



Středoškolská technika 2024

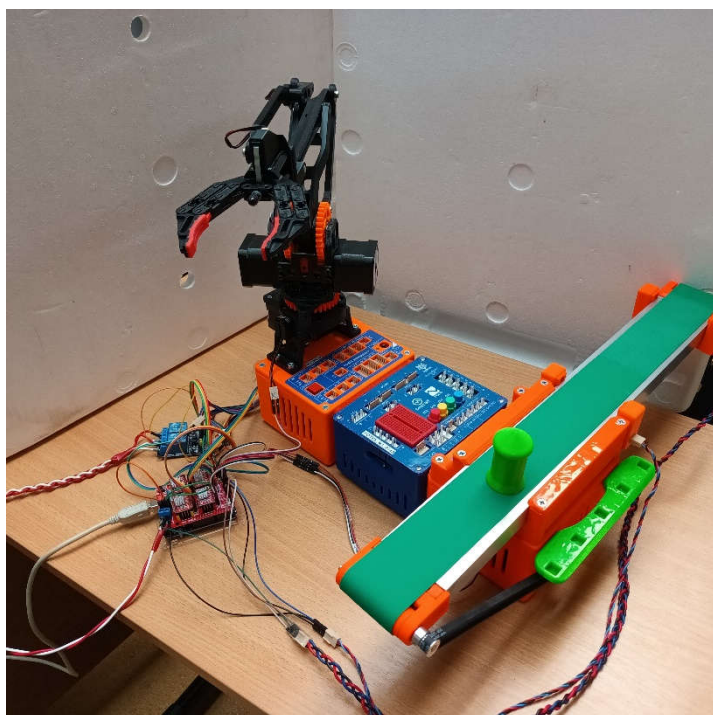
Setkání a prezentace prací středoškolských studentů na ČVUT

REPASE ROBOTICKÉ RUKY

Filip Houžvička

SPŠ, SOŠ a SOU Hradec Králové

Hradební 1029, Hradec Králové



**Střední průmyslová škola
Střední odborná škola
Střední odborné učiliště
Hradec Králové**



PRAKTICKÁ ROČNÍKOVÁ PRÁCE MANIPULÁTOR

Filip Houžvička

Obor: 26-41-L/01 – Mechanik Elektrotechnik

Školní rok: 2023/24

Anotace

Hlavním cílem této ročníkové práce je nahradit nefunkční ovládací jednotku FOXEE, programovací platformou Arduino Uno s pomocí CNC shieldu V3, a také pomocí Arduina rozhýbat a naprogramovat manipulátor.

Annotation

The main goal of this thesis is to replace the non-functioning FOXEE control unit with an Arduino Uno programming platform with the help of a CNC shield V3, and also to use Arduino to move and program the manipulator.

Klíčová slova

Krokové motory, osy, arduino, CNC shield, zdroj, pohybový pás, senzory, schéma, program, elektrický proud, řídicí jednotka, FOXEE, relé, senzory, mikrospínač

Key words

Stepper motors, axes, arduino, CNC shield, source, motion belt, sensors, diagram, program, electric current, control unit, FOXEE, relay, sensors, microswitch

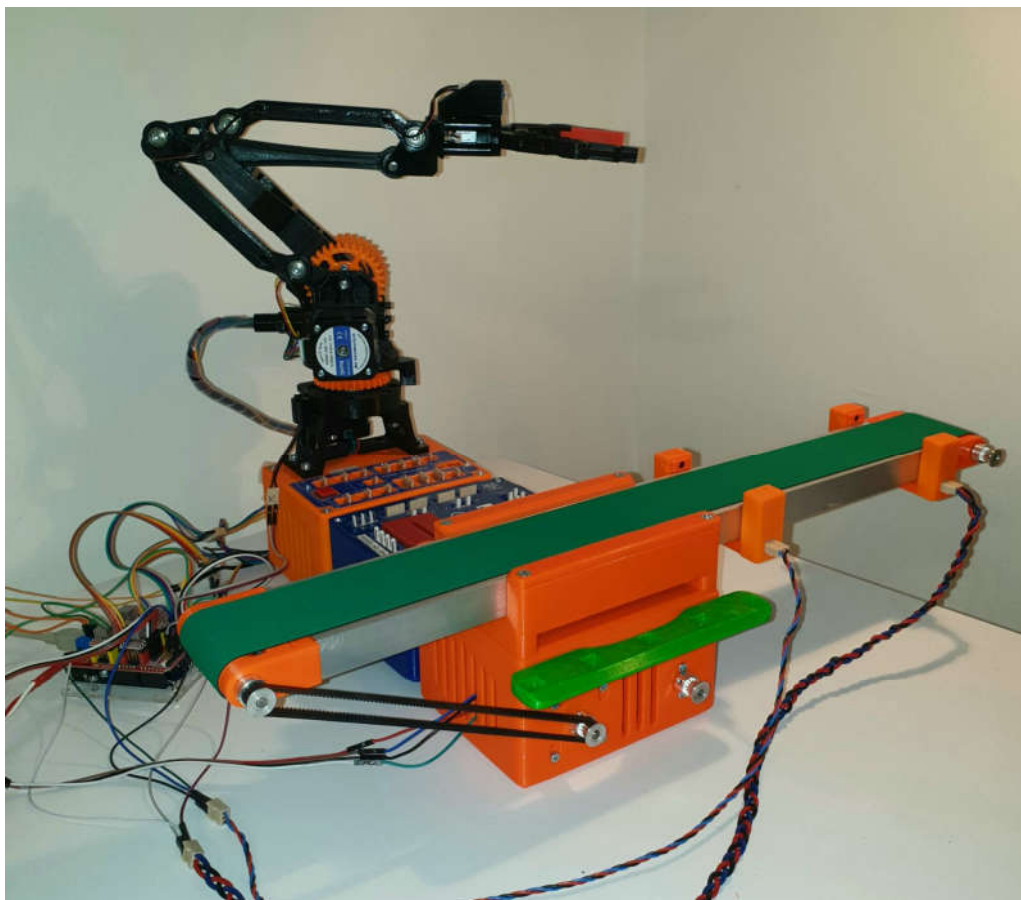
Obsah

ÚVOD.....	5
SEZNAM KOMPONENTŮ	6
SEZNAM ZKRATEK A SYMBOLŮ	6
1 TEORETICKÁ ČÁST	7
1.1 ZDROJE.....	7
1.1.1 NAPÁJECÍ ZDROJ	8
1.2 MIKROSPÍNAČE, SENZORY A RELÉ	10
1.2.1 MIKROSPÍNAČE.....	10
1.2.2 SENZORY	12
1.2.3. RELÉ	13
1.3 ŘÍDÍCÍ JEDNOTKA	14
1.3.1 Arduino Uno	14
1.3.2 CNC Shield	15
1.3.3 A4988 driver	16
2 PRAKTICKÁ ČÁST	17
2.1 NÁHRADA ŘÍDÍCÍ JEDNOTKY FOXEE ARDUINEM	17
2.2 PŘIDÁNÍ CNC SHIELDU	18
2.3 ZAPOJENÍ KROKOVÝCH MOTORŮ A KONCOVÝCH ČIDEL.....	18
2.4 KLEŠTĚ.....	20
2.5 ČTVRTÁ OSA A	20
2.6 SENZORY	21
2.7 KÓD PRO ARDUINO	23
3 ZÁVĚR	32
4 POUŽITÁ LITERATURA.....	33
5 ZDROJE K OBRÁZKŮM.....	34

ÚVOD

V dnešní době je čím dál tím větší zájem o usnadnění si práce, a to možná o co nejvíce. Pro to, aby všechno běželo, jak má a navíc samo, je potřeba si věci zautomatizovat. Kdo a kde jinde si má někdo něco takového vyzkoušet než mladý žáci na středních školách. Pro takové zaměření je ideální si zvolit obor ohledně elektro. Osobně vidím v zaměření automatizace velkou budoucnost, a proto jsem si také vybral tuto ročníkovou práci. Na mé ročníkové práci je možné si vyzkoušet a nejlépe se i naučit programování a sestavu řídicího systému pro ovládání manipulátoru s pohyblivým pásem. Nejenom že jsem musel vymyslet a zcela nahradit řídicí jednotku za jinou, ale také jsem měl velkou šanci si vyzkoušet něco nového, a to programovat v noví pro mě zcela cizí aplikaci Arduino IDE. Nejvíce se mi líbí, že jsem si mohl vyzkoušet zprovoznit zmenšeninu „výrobní linky v průmyslu“.

Obrázek 0: Manipulátor



SEZNAM KOMPONENTŮ

- Arduino UNO
- CNC shield
- motor driver A4988
- 2-kanálové relé
- propojovací kabely
- počítačový zdroj
- krokové motory
- manipulátor
- pohyblivý pás
- mikropínače
- senzory

SEZNAM ZKRATEK A SYMBOLŮ

- AC – Alternate current – střídavé
- DC – Direct current – stejnosměrné
- PSU – z anglického "Power Supply Unit" = napájecí jednotka
- GND – Ground = zem
- V – Volt, jednotka elektrického napětí

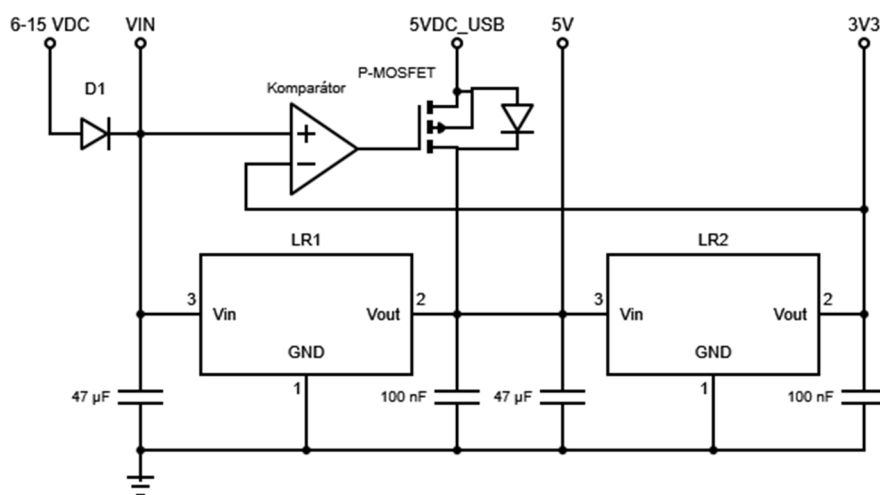
1 TEORETICKÁ ČÁST

V teoretické části se budu zabývat obecným vysvětlením jednotlivých částí manipulátoru a rozeberu veškeré jeho ovládací komponenty.

1.1 ZDROJE

Nezbytnou součástí systému je samotné napájení, ke kterému je použit napájecí zdroj z počítače, bez kterého by se manipulátor ani nepohnul. CNC shield je napájen 12V, které jsou přivedeny ze zdroje. Zdroj také napájí 2-kanálové relé, které slouží k otevírání a zavírání kleští na manipulátoru. K napájení Arduina nám bohatě stačí přívod z počítače, ale může být také napájeno přes souosý konektor, s přívodem 6–15 V DC jak je uvedeno ve schématu.

Obrázek 1: Obvod napájení Arduina Uno



1.1.1 NAPÁJECÍ ZDROJ

Napájecí zdroj (PSU, z anglického "Power Supply Unit") slouží k převodu střídavého proudu z elektrické sítě na stejnosměrný proud potřebný pro různé elektronické komponenty. Jeho význam spočívá v zajištění stabilního a spolehlivého napájení, což je klíčové pro správnou funkci zařízení. Kvalitní PSU chrání před přepětím a výkyvy v elektrické síti, čímž předchází potenciálnímu poškození citlivých elektronických komponent. Důležitým aspektem napájecího zdroje je jeho účinnost, kterou vyjadřuje certifikace jako 80 PLUS, zaručující efektivní přeměnu energie s minimálními ztrátami.

Napájecí zdroje se liší nejen výkonem, ale i typem konektorů a dalšími vlastnostmi. Některé PSU nabízejí modulární kabeláž, což umožňuje uživateli připojit pouze ty kabely, které skutečně potřebuje, čímž se zlepšuje proudění vzduchu uvnitř skříně a usnadňuje se správa kabelů. Kromě toho mohou mít napájecí zdroje různé bezpečnostní funkce, jako jsou ochrana proti přepětí, ochrana proti nadproudu a ochrana proti zkratu, které dále zvyšují jejich spolehlivost a bezpečnost.

Výběr správného napájecího zdroje závisí na energetických požadavcích celého systému. Je důležité zvolit PSU s dostatečným výkonem, aby mohl bez problémů napájet všechny komponenty, a zároveň poskytoval určitou rezervu pro budoucí rozšíření nebo upgrade. Kromě výkonu a účinnosti je také dobré zvážit hlučnost napájecího zdroje, protože některé modely jsou vybaveny tichými nebo dokonce pasivními chladiči, které minimalizují hluk. Celkově je napájecí zdroj jedním z nejdůležitějších komponent každého elektronického systému, který zajišťuje jeho stabilní a bezpečný provoz.

Obrázek 2: Napájecí zdroj z počítače



Obrázek 3: Napájecí zdroj z počítače



1.2 MIKROSPÍNAČE, SENZORY A RELÉ

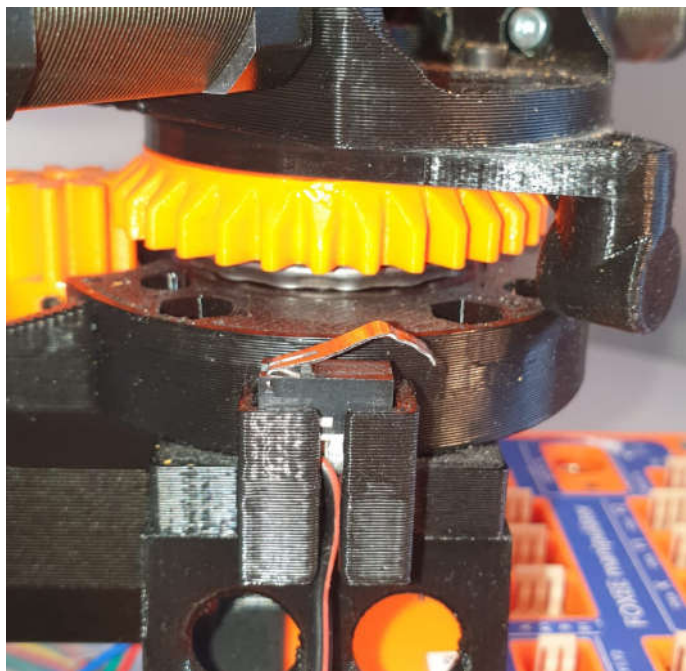
1.2.1 MIKROSPÍNAČE

Mikrospínače neboli koncová čidla jsou nezbytnou součástí manipulátoru/ramene. Funguje na jednoduchém principu, když se mikrospínač aktivuje čili rameno ho při pohybu zamáčkne tak mikrospínač vyšle signál do CNC shieldu, a s tímto už můžeme pracovat dál, např. nadefinovat kód kam se má rameno pohnout.

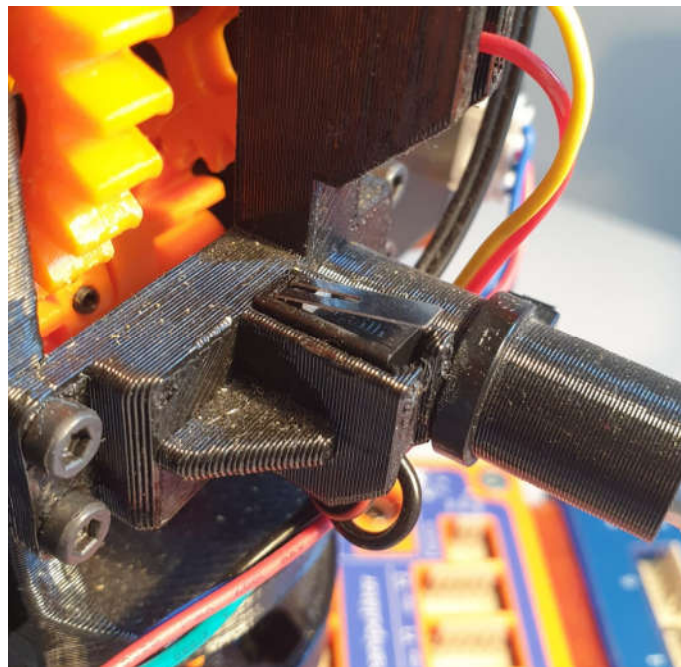
Otáčení ramene zajišťují tři osy řízené krokovými motory, každá z os řídí jiný směr pohybu. Máme Osu X a ta pohybuje ramenem doleva a doprava, potom je tu osa Y, která pohybuje ramenem dopředu a dozadu a třetí a poslední osa ramene je Osa Z, která pohybuje ramenem nahoru a dolů.

Tyto osy by nebylo možné programovat, kdybychom neměli u každé z nich koncové čidlo. Říká se jim koncová, protože jsou na konci pohybu ramene, a jsou nesmírně důležitá neboli bez nich by rameno nevědělo kdy a kde se nachází. Tím že rameno má konečnou neboli výchozí pozici, tak můžeme definovat kam a o kolik se má rameno posunout.

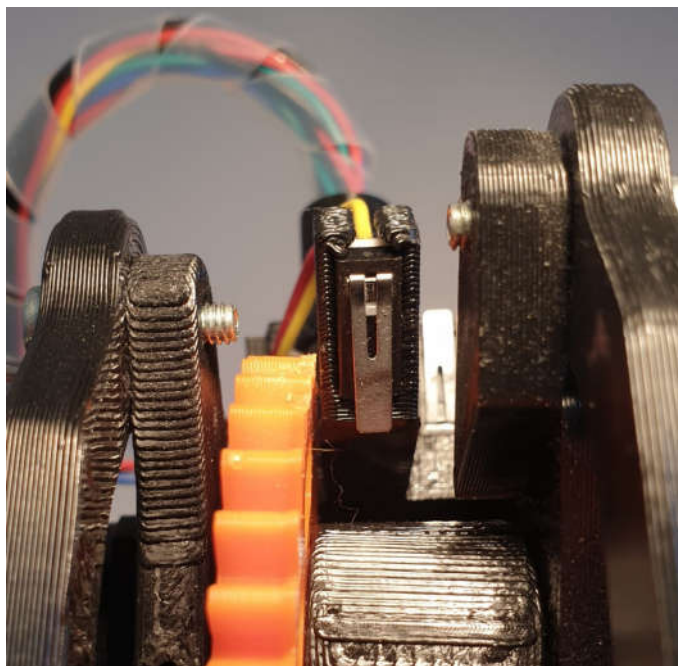
Obrázek 4: Mikrospínač pro osu X



Obrázek 5: Mikrospínač pro osu Z



Obrázek 6: Mikrospínač pro osu Y



1.2.2 SENZORY

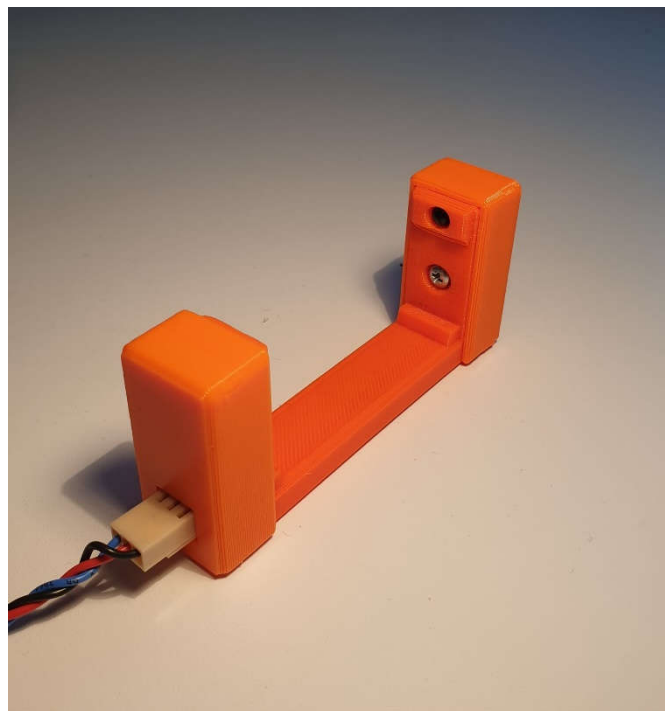
U pohybového pásu jsou zapojeny dva senzory, které umožňují měnit směr pásu podle toho, skrz jaké čidlo projede předmět. Jsou využity dva senzory, které byli součástí manipulátoru.

Senzor se používá jako obecnější termín pro zařízení, které měří fyzikální jevy a převádí je na elektrické signály. Senzory, které zde zmiňuji jsou konkrétně infračervené senzory typu závorové, které se často používají pro detekci přerušení světelného paprsku. Tento typ senzoru se skládá ze dvou hlavních částí: vysílače (diody) a přijímače (fototranzistoru). Vysílač (dioda) emituje infračervené světlo a přijímač (fototranzistor) detekuje přítomnost tohoto světla. Pokud se mezi vysílač a přijímač dostane překážka, signál se přeruší, toto přerušení signálu je pak detekováno a informace o přerušení je poslána do řídicí jednotky neboli opět do CNC shieldu. A stejně jako u Mikrospínačů můžeme s tímto signálem pracovat.

Obrázek 7: Senzory pro pás



Obrázek 8: Senzor



1.2.3. RELÉ

Pro ovládání kleští je použito 2-kanálové relé, které zajišťuje otevírání a zavírání kleští. Relé je napájeno ze zdroje 12V. Jedno relé zajišťuje otevírání kleští a druhé zavírání kleští.

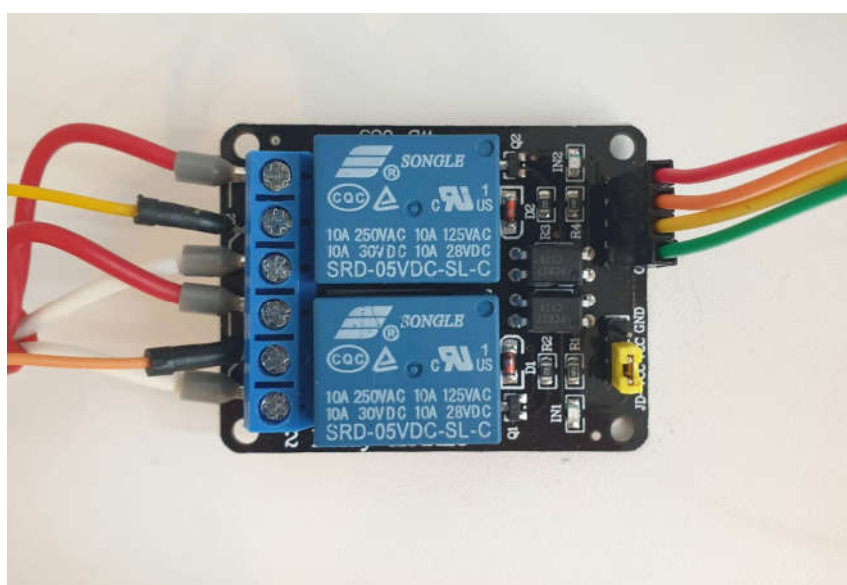
2-kanálové relé je elektronické zařízení, které umožňuje ovládat dva samostatné elektrické okruhy pomocí jednoho řídicího signálu. Relé funguje jako spínač, který může zapínat nebo vypínat elektrický proud v okruzích, které ovládá. Hlavními komponenty relé jsou cívka, kontakty a izolační bariéra. Když je na cívku aplikováno napětí, vytváří magnetické pole, které přitahuje kotvu a způsobí změnu stavu kontaktů. Když je napětí odstraněno, pružina vrátí kotvu zpět do původní polohy.

Často se používá v domácí automatizaci pro ovládání osvětlení, topení nebo zavlažovacích systémů, v průmyslových automatech pro řízení motorů a ventilů a v bezpečnostních systémech pro ovládání dveřních zámků nebo alarmů.

Výhodou relé je bezpečné oddělení nízkonapěťového řídicího obvodu od vysokonapěťových výkonových obvodů, spolehlivé spínání méně citlivé na šum než elektronické spínače a flexibilita v ovládání různých typů zátěží (AC i DC).

Typické specifikace zahrnují napětí cívky 5V, 12V nebo 24V DC, spínací kapacitu 10A při 250V AC nebo 10A při 30V DC a nezávislé ovládání dvou kanálů.

Obrázek 9: 2-kanálové relé



1.2 ŘÍDÍCÍ JEDNOTKA

Řídící jednotka se skládá ze tří částí: Arduino Uno, CNC Shieldu V3 a A4988 driveru pro krokové motory. Všechny tyto části jsou velice důležité pro řízení manipulátoru.

Nyní si rozebereme každou část jednotlivě.

1.3.1 Arduino Uno

Uplatnění Arduina v ročkové práci:

Arduino je použito jako klíčový nástroj pro ovládání tříosého manipulátoru a dopravního pásu. Pomocí této platformy lze přesně řídit pohyb jednotlivých os manipulátoru (X, Y a Z), což umožňuje efektivní a přesné polohování a manipulaci s předměty. Kromě toho Arduino řídí pohyb dopravního pásu, který transportuje objekty k manipulátoru. Senzory připojené k Arduinu poskytují zpětnou vazbu o poloze a stavu předmětů, což umožňuje automatizaci celého procesu manipulace a transportu. Tento systém přispívá k optimalizaci pracovních postupů a zvyšuje produktivitu a přesnost operací v naší aplikaci.

Uplatnění Arduina obecně:

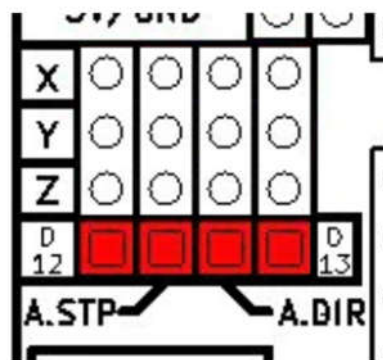
Arduino je univerzální programovací platforma, která se používá k vytváření a prototypování elektronických projektů. Díky snadnému použití a široké komunitě uživatelů umožňuje Arduino vývojářům rychle a efektivně vytvářet různé aplikace, od jednoduchých LED světel až po složité robotické systémy. Arduino desky jsou vybaveny mikrokontroléry, které lze naprogramovat pomocí jazyka založeného na C/C++. To umožňuje uživatelům číst vstupy z různých senzorů a zařízení a na základě těchto dat provádět akce, jako je ovládání motorů, zobrazování informací na displejích nebo komunikace s jinými zařízeními. Arduino tak otevírá dveře k široké škále projektů v oblasti vzdělávání, průmyslu a hobby.

1.3.2 CNC Shield

CNC shield zapojíme do Arduina, což znamená, že už nepracujeme s piny přímo na Arduinu, ale s piny na CNC shieldu. Nejsložitější částí ročníkové práce bylo zjistit, kam jednotlivé komponenty zapojit a jaká čísla mají tyto piny.

CNC shield se často používá pro 3D tiskárny, ale lze ho využít i jinak. Pomocí CNC shieldu ovládáme všechny tři osy manipulátoru. CNC shield také umožňuje využití čtvrté osy, kterou lze řídit dalším krokovým motorem. Tato osa je označena jako A a jejím primárním účelem je klonování jedné ze tří os (X, Y nebo Z), ale propojením správných pinů nám umožňuje samostatné ovládání čtvrté osy A. Tuto osu využíváme k ovládání pásu, který je rovněž řízen krokovým motorem.

Obrázek 10: Propojení pinů k vytvoření osy A

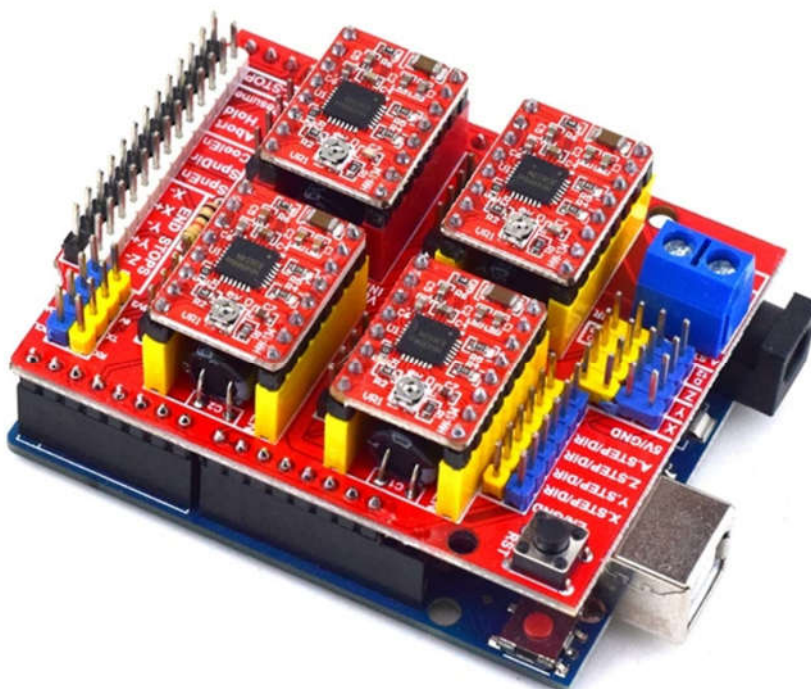


Use D12 and D13 to drive the 4th
stepper driver
(marked as A)

1.3.3 A4988 driver

A4988 driver je důležitou součástí CNC shieldu, který umožňuje řízení krokových motorů. Tento driver je široce využíván pro svou schopnost poskytovat vysoký výkon a přesnost při pohybu motorů. A4988 je snadno integrovatelný s Arduinem a dalšími mikrokontroléry díky svému jednoduchému rozhraní. Jeho schopnost nastavitelných mikrokroků umožňuje jemné řízení pohybu motorů s vysokou přesností. Díky těmto vlastnostem je A4988 driver ideální volbou pro naše CNC aplikace, kde vyžadujeme spolehlivé a přesné řízení pohybu motorů při zachování kompaktního designu a snadné integrace s celým systémem.

Obrázek 11: Osazení Arduina CNC shieldem a A4988 driverem



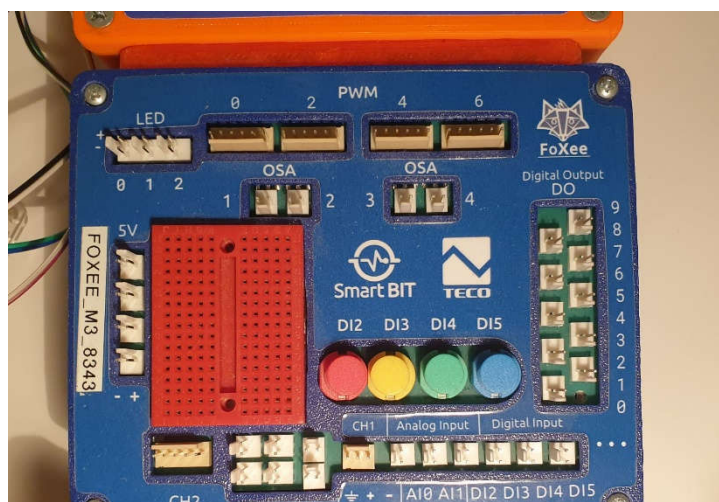
2 PRAKTICKÁ ČÁST

V praktické části vysvětlím, k čemu jednotlivé díly slouží, jakou mají funkci a jak jsem postupoval při konstrukci manipulátoru.

2.1 NÁHRADA ŘÍDÍCÍ JEDNOTKY FOXEE ARDUINEM

Prvním krokem při práci s manipulátorem bylo jeho zapojení a uvedení do pohybu pomocí základní jednotky FOXEE. U manipulátoru byla i základní dokumentace, která však neobsahovala žádné užitečné informace o sestavení a propojení jednotky s manipulátorem a pásem. Bylo zjištěno, že jedna z ovládacích desek pro krokový motor nefunguje, a proto bylo nutné přistoupit k náhradnímu řešení, konkrétně nahrazení řídicí jednotky Arduinem.

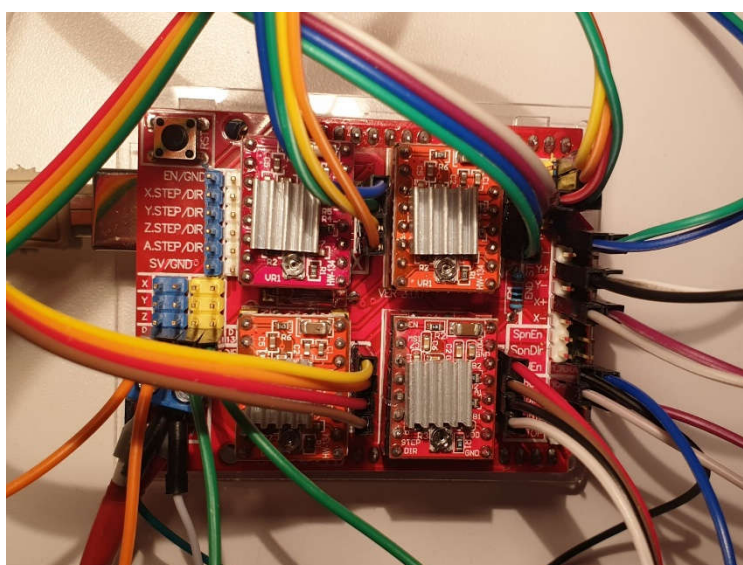
Obrázek 12: Nefunkční ovládací jednotka FOXEE



2.2 PŘIDÁNÍ CNC SHIELDU

Arduino má označené piny, což usnadňuje vytvoření kódu pro ovládání jednotlivých os. Nicméně Arduino nedokáže řídit motory plynulým pohybem, takže bylo nutné zakoupit CNC shield, který tuto funkci umožňuje. Pro řízení krokových motorů je třeba CNC shield kombinovat s driverem A4988. Po objednání potřebných součástek bylo nutné všechny části sestavit.

Obrázek 14: Plně osazený CNC shield zapojený v Arduino

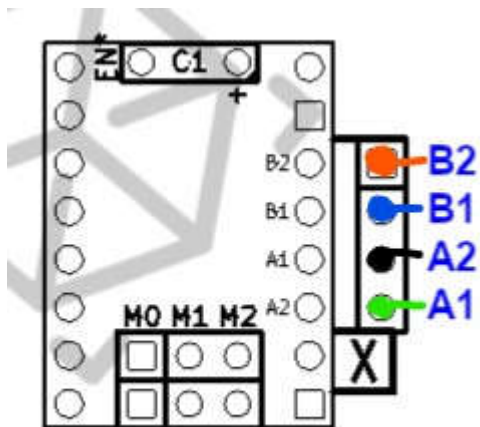


2.3 ZAPOJENÍ KROKOVÝCH MOTORŮ A KONCOVÝCH ČIDEL

K jednotlivým třem osám byly pomocí propojovacích kabelů připojeny krokové motory. Zpočátku bylo obtížné najít správné schéma zapojení kabelů. Po chvíli hledání na internetu se podařilo nalézt správné schéma, podle kterého byly motory zapojeny do CNC shieldu. Viz obrázek 15 a 16.

Poté byly jednotlivé osy uvedeny do pohybu, ale nebylo možné je koordinovaně řídit, protože nevěděly, kdy a kde se nacházejí. Proto bylo nezbytné připojit koncová čidla. Opět bylo nutné najít na internetu správné zapojení, které se podařilo nalézt. Zjistilo se, že CNC shield má piny přímo určené pro koncová čidla. Viz obrázek 17. Čidla umožnila dát pokyn jednotlivým osám, kam a o kolik se mají pohnout, což usnadnilo jejich naprogramování.

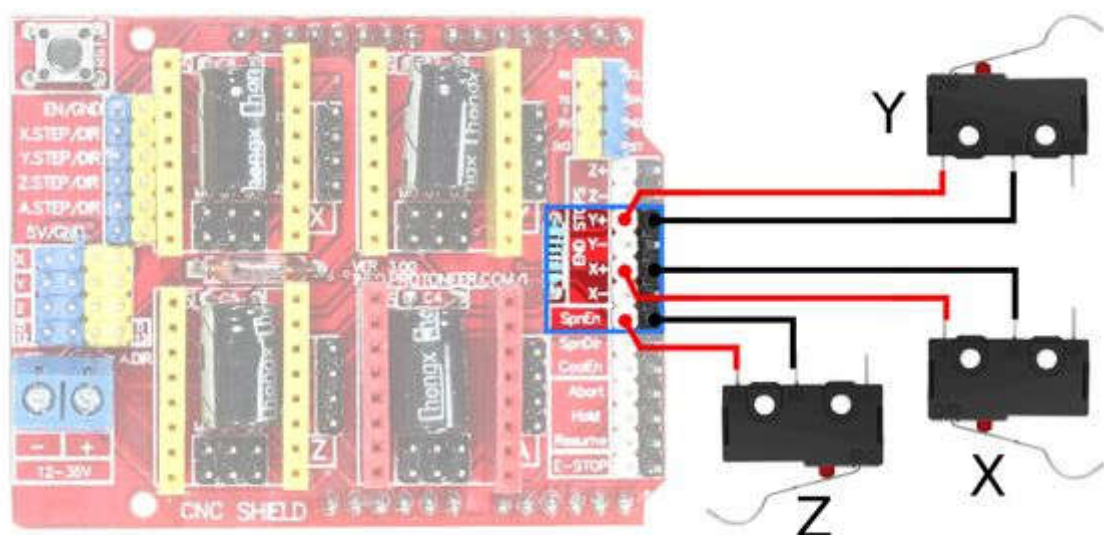
Obrázek 15: Zapojení motorů do CNC shieldu



Obrázek 16: Zapojení krokových motorů



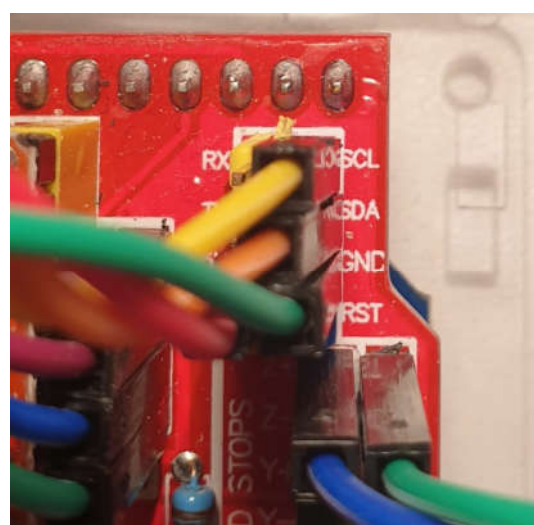
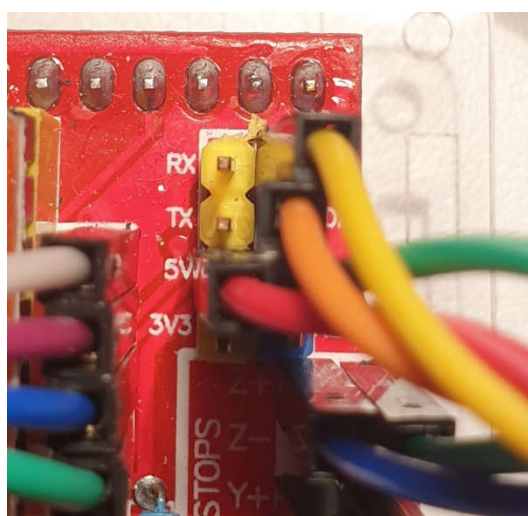
Obrázek 17: Zapojení mikropínačů



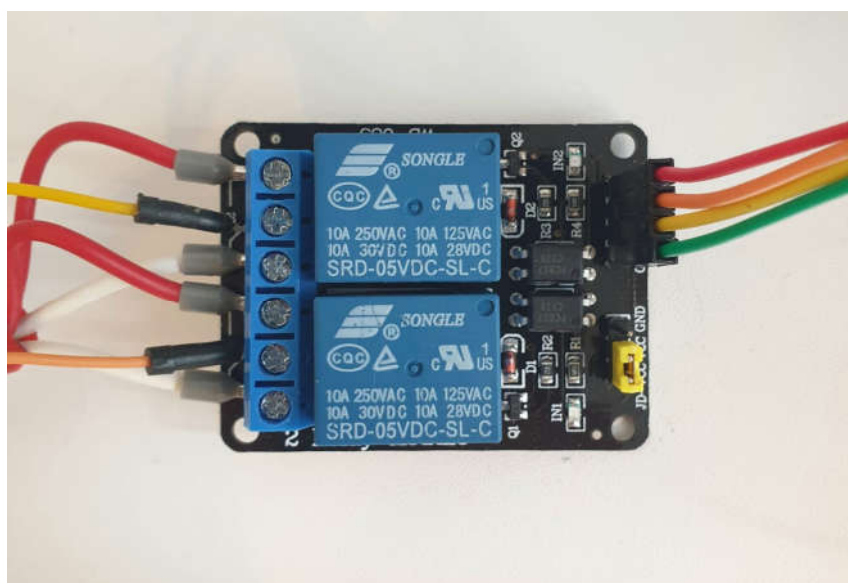
2.4 KLEŠTĚ

Po vyřešení pohybu ramena bylo třeba rozpohybovat kleště. Ty se nejjednodušeji ovládají přes relé, které umožňuje jejich otevření a zavření. Bylo potřeba dvoukanálové relé, do kterého byl přiveden proud (12V) ze zdroje. Relé bylo napojeno do CNC shieldu, což umožnilo řízení kleští přes kód.

Obrázek 18 a 19: Zapojení relé do CNC shieldu



Obrázek 9: 2-kanálové relé

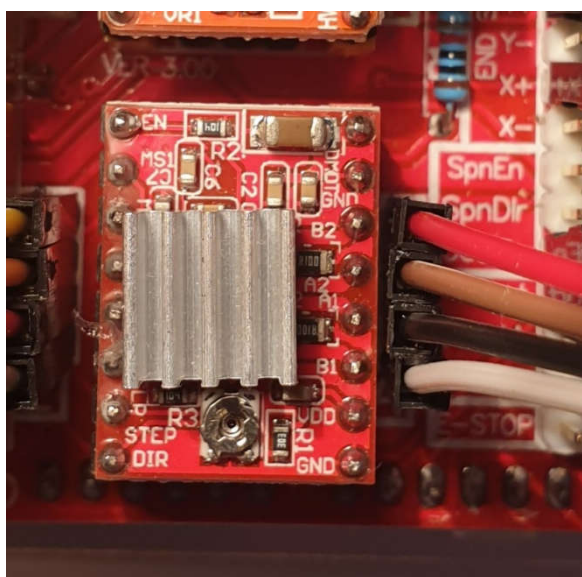


2.5 ČTVRTÁ OSA A

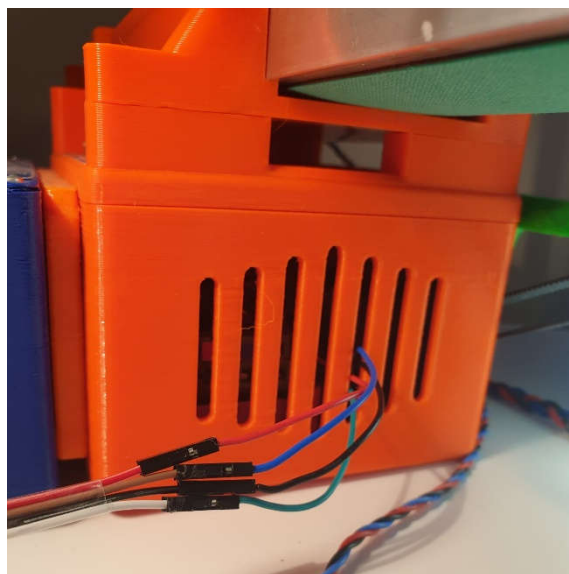
Dalším krokem bylo zprovoznění čtvrté osy, osy A, aby mohla být ovládána samostatně a nesloužila jako klonovací osa, museli se propojit dané piny. Viz obrázek 15.

Na internetu byl nalezen vzorový kód pro pohyb osy A, který byl upraven podle mých požadavků pro pás. Krokový motor pro ovládání pásu je stejně zapojený jako ostatní motory. Viz obrázek 12.

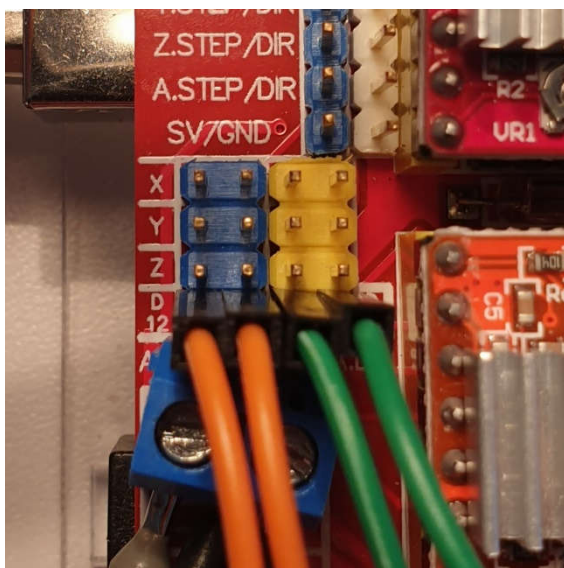
Obrázek 20: Připojení motoru od pásu do CNC shieldu



Obrázek 21: Připojení krokového motoru



Obrázek 22: Propojení pinu ke vznikutí osy A

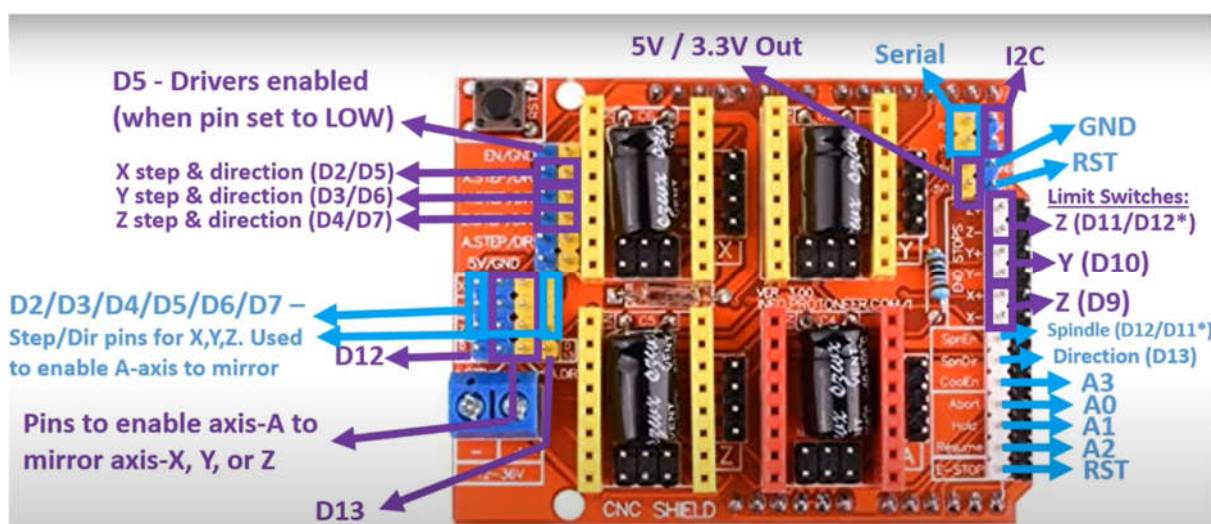


2.6 SENZORY

Posledním krokem bylo zprovoznění senzorů. Nastal problém, protože nebylo jasné, jak je senzor sestaven, což ztěžovalo jeho zapojení do CNC shieldu. Po důkladném proměření se podařilo zjistit konstrukci senzoru. Při pokusu o zjištění polarit byl jeden ze senzorů poškozen, ale následně opraven.

Zapojení senzoru je následující: modrý drátek je mínus (GND) a lze ho zapojit kamkoliv, kde je označení GND neboli zem (nachází se například vedle pinu A1, označený černou barvou; všechny tyto černé piny ve sloupci slouží jako GND). Červený drátek je plus neboli přívod (12V) a zapojuje se k přívodu. Černý drátek je signál a připojuje se tam, kde je označení A (0, 1, 2 nebo 3). Viz obrázek 8 a 16.

Obrázek 23: Onačení pinu na CNC shieldu



2.7 KÓD PRO ARDUINO

Po zprovoznění všech jednotlivých částí byly všechny kódy sloučeny do jednoho celku, čímž byl celý systém kompletně funkční.

Kód pro Arduino pro manipulátor:

```
// nastavení čísel projevacích pinů

#define ENABLE 8

#define X_STEP 2

#define Y_STEP 3

#define Z_STEP 4

#define X_DIR 5

#define Y_DIR 6

#define Z_DIR 7

#define RELAY_SEL A4

#define RELAY_SDA A5

#define CW 0

#define CCW 1

const int limitPinX = 9;

const int limitPinY = 10;

const int limitPinZ = 11;

const byte stepPin = 12; // x axis > 2, y axis > 3, z axis > 4 a axis > 12

const byte dirPin = 13; // x axis > 5, y axis > 6, z axis > 7 a axis > 13

const byte enablePin = 8; // enable on the CNC shield is held HIGH (disabled) by default

const byte sensorPin = A3; // Pin pro připojení čidla

const byte sensorPin2 = A0; // Pin pro připojení čidla
```

```
unsigned long stepTime = 2400;
```

```
// Definice pinů pro mikrospínač
```

```
const int mikrospinaX = limitPinX;
```

```
const int mikrospinaY = limitPinY;
```

```
const int mikrospinaZ = limitPinZ;
```

```
void setup() {
```

```
    // Nastavení pinů jako výstupní (dioda) a vstupní (mikrospínač)
```

```
    pinMode(mikrospinaX, INPUT_PULLUP); // Použití interního pull-up rezistoru  
    pro mikrospínač
```

```
    pinMode(mikrospinaY, INPUT_PULLUP);
```

```
    pinMode(mikrospinaZ, INPUT_PULLUP);
```

```
    Serial.begin(115200);
```

```
    pinMode(stepPin, OUTPUT);
```

```
    pinMode(dirPin, OUTPUT);
```

```
    pinMode(enablePin, OUTPUT);
```

```
    pinMode(sensorPin, INPUT); // Nastavení pinu čidla jako vstup
```

```
    digitalWrite(enablePin, LOW); // enable steppers
```

```
    digitalWrite(dirPin, CCW);
```

```
    // nastavení směrů pro všechny piny
```

```
    pinMode(X_STEP, OUTPUT);
```

```

pinMode(Y_STEP, OUTPUT);
pinMode(Z_STEP, OUTPUT);
pinMode(X_DIR, OUTPUT);
pinMode(Y_DIR, OUTPUT);
pinMode(Z_DIR, OUTPUT);
pinMode(ENABLE, OUTPUT);
pinMode(RELAY_SEL, OUTPUT);
pinMode(RELAY_SDA, OUTPUT);

// povolení řízení pro všechny drivery
digitalWrite(ENABLE, LOW);
}

void loop() {
  while(true){
    //pro položení předmětu v pravo
    centrALL(); //všechno se vicentruje
    kleste(true); //kleště se otevřou
    prijezdpasu(); //vicentruje se předmět na pásu
    VezmiPredmen(); //ruka se posune dolu a vezme
    kleste(false); //zavře kleště
    poloz(false); //posune se (true - pravá, false - levá)
    kleste(true); //otevře kleště

    //pro položení předmětu v levo
    centrALL();
  }
}

```

```

    kleste(true);
    prijezdpasu();
    VezmiPredmen();
    kleste(false);
    poloz(true);
    kleste(true);
}

}

void prijezdpasu(){
    //hlídá jestli se jeden senzor spustil jinak jede doleva pásem
    while(!digitalRead(sensorPin) == HIGH && !digitalRead(sensorPin2) == HIGH){
        oneStep(true);
    }

    //potom kontroluje zda je senzor blíží k pásu aktivní pokud ne tak pás otočí
    směr jízdy
    if(!digitalRead(sensorPin) == HIGH){
        while(!digitalRead(sensorPin) == HIGH){
            oneStep(false);
        }
    }
    else{
        for (int i = 0; i < 10000; i++) {
            oneStep(true);
        }
    }
}

```

```

// pototm se vycentruje
for (int i = 0; i < 30000; i++) {
    oneStep(false);
}
for (int i = 0; i < 30000; i++) {
    oneStep(false);
}
}

void pohybOsy(byte dirPin, byte stepPin, int kroky, int speed, bool direction) {
    // Nastavení směru na opačný
    digitalWrite(dirPin, direction);
    delay(50);
    // smyčka pro provedení předaného množství kroků
    for (int i = 0; i < kroky; i++) {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(speed);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(speed);
    }
}

void VezmiPredmen(){
    //několik pohybů v různých směrech aby se přesunul z vycentrované polohy
    dolů k předmětu

    pohybOsy(Y_DIR, Y_STEP, 100, 3000, false);
    pohybOsy(Z_DIR, Z_STEP, 100, 3000, false);
    pohybOsy(Y_DIR, Y_STEP, 100, 3000, false);
}

```

```
pohybOsy(Z_DIR, Z_STEP, 100, 3000, false);  
pohybOsy(Y_DIR, Y_STEP, 50, 3000, false);  
  
}
```

```
void kleste(bool otevrit){  
    //pokud je otevři true tak se kleště otevřou  
    if(otevrit){  
        digitalWrite(RELAY_SEL, HIGH);  
        delay(1000);  
        digitalWrite(RELAY_SEL, LOW);  
        delay(1000);  
    }  
    //pokud jinak tak se zavře  
    else{  
        digitalWrite(RELAY_SDA, HIGH);  
        delay(1000);  
        digitalWrite(RELAY_SDA, LOW);  
        delay(1000);  
    }  
}
```

```

void centrYZ(){
    //vycentruje osy Y a Z protože je musíme dělat najednou
    while(true){
        if (digitalRead(mikrospinacY) && digitalRead(mikrospinacZ)) {
            return;
        }
        else {
            //pohybuje osu dokud není aktivní senzor Y
            if(!digitalRead(mikrospinacY)){
                pohybOsy(Y_DIR, Y_STEP, 100, 3000, true);
            }
            //pohybuje osu dokud není aktivní senzor Z
            if(!digitalRead(mikrospinacZ)){
                pohybOsy(Z_DIR, Z_STEP, 100, 3000, true);
            }
        }
    }
}

```

```

void centrX(){
    //vycentruje osu X
    while(true){
        //pohybuje osu dokud není senzor X aktivní
        if (digitalRead(mikrospinacX)) {
            //pokud je aktivní tak ho to posune o 175 kroku což ho přibližně vycentruje
            na střed
            pohybOsy(X_DIR, X_STEP, 175, 3000, false);
        }
    }
}

```

```

    return;
}
else {
    pohybOsy(X_DIR, X_STEP, 20, 3000, true);
}
}
}

void centrALL(){
    //tady se pak spojují všechny centrovací funkce
    centrYZ();
    centrX();
    kleste(false); //součástí cetrování je i že se kleště otevřou
}

void poloz(bool left){
    //tahle funkce z bodu kdy drží předmět tak se přesune doprava nebo doleva
    podle logické proměné left
    if(left){
        //pokud proměná left je true tak se spustí řada příkazů které to přesunou
        doleva
        pohybOsy(Z_DIR, Z_STEP, 100, 3000, true);
        pohybOsy(X_DIR, X_STEP, 50, 3000, true);
        pohybOsy(Y_DIR, Y_STEP, 50, 3000, true);
        pohybOsy(Z_DIR, Z_STEP, 70, 3000, false);
        pohybOsy(Y_DIR, Y_STEP, 70, 3000, false);
    }
    else{

```

```
//pokud proměná left je false tak se spustí řada příkazů které to přesunou  
doprava
```

```
  pohybOsy(Z_DIR, Z_STEP, 80, 3000, true);  
  pohybOsy(X_DIR, X_STEP, 80, 3000, false);  
  pohybOsy(Y_DIR, Y_STEP, 80, 3000, false);  
}  
}
```

```
void oneStep(bool left)
```

```
//tahle funkce hýbe pásem
```

```
{  
  if(left){  
    digitalWrite(dirPin, LOW); //pokud je left true tak směr pásu bude doleva  
  }  
  else{  
    digitalWrite(dirPin, HIGH); //pokud je left false tak směr pásu bude doprava  
  }  
  static unsigned long timer = 0;  
  unsigned long interval = stepTime;  
  if (micros() - timer >= interval)  
  {  
    timer = micros();  
    digitalWrite(stepPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(stepPin, LOW);  
  }  
}
```

3 ZÁVĚR

Když jsem poprvé dostal manipulátor do ruky a bylo mi řečeno, že to bude moje ročníková práce, nebyl jsem z toho nijak nadšený, protože jsem nevěděl, co s tím mám dělat. Jelikož to není známý a prodáváný produkt, bylo obtížné najít jakékoliv informace, které by mi pomohly.

Když jsem zjistil, že hlavní jednotka FOXEE není funkční, docela se mi ulevilo, protože jsem nevěděl nic o jejím provozu. Zjistění, že můžu použít Arduino, mě potěšilo, protože programování pomocí Arduina, jak jsem zjistil, není tak složité.

Po přidání CNC shieldu se všechno rázem zkomplikovalo. Nutnost zapojit mnoho součástek z manipulátoru do CNC shieldu mi práci značně ztížila, a hledání informací o tom, jaký pin má jaké číslo, mi zabralo spoustu času. Jak se ale nakonec ukázalo, stálo to za to.

Rozpohybování samotného ramene mi zabralo asi nejvíce času a bylo nejsložitější, ale jakmile jsem překonal tuto překážku, viděl jsem velký pokrok. Do té doby jsem nevěděl, jestli má vůbec cenu se snažit, protože jsem se na dlouhou dobu zasekl na začátku s pohybem ramene.

Jakmile jsem překonal tento krizový bod, začalo mě věnování se manipulátoru bavit. Postupně jsem rozpohyboval zbytek a viděl postupné pokroky. Kdybych to měl shrnout, musím říct, že na to, že jsem na začátku vůbec nevěděl, co s tím, se mi to nakonec hodně povedlo a naučil jsem se spoustu nových věcí. Naučil jsem se porozumět Arduinu a CNC shieldu, stejně jako pohybu a řízení manipulátoru, což jsou hlavní části včetně krokových motorů.

4 POUŽITÁ LITERATURA

<https://navody.drateg.cz/technikuv-blog/napajeni-arduina.html>

<https://www.zyltech.com/arduino-cnc-shield-instructions/>

<https://forum.arduino.cc/t/arduino-uno-cnc-shield-controlling-4-stepper-motors/913742/3>

<https://forum.arduino.cc/t/arduino-uno-cnc-shield-controlling-4-stepper-motors/913742>

<https://domoticx.com/arduino-shield-cnc-shield/>

<http://www.zyltech.com/arduino-cnc-shield-instructions/>

<https://navody.drateg.cz/navody-k-produktum/arduino-cnc-shield-driver-a4988-motor-28byj-48.html>

<https://drateg.cz/arduino/1132-cnc-shield-pro-3d-tiskarny-pro-arduino.html>

<https://www.laskakit.cz/arduino-uno-r3--atmega328p--precizni-klon/>

<https://www.laskakit.cz/2-kanaly-rele-modul--5vdc-250vac-10a/>

<https://www.youtube.com/watch?v=JlhjcTh4yts>

<https://www.youtube.com/watch?v=zUb8tiFCwmk&t=216s>

5 ZDROJE K OBRÁZKŮM

Obrázek 0: Vlastní

Obrázek 1: <https://navody.drateg.cz/technikuv-blog/napajeni-arduina.html>

Obrázek 2: Vlastní

Obrázek 3: Vlastní

Obrázek 4: Vlastní

Obrázek 5: Vlastní

Obrázek 6: Vlastní

Obrázek 7: Vlastní

Obrázek 8: Vlastní

Obrázek 9: Vlastní

Obrázek 10: <https://www.zyltech.com/arduino-cnc-shield-instructions/>

Obrázek 11: <https://www.amazon.co.uk/Engraving-Machine-Printer-Expansion-arduino/dp/B0BNXLFP?th=1>

Obrázek 12: Vlastní

Obrázek 13: Vlastní

Obrázek 14: Vlastní

Obrázek 15: Vlastní

Obrázek 16: Vlastní

Obrázek 17: https://sixselov.live/product_details/60042645.html

Obrázek 18: Vlastní

Obrázek 19: Vlastní

Obrázek 20: Vlastní

Obrázek 21: Vlastní

Obrázek 22: Vlastní

Obrázek 23: <https://www.youtube.com/watch?v=JlhjcTh4yts&t=329s>