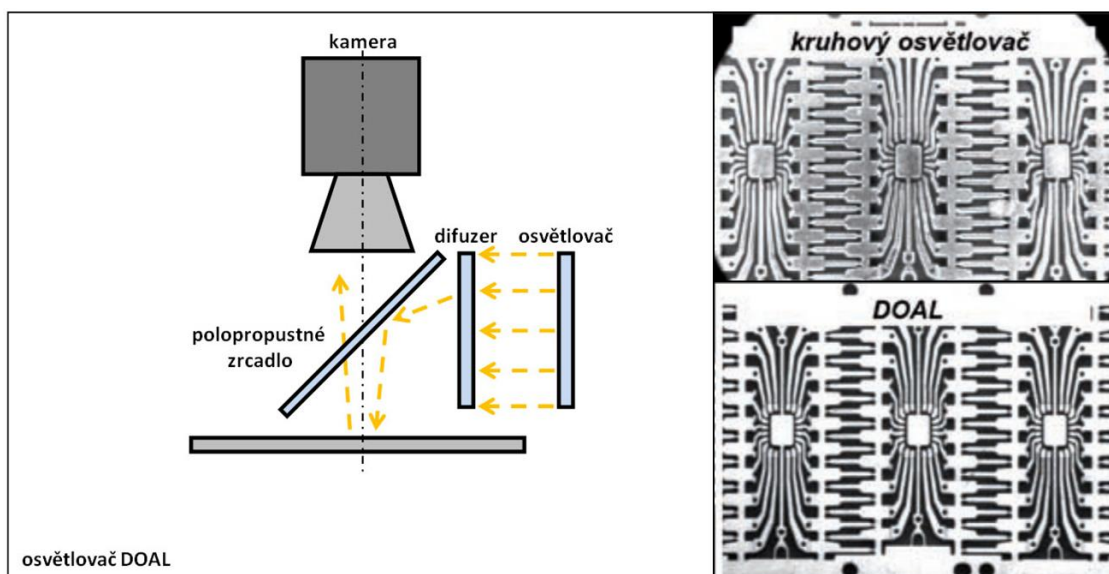
Kopulové osvětlovače se používají pro osvětlení s jasným obrazovým polem, kdy je nezbytně nutné osvětlit objekt světlem skutečně rozptýleným. (obrázek 21) To je například úloha čtení tištěných nápisů na zmačkané lesklé fólii. Rozptýlené světlo se v osvětlovači vytváří difuzním odrazem na vnitřní ploše polokoule. Osvětlení se někdy říká bezestínové (shadowless).



Obrázek 21 vpravo schéma kopulového osvětlovacího pole vlevo příklad použití (převzato z [10])

2.5.2.4 Osvětlovače DOAL

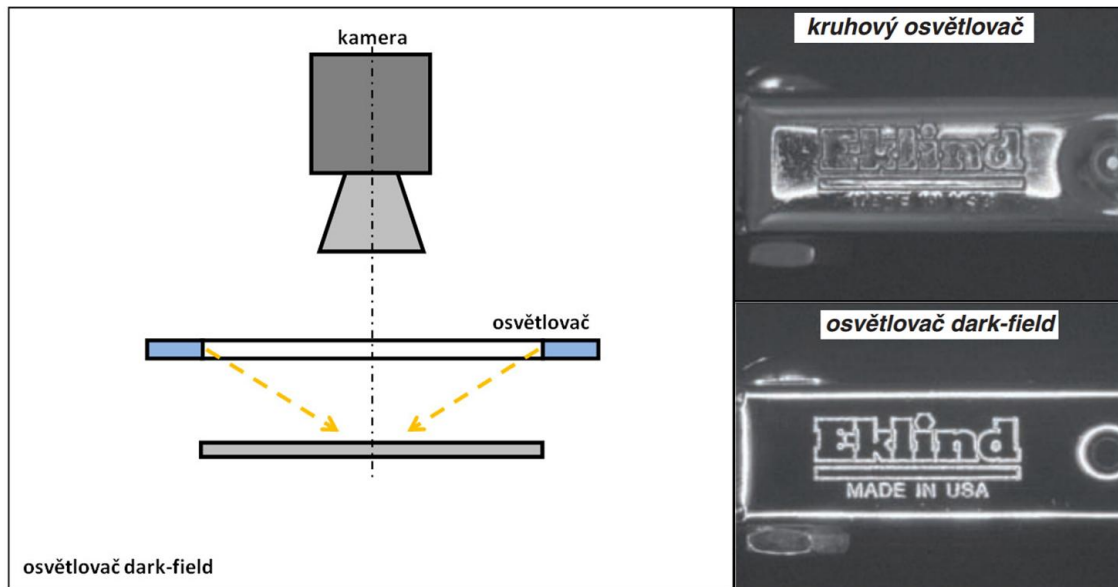
Osvětlovač DOAL (Diffused ON Axis Lighting) poskytuje specifické osvětlení s jasným obrazovým polem a rozptýleným světlem. Použitý difuzor zaručuje lepší rozptýlení světla. Díky konstrukci s polopropustným zrcadlem světlo dopadá rovnoměrně z celé plochy nad objektem, tedy i z prostoru před objektivem kamery. (obrázek 22) Nevýhodou je vyšší cena a omezené zorné pole.



Obrázek 22 vpravo schéma osvětlovače DOAL a vlevo příklad použití [10]

2.5.2.5 Osvětlovače dark-field

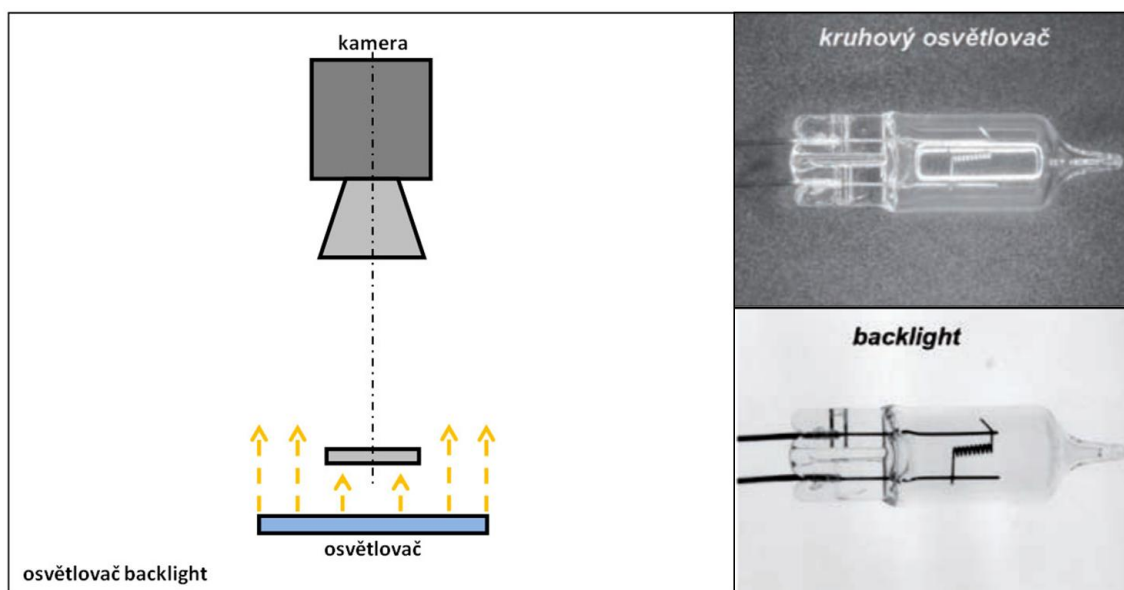
Z názvu vyplývá, že poskytuje přední osvětlení s tmným zorným polem. Realizuje je jako světlo směřované v rovině téměř kolmé k ose objektivu kamery. Typickým příkladem využití osvětlovače dark-field je zdůraznění gravírovaných nebo laserem vypálených značek či nápisů. Osvětlovač dark-field používaný pro takové účely je obvykle tvořen kruhem diod LED s úzkou vyzařovací charakteristikou, které svítí téměř v rovině kruhu do jeho středu.



Obrázek 23 vpravo schéma osvětlovače dark-field a vlevo příklad použití [10]

2.5.2.6 Osvětlovače backlight

Zadní rozptýlené světlo neboli backlight je nejčastěji používaný typ zadního osvětlení. Je vhodné především při měřeních, kde je třeba zobrazit obrys měřené součásti. Dalším typickým příkladem je úloha rozpoznání nebo nalezení polohy součásti v průhledném pouzdru – kontrola vlákna žárovky. (obrázek 24) Osvětlovač backlight je konstruován jako pole diod LED s předřazeným velkoplošným rozptylovačem (difuzorem). Měření obrysu vyžaduje zcela homogenní zadní světlo.



Obrázek 24 vpravo schéma osvětlovače backlight a vlevo příklad použití [10]

2.5.3 Řízení osvětlovačů

Je-li požadováno nepřetržité osvětlení, není třeba osvětlovač nijak řídit. Pokud se světlo zapíná jen v okamžiku expozice, je nutné využít řídicí jednotku, kterou obvykle dodává výrobce osvětlovačů. Téměř všechny systémy strojového vidění mají digitální výstup umožňující synchronizaci světla se snímkem, který lze přímo propojit s řídicí jednotkou osvětlovače. Někdy je výhodné využít i možnost řídit výkon (světelného toku) osvětlovače. Pak je možné eliminovat případné snížení světelného toku způsobené např. stárnutím světelného zdroje zvýšením příkonu.

2.5.4 Eliminace rušivých vlivů

Pokud by byl jediným zdrojem světla správně zvolený osvětlovač, bylo by snadné pro každý snímek docílit stále stejného jasového obrazu na snímači obrazu. Výsledky inspekce by pak byly opakovatelné. V prostředí průmyslového provozu však téměř vždy existují i jiné zdroje světla. Patří sem v první řadě denní světlo a umělé osvětlení provozu. Parametry těchto zdrojů jsou v podstatě nepředvídatelné; závisí na denní a roční době, na počasí a mnoha dalších faktorech. Nejjednodušším řešením je tyto nežádoucí zdroje světla odstínit. Je-li to technicky nemožné, použije se osvětlovač, který je schopen vytvořit řádově vyšší osvětlení než nežádoucí zdroje. Vliv rušivého světla pak působí jen relativně malé změny v celkovém osvětlení. Nutný světelný tok takového osvětlovače je však většinou velmi vysoký a relativně vysoký je i jeho příkon. Proto se osvětlovač zapíná jen po dobu expozice. Blikání intenzivního světla však často působí nepříjemně na případnou lidskou obsluhu, způsobuje únavu a při dlouhodobém působení i zdravotní potíže. Vysoké osvětlení také často vyžaduje velké zaclonění objektivu, což někdy nemusí být žádoucí. Další příčinou kolísání osvětlení mohou být nečistoty – prach, který se usazuje na aktivní ploše osvětlovače. Je tedy třeba zajistit pravidelné čištění osvětlovače. Ve skutečně prašných provozech je dokonce třeba osvětlovač čistit průběžně ofukováním stlačeným vzduchem. Při nevhodné konstrukci zapouzdřeného osvětlovače se prach může dostat dovnitř a usadit se např. na rozptylovačích, zrcadlech a dalších optických prvcích. Snížení svítivosti nemusí být okem registrovatelné, a přesto může způsobit nevysvětlitelné zvýšení chyby měření.

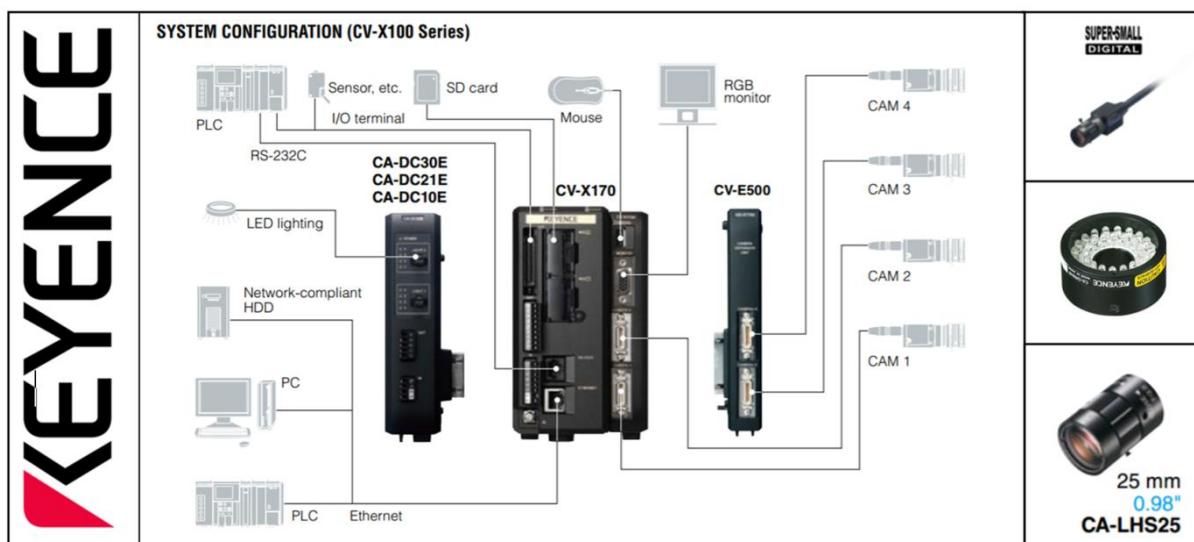
2.6 Přehled výrobců systémů strojového vidění

Tímto přehledem mají být zmapováni někteří výrobci nabízející systémy strojového vidění a uvést příklad komerčně dostupného systému použitelného jako alternativa pro úlohu této bakalářské práce.

Mezi výrobce patří například firmy Turck, Baumer, Ximea, AccuSentry, Basler, National Instruments, Banner, Pilz, Cognex, nebo Keyence.

2.6.1 Komerční systém

Možný komerční systém pro úlohu této bakalářské práce by mohla být například sestava od firmy KEYENCE skládající se z ovládací jednotky CV-X150, barevné kamery CV-S200C s rozlišením 1600x1200 pixelů, objektivem CA-LHS25 s ohniskovou vzdáleností 25 mm a kruhovým LED osvětlovačem CA-DRW4F. Ovládací jednotka je podle obrázku 27 snadno propojitelná nejen s dalšími částmi sestavy, ale i s monitorem (VGA) a myší (USB) pro tvorbu inspekčního programu a nebo PLC (Ethernet). Navíc s PLC od firmy SIEMENS, jak je tomu i v případě této práce, je spárování velmi snadné.



Obrázek 25 vlevo schéma propojení systému řady CV-X100 od firmy KEYENCE a vpravo shora kamera CV-S200C, osvětlovač CA-DRW4F a objektiv CA-LHS25 [11]

2.7 Charakteristika zařízení použitého v naší úloze

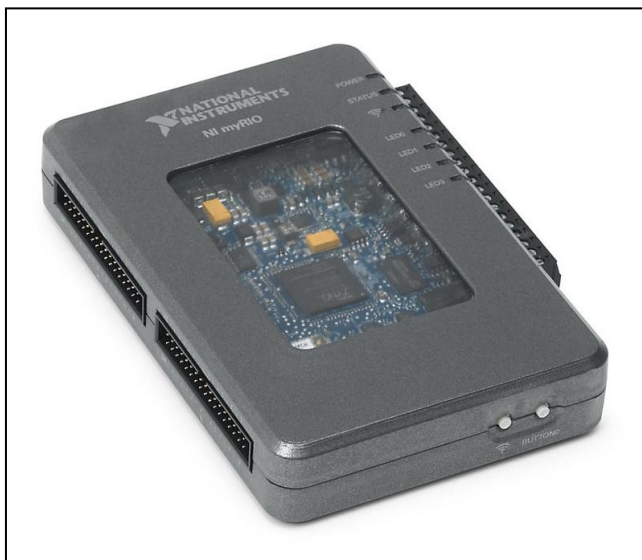
Pro naši úlohu použijeme kameru Trust webcam 1.3, vývojové prostředí LabVIEW a vestavné zařízení NI myRIO. Půjde tedy o třetí druh systému strojového vidění, neboli PC vision.

2.7.1 Vestavné zařízení NI myRIO

Vestavné zařízení NI myRIO je přímo určené na podobné aplikace pro studenty. Obsahuje nejnovější SoC (System on a Chip) Zynq® od společnosti Xilinx s dvoujádrovým procesorem ARM Cortex-A9 a programovatelným obvodem FPGA s 28 000 programovatelných logických bloků, který lze programovat pomocí grafického vývojového prostředí LabVIEW. Výukové materiály jsou k němu dodávány bezplatně. Pro naše účely je důležitá možnost připojení přes USB a WiFi. (obrázek 26)

2.7.2 Kamera Trust 1.3MPx

Jde o kameru určenou pro běžné video hovory. Nemá příliš velké rozlišení (1280x1024), ale pro danou úlohu to bude dostatečné, protože nejde o měření přesných rozměrů, pouze pozici a tvar dílců. Kamera je barevná, což umožní i rozpoznání barev sledovaných objektů. Kolem objektivu je klasické vestavné kruhové osvětlení z LED diod, jehož intenzita osvětlení jde regulovat. Výstupní port je USB typu A.



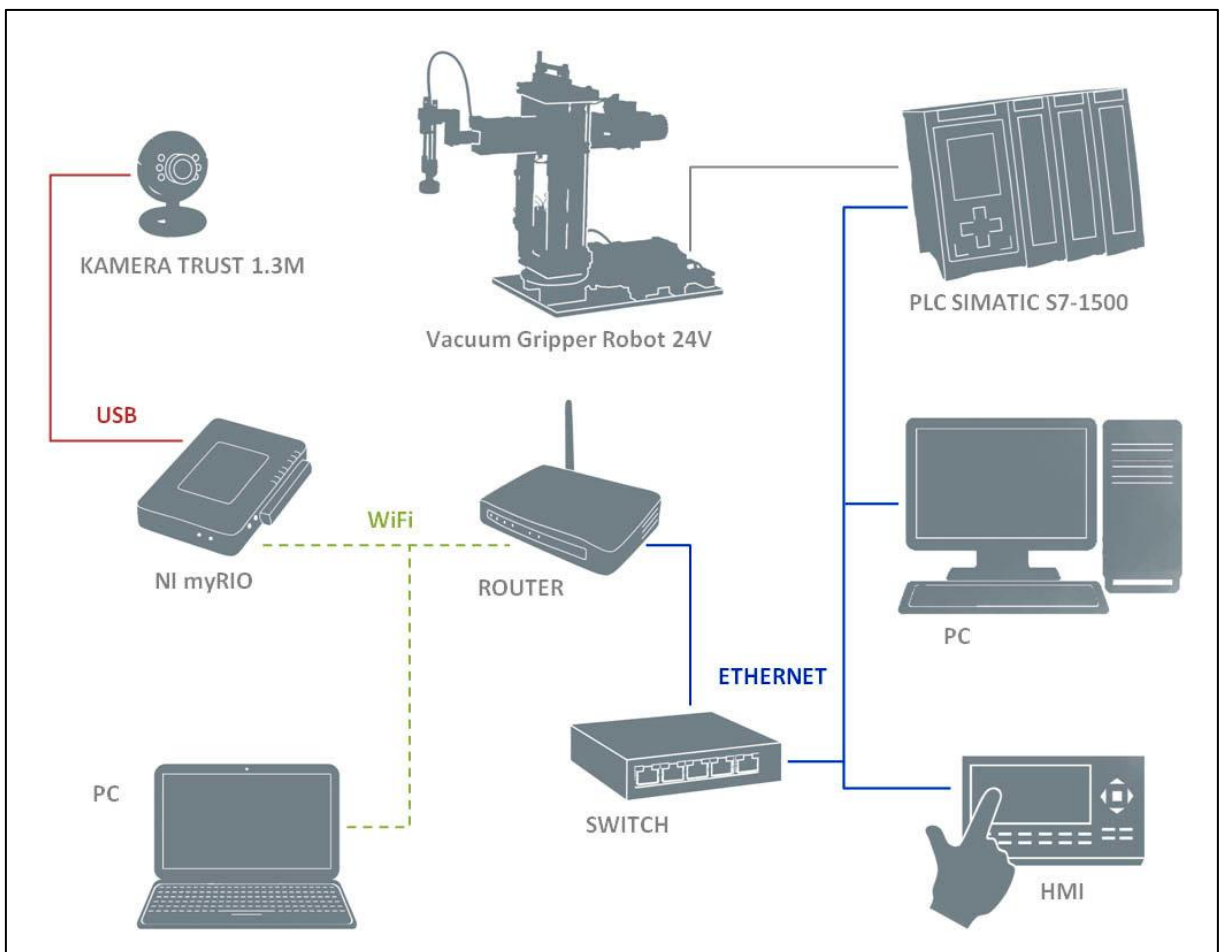
Obrázek 26 Vestavné zařízení NI myRIO [4]



Obrázek 27 Kamera Trust 1.3 MPx [15]

2.8 Přehled komunikačních rozhraní a protokolů

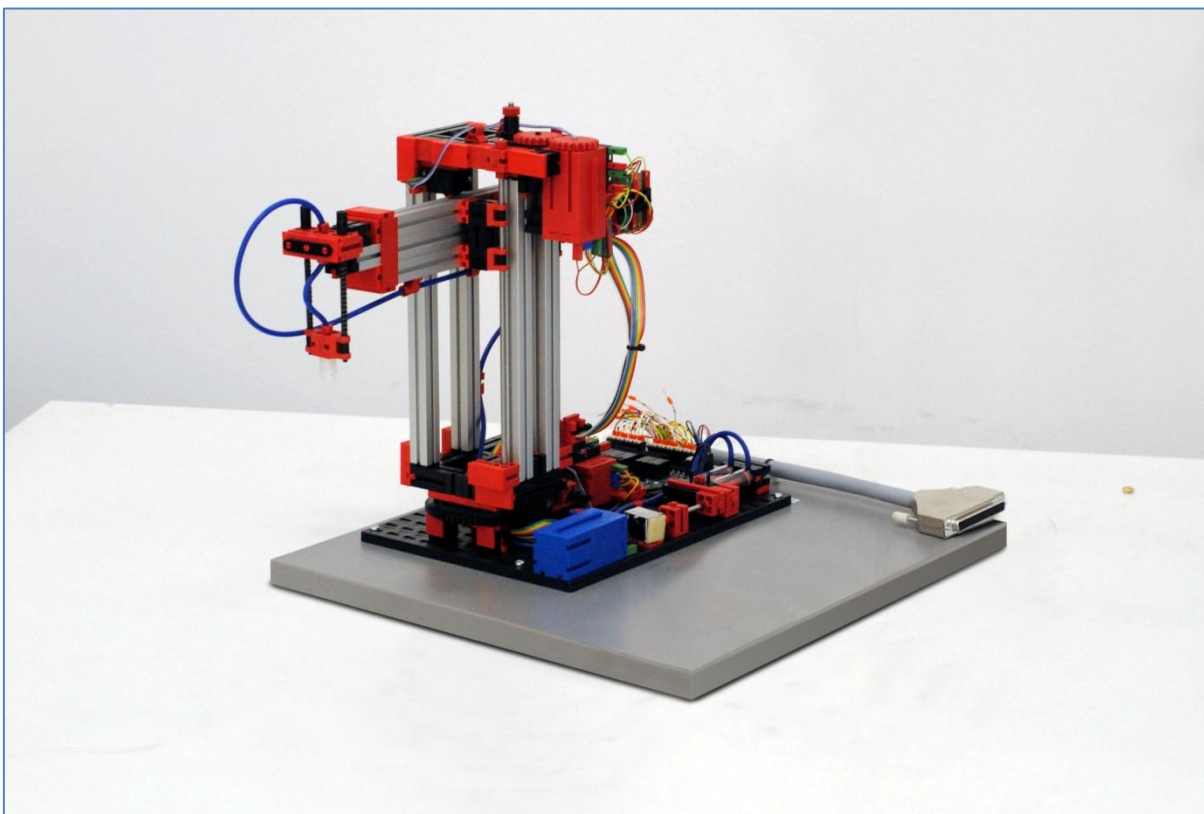
Pro vytvoření funkčního řešení strojového vidění je také potřeba správně propojit jednotlivé části systému tak, aby spolu mohly komunikovat. V případě této práce je nejprve potřeba propojit kameru se zařízením na zpracování obrazu, tedy NI myRIO. (obrázek 28) Toto propojení bude realizováno portem USB 2.1 typu A (Universal Serial Bus), který je výstupem kamery a umožňuje rychlé propojení s počítačem díky technologii plug&play. Informace získané z vyhodnocení obrazu je potřeba předat okolí, což je v tomto případě PLC SIMATIC S7-1500 od firmy SIEMENS. Komunikace s okolím probíhá prostřednictvím bezdrátového připojení wifi. Aby bylo možné propojit PLC s NI my RIO je zapotřebí routeru, který převede bezdrátový signál wifi na sběrnici metalického (může být i optické) vedení ethernetu v daném protokolu (způsob posílání dat), v tomto případě jde o průmyslový automatizační protokol profinet. Mezi router a PLC je ještě vřazen switch (přepínač), který umožňuje propojit více vedení ethernetu a na základě MAC adres (Media Access Control) rozesílat pakety (přenášené bloky dat) příslušným zařízením.



Obrázek 28 Schéma propojení jednotlivých zařízení

3 Vlastní řešení

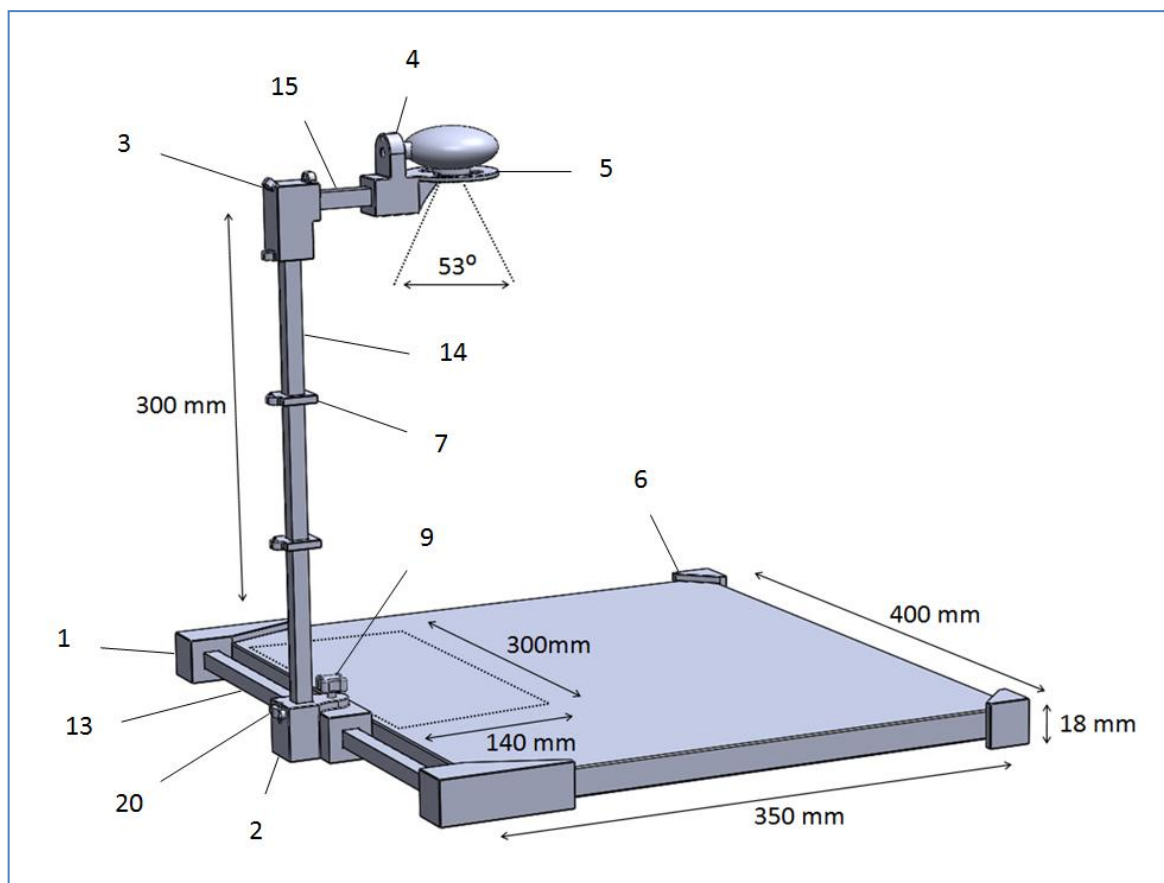
Po vypracování řešerše k vytvoření potřebného přehledu o dané problematice a zmapování výchozích prostředků, kterými jsou použita zařízení, lze přejít k samotnému řešení. Řešení se skládá z návrhu a výroby stojanu na kameru, zajišťující její vhodnou polohu vůči manipulačnímu robotovi, návrhu a výroby dílců detekovaných systémem, vytvoření programu vyhodnocující obraz z kamery a propojení těchto částí do funkčního celku systému strojového vidění. Na obrázku 29 je výchozí stav laboratorního robota upevněného na desce. Rozměry robota a desky jsou směrodatné pro další postup řešení.



Obrázek 29 Původní stav laboratorního robota

3.1 Stojan na kameru

Základní rozměry stojanu byly určeny tak, aby byl snadno a přesně připevnitelný k desce laboratorního robota, nijak nepřekážel v pohybu robota či nedošlo ke kolizi s jeho pohyblivými částmi a jeho výška umožňovala kameře snímat celou požadovanou oblast. Stojan se skládá z plastových dílců s drážkovanými otvory, do nichž se zasunují čtvercové duté hliníkové profily, jimiž jsou jednotlivé části pospojovány. Díly byly navrženy v programu SOLIDWORKS a vytisknuty na 3D tiskárně. Díl pro uchycení kamery má zajistit stálou polohu objektivu, jinak by mohlo dojít k posunu, což by vneslo chyby do odměřování polohy detekovaných dílců. Držák objektivu má ještě postranní otvory kvůli diodám umístěným po obvodu objektivu. Některé dílce jsou opatřeny úchytkami, do kterých se vmáčkne kabel vycházející z kamery. Poslední úchyt je pak tvořen drážkou a přišroubovatelným můstkem, takže nedojde k vyvléknutí kabelu. Posuvný díl umístěný na vodícím profilu je opatřen stahovacím šroubem pro aretaci polohy. Pro zachování vodorovné polohy jsou na dva protilehlé rohy desky nasazeny rohové dílce sloužící jako nožičky.



Obrázek 30 3D model stojanu se základními rozměry a pozicemi

Č.	Název 3D modelu	popis	počet
1	DÍL 1	Patka pro uchycení k desce	2
2	DÍL 2	Posuvný díl s aretací a pevným uchycením kabelu	1
3	DÍL 3	Pravoúhlý spojovací díl s přichytkami na kabel	1
4	DÍL 4	První část koncového dílu k připevnění kamery	1
5	DÍL 5	Druhá část koncového dílu k připevnění kamery	1
6	DÍL 6	Nožičky na protilehlé rohy desky	2
7	DÍL 7	Posuvné přichytky kabelu	2
8	DÍL 8	Podložka pod aretační šroub doléhající na vodící profil	1
9	DÍL 9	Nástavec na hlavu aretačního šroubu	1
10	DÍL 10	Dílec s čtvercovým půdorysem	3
11	DÍL 11	Dílec s kruhovým půdorysem	3
12	DÍL 12	Dílec s šestiúhelníkovým půdorysem	3
13		Čtvercový hliníkový profil 10x10x1 mm délky 400 mm	1
14		Čtvercový hliníkový profil 10x10x1 mm délky 320 mm	1
15		Čtvercový hliníkový profil 10x10x1 mm délky 74 mm	1
16		Šroub se šestihrannou hlavou M5x16 ISO 4017 pro aretaci	1
17		Šestihranná matice M5 ISO 4032 pro aretaci	1
18		Šroub M4x16 DIN7985 pro spojení dílů 4 a 5	1
19		Šestihranná matice M4 ISO 4032 pro spojení dílů 4 a 5	1
20		Kabelový držák plastový 20x5,5 mm	1

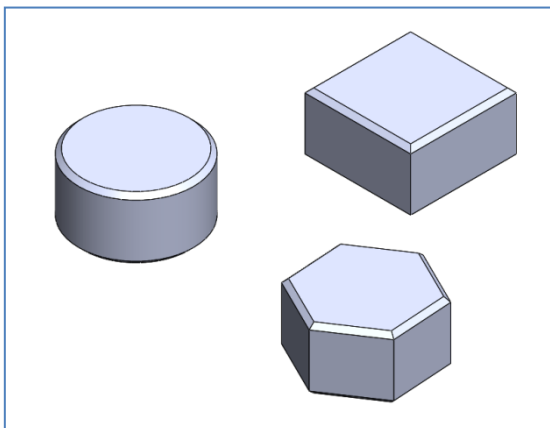
Tabulka 1 Kusovník

3.2 Dílce

Požadavkem na systém strojového vidění ze zadání bakalářské práce je, aby rozeznal tři tvary a tři barvy dílců, proto byli vytvořeny 3D modely tří tvarů (čtverec, kruh, šestiúhelník), na nichž se systém otestuje. Každý byl vytisknut na 3D tiskárně ve třech barevných variantách (červená, modrá, bílá). Při návrhu dílců bylo pouze potřeba zajistit, aby jejich horní plocha byla dostatečně velká pro uchycení podtlakovým úchopem manipulačního robota, který má průměr 12 mm.



Obrázek 31 Dílce vytištěné na 3D tiskárně



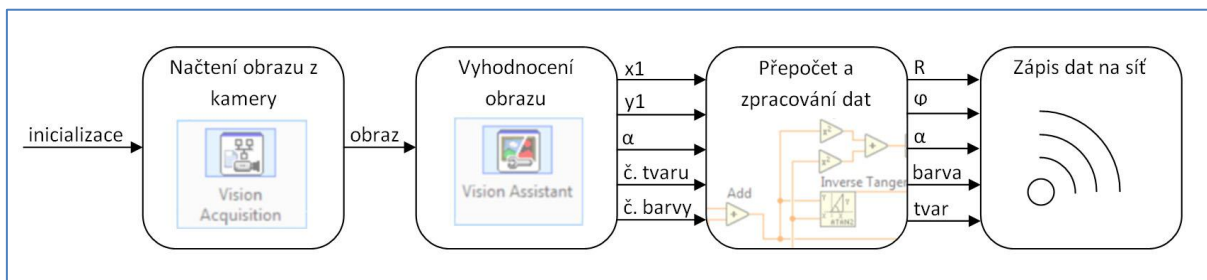
Obrázek 32 3D modely dílců

4 Vytvoření inspekčního programu v prostředí LabVIEW

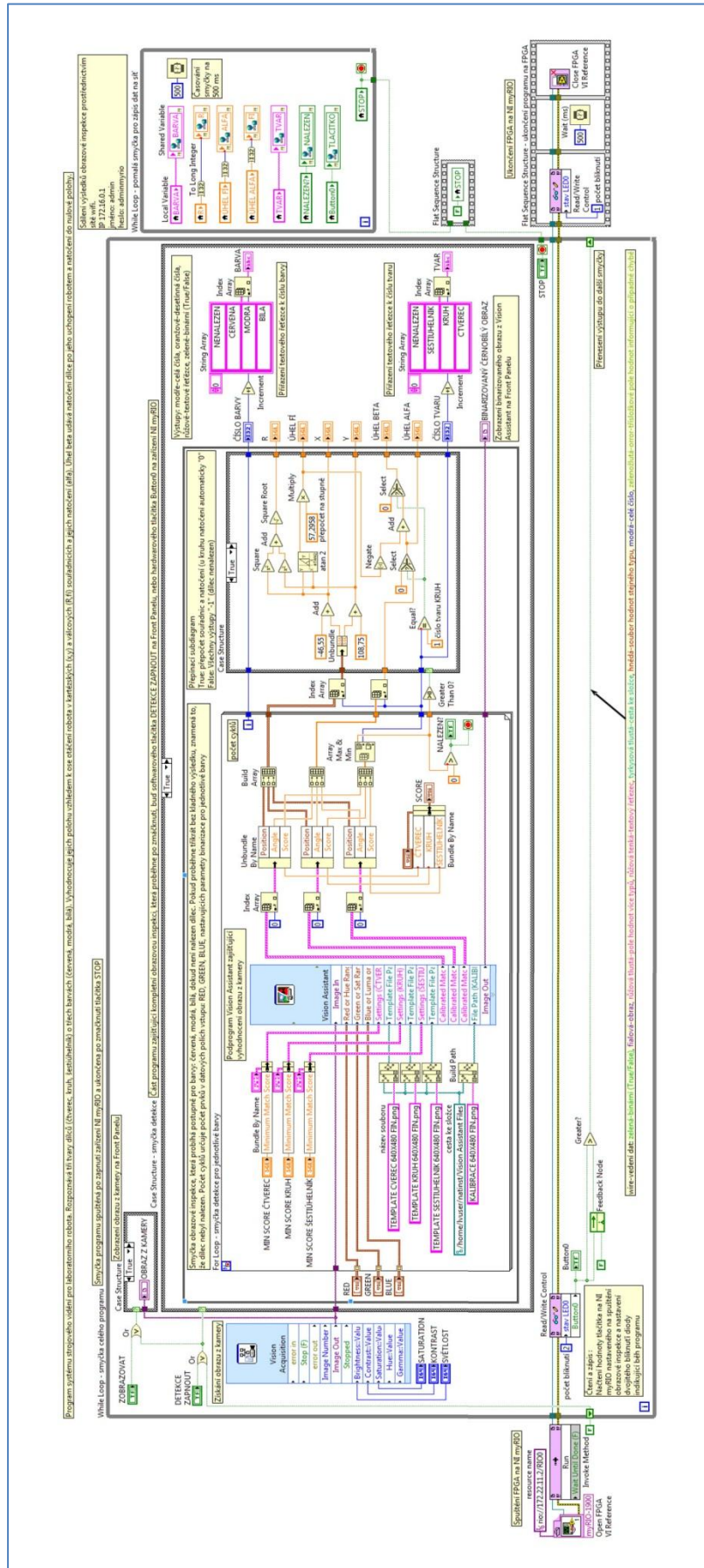
Inspekční program pro systém strojového vidění, který byl vytvořen v rámci této bakalářské práce byl sestaven v prostředí LabVIEW od společnosti National Instruments. Protože jde o systém, který má obohatit výuku, je vhodné vytvořit i návod na jeho tvorbu, aby každý student byl schopen jeho rekonstrukce na základě které by mohl pochopit princip fungování celého systému.

4.1 Požadavky na program

Požaduje se rozpoznání tří tvarů, tří barev, polohy a orientace snímaných dílců. Program je spouštěn buď v rozhraní **LabVIEW** přes **Front Panel**, sloužící jako kontrolní okno, kde je možné vytvořit virtuální tlačítko, nebo hardwarovým tlačítkem na zařízení **NI myRIO**. Výstupy programu jsou souřadnice dílce a to jak souřadnice x a y v kartézském souřadném systému, tak i přepočtené souřadnice do válcového souřadného systému robota R a φ . Natočení dílce udává úhel α . To jsou výstupy číselné. Pak jsou výstupy textové udávající barvu a tvar. Nakonec je dvouhodnotový výstup udávající, zda byl nalezen dílec.



Obrázek 33 Schéma rozvržení programu



Obrázek 34 Celé zmenšené komentované blokové schéma programu

4.2 Rozvržení programu

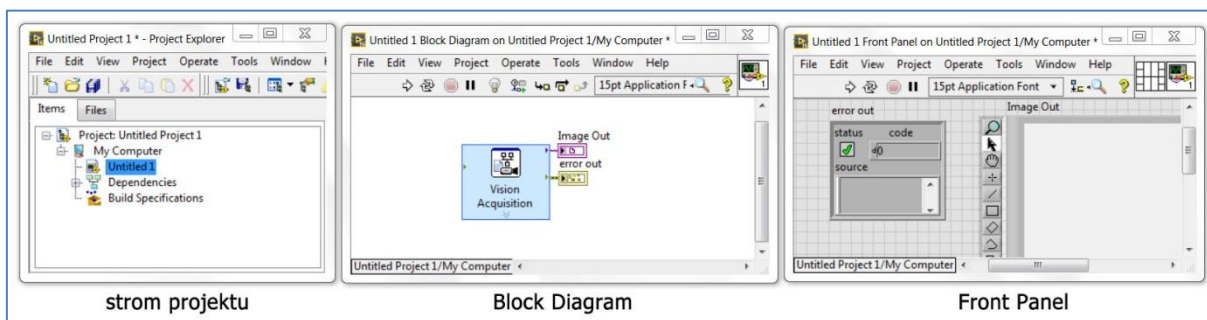
Nejprve je třeba získat obraz z kamery, to zajistí podprogram **Vision Acquisition**. Tento obraz bude zpracován dalším podprogramem **Vision Assistant**. Při zpracování obrazu dochází také k binarizaci, což je přetransformování obrazu na binární tak, aby jednou barvou byly objekty které detekujeme a druhou jejich pozadí. Z důvodů rozložení barev pozadí v histogramu, jak ukazuje obrázek 35, nebylo možné nastavit hodnoty tak, aby zahrnovali všechny barvy dílců a přitom nedošlo i k zvýraznění částí pozadí. Řešením je opakovaně spouštět **Vision Assistant** ve smyčce, kde dochází postupně k vyhodnocení obrazu s binarizací nastavenou vždy pouze pro jednu barvu. Smyčka potom probíhá dokud buď není nalezen dílec, nebo neproběhne třikrát pro všechny barvy. Barva je určena z počtu proběhnutých cyklů. Získaná data jsou v další části programu upravena na potřebné výstupy. Souřadnice jsou přepočteny. Čísla tvarů a barev převedena na textové řetězce. Nakonec jsou výstupy programu sdíleny s okolím prostřednictvím sítě wifi (obrázek 33).



Obrázek 35 Rozložení histogramu pro jednotlivé barvy a pozadí

4.3 Vytvoření projektu

Pro sestavení programu v **LabVIEW** je třeba vytvořit takzvaný projekt, což je soubor obsahující všechny potřebné části programu, jako jsou například jednotlivé podprogramy, obrázky, nastavení atd. Nový projekt vytvoříme tím, že při otevření programu **LabVIEW** klikneme v levé části okna v záložce **Create Project** na **Blank Project**, čímž se otevře strom projektu (obrázek 36). Je třeba kliknout na hlavní složku projektu pravým tlačítkem myši a vybrat **New/Targets and Devices...** Otevře se okno pro výběr nového zařízení, kde pod výběrem **Existing target or device/Discover an existing target(s) or device(s)** ve složce **myRIO** vybereme (je nutno mít zařízení správně připojené) **NI-myRIO-1900...** Takto vznikne nový cíl, kam později nahrajeme inspekční program (kapitola 4.11). Dále pravým tlačítkem myši klikneme ve stromu projektu na záložku **My Computer** a vybereme **New/VI**. Otevřou se dvě okna pro tvorbu nového programu a to **Block Diagram**, kde se vytváří program pomocí vkládání a propojování bloků nástrojů a podprogramů a **Front Panel**, který má funkci kontrolního panelu k vizualizaci nastavení (**controls**) a výsledků (**indicators**).



Obrázek 36 Tři okna projektu v LabVIEW

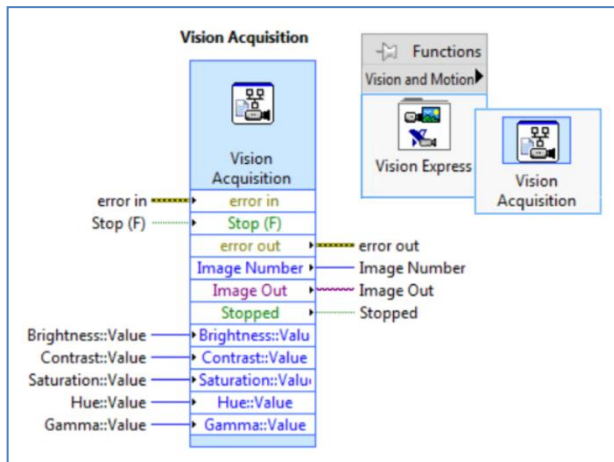
4.4 Získání obrazu z kamery

K získání obrazu slouží podprogramu **Vision Acquisition** (obrázek 37). K jeho vložení stačí zadat jeho název do vyhledávání (**Search**), jež je v pravém horním rohu okna, nebo se otevře při kliknutí pravým tlačítkem myši na bílou plochu. Také ho můžeme najít v takto otevřeném okně ve složce **Functions/Vision and Motion/Vision Express**. Po umístění bloku se automaticky otevře okno pro nastavení získávání obrazu. Mezi jednotlivými částmi nastavení přecházíme tlačítkem **Next**. Nejprve vybereme zdroj obrazu, tedy kameru (**Select Acquisition Source**). V dalším kroku (**Select Acquisition Type**) je třeba zvolit **Continuous Acquisition with inline processing** pro průběžné získávání dat z kamery, aby mohlo dojít k ustálení obrazu, který je zpočátku přesvětlen z důvodu naakumulovaného náboje na snímači kamery. Tímto nastavením bude **Vision Acquisition** vložen do **While Loop** smyčky do které se posléze vloží i většina ostatních bloků programu. V rámci **Configure Acquisition Settings** se nastavují parametry pro úpravu obrazových dat. Tyto parametry lze dodatečně upravovat podle konkrétních světelných podmínek a potřeb, neboť systém nemá dostatečně silný osvětlovač, který by zajistil stabilitu vůči proměnným světelným podmínkám. Z důvodu omezené kapacity zařízení **NI myRIO** je třeba snížit rozlišení obrazu z maximálního 1280x1024 bodů na 640x480 bodů. Další nastavení základních hodnot ukazuje tabulka 1.

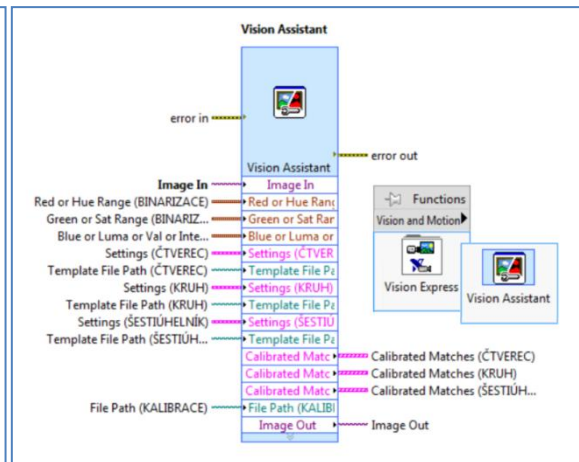
brightness	40
contrast	0,33
gamma	0,72
sharpness	3
saturation	119

Tabulka 2 Configure Acquisition Settings

Tučně vyznačené hodnoty jsou nejdůležitější a proto budou i nastavitelné z **Front Panel**. Ostatní hodnoty, které nejsou v tabulce, můžeme ponechat jak byly přednastaveny. Cestu ke složce pro ukládání pod **Configure Image Logging Settings** nebudeme nastavovat. Nakonec se nastavují vstupy (**controls**) a výstupy (**indicators**) bloku podprogramu v záložce **Select Controls/Indicators**. Některé jsou pevně dány, jako například **error in** a **error out**, jež slouží k případné indikaci chyby, avšak nemusí být nutně připojeny, či **Image Out**, kterým vystupuje vzniklý obraz a musí být připojen. Další volitelné vstupy a výstupy můžeme vybrat zaškrtnutím políčkem. Pro nás jsou podstatné vstupy **Brightness/Value**, **Contrast/Value** a **Saturation/Value**. Po dokončení nastavení kliknutím na tlačítko **Finish**, je třeba zapojit vstupy a výstupy. Vstup **Brightness::Value** se připojí pravým kliknutím myši na vstup bloku a výběrem **Create/Control**. Tím se vytvoří **Control** v okně **Block Diagram** a nastavitelné políčko v okně **Front Panel**. Stejně postupujeme i u dalších vstupů. U výstupů pouze namísto **Control** vytvoříme **Indicator**.



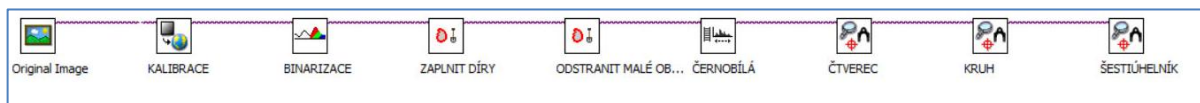
Obrázek 37 Blok podprogramu Vision Acquisition



Obrázek 38 Blok podprogramu Vision Assistant

4.5 Vyhodnocení získaného obrazu ve Vision Assistant

Vyhodnocení obrazu zajišťuje inspekční podprogram **Vision Assistant** (obrázek 38). Do blokového schématu ho přidáme pravým kliknutím myši na bílou plochu a výběrem **Functions/Vision and Motion/Vision Express/ Vision Assistant**. Po přidání se otevře rozhraní pro tvorbu inspekčního programu. Blokové schéma postupu obrazové inspekce je na obrázku 39.



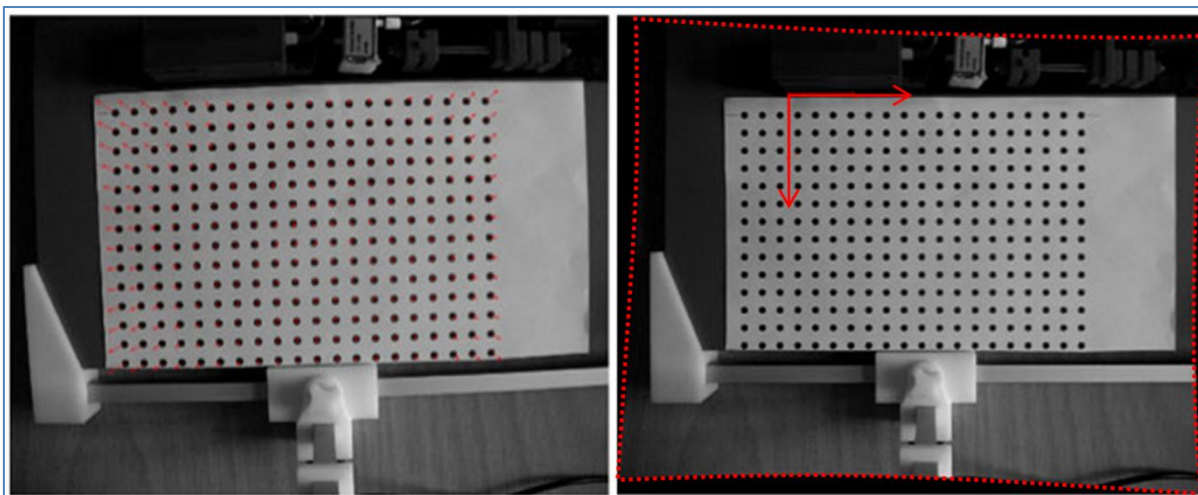
Obrázek 39 Blokové schéma postupu obrazové inspekce ve Vision Assistant

4.5.1 Kalibrace

Nejprve je třeba provést kalibraci. Ta spočívá v tom, že se zachytí kamerou snímek mřížky známých rozměrů, na základě čehož vznikne kompenzační algoritmus pro přepočítání dat, tak aby odpovídala skutečné situaci. Takto jsou kompenzovány chyby způsobené optikou soustavy či perspektivou a obrazovým bodům jsou přiřazeny příslušné jednotky. Pro tento krok je třeba mít vytištěnou kalibrační mřížku, která je součástí volně dostupného průvodce od National Instrument ze zdroje [13], kde se nachází ve složce "NI myRIO Vision Essentials Guide/calibration grids/dot grid - 1cm.pdf". Je nutno nastavit tisk tak, aby nedošlo ke změně rozměrů obrazu a následně ještě přeměřit, zda je vzdálenost mezi tečkami skutečně 1 cm. Vytisknutý list s mřížkou je třeba vhodně umístit pod kameru, aby byla pokud možno v středu a rovnoběžně. Je-li kamera správně připojena, mělo by jít pořídit snímek s kalibrační mřížkou, který vložíme do **Vision Assistant**. (Návod na získávání a vkládání snímků je případně v manuálu [12] na straně 26, došlo-li by i tak k potížím, lze pro pořízení snímku použít také **NI MAX**, který bývá součástí **LabVIEW**). Pro kalibraci je vhodnější černobílý obraz, proto před ní dočasně vložíme nástroj **Color Plane Extraction** ze záložky **Processing Functions: Color** v levém dolním okně a vybereme možnost **HSL - Luminance Plane**. Dále ze záložky **Processing Functions: Image** zvolíme **Image Calibration**. Tlačítkem **New Calibration** otevřeme okno pro nastavení v těchto krocích.

1. **Select Calibration Type:** Zvolíme možnost **Distortion Model (Grid)**
2. **Select Image Source:** Ponecháme vybraný aktuální obrázek.
3. **Extract Grid Features:** Ponecháme přednastavené hodnoty.
4. **Specify Grid Parameters:** Nastavíme hodnoty **X Spacing** a **Y Spacing** na **10** a u **Unit** (jednotek) vybereme možnost **millimeter**.
5. **Review Calibration Results:** Ponecháme.
6. **Specify Calibration Axis:** Nastavíme souřadnice tak, že nejprve klikneme na levý roh černé základové desky robota a poté co nejdále vpravo na hranu téže desky, aby osa x s ní byla rovnoběžná.
7. **Summary:** Potvrdíme tlačítkem **OK**.

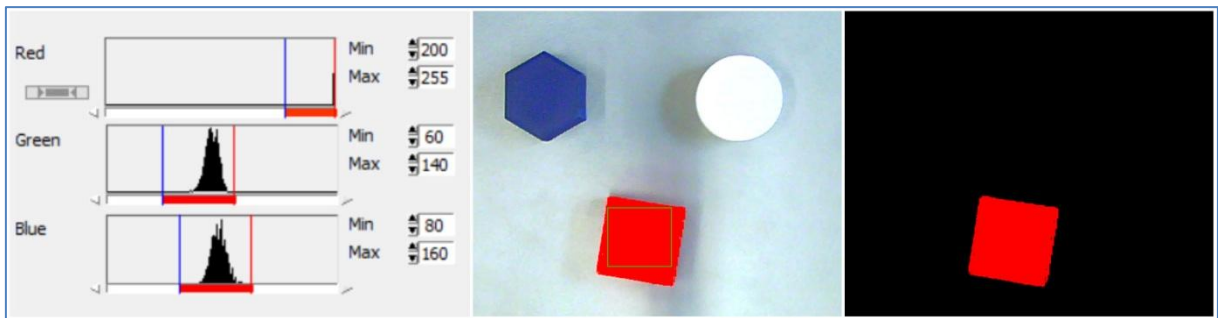
Nyní pojmenujeme a uložíme kalibraci do složky projektu. Ještě je třeba smazat nástroj **Color Plane Extraction**, aby bylo možno pracovat s barevným obrazem, což uděláme tak, že na něho klikneme pravým tlačítkem myši a vybereme možnost **delete**.



Obrázek 40 Vizualizace kalibrace: vlevo původní obraz (s vektory deformace), vpravo zkalibrovaný obraz

4.5.2 Binarizace

Pro detekci tvaru je potřeba obraz binarizovat, což znamená, že ho rozdělíme na dvě barvy tak, aby jednou (červenou) byly zobrazeny objekty, které detekujeme a druhou barvou (černou) jejich pozadí. Tak lze najít jejich hrany a vzniklý obrys porovnat s tvarem vzorovým. Nástroj pro binarizaci je v záložce **Processing Functions: Color** a jmenuje se **Color Threshold**. Nastavení spočívá ve vytvoření rozsahů v histogramu pro tři režimy barev RED, GREEN a BLUE tak aby došlo k potřebnému rozdělení obrazu. Zadává se vždy dolní a horní hranice intervalu, který může ležet v rozsahu 0-255. Tyto hodnoty budou sice nastavovány v rozhraní **Front Panel**, a proto je nyní nemusíme nastavit přesně, avšak je potřeba si zaznamenat potřebné intervaly pro jednotlivé barvy, abychom je poté měly k dispozici. Lze je získat tak, že uděláme snímek dílců o různých barvách, poté vždy označíme část daného dílce a z histogramu vyčteme potřebné hraniční hodnoty tak, aby se vzniklé píky nacházely uvnitř intervalu, avšak je dobré ponechat co největší rezervu (obrázek 41).



Obrázek 41 Nastavení binarizace: vlevo histogram uprostřed barevné dílce vpravo binarizovaný obraz

4.5.3 Převod na černobílý obraz

Pro možnost detekce tvaru je třeba převést obraz na požadovaný binární 8 bitový černobílý formát. Na to se použije nástroj **Lookup Table** v záložce **Processing Functions: Grayscale**, který nastavíme na možnost **Equalize**.

4.5.4 Zaplnění děr a odstranění malých objektů

Při binarizaci obrazu mohou vzniknout slepá místa (díry) uprostřed objektů (obrázek 42). Tyto chyby lze odstranit nástrojem **Adv. Morphology** v záložce **Processing Functions: Binary**, pokud v jeho nabídce vybereme možnost **Fill holes**. Tento samý nástroj lze použít i pro odstranění malých škodlivých objektů a to tak, že vybereme možnost **Remove small objects**, je však třeba ještě nastavit hodnotu **Iterations**, jež určuje maximální velikost odstraňovaných objektů. Tato hodnota nesmí být příliš vysoká, aby nebyly odstraněny i detekované objekty, proto byla zvolena hodnota 15.



Obrázek 42 zleva: původní binarizovaný černobílý obraz, zaplnění děr v objektu, odstranění malých objektů

4.5.5 Detekce tvaru orientace a polohy

Přiřazení tvaru funguje tak, že se porovnává tvar objektu na aktuálním snímku s uloženým vzorem, a pokud je shoda dostatečná, dojde k přiřazení k tomuto tvaru spolu se souřadnicemi a natočením. K tomuto účelu je určen v záložce **Processing Functions: Machine Vision** nástroj **Geometric Matching**. Pro jeho nastavení je zapotřebí mít k dispozici snímky objektů, které chceme detekovat, vložit je do **Vision Assistant** a zpracovat předešlými nástroji. Nejprve je třeba otevřít možnost **New Template**, počemž se otevře okno pro vytvoření vzorového obrázku. Z původního celého obrazu vybereme jen jeho část, tak aby obsahovala celý vzorový tvar. V dalším kroku nastavení změňme

Extraction Mode z volby **Normal** na **Uniform Regions**, což zlepší ostrost rohů aproximovaných okrajů. Velikost kroku vyhledávání ve vodorovném směru (**Row Search Step Size**) snížíme na hodnotu 10, stejně jako velikost kroku vyhledávání ve směru svislém (**Column Search Step Size**). V úseku **Customize Corelation Scoring** ponecháme přednastavené hodnoty, avšak v posledním kroku je potřeba správně umístit a natočit lokální souřadný systém objektu (obrázek 43 vpravo), aby počátek byl ve středu (těžišti) plochy dílce a osy orientovány tak, že při nulovém natočení budou rovnoběžné s osami globálního souřadného systému nastaveného při kalibraci. Také můžeme nastavit, v jakém rozsahu úhlu má být natočení dílce. Jde o symetrické útvary, a tak je úhel natočení relativní, avšak je vhodné brát ten nejmenší možný. Pro konkrétní tvary je nastavení různé. Nejsnazší způsob určení rozsahu v našem případě je asi vydělit úhel 360 stupňů dvojnásobkem počtu os symetrie tvarů. U čtverce je to 0-45 stupňů, u šestiúhelníku 0-30 stupňů a kruh, který má nekonečně os symetrie, nepotřebuje údaj o natočení vůbec. Po vytvoření vzoru je ještě možno v záložce **Settings** upravit minimální hodnotu podobnosti (**Minimum Score**), při které dojde k přiřazení. Tato hodnota může být maximálně 1000 při naprosté shodě, ale většinou se pohybuje mezi hodnotami 800 a 900, jde hlavně o to, aby nedocházelo k záměně tvarů, či přiřazení více tvarů jednomu objektu, nicméně hodnoty bude možno upravovat na ovládacím panelu i dodatečně podle konkrétní situace a případných kolizí. Nástroj přiřazení bude vložen pro každý tvar zvlášť, v našem případě tedy třikrát pro tři různé tvary. Před dokončením práce ve **Vision Assistant** tlačítkem **Finish** je ještě třeba nastavit vstupy a výstupy kliknutím na **Select Controls** vpravo na dolní liště a zaškrtnutím vybrat vstupy: **File Path** (kalibrace), **Red or Hue Range**, **Green or Sat Range**, **Blue or Luma or Val or Inten Range**, **Settings** a **Template File Path** (pro všechny detekce tvaru); a výstupy: **Calibrated Matches** (pro všechny tvary). To vytvoří na bloku podprogramu Vision Assistant segmenty ke kterým je poté možno připojit dané vstupy a výstupy.

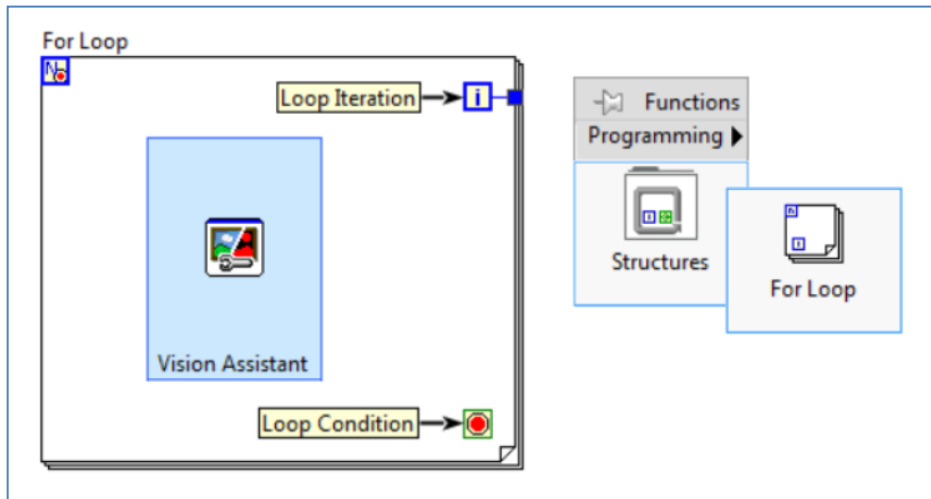


Obrázek 43 Postup obrazové inspekce od původního snímku dílce přes binarizaci až k samotnému rozpoznání tvaru

4.6 Vložení Vision Assistant do smyčky

Celý program je vždy spouštěn ve smyčce, která probíhá buď třikrát podle počtu barev dílců, nebo je podmíněně ukončena kladným výsledkem detekce. Blok **Vision Assistant** vložíme do smyčky tím, že klikneme pravým tlačítkem myši na prázdnou plochu v **Block Diagramu**, vybereme **Functions/Programming/ Structure/For Loop** a držením levého tlačítka myši roztáhneme rámec smyčky přes blok **Vision Assistant** (obrázek 44). Poté klikneme pravým tlačítkem myši na okraj rámce a zaškrtneme možnost **Conditional Terminal**, čímž vytvoříme tlačítko pro podmíněné ukončení

smyčky. Protože budeme také potřebovat index poslední proběhnuté smyčky, přesuneme blok **Loop Iteration** (modrý rámeček s "i" uvnitř) do pravého horního rohu **For Loop** smyčky a vytáhneme jeho výstup na okraj rámce. Na vzniklý **Tunnel** na přechodu smyčky klikneme pravým tlačítkem myši a vybereme **Tunnel Mode/Last Value**.



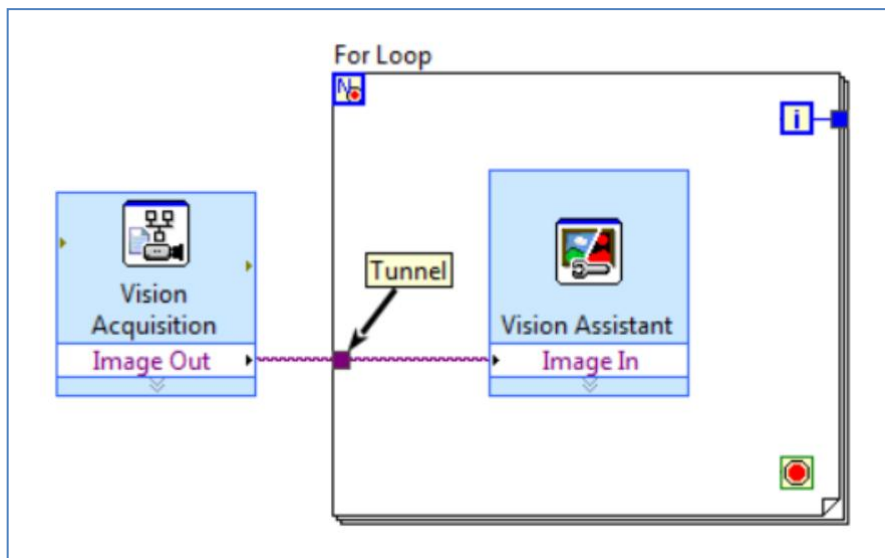
Obrázek 44 Vložení Vision Assistant do For Loop smyčky

4.7 Připojení vstupů a výstupů

Při připojování vstupů a výstupů je důležité vědět, které hodnoty jsou pro všechny cykly stejné a které se mění. Proměnné jsou pouze vstupy pro nastavení binarizace. Jejich počet určuje počet cyklů, jelikož, není-li nastaveno jinak, probíhá smyčka tak dlouho, dokud nezpracuje všechny hodnoty na vstupu. Proto jsou jednotlivé hodnoty vloženy do struktury **Array**, která vytvoří ze sady hodnot matici a oindexuje jednotlivé členy.

4.7.1 Vstupní obraz

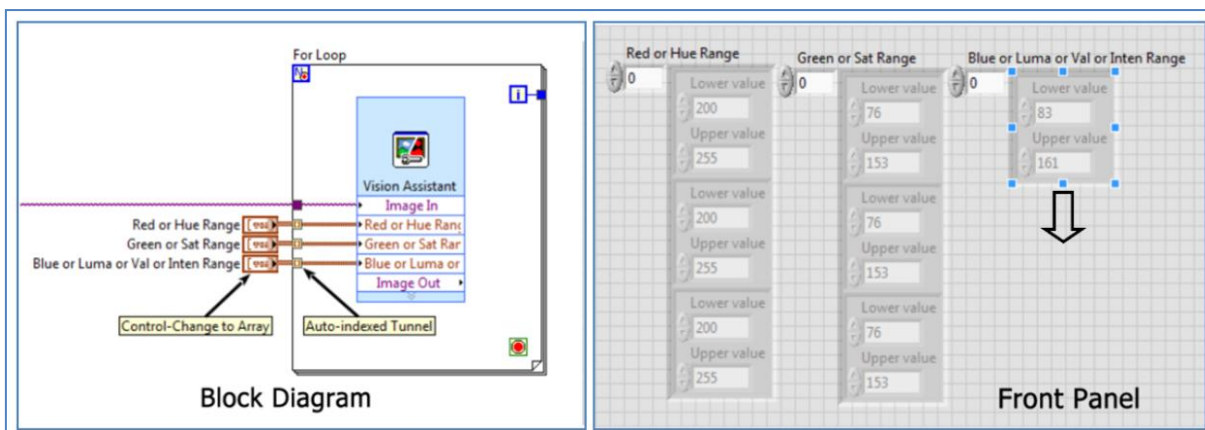
Připojení vstupního obrazu se vytvoří jednoduše přetažením výstupu **Image Out** z **Vision Acquisition** do vstupu **Image In** u **Vision Assistant**, přičemž se automaticky vytvoří **Tunnel** na přechodu rámce smyčky (obrázek 45).



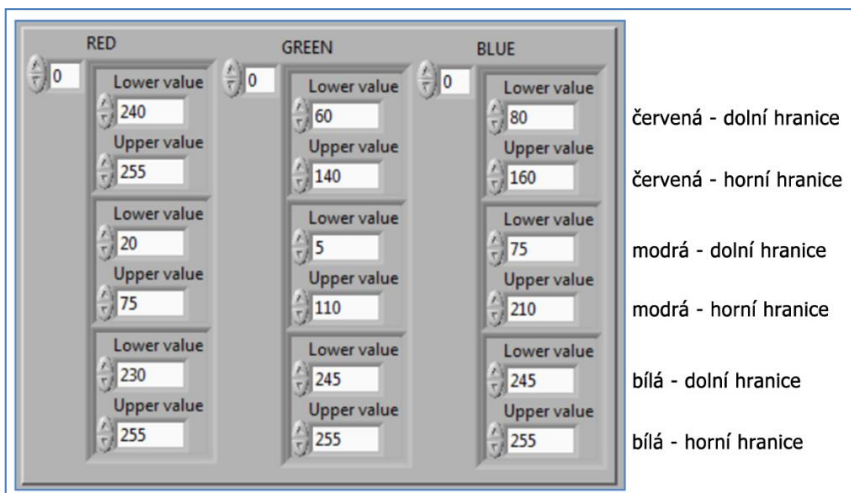
Obrázek 45 Připojení vstupního obrazu z Vision Acquisition k Vision Assistant

4.7.2 Vytvoření pole vstupních hodnot pro binarizaci

Vstupům **Red or Hue Range**, **Green or Sat Range** a **Blue or Luma or Val or Inten Range** vytvoříme **Controls**, které poté vytáhneme ze smyčky a znovu propojíme, čímž vznikne **Tunnel** pro přenos dat do smyčky (obrázek 46 vlevo). Kliknutím pravého tlačítka myši na vniklý **Control** a výběrem **Change to Array** z něho místo jedné dvojice proměnných vytvoříme jejich pole. Dvojitým kliknutím na tento blok se otevře rozhraní **Front Panel** s vytvořeným polem vstupních hodnot. Toto pole je však třeba tažením dolů za jeho dolní okraj rozšířit z jednoprvkového pole na potřebnou velikost tříprvkové pole, viz obrázek 46 vpravo. Tento postup je třeba opakovat pro všechny tři vstupy. Nakonec máme k dispozici tři pole hodnot, každé se třemi svazky o dvou hodnotách. Tato pole vyplníme hodnotami získanými při vytváření binarizace ve **Vision Assistant**, konkrétně jde o tři sady hodnot, kde každá obsahuje tři dvojice hraničních hodnot intervalů pro tři režimy barev RGB. Těchto osmnáct hodnot je třeba správně vyplnit do třech datových polí tak, aby byla zachována posloupnost barev (červená - modrá - bílá) a posloupnost hraničních hodnot intervalů (nižší hodnota - vyšší hodnota) jako na obrázku 47.



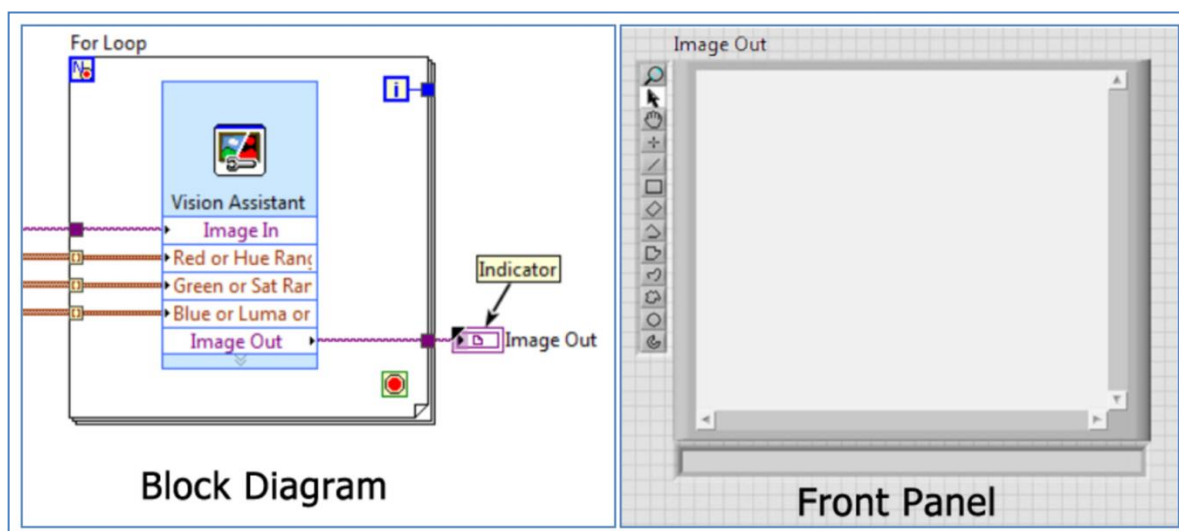
Obrázek 46 Vytvoření vstupního pole hodnot pro binarizaci



Obrázek 47 vyplněné pole hodnot pro červenou, modrou a bílou barvu

4.7.3 Vystupující obraz

Pro vizualizaci binarizovaného obrazu vzniklého obrazovou inspekcí je třeba vytvořit **Indicator** pravým kliknutím myši na výstup **Vision Assistant** pro **Image Out** a výběrem **Create/Indicator**. Tento **Indicator** přetáhnout vně rámce smyčky a znovu ho propojit. Kliknutím pravého tlačítka myši na vzniklý **Tunnel** v záložce **Tunnel Mode** vybereme **Last Value**, aby ze smyčky vystupovala pouze data z posledního cyklu.

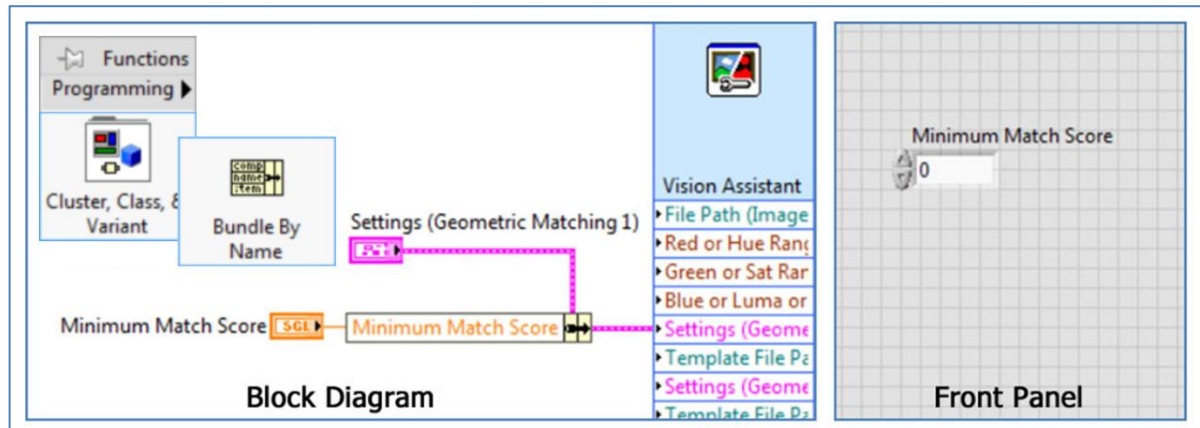


Obrázek 48 Připojení výstupu Image Out a jeho vizualizace v rozhraní Front Panel

4.7.4 Nastavení detekce tvaru

Připojení vstupních parametrů pro detekci tvaru může být i uvnitř smyčky, avšak protože ne všechny hodnoty je třeba měnit v rámci rozhraní **Front Panel**, propojíme **Control** vstupu s dalšími nástroji tak, aby obsahoval jen potřebné hodnoty (obrázek 49). Na začátek vytvoříme **Control** na vstupu **Settings**. Poté vložíme nástroj **Bundle By Name** umístěný ve složce **Functions/Programming/Cluster, Class, & Variant**. Vytvořený **Control** přepojíme shora k **Bundle By Name** (pro nápovědu označte **Bundle**

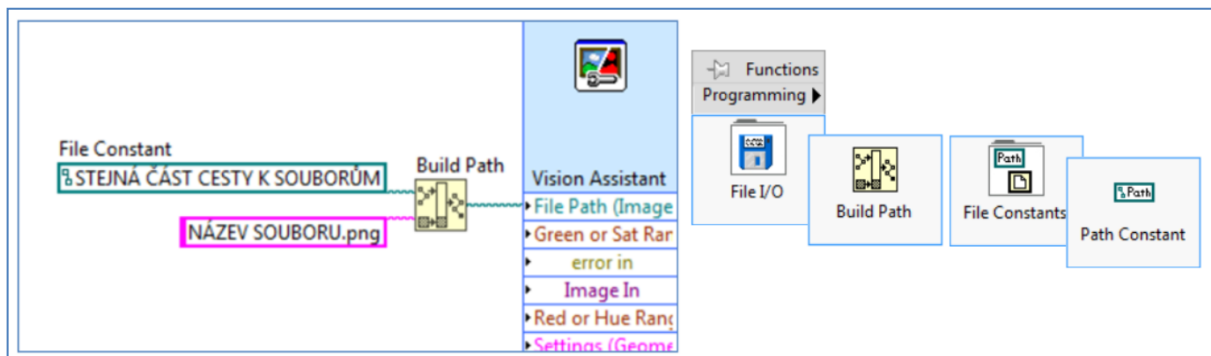
a stiskněte *ctrl H*). Levým tlačítkem myši klikněte na nápis v okně nástroje **Bundle By Name** a vyberte **Minimum Match Score**, aby bylo možné zleva připojit **Control** pro tuto hodnotu



Obrázek 49 Nastavení detekce tvaru vlevo Block Diagram, vpravo Front Panel

4.7.5 Cesta ke složce se vzorovými tvary a kalibrací

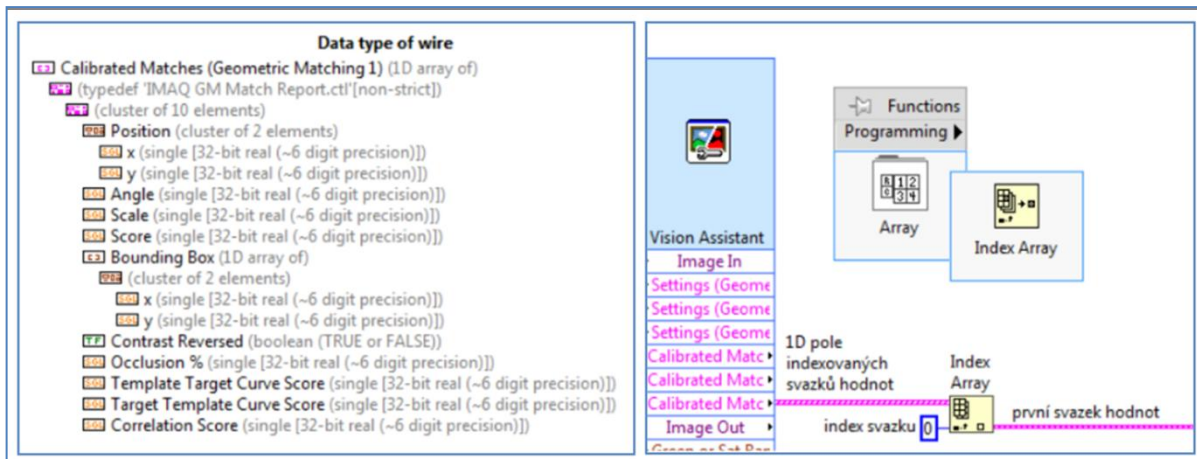
Vstupy jako **File Path**, či **Template File** vyžadují připojit **Control** obsahující cestu k danému souboru s obrazem pro kalibraci, nebo detekci tvaru. Tyto soubory jsou většinou uloženy ve stejné složce, proto je možné použít u všech nástroj, který ke stejné cestě připojí vždy jen odlišný název souboru v cílové složce, aby se vše nemuselo vícekrát vypisovat (obrázek 50). Tento nástroj vložíme z **Functions/Programming/File I/O/Build Path** a připojíme k němu **Path Constant** ze složky **Functions/Programming/File I/O/File Constants**.



Obrázek 50 Vytvoření cesty k souboru v rozhraní Block Diagram

4.7.6 Získání potřebných výsledků detekce z datového pole na výstupu

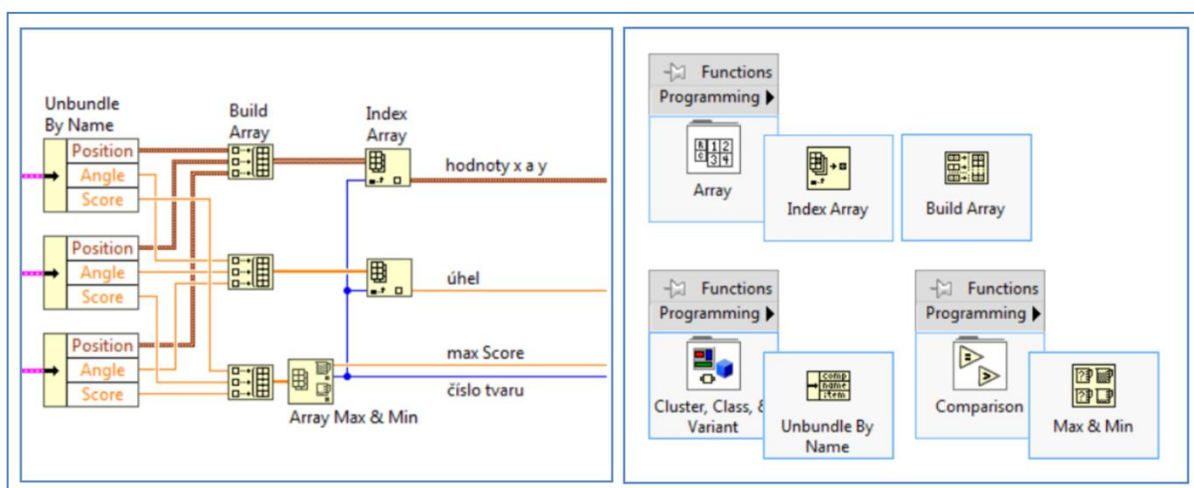
Výsledky detekcí uskutečněných ve smyčce jsou uspořádány v 1D matici a indexované od 0 podle cyklu, ve kterém byly získány (obrázek 51). Pro nás jsou důležité jen složky **Position** (obsahující hodnotu **x** a **y**), **Angle** a **Score**. Abychom je získaly, musíme použít nástroj **Index Array** na jehož horní vstup připojíme výstup **Calibrated Matches** z **Vision Assistant** a na dolní vstup klikneme pravým tlačítkem myši a vybereme **Create/Constant**, do které napíšeme hodnotu **0**, jelikož výsledky jsou automaticky indexovány od 0 pro jednotlivá přiřazení, ač bylo nastaveno, aby bylo provedeno maximálně jedno přiřazení.



Obrázek 51 Vlevo složení 1D pole hodnot, vpravo získání první složky pole

4.7.7 Rozdělení jednotlivých složek výsledků

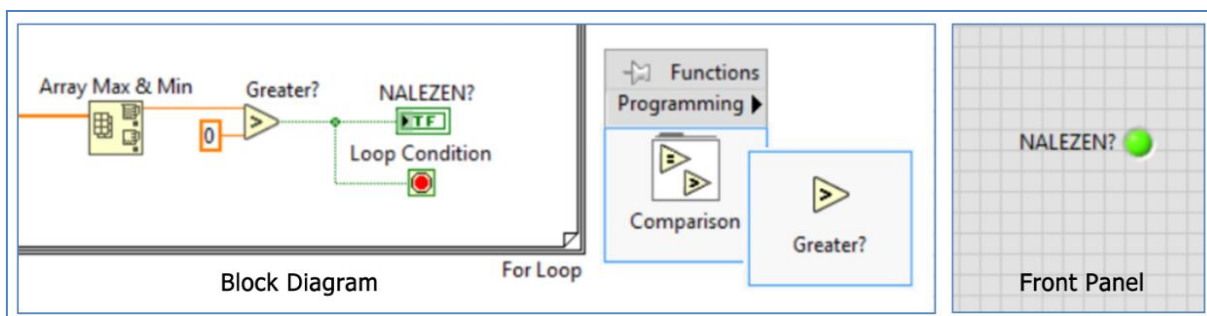
Nyní je k dispozici soubor hodnot jen pro jedno přiřazení, ze kterých se pomocí nástroje **Unbundle By Name** získají jen potřebné složky. Protože však jsou výstupy tři pro jednotlivé tvary, budou i tři sady hodnot, předpokládá se však, že detekujeme vždy jen jeden dílec, proto by měl být kladný jen jeden výstup. Který to je, se zjistí porovnáním hodnoty **Score** nástrojem **Array Max & Min**, kterému se předradí nástroj **Build Array** pro uspořádání dat do pole požadovaného na jeho vstupu. Zleva se tedy připojí na **Build Array** jednotlivá **Score** v daném pořadí, které je třeba stále zachovávat i v další tvorbě blokového schématu. Vpravo pak je výstup jediný, který se připojí na vstup **Array Max & Min**. Výstupem porovnání hodnot jsou čtyři čísla z nichž jsou dále důležité hodnoty **max value** (první shora), podle níž se případně podmíněně ukončí **For Loop** smyčka a **max index** (druhá shora), podle níž se vybere potřebná část z datového pole obsahující hodnoty **Position** a **Angle** a to tak, že se nejprve pomocí nástroje **Build Array** stejně jako u hodnot **Score** pospojují hodnoty **Position** a **Angle** a nástrojem **Index Array** podle hodnoty **max index** se vyberou potřebné složky (obrázek 52).



Obrázek 52 Získání potřebných informací ze tří svazků hodnot výsledků obrazové inspekce

4.7.8 Podmíněné ukončení For Loop smyčky

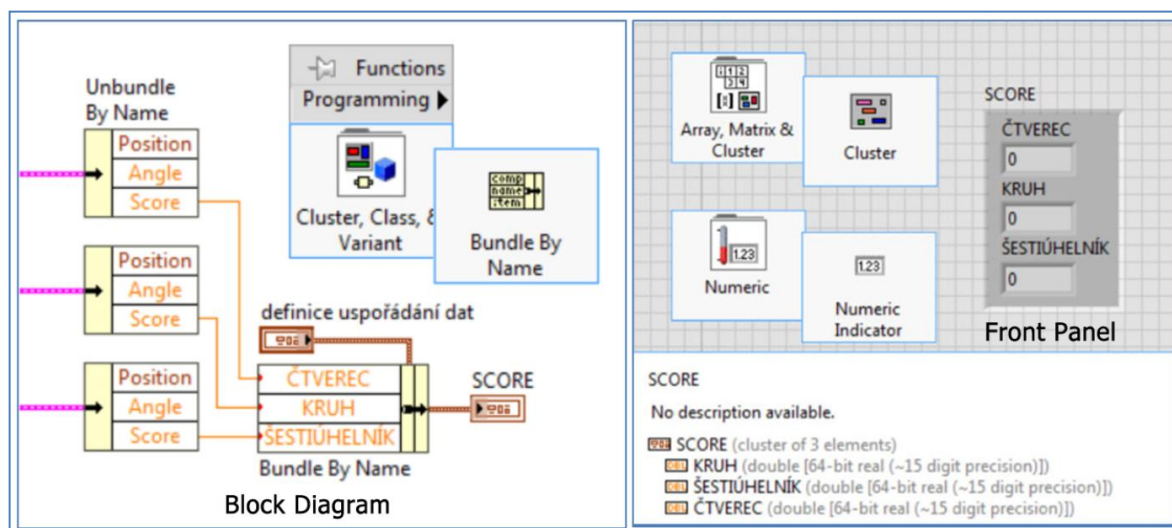
Ukončení smyčky při kladném nález se zajistí tak, že hodnota **max value** je porovnána s nulou nástrojem **Greater?** ze složky nástrojů **Functions/Programming/Comparison** a výsledek je připojen na tlačítko **Loop Condition**.



Obrázek 53 Ukončení For Loop smyčky při kladném výsledku detekce

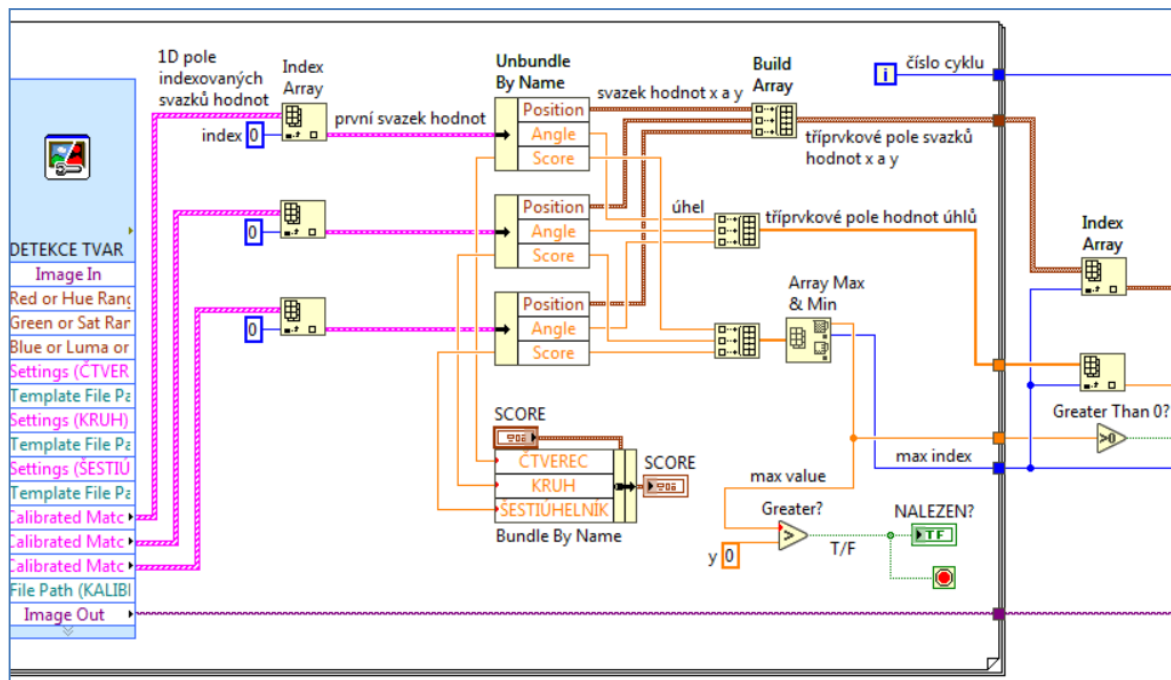
4.7.9 Vizualizace hodnot Score

Pro zobrazení hodnot **Score** v rozhraní **Front Panel** je třeba vytvořit **Indicator**, aby však byly hodnoty vhodně uspořádány, seskupí se do svazku (**cluster**) použitím nástroje **Bundle By Name**. Ten však potřebuje definovat, jak má hodnoty uspořádat, proto je k němu shora připojen blok obsahující požadovanou strukturu dat. Tuto strukturu lze vytvořit v rozhraní **Block Diagram** (obrázek 54 vpravo). Nejprve se pravým tlačítkem myši otevře nabídka nástrojů, kde se vybere **Array Matrix & Cluster/Cluster**. Do vzniklého rámce klastru se vloží tři číselné ukazatele z **Numeric/Numeric Indicator**. Pro ulehčení stačí vložit jeden a ten přetažením myši při současném držení **ctrl** duplikovat.



Obrázek 54 Vizualizace hodnot SCORE

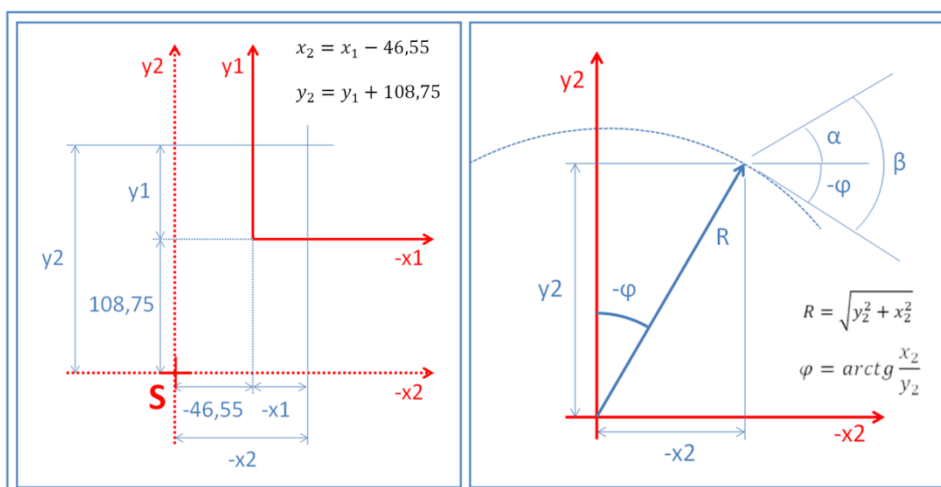
Nyní už jsou k dispozici oddělené hodnoty souřadnic x, y, úhel natočení a index barvy a tvaru (obrázek 55). Úhel natočení je již konečným výstupem, souřadnice je třeba přepočítat do souřadného systému manipulačního robota a indexy tvaru a barvy budou převedeny na textové řetězce.



Obrázek 55 Část blokového schématu inspekčního programu s popisky

4.8 Zpracování a přepoččet dat

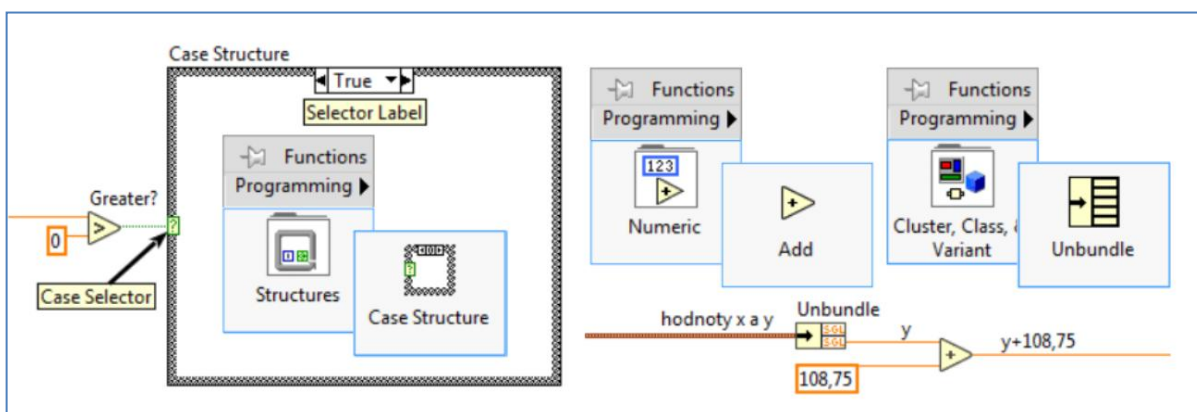
Z podprogramu **Vision Assistant** se získají sice souřadnice detekovaného dílce, avšak protože je třeba pomocí získaných informací řídit manipulačního robota, je třeba vše přepočítat do příslušného souřadného systému. Při tom je nutné dát pozor na kladný směr osy x . Souřadný systém byl definován z pohledu kamery, proto z pohledu robota směřuje doprava záporná osa x . Nejprve se hodnoty x_1 a y_1 (obrázek 56 vlevo) vztažené k rohu desky robota, jak bylo definováno při kalibraci (obrázek 40), převedou na x_2 a y_2 v kartézském souřadném systému, jehož osy jsou rovnoběžné s původními, ale protínají se v ose rotace robota. Poté se přepočítají hodnoty x_2 a y_2 na R a φ (obrázek 56 vpravo), které jim odpovídají v cylindrickém souřadném systému manipulačního robota. R lze vypočítat pomocí Pythagorovy věty a natočení φ pomocí cyklometrické funkce arkustangens.



Obrázek 56 Přepoččet souřadnic, vlevo posunutí os do středu otáčení robota, vpravo přepoččet do cylindrických souřadnic

4.8.1 Vytvoření struktury přepínacího subdiagramu

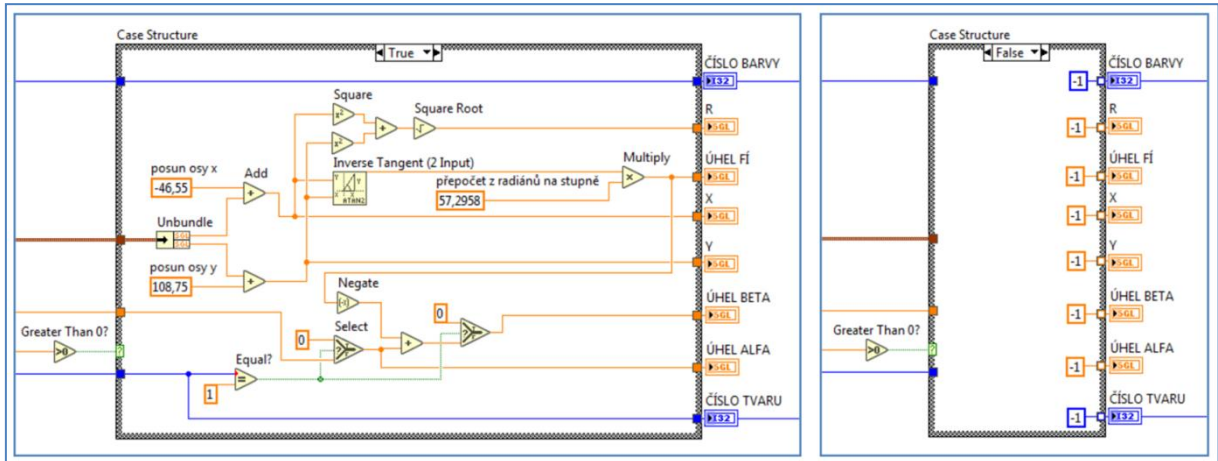
Výsledek detekce je buď kladný, nebo není nalezen žádný dílec, proto se umístí celý přepočít souřadnic do přepínacího subdiagramu **Case Structure** (obrázek 57 vlevo). V něm poté, pokud bude kladný nález, dojde k přepočtu souřadnic, jinak na všech výstupech naskočí "-1". Pro vytvoření **Case Structure** se klikne pravým tlačítkem myši na bílou plochu a z nabídky nástrojů se vybere **Functions/Programming/Structures/Case Structure**, poté se jen držením levého tlačítka myši roztáhne rámec. Na levém okraji bude **Case Selector**, zelený rámeček, v němž je "?". Na něho se připojí výstup funkce **Greater Than 0?** do níž je zleva připojena hodnota **Score** a která vyhodnotí, zda je vstupní hodnota větší než nula, podle čehož má na výstupu hodnotu "1" jako **True**, nebo "0" jako **False**. Na vrchu rámce subdiagramu **Case Structure** je přepínací okénko. Pro každou hodnotu v tomto okénku se vytváří jiný obsah. V tomto případě zde bude pro hodnotu **False** pouze sada konstant s hodnotou "-1". Pro hodnotu **True** bude vytvořen přepočít dat (obrázek 58).



Obrázek 57 Vlevo Case Structure pro přepočít souřadnic, vpravo rozbalení clusteru se souřadnicemi x a y a přičtení konstanty k hodnotě y

4.8.2 Přepočít souřadnic x a y

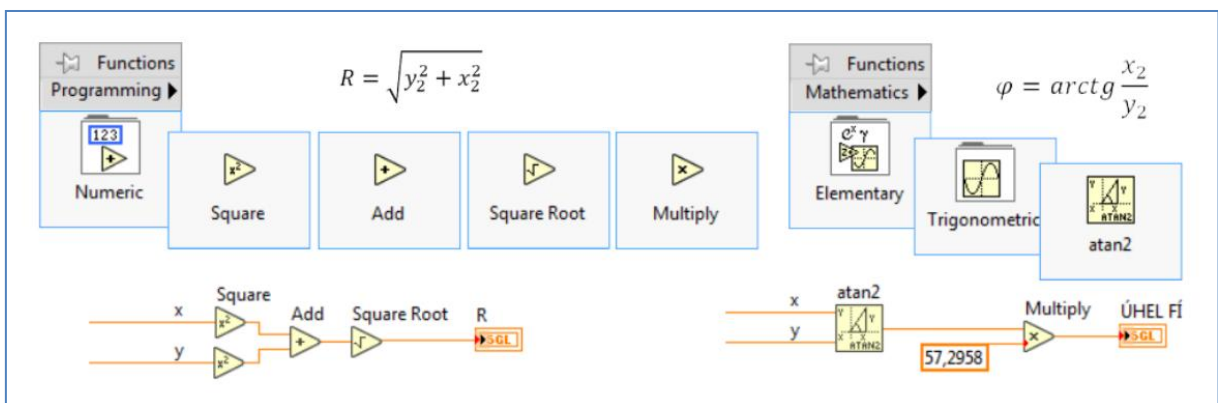
Nejprve je třeba ze svazky hodnot **Position** získat jednotlivé hodnoty **x1** a **y1** nástrojem **Unbundle** ze složky nástrojů **Functions/Programming/Cluster, Class & Variant** (obrázek 57 vpravo). Poté se ke každé složce přičte pouze konstanta, která představuje naměřenou vzdálenost (v milimetrech) osy otáčení robota od počátku zvoleného souřadného systému v obrazové inspekci na rohu základové desky robota. Tak dostaneme souřadnice **x2** a **y2**. Součet se vytvoří nástrojem **Functions/Programming/Numeric/Add**. K hodnotě **x1** se přičte konstanta "-46,55" a k hodnotě **y1** se přičte konstanta "108,75" (obrázek 56 vlevo).



Obrázek 58 Blokové schéma Case Structure vlevo poloha True, vpravo poloha False

4.8.3 Převod do cylindrických souřadnic

Rameno **R** se získá odmocněním součtu umocněných souřadnic **x2** a **y2** (obrázek 59 vlevo). Druhá mocnina vstupu se vytvoří nástrojem **Functions/Programming/Numeric/Square**, takto získané druhé mocniny souřadnic se sečtou funkcí **Functions/Programming/Numeric/Add** a součet se odmocní vřazením **Functions/ Programming/Numeric/Square Root**. Úhel natočení φ se vypočítá jako arkustangens podílu souřadnic **x2** a **y2** (obrázek 59 vpravo), využitím nástroje **Functions/ Mathematics/Elementary/Trigonometric/atan2**. Výsledek je však v radiánech, proto se převede na stupně vynásobením konstantou 57,2958 nástrojem **Functions/Programing/Numeric/Multiply**.



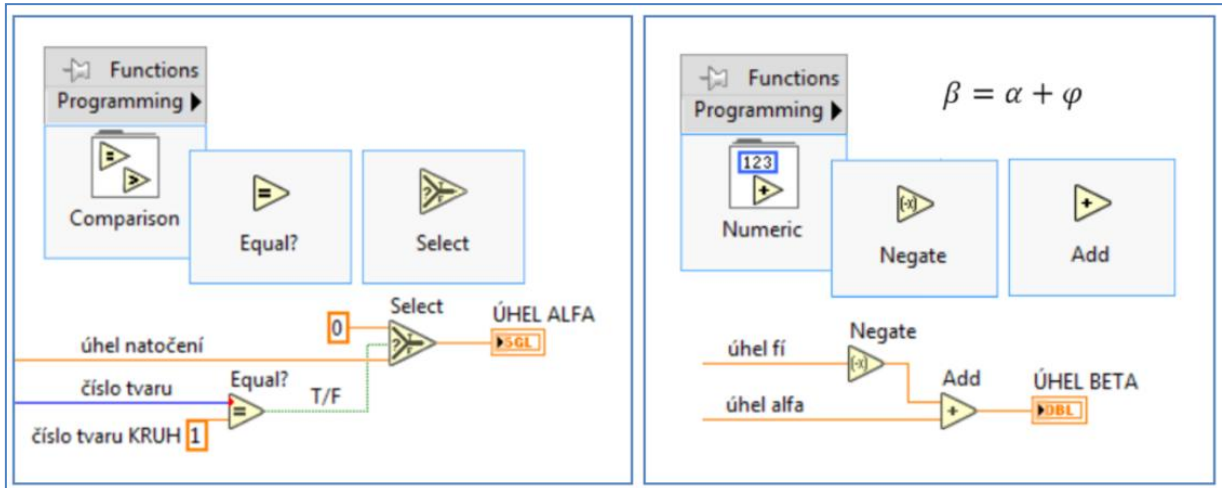
Obrázek 59 Přepočít souřadnic x a y na souřadnice R a φ

4.8.4 Nulové hodnoty natočení kruhu

U tvaru **KRUH** nelze jednoznačně určit úhel natočení, proto je tento údaj zbytečný, avšak při detekci je vždy nějaký úhel přiřazen. Z tohoto důvodu se do rámce **Case Structure** vkládá ještě nástroj **Equal?**, který při hodnotě čísla rovnající se nálezu tvaru **KRUH** generuje hodnotu **True**, na základě které funkce **Select** přepíše hodnoty na "0" (obrázek 60 vlevo).

4.8.5 Výsledné natočení

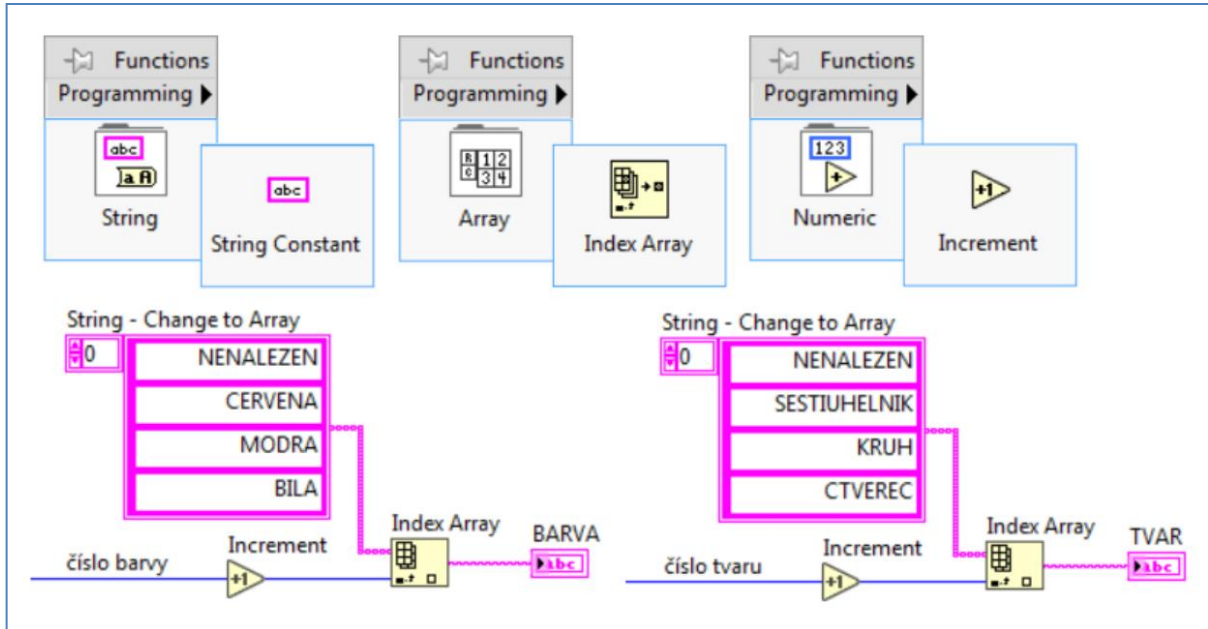
Výsledné natočení dílce se skládá z úhlu α a úhlu φ , neboť uchopí-li robot dílec a natočí se do nulové pozice, bude natočení zvětšeno o úhel pootočení robota, tedy úhel φ . Součet obou úhlů se označí β . Součet se vytvoří nástrojem **Add**, akorát před vstup hodnoty úhlu φ vřadíme funkci **Negate**, protože záporná hodnota úhlu φ celkové natočení zvětšuje (obrázek 60 vpravo).



Obrázek 60 Vlevo realizace zajištění nulových hodnot pro tvar KRUH, vpravo součet úhlu α a úhlu φ

4.8.6 Zobrazení textových řetězců

Aby se k číslu přiřadilo textové označení, je třeba vytvořit **String Constant**, která se změní na pole textových konstant kliknutím pravého tlačítka myši a výběrem **Change to Array**. Takto vzniklý blok lze roztáhnout na potřebný počet oken, do kterých se napíše textová označení odpovídající jednotlivým číslům. Okna jsou indexovaná shora od "0". Protože předchozí číslování začínalo "-1" pro negativní nálezy, je třeba ještě pomocí funkce **Increment** přičíst ke každé hodnotě "+1". Z textového bloku vystupuje indexované pole textových hodnot, ze kterého se získá potřebná část pomocí **Index Array** na základě čísla tvaru, nebo barvy (obrázek 61).



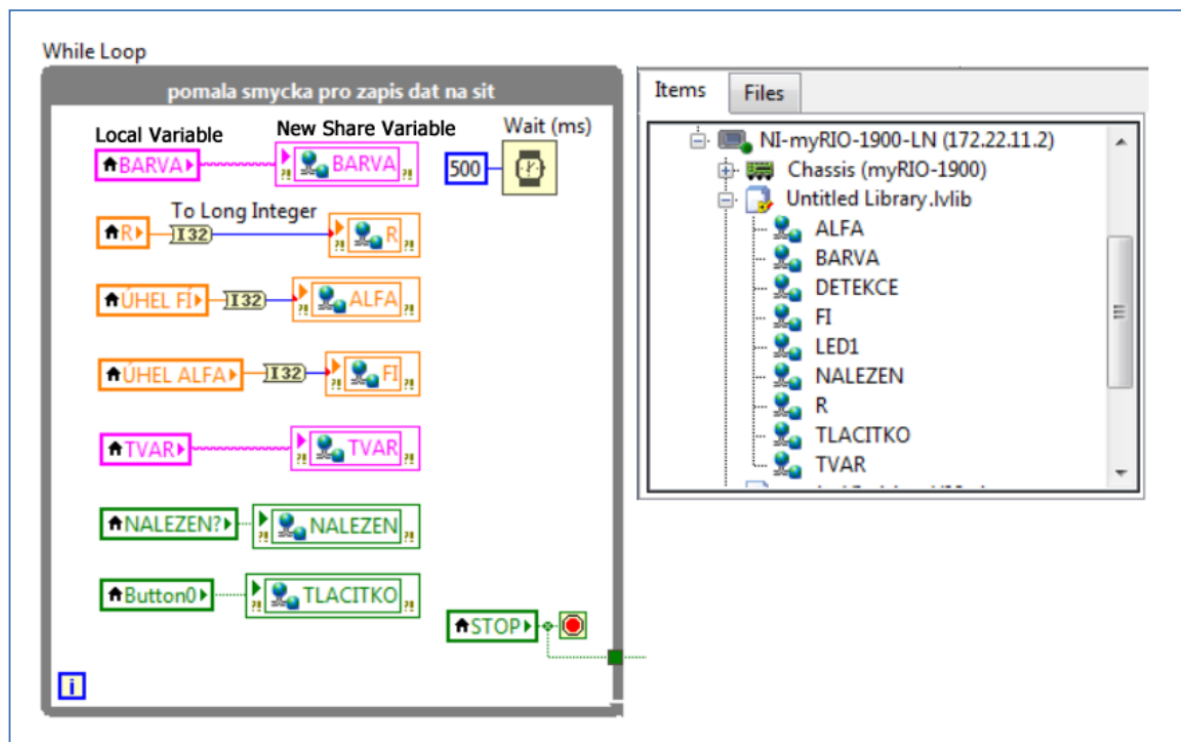
Obrázek 61 Zobrazení textových řetězců

4.9 Spuštění programu a jeho komunikace s okolím

Celý program bude spuštěn zároveň se zapnutím zařízení **NI myRIO**, avšak samotná obrazová inspekce a vyhodnocení bude inicializováno vnějším podnětem. Tím bude buď stisknutí softwarového tlačítka na **Front Panelu**, nebo hardwarového tlačítka na boku zařízení **NI myRIO**. Po proběhnutí obrazové inspekce je třeba z ní získaná data předat jiným zařízením, například PLC řídící robota. Toto předávání dat probíhá přes bezdrátovou síť wifi.

4.9.1 Sdílené proměnné

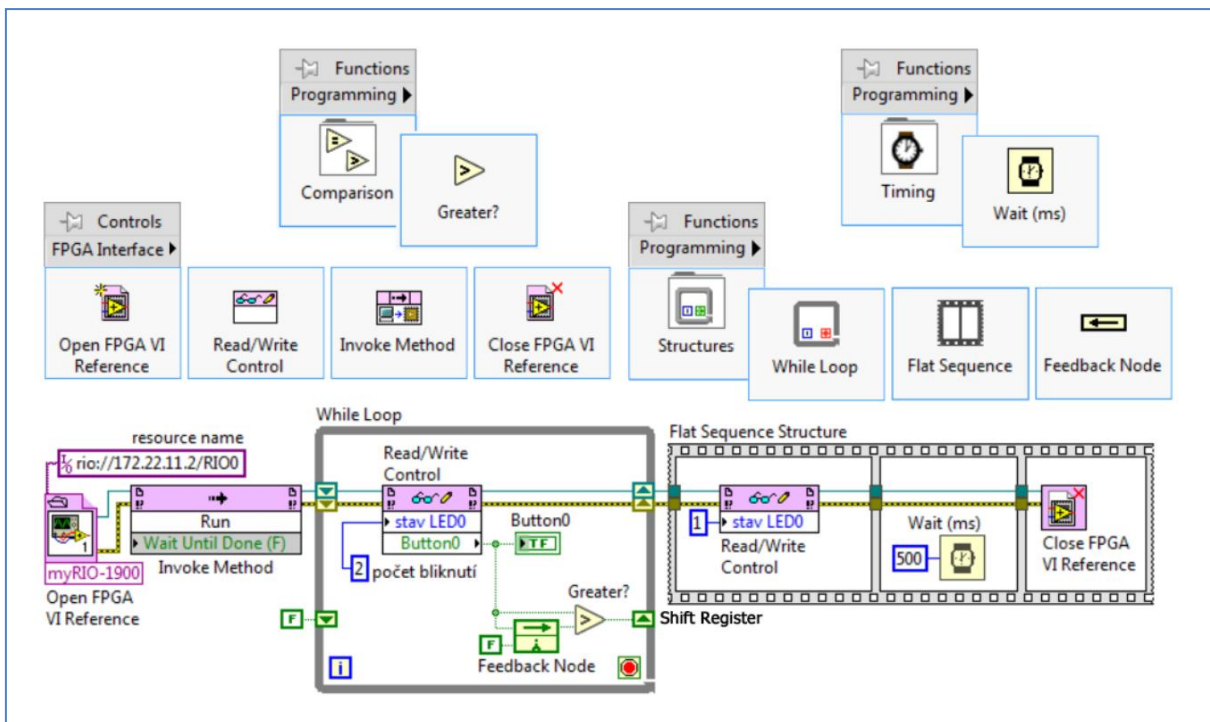
Pro získávání výsledných dat inspekčního programu je třeba vytvořit sdílené proměnné, které slouží jako výstupy sdílené přes síť wifi s jinými zařízeními. Nejprve je třeba ve stromu projektu vytvořit knihovnu proměnných (obrázek 62 vpravo). Pravým tlačítkem myši se klikne na složku **NI-myRIO-1900...** a vybere se **New/Library**. Tak se vytvoří složka **Untitled Library**, na ní se klikne pravým tlačítkem myši a vybere se **New/Variable** zvolí se typ proměnné a její název. Takto se vytvoří všechny proměnné, které budou sdíleny. V **Block Diagram** se vytvoří rámec **While Loop** smyčky (obrázek 62 vlevo), do které se vloží časovač **Wait (ms)**, k němuž se připojí konstanta s hodnotou "500", což znamená, že smyčka proběhne vždy po 500-ti milisekundách. Dále se vytvoří **Local Variable** pro všechny sdílené hodnoty a pro spouštěcí tlačítko **DETEKCE**. Pravým tlačítkem myši se klikne na **Indicator** dané proměnné a vybere se **Create/Local Variable**. Vzniklé lokální proměnné se přetáhnou do **While Loop** smyčky a připojí se k nim sdílené proměnné pravým kliknutím myši na výstup a výběrem **Create/Shared Variable Nodes/New Share Variable**. U hodnot, které stačí jako celá čísla, lze mezi tyto bloky vřadit funkci **To Long Integer**.



Obrázek 62 Vlevo While Loop smyčka pro sdílení proměnných, vpravo strom projektu s vytvořenou knihovnou proměnných

4.9.2 Spuštění obrazové inspekce přes tlačítko na zařízení NI myRIO

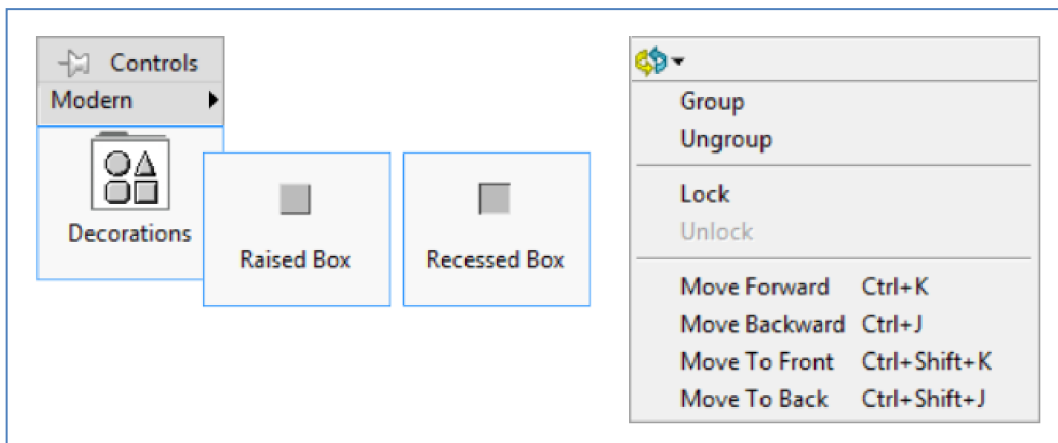
Pro iniciaci průběhu inspekčního programu může být použito tlačítko na zařízení **NI myRIO**. Je třeba vytvořit **While Loop** smyčku, do které se vloží dosud vytvořený program. Dále se vytvoří bloky **Open FPGA VI Reference**, **Invoke Method**, **Read/Write Control**, **Close FPGA VI Reference** a zabudují se do blokového diagramu dle obrázku 63. Z názvu bloků je patrné, že jde o část programu spouštěného na **FPGA** (Field Programmable Gate Array - Programovatelná hradlová pole), což je hardwarová část **NI myRIO**, která dokáže pracovat rychleji než procesor téhož zařízení, je však omezena svou kapacitou, danou počtem programovatelných logických obvodů. V rámci bloku **Read/Write Control** můžeme použít tlačítko **Button0** jako vstup našeho programu. Také lze nastavit počet a čas blikání tří diod. V našem případě je program sestaven tak, aby při zmáčknutí tlačítka **Button0** na boku zařízení **NI myRIO** došlo ke spuštění programu obrazové inspekce a dioda **LED0** indikovala běh programu dvojitým bliknutím, jeho ukončení pak bliknutím jediným. Porovnávání hodnoty tlačítka **Button0** nástrojem **Greater?** s hodnotou předchozí pomocí funkce **Feedback Node** zajišťuje, aby při držení tlačítka došlo pouze jednou k iniciaci spuštění. Další iniciace může nastat až po uvolnění tlačítka a opětovném stlačení. Aby byly převedené výstupy z jedné smyčky do druhé, je potřeba změnit přechod **Tunnel**, který je na okraji rámece **While Loop** smyčky, kam jsou připojeny výstupy, na **Shift Register** pravým kliknutím myši a výběrem **Replace with Shift Register**. Takto vzniknou dva paralelní bloky, které hodnotu přivedenou na pravý blok v jednom cyklu převedou na blok levý v dalším cyklu. To způsobí, že hodnota **True** získaná stlačením tlačítka **Button0** v průběhu některého z cyklů smyčky spustí v cyklu následujícím program obrazové inspekce.



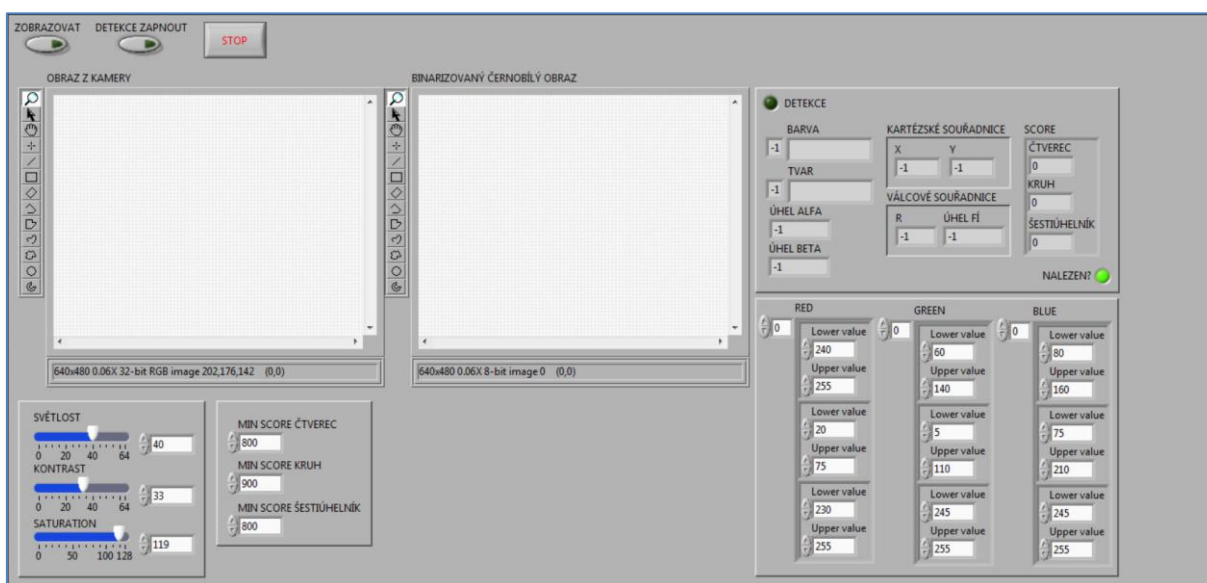
Obrázek 63 Část programu pro získání vstupu Button0 a nástroje v něm použitých

4.9.3 Konečné propojení jednotlivých částí blokového diagramu

Konečné propojení zajišťuje správné spuštění a ukončování celého programu a jeho podčástí (obrázek 64). Detekce je spuštěna buď virtuálním tlačítkem **DETEKCE ZAPNOUT** na **Front Panelu**,



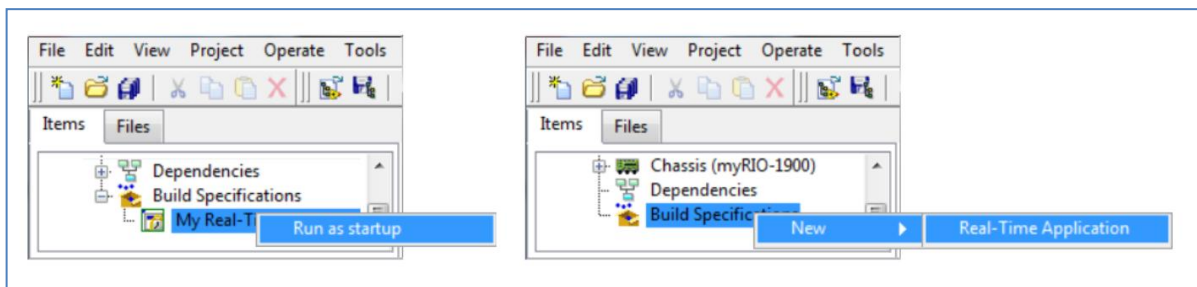
Obrázek 65 Vlevo nástroje pro tvorbu Front Panelu, vpravo záložka pro uspořádání bloků



Obrázek 66 Uspořádání Front Panelu

4.11 Nahrání programu na NI myRIO a nastavení automatického spuštění

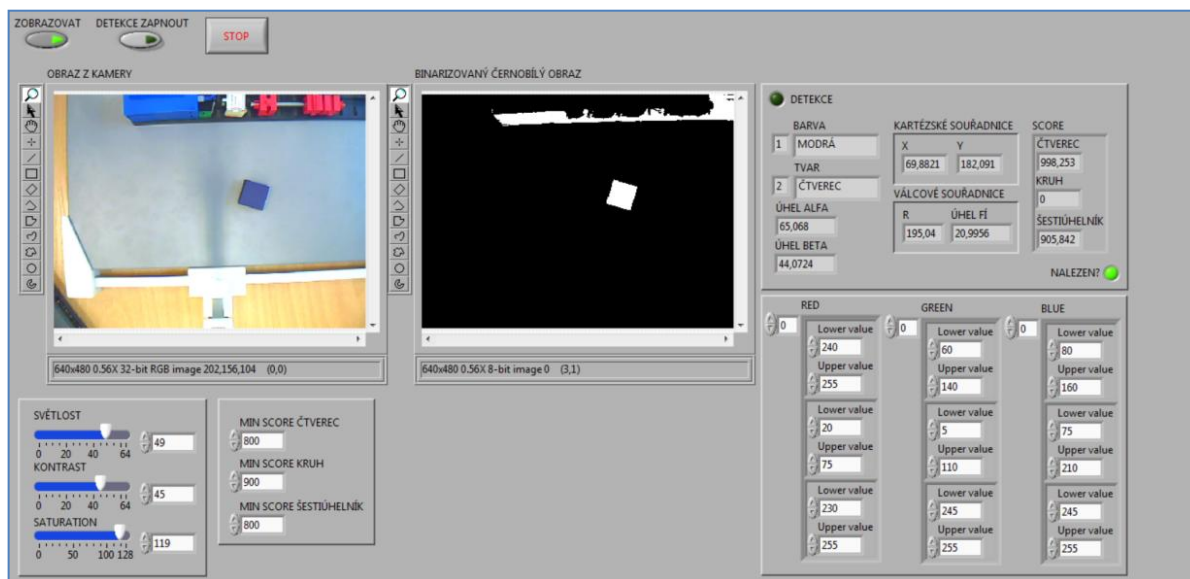
Vytvořený program se ve stromu projektu nachází ve složce **My Computer**, proto se s pomocí na PC, ve kterém je program sestavován. Aby se program spouštěl na zařízení **NI myRIO**, je třeba ho přesunout ve stromu do složky **NI-myRIO-1900...** Chceme-li spouštět program nezávisle na PC, je třeba vytvořit **My Real-Time Application**, jež se uloží přímo do paměti zařízení **NI myRIO**. Na samém konci stromu projektu se pravým tlačítkem myši klikne na záložku **Build Specifications**, vybere se **New/Real-Time Application** (obrázek 67 vpravo), v okně nastavení se vybere v záložce **Source Files** pro **Startup VIs** vytvořený inspekční program (koncovka .vi) a pro **Always Included** knihovna proměnných (koncovka .lvlib), nakonec se dokončí kliknutím na dolní tlačítko **Build**. Pro nastavení spuštění této aplikace po spuštění zařízení na ní klikneme ve stromu projektu pravým tlačítkem myši a vybereme **Run as startup** (obrázek 67 vlevo). Tím se po zapnutí zařízení **NI myRIO** automaticky načte a spustí i program.



Obrázek 67 Vlevo nastavení jako aplikace spouštěná po zapnutí, vpravo vytvoření aplikace v paměti zařízení NI myRIO

4.12 Ověření funkčnosti

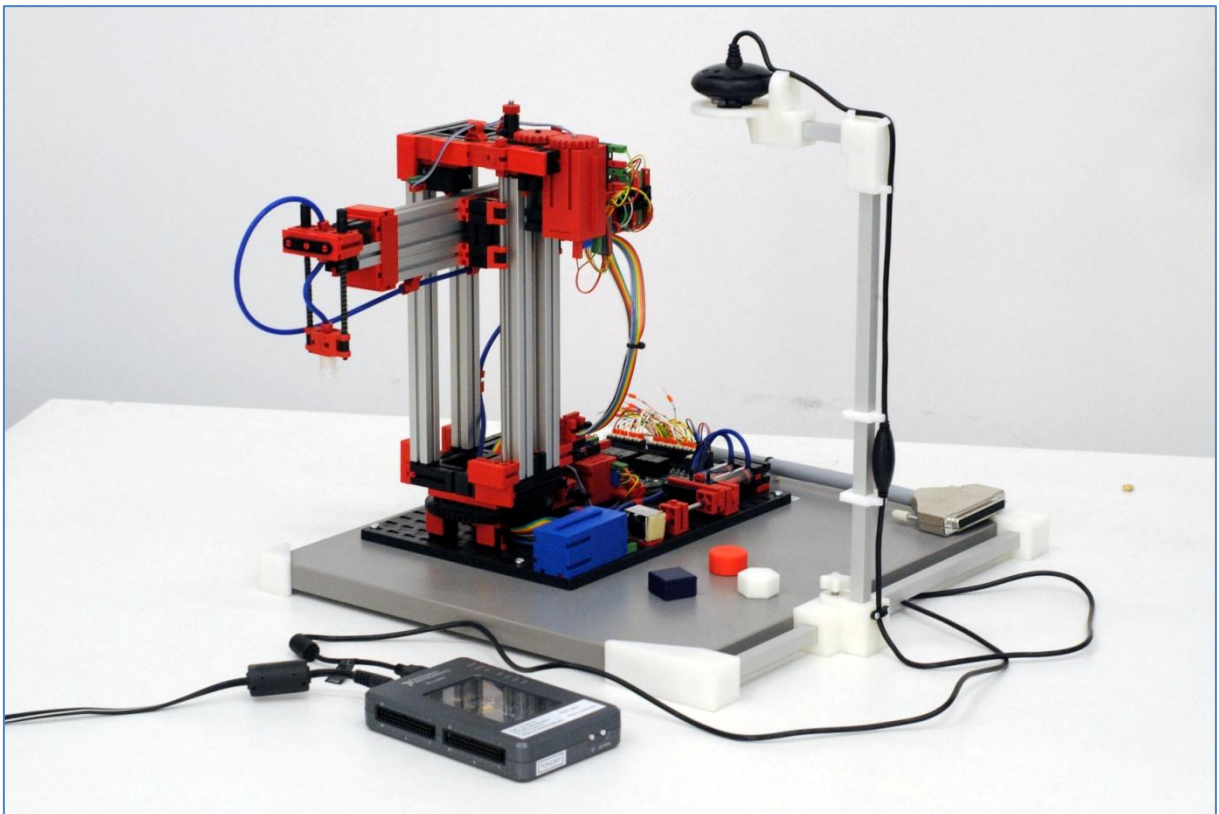
Pro ověření funkčnosti programu byla nainstalována na zařízení s operačním systémem Android aplikace **Data Dashboard** od společnosti National Instrument. Díky této aplikaci je možné získávat data ze zařízení **NI myRIO** přes síť wifi. Pokud byly zajištěny přibližně stejné světelné podmínky při jakých byl systém nastaven, dokázal rozpoznat všechny tři tvary i barvy dílců. Byla ověřena funkčnost tlačítka pro spuštění detekce. I ověření sdílení dat přes síť wifi proběhlo úspěšně, jen bylo třeba odstranit diakritiku z textových řetězců, neboť některé znaky byly špatně dekodovány. Nejnáchylnější na změnu osvětlení byla detekce červených dílců. Při změně osvětlení také docházelo k záměně šestiúhelníku za kruh. To byl problém hlavně po tom co byl program přenastaven na nižší rozlišení obrazu z důvodu malé kapacity zařízení. Obrázek 68 z uživatelského rozhraní **Front Panel** ukazuje vizualizaci nastavení a výsledků detekce v prostředí **LabVIEW**.



Obrázek 68 Výsledky detekce modrého čtvercového dílce zobrazené na Front Panelu

5 Závěr

Byl zpracován přehled výrobců a systémů strojového vidění. Byl navržen systém strojového vidění pro manipulátor s cylindrickou kinematickou strukturou od firmy Fishertechnik na platformě National Instrument, který rozpozná tři tvary dílců (čtverec, kruh, šestiúhelník), tři barvy (červená, modrá, bílá) a jejich polohu a natočení vzhledem k souřadnému systému manipulačního robota. Systém byl otestován a shledán funkčním. Z důvodu slabého kruhového osvětlovače kamery a závislosti na okolním osvětlení však vykazoval nestabilitu. To by nejspíše řešil silnější přední plošný osvětlovač s tmným zorným polem, aby nevytvářel odlesky na pozadí. Tím byly splněny pokyny zadání. Do budoucna, avšak již nad rámec této bakalářské práce, by mělo dojít k oživení robota řízeného vzniklým systémem strojového vidění. K tomu bude ještě třeba propojit zařízení NI myRIO s PLC SIMATIC S7-1500 řídící samotného robota.



Obrázek 69 Fotografie laboratorního robota s kamerou, stojanem, třemi dílci a zařízením NI myRIO

Zdroje a Citace

- [1] FCC Průmyslové Systémy. Strojové vidění [online]. [cit. 2016-12-05]. Dostupné z: <http://www.strojove-videni.cz>
- [2] AJP-Tech. Baumer kamery [online]. [cit. 2016-12-05]. Dostupné z: <http://www.ajptech.cz/produkty/prumyslova-automatizace/baumer-kamery/>
- [3] Fischertechnik: training models [online]. [cit. 2016-12-05]. Dostupné z: http://www.fischertechnik.de/en/desktopdefault.aspx/tabid-24/41_read-489/usetemplate-2_column_pano/
- [4] NI myRIO. National Instruments [online]. [cit. 2016-12-05]. Dostupné z: <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/211694>
- [5] MyRIO Robot Project. MICHIGAN STATE UNIVERSITY [online]. [cit. 2016-12-05]. Dostupné z: http://www.egr.msu.edu/classes/me451/me451_labs/robot/MyRIO%20Robot%20Project_CartMaking.html
- [6] Vestavné zařízení pro studenty NI myRIO. AUTOMA časopis pro automatizační techniku [online]. [cit. 2016-12-05]. Dostupné z: http://automa.cz/cz/web-clanky/vestavne-zarizeni-pro-studenty-ni-myrio-52193_08341/
- [7] Strojové vidění I: Principy a charakteristiky. AUTOMA časopis pro automatizační techniku [online]. [cit. 2016-12-05]. Dostupné z: http://automa.cz/cz/casopis-clanky/strojove-videni-i-principy-a-charakteristiky-2008_01_36550_5518/
- [8] Strojové vidění II: Úlohy, nástroje a algoritmy. AUTOMA časopis pro automatizační techniku [online]. [cit. 2016-12-05]. Dostupné z: http://automa.cz/cz/casopis-clanky/strojove-videni-ii-ulohy-nastroje-a-algoritmy-2008_02_36676_5521/
- [9] Strojové vidění III: Kamery a jejich části. AUTOMA časopis pro automatizační techniku [online]. [cit. 2016-12-05]. Dostupné z: http://automa.cz/cz/casopis-clanky/strojove-videni-iii-kamery-a-jejich-casti-2008_03_36925_5342/
- [10] Strojové vidění IV: Osvětlovače. AUTOMA časopis pro automatizační techniku [online]. [cit. 2016-12-05]. Dostupné z: http://automa.cz/cz/casopis-clanky/strojove-videni-iv-osvetlovace-2008_04_36988_5853/
- [11] KEYENCE: Vision / Auto ID [online]. [cit. 2017-02-24]. Dostupné z: <http://www.keyence.com/products/vision/index.jsp>
- [12] NI Vision: NI Vision Assistant Tutorial [online]. 372228M. Austin, Texas, USA, 2011 [cit. 2017-03-30]. Dostupné z: <http://www.ni.com/pdf/manuals/372228m.pdf>
- [13] DOERING, Ed. NI myRIO Vision Essentials Guide [online]. July 23, 2015. National Technology and Science Press, 2015 [cit. 2017-03-30]. Dostupné z: <http://www.ni.com/white-paper/52475/en/>
- [14] BERGER, Hans. Automating with SIMATIC S7-1500 Configuring, Programming, Motion Control and Security inside TIA Portal. Erlangen: PUBLICIS, 2013. ISBN 9783895784040.

[15] Trust: Webcams [<http://www.trust.com/en/product/16428-spotlight-we>]. [cit. 2017-04-15].

Seznam obrázků

Obrázek 1 Vacuum Gripper Robot 24V od firmy fischertechnik [3]	10
Obrázek 2 Obecné uspořádání systému strojového vidění [7]	11
Obrázek 3 Blokové schéma inteligentní kamery [7]	12
Obrázek 4 Blokové schéma kamerového senzoru [7].....	12
Obrázek 5 Blokové schéma PC systému strojového vidění [7]	13
Obrázek 6 Vztah úlohy nástroje a algoritmu v inspekčním programu [8].....	14
Obrázek 7 Příklad úlohy nalezení povrchové vady: housenka těsnícího tmelu [1]	14
Obrázek 8 Příklad úlohy počítání: kontrola naplnění blistru s léky [1]	15
Obrázek 9 Příklad úlohy kontroly tolerancí: měření závitu a kužele na kulovém čepu [1]	15
Obrázek 10 Příklad úlohy identifikace barev: třídění dílců podle barvy [1]	16
Obrázek 11 Příklad úlohy čtení textu: kontrola data výroby nebo expirace [1]	16
Obrázek 12 Nejčastěji používané rozměry snímačů pro snímání obrazu [9]	18
Obrázek 13 Znázornění pixelu a rozlišení na černobílém a barevném snímači	19
Obrázek 14 Snímání obrazu kamerou	20
Obrázek 15 Zobrazení standardním a telecentrickým objektivem	21
Obrázek 16 Schéma předního osvětlení s jasným zorným polem	22
Obrázek 17 Schéma předního osvětlení s tmným zorným polem	23
Obrázek 18 Schéma zadního osvětlení.....	23
Obrázek 19 Schéma plošného osvětlovacího pole	24
Obrázek 20 Schéma kruhového osvětlovacího pole	24
Obrázek 21 vpravo schéma kopulového osvětlovacího pole vlevo příklad použití (převzato z [10]) ...	25
Obrázek 22 vpravo schéma osvětlovače DOAL a vlevo příklad použití [10]	25
Obrázek 23 vpravo schéma osvětlovače dark-field a vlevo příklad použití [10]	26
Obrázek 24 vpravo schéma osvětlovače backlight a vlevo příklad použití [10]	26
Obrázek 25 vlevo schéma propojení systému řady CV-X100 od firmy KEYENCE a vpravo shora kamera CV-S200C, osvětlovač CA-DRW4F a objektiv CA-LHS25 [11].....	28
Obrázek 26 Vestavné zařízení NI myRIO [4]	29
Obrázek 27 Kamera Trust 1.3 MPx [15].....	29
Obrázek 28 Schéma propojení jednotlivých zařízení.....	30
Obrázek 29 Původní stav laboratorního robota	31
Obrázek 30 3D model stojanu se základními rozměry a pozicemi.....	32
Obrázek 31 Dílce vytištěné na 3D tiskárně	33
Obrázek 32 3D modely dílců	33
Obrázek 33 Schéma rozvržení programu	33
Obrázek 34 Celé zmenšené komentované blokové schéma programu	34
Obrázek 35 Rozložení histogramu pro jednotlivé barvy a pozadí.....	35
Obrázek 36 Tři okna projektu v LabVIEW	35
Obrázek 37 Blok podprogramu Vision Assistant.....	37

Obrázek 38 Blok podprogramu Vision Acquisition	37
Obrázek 39 Blokové schéma postupu obrazové inspekce ve Vision Assistant	37
Obrázek 40 Vizualizace kalibrace: vlevo původní obraz (s vektory deformace), vpravo zkalibrovaný obraz	38
Obrázek 41 Nastavení binarizace: vlevo histogram uprostřed barevné dílce vpravo binarizovaný obraz	39
Obrázek 42 z leva: původní binarizovaný černobílý obraz, zaplnění děr v objektu, odstranění malých objektů	39
Obrázek 43 Postup obrazové inspekce od původního snímku dílce přes binarizaci až k samotnému rozpoznání tvaru	40
Obrázek 44 Vložení Vision Assistant do For Loop smyčky	41
Obrázek 45 Připojení vstupního obrazu z Vision Acquisition k Vision Assistant	42
Obrázek 46 Vytvoření vstupního pole hodnot pro binarizaci	42
Obrázek 47 vyplněné pole hodnot pro červenou, modrou a bílou barvu	43
Obrázek 48 Připojení výstupu Image Out a jeho vizualizace v rozhraní Front Panel	43
Obrázek 49 Nastavení detekce tvaru vlevo Block Diagram, vpravo Front Panel	44
Obrázek 50 Vytvoření cesty k souboru v rozhraní Block Diagram	44
Obrázek 51 Vlevo složení 1D pole hodnot, vpravo získání první složky pole	45
Obrázek 52 Získání potřebných informací ze tří svazků hodnot výsledků obrazové inspekce	45
Obrázek 53 Ukončení For Loop smyčky při kladném výsledku detekce	46
Obrázek 54 Vizualizace hodnot SCORE	46
Obrázek 55 Část blokového schématu inspekčního programu s popisky	47
Obrázek 56 Přepočítání souřadnic, vlevo posunutí os do středu otáčení robota, vpravo přepočítání do cylindrických souřadnic	47
Obrázek 57 Vlevo Case Structure pro přepočítání souřadnic, vpravo rozbalení clusteru se souřadnicemi x a y a přičtení konstanty k hodnotě y	48
Obrázek 58 Blokové schéma Case Structure vlevo poloha True, vpravo poloha False	49
Obrázek 59 Přepočítání souřadnic x a y na souřadnice R a ϕ	49
Obrázek 60 Vlevo realizace zajištění nulových hodnot pro tvar KRUH, vpravo součet úhlu α a úhlu ϕ	50
Obrázek 61 Zobrazení textových řetězců	50
Obrázek 62 Vlevo While Loop smyčka pro sdílení proměnných, vpravo strom projektu s vytvořenou knihovnou proměnných	51
Obrázek 63 Část programu pro získání vstupu Button0 a nástroje v něm použité	52
Obrázek 64 Blokové schéma konečného propojení programu se zjednodušenými částmi hlavního programu s Vision Assistant, Vision Acquisition a smyčky pro zápis dat na síť	53
Obrázek 65 Vlevo nástroje pro tvorbu Front Panelu, vpravo záložka pro uspořádání bloků	54
Obrázek 66 Uspořádání Front Panelu	54
Obrázek 67 Vlevo nastavení jako aplikace spouštěná po zapnutí, vpravo vytvoření aplikace v paměti zařízení NI myRIO	55
Obrázek 68 Výsledky detekce modrého čtvercového dílce zobrazené na Front Panelu	55
Obrázek 69 Fotografie laboratorního robota s kamerou, stojanem, třemi dílci a zařízením NI myRIO	56

Seznam tabulek

Tabulka 1 Kusovník.....	32
Tabulka 2 Configure Acquisition Settings	36

Seznam příloh

Uživatelský návod

CD obsahující:

- 3D modely dílců
- program s komentáři
- uživatelský návod