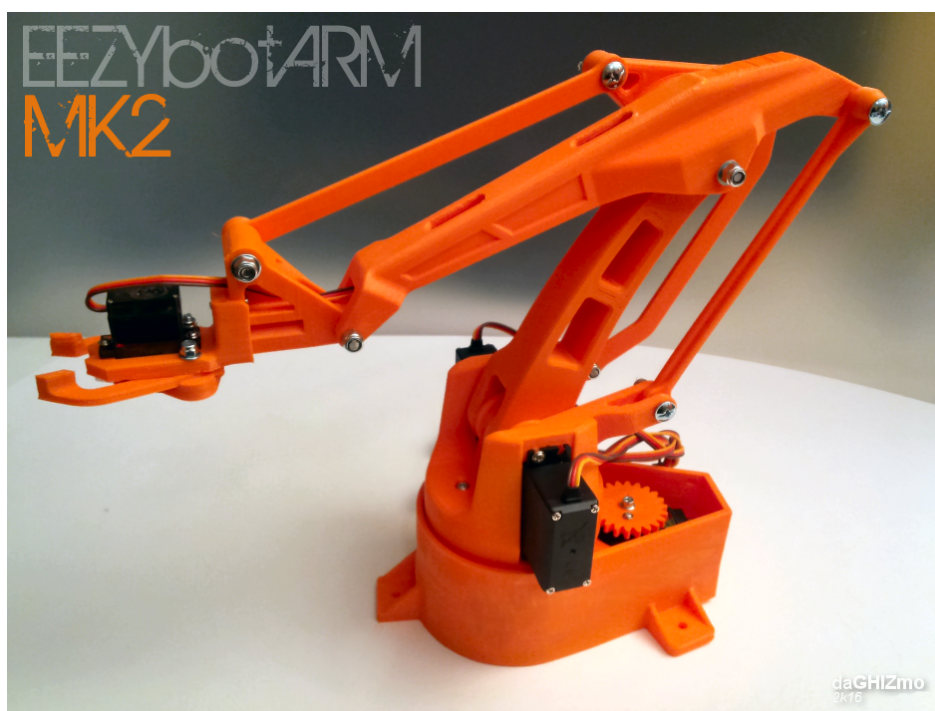


# ZÁVĚREČNÁ STUDIJNÍ PRÁCE

## dokumentace

### Robotické rameno pohybující se pomocí gest



**Autor:** Sofja Klopová  
**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
se zaměřením na počítačové sítě a programování  
**Třída:** IT4  
**Školní rok:** 2023/24



## **Poděkování**

Chci poděkovat Pánu učiteli Godovskému za poskytnutí veškerého hardweru a také konzultací během celé práci.

## **Prohlášení**

Prohlašuji, že jsem závěrečnou práci vypracovala samostatně a uvedla veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 1. 1. 2024

.....  
Podpis autora



## Abstrakt

Výsledkem projektu je funkční pohyblivé rameno, které zvládá chytnout kuličku z konce stavebnice a posune ji na začátek a také za dobře odvedenou práci se uklonit. Ovládáno je gesty které jsou snímány kamerou notebooku. Hlavním aspektem projektu je kontrola jednotlivých serv ramene pomocí gest. Kód který umožní zobrazovat gesta je psaný v Pythonu, a holá pohyblivost ramene je obstarána Arduinem které využívá C++.

## Klíčová slova

pohyblivé rameno, gesta, Python, C++ ...

## Abstract

Write your abstract here! Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## Keywords

Template,  $\LaTeX$ , High school professional activity, ...

# Obsah

<b>Úvod</b>	<b>3</b>
<b>1 Princip fungování projektu</b>	<b>5</b>
<b>2 Hardware</b>	<b>7</b>
2.1 Netištěné součástky . . . . .	7
<b>3 Software</b>	<b>11</b>
3.1 Získání gest . . . . .	11
3.2 Ověření funkčnostní ramene . . . . .	13
3.3 Práce s Arduinem . . . . .	15
<b>Závěr</b>	<b>19</b>
<b>Literatura</b>	<b>21</b>



# Úvod

Tahle dokumentace vysvětluje celkovou funčnost mého projektu. Již hned jakmile jsme byli seznámení s tím, že čtvrtý ročník musíme zakončit ročníkovým projektem. Byla jsem rozhodlá pro Hardware. Po dlouhém a náročném pátrání a s dopomocí pana Godovského, bylo zvoleno robotické rameno. Bohužel samotné ovládání mě zavedlo primárně do Softwaru.

Rameno je ovládáno předvolenými gesty snímanými z kamery notebooku. Byl také nápad kompletně ovládat rameno způsobem, že rameno kopíruje pohyb ruky. Ale k tomu bylo potřeba dokoupit Pohyblivý senzor Kinect pro Xbox. To je ale celkem finančně náročné, tudíž od toho jsem odstoupila.

Cílem projektu bylo zprovoznit pohyb ramene celkem jednoduchým ale také efektivním způsobem. Díky tomu jsem si skvěle procvičila Python a C++. Motivoval mě ten pocit že to rameno rozpohybují.

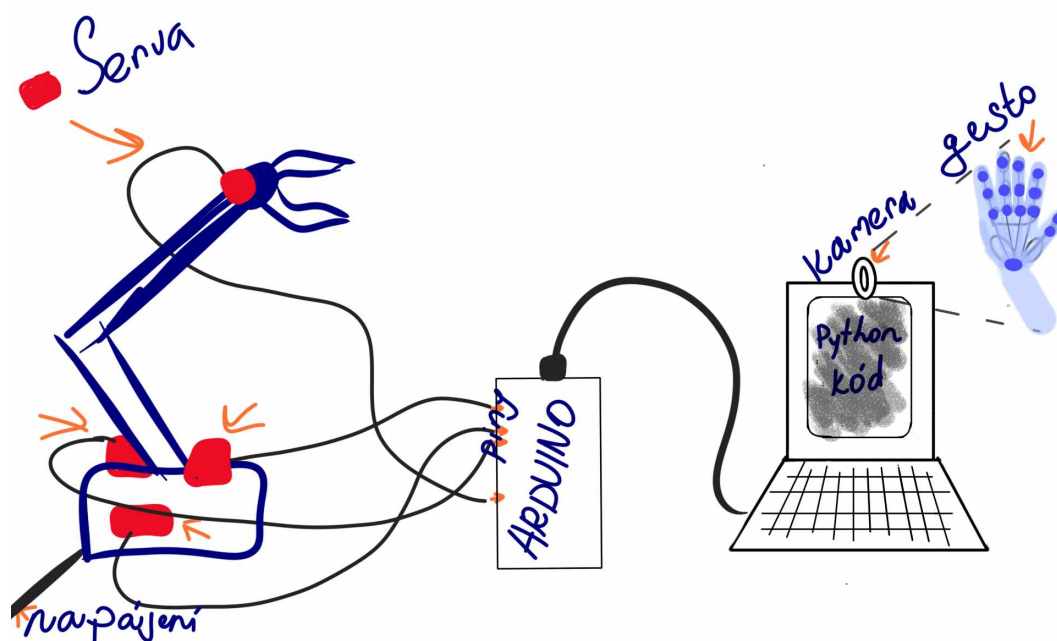
Hnedka v první kapitole naleznete všechny potřebné součástky k sestavení samotného ramene. To jsem ale nesestavovala já tudíž se budu věnovat spíš softwaru. Je to doprovázené krátkou teorií o využitých technologiích. Nakonec také samozřejmě nebudou chybět odkazy na všechny využité zdroje.





# 1 PRINCIP FUNGOVÁNÍ PROJEKTU

Tady se nachází strukturovaný náčrt, jak funguje celý projekt. A to tak že kamera snímá určité gesto. Kód s využitím knihoven obraz převádí na potřebné hodnoty které rozpozná rameno. Potom se pošlou do Arduino a ono ve své řadě je dá na určité piny aby serva rozpoznala co mají dělat.



Obrázek 1.1: princip fungování projektu [?].



## 2 HARDWARE

### 2.1 NETIŠTĚNÉ SOUČÁSTKY

Jelikož jsem to rameno nesestavovala tak nemám možnost popisovat postup, ale tady je výčet potřebných součástí

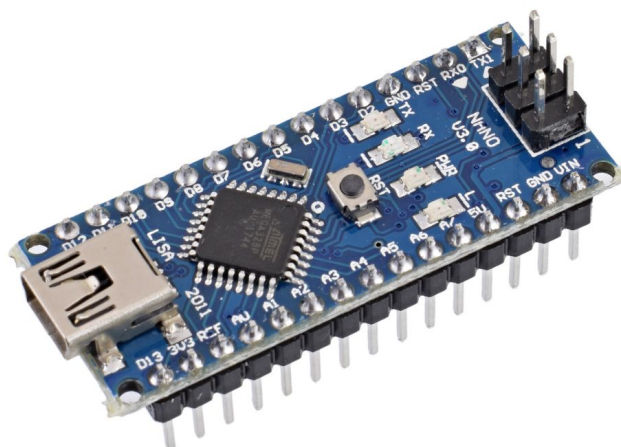
- 3 - 955 nebo 946 servo
- 1 - SG90 SERVO
- 1 - Samojistná matice M6
- 1 - Šroub M6x25
- 2 - Samojistné matice M3
- 2 - Šrouby M3 x 20
- 1 - Šroub M3 x 10 se šestihrannou hlavou
- 9 - Samojistné matice M4
- 1 - Šroub M4 x 40
- 1 - Šroub M4 x 30
- 5 - Šroub M4 x 20
- 1 - Závitová tyč M4 x 60mm
- 1 - Závitová tyč M4 x 32 mm
- 25 - Kulové koule o průměru 6 mm
- Ložisko 1 - 606zz
- Některé podložky M4

Návod na samotné sestavení ramene naleznete na tomhle odkazu

## 2.1.1 Arduino nano 328

### popis

Arduino napomáha rozhýbat samotný hardware. Napsaný kód, který popíšu v další kapitole nahrajeme pomocí USB kabelu do něj a díky správnému zapojení ramene vše skvěle funguje.



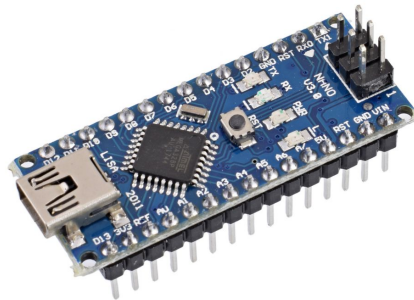
Obrázek 2.1: Arduino nano 328 [?].

## 2.1.2 Servo

### popis

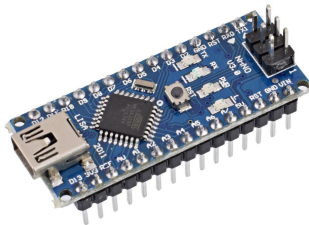
Tohle rameno obsahuje tři velké Serva a jedno malé. Každý z nich má také tři výstupy, a to: kabel určený na nějaký port, mínus a plus.

To nejmenší se nachází v kleštičkách



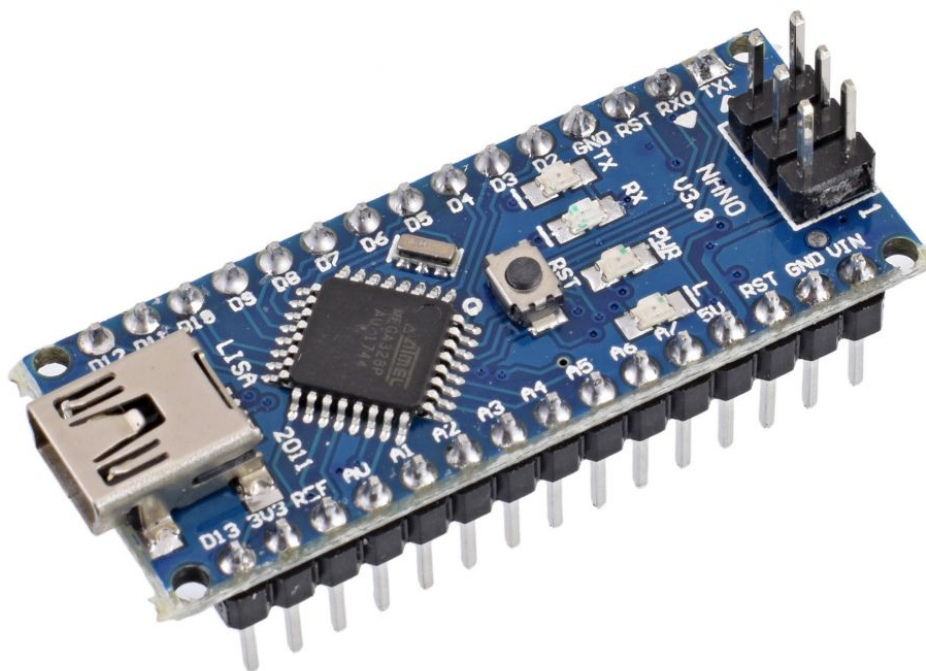
Obrázek 2.2: servo nacházející se v kleštích ramene[?].

Zbylé tři se zabývají rotací a pohyby nahoru-dolů, dopředu-dozadu



Obrázek 2.3: Serva v těle ramene[?].

### 2.1.3 Obvod



Obrázek 2.4: zapojení ramene [?].

## 3 SOFTWARE

### 3.1 ZÍSKÁNÍ GEST

Začneme importem knihoven, a to **OpenCV math** a **mediapipe 0.10.8**

#### Knihovna OpenCV

OpenCV je multiplatformní svobodná knihovna pro práci s obrazem.

Samotná knihovna je z velké části napsána v C++, ale poskytuje Python wrapper, díky kterému ji můžeme používat i v Pythonu. OpenCV se používá pro zpracování obrazu z kamer, rozpoznání psaného textu nebo obličejů.. Tuto knihovnu jsem využila na celkové zpracování obrazu z kamery mého notebooku

```
1 cap = cv2.VideoCapture(0)
2 ...
3 image = cap.read()
4 imageRGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
5 results = hands.process(imageRGB)
```

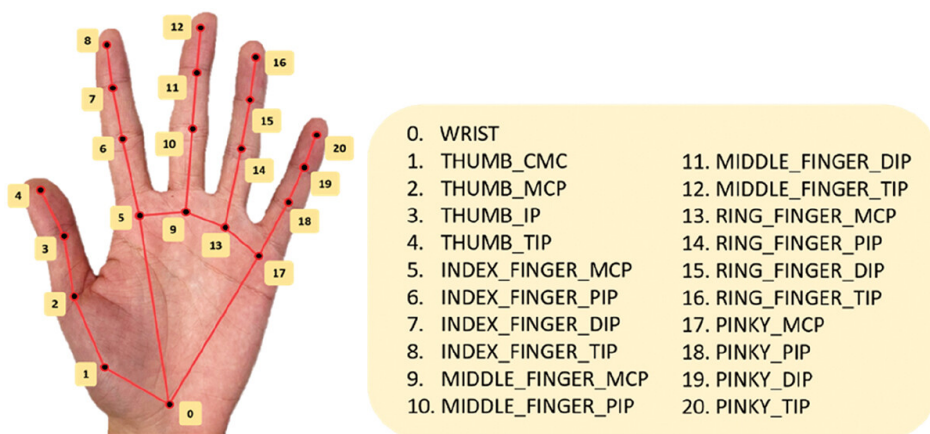
Kód 3.1: obraz, a jeho převedení do správného formátu

Převede snímáný obraz do formátu RGB, který potřebujem pro vykreslení pixelů ve správném pořadí.

#### knihovna mediapipe

Tahle knihovna se zaměřuje na detekci lidských částí těla. Já využila pro svůj projekt pouze detekci dlaní. Pro detekci potřebuje mít orientační body ruky. Model ručního orientačního bodu dokáže předpovědět 21 přesných souřadnic umístění každého orientačního bodu na ruce.

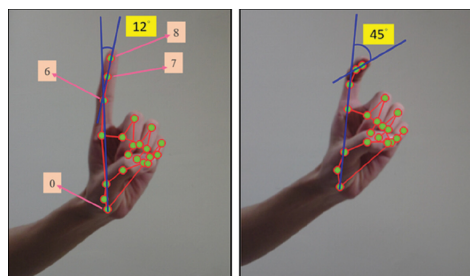




Obrázek 3.1: orientační body dlaně [?].

## Knihovna math

Ve spolupráci se výše zmíněnou knihovnou mi poskytla souřadnice jednotlivých kloubů, pro přesné zachycení ruky. Další výpočty provedla také při zachycení vzdálenosti prstů potřebné k rozhýbání kleštíček



Obrázek 3.2: použití úhlu v gestech [?].



Obrázek 3.3: použití úhlu v gestech [?].

## 3.2 OVĚŘENÍ FUNKČNOSTNÍ RAMENE

Než jsem mohla začít se sprovedením ramene dle myšlenky, jsem musela ověřit její funkčnost a taky zjistit jak se vlastně jednotlivá část hýbe. Proto jsem použila jednoduchý kód. Připojením pokaždé jednoho drátku do desky s arduinem jsem se postupně dozvěděla zaprvé jaký drátek zodpovídá za co a také i jak se hybou. Byl tam jeden zadrhel a to to že serva stávkovaly, když jsem posílala ty krajní hodnoty(0,180) po úpravě intervalu ve funkci, se všechno srovnalo.

```
1 #include <Servo.h>
2
3 Servo myservo;
4
5 int pos = 0;
6 void setup() {
7     myservo.attach(9);
8 }
9
10 void loop() {
11     for (pos = 2; pos <= 160; pos += 1) {
12         myservo.write(pos);
13         delay(15);
14     }
15     for (pos = 160; pos >= 2; pos -= 1) {
16         myservo.write(pos);
17         delay(15);
18     }
19 }
```

Kód 3.2: Ukázka kódu k ověření funkčnosti

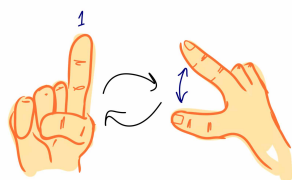
### 3.3 PRÁCE S ARDUINEM

První věc, kterou musíme vzít v úvahu, je, že nezbytně potřebujeme mít na svém počítači program Python a také PySerial. Většinu kódu v Pythonu se dá sehnat na internetu a odkazy na ty stránky naleznete níže v použité literatuře. Takže se jdeme přesunout k popisu. Pro kód v C++ je na začátek potřeba inicializovat veškeré potřebné hodnoty. např:

```
1 const int min= 2; // minimální hodnota pro pozici ramene
2 const int max = 160; // maximální hodnota pro pozici ramene
3
4 bool end;
5 int data = 0;
6 int serialData =0;
7 bool changeMode;
8 int mode;
9 const int servoPose = 6; //kolik stavu má rameno
10 const int servoCount = 4; // počet serv
11
12 Servo servo[4]; // pole pro serva
13 const byte servoPins [] = {2,3,4,5}; //piny pro serva
```

Kód 3.3: Inicializace potřebných hodnot

- Jak jsem již zmiňovala výše zádrhel byl, když se do ramene posílali krajní hodnoty(0 a 180). Kvůli tomu jsem je nastavila na 2 a 160. a to je uloženo v proměnných min a max.
- changeMode je pro přehazování mezi snímáním počtu prstu nebo vzdálenosti mezi prsty



Obrázek 3.4: dvě módy pro kontrolu ramene [?].

- V proměnné servoPos je uložen počet všech stavu ve kterých se moje rameno může nacházet.

- Jelikož obsahuje 4 serva, tak jsem pro ně vytvořila pole. A každé potřebuje svůj pin k tomu je proměnná servoPins.

```

1 if(j == (servoMoves[data][0] -1)) {
2     if(data == 0 || data == 1) {
3         changeMode = true;
4         mode = data;
5     }

```

Kód 3.4: podmínka pro přepínání módu

- Tahle jednoduchá podmínka je vytvořena pro kontrolu, kde se zroná nachází rameno. A když se nachází nahoře nebo dole, tak se přepne mód pro měření vzdálenosti. Ať se může chytnout nebo pustit kulička.

```

1 \begin{lstlisting}[style=Python, caption={dvourozměrné pole na potřebné hodnoty}]
2     int servoValues [13] [3] = {
3         {20,50,20}, //puvodni stav
4         {170,80,45}, //dole sběr kuličky cast1
5         {123,113,45}, //dolu sběr kuličky cast2
6         ...
7         {123,130,133}, //uklona cast2
8     };
9

```

Kód 3.5: dvourozměrné pole na potřebné hodnoty

- Dále máme dvourozměrné pole pro hodnoty pro serva. jsou vždy v jednom řádku jen tři, protože pro ovládání serva v kleštích jsou potřeba jiné hodnoty.

```

1 \begin{lstlisting}[style=Python, caption={podmínka pro přepínání módu}]
2     int servoMoves [servoPose] [3] = {
3         {2,1,500}, //dolu ke kulice 1
4         {2,3,500}, //nahoru ke kulice 2
5         {5,5,700}, //ne 3
6         ...
7     };
8

```

Kód 3.6: další pole pro

- Další pole se skládá z několika různých hodnot. A to na prvním místě počet opakování pohybů, na druhém místě se nachází pořadí hodnot serv z předchozího pole. Poslední hodnota je pro delay.

```

1  \begin{lstlisting}[style=Python, caption={funkce na rozpohybování ramene}]
2
3  void move(int data) {
4      int cycle = 0;
5      for(int j = 0 ; j < servoMoves[data][0];j++) {
6          for(int i = 1; i < servoCount;i++) servo[i].write(servoValues[
7              (servoMoves[data][1])+cycle][i -1]);
8
9          cycle++;
10         if(cycle > 1) cycle=0;
11         if(j == (servoMoves[data][0] -1)) {
12             if(data == 0 || data == 1) {
13                 changeMode = true;
14                 mode = data;
15             }
16             end = true;
17             Serial.begin(9600);
18         }
19         delay(servoMoves[data][2]);
20     }
21 }
22

```

Kód 3.7: funkce na rozpohybování ramene

- Jedna z nejpodstatnějších funkcí. Je pro rozpohybování ramene. Jakmile se načte jedna hodnota, funkce se posune v cyklu o jedno políčko dál v servoValues, a tak než dojde na konec. Potom se zkontroluje zda data jsou 0 nebo 1 pro pochopení že se rameno nachází v pozici u kuličky. Tehdy se změní changeMode a přepne se na měření vzdálenosti mezi prsty.

## ZÁVĚR

Cílem samotné práce bylo rozpohybovat rameno a udělat si jasno jak fungují knihovny. Ve finální verzi rameno je schopno uchytit kuličku a přenést ji z konce na začátek. Všechno to je vyřešeno pomocí Pythonu a C++. Využití tohoto projektu může být různé. Například v továrnách, ve skladech, v nanochirurgii nebo jako vzdělávací hračky. Začátečný cíl se mi podařilo splnit a rameno funguje tak, jak jsem si původně představovala při zahájení vývoje. I přes můj výkon kód přece jen není napsán nejoptimálněji a věřím, že by mohl být mnohem lépe zkonstruován, což do budoucna bych mohla opravit. Také později bych se chtěla víc zaměřit na knihovnu OpenCV a prozkoumat další její funkce.



# LITERATURA

- [1] Základní informace o gestech [Online]. 2023 CEMI MBA Studies s.r.o. [cit. 2020-08-24]. Dostupné z: <https://www.cemi.cz/blog/neverbalni-komunikace-co-prozradi-gesta>
- [2] Typy gest v běžném životě [online]. 1998 [cit. 2020-08-24]. Dostupné z: <https://orangeacademy.cz/clanky/rec-tela/>
- [3] Řeč tela a znakový jazyk, 2023, Slune - svět jazyků, s.r.o. Dostupné z: <https://www.slune.cz/aktualita/znakovy-jazyk/>
- [4] knihovna math v pythonu Copyright © 2023 itnetwork.cz. <https://www.itnetwork.cz/python/zaklady/python-tutorial-knihovny-math-a-random>
- [5] knihovna opencv v pythonu Copyright © 2023 itnetwork.cz. <https://www.python-programator.cz/clanky/python-pro-zpracovani-obrazu-opencv>
- [6] komunikace s arduinem © 2023 Arduino <https://projecthub.arduino.cc/ansh2919/serial-communication-between-python-and-arduino-663756>
- [7] komunikace s arduinem 2 Copyright © 2023 Circuit Digest. All rights reserved. <https://circuitdigest.com/microcontroller-projects/arduino-python-tutorial>
- [8] zkouška funkčnosti <https://docs.arduino.cc/learn/electronics/servo-motors>
- [9] zkouška funkčnosti <https://problemsolvingwithpython.com/11-Python-and-External-Hardware/11.03-Controlling-an-LED/>
- [10] *TeX - LaTeX Stack Exchange* [online]. Stack Exchange, 2020 [cit. 2020-09-01]. Dostupné z: <https://tex.stackexchange.com>
- [11] *Střední škola průmyslová a umělecká Opava* [online]. [cit. 2023-11-11]. Dostupné z: <https://www.sspu-opava.cz>
- [12] *Citace PRO* [online]. Citace.com, 2020 [cit. 2020-08-31]. Dostupné z: <https://www.citacepro.com>



- [13] BORN, Max a Emil WOLF. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. 7th (expanded) edition. Reprinted with corrections 2002. 15th printing 2019. Cambridge: Cambridge University Press, 2019. ISBN 978-0-521-64222-4.

## Seznam obrázků

1.1	princip fungování projektu [?]. . . . .	5
2.1	Arduino nano 328 [?]. . . . .	8
2.2	servo nacházející se v kleštích ramene[?]. . . . .	9
2.3	Serva v těle ramene[?]. . . . .	9
2.4	zapojení ramene [?]. . . . .	10
3.1	orientační body dlaně [?]. . . . .	12
3.2	použití úhlu v gestech [?]. . . . .	12
3.3	použití úhlu v gestech [?]. . . . .	13
3.4	dvá módy pro kontrolu ramene [?]. . . . .	15