|  |  |  |
| --- | --- | --- |
| 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 | |

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

OBSAH

[Úvod 5](#_Toc533329789)

[1 Teoretická a metodická východiska 6](#_Toc533329790)

[1.1 Princip Software 6](#_Toc533329791)

[1.1.1 Testování krokového motoru 6](#_Toc533329792)

[1.1.2 Schéma zapojení 8](#_Toc533329793)

[1.1.3 Zdrojový kód 8](#_Toc533329794)

[2 Využité technologie 9](#_Toc533329795)

[2.1 Hardware 9](#_Toc533329796)

[2.1.1 Arduino Nano 9](#_Toc533329797)

[2.1.2 Krokový motor 9](#_Toc533329798)

[2.1.3 Line laser 9](#_Toc533329799)

[2.1.4 Webkamera 9](#_Toc533329800)

[2.1.5 Driver 9](#_Toc533329801)

[2.2 Software 9](#_Toc533329802)

[2.2.1 FabScan100 9](#_Toc533329803)

[2.2.2 Arduino IDE 10](#_Toc533329804)

[2.2.3 FabScanPi 10](#_Toc533329805)

[3 Způsoby řešení a použité postupy 11](#_Toc533329806)

[3.1 Finální kompletování 11](#_Toc533329807)

[3.1.1 Pájení na desku 11](#_Toc533329808)

[3.1.2 Sestavení boxu 11](#_Toc533329809)

[4 Výsledky řešení, výstupy, uživatelský manuál 12](#_Toc533329810)

[Závěr 13](#_Toc533329811)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 14](#_Toc533329812)

[Seznam příloh 15](#_Toc533329813)

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 pro svůj projekt použil starší verzi programu - FabScan100, která běží na operačním systému Ubuntu 13.04. Celý program i s OS je zabalený v instalačním souboru ISO, navíc ještě obsahuje pár další programů na úpravu objektů, proto jsem zvolil právě tuto verzi. Zkoušel jsem samozřejmě také druhu verzi, kterou se podrobněji pokusím popsat v první kapitole své dokumentace.

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 uplatnění v praxi a další 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.

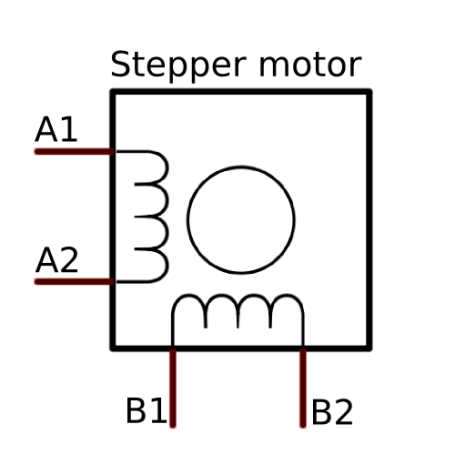
Nyní zde zmíním ještě druhou možnost, kterou jsem vyzkoušel a tou je FabScanPi, jenž umožňuje ovládat scanner přes webový prohlížeč. Tato verze programu je dostupná zatím pouze pro OS Linux a je napsaná v programovacím jazyce Python. Webové uživatelské rozhraní je napsáno pomocí JavaScriptu[[1]](#footnote-1) 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.

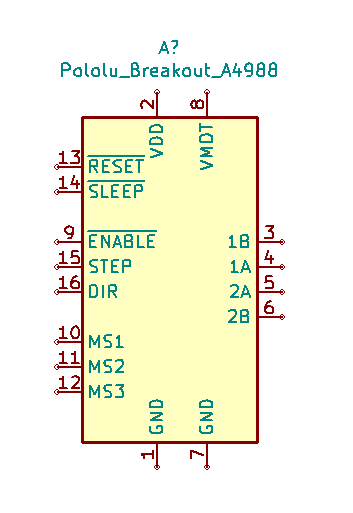
### Testování krokového motoru

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.

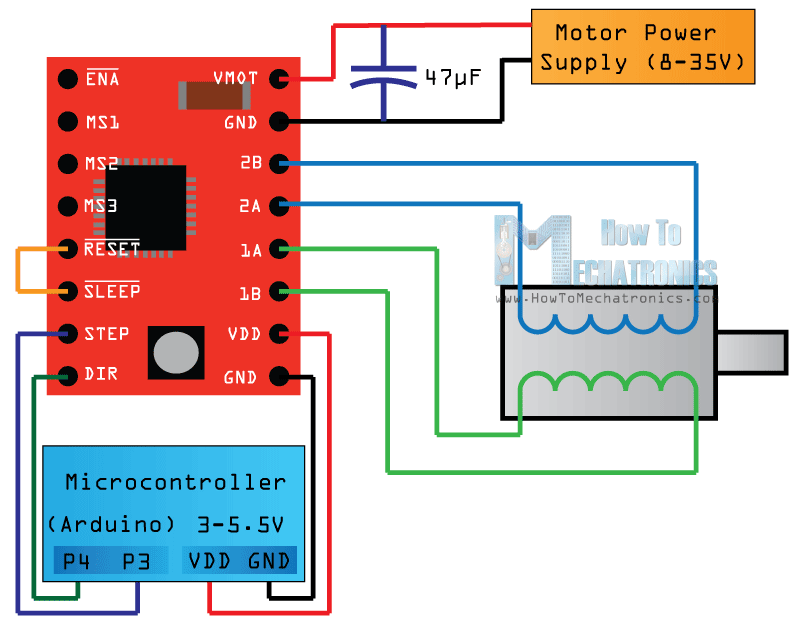


Obrázek 1. - Krokový motor 1



Obrázek 2. - Polulu A4988 1

### Schéma zapojení



Obrázek 3. - Schéma zapojení 1

### 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**);**

**}**

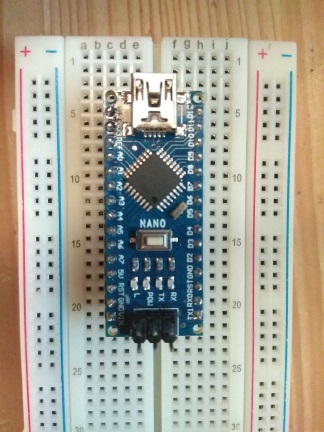
# Využité technologie

## 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.



Obrázek 4. - Arduino Nano 1

### 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 5. Krokový motor 1

### 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 5mW a je napájen 3 – 5V.



Obrázek 6. - Line laser 1

### 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:

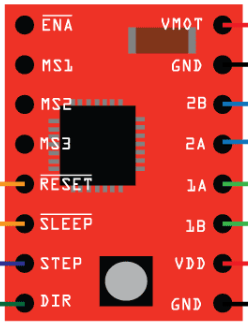
-VDD pin pro napájení čipu z Arduina „plus“ (3-5.5V)

-GND pin pro „mínus“ z Arduina

Step a dir jsou piny určené pro ovládání motoru

-STEP pin řídí

-DIR pin řídí směr rotace motoru



Obrázek 7. - Polulu A4988 1

## Software

Programový část obsahuje program na tvorbu objektů, vývojové prostředí a kód pro Arduino.

### FabScan100

FabScan je software, na kterém probíhá zpracovánía tvorba výsledného objektu ze získaných snímků z webkamery. Arduino je spojeno s počítačem, na kterém běží programem FabScan a ten přes něj posílá příkazy a řídí celý scanner. Umožňuje vypínat a zapínat laser, otáčet motorem a spouštět samotný scanner.

### Arduino IDE

### FabScanPi

* 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

### Pájení na desku

### Sestavení boxu

* popis řešení úkolu včetně, použité postupy a jejich vysvětlení, způsoby testování funkčnosti, parametry výrobku (programu, hotového řešení), schémata, obrázky z tvorby a finálního provedení, výpočty, použité příkazy…

# 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

[1] BOHMAN, Ludvík. Zákon o pojistné smlouvě. Praha: Linde Praha a. s., 2004. 381 s. ISBN80-7201-504-4

[2] DUCHÁČKOVÁ, Eva. Principy pojištění a pojišťovnictví. 3. aktualizované vydání. Praha: Ekopress 2009. 224 s. ISBN 978-80-86929-51-4

[3] KUBALA, Petr. Planetární dvojcata - Věda a technika (Český rozhlas) [online].   
Č. 2000-2008, poslední revize 19. 3. 2008 [cit. 2008-03-20].  
<http://www.rozhlas.cz/veda/vesmir/\_zprava/435849>.

[4] KULDOVÁ, O., FLEISCHMANNOVÁ, E. Metodická příručka k technice administrativy a obchodní korespondence. 1.vyd. Praha: Fortuna 1998. 111 s.   
ISBN 80-7168-574-7. Kapitola 6, Metody nácviku psaní hmatovou metodou,   
s. 28-29.

[5] VLACH, J. JE Temelín a zásobování teplem. Energetika, 2001, roč. 51, č. 3, s. 84 -85. ISSN 0375-8842.

* musí zahrnovat všechny prameny, knihy, internetové odkazy a další studijní podklady, z nichž jsme čerpali;
* kapitola se nečísluje a zde končí číslování stránek práce;
* jednotlivé publikace se uvádějí v abecedním pořadí podle příjmení autorů a iniciál jeho jména, který se píše za čárkou;
* příjmení autora se píše velkými písmeny;
* název publikace se zvýrazňuje kurzívou;
* jestliže jsou uvedeni více než tři autoři, je možné vypsat hlavního autora s poznámkou „a kol.“(a kolektiv).

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)