



Středoškolská technika 2022

Setkání a prezentace prací středoškolských studentů na ČVUT

Robotický manipulátor

Pavel Pewner

SPŠ a VOŠ Písek, Karla Čapka 402, 397 11 Písek

Anotace

Práce se zabývá návrhem, strojních konstrukcí, způsoby řízení a zpracováním vstupních informací u robotického manipulátoru. V práci jsou popsány postupy návrhu a výroby mechanické a elektrické části zařízení, postup výroby desky plošných spojů za použití CNC stroje a principy fungování řídicího programu, který zahrnuje i zpracování obrazu z kamer. Dále je zde popsán jednoduchý ukázkový program, který demonstruje autonomní práci manipulátoru při provádění naprogramované úlohy.

Klíčová slova:

Manipulátor, lineární pohyb, Arduino, Raspberry, krokový motor, joystick, řízení, 3D tisk

Annotation

Graduation thesis is focused on machine design, control methods and processing of input informations of the robotic manipulator. In thesis are described procedures of design and production of mechanical and electrical parts, production process of PCB with using the CNC machine and main program principles, which also includes processing of images from cameras. Further is described simple exemplary program, which shows autonomy work of the manipulator control by programmed task.

Key words:

Manipulator, linear movement, Arduino, Raspberry, stepper motor, joystick, 3D print

Obsah

| | |
|--|----|
| Obecný úvod do tématu..... | 5 |
| Popis projektu | 7 |
| Hardware..... | 10 |
| Rameno manipulátoru | 10 |
| Mechanická část..... | 10 |
| Výroba ramene manipulátoru..... | 10 |
| 3D tisk konstrukce..... | 11 |
| výroba mechanické části a částečná kompletace | 12 |
| Popis lineárních pojezdů | 24 |
| Elektronická výbava a kabelový rozvod..... | 27 |
| Kabely a vodiče..... | 27 |
| Krabice s řídicí elektronikou | 28 |
| Napájení | 28 |
| Základní deska | 29 |
| Výroba základní desky | 31 |
| Porty a moduly osazené na základní desce..... | 36 |
| Drivery | 38 |
| Zbytek elektroniky v krabici | 40 |
| Ovládací panel | 42 |
| Joystick | 44 |
| Způsoby řízení pohybu ramene manipulátoru..... | 46 |
| Ukázkový program..... | 47 |
| Software | 50 |
| Arduino software | 50 |
| Řízení krokových motorů | 52 |
| Homing – parkování do nulového bodu..... | 56 |
| Komunikace s dálkovými ovladači..... | 56 |
| Komunikace přes sériový port z Arduina do ovládacího programu na počítači..... | 57 |
| Vývojové prostředí Processing | 58 |
| Vzhled a ovládací prvky..... | 59 |
| Komunikace přes sériový port z ovládacího programu do Arduina. | 61 |
| Funkce pro rozpoznávání barev pomocí kamery..... | 63 |
| Software joysticku..... | 65 |

| | |
|-----------------------|----|
| Závěr..... | 66 |
| Citované zdroje | 68 |
| Seznam obrázků | 69 |
| Seznam tabulek..... | 70 |
| Přílohy | 71 |

Obecný úvod do tématu

Robot je stroj pracující s určitou mírou samostatnosti, vykonávající určené úkoly, a to předepsaným způsobem a při různých mírách potřeby interakce s okolním světem a se zadavatelem: Robot je schopen své okolí vnímat pomocí senzorů, reagovat na něj, zasahovat do něj, případně si o něm vytvářet vlastní představu, model. Vnímáním světa nejenže může poznávat svět samotný, ale může také vyhodnocovat svůj vliv na něj a využívat tak zpětnou vazbu. Robot je fyzickou realizací obecnějšího pojmu agent.

Dělení robotů:

Podle generace na:

- roboty 1. generace – pracují na základě pevného programu
- roboty 2. generace – vybavené senzory a čidly, díky nimž reagují na okolní podmínky

Podle jejich schopnosti přemísťovat se na:

- stacionární – nemohou se pohybovat z místa na místo (například průmyslové manipulátory)
- mobilní – mohou se přemísťovat (například vesmírné sondy a vozítka na Marsu)

Dále také podle:

- pohybových možností,
- autonomie,
- účelu (boj, výroba, tiskárny a plotry, přeprava, průzkum),
- způsobu programování, a i jinak.

Podle účelu, vzhledu, způsobu vzniku, schopností a dalších aspektů rozlišujeme tyto roboty:

- **Manipulátor** – stroj nemající vlastní inteligenci. Je ovládán na dálku.
- **Kuchyňský robot** – kombinace mixéru, hnětače a dalších kuchyňských strojů, obvykle provedený jako motorová jednotka s nastavci
- **Android** – robot podobný člověku – obvykle se očekává biologické složení. Roboti v R.U.R. byli podle tohoto dělení androidi.
 - **Droid** – jakýkoliv inteligentní a samočinný robot, jeho typem je i dron, tedy droid pracující jako dělník.
 - **Humanoid** – robot podobný člověku principiální stavbou těla a zejména způsobem pohybu.
 - **Anthropomorfní** – stroj, který se člověku přibližuje (napodobuje ho) buď fyzicky, způsobem pohybu, nebo naopak mentálně (např. HAL 9000).
- **Kyborg (kybernetický organismus)** – umělá bytost či mysl, biologické, přírodní tělo plně pod vládou stroje, skrze nějaké bio-kybernetické

propojení. Naproti tomu opačný pól je živá, přírodní bytost či mysl s uměle upraveným tělem, např. obohacným o mechanické či elektronické součástky, ze kterého v extrémním případě mohl zůstat i jen mozek (viz bionika), ale stále považovaná za člověka. [1]

Popis projektu

Mnou navržený robotický manipulátor je zařízení určené k přemísťování předmětů po pracovní ploše. Rameno robotického manipulátoru je umístěno na desce stolu, která slouží také jako manipulační prostor. Manipulátor má tři osy pohybu a čtvrtou osu sloužící jako čelist pro uchopení předmětů. Manipulační plocha má rozměry 400x400 mm. Jednotlivé osy se pohybují pomocí lineárních šroubových pojezdů, jejichž pohyb zajišťují krokové motory. Vodícím prvkem jsou ocelové vodící tyče. Celá konstrukce byla vyrobena pomocí technologie 3D tisku.

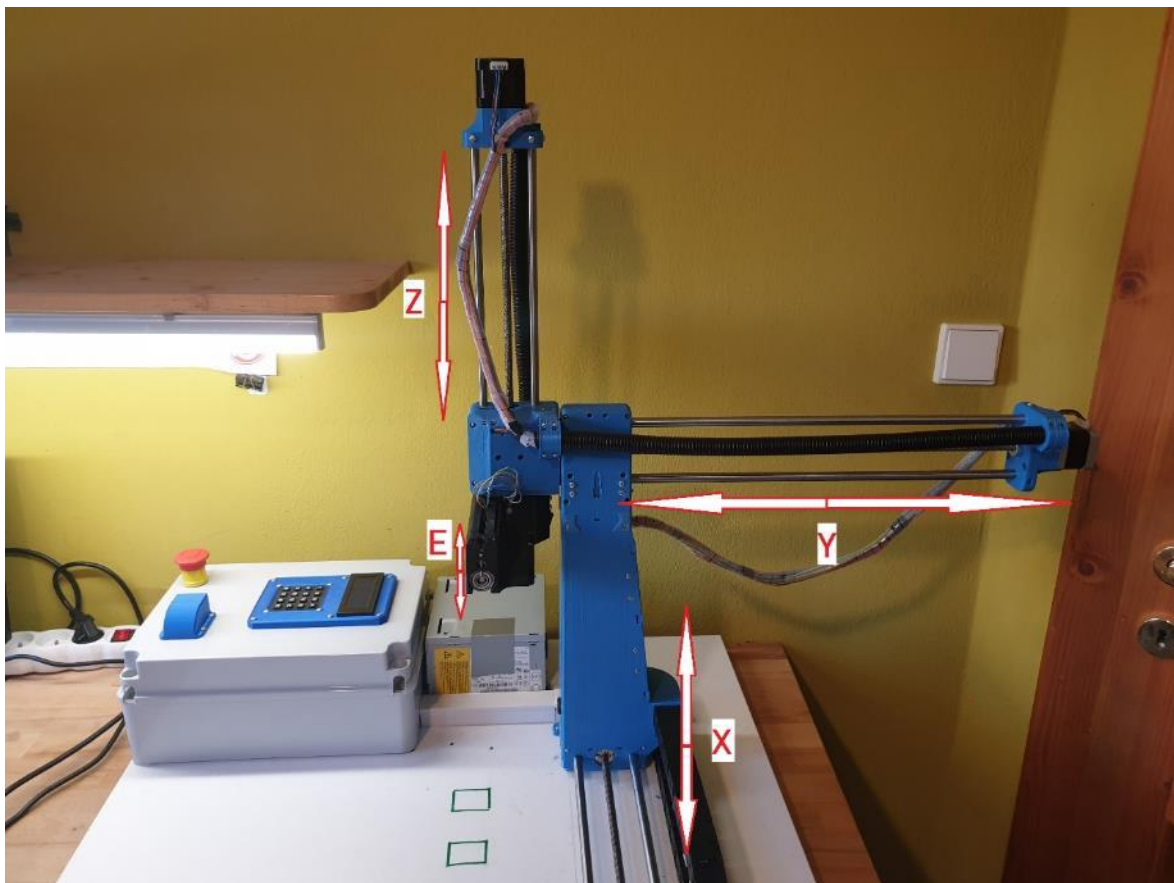


Obrázek 1: současný stav projektu, pohled na celek

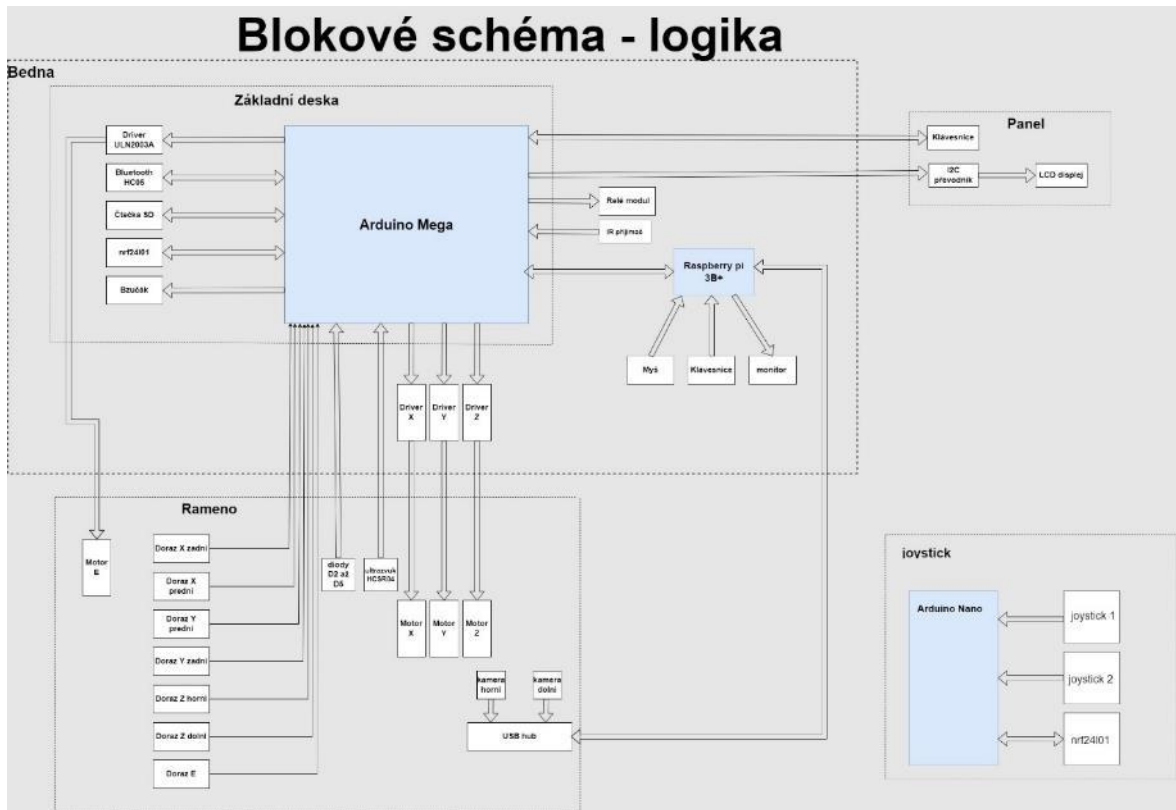
Manipulátor má několik vstupních periférií, které slouží k autonomnímu řešení zadané úlohy. Těmito perifériemi jsou dvě kamery, jedna umístěná u čelisti a druhá v základně (středu) robotického manipulátoru. Dalšími vstupními prvky jsou ultrazvukový měřič vzdálenosti a úchopový senzor v čelisti.

Řídící elektronika je umístěna v instalační krabici, která se nachází na podkladové desce. Ve víku této krabice jsou umístěny ovládací prvky pro operátora. Uvnitř krabice je umístěna základní deska a další řídicí elektronika. Celé zařízení je řízeno pomocí Arduina. Počítačový program pro řízení pohybu a zpracování obrazu z kamer běží na počítači Raspberry pi. Napájení celého zařízení je zajištěno počítačovým zdrojem, který je připevněn mezi manipulátorem a krabicí s elektronikou. Signálové a napájecí kabely z řídicí elektroniky do manipulátoru jsou vedeny instalačními lištami a kabelovým řetězem, který ústí do středového prvku manipulátoru.

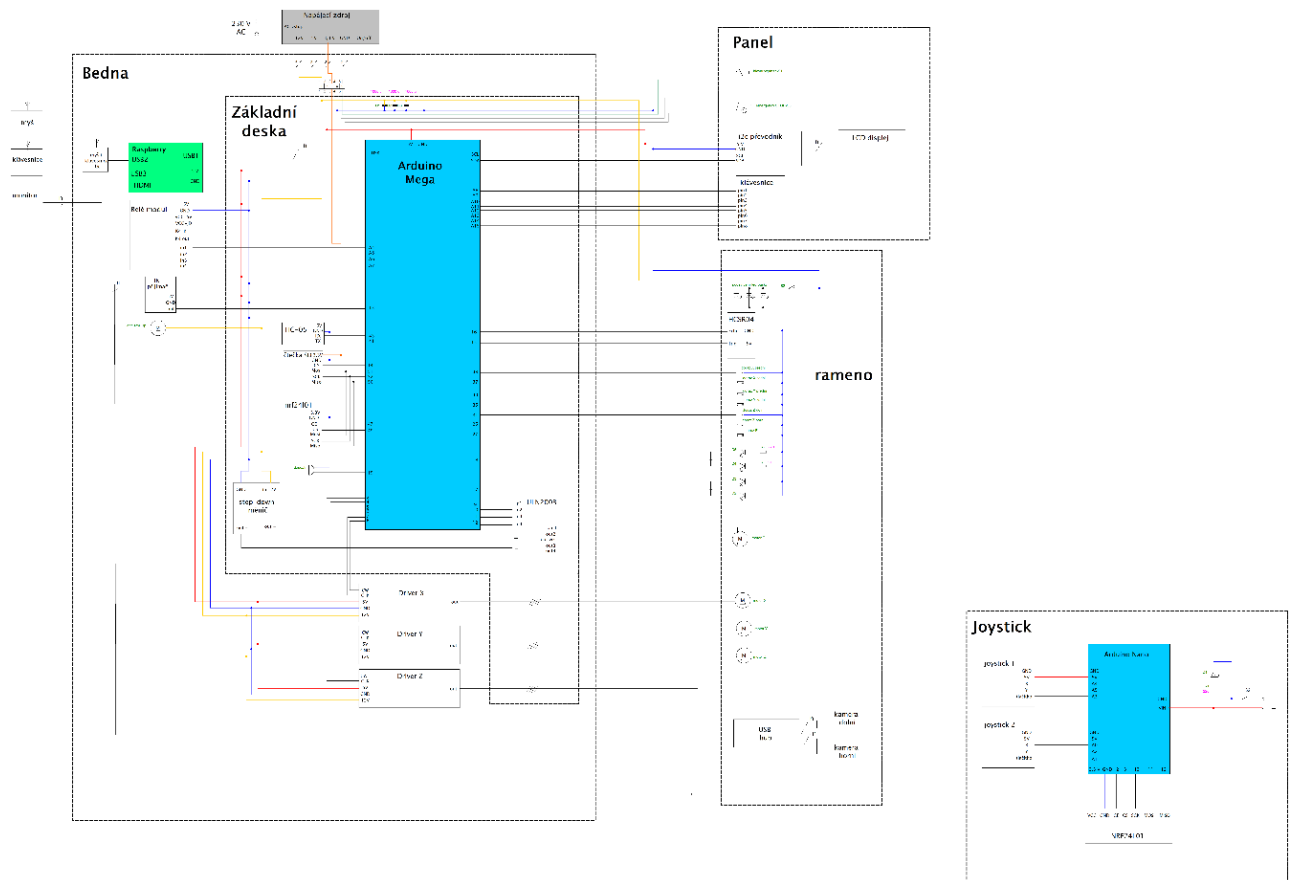
Manipulátor je možné ovládat počítačovým programem, dálkovým ovladačem a nebo bezdrátovým joystickem, jehož konstrukce je součástí projektu. Dále je možné spustit předepsaný program uložený v paměti. Tyto funkce jsou voleny přes klávesnici na ovládacím panelu zařízení.



Obrázek 2: popis os pohybu ramene manipulátoru



Obrázek 3: blokové schéma logické části řízení



Obrázek 4: celkové elektrické schéma

Hardware

Rameno manipulátoru

Mechanická část

Výroba ramene manipulátoru

Výroba mechanických ramene manipulátoru byla velice složitá a časově náročná. Dala by se rozdělit na 3 hlavní fáze: 3D tisk + návrh mechanické části a částečná kompletace, celková kompletace, kabeláž a elektronika.

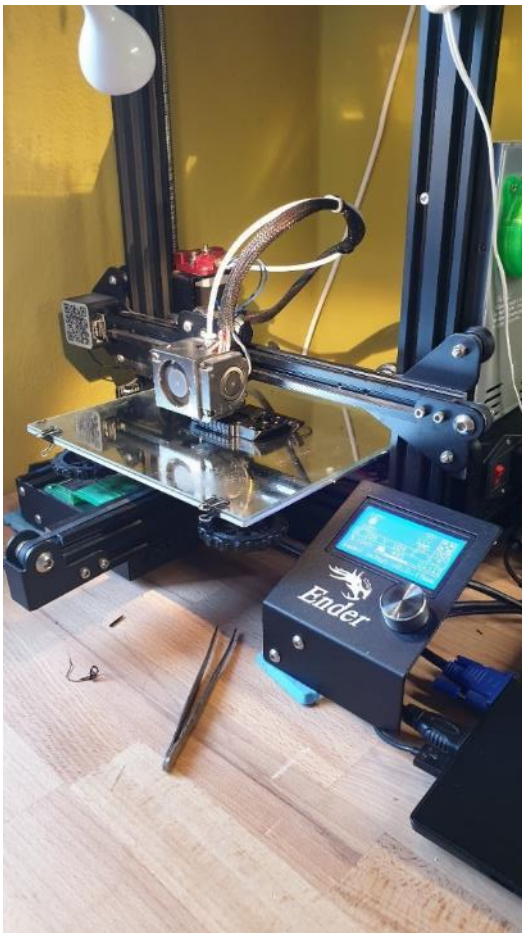
Nejprve jsem navrhnul a vytisknul všechny mechanické díly, poté jsem zkompletoval veškeré tyto části a vložil do nich elektrické součástky a propojovací kabelové svazky, které bylo potřeba přidělat už během kompletace mechanických dílů. Po kompletaci jsem přidělal zbytek elektronických součástek a natahal kabeláž.

3D tisk konstrukce

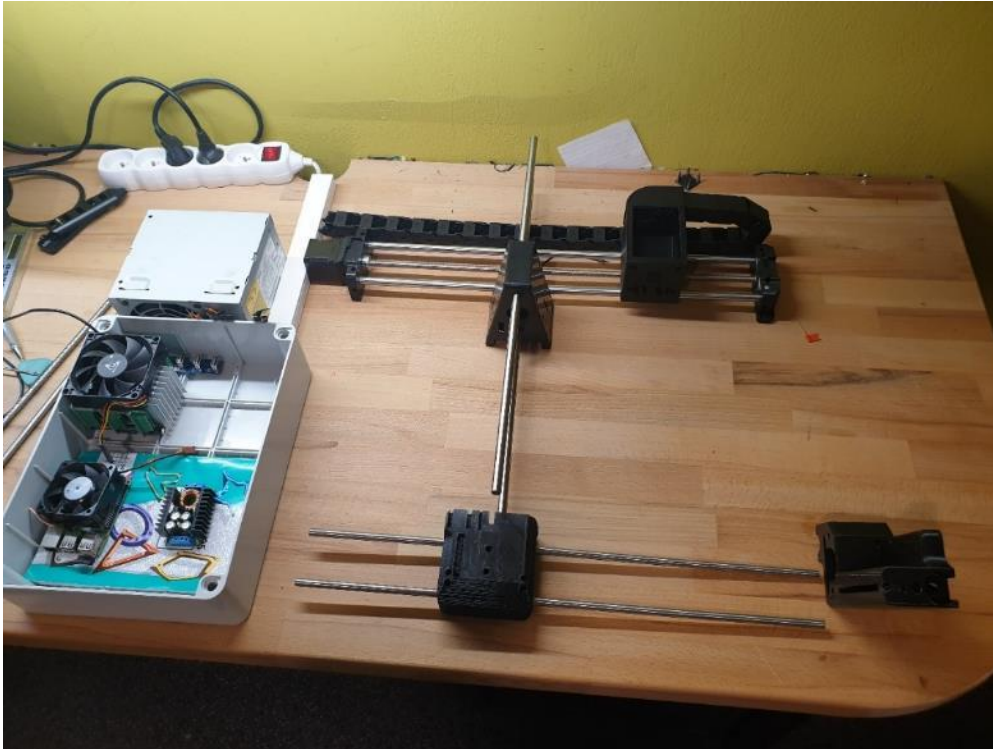
Rameno manipulátoru je složeno z celkem sedmnácti plastových dílů. Všechny plastové díly jsem zhotovil pomocí metody FDM 3D tisku. Vždy jsem nejprve vytisknul prototyp z PLA a malou výplň a když jsem si byl jistý, že součástka pasuje tak, jak potřebuji vytiskl jsem ji „naostro“ z PETG materiálu. Všechny díly jsem kreslil v prostředí Autodesk Inventor

Všechny díly jsem tisknul na 3D tiskárna CREALITY Ender 3 a Flashforge Creator 3. Při tisku na Ender 3 jsem G-kód generoval v softwaru Prusa Slicer a při tisku na Creator 3 jsem G-kód generoval v softwaru Flash Print.

Po vytisknutí každého dílu byl nutný post processing. Většinou bylo potřeba vrtákem vyčistit otvory pro šrouby a tyče, odstranit ostré hrany a zabrousit různé výčnělky. U některých dílů bylo potřeba zarovnat brusným papírem nerovný povrch.

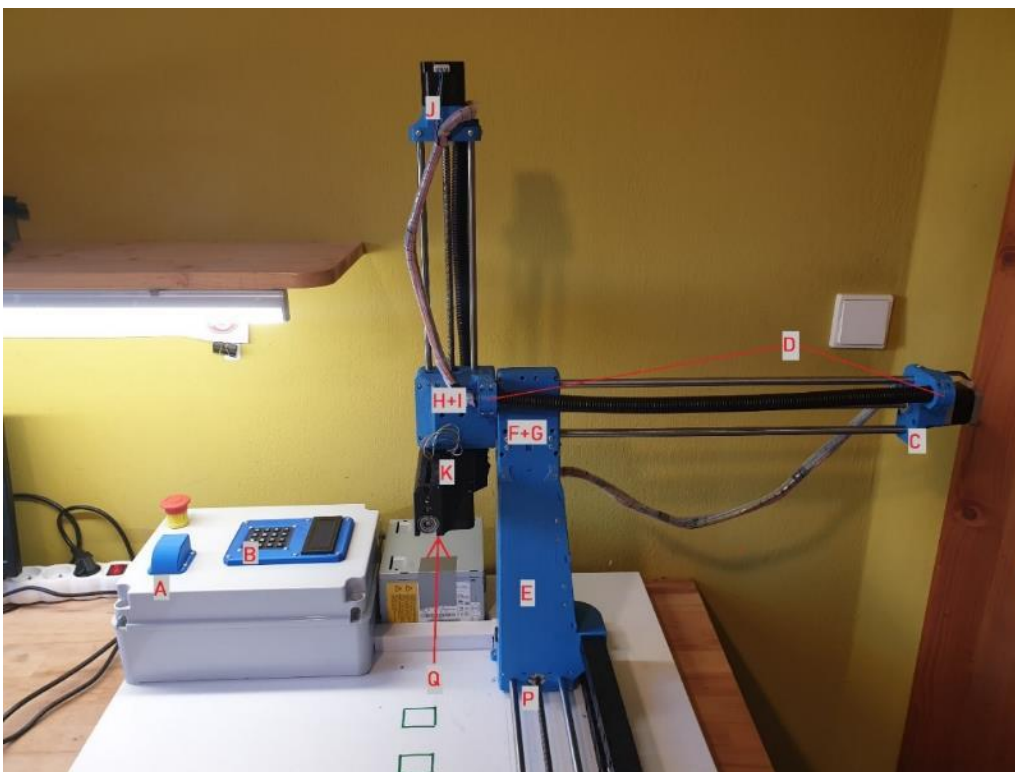


Obrázek 5: tisk prototypu na tiskárně Ender 3



Obrázek 6: skládání prototypů jednotlivých dílů

výroba mechanické části a částečná kompletace



Obrázek 7: popis tištěných dílů - pohled z boku



Obrázek 8: popis tištěných dílů - pohled zezadu

Tabulka 1: vysvětlivky popisu tištěných dílů

| index | Název dílu |
|-------|----------------------------------|
| A | Pouzdro hlavního vypínače |
| B | Rámeček klávesnice a displeje |
| C | Držák motoru osy Y |
| D | Třmen kabelové chráničky |
| E | Střed / středová základna |
| F | Pouzdro ložisek osy Y |
| G | Pouzdro ložisek osy Y - protikus |
| H | Pouzdro ložisek osy Z |
| I | Pouzdro ložisek osy Z - protikus |
| J | Držák motoru osy E |
| K | Hlava |
| L | Základna – osa X |
| M | Základna + držák motoru osy X |
| N | Úchyt řetězu |
| O | Krytka středu |
| P | Střed-protikus |
| Q | Čelist protikus |

Sestavování manipulátoru jsem začal právě mechanickou částí. Postupoval jsem od nejspodnějších dílů a skončil jsem u čelisti. Postupně jsem tisknul jednotlivé díly a zkoušel jsem, jak do nich zapadají potřebné součásti, spojovací a mechanické díly.

Jako spojovací materiál byly použity šrouby M3 o různých délkách. Matice jsem často zajišťoval proti vypadnutí lepidlem.

Úplně nejdříve jsem vytisknul díl, ve kterém je upevněn krokový motor a vodící tyče osy X, základnu osy Y. Následně jsem vytisknul druhý díl, který drží druhou stranu vodících tyčí a končí v něm trapézový šroub, který je uložen v ložisku s přírubou. Poté jsem řešil, jak zakomponovat do obou těchto dílů koncové dorazy tak, aby byly uvnitř součástky, daly se snadno vyjmout a nezabíraly moc místa.



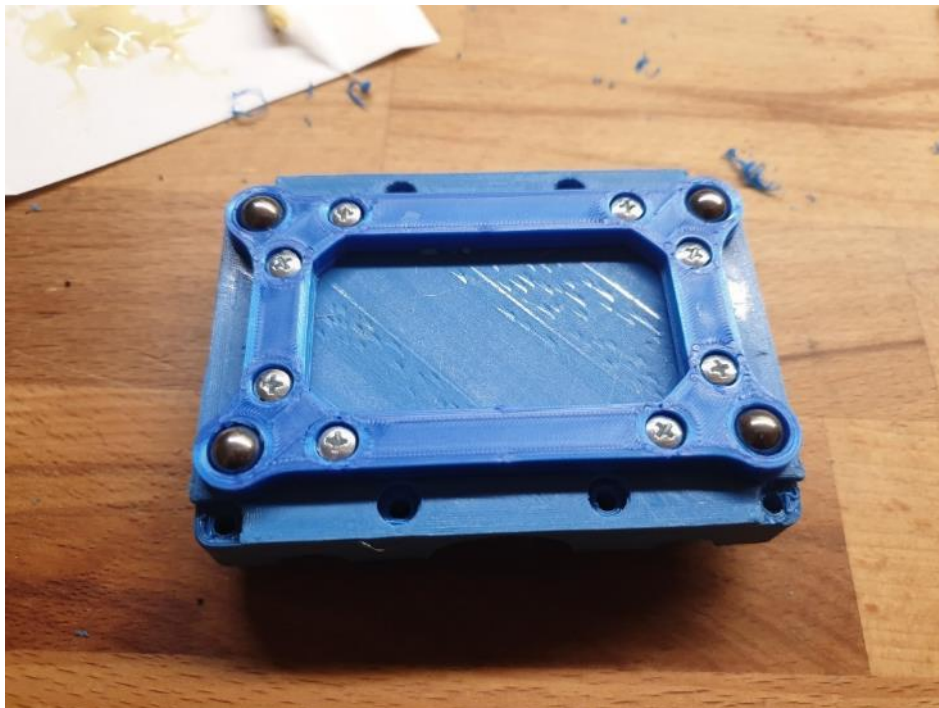
Obrázek 9: první prototyp základny osy X

Vodící tyče jsou zapuštěny 10mm do hloubky a každá tyč je přitažena dvěma šrouby. Šrouby jsou utažené do matic umístěných v šestiúhelníkovém otvoru a dotýkají se vodících tyčí.

Jako poslední jsem udělal „podpatek“, kterým jsou tyto dva díly přišroubované k podložce. Musel jsem pamatovat, že to jsou jediné dvě spojnice celého ramene s podložkou, tudíž musejí být pevné a vydržet veliký tlak.

Začal jsem navrhovat díl, který slouží jako protikus ke středové základně a jsou v něm uloženy lineární ložiska. Pro zachování tuhosti konstrukce je nutné, aby ložiska přesně pasovala do dílu a byla v konstrukci pevně sevřená. Mezi ložisky jsou umístěny dvě trapézové matice. Metodou pokus-omyl jsem se snažil optimalizovat jejich vzdálenost a úhel natočení pro co nejlepší chod šroubu v maticích. To se mi ovšem moc nepovedlo a ve finále celé stavby jsem to musel předělávat. Na spodek protikusu středové základny, nebo-li pouzdra pro ložiska osy X, jsem přidělal rámeček s vloženými

ocelovými kuličkami. Tento díl slouží jako podvozek minimalizující nežádoucí vychylování ramene manipulátoru na prohýbajících se vodících tyčích osy X.



Obrázek 10: pouzdro ložisek osy X s podvozkem

Následně jsem začal s navrhováním několika součástek najednou. Ale půjdu popořadě.

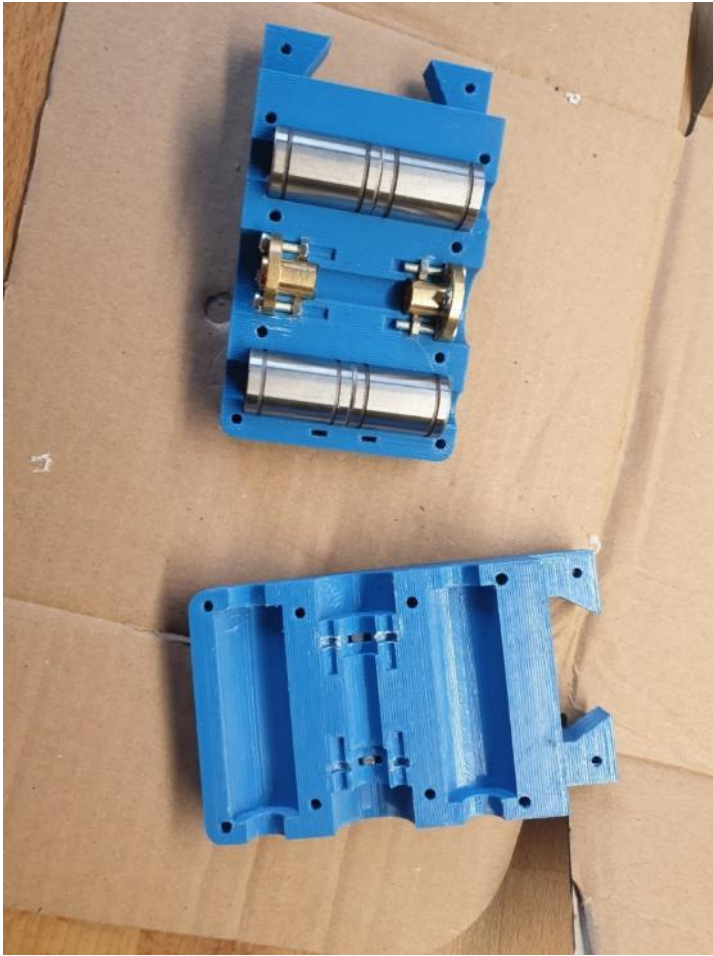
Jedním z nejtěžších dílů byl středový díl. Napojuje se do něho hned několik tištěných dílů, několik mechanických součástek a několik elektronických součástek. Jeho konstrukci jsem musel navrhnout tak, aby umožňovat co nejpohodlnější montáž a přístup ke všem součástkám a zároveň velikou tuhost, jelikož na něho působí velická páka a otřesy. Uvnitř je uložena elektronika a kabeláž, která se větví do dalších částí konstrukce. Myslet jsem musel také na upevnění vnitřní elektroniku a kabeláž vedoucí dovnitř.

Nejdříve jsem navrhnul prostor pro uložení ložisek a matic pro lineární pohyb osy X. V této části je ke středu několika šrouby přitažen protikus svírající mechanické součástky. Poté jsem nahrubo udělal celkový tvar středového dílu. Pokračoval jsem vstupem pro kabelový řetěz. Kabelový řetěz nebylo lehké zakomponovat, jelikož jsem nevěděl, jak ho do zařízení upevnit. Nakonec jsem zvolil jeho prosté zasunutí a za aretování dvěma šrouby, které procházejí výřezem kabelového řetězu a brání v jeho vysunutí.

Na středový díl jsou ve vrchní části připevněny pouzdra pro ložiska a matice osy Y. Jako jednou z počátečních částí návrhu bylo to, že jsem vymyslel zámkový systém, do kterého se tyto dva díly zasunou a sešroubováním k sobě utáhnou.

Dalším krokem bylo vytvoření dutého prostoru pro uložení elektroniky a kabeláže. Prostor jsem navrhnul tak veliký, abych do něho mohl co nejlépe uložit elektroniku a měl co nejlepší přístup pro montáž dílů uložených ve středu. Poté jsem nakreslil montážní otvory pro ultrazvukový měřič vzdálenosti HCSR04 a dolní kameru. Jako poslední jsem na středovém dílu navrhnul prvky pro přidělení zadního krytu, který je držen osmi šrouby a je pomocí zubových výběžků zakousnut do středu. Kryt středu nese mé jméno a jméno mé školy.

Jako další díly jsem navrhnul pouzdra ložisek a matic pro osu Y. V nich jsou sevřeny ložiska a matice pro osu Y. Zasazení těchto součástí jsem realizoval stejně, jako v případě osy X. V jednom z těchto dvou tištěných dílů jsou namontovány koncové dorazy pro osu Y. Jejich zakomponování pro mě nebylo složité, jelikož je v tomto díle poměrně dost prostoru. Jejich vodiče jsou vyvedeny otvorem rovnou do vnitřku středového dílu. Na špičkách těchto dílů jsem původně zamýšlel udělat nějaké vedení pro kabelovou chráničku s kabeláží, od toho jsem nakonec opustil. Ale ve 3D modelu mi zůstaly dva nevyužité otvory.



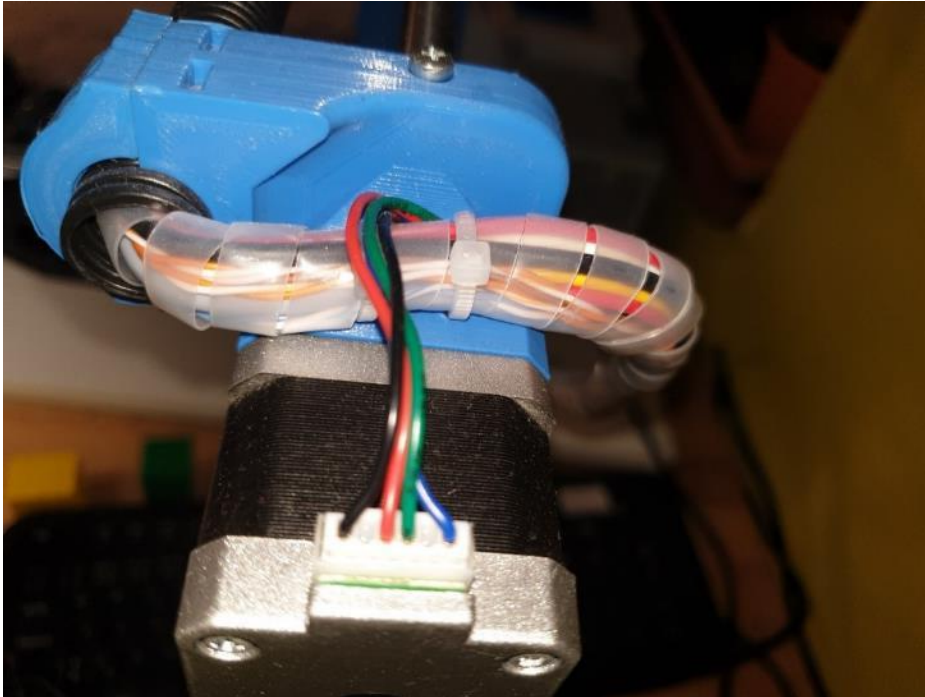
Obrázek 11: pouzdra ložisek osy X se zámký

Dalším konstrukčním prvkem na ose Y je úchyt motorů a konců lineárního vedení na zadní straně zařízení. Návrh tohoto dílu nebyl moc složitý. Jediné, čeho jsem se bál, byla pouze hloubka zapuštění vodičích tyčí, která činí také 10mm. Ale chtěl jsem mít co nejdelší dráhu pohybu. Upevnění tyčí jsem zvolil stejné, jako u osy X, tedy každá tyč je přitažena dvěma šrouby.

Mimo jiné je přes tento díl vedena kabeláž. Při návrhu kabeláže jsem musel myslet na několik faktorů. Elektroniku jsem se snažil maximálně zapouzdřit. Kabely musejí být vedeny tak, aby nebyly volné a nepřekážely v pohybu manipulátoru. Proto jsou kabelové svazky vedeny na první pohled trochu nelogicky přes zadní část manipulátoru, čímž se značně prodlužuje jejich délka.

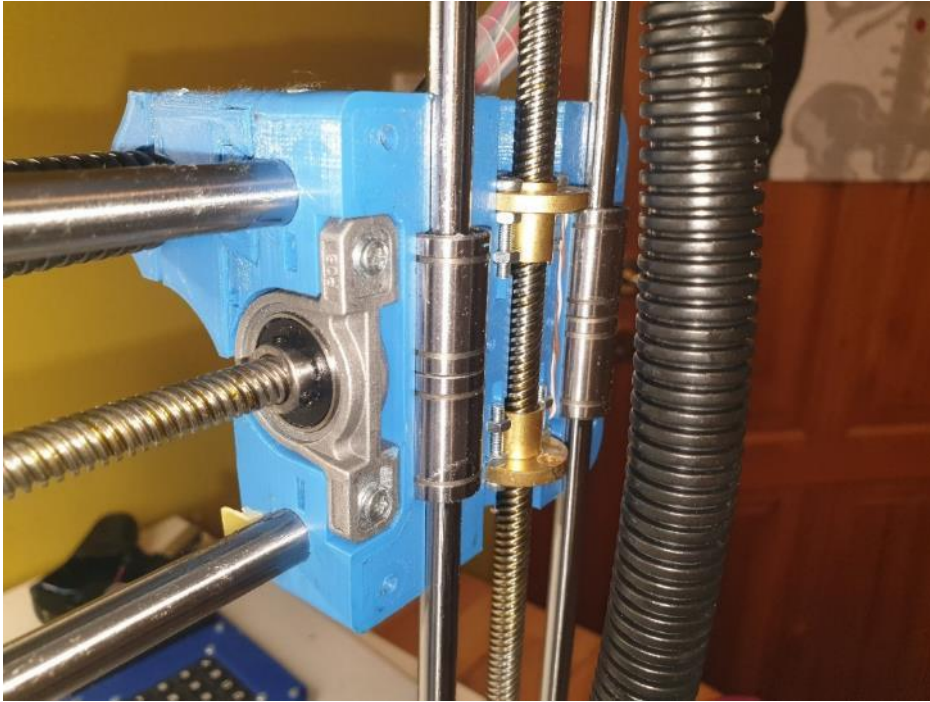
V tomto dílu přechází kabelový svazek svázaný univerzální omotávkou do kabelové chráničky, který vede souběžně s mechanickým pojezdem osy Y až na její konec, respektive začátek, z operátora pohledu. Kabelová chránička je připevněna třmenem k danému dílu. V obou těchto částech jsou drážky zapadající do drážek

v kabelové chráničce. Kabelový svazek s omotávkou jsem připevnil pomocí stahovacích pásek.



Obrázek 12: úchyt motoru osy Y s přechodem kabelového svazku do kabelové chráničky

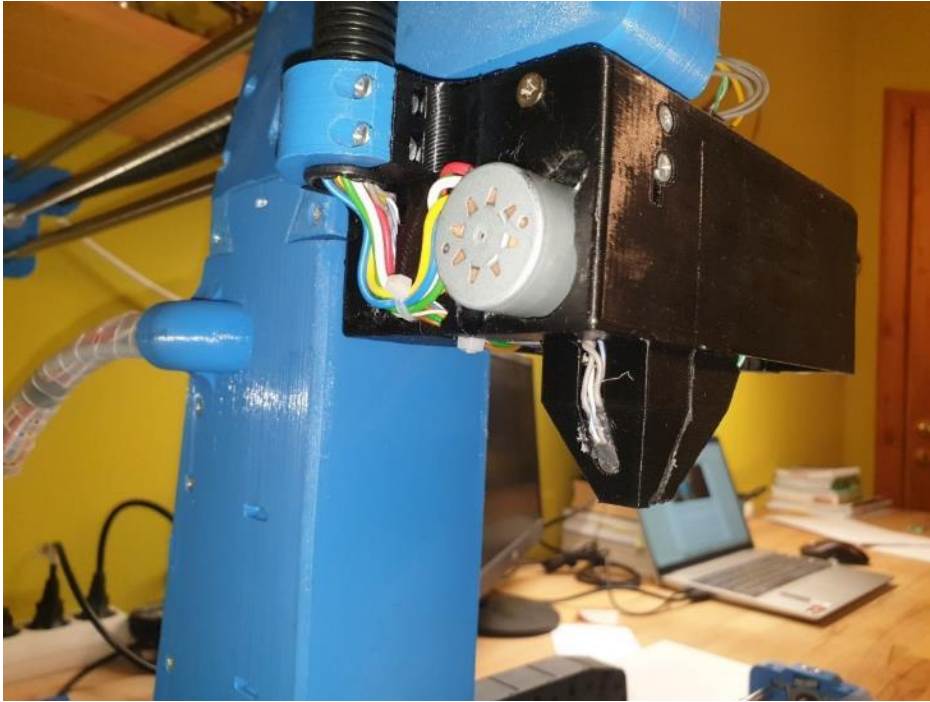
Na začátku osy Y jsou dva díly, pouzdra ložisek a matic pro osu Z. Návrh tohoto dílu byl značně složitý, jelikož jsem do malého prostoru musel umístit velké množství součástek. Vodící tyče jsou zde kvůli zvýšené páce zapuštěny 20mm do hloubky. Uvnitř těchto dvou sešroubovaných dílů jsou sevřeny lineární ložiska a trapézové matice pro osu Z. Díly jsou sešroubovány pomocí šesti šroubů. To byla první část návrhu. Jelikož osa Z je vedena na vodících tyčích s menším průměrem, bylo nutné změnit velikost otvorů pro uložení ložisek. V tomto díle opět přechází kabelová chránička na svazek s univerzální omotávkou. Oba tyto prvky jsou připevněny stejně jako na držáku motoru osy Y.



Obrázek 13: vnitřek pouzder ložisek osy Z

Pokud půjdeme po kabelech dál, dojdeme na vrchní část osy Z, kde se nachází držák motoru osy Z. Držák motoru osy Z je téměř totožný s držákem motoru osy Y, má pouze jiné rozměry.

Jako jedna z posledních částí návrhu konstrukce ramene byla spodní část osy Z, neboli hlava/čelist. Tento díl byl zároveň i pravděpodobně nejsložitější díl celé mechanické sestavy. Musel jsem do malého prostoru zakomponovat mnoho součástí, jako je pojezd čelisti, podsvícení, kamera, čelist a další. A zároveň jsem musel udržet co nejmenší hmotnost. Bylo těžké vymyslet, kudy povedou jednotlivé vodiče, aby byly chráněny a aby plnily svou funkci. Vodící tyče jsem zapustil do hloubky 20mm, jelikož jsem se obával jejich vysunutí při zatížení. Každá tyč je přitažena dvěma šrouby.



Obrázek 14: pohled na hlavu / čelist z boku



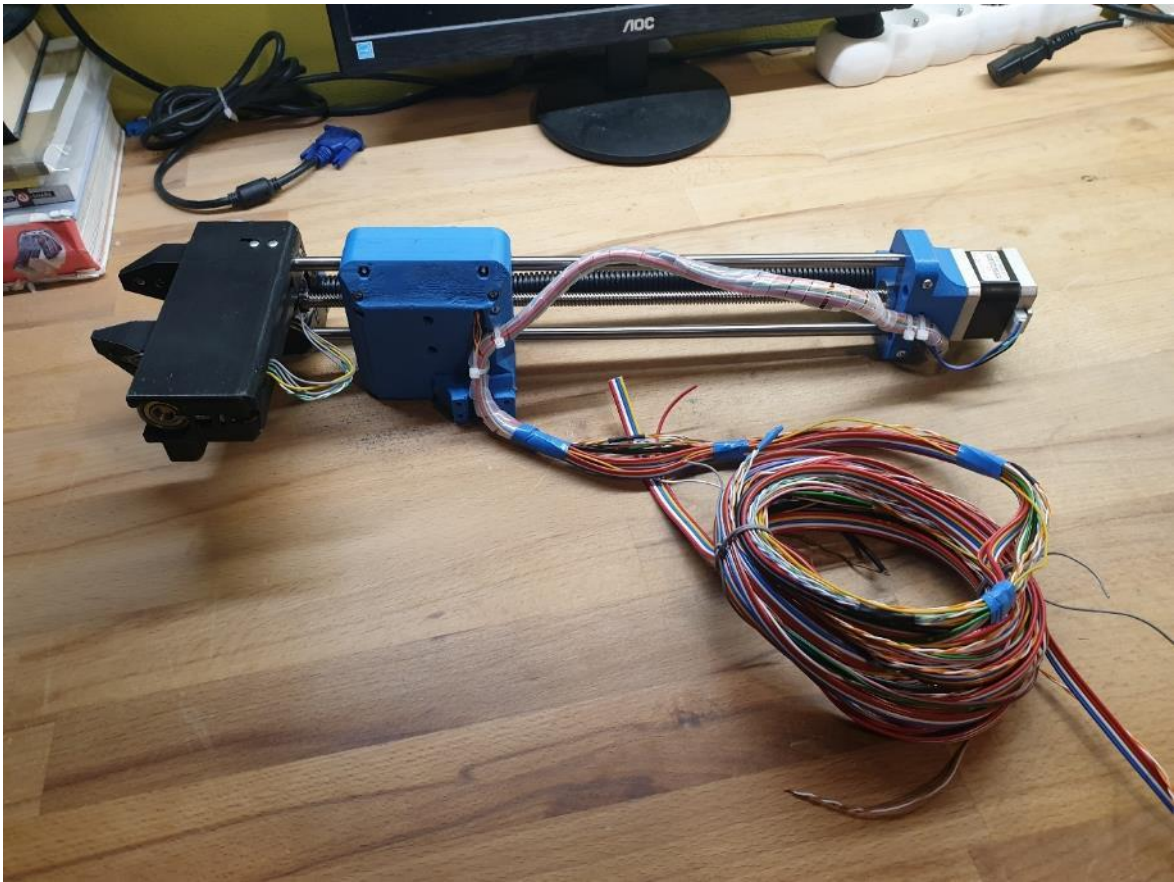
Obrázek 15: pohled na hlavu/čelist zespoda



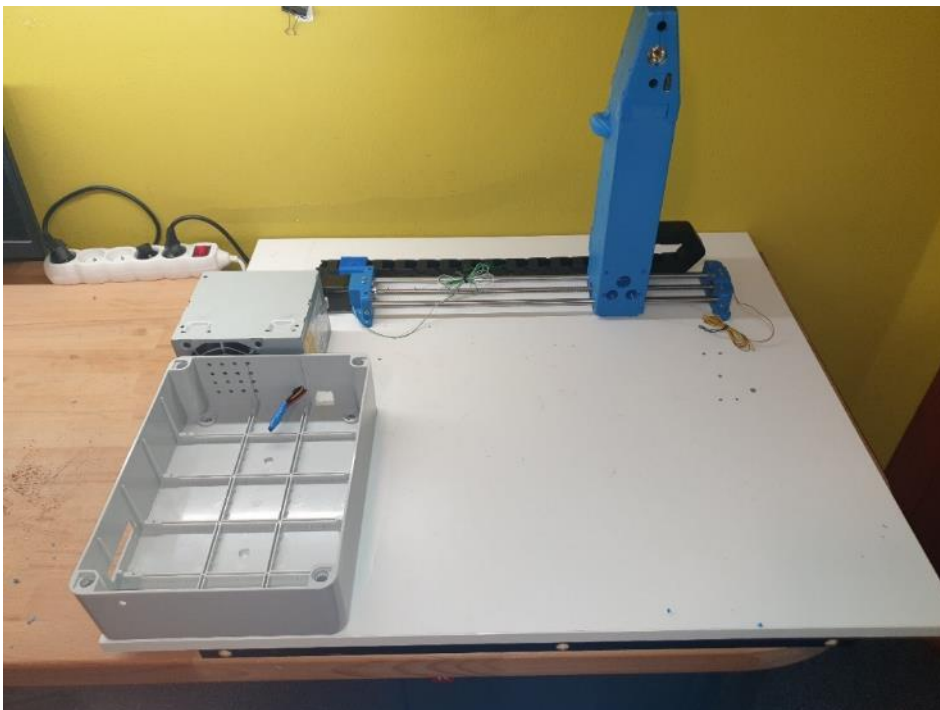
Obrázek 16: pohyblivý díl čelisti

Návrh každého dílu trval řádově jednotky až desítky hodin a vyžádal si mnoho prototypů.

Po dokončení čelisti jsem začal celé rameno kompletovat. V této fázi to již byla taková, nadneseně řečeno, stavebnice. Ne všechny díly do sebe seděly a byla nutná obrovská spousta mechanických zásahů a úprav.

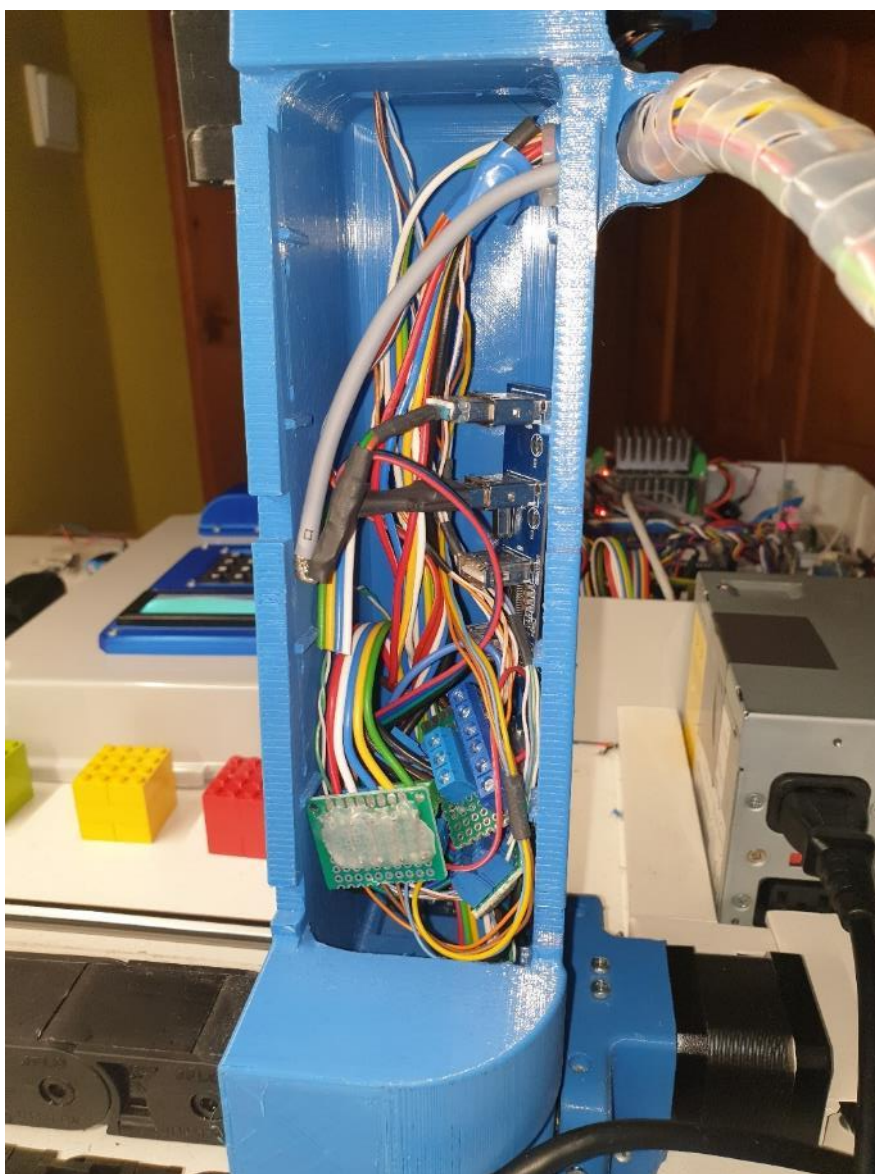


Obrázek 17: osa Z s kabelovým svazkem připravená na montáž



Obrázek 18: obrázek z kompletace zařízení

Při stavbě kompletaci elektrické části manipulátoru jsem postupoval v opačném směru, než při stavbě. Jako první jsem přidělal elektronické součástky v oblasti čelisti. Následně jsem vyvedl kabelový svazek skrz kabelovou chráničku nahoru k držáku motoru osy Z. Motor jsem připojil a omotaný svazek jsem vedl k pouzdrům ložisek osy Z. Tam jsem připojil koncové dorazy a skrz kabelovou chráničku podél osy Y jsem vedl svazek k držáku motoru osy Y. Tam jsem opět připojil krokový motor a omotaný kabelový svazek jsem vedl do středového dílu. Ve středovém dílu jsem připojil koncové dorazy, HCSR04 a připojil kamery do USB hubu.



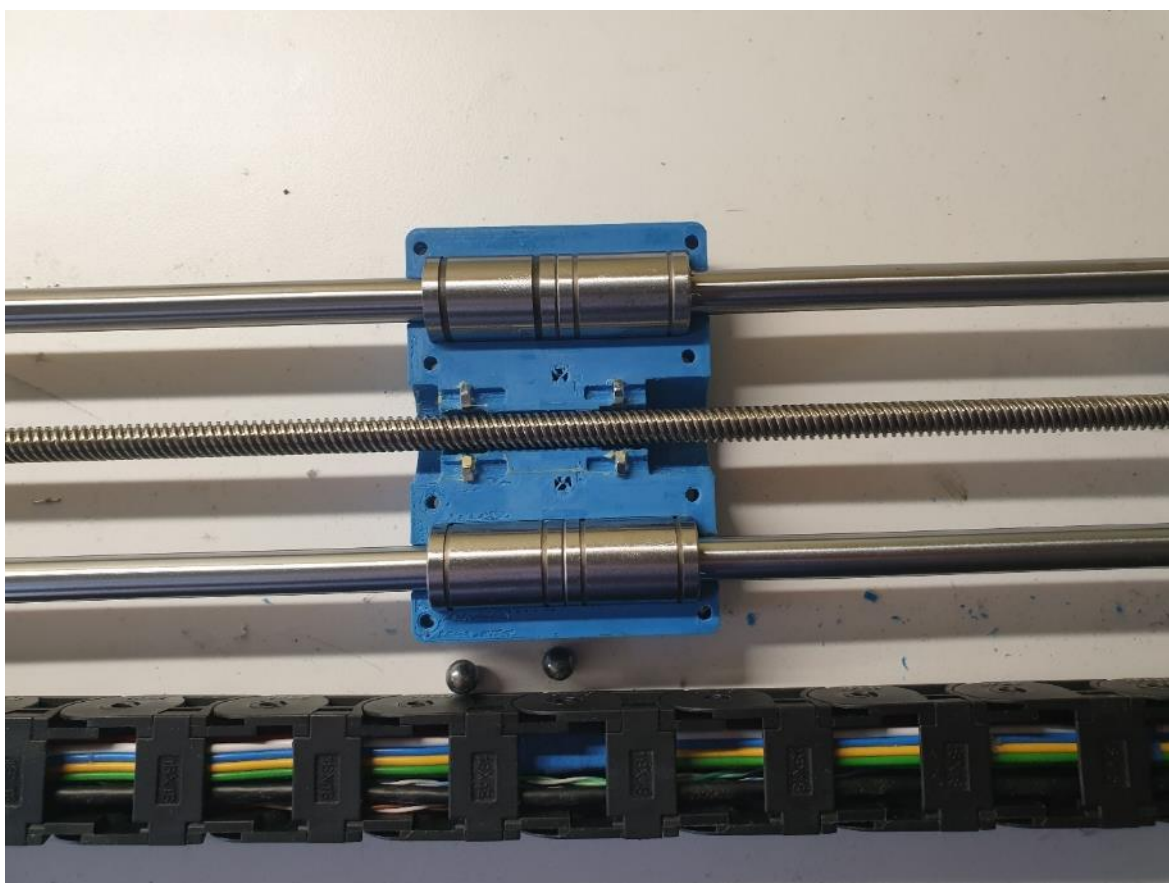
Obrázek 19: vnitřek středního dílu

Následně jsem kabelový svazek vedl kabelovým řetězem a instalačními lištami až do krabice. Tato část byla během sestavování elektrické části možná nejsložitější. Kvůli

stísněnému prostoru uvnitř středového dílu bylo velice obtížné kabelový svazek protáhnout skrz ohyb, přes který je kabelový řetěz spojený se středovým dílem. Kabelový řetěz jsem k podložce uchytil součástkou zhotovenou na 3D tiskárně.

Popis lineárních pojezdů

Jako pohybovou soustavu manipulátoru jsem zvolil lineární pojezdy s trapézovými šrouby a krokovými motory. Tento druh pohybového aparátu jsem zvolil kvůli jednoduchosti, univerzálnosti a programové nenáročnosti. Jelikož je kroučící moment převáděn přes šroub a většina váhy manipulátoru spočívá na vodících tyčích, lze díky tomu posouvat mnohem větší hmotnost, než by tomu bylo se stejnými motory u jiného druhu pohybového aparátu.



Obrázek 20: pouzdro ložisek osy X - vnitřek (bez trapézových matic)

Jako krokové motory jsem použil motory typu NEMA17 pro osy X, Y, Z a malý převodovaný krokový motor pro osu E.

Lineární pohybovou soustavu osy X jsem realizoval pomocí bipolárního krokového motoru 42HB40 se silou 0,5 Nm. Motor je napevno připevněn k podložce. K němu je

přes pružnou spojku připevněna trapézová závitová tyč o průměru 8mm, délce zhruba 500mm a stoupáním 8mm. Druhý konec závitové tyče je upevněn v ložisku s přírubou.

Na závitové tyči jsou dvě trapézové matice, které jsou napevno spojeny se středovým dílem a pouzdrem ložisek osy X.

Závitová tyč je umístěna v ose mezi dvěma vodícími tyčemi o délce 50mm a průměrem 10mm. Je velice důležitá sousost všech tyčí, jelikož odchylka i půl milimetru může mít za následek absolutní nepohyblivost osy. Na každé tyči jsou umístěny dvě lineární ložiska o průměru 10mm a jsou uzavřeny mezi pouzdrem ložisek osy X středovým dílem.

Pohybovou soustavu osy Y jsem realizoval totožně, liší se pouze modelem krokového motoru, kterým je 17HS4401 se silou 0,4 Nm. Slabší krokový motor jsem zde umístil, jelikož pohyb této osy nevyžaduje takovou sílu, jako osy Y a Z. Na této ose motor pohybuje spolu s celou horní sestavou.

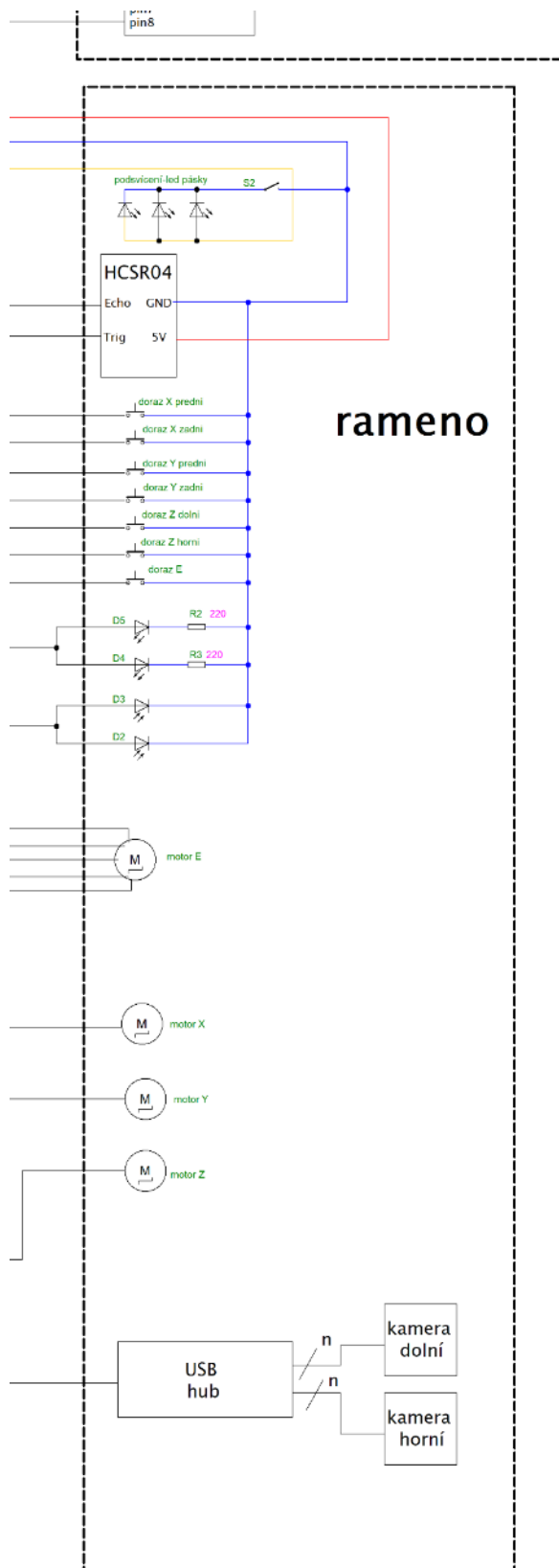
Pohybovou soustavu osy Z jsem realizoval podobně, jako u předešlých os. Zásadní rozdíl je v použití 8mm vodících tyčí a zmenšení jejich rozteče, jelikož tato osa není namáhána v horizontálním, nýbrž ve vertikálním směru.

V hlavě je uložena pohybová soustava osy E. Zde jsem použil bipolární krokový motor 28byj-48. Ten je napevno přidělaný k hlavě a pružnou spojkou je k němu připojena trapézová závitová tyč, která končí v ložisku, které je volně uložené na druhé straně hlavy. Na trapézovém šroubu je umístěna matice, pevně spojená s pohyblivou čelistí. Čelist klouže po 8mm vodící tyči, která je umístěna nad tyčí trapézovou.



Obrázek 21: pohybová soustava osy E

Popis elektronické části ramene manipulátoru



Obrázek 22: zvětšené elektrické schéma ramene manipulátoru

Elektronická výbava a kabelový rozvod

Samotná rameno manipulátoru má v sobě zakomponované jak vstupní, tak výstupní periferie. Na každé ose se nachází koncové dorazy s mikropsínači pro snímání polohy. V hlavě můžeme nalézt podsvícení kamery ve formě 12V LED pásků s teplým světlem, které se dá pomocí spínače manuálně zapnout či vypnout. Dále také kameru s automatickým ostřením, určenou pro snímání manipulační plochy a rozeznávání barev. A úchopová čidla v samotné čelisti, realizované IR diodami a foto diodami na protější straně čelisti. Ty mají za úkol zjistit přerušování světla z IR diod uchopeným předmětem. Ve středovém dílu se nachází druhá kamera, která je vybavena manuálním ostřením a ultrazvukový měřič vzdálenosti HCSR04, oboje pro snímání manipulační plochy před středovou částí. Dále pak USB hub, který propojuje kamery s Raspberry v krabici. Na každé ose je samozřejmě umístěn jeden krokový motor.

Jako kamery jsem použil obyčejné webkamery, které jsem vyňal z jejich plastových krytů.

Kabely a vodiče

Během stavby ramene manipulátoru jsem používal různé druhy vodičů a kabelů.

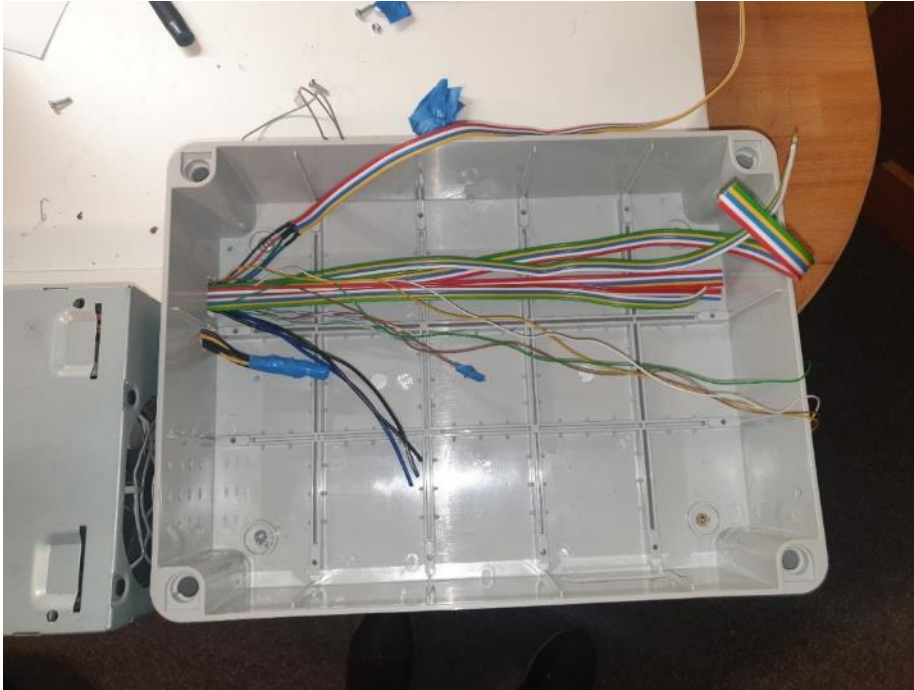
Pro připojení krokových motorů jsem použil plochý kabel o čtyřech či pěti vodičích s průřezem $0,35 \text{ mm}^2$

Pro připojení horní kamery jsem použil čtyř žilový stíněný kabel s průřezem vodičů $0,25 \text{ mm}^2$

Pro rozvod napájení jsem použil vodiče s průřezem 1 mm^2

Pro připojení koncových dorazů s mikropsínači jsem použil vodiče z UTP kabelu, mezi středem a krabicí jsem pro tyto účely použil opět plochý kabel.

Ploché kabely jsem používal především pro jejich prostorovou nenáročnost, jednoduchou práci a dobrou ohebnost uvnitř kabelového řetězu. Vodiče z UTP kabelu jsem použil pro jejich dostupnost a malý průřez.



Obrázek 23: kabely vedoucí do krabice

Krabice s řídicí elektronikou

Řídicí elektroniku jsem umístil do bílé instalační krabice, kterou jsem situoval do rohu základny manipulátoru. Víko krabice jsem využil jako ovládací panel a umístil jsem na něj ovládací prvky a LCD Displej. Za krabicí s elektronikou (z pohledu operátora) jsem umístil napájecí zdroj.

Napájení

Jako napájecí zdroj jsem použil napájecí zdroj ze starého serveru. Tento zdroj jsem zvolil pro jeho dostupnost, výkon a různá napájecí napětí, na kterých pracují součástky a moduly uvnitř krabice i na rameni manipulátoru.

Výstupy zdroje:

12 V – žlutý vodič

5 V – červený vodič

3,3 V – oranžový vodič

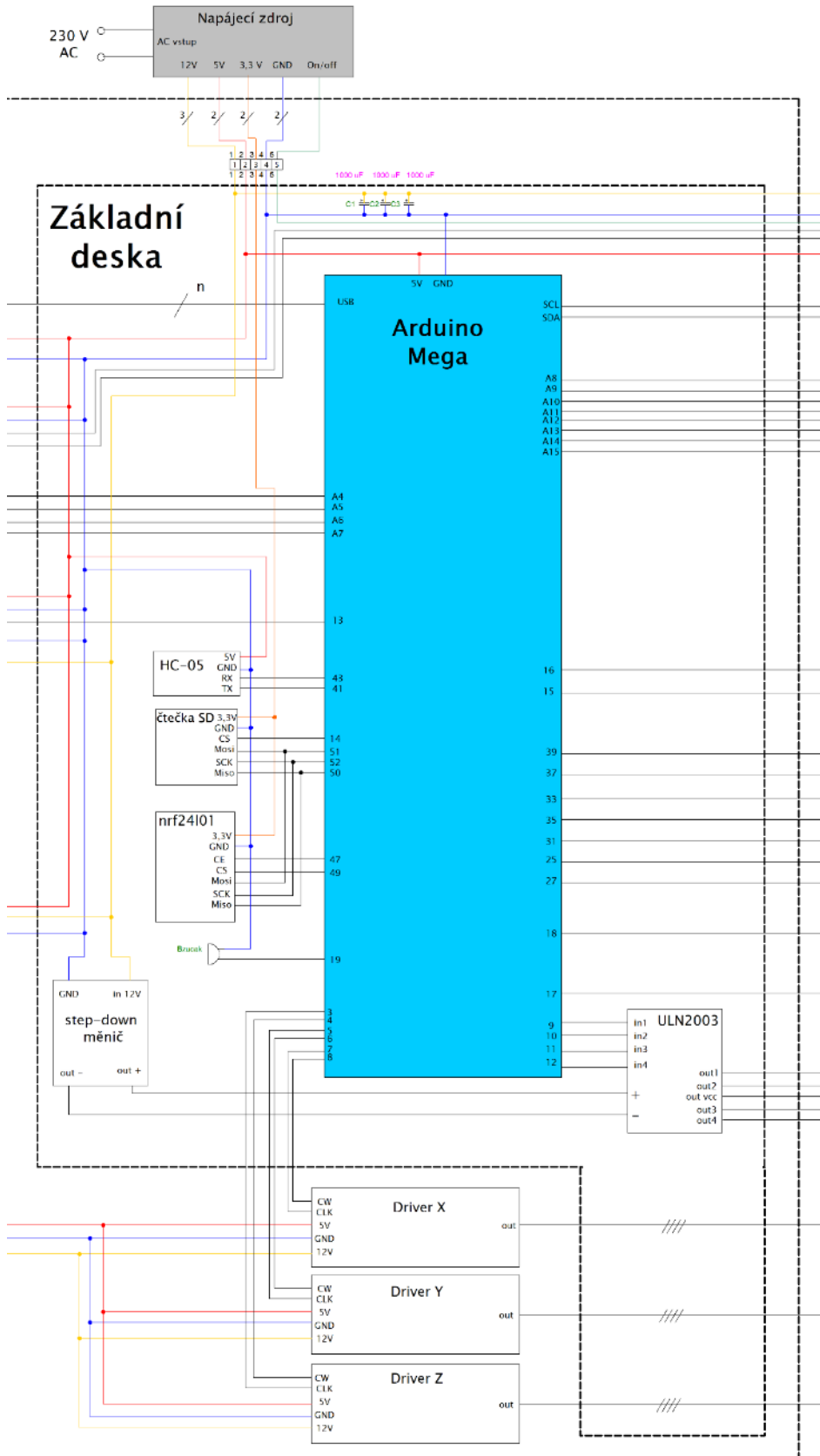
GND – černý vodič

on/off – zapnutí a vypnutí zdroje- zelený vodič

Napájecí zdroj je před dvě svorkovkovernice připojen do základní desky a je spínán spínačem, který je umístěn na ovládacím panelu a zapojen taktéž do základní desky. Základní deska funguje jako uzel pro distribuci napájení. Je z ní vyvedeno napájení ramene manipulátoru, napájení Raspberry, relé modulu a driverů krokových motorů. Záporný pól pro krokové motory je veden přes spínací kontakt K1 na relé modulu. Tento kontakt může být rozepnut buď elektronicky výstupem z Arduina a nebo mechanicky stisknutím Emergency STOP tlačítka na ovládacím panelu. Při rozepnutí kontaktu dojde k vypnutí napájení driverů a krokové motory os X, Y a Z se přestanou pohybovat. Při opětovném připojení driverů do obvodu dojde k velikému skokovému zatížení zdroje a ten se vypne. To jsem vyřešil připojením tří paralelně zapojených kondenzátorů. Každý má kapacitu 1000 mikroFaradů.

Základní deska

Základní deska je srdcem celého zařízení. Je na ni umístěno Arduino, které řídí pohyb celého zařízení, větví se zde napájení. Je umístěna zhruba uprostřed krabice a vede do ní většina vodičů.



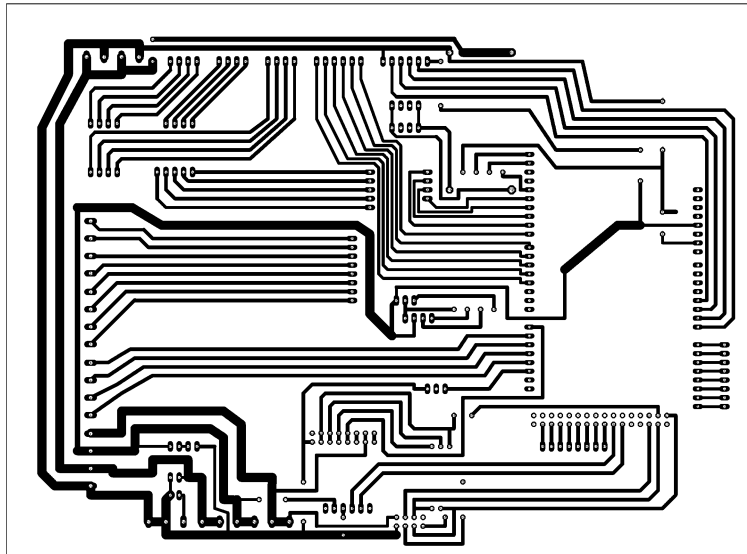
Obrázek 24: zvětšené schéma základní desky



Obrázek 25: osazená základní deska

Výroba základní desky

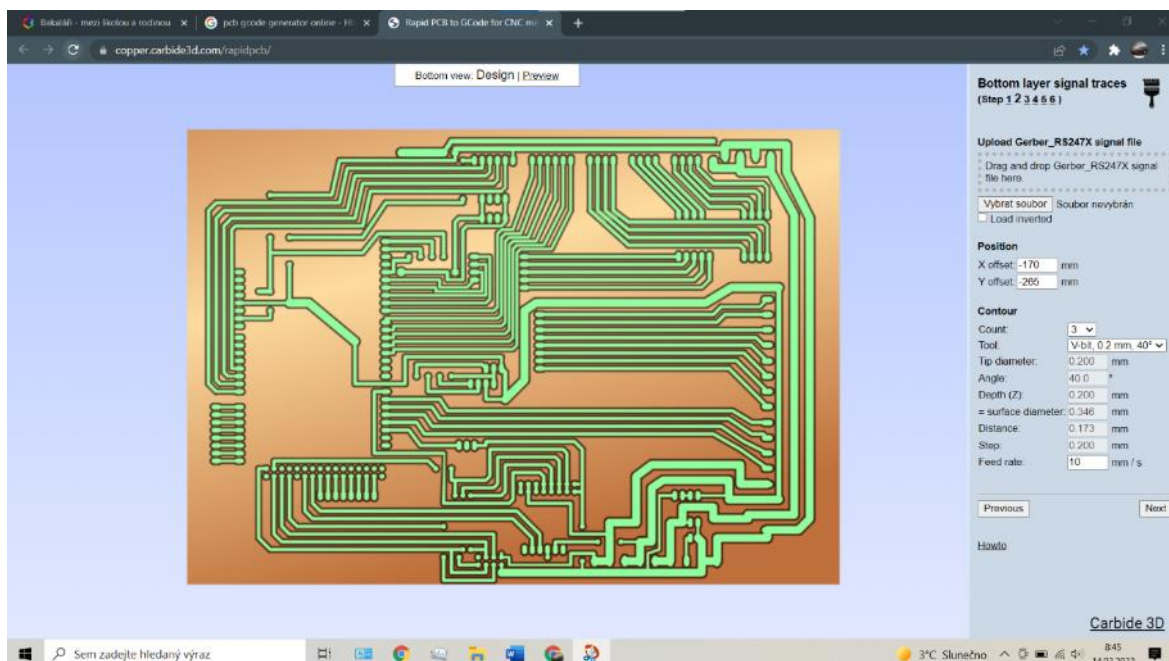
Nejprve jsem v prostředí Autodesk Eagle nakreslil funkční schéma, následně jsem začal navrhovat plošný spoj. Jako první jsem si ohraničil velikost desky, poté jsem podle svých požadavků rozmístil součástky. Chtěl jsem původně vygenerovat vodivé cesty pomocí funkce Auto router, ale program nebyl schopen vygenerovat efektivní sestavu vodivých spojů. Proto jsem musel vodivé spoje navrhnout sám a zároveň jsem také musel myslet na technologii výroby, kterou mělo být chemické leptání plošného spoje ve školní dílně. Vodivé cesty jsem musel udělat dostatečně široké a v dostatečné vzdálenosti od sebe, aby i při nedokonalém vyleptání desky byl obvod funkční.



Obrázek 26: cesty na základní desce - obrázek z EAGLE

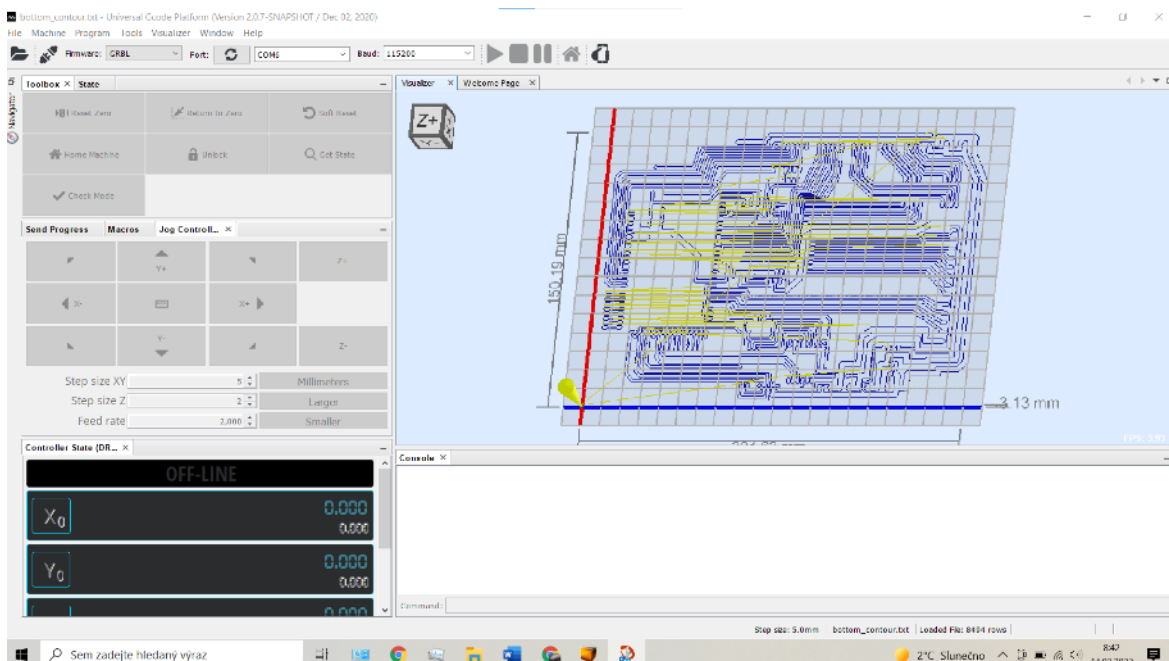
Po dokončení návrhu jednostranné DPS jsem vytisknul osvitovou předlohu pro přenos leptání DPS. Plošný spoj jsem vyleptal na jednostranný cuprexit o tloušťce cca 1,25 mm. Následně jsem si měděnou stranu plošného spoje nechal profesionálně pocínovat v externí firmě. Pak už stačilo pouze vyvrtat otvory pro THT součástky. Při osazování desky jsem bohužel narazil na problém, že mi součástky do desky nepasují. To bylo zapříčiněno tím, že jsem špatně vytisknul šablonu pro leptání, která byla o několik procent zmenšená oproti originálu. Tento problém mě ve výrobě zařízení velice zpomalil. Po tomto nezdaru jsem se výrobě nové DPS nevěnoval, jelikož začali prázdniny a tudíž jsem ve škole nemohl vyleptat nový plošný spoj. Po prázdninách jsem si za peníze z brigády koupil malý CNC obráběcí stroj a jal jsem se DPS vyfrézovat. To bylo mnohem složitější, než se zdá, proto popis postupu frézování a problémy, které nastaly, popíši pouze ve zkratce. Jelikož základní deska je plošný spoj poměrně velikých rozměrů, měl jsem problém s upnutím cuprexitu k upínacímu stolu tak, aby se cuprexit neprohýbal. Při frézování zajíždí obráběcí nástroj pouze desetinu milimetru do hloubky obráběného materiálu. Proto sebemenší prohnutí cuprexitu znamená, že se izolační drážky buď nevyfrézují, a nebo se ubere moc materiálu a cestičky se úplně odfrézují.

Z Eaglu jsem vyexportoval Gerber data potřebná pro výrobu DPS. Přes online nástroj Rapid-PCB jsem vygeneroval G-kód pro frézování cestiček a pro vyvrtání otvoru pro THT součástky.



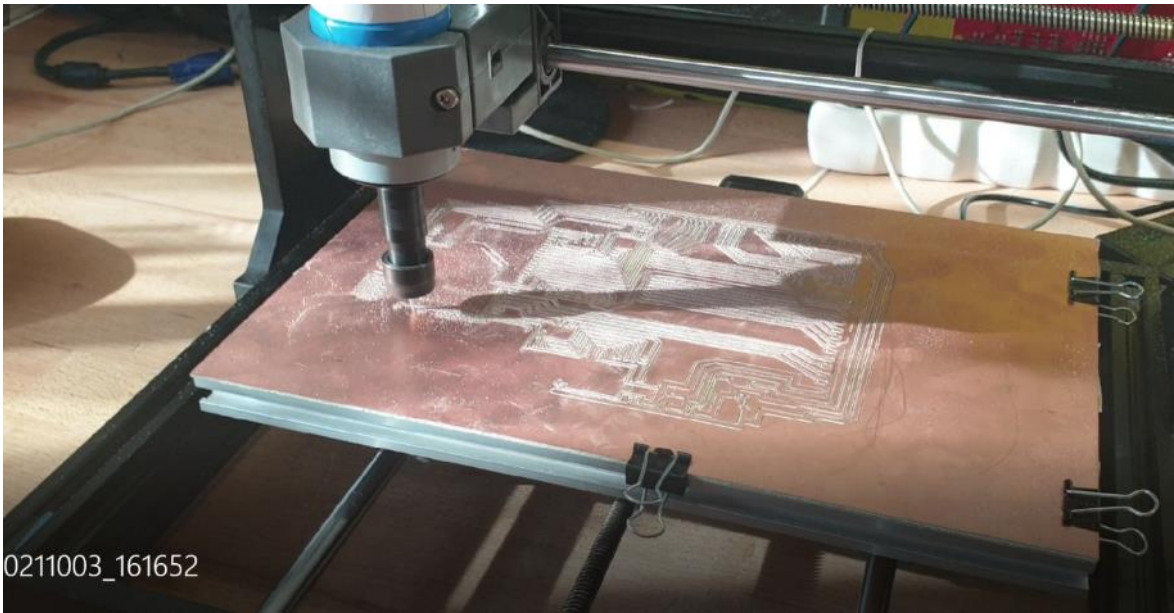
Obrázek 27: screenshot z programu Rapid PCB

Tento G-kód jsem vložil do programu Universal Gcode Sender, který slouží k odesílání G-kódu do obráběcího stroje.

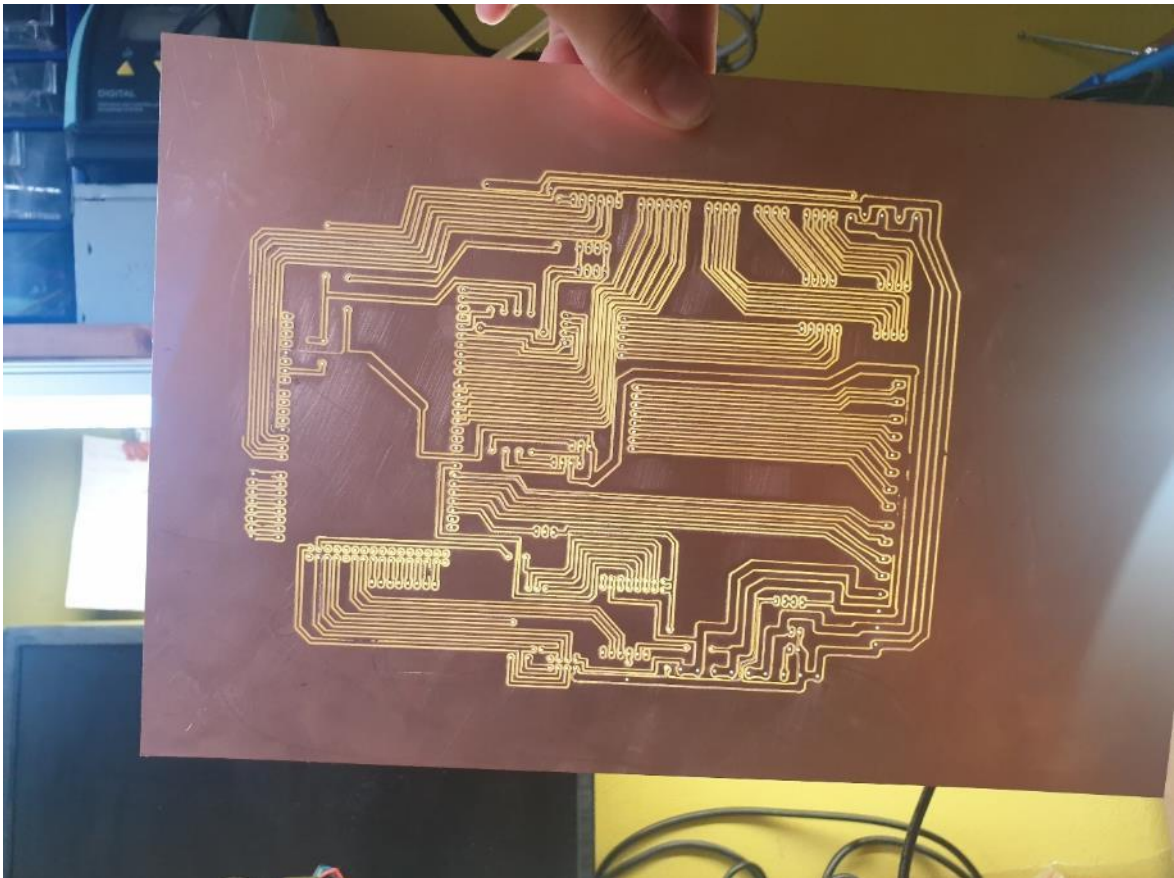


Obrázek 28: screenshot z programu Universal Gcode Sender

Poté jsem do vřetena vložil gravírovací frézu, nastavil nulový bod obrobku a začal frézovat cestičky. Po vyfrézování cestiček jsem vyměnil gravírovací frézu za 0,8mm vrták. Poté jsem v UGS otevřel G-kód pro vrtání otvorů, opět nastavil nulový bod a začal jsem vrtat. Po dokončení vrtání byla deska hotová.



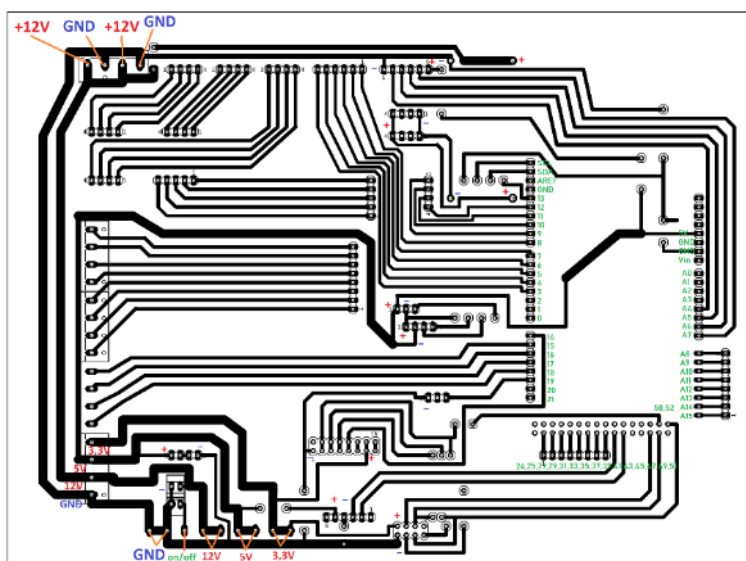
Obrázek 29: foto z frézování cest na CNC



Obrázek 30: Hotová základní deska včetně nedokonalostí z výroby

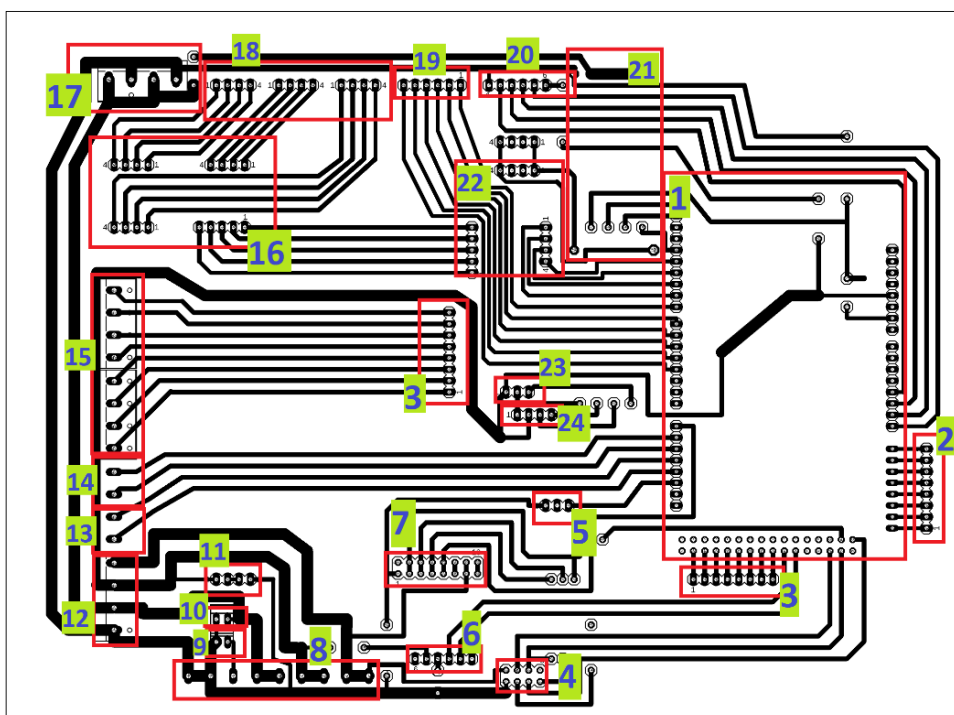
Finální DPS měla mnoho nedokonalostí, které jsem musel ručně opravovat. Po tomto procesu byla deska připravena na pájení součástek.

Následné pájení součástek proběhlo hladce. Když byli všechny součástky osazené, ověřil jsem ohmmetrem správné vodivosti a nepřístupné zkraty, tedy potvrdil jsem funkčnost. V tento moment byla deska hotová.



Obrázek 31: Zapojení napájení na základní desce a zapojení vstupních/výstupních pinů

Porty a moduly osazené na základní desce



Obrázek 32: popis osazení základní desky (vysvětlivky v tabulce 3)

(1) Arduino mega, které řídí celé zařízení a je přes sériový port připojeno k Raspberry.

(6) Bluetooth modul HC-05, který lze využít ke komunikaci s jakýmkoliv zařízením. Já ho ve svém projektu využívám k odesílání diagnostických dat do svého notebooku. To mi pomáhá při odladování programu. Tyto data nemůžu odeslat normálně přes sériový port Arduina, jelikož jej používám pro komunikaci s počítačovým ovládacím softwarem.

(7) Čtečka SD karet, tento konektor momentálně není osazen modulem, jelikož ho v této fázi projektu nepotřebuji. Tento modul jsem plánoval využít pro ukládání a čtení .txt souborů, které by obsahovaly souřadnice pro pohyb manipulátoru.

(4) Komunikační modul nrf24l01, tento modul využívám pro bezdrátový přenos dat z joysticku, který lze použít pro ovládání, do Arduina.

(21) Step-down měnič, tento modul snižuje napětí pro napájení motoru a driveru osy E.

(22) Driver pro motor osy E, ten je osazen tranzistorovým polem ULN2003. Jeho provozní napětí driveru a motoru činí 5-12 V

(5) Bzučák, slouží pro akustickou signalizaci

Dále se na základní desce nachází mnoho konektorů a svorkovnic pro připojení periferií, napájení atd. Ty zde nebudu popisovat

Tabulka 2: zapojení I/O Arduina

| Označení pinu | Popis vstupu na modulu/součástce | Modul/součástka |
|---------------|----------------------------------|-----------------|
| SCL | SCL | I2c převodník |
| SDA | SDA | I2c převodník |
| A8 | | klávesnice |
| A9 | | klávesnice |
| A10 | | klávesnice |
| A11 | | klávesnice |
| A12 | | klávesnice |
| A13 | | klávesnice |
| A14 | | klávesnice |
| A15 | | klávesnice |
| 16 | echo | HCSR04 |
| 15 | trig | HCSR04 |
| 39 | | Doraz X predni |
| 37 | | Doraz X zadni |
| 33 | | Doraz Y predni |
| 35 | | Doraz Y zadni |
| 31 | | Doraz Z dolni |
| 25 | | Doraz Z horni |
| 27 | | Doraz E |
| 18 | | fotodiody |
| 17 | | IR diody |
| 9 | in 1 | ULN2003 |
| 10 | In 2 | ULN2003 |
| 11 | in 3 | ULN2003 |
| 12 | in 4 | ULN2003 |
| A4 | in1 | Relé modul |
| A5 | in2 | Relé modul |
| A6 | in3 | Relé modul |
| A7 | in4 | Relé modul |
| 13 | out | IR přijímač |
| 43 | RX | Bluetooth HC-05 |
| 51 | TX | Bluetooth HC-05 |
| 14 | CS | Čtečka SD Karet |
| 51 | Mosi | Čtečka SD Karet |
| 52 | SCK | Čtečka SD Karet |
| 50 | Miso | Čtečka SD Karet |
| 47 | CE | Nrf24l01 |
| 49 | CS | Nrf24l01 |

| | | |
|----|------|----------|
| 51 | Mosi | Nrf24l01 |
| 52 | SCK | Nrf24l01 |
| 50 | Miso | Nrf24l01 |
| 3 | CLK | Driver Z |
| 4 | CW | Driver Z |
| 5 | CLK | Driver Y |
| 6 | CW | Driver Y |
| 7 | CLK | Driver X |
| 8 | CW | Driver X |

Drivery

Jako driver krokových motorů pro osy X, Y, Z jsem použil TB6600 V1.2, které jsou umístěny vedle základní desky. Tyto drivery mají veliký rozsah napájecího napětí cca 9-32 a výstupní proud až 4,5 A. Lze na nich nastavit výstupní proud i délku kroku. Jejich velikou výhodou je ovládání krokového motoru pouhými dvěma vodiči CW a CLK, které jsou připojeny na výstup Arduina.

Mají celkem 4 vstupy pro ovládání:

5V – napájení logické části

CLK- krokovací impulsy

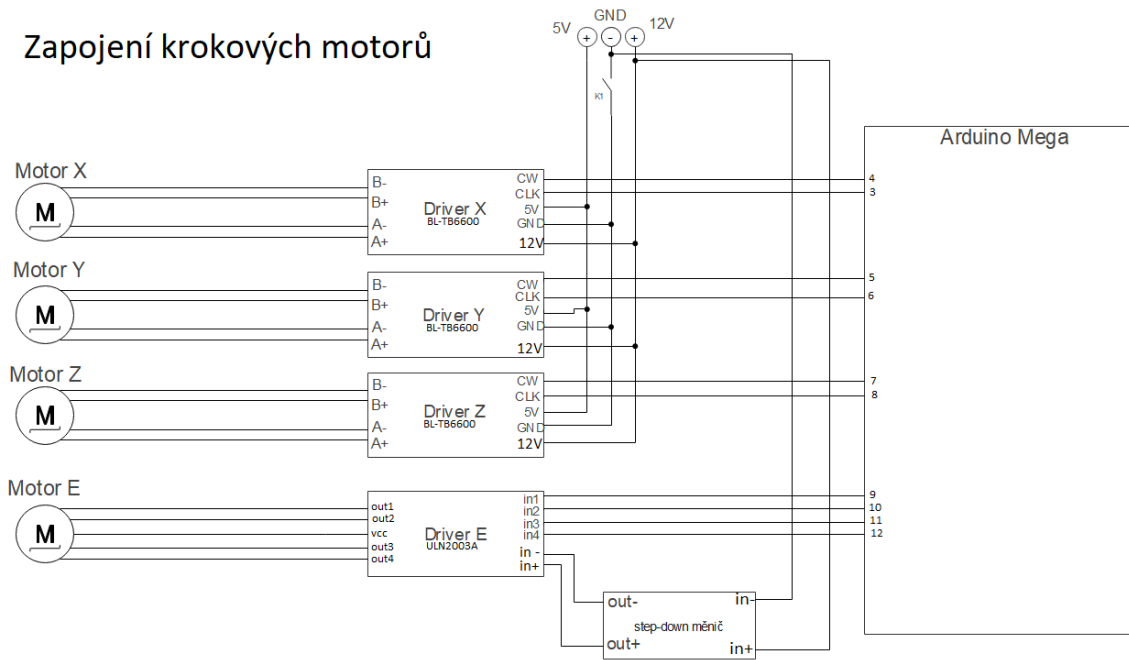
CW – směr otáčení

EN – aktivace driveru

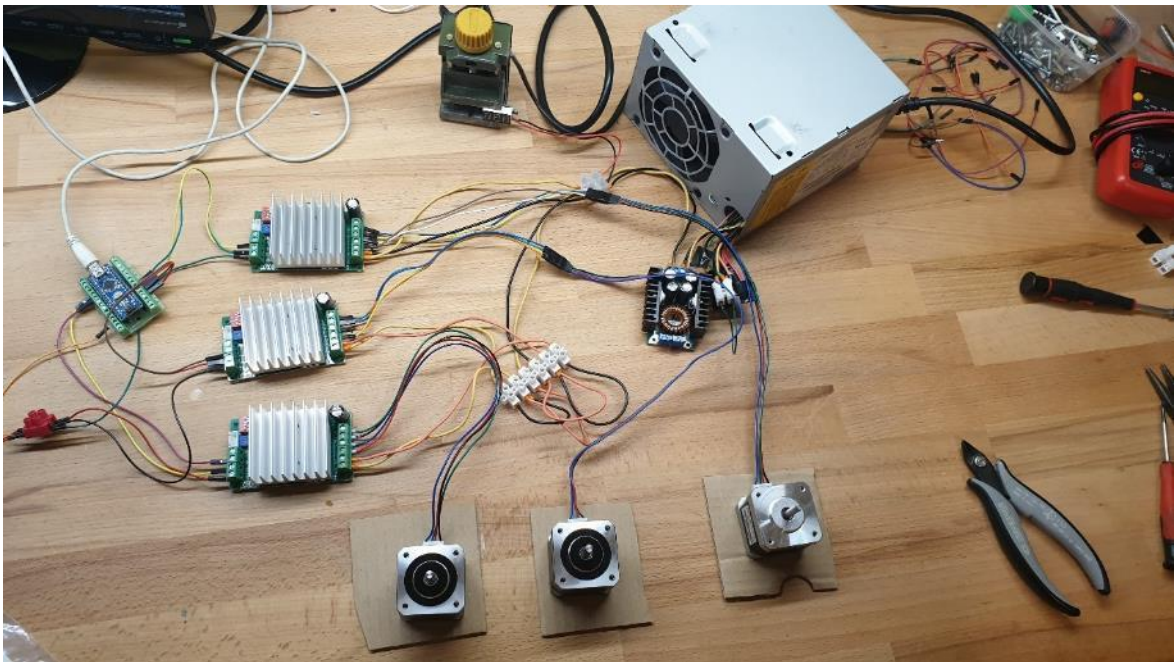
A celkem 4 výstupy pro jednotlivé fáze krokového motoru: A+, A-, B+, B-. Tyto výstupy vedou do základní desky, kde jsou na ně připojeny jednotlivé krokové motory.

Pro řízení motoru osy E jsem použil tranzistorové pole ULN2003, umístěné v modulu přímo na základní desce. Krokový motor a tranzistorové pole jsou napájen přes regulovatelný měnič napětí. Toto tranzistorové pole má 4 vstupy (in1 až in4) a proti nim 4 výstupy (out1 až out4) a napájení. Vstupy jsou tranzistorového pole jsou připojeny na výstupy Arduina.

Zapojení krokových motorů

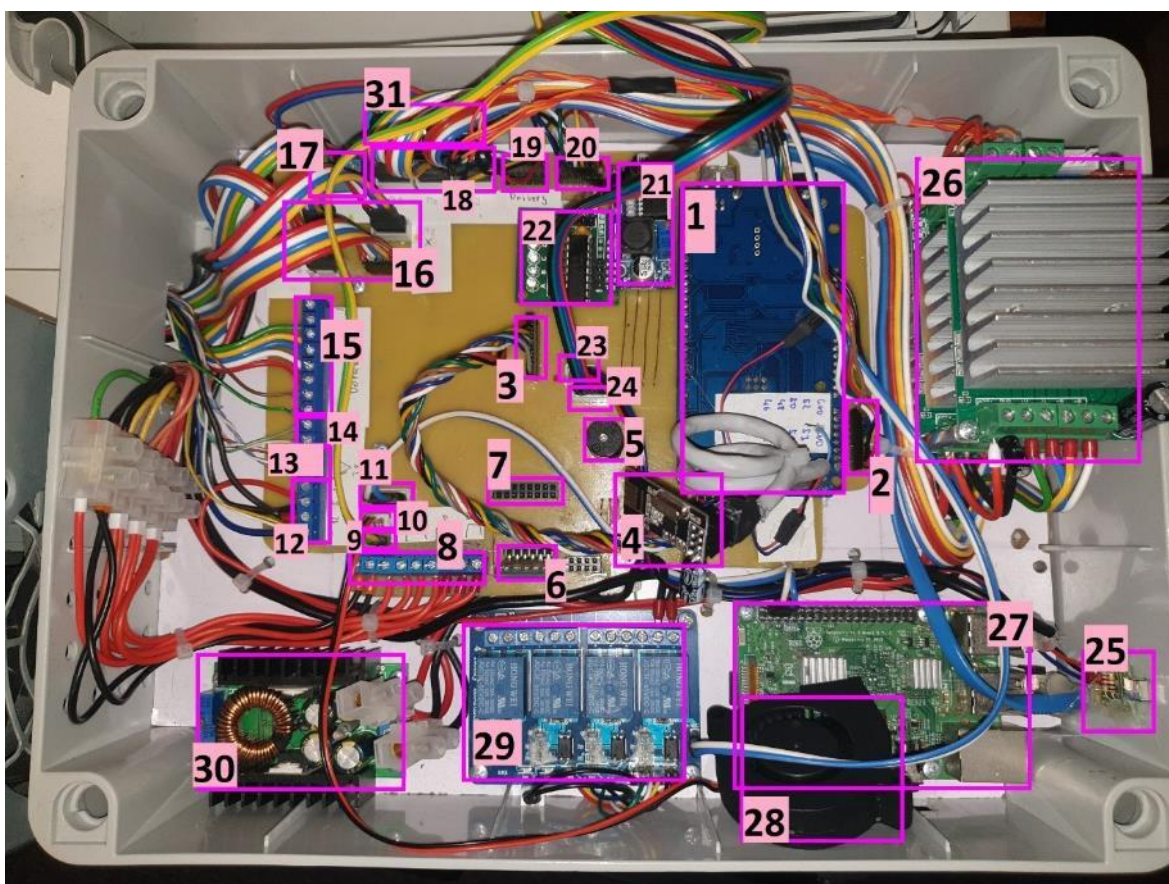


Obrázek 33: podrobné schéma zapojení krokových motorů



Obrázek 34: zapojení krokových motorů při jejich prvním testování před stavbou zařízení

Zbytek elektroniky v krabici



Obrázek 35: popis obsahu krabice (vysvětlivky v tabulce 3)

Tabulka 3: vysvětlivky popisu obsahu krabice

| index | název |
|-------|---|
| 1 | Arduino Mega |
| 2 | Konektor pro připojení klávesnice |
| 3 | Propojovací konektory na DPS |
| 4 | Komunikační modul nrf24l01 |
| 5 | Bzučák |
| 6 | Bluetooth modul HC-05 |
| 7 | Čtečka SD karet |
| 8 | Svorkovnice pro připojení napájecího zdroje |
| 9 | Konektor pro připojení hlavního vypínače |
| 10 | Konektor pro připojení ventilátoru |
| 11 | Konektor pro napájení Raspberry |
| 12 | Svorkovnice pro napájení Ramene |
| 13 | Svorkovnice – Diody v Rameni |
| 14 | Svorkovnice - vstup z HCSR04 |
| 15 | Svorkovnice – vstup z dorazů |
| 16 | Konektory pro připojení motorů |
| 17 | 12V výstup pro napájení motorů a připojení kondenz. |
| 18 | Konektory pro připojení motorů |
| 19 | Konektor pro připojení driverů |

| | |
|----|-------------------------------------|
| 20 | Konektor pro připojení relé |
| 21 | Měnič pro napájení motoru osy E |
| 22 | ULN2003 – driver motoru osy E |
| 23 | Konektor pro připojení displeje |
| 24 | Konektor pro připojení IR přijímače |
| 25 | IR přijímač |
| 26 | Driver krokový motorů os X, Y, Z |
| 27 | Raspberry pi |
| 28 | Ventilátor pro Raspberry |
| 29 | Relé modul |
| 30 | Měnič - nevyužitý |
| 31 | Kondenzátory |

(27) Raspberry pi 3B, jednodeskový počítač, který slouží k ovládání manipulátoru přes počítačový software a ke zpracování obrazu z kamer. Přes sériový port komunikuje s Arduinem. Je k němu připojena klávesnice, myš a monitor. Raspberry je chlazeno 12 V ventilátorem a napájeno napětím 5V ze základní desky.

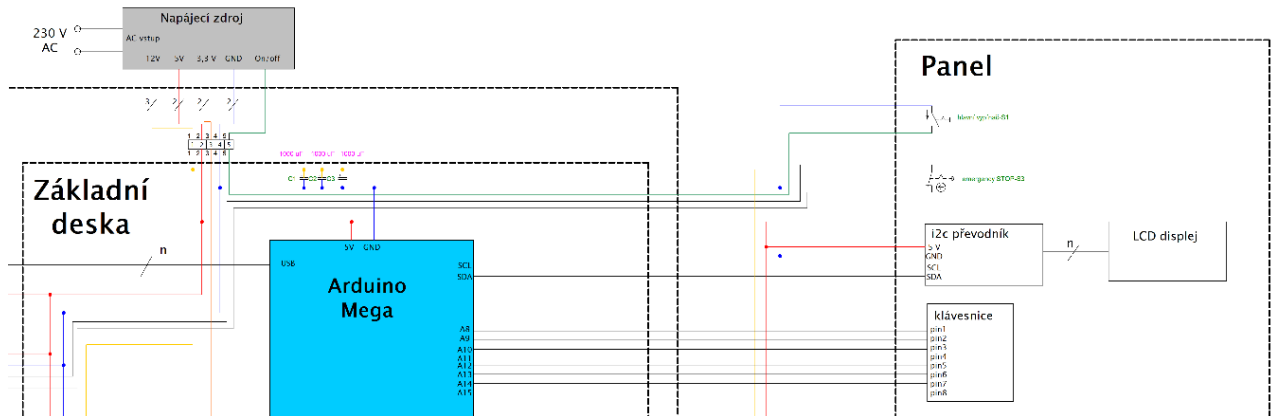
(29) Relé modul obsahuje čtyři relé. Zapojeno je pouze relé K1, které slouží k zapínání a vypínání driverů krokových motorů. Relé modul je napájen napětím 5V a lze ho vypnout Emergency STOP tlačítkem.

(30) step-down měnič. Původně jsem s ním chtěl snižovat napájecí napětí driverů krokových motorů, ale nakonec jsem drivery připojil napřímo a měnič proto neplní žádnou funkci

(25) IR přijímač je umístěn na stěně krabice směrem k operátorovi. Slouží pro příjem signálu z televizního dálkového ovladače. Tento signál následně zpracovává a předává do Arduina.

Ovládací panel

Ovládací panel se nachází na víku krabice s řídicí elektronikou. Slouží k přepínání režimu ovládání, zobrazování informací, zapnutí a vypnutí napájení a k nouzovému ukončení pohybu ramene manipulátoru. Obsahuje znakový LCD displej 20x4, maticovou klávesnici, hlavní vypínač (S1) a emergency STOP tlačítko s rozpínacím kontaktem.



Obrázek 36: zvětšené schéma panelu



Obrázek 37: popis ovládacího panelu

Otvory pro součástky jsem do panelu vrtal a následně ručními nástroji upravil.

Displej s klávesnicí jsou umístěny v plastovém rámečku, který jsem vyrobil za použití technologie 3D tisku. Rámeček je přišroubovaný do víka krabice.

Blíže k operátorovi se nachází hlavní vypínač, který je položený naležato a ukrytý v plastovém pouzdře. Hlavní vypínač jsem takto ukryl, aby nemohlo dojít k jeho náhodnému vypnutí a tím ztracení např. neuloženého programu na Raspberry.

Na levé straně od hlavního vypínače se nachází emergency STOP tlačítko, které slouží k odpojení driverů od napájecího zdroje.

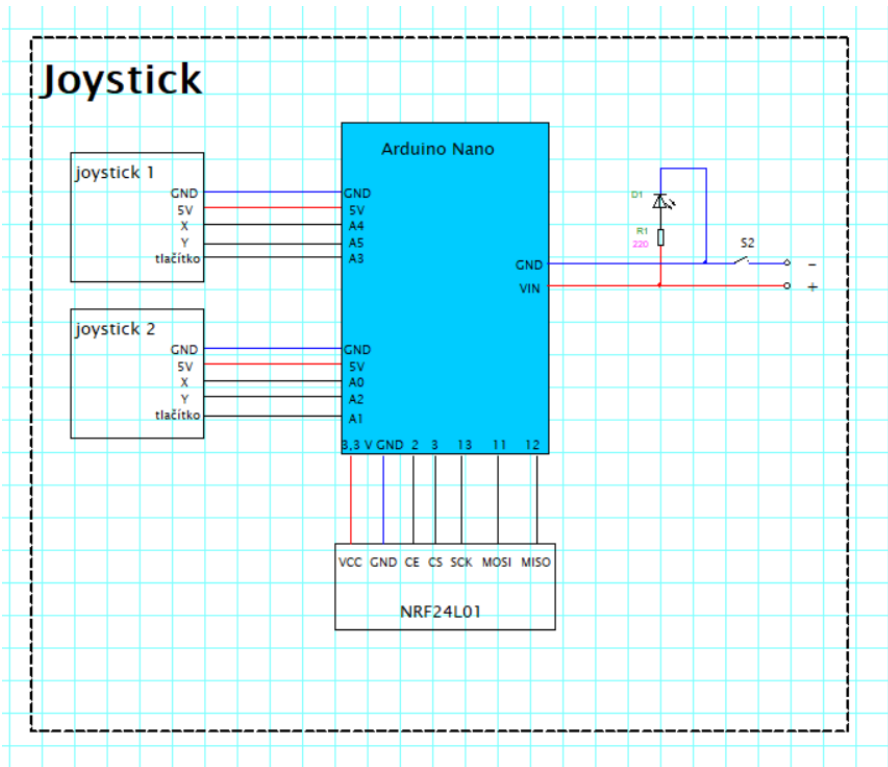
Joystick

Jedná se o samostatné zařízení sloužící k dálkovému ovládní pohybu ramene manipulátoru. Elektronika joysticku je uložena uvnitř anatomicky tvarovaného pouzdra. K jeho tvaru jsem se inspiroval od ovladačů u herních konzol. Pouzdro se skládá z korpusu a víka. Obě tyto části jsem vyrobil za použití technologie 3D tisku z materiálu PLA.



Obrázek 38: bezdrátový joystick

Elektronika joysticku se skládá z mikrokontroléru Arduino Nano, dvou joystick modulů, komunikačního modulu nrf24l01 a měniče. Jednotlivé moduly jsem připájel a pospojoval na univerzálním plošném spoji. Obvod je napájen 9V baterií, jejíž kladný pól je připojen na VIN pin Arduina.



Obrázek 39: schéma zapojení bezdrátového joysticku



Obrázek 40: elektronika bezdrátového joysticku

Způsoby řízení pohybu ramene manipulátoru

Manipulátor má celkem čtyři způsoby řízení pohybu ramene manipulátoru. Tyto způsoby řízení lze vybrat pomocí klávesnice na panelu operátora. Jejich nabídka se zobrazuje na LCD displeji formou menu.

Během řízení ramene robotického manipulátoru se nám zobrazují souřadnice jak na displeji, tak v počítačovém programu.

Rameno manipulátoru vždy urazí pouze definovanou vzdálenost. Tuto vzdálenost lze změnit na panelu operátora.

(1) Ovládání pomocí počítačového softwaru

Počítačový software běží neustále na Raspberry. Pokud však není aktivovaný režim ovládání přes tento software, jsou ovládací tlačítka zablokována. V tomto případě jsou tlačítka podbarvena šedou barvou. Při aktivování režimu ovládání přes počítačový software se tlačítka odemknou a změní barvu na zelenou. Zároveň se vypne funkce rozpoznávání barvy, aby byla zajištěna co největší plynulost běhu programu.

Po stisknutí tlačítka urazí daná osa manipulátoru vždy pouze přednastavenou vzdálenost. V tomto režimu ovládání lze tuto vzdálenost přenastavit také přímo v počítačovém softwaru.

(2) Ovládání přes bezdrátový joystick

Konstrukci a popis bezdrátového joysticku jsem popsal v samostatné kapitole. Nyní je pro nás důležité, že obsahuje dvě ovládací páčky, jejich nakloněním lze uvést rameno manipulátoru do pohybu. Každá páčka ovládá dvě osy. Jednu osu nakloněním ve vertikálním a druhou osu nakloněním v horizontálním směru. Velikou výhodou je možnost citlivějšího ovládání, jelikož vzdálenost pohybu se mění v závislosti na úhlu naklonění páčky joysticku. Naklonění má 3 intenzity, při kterých rameno manipulátoru urazí buď celou předdefinovanou vzdálenost, nebo $\frac{3}{4}$ vzdálenost a nebo $\frac{1}{2}$ vzdálenosti.



Obrázek 41: televizní ovladač s označenými tlačítky

(3) Bezdrátové ovládání přes televizní ovladač

Ovládání ramene manipulátoru přes dálkový IR televizní ovladač je velice jednoduché. Stačí ovladač namířit na krabici sovládací jednotkou a stlačit příslušné tlačítko. Následně rameno manipulátoru urazí předdefinovanou vzdálenost.

Tento druh ovládání jsem využíval především během vývoje, kdy jsem potřeboval zkoušet různé možnosti pohybu os a nebo jejich přesné polohování.

(4) Pohyb ramene manipulátoru na základě programu v paměti

Poslední možností je zvolit program uložený v paměti (ukázkový program). Po jeho vybrání přes terminál se spustí automatický pohyb podle předepsaného programu.

Ukázkový program

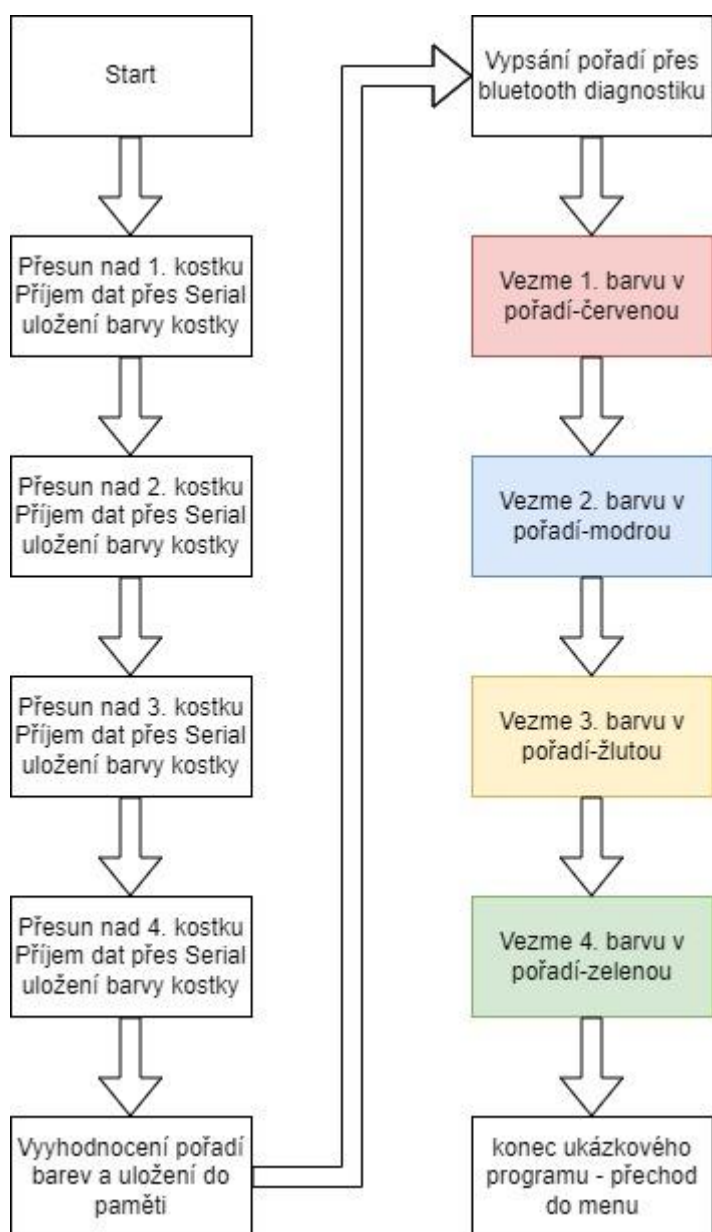
Při vytváření ukázkového programu jsem se chtěl co nejvíce zaměřit na jednoduchost a na autonomní řízení.

Manipulátor má za úkol naskenovat čtyři různě barevné kostičky náhodně rozestavěné na určitých místech a následně je na základě zadání vyskládat do komínku v určitém pořadí barev.

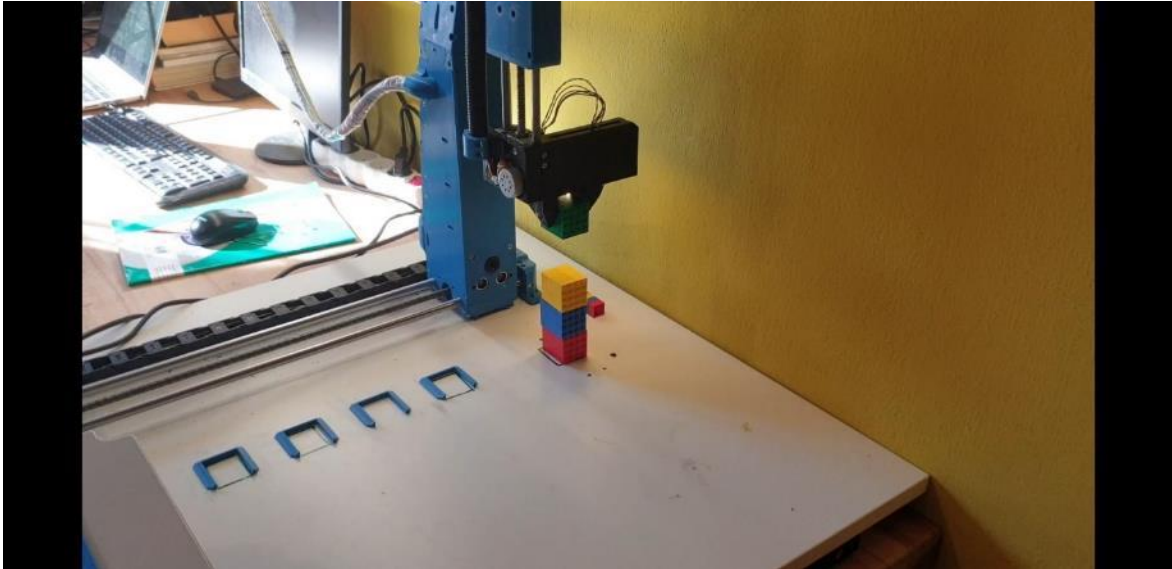
Kostičky jsou postaveny na předem definovaných místech v rámečcích, jelikož vlivem vibrací, které jsou způsobeny pohybem osy X, docházelo k pohybu kostiček po manipulační ploše.

Požadované pořadí: červená, modrá, žlutá zelená.

Manipulátor nejprve najede kamerou nad 1. kostku, vyšle žádost o odeslání barvy z počítačového programu. Barvu kostičky vyhodnotí program běžící na Raspbery a informaci předá Arduinu přes sériový port. Arduino tuto barvu uloží do paměti a najede nad další kostku v pořadí. Tak to opakuje se všemi čtyřmi kostkami. Následně vyhodnotí barvy na jednotlivých pozicích a seřadí je. Následně uchopí první kostku v pořadí, červenou, a umístí ji na určité místo na manipulační ploše. Poté uchopí druhou kostičku v pořadí, modrou, a umístí ji na červenou kostičku. Tímto způsobem pokračuje, dokud neposkládá do komínku všechny čtyři kostičky. Následně manipulátor přejde do režimu menu a zaparkuje se do nulového bodu.



Obrázek 42: blokové schéma běhu ukázkového programu



Obrázek 43: skládání kostiček ramenem manipulátoru

Software

Softwarovou část zařízení lze rozdělit na 3 části: software pro řídicí elektroniku, software pro bezdrátový joystick a software pro počítač, přes který je možné řídit manipulátor a zpracovávat obraz z kamer.

Všechny programy jsem psal v jazyce C++, ve vývojových prostředích Arduino a Processing.

Jelikož software nespadá do zadání mé práce, popíši tuto část obecně a zaměřím se pouze na důležité a zajímavé věci.

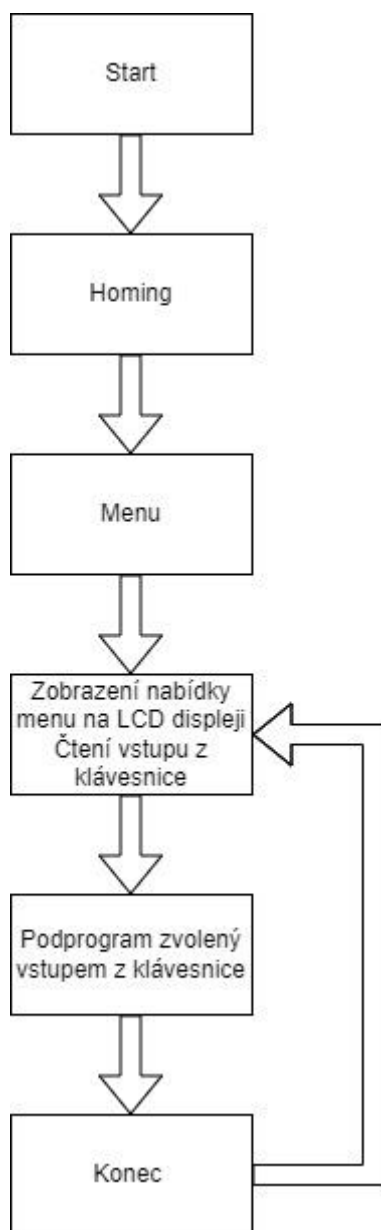
Arduino software

Řídicí program pro Arduino jsem napsal ve vývojovém prostředí Arduino IDE, v programovacím jazyce C++.

Tabulka 4: přehled funkcí pro řízení manipulátoru

| Název funkce | Vstupní parametry | Co funkce dělá |
|-------------------|---------------------------------------|--|
| Setup() | | inicializace |
| Loop() | | Hlavní smyčka programu. |
| Doraz() | | Kontroluje sepnutí koncových dorazů. |
| posunY()/X/Z | Směr pohybu, vzdálenost pohybu | Funkce pro pohyb osy. Volá další funkce - akceleraci, deceleraci, atd... |
| otaceniY1mm()/X/Z | Rychlost pohybu | Pohyb osy o 1mm. |
| otaceniY() | Vzdálenost pohybu, rychlost pohybu | Pohyb osy plnou rychlostí. |

| | | |
|-----------------------|--|--|
| otaceniY_akcelerace() | Vzdálenost pohybu, startovací rychlost, finální rychlost | Pohyb osy s postupným zvyšováním rychlosti. |
| otaceniY_Decelerace() | Vzdálenost pohybu, startovací rychlost, finální rychlost | Pohyb osy s postupným snižováním rychlosti. |
| roztahovaniCelisti() | | Pohyb osy E – čelisti směrem od sebe |
| stahovaniCelisti() | | Pohyb osy E – čelisti směrem k sobě |
| krok1() | | Posílání impulsů do motoru osy E |
| posunE() | Směr pohybu, vzdálenost pohybu | Funkce pro pohyb osy E |
| posunNaSouřadnice() | Souřadnice X,Y,Z,E | Funkce pro absolutní programování formou zadávání souřadnic. |
| homing() | | Zaparkuje všechny osy do jejich nulového bodu |
| prijemDat() | | Příjem dat pomocí modulu nrf24l01 z bezdrátového joysticku. |
| posunNaPrijem() | | Pohyb os na základě dat přijatých z bezdrátového joysticku. |
| IROvladani() | | Příjem dat z televizního ovladače a pohyb os. |
| menu() | | Volá ostatní funkce, zobrazuje nabídku na LCD, čte vstup z klávesnice. |
| ovIPC() | | Manipulátor v režimu ovládání přes počítačový program. |
| ovJoy() | | Manipulátor v režimu ovládání přes bezdrátový joystick. |
| ovOvl() | | Manipulátor v režimu ovládání přes bezdrátový ovladač od televize. |
| zakladLCD() | | Vykresluje univerzální šablonu na LCD displeji během různých režimů ovládání. |
| SeriovPortPrijem() | | Příjem dat přes sériový port. |
| seriovPortOdeslani() | | Odeslání dat přes sériový port. |
| diagnostika() | | Odeslání hodnoty různých proměnných přes Bluetooth. |
| progPam() | | Manipulátor v režimu samostatného pohybu na základě programu uloženého v paměti. |
| odnes1Kostku() | | Součást ukázkového programu. Vezme kostku na 1. pozici. |
| _1mistoShozu() | | Součást ukázkového programu. Položí kostku na spodní pozici v odkladovém prostoru. |



Obrázek 44: blokové schéma běhu Arduino programu

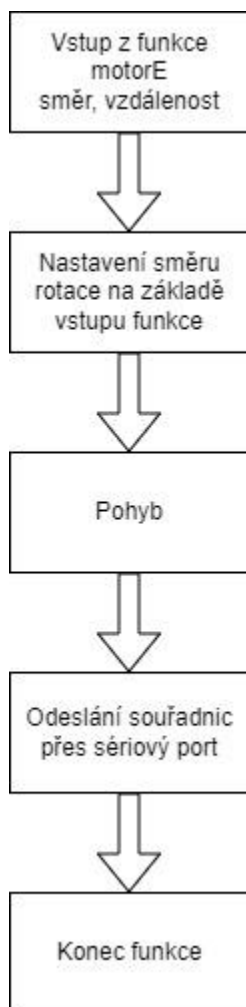
Řízení krokových motorů

Princip řízení krokového motoru s driverem TB6600 je velice jednoduchý. Driver je ovládán dvěma vodiči. Na vstup CW pošleme logickou 1 nebo logickou 0, podle toho, jakým směrem chceme, aby se motor otáčel. Na vstup CLK posíláme impulsy, na jejichž frekvenci závisí rychlost točení motoru.

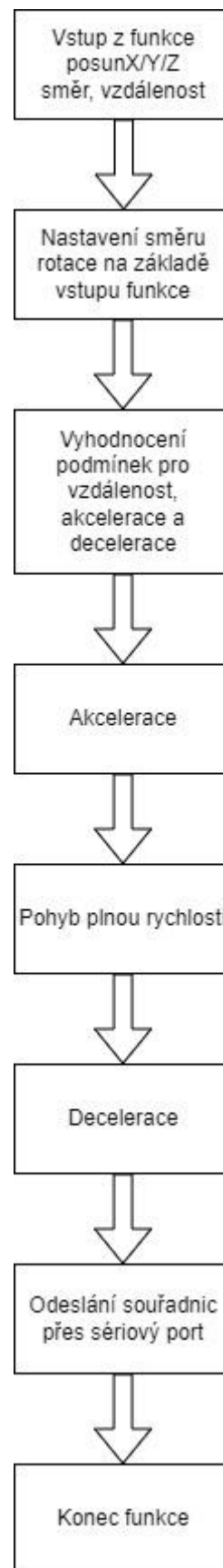
Původně jsem chtěl použít pro řízení krokových motorů knihovnu, ale ve funkci této knihovny běží cyklus, do kterého nelze zasahovat. Tudíž jsem při otáčení motoru nemohl hlídat stav koncových spínačů a provádět další úkony. Proto jsem si vytvořit funkce pro pohyb motorů napsat sám. To nakonec bylo mnohem těžší, než se původně

zdálo. Program zabývající se pohybem jednoho krokového motoru nakonec zabírá bezmála 200 řádek. Výsledkem je funkce jménem *posunX()* (příklad pro osu X), do níž stačí zadat směr a vzdálenost a sama se postará o pohyb motoru včetně postupného zrychlení a zpomalení. Funkce volí vzdálenost akcelerace a decelerace podle zadané vzdálenosti. Také počítá souřadnice a hlídá stav koncových spínačů. Tato funkce slouží k přírůstkovému programování pohybu ramene manipulátoru. Pro absolutní programování lze využít funkci *posunNaSouradnice()*.

U motoru osy E to bylo značně jednodušší. Jelikož se otáčí konstantní rychlostí, stačí pouze změnit pořadí posílaných impulsů.



Obrázek 45: blokové schéma řízení motoru osy E



Obrázek 46: blokové schéma řízení motorů os X, Y, Z

```

void posunY( String smerY, unsigned int vzdalenostY){//funkce pro pohyb motoru osy Y

    delay(100); //časová prodleva
    doraz(); //zkontroluje, zda je sepnutý některý z koncových dorazů

    if(smerY.equals("dopredu")){ //zjistí, jaký směr otáčení byl zadán do funkce
        smerYvoid="dopredu";//proměnná směru pro další operace
        digitalWrite(dirPinY, LOW);//nastaví směr rotace
    }
    if((smerY.equals("dozadu"))&&(homeY==false)){ //zjistí, jaký směr otáčení byl zadán do funkce
        smerYvoid="dozadu";//proměnná směru pro další operace
        digitalWrite(dirPinY, HIGH);//nastaví směr rotace
    }
}

```

Obrázek 47: ukázka funkce pro pohyb motoru posunY()

```

if((vzdalenostY>2)&&(vzdalenostY<6)){ //pokud je zadána vzdálenost v tomto rozsahu
    otaceniY_akcelerace(1, rychlostY, rychlostYZrychleni);//popojede o určitou část vzdálenosti pohybu se zvyšováním rychlosti
    if(homeY==true)return;
    otaceniY(vzdalenostY-2, rychlostY); //odečte vzdálenost akcelerace a decelerace od celkové zadan vzdálenosti a tuto dráhu ujede
    if(homeY==true)return;
    otaceniY_Decelerace(1, rychlostYZpomaleni, rychlostY);//popojede o určitou část vzdálenosti pohybu se snižováním rychlosti
}

```

Obrázek 48: ukázka vyhodnocování vzdálenosti pohybu akcelerace, plné rychlosti a decelerace

```

void otaceniY_akcelerace(unsigned int pocetMMA, unsigned int pozadovanaRychlostY, unsigned int startovaciRychlostY){ //otáčení motoru se zvyšováním rychlosti
    if((homeY==true)&&(smerYvoid.equals("dozadu"))){return; //pokud je doraz sepnut, vyskočí z funkce
    int odečetYa=(startovaciRychlostY-pozadovanaRychlostY)/(pocetMMA*25);//vypočítá odečet rychlosti pro každý krok

    for(int pocitadloPosunY=0;pocitadloPosunY<pocetMMA;pocitadloPosunY++){//samotný posun se zvyšováním rychlosti
    {
        for(int i=0;i<25;i++){//posun o 1mm (25 kroků)
            doraz();
            if((homeY==true)&&(smerYvoid.equals("dozadu"))){break; //pokud je doraz sepnut, vyskočí z funkce
            startovaciRychlostY=startovaciRychlostY-odečetYa;//výpočet prodlevy mezi impulsy

            digitalWrite(stepPinY, HIGH);//impulsy do driver -> otáčení motoru
            delayMicroseconds(startovaciRychlostY);
            digitalWrite(stepPinY, LOW);
            delayMicroseconds(startovaciRychlostY);
        }

        if((smerYvoid.equals("dozadu"))&&(homeY==false))souradniceY--;//podle směru přičte/odečte souřadnice
        if(smerYvoid.equals("dopredu"))souradniceY++; //podle směru přičte/odečte souřadnice
    }

    return 0;
}
}

```

Obrázek 49: ukázka funkce pro akceleraci krokového motoru

```

void homing() {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("homing...");
    doraz();
    delay(150);
    posunE("odSebe",3);

    if(homeZ==false){
        posunZ("dozadu",500);
        souradniceZ=0;
    }
    if(homeE==false){
        posunE("kSobe",70);
        souradniceE=0;
    }
    if(homeX==false){
        posunX("dozadu",500);
        souradniceX=0;
    }
    if(homeY==false){
        posunY("dozadu",500);
        souradniceY=0;
    }
}

```

Obrázek 50: ukázka funkce homing()

Homing – parkování do nulového bodu

Funkce homing() zaparkuje rameno manipulátoru do nulového bodu – výchozí souřadnice. Funguje tak, že postupně rozjede všechny osy a udržuje je v pohybu, dokud nezaznamená na dané ose sepnutí koncového spínač dorazu. Následuje pohybová sekvence další osy.

Komunikace s dálkovými ovladači

Manipulátor využívá pro vzdálené ovládání dva bezdrátové ovladače. Je jím bezdrátový joystick a televizní ovladač.

Nejpraktičtější volbou ovládání je bezdrátový joystick. Je jím možné i regulovat vzdálenost pohybu, která závisí na úhlu naklonění páčky joysticku. Ta má tři polohy, tedy tři rychlosti. Bezdrátový joystick vysílá data pomocí komunikačního modulu nrf24l01 a stejný komunikační modul v manipulátoru data přijímá. Joystick vysílá

dvojciferné číslo, kde první cifra značí osu a směr, druhá cifra vzdálenost pohybu. Takto se rameno manipulátoru může pohybovat o celou, tři čtvrtiny a nebo půlku předdefinované vzdálenosti.

```
////////////////////////////////////  
if(prijem==33){//podmínka pro přijatá data  
  diagnostika();//odešle přijatá data přes bluetooth  
  posunX("dopredu",delkaKroku);//posune osu  
  příjem=0;//vynuluje přijatá data  
}  
  
else if((prijem==32)&&(delkaKroku>1)){  
  diagnostika();  
  posunX("dopredu",delkaKroku*0.75);  
  příjem=0;  
}  
  
else if((prijem==31)&&(delkaKroku>1)){  
  diagnostika();  
  posunX("dopredu",delkaKroku*0.5);  
  příjem=0;  
},
```

Obrázek 51: ukázka podmínek vyhodnocování dat přijímaných z bezdrátového joysticku

U ovládání televizním ovladačem je to značně jednodušší, ale s omezením. Stačilo si načíst, který signál se odešle při zmáčknutí jakého tlačítka. Pak jsem tyto signály pouze přepsal do podmínek. Toto ovládání umožňuje pohyb pouze na předdefinovanou vzdálenost.

Komunikace přes sériový port z Arduina do ovládacího programu na počítači.

Vymyslet způsob, jak přes sériový port poslat několik čísel naráz a pak je na druhé straně patřičně rozdělit, nebylo úplně jednoduché. Možností je několik. Já jsem nakonec zvolil metodu pro mě nejjednodušší - kvantování. Odeslaná data rozdělím do intervalů s rozsahem 1000. Tudíž ke každé odeslané proměnné přičtu určitý násobek čísla 1000 a na druhé straně si program na základě podmínek určí, o jakou proměnnou se jedná a přičtenou hodnotu opět odečte.

```

void seriovyPortOdeslani() { //odeslání dat přes sériový port

    //zvětšení hodnoty odeslaných dat

    xData=souradniceX+1000;
    yData=souradniceY+2000;
    zData=souradniceZ+3000;
    eData=souradniceE+4000;
    enableOvladaniData=enableOvladani+5000;
    delkaKrokuData=delkaKroku+6000;

    //zvětšení hodnoty odeslaných dat

    Serial.println(xData);
    delay(1);
    Serial.println(yData);
    delay(1);
    Serial.println(zData);
    delay(1);
    Serial.println(eData);
    delay(1);
    Serial.println(enableOvladaniData);
    delay(1);
    Serial.println(delkaKrokuData);
    delay(1);

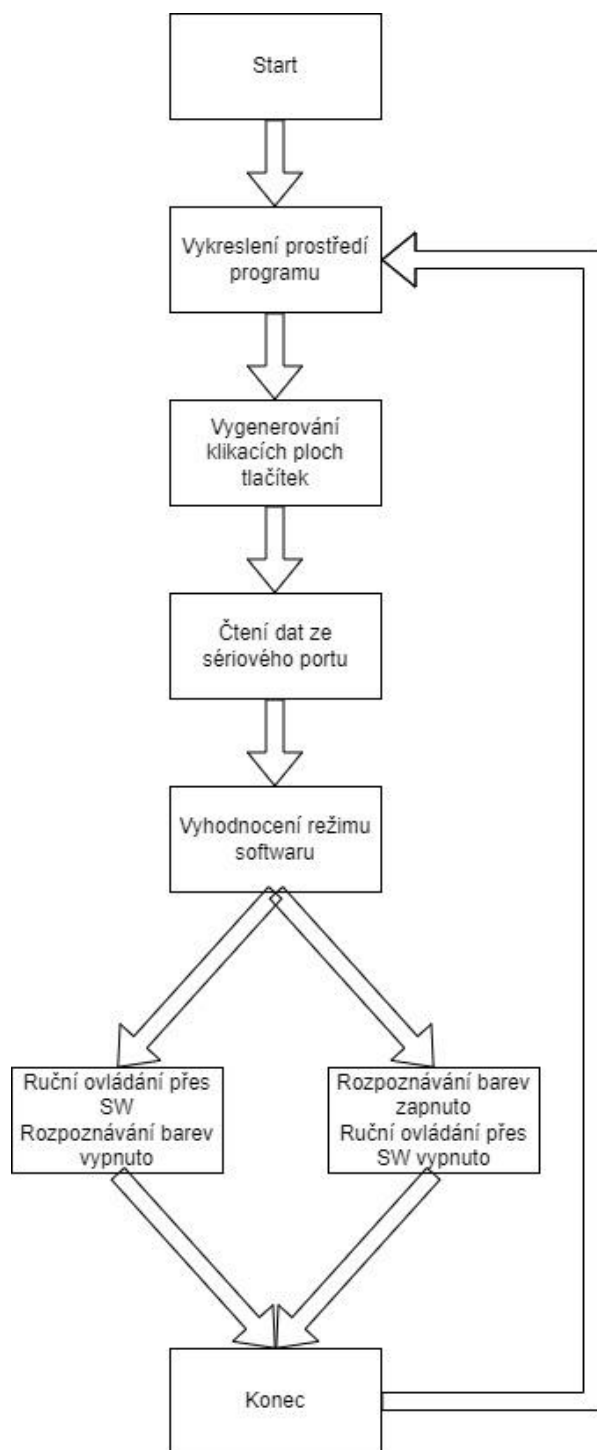
}

```

Obrázek 52: ukázka funkce pro odeslání dat přes sériový port

Vývojové prostředí Processing

Počítačový program pro ovládání ramene manipulátoru a zpracování obrazu z kamer jsem napsal ve vývojovém prostředí Processing, v jazyce C++. Toto prostředí používám velice rád, jelikož je jednoduché a pracuje na principu Arduino IDE. Veškerá grafika se zde programuje zadáváním souřadnic a hodnot RGB složek.



Obrázek 53: blokové schéma běhu počítačového programu pro ovládání

do režimu ovládání přes počítač.

Největší plochu okenní aplikace zabírají obrazovky z kamer. Levá kamera je dolní kamera umístěná ve středovém prvku ramene manipulátoru. Pravá kamera je horní

Vzhled a ovládací prvky

V levém horním rohu se nachází textové pole zobrazující souřadnice polohy ramene. Tyto souřadnice jsou Arduinem vyslány po každém dokončeném pohybu ramene manipulátoru.

Pod nimi se nachází virtuální ovládací tlačítka.

Tyto tlačítka jsem jako ovládací prvky zvolil z důvodu časové tísně, původně jsem chtěl vytvořit posuvné ovládací prvky. Princip ovládání funguje stejně jako u ovládání bezdrátovými ovladači. Zmáčknu tlačítko a rameno manipulátoru popojede o předdefinovanou vzdálenost.

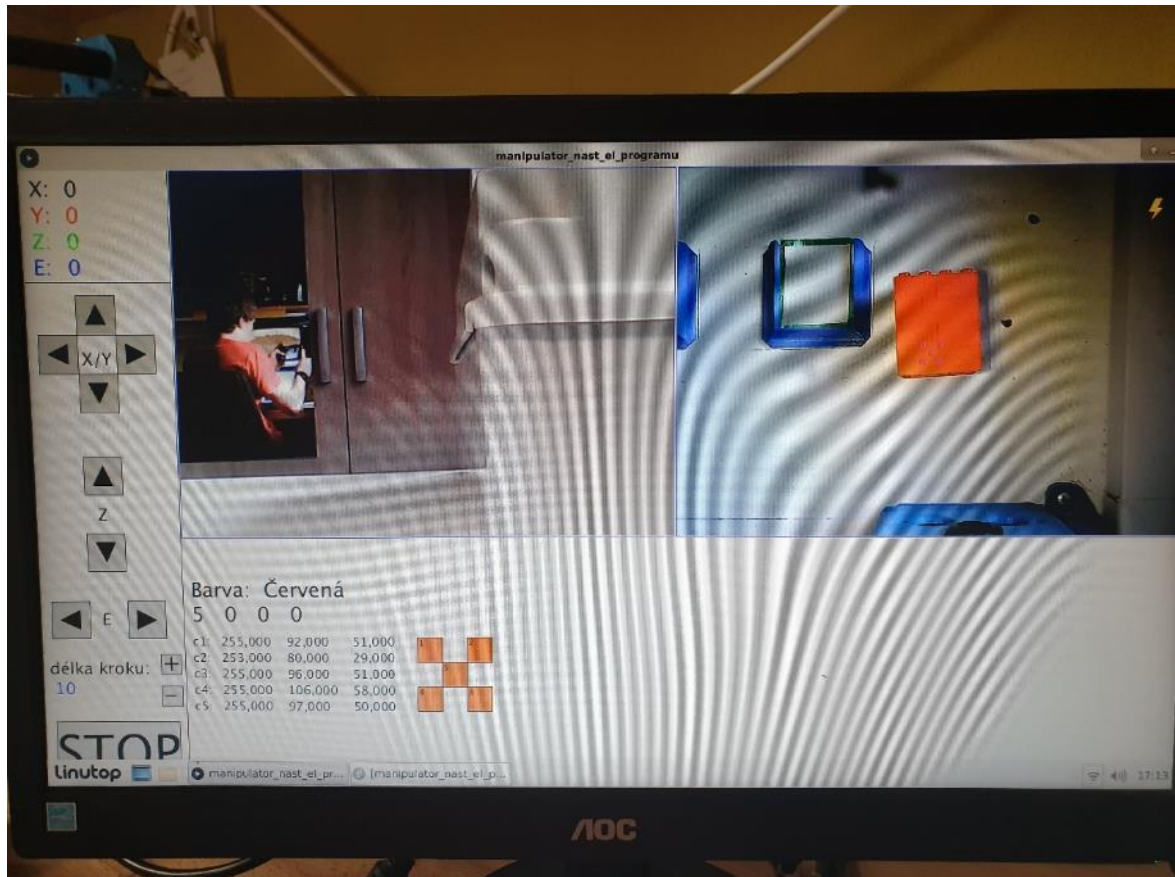
Nachází se zde tlačítka pro ovládání os X a Y, pod nimi ovládání osy Z a pod ním ovládání osy E. Tlačítka jsem koncipoval podobně jako na televizním ovladači.

Ve spodní části se nachází ovládací prvek pro změnu vzdálenosti pohybu ramene manipulátoru. Pod ním STOP tlačítko, v této fázi projektu jsem ho doposud neaktivoval.

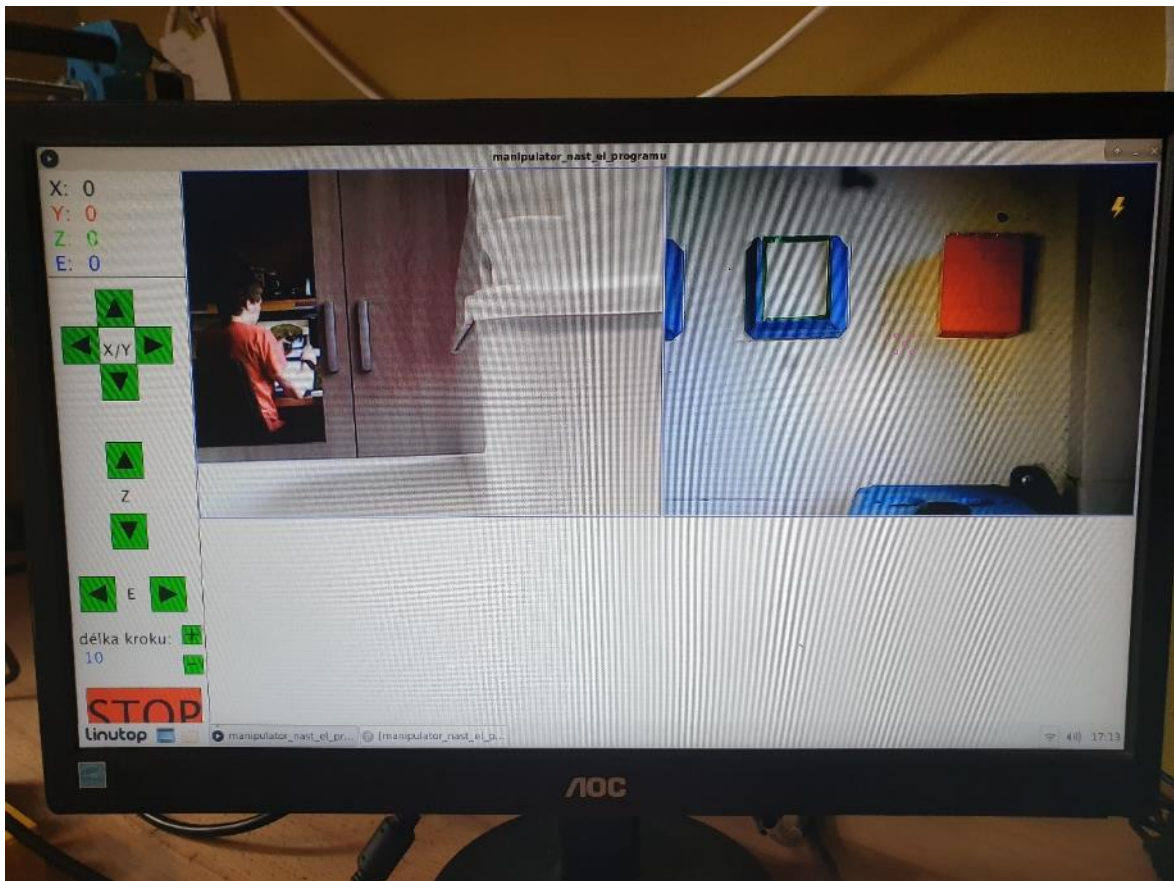
Všechny tyto ovládací prvky se aktivují až v momentě, kdy je manipulátor přepnut

kamera umístěná u čelisti manipulátoru. Tato kamera je také určena pro rozpoznávání barev v ukázkovém programu.

Pod kamerovými obrazovkami se nachází informační pole, kde se vypisují barvy a jednotlivé barevné složky pixelů z kamery. Tato funkce je při režimu ovládání přes počítač deaktivována.



Obrázek 54: počítačový program s rozpoznáváním barev a deaktivovaným ovládáním



Obrázek 55: počítačový program s aktivovaným ovládáním

Komunikace přes sériový port z ovládacího programu do Arduina.

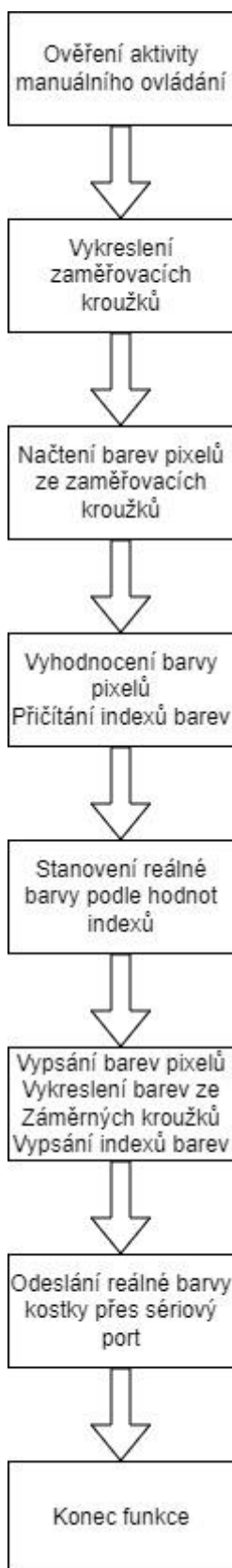
Z důvodu problému s konverzí datových typů a číselných soustav jsem nemohl použít stejnou metodu jako při posílání dat z Arduina. Metodou pokusů jsem zjistil, že mi jde posílat pouze čísla do hodnoty 127. Tudíž jsem komunikaci zjednodušil na minimum. Takže každému tlačítku a každé barvě, která byla rozpoznána, jsem přiřadil index, který je přes sériový port odeslán, je-li dané tlačítko stisknuto, či rozpoznána daná barva.

Příjem dat přes sériový port je jednoduchý. Data jsou rozdělena do jednotlivých kvantizačních hladin, které jsou násobkem čísla 1000. Přes podmínky si program rozpozná hladiny a hodnoty přidělí k příslušným proměnným.

```
if((prijemSerial>999)&&(prijemSerial<2000)){//kvantování přijatých dat
    xPrijem=prijemSerial-1000;
}
else if((prijemSerial>1999)&&(prijemSerial<3000)){
    yPrijem=prijemSerial-2000;
}
else if((prijemSerial>2999)&&(prijemSerial<4000)){
    zPrijem=prijemSerial-3000;
}
else if((prijemSerial>3999)&&(prijemSerial<5000)){
    ePrijem=prijemSerial-4000;
}
else if((prijemSerial>4999)&&(prijemSerial<6000)){
    enableOvladaniPrijem=prijemSerial-5000;
}

else if((prijemSerial>5999)&&(prijemSerial<7000)){
    delkaKrokuPrijem=prijemSerial-6000;
}
else if((prijemSerial>6999)&&(prijemSerial<8000)){
    posliBarvu=prijemSerial-7000;
}
```

Obrázek 56: ukázka zpracování dat přijatých přes sériový port



Funkce pro rozpoznávání barev pomocí kamery

Vytvořit funkční detekci barev objektů z kamerového obrazu, bylo pro mě jedním z největších cílů mé práce. Z důvodu časové tísně jsem zvolil co nejjednodušší cestu. Program pomocí funkce `get()` uloží do proměnných barevné složky pěti pixelů uprostřed kamerové obrazovky, označených fialovým kroužkem. Následně podle podmínek vyhodnotí barvu každého z nich. Pokud se alespoň 4 z 5 barev pixelů shodují, program rozhodne, že se jedná o danou barvu.

Obrázek 57: blokové schéma funkce pro rozpoznávání barev

```
//podminky pro určení barev
if((red(c1)>180)&&(red(c1)<256)&&(green(c1)>50)&&(green(c1)<135)&&(blue(c1)>-1)&&(blue(c1)<110)) barvaPixeluCervena++; //c1 cervena
if((red(c1)>-1)&&(red(c1)<110)&&(green(c1)>10)&&(green(c1)<160)&&(blue(c1)>105)&&(blue(c1)<270)) barvaPixeluModra++; //c1 modra
if((red(c1)>150)&&(red(c1)<256)&&(green(c1)>120)&&(green(c1)<256)&&(blue(c1)>-1)&&(blue(c1)<100))barvaPixeluZluta++; //c1 zluta
if((red(c1)>-1)&&(red(c1)<100)&&(green(c1)>70)&&(green(c1)<200)&&(blue(c1)>-1)&&(blue(c1)<110))barvaPixeluZelena++; //c1 zelena
// if((red(c1)>100)&&(red(c1)<200)&&(green(c1)>150)&&(green(c1)<256)&&(blue(c1)>-1)&&(blue(c1)<140))barvaPixeluSvetleZelena++; //c1 svetle zelena
```

Obrázek 58: podmínky pro vyhodnocování barev

```
if(barvaPixeluCervena>3){

    text("Barva: Červená", (width/8)+10,height/2+120);
    barvaKostky=1;

}

else if(barvaPixeluModra>3){

    text("Barva: Modrá", (width/8)+10,height/2+120);
    barvaKostky=2;

}

else if(barvaPixeluZluta>3){
    text("Barva: Žlutá", (width/8)+10,height/2+120);
    barvaKostky=3;

}

else if(barvaPixeluZelena>3){
    text("Barva: Zelená", (width/8)+10,height/2+120);
    barvaKostky=4;
}

}
```

Obrázek 59: podmínky pro vyhodnocování shody barvy skenovaných pixelů

Zároveň překresluje barvy pixelů, přepisuje potvrzenou barvu, barevné složky a index počtu shod do pole pod kamerovými obrazovkami. Tyto informace slouží především k vývojové kalibraci snímání barev vůči rozdílným světelným podmínkám. Index potvrzené barvy je odeslán až v momentě, kdy program obdrží žádost od Arduina.


```

if(aktX>94) i=33;
else if((aktX>70) && (aktX<95)) i=32;
else if((aktX>10) && (aktX<61)) i=31;

else if(aktX<-94) i=23;
else if((aktX<-70) && (aktX>-95)) i=22;
else if((aktX<-10) && (aktX>-61)) i=21;

else if(aktY>94) i=13;
else if((aktY>70) && (aktY<95)) i=12;
else if((aktY>10) && (aktY<61)) i=11;

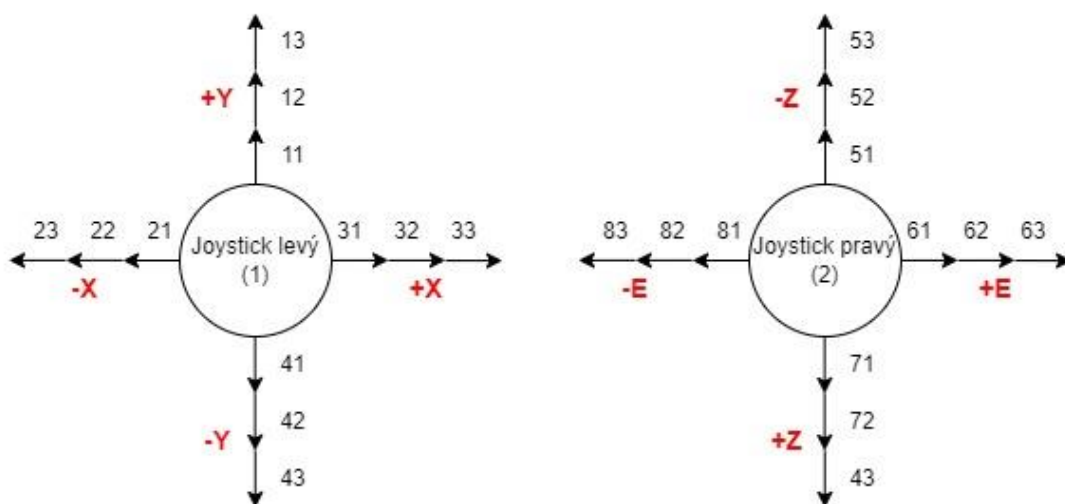
else if(aktY<-94) i=43;
else if((aktY<-70) && (aktY>-95)) i=42;
else if((aktY<-10) && (aktY>-61)) i=41;

```

Obrázek 60: ukázka podmínek vyhodnocujících náklon páčky joysticku

Software joysticku

Program pro joystick je velice jednoduchý. Oba páčkové ovladače jsou vyrobeny z potenciometrů, jejichž odpor se mění s náklonem páčky. Tím se na vstup Arduina dostává různá hodnota napětí. Ta je přes funkci `map()` převedena na rozsah -100 až +100. Podmínkami se vyhodnocuje míra náklonu a podle ní je odeslána patřičné dvojčíferné číslo. První cifra označuje směr a osu, druhá cifra označuje vzdálenost.



Obrázek 61: obrázek zobrazující odeslaná data v závislosti na náklonu páčky joysticku

Závěr

Při dokončení práce jsem v zásadě dosáhl zamýšlených výsledků. I přes to, že to byla velice strastiplná cesta a neustále mě zrazovala technika ale i vlastní hlava. Nad projektem jsem strávil řádově stovky hodin. Z velké části bylo časově náročné hledání příčin nefunkčnosti od vývojového prostředí až po produkční problémy spojené s CNC frézou či 3D tiskárnou. S velkými obtížemi jsem se potýkal během uvádění kompletního zařízení do chodu, tedy jeho funkčního oživení. Některé součástky nebo funkční celky nefungovaly podle předpokladu. Někdy to bylo způsobeno špatným zapojením, nefunkčností součástky, mou neznalostí některých praktických principů elektroniky a mechaniky.

Pokusím se zmínit problémy, které mne nejvíce mrzely a které nejvíce hanobí vzhled a finální zpracování manipulátoru a řídicí elektroniky. Jedním z těchto problémů byl pokus o oživení krokového motoru osy E, který mi nechtěl fungovat. Motor jsem mnohokrát vyměňoval, přepojoval jsem obvody a zkoušel zapojení na NKP. Po týdnu hledání příčin jsem zjistil, že mám fyzickou záměnu mezi dvěma řídicími vodiči. Velice mě mrzela stabilita ramene manipulátoru, která vůbec neodpovídá mým představám. Tato nestabilita je způsobena prohýbáním vodících tyčí, především na ose X. Dále má detailní neznalost signálového rozhraní USB, způsobila, že jsem nepoužil pro přenos stíněný kabel. Já jsem při připojování kamer použil kabel nestíněný a nemohl jsem kamery zprovoznit a jeho výměna byla časově náročná.

Dále při ožívování komunikačního modulu nrf24l01 jsem bohužel zjistil, že na základní desce je nějaký problém a komunikační modul jsem musel připájet přímo na Arduino. Také se ukázal problém s vibracemi, které způsobovaly povolování šroubových spojů, řešení jsou samojisticí matice, které nejsou bohužel použity.

Další konstrukční vadou je krabice, která zasahuje do manipulačního prostoru. V tomto případě bohužel nebyla jiná možnost umístění. Také jsem nevhodně navrhnul pouzdra ložisek osy Z a kvůli jejich tvaru nefungovalo spínání dorazu osy Y. Musel jsem zde dolepit pomocnou plošku, což je dost nevzhledné. Při návrhu čelisti jsem zapomenul na spínač koncového dorazu a nakonec jsem ho musel dodatečně přilepit. Při návrhu jedné ze základen osy X jsem ji nezáměrně udělal o 1mm nižší, následně jsem ji musel podložit. V tomto dílu mi také vznikla mírná chyba v uložení ložiska, a tím nebyla závitová tyč v ose s tyčemi vodícími. Proto do tohoto místa nebylo rameno manipulátoru schopno dojet. Musel jsem ložisko zbrousit a podložit. Při řešení všech potíží při stavbě manipulátoru jsem došel k poučce, kterou jsem si napsal na titulní stranu svého projektového poznámkové bloku, a ta zní: ***„I ten nejsložitější problém má většinou to nejjednodušší řešení“***

Bohužel jsem kvůli časové tísni nestihl rozšířit některé části projektu. Především v softwarové části, kde jsem chtěl udělat sofistikovanější program pro zpracování obrazu z kamer, který by nejen rozpoznal barvy, ale třeba i tvar objektu, četl QR kódy a byl schopný samostatně identifikovat objekty a mechanicky s nimi pracovat.

Dále jsem nestihl uvést do provozu některé moduly, jako je čtečka SD karet, na níž jsem zamýšlel ukládat textový soubor, který by obsahoval obslužný program procesní

úlohy manipulátoru. Ultrazvukový měřič vzdálenosti jsem bohužel také nestihl využít. Dal by se krásně využít ke skenování pracovní plochy manipulátoru. Stejně tak jsem nevyužil senzor uvnitř čelisti, který jsem přelepil gumou, abych zvýšil adhezi čelisti při uchopování kluzkých kostiček. Dalo by se říct, **že finální funkčnost projektu je pouze takové brouzdání po povrchu možností, které by se na tomto projektu daly realizovat.**

Každopádně můžu říci, že mi tento projekt přinesl opravdu mnoho. Zdokonalil jsem se v mnoha oblastech techniky. Především v návrhu 3D dílů konstrukce zařízení, v programování, pochopení funkčnosti některých elektronických částí a modulů, konstrukce lineárních pojezdů. Velice přínosné pro mě bylo, že jsem dovedl vytvořit jednoduchou funkci pro rozpoznání barvy kostiček z několika pixelů. Tato funkce mi udělala opravdu radost. Ale jedním z největších přínosů pro mě bylo frézování plošného spoje pomocí CNC frézy. I přes to, že bylo s tímto nízkonákladovým obráběcím strojem opravdu náročné, byl jsem z výsledku do jisté míry mile překvapen. Pochopil jsem, že je to velice jednoduchý a efektivní způsob, jak si v domácích podmínkách vyrábět desky plošných spojů. Po nutných úpravách se touto technologií dají vyrábět i desky na téměř profesionální úrovni. Jakmile zvládnou maturitu, budu se této problematice dále věnovat.

Citované zdroje

[1] <https://cs.wikipedia.org/wiki/Robot>

Seznam obrázků

| | |
|--|----|
| Obrázek 1: současný stav projektu, pohled na celek | 7 |
| Obrázek 2: popis os pohybu ramene manipulátoru..... | 8 |
| Obrázek 3: blokové schéma logické části řízení | 9 |
| Obrázek 4: celkové elektrické schéma | 10 |
| Obrázek 5: tisk prototypu na tiskárně Ender 3 | 11 |
| Obrázek 6: skládání prototypů jednotlivých dílů | 12 |
| Obrázek 7: popis tištěných dílů - pohled z boku | 12 |
| Obrázek 8: popis tištěných dílů - pohled zezadu..... | 13 |
| Obrázek 9: první prototyp základny osy X | 14 |
| Obrázek 10: pouzdro ložisek osy X s podvozkem..... | 15 |
| Obrázek 11: pouzdra ložisek osy X se zámkou | 17 |
| Obrázek 12: úchyt motoru osy Y s přechodem kabelového svazku do kabelové chráničky..... | 18 |
| Obrázek 13: vnitřek pouzder ložisek osy Z..... | 19 |
| Obrázek 14: pohled na hlavu / čelist z boku | 20 |
| Obrázek 15: pohled na hlavu/čelist zespoda | 20 |
| Obrázek 16: pohyblivý díl čelisti..... | 21 |
| Obrázek 17: osa Z s kabelovým svazkem připravená na montáž | 22 |
| Obrázek 18: obrázek z kompletace zařízení | 22 |
| Obrázek 19: vnitřek středního dílu..... | 23 |
| Obrázek 20: pouzdro ložisek osy X - vnitřek (bez trapézových matic) | 24 |
| Obrázek 21: pohybová soustava osy E..... | 25 |
| Obrázek 22: zvětšené elektrické schéma ramene manipulátoru..... | 26 |
| Obrázek 23: kabely vedoucí do krabice..... | 28 |
| Obrázek 24: zvětšené schéma základní desky | 30 |
| Obrázek 25: osazená základní deska | 31 |
| Obrázek 26: cesty na základní desce - obrázek z EAGLE | 32 |
| Obrázek 27: screenshot z programu Rapid PCB..... | 33 |
| Obrázek 28: screenshot z programu Universal Gcode Sender..... | 33 |
| Obrázek 29: foto z frézování cest na CNC | 34 |
| Obrázek 30: Hotová základní deska včetně nedokonalostí z výroby | 34 |
| Obrázek 31: Zapojení napájení na základní desce a zapojení vstupních/výstupních pinů | 35 |
| Obrázek 32: popis osazení základní desky (vysvětlivky v tabulce 3)..... | 36 |
| Obrázek 33: podrobné schéma zapojení krokových motorů | 39 |
| Obrázek 34: zapojení krokových motorů při jejich prvním testování před stavbou zařízení..... | 39 |
| Obrázek 35: popis obsahu krabice (vysvětlivky v tabulce 3)..... | 40 |
| Obrázek 36: zvětšené schéma panelu..... | 42 |
| Obrázek 37: popis ovládacího panelu | 43 |
| Obrázek 38: bezdrátový joystick | 44 |
| Obrázek 39: schéma zapojení bezdrátového joysticku | 45 |
| Obrázek 40: elektronika bezdrátového joysticku..... | 45 |
| Obrázek 41: televizní ovladač s označenými tlačítky | 47 |
| Obrázek 42: blokové schéma běhu ukázkového programu | 49 |
| Obrázek 43: skládání kostiček ramenem manipulátoru | 50 |
| Obrázek 44: blokové schéma běhu Arduino programu | 52 |
| Obrázek 45: blokové schéma řízení motoru osy E | 54 |

| | |
|---|----|
| Obrázek 46: blokové..... | 54 |
| Obrázek 47: ukázka funkce pro pohyb motoru posunY()..... | 55 |
| Obrázek 48:ukázka vyhodnocování vzdálenosti pohybu akcelerace, plné rychlosti a decelerace .. | 55 |
| Obrázek 49: ukázka funkce pro akceleraci krokového motoru..... | 55 |
| Obrázek 50: ukázka funkce homing()..... | 56 |
| Obrázek 51: ukázka podmínek vyhodnocování dat přijímaných z bezdrátového joysticku | 57 |
| Obrázek 52: ukázka funkce pro odeslání dat přes sériový port | 58 |
| Obrázek 53: blokové schéma běhu počítačového programu pro ovládání | 59 |
| Obrázek 54: počítačový program s rozpoznáváním barev a deaktivovaným ovládáním..... | 60 |
| Obrázek 55: počítačový program s aktivovaným ovládáním | 61 |
| Obrázek 56: ukázka zpracování dat přijatých přes sériový port | 62 |
| Obrázek 57: blokové schéma funkce pro rozpoznávání barev | 63 |
| Obrázek 58: podmínky pro vyhodnocování barev | 64 |
| Obrázek 59: podmínky pro vyhodnocování shody barvy skenovaných pixelů | 64 |
| Obrázek 60: ukázka podmínek vyhodnocujících náklon páčky joysticku | 65 |
| Obrázek 61: obrázek zobrazující odeslaná data v závislosti na náklonu páčky joysticku | 65 |

Seznam tabulek

| | |
|--|----|
| Tabulka 1: vysvětlivky popisu tištěných dílů | 13 |
| Tabulka 2: zapojení I/O Arduina..... | 37 |
| Tabulka 3: vysvětlivky popisu obsahu krabice | 40 |
| Tabulka 4: přehled funkcí pro řízení manipulátoru..... | 50 |

Přílohy

Na internetu

- 1) Prezentační video - <https://www.youtube.com/watch?v=wDfgv03POno&t=793s>

V samostatných souborech

- 2) 3D modely
- 3) Kompletní elektrické schéma
- 4) Rozvržení základní desky
- 5) Schéma základní desky
- 6) Zdrojový kód pro Arduino
- 7) Zdrojový kód pro počítačový program