|  |  |  |
| --- | --- | --- |
| Logo__SSPU_2016_Barva | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **FabScan 3D scanner** | | |
| Josef Galvas | | |
| C:\Users\Josef\Desktop\logo.jpg | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2018/2019 | |

#### Poděkování

*Děkuji Ing. Petru Grussmannovi za pomoc při vedení ročníkové práce.*

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

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

V Opavě 31. 12. 2018

*podpis autora práce*

**ANOTACE**

Cílem projektu bylo vytvořit funkční 3D scanner, který by umožňoval skenování 3D objektů a následné zobrazení na počítači. Projekt využívá open-source software FabScan, který dokáže z nasnímaných fotografií sestavit 3D objekt a běží na OS Linux. Samotný objekt je umístěn na podstavci otáčeném pomocí krokovacího motorku. Motor je ovládán ovladačem Polulu A4988 a ten je připojen k mikrokontroléru Arduino Nano, v němž je nahrán samotný kód. Na snímaný předmět svítí přímkový laser a ten v určitých intervalech zachytává webkamera a vyfocené snímky posílá do programu FabScan přes USB vstup. Na obrazovce lze vidět postupné skenování obrazu a výsledkem je hotový objekt, který lze naformátovat do STL podoby určené pro 3D tiskárny.

**Klíčová slova:** scanner, open-source, mikro-kontrolér, ovladač, formát

**ANNOTATION**

The goal of the project was to create a functional 3D scanner that would allow the scanning of 3D objects and subsequent display on the computer. The project uses FabScan open source software to build a 3D object from scanned photos and run on Linux. The object itself is placed on a pedestal rotated by a stepper motor. The engine is controlled by the Polulu A4988 which is connected to the Arduino Nano microcontroller. A direct laser is lit on the subject, and at certain intervals capture the webcam and take the captured images to the FabScan via USB input. You can see a progressive image scan on the screen, resulting in a finished object that can be formatted into the STL format for 3D printers.

Keywords: scanner, open source, micro-driver, driver, format

OBSAH

[Úvod 5](#_Toc534468286)

[1 Teoretická a metodická východiska 6](#_Toc534468287)

[1.1 Princip Software 6](#_Toc534468288)

[1.1.1 Testování krokového motoru 6](#_Toc534468289)

[1.1.2 Schéma zapojení 8](#_Toc534468290)

[1.1.3 Testovací zdrojový kód 8](#_Toc534468291)

[2 Využité technologie 9](#_Toc534468292)

[2.1 Hardware 9](#_Toc534468293)

[2.1.1 Arduino Nano 9](#_Toc534468294)

[2.1.2 Krokový motor 9](#_Toc534468295)

[2.1.3 Line laser 10](#_Toc534468296)

[2.1.4 Webkamera 10](#_Toc534468297)

[2.1.5 Polulu A4988 – ovladač motoru 11](#_Toc534468298)

[2.2 Software 12](#_Toc534468299)

[2.2.1 FabScan100 12](#_Toc534468300)

[2.2.2 FabScanPi 12](#_Toc534468301)

[2.3 Vývojová prostředí 12](#_Toc534468302)

[2.3.1 PyCharm ?? 12](#_Toc534468303)

[3 Způsoby řešení a použité postupy 13](#_Toc534468304)

[3.1 Finální kompletování 13](#_Toc534468305)

[3.1.1 Pájení na desku 13](#_Toc534468306)

[3.1.2 Sestavení boxu 14](#_Toc534468307)

[3.2 Řešení problémů 14](#_Toc534468308)

[3.2.1 Problém s instalací Linuxu 14](#_Toc534468309)

[3.2.2 Webkamera 14](#_Toc534468310)

[3.2.3 Odpálení Arduina 14](#_Toc534468311)

[4 Výsledky řešení, výstupy, uživatelský manuál 15](#_Toc534468312)

[Závěr 16](#_Toc534468313)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 17](#_Toc534468314)

[Seznam příloh 18](#_Toc534468315)

Úvod

V úvodu této práce bych chtěl vysvětlit, proč jsem si vybral zrovna tento projekt a co všechno mě k tomu vedlo. Protože již od mala se mi hodně líbily různé programy na modelování a tvorbu všelijakých objektů, rozhodl jsem se tomu věnovat trochu svého času a pokusil jsem se najít něco, co by mě zaujalo natolik, abych z toho udělal svou závěrečnou práci. O projektu FabScan jsem se náhodou dozvěděl od mého učitele z počítačových sítí, když nám dával různé návrhy na závěrečné ročníkové práce. Tento projekt mě zaujal hned na první pohled, a proto jsem si řekl, že se do toho pustím a zkusím si vytvořit svůj vlastní 3D scanner doma a použít ho zároveň i jako svou závěrečnou práci.

Před začátkem realizace svého projektu jsem musel zvážit, kterou distribuci programu zvolím. Na internetu je celá řada různých verzí a já jsem si nakonec vybral tu, jenž v sobě obsahuje vše, co potřebuji ke kompletaci a to od hardwarových součástí, až po konečný software. Původní software vytvořil Francis Engelmann, který na něm pracoval déle než pět let. Nyní se tomuto projektu věnuje více lidí a postupně se jej snaží vylepšovat a dolaďovat. Nejznámější vývojář, který momentálně pracuje na vývoji se jmenuje Mario Lukas. Vytvořil nejnovější verzi zařízení – FabScanPi-Server, založenou na Raspberry Pi. Zařízení je ovládáno pomocí internetového prohlížeče a také využívá lepší snímací techniky. Já jsem vyzkoušel obě verze. FabScan100, která běží na operačním systému Ubuntu 13.04. Celý program i s OS je zabalený v instalačním souboru ISO a obsahuje pár další programů na úpravu objektů, proto jsem zprvu zvolil právě tuto verzi. Zkoušel jsem samozřejmě také druhou verzi, kterou je podrobněji popsána později.

První část mé dokumentace se zaměřuje především na využité technologie a použité hardwarové součásti, které jsem potřeboval během řešení svého projektu. V další, rozsáhlejší části, vysvětlím, jak celý projekt funguje a co je jeho výsledkem. Jsou zde uvedeny také případné možné vylepšení do budoucna.

# Teoretická a metodická východiska

## Princip Software

Protože je FabScan open-source software, existují různé verze projektů s různými druhy řešení. Já jsem vyzkoušel dvě. První verze běží na OS Linux a jmenuje se FabScan100. Lze jej stáhnout jako instalační soubor ISO, který obsahuje operační systém Linux 13.04 a samotný program pro skenování objektů. Nejprve jsem zkoušel stáhnout a nainstalovat tento instalační soubor jako virtuální počítač přes VIrtualBox od firmy Oracle. Toto řešení se později ukázalo jako nevhodné, protože do virtuálního počítače nešlo připojit USB port pro webovou kameru ani pro mikro-kontrolér Arduino. Z tohoto důvodu jsem musel zvolit jinou možnost. A tou byl můj starý notebook značky Dell 6300. Zde se vyskytl další problém, nefunkční disk. Po vyřešení problému s diskem jsem konečně mohl začít se samotnou instalací. Nejprve jsem si vytvořil bootovací USB flash disk, který obsahoval instalační soubor. Následně jsem jej použil pro nahrání OS do svého počítače. Instalace proběhla úspěšně a já jsem mohl pokračovat v dalších částech projektu.

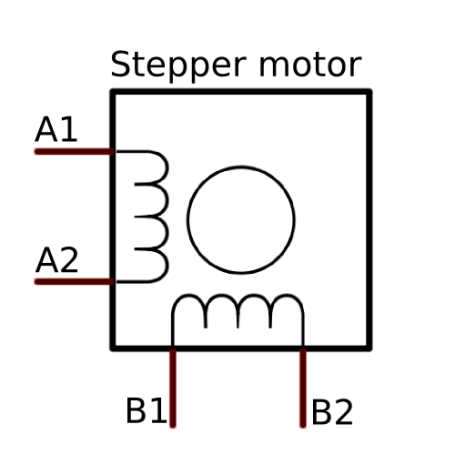
Nyní zde zmíním ještě druhou možnost, kterou jsem vyzkoušel a tou je FabScanPi, jenž umožňuje ovládat scanner přes webový prohlížeč. Tato verze programu je dostupná zatím pouze pro OS Linux a je napsaná v programovacím jazyce Python. Webové uživatelské rozhraní je napsáno pomocí JavaScriptu[[1]](#footnote-1) a HTML 5[[2]](#footnote-2) používající WebJL[[3]](#footnote-3). FabScanPi je také open-source projekt vydaný pod [GNU General Public License V2](https://www.gnu.org/licenses/gpl-2.0.html).

Obě verze používají pro scanner podobné hardwarové součástky. Webovou kameru, krokový motorek, přímkový laser, ovladač motoru a mokro-kontrolér Arduino. Princip hardwarových částí je vysvětlen v kapitole Hardware.

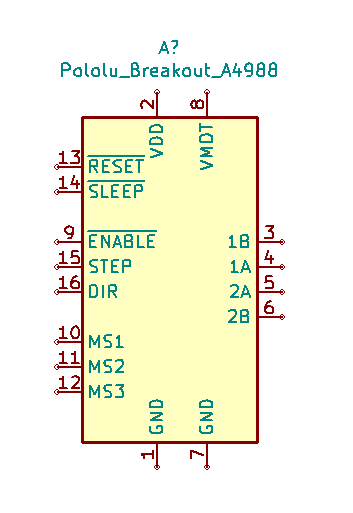
### Testování krokového motoru

Podle schématu zapojení jsem se pokusil zapojit a vyzkoušet motor na nepájivém poli. Krokový motor je připojen pomocí čtyř drátů k ovladači Polulu – A4988. Na jednotlivé piny na ovladači se postupně připojují drátky od motoru podle označení A1 – B2, podle přiloženého obrázku. Různý motor může mít různé zapojení, takže je třeba najít vhodné schéma pro konkrétní typ motoru. Motor musí být poháněn externím zdrojem. Jako zdroj však nelze použít výstup 5V z Arduina. Při tomto zapojení Arduino začne hořet. Pro začátek jsem tedy použil univerzální laboratorní elektrický zdroj. Poté jsem připojil samotné Arduino a nahrál do něj jednoduchý kód pro test, zda se motor točí. K propojení je potřeba čtyři piny. Dva jsou pro ovládání pohybu motoru a další dva pro napájení ovladače motoru.

Všechny součásti jsem zkoušel a testoval na nepájivém poli. Ukázka zapojení a kód je popsán v kapitole 1.1.2.

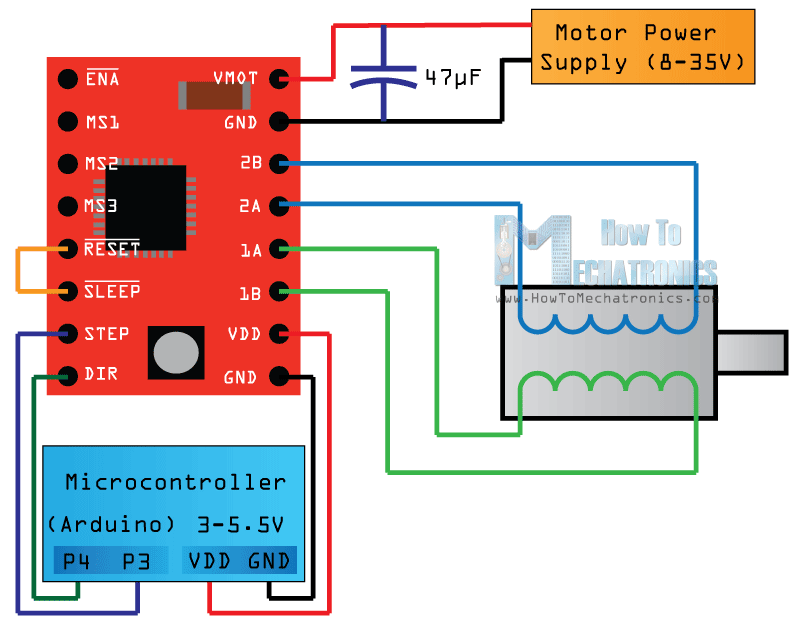


Obrázek č. 1. - Krokový motor



Obrázek č. 2. - Polulu A4988

### Schéma zapojení



Obrázek č. 3. - Schéma zapojení

### Testovací zdrojový kód

const int stepPin **=** 3**;** /\*deklarace pinů

const int dirPin **=** 4**;** \*/

void setup**()** **{**

// Nastavení dvou výstupních pinů

pinMode**(**stepPin**,**OUTPUT**);**

pinMode**(**dirPin**,**OUTPUT**);**

**}**

void loop**()** **{**

digitalWrite**(**dirPin**,**HIGH**);** // Umožňuje točit motorem v určitém směru

// Cyklus „for“, který zajistí aby se motor otočil o jednu otáčku

**for(**int x **=** 0**;** x **<** 200**;** x**++)** **{**

digitalWrite**(**stepPin**,**HIGH**);**

delayMicroseconds**(**500**);**

digitalWrite**(**stepPin**,**LOW**);**

delayMicroseconds**(**500**);**

**}**

delay**(**1000**);** // One second delay

digitalWrite**(**dirPin**,**LOW**);** //Změní rotaci směru

// Cyklus, který udělá 400 pulsů a motor se otočí dvakrát

**for(**int x **=** 0**;** x **<** 400**;** x**++)** **{**

digitalWrite**(**stepPin**,**HIGH**);**

delayMicroseconds**(**500**);**

digitalWrite**(**stepPin**,**LOW**);**

delayMicroseconds**(**500**);**

**}**

delay**(**1000**);**

**}**

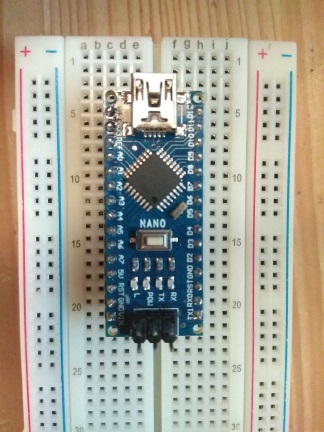
# Využité technologie

## Hardware

Pro zhotovení tohoto projektu bylo zapotřebí jak softwarové, tak i hardwarové součásti. V téhle kapitole budu popisovat jednotlivé prostředky, které jsem použil. Začnu hardwarem. Jako první jsem musel sehnat mikro-kontrolér, který umožňuje řídit krokový motor a také napájet laser. K tomuto účelu jsem použil Arduino Nano. Dále krokový motor, který otáčí daný objekt. Motor můžeme zvolit v závislosti na tom, jak velké objekty chceme skenovat. Na objekt svítí červený přímkový laser a ten snímá webová kamera. Webkameru můžeme zvolit podle výsledné kvality, jaké chceme docílit při skenování. Já jsem zvolil jako konečné řešení Full HD kameru Logitech C922 PRO. To je zároveň poslední hardwarová část. V této kapitole je uveden popis všech součástí, které jsem potřeboval.

### Arduino Nano

Arduino Nano je v informatice název malého jednodeskového počítače z otevřené platformy Arduino, který je založen na mikrokontrolerech ATmega od firmy Atmel. Arduino je připojené k počítači a pomocí něj se otáčí motor. K Arduinu je připojen laser a také driver A4988. Podrobné schéma je v příloze dokumentace.



Obrázek č. 4. - Arduino Nano

### Krokový motor

Krokový motor je synchronní točivý stroj, většinou napájený impulsy stejnosměrného proudu. Magnetické pole je generováno postupným napájením jednotlivých pólových dvojic. Pohyb rotoru krokového motoru je při nízkých rychlostech nespojitý, rotor se pohybuje mezi stabilními polohami vždy v určitém úhlu – mluvíme o pohybu v krocích.

Motor je zde použit, pro otáčení daného objektu a jeho následné snímání. Krokový motor musí být napájen externím zdrojem. Tento zdroj je připojen přes ovladač Polulu A4988 a ten posílá proud do motoru. Stačí okolo 5V. Jako zdroj lze použít obyčejný 5V transformátor z 230V.



Obrázek č. 5. Krokový motor

### Line laser

Přímkový laser jsem ve svém projektu použil pro snímání objektu. Laser se vykreslí přesně podle tvaru objektu a kamera toto vykreslení zaznamená a vyfotí. Následně se z těchto výsledných snímků postupně skládá daný objekt. Laser má výkon 5 mW a je napájen 3 V– 5 V.



Obrázek č. 6. - Line laser

### Webkamera

Použil jsem nejprve webovou kameru s nízkým rozlišením pro test, ale pro nedostačující snímaný výkon jsem musel zvolit kvalitnější zařízení. Proto jsem zvolil webkameru Logitech Pro Stream Webcam C922 PRO s rozlišením 1920x1080. Tato kamera již dostačuje pro kvalitní snímání. Kamera snímá laser, který se vykresluje podle objektu a následně tento snímek pošle do programu, který z těchto snímků skládá výsledný objekt.

### Polulu A4988 – ovladač motoru

Piny pro ovládání motoru:

-VMOT pin je pro vstup externího zdroje jako „plus“ (5V)

(před připojením externího zdroje je ještě třeba na piny VMOT a GND připojit 47  µF kondenzátor pro ochranu ovladače od nebezpečného napětí)

-GND pin pro vstup externího zdroje jako „mínus“

Piny 2A, 2B jsou připojeny k jedné cívce motoru a druhé dva piny 1A, 1B jsou při pojeny k druhé cívce.

Piny pro Arduino:

-VDD pin pro napájení čipu z Arduina „plus“ (3-5.5V)

-GND pin pro „mínus“ z Arduina

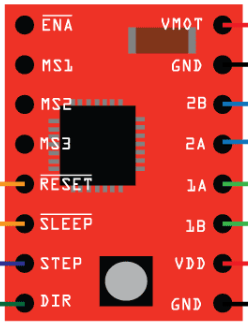
Step a dir jsou piny určené pro ovládání motoru

-STEP pin řídí kroky motoru, každý puls, který se pošle na tento pin, se motor posune o jeden krok, je připojen na digitální pin 3 v Arduinu

-DIR pin řídí směr rotace motoru, je připojen na digitální pin 4 v Arduinu

-SLEEP pin je použit pro šetření energie, když motor neběží

-RESET pin nepotřebuji, takže jej stačí připojit na pin SLEEP



Obrázek č. 7. - Polulu A4988

## Software

### FabScan100

FabScan je software, na kterém probíhá zpracování a tvorba výsledného objektu ze získaných snímků z webkamery. Tato verze běží na operačním systému Linux. Obsahuje konfigurační konzoli, na které je možno otestovat, zda všechny součástí fungují správně a také se zde musí nastavit správná pozice kamery a laseru.

### FabScanPi

Jak jsem již popsal v předchozí kapitole 1.1, FabScanPi je nejnovější verze programu, napsaná v pythonu.

## Vývojová prostředí

Pracoval jsem ve vývojovém prostředí Arduino IDE a Visual Studio Code. Protože jsem nepotřeboval posílat žádná data na server ani instalovat žádné knihovny, stačilo mi pouze nahrát kód pro řízení scanneru do Arduina. Pro tento účel mi stačilo Arduino IDE. V tomto prostředí mi stačilo vybrat typ mikro-kontroléru a USB vstup. VS Code jsem použil pro vyzkoušení a seznámení se s tímto prostředím. Tato platforma nabízí velkou škálu různých rozšíření a pluginů, které je možno snadno doinstalovat. Jednou z mnoha výhod je, možnost používání knihoven pouze pro daný projekt, nemusíme se tedy bát, že by nám někdo knihovnu omylem smazal nebo upravoval, nebo nástroj na opravu kódu debugger. Já jsem si stáhl rozšíření Platformio, které je přímo určené pro práci s mikro-kontroléry.

### PyCharm ??

* nástroje, přístroje, programy a jiné materiální prostředky, včetně zdůvodnění jejich výběru, použité při řešení zadaného úkolu
* informace o použitých technologiích by neměly mít charakter reklamy na daný výrobek, ale měly by přinášet objektivní technický popis použitých prostředků včetně zdůraznění a vysvětlení klíčových parametrů

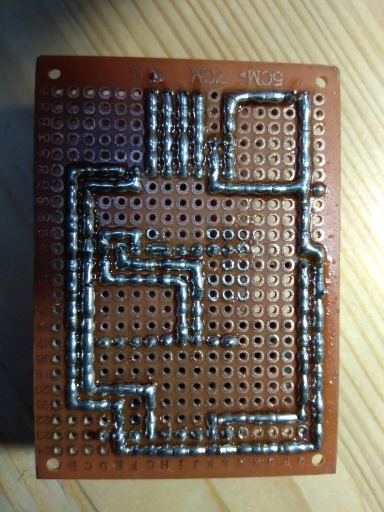
# Způsoby řešení a použité postupy

## Finální kompletování

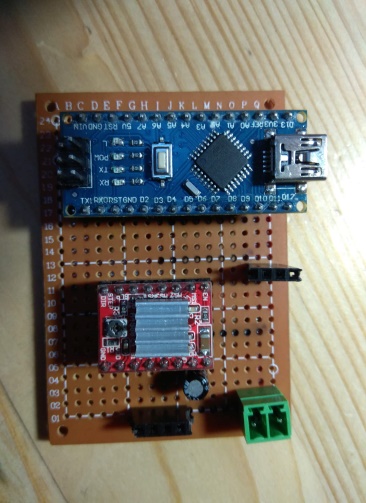
Zde jsem spojil všechny součásti dohromady. Abych mohl lépe pracovat s jednotlivými díly, musel jsem si je trochu upravit. Začal jsem prodloužením drátů u laseru a krokového motoru. Odizoloval jsem drátky, poté jsem k nim přimotal další dráty a pomocí pájky je připájel k sobě. Místo laboratorního zdroje jsem použil transformátor na 5 V. Pro lepší způsob zapojení je většina drátů upravena tak, aby se dala jednoduše připojit a odpojit z desky.

### Pájení na desku

Abych mohl lépe manipulovat a pracovat s hardwarovými součástmi, rozhodl jsem se je připájet na desku s plošnými spoji pomocí pin lišt. Toto řešení jsem zvolil, protože můžu zařízení kdykoliv odpojit a připojit. Na desce je dále připájen kondenzátor, vstup pro laser, vstup pro ovládání motoru a také napájení motoru.



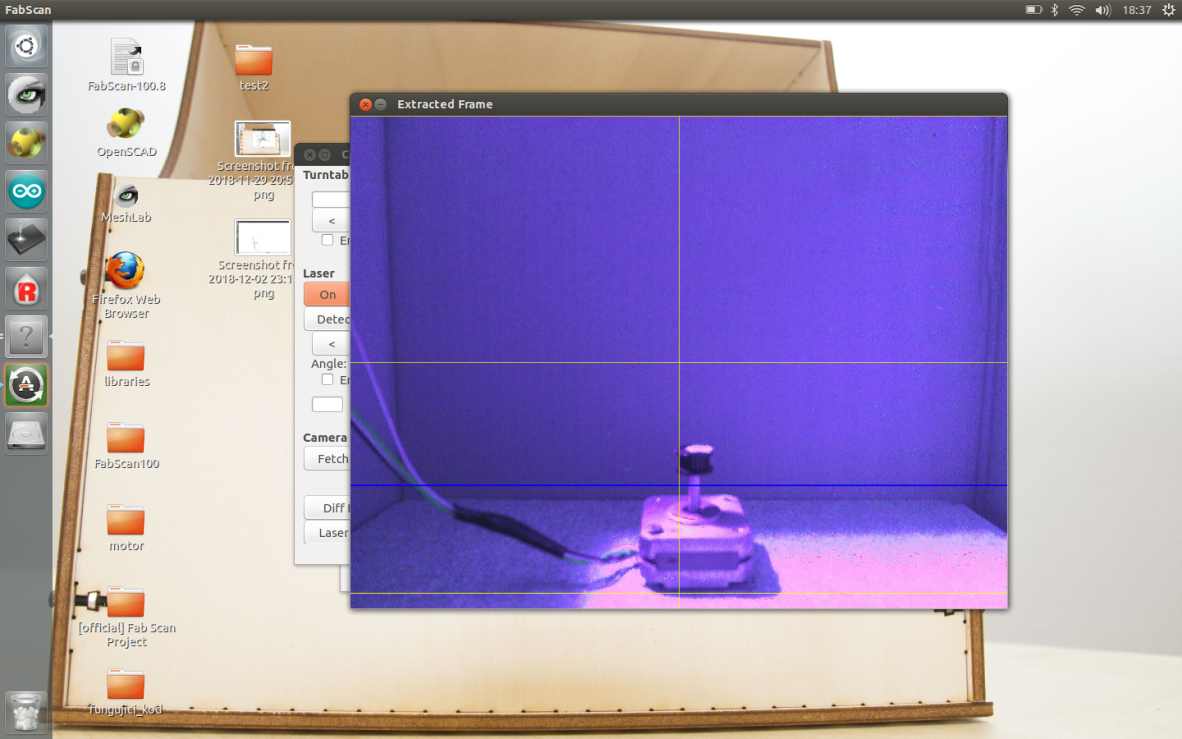
Obrázek č. 8 - Plošný spoj



Obrázek č. 9 - Výsledné zapojení

### Sestavení boxu

Všechny díly jsou uloženy v jednom boxu. Použil jsem kartónovou krabici velikosti 300mm x 400mm. Do ní jsem postupně umístil jednotlivé díly. Nejprve jsem upevnil motor doprostřed kratší strany úplně ke hraně. Na protější stranu jsem nastavil kameru přesně naproti motoru. Vedle kamery je umístěn laser, který míří přímo na doprostřed motoru – na místo na kterém se bude točit objekt. Úhel kamery a laseru je nutné nastavit pomocí konfiguračního panelu v samotném programu. Deska s plošnými spoji je umístěna na dně krabice a vede z ní vývod na USB do počítače pro Arduino, další USB pro webovou kameru a napájení motoru do sítě na 220 – 230 V.



Obrázek č. 10 – Nastavení kamery 1

## Řešení problémů

### Webkamera

Svoje první testy jsem prováděl s webkamerou, která neměla dostatečné rozlišení pro skenování. Kamera neuměla rozpoznat laser a program měl poté problémy se zpracováním objektu. Z tohoto důvodu jsem použil full HD kameru Logitech Pro Stream Webcam C922 PRO.

### Odpálení Arduina

Při prvních testech zapojení motoru jsem se pokoušel napájet krokový motor přímo z Arduina, které má vývod na 5 V, ale bohužel jsem zjistil, že takové zapojení nefunguje a Arduino se spálilo. Došlo zde nejspíše ke zkratu. Proto jsem se o další zbytečné zkoušky nepokoušel a krokový motor napojil na externí zdroj 5 V.

# Výsledky řešení, výstupy, uživatelský manuál

Text čtvrté kapitoly

* výčet splněných a nesplněných cílů, obrázky (schémata, vzorce apod.) z finálního provedení, prokázání funkčnosti, výsledné parametry výrobku apod.
* podle zaměření a charakteru práce je třeba volit vhodný nadpis pro tuto kapitolu, je samozřejmě možné i rozdělení na více kapitol (např. Uživatelské rozhraní internetové aplikace; Administrace internetové aplikace…)

# **Závěr**

Text závěru

* povinná část,
* shrnuje výsledky, hodnotí splnění cíle práce, uvádí možnost uplatnění řešení v praxi a nastínění případných dalších budoucích vylepšení
* kapitola se nečísluje (stejné jako úvod)

Seznam použitýCH INFORMAČNÍCH ZDROJů

[] Home - FabScanPi. Site not found · GitHub Pages [online]. Copyright © Copyright 2015 [cit. 06.01.2019]. Dostupné z: <https://mariolukas.github.io/FabScanPi-Server/>

[2] GitHub - mariolukas/FabScanPi-Server: FabScan Pi Open Source 3D Laser Scanner - Server application. The world’s leading software development platform · GitHub [online]. Copyright © 2019 [cit. 06.01.2019]. Dostupné z: <https://github.com/mariolukas/FabScanPi-Server>

[3] FabScan – how to build a 3D scanner? A DIY tutorial… | 3D Printing Center - news, opinions, tests & reviews. CD3D - 3D Printing Center [online]. Copyright © [cit. 06.01.2019]. Dostupné z: <http://3dprintingcenter.net/2016/06/03/fabscan-how-to-build-a-3d-scanner-do-it-yourself/>

[4] Downloads – Homepage von Wolfgang Schmidt. Homepage von Wolfgang Schmidt [online]. Dostupné z: <http://wolfgangschmidt.info/simple3dscan-download/>

[5] How To Control Stepper Motor with A4988 Driver and Arduino. How To Mechatronics [online]. Copyright © 2018 HowToMechatronics.com. All rights reserved. [cit. 06.01.2019]. Dostupné z: <https://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/>

Seznam příloh

č. 1 Titulní list

č. 2 Čestné prohlášení

č. 3 Poděkování

Nepovinná část – pokud nemáte žádné přílohy ke své práci, tuto část odstraňte!

* Přílohy se zařazují na konec práce.
* Jsou to texty, obrázky, grafy, tabulky, které by přímo v textu byly zbytečně detailní, ale mají být po ruce k dokreslení východisek i výsledku řešení.
* Jsou číslovány a v textu se na ně může odkazovat.
* Před první přílohu se umisťuje seznam příloh.
* Každá příloha je označena číslem - např. Tabulka č.. 1, Schéma č. 2, Obrázek č. 3.
* Každá tabulka by měla mít i vlastní název, který stručně vystihuje její obsah.
* (Tabulka č. 1 Zakázky stavebních prací v roce 2009-2010).
* Pokud je z tabulky vytvořen graf, umístíme jej na stejné stránce jako tabulku.

**Příloha č. 1: Titulní list**

1. Objektově orientovaný skriptovací jazyk. [↑](#footnote-ref-1)
2. Značkovací jazyk pro webové stránky. [↑](#footnote-ref-2)
3. JavaScriptové aplikační rozhraní pro nativní zobrazování 3D grafiky. [↑](#footnote-ref-3)