|  |  |  |
| --- | --- | --- |
|  | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Arduino servo rameno** | | |
| Richard West | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2021/2022 | |

#### Poděkování

* *Chtěl bych poděkovat všem učitelům, kteří mi pomáhali při vývoji projektu. A také rád poděkoval i mým spolužákům za případné poznámky pro vylepšení projektu a následnou pomoc.*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

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

V Opavě 31. 12. 2021

*podpis autora práce*

**ANOTACE**

Projekt se zabývá tvorbou robotického ramena na platformě arduino s použitím servo motorů. Na projekt byl využit mikrokontroler Arduino MEGA2560. Konstrukce ramena je tisknutá na 3D tiskárně z materiálu PLA a do pohybu jí uvádí 3 MG995 servo motory a uchopovacím mechanismem hýbe SG90 servo motor. Kód je napsán v jazyce C++ a je přizpůsoben pro jednoduché upravení pro změnu práce, kterou rameno vykonává.

**Klíčová slova:** Arduino, Servo motory, Robotické rameno, 3d tisk, Automatizace

OBSAH

[Úvod 5](#_Toc92645202)

[1 Využité technologie 6](#_Toc92645203)

[1.1 Hardware 6](#_Toc92645204)

[1.1.1 Arduino Mega 6](#_Toc92645205)

[1.1.2 MG995 Servo motor 7](#_Toc92645206)

[1.2 Software 8](#_Toc92645207)

[1.2.1 Wiring 8](#_Toc92645208)

[1.2.2 Arduino IDE 8](#_Toc92645209)

[1.2.3 TinkerCAD 8](#_Toc92645210)

[2 Sestavení a testování ramena 9](#_Toc92645211)

[2.1 Sestavení ramena 9](#_Toc92645212)

[2.2 Ovládání ramena pomocí arduina 10](#_Toc92645213)

[2.3 Problémy se servy 11](#_Toc92645214)

[2.4 Shromáždění potřebných souřadnic 11](#_Toc92645215)

[2.5 Implementace systémového přerušení 11](#_Toc92645216)

[3 Způsoby řešení a použité postupy 12](#_Toc92645217)

[3.1 Použité součástky 12](#_Toc92645218)

[3.2 Schéma zapojení 13](#_Toc92645219)

[3.3 Kód 14](#_Toc92645220)

[4 Výsledky řešení, výstupy, uživatelský manuál 15](#_Toc92645221)

[4.1 Finální provedení 15](#_Toc92645222)

[Závěr 16](#_Toc92645223)

[SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJů 17](#_Toc92645224)

Úvod

K výběru mého závěrečného projektu mi pomohl můj zájem o elektrotechniku a fascinace automatizací a robotikou.

Většina průmyslových firem již robotická ramena používá. Jsou ideálním způsobem, jak snížit náklady na výrobu a zvýšit efektivitu produkce. Ramena umožňují téměř nulovou chybovost provozu a snižují tím počet potřebných pracovníků ve firmě.

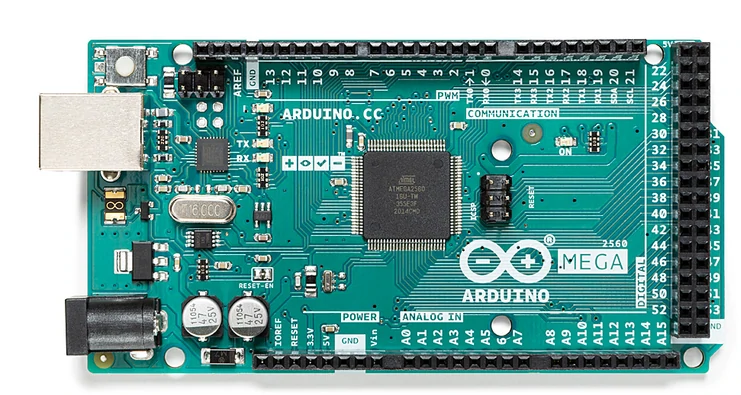
V projektu se věnuji převážně experimentaci a fungování servo motorů. Se servo motory jsem neměl žádnou předchozí zkušenost. V jednotlivých kapitolách postupně popisuji svůj postup projektem a případné problémy na které jsem narazil. Na projekt jsem se rozhodl využít mikrokontroler Arduino MEGA2560, protože za jeho cenu nabízí dostatek vstupů a výstupů pro budoucí rozšiřování funkcí ramena. Konstrukce ramena je tisknutá na 3D tiskárně z materiálu PLA, 3d tisk všech potřebných částí trval téměř 30 hodin. Do pohybu rameno uvádí 3 MG995 servo motory s kovovými převody a uchopovacím mechanismem hýbe SG90 servo motor. Kód je napsán v jazyce C++ a je přizpůsoben pro jednoduché upravení pro změnu práce, kterou rameno vykonává.

# Využité technologie

## Hardware

### Arduino Mega

Celý tento projekt běží na Arduino Mega.Arduino MEGA 2560 je deska založená na známém mikrokontroleru ATmega2560. Deska obsahuje 54 digitálních vstupů/výstupů (15 PWM výstupů, 4 hardwarové sériové porty, externí přerušení), 16 analogových vstupů, 16MHz krystalový oscilátor, USB konektor, napájecí konektor, ICSP konektor a resetovací tlačítko.



*Obrázek 1: Arduino Mega2560*

### MG995 Servo motor

Pro pohyb celým ramenem jsem použil MG995 serva pro jejich dostatečnou sílu a odolné kovové převody uvnitř serva zaručí výdrž při používání. Ze serva vedou tři vodiče, zem(černý vodič), 4.8-7.2V(červený vodič) a oranžový vodič je na posílání dat z Arduina.



*Obrázek 2: MG995 Servo*

## Software

### Wiring

Celý program je napsaný v jazyce Wiring, který je vlastně obohacená verze jazyka C++ o knihovny umožňující práci s vývojovými deskami Arduino.

### Arduino IDE

Pro psaní kódu jsem použil prostředí Arduino IDE protože můj projekt nevyžadoval velké množství knihoven a nijak komplexní programování, v takovém případě bych zvolil VSC v kombinaci s PlatformIO

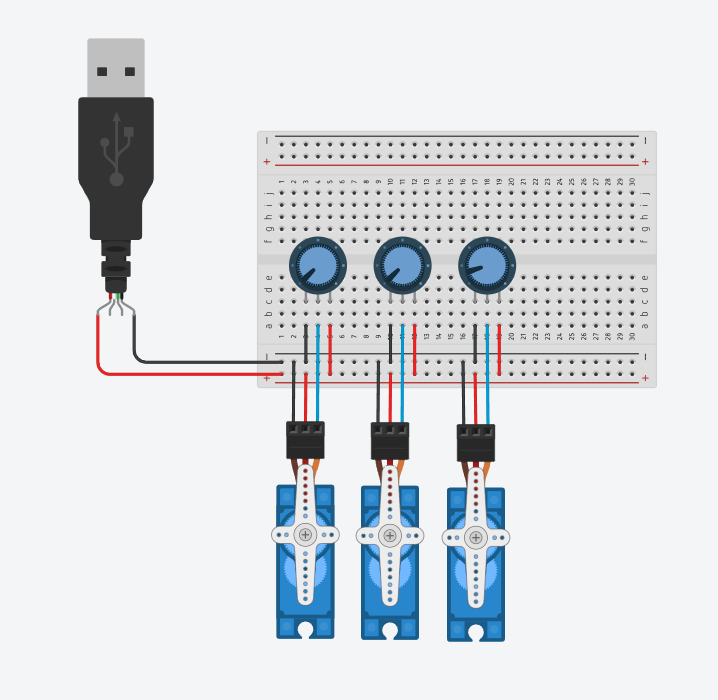
### TinkerCAD

Pro vytvoření nákresů a schémat jsem použil tuto webovou aplikaci, která je zdarma a dostupná všem. Umožňuje i simulaci kódu a vizualizaci chování obvodu.

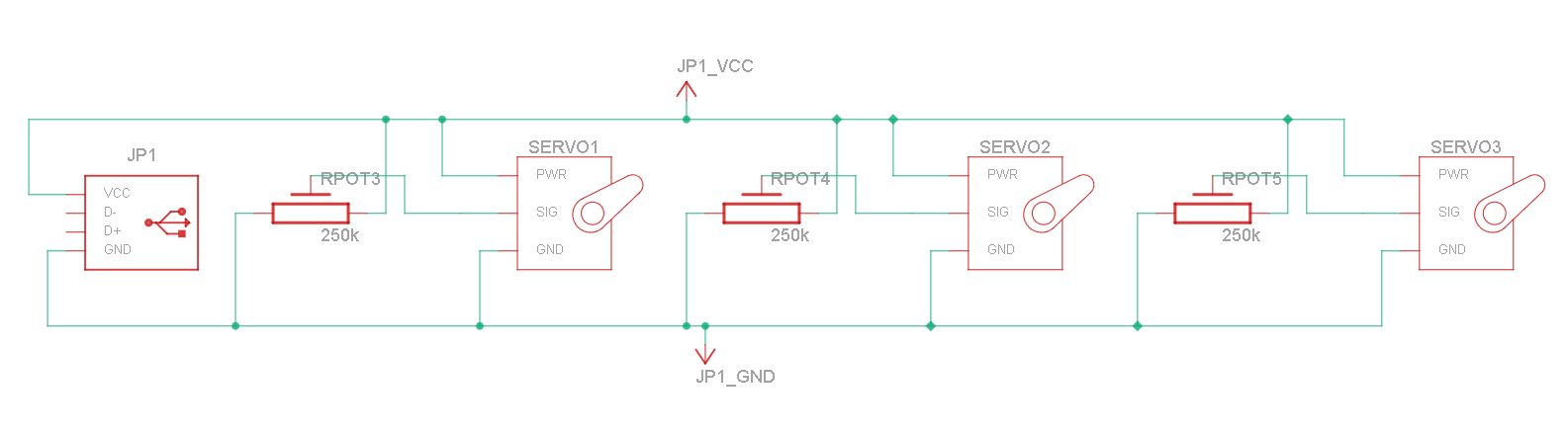
# Sestavení a testování ramena

## Sestavení ramena

Rameno jsem dostal od pana učitele Godovského sestavené, avšak jsem si ho rozebral abych lépe porozuměl, jak součástky spolu fungují a pasují do sebe. Poté co jsem rameno složil, jsem ho zapojil pomocí potenciometrů na 5v zdroj.



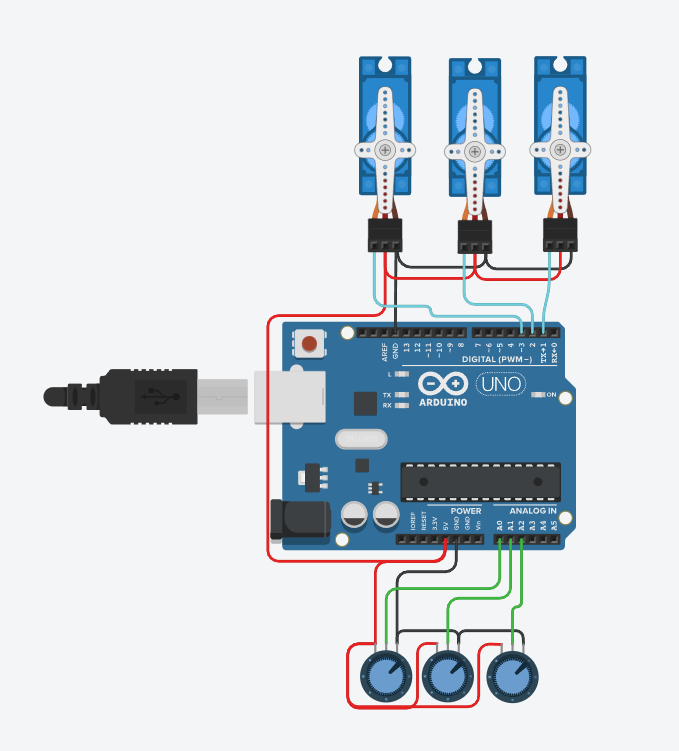
*Obrázek 3: Zapojení servo motorů s potenciometry*



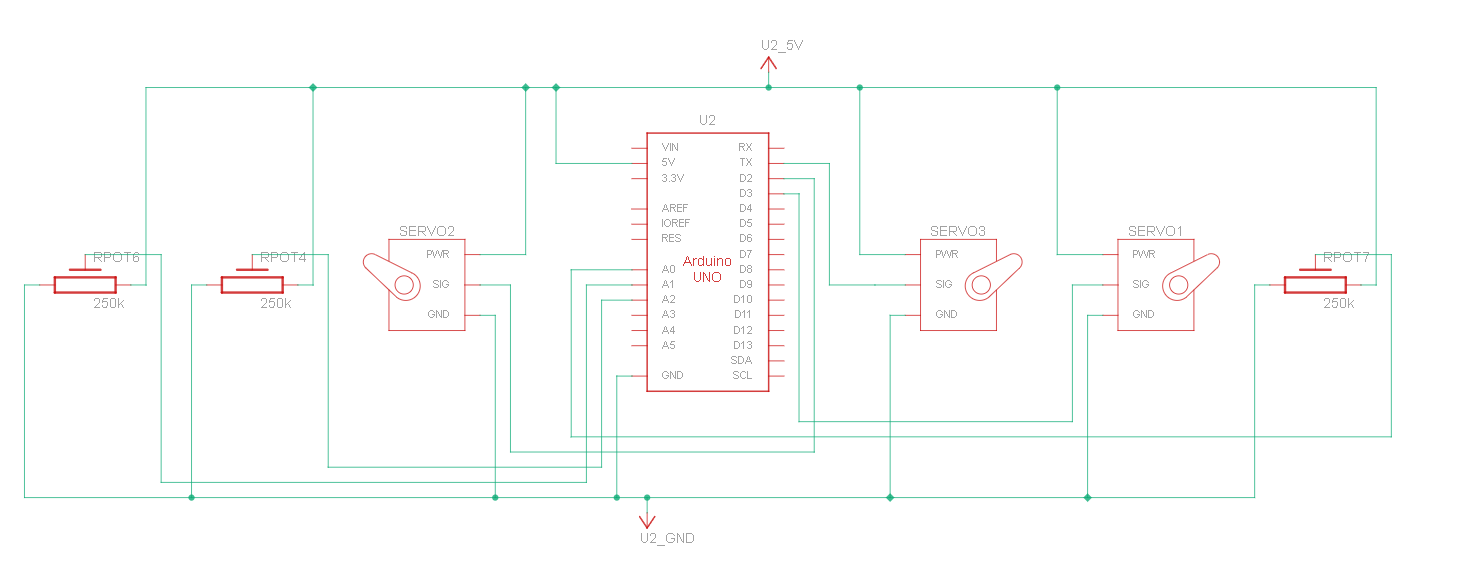
*Obrázek 4: Schéma zapojení servo motorů s potenciometry*

## Ovládání ramena pomocí arduina

Po prvotním testování s potenciometry jsem zapojil serva do arduina a udělal jednoduchý kód který četl hodnoty z potenciometru a posílal je do servo motorů abych si odzkoušel, jak funguje knihovna servo.h. V této fázi projektu nastal problém, servo v základně se nedokázalo hnout a někdy se arduino restartovalo, když jsem hýbal potenciometrem příliš rychle.



*Obrázek 5: Zapojení servo motorů a potenciometrů do arduina*



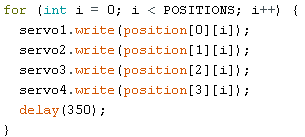
*Obrázek 6: Zapojení servo motorů a potenciometrů do arduina*

## Problémy se servy

Nejprve jsem si myslel, že problémy byly způsobené nedostatečným tokem proudu z USB portu stolního počítače, avšak i při připojení 12v zdroje do napájecího portu arduina problém přetrvával, už se arduino nerestartovalo ale po ~30 minutách práce se arduino vyplo a serva se ani nehly. Integrovaný 5V regulátor se přehřál a aby zabránil poškození se vypl. Tudíž řešení bylo jasné, obejít tento regulátor 5V AC adaptérem.

## Shromáždění potřebných souřadnic

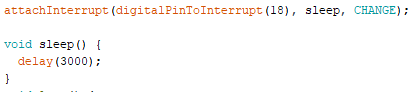
Následně jsem musel zvolit efektivní způsob pro uložení pozicových hodnot pro serva, jako nejpřehlednější způsob mi přišla dvourozměrná matice o 4 řádcích. Je to čitelné i z pohledu člověka, a tudíž i lehce upravitelné. Poté stačilo vytvořit cyklus for obsahující funkce pro posílání hodnot servům.



*Obrázek 7: Cyklus posílání hodnot pro serva*

## Implementace systémového přerušení

Přidání systémového přerušení je důležité pro funkční ovládání ramena nezávisle kde v cyklu rameno zrovna je. Systémové přerušení jsem přidal pomocí funkce   
attachInterrupt() z knihovny Arduino.h. Potřebná část kódu pro funkční přerušení vypadá následovně:



*Obrázek 8: Systémové přerušení*

# Způsoby řešení a použité postupy

## Použité součástky

1x Arduino MEGA2560

3x MG995 Servo

1x S90 Servo

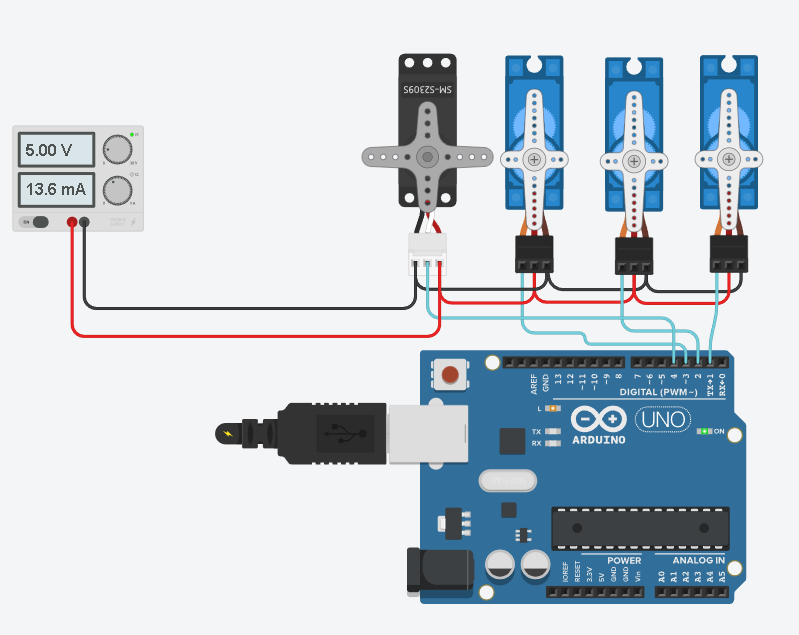
1x 5V 2A AC/DC adaptér

1x dětská hračka, skluzavka pro skleněné kuličky

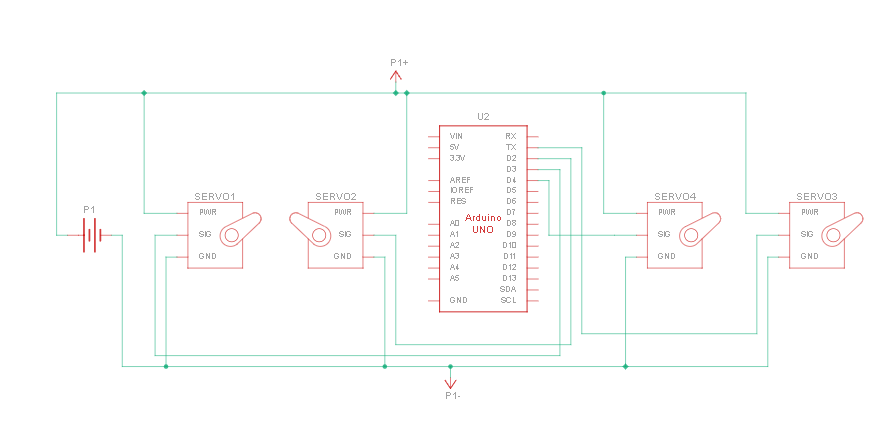
3x tisknuté uchycení pro skluzavku

1x tisknutý vývod pro skleněnou kuličku na konci dráhy

## Schéma zapojení



*Obrázek 9: Obrázek finálního zapojení*



*Obrázek 10: Schéma finálního zapojení*

## Kód

#include <Servo.h>

#include <Arduino.h>

#define SERVOS 4 //Počet servo motorů.

#define POSITIONS 8 //Počet pozic, kterýma rameno projde.

Servo servo1; //Definování objektů typu servo.

Servo servo2;

Servo servo3;

Servo servo4;

int pins[SERVOS] = {2, 3, 4, 5}; //Piny na které jsou serva připojeny.

int position[SERVOS][POSITIONS] = {{ 45, 60, 60, 90, 90, 60, 60, 45}, {116, 85, 85, 85, 85, 85, 85, 116},

{123, 150, 175, 175, 175, 175, 140, 123}, {6, 6, 6, 6, 30, 30, 30, 30}};

void setup() {

attachInterrupt(digitalPinToInterrupt(18), sleep, CHANGE); //systémové přerušení, které bez uzemnění pinu 18 tlačítkem pouze čeká

servo1.attach(pins[0]); //Přiřazení objektů k digitálním výstupním pinům.

servo2.attach(pins[1]);

servo3.attach(pins[2]);

servo4.attach(pins[3]);

}

void sleep() { //pozastavení práce ramena

delay(3000);

}

void loop() {

//Cyklus for, který při každém průběhu nastaví jednotlivé hodnoty serv na hodnoty pro danou polohu ramena. Cyklus proběhne tolikrát, kolik je nastaveno v proměnné POSITIONS.

for (int i = 0; i < POSITIONS; i++) {

servo1.write(position[0][i]);

delay(20);

servo2.write(position[1][i]);

delay(20);

servo3.write(position[2][i]);

delay(20);

servo4.write(position[3][i]);

delay(100);

}

delay(1200);

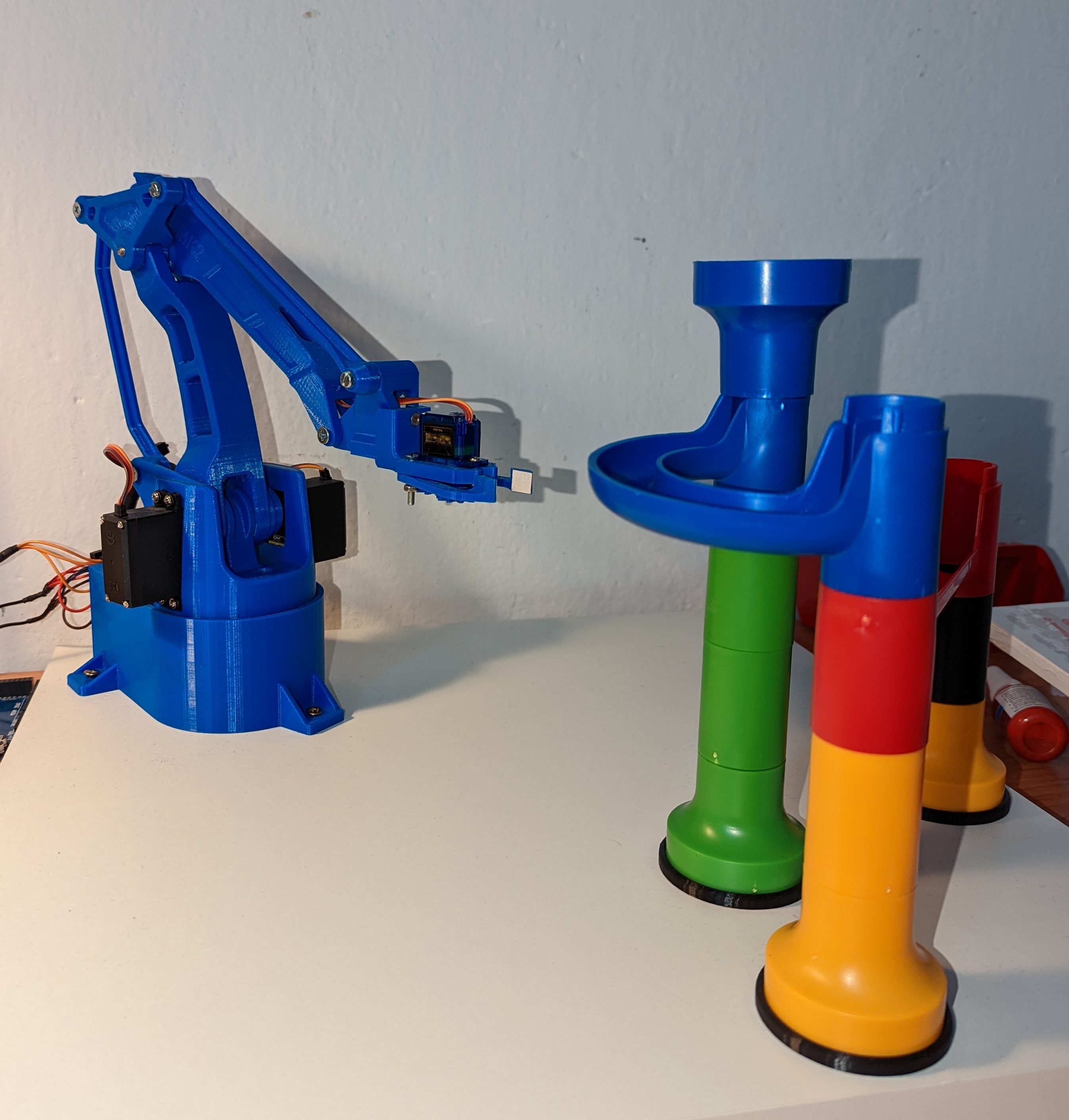
}

# Výsledky řešení, výstupy, uživatelský manuál

## Finální provedení

Výsledné provedení vychází z mé původní vize o projektu, avšak kdybych začínal projekt znova tak bych spíše zvolil krokové motory a patřičné drivery, abych mohl ovládat akceleraci a rychlost pohybu a mohl mít i zpětnou vazbu o motorech jako je síla pohybu a případná detekce kolize.

*Obrázek 11: Finální provedení ramena*



Kvůli ne úplně dobré preciznosti pohybu se občas kulička špatně odrazí a z dráhy uteče.

# Závěr

Cílem bylo vytvoření robotického ramena s možností lehkého předělání na více typů úloh. Zadaný cíl byl splněn, protože v momentálním stavu je to pouze otázka nahrazení stávajících hodnot v matici a změnění časování pro zajištění plynulého pohybu.

Při realizaci projektu ve větším měřítku by však bylo jistě potřeba udělat několik úprav a mimo ty co jsem již zmínil výše přidat alespoň display a uživatelské rozhraní.

Odkaz na GitHub: https://github.com/Brr0/roborameno

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJů

[] *EEZYbotARM MK2* [online]. [cit. 2021-12-10]. Dostupné z: <https://www.thingiverse.com/thing:1454048>

[2] *Arduino Servo.h library* [online]. [cit. 2021-12-10]. Dostupné z: https://www.arduino.cc/reference/en/libraries/servo/

[3] *EEZYbotARM MK2 3D Printed Robot Build - Chris's Basement* [online]. [cit. 2021-12-10]. Dostupné z: https://www.youtube.com/watch?v=R2MI-tpXyS4&ab\_channel=ChrisRiley

[4] *The EEZYbotARM 3D printed easy bot arm build* [youtube.com]. [cit. 2021-12-12]. Dostupné z: https://www.youtube.com/watch?v=zBBRGoHiQgc&ab\_channel=ChrisRiley

[5] *EEZYbotARM Mk2 - 3D Printed Robot* [online]. [cit. 2021-12-12]. Dostupné z: <https://www.instructables.com/EEZYbotARM-Mk2-3D-Printed-Robot/>

[6] *Servo.h knob example* [online]. [cit. 2021-12-12]. Dostupné z: https://www.arduino.cc/en/tutorial/knob?\_gl=1\*1bih3jl\*\_ga\*MTQwMDk2MTU0MC4xNjMzNjc3Mzgy\*\_ga\_NEXN8H46L5\*MTY0MDM2MTg2NC4xMC4wLjE2NDAzNjE4NjQuMA..

[7] *ARDUINO MEGA 2560 R3* [online]. [cit. 2021-12-14]. Dostupné z: https://www.hwkitchen.cz/arduino-mega-2560-r3/

[8] *Servo MG995* [online]. [cit. 2021-12-14]. Dostupné z: https://www.laskakit.cz/servo-mg995-s-kovovymi-prevody-180---extra-silne/#ratingTab