

Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta

Katedra informatiky

Bakalářská práce
Paralelní robot CoreXY

Vypracoval: Lukáš Rajzl
Vedoucí práce: Ing. Michal Šerý, Ph.D.
České Budějovice 2016

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci na téma Paralelní robot CoreXY vypracoval samostatně, pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích 19.12.2016

Lukáš Rajzl

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lukáš RAJZL**
Osobní číslo: **P13815**
Studijní program: **B7507 Specializace v pedagogice**
Studijní obor: **Informační technologie a e-learning**
Název tématu: **Paralelní robot CoreXY**
Zadávací katedra: **Katedra informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Úvod
2. Základní pojmy, uvedení do problematiky pohybových zařízení, 3D tisk
3. Seznámení se s vývojovou sadou Arduino - základní představení platformy
4. Možnosti vývojové sady Arduino
5. Možnosti programování platformy Arduino - sada praktických ukázek:
 - Teplotní čidlo (Základní syntaxe, ovládání vstupu)
 - Řízení teploty (podmínky if, ovládání vstupu a výstupu)
 - Rozsvěcení diody a blikání diody (Cykly)
 - Řízení RGB diody (PWM regulace)
 - Práce ze sedmisegmentovým displejem (Cykly a pole)
 - Řízení krokového motoru
 - Komunikace mezi moduly
 - Komunikace s nadřazeným PC
6. Komplexní příklad: robot CoreXY
7. Konstrukce CoreXY (mechanika a elektronika), vypracování návrhu
8. Sestavení a oživení CoreXY
9. Technická dokumentace a test zařízení
10. Zhodnocení a závěr

Rozsah grafických prací: **CD ROM**

Rozsah pracovní zprávy: **40**

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

Internetové zdroje:

1. Průvodce světem ARDUINA [online]. 2014 [cit. 2015-08-28]. Dostupné z: <http://arduino.cz/>
2. Základy 3D tisku [online]. 2014 [cit. 2015-08-28]. Dostupné z: <http://www.prusa3d.cz/>
3. CoreXY — Cartesian Motion Platform [online]. 2014 Dostupné z: <http://www.corexy.com/theory.html>
4. RepRap [online] 2015 Dostupné z: http://reprap.org/wiki/General_FAQ#parallel_manipulator_systems

Vedoucí bakalářské práce:

Ing. Michal Šerý, Ph.D.

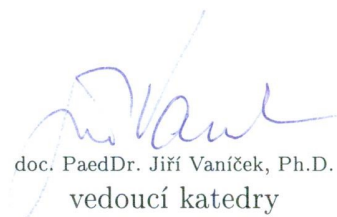
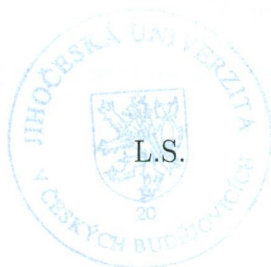
Katedra aplikované fyziky a techniky

Datum zadání bakalářské práce: **19. listopadu 2015**

Termín odevzdání bakalářské práce: **29. dubna 2016**



Mgr. Michal Vančura, Ph.D.
děkan



doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 19. listopadu 2015

Abstrakt

Bakalářské práce je zaměřena na rozbor aktuálního stavu v oblasti paralelních robotů, jejich porovnání, poté se bude zabývat návrhem vlastního zařízení typu CoreXY. Posledním bodem bude návrh realizován a otestována funkčnost navržené a realizované koncepce. V rámci práce bude vypracován konstrukční návod.

Abstract

Thesis focuses on the analysis of the current state of parallel robots , comparing them, then deal with the design your own device type CoreXY . The last point of the proposal will be implemented and tested functionality designed and implemented the concept. As part of the work will be developed design instruction.

Klíčová slova

CoreXY, Delta, paralelní robot, Arduino

Keywords

CoreXY, Delta, parallel robot, Arduino

Obsah

1	Úvod	7
2	Elektrotechnika	7
2.1	Základní elektrické veličiny	7
2.2	Základní vztahy v elektronice	8
2.3	Elektronické součástky	8
3	Arduino	10
3.1	Arduino desky a shieldy	11
3.2	Možnosti programování vývojové desky Arduiono	15
4	CAD - Computer-aided drafting	28
4.1	Historie CAD	28
5	Návrh paralelního robota	29
5.1	Specifikace robota	29
5.2	Typ konstrukce robota	30
5.2.1	Delta	31
5.2.2	CoreXY	32
5.2.3	Porovnání	32
5.3	Úprava referenčního mechanismu CoreXY	33
5.4	Osa Z	34
5.5	Výběr materiálu	34
5.6	Výběr řídicí elektroniky	35
5.7	Nákresy	38
5.8	Skládání paralelního robota	39
5.9	Ovládací software a výběr firmware	42
5.10	Teacup	44
5.11	Aprinter	46
5.12	Repetier	48
5.13	Srovnání	49
6	Test funkčnosti zařízení	50
7	Závěr	51

1 Úvod

Mnozí lidé se bojí technických předmětů a programování. Většinou zbytečně. Na jednoduchých příkladech si předvedeme základy programování a připojování periférií na výukovou desku Arduino. Aby jsme toto všechno zvládli jiným způsobem než pouhým kopírováním již vytvořených návrhů, budeme potřebovat určitý vědomostní základ z několika oborů. Proto část této práce je věnována seznámením se s elektrotechnikou a programováním vývojových desek Arduino.

Všechny tyto znalosti budeme prakticky využívat tak, aby jsme je prakticky ovládli a byli schopni s těmito znalostmi pracovat, což zároveň ulehčuje pochopení teorie jednotlivých oborů. Cílem této práce není z jejich čtenářů udělat odborníky v dané problematice, ale má sloužit jako úvod do světa techniky a praktického programování.

Zároveň tato práce opravuje některé chyby v referenčním návrhu paralelního robota typu CoreXY a zároveň přinese čtenáři přehled, co taková konstrukce paralelního robota obnáší.

2 Elektrotechnika

V této části se budeme věnovat elektrotechnice, tedy základním elektrickým veličinám, zákonům platným v elektrických obvodech a elektrosoučástkám. Nebudu zde zmiňovat všechny součástky, ale převážně ty části, které budeme potřebovat později v sekci Možnosti programování vývojové desky Arduino.15.

2.1 Základní elektrické veličiny

Elektrický náboj - vyjadřuje schopnost působit elektrickou silou. Značíme jej Q a jeho jednotkou je Coloumb [C]. Může nabývat kladnou i zápornou hodnotu. Nositel elektrického náboje bývá buď proton (kladná náboj) nebo elektron (záporný náboj).

Elektrický proud - vyjadřuje množství elektrických nábojů procházejících vodičem o daném průřezu za jednotku času. Značíme ho velkým písmenem I a jeho jednotkou je ampér [A]. Měříme ho ampérmetrem zapojeným sériově. Může mít buď střídavý nebo stejnosměrný průběh.

Elektrické napětí - je rozdíl elektrických potenciálů mezi dvěma body. Značíme ho písmenem U a jeho jednotkou je volt [V]. Můžeme mít střídavé nebo stejnosměrné. V případě střídavého napětí (jenž má nejčastěji sinusový průběh) se udává efektivní hodnota napětí.

Elektrický výkon - je elektrická práce vykonaná za jednotku času. Značíme jí velkým písmenem P a její jednotkou je watt [W]. U střídavého proudu rozlišujeme několik druhů výkonů (činný jalový, zdánlivý), u stejnosměrného proudu jej jednoduše spočítáme pomocí vzorečku $P=U \cdot I$.

Elektrický odpor - schopnost vést elektrický proud. Značíme jej písmenem R a jeho jednotkou je ohm [Ω]. Převrácená hodnota elektrického odporu je vodivost (značíme písmenem G jednotka je siemens [S]).

2.2 Základní vztahy v elektronice

Ohmův zákon

Ohmův zákon udává vztah mezi napětím, proudem a odporem. „Je-li napětí na koncích vodiče stálé, je proud nepřímo úměrný odporu vodiče.“ [1] Vyjádřeno matematicky: $R= U/I$.

2.3 Elektronické součástky

Rezistor

Rezistor neboli odpor je součástka, která by se měla projevat pouze elektrickým odporem. Používá se většinou pro snížení proudu nebo napětí. Je několik druhů rezistorů. Například termistor má proměnlivý odpor na základě teploty. Dále máme potenciometr a trimr, což jsou součástky, na kterých můžeme plynule regulovat velikost odporu. Využívají se například pro změnu intenzity podsvícení displejů, regulaci hlasitosti atd.



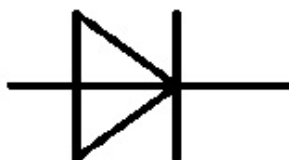
Obrázek 1: Schématická značka rezistoru



Obrázek 2: Schématická značka potenciometru

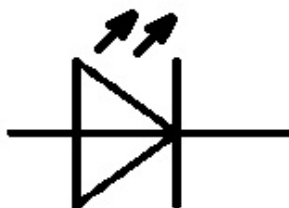
Dioda

Dioda je polovodičová součástka, která má velmi rozdílné vlastnosti v závislosti na tom, jak je připojena do obvodu. Skládá se z dvou elektrod označovaných jako katoda a anoda. Používáme ji v zapojení propustném (anoda má připojené kladné napětí) nebo závěrném (katoda má připojené kladné napětí). Mezi její podstatné parametry patří maximální proud v propustném směru a maximální napětí v závěrném směru. V propustném směru dioda do dosažení prahového napětí nevede, po jeho dosažení (obvykle kolem 0,6 V) začne vést. V případě dosažení maximálního propustného proudu dojde k tepelné destrukci diody. V závěrném směru diodou protéká velmi malý proud, v případě dosažení maximálního závěrného napětí, začne proud opět růst a pokud dosáhne maximálního proudu, dojde k destrukci diody¹.



Obrázek 3: Schématická značka diody

LED - dioda emitující světlo (Light-Emitting Diode) je speciální druh diody, která pokud je zapojena v propustném směru, vyzařuje světlo nebo infračervené či ultrafialové záření. V závislosti na vlnové délce vyzařovaného světla se mění napětí velikost napětí v propustném směru (například modrá led má prahové napětí kolem 2,5 V). Anoda má většinou delší vývod než katoda. Zbývá specifikace je prakticky stejná jako u klasické diody.



Obrázek 4: Schématická značka LED

¹Je mnoho druhů diod a některé se převážně používají v závěrném směru (Zenerova dioda) a jiné v propustném (LED)

Tranzistor

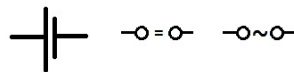
Jedná se o polovodičovou součástku minimálně se třemi vývody. Byla vynalezena v roce 1947 v USA. Vynález tranzistoru vedl k převratu v elektrotechnice a tranzistor je dnes běžnou součástkou většiny integrovaných součástek. Dle potřeby se dá využít jako spínač, zesilovač nebo inventar.

Krokový motor

Je druh elektrického motoru, jehož rotor² se pohybuje nespojitě. Každý krokový motor se pohybuje mezi stabilními klidovými polohami. To je řízeno pomocí impulsů, které jsou v určité posloupnosti. Lze je využít kdekoliv, kde potřebujeme otočit motorem pouze o určitý úhel. Navíc i po otočení zůstávají jeho cívky pod napětím, čily se v dané pozici motor drží. Dělíme je na dva základní typy a to unipolární a bipolární.

Elektrický zdroj

Elektrický zdroj je zařízení, které je schopné na svých výstupech udržovat napětí a zásobit obvod elektrickým proudem. Zdroj má svůj vlastní vnitřní odpor, proto se zvedající zátěží (odebíraným proudem) klesá napětí na zdroji. Podle velikosti vnitřního odporu dělíme zdroje na tvrdé a měkké.



Obrázek 5: Schématické značka různých zdrojů

3 Arduino

Arduino³ je open source projekt univerzální prototypové desky. Založená je nejčastěji na mikrokontroléru od společnosti Atmel. Tento projekt vznikl v Itálii ve městě Ivrea v roce 2005. Cílem tohoto projektu bylo dát studentům levnější alternativu do té doby používanému BASIC STAMP, jenž se prodával za cenu kolem 100\$. K této desce je dostupné jednoduché vývojové prostředí, jako programovací jazyk se využívá Wiring. K Arduino desce můžeme lehce připojovat mnoho elektronických součástek a periférií.

²Pohyblivá část motorů

³Oficiálně by se měl mimo USA používat název Genuino

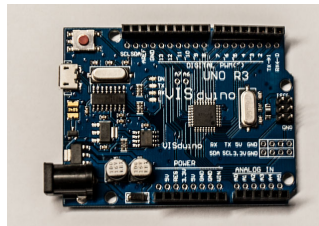
Arduino není učeno jen pro profesionální použití. Díky obsáhle komunitě a nízké ceně může s arduinem začít opravdu každý. Projekty založené na těchto deskách mohou být malé a vcelku jednoduché, jako například stavba domácí meteo stanice. Nebo rozsáhlé projekty jako kompletní řízení domácnosti, 3D tiskárny atd.

Mezi hlavní výhody Arduina patří:

- Multiplatformnost
- Nízké pořizovací náklady
- Jednoduchost
- Dostupnost návodů a komunitní podpory

Samotné desky většinou obsahují převodník USB-serial a několik pinů určených pro vstup a výstup. K těmto deskám se dají připojit rozšiřující karty zvané „shieldy“. Tyto shieldy nám umožňují přidat funkcionalitu, kterou by jsme jinak tvořili velmi složitě a zdlouhavě. Velmi populární je například Ethernet shield (rozšíření Arduina o komunikaci skrze ethernet).

Díky open-source licenci vzniká mnoho klonů Arduina. Od svých originálních předloh se velmi často liší v použitých konektorech a často i čipech (například pro USB, několik desek používalo falešný FTDI čip, tento čip poté jedna verze ovladačů „odpalovala“). Častý problém klonů je jejich kvalita. Velmi špatně napájené součástky na desku způsobují nefunkčnost, oprava však bývá velmi snadná.



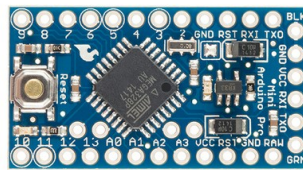
Obrázek 6: Klon Arduino UNO

3.1 Arduino desky a shieldy

Arduino není v současnosti pouze jedna prototypová deska ale několik. Díky dostupnosti více desek, jenž se liší vybavením a velikostí, si každý může vybrat tu správnou pro svůj projekt. Vzhledem k velkému množství desek zde uvedu pouze nejznámější desky a jejich základní popis. Pro kompletní popis navštivte oficiální stránky www.Arduino.cc

Arduino Mini Pro

Jedna z nejmenších desek. Je osazena mikrokontrolérem ATmega328, disponuje 14 digitálními vstupy/výstupy, 6 analogovými vstupy, je vybavena dírami pro napájení pinů. Vyrábí se ve dvou verzích. Jedna s napětím 3,3 V a frekvencí 8 MHz, druhá s napětím 5 V a frekvencí 16 MHz. Oběma verzím chybí USB-serial převodník.



Obrázek 7: Arduino Pro Mini

Arduino Nano

Další z malých desek. Od Mini se liší hlavně přítomností USB-Serial převodníku. Je osazena mikrokontrolérem ATmega328 (verze 3.X) nebo ATmega168 (verze 2.X). Disponuje 14 digitálními vstupy/výstupy a 8 analogovými vstupy.



Obrázek 8: Arduino Nano

Arduino Micro

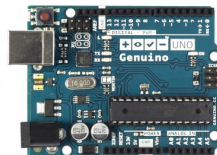
Tato deska je založena na procesoru ATmega32U4 na 16 MHz, který má v sobě zabudovanou podporu pro USB, což umožňuje s ním lehce emulovat klávesnici nebo myš. Disponuje 20 digitálními vstupy/výstupy, z nichž 12 může být využito jako analogové vstupy a 7 jako PWM výstup.



Obrázek 9: Arduino Micro

Arduino UNO

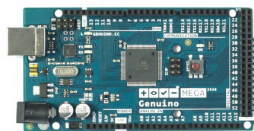
Jedna z nejvíce používaných desek. Obsahuje procesor ATmega328P, 14 digitálních vstupů/výstupů, 6 analogových vstupů, USB konektor, a napájecí konektor přes který můžete tuto desku napájet. Rozsah vstupního napětí je poměrně velký (6-20 V, doporučeno 7-12 V).



Obrázek 10: Arduino UNO

Arduino MEGA 2560

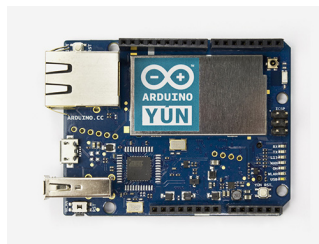
Deska doporučována pro větší projekty typu 3D tiskárna nebo projekty z oblasti robotiky. Tato deska je osazena mikrokontrolérem ATmega2560 na 16 MHz a má 54 digitálních vstupů/výstupů (z toho 15 může být využito jako PWM výstup), 16 analogových vstupů, 4x UART a napájecí konektor.



Obrázek 11: Arduino MEGA 2560

Arduino Yún

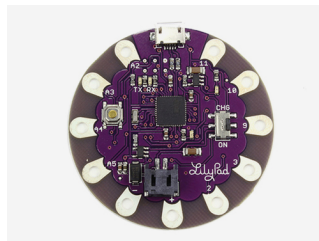
Na rozdíl od většiny desek, tato obsahuje rovnou dva procesory, ATmega32u4 a Atheros AR9331. Deska obsahuje 20 digitálních vstupů/výstupů, 12 analogových vstupů, ethernet, wifi, čtečku pamětových karet, mikro USB port a host USB port. K procesoru AR9331 je připojen USB host port, čtečka SD karet, Ethernet, wifi a má podporu pro OpenWRT distribuci linuxu. Procesor ATmega32u4 má na starosti zbytek komponent. Mezi sebou tyto procesory komunikují pomocí mostu.



Obrázek 12: Arduino Yún

LilyPad Arduino USB

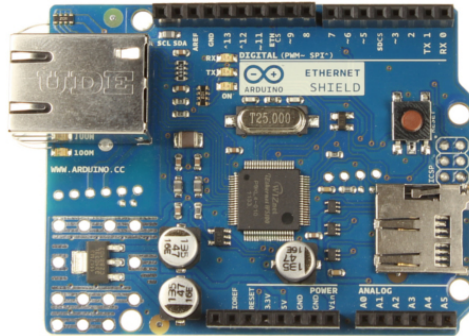
Tato deska se zaměřuje na nositelnou elektroniku, tzv. e-textil. Je založena na mikrokontroléru ATmega32u4 na 8 MHz. Obsahuje 9 digitálních vstupů/výstupů, z nichž 4 mohou být použity jako PWM výstup a 4 jako analogový vstup. Má konektor pro připojení 3,7 V LiPo baterie.



Obrázek 13: LilyPad Arduino USB

Arduino Ethernet Shield USB

Rozšiřující deska pro Arduino, obsahující ethernetový řadič W5100 a slot na Mikro SD kartu. K Arduino se připojuje pomocí SPI portu. Tento shield je ještě možné rozšířit o PoE (Power Over Ethernet), kterým poté můžeme napájet celou sestavu desek.



Obrázek 14: Arduino Ethernet Shield

3.2 Možnosti programování vývojové desky Arduino

Programování Arduino desek je velmi jednoduché a dá se lehce využít k výuce programování. Na rozdíl od běžné výuky studenti vidí praktický výsledek programování, což jim umožňuje lépe pochopit jednotlivé funkce. Základní kód aplikace obsahuje dvě části. První je `void setup()`, v které definujeme věci, které se provedou pouze jednou při zapnutí Arduina. Obvykle v ní definujeme, které piny budou vstupní a které výstupní. Druhá je `loop`, která se provádí opakovaně, jedná se tedy o cyklus. Nejčastějším příkladem je rozsvěcování led diody.

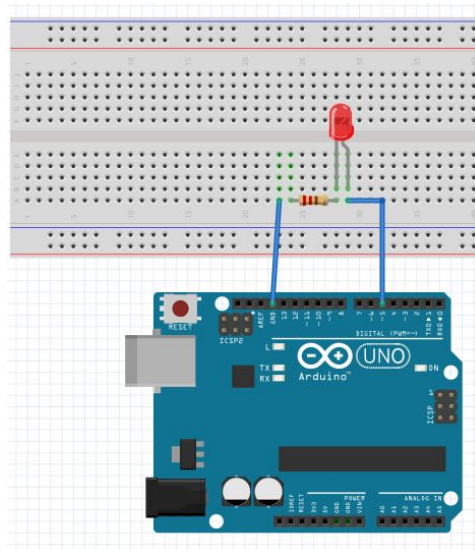
Výsledkem tohoto kódu bude, že se dioda rozsvítí, bude svítit 3 vteřiny, poté zhasne na 3 vteřiny a poté se celý tento cyklus opakuje. K ukázkám programování Arduina se přidává ještě obvykle náčrt zapojení. K tomu můžeme využít jeden z mnoha bezplatných nástrojů, já osobně používám Fritzing (Naleznete na fritzing.org)

V zapojení si můžeme všimnout kromě Arduina a led diody ještě rezistoru. Ten slouží ke snížení napětí na led diodě tak, aby nebylo překročeno její prahové napětí. Pro jeho správnou hodnotu můžeme použít výpočet podle Ohmova zákona^{2.2}

Ukázka kódu 1: Rozsvěcení LED diody

```
void setup() {
  pinMode(5, OUTPUT); //Nastaví pin 5 jako výstupní
}

void loop() {
  digitalWrite(5,HIGH); //Nastaví pin 5 na zapnuto
  delay(3000);           //Počká 3000 milisekund
  digitalWrite(5,LOW);  //Nastaví pin 5 na vypnuto
  delay(3000);          //Počká 3000 milisekund
}
```



Obrázek 15: Zapojení LED diody

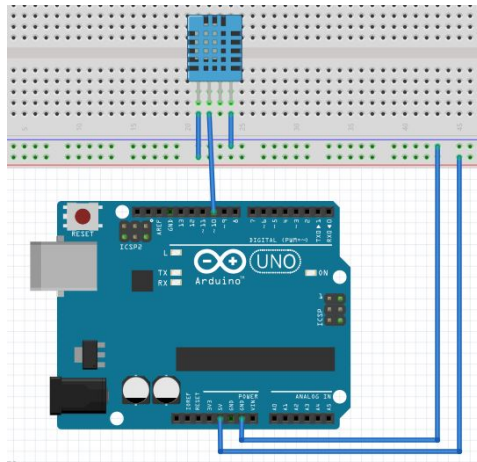
V zapojení si můžeme všimnout kromě Arduina a led diody ještě rezistoru. Ten slouží ke snížení napětí na led diodě tak, aby nebylo překročeno její prahové napětí. Pro jeho správnou hodnotu můžeme použít výpočet podle Ohmova zákona 2.2

Teplotní a vlhkostní čidlo

Arduino můžeme rozšiřovat o různé senzory. Jeden z velmi často užívaných je senzor DHT11, jedná se o teplotní a vlhkostní čidlo. Aby jsme si ho mohli využít, budeme si muset naimportovat knihovnu pro toto čidlo. Tu nalezneme například na oficiálních stránkách Arduina (Arduino.cc). Jednoduchý program, který načítá hodnotu z čidla, může vypadat například takto:

Ukázka kódu 2: Teplotní čidlo

```
#include <dht11.h> // Naimportuje knihovnu pro čidlo DHT11
dht11 cidlo; // Vytvoří objekt čidlo
void setup() {
}
void loop() {
  cidlo.read(10); // Načte hodnoty z čidla na pinu 10
  teplota=cidlo.temperature; // uloží teplotu do proměnné teplota
  vlhkost=cidlo.humidity; // uloží vlhkost do proměnné vlhkost
}
```

Obrázek 16: zapojení čidla DHT11

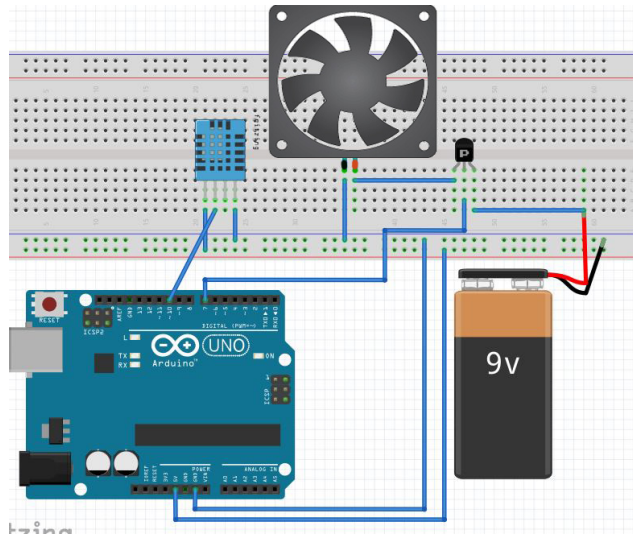
Automatické řízení teploty

Když už jsme si předvedli, jak jednoduše se dá načítat teplota, určitě by jsme jí chtěli i nějakou cestou regulovat. V našem případě zvolíme regulaci teploty pomocí ventilátoru. Zde ovšem nastává problém. Výstup z Arduino umožňuje pouze malý výkon, proto budeme muset sáhnout po nějaké součástce, která nám umožní spínat větší zátěž. Touto součástkou bude PNP tranzistor. V tomto kódu budou již užity podmínky `if`. Jejich syntaxe je jednoduchá `if(podmínka){ Co má provést pokud je splněno }`. Tyto podmínky se dají dále rozvést pomocí `else`, které nám říká, co dělat pokud podmínka není splněna.

Ukázka kódu 3: Řízení teploty

```
#include <dht11.h> //Naimportuje knihovnu pro čidlo DHT11
dht11 cidlo; //Vytvoří objekt čidlo
void setup() {
  pinMode(7, OUTPUT);
  digitalWrite(7, HIGH);
}

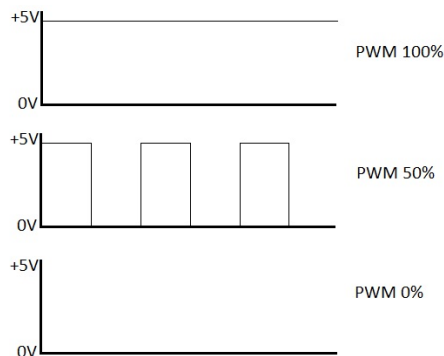
void loop() {
  cidlo.read(10); //Přečte hodnotu z čidla
  if(cidlo.temperature > 25){ //První podmínka pokud je teplota větší než 25
    digitalWrite(7,LOW); //Nastaví tranzistor jako průchozí
  }
  else{
    digitalWrite(7,HIGH); //Nastaví tranzistor jako neprůchozí
  }
  delay(1000);
}
```



Obrázek 17: Řízení teploty

Řízení RGB diody

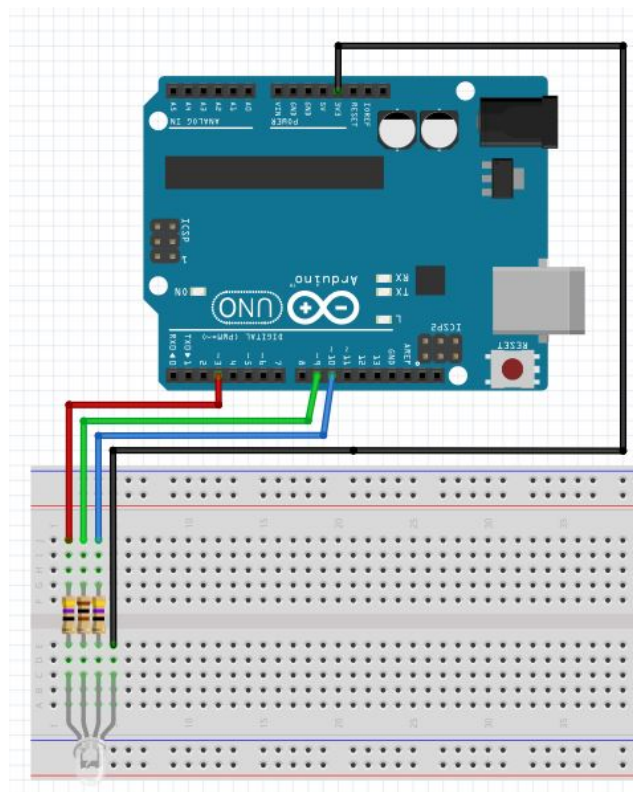
Než začneme psát o samotném řízení RGB diody, je třeba si říct, co tato dioda je. Ve své podstatě se jedná o 3 LED2.3 diody v jednu pouzdrě. Tato dioda se vyrábí ve dvou provedeních, tvojí a to buď se společnou anodou nebo katodou. Další potřebnou věcí je vědět, co je to PWM (Pulse Width Modulation), česky Pulzně šířková modulace. Princip této modulace je velmi jednoduchý. O určité frekvenci (u Arduina okolo 500 Hz), se střídá stav zapnutí a vypnutí. V rozmezí 0-255 můžeme řídit, jaký poměr bude mezi zapnutým a vypnutým stavem (hodnota 0 trvale vypnuté - zem, hodnota 255 trvale zapnuté - +5 V). Lépe se tato modulace dá pochopit z grafu průběhu signálu. V současnosti se



Obrázek 18: Pulzně šířková modulace

tato modulace využívá například při regulaci otáček ventilátoru nebo řízení intenzity podsvícení. My s touto modulací budeme měnit barvy na RGB diodě⁴.

Ve schématu si všimněte 3 různých odporů. Každá barva diody má jiné napětí. Pro správnou velikost jednotlivých odporů se podívejte do datasheetu ke své diodě. V případě, že budete mít diodu se společnou katodou, stačí ji třeba místo +3,3 V připojit na vývod GND.



Obrázek 19: Řízení RGB diody

Ve zdrojovém kódu se objevují nové prvky. Jsou to cyklus while a jedno-rozměrná pole. Jednorozměrné pole si můžeme představit jako jeden řádek tabulky, v kterém jsou určité hodnoty. V tomto případě piny, kam je připojena RGB dioda. Cyklus while je cyklus, který se provádí do doby, dokud platí podmínka v něm uvedená. Další novinkou jsou logické operátory, v našem případě logické AND, (&&) které způsobí, že podmínka je splněna, pokud jsou splněny obě její části.

⁴Dioda kterou mám k dispozici má společnou anodu proto při hodnotě 255 nesvítí a při hodnotě 0 svítí

Ukázka kódu 4: Řízení RGB diody

```

byte piny[]={3,9,10}; //Definice pole
int i=0;
int x=0;
void setup() {
}
void loop() {
barvy();
delay(5000);
}
// Metoda která bude postupně měnit barvy kterými dioda svítí
void barvy(){
// První část kódu rozsvítí všechny led v~RGB diode
analogWrite(piny[i],x);
analogWrite(piny[i+1],x);
analogWrite(piny[i+2],x);
x=255;
/*Cyklus která postupně zvyšuje jas jednotlivých led při jeho
* průběhu můžeme vidět jak led mění své barvy */
while(x!=0 && i!=3){
x=x-5;
analogWrite(piny[i],x);
delay(200);
if(x==0 && i<3){
i++;
x=255;
}
}
//Nastaví výchozí hodnoty proměnných do výchozího stavu a zhasne diodu
i=0;
x=0;
analogWrite(piny[i],255);
analogWrite(piny[i+1],255);
analogWrite(piny[i+2],255);
}

```

Komunikace s počítačem

Předchozí příklady byly sice funkční, ale my jsme neměli způsob jak si to ověřit. Jedna forma možnosti ověření komunikace je pomocí připojeného pc. Pro komunikace se využívá sériová komunikace, jako klienta na počítači můžete využít například program putty. Nyní, když se připojíme pomocí putty na port, na kterém máme připojené Arduino a zadáme číslo 1, vypíše nám sériová linka aktuální teplotu. V případě, že zadáme číslo 2 vlhkost a v případě, že zadáme cokoliv jiného, vypíše nám "Stiskni 1 pro teplotu nebo 2 pro vlhkost".⁵ Zapojení je v tomto případě stejné jako v příkladu připojení čidla DHT11.

⁵Sériová komunikace převádí znak na číslo podle ASCII tabulky, proto je v podmínce 49, což odpovídá znaku 1

Ukázka kódu 5: Komunikace s pc

```

#include <dht11.h> //Naimportuje knihovnu pro čidlo DHT11
dht11 cidlo; //Vytvoří objekt čidlo
void setup() {
  Serial.begin(9600); //Zahájí sériovou komunikaci
}
void loop() {
  cidlo.read(10); //Přečte hodnotu z čidla
  if (Serial.available()){ // Ověření dostupnost komunikace
    int input = Serial.read(); //Spustí čtení sériové komunikace
    answer(input);
  }
}
void answer(int input) //definice metody answer
{
  if (input == 49) //Podmínka pokud je vstup 1
  {
    Serial.print("Vlhkost je "); //Vypíše na sériovou linku text
    Serial.println(cidlo.humidity);
  }
  else if (input == 50) //Podmínka pokud je vstup 2
  {
    Serial.print("Teplota je: ");
    Serial.println(cidlo.temperature);
  }
  else //pokud nebyla splněna ani jedna z předchozích podmínek
  {
    Serial.println("Stiskni 1 pro teplotu nebo 2 pro vlhkost");
  }
}
}

```

Komunikace mezi moduly

K arduinu můžeme připojit mnoho modulů, počet pinů však není neomezený. Z tohoto důvodu se na komunikaci využívají různé sběrnice. Například LCD displej potřebuje ke svému fungování 6 datových vodičů (a 4 pro napájení). Abychom se vyhnuli takovému plýtvání piny, využijeme k jeho připojení obvod, k tomuto účelu přímo stvořený. Tento obvod je LCM1602 IIC. Nejprve si ale něco řekneme o I2C sběrnici.

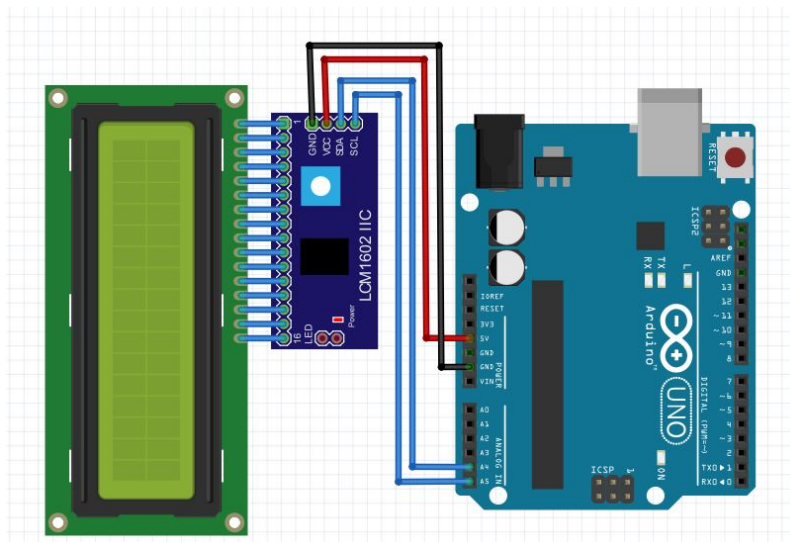
I2C

Je sběrnice vyvinutá firmou Philips Semiconductors kolem roku 1980. Její účel byl lehce propojit cpu a ostatní čipy v televizích a ostatní multimediální technice. Tato sběrnice rozděluje zařízení na dva - master a slave. Sběrnice umožňuje pomocí dvou vodičů připojit až 128 různých zařízení. V praxi se využívají ještě dva další vodiče pro napájení připojených zařízení. Dva hlavní vodiče jsou označeny SDA a SCL. Linka SDA je určena pro data a linka SCL pro hodinový signál. Nebudeme si zde popisovat přesné fungování této sběrnice, jen lehce uvedeme vše potřebné, co je třeba znát. Master zařízení

ovládá sběrnici. Každé zařízení na sběrnici má svou vlastní 7 bitovou adresu. Slave zařízení mohou vysílat i přijímat data (ovšem nemusí). Master zahájí komunikaci se slavem nastavením stavu linky na start a odesláním adresy zařízení. Všechny zařízení jsou paralelně připojeny na sběrnici.

Arduino a displej přes I2C

Jak jsem již dříve zmínil, ke komunikaci využijte připravený obvod LCM1602 IIC. Displej bude zapojen celkem pomocí 4 vodičů. Tyto vodiče jsou Napájecí (VCC), Zem (GND), SDA a SCL. Ke komunikaci s displejem využijeme knihovny LCDi2C a Wire, kterou nalezneme stejně jako knihovnu k čidlu DHT11 na oficiálních stránkách. Každé Arduino má umístěné piny na I2C jinde. U Arduino Uno, které využívám, jsou to piny A4(SDA) a A5(SCL). Na našem obvodu nalezneme trimr, který je určen k regulaci podsvícení displeje



Obrázek 20: Arduino a LCD displej přes I2C

A samozřejmě nesmí chybět ukázkový program, který nám předvede základní práci s tímto displejem.

Jak vidíte, kód je velice jednoduchý a dává nám možnost dostat informace z Arduina lehce a bez nutnosti použít počítač. Pokud by jsme například připojili čidlo DHT11, může nám displej zobrazovat aktuální vlhkost a teplotu.

Ukázka kódu 6: LCD displej přes I2C

```

#include <Wire.h> //Import knihovny Wire kvůli podpoře I2C
#include <LiquidCrystal_I2C.h> //Import knihovny k LCD displeji přes I2C
LiquidCrystal_I2C displej(0x26, 16, 2);
//Nastavení displeje adresa a poté velikost
void setup() {
  displej.begin(); //Zapnutí LCD
  displej.backlight(); //Zapnutí podsvícení displeje
}
void loop() {
  displej.setCursor( 0 , 0)
  //Nastaví kurzor displeje na 1. znak 1. řádku
  displej.print("Hrátky s~") //Vypíše na displej Hrátky
  s~displej.setCursor( 0, 1) //Nastaví kurzor na druhý řádek 1. znak
  displej.print("displejem ")//Vypíše na druhý řádek displejem
  delay(3000); //počká 3 sekundy
  displej.clear();
  //smaže obsah displeje a nastaví kurzor na 1. řádek, 1. znak
  displej.print("Hrátky s~displejem ") //vypíše na displej Hrátky s~displejem
  delay(3000);
  displej.clear();
}

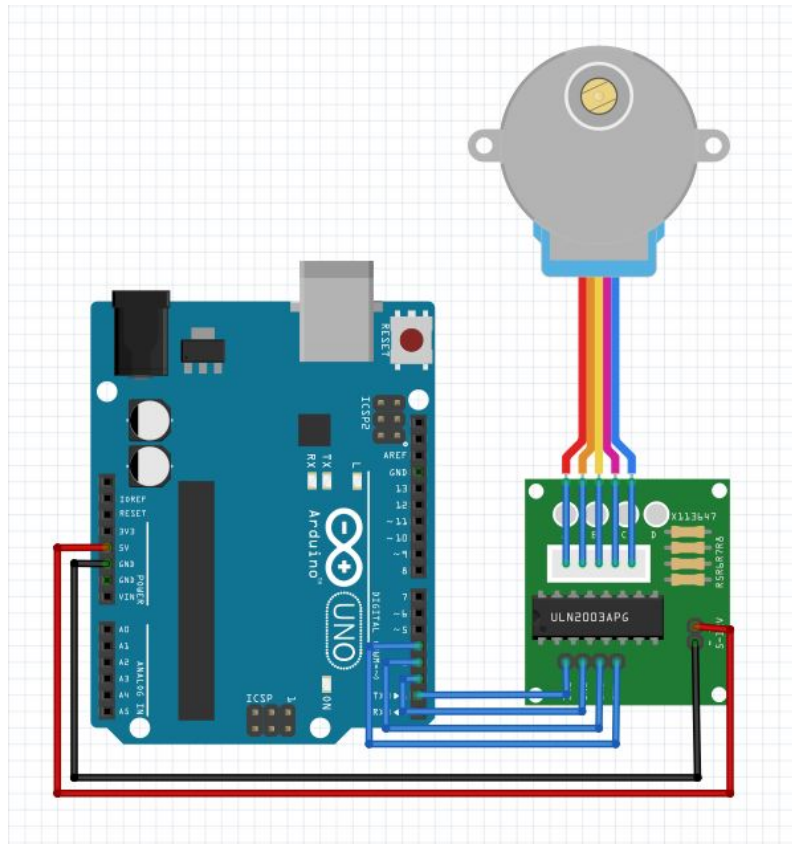
```

Arduino a krokový motor

Pro stavbu robotů i různých manipulátorů potřebuje provést občas pohyb o definované velikosti. Tento pohyb nám pomůže zrealizovat krokový motor. V našem robotovi CoreXY, je například využíváme pro pohyb platformy. Napájení výstupních pinů Arduina není ovšem dost silné, aby spínalo krokový motor přímo. Proto využijeme předpřipravený obvod. Ten využívá tranzistorové pole ULN2003. Na internetu lze zakoupit unipolární krokový motor i s tímto polem přímo. Samozřejmě můžeme si jednoduchý zesilovací obvod postavit sami, toto je ovšem jednodušší řešení.

Ze zapojení je patrné, že se nejedná o nic složitého. Z Arduina nám stačí vyvést pouhé 4 piny a napájení, v případě silnějšího motoru můžeme využít externí napájení. Zem na obvodové desce ovšem musíme propojit se zemí Arduina, jako je například u zapojení u řízení teploty. Kód programu je také velmi jednoduchý. Můžeme využít knihovnu z Arduina pro řízení krokového motoru. S ní ovšem jsou problémy, a proto jsem se rozhodl sepsat vlastní řízení krokového motoru. Aby jsme mohli řídit krokový motor, je třeba znát správnou sekvenci. Sekvence není jediná, můžeme jich použít několik. Plný krok, plný krok dvoufázově a poloviční kroky⁶. Plný krok vždy sepne pouze jeden cívkou. Tento způsob je nejméně energeticky náročný. Alternativou k němu je dvoufázový plný krok, kdy spínáme vždy dvě cívkou najednou. Výhoda tohoto řešení spočívá v cca o 30 % větším točivém momentu. Toto

⁶Dále existují mikrokroky, těmi se ovšem zabývat nebudeme



Obrázek 21: Arduino a krokový motor

řešení ovšem požaduje více energie. Poslední si ukážeme chůzi krokového motoru po polovičních krocích. To nám umožní zvýšit rozlišení motoru. Pro přehlednost přidávám tabulky jednotlivých spínání. Tato sekvence se postupně opakuje. Není nutné dodržet směr, můžeme jít po směru i protisměru hodinových ručiček.

Tabulka 1: Spínání plné kroky

Jednofázové				Dvoufázové			
1	2	3	4	1	2	3	4
1	0	0	0	1	1	0	0
0	1	0	0	1	1	0	0
0	0	1	0	0	0	1	1
0	0	0	1	1	0	0	1

Tabulka 2: Spínání poloviční kroky

Pin			
1	2	3	4
1	0	0	0
1	1	0	0
0	1	0	0
0	1	1	0
0	0	1	0
0	0	1	1
0	0	0	1
1	0	0	1

Při pohledu na tyto sekvence je vidět, že první i druhá jsou velmi jednoduché. Poloviční kroky nejsou také složité. V podstatě je to kombinace obou tabulek dohromady. A nyní již přichází na řadu samotný kód programu. V tomto kódu uvedu pouze část programu, mající na starosti jednotlivý krok vpřed.

Ukázka kódu 7: Arduino krokový motor plné kroky

```
void plnykrokJednofazovy(){ //Název funkce
  int nasledujici =2; //Definice následujícího pinu
  for(int i=1;i<5;i++){ //cyklus který se bude provádět pokud bude podmínka platná
    if(nasledujici == 5){
      nasledujici= 1;}
    //Podmínka zajišťující že následující pin po 4 je 1.
    if(digitalRead(i)==HIGH){ //načte hodnotu z~pinu
      digitalWrite(nasledujici ,HIGH);
      digitalWrite(i,LOW);
      return;}
    nasledujici++;}}
```

V případě, že by jsme řídily motor dvoufázově, stačí změnit hodnotu proměnné následující na 3. Motor by měl být před spuštěním funkce zapnutý, to znamená ve výchozím stavu. Jak je zde vidět, na řízení krokového motoru se dá při výuce v programování využít k naučení for cyklů. O něco složitější je to v případě půl-kroků. Opět zde použijeme for cyklus, jen zde přibude více podmínek. Ve všech kódech si není třeba pamatovat, jaký je aktuální krok. Vždy se provede ten správný.

Ukázka kódu 8: Arduino krokový motor poloviční kroky

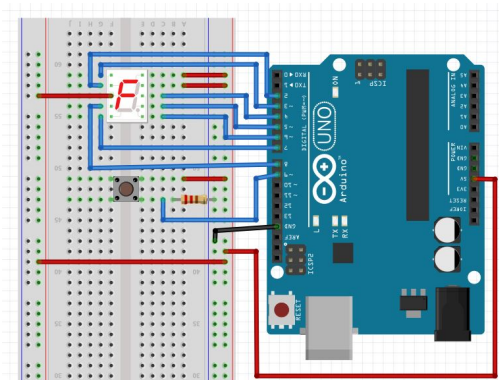
```

void pulkrok(){
  int nasledujici = 2;
  for(int i=1;i<5;i++){
    if(nasledujici == 5){
      nasledujici= 1;
    }
    if(digitalRead(i)==HIGH){
      if(i==1 && digitalRead(4)==HIGH){
        digitalWrite(4,LOW);
        //Podmínka kvůli koncovému stavu, kdyby zde nebyla,
        program by sepnul v jednom případě 3 cívky
      }
      else if(digitalRead(nasledujici)==HIGH){
        digitalWrite(i,LOW);
        return;
      }
    }else {
      digitalWrite(nasledujici,HIGH);
      return;
    }
    nasledujici++;
  }
}

```

Arduino a sedmsegmentový displej

Sedmsegmentový displej není v podstatě nic jiného, než 7 LED diod v jednom pouzdře. Jsou uspořádány tak, aby mohly zobrazovat jakékoliv číslo v rozsahu 0-9. Jsou k dostání se společnou anodou nebo katodou. K dispozici jsou i displeje s více číslicemi, ty ovšem většinou mají vlastní řadič. My si ukážeme práci s jednomístným displejem Tesla LQ450, jenž má společné anody na pinech 3, 9 a 10. Vždy si zkontrolujte dle displeje, jaký vývod je určen na kterou diodu. Navíc číslice na displeji budeme měnit pomocí tlačítka.



Obrázek 22: Arduino a sedmsegmentový displej

Zde je vidět, že budeme potřebovat doopravdy hodně vývodů z Arduina, konkrétně 8. Ostatní zapojené vývody jsou určeny pro napájení.

Ukázka kódu 9: Arduino krokový motor poloviční kroky

```
int cisla =0;
byte piny[]={2,3,4,5,6,7,8};
byte zobrazeni_cisel[11][7] ={
//Definice pole pro každé číslo jeden řádek + pokud je mimo rozsah
  {0,0,0,0,0,0,1},//0
  {1,0,0,1,1,1,1},//1
  {0,0,1,0,0,1,0},//2
  {0,0,0,0,1,1,0},//3
  {1,0,0,1,1,0,0},//4
  {0,1,0,0,1,0,0},//5
  {0,1,0,0,0,0,0},//6
  {0,0,0,1,1,1,1},//7
  {0,0,0,0,0,0,0},//8
  {0,0,0,0,1,0,0},//9
  {0,1,1,0,0,0,0},//E };
void setup() {
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,INPUT);
ukazcislo(cisla);}
void loop() {
  if(digitalRead(9)==HIGH){
    cisla++;};
    if(cisla >10){
      cisla=0;}
      ukazcislo(cisla);
      delay(300);}
  void ukazcislo(int hodnota){
    for(int i=0;i<7;i++){
      digitalWrite(piny[i],zobrazeni_cisel[hodnota][i]);}}
```

Kód je krátký a vcelku přehledný. Nejprve jsme si definovali pole, ve kterém máme nutný uvedený stav pinu pro zobrazení konkrétního čísla ⁷. V metodě loop, neustále čteme vstup z pinu 9, pokud je tlačítko zmáčklé, odpovídá to stavu HIGH. Poté zvětšíme globální proměnou o 1. Dále nechybí podmínka, pokud by tato proměnná byla větší než 10, její nastavení na hodnotu 0.

Mnohem zajímavější metoda je ovšem ukazcislo. Ta jako parametr přijme číslo, které má zobrazit. Tato metoda pomocí for cyklu zapíše na pin, dle hodnoty v poli piny, hodnotu z pole zobrazeni_cisel z řádku, odpovídajícímu číslu, které má zobrazit hodnotu 1 nebo 0. Toto osobně považuji za výbornou ukázkou práce s jedno a dvourozměrným polem.

⁷1 odpovídá HIGH, 0 odpovídá LOW

4 CAD - Computer-aided drafting

Computer-aided design, nebo-li česky počítačem podporované projektování (také se využívá pojem CADD Computer Aided Design and Drafting), je počítačový systém zaměřený na vytváření, analýzu a optimalizaci návrhů v různých oblastech. CAD software zlepšuje efektivitu návrhářů, pomáhá zkvalitnit dokumentaci a umožňuje snadné sdílení dokumentace a vytváření databází produktů. Může též pomoci při vytváření produktů návrhem optimálního řešení výroby.

CAD je použit v mnoha oborech, podle kterých můžeme CAD aplikace dělit na:

- Obecné CAD systémy
- Stavebnictví a architektura (AEC Architecture Engineering Construction)
- Strojírenství CAM (Computer-Aided Manufacturing) CAE (Computer-aided engineering)
- Správa nemovitostí (FM Facility Management)
- Elektronika EDA (Electronic design automation)
- Územní plánování a geografie GIS (Geographic Information Systems)

4.1 Historie CAD

Historie CAD aplikací sahá do 50. let 20. století, kdy došlo k vynálezu světelného pera. Jako plátno a paměť zároveň sloužilo stínítko obrazovky. Ovšem jako první CAD aplikaci můžeme označit výsledek disertační práce Ivana Sutherlanda. Tématem této práce byla aplikace počítačů v grafice a návrhu. Výsledek této práce vznikl v roce 1962 a jmenoval se Sketchpad.

Prakticky všechny aplikace až do roku 1978, využívali CAD systémy vektorové grafiky. Tato situaci se změnila a počítače začali využívat k zobrazení rastrové grafiky. Další milník ve vývoji CAD aplikací nastal v roce 1980 a postaral se o něj software ARCH MODEL. Poprvé bylo možné využívat 3D grafiky ploch, do té doby se využívali pouze drátěné modely a plochy byly vyplněny pomocí čar.

Jak šel vývoj počítačů dopředu, začali se hojně objevovat dostatečně výkonné počítače v nižší cenové hladině. I pro tyto počítače se začaly vyvíjet CAD aplikace, například i v současnosti velmi známá AutoCAD.

I když editory obsahovali i možnost 3D zobrazení, modelovalo se převážně ve 2D a 3D zobrazení sloužilo hlavně jako kontrola návrhu. V tomto 2D modelování se vytvořil základní obrys a z něj se generoval 3D model. Pokud se ukázala chyba v základním obrysu, její oprava byla velice komplikovaná a ne zřídka se stávalo, u komplikovanějších modelů, že se celý model smazal a tvořil znovu. Vzhledem k náročnosti takového přístupu byla pouze otázka času, než se objeví na tento problém řešení. S tím přišla v roce 1988 firma PRC PRO/Engineer a jednalo se o parametrické modelování. Jeho hlavní výhoda spočívá v tom, že se nahrubo navrhne tvar, okótuje se a poté se postupně zpřesňuje. Tuto koncepci brzo převzali další společnosti, jako například Dassault Systèmes SolidWorks Corporation, známá svými produkty SolidWorks.

5 Návrh paralelního robota

Než se pustíme do samotného návrhu, je třeba si stanovit účel paralelního robota, jeho očekávané parametry a vlastnosti. Tuto část si označíme jako specifikaci. Poté co budeme mít hotovou specifikaci, je třeba vybrat materiál, z kterého budeme robota stavět. I když tato část částečně vyplývá ze specifikace. Když jsou tyto základní požadavky splněny, přejde se k výběru typu konstrukce paralelního robota a následně k samotnému návrhu. Po dokončení prvního návrhu přijde jeho úprava, na základě konzultací s lidmi pohybujícími se v oblasti strojírenství a samozřejmě na základě možnosti výroby jednotlivých dílů pro tohoto robota. Až bude návrh dostatečně optimalizován, přejde se k realizaci samotného robota. Poté bude následovat test jednotlivých komponent a případné opravy v návrhu a realizaci.

5.1 Specifikace robota

Nejprve si musíme určit účel, za kterým našeho robota postavíme. V našem případě použití jako laserové gravírky, 3D tiskárny a frézky lehce obrobitelných materiálů, jako například měkké dřevo.

Nyní, když je již jasný účel, je třeba stanovit další parametry. Mezi základní parametry každého robota patří jeho operativní prostor. Velká část 3D tiskáren využívá prostor o velikosti 20x20x20 cm. Vzhledem k tomu, že my ovšem tohoto robota chceme využít i jako laserovou gravírku, hodí se nám větší rozměr. Proto operativní prostor stanovuji na 40x40x40 cm.

Mezi další specifikace patří požadavky na výrobní náklady, které jsem na konstrukci robota stanovil na maximální cenu 20 000 Kč. Dále přesnost, rychlost a snadná výroba. Samozřejmě tyto parametry chceme co nejlepší a

uvidíme, co nám jednotlivé typy nabídnou.

5.2 Typ konstrukce robota

Roboty můžeme rozlišit na dva základní typy, a to sériové nebo paralelní. Rozebírat jednotlivé typy není součástí této práce. Pro úplnost zde uvedu pouze shrnutí vlastností jednotlivých typů.

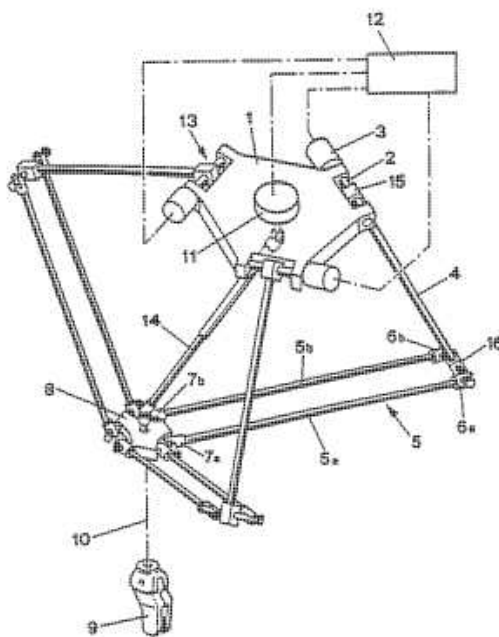
Tabulka 3: Porovnání sériových a paralelních kinematických struktur [2]

Vlastnosti	Paralelní kinematický mechanismus	Sériový kinematický mechanismus
Tuhost	Vysoká (zejména jen tah a tlak)	Nízká (zatěžování především na ohyb)
Rychlost a zrychlení stroje	Vysoké	Průměrné
Šíření chyb	Chyby jsou průměrem nashromážděných chyb v jednotlivých osách	Hromadění v jednotlivých osách
Hmotnost pohybujících se částí	Nízká (pohybují se jen rameny a vřeteno)	Vysoká (první osa je vždy namáhána osami následujícími)
Řízení (regulace)	Komplikované (celý mechanismus musí být regulován jako celek)	Jednoduchá (každá osa se reguluje jednotlivě)
Kalibrace	Komplikovaná (celková kalibrace je složitá, jelikož jednotlivé části se navzájem ovlivňují)	Jednoduchá
Kinematika	Inverzní kinematika, transformace souřadnic je zcela nutná	Přímá kinematika (je jednoduchá)
Vazba mezi osami	Osy spojené a vazba je nelineární	Jen malé vazby
Výroba a montáž	Jednoduchá (prvky jsou lehce nahraditelné)	Složitá (nákladná a časově náročná)
Flexibilita	Vysoká (zaměnitelné díly)	Malá

Zde pro nás hlavně důležité parametry jsou šíření chyb, výroba, montáž a dále flexibilita. Z toho je jasné, že se rozhodneme pro konstrukci paralelního robota. Ovšem i způsobů jak realizovat paralelního robota je několik. Proto jsem vybral dva typy, které by se mohli pro náš účel hodit. Ty porovnáme a vybereme pro nás ten nejvhodnější.

5.2.1 Delta

Robot typu Delta, využívá 3 paralelní paže k manipulaci s platformou. Vynalezl ho v 80. letech profesor Reymond Clavel. Tento typ robota je hlavně využíván v průmyslu, jako třídící robot či manipulátor. Patří mezi nejúspěšnější paralelní roboty. Za tento vynález byl v roce 1999 odměněn cenou Golden Robot Award [3].

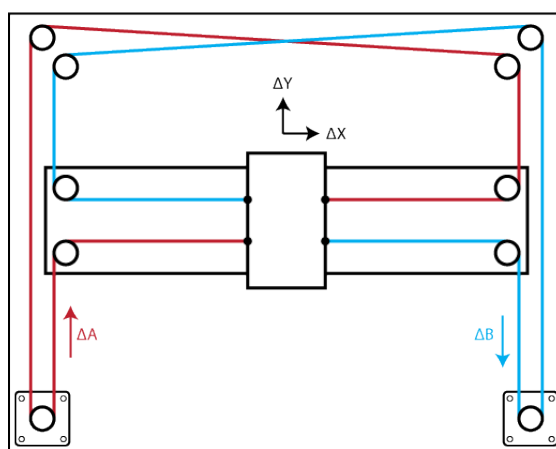


Obrázek 23: Koncepce delta robota [4]

Mezi hlavní výhody delta robota patří velká rychlost ve všech 3 osách. Jeho konstrukce se dá provést i jednodušeji, než znázorňuje obrázek z patentu. V případě upravené konstrukce je velmi jednoduchý na výrobu. Jeho řízení je ovšem složitější a nároky na ramena jsou poměrně vysoké. Navíc je vcelku omezený vahou platformy a silou ramen, proto je určen na rychlou manipulaci s lehkými předměty.

5.2.2 CoreXY

Jedná se o paralelní manipulátor, kdy jsou oba motory stacionární. V případě pohybu jednoho motoru se platforma pohybuje pod úhlem 45 stupňů v ose X a Y. Narozdíl od Delt CoreXY, neřeší pohyb v Z ose. Ta se tedy obvykle řeší zvlášť. Mezi jeho výhody patří tuhost, rychlost v osách XY, jednoduchost, a flexibilita. Operativní prostor CoreXY se dá zvětšovat pomocí výměny vodících dílů, nebo například zvětšením řemene a jedné vodící tyče, jako například v projektu Bot of the Cloth [5]



Equations of Motion:

$$\Delta X = \frac{1}{2}(\Delta A + \Delta B), \quad \Delta Y = \frac{1}{2}(\Delta A - \Delta B)$$

$$\Delta A = \Delta X + \Delta Y, \quad \Delta B = \Delta X - \Delta Y$$

Obrázek 24: Referenční mechanismus CoreXY [6]

Jak si můžeme všimnout na obrázku referenčního mechanismu, tak i ten se dopouští několika chyb po stránkách mechaniky. Například křížení řemenů není v mechanice správné, nerovnoměrným zatížením plochy řemenu dochází k jeho rychlejšímu opotřebení.

5.2.3 Porovnání

Ve stručnosti jsme si představili dvě konstrukce, které by se dali využít při konstrukci paralelního robota. Nyní nezbyvá nic jiného, než jednotlivé konstrukce porovnat a zjistit, která více vyhovuje našim požadavkům.

V případě, že by jsme se spokojili pouze s 3D tiskárnou a laserovou gravírkou, byla by delta patrně lepší řešení. Vzhledem k tomu, že v požadavcích máme i možnost využití jako frézky na lehčí materiály, se

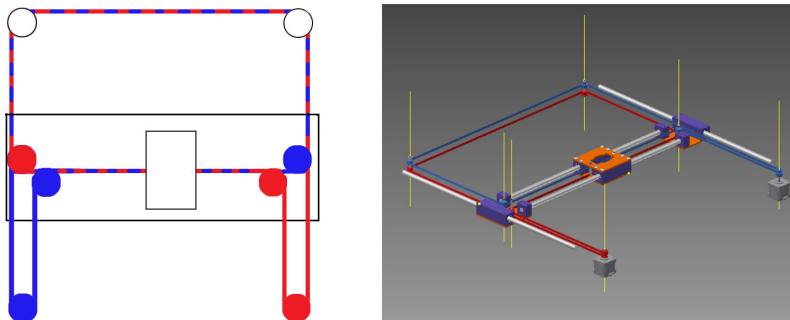
Tabulka 4: Porovnání Delta a CoreXY

Vlastnost	Delta	CoreXY
Zatížení platformy	Malé	Vysoké
Rychlost	Vysoké	Vysoké pouze v XY
Tuhost	Menší	Větší
Obtížnost řízení	Složitá transformace souřadnic	Jednoduchá transformace
Možnost rozšíření	Pouze v Z ose	Ve všech osách
Náročnost na konstrukci	Malá	Malá
Další nevýhody	Žádné	Nutno vytvořit Z osu

ovšem situace mění. Kvůli možnosti nízkému zatížení platformy a též kvůli menší tuhosti, a tím pádem i menší odolnosti na vibrace, jsem jako vítěze porovnání stanovil CoreXY

5.3 Úprava referenčního mechanismu CoreXY

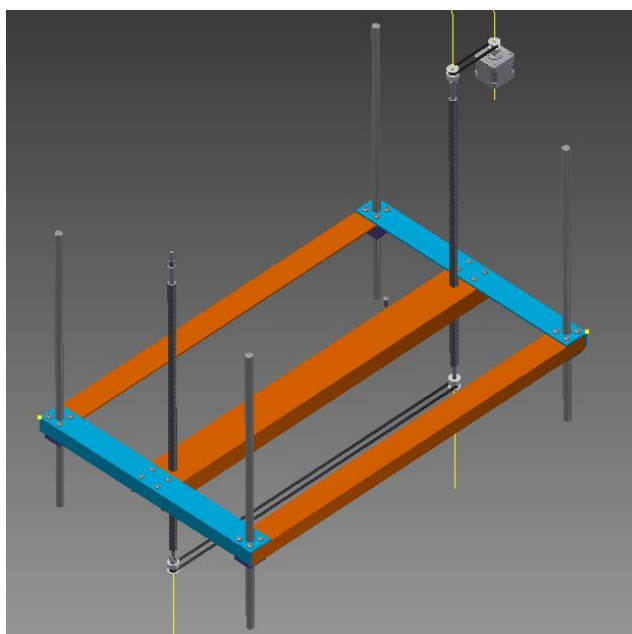
Jak jsem již psal, referenční návrh paralelního robota CoreXY se dopouští jedné velké chyby a tou je křížení řemenů. Abych se nedopustil stejné chyby, rozhodl jsem se upravit referenční mechanismus tak, aby obsahoval výhody CoreXY a zároveň se v něm nekřížili řemeny. Umístění řemenů jsem upravil tak, aby byly chyceny vždy za střed platformy. Tím jsem zmenšil počet os, přes které vedou řemeny a zajistil tím větší symetrii celého návrhu.



Obrázek 25: Upravený mechanismus CoreXY

5.4 Osa Z

Obvykle se u mnoha 3D tiskáren využívá na osu Z dvou paralelně zapojených krokových motorů, na které jsou závitové tyče, svým pohybem poté zvedají desku. Toto řešení má několik úskalí. Jedno z nich například nastává při zapnutí krokových motorů. Občas se stane, že jeden z motorů "cukne" a tím pádem se začne deska křížit. Proto jsem se rozhodl využít pro pohon pouze jeden motor a dva trapézové šrouby. První šroub bude napojena na horní straně řemenem s převodem 1:1 na krokový motor, na její spodní části bude řemenem převodem 1:1 napojen na druhý trapézový šroub. Na těchto trapézových šroubech bude umístěna matka, a na ní bude umístěna základna pro desku určenou k uchycení buď opracovávaného materiálu nebo deska pro tisk. Každý trapézový šroub bude mít umístěnou zleva i zprava vodící tyč, která bude omezovat síly, které by jinak působily na šroub.



Obrázek 26: Osa Z

5.5 Výběr materiálu

V současné době se využívá pro stavbu 3D tiskáren a laserových gravírek mnoho různých materiálů, od plastů přes laserem řezané dřevo po kov. Musíme si však uvědomit naše požadavky, mezi kterými je i frézka lehčích materiálů. Z tohoto důvodu jsem se rozhodl, mít celou tiskárnu provedenou

z kovů. Po dlouhém vybírání jsem se rozhodl jako stavební materiál využít hliníkové profily. Mezi jejich přednosti patří nízká hmotnost, slušná tuhost, dobrá dostupnost a cena.

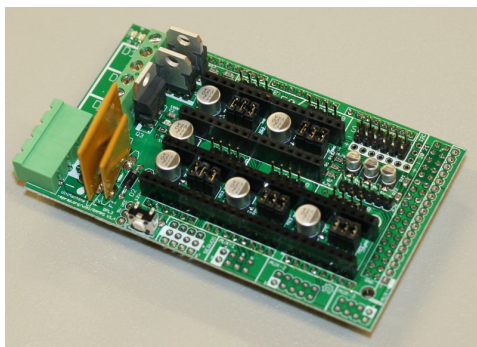
5.6 Výběr řídicí elektroniky

I když máme po mechanické stránce jasno, ještě je třeba vybrat vhodnou elektroniku pro našeho paralelního robota. Samozřejmě je i možnost si postavit vlastní elektroniku na míru, ovšem toto řešení je značně komplikované a náročné, proto jsem ho zavrhl. Hlavní úkol elektroniky nespočívá pouze v ovládní patřičných motorů, ale také v ovládní nástrojů, které na paralelního robota umístíme. Když vezmeme v potaz běžnou 3D tiskárnu, potřebuje ke svému správnému fungování minimálně 2, optimálně 3 tepelné výstupy. Jeden je určen pro tavení materiálu, druhý pro ohřev podložky a třetí pro ventilátor.

Mezi nejpopulárnější elektroniky pro tento účel patří Arduino Mega v kombinaci s RAMPS. Dále v současnosti nabírá na popularitě Smoothie-board a jako další elektroniku, která by připadala v úvahu jsem zvolil Duet.

RAMPS

RAMPS je zkratka pro RepRap Arduino Mega Pololu Shield, jak již z názvu vyplývá, jedná se o rozšíření vývojové desky Arduino a to konkrétně pro Arduino Mega. Využití tohoto shieldu se neomezuje ovšem pouze na 3D tiskárny, ale je možné ho využít v jakémkoli projektu. Mezi jeho hlavní výhody patří kvalitně zpracovaná dokumentace, cena a v neposlední řadě rozšířenost. Díky ní je kompatibilní s celou řadou firmwarů určených pro 3D tiskárny, laserové gravírky atd. Tato deska je ovšem, na rozdíl od ostatních v mém výběru, neosazena moduly pro krokový motor.



Obrázek 27: Elektronika RAMPS

Tabulka 5: Vlastnosti RAMPS

Procesor	Dle Arduina
Max. krokových motorů	5
Počet teplotních čidel	3
Počet výkonných výstupů	3 všechny PWM
Počet endstopů	6
Podpora	Velká
Firmware	Mnoho
Možná rozšíření	LCD displej, SD karta, bluetooth modul
Základní výbava	Nic
Cena	sada s LCD 1290 Kč

SmoothieBoard

SmoothieBoard začal jako mnoho současných projektů na serveru kickstarter.com 27. září 2013. Během 48 hodin zvládla vybrat požadovanou částku 20 000 dolarů. Druhého března 2014 začali odesílat všechny předobjednané desky a 21. března byly již odeslány. Zároveň se objevila možnost k zakoupení těchto desek. Aktuální cena této desky se pohybuje kolem 110 Euro, za variantu určenou k řízení 4 krokových motorů. Mezi hlavní výhody této desky patří integrovaný slot na SD kartu a ethernet. Samozřejmě procesor je výrazně výkonnější než typické Arduino.

Tabulka 6: Vlastnosti Smoothieboard

Procesor	NXP LPC 1768/9 32-bits Cortex-M3 96-120 Mhz
Max. krokových motorů	5
Počet teplotních čidel	4
Počet výkonných výstupů	4
Počet endstopů	6
Podpora	Průměrná
Firmware	Oficiální, modifikovatelný
Základní výbava	Ethernet, čtečka SD karet
Možná rozšíření	Displej, I2C Serial
Cena	cca 3 300 Kč

Duet

Duet je elektronika, která je kompatibilní s Arduino Due, což ovšem neznamená, že ho potřebuje ke svému chodu podobně jako RAMPS, nýbrž že ho v sobě obsahuje. Tuto desku vyvinul Andy Hingston a Tony Lock ve spolupráci s komunitou RepRapPro a radamy od Chrise Palmer.[8] Tato elektronika, stejně jako ostatní výše uvedené elektroniky, jsou Open Hardware, což znamená, že je k dispozici kompletní návrh elektroniky a každý kdo má potřebné nástroje, si jí může vyrobit.

Tabulka 7: Vlastnosti Duet

Procesor	SAM3X8E 32-bits
Max. krokových motorů	5
Počet teplotních čidel	4
Počet výkonných výstupů	3 všechny PWM
Počet endstopů	7
Podpora	Průměrná
Firmware	2
Možná rozšíření	Displej, DUEX4
Základní výbava	Ethernet/Wi-fi, čtečka SD karet
Cena	cca 3 300 Kč

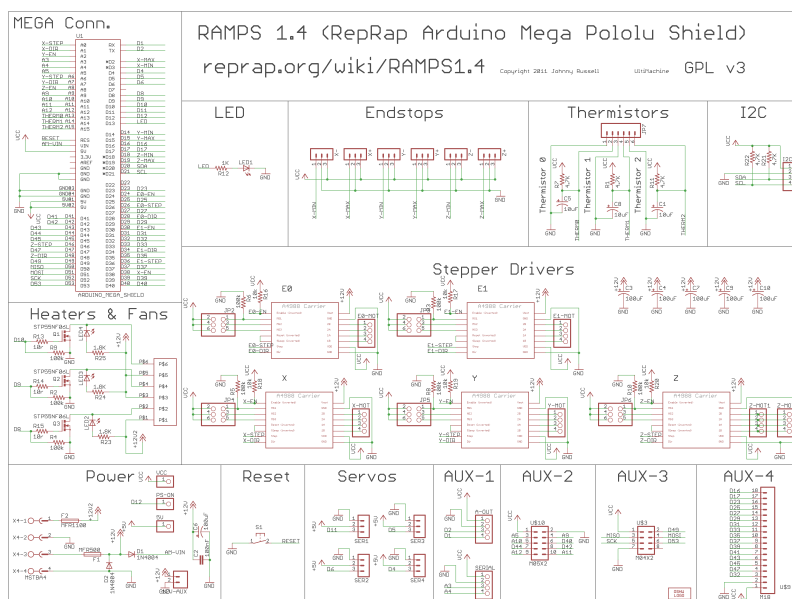
Aktuální verze desky je 0.85, která se prodává ve dvou verzích, a to klasický Duet nebo DuetWifi. Verze Duetwifi se liší výměnou ethernet portu za wifi kartu. Nejzajímavější ovšem na této elektronice je její příslušenství. Vyjma obvyklého zde nalezneme desku DUEX4, která rozšiřuje desku Duet o další 4 výstupy na krokové motory, samozřejmě včetně 4 výkonných výstupů, 4 vstupy pro teplotní čidla a přidává další 4 porty pro endstopy. To ve výsledku pro 3 osou 3D tiskárnu znamená možnost ovládat až 6 extrudrů.⁸ Samozřejmě můžete využít tyto výstupy pro postavení robota majícího více os.

Porovnání

Vezmeme-li v potaz užití paralelního robota, jako nejvhodnější varianta elektroniky nám vychází RAMPS. Jako hlavní výhodu vidím cenu a dostatek portů pro vše, co potřebuji. Ostatní desky jsou sice v mnoho parametrech

⁸Extruder je u 3D tiskáren sloužící k dávkování materiálu do trysky a jeho ohřevu

lepší, ovšem pokud tyto vlastnosti nevyužijeme, jsou pro nás zbytečné. Navíc v případě, že bychom chtěli robota rozšířit o některé populární funkce, jeví se mi jako lepší varianta využití Raspberry Pi jako kontrolního počítače, který by navíc mohl přinést mnoho dalších užitečných funkcí. Například sledování práce zařízení za pomoci kamery atd. V budoucnosti, pokud bych chtěl robota rozšířit o více os, či využít více extrudrů, rozhodně bych sáhl po desce Duet a jejímu rozšíření DUEX4. Pro úplnost přikládám schéma vybrané elektroniky



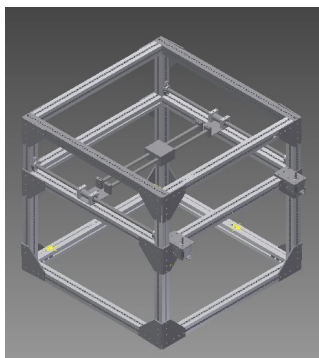
Obrázek 28: Schéma elektroniky RAMPS [10]

5.7 Nákresy

Prakticky všechny nákresy a nápady končily na papíře. Vzhledem k rozsahu projektu bylo nutno vybrat CAD aplikaci, v které budu návrh provádět. Vzhledem k mým zkušenostem ze střední školy, jsem zvolil Autodesk Inventor 2015. Práce ve všech aplikacích podobného zaměření se v zásadě nijak neliší. Ovšem v čem mě Inventor nemile překvapil, byl problém s vazbami. Vazby s rostoucí složitostí návrhu přestávali fungovat.

V návrhu jsem postupoval nejdříve hrubým návrhem nosné konstrukce, kterou jsem posléze osadil vodícími tyčemi. Následovala konstrukce funkčních dílů a následné konzultace s vedoucím mé práce, nebo s lidmi pohybujícími se v oboru strojírenství. Následná úprava modelu, opětovné konzultace a úprava, dokud jsme návrh každé komponenty dostatečně neoptimalizovali.

Zejména jako problém se objevila vyrobiteľnosť jednotlivých komponent, proto jsem se konstrukci snažil dělat jednoduchou a v rámci mých výrobních možností.



Obrázek 29: Hrubá konstrukce

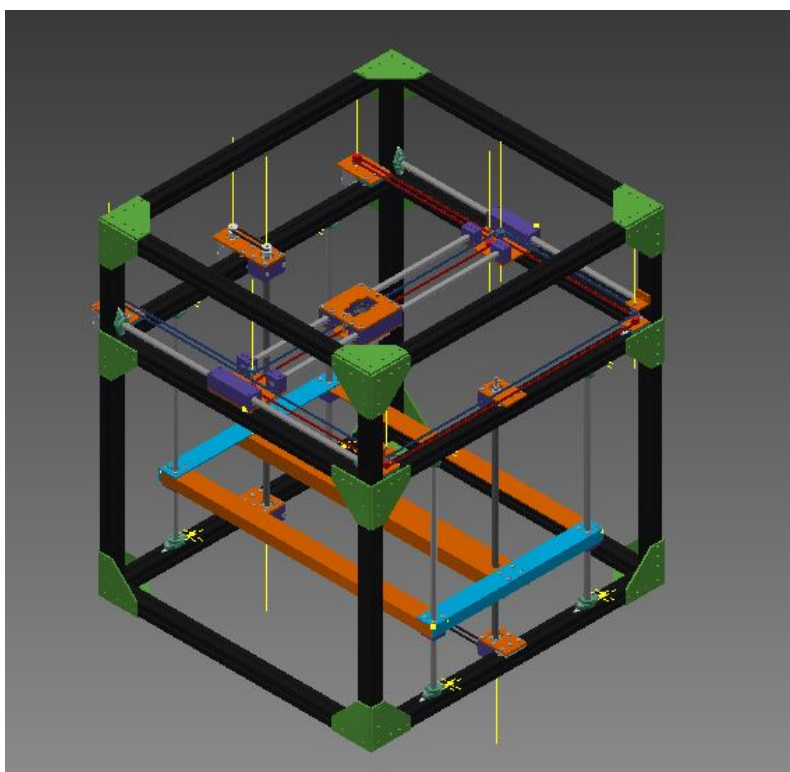
Z daného obrázku je patrné že hrubá konstrukce byl pouze jen hrubý návrh. Po úpravě jednotlivých komponent, způsobem popsaným výše, přišla nakonec stanovení finálních rozměru tiskárny, tak aby byl splněn zadávací parametr na rozměr operativní plochy 40x40x40 cm.

Většina aktivních komponent již měla správné rozměry, bylo však nutné změnit rozměry nosné konstrukce. Tato úprava znamenala vytvoření nové sestavy, která se postupně osazovala jednotlivými částmi. Následovala poslední kontrola návrhu, zejména finální kontrola složitosti. Samozřejmě se ukázalo mnoho chyb, které bylo třeba opravit a zajistit. Například na ose Z, mezi trapézovými šrouby musí být taková vzdálenost, aby se na ní vyráběl nekonečný řemen. Jednotlivé nákresy komponent i celou sestavu naleznete v digitální podobě na cd, přiloženém k této práci.

Před samotnou realizací bylo třeba samozřejmě třeba zajistit výrobu jednotlivých komponentů. To často znamenalo obvolávání firem poskytující potřebné služby nebo zdlouhavou emailovou komunikaci. I když se tato část nezdá složitá, tak zabrala skoro týden čistého času, což znamená měsíc, než bylo vše připravené k objednání. Některé části byly řezány laserem, další bylo třeba objednat kvůli místní ceně nebo dostupnosti z ciziny. Tento paralelní robot v sobě obsahuje části vyrobené v České republice, Číně nebo například USA. Lze říci, že tento paralelní robot obsahuje komponenty z celého světa.

5.8 Skládání paralelního robota

Realizace sebou nese vždy řadu neočekávaných problémů, které při ní nastanou. Jako první a poměrně náročný problém na řešení se objevil použitý



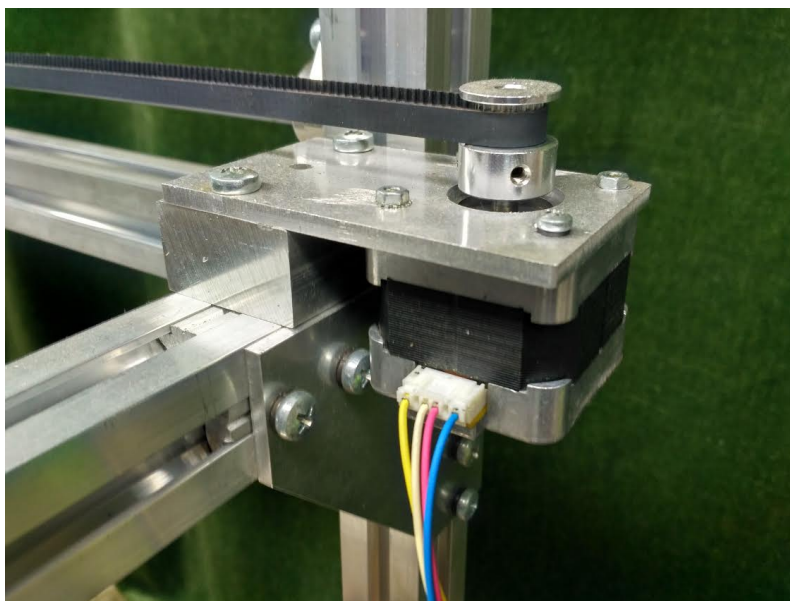
Obrázek 30: Konstrukce před realizací

materiál a tím byl hliník. Je to velmi měkký materiál, který je špatně obrobitelný. Tento problém se projevil zejména u matek určených do profilu. Tyto matky chodí jako nevyvrtaný profil, který je třeba nařezat na požadované délky, posléze provrtat a vytvořit v něm závit. Bohužel jsme zjistili, že bruskou tyto profily nejdu řežat, jelikož se v místě řezu hliník roztaví a ztráta na řezu by dosahovala skoro jednoho centimetru. Řešení tohoto problému byla přímočará pila s plátkem, určeným přímo na hliník. I když profil na matky je tenký a má zhruba rozměry 10 mm x 7 mm, i tak se musel při řezání chladit lihem.

Nejednalo se však o poslední komplikaci, která se týkala matek. I když jsem je vrtal ve stojanové vrtačce, nebyl jsem schopen docílit přesné roztoče 4 děr (nutná přesnost minimálně 0,5 mm). Z toho důvodu došlo na řezání matek na kousky o délce 1,5 cm a jejich následnému vyvrtání. Celý proces výroby matek zabral zhruba 3 dny práce a pro příště využiji matky rovnou kupované, které sice vychází cenově o něco draž, ovšem časová úspora je velká a navíc se dají koupit matky z odolnějšího materiálu jako například pozinkované železo.

Další problém, který vznikl využitím motorů z použitých jehličkových tiskáren, byly jejich osazené řemeničky. Tyto řemeničky byly na osy krokových motorů nalisovány a bylo potřeba je sundat a nasadit jiné. Po mnoho neúspěšných pokusech jsme objevili asi jako nejjednodušší způsob dané řemeničky zbrousit a poté při zafixované osičce pomalu sundávat.

Nastal další problém, který souvisel s atypickou délkou osiček u těchto krokových motorů. U klasického motoru má osička krokového motoru vyčnívající délku 24 mm [7]. U těchto motorů ovšem osička byla kratší a měla vyčnívající délku 13 mm. Což znamenalo podložit pravý motor tak, aby výška odpovídala výšce zbytku dráhy řemene. Naštěstí jsem při objednávání materiálu počítal s možnými chybami v návrhu, a proto jsem měl nakoupen i materiál, z kterého jsem udělal podložku pod motor (stačila pouze jedna podložka, na ostatních motorech nebyla třeba jelikož se počítalo s nasazením řemenic velice nízko).



Obrázek 31: Přizvednutý motor

Další problém nastal při umístění ložisek určených pro vedení řemene. Tyto ložiska jsou umístěna na řemenu, v návrhu byla ložiska dána do požadované výšky pomocí trubičky z plastu a podložena tenkou podložkou. Toto řešení nakonec nebylo třeba, jelikož ložiska šly umístit na osičku tak špatně, že jsem je tam musel "nalisovat". Tento proces spočíval ve vytvoření závitů o délce 10 mm na osičce, poté nasazení ložiska (při vytváření závitů se původní průměr osičky z 5 mm změnil na maximálně 4,9 mm), umístění podložky a šroubováním matky na závit se ložisko lisovalo dolů, dokud nebylo

v požadované výšce.

Mezi problémy, které byly zaviněny na stranách dodavatele, byl špatně zpracovaný hliníkový U profil, který měl na sobě vyvrtané díry se špatnou roztečí. Tuto chybu jsem však byl schopen rychle opravit za pomoci stojanové vrtačky.

Jedna z věcí, na kterou jsem při návrhu zapomněl, byl spoj řemenů u platformy. Toto spojení jsem nakonec realizoval pomocí "klubalek".⁹ Přes prvotní obavy, zda-li mají klubalky dostatečnou sílu tlaku na udržení řemene v pohybu, po otestování funkčnosti se tyto obavy ukázaly jako zbytečné. Klubalky svou funkci splnily bez jakéhokoliv problému.

A asi jako poslední věc, na kterou jsem v návrhu úplně zapomněl, byly "nožičky". Problém chybějících "nožiček" se objevil při úhlování nosné konstrukce, která po hladkých hlavičkách šroubů klouzala po stole. Jako řešení bylo použito plát železa, na něm přivařena matka a do ní zašroubovaná nožička z váhy Tonava typ TC-33.

Vyjma výše uvedených problémů se žádná další komplikace při sestavování konstrukce paralelního robota neobjevila, což mě osobně velice překvapilo.

5.9 Ovládací software a výběr firmware

Stejně jako elektroniku pro tiskárny, kterou si můžeme poskládat samy, můžeme si i sami napsat řídicí firmware. Je třeba si však uvědomit všechny úlohy řídicího firmware. Tento firmware nemá na starosti pouze ovládání krokových motorů, ale funguje jako interpret G-kódu.

G-code je číselný programovací jazyk určený k řízení robotů, CNC strojů atd. Tento programovací jazyk vznikl v 50 letech na MIT. Standardizace se mu dostala začátkem 60. let v USA, finální verze byla uvolněna v roce 1980. Je však potřeba říci, že plně standardizovanou verzi využívá pouze velice málo strojů a aplikací. Z čehož plyne další problém co se týče vlastní implementace. Je nutné být obeznámen nejen s příslušnými kódy a ty implementovat do firmwaru, ale je třeba též znát generátory, které se budou využívat pro naše aplikace. Například některé aplikace generují kód na puštění laseru jako M3 a některé jako M103. Dále je mnoho instrukcí, které jsou konkrétní k danému typu zařízení a ve standardizovaném G-codu nejsou.

Další nutné věci které je potřeba mít ve firmware, je nastavení kontroly chlazení nebo odsávání, pomocí hlídání různých teplot. Též spolupráce s různými perifériemi jako například LCD displej, vstup dat z paměťové karty atd. Z těchto důvodů jsem se rozhodl využít již existující firmware. Vyzkoušel

⁹ jinak řečeno kolíčků na prádlo

jsem více firmwarů, ale vybral jsem tři, které si blíže představíme a vybereme si takový, který nejvíce bude vyhovovat našim potřebám.

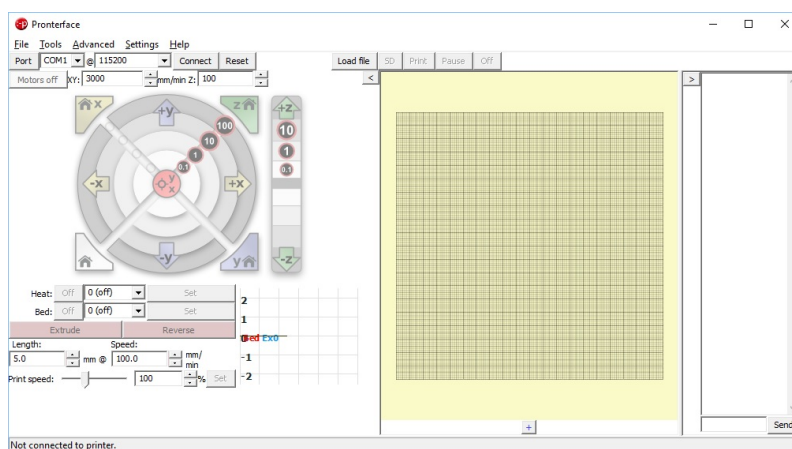
Firmware budeme do Arduina nahrávat pomocí počítače s nainstalovaným linuxovým operačním systémem Debian, ve verzi Wheezy. Aby jsme měli počítač připravený na instalaci firmware, nainstalujeme si nejdříve potřebné balíčky pro komunikaci s deskou.

Ukázka kódu 10: Instalace základních balíčků pro komunikaci s Arduinem

```
su
aptitude install avrdude gcc-avr avr-libc
```

V případě že používáme v systému sudo, využijeme pouze druhý příkaz, před který napíšeme sudo.

Pro testování firmware jsme si připravili dva různé podklady, a to model kostky určený k 3D tisku, generovaný z modelu pomocí aplikace Slic3r a dva nápisy určené k laserovému gravírování, generovaná pomocí pluginu určeným pro aplikaci Inkscape. Tyto G-cody budeme nahrávat pomocí aplikace Pronterface. Na platformu paralelního robota byla umístěna tužka, která kreslí na papír umístění pod ní.



Obrázek 32: Prostředí Pronterface

Všechny firmware nabízí možnost ruční editace konfiguračních souborů. My se ovšem podíváme na jejich konfigurační nástroje, které mají za úkol tuto konfiguraci značně zjednodušit.

5.10 Teacup

Jako první firmware, který mne zaujal a také firmware, který posloužil při kontrole funkčnosti čerstvě postaveného paralelního robota. Celý tento projekt odstartoval 20. ledna 2010 Triffid Hunter příspěvkem na blogu, v kterém vyjádřil nespokojenost se současnou situací nad firmwarem pro jeho elektroniku Sanguiono. Tento oficiální firmware byl napsán v C++ a počítal s plovoucí desetinnou čárkou. Triffid Hunter tento firmware přepsal do programovacího jazyka C a upravil tak, aby počítal s celými čísly. Vzhledem k čistotě jeho kódu a znatelnému zrychlení práce s jeho robotem, mu začala s firmwarem pomáhat komunita a brzy se zrodil firmware FiveD.

Jak plynul čas, objevovalo se mnoho různých elektronik. Stejně tak se rozšiřoval i tento firmware, který se přejmenoval na jeho současný název Teacup.

Firmware podporuje tyto elektroniky:

- Arduino Mega/RAMPS
- RADDs a jeho klony
- Gen7
- Gen3
- NanoHeart
- Teensy 2.X/3.X

Samozřejmě nejste nuceni využít tyto elektroniky v jeho konfigurátoru a stejně tak u ostatních firmware si lze vytvořit návrh pro svou elektroniku, ovšem elektroniky uvedené v seznamu spolehlivě s daným firmware chodí.

Instalace a konfigurace Teacup

Máme několik možných postupů instalace, buď přímé stažení firmware z webu, nebo doporučené řešení, naklonování git repozitáře. Před samotným klonováním si nejprve ovšem nainstalujeme potřebné balíčky pro konfigurační nástroj. Ten je psán v pythonu, který nejdříve budeme muset nainstalovat. To provedeme následujícím příkazem

Nyní se nám otevřel nástroj pro konfiguraci našeho zařízení. Jako první věc, kterou je potřeba udělat je, vybrat si v menu podobnou konstrukci tiskárny jako využíváme. To uděláme v položce File, umístěné vlevo nahoře a vybereme možnost Load Printer. Poté opět otevřeme menu File a vybereme možnost Load Board, kde si vybereme verzi naší elektroniky.

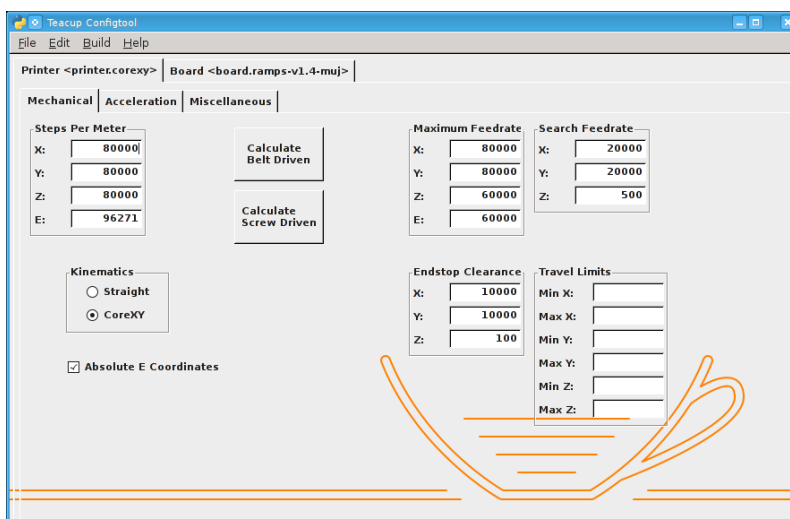
Ukázka kódu 11: Instalace Teacup

```

su
aptitude install python-wxgtk3.0
//Po nainstalování balíčku přejdeme si naklonujeme aktuální git repozitář.
git clone https://github.com/Traumflug/Teacup_Firmware.git
//Pomocí příkazu cd se přepneme do adresáře s repozitářem.
cd Teacup_Firmware
//Samotnou konfigurační utilitu spustíme následujícím příkazem
python configtool.py

```

Poté musíme upravit nastavení kinematicky. To provedeme vcelku jednoduše, stačí pouze zaškrtnout přepínač CoreXY na záložce Mechanical. Mezi další skvělé vlastnosti tohoto konfiguračního nástroje patří integrovaný kalkulátor na šrouby a řemeny, dle krokových motorů a využitých řemenic.



Obrázek 33: Teacup konfigurační nástroj

Po využití těchto nástrojů, doplnění výsledků z nich do nastavení, a v případě že jsme zvolily správnou verzi elektroniky můžeme uložit naše nastavení. To provedeme tak, že vybereme položku v menu File a poté zvolíme Save config.h. Po tomto kroku je třeba daný firmware zkompilovat a následně nahrát do naší elektroniky. To provedeme v menu Build, pomocí položky Build a následně položky Upload. Elektronika zabliká a firmware je připraven v tiskárně.

Testování firmware Teacup

Testování probíhalo pomocí připojeného laseru na tiskárně a smrkové destičky položené na podložce. Jako první přišla na řadu g-code, genero-

vaný aplikací Slic3r, určený pro 3D tisk. První problém nastal s nastavením tiskové plochy. Softwarové endstopy nevzaly v potaz kinematiku CoreXY a zmenšily nastavenou tiskovou plochu asi o polovinu. Tento problém jsem však opravil jednoduchým nastavením větší tiskové plochy.

Kód generovaný přes aplikaci Slic3r vyjel bez problému, rychlost pohybu platformy odpovídala nastavení provedeném v G-codu. Na řadu přišel kód generovaný přes plugin v aplikaci InkScape. První nápis vyjel bez problémů, ovšem místo druhého nápisu byla pouze čára. Po chvíli pátrání jsem zjistil, že tento firmware neumí G2/G3 kódy, určené pro tisk křivek.

5.11 Aprinter

Aprinter začal vyvíjet a stále vyvíjí Ambroz Bizjak. Podobně jako Teacup, je určen pro mnoho různých elektronik a mnoho různých typů kinematik (v podpoře nechybí samozřejmě CoreXY mezi další může zmínit například delta). V jeho firmware je i experimentální podpora pro ovládání laseru, umožňující měnit jeho intenzitu, pomocí PWM. Celý firmware je napsán v programovacím jazyce C++. Mezi hlavní výhody tohoto firmware patří velká podpora různých kinematik.

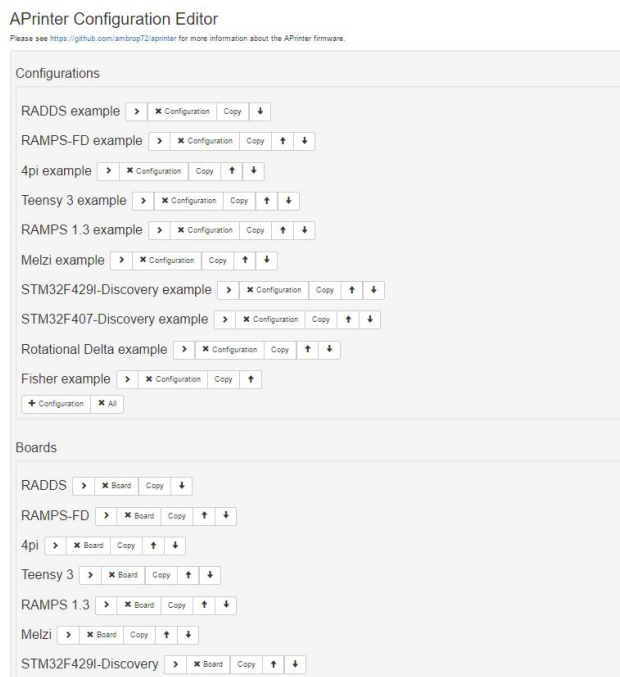
Firmware podporuje tyto elektroniky:

- Arduino Mega/RAMPS
- RADDs a jeho klony
- 4pi
- Melzi
- Teensy 3.1/3.2
- STM32F4-Discovery

Instalace a konfigurace Aprinter

Konfigurace firmwaru se provádí skrze webové stránky. Je možné stáhnout si webové stránky a mít konfigurační nástroj lokálně na počítači nebo využít veřejný server, běžící na webové adrese www.aprinter.eu. Této možnosti jsme využili a na konfiguraci firmware nám stačil libovolný webový prohlížeč.

Z obrázku je patrné, že webový konfigurátor nepatří mezi uživatelsky přívětivé. Sice zde naleznete všechny možné konfigurace, včetně úpravy pinout, ovšem je velice chaotický a občas vykazuje chyby. Problém s konfigurací nastal při nastavení tiskárny na CoreXY. Při zaškrtnutí tohoto pole je třeba



Obrázek 34: Aprinter konfigurační nástroj

přejmenovat motory z výchozího názvu X,Y,Z na A,B,Z a u každého navíc nastavit možnost "Is cartesian" na hodnotu "No". Na nastavení rozsahu velikosti tiskové plochy musíme využít následující vzorec $Arange = [Xmin + Ymin, Xmax + Ymax]$, $Brange = [Xmin - Ymax, Xmax - Ymin]$ [9].

Instalace firmwaru provedeme v terminálu následujícím příkazem.

Ukázka kódu 12: Instalace Aprinter

```
avrdude -p atmega2560 -P /dev/ttyACM0 -b 115200 -c stk500v2 -D -U
"flash:w:$HOME/aprinter-build/aprinter-nixbuild.hex:i"
```

Testování firmware Aprinter

Test probíhal stejně jako u Teacupu, to znamená nejdřív G-code připravený přes aplikaci Slic3r a poté G-code z aplikace Inkscape. S prvním kódem neměl paralelní robot s výchozím nastavením žádný problém. Problém ovšem nastal při zvýšení rychlosti, kdy se zdálo, že firmware nestíhal a nereagoval na příkazy stop, což mě velice nemile překvapilo. Na tomto je jasně vidět, že optimalizace kódu není stejně kvalitní jako například u firmwaru Teacup.

U druhé sady G-codu se objevil stejný problém jako u Teacupu a to nefunkčnost G2/3 kódu. Vzhledem k vyšším nárokům na řídicí elektroniku, než má například Teacup, nepřehledný konfigurační nástroj, nemohu tento firmware nikomu doporučit.

5.12 Repetier

Tento firmware pochází z Německa a jeho vznik se datuje na rok 2011. Jak autoři přiznávají, v těchto letech nebylo mnoho kvalitních firmware pro jejich 3D tiskárnu a proto se rozhodli napsat firmware vlastní. Z jejich hobby se stala práce na plný úvazek a proto založili firmu Hot-World GmbH. V současnosti nenabízí pouze firmware, ale také Repetier-Host, což je aplikace, která v sobě kombinuje aplikaci na převod modelu z 3D do g-kódu (aktuálně v sobě zahrnuje 3 tyto různé aplikace a to Slic3r, CuraEngine a Skainforge). Mezi jejich další software patří Repetier-Server, aplikace, která umožňuje dálkovou kontrolu 3D tiskárny.

Samozřejmostí v tomto firmwaru je podpora pro paměťové karty, ale také například pro displej, přes který lze provést úpravu nastavení tiskárny a zároveň ukazuje stavové informace. Podpora různých elektronik je zde také velmi rozsáhlá. Firmware podporuje tyto elektroniky:

- Arduino Mega/RAMPS
- Gen 6
- Sanguinololu
- Gen7 1.1 s 16 nebo 20 MHz
- Teensylu
- Printrboard
- MegaTronics
- RUMBA
- Rambo
- RADDs Shield for Arduino Due

Instalace a konfigurace Repetier

Repetier, podobně jako Aprinter, lze konfigurovat pomocí úpravy zdrojových souborů nebo přes webovou utilitu. Konfigurace je více přehledná a rozdělena do několika záložek. Tyto záložky jsou General, Mechanics, Tools, Features, User Interface, Manual a Download. Toto webové rozhraní mi přišlo velice jednoduché a přehledné, nabízející lepší roztřídění položek než konfigurační nástroj Teacupu.

Repetier-Firmware configuration tool for version 0.92.9 version

Start	General	Mechanics	Tools	Features	User Interface	Manual	Download
Configuration level	Normal, hide only internal settings						
Processor	Atmel 8-bit based board (e.g. Arduino Mega)						
Motherboard	RAMPS 1.3/RAMPS 1.4						
Printer type	Cartesian printer						
EEPROM usage	EEPROM Set 1						
<small>If you enable eeprom, you can change the most important parameter after installation over the host. Please be aware that the eeprom values overwrite settings in Configuration! To overwrite existing settings, select a different eeprom set.</small>							
Primary Port	Default port						
Baud rate	115200 ANSI						
<small>If you intend to use the printer from a linux pc, select a serial baud rate.</small>							
Bluetooth serial port	No bluetooth connected						
Baud rate bluetooth	115200 ANSI						
<small>If you intend to use the printer from a linux pc, select a serial baud rate.</small>							
Kill method	Reset controller: Will not reset separate communication chips!						
Startup GCode							
Dimensions							
X min position	0	[mm]					
Y min position	0	[mm]					
Z min position							

Obrázek 35: Repetier konfigurační nástroj

Po dokončení konfigurace a stažení firmwaru, spustíme Arduino IDE a otevřeme soubor Repetier.pde. Arduino IDE otevře i všechny ostatní potřebné soubory. Poté dáme zkompilovat a nahrát. Tím je firmware nainstalován.

Testování firmware Repetier

Test tohoto firmware probíhal nad očekávání dobře. Ani s jedním G-codem neměl tento firmware sebemenší problém a jako jediný v našem testu zvládl i G2/3 kódy, které najdou své využití zejména u laserové gravírky.

5.13 Srovnání

Po představení jednotlivých firmwaru je třeba si udělat jejich vzájemné porovnání a vybrat firmware pro paralelního robota. Srovnání se pokusím

provést co nejvíce objektivně, jednotlivé firmware umístím do tabulky, kde vypíšu jejich parametry.

Tabulka 8: Porovnání firmwarů

Vlastnosti	Teacup	Aprinter	Repetier
Přehlednost konfiguratoru	Skvělá	Podprůměrný	Průměrný
Složitost instalace	Malá	Malá	Malá
Podpora pro SD karty	Ano	Ano	Ano
Podpora pro displej	Ne	Ne	Ano
Podpora 3D tisku	Ano	Ano	Ano
Podpora pro laser	Ne	Ano (experimentální)	Ano
Podpora pro frézku	Ne	Ne	Ano
Zvládl testovací G-code	Ne	Ne	Ano

Asi nemá smysl tuto tabulku nikterak více rozšiřovat a vítěz pro naše zařízení je jasný a stává se jím Repetier. Pokud bychom chtěli pouze jednoduchou 3D tiskárnu, určitě by jsme doporučili Teacup.

6 Test funkčnosti zařízení

Na test funkčnosti paralelního robota jsme zvolili 2,5 W laser, určený ke gravírování do dřeva a řezání například papíru nebo fólií. Laser jsme připojili na port D9 na naší vývojové desce RAMPS. Zde však musím říci důležité varování pro všechny, kteří by též pracovali s laserem. Jako první materiál jsme zvolili smrkové dřevo. Test proběhl a paralelní robot vypálil obrazec přesně podle zadání, bez sebemenších problémů.

Podruhé jsme zkusili otestovat řezání plastových fólií. S tím nebyl žádný problém a laser si s řezem poradil velice dobře, jen je třeba si uvědomit, že při tomto vzniká poměrně silný zápach, proto, jako další věc kterou jsme přidali do seznamu co je třeba na tohoto paralelního robota dodělat, je odsávání.

Dále jsme testovali frézku a smrkové dřevo. Tento robot s tím neměl žádný problém, jen je třeba zvolit správnou šíři nástroje a opět se ukázalo jako nutnost mít odsávání okolo pracovní plochy nástroje, jelikož piliny létaly všude a zašpinily vodící tyče, což by mohlo zkracovat životnost ložisek a jejich odpor na vodících tyčích.

7 Závěr

Tato práce si kladla za cíl osvětlit neznalému čtenáři základy praktického programování a základní znalosti, potřebné ke konstrukci paralelního robota. Tento účel tato práce splnila a jako důkaz splnění, jsem postavil paralelního robota typu CoreXY. Tento robot je plně funkční a videa s demonstrací její funkčnosti, stejně tak jako fotodokumentaci a veškeré modely, naleznete v digitální podobě jako přílohu této práce. Časem budou uvolněny na reppap fórum, volně k dispozici široké komunitě a veřejnosti.

I když tato práce byla z části o konstrukci paralelního robota CoreXY, neopomenula i na další velmi populární typ a to je robot typu Delta. A i když tato práce neslouží jako návod krok za krokem, pozornému čtenáři poskytla dostatek informací, aby se mohl pustit do konstrukce vlastní 3D tiskárny, laserové gravírky či jiného robota, bez ohledu na to, zda-li se bude jednat o typ delta, CoreXY nebo typ úplně jiný.

A úplným závěrem bych chtěl poděkovat všem, kteří se na této práci a realizaci podíleli, zejména vedoucímu mé práce, panu Ing. Michal Šerému, že mi toto téma umožnil realizovat. Dále Vojtěchu Urbanovi, za jeho podnětné rady a pomoc při skládání tohoto paralelního robota. A samozřejmě své rodině, která mě v této práci podporovala a komunitě na CNC fóru, jenž je nepřebornou studnicí nápadu.

Použitá literatura a zdroje

- [1] *Ohmův zákon*. In: *Wikipedia: the free encyclopedia [online]*. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-02-21].
Dostupné z https://cs.wikipedia.org/wiki/Ohmův_zákon
- [2] PAVLICA, Jiří. *Paralelní kinematické struktury průmyslových robotů [online]*. Brno: Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2012 [cit. 2016-11-13].
Dostupné z: <http://hdl.handle.net/11012/5643>
- [3] BONEV, Ilian. *Delta Parallel Robot — the Story of Success [online]*. [cit. 2016-11-14]..
Dostupné z: <http://www.parallemic.org/Reviews/Review002.html>
- [4] REYMOND, CLEVEL. *Device for the movement and positioning of an element in space. US4976582 (A)*. Uděleno 1990-12-11.
- [5] *Massachusetts Institute of Technology [online]*. Massachusetts Institute of Technology: Ilan E. Moyer, 2011 [cit. 2016-11-14].
Dostupné z: <http://web.mit.edu/imoyer/www/portfolio/botofthecloth/index.html>
- [6] *Reference Mechanism*. In: *CoreXY.com [online]*. Ilan E. Moyer, 2012 [cit. 2016-11-15].
Dostupné z: <http://corexy.com/theory.html>
- [7] *RepRap Wiki [online]*. RepRap.org, 2016-8-16 [cit. 2016-11-15].
Dostupné z: http://reprap.org/wiki/NEMA_17_Stepper_motor
- [8] *Think3dPrint3d [online]*. United Kingdom: Think3DPrint3D Ltd, 2013 [cit. 2016-11-15].
Dostupné z: <http://blog.think3dprint3d.com/2013/12/Duet-Arduino-Due-compatible-3DPrinter-controller.html>
- [9] *GitHub [online]*. San Francisco: Ambroz Bizjak, 2015 [cit. 2016-11-16].
Dostupné z: <https://github.com/ambrop72/aprinter>
- [10] *Reprap*. Reprap [online]. Tennessee, USA: Johnny Russell, 2016 [cit. 2016-12-08].
Dostupné z: http://reprap.org/wiki/arduino_Mega_Pololu_Shield#Schematic

Seznam obrázků

1	Schématická značka rezistoru	8
2	Schématická značka potenciometru	8
3	Schématická značka diody	9
4	Schématická značka LED	9
5	Schématické značka různých zdrojů	10
6	Klon Arduino UNO	11
7	Arduino Pro Mini	12
8	Arduino Nano	12
9	Arduino Micro	13
10	Arduino UNO	13
11	Arduino MEGA 2560	13
12	Arduino Yún	14
13	LilyPad Arduino USB	14
14	Arduino Ethernet Shield	15
15	Arduino zapojení LED diody	16
16	Arduino zapojení čidla DHT11	17
17	Arduino řízení teploty	18
18	Pulzně šířková modulace	18
19	Arduino řízení RGB diody	19
20	Arduino a LCD displej přes I2C	22
21	Arduino a krokový motor	24
22	Arduino a sedmissegmentový displej	26
23	Koncepce delta robota	31
24	Referenční mechanismus CoreXY	32
25	Upravený mechanismus CoreXY	33
26	Osa Z	34
27	Elektronika RAMPS	35
28	Schéma elektroniky RAMPS	38
29	Hrubá konstrukce	39
30	Konstrukce před realizací	40
31	Přizvednutý motor	41
32	Prostředí Pronterface	43
33	Teacup konfigurační nástroj	45
34	Aprinter konfigurační nástroj	47
35	Repetier konfigurační nástroj	49

Seznam tabulek

1	Spínání plné kroky	24
2	Spínání poloviční kroky	25
3	Porovnání sériových a paralelních kinematických struktur . . .	30
4	Porovnání Delta a CoreXY	33
5	Vlastnosti RAMPS	36
6	Vlastnosti Smothieboard	36
7	Vlastnosti Duet	37
8	Porovnání firmwarů	50

Výpisy kódu

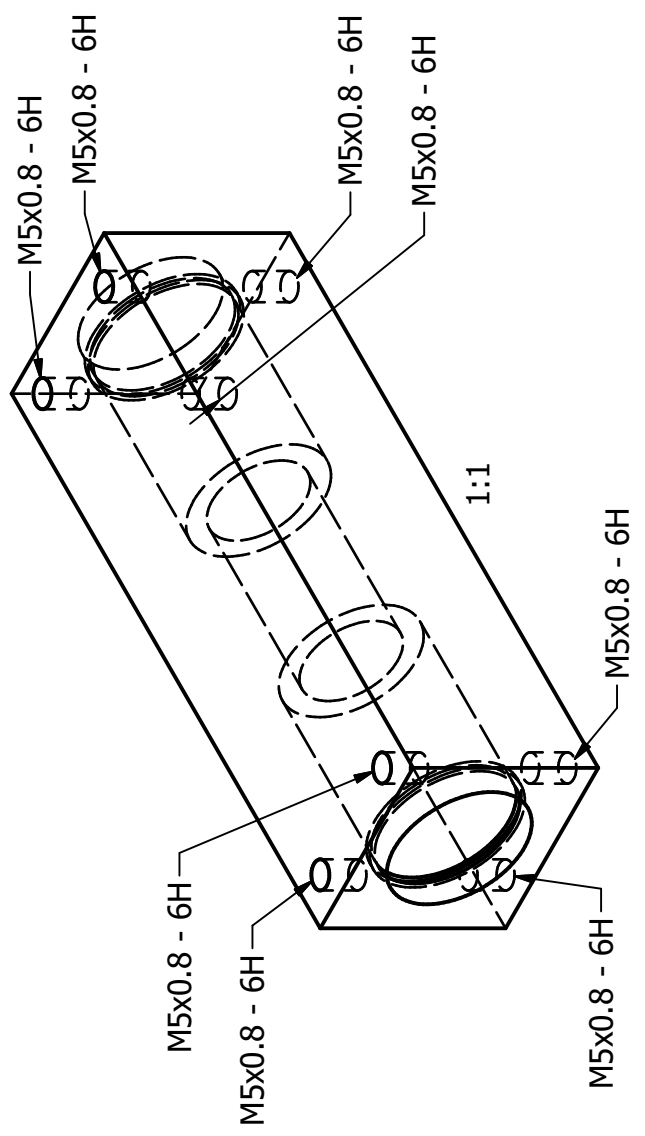
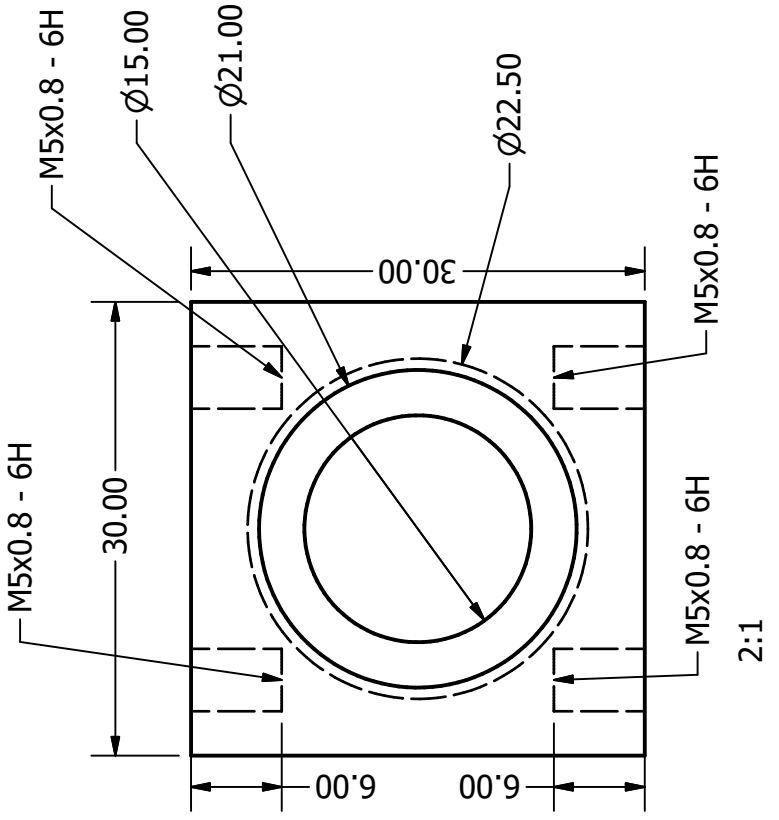
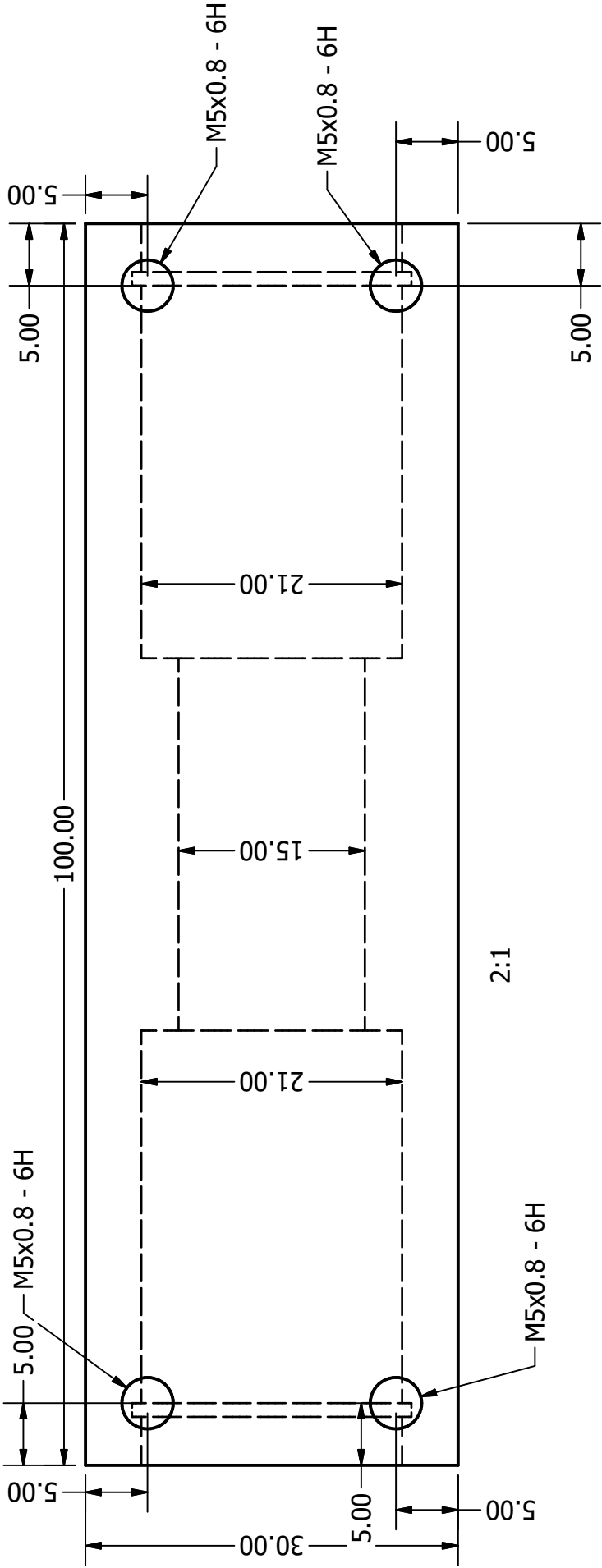
1	Rozsvěcení LED diody	15
2	Teplotní čidlo	16
3	Řízení teploty	17
4	Řízení RGB diody	20
5	Komunikace s pc	21
6	LCD displej přes I2C	23
7	Arduino krokový motor plné kroky	25
8	Arduino krokový motor poloviční kroky	26
9	Arduino krokový motor poloviční kroky	27
10	Instalace základních balíčků pro komunikaci s Arduinem	43
11	Instalace Teacup	45
12	Instalace Aprinter	47

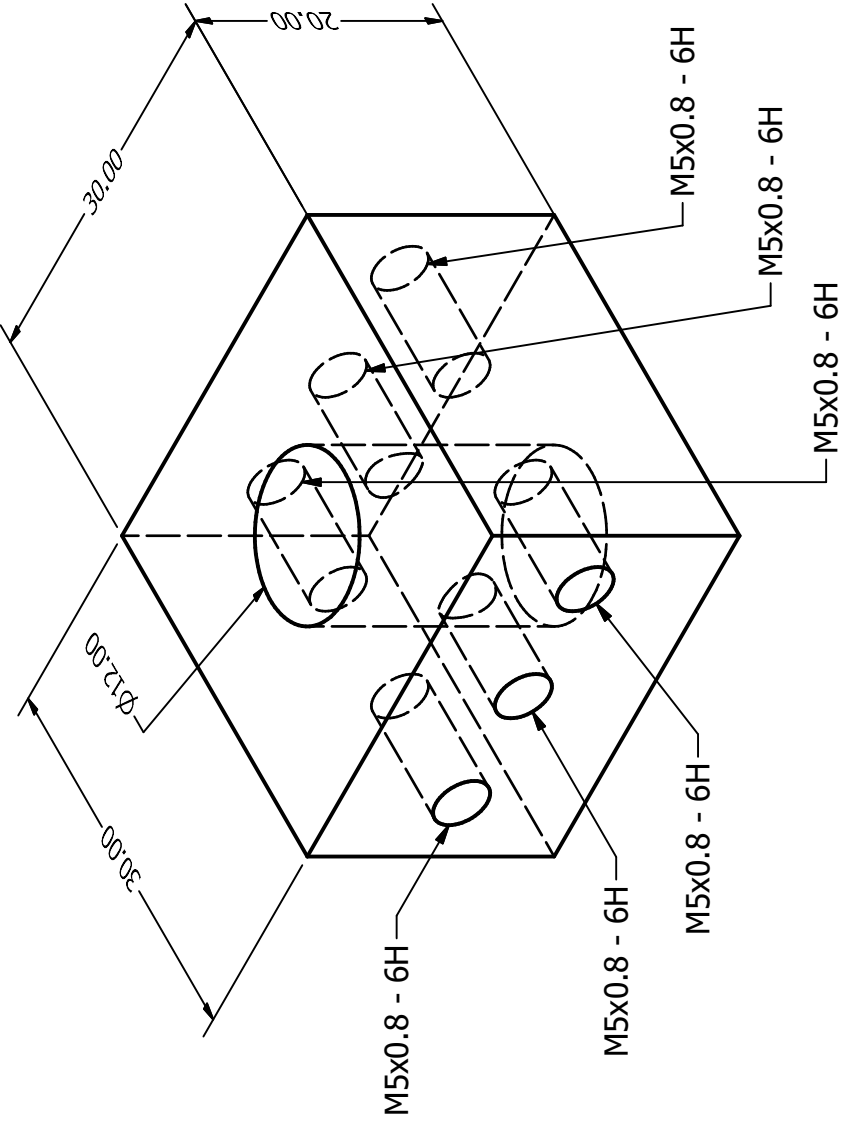
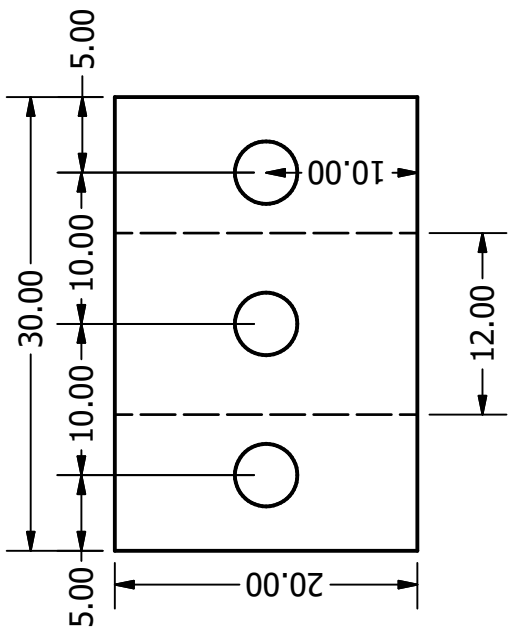
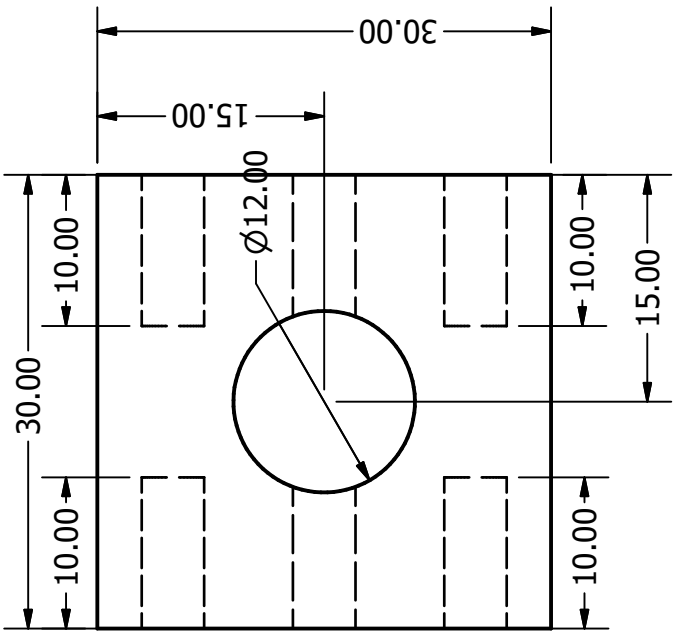
Přílohy

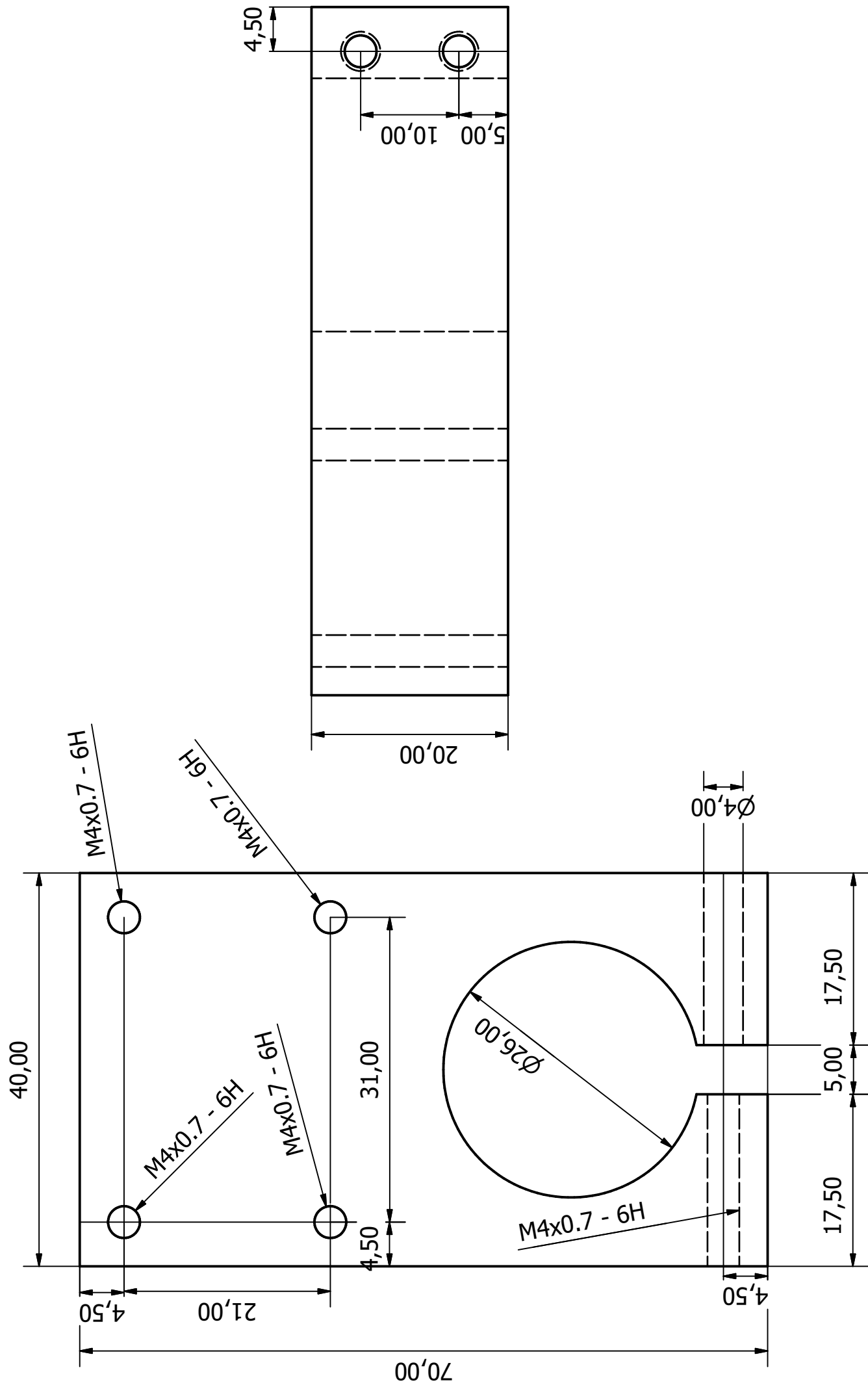
1. CD - na tom dvd najdete výkresy k párnímu robotu v digitální podobě, modely jednotlivých komponent i celého robota a veškerou zbylou dokumentaci
2. Výkresová dokumentace

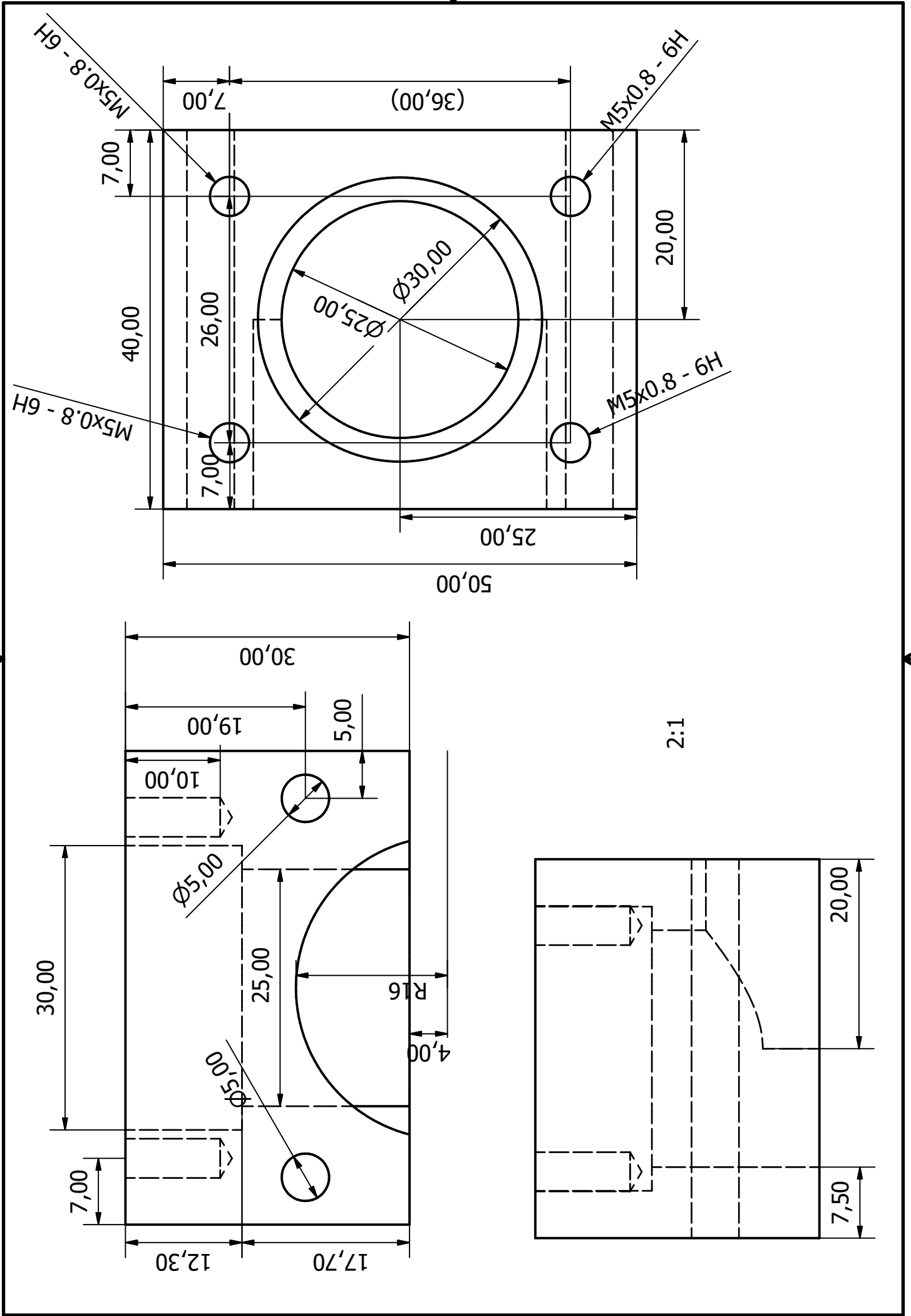
Příloha číslo 1 CD

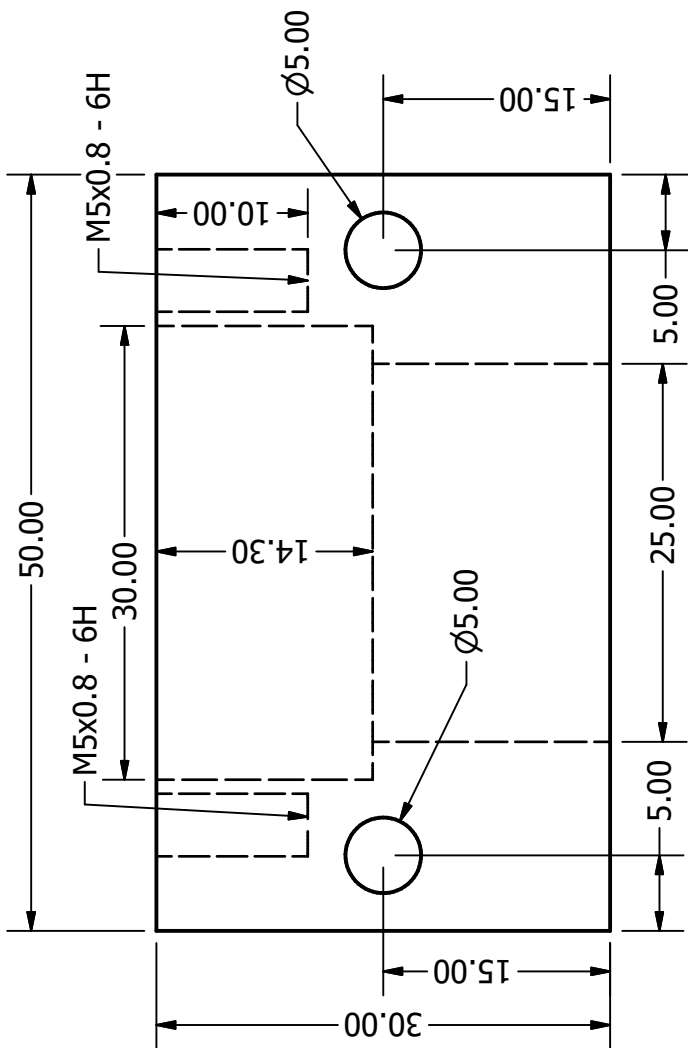
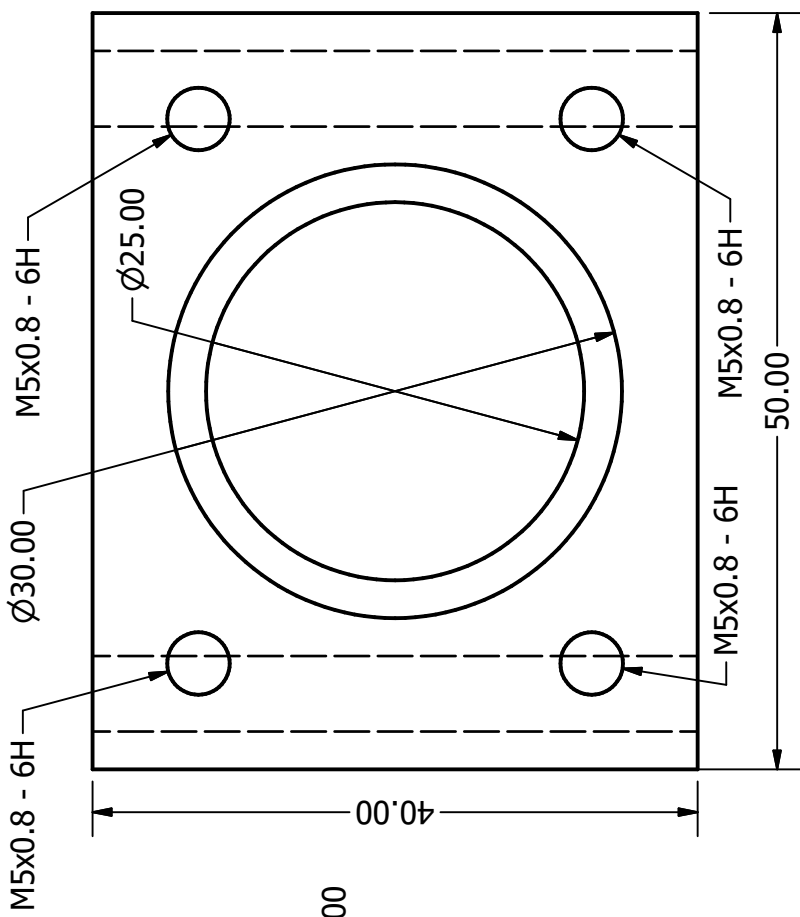
Příloha číslo 2 Výkresy

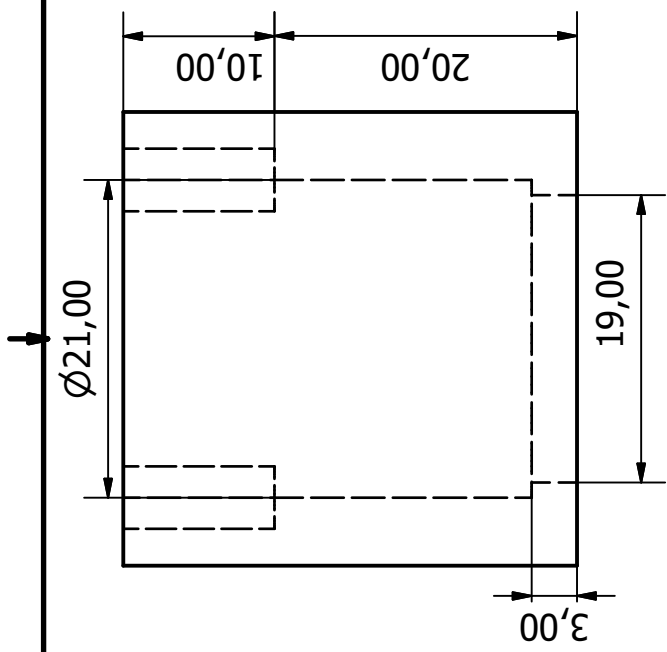




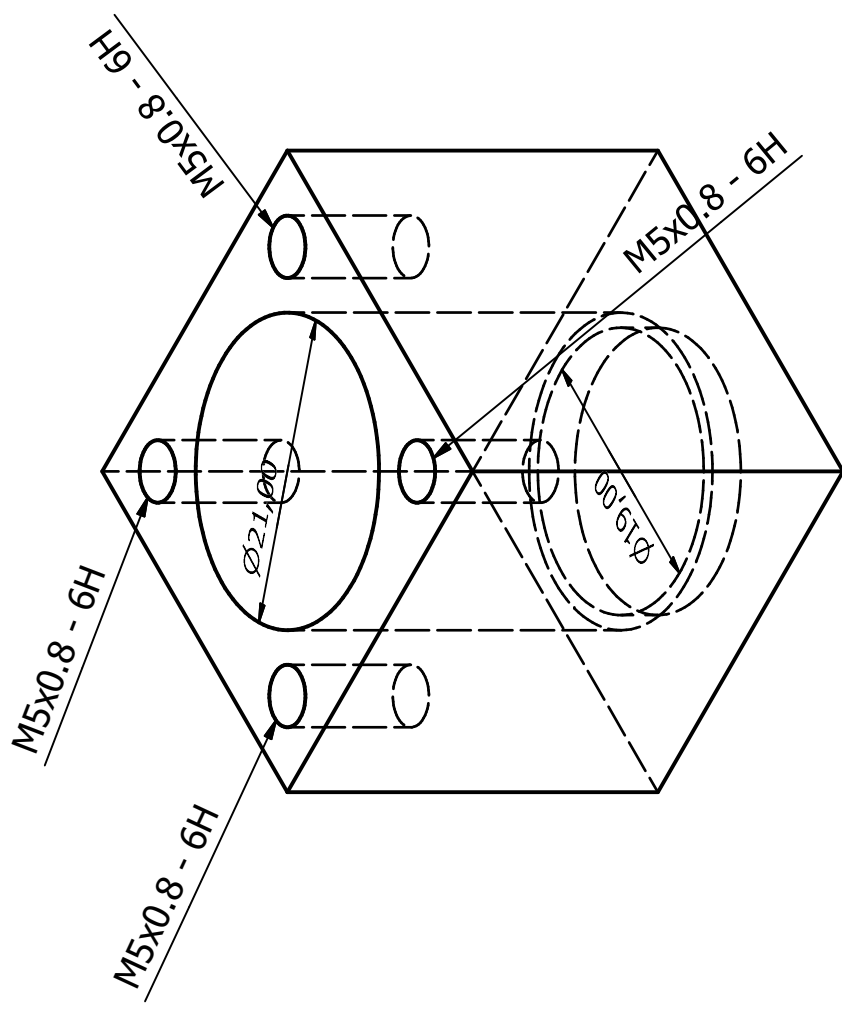
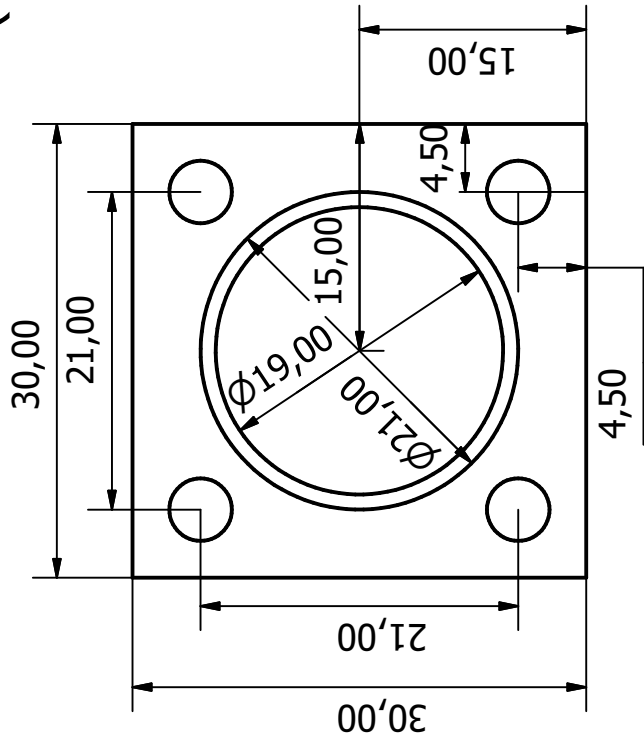


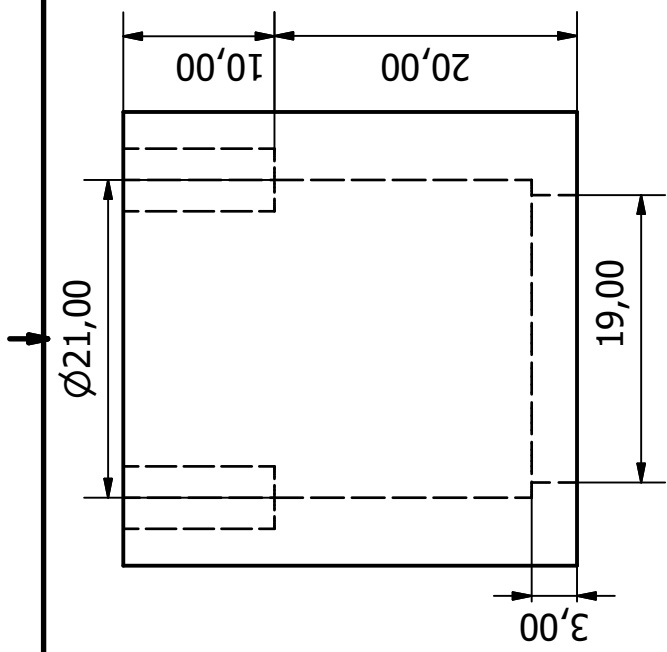




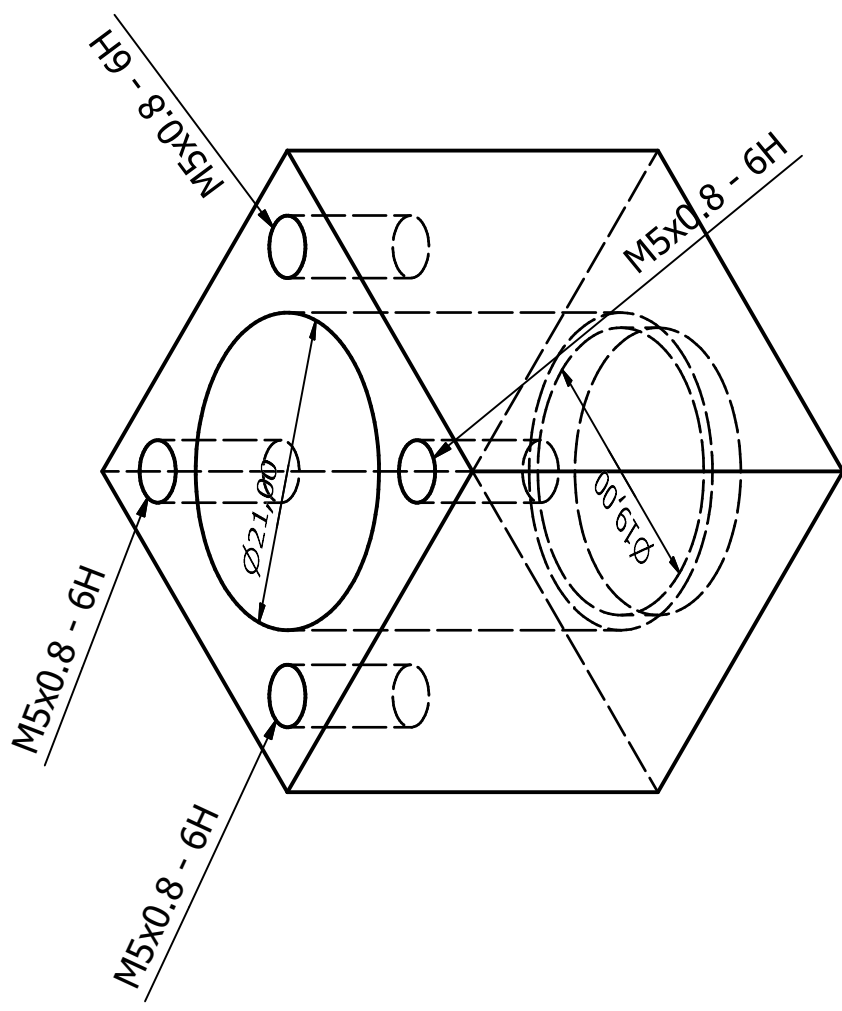
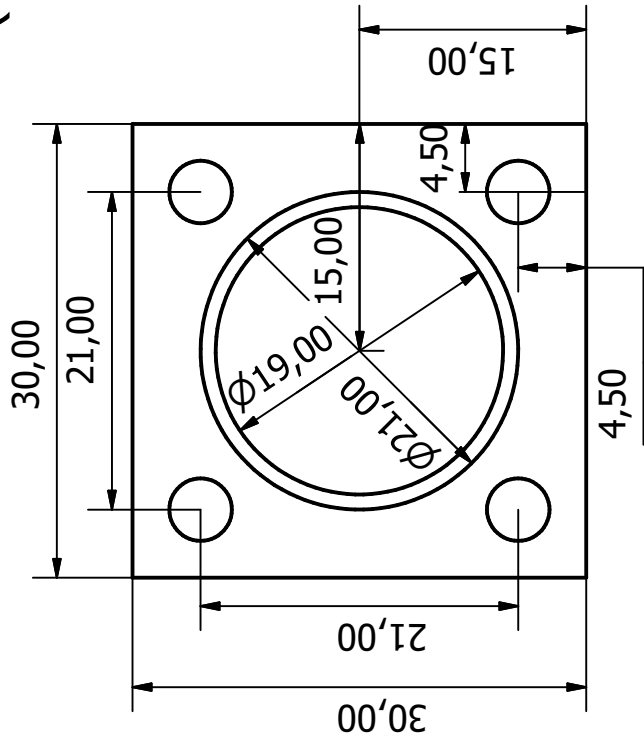


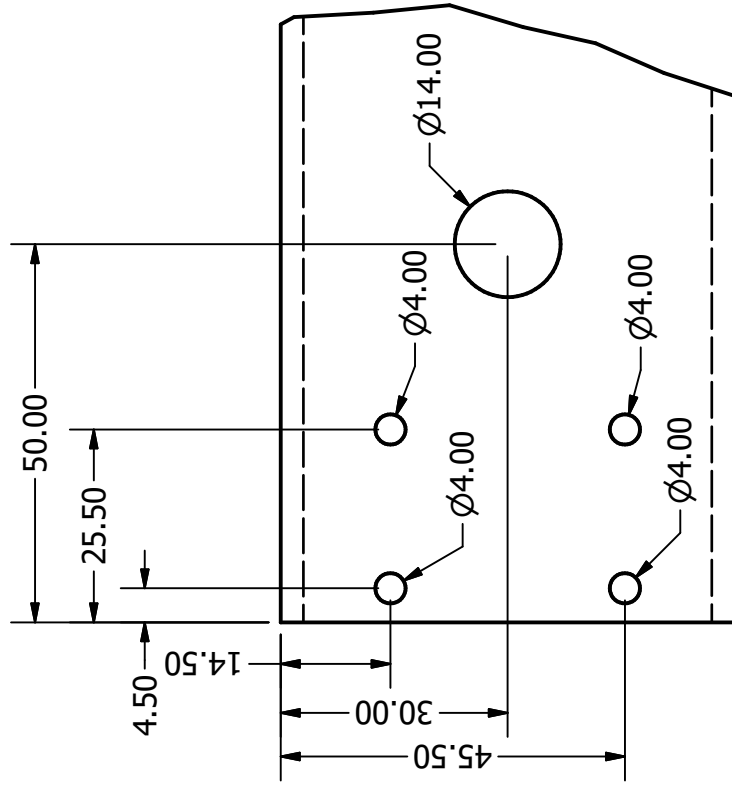
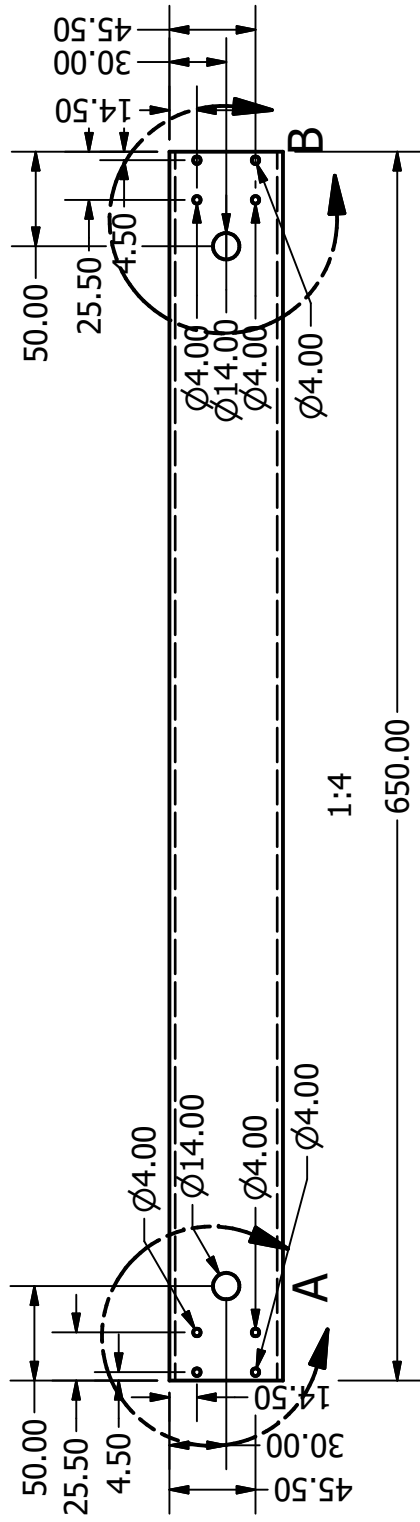
A (2:1)



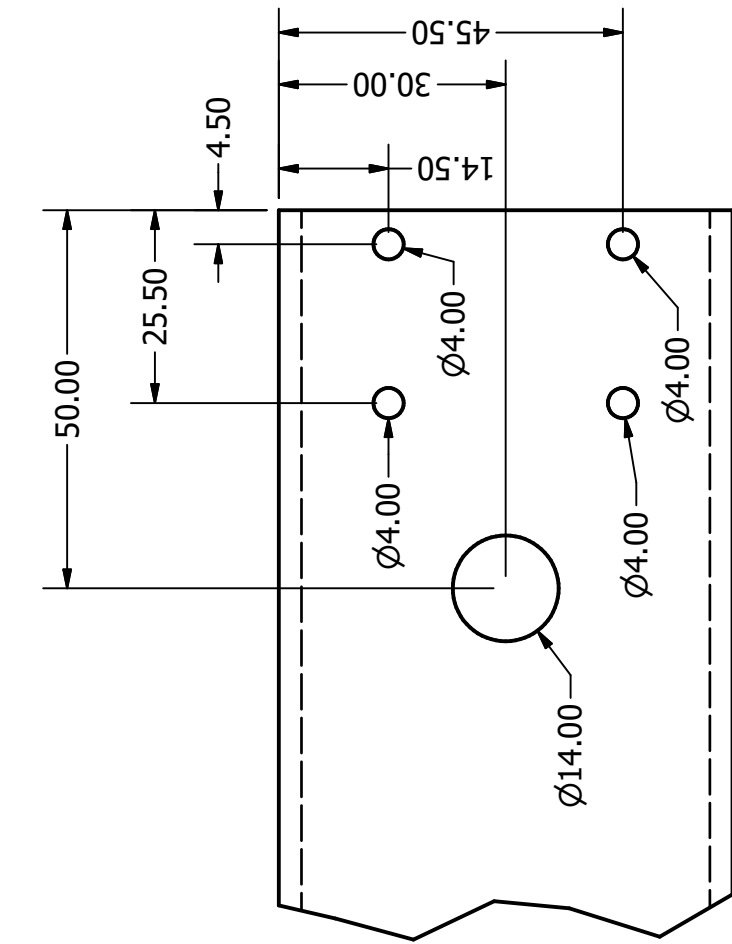


A (2:1)

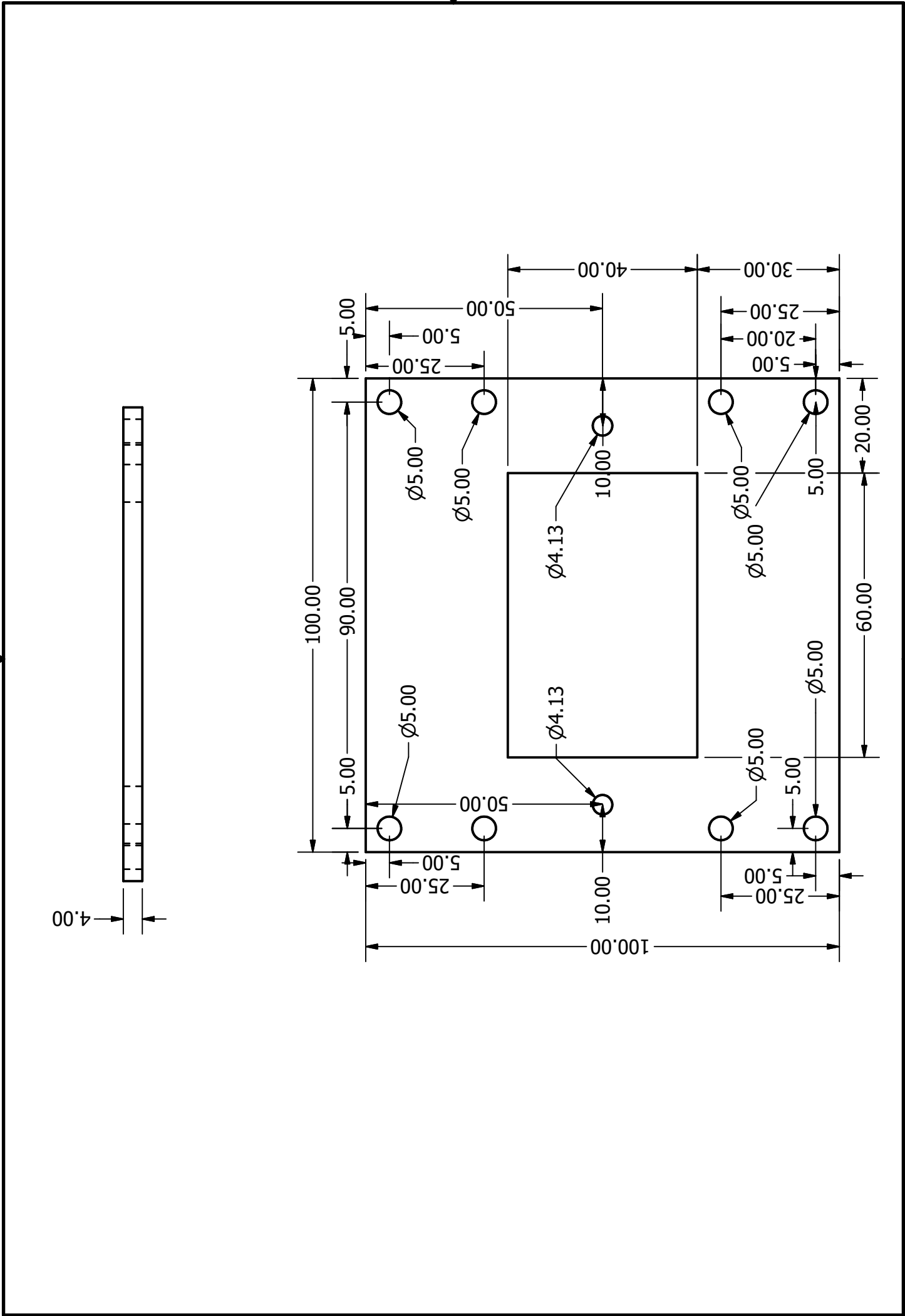


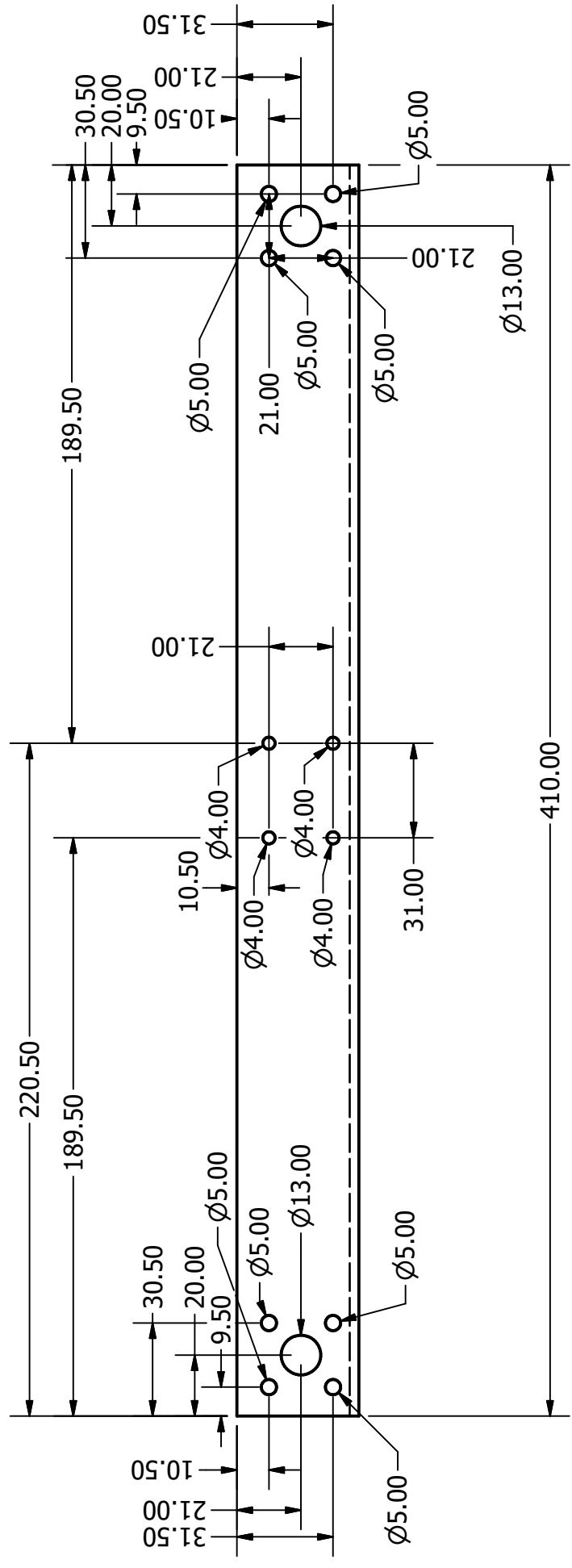


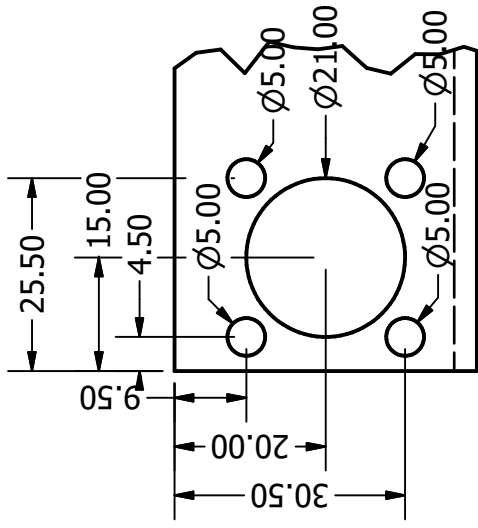
DETAIL A
 MĚŘITKO 1 : 1



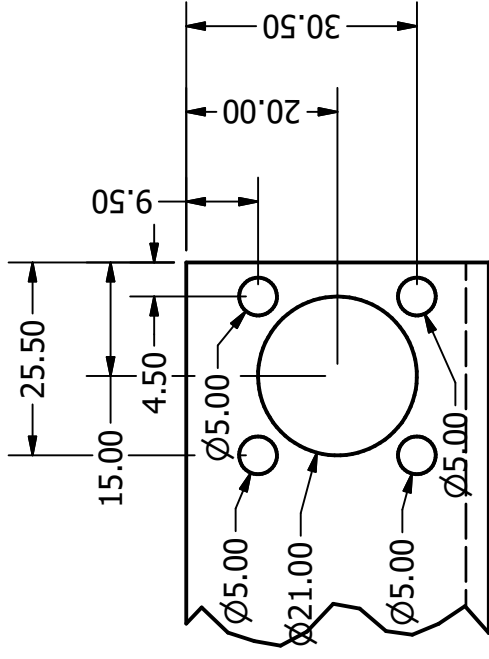
DETAIL B
 MĚŘITKO 1 : 1







DETAIL A
MĚŘÍTKO 1 : 1



DETAIL B
MĚŘÍTKO 1 : 1

