



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

## KOLABORATIVNÍ ROBOT S INTEGROVANÝM KAMEROVÝM SYSTÉMEM

COLLABORATIVE ROBOT WITH INTEGRATED CAMERA SYSTEM

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Miroslav Doseděl

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Barnabás Dobossy

BRNO 2019

# Zadání bakalářské práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	<b>Miroslav Doseděl</b>
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	<b>Ing. Barnabás Dobossy</b>
Akademický rok:	2019/20

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

## **Kolaborativní robot s integrovaným kamerovým systémem**

### **Stručná charakteristika problematiky úkolu:**

Kolaborativní roboty jsou v dnešní době stále větším trendem. Tyto roboty spolupracují s člověkem, pomáhají mu při vykonání úkolů, které vyžadují vysokou a neustále stejnou přesnost. Díky své univerzalitě mají tyto roboty široké uplatnění v aplikacích jako svařování, lepení a umístování předmětů. Tato práce vznikla v souladu s tímto trendem a zaměřuje se na zkoumání možnosti ovládní kolaborativního robotu Techman s pomocí PC. Výstupem práce bude ukázková úloha přemístování předmětů s využitím zpracování obrazu.

### **Cíle bakalářské práce:**

1. Seznamte se s kolaborativním robotem Techman.
2. Zjistěte možnosti jeho ovládní a naprogramování z PC.
3. Otestujte možnosti kamery a funkcí pro zpracování obrazu.
4. Naprogramujte ukázkovou úlohu využívající zpracování obrazu z kamery a přemístování předmětů.

### **Seznam doporučené literatury:**

VALÁŠEK, Michael. Mechatronika. Vydavatelství ČVUT, 1995. ISBN 80-01-01276-X.

GREPL, Robert. Kinematika a dynamika mechatronických systémů. Brno: Akademické nakladatelství CERM, 2007, 158 s. : il. ; 26 cm. ISBN 978-80-214-3530-8.

STARCK, Jean-Luc, Albert BIJAOUI a Fionn D MURTAGH. Image Processing and Data Analysis. Cambridge: Cambridge University Press, 2011. ISBN 9780521590846.

PRATT, William K. Digital image processing: PIKS Scientific inside. 4th ed. Hoboken: Wiley-Interscience, 2007, xix, 782 s. : obr., čb. fot., tabulky, grafy + 1 CD-ROM. ISBN 978-0-471-76777-0.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne

L. S.

---

prof. Ing. Jindřich Petruška, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

# **ABSTRAKT**

Tato bakalářská práce se zabývá kolaborativním robotem TM 5-900 s integrovanou kamerou od firmy Techman. Zjišťuje způsoby, jakými ho lze programovat, možnosti jeho využití, testuje zpracování obrazu za pomoci integrované kamery a následně představuje tyto funkce na ukázkové úloze.

# **ABSTRACT**

This Bachelor thesis deals with the collaborative robot TM 5-900 with an integrated camera from Techman. It detects the ways in which it can be programmed, the possibilities for its use, tests image processing using an integrated camera, and then presents these functions in a demonstration role.

# **KLÍČOVÁ SLOVA**

Kolaborativní robot, integrovaná kamera, Techman, ZLÍN ROBOTICS, TM 5-900, automat na lízátka

# **KEYWORDS**

Collaborative robots, integrated camera, Techman, ZLÍN ROBOTICS, TM 5-900, lollipop machine

# **BIBLIOGRAFICKÁ CITACE**

DOSEDĚL, Miroslav. *Kolaborativní robot s integrovaným kamerovým systémem*. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/125006>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Barnabás Dobossy.

# ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci na téma „Kolaborativní robot s integrovaným kamerovým systémem“ vypracoval samostatně s použitím pramenů a odborné literatury uvedených v seznamu, který tvoří přílohu této práce.

V Brně dne 24. června 2020

.....  
Miroslav Doseděl

# PODĚKOVÁNÍ

Rád bych poděkoval svému vedoucímu práce Ing. Barnabáši Dobossy za cenné rady, jeho vedení a konzultace. Dále bych chtěl poděkovat Petrovi Vizvárymu za jeho pomoc.

# OBSAH

<b>ÚVOD .....</b>	<b>9</b>
<b>CÍLE PRÁCE.....</b>	<b>10</b>
<b>1. KOLABORATIVNÍ ROBOTY.....</b>	<b>11</b>
<b>2. TECHMAN TM 5-900 .....</b>	<b>13</b>
2.1. POPIS RAMENA .....	13
2.2. POPIS UNIVERZÁLNÍHO MOBILNÍHO STOJANU.....	16
2.3. BEZPEČNOST .....	17
2.4. PŘIPOJENÍ ROBOTU .....	19
<b>3. TMFLOW .....</b>	<b>20</b>
3.1. PRÁCE S PROGRAMEM.....	20
3.2. ZÁKLADNÍ FUNKCE.....	21
3.3. VISION.....	27
3.4. KALIBRACE A POLOHOVÁNÍ ROBOTU .....	32
<b>4. UKÁZKOVÁ ÚLOHA – AUTOMAT NA LÍZÁTKA .....</b>	<b>34</b>
4.1. CHARAKTERISTIKA ÚLOHY .....	34
4.2. CHAPADLO .....	34
4.3. ZÁSOBNÍK LÍZÁTEK .....	35
4.4. MANIPULACE S ŽETONY .....	36
4.5. POPIS PROGRAMU ROBOTU A ARDUINA .....	37
4.5.1. Zkoumání žetonu .....	38
4.5.2. Braní a podání lízátko .....	39
4.6. ARDUINO .....	41
4.6.1. Zapojení .....	42
4.6.2. Komunikace s robotem a obsluhou .....	42
4.6.3. Počítání lízátek .....	44
<b>ZÁVĚR .....</b>	<b>45</b>
<b>SEZNAM POUŽITÝCH ZDROJŮ.....</b>	<b>47</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>49</b>



# ÚVOD

Kolaborativní roboty se v posledních letech stávají běžnou součástí průmyslové výroby. Na rozdíl od běžných průmyslových robotů mohou pracovat v těsné blízkosti člověka, dokonce s ním i přímo spolupracovat. Ve výrobních sektorech se kolaborativní roboty používají pro ulehčení práce, při monotónních činnostech, pro zvýšení přesnosti výroby, zefektivnění výrobních procesů a vykonávání nebezpečných úkonů.

Robot, který byl v této práci využit, byl zapůjčen od firmy ZLÍN ROBOTICS. Jednalo se o kolaborativní robot TM-5 900 s integrovanou kamerou, který byl vybrán pro zjištění jeho možností, schopností a způsobů programování.

Při zkoumání možností robotu lze postup v této práci shrnout do následujících bodů:

- seznámení se s robotem, zjištění možností jeho programování, ovládání a osvojení si programovacího prostředí,
- přezkoumání možností kamery, jednotlivých nastavení a způsobů, které lze využít pro detekci objektů, rozpoznávání barvy a čtení QR kódů,
- aplikace získaných dovedností na ukázkové úloze.

Pro představení schopností robotu byla navržena ukázková úloha. Její podstatou je vytvořit z robotu „automat na lízátko“. Robot by měl ukázat svoji schopnost uchopit a přesunout předmět (lízátko), rozpoznat za pomoci kamery vzor (správný žeton), nalézt objekt (zásobník lízátek) a díky značce (TM Landmark) umístěné na zásobníku zjistit přesnou polohu předmětů (lízátek). Robot má navíc několik zajímavých funkcí, které byly využity v ukázkové úloze. Příkladem těchto funkcí byla schopnost robotu promluvit nebo signalizovat různé stavy, ve kterých se nachází, pomocí identifikačního světelného kroužku.

# CÍLE PRÁCE

Hlavním tématem bakalářské práce bylo vytvořit demonstrativní ukázkovou úlohu, na které byly znázorněny schopnosti robotu, jeho základní funkce a vybavení.

Zadání bakalářské práce obsahovalo i další cíle, a to seznámení se s kolaborativním robotem a jeho ovládáním, možnostmi jeho programování přes PC a otestování funkcí kamery.

Mezi cíle, které se objevily v průběhu tvoření ukázkové úlohy, patří vytvoření zásobníku lízátek a prstů koncového efektoru, vymyšlení způsobu odstranění žetonů a nutnost zajištění komunikace mezi robotem a obsluhou robotu, například z důvodu doplnění zásobníku lízátek.

# 1. Kolaborativní roboty

V následující kapitole jsou popsány průmyslové a kolaborativní roboty, rozdíly mezi nimi a jejich využití.

Průmyslové roboty slouží k automatizaci a zefektivnění průmyslových procesů. Roboty nahradily lidskou pracovní sílu hlavně ve velkých sériových výrobnách, kde je zapotřebí velká síla, preciznost a rychlost. Dokáží vykonávat monotónní nebo nebezpečnou činnost bez vykazování známek únavy, polevení pozornosti nebo znučení, což jsou typické lidské vlastnosti při vykonávání tohoto typu práce. Častokrát se jedná o velké a těžké stroje, jejichž pohyby mají vysokou rychlost a musí být pro zajištění bezpečnosti odděleny od svého okolí. Roboty mají širokou oblast využití a uplatnění, od zemědělství přes stavebnictví, těžký průmysl, potravinářství, výrobu až po lékařství. [1]

Pro práci v blízkosti lidí jsou určeny kolaborativní roboty (ve zkratce se označují jako koboty), které mohou přijít do vzájemného kontaktu. Kolaborativní roboty se liší od průmyslových robotů ve způsobu zabezpečení, oblasti použití a přípravě k práci.

## Zabezpečení

Jedním z hlavních rozdílů mezi kolaborativními a průmyslovými roboty je, že kolaborativní roboty mohou pracovat v blízkosti člověka. Na rozdíl od průmyslových robotů nemusí být odděleny od lidské obsluhy za pomoci oplocení nebo optických závor, které by při porušení zastavily práci robotu. Protože mají roboty obecně velkou sílu, podléhají přísným bezpečnostním předpisům. Musí splňovat normu ČSN EN ISO 10218-2, podle které existují čtyři režimy provozní spolupráce (viz obrázek 1):



Obrázek 1 – Režimy provozní spolupráce [16]

- **Ruční vedení** znamená, že se robot učí prostřednictvím zapamatování si pohybů, když je veden obsluhou.
- **Bezpečnostní monitorované zastavení** se spustí, pokud obsluha vstoupí do monitorovaného prostoru, a dojde k řízenému zastavení robotu. Využívá se, pokud je robot oplocen.

- **Monitorování vzdálenosti a rychlosti** je režim, kdy robot nemusí být oplocen a sleduje vzdálenost a rychlost osob, které se pohybují v jeho okolí. Při vyhodnocení bezpečné vzdálenosti obsluhu pracuje normální rychlostí, zatímco při přiblížení osob se jeho rychlost postupně snižuje a při překročení minimální vzdálenosti se robot úplně zastaví.
- **Omezení síly a výkonu** je charakteristické pro kolaborativní roboty. Síla a rychlost robotu je omezena do takové míry, aby robot nemohl obsluhu způsobit zranění.

### **Oblasti použití**

Kolaborativní roboty se zejména používají pro úlohy, které není možné úplně automatizovat, a v činnostech, kde musí robot spolupracovat s člověkem. Zatímco průmyslové roboty jsou využívány pro činnosti vyžadující vysokou přesnost a rychlost, pro nebezpečné práce, manipulaci s těžkými objekty anebo v plně automatizovaných výrobních linkách.

### **Příprava k práci**

Instalace kolaborativních robotů do výroby je podstatně jednodušší a rychlejší než u běžných průmyslových robotů, kde instalace a programování může trvat i několik týdnů. [2]

V současnosti má na trhu s kolaborativními roboty největší zastoupení dánská firma Universal Robots. Nové generace kolaborativních robotů představuje japonská firma FUNUC a německá KUKA. Mezi další významné společnosti, které vytvářejí kolaborativní roboty a expandují do nových regionů díky autorizovaným distributorům, patří Techman Robot (Taiwan) a Doosan Robotics (Jižní Korea). [3]

Na obrázku 2 je vyobrazen kolaborativní robot od firmy Universal Robots.



*Obrázek 2 – Kolaborativní robot UR5e od firmy Universal Robots [15]*

## 2. Techman TM 5-900

Robot, který byl zapůjčen od firmy ZLÍN ROBOTICS, patří do řady robotů TM od taiwanské společnosti Techman Robotics. Tato řada byla vyvinuta v roce 2015 pro usnadnění a automatizaci jednotvárných úkolů, jako je například vybalování, šroubování či přichystání komponentů. Výhoda kolaborativních robotů řady TM je rychlá instalace, kratší doba programování a nastavování robotu. [4]

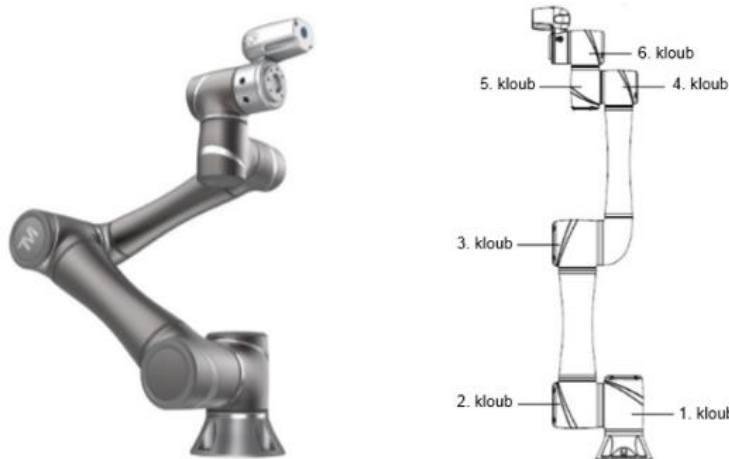
Firma ZLÍN ROBOTICS je českým distributorem robotických technologií pro průmyslovou výrobu. Jedná se například o průmyslové roboty a manipulátory, jednoúčelové stroje, průmyslové robotické buňky, mobilní stojany pro roboty Techman a jiné. [5]

V následujících kapitolách jsou podrobně popsány jednotlivé části robotu Techman TM 5-900, jeho rameno i stojan, který byl vyroben firmou ZLÍN ROBOTICS, bezpečnostní opatření robotu a způsoby, jakými ho lze připojit k počítači a začít programovat.

### 2.1. Popis ramena

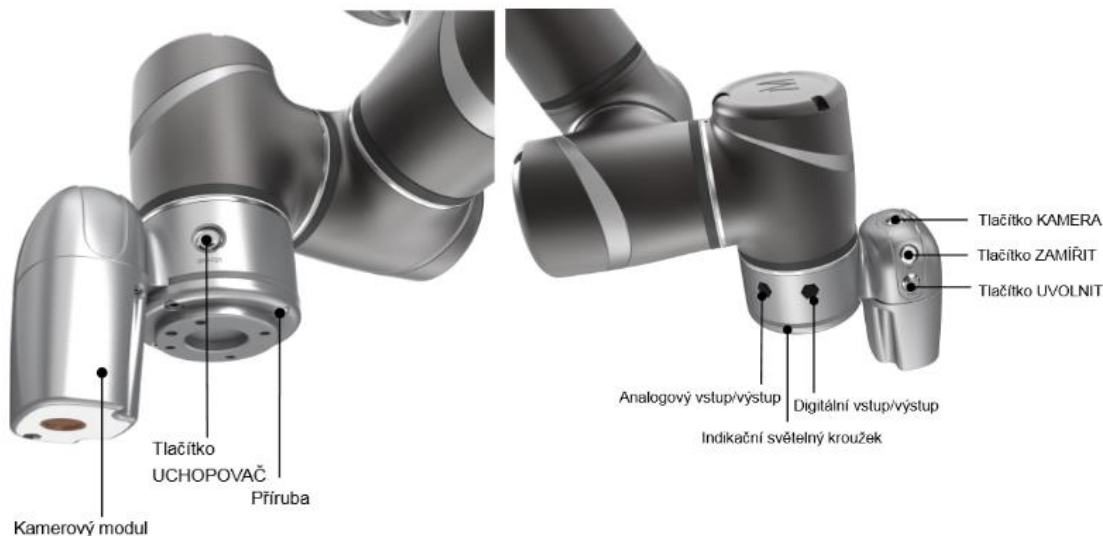
Jedná se o robot s šestiosým pohybem a funkcemi pro omezení výkonu a síly. Hmotnost ramena je 22,6 kg. Jeho dosah, jak už název napovídá, je 900 mm. Unese zátěž až 4 kg. Maximální rychlost, které je schopen dosáhnout, je  $1,4 \text{ m}\cdot\text{s}^{-1}$ .

Na obrázku 3 je znázorněno rozmístění kloubů.



Obrázek 3 – Obrázek ramena a rozmístění jednotlivých kloubů ramena [7]

Rozsah natočení kloubů je následující: pro klouby 1 a 6 je  $\pm 270^\circ$ , pro klouby 2, 4 a 5 je  $\pm 360^\circ$  a pro kloub 3 je  $\pm 155^\circ$ . Na obrázku 4 je zobrazeno zakončení ramena.



Obrázek 4 – Popis zakončení ramena [7]

Na konci ramena se nachází místo pro připojení koncového efektoru, kamera, identifikační světelný kroužek a několik tlačítek. Tlačítka mají různé funkce, které jsou popsány v tabulce 1.

Tabulka 1 – Popis funkcí jednotlivých tlačítek na ramenu robotu

Název tlačítka	Funkce
<i>Vision</i>	vytvoří v programu uzel <i>Vision</i> sloužící k práci s kamerou
<i>Point</i>	vytvoří v programu uzel <i>Point</i> nesoucí záznam polohy robotu (úhly natočení jednotlivých kloubů), ve kterém se robot právě nachází
<i>Gripper</i>	tlačítko je propojené s koncovým efektozem, při jeho stisknutí se koncový efektor, pokud je to chapadlo, uzavře nebo otevře, v programu se vytvoří odpovídající uzel
<i>Free</i>	tlačítko uvolní klouby robotu a umožní manuální změnu jeho polohy

Na koncový efektor ramena se dá připojit chapadlo Schunk, Robotiq anebo chapadlo od dodavatele robotické aplikace (integrátora).

Podle toho, jakým způsobem identifikační světelný kroužek svítí (jestli stále svítí jednou barvou, nebo bliká mezi různými barvami), lze zjistit, ve kterém stavu se robot nachází anebo druh chyby, která právě nastala. Například, pokud svítí zelená barva, znamená to, že robot běží v ručním režimu, modrá naopak znamená běh v automatickém režimu. Blikání mezi červenou a zelenou barvou označuje chybu v ručním režimu. Všechny další možnosti jsou vypsány v manuálu k robotu *Příručka k softwaru TMflow* [6].

Zabudovaná barevná kamera na konci ramena má rozlišení 1,2 Mpx/5 Mpx a je vybavena možností světelného přisvícení pracovní plochy.

K Univerzálnímu Mobilnímu Stojanu je připevněn ovladač s tlačítky sloužícími k řízení robotu a světelnými indikátory napájení a režimu. Ovladač robotu je zobrazen na obrázku 5. Funkce tlačítek na ovladači robotu jsou uvedeny v tabulce 2.

Tabulka 2 - Popis funkcí jednotlivých tlačítek na ovladači robotu

Název tlačítka	Funkce
Nouzový vypínač	spouští nouzové zastavení robotu
Napájení	zapne a vypne robot
Přepínání režimů M/A	přepíná mezi režimem manuálním a automatickým
+/-	mění rychlost pohybu robotu při běhu programu
Spustit/Pozastavit	spustí a následně pozastaví program
Zastavit	zastaví, ukončí běh programu



Obrázek 5 – Ovladač robotu [7]

Robot může pracovat ve dvou režimech - ručním a automatickým. Aktuálně zvolený režim je barevně znázorněn na identifikačním kroužku robotu.

V ručním režimu lze robot ovládat pomocí *Controlleru* (kapitola 5.1) v programu TMflow a ovladačem robotu. V tomto případě je rychlost omezena na maximální hodnotu  $250 \text{ mm.s}^{-1}$ . Dále je možné robotem manipulovat stisknutím a podržením tlačítka *Free* na konci ramena a následnou změnou jeho polohy. Při spuštění projektu je rychlost pohybu robotu automaticky nastavena na 10 % rychlosti projektu. Při běhu projektu se může rychlost robotu snižovat nebo zvyšovat po 5 % (stisknutím tlačítka +/- na ovladači), ale pohyb robotu bude nadále omezen maximální rychlostí  $250 \text{ mm.s}^{-1}$ .

Při spuštění automatického režimu robot pracuje podle programu, který byl zvolen pro tento režim. Při běhu programu je rychlost robotu rovna rychlosti projektu. Stisknutím tlačítek *Play* a *Pause* lze program robotu spustit nebo pozastavit. Tlačítko *Free* v automatickém režimu nefunguje. [7]

## 2.2. Popis Univerzálního Mobilního Stojanu

Univerzální Mobilní Stojan (obrázek 6) slouží pro snadnou manipulaci a připojení robotu na jakémkoliv místě. Stabilita stojanu je zajištěna závažím na spodní části a třemi podpůrnými nohami, které se podle potřeby vyšroubují. Tento stojan se skládá z několika částí, které jsou následně popsány.



Obrázek 6 – Robot s mobilním stojanem [17]

**Deska pro umístění robotu** slouží pro upevnění robotu.

Na **rozvaděči elektro částí** (obrázek 7) je umístěn hlavní napájecí konektor pro napájení ze sítě 230 V, hlavní vypínač celého robotu, konektory digitálních vstupů a výstupů. Konektory byly speciálně složeny pro připojení k robotu. Je na nich vyvedeno všech 16 digitálních vstupů a výstupů. Jako logická jednička se bere napětí 24 V. Pro připojení dalších periférií robotu slouží kombinovaný komunikační konektor (USB, LAN, EtherCAT, GigE, HDMI a napájení pro externí kameru). Další tři USB konektory se nachází uvnitř v rozvaděči. Na vrchní straně stojanu jsou umístěné konektory pro chapadlo Schunk a Robotiq, připojení vzduchu, USB a LAN konektor. Konektory USB lze použít pro připojení hardwarových klíčů, které slouží pro odemknutí nadstavbových funkcí robotu, jako je využití externí kamery, funkcí pro měření a jiné. [8]





Obrázek 7 – Univerzální Mobilní Stojan zepředu (elektrorozvaděč)

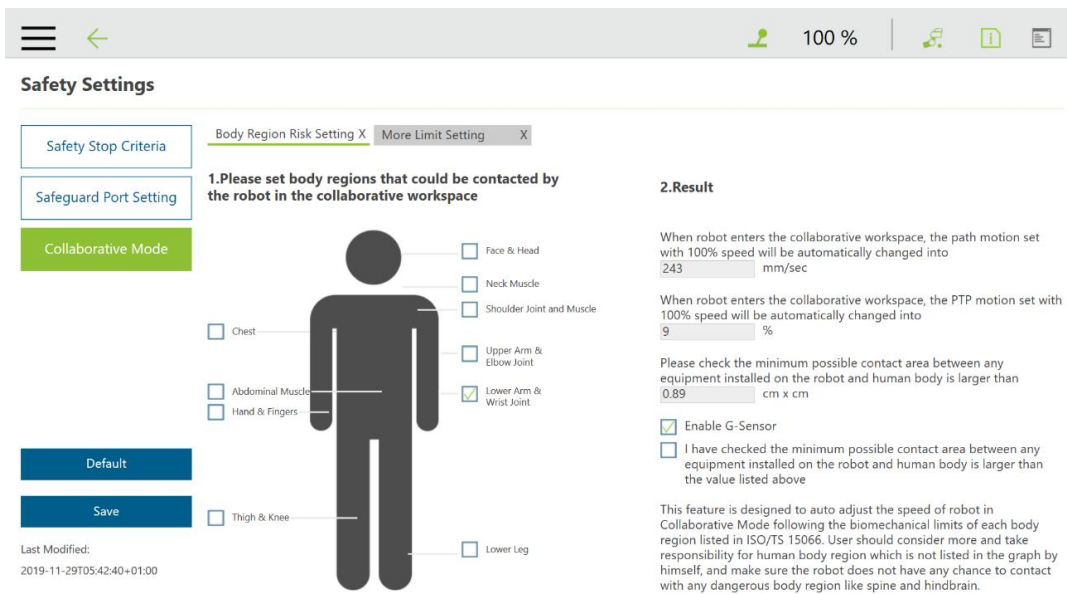
## 2.3. Bezpečnost

Na úroveň bezpečnosti robotu má vliv to, v jakém režimu se robot právě nachází. Robot může pracovat v průmyslovém nebo kolaborativním režimu (robot spolupracuje s člověkem nebo pracuje v jeho blízkosti). Kolaborativní robot má sníženou rychlost a menší točivý moment, což zvyšuje bezpečnost při práci v blízkosti člověka. Tělo robotu je úmyslně tvořeno pouze oblými tvary, nemá žádné ostré hrany, aby i v případě kontaktu s člověkem nedošlo k vážnému úrazu. K dalším opatřením při kontaktu robotu s překážkou či osobou patří nadproudová ochrana, kterou má robot umístěnou v každém kloubu. Při nárazu se zvýší proud a robot se zastaví.

Robot má několik dalších bezpečnostních funkcí, které nezávisí na režimu, ve kterém se nachází. K zastavení robotu dojde v následující situacích:

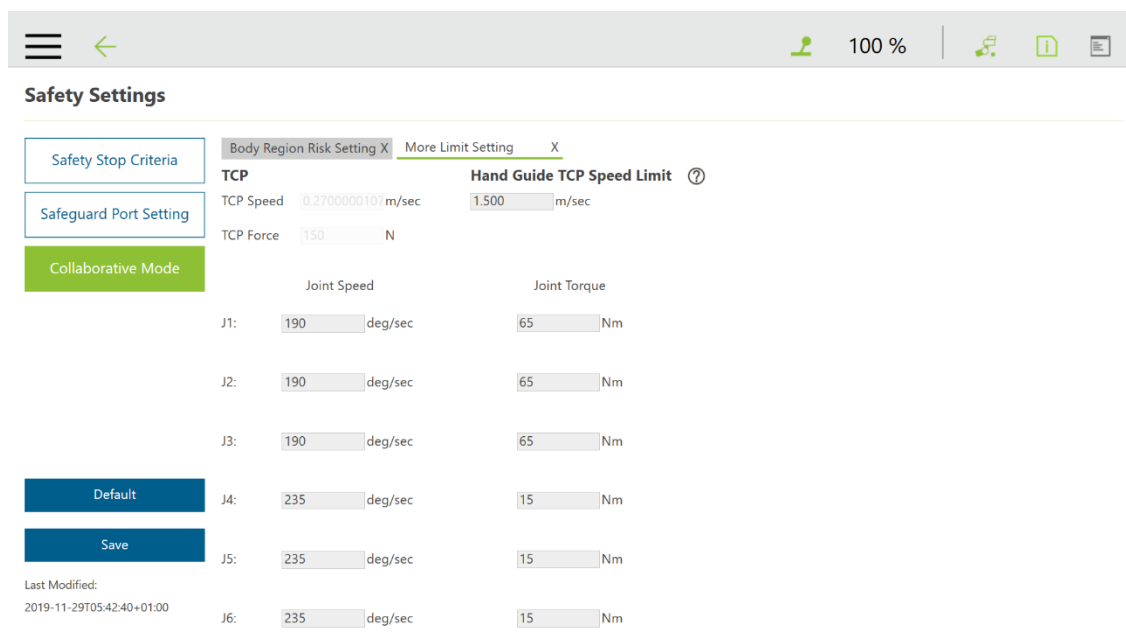
- stisknutím tlačítka nouzového zastavení na ovladači robotu nebo externího nouzového tlačítka připojeného k robotu,
- při překročení maximální hodnoty točivého momentu v kloubech,
- při překročení limitní hodnoty úhlu pohybu jednotlivých kloubů,
- při překročení nastaveného rozsahu rychlostí kloubů,
- při překročení limitní rychlosti koncového bodu nástroje,
- při překročení nastavené hodnoty mezní síly koncového bodu nástroje (tato síla je odhadnuta za pomoci systému robotu).

V nastavení robotu, přesněji v nastavení bezpečnosti a při práci v kolaborativním režimu, lze vybrat (zaškrtnout) rizikové části těla (zobrazeno na obrázku 8), se kterými může robot přijít do kontaktu. Na pravé straně v tomto nastavení se zobrazí vypočtená maximální rychlost robotu v kolaborativním režimu, která se přizpůsobuje podle toho, která políčka byla zvolena. Výslednou hodnotu rychlosti lze změnit, ale pouze snížit.



Obrázek 8 – Nastavení bezpečnosti robotu, záložka *Body Region Risk Settings*

Podrobnější nastavení limitních hodnot maximálních rychlostí a točivých momentů jednotlivých kloubů se nachází v záložce *More Limit Settings*, znázorněno na obrázku 9. Tyto hodnoty je možné pouze snížit. [9]



Obrázek 9 – Nastavení bezpečnosti robotu, záložka *More Limit Settings*

Existuje několik druhů zastavení, rozdělených do kategorií podle závažnosti situace:

- zastavení **kategorie 0** znamená zastavení stroje přerušením přívodu elektrické energie do motorů, v činnosti zůstávají pouze součásti pro zabrzdění stroje (brzdy),
- zastavení **kategorie 1** se provádí sestupnou hranou signálu, jedná se tedy o kontrolované zastavení,
- při zastavení **kategorie 2** zůstávají akční členy pod napětím. [10]

V případě nouzového zastavení, stisknutím tlačítka nouzového zastavení na ovladači robotu, dojde k zastavení kategorie 1. Při překročení limitních hodnot (maximální hodnota točivého momentu, natočení kloubu, rychlost koncového bodu atd.) nastane zastavení kategorie 2. Když nastane zastavení kategorie 2, spustí se funkce zablokování kodéru a při jakémkoliv náhodném pohybu robotu, se aktivuje zastavení kategorie 0. [9]

## 2.4. Připojení robotu

Robot může být ovládán i programován vždy pouze jedním uživatelem. K tomu, aby bylo možné programovat robot či změnit jeho nastavení, je zapotřebí nejdříve připojit počítač k robotu. Je možné toho docílit čtyřmi způsoby:

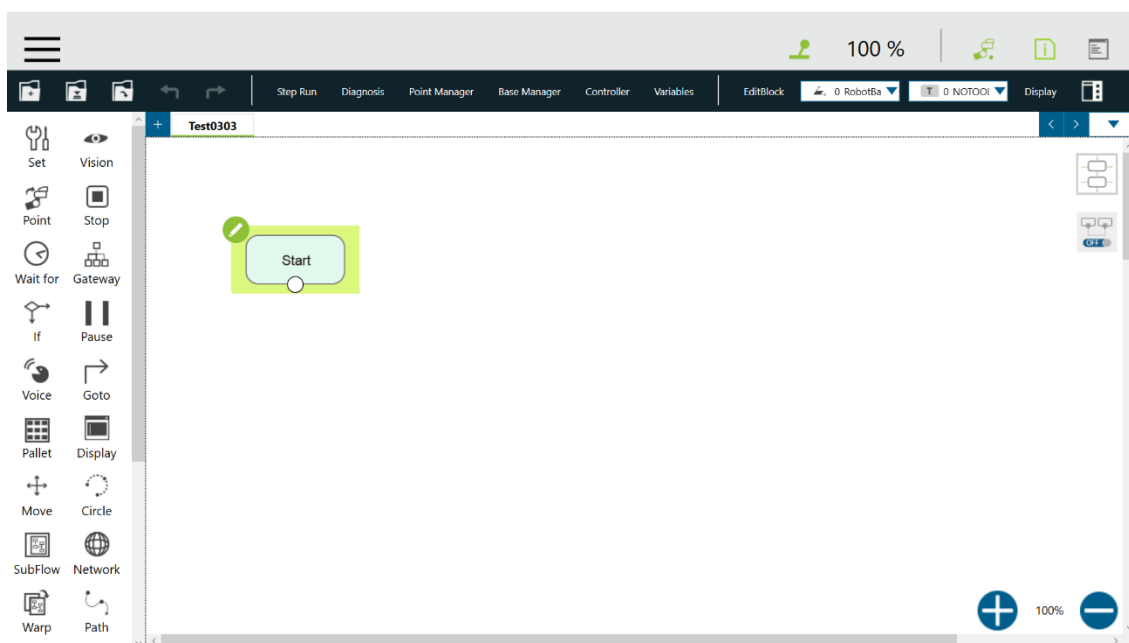
- připojení monitoru, klávesnice a myši přímo k ovládací skříňce robotu umožní spustit program TMflow v robotu bez připojení vnějšího počítače,
- nainstalováním klientské verze programu TMflow na počítač s operačním systémem Windows a připojením počítače a robotu ke stejnému fyzickému přístupovému bodu,
- nainstalováním klientské verze programu TMflow na počítač s operačním systémem Windows a připojením počítače k robotu přes síťový kabel,
- v případě omezeného přístupu k robotu, existuje možnost programovat offline, tzn. programovat bez připojení robotu, tato funkce je však zpoplatněna a je zapotřebí hardwarový klíč, který odemkne programování v režimu offline. [6]

## 3. TMflow

V následujících kapitolách je popsán software TMflow, který tvoří jedinou možnost, jak naprogramovat robot. Je objasněno programovací prostředí, jednotlivé funkce programu, funkce pro práci s kamerou a způsoby polohování a kalibrace robotu. Jedná se o grafické programování, ve kterém jde o spojování bloků, které lze spojovat vodiči. Vodič se vytvoří kliknutím do kroužku u bloku (funkce) a potáhnutím myši na následující blok.

### 3.1. Práce s programem

Obrázek 10 zobrazuje, jak vypadá nově vytvořený projekt. Každý projekt obsahuje uzel *Start*, na který se následně připojují další funkce, uzly. Na horní liště se nachází tlačítka pro vytvoření nového programu, uložení programu nebo jeho načtení.



Obrázek 10 – Nově vytvořený projekt

V tabulce 3 jsou popsány nejdůležitější záložky a popis jejich funkce.

Tabulka 3 – Popis funkcí některých záložek

Název záložky	Funkce
<i>Step run</i>	spouštění programu od určitého místa se sníženou rychlostí
<i>Point Manager</i>	obsahuje všechny uložené body a umožňuje jejich změnu
<i>Base Manager</i>	obsahuje všechny vytvořené souřadné systémy, ve kterých robot pracuje
<i>Controller</i>	ovládání robotu, natáčení jednotlivých kloubů, pohyb koncového efektoru ve všech směrech, ovládání digitálních výstupů
<i>Variables</i>	obsahuje všechny vytvořené proměnné a umožňuje jejich změnu
<i>Edit Block</i>	kopírování, vkládání a mazání jednotlivých bloků (uzly, funkce) v programu

Vedle záložky *Edit Block* se nachází výběr souřadných systémů a koncových efektorů. [6]

## 3.2. Základní funkce

Program TMflow zahrnuje jak funkce pro manipulaci s robotem, tak funkce pro práci s kamerou. V této kapitole, jsou popsány nejdůležitější funkce pro ovládání robotu, funkce pro práci s kamerou jsou popsány v kapitole 5.3.

Přehled jednotlivých funkcí:

### 1. *Point* (obrázek 11)

Funkce *Point* vytváří bod, který obsahuje natočení jednotlivých kloubů, a tedy i polohu koncového efektoru ve zvoleném souřadném systému. Bod může být vytvořen dvěma způsoby:

- přetáhnutím funkce *Point* do programu,
- stisknutím tlačítka *Point* na robotu.

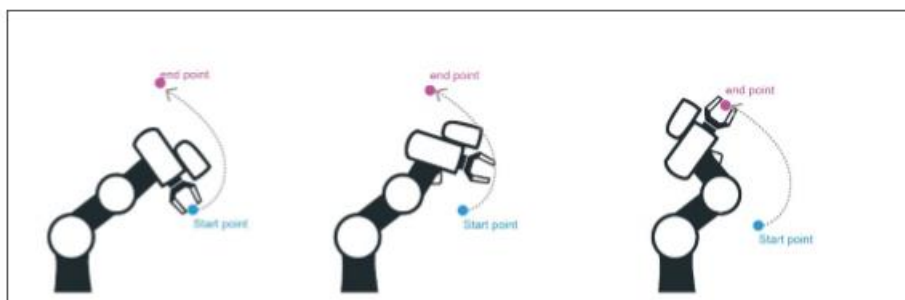


Obrázek 11 – Ikona funkce *Point*

Způsob pohybu, kterým se robot dostane do daného bodu, je možné vybrat ze tří možností.

- **PTP (Point-to-Point)**

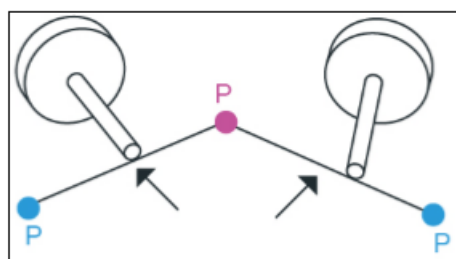
Jedná se o nejrychlejší způsob pohybu. Robot se přesune do cílového bodu po nejkratší a nejvýhodnější dráze (z hlediska natočení kloubů). Na obrázku 12 je příklad pohybu PTP.



Obrázek 12 – Znázornění pohybu typu PTP [6]

- **Line**

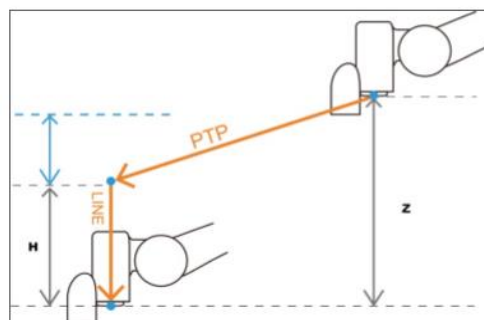
Koncový efektor se přesune do cílového bodu po přímce, což je zobrazeno na obrázku 13.



Obrázek 13 – Znázornění pohybu typu Line [6]

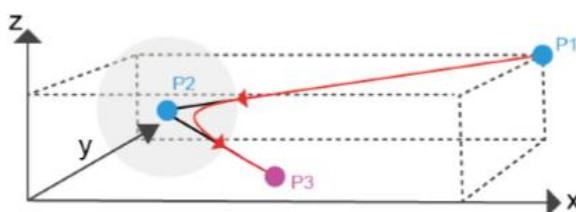
- **Waypoint**

Koncový efektor se přemístí do určené výšky nad cílovým bodem pomocí pohybu PTP a následně pohybem typu Line sestoupí dolů ve směru osy z. Tento pohyb je ukázán na obrázku 14.



Obrázek 14 – Znázornění pohybu typu Waypoint [6]

Pro dosažení plynulosti pohybu koncového efektoru je vhodné použít nastavení *Blending*, které nastaví určitý bod průchozím. Bez nastavení *Blending* by robot do zvoleného bodu dojel a z něj by následně pokračoval do koncového bodu (například při vyhnutí se překážce). Při použití tohoto nastavení se robot blíží k průchozímu bodu, tímto bodem však neprojde, ale obloukem pokračuje do koncového bodu. Použití nastavení *Blending* je předvedeno na obrázku 15.



Obrázek 15 – Ukázka nastavení *Blending* v prostoru [6]

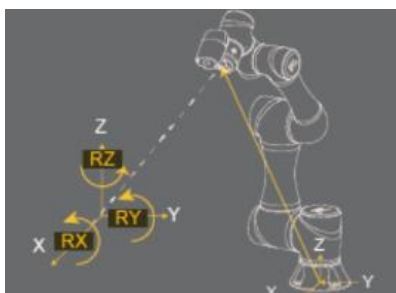
## 2. *Move* (obrázek 16)



Move

Obrázek 16 – Ikona funkce *Move*

Umožňuje změnu polohy koncového efektoru v určitém souřadnicovém systému (obrázek 17) pomocí nastavení posunu v milimetrech v osách  $x$ ,  $y$ ,  $z$  a natočení  $rx$ ,  $ry$ ,  $rz$  ve stupních, nebo nastavením úhlů natočení jednotlivých kloubů.



Obrázek 17 – Znárodnění souřadného systému

## 3. *Circle* (obrázek 18)

Nastavení tří bodů (počátečního bodu, průchozího bodu, koncového bodu) a délky oblouku ve stupních umožní pohyb koncového efektoru po kružnici.



Circle

Obrázek 18 – Ikona funkce *Circle*

#### 4. *Pallet* (obrázek 19)

Jak už název napovídá, tato funkce se používá pro práci s paletou nebo když je nutné použít rozložení předmětů do určitého typu mřížky. Stačí určit tři krajní body, které plně definují její tvar (prvním bodem se určí počátek, druhým bodem šířka a posledním bodem délka mřížky), počet řad a sloupců. Program si následně sám přepočítá polohy jednotlivých pozic.



Pallet

Obrázek 19 – Ikona funkce Pallet

#### 5. *WaitFor* (obrázek 20)

Funkce *WaitFor* pozastaví program, dokud není splněna jedna nebo všechny z nastavených podmínek. Podmínky mohou být typu uplynutí určité doby, stavu digitálních nebo analogových vstupů a výstupů. Program poté pokračuje v běhu.



Wait for

Obrázek 20 – Ikona funkce WaitFor

#### 6. *Set* (obrázek 21)

Nastavuje stavy IO a proměnné. Používá se pro vytvoření proměnných, změny jejich hodnot, nastavení digitálních a analogových výstupů během chodu programu.



Set

Obrázek 21 – Ikona funkce Set

#### 7. *If* (obrázek 22)

Tato funkce pracuje jako funkce *If* v jiných programech. Vyřeší se zadaná podmínka nebo více podmínek. Nastavuje se, jestli má platit alespoň jedna, nebo všechny podmínky. Porovnávání mohou být hodnoty nebo stavy IO. Při splnění podmínky pokračuje program větví *Yes*, jinak pokračuje větví *No*.



If

Obrázek 22 – Ikona funkce If



### 8. *Gateway* (obrázek 23)

Pracuje stejně jako funkce *Switch* v ostatních programovacích jazycích.



Obrázek 23 – Ikona funkce *Gateway*

### 9. *Voice* (obrázek 24)

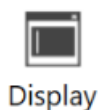
Při použití této funkce robot promluví, přečte text zapsaný v uvozovkách nebo uložený v proměnné typu textového řetězce. Způsoby je možné kombinovat. Lze nastavit, jestli má robot promluvit, a poté pokračovat v programu (*Speak, than Move*), nebo mluvit během vykonávání dalších funkcí (*Speak and Move*).



Obrázek 24 – Ikona funkce *Voice*

### 10. *Display* (obrázek 25)

Slouží k zobrazení proměnných a textů na obrazovce během chodu programu.



Obrázek 25 – Ikona funkce *Display*

### 11. *Pause* (obrázek 26)

Funguje stejně jako tlačítko *Pause* na ovladači robotu.



Obrázek 26 – Ikona funkce *Pause*

**12. Stop (obrázek 27)**

Funguje stejně jako tlačítko *Stop* na ovladači robotu.



Stop

Obrázek 27 – Ikona funkce Stop

**13. Goto (obrázek 28)**

Slouží k přechodu do jiné části programu za běhu.



Goto

Obrázek 28 – Ikona funkce Goto

**14. SubFlow (obrázek 29)**

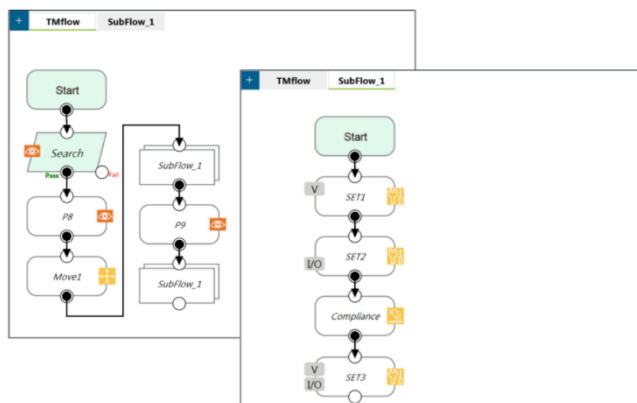
Použitím funkce *SubFlow* je možné vytvořit v programu podfunkci, která se spustí v určité části programu, nebo podfunkci, která pracuje souběžně s programem a při zastavení programu je taktéž zastavena.



SubFlow

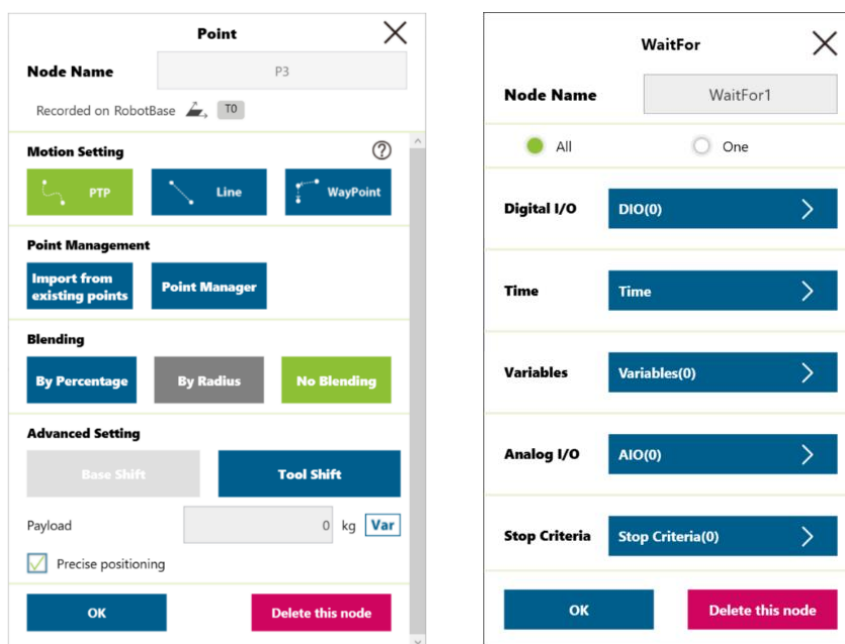
Obrázek 29 – Ikona funkce SubFlow

Používá se u bloků, které se v programu vícekrát opakují. V programu se vytvoří nová záložka, podfunkce bude s původním programem sdílet proměnné a parametry. Další výhodou využití funkce *SubFlow* je zpřehlednění a zjednodušení programu, znázorněno na obrázku 30. [6]



Obrázek 30 – Ukázka funkce SubFlow a její zařazení v programu [6]

U většiny funkcí, kromě *Stop* a *Pause*, se po jejich rozkliknutí otevře detailnější nastavení. Příklad tohoto nastavení pro funkce *Point* a *WaitFor* je uveden na obrázku 31.



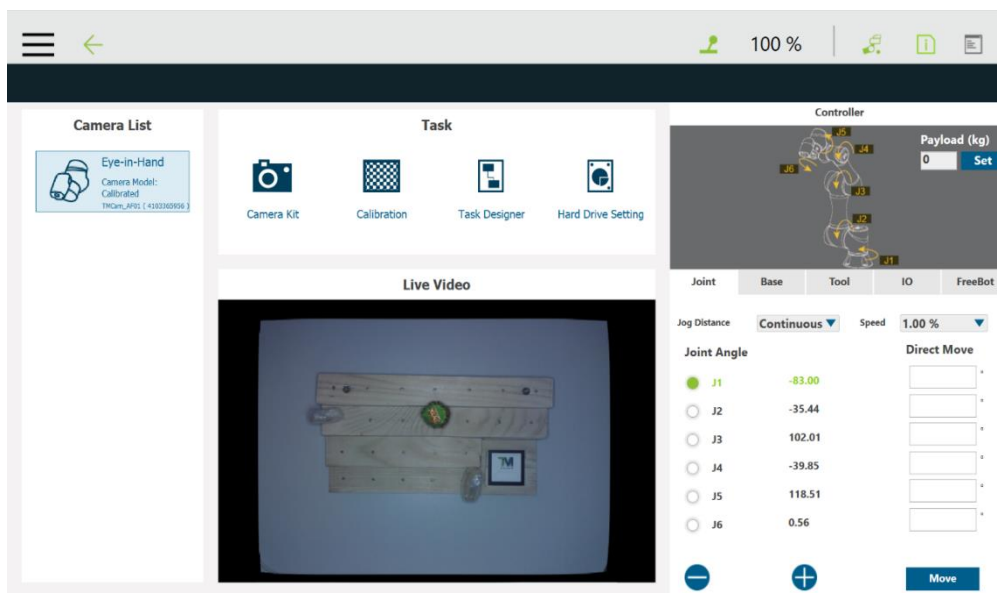
Obrázek 31 – Příklad nastavení funkce *Point* a *WaitFor*

### 3.3. Vision

V této kapitole je popsána funkce *Vision*, nastavení kamery a možnosti zpracování obrazu.

Funkce *Vision*, přesněji *TMvision*, slouží pro práci s kamerou. Obsahuje standardní funkce pro práci s kamerou, pomocí kterých lze nastavit, co má robot rozpoznávat (barvy, předměty, QR a čárové kódy), kam se má přemístit, jak se má natočit a jiné. Další funkce, například pro identifikaci odchylek rozměrů nebo určení počtu předmětů, patří mezi zpoplatněné funkce, ve standardním balíku se nenachází. Vkládá se do programu stejným způsobem jako ostatní funkce, včetně klasického způsobu přetažením ikonky do programu, je možné ji vložit i stisknutím tlačítka *Vision* umístěného na kameře (na konci ramena).

Při rozkliknutí *Vision Job* se objeví následující okno, viz obrázek 32:

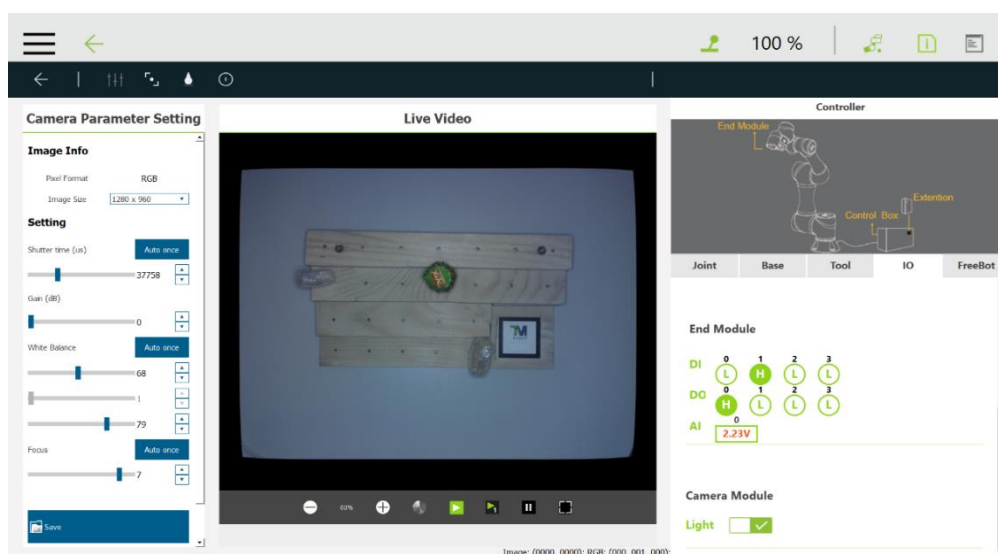


Obrázek 32 – Pracovní okno funkce *Vision Job*

Na levé straně se nachází seznam připojených kamer, napravo je ovladač ramena a uprostřed běží živé video z kamery. Nad ním jsou umístěny čtyři ikony sloužící k nastavení kamery, kalibraci, úlohy (k čemu bude kamera využita) a úložiště.

## Nastavení kamery

Možnost nastavení kamery (obrázek 33) umožňuje manuální i automatickou změnu jednotlivých parametrů od rychlosti závěrky po zaostření kamery.



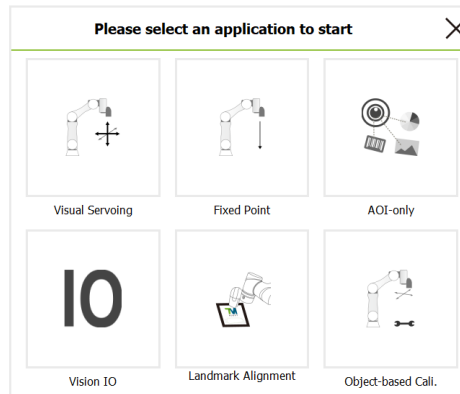
Obrázek 33 – Nastavení kamery

## Kalibrace kamery

Způsoby kalibrace kamery jsou řešeny v kapitole 5.4.

## Výběr úlohy kamery

Podle toho, k čemu má být kamera použita, se vybere jedna z následujících funkcí (obrázek 34):

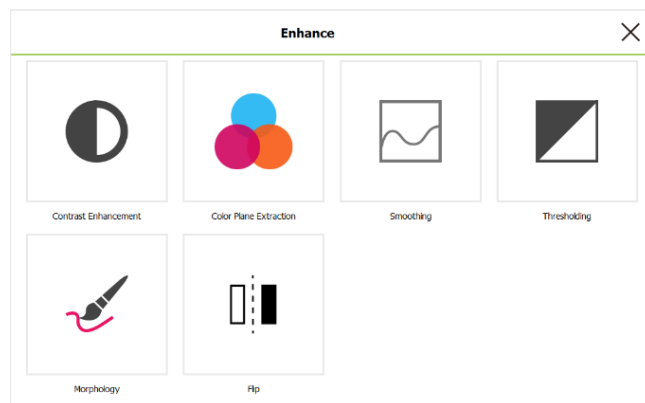


Obrázek 34 – Funkce kamery

- **Visual Servoing** (vizuální polohování) a **Fixed Point** (pevné polohování) jsou popsány v kapitole 5.4,
- **AOI-only** se používá při čtení QR kódu, čárového kódu a k identifikaci barev bez potřeby určení pracovního prostoru, při jeho použití se nevytvoří žádný nový souřadný systém,
- **Vision IO** slouží k zaznamenání změn na sledovaném úseku obrazu,
- **Landmark Aligment** a **Object-based calibration** (kalibrace na základě objektu) se využívají pro kalibraci za pomoci TM Landmarku a na základě objektu, tyto funkce jsou popsány v kapitole 5.4.

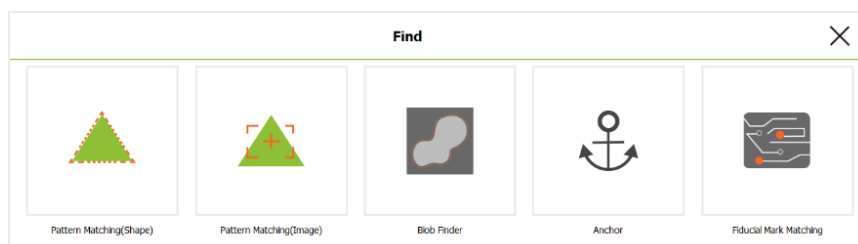
Poté co se vybere jedna funkcí kamery, objeví se na horní liště tři záložky pro úpravu obrazu, detekci předmětů a identifikaci QR kódů a barev.

Záložka **Enhance** (obrázek 35) obsahuje možnosti úpravy obrazu, jako je úprava kontrastu, extrahování specifické barvy, vyhlazení obrazu pomocí odfiltrování šumu, převedení obrazu na černobílý a překlopení.



Obrázek 35 – Možnosti v záložce Enhance

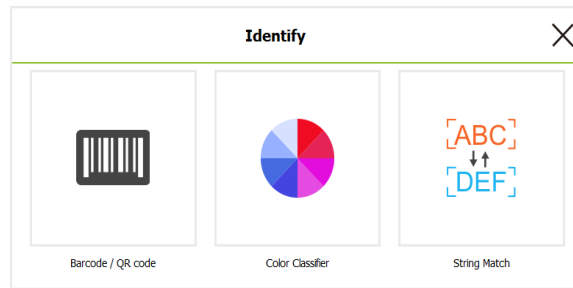
Záložka **Find** (obrázek 36) umožní detekci objektů různými způsoby. Výstupem jsou souřadnice hledaného objektu  $x$ ,  $y$  a úhel natočení  $r$ . Tyto způsoby jsou:



Obrázek 36 – Možnosti v záložce Find

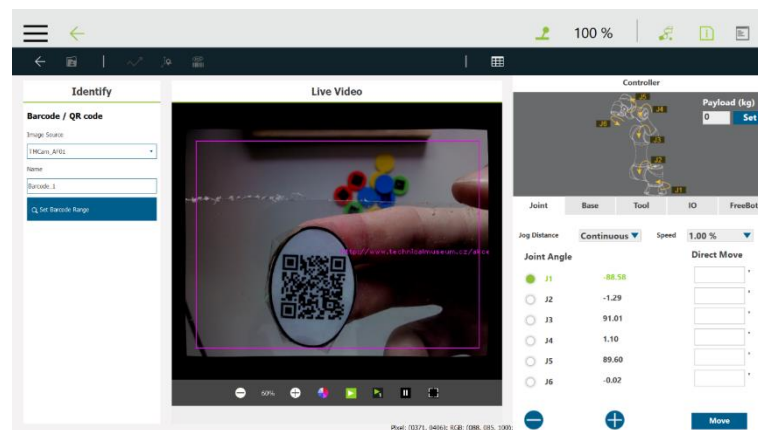
- **Pattern Matching (Shape)** porovnává objekty na základě geometrického tvaru předlohy, objekt může být zvětšený, zmenšený či pootočený,
- **Pattern Matching (Image)** porovnává objekty na základě jejich vlastních obrazů (fotografií), neumožňuje změnu rozměrů a může trvat delší dobu,
- **Blob finder** rozpozná objekt barevným rozdílem mezi objektem a pozadím,
- **Anchor** slouží ke změně pozice a natočení souřadného systému vytvořeného na objektu,
- **Fiducial Mark Matching** využívá dvou společných referenčních značek pro zarovnání objektu a rychlé a přesné nalezení objektu. Používá se pro manipulaci s deskami plošných spojů.

Záložka **Identify** (obrázek 37) obsahuje následující funkce:



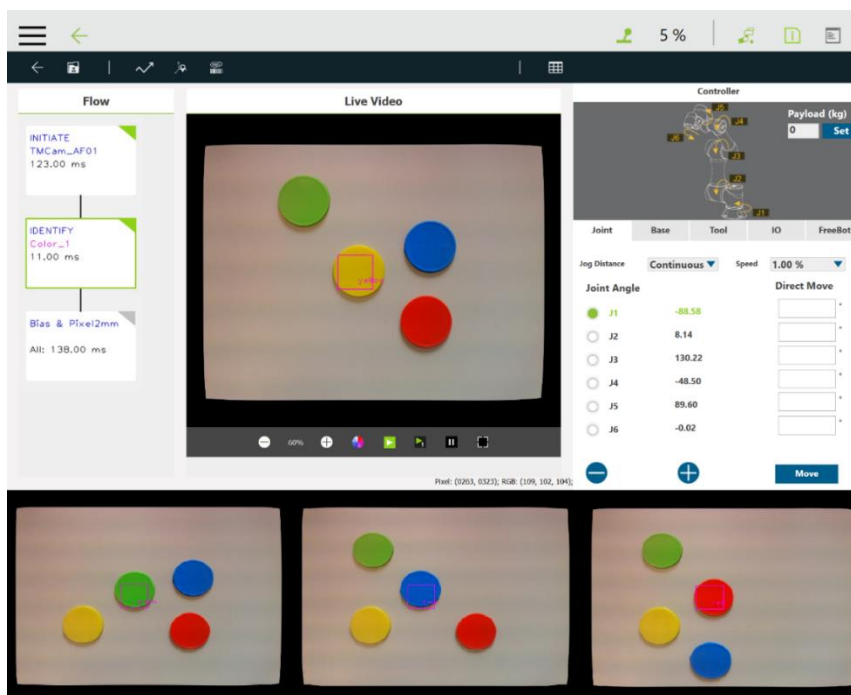
Obrázek 37 – Možnosti v záložce Identify

- **Barcode / QR code** (obrázek 38) slouží ke čtení čárového kódu a QR kódu,



Obrázek 38 – Ukázka čtení QR kód přes kus plexiskla

- **Color Classifier** (obrázek 39) umožňuje rozpoznávání barev určených uživatelem na předem vybrané oblasti,
- **String Match** porovnává textové řetězce. [11]



Obrázek 39 – Ukázka rozpoznávání barev žlutá, zelená, modrá a červená

### 3.4. Kalibrace a polohování robotu

Software TM Flow nabízí čtyři způsoby polohování.

#### 1. TM Landmark (dále jen Landmark)

Landmark (obrázek 40) je malá plastová destička o rozměrech 5x5 cm a tloušťce 0,2 cm. Slouží k polohování robotu. Používá se k 3D kalibraci robotu v prostoru. Body jsou vztaženy relativně k souřadnicím Landmarku. Když se změní relativní poloha mezi robotem a Landmarkem, robot vyfotí Landmark, aby získal informace o jeho nové poloze a natočení. Pozice, které byly nahrány v souřadném systému Landmarku jsou přepočteny do souřadného systému robotu.

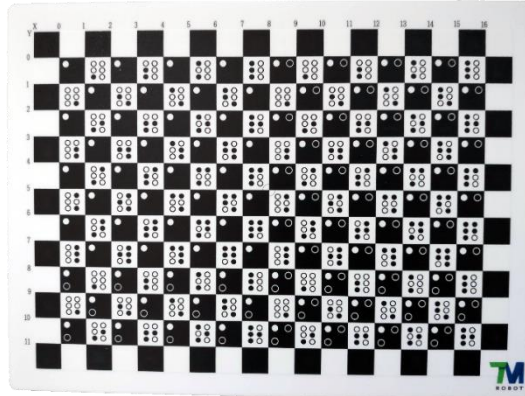


Obrázek 40 – TM Landmark



## 2. Pevné polohování

Tento typ polohování využívá pracovní prostor, který byl vytvořen za pomoci kalibrační desky (obrázek 41). Pracovní prostory se vytvářejí za pomoci automatické nebo ruční kalibrace.



Obrázek 41 – Kalibrační deska

## 3. Vizuální polohování (zpětná vazba)

Při využití vizuálního polohování je poloha ramena automaticky upravována, aby došlo k zarovnání referenčního obrazu pomocí kamery s objektem.

## 4. Kalibrace na základě objektu

Při této kalibraci se robot přemístí tak, aby viděl předmět postupně ve všech čtyřech rozích zorného pole kamery, čímž se definuje pracovní prostor. [11]

## 4. Ukázková úloha – automat na lízátka

Ukázková úloha byla založena na možnostech, které je robot schopen vykonávat. Důraz byl kladen na práci s kamerou a schopnost robotu uchopit a přemístit předměty, zjištění náročnosti programování a způsobu ovládní.

V následujících kapitolách je objasněn princip ukázkové úlohy a její realizace.

### 4.1. Charakteristika úlohy

Hlavním cílem ukázkové úlohy bylo vytvoření takzvaného automatu na lízátka. Princip funkce je daný následujícím popisem:

Člověk vhodí žeton, který dopadne před robot. Ten kamerou zjistí, jestli se jedná o žeton, nebo o něco jiného. Žeton je poté smeten. Robot vezme ze zásobníku lízátka a podá ho uživateli. Člověk z bezpečnostních důvodů nepříjde s robotem do kontaktu.

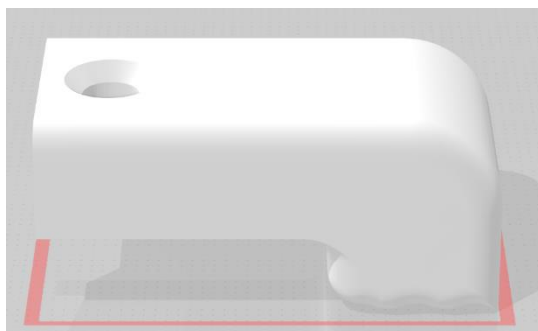
### 4.2. Chapadlo

Aby byl robot schopen uchopit předmět, bylo zapotřebí připevnit na konec ramena chapadlo. Použité chapadlo, vyrobené firmou Schunk (obrázek 42), bylo zapůjčeno společně s robotem. Existuje mnoho variant chapadel, které se liší uchopovací silou a rozevřením čelistí. V tomto případě se jednalo o minimální sílu 35 N, maximální sílu 140 N a rozevření čelistí o 12 mm. [12]



*Obrázek 42 – Chapadlo od firmy Schunk*

K použití chapadla bylo zapotřebí připevnit na jeho čelisti prsty koncového efektoru (obrázek 43), kterými by byl robot schopen uchopit lízátko. Z důvodu různorodosti tvaru lízátek bylo rozhodnuto, že nejvýhodnější bude uchopovat lízátko za jejich tyčku. Jelikož se průměr tyček lízátek pohyboval v rozmezí od 1 do 3 mm, byl vyroben koncový efektor tak, aby byl schopen úplného sevření (aby se jeho dvě poloviny dotkly).



Obrázek 43 – Návrh prstů koncového efektoru

### 4.3. Zásobník lízátek

Úlohou robotu bylo podávání lízátek a pro jejich uskladnění byl navržen zásobník (obrázek 44), ze kterého si robot bral lízátko jedno po druhém.



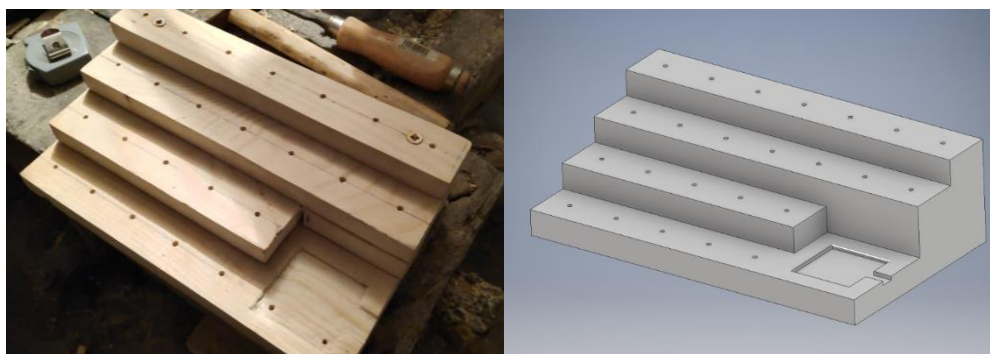
Obrázek 44 – Zásobník s lízátky a Landmarkem

První prototyp zásobníku byl vytvořen z polystyrénu, do kterého byly v pravidelných rozstupech rozmístěny lízátko do dvou řad. Po uchopení lízátko se robotu nepodařilo lízátko uvolnit ze zásobníku a bylo nadzvednuto i se zásobníkem. Otvor pro lízátko se nemohl zvětšit, protože by docházelo k jeho náklonu a robot by nebyl schopen ho uchopit (nedostalo by se mezi prsty koncového efektoru).

Druhým nefunkčním prototypem byl zásobník vytvořený opět z polystyrénu, který byl zatížen, aby ho robot nemohl vzít společně s lízátkem. Lízátko zde byla rozmístěna do mřížky o velikosti 3x3 s pravidelnými rozestupy. Tento prototyp nefungoval. První

problém byl způsoben typem materiálu, ze kterého byl zásobník vyroben, a druhý jeho konstrukcí. Po několikanásobném vytažení lízátek a jejich opětovném umístění do zásobníku byly otvory v polystyrénu zvětšeny, lízátka už v nich nedržela pevně a nakláněla se. Robot mohl lízátko špatně uchopit, mohlo mu při přesunu vypadnout, nemuselo být vůbec sebráno nebo mohl robot o lízátko zavazit a odsunout celý zásobník stranou. Pro uchopení lízátka za tyčku neměl robot dostatečný prostor, protože část tyčky byla umístěna v zásobníku, a pokud bylo zapotřebí vzít lízátka ze zadních řad, robot do zásobníku narazil a odsunul ho stranou.

Nakonec byl vytvořen zásobník schodovitého tvaru (obrázek 45). Jako materiál bylo zvoleno dřevo, pro jeho pevnost a snadnou úpravu. Každá další řada byla alespoň o 15 mm výše než předchozí, aby se zamezilo kolizi robotu se zásobníkem. Do jednotlivých řad byly v pravidelných intervalech vyvrtané díry pro umístění lízátek a v pravém dolním rohu byl vytvořen otvor, do kterého byl umístěn Landmark sloužící k orientaci robotu.



Obrázek 45 – Konečná varianta zásobníku a jeho návrh

## 4.4. Manipulace s žetony

Pokud žeton (obrázek 46) nebo jakýkoliv jiný předmět dopadl před robot, bylo zapotřebí jej odstranit. Robot sám o sobě nebyl schopen předmět sebrat, kvůli jeho velikosti (pokud předmět přesahoval maximální velikost rozevření čelistí robotu nebo pokud byl naopak předmět příliš tenký, robot ho nebyl schopen pořádně uchopit a mohl mu vyklouznout). Z tohoto důvodu bylo potřeba zajistit způsob, jak odstranit všechny předměty před robotem. Funkce počítání předmětů patří mezi zpoplatněné funkce a nebylo možné ji u této práce využít.



Obrázek 46 – Žeton s logem

Pro odstranění předmětů bylo navrženo několik variant.

- **Propadliště**

Dopadová plocha by byla vytvořena z jedné nebo ze dvou desek, které by se pomocí motorů sklopily, předměty by propadly a desky by se vrátily do původní polohy. Mohlo by se ale stát, že při příliš malém sklopení desek by předměty nemusely sklouznout dolů.

- **Robot by shrnul předměty pomocí stěrky**

Robot by uchopil stěrku, pomocí které by shrnul předměty.

- **Předměty by byly shrnuty pomocí lineárního motoru**

Princip funkce by se podobal předešlému návrhu, stěrka by avšak byla ovládaná motorem, namísto robotu.

- **Pásový dopravník (obrázek 47)**

Předmět by dopadl na pás a robot by začal otáčet klikou spojenou přes řetězový převod s hnací hřídelí dopravníku, nebo by motor poháněl pás místo kliky. Pás by se začal pohybovat a odstranil by předmět.



*Obrázek 47 – Pásový dopravník*

## 4.5. Popis programu robotu a arduina

V této kapitole je shrnut celkový program robotu a sjednocení jednotlivých částí, ať už zásobníků, použití kamery, tak i přesun předmětů a komunikace s arduinem a obsluhou.

## 4.5.1. Zkoumání žetonu

Program začínal přesunem robotu nad plochu, kam dopadaly žetony. V této pozici neustále kontroloval, jestli pod ním nenastal pohyb. Pokud pohyb nastal, podíval se čím byl pohyb způsoben. Mohly nastat dvě možnosti.

- **Pohyb byl způsoben žetonem s logem**

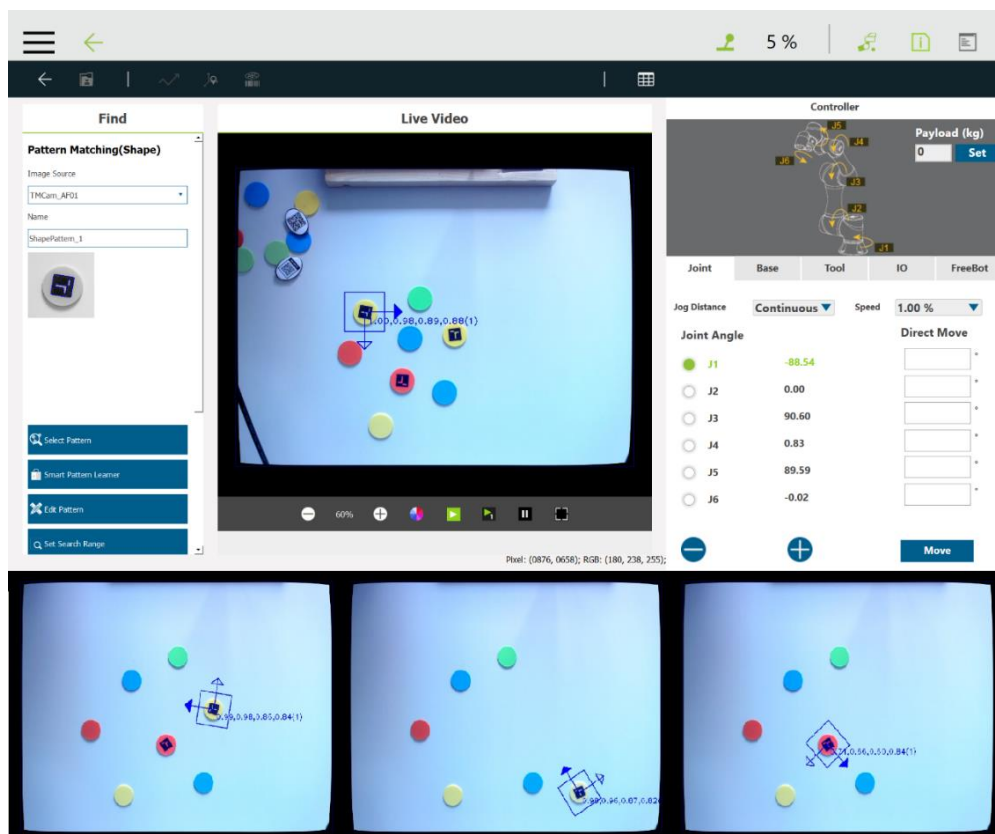
Robot rozpoznal žeton s logem a přešel k zásobníku lízátek.

- **Pohyb nebyl způsoben žetonem s logem**

Pokud byl vhozen jiný předmět než žeton s logem, robot daný předmět ignoroval a vrátil se k detekci pohybu.

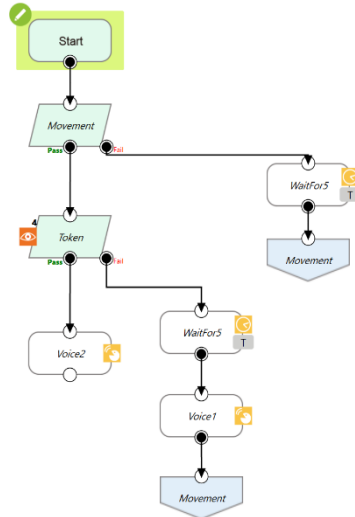
K rozpoznání žetonu byla využita funkce *Vision* a úloha *Fixed Point* s vytvořeným pracovním prostorem na desce stolu, kam dopadaly žetony. K následnému rozpoznání žetonu byl použit způsob *Pattern Matching (Shape)*, kde jako vzor (*pattern*) bylo vybráno logo znázorněné na žetonu.

Na obrázku 48 je uvedeno rozpoznávání žetonů. Pokud se v pracovní oblasti nacházelo více žetonů s logem, byl vybrán ten, který měl největší podobnost se vzorem.



Obrázek 48 – Rozpoznávání žetonů

Při nalezení žetonu vyslovil robot slovo „yes“, v opačném případě řekl slovo „no“ a počkal 5 vteřin, než se vrátil k detekci pohybu (obrázek 49).



Obrázek 49 – Část programu zabývající se rozpoznání žetonu s logem

## 4.5.2. Braní a podání lízátka

Po úspěšné detekci žetonu s logem se robot přesunul nad pracovní plochu a pokusil se nalézt *Landmark* nacházející se na zásobníku lízátek. Pokud byl *Landmark* nalezen, znamenalo to, že zásobník lízátek byl na svém místě. V opačném případě poslal zprávu obsluze, že zásobník nebyl nalezen, a čekal na nápravu. Poté se přesunul blíže nad zásobník a vyfotil si *Landmark* pro obnovení jeho pozice a natočení.

Robot přešel přes průchozí bod, tak aby nenarazil do zásobníku, do bodu, který se nacházel 5 cm před lízátkem. Poté se posunul k lízátku, až se tyčka lízátka objevila mezi rozevřenými prsty koncového efektoru robotu. Robot lízátko uchopil a nadzvedl ho o 20 cm směrem nahoru, aby se vyhnul kolizi při dalším pohybu.

Pro práci se zásobníkem byla využita funkce *Pallet*, jelikož usnadní programování a orientaci v zásobníku.

Dále je rozebráno nastavení funkce *Pallet* při tomto použití (obrázek 50)

The screenshot shows the 'Pallet' configuration window. The 'Node Name' is 'Pallet1'. It was recorded on 'vision\_Mark2' with 6 points and T1. The motion setting is 'PTP'. The pallet pattern is a 4x7 grid. The number of rows is 4 and the number of columns is 7. The number of layers is 0 and the thickness of each layer is 0 mm. The direction of the pallet is set to the first option. The base shifting is 'vision\_Mark2' and the payload is 0 kg. Precise positioning is checked. The 'OK' and 'Delete this node' buttons are visible at the bottom.

Obrázek 50 – Nastavení funkce *Pallet*

Tato funkce obsahuje několik parametrů, které bylo třeba nastavit. Jméno uzlu bylo ponecháno defaultní. Pod tímto jménem je napsáno, v jakém souřadném systému byla funkce nahrána. Při tomto použití je důležité, aby byl vybrán souřadný systém, který byl vytvořen pomocí *Landmarku* na zásobníku a jednotlivé polohy lízátek byly k němu vztaženy. V nastavení *Motion Settings* byl vybrán pro co nejkratší a nejvýhodnější posun robota pohyb PTP. Jako *Pallet Patern* byl zvolen způsob průchodu paletou řádek po řádku, začínající vždy na stejné straně zásobníku. Následně bylo potřeba naučit robota tři body, pomocí nichž se definoval zásobník. Jako první bod se zvolilo místo na zásobníku vpravo dole (toto místo se tam ve skutečnosti nenacházelo kvůli umístěnému *Landmarku*). Robot se přemístil do tohoto místa a natočil se kamerou nahoru, aby mu nepřekážela v cestě a byl schopen uchopit lízátka za jeho tyčku. Z důvodu, že při pohybu do tohoto bodu by robot mohl narazit do lízátka, byl tento bod ještě posunut o 5 cm před lízátka. Druhý a třetí bod byl zvolen stejným způsobem. Pozice bodů je znázorněna na obrázku 50 v nastavení *Pallet Patern*. Druhý bod se nacházel na konci prvního řádku a třetí bod byl umístěn na konci posledního sloupce a posledního řádku.

Poté se zvolil počet řádků 4 (znázorněno třemi vodorovnými čarami) a počet sloupců 7 (znázorněno třemi svislými čarami). Bylo vynecháno nastavení vrstev, volba způsobu brání předmětů z vrchního patra nebo z nejnižšího patra a nastavení hmotnosti předmětu, se kterým bude robot manipulovat, a tedy by nebyl nijak ovlivněn tíhou daného předmětu.



Robot si sám vytvořil polohy lízátek podle toho, kolik řádků a sloupců bylo zvoleno a jaké tři body byly vybrány pro popis palety.

Při spuštění programu byly všechny hodnoty (sloupce, řádku i patra) automaticky nastaveny na hodnotu 0. Při každém průchodu tímto uzlem se robot sám přesunuje v paletě podle nastaveného pořadí v *Pallet Patern*. Pokaždé při průchodu tímto uzlem (před uskutečněním pohybu) se hodnota řádku zvýší o 1. Pokud se hodnota řádku rovná přednastavené hodnotě celkového počtu řádků, nastaví se proměnná řádku zpátky na hodnotu 0 a proměnná sloupce se zvedne o 1. Při poslední pozici se proměnné vrátí na původní hodnoty a proměnná patra se zvedne o 1.

Zásobník se od palety lišil tím, že se na prvních dvou pozicích řádku i sloupce nenacházelo místo pro lízátka, nýbrž byl na toto místo připevněn *Landmark*. Bylo potřeba zařídit, aby robot tyto pozice vynechával. Na začátku programu byla použitím funkce *Set* přednastavena hodnota proměnné řádku na 2 a sloupce na 0. Po průchodu tímto uzlem se hodnota řádku zvýšila na 3 a robot začal procházet paletu od třetí pozice prvního sloupce. Pro druhý sloupec bylo nastavení obdobné. Pokud hodnota řádku byla 0 a hodnota sloupce se rovnala přednastavené hodnotě celkového počtu řádků, tak jím byla nastavena nová hodnota, pro řádek 1 a pro sloupec 2.

Poté, co robot uchopil a nadzvedl lízátka, se přemístí nad skluzavku, otočí koncový efektor a pustí lízátka.

## 4.6. Arduino

Z důvodu potřeby upozornit obsluhu robotu o nedostatku lízátek a nutnosti doplnění zásobníku, musela být zabezpečena komunikace mezi robotem a obsluhou. Robot komunikoval se svým okolím přes digitální piny. Pro zjednodušení této komunikace byly signály zpracovány přes arduino s wifi modulem a následné upozornění obsluhy proběhlo prostřednictvím emailu. Arduino bylo dále použito k signalizaci, zda robot rozpoznal žeton.

V programu bylo využito několik knihoven:

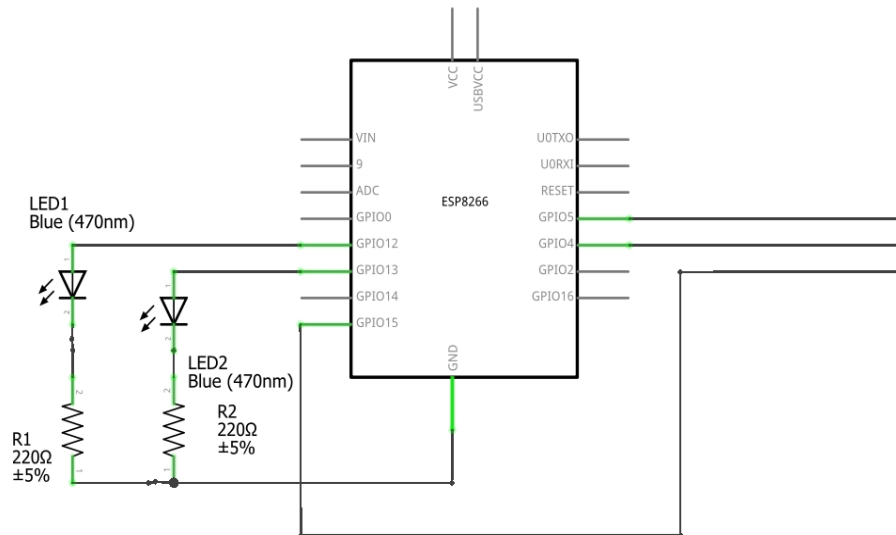
- knihovna ESP8266WiFi pro práci s wifi modulem,
- knihovna EMailSender, která byla převzata z GitHubu [13] pro odesílání emailů,
- knihovna ThingSpeak pro ukládání a načítání dat ze stejnojmenného serveru.

Celý program i s poznámkami lze nalézt v příloze 2 bakalářské práce.

## 4.6.1. Zapojení

V této podkapitole je uvedeno zapojení (obrázek 51) arduina.

Pro všechny účely byl využit IoT ESP8266 Lua NodeMcu Amica CP2102 WIFI modul.

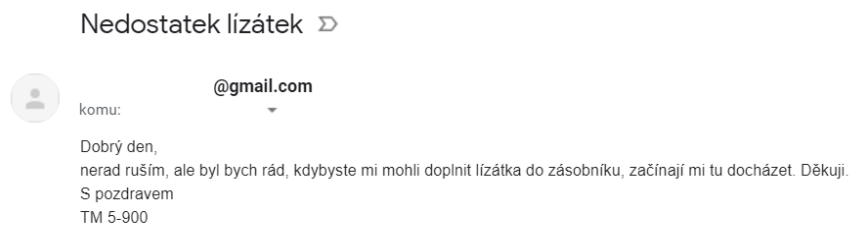


Obrázek 51 – Zapojení wifi modulu

## 4.6.2. Komunikace s robotem a obsluhou

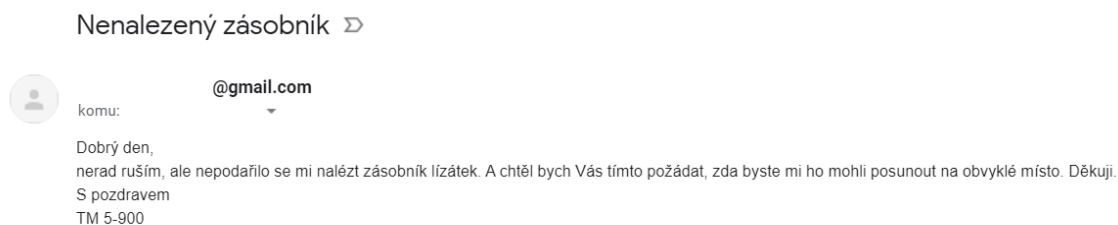
Robot komunikoval s obsluhou pomocí emailů, ve kterých bylo uvedeno, co se stalo a s čím by potřeboval robot asistenci. Toto jsou jednotlivé případy:

- **Nedostatek lizátek** – pokud začnou v zásobníku docházet lizátka, je zapotřebí je doplnit. Jakmile se robot dostane na poslední řadu lizátek v zásobníku, odešle přes arduino email o nedostatku lizátek. Jak vypadá odeslaný mail arduinem, je znázorněno na následujícím obrázku 52.



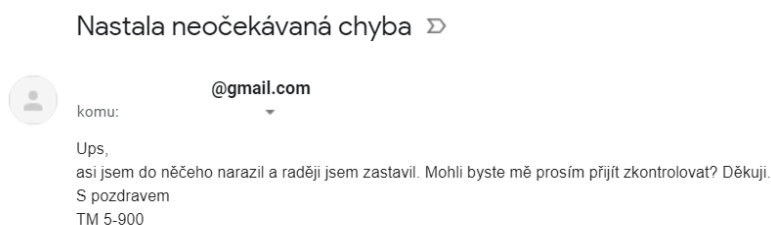
Obrázek 52 – Email odeslaný robotem upozorňující na nedostatek lizátek v zásobníku. Jméno odesílatele a příjemce bylo odstraněno z důvodu ochrany osobních údajů.

- **Nenalezený zásobník** (obrázek 53) – může dojít ke stavu, kdy robot nemůže nalézt zásobník lízátek, když byl po doplnění přesunut mimo zorné pole kamery.



*Obrázek 53 – Email odeslaný robotem upozorňující na nenalezení zásobníku. Jméno odesílatele a příjemce bylo odstraněno z důvodu ochrany osobních údajů.*

- **Nastala neočekávaná chyba** (obrázek 54) – když robot narazí do překážky, která se mu ocitla v cestě, zastaví se a přejde do stavu reprezentujícího chybu (identifikační světelný kroužek začne blikat červeně).



*Obrázek 54 – Email odeslaný robotem upozorňující na neočekávanou chybu. Jméno odesílatele a příjemce bylo odstraněno z důvodu ochrany osobních údajů.*

Pro odesílání těchto emailů pomocí arduina byla využita knihovna EMailSender. Následující úsek představuje část programu, který je zodpovědný za odesílání emailu ohledně nedostatku lízátek (zbylé emaily jsou řešeny stejným způsobem): [14]

```
//----- Send E-Mail Nedostatek -----
if ((digitalRead(D1)==1) && (digitalRead(D2)==1) && (digitalRead(D8)==0)){
  Serial.println("email 1 sent");
  // email odesílatele (gmail) a heslo
  EMailSender emailSend(SENDER, PASSWORD);
  EMailSender::EMailMessage message;
  // předmět a zpráva emailu
  message.subject = "Nedostatek lízátek";
  message.message = "Dobrý den,<br>nerad ruším, ale byl bych rád, kdybyste mi mohli doplnit
  lízátka do zásobníku, začínají mi tu docházet. Děkuji.<br>S pozdravem<br>TM 5-900";
  // poslání emailu na emailovou adresu
  EMailSender::Response resp = emailSend.send(RECIEVER, message);
  // zjištění zda byl email poslán
  Serial.println("Sending status: ");
  Serial.println(resp.code);
  Serial.println(resp.desc);
  Serial.println(resp.status);
}
```

### 4.6.3. Počítání lízátek

Pro zjištění počtu lízátek, který bude přibližně potřeba během jednoho dne pro robot zajistit, byla zavedena statistika počítání lízátek. Pomocí wifi modulu byla data odesílána a ukládána na server ThingSpeak. Hodnoty lze následně zpracovat a zjistit, kolik lízátek se rozebere během dne a v jakou denní dobu je nejčastěji robot používán.

```
//----- Write Data -----
// čas od spuštění v sec
stat = millis()/1000;
}
// zjištění zda uběhlo 30s od posledního odeslání dat (data se mohou nahrávat v rozmezí alespoň 15s)
else if((digitalRead(D1)==1) && (digitalRead(D2)==0) && (digitalRead(D8)==0) && ((times+30)<stat)){
    times = stat;
// zvýšení počtu lízátek o jedno
    number++;
// nahrání nového počtu lízátek na server
    int x = ThingSpeak.writeField(myChannelNumber, counterFieldNumber, number, myAPIKey);
    if(x == 200){
        Serial.println("Channel update successful.");
    }
    else{
        Serial.println("Problem updating channel. HTTP error code " + String(x));
    }
}
```

# ZÁVĚR

Hlavním cílem této bakalářské práce bylo vytvořit ukázkovou úlohu, na které budou znázorněny schopnosti robotu, jeho základní funkce a vybavení. Další cíle jsou uvedeny v následujících bodech:

- seznámení se s kolaborativním robotem,
- zjištění způsobů ovládání a možností programování přes počítač,
- otestování funkcí kamery.

V první části této práce byla zpracována krátká rešerše o kolaborativních robotech. Následně se přistoupilo k robotu TM 5-900 a byly popsány jeho jednotlivé části, od ramena a kamery, přes ovladač robotu, až po Univerzální Mobilní Stojan, na kterém byl robot upevněn. Protože se jedná o kolaborativní robot, který pracuje v blízkosti lidí, musí robot splňovat přísnější bezpečnostní normy a jeho pohyb musí být omezen s ohledem na obsluhu, která s ním pracuje.

Robot je možné programovat přes počítač, pouze pokud je robot s počítačem připojen ke stejnému fyzickému přístupovému bodu nebo pokud je počítač přímo připojen k robotu za pomoci síťového kabelu. Na tomto počítači však musí být nainstalována klientská verze softwaru TMflow. Další možností, jak lze robot programovat, je připojením klávesnice, myši a monitoru k robotu (Univerzálnímu Mobilnímu Stojanu) a spuštěním programu TMflow v robotu. Nevýhodou je, že bez hardwarového klíče, který je nutné si pořídit, nelze programovat v programu TMflow bez připojeného robotu.

Jak už bylo zmíněno, programování robotu probíhá v programu TMflow. Jedná se o grafické programovací prostředí, kde jsou jednotlivé funkce (uzly, bloky) propojovány vodiči. Software je přizpůsobený pro vertikální stavbu programu (funkce jsou propojeny ze shora a ze spodu), z tohoto důvodu s rostoucí délkou programu je nutné se často posouvat nahorů a dolů v programu a je obtížné se v něm orientovat. Navíc při potřebě duplikovat funkci nelze využít klávesových zkratk pro kopírování a přilepování údajů (ctrl+c a ctrl+v), ale je zapotřebí vybrat daný blok, rozkliknout možnost *edit block* a vybrat tlačítko *copy* a *paste*. Při práci v TMflow se nesmí zapomínat, že po úpravě jednotlivých bloků je zapotřebí stisknout tlačítko *OK*, jinak se provedené úpravy neuloží.

Pro práci s kamerou se používá funkce *Vision* v programu *TMflow*. V této funkci lze změnit nastavení kamery a vybrat úlohu, která je vhodná pro práci s kamerou. Mezi dostupné úpravy obrazu, patří například změna kontrastu. Dále je možné pomocí kamery nalézt objekt několika způsoby, přečíst QR kód nebo zjistit barvu předmětů. Ke kalibraci a polohování slouží *Landmark* a kalibrační desky.

Ukázková úloha byla postavena na principu automatu na lízátka, kde po vhození žetonu podá robot člověku lízátko. Aby byl vůbec robot schopen uchopit a podat lízátko, byly pro chapadlo vytvořeny prsty na 3D tiskárně tak, aby bylo možné vzít lízátko za jeho tyčku. Lízátka pak musela být pro lehčí uchopení robotem rozestavěna v pravidelných intervalech v zásobníku schodovitého tvaru. Ukázková úloha spočívala v tom, že člověk vhodí před robot žeton, ten jej rozpozná a následně je žeton odstraněn. Robot se přesune k zásobníku lízátek, odkud jedno lízátko vezme a podá ho člověku.

Jelikož by po určité době došla lízátka kontaktuje robot obsluhu za pomoci arduina s wifi modulem prostřednictvím emailů. Odesílány byly také emaily ohledně nenalezeného zásobníku a rovněž v případech, kdy robot vyžadoval asistenci. Arduino kromě toho posloužilo k počítání lízátek, která byla rozdána během dne.

Možné vylepšení této úlohy by spočívalo ve čtení jedinečných QR kódů přes plexisklo namísto rozpoznávání žetonů (není nutné nic odklízet).

# SEZNAM POUŽITÝCH ZDROJŮ

- [1] ŽÁČEK, Michal. Průmyslové roboty: Jaké jsou jejich druhy?. In: *Factory Automation* [online]. FANUC Czech, 2018 [cit. 2020-05-31]. Dostupné z: <https://factoryautomation.cz/prumyslove-roboty-jake-jsou-jejich-druhy/>
- [2] Robot alebo kobot. Podáte ruku robotovi?. In: *Talentica* [online]. Bratislava: Talentica, 2018 [cit. 2020-05-29]. Dostupné z: [https://www.talentica.sk/robot-alebo-kobot/?fbclid=IwAR0w-ulPBza3Z6TW1\\_YsTjA3sEcc4EjhdnBXXgh9W7sCVozY4ufnemVUY4s](https://www.talentica.sk/robot-alebo-kobot/?fbclid=IwAR0w-ulPBza3Z6TW1_YsTjA3sEcc4EjhdnBXXgh9W7sCVozY4ufnemVUY4s)
- [3] *Collaborative Robot (Cobot) Market* [online]. MarketsandMarkets Research Private, 2020 [cit. 2020-06-25]. Dostupné z: <https://www.marketsandmarkets.com/Market-Reports/collaborative-robot-market-194541294.html>
- [4] PEREZ, Denrie Caila. *OMRON and Techman Launch TM Series Collaborative Robot* [online]. engineering.com, 2018 [cit. 2020-06-23]. Dostupné z: <https://www.engineering.com/AdvancedManufacturing/ArticleID/17931/OMRON-and-Techman-Launch-TM-Series-Collaborative-Robot.aspx>
- [5] *ZLÍN ROBOTICS: Průmyslová automatizace a robotizace výroby* [online]. Slušovice: ZLÍN ROBOTICS, 2020 [cit. 2020-05-28]. Dostupné z: <https://www.zlinrobotics.cz/>
- [6] *Příručka k softwaru TMflow*. 1.03. Tchaj-wan: Techman Robot, 2018.
- [7] *Návod na instalaci hardwaru: Série běžného nákladu*. 1.03. Tchaj-wan: Techman Robot, 2018.
- [8] *Návod*. Slušovice: ZLÍN ROBOTICS, 2018.
- [9] *Bezpečnostní návod*. 1.03. Tchaj-wan: Techman Robot, 2018.
- [10] JIŘÍ, Hlinovský. Zabezpečení strojů a strojních zařízení proti následkům poruchy jejich vlastního elektrického řídicího systému. *Elektro: časopis pro elektrotechniku* [online]. Praha: FCC PUBLIC, 2000, (042000) [cit. 2020-06-16]. Dostupné z: <http://www.odbornecasopisy.cz/elektro/casopis/tema/zabezpeceni-stroju-a-strojnich-zarizeni-proti-nasledkum-poruchy-jejich-vlastniho-elektrickeho-ridiciho-systemu--15159>
- [11] *Příručka k softwaru TMvision*. 1.00. Tchaj-wan: Techman Robot, 2018.
- [12] Co-act EGP-C 40-N-N-URID. *Schunk* [online]. Lauffen am Neckar: Schunk GmbH, 2020 [cit. 2020-06-17]. Dostupné z: [https://schunk.com/cz\\_cs/uchopovaci-systemy/product/63296-1326455-co-act-egp-c-40-n-n-urid/](https://schunk.com/cz_cs/uchopovaci-systemy/product/63296-1326455-co-act-egp-c-40-n-n-urid/)

- [13] *Library to send EMail with attachments*. Dostupné z: <https://github.com/xreef/EMailSender>
- [14] Control an LED from Webserver using NodeMcu or Esp8266 programming with Arduino IDE. In: *Internet Of Things* [online]. 2014 [cit. 2020-06-26]. Dostupné z: <http://internetofthinking.blogspot.com/2015/12/control-led-from-webserver-using.html>
- [15] ROBOT UR5e. In: *Amtech* [online]. Amtech, 2020 [cit. 2020-05-31]. Dostupné z: <https://www.amtech.cz/universal-robots/robot-ur5e>
- [16] Safe Robotics. In: *Safe Productivity – bezpečné roboty | SICK* [online]. Waldkirch: SICK AG, 2016 [cit. 2020-05-29]. Dostupné z: [https://cdn.sick.com/media/docs/2/92/292/Special\\_information\\_Safe\\_Robotics\\_HUMAN\\_ROBOT\\_COLLABORATION\\_en\\_IM0073292.PDF](https://cdn.sick.com/media/docs/2/92/292/Special_information_Safe_Robotics_HUMAN_ROBOT_COLLABORATION_en_IM0073292.PDF)
- [17] Mobilní stojany pro roboty Techman. In: *ZLÍN ROBOTICS: Průmyslová automatizace a robotizace výroby* [online]. Slušovice: ZLÍN ROBOTICS, 2020 [cit. 2020-06-17]. Dostupné z: [https://www.zlinrobotics.cz/images/UMS-NEW/robot\\_06-pruhl.png](https://www.zlinrobotics.cz/images/UMS-NEW/robot_06-pruhl.png)



# SEZNAM PŘÍLOH

Příloha 1

Ukázková úloha

Video

Příloha 2

Program arduina

Arduino IDE