|  |  |  |
| --- | --- | --- |
|  | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Robotické rameno s využitím Arduina** | | |
| Michal Semenišín | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2020/2021 | |

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. 2020

*podpis autora práce*

**ANOTACE**

Projekt se zabývá tvorbou robotického ramene s využitím řízení pomocí vývojové desky Arduino. Řešení se tedy skládá jak z hardwarové, tak i softwarové části. Zařízení je jako celek z důvodů stability a možnosti přenosu umístěno na základové desce z překližky. Základ hardwarové části řízení tvoří vývojová deska Arduino Uno Rev3. Programová část ramene je řešena v jazyce Arduino, což je kombinace jazyků C a C++. Samotný pohyb robotického ramene je potom zajištěn pomocí 6 servomotorů, které jsou v první části projektu ovládány joysticky. Zařízení se uvádí do chodu pomocí kolébkového vypínače umístěného na vlastním napájecím modulu. Tímto krokem se spouští program, rameno se nastavuje so výchozí polohy a je možné jeho ovládání pomocí joysticků. Druhá část projektu řeší softwarové ovládání ramene pomocí grafického uživatelské rozhraní vyvinutého v Pythonu.

OBSAH

[Úvod 5](#_Toc62425039)

[1 Seznam použitého hardware 6](#_Toc62425040)

[1.1 Seznam součástek 6](#_Toc62425041)

[1.2 Spojovací a pomocný materiál 6](#_Toc62425042)

[1.3 Použité nářadí 6](#_Toc62425043)

[2 sestavení ROBOTICKÉHO RAMENE 7](#_Toc62425044)

[3 Využité technologie 8](#_Toc62425045)

[3.1 Popis hardware 8](#_Toc62425046)

[3.1.1 Vývojová deska Arduino Uno Rev 3 8](#_Toc62425047)

[3.1.2 Servomotory MG996R 9](#_Toc62425048)

[3.1.3 Dual-axis XY joystick 9](#_Toc62425049)

[3.1.4 6 DOF hliníková robotická konstrukce ramene 10](#_Toc62425050)

[3.2 Software 11](#_Toc62425051)

[3.2.1 Jazyk Arduino 11](#_Toc62425052)

[3.2.2 Python 3.9.1 11](#_Toc62425053)

[3.2.3 Arduino IDE 1.8.13 11](#_Toc62425054)

[3.2.4 KiCAD 5.1.9 11](#_Toc62425055)

[4 Způsoby řešení a použité postupy 12](#_Toc62425056)

[4.1 Kabeláž 12](#_Toc62425057)

[4.2 Napájení 13](#_Toc62425058)

[4.3 Princip fungování servomotoru 13](#_Toc62425059)

[4.4 Princip fungování programu 14](#_Toc62425060)

[4.4.1 Inicializace základní polohy 15](#_Toc62425061)

[4.4.2 Pohyb 15](#_Toc62425062)

[4.4.3 Uživatelskérozhraní 16](#_Toc62425063)

[5 Výsledky řešení 17](#_Toc62425064)

[Závěr 18](#_Toc62425065)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 19](#_Toc62425066)

Úvod

V rámci závěrečné studijní práce jsem se rozhodl nahlédnout do oblasti robotiky, a to prostřednictvím bližšího seznámení se s robotickými rameny. Seznámil jsem se např. s pojmy jako kinematika, kinematická dvojice, stupeň volnosti atd.

Na základě nastudované teorie problematiky robotických ramen jsem se rozhodl pro konstrukci cenově dostupného šestiosého robotického ramene, které by mohlo následně posloužit edukačním účelům.

šestiosá robotická ramena umožňují vyšší flexibilitu a mohou nalézt častější využití než roboti s méně osami.

Pořídil jsem si tedy stavebnici šestiosého robotického ramene včetně servomotorů a začal s přípravami. Má představa spočívala ve vytvoření dostatečně velkého pracovního prostoru ramene – základové desky, ke které bude rameno kotveno. Tato základová deska by posloužila také k montáži desky mikropočítače, nepájivého kontaktního pole a samozřejmě ovládacích prvků ramene.

Po úvaze jsem pro řízení pohybu robotického ramene zvolil vývojovou desku Arduino UNO Rev3 založenou na mikrokontroleru ATmega328 od firmy Atmel. Tento mini počítač má dostatečný počet vstupů a výstupů a je cenově velmi výhodný. Dostal tedy přednost např. před Raspberry Pi

Ovládání pohybu ramene jsem chtěl původně řešit pomocí 6 kusů potenciometrů. Vzhledem k lepší možnosti uchycení k základové desce jsem nakonec zvolil Dual-axis XY joysticky KY-023. Tyto joysticky jsou osazeny 2 trimry (jedním pro osu X, druhým pro osu Y) a spínačem. Místo šesti ovládacích prvků budou použity pouze tři., což bude praktičtější a přinese to i úsporu místa na základové desce.

Program pro ovládání ramene bude tedy naprogramován v jazyce Arduino, což je kombinace jazyků C a C++. K ramenu jsem také připravil jednoduché uživatelské rozhraní v Pythonu.

V dalších částech této dokumentaci popíšu samotné sestavení ramene a princip jeho fungování. Také se zmíním o různých problémech, na které jsem v průběhu vývoje ramene narazil.

Závěrem zhodnotím výsledek provedené práce.

# Seznam použitého hardware

## Seznam součástek

* Vývojová deska Arduino UNO Rev 3,
* nepájivé kontaktní pole s 830 kontakty,
* dual-axis XY joystick - 3 kusy,
* napájecí adaptér AC 230 V, DC 5 V,
* DC svorka samice,
* servomotor MG996R 6 kusů,
* 6 DOF hliníková robotická konstrukce ramene,
* modelářská překližka 600x300x6,0 mm.

## Spojovací a pomocný materiál

* Propojovací DuPont kabely
* matice M3 16 kusů,
* šrouby M3x16mm 16 kusů, M3x10mm 4 kusy, šroub M3 pro CD mechaniku 4 kusy,
* podložka plochá 6,8/3,2mm 20 kusů,
* vějířová podložka 4 kusy,
* distanční sloupek kovový šestihran 4 kusy,
* plastový distanční sloupek 12 kusů,
* oboustranná lepicí páska,
* samolepící nožička 6 kusů,
* stahovací pásky,
* smršťovací bužírky.

## Použité nářadí

* Páječka,
* vrtačka + sada vrtáků,
* štípací a odizolovací kleště,
* odlamovací nůž, nůžky,
* sada pilníků,
* šroubováky.

# sestavení ROBOTICKÉHO RAMENE

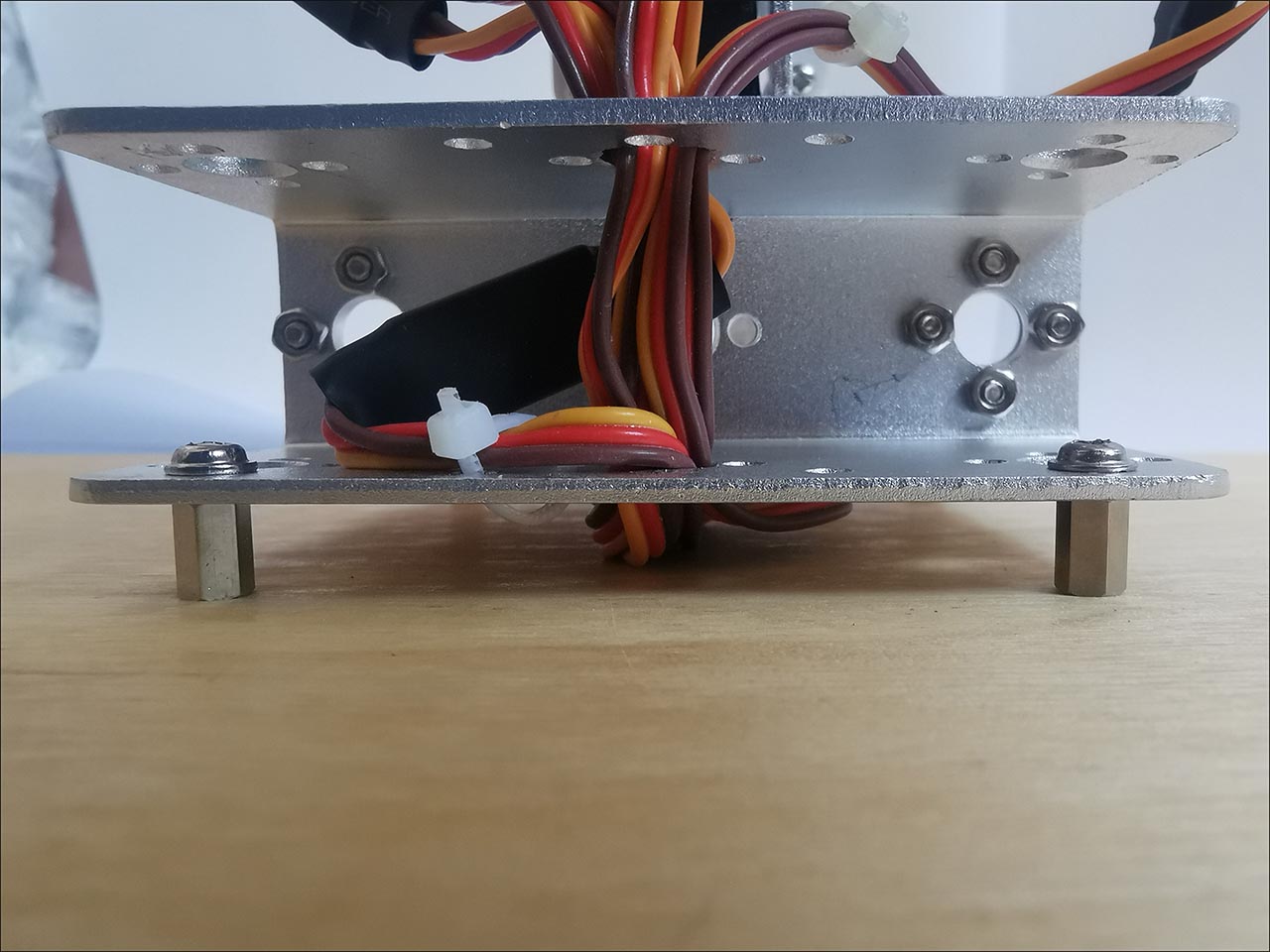
Po přípravné fázi jsem se tedy pustil do samotného sestavení ramene. Situaci mi trochu zkomplikovala absence montážního návodu. Montáž jsem tedy provedl podle internetového instruktážního videa jednoho z nadšenců, vlastníka stejné stavebnice.

Ještě před sestavením ramene jsem si ověřil funkčnost všech servomotorů. Následná výměna vadného serva by nemusela být až tak jednoduchá.

Samotné sestavení ramene proběhlo bez problémů.

Nyní jsem musel vymyslet systém ukotvení robotického ramene k základové desce. Jako základovou desku jsem použil modelářskou překližku o rozměrech 600x300x6,0 mm. Tato velikost mi přišla jako vyhovující pro demonstraci funkčnosti ramene, a i pro případný přenos pro osobní prezentaci ve škole. Do tohoto materiálu lze velmi jednoduše navrtávat otvory, popř. je dále upravovat. Pro lepší údržbu jsem desku z vrchní strany nastříkal čirým 2K lakem.

Základnu robota jsem se rozhodl umístit na kovové šestihranné distanční sloupky. To mi umožnilo elegantní skryté vyvedení kabelů servomotorů pod základovou desku.



*Obrázek č. 1 Uchycení základny robota a průchod kabeláže základnovou deskou*

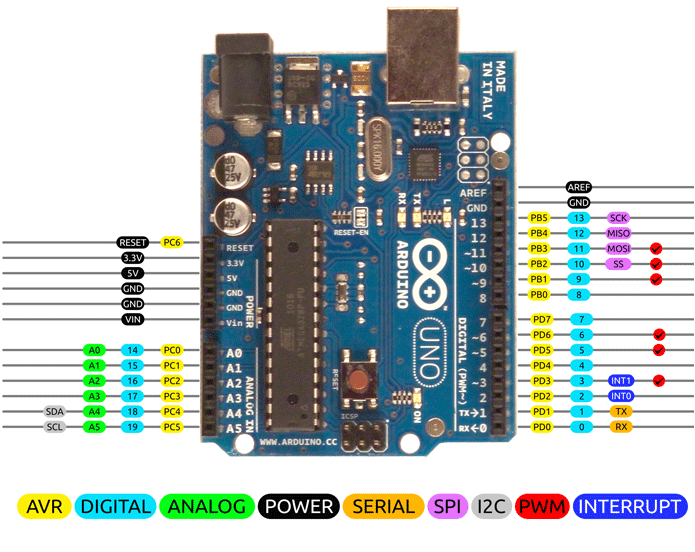
# Využité technologie

## Popis hardware

### Vývojová deska Arduino Uno Rev 3

Základ projektu tvoří vývojová deska Arduino Uno (obrázek č. 2). Tento model Arduina jsem zvolil s ohledem na potřeby na ovládání šesti osého ramene. Disponuje 6 analogovými vstupy pro ovládání joysticky a 6 PWM (pulzní šířková modulace) výstupy pro ovládání servomotorů. Jak jsem již zmínil, zejména z finančních důvodů a také prototypování jsem dal přednost Arduinu před Raspberry Pi.

* Mikroprocesor ATmega328P
* Provozní napětí 5 V
* Vstupní napětí 7–12 V
* Digitální piny 14 (6 z nich PWM)
* Analogové vstupy 6





*Obrázek č. 2 Arduino Uno Rev 3 Obrázek č. 3 Arduino Pinout Diagram*

### Servomotory MG996R

Servomotory byly součástí zakoupené stavebnice. Jejich hlavní výhodou je větší točivý moment než u krokových motorů. Dále k nim nejsou potřeba žádné motorové drivery. Jejich provozní napětí činí 4,8 – 7,2 V, točivý moment 9,4 kg při 4,8 V, mohou se otáčet až o 180°. Zkratka MG znamená, že je servo osazeno kovovými převody (metal gear), takže mají delší životnost.



*Obrázek č. 4 MG996R*

### Dual-axis XY joystick

Jedná se o modul joysticku KY-023.

Tento joystick umožňuje snímat pohyb ve dvou osách X a Y. Modul joysticku je také vybaven tlačítkem, které se aktivuje zmáčknutím hmatníku joysticku. Hodnotám X a Y odpovídá analogová úroveň napětí 0-5 V. Mechanické provedení tvoří deska s trimry 10 kΩ, tlačítkem a mechanismem pro uchycení a pohyb osy joysticku. Na okraji modulu je pin lišta pro připojení k řídící desce. Ovládání joysticku je velmi pohodlné a přesné.

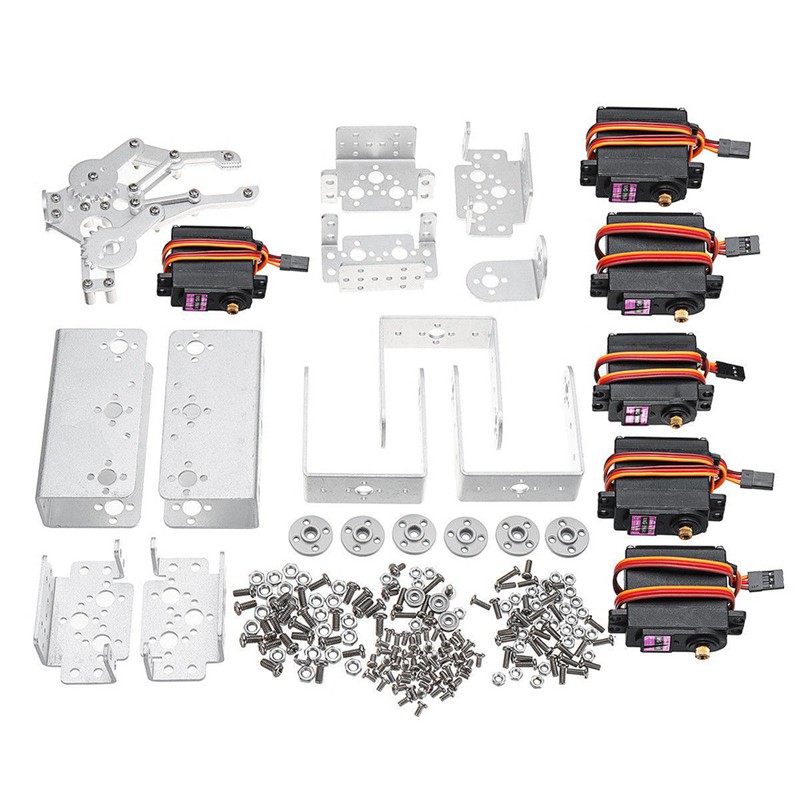
Výstupy modulu joysticku VRx a VRy budou zapojeny na analogové vstupní piny Arduina (A0 – A5). Úroveň napětí na vstupních pinech bude následně pomocí A/D převodníku Arduina převedena na číselnou hodnotu 0-1023. Tento rozsah úrovní odpovídá desetibitovému A/D převodníku.

Pohyb hmatníku joysticku z jedné krajní polohy do druhé provede změnu získané hodnoty od 0 do 1023. Toto platí jak pro osu X, tak i pro osu Y. Středové poloze hmatníku odpovídá hodnota kolem 512.

*Obrázek č. 5 Dvouosý joystick modul*

### 6 DOF hliníková robotická konstrukce ramene

Prostřednictvím internetu jsem si vyhledal a zakoupil cenově dostupnou stavebnici. Zvolil jsem verzi stavebnice, jejíž součástí bylo také 6 servomotorů (popsaných v části **3.1.2)**. Jedná se o set hliníkových konstrukčních prvků, již sestaveného chapadla, šesti servomotorů a spojovacího materiálu. Robot této konstrukce umožňuje otáčecí úhel 180°.



*Obrázek č. 6 stavebnice 6 DOF robotické rameno*

## Software

### Jazyk Arduino

Program sloužící k ovládání ramene jsem napsal v jazyce Arduino. Ten vychází z jazyka Wiring, který je velice podobný jazyku C nebo C++. Lze v něm využít návyků z C jazyků, čímž je mi blízký. Program psaný v jazyce C má vždy hlavní funkci main(). U Arduina jsou takové funkce dvě – setup() a loop(). Setup() slouží k inicializaci a nastavení hodnot a loop() je stále se opakující funkce programu.

### Python 3.9.1

Grafické uživatelské rozhraní je napsané v programovacím jazyce Python. Python je velice přehledný a jednoduchý pro čtení. Použil jsem verzi, která podporuje knihovnu Tkinter. Tedy nejnovější verzi tohoto jazyku.

### Arduino IDE 1.8.13

Jako integrované vývojové prostředí jsem si na otestování součástek vybral Arduino IDE 1.8.13. V něm jsem využil příklad „sweep“ k seznámení se servomotory.

### KiCAD 5.1.9

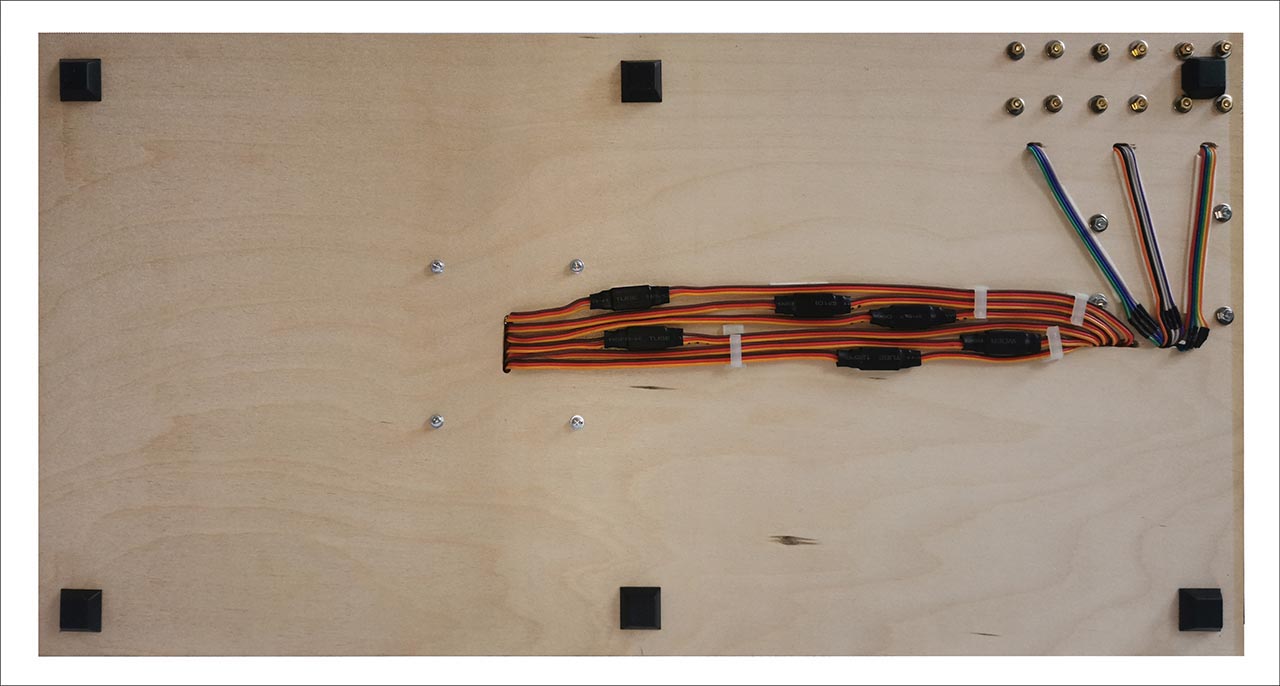
Pro tvorbu návrhu a schématu jsem využil program KiCAD. Jedná se o multiplatformní open-source software pro návrh schémat a plošných spojů. Důležitým faktorem při navrhování byla funkce, která kontroluje chyby ve schématu, jako je například nepřipojený pin.

# Způsoby řešení a použité postupy

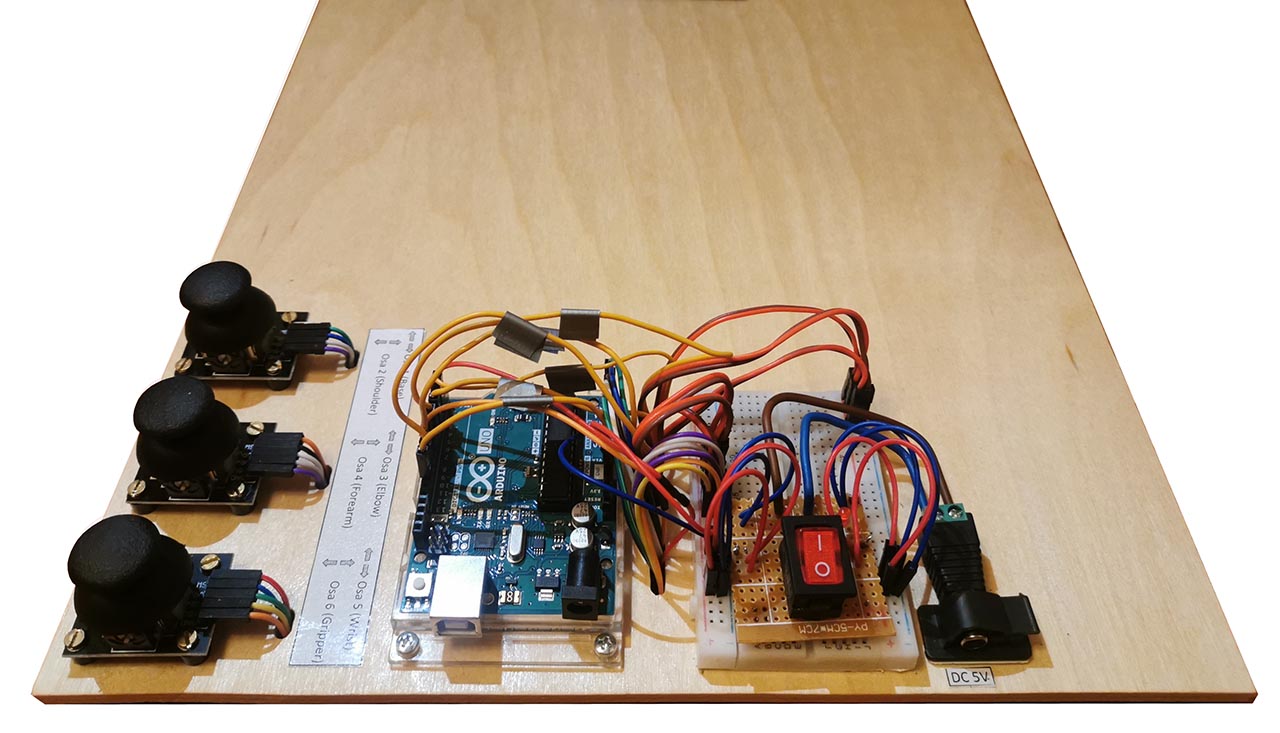
## Kabeláž

Kabely servomotorů jsem ke konstrukci robotického ramene přichytil pomocí stahovacích pásků a to tak, aby bylo možné využít maximálního rozsahu pohybu všech jeho částí. Svazek šesti kabelů jsem protáhl rozšířeným otvorem základnové části ramene pod tuto základnu a protáhl připraveným otvorem pod překližkovou základnovou desku. Tímto řešením se mi vyčistil pracovní prostor ramene. Kabely servomotorů bylo také zapotřebí prodloužit, a to pomocí prodlužovacích kabelů s konektory JR. Všechny konektorové spoje těchto kabelů jsem zajistil proti rozpojení smršťovacími bužírkami. Následně jsem přichytil kabely k základové desce pomocí plastových klipů a oboustranné lepicí pásky (tou zejména konektorové spoje). Takto prodloužené kabely jsem ještě délkově upravil dle potřeby, zakončil konektory DuPont a dalším otvorem vyvedl zpět nad základnovou desku pro jejich zapojení do desky Arduina a nepájivého kontaktního pole.

Kabely joysticků jsem opět vyvedl pod základnovou desku. Jejich délku jsem upravil pájeným spojem izolovaným smršťovací bužírkou. I tyto kabely jsem přilepil k základnové desce oboustrannou lepicí páskou a stejně jako kabely servomotorů vyvedl zpět nad základnovou desku otvorem mezi deskou Arduina a deskou nepájivého kontaktního pole.



*Obrázek č. 7 Kabeláž – spodní strana*



*Obrázek č. 8 Kabeláž – horní strana*

## Napájení

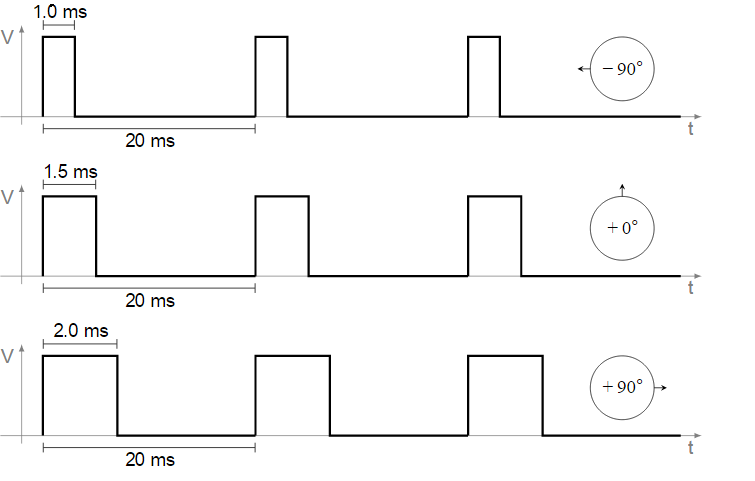
Pro napájení desky Arduina a servomotorů jsem zvolil externí zdroj napětí 5 V. Arduino samotné by nebylo schopno dodat dostatečné množství proudu pro všechna použitá serva a robotické rameno by nefungovalo správně. V ideálním případě by měl být napájecí zdroj schopen dodat alespoň 2 A pro každé použité servo. Vzhledem k vysokým proudovým nárokům servomotorů jsem si vyrobil vlastní napájecí modul. K napájecímu konektoru je tento modul připojen vodiči o průřezu 1,5mm2. Modul je osazen indikační LED diodou. Výstup z tohoto modulu je řešen čtyřmi plusovými (červenými) vodiči DuPont a čtyřmi minusovými (modrými) vodiči DuPont. Toto řešení jsem zvolil za účelem snížení proudového zatížení jak vodičů samotných, tak i snížení proudového zatížení desky nepájivého kontaktního pole. Vždy dva plusové a dva mínusové vodiče jsou zapojeny na samostatnou řadu napájecích pinů desky nepájivého kontaktního pole. Na každé straně této desky je potom zapojeno napájení tří servomotorů. Na napájecí modul jsem připájel čtyři piny a těmito jsem tento modul mechanicky (neelektricky) zasadil do nepájivého pole.

## Princip fungování servomotoru

Aktuální poloha hřídele je pomocí potenciometru převedena na elektrický signál a porovnána s příkazovým vstupním signálem. Příkazový vstup je zadán podle požadovaných parametrů hřídele. Pokud se signál zpětné vazby liší od zadaného vstupu, vygeneruje se chybový signál. Chybový signál se poté zesílí a použije jako vstup do motoru, což způsobí otáčení. Když se hřídel dostane do požadované polohy, chybový signál se stane nulovým, tím pádem motor zůstane stát v klidu. Vstup příkazu je uveden ve tvaru elektrických impulsů. Protože skutečný vstup aplikovaný na motor je rozdíl mezi signálem zpětné vazby (aktuální poloha) a aplikovaným signálem (požadovaná poloha), rychlost motoru je úměrná rozdílu mezi aktuální polohou a požadovanou polohou. Servomotor je řízen vysláním signálu PWM (pulzně šířkově modulovaný) přes řídící vodič. Tento pulz je vyslán každých 20 milisekund, jeho šířka určuje polohu hřídele.

U Arduina k nastavení frekvence signálu PWM stačí vyvolat knihovnu Servo.h, frekvence se díky této knihovně nastaví automaticky.

#include <Servo.h>



*Obrázek č. 9 Princip servomotorů*

A/D převodník od Arduina má 10bitové rozlišení, takže se celočíselné hodnoty pohybují od 0-1023 (2^10), to znamená, že bude mapovat vstupní napětí mezi 0 a 5 volty na celočíselné hodnoty mezi 0 a 1023. Servo se ale může pohybovat pouze od 0 do 180, takže je nutné použít speciální funkci map() v Arduinu, která automaticky spojí hodnoty A/D převodníku a servomotoru, nakonec tato funkce změní hodnoty joysticku.

float dx = map(analogRead(xdirPin), 0, 1023, 0, 179);

## Princip fungování programu

Pro uvedení ramene do pohybu je zapotřebí vytvořit si objekty typu Servo. Ke každému takovému objektu je nutné připojit různé piny (pouze PWM), které jsem si nastavil do proměnných. Každý servomotor se potom vyvolává zvlášť.

Servo servoBase, servoShoulder, servoElbow, servoGripper, servoWrist, servoForearm;

*Vytvoření objektů Servo*

Přiřazení jednotlivých pinů k servomotorům.

servoBase.attach(basePin);

...

*přiřazení pinu*

Pro přiřazení joysticků na analogový pin je nutné vytvořit si proměnnou typu int.

**int** xdirPin = A0;

**...**

Poté ve funkci setup() jsem nastavil všechny proměnné jako input

pinMode(xdirPin, INPUT);

### Inicializace základní polohy

Rameno se po připojení napájecího zdroje samo nastaví do výchozí polohy. To proto, aby se s ním snadněji operovalo, popř. nepoškodilo desku nebo samo sebe. Pro tento pohyb slouží funkce StartingPosition(). Stačí si nastavit proměnné a poté vyvolat funkci write(), tato funkce je vyvolaná hned po přiřazení pinu ve funkci setup().

void StartPosition() {

posBase = startBase;

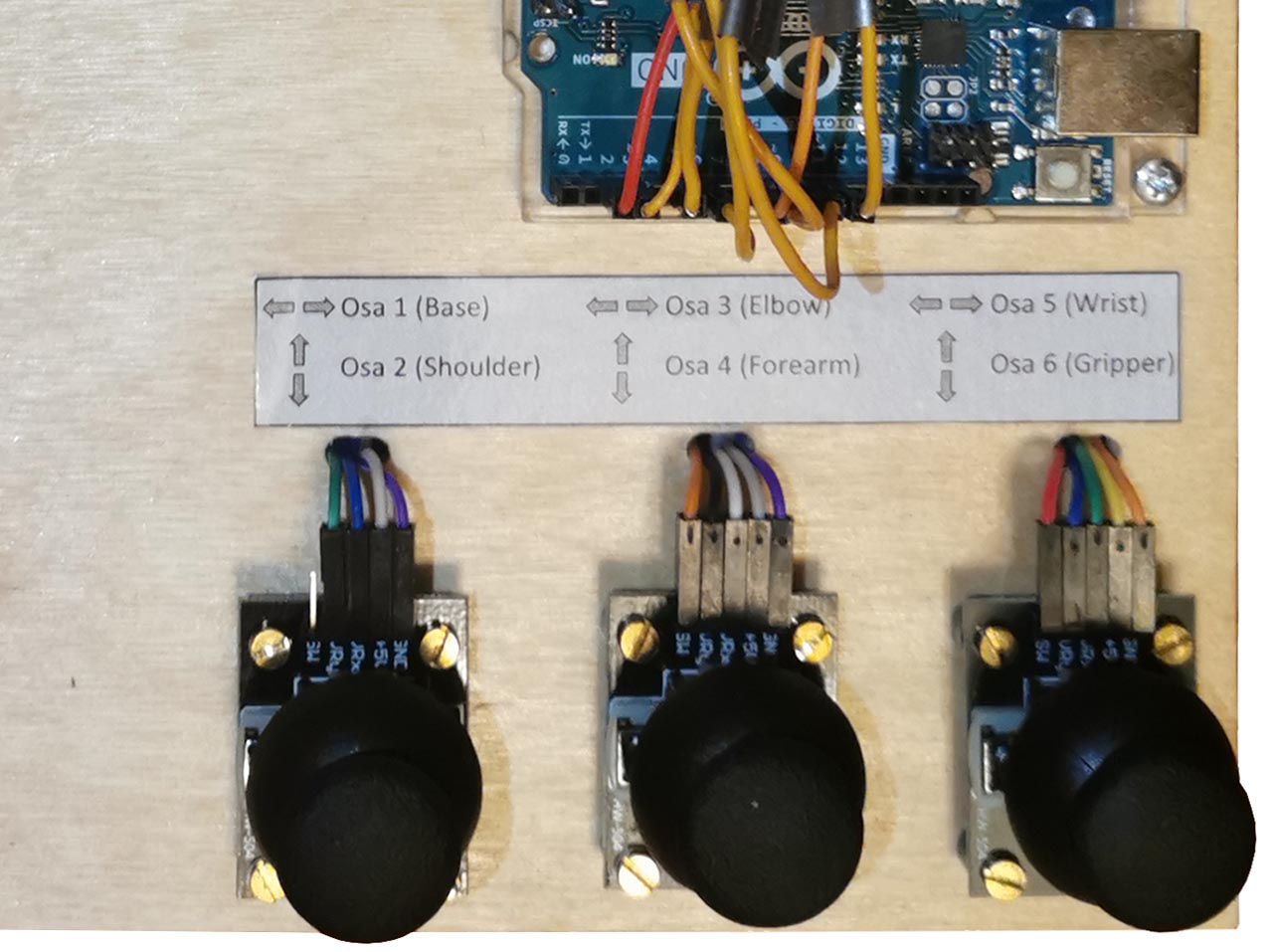
...

servoBase.write(posBase);

}

### Pohyb

Nejdříve se zmapuje a převede hodnota joysticku pomocí funkce map(). Tuto zmapovanou hodnotu přiřazujeme dále do proměnné, pomocí které měníme polohu servomotoru funkcí write(). Je nutné omezit otáčecí úhel, pro každý motor si inicializujeme proměnnou min a max. Pokud je stávající hodnota menší, nebo větší než nastavený limit, program pokaždé nastaví hodnotu dosaženého limitu jako polohu.



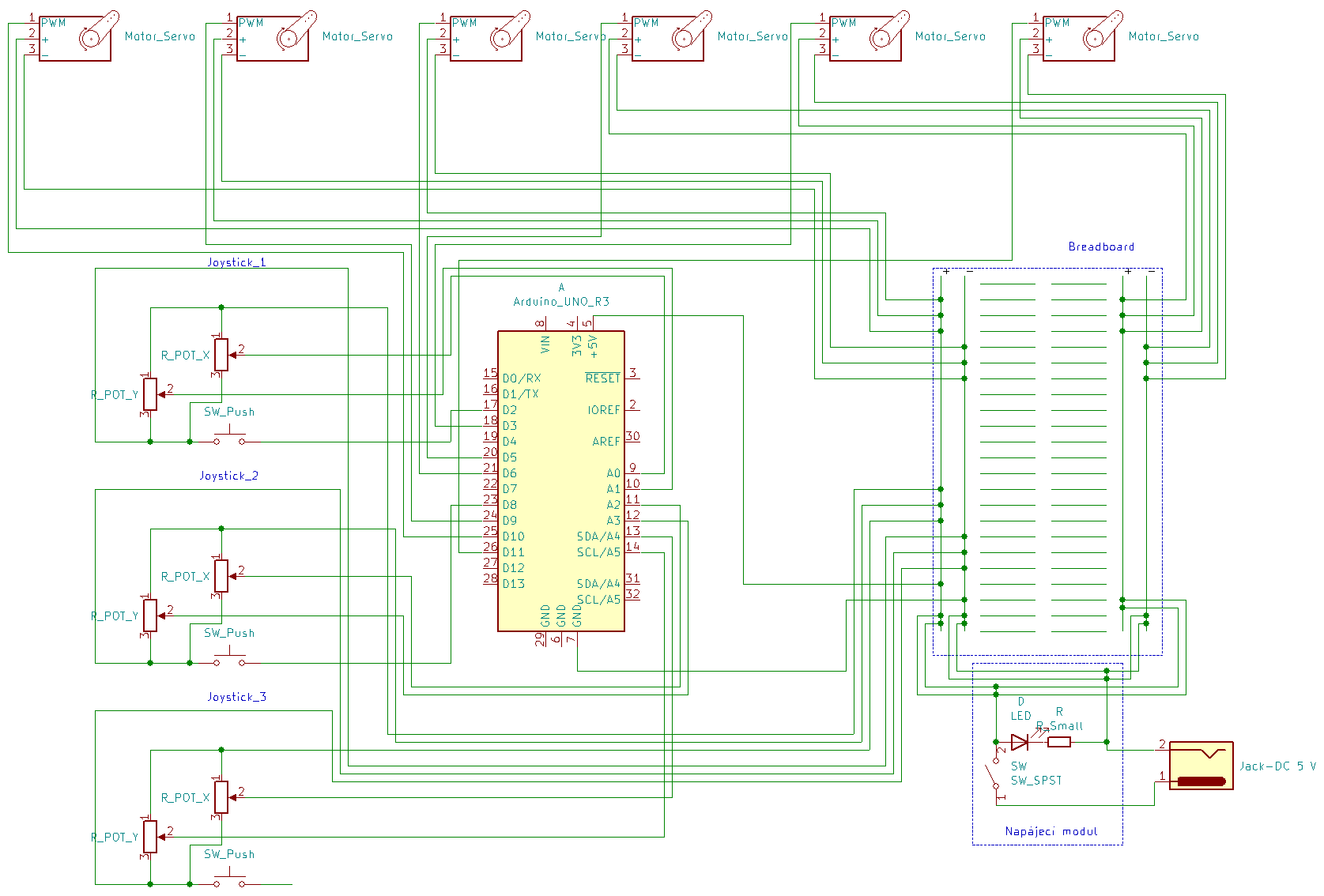
*Obrázek č. 10 Joysticky - popis*

### Uživatelské rozhraní

Ve druhé verzi programu se rameno ovládá grafickým uživatelským rozhraním místo joysticku. K takovému ovládání je nutno importovat knihovnu PySerial, která otevře sériový port pro komunikaci s Arduinem, název tohoto portu zjistíme ve správci zařízení, v mém případě se jednalo o port COM1. Samotné okno je řešeno pomocí knihovny Tkinter, která poskytuje možnost vytvořit grafická okna. V rozhraní jsou tlačítka, pomocí kterých můžeme nastavovat hodnotu pro určitý servomotor a textové pole, ve kterém se zobrazují aktuální souřadnice. Python odesílá informace skrze sériový port, kliknutím na některé z tlačítek program zjistí kód a podle tohoto kódu se vyvolá funkce callback, která odešle úhel, o který se má servo pootočit. Rozhraní má tlačítko pro resetování polohy a pro získání náhodné pozice.

# Výsledky řešení

Rameno je v současné době plně funkční a s výsledkem jsem spokojen. Po několikadenním testování robot splnil má počáteční očekávání. Vodiče napájení zařízení jsou zatím vedeny skrze nepájivé kontaktní pole. Tyto vodiče by šlo také zapojit do svorkovnic a zajistit tak spolehlivější kontakty. Prvotním cílem bylo zprovoznit ovládání pomocí potenciometrů, našel jsem ale mnohem efektivnější řešení a to joysticky. Jako další cíl bylo ovládání pomocí grafického uživatelského rozhraní, které je také funkční. Pro zpřehlednění jsem doplnil desku popisem ovládacích prvků.



*Obrázek č. 11 Schéma zapojení*

# Závěr

Cílem projektu bylo vytvořit funkční robotické rameno, které by se dokázalo pohybovat pomocí joysticků anebo grafického uživatelského rozhraní a mohlo sloužit k edukačním účelům. Všechny mé cíle byly splněny a zařízení funguje bez problému.

Do budoucna bych chtěl vylepšit pohyb pomocí trigonometrických vzorců a zprovoznit inverzní kinematiku. Zakoupil jsem bluetooth modul HC-05, takže by bylo možné vylepšení o ovládání pomocí mobilní aplikace, také lze vytvořit ovládání pomocí webové stránky, další možností je vytvoření konstrukce ramene vytištěné na 3D tiskárně. V současné době pracuji na funkci Record and Play, robot díky této funkci po stisknutí tlačítka nahraje pohyb a stiskem druhého tlačítka pohyb začne opakovat.

GitHub: https://github.com/michalsemenisin/zaverecny\_projekt

Seznam použitýCH INFORMAČNÍCH ZDROJů

[] *Joystick control* [online]. [cit. 2021-01-18]. Dostupné z: https://create.arduino.cc/projecthub/RiddledExistence/controlling-a-servo-motor-with-thumb-joystick-46a4d3

[2] *Servo control* [online]. [cit. 2021-01-18]. Dostupné z: https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/

[3] *Python gui* [online]. [cit. 2021-01-18]. Dostupné z: http://arduinolearning.com/code/led-control-with-arduino-and-python-tkinter.php

[4] *Seznámení s Arduinem [online]. 2016 [cit. 2021-01-18]. Dostupné z: https://navody.dratek.cz/zaciname-s-arduinem/*

[5] *Robotic arm [online]. [cit. 2021-01-18]. Dostupné z:* [*https://create.arduino.cc/projecthub/gmel31/most-simple-robot-arm-out-there-easy-3912f1*](https://create.arduino.cc/projecthub/gmel31/most-simple-robot-arm-out-there-easy-3912f1)

[6] *Robotic arm [online]. [cit. 2021-01-18]. Dostupné z: https://github.com/netlabtoolkit/VarSpeedServo*