



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV AUTOMATIZACE A INFORMATIKY**

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**NÁVRH ŘÍZENÍ HOBBY ROBOTICKÉHO  
RAMENE**

HOBBY ROBOTIC ARM CONTROL DESIGN

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

Aleksei Chursin

**VEDOUCÍ PRÁCE**

SUPERVISOR

Ing. Zdeněk Cejpek

BRNO 2021



## Zadání bakalářské práce

Ústav:	Ústav automatizace a informatiky
Student:	<b>Aleksei Chursin</b>
Studijní program:	Strojírenství
Studijní obor:	Aplikovaná informatika a řízení
Vedoucí práce:	<b>Ing. Zdeněk Cejpek</b>
Akademický rok:	2020/21

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:

### – **Návrh řízení hobby robotického ramene**

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

Robotická ramena nabízí agilní řešení při automatizaci řady úkonů ve výrobě, logistice i dalších odvětvích. Značná poptávka po pracovnících znalých práce s robotickými zařízeními vedla nejen k rozšíření výuky práce s profesionálními systémy, nýbrž i k rozmachu různých nízkonákladových hobby řešení, jež poskytují hravou formu vzdělání a zábavy.

Cílem této práce je návrh a realizace řízení již existujícího hobby robotického ramene na platformě Arduino/Matlab a zpracování jednoduché laboratorní úlohy vhodné pro seznámení studentů bakalářského studia s prací s robotickými manipulátory.

#### **Cíle bakalářské práce:**

- výběr a zapojení vhodné řídicí desky Arduino
- návrh jednoduché laboratorní úlohy s využitím konektivity platformem Arduino a Matlab
- vypracování podkladů pro laboratorní cvičení
- realizace řízení robota po vybrané sběrnici

#### **Seznam doporučené literatury:**

KOLÍBAL, Zdeněk. Roboty a robotizované výrobní technologie. Brno: Vysoké učení technické v Brně nakladatelství VUTIUM, 2016. ISBN 978-80-214-4828-5.

SICILIANO, Bruno a Oussama KHATIB. Springer handbook of robotics. 2. Berlin: Springer, 2016. ISBN 978-3-319-32550-7.

VODA, Zbyšek a Martin STRÍŽ. Průvodce světem Arduina. 2015. Bučovice. ISBN 978-80-87106-90-7.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.  
ředitel ústavu

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

## **ABSTRAKT**

Tato bakalářská práce se zabývá sestavením robotického ramene navrženého skupinou CustomElectronics, realizací jeho řízení a následným návrhem laboratorní úlohy s využitím tohoto manipulátoru. Teoretická část popisuje konstrukci mechanismu, vlastnosti manipulátoru a dalších zařízení použitých při návrhu. Praktická část je zaměřena na realizaci navržené úlohy a vypracování podkladů.

## **ABSTRACT**

This thesis deals with the assembly of a robotic arm designed by the CustomElectronics group, the implementation of its control and the subsequent design of a laboratory task using this manipulator. The theoretical part describes the construction of the mechanism, the properties of the manipulator and other devices used in the design. The practical part is focused on the implementation of the proposed task and the development of documents.

## **KLÍČOVÁ SLOVA**

Řízení dopravníků, řízení manipulátorů, programování v arduino, arduino-matlab package, školení studentů, kolaborativní roboty, rozpoznávání barvy objektů, sběrnice

## **KEYWORDS**

Conveyor control, manipulator control, arduino programming, arduino-matlab package, student training, collaborative robots, object color recognition, bus.

## **BIBLIOGRAFICKÁ CITACE**

CHURSIN, Aleksei. *Návrh řízení hobby robotického ramene*, Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky.







## **PODĚKOVÁNÍ**

Chtěl bych velice poděkovat vedoucímu moje bakalářské práce Ing. Zdenku Cejpkovi, za odborné rady, poskytnutí pozornosti, času, a pomoc při realizaci. Dekuji taky své přítelkyni, která je mi v životě oporou.





# ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Zdenka Cejpka a s použitím literatury uvedené v seznamu literatury.

V Brně dne 21. 5. 2021

.....  
Aleksei Chursin





# OBSAH

<b>1</b>	<b>ÚVOD .....</b>	<b>19</b>
<b>2</b>	<b>NÁVRH LABORATORNÍ ÚLOHY .....</b>	<b>20</b>
2.1	Cíle úlohy .....	20
2.2	Základní představení úlohy .....	20
<b>3</b>	<b>KONSTRUKCE MANIPULÁTORU CUSTOMELECTRONICS .....</b>	<b>21</b>
3.1	Konstrukce .....	22
3.1.1	Přizpůsobení konstrukce pro účely úlohy .....	23
3.2	Programování .....	23
3.3	Kinematický model .....	24
3.3.1	Dopředná kinematická úloha .....	24
<b>4</b>	<b>NÁVRH ZÁSOBNÍKU .....</b>	<b>27</b>
4.1	Vertikální zásobník .....	27
4.2	Pásový dopravník .....	28
4.3	Realizace .....	30
<b>5</b>	<b>VÝBĚR VHODNÉ ŘÍDÍCÍ DESKY ARDUINO .....</b>	<b>31</b>
5.1	Charakteristiky Arduino Mini Pro .....	31
5.2	Charakteristiky Arduino Mega 2560 .....	32
5.3	Charakteristiky Arduino Uno .....	32
<b>6</b>	<b>ELEKTRICKÉ VYBAVENÍ LABORATORNÍ ÚLOHY .....</b>	<b>34</b>
6.1	DC MOTORY .....	34
6.2	H můstek .....	35
6.3	Modelářské servomotory .....	37
6.3.1	Ovládání servomotoru .....	38
6.4	Vstupní senzory .....	41
6.4.1	Senzor přítomnosti .....	41
6.4.2	Senzor barev .....	42
6.5	Napájení .....	44
<b>7</b>	<b>ŘÍZENÍ ROBOTY PO VYBRANÉ SBĚRNICI .....</b>	<b>44</b>
7.1	Komunikační rozhraní UART .....	44
7.2	Komunikační rozhraní I2C .....	45
7.3	Komunikační rozhraní SPI .....	46
7.4	Komunikační rozhraní RS-232 .....	47
7.5	Komunikační rozhraní RS-485 .....	48
7.6	Komunikační rozhraní CAN .....	48
7.7	Komunikační rozhraní Ethernet .....	49
7.8	Realizace řízení ramene po SPI sběrnici .....	49
<b>8</b>	<b>REALIZACE LABORATORNÍ ÚLOHY .....</b>	<b>50</b>
8.1	Zapojení celé soustavy do řídicí desky Arduino Uno .....	51
8.2	Montáž .....	53
8.3	Řízení úlohy pomocí Matlabu .....	53
<b>9</b>	<b>ZÁVĚR .....</b>	<b>55</b>



<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>57</b>
<b>SEZNAM OBRÁZKŮ A TABULEK.....</b>	<b>59</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>61</b>



# 1 ÚVOD

V současné době se v průmyslové automatizaci hojně využívá robotických ramen v roli manipulátorů nejrůznějšího materiálu. K tomuto rozšíření dochází především velké flexibilitě těchto manipulátorů, kdy pouhou změnou programu mohou zásadně změnit vykonávané pohyby, čímž předčí manipulátory konstruované jednoúčelově. Je tedy žádoucí studenty seznámit nejen s roboty jako takovými, ale i výhodami, jež přináší nasazení programovatelných zařízení a vzbudit tak zájem o programování strojů již na počátku studia.

Hlavním cílem této bakalářské práce je tedy oživení již hotového robotického ramene a jeho nasazení v zbrusu nové laboratorní úloze typu "pick and place". Podstatou úlohy je třídění manipulovaných součástí podle barvy. Součástí práce je nejen samotná výroba a oživení laboratorní úlohy, ale též podklady potřebné k jejímu nasazení ve výuce.

Úloha je koncipována pro studenty prvního ročníku a její nasazení je předpokládáno v předmětu IIN-Informatika, čemuž odpovídá složitost dílčích úkolů. Úloha však nabízí značný prostor pro procvičení programování i pro pokročilejší studenty automatizace.

Dále tato bakalářská práce obsahuje krátké představení všech elektrických a elektronických komponent laboratorní úlohy a programové vybavení umožňující robotické rameno provozovat jako "periferii" ovládanou přes sběrnici SPI.

## 2 NÁVRH LABORATORNÍ ÚLOHY

### 2.1 Cíle úlohy

Tato kapitola zahrnuje popis návrhu laboratorní úlohy zaměřené na řízení robota s využitím konektivity platformem *Arduino* a *Matlab*. Laboratorní úloha by měla být řešitelná pro studenty prvního ročníku Fakulty strojního inženýrství předmětu **Informatika-1IN**. Aby bylo možné studentům názorně předvést komplexní (avšak přiměřeně jednoduchou) ukázkou nasazení robotické aplikace, bylo nutné úlohu rozšířit o vhodná zařízení, která lze snadno řídit přes *MATLAB Support Package for Arduino Hardware*.

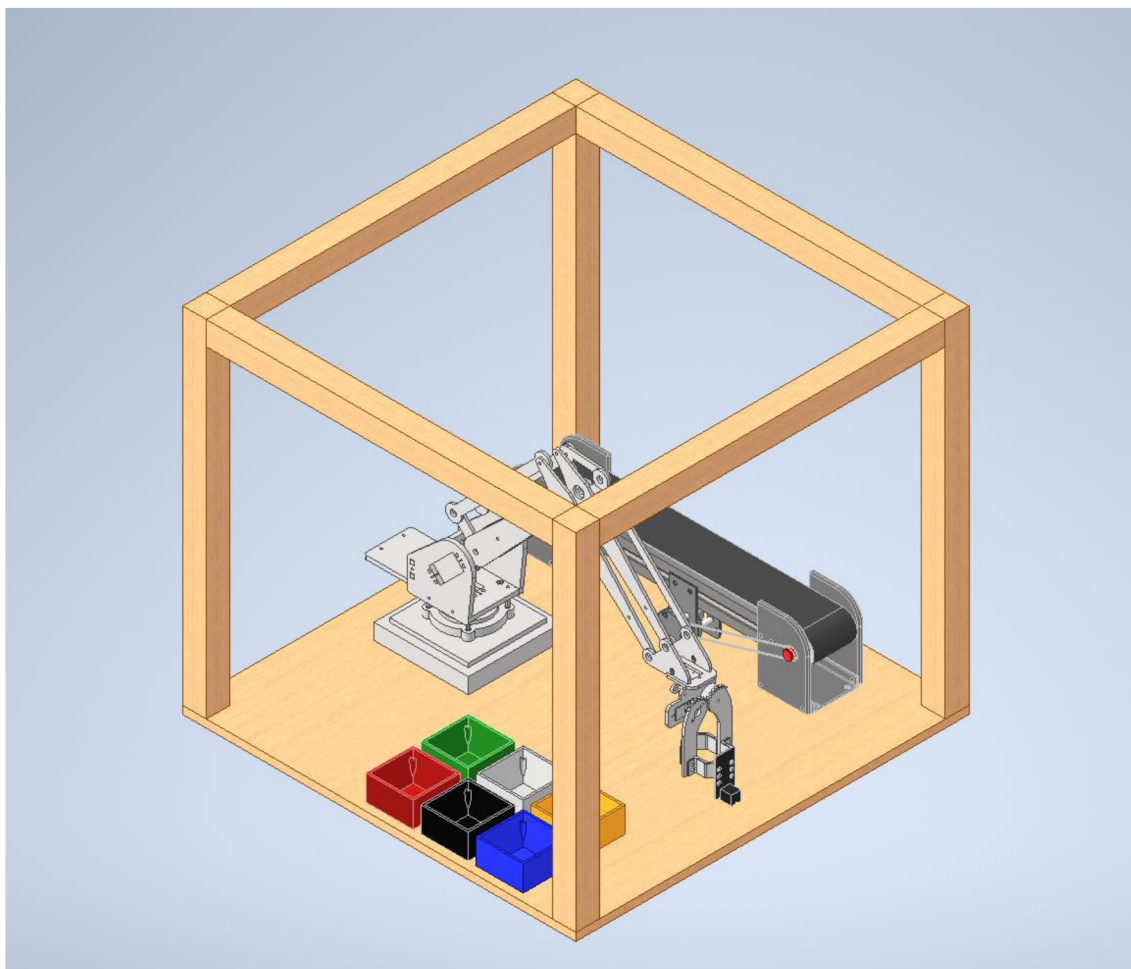
### 2.2 Základní představení úlohy

V laboratorní úloze se používá robotické rameno navržené společností *CustomElectronics*. Pomocí tohoto manipulátoru a dalších zařízení byla navržena úloha typu „*pick and place*“, kde nejdůležitější roli hraje robotické rameno. Hlavní funkcí manipulátoru v úloze je přesouvání krabiček z vstupního zásobníku do příslušného odkladiště.

Výroba nestandardních dílů byla uskutečněna pomocí moderní technologie 3D tisku. Mezi použitými zařízení patří senzory barev, senzory sledování čáry a stejnosměrný motor řízený pomocí H můstku. Jedním z hlavních záměrů autora práce při návrhu úlohy bylo nadchnout studenty k tomu, aby začali vytvářet svoje vlastní projekty. Důležitým kritériem při návrhu úlohy byla nízká výrobní cena a jednoduchost výsledné konstrukce.

Ovládání úlohy bylo realizováno pomocí desky *Arduino* řízené ze studentům známého prostředí *Matlab*. Toho bylo dosaženo využitím knihovny *MATLAB Support Package for Arduino Hardware*.

Jako manipulované součástky byly zvoleny krabičky o straně 30mm vyrobené pomocí technologie 3D tisku.



Obr. 2.2: 3D model finální verze návrhu úlohy

Součástí návrhu je i polouzavřená konstrukce chránící úlohu před poškozením. Rovněž usnadňuje manipulaci a skladovatelnost.

### 3 KONSTRUKCE MANIPULÁTORU CUSTOMELECTRONICS

V bakalářské práci je využito nízkonákladové řešení robotického ramene od týmu *CustomElectronics*. *CustomElectronics* je komunita inženýrů a programátorů elektroniky. Spojuje je touha po kreativní seberealizaci. Většina jejich projektů se vyvíjí výhradně pro akademický zájem, ale někdy se provádí i malé komerční zakázky.

Na webových stránkách společnosti je v části Školení umístěna celá řada informací od autorů se všemi zdrojovými materiály.

### 3.1 Konstrukce

Manipulátor je poskládaný z dílů vyřezaných z plexiskla tloušťky 3 a 5 mm, třech silnějších servomotorů MG995, dvou servomotorů MG90S, radiálních kuličkových ložisek. Pro montáž se využívali M3 a M4 šrouby.

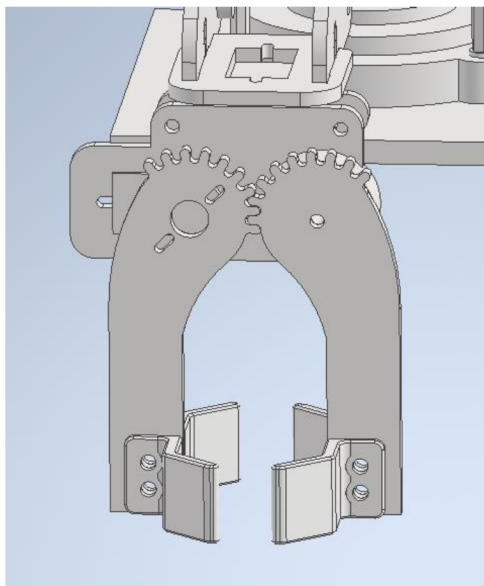
Robotická ruka byla jedním z nejdelších projektů společnosti *CustomElectronics*. Výsledné řešení má hned několik výhod oproti modelům momentálně dostupných na trhu. Níže jsou uvedené některé vlastnosti manipulátoru [1]:

1. Všechny klouby jsou opatřeny kuličkovými ložisky, které zvyšují přesnost a především životnost robota. Celkový počet ložisek je 11 kusů.
2. Jednoduchá montáž. Při návrhu řešení bylo záměrem věnovat co největší pozornost postupu poskládání manipulátoru tak, aby se montáž prováděla co nejsnadněji.
3. Veškeré výkonné servomotory jsou uspořádány u základny a to tak, že “dolní” servomotory netáhnou “horní” aby jejich hmotnost nesnižovala nosnost výsledné konstrukce..
4. Díky smíšené sério-paralelní kinematice gripper vždy zůstává v natočení kolmém nebo paralelním k podložce.
5. Otevřenost projektu. Veškeré výrobní podklady jsou volně k dispozici k užití i úpravám
6. Hotový ukázkový Arduino-kompatibilní software.
7. Při výšce 300 mm je pracovní plocha na podložce vymezena mezikružím o poloměrech 140 a 300 mm.
8. Maximální váha, se kterou robotická ruka může operovat, je 200 gramů, a to i při zachování maximálního dosahu.

- Potřebný proud je maximálně špičkově 6A. Tento odběr není trvalý a působí jen po dobu max. 3 ms v 20 ms cyklu. Vhodným programovým řešením však lze odběr snížit na úroveň jediného servomotoru.

### 3.1.1 Přizpůsobení konstrukce pro účely úlohy

Protože se výchozí kleště robota pro manipulaci se zvolenými krabičkami neosvědčily, byl navržen vlastní gripper.



Obr. 3.1: Model kleští pro manipulaci s krabičkami

Při návrhu se vycházelo z tvaru již existujících kleští. Gripper se skládá ze dvou prstů se synchronizovaným otáčivým pohybem a čtyř S profilů, je navržený speciálně pro práci s použitými krabičkami. Montáž byla provedena pomocí šroubů a matic M3.

## 3.2 Programování

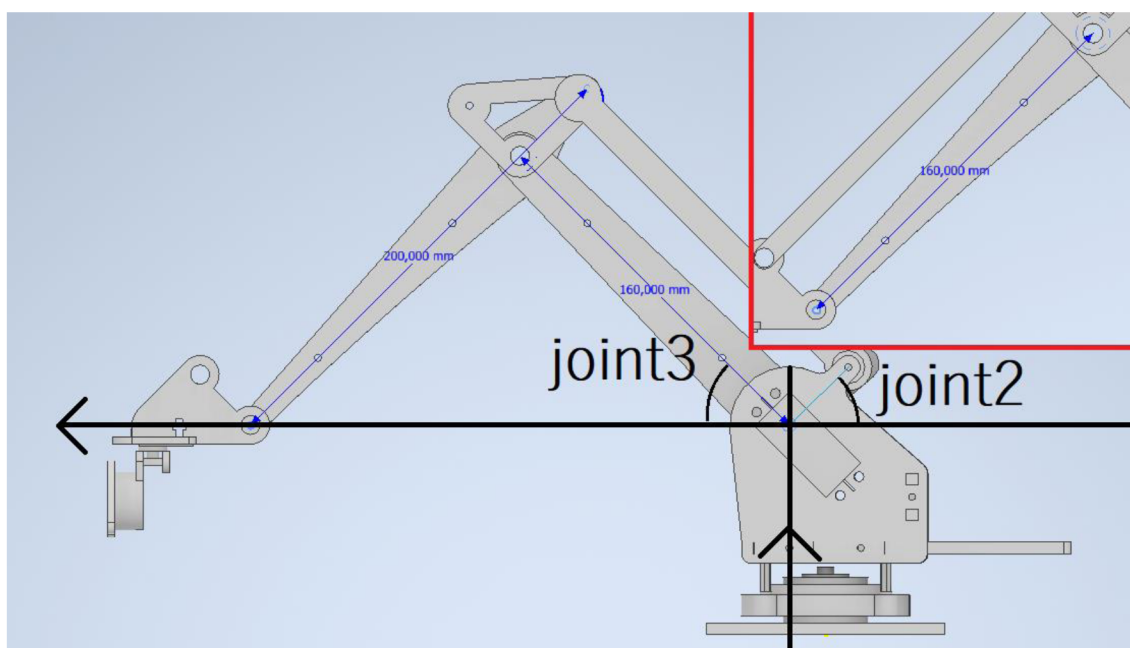
Na stránkách *CustomElectronics* jsou k dispozici tři různá hotová software řešení pro práci ve třech režimech, například řízení pomocí příkazu z počítače. Využívá se tady software od *uFactory*, který umožňuje ovládat manipulátor pomocí myši. V navržené

úloze bylo ale použito vlastní řešení, jež dovolí studentům neinformatického oboru využít znalosti jazyku *Matlab*. Současně to bude představovat minimální zátěž pro studenty z hlediska zahlcení přebytečnými podrobnostmi této aplikace nad rámec základního kurzu informatiky, pro který je využití výstupu této práce zamýšleno.

### 3.3 Kinematický model

#### 3.3.1 Dopředná kinematická úloha

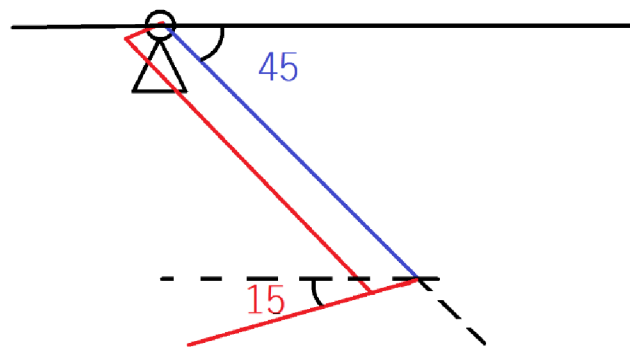
Pro řízení robota je nezbytně nutná definice jeho kinematického modelu. Kinematický model dovoluje vypočítat pozice jednotlivých kloubů a především pak koncového bodu manipulátoru. Natočení koncového bodu není v případě kinematiky tohoto manipulátoru příliš zajímavé, neboť je udržováno paralelními vazbami konstantně v pozici vodorovné s podložkou. Přímá, neboli dopředná úloha kinematického modelu vychází z toho že jsou známé délky kloubů a úhly natočení. Je tedy potřeba zjistit souřadnice koncového bodu.



Obr. 3.3.1.1 model manipulátoru: pohled z boku

Na obrázku Obr. 3.3.1.1 jsou uvedené uhly motoru joint3 a joint2, které se rovnají  $45^\circ$ . Manipulátor je ale sebrán tak, že v této poloze uhly natočení na motorech joint3 a joint2 jsou  $90^\circ$  a  $60^\circ$ . Je potřeba pro výpočet dopředné kinematiky to vzít v úvahu. Pozice manipulátoru s nulovou hodnotou natočení servomotorů je:





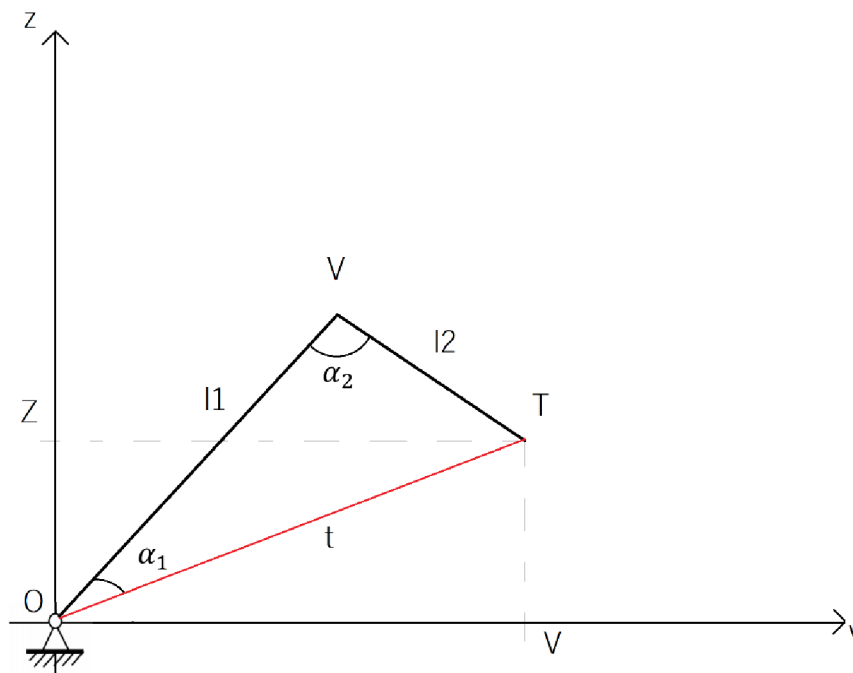
Uhel mezi dvěma částmi ramene v “nulovém” stavu je:

$$\delta = 15^\circ + 45^\circ = 60^\circ$$

Což je rozdíl mezi pracovními rozsahy servomotoru Joint2 a Joint3. Pak platí:

$$\beta_1 = \alpha_1 - 45^\circ$$

$$\beta_2 = \alpha_2 - 120$$

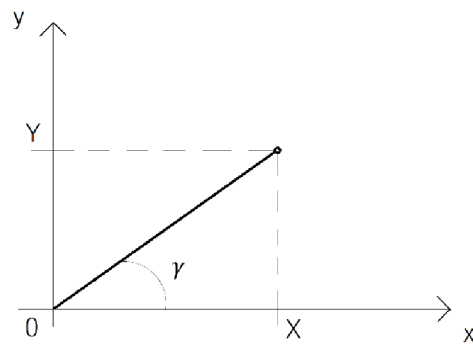


$$T_v = (\cos \alpha_1) * l_1 + \cos(180 - \alpha_1 - \alpha_2) * l_2 \quad (1)$$

$$T_z = (\sin \alpha_1) * l_1 - \sin(180 - \alpha_1 - \alpha_2) * l_2 \quad (2)$$

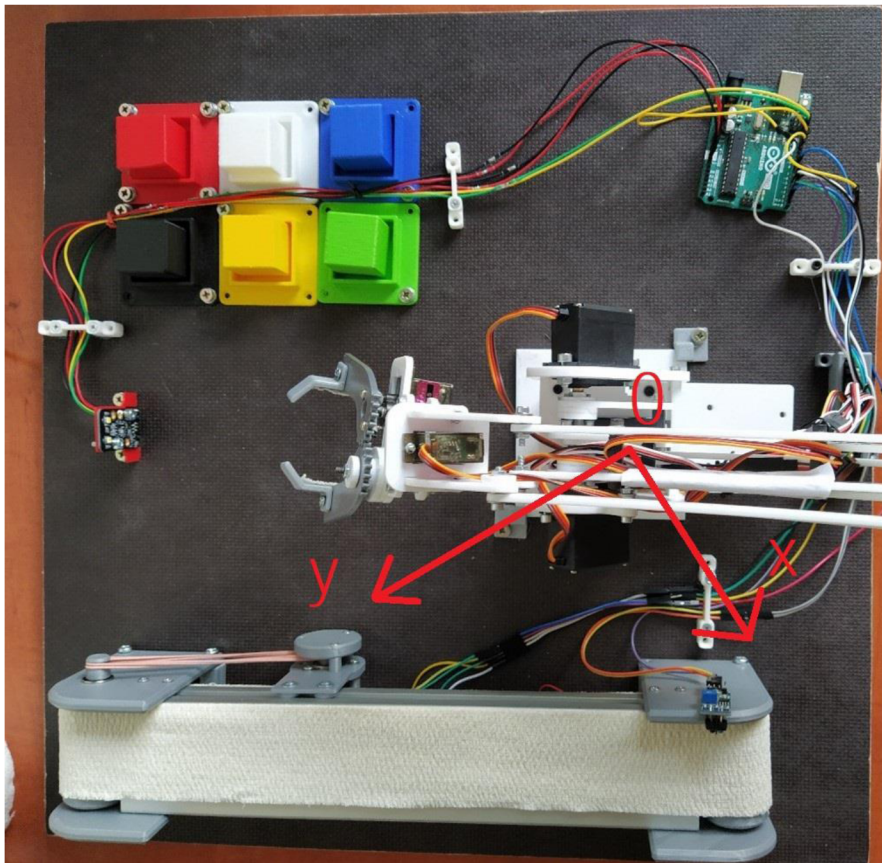
Kde  $T_v$  a  $T_z$  jsou souřadnice koncového bodu manipulátoru při uhlech natočení  $\alpha_1$   $\alpha_2$

Zbývající souřadnice se vypočítají jednoduše:



$$T_y = T_v \cdot \sin(\gamma) \quad (3)$$

$$T_x = T_v \cdot \cos(\gamma) \quad (4)$$



Obr. 3.3.1.2: Umístění os v úloze.

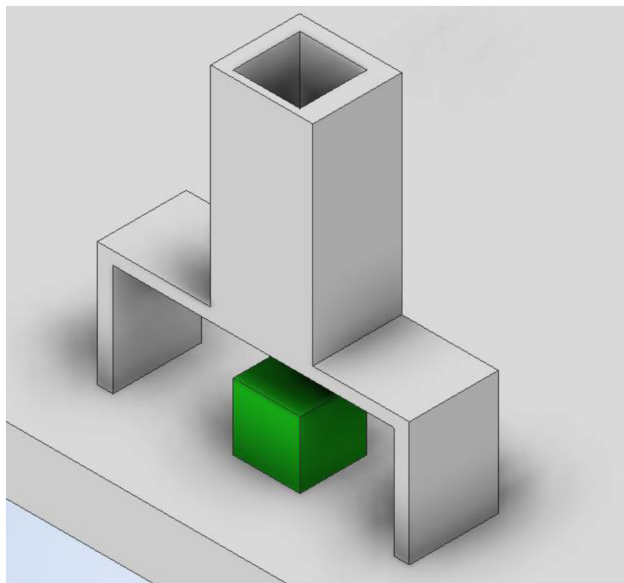
## 4 NÁVRH ZÁSOBNÍKU

V této kapitole bude následovat popis postupu návrhu zásobníku pro manipulované součástky použité při návrhu úlohy.

Při návrhu úlohy typu „*pick and place*“ jsou použité dva zásobníky, pojmenované jako "zdroj" a "odběratel" manipulovaných kostek. Jeden zásobník obsahuje součástky ještě netříděné. Druhý zásobník přijme součástky již vytríděné a má v sobě rozdělení pro každý typ součástky. Cílem návrhu bylo také dodržet možnost jak vkládání, tak i sbírání součástek z obou zásobníků.

### 4.1 Vertikální zásobník

První navrženou variantou "zdroje" byl vertikální nepohyblivý zásobník, který přijímá krabičky otvorem v horní části, odkud se krabičky vlivem gravitace pohybují směrem dolů k otvoru pro odběr krabiček manipulátorem. Z tohoto otvoru by krabičky mohly být odebírány manipulátorem s gripperem instalovaném v horizontálním směru (viz Obr. 4.2). Výhodou tohoto návrhu je jednoduchost výroby a snadná montáž. Nevýhodou je však omezená kapacita závislá na výšce zásobníku. Kvůli dosažení dobré manipulovatelnosti a skladovatelnosti modelu je výška zásobníku dosti omezená. Pokud by bylo potřeba v zásobníku umístit více krabiček, musel by být zásobník prodloužen, což by nejen neúměrně zvětšilo zástavbové rozměry modelu (tedy zkomplikovalo transport a skladování), ale též znemožnilo robotu vracet již roztříděné kostky zpět do zásobníku (to lze využít například při prezentaci modelu veřejnosti a při podobných příležitostech.).

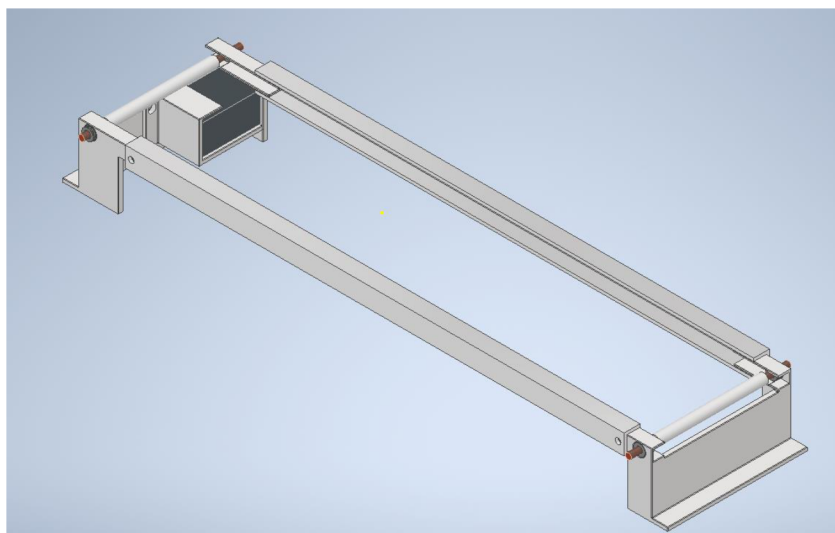


Obr. 4.1: Vertikální zásobník

## 4.2 Pásový dopravník

Další možností realizace "zdroje" manipulovaných krabiček byl pásový dopravník vlastního návrhu, který vystupoval v roli přívodu nových manipulovaných objektů. V první iteraci návrhu byl dopravník sestaven z:

- 3 součástek tištěných pomocí technologie 3D tisku
- Dvou L profilů



Obr. 4.2.1: První iterace návrhu pásového dopravníku

První iterace návrhu pásového dopravníku měla několik výhod, a to hlavně malé rozměry, celkovou kompaktnost a možnost upravení délky, což by mohlo být využitelné k nastavení správného napnutí pásu. Kvůli obtížnosti tisku však tento návrh realizován nebyl.

Druhá iterace návrhu pásového dopravníku byla převzata z bakalářské práce od pana Bc. Vojtěcha Drážky na téma „*Návrh malého pásového dopravníku*“, která se zabývala návrhem, konstrukcí a realizací malého pásového dopravníku pro potřeby laboratoří Ústavu automatizace a informatiky na VUT FSI.



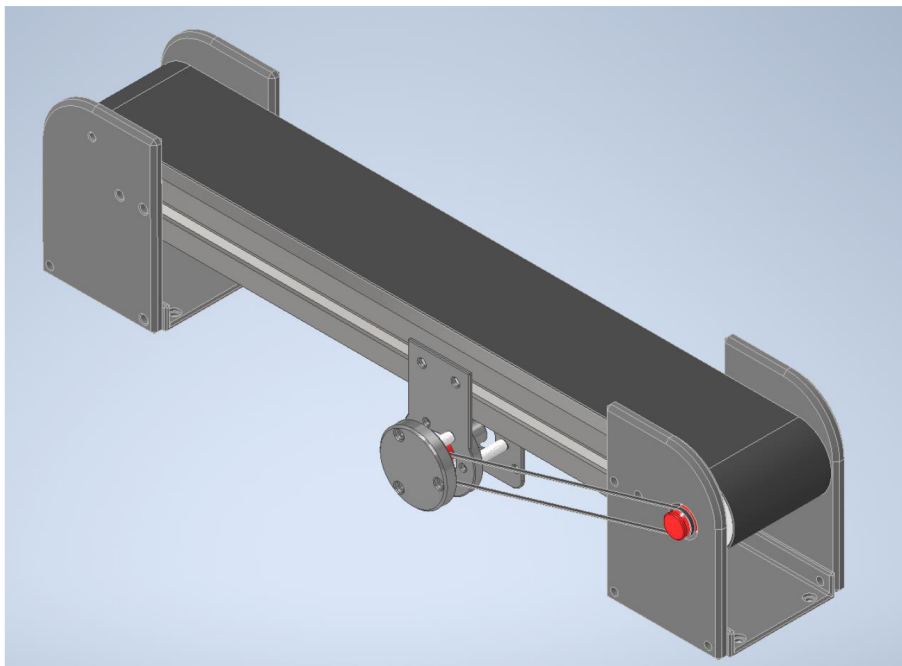
Obr. 4.2.2: Foto pásového dopravníku z bakalářské práce „*Návrh malého pásového dopravníku*“ [2]



Obr. 4.2.3: Foto pásového dopravníku z bakalářské práce „*Návrh malého pásového dopravníku*“ [2]

Návrh představený v této bakalářské práci je velice praktický, při návrhu laboratorní úlohy však bylo třeba provést několik úprav:

1. Cela konstrukce se zkrátila a zmenšila
2. Místo krokového motoru byl použit snáze ovladatelný DC motor s převodovkou řízený přes H můstek
3. Pro delší životnost motoru (přesněji jeho integrované převodovky) byla hnací kladka vybavena vlastní oddělenou hřídelí spojenou s motorem pružnou spojkou. Síly působící na hnací kladku jsou zachyceny přidáním ložisky, nikoliv přímo ložisky převodovky motoru.
4. Pás byl vybaven senzorem přítomnosti manipulované kostky, jež poskytuje nezbytnou zpětnou vazbu, umožňující efektivnější chod dopravníku. Tento senzor je dále nezbytný pro spolehlivý chod manipulátoru, který pro svou jednoduchost neposkytuje žádnou (prostředky navržené úlohy snadno zajistitelnou) zpětnou vazbu.

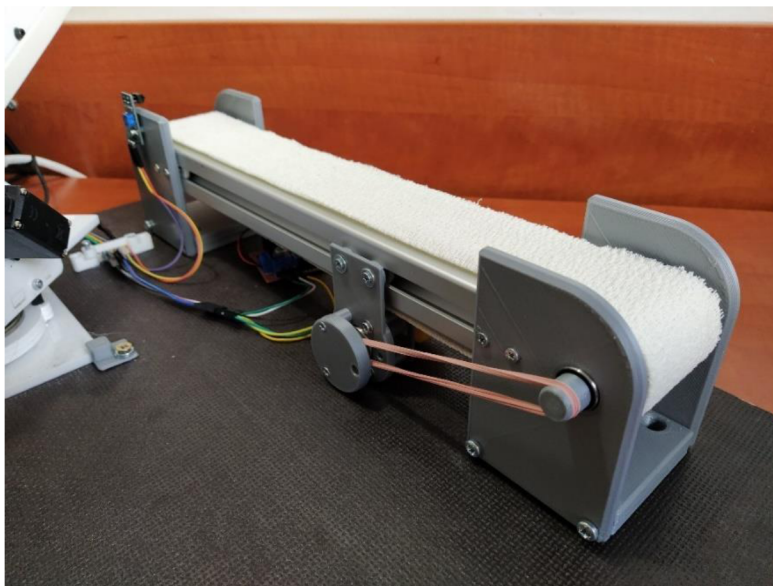


Obr. 4.2.4: Finální iterace návrhu pásového dopravníku

### 4.3 Realizace

Dopravník byl vyroben a sestaven v dílnách VUT FSI s asistencí vedoucího práce Ing. Zdeňka Cejпка. Převážnou většinu komponentů tvoří součástky vyrobené pomocí 3D

tisku. Pro zajištění snadného pohonu byla použita ložiska. Protože byla potřeba nestandardní šířka pásu 50mm, byl vyroben vlastní pas z elastického obvazu.



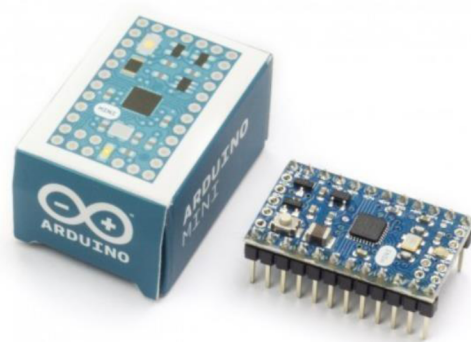
Obr. 4.3: Sestavený pásový dopravník

## 5 VÝBĚR VHODNÉ ŘÍDÍCÍ DESKY ARDUINO

Řídící deska je nezbytnou součástí úlohy. Řízení se uskuteční pomocí *MATLAB Support Package for Arduino*. To je rozšíření Matlabu, které dovoluje číst a posílat data přes řídící desku Arduino a okamžitě dostávat výsledky, aniž by se musel kompilovat kód. Při této spolupráci však může řídící desce snadno dojít paměť určená k uložení programu, a je proto nutné toto zohlednit již při výběru vhodného typu desky. Úkolem desky bude řízení nejen manipulátoru, ale i dopravníku se senzorem. Z těchto důvodů musí mít deska vhodný výkon i počet pinů. Na trhu se vyskytuje velké množství různých řídicích desek, ze kterých se volilo mezi deskou Arduino Mini Pro, ArduinoUno a Arduino Mega 2560.

### 5.1 Charakteristiky Arduino Mini Pro

*Arduino Mini Pro* je nejlevnější, má nejmenší výkon a počet pinů. Tato deska je velmi kompaktní, ale neobsahuje USB konektor ani převodník z USB/UART, přes který se komfortně přenáší již zkompilovaný program ve formátu .hex, či probíhá komunikace s prostředím Matlab na PC.



Obr. 5.1: Řídící deska Arduino Mini Pro [3]

## 5.2 Charakteristiky Arduino Mega 2560

*Arduino Mega 2560* je naopak ideální pro velmi rozsáhlé projekty. Deska je založená na mikrokontroleru ATmega2560, má větší výkon a 54 digitálních pinů, z nichž 4 jsou hardwarové sériové porty a 15 PWM výstupů. Nevýhodou je větší rozměr desky a vysoká cena.



Obr. 5.2: Řídící deska Arduino Mega 2560. [3]

## 5.3 Charakteristiky Arduino Uno

Pro realizace návrhu úlohy budou stačit parametry desky *Arduino Uno*, která byla v laboratoři k zapůjčení na vyzkoušení a osvědčila se. Pro použití v úloze byla zásadní podpora *MATLAB Support Package for Arduino*, kterou Arduino Uno má. Tato deska je populární nejen díky rozsáhlým možnostem realizace nejrůznějších hobby aplikací, ale i díky příznivé ceně a snadné dostupnosti. Řídící deska *Arduino Uno* je rozšířená mezi začátečníky a má velký počet dokumentací i návodů.

Protože od začátku sestavení úlohy bylo záměrem dodržet co nejnižší cenu celého projektu, byl zvolen kvalitně vypracovaný čínský klon *Arduino Uno*, který má shodné parametry a mnohem menší cenu.





Obr. 5.3: Řídící deska Arduino Uno. [3]

Níže je představená tabulka s podrobnými technickými charakteristiky *Arduino Uno* podle výrobce:

Tab. 1: Přehled dat řídicí desky Arduino Uno [3]

Microcontroller	ATmega328P
Operating Voltage	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limit)	6-20 V
Digital I/O Pins	6
Analog input Pins	6
DC Current per I/o Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	18 MHz
LED_BULTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

## 6 ELEKTRICKÉ VYBAVENÍ LABORATORNÍ ÚLOHY

### 6.1 DC MOTORY

Stejnoseměrné motory jsou elektromotory napájené stejnosměrným napětím. Výhodou těchto motorů je jejich cena a snadná regulace otáček pomocí změny napětí. Dále umožňují snadnou reverzaci chodu pomocí změny směru napětí, čímž lze motor i brzdít. Jejich řízení je levnější než řízení asynchronních motorů. [2]

V navržené úloze stejnosměrný motor slouží k pohonu pásového dopravníku. Pro tyto účely byl zvolen motor s integrovanou převodovkou:



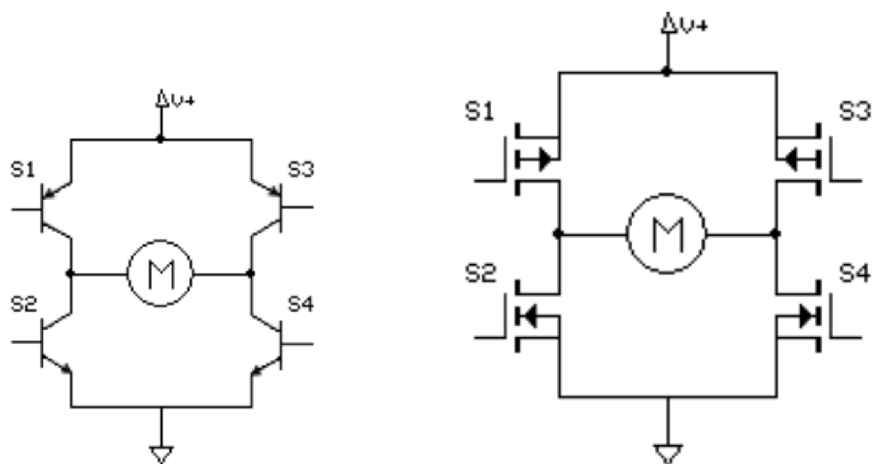
Obr. 6.1: DC motor s převodovkou – plastové převody [4]

Tab. 2: Parametry použitého DC motoru [4]

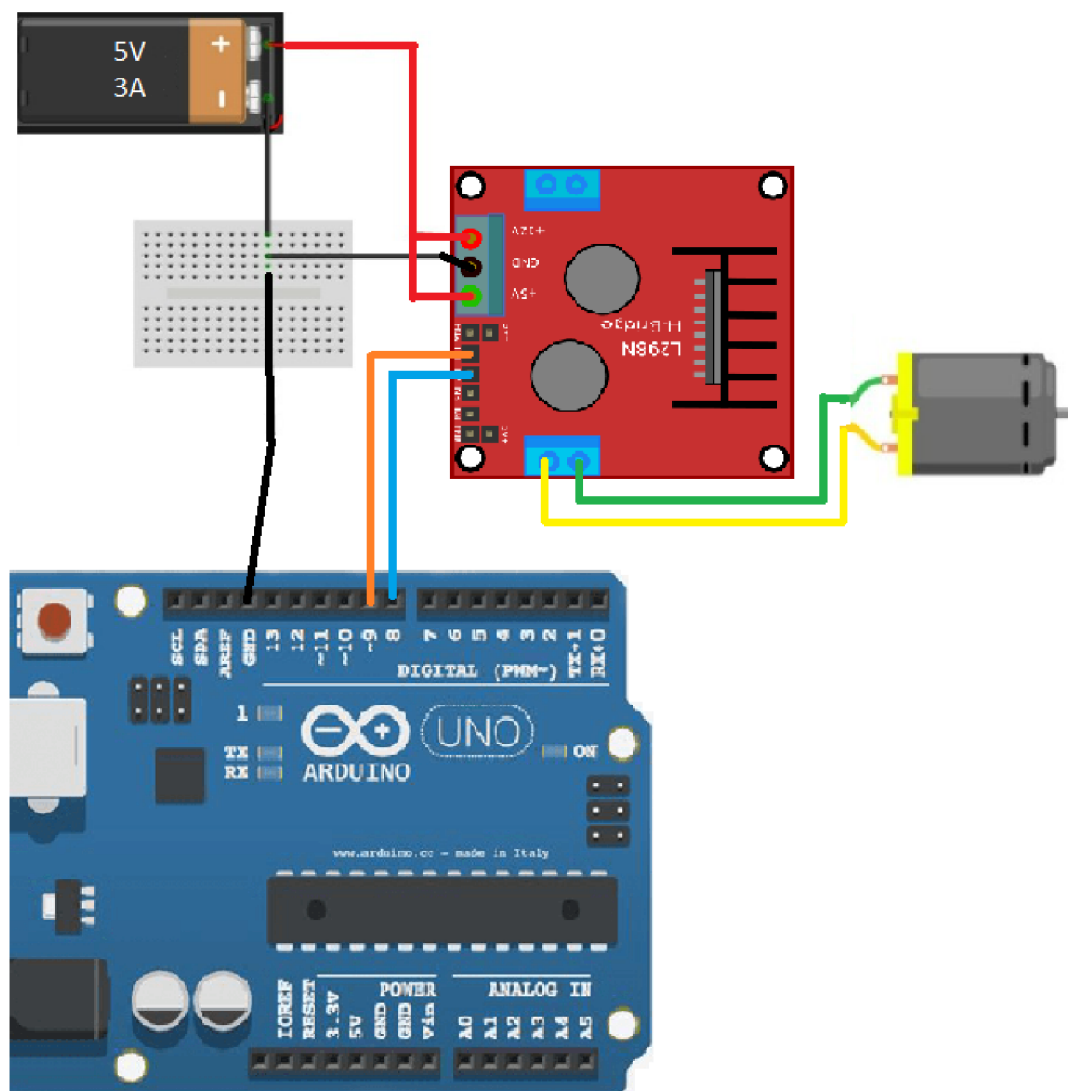
Napájení (V):	DC 3	DC 5	DC 6
Proud (mA):	100	100	100
Převod	48:1	48:1	48:1
Výstupní otáčky (ot/min):	114	197	240
Točivý moment(g/cm /3V):	800	800	800
Váha motoru (g):	50	50	50
Rozměry motoru(mm):	70x22x18	70x22x18	70x22x18
Hluk (dB):	65	65	65

## 6.2 H můstek

H můstek je elektrický obvod sestavený ze čtyř tranzistorů. Tyto tranzistory jsou schopné připojit oba póly zátěže k napájecímu napětí. Obvykle se prodávají integrované obvody či moduly, které obsahují dva H můstky. Tyto obvody je možné použít pro řízení jednoho krokového motoru nebo dvou stejnosměrných motorů [5].

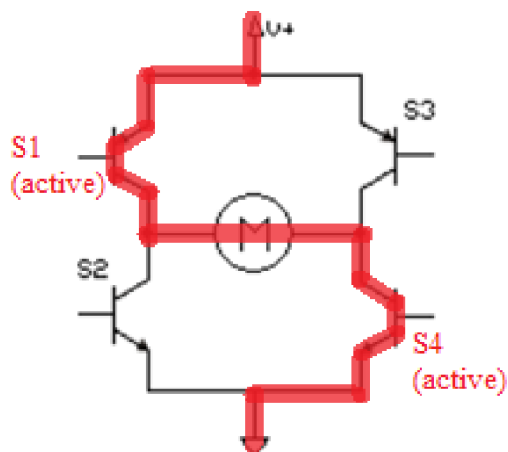


Obr. 6.2.1: Vlevo: Schéma *H-bridge* s bipolárními tranzistory; vpravo: *H-bridge* s MOSFET tranzistory [5].



Obr. 6.2.2: Připojení H můstku (*L298N*) a DC motoru k *Arduino*

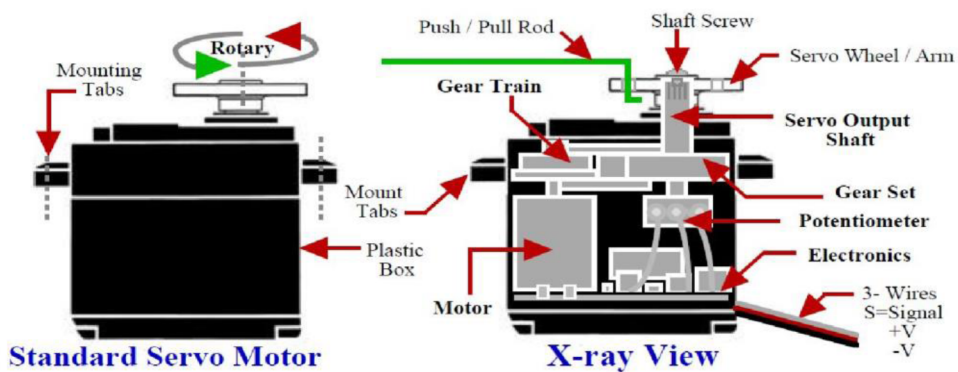
V úloze H můstek *L298N* bude sloužit pro řízení jednoho DC motoru, který pohání pásový dopravník. Vstupy č.1 a č.2 jsou určeny pro řízení směru otáčení motoru. Pokud na oba vstupy přicházejí stejné hodnoty (buď *LOW* nebo *HIGH*), motor se zastaví. Pokud se na jednom vstupu zachová digitální 0 a na druhý vstup H můstku se přivede digitální 1, sepnou se tím tranzistory *S1* a *S4* (Obr. 5.2.3) a motor se roztočí v jednom směru. Pokud digitální jednička bude na druhém vstupu H můstku, sepnou se tranzistory *S2* a *S3*, čímž se docílí otáčení motoru v opačném směru.



Obr. 6.2.3: Řízení stejnosměrného motoru přes H můstek

### 6.3 Modelářské servomotory

Servomotor není nic jiného než stejnosměrný motor s převodovkou, který je vybaven potenciometrem snímajícím natočení. Odpor potenciometru je zabudovanou řídicí elektronikou převeden na pulzy, jež jsou dále porovnávány s řídicími pulzy. Takto získaná informace o regulační odchylce, opět ve formě pulzů, je přímo použita k "dávkování" akční veličiny - napětí.



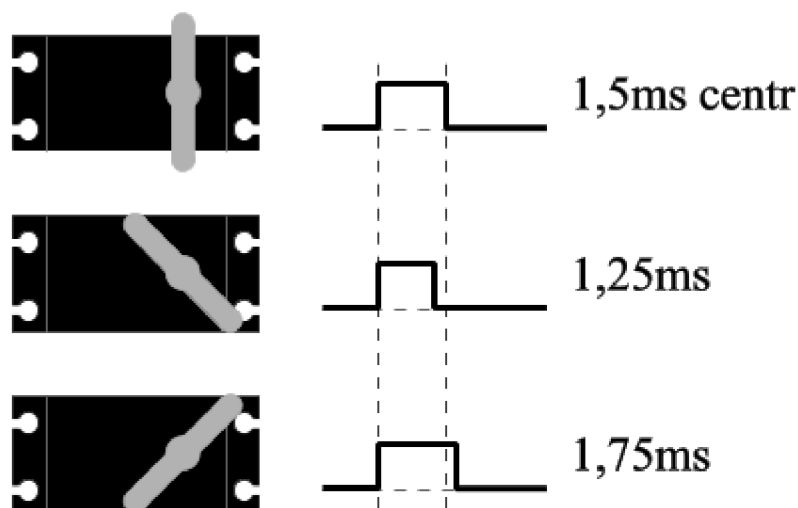
Obr. 6.3 Schéma servomotoru [6]

### 6.3.1 Ovládání servomotoru

Daná podkapitola je inspirovaná [6] a [7]

Součástí servomotoru jsou tři vystupující kabely (GND - země, VCC - napětí a řídicí signál S. Signální vodič dostává PWM signály poslané z externího řízení. Část obvodu analyzuje a zpracovává všechny příchozí signály. Tyto upravené signály pak slouží k řízení motoru. Servomotor se pohání hodnotou odporu, která je získána převáděním vstupního PWM signálu.

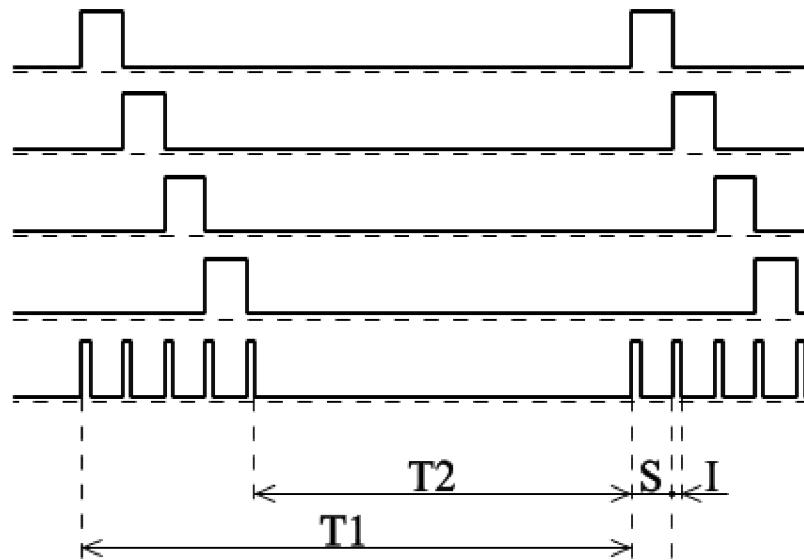
Dnes používané servomotory pracují s kladnými řídicími impulsy o délce 1- 2 ms. Jsou i takové, co pracují s pulzy délky 0,75 – 2,5 ms.



Obr. 6.3.1.1: Natočení výstupní páky servomotoru dle velikosti středy. Pro různé servomotory se uvedené hodnoty mohou lišit. [7]



Pulzní šířce signálu 1,5ms odpovídá střední poloha servomotoru, 1ms a 2ms jsou dvě protilehlé krajní polohy (přesné hodnoty se mohou u různých servomotoru lišit). Existují i taková mechanická provedení motoru, která umožňují pohyb v rozsahu větším než 180 stupňů, ale není to pravidlem. Většina servomotorů má na koncích rozsahu pohybu mechanické blokování. Při dojetí na doraz se tím výrazně zvýší odebíraný proud a může dojít ke shoření elektroniky servomotoru.

Na Obrázku 5.3.1.2 je znázorněna inteligentnější varianta řízení servomotoru, kdy každý servomotor odebírá výkon v sobě vyhrazeném čase. První čtyři průběhy vyjadřují impulsy na výstupu modelářského přijímače. Pátý průběh ukazuje výstupní signál kodéru modelářského vysílače pro 4 servomotory.



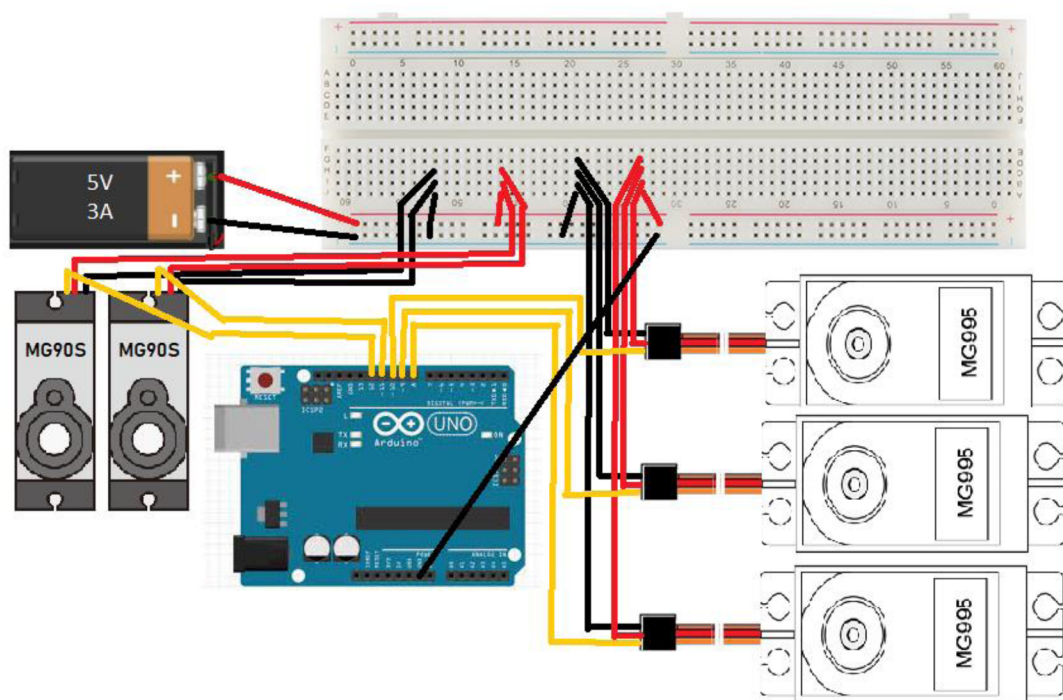
Obr. 6.3.1.2: Impulzy na výstupu modelářského přijímače. [7]

Dále jsou uvedené charakteristiky použitých servomotorů MG995 a MG90S:

	MG995		MG90S
Velikost serva	Micro	Velikost serva	Micro
Typ serva	analogové	Typ serva	analogové
Krouticí moment při 4,8V [kg·cm]	13,0	Krouticí moment při 4,8V [kg·cm]	1,6
Krouticí moment při 6,0V [kg·cm]	15,0	Krouticí moment při 6,0V [kg·cm]	1,8
Rychlost při 4,8V [s/60st.]	0,17	Rychlost při 4,8V [s/60st.]	0,10
Rychlost při 6,0V [s/60st.]	0,13	Rychlost při 6,0V [s/60st.]	0,09
Převody serva	Metal	Převody serva	Metal
Kuličkové ložiska	žádné	Kuličkové ložiska	žádné
Napájení [V]	4,8-6	Napájení [V]	4,8-6
Rozměry [mm]	40 x 19 x 43	Rozměry [mm]	22,6 x 12,1 x 22,5
Hmotnost [g]	75,0	Hmotnost [g]	12,0

Tab. 3,4 Přehled dat servomotoru MG995 a MG90S podle [8] a [9]



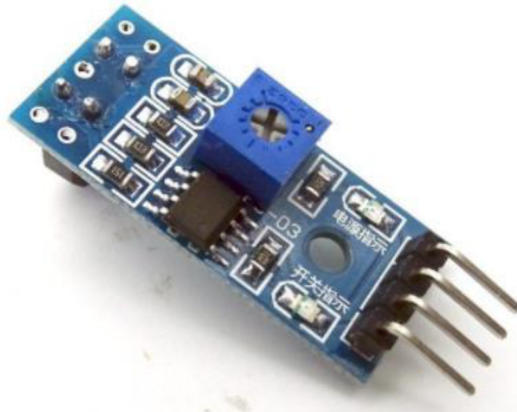


Obr. 6.3.1.3: Schéma zapojení servomotorů k Arduino

## 6.4 Vstupní senzory

### 6.4.1 Senzor přítomnosti

Senzor se skládá z infračerveného vysílače a detektoru záření. infračervený vysílač vyzařuje infračervené vlny, které se odrážejí od překážky a jsou zachyceny detektorem záření. Senzor detekuje překážky v rozsahu vzdáleností od nuly do nastaveného limitu. Je postaven na základě komparátoru LM393, který na výstup přivádí napětí podle principu: detekována překážka – logická úroveň HIGH, nedetekována - logická úroveň LOW, tento stav je indikován také zelenou LED na senzoru. Mezní sčítaná vzdálenost se nastavuje pomocí potenciometru nainstalovaného na modulu. Použitý senzor čáry má čtyři piny: VCC, GND, D0 a A0. Sčítat signál ze senzoru se může jak z analogového výstupu A0, tak i z digitálního D0. Pro detekci předmětů na pásovém dopravníku bude jednodušší využít výstup digitální. Napájení senzoru se může realizovat pomocí zdroje 5V. Detekční rozsah tohoto senzoru je 1 až 15mm. [10]

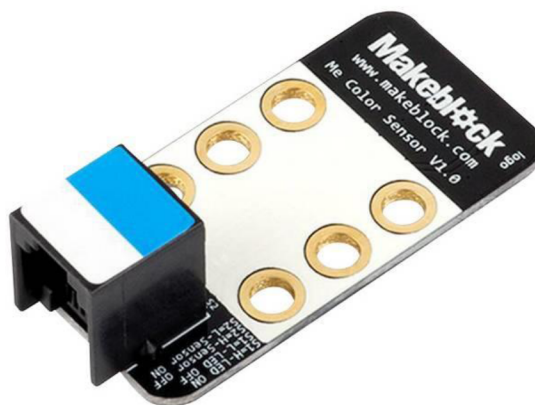


Obr. 6.4.1: Infračervený senzor sledování čáry s *LM393* [10]

### 6.4.2 Senzor barev

V navržené úloze dochází ke třídění podle barvy. Pro tyto účely slouží senzory, které jsou schopné detekovat a rozlišovat barvy podle spektrálního složení světla odráženého z povrchu. Senzor barev se skládá z několika důležitých částí:

1. Snímač, v sobě obsahuje fotocitlivé prvky a další prvky zajišťující zpracování signálu. V současné době se využívá systém obdobný lidskému oku, tedy snímání intenzity tří základních barev (vlnových délek) - RGB (Red, Green, Blue).
2. Zdroje světla, které mají za úkol osvětlovat měřený objekt definovaným způsobem a tím zlepšit spolehlivost výsledků.
3. Prvky, zajišťující processing naměřených hodnot a komunikaci po příslušné sběrnici.

Obr. 6.4.2: Použitý senzor *Me color sensor V1* [11]

Č.	Pin	Funkce
1	SCL	I2C communications (clock pin)
2	SDA	I2C communications (data pin)
3	GND	Ground line
4	VCC	Power line
5	S1	Fill light control pin
6	S2	Color acquisition control pin

Tab. 5: Piny použitého senzoru *Me Color Sensor V1* [11]

#### 6.4.2.1 Parametry použitého senzoru [11]

- Piny se standardní roztečí 2,54 mm pro připojení vodičů
- Obsahuje vestavěnou ochranu proti přepólování
- Má 6 otvorů formátu “LEGO”, je kompatibilní s detaily této značky.
- Těsně vedle snímače jsou umístěny dvě bílé světelné diody, které mají dvě bílé světodiody z obou stran těsně vedle snímače, které mají za úkol zajistit vhodné podmínky pro snímání barvy. Komunikace s řídicí deskou probíhá pomocí sběrnice I2C

## 6.5 Napájení

Pro připojení bylo použito nepájivé kontaktní pole a později deska plošných spojů (dál už jen DPS). Obsahuje vhodně umístěné konektory pro napájení motorů a signální kontakty. Napájení je realizováno z jednotky 5V 3A, ale pokud jsou plánována větší zatížení současně ve všech osách, pak je doporučeno použít napájecí zdroj, který je schopný dodat 5A.

# 7 ŘÍZENÍ ROBOTY PO VYBRANÉ SBĚRNICI

Samostatným bodem v zadání bakalářské práce je realizace řízení robotického ramene pomocí příkazů vysílaných po vybrané sběrnici. Kapitola 7 se bude zabývat řešením této problematiky.

V současné době se lze setkat s komunikačními rozhraními, jako například UART, SPI, RS-232, RS-485, CAN, Ethernet, I2C a řada dalších, jež je možné využít k zadávání příkazů z manipulátoru nadřazeného řídicího systému. Dále následuje krátký popis principu činnosti každého rozhraní.

## 7.1 Komunikační rozhraní UART

Sériové komunikační rozhraní UART využívá asynchronní přenos. Používá se tady *half-duplex* metoda komunikace. Pro přenos UART potřebuje dva datové vodiče: Tx (Transmit), použitý pro vysílání a Rx (Receiving) pro příjem. Aby se zajistila správnost komunikace, musí se dodržet zapojení těchto vodičů do kříže.

Ve výchozím stavu na obou datových vodičích je nastavená logická 1, start komunikace se tady signalizuje start bitem neboli logickou 0. Dále následuje 8 datových bitů. Zpráva končí stop bitem, uvádějícím datový vodič do výchozí logické úrovně. Tělo zprávy lze taky modifikovat jak paritním bitem, tak i dvěma stop bity, čímž se může zajistit ochrana přenášených dat a náhrada kontrolního součtu. Paritní bit je nastaven buď do logické 1 nebo 0, v závislosti na tom, jestli zpráva obsahuje sudý nebo lichý počet jedniček. Touto kontrolou se vylučuje případ, kdy vlivem rušení dojde ke znehodnocení jednoho bitu z původně dobře odeslané zprávy. Nevýhodou použití paritního bitu je to, že tím se o jeden bit zkracuje maximální možný obsah zprávy, tedy pouze 7 bitů.

Taktování rozhraní UART je řešeno pomocí parametru „*baud rate*“, který vyjadřuje počet bitů přenesených za sekundu. Aby došlo ke správnému zpracování posílané nebo přijímané zprávy, musí tento parametr mít na obou stranách stejné hodnoty. Díky nastavenému „*baud rate*“ pak obě zařízení ví, kdy mají daný bit odesílat a kdy přijímat.

Toto komunikační rozhraní je mnohdy využíváno pro rozsáhlé aplikace. Nevýhodou je jeho rychlost, která nepřevyšuje 115 kb/s. Tento problém byl vyřešen novým rozhraním nazývaným USART, které vychází z rozhraní UART. Rozdíl USART rozhraní je především v možnosti synchronizace za pomoci externího zdroje taktu. Vstupní pin bývá označován XCK. Rychlost tohoto rozhraní může být mnohonásobně vyšší (ne vyšší, než u SPI). Další nevýhodou je propojení pouze dvou zařízení skrze jednu sběrnici [12].

## 7.2 Komunikační rozhraní I2C

Komunikační rozhraní I2C je stejně jak UART sériovým rozhraním, využívajícím „*Master-Slave*“ typ komunikace. Pro zajištění přenosu zde také postačí dva signální kabely označené jako SDA a SCL. SDA (Serial Data) vodič zajišťuje obousměrný přenos dat. SCL (Serial Clock) vodič umožňuje zařízení typu *Master* posílat ostatním zařízením časový signál. V navržené úloze protokol I2C bude použit při komunikaci senzoru barev s řídicí deskou Arduino Uno.

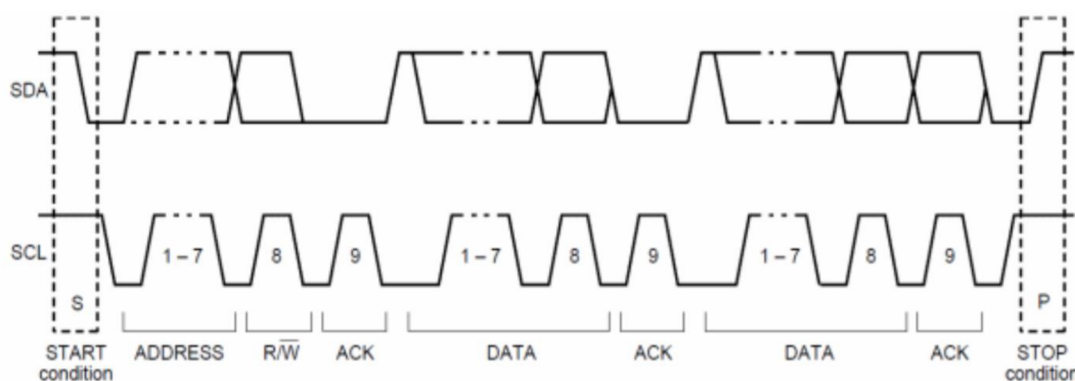
Tab. 6: Tabulka možných rychlostí sběrnice I2C [13].

Přenosová rychlost	Označení
10 kb/s	Low speed mode
100 kb/s	Standart mode
400 kb/s	Fast mode
1 Mb/s	Fast mode +
3,4 Mb/s	High speed mode

Pro úspěšnou spolehlivou komunikaci se v praxi na oba signální vodiče, jak SDA tak i SCL musí připojit „pull-up“ rezistory o odporu cca 1.5 kΩ. Je to nutné z důvodu potřeby logické jedničky na obou vodičích v momentě, kdy jsou všechny sběrnice nečinné.

U komunikačního rozhraní I2C, stejně tak u UART, má veškeré řízení sběrnice na starosti *Master*. Pouze *Master* je oprávněný začít komunikaci, což znamená, že na sběrnici nikdy nedojde ke kolizím. A protože komunikace je deterministická, při správné konfiguraci lze také zaručit i dobu odezvy. Nevýhodou komunikace typu “*Master-Slave*” je však to, že ostatní zařízení, která pracují jako *Slave*, nemohou nejen řídit sběrnici, ale i zažádat o vysílání nebo příjem dat. Jedním možným řešením je použití dotazovací metody “*polling*”. Podstata této metody spočívá v neustálém posílání dotazů na stav všech zařízení na sběrnici. Dalším řešením by mohlo být například použití zařízením *Slave* externího přerušovacího signálu neboli dalšího vodiče, přes který se bude *Master* informovat o nutnosti zahájení komunikace. Pokud je podporován, lze tento problém také vyřešit režimem “*Multi-Master*”.

Podobně jako u UART, se komunikace zahajuje pomocí start bitu a následujícím vysláním adresy zařízení, se kterým *Master* bude komunikovat. Pro ukončení komunikace se využívá stop bit. [13]



Obr. 7.3: Princip datového přenosu na sběrnici I2C [13].

### 7.3 Komunikační rozhraní SPI

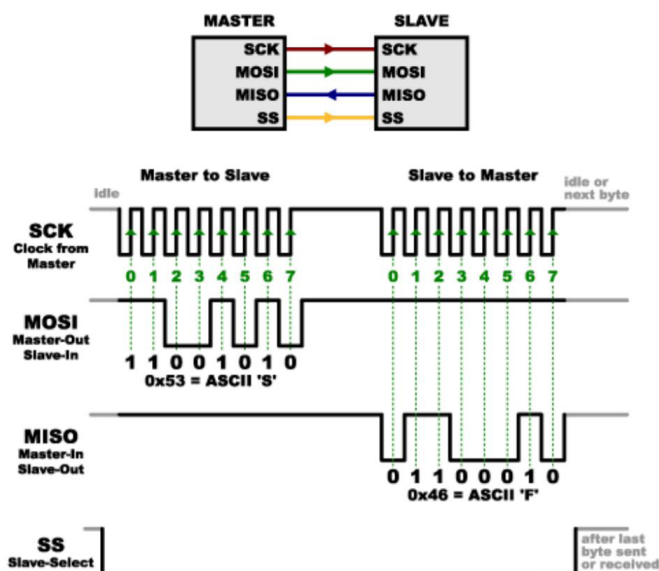
Komunikační rozhraní SPI je stejně jako předchozí sériové. Využívá ale synchronní přenos, a jelikož je to protokol s paralelní komunikací, pro propojení bude vyžadovat 4 vodiče:

- MISO (Master In, Slave Out)
- MOSI (Master Out, Slave In)
- SCK, který slouží pro synchronizace podobně jak SCL u komunikačního rozhraní I2C.
- SS.

Princip komunikace, podobně jako u UART a I2C, je typu „*Master-Slave*“. Protože komunikace je obousměrná, vzniká tady problém se zahájením komunikace mezi dvěma

jednotlivými zařízeními. Ten problém je vyřešen pomocí dalšího signálního vodiče SS, přes který *Master* posílá signál tomu zařízení, se kterým chce zahájit komunikaci. Nevýhodou je to, že se tímto vodičem musí připojit každé zařízení. Tím pádem vznikne v podstatě kruhový posuvný registr. Když se vybere zařízení, se kterým master bude komunikovat, vstupuje SCK vodič, který dále řídí přenos. U SPI je více možností pro synchronizaci dat, například nástupní nebo sestupná hrana, úroveň SCK v době nečinnosti. Data z *masteru* jsou přenášena vodičem MOSI, a zároveň jsou s každým taktem posunuta i ze *slave* do *masteru* vodičem MISO.

Sběrnice SPI má velmi vysokou taktovací frekvenci a nemusí se přepínat mezi vysíláním a příjmem, avšak je nevhodná pro velké vzdálenosti, nemá ACK bit a komunikace s větším množstvím zařízení je pro toto rozhraní obtížná. [14]

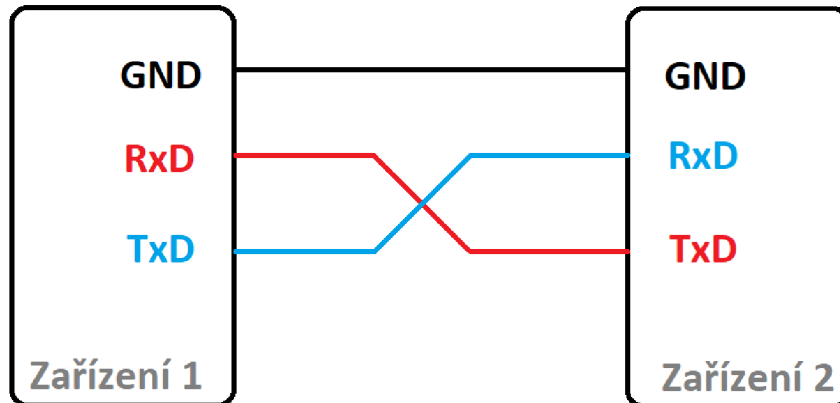


Obr. 7.4: Princip datového přenosu na sběrnici SPI. [14]

## 7.4 Komunikační rozhraní RS-232

Sériové komunikační rozhraní RS-232 je určeno pro dvoubodové spojení, což znamená, že propojení je možné pouze mezi dvěma zařízeními. Toto rozhraní má podobný princip přenosu jak UART. Za svou dlouhou historii rozhraní RS-232 bylo používáno až s 22 signálními vodiči (TXD - out, RXD - in, RTS, CTS a tak dále). Standardní linka ale obsahuje celkově jenom 9 datových a řídicích vodičů, a pokud není potřeba data přenášet synchronně, budou pro komunikaci stačit jenom tři vodiče: TxD, RxD a GND. [13]

Sériové komunikační rozhraní RS-232 je jedním ze starších rozhraní. Nicméně v současné době je pořád využíváno a je průmyslovým standardem.



Obr. 7.5: Princip propojení RS-232.

## 7.5 Komunikační rozhraní RS-485

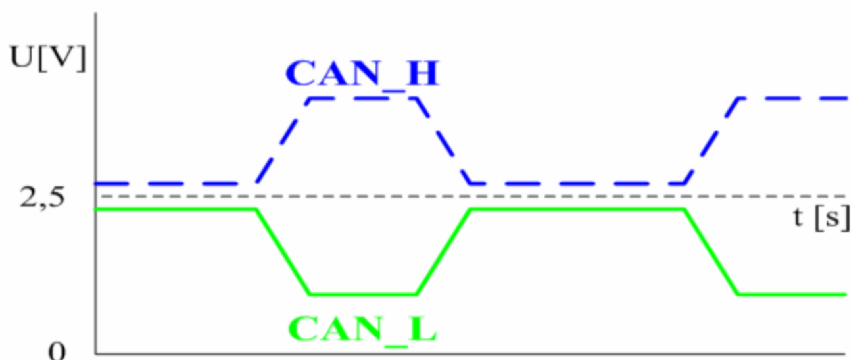
Rozhraní RS-485 má stejnou podstatu jako rozhraní RS-232. Liší se hlavně v úrovních napětí, které reprezentují logickou 1 nebo 0. Na rozdíl od RS-232 rozhraní RS-485 vztahuje úroveň napětí k protějšímu vodiči. Tímto se docílí mnohem vyšší odolnosti proti rušení a spolehlivé komunikace na větší vzdálenosti. Signální dráty jsou označeny jako A a B, ale existuje také *full duplex* varianta zapojení využívající 4 vodiče. [15]

## 7.6 Komunikační rozhraní CAN

Komunikační rozhraní CAN využívá synchronní typ komunikace „*multimaster*“. Synchronizace je vnitřní a provádí se pravidelným zasíláním logické 1 a 0, které se střídají. Pokud se vysílá více než 5 bitů na stejné logické úrovni, přidává se ke zprávě tzv. *bit stuffing* nebo bit opačné úrovně. Tím se docílí synchronizace vždy maximálně po pěti bitových intervalech.

Podstatný rozdíl mezi sběrnici CAN a RS-485 spočívá ve vyhodnocení logických úrovní na sběrnici. CAN dokáže rozlišit dvě úrovně signálu. První úroveň je dominantní. Odpovídá úrovni signálu logické 1 a nastává při větším než 0,9 V rozdílu napětí mezi CAN\_H a CAN\_L. Druhou úrovní je úroveň recesivní, která naproti tomu odpovídá logické 0 a vyjadřuje rozdíl napětí menší než 0,5 V. [16]





Obr. 7.6: Logické úrovně na sběrnici CAN [16]

## 7.7 Komunikační rozhraní Ethernet

Rozhraní Ethernet je v současné době velmi používané a oblíbené. Využívá se hlavně pro počítačové sítě. Komunikace na rozhraní Ethernet je realizována pomocí paketů. Díky další verzi IPv4 – IPv6 je přenášený paket výrazně jednodušší. Paket tvoří preamble, MAC adresu komunikujících zařízení, protokoly vyšších vrstev a datové části s kontrolním součtem. [17]

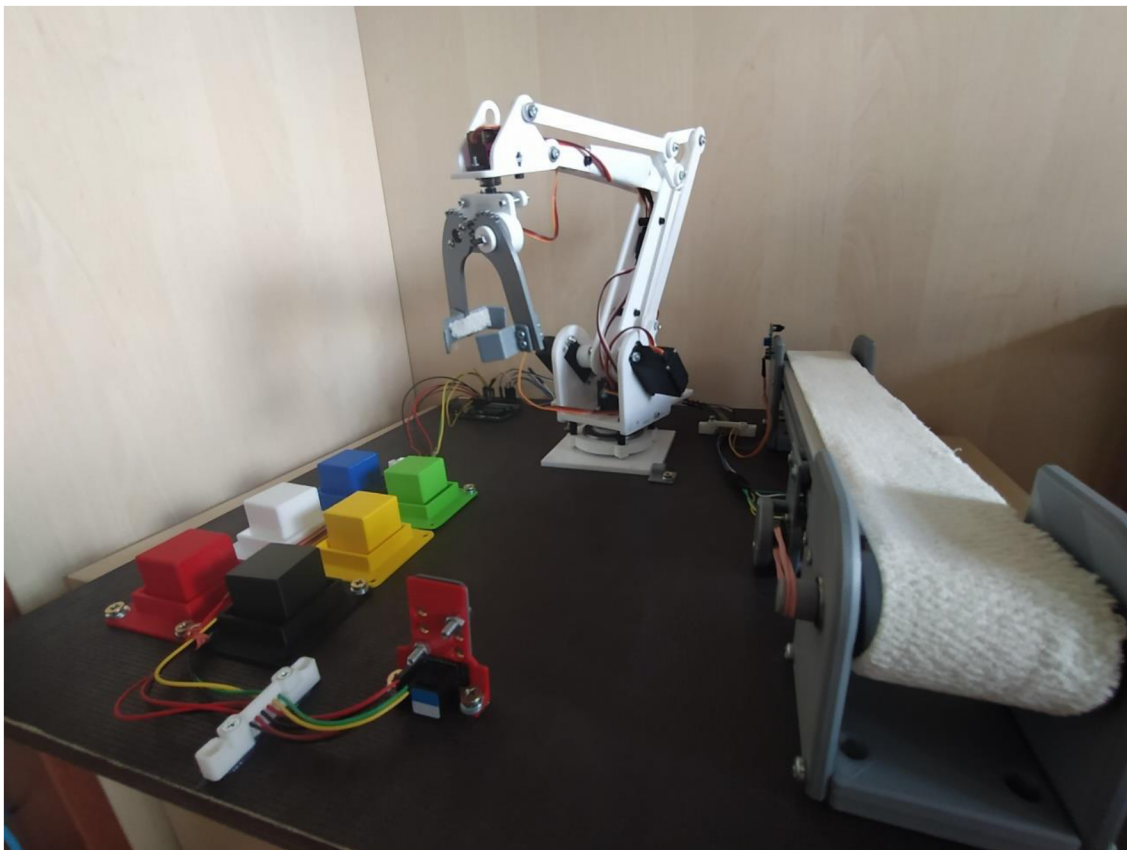
## 7.8 Realizace řízení ramene po SPI sběrnici

Pro řízení robota byla vybrána sběrnice SPI, která podporuje *full-duplex* komunikaci a na mikrokontroleru ATmega328P má hardwarovou podporu. Dalším důvodem výběru této sběrnice bylo přání autora, jelikož s touto sběrnici nikdy nepracoval a chtěl si to vyzkoušet. Řízení se provádělo pomocí dvou řídicích desek Arduino Uno, ze kterých jedna vystupuje v roli *Master* a druhá v roli *Slave*. Pro vyzkoušení řízení byl pro každé zařízení napsán zkušební kód (viz přílohy), jehož výstupem jsou hodnoty posílané přímo na servomotory uvádějící do pohybu manipulátor:

- Master Arduino Code – skript, zajišťující vysílání dat z masteru a kontrolu
- Slave Arduino Code – skript, příjem a vracení dat do masteru, následující zpracování a nastavení servomotoru podle přijatých dat.

## 8 REALIZACE LABORATORNI ÚLOHY

Při návrhu se předpokládala realizace s použitím 3D tisku. Technologie 3D tisku nezaručuje úplně přesné výsledné výrobky, ale vzhledem k relativně rychlému a levnému procesu výroby je ideální pro návrhy a testovací verze součástek. Dalším důvodem výběru technologie 3D tisku pro výrobu součástek je možnost seznámit studenta s moderními trendy výroby.



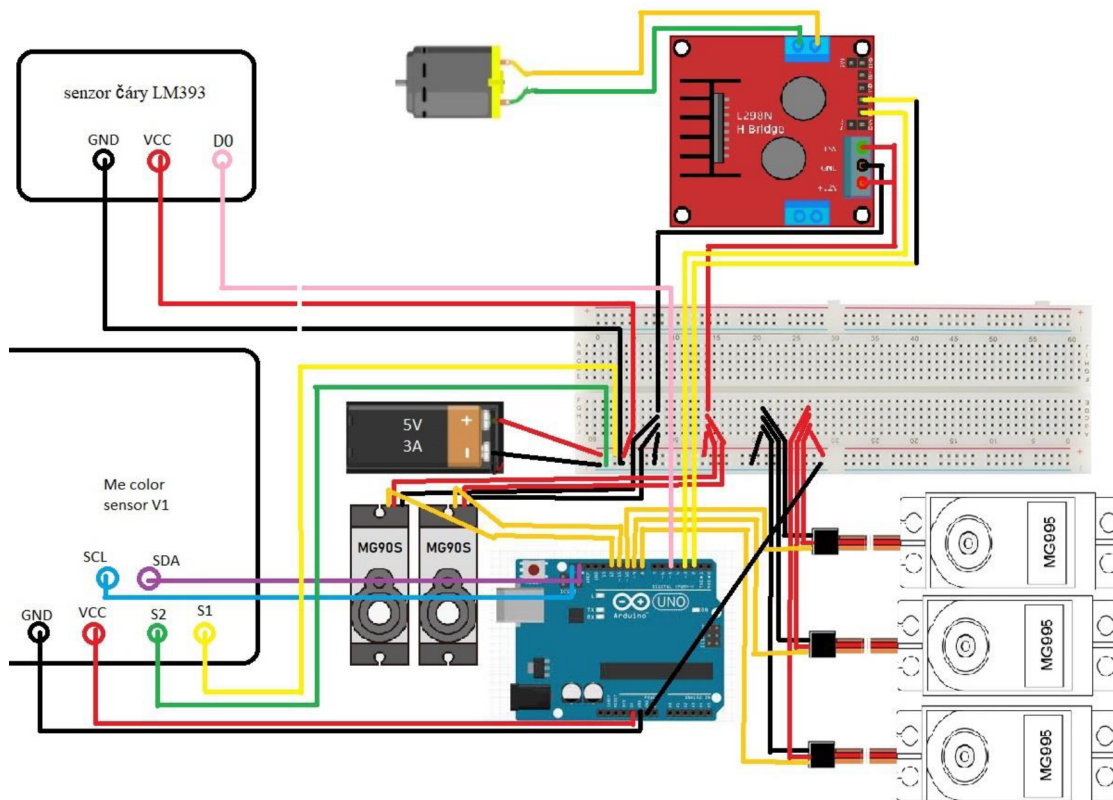
Obr. 8.1: Foto realizované úlohy

Po realizaci se osvědčilo několik nedostatků:

- Nízkou cenou jsou dány absence zpětné vazby s manipulátorem a nepříliš tuhá konstrukce.
- Použité čínské servomotory mají významný vliv na přesnost prací manipulátoru. Každý kloub se může v rámci 2-3 stupně kmitat, což může způsobit odchylky při dosažení stejné pozice z různých stran. Tento problém lze částečně odstranit programovou korekcí při nájezdu na pozici.

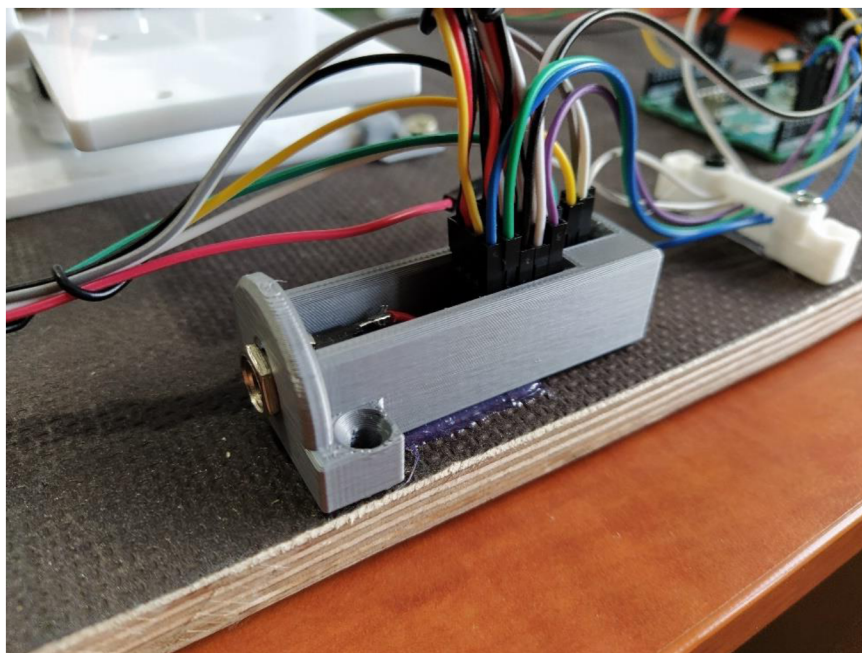
- Později se osvědčilo, že infračervený senzor sledování čáry nebyl vhodnou volbou pro zajištění zpětné vazby mezi dopravníkem a řídicí deskou. Pokud při snímání senzor bude orientován směrem k oknům, dopadem paprsků denního světla může dojít ke ovlivnění výsledku měření.

## 8.1 Zapojení celé soustavy do řídicí desky Arduino Uno



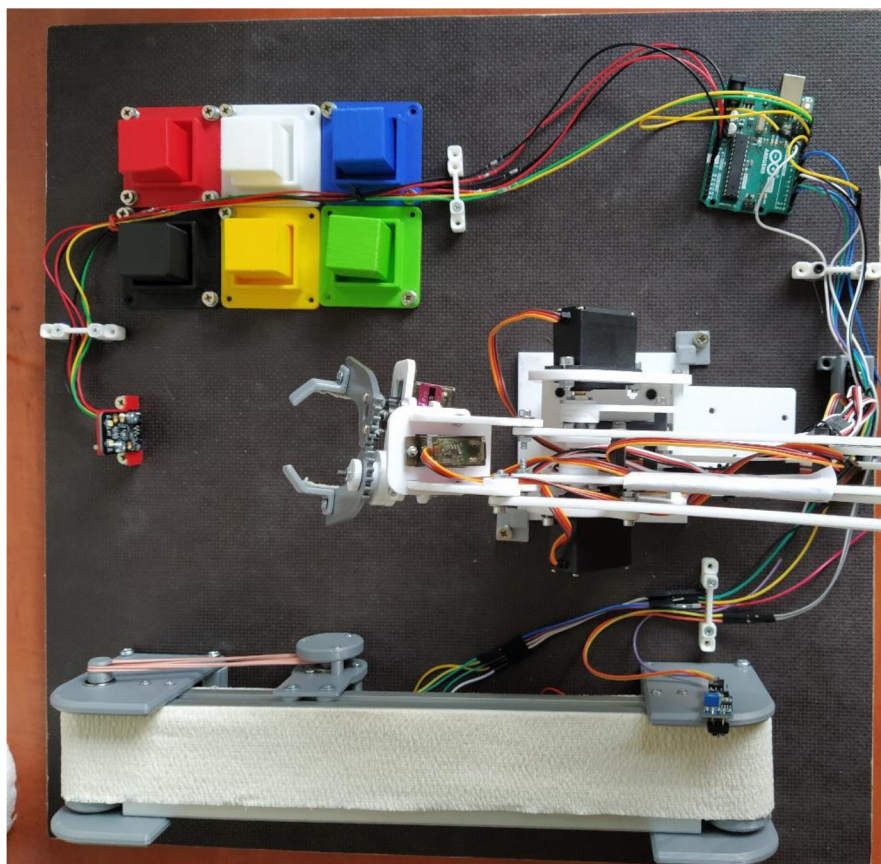
Obr. 8.2: Zapojení celé soustavy do řídicí desky Arduino Uno

Nepájivé kontaktní pole se později zaměnilo za speciálně vyrobenou kompaktnější desku plošných spojů, ke které jsou připojeny konektory:



Obr. 8.1.2: Konektor

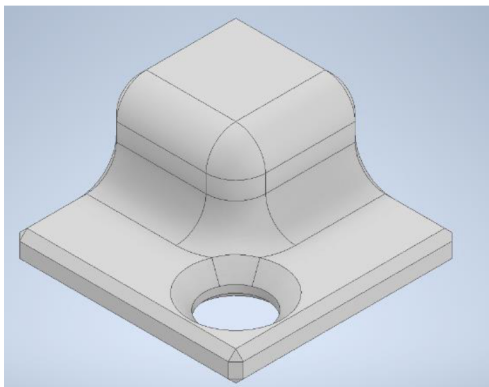
Pro zajištění přehledného a současně bezpečného uložení drátů se použily speciálně vytištěné součástky:



Obr. 8.1.3: Pohled na ulohu shora

## 8.2 Montáž

Jednotlivé komponenty úlohy se upevnily na dřevěné desce 50x50cm pomocí vrutu, protože se u základní konstrukce manipulátoru nepředpokládala montáž, pro tento účel byly vytištěny speciální součástky:



Obr. 8.2: Držák pro manipulátor

## 8.3 Řízení úlohy pomocí Matlabu

Úloha je rozdělená na více částí a ke každé části je vypracované samostatné ukázkové řešení. Celkovým řešením a hlavním výstupem práce jsou soubory:

- **Initialization.m** – skript, který se musí spustit jako první. Uvnitř tohoto souboru se definuje port počítače, ke kterému je připojena řídicí deska Arduino a přidávají se knihovny „Servo.h“ a „I2C.h“. V sekci, která se jmenuje „*color sensor part*“ je vykonána inicializace senzoru barev. V sekci „*servomotor part*“ se přiřazují signální piny pro řízení servomotorů a nastavuje se jejich počáteční poloha. Soubor „*Initialization*“ je nutné spustit jednou před začátkem práce, a protože nemusí být zřejmé v jakých pozicích se vyplí servomotory naposled, napájení servomotorů je doporučeno připojit až celý skript proběhne.

**PresentationCode.m** – hlavní soubor, ve kterém jsou uspořádány funkce zajišťující běh úlohy. Aby se zajistila cykličnost úlohy, veškeré funkce byly umístěny do nekonečného cyklu „*while*“. Uvnitř cyklu „*while*“ je smyčka „*for b=1:x*“, která zajistí vykonání programu pro  $x$  krabiček (max. 6ks). Průběh zpracování každé krabičky začíná nastavením manipulátoru do „*home*“ pozice. Dále následuje kontrola přítomnosti krabičky prozatím neznámé barvy na výdejním místě dopravníku. Pokud tam krabička není, spustí se pás, a zůstane zapnutý, dokud senzor nezaznamená přítomnost krabičky. Když je krabička detekována, pás se zastaví a manipulátor

se přemístí do pozice „*point1*“, která je přímo nad výdejním místem kde se nachází krabíčka. Dále se rameno postupně přemístí do pozic „*point2*“ a „*point3*“, čímž krabíčku uchopí a nadzvedne jí nad pásovým dopravníkem. Příkazy umístěné uvnitř sekce „*point4*“ zajišťují posun koncového bodu manipulátoru s krabíčkou do pracovního prostoru senzoru barev, kde se díky kódu v sekci „*Check the color of the box*“ stanoví barva momentálně manipulované součástky. Poslední sekce „*Move box with defined color to spot of the same color*“ zahrnuje přesun krabíčky do zásobníku příslušné barvy. Po ukončení zpracování všech krabíček následuje náhodné přiřazování pořadí pro vrácení krabíček zpět do pásového dopravníku. Sekce "moving boxes back" je zodpovědná za přesunutí vybraných krabíček zpět na pásový dopravník.

- **GetBoxColor.m** – funkce, zahrnující příkazy pro sčítání dat z registru senzoru barev a logiku pro vyhodnocení sčítané barvy.
- **smoothMove.m** – soubor zahrnující funkci, která zajišťuje plynulý pohyb manipulátoru s možností nastavení rychlosti.
- **moveManipulatorToHomePosition.m** – obsahuje příkazy pro nastavení manipulátoru do výchozí „*home*“ pozice
- **moveManipulatorToBoxBeltReadyToTake.m** – obsahuje příkazy pro nastavení manipulátoru do pozice nad výdejním místem pásu.
- **moveManipulatorToBoxBeltTake.m** – obsahuje příkazy pro přenesení krabíčky na pás.
- **moveManipulatorBackToBoxBeltReadyToTake.m** - obsahuje příkazy pro vrácení manipulátoru do pozice nad výdejním místem pásu (liší se od `moveManipulatorToBoxBeltReadyToTake.m` jenom pořadím příkazu) .
- **moveManipulatorToColorSensorReadPosition.m** – soubor zahrnující řadu příkazů pro posun koncového bodu manipulátoru spolu s krabíčkou do pracovního prostoru senzoru barev
- **moveBoxToItsColorSpot.m** – Funkce zajišťující přesun krabíčky do příslušného odkladiště.
- Šest souboru `bringBlackBoxBackOnBelt.m`, `bringBlueBoxBackOnBelt.m`, `bringGreenBoxBackOnBelt.m`, `bringRedBoxBackOnBelt.m`, `bringWhiteBoxBackOnBelt.m` a `bringYellowBoxBackOnBelt.m` zahrnují v sobě příkazy pro přesun krabíček jednotlivých barev zpět do pásového dopravníku.

## 9 ZÁVĚR

Cílem této bakalářské práce bylo oživení již hotového robotického ramene a jeho nasazení do laboratorní úlohy typu "pick and place". Řízení bylo realizováno na platformě Arduino řízené z prostředí Matlab. K tomuto účelu byla využita knihovna „MATLAB Support Package for Arduino Hardware“.

Při návrhu mechanických částí bylo hojně využíváno modelování v prostředí Autodesk Inventor a technologie 3D tisku.

Dále byl sestaven pásový dopravník sloužící jako „zdroj“ manipulovaných objektů. K systému byl připojen i senzor barev, díky kterému lze manipulované objekty třídit dle barvy. Pro tato doplňková zařízení byl v prostředí Matlab připraven jednoduchý interface.

Výše zmíněné prvky byly spojeny do výukového setu. K tomuto výukovému setu byl sestaven komplexnější program. Tento program provádí rozřídění manipulovaných objektů odebíraných z pásového dopravníku. Po rozřídění dochází k náhodnému navrácení manipulovaných objektů zpět na pásový dopravník. Tento program poslouží k demonstraci aplikace nejen ve výuce, ale rovněž při prezentaci veřejnosti, neboť pracuje v uzavřené smyčce - bez nutnosti navracet manipulované objekty zpět na vstupní dopravník.

Pro nasazení ve výuce byly sestaveny jednodušší úlohy a návod použitelný v předmětu IIN-Informatika.





## SEZNAM POUŽITÉ LITERATURY

- [1] CustomElectronics. *Робо-пыка* [online]. [cit. 03.04.2021]. Dostupné z: <http://www.customelectronics.ru/>
- [2] DRAŽKA, Vojtěch. *Návrh malého pásového dopravníku* [online]. Brno, 2020 [cit. 04.04.2021]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/125034>.
- [3] ARDUINO. *Arduino products* [online]. [cit. 06.04.2021]. Dostupné z: <https://store.arduino.cc/arduino>
- [4] Laskarduino. *TT motor s převodovkou - plastové převody* [online]. 2021 [cit. 10.04.2021]. Dostupné z: <https://www.laskarduino.cz/tt-motor-s-prevodovkou-plastove-prevody/>
- [5] PAŠTYKA, Jan. DC motory řízené pomocí H můstku. *České vysoké učení technické v Praze*. 2017. Dostupné také z: [https://embedded.fel.cvut.cz/sites/default/files/kurzy/lpe/h-bridge/H\\_Bridge.pdf](https://embedded.fel.cvut.cz/sites/default/files/kurzy/lpe/h-bridge/H_Bridge.pdf)
- [6] MAREČEK, P. *Manipulátor pro třídění kuliček řízený PLC*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. Vedoucí bakalářské práce Ing. Lukáš Knispel.
- [7] Vlastikd. *Jak funguje modelářské servo* [online]. [cit. 15.04.2021]. Dostupné z: <http://vlastikd.webz.cz/bastl/serva.htm>
- [8] AliExpress. *MG995 High Torque Servo* [online]. [cit. 15.04.2021]. Dostupné z: [https://aliexpress.ru/item/32854187688.html?spm=a2g0s.9042311.0.0.27424c4d7oO1uG&sku\\_id=65269401290](https://aliexpress.ru/item/32854187688.html?spm=a2g0s.9042311.0.0.27424c4d7oO1uG&sku_id=65269401290)
- [9] AliExpress. *MG90S Digital Servo* [online]. [cit. 15.04.2021]. Dostupné z: [https://www.aliexpress.com/item/32952046848.html?spm=a2g0s.9042311.0.0.27424c4d7oO1uG&sku\\_id=66302641688](https://www.aliexpress.com/item/32952046848.html?spm=a2g0s.9042311.0.0.27424c4d7oO1uG&sku_id=66302641688)
- [10] LASKARDUINO. *Infračervený senzor sledování čáry s LM393*. [online]. [cit. 16.05.2021]. Dostupné z: <https://www.laskarduino.cz/arduino-infracervený-senzor-sledovani-cary-s-lm393/>.
- [11] <https://www.hwkitchen.cz/>. *ME SNÍMAČ BAREV* [online]. 2021 [cit. 23.05.2021].
- [12] CHYTIL, L. *Řídicí systém s HMI displejem*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 82 s., 11 s. příloh. Diplomová práce. Vedoucí práce: prof. Ing. Jaroslav Boušek, CSc.
- [13] I2C., Komunikace. *Root.cz*. [online]. Brno: Root.cz, 2009 [cit. 2021-05-13] Dostupné z: <https://www.root.cz/clanky/komunikace-po-seriove-sbornici-isup2supc/>.

- [14] (SPI)., Serial. *Sparkfun*. [online]. USA: Sparkfun, 2012 [cit. 2021-05-14]. Dostupné z: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/slave-selectss>.
- [15] RS-485. *Wikipedie*. [online]. Praha: Wikipedie, 2016 [cit. 2021-04-20]. Dostupné z: [https://cs.wikipedia.org/wiki/RS-485#Technick.C3.A9\\_parametry](https://cs.wikipedia.org/wiki/RS-485#Technick.C3.A9_parametry).
- [16] PATA, Jaroslav. *Senzorický systém mobilního robotu*. Brno, 2008. 59 s. , CD-ROM. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí bakalářské práce Ing. František Burian.
- [17] ETHERNET. *Wikipedie*. [online]. Wikipedie, 2017 [cit. 2021-03-17]. Dostupné z: <https://cs.wikipedia.org/wiki/Ethernet>.
- [18] KOLÍBAL, Zdeněk. *Roboty a robotizované výrobní technologie*. Brno: Vysoké učení technické v Brně - nakladatelství VUTIUM, 2016.
- [19] ZEŽULKA, František. *Prostředky průmyslové automatizace*. Brno : VUTIUM, [2004]. 161 s. .

## SEZNAM OBRÁZKŮ A TABULEK

- Obr. 2.2: 3D model finální verze návrhu úlohy.
- Obr. 3.1: Model kleští pro manipulace s krabičky.
- Obr. 3.3.1.1 Model manipulátoru: pohled z boku.
- Obr. 3.3.1.2: Umístění os v úloze.
- Obr. 4.1: Vertikální zásobník.
- Obr. 4.2.1: První iterace návrhu pásového dopravníku.
- Obr. 4.2.2: Foto pásového dopravníku z bakalářské práce „Návrh malého pásového dopravníku“.
- Obr. 4.2.3: Foto pásového dopravníku z bakalářské práce „Návrh malého pásového dopravníku“.
- Obr. 4.2.4: Finální iterace návrhu pásového dopravníku.
- Obr. 4.3: Sestavený pásový dopravník.
- Obr. 5.1: Řídící deska Arduino Mini Pro
- Obr. 5.2: Řídící deska Arduino Mega 2560.
- Obr. 5.3: Řídící deska Arduino Uno.
- Obr. 6.1: DC motor s převodovkou – plastové převody.
- Obr. 6.2.1: Vlevo: Schéma H-bridge s bipolárními tranzistory; vpravo: H-bridge s MOSFET tranzistory.
- Obr. 6.2.2: Připojení H můstku (L298N) a DC motoru k Arduino.
- Obr. 6.2.3: Řízení stejnosměrného motoru přes H můstek.
- Obr. 6.3 Schéma servomotoru.
- Obr. 6.3.1.1: Natočení výstupní páky servomotoru dle velikosti střídy.
- Obr. 6.3.1.2: Impulzy na výstupu modelářského přijímače.
- Obr. 6.3.1.3: Schéma zapojení servomotorů k Arduino.
- Obr. 6.4.1: Infračervený senzor sledování čáry s LM393.
- Obr. 6.4.2: Použitý senzor Me color sensor V1.
- Obr. 7.3: Princip datového přenosu na sběrnici I2C.
- Obr. 7.4: Princip datového přenosu na sběrnici SPI.
- Obr. 7.5: Princip propojení RS-232.
- Obr. 7.6: Logické úrovně na sběrnici CAN.
- Obr. 8: Foto realizované úlohy.
- Obr. 8.1.1: Zapojení celé soustavy do řídicí desky Arduino Uno.
- Obr. 8.1.2: Konektor.
- Obr. 8.1.3: Pohled na úlohu shora.
- Obr. 8.2: Držák pro manipulátor.

Tab. 1: Přehled dat řídicí desky Arduino Uno.

Tab. 2: Parametry použitého DC motoru.

Tab. 3,4 Přehled dat servomotoru MG995 a MG90S.

Tab. 5: Piny použitého senzoru Me Color Sensor V1.

Tab. 6: Tabulka možných rychlostí sběrnice I2C.

# SEZNAM PŘÍLOH

## **PŘÍLOHA I**

Návod

- 1) Laboratorní cvičení z IIN - Ovladání robotického ramene.odt

## **PŘÍLOHA II**

3D model

- 1) 3D\_Uloha

## **PŘÍLOHA III**

Kódy napsané v Arduino IDE

- 1) SPIcode

## **PŘÍLOHA IV**

Kódy napsané v Matlab

- 1) bringBlackBoxBackOnBelt.m
- 2) bringBlueBoxBackOnBelt.m
- 3) bringGreenBoxBackOnBelt.m
- 4) bringRedBoxBackOnBelt.m
- 5) bringWhiteBoxBackOnBelt.m
- 6) bringYellowBoxBackOnBelt.m
- 7) GetBoxColor.m
- 8) Initialization.m
- 9) moveBoxToItsColorSpot.m
- 10) moveManipulatorBackToBoxBeltReadyToTake.m
- 11) moveManipulatorToBoxBeltReadyToTake.m
- 12) moveManipulatorToBoxBeltTake.m
- 13) moveManipulatorToColorSensorReadPosition.m
- 14) moveManipulatorToHomePosition.m
- 15) PresentationCode.m
- 16) smoothMove.m

- 17) Kinematika

## **PŘÍLOHA V**

Video ukázka

- 1) Video