

Implementace 3D tisku prostřednictvím průmyslového robotu Stäubli RS40B

The Implementation of 3D Printing Using a Stäubli RS40B
Industrial Robot

Bc. Adam Škamrala



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2014/2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Adam Škamrala**
Osobní číslo: **A13751**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Automatické řízení a informatika**
Forma studia: **prezenční**

Téma práce: **Implementace 3D tisku prostřednictvím průmyslového robotu
Stäubli RS40B**

Téma anglicky: **The Implementation of 3D Printing Using a Stäubli RS40B
Industrial Robot**

Zásady pro vypracování:

1. Provedte literární rešerši na zadané téma.
2. Vyberte vhodnou tiskovou hlavu a doplňující hardware za účelem propojení s řídicí jednotkou robotu.
3. Zvolte vhodné programové vybavení pro vytváření 3D modelů a export řezů do G-kódu.
4. Vytvořte programové vybavení ve vybraném programovém prostředí pro interpretaci G-kódu do jazyku robotu za účelem řízení pohybu po zadané dráze a ovládání tiskové hlavy.
5. Demonstrujte na vhodně zvoleném příkladu možnosti 3D tisku prostřednictvím průmyslového robotu.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. CS8C Controller: Instruction manual. Stäubli Faverges, 2008, 190 s.
2. VAL3 REFERENCE MANUAL: Version 6. Stäubli Faverges, 2008, 186 s.
3. BANZI, Massimo. Getting started with Arduino. 1st ed. Make: Books / O'Reilly, 2009. ISBN 978-059-6155-513.
4. EVANS, Brian. Beginning Arduino Programming. New York: Apress, 2011, 272 s. ISBN 978-1-4302-3778-5.
5. HORVATH, Joan. Mastering 3D Printing. New York: Apress, 2014, 224 s. ISBN 978-1-4842-0025-4.
6. CANESSA, Enrique, Carlo FONDA a Marco ZENNARO. Low-cost 3D Printing for Science, Education & Sustainable Development. First Edition. Trieste: ICTP The Abdus Salam International Centre for Theoretical Physics, 2013. ISBN 92-95003-48-9. Dostupné z: <http://sdu.ictp.it/3D/book.html>
7. MAGAZINE, Make. MAKE: ultimate guide to 3D printing. MAKE special issue. Sebastopol, CALIF: O'Reilly Media, 2012. ISBN 978-144-9357-375.
8. ROTOLO, By Isaac Budmen and Anthony. The book on 3D printing. 2013. ISBN 978-1489529442.

Vedoucí diplomové práce:

Ing. Petr Navrátil, Ph.D.

Ústav řízení procesů

Datum zadání diplomové práce:

27. února 2015

Termín odevzdání diplomové práce:

20. května 2015

Ve Zlíně dne 27. února 2015

doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

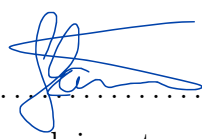
Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomové práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 20.5.2015

.....

podpis autora

ABSTRAKT

Účelem práce bylo navržení funkce 3D tisku na průmyslovém robotu. Teoretická část podává základní informace o problematice a vývoji 3D tisku s následným zaměřením na metodu FDM, které bylo využito. Dále rozebírá technické a programové specifikace použitého mikrokontroléru a 4-osého robota. V praktické části je celé vlastní řešení daného tématu. Byla sestavena tisková hlava, kterou bude prováděn tisk. Dále byla navržena a realizována elektronická část k řízení tiskové hlavy. Nakonec byla provedena implementace programu mikrokontroléru, kterým je dosaženo programové řízení všech součástí tiskové hlavy a vyhřívané podložky.

Klíčová slova: 3D tisk, tisková hlava, PWM řízení, PID řízení, mikrokontrolér Arduino Uno, 4-osý robot Stäubli RS40B, sériová komunikace RS232

ABSTRACT

The goal of this thesis was the design of 3D printer using an industrial robot. Theoretical part includes basic information about the problem and development of 3D printing with the focus on FDM method, which is used later on. Furthermore it analyzes technical and program specifications of used microcontroller and 4-axis robot. Practical part deals with the actual implementation of the topic. Printer head was constructed, which will be used for the printing. Next there is the design and development of electronic parts for the printer head control. Finally, software for the microcontroller was developed, which is for the control of printer head parts and heated bed.

Keywords: 3D printing, printing head, PWM control, PID control, Arduino Uno microcontroller, 4-axis robot Stäubli RS40B, serial communication RS232

Na tomto místě bych chtěl poděkovat vedoucímu mé závěrečné diplomové práce panu Ing. Petru Navrátilovi, Ph.D. za jeho cenné rady, podporu a poskytnutí materiálů a zdrojů informací a za celkové vedení při práci.

Děkuji také svým nejbližším za podporu po celou dobu mého studia.

„Pokud jsi pracovitý a snaživý, jsi takový kvůli sobě, ne kvůli ostatním lidem. Vítězit je návyk. Vzdávat naděje na vítězství je bohužel také návyk.“

- Vince Lombardi

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	9
1 3D TISK	11
1.1 TECHNOLOGIE 3D TISKU	11
1.1.1 Fused Deposition Modeling (FDM)	11
1.1.2 Selective Laser Sintering (SLS)	12
1.1.3 Stereolitografie (SLA)	13
1.2 SOUČÁSTI TISKOVÉ HLAVY	14
1.2.1 Extruder	14
1.2.2 Hrot	15
1.3 VYHŘÍVANÁ PODLOŽKA	16
1.4 MATERIÁL PRO 3D TISK	16
2 SOFTWARE	18
2.1 PRO TVORBU 3D MODELŮ	18
2.2 PRO TVORBU ŘEZŮ A GENEROVÁNÍ G-KÓDU	20
3 MIKROKONTROLÉR ARDUINO UNO	22
3.1 POPIS MIKROKONTROLÉRU	22
3.2 PROGRAMOVÉ VYBAVENÍ	23
3.2.1 Datové typy	24
3.3 VÝVOJOVÉ PROSTŘEDÍ	24
4 PRŮMYSLOVÝ ROBOT STÄUBLI RS40B	26
4.1 ROBOTICKÉ RAMENO RS40B	26
4.1.1 Rozsah pohybů, rychlost a hmotnostní zatížení	26
4.2 KONTROLÉR CS8C	28
4.2.1 Ruční programovací panel MCP	28
4.2.2 Komunikační rozhraní	29
4.3 PROGRAMOVACÍ JAZYK VAL3	30
4.3.1 Datové typy	30
4.3.2 Vývojové rozhraní	30
II PRAKTICKÁ ČÁST	31
5 HARDWARE	33
5.1 ARDUINO UNO	33
5.2 TISKOVÁ HLAVA	33

5.3	VYHŘÍVANÁ PODLOŽKA.....	35
5.4	NAPÁJECÍ ZDROJ	35
5.5	NÁVRH DESKY	36
5.5.1	Schéma zapojení.....	36
5.5.2	Přehled součástek	37
5.5.3	Ovladač krokového motoru	38
5.5.4	Převodník RS232.....	39
5.5.5	Výsledná realizace	40
6	CELKOVÉ PROPOJENÍ JEDNOTLIVÝCH PRVKŮ SYSTÉMU	43
7	NÁVRH REGULÁTORŮ	44
7.1	IDENTIFIKACE	44
7.1.1	Hrot.....	45
7.1.2	Vyhřívaná deska	46
7.2	VÝPOČET PARAMETRŮ.....	47
7.2.1	Regulační pochody	48
8	PROGRAMOVÁNÍ ARDUINO UNO	51
8.1	ŘÍZENÍ.....	51
8.2	KOMUNIKAČNÍ PROTOKOL.....	52
	ZÁVĚR.....	53
	SEZNAM POUŽITÉ LITERATURY	55
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	58
	SEZNAM OBRÁZKŮ	59
	SEZNAM TABULEK	61
	SEZNAM PŘÍLOH	62

ÚVOD

Když se řekne průmyslový robot, každý si představí mechanické ústrojí zpracovávající určitou činnost dle zadaných pokynů. Prvotní myšlenka pro tuto práci bylo využití vlastností průmyslového robota k činnosti, která pro něj není příliš typická. Není prováděna žádná manipulace s objekty nebo sváření, ale ideou bylo jej využít pro přesné řízení polohy koncového členu, ke kterému by byla instalována tisková hlava pro 3D tisk.

3D tisk není žádnou novinkou a je s námi už dobrých pár let. Ale teprve v posledních letech se dostává do podvědomí širší veřejnosti. Jedním z důvodů je pokles pořizovacích nákladů. Potenciální uživatelé mohou pořízení řešit nákupem jednotlivých dílů a vlastní tiskárnu si vlastnoručně složit. Nebo lze zakoupit hotové sety, kde ale cena pochopitelně vzroste, ale uživatel tento typ tiskáren pouze zapne a může tisknout. První možnost je takové kutilství, protože vyžaduje zásahy do mechanických částí, elektroniky a programování nebo využití existujících firmwarů. Ty ale nikdy nezajistí bezproblémový chod a nezbytná kalibrace 3D tiskárny je běh na dlouhou trať. Nicméně u tohoto typu tiskáren může uživatel improvizovat. Jakmile je tiskárna uvedena do provozu a je schopna tisknout akceptovatelné výstupy, je následně možné tisknout součástky pro sebe samu. Obecně to mezi uživateli funguje tak, že někdo vymyslí např. promyšlenější součástku pro držení tiskové hlavy a po nahrání odpovídajícího 3D modelu je schopen distribuovat své řešení napříč všemi uživateli po světě. 3D tiskárny jsou v současné době nejrychlejším možným způsobem, jak mezi lidmi šířit všemožné prototypy určené k tisku.

Ale zpět k věci. Plánovaného řešení se týká pouze tisková hlava 3D tiskárny a zbytek realizace včetně navázání komunikace s robotem bylo nutné navrhnout takříkajíc od píky. Celá diplomová práce se dělí na teoretickou a praktickou část. Teoretická část se zabývá seznámením čtenáře se stručnou historií 3D tisku jako takového včetně metod, které lze považovat za klíčové v tom smyslu, že měly podstatný vliv na současnou podobu. Dále jsou uvedeny podklady jednotlivých prostředků potřebných pro výsledné praktické řešení. Praktická část je potom chronologicky řazený postup práce.

I. TEORETICKÁ ČÁST

1 3D TISK

Charles W. Hull je obecně označován jako tvůrce první 3D tiskárny z roku 1984 [3]. Následně si nechal v roce 1986 patentovat technologii stereolitografie. Princip je takový, že se vytváří objekty postupným vytvrzováním polymerů pomocí UV záření při různých vlnových délkách. Později v roce 1989 byla tato tiskárna komercializována společností 3D Systems založenou právě Charlesem W. Hullem. [1]

V posledních letech zažívá 3D tisk značné rozšíření, kde jedním z důvodů je vypršení patentů [3]. Dále rozmachu domácích 3D tiskáren pomáhá značné množství otevřených komunit zabývajících se touto problematikou. 3D tisk neboli aditivní výroba je proces, při kterém je vytvářen trojrozměrný model objektu na základě digitální předlohy modelu. V aditivním procesu je objekt vytvářen postupným nanášením vrstev materiálu, dokud není vytvořen celý objekt. Každá z těchto vrstev může být viděna jako tenké plátky ve vodorovném průřezu vytvořeného objektu. [1] [2]

Zpočátku se 3D tisk primárně označoval jako Rapid Prototyping (rychlá výroba prototypů) a dodnes se s tímto označením lze setkávat. Před příchodem dostupných tiskáren se technologie 3D tisku používala výhradně pro výrobu prototypů. Rapid Prototyping obsahuje tři procesy, a to pre-processing, processing a post-processing. [4] [6]

Výrobní postup začíná návrhem 3D modelu v CAD (Computer Aided Design) softwaru nebo pokud chceme nějaký existující objekt replikovat, tak lze použít 3D skener [2]. V obou případech je nutné, aby výstupem byl model ve standardním formátu STL, který je vstupem pro CAM (Computer Aided Manufacturing) software. Následně je v pre-processingovém softwaru model rozdělen na tenké vrstvy (řezy) a jsou dopočítány dráhy pohybu pro tiskové hlavy v 3D tiskárně. Processing potom zahrnuje samotný tisk trojrozměrného modelu a dodatečné úpravy (čištění, lakování, lepení) patří do post-processingu. [4]

1.1 Technologie 3D tisku

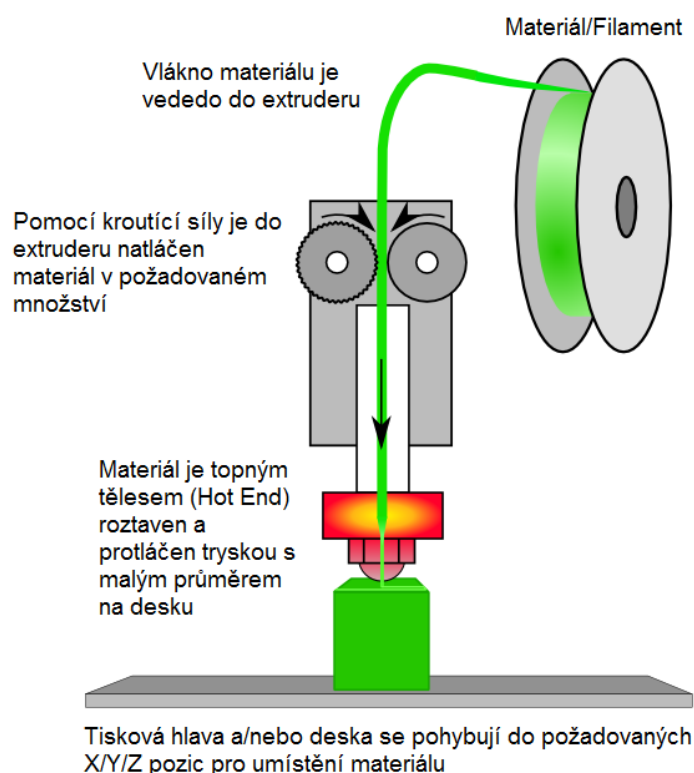
V této podkapitole je popsán pouze zlomek dostupných technologií pro 3D tisk. Pozornost je kladena na použitou technologii v praktické části a další uvedené technologie lze považovat za hlavní milníky technologického vývoje.

1.1.1 Fused Deposition Modeling (FDM)

Vzhledem k existenci mnoha různých metod 3D tisku, je v současnosti technologie FDM (taktéž označována jako FFF) ta nejobvyklejší a zároveň nejrozšířenější. Princip technologie je velice jednoduchý. Do tiskové hlavy je pod tlakem natlačováno obvykle plastové vlákno, přesněji řečeno, některý z tzv. termoplastů. Tisková hlava daný ma-

terál roztaví a vytlačuje jej ven tryskou. Metoda tedy využívá vlastnosti stavebního materiálu, za určitých teplot vratně měnit svoje skupenství z pevného na viskozně-tekutý stav [8]. Celá tisková hlava se pohybuje nad pracovní plochou určenou pro tisk a z vytlačovaného materiálu se po jednotlivých vrstvách tvoří výsledný objekt. Pohyb je obvykle zajištěn krokovými motory nebo servopohony včetně zajištění pohybu pro vytlačování materiálu. Tato technologie byla vynalezena a patentována S. Scottem Crumpem v roce 1989. Ke komercializaci došlo v roce 1990 společností Stratasys [7], které byl Crump spoluzakladatelem. [5] [6]

Nevýhodou je rozlišení, které nedovoluje dosáhnout vysokých detailů a FDM nezvládne plnobarevný tisk. Výhodou je variace různých materiálů, které jsou tyto tiskárny schopny používat. Nelze opomenout bezpečnost procesu tisku, proto není problém tyto tiskárny provozovat doma nebo v běžné kanceláři. [4]



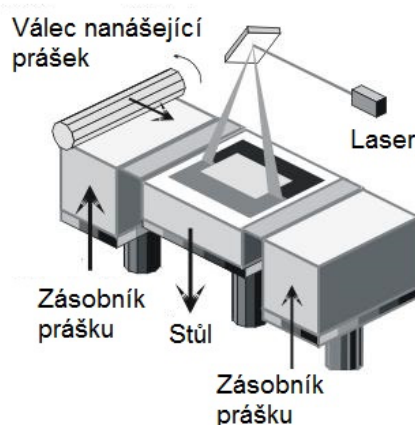
Obr. 1.1 Princip technologie FDM [10].

1.1.2 Selective Laser Sintering (SLS)

SLS je jednou z nejstarších, ale velmi efektivních metod výroby prototypů. Během procesu jsou malé částčky plastu, keramiky nebo skla spékány dohromady působením

výkonného laseru do trojrozměrného tvaru. Tento proces byl vyvinut a patentován Carlem Deckardem v roce 1980. [5] [11]

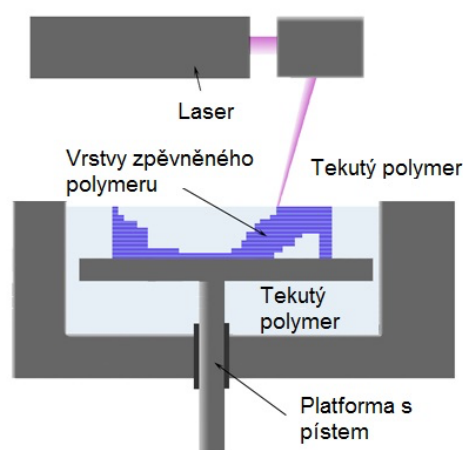
V porovnání s ostatními metodami, SLS používá širokou škálu materiálů pro tvorbu objektů, ale je omezen tiskem pouze z jedné barvy. Jako další nevýhodu lze uvést vysoké pořizovací náklady. [9]



Obr. 1.2 Princip technologie SLS [12].

1.1.3 Stereolitografie (SLA)

Stereolitografie je dnes také velmi rozšířenou technologií pro tvorbu trojrozměrných objektů. Její princip je analogický s technologií SLS. Pro tvorbu jednotlivých vrstev objektu se využívá ultrafialového laserového paprsku, který vrstvu vykresluje na hladinu polymerové tekutiny, vytvrzované UV světlem. Po dokončení vrstvy se tištěný objekt ponoří do polymeru a začne tvorba další vrstvy. Tato technologie byla vynalezena a patentována v roce 1986 Charlesem W. Hullem. [5]



Obr. 1.3 Princip technologie SLA [5].

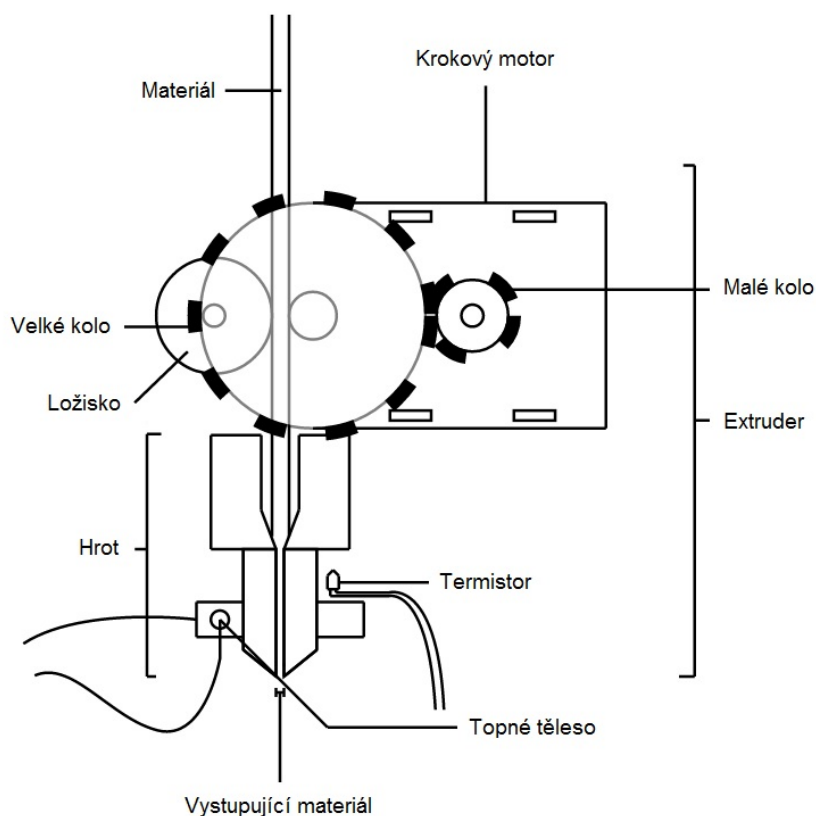
Zásadní nevýhodou této technologie je nemožnost výběru materiálu. Ačkoli UV laser potřebuje pro vytvrzování polymeru pouze zlomek výkonu v porovnání s technologií SLS, zůstává tato technologie nadále cenově nedostupná běžnému uživateli. Naopak výhodou této technologie jsou velmi detailní součástky. [4] [5]

1.2 Součásti tiskové hlavy

V této podkapitole jsou popsány nezbytné součásti tiskové hlavy 3D tiskárny využívající technologii FDM.

1.2.1 Extruder

Extruder je zařízení, které plní funkci dodávání materiálu do žhavicí trysky. Jeho součástí je pohon, kterým je obvykle krokový motor spolu s mechanismem, který tlačí materiál do hrotu. Mechanismem je soustava ozubených kol, kde na jednom konci je krokový motor a na druhém je šroub opírající se o ložisko, kterým se dávkuje materiál. [2] [15]



Obr. 1.4 Popis tiskové hlavy [15].

1.2.2 Hrot

Hrot neboli Hot-end [9] je část tiskové hlavy, kde dochází k tavení extruderem dodávaného materiálu. Skládá se z topného tělesa, teplotního senzoru pro měření teploty, tepelné bariéry pro oddělení studené a horké části a trysky pro nanášení roztaveného termoplastu.

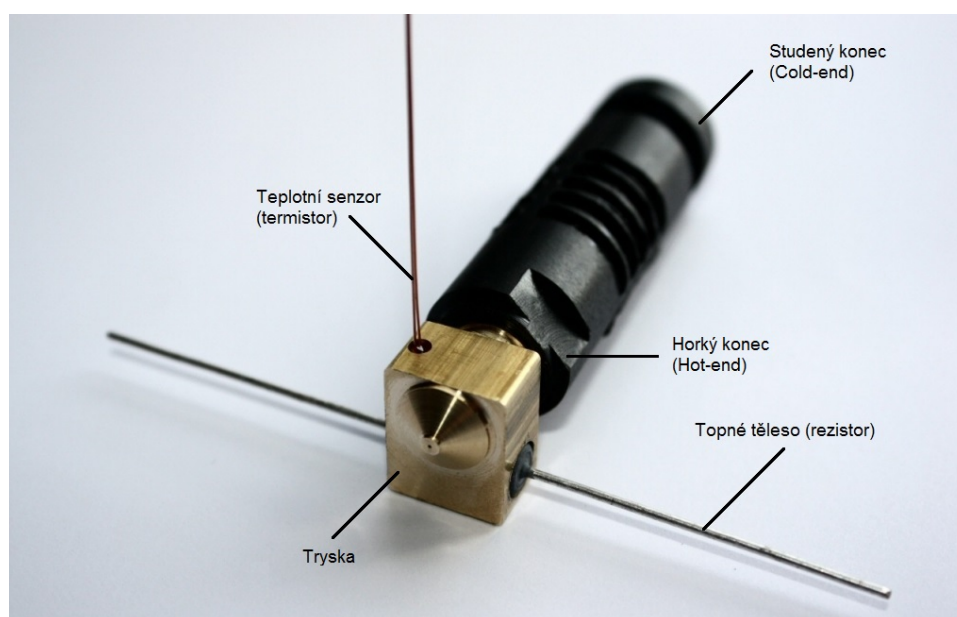
Jako topné těleso pro kovovou trysku se běžně používá rezistor s nízkým odporem. Drátový rezistor tak vytváří dostatečné množství tepla a výhodou je jeho nízký odpor, protože čím více proudu teče odporem, tím více tepla se vytváří. Topné tělesa jsou umístěna na vyhřívacím bloku trysky. [6]

Pro měření teploty se používá termistor, kterým lze po připojení k elektronice měřit a následně regulovat teplotu topného tělesa trysky. Pomocí rezistoru a měřením teploty lze tedy trysku vyhřát na požadovanou teplotu a udržovat ji.

Pod tepelnou bariérou [1] je charakterizována trubice z teflonu (PTFE) s vysokou tepelnou odolností, skrze kterou je přiváděn materiál od extruderu. Dělí se na horký a studený konec (Hot-end a Cold-end) [9]. Horký konec je spojen s tryskou, studený s extruderem. Tím je zabráněno nadměrnému zahřívání mechanismu tiskové hlavy.

Tryska má za účel roztavit materiál a spolehlivě ho nanášet na určená místa. Kvalitní tryska patří mezi nejdůležitější prvky tiskárny. Na trysku jsou kladeny specifické nároky ohledně vysoké teploty, tlaku, mechanického namáhání a přesnosti provedení. Konec trysky má otvor o průměru 0.5 až 0.2 mm [2] v závislosti na průměru použitého materiálu.

Na obrázku níže (Obr. 5.2) je popis jednotlivých součástí hrotu.



Obr. 1.5 Popis hrotu [16].

1.3 Vyhřívaná podložka

Podložka poskytuje pracovní plochu pro 3D tisk. Může být s ohřevem nebo bez. Některé materiály nevyžadují ohřev a na tiskovou plochu stačí nanést vrstvu malířské pásky, ke které roztavený materiál přilne. Jiné materiály na druhou stranu výhřev desky bez výhrady vyžadují. Požadované teploty pro výhřev desky se pohybují v rozsahu 40 °C až 110 °C v závislosti na použitém materiálu pro tisk. [2]

Jakmile je z trysky vytlačen roztavený plast, tak okamžitě začne chladnout a deformovat se. Za použití vyhřívané podložky se tomuto efektu zabrání a obecně stálý výhřev tištěného objektu zajišťuje výrazné zlepšení celkové kvality. Podložka je důležitá při tisku první vrstvy, aby se roztavený plast zachytil na podložku. Z konstrukčního hlediska je vyhřívaná podložka obvykle vyrobena z plošného spoje, na kterém je vhodně vyleptána topná spirála. [13]



Obr. 1.6 Vyhřívaná podložka [13].

1.4 Materiál pro 3D tisk

V současné době se u FDM nejčastěji používají dva základní typy materiálů. Jedná se o termoplasty ABS a PLA. Materiály jsou obvykle k dispozici ve formě struny o průměrech 1.75 a 3 mm.

- **ABS** (Acrylonitrile Butadiene Styrene) je jeden z nejpoužívanějších materiálů při metodě FDM. Všechny vytisknuté součástky na 3D tiskárnách jsou právě z tohoto materiálu [4]. Hlavními výhodami tohoto plastu jsou jeho tvrdost a odolnost při zachování nízké hmotnosti. Pro metodu FDM je důležitý koeficient tření, který

je v porovnání s PLA nižší, čímž je umožněno použití nižších protlačovacích sil při tisku. Z toho plyne výhoda při tvorbě malých objektů. Nevýhodou je tepelná roztažnost v porovnání s ostatními materiály. To při tisku způsobuje deformace objektů a je proto vyžadována vyhřívaná deska. Má dobrou tepelnou odolnost a je relativně levný (přibližně 500 Kč/kg). Jedná se o velice stálý plast s vysokou pevností. ABS plast je schopen odolávat teplotám až do 100 °C bez velkých ztrát na pevnosti (nejlepší vlastnosti vykazuje při teplotách -20 až 80 °C [9]). Tisková teplota se pohybuje v rozsahu 230 až 250 °C. [2] [18]

- **PLA** (Polylactic Acid) je zřejmě jako jediný vyroben z kukuřičného škrobu, čili se jedná o biologicky odbouratelný materiál. Ve 3D tisku je velmi rozšířen, hlavně kvůli jeho velmi nízké teplotní roztažnosti. Z tohoto důvodu není třeba používat vyhřívanou podložku. Teplota tání plastu PLA je vzhledem k ostatním termoplastům poměrně nízká (od 180 do 220 °C). Koeficient tření je v porovnání s ABS vyšší. Je rozměrově stálý a proto ve většině případů nevyžaduje vyhřívanou podložku. Prakticky jako jediný jej lze použít k tisku objektu přesahující rozměry 20 cm. Nevýhodou je vysoká míra křehkosti a zmíněná nízká teplota tání (výtisky jsou použitelné maximálně do 60 °C [4]). [9] [14] [19]

U obou materiálů je nutné zabezpečit vhodné skladování s minimální vlhkostí, protože přítomnost vlhkosti v materiálu zapříčiní tvorbu bublin v trysce [18] a náhlé vystřikování materiálu ven z trysky. Následky mají vliv na celkové vlastnosti modelu.

Pro kancelářské prostory je vhodnější použití PLA [18] z hlediska komfortu tisku, protože při tavení nemá dráždivý zápach jako ABS, u kterého je nutné zajistit dostatečnou ventilaci prostor.



Obr. 1.7 Porovnání ABS a PLA u vytištěného modelu [19].

2 SOFTWARE

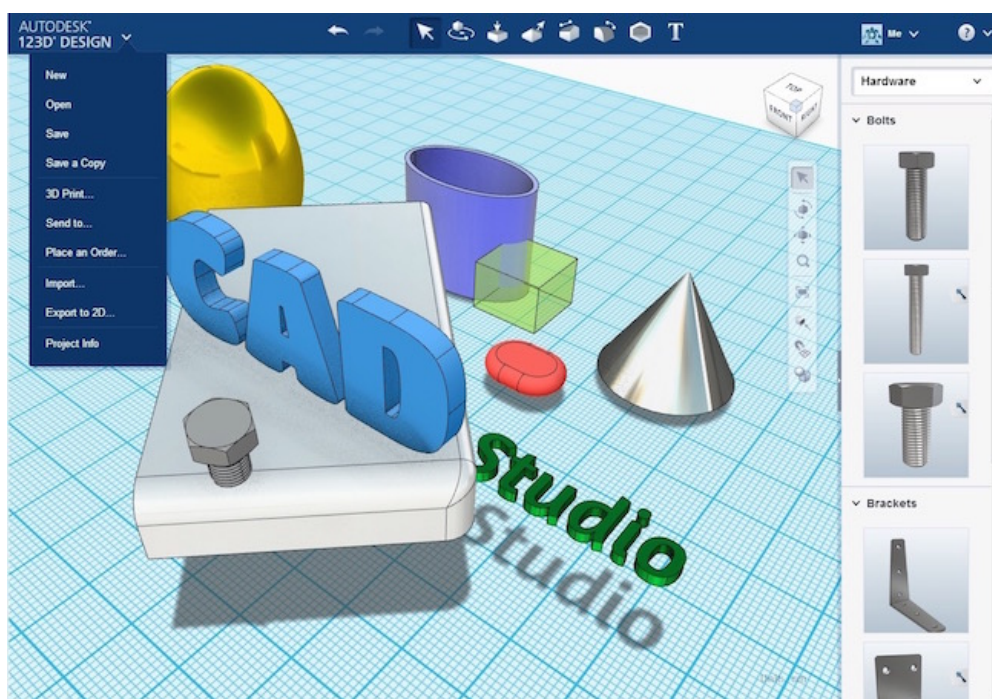
V této podkapitole je uvedeno několik CAD programů s podporou převodu objektů do formátu STL následované výčtem CAM programů pro generování G-kódu.

2.1 Pro tvorbu 3D modelů

Před samotným 3D tiskem je nutné objekt navrhnout ve vhodném CAD software.

- **Autodesk 123D Design** umožňuje jednoduše vytvářet a editovat 3D modely. K dispozici je online verze běžící v prostředí webového prohlížeče [21]. V rámci online aplikace lze vytvořené modely ukládat na cloudové uložení a odesílat je do 3D tisku nebo CNC výroby. Pro generování CNC kódu podle zadaných parametrů (G-kód) je využíváno jádro CAM systému HSMworks.

Je možné pracovat s materiály a modely skládat z připravených geometrických primitiv. Aplikace podporuje generování modelu do formátu STL. [20]

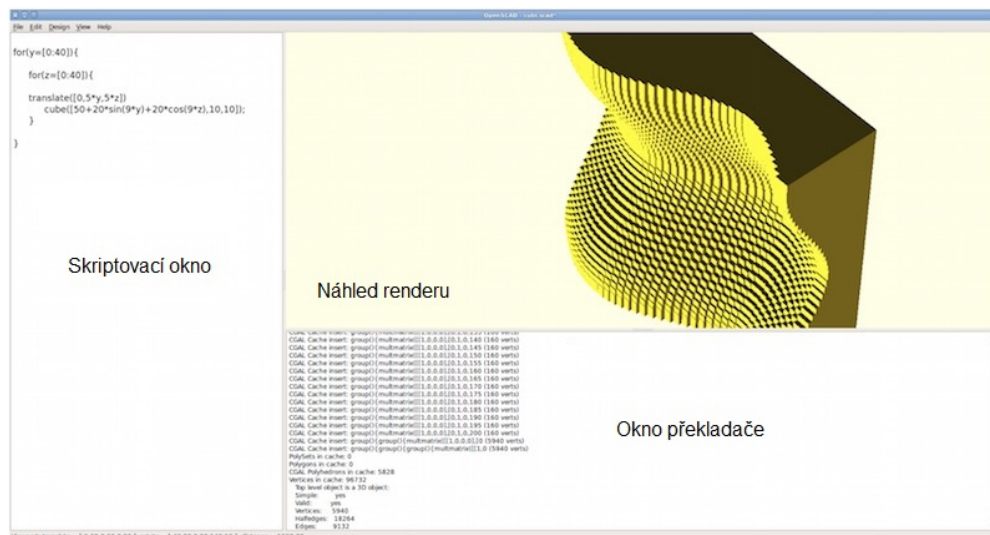


Obr. 2.1 Uživatelské rozhraní Autodesk 123D [20].

- **OpenSCAD** je open source 3D modelovací software. Není zaměřen pro umělecké 3D modelování, ale cílí na aspekty v oblasti CAD systémů. Modely nejsou vytvářeny vizuálně, ale jednoduchým textovým zápisem podobným základnímu programování. Není to tedy interaktivní modelovací software, ale jedná se o 3D překladač na základě vytvořeného skriptu. Umožňuje vytvářet parametrické mo-

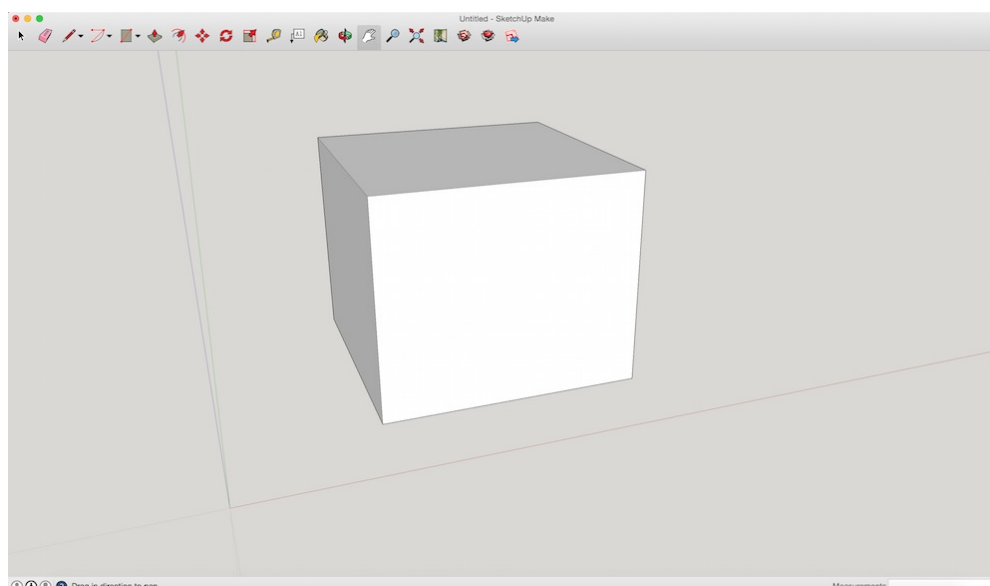
dely, tzn. že zadáním parametru se mění výsledný objekt bez přemodelování celého objektu. [6] [22].

Mimo svých podporovaných formátů dokáže OpenSCAD vytvářet 3D modely a následně je nativně exportovat do formátu STL pro 3D tisk.



Obr. 2.2 Uživatelské rozhraní OpenSCAD [6].

- **SketchUp** je velice jednoduchý nástroj pro vytváření 3D modelů. Nabízí většinu standardních nástrojů pro modelování a jeho funkcionalita lze ovlivnit instalací různých rozšíření. Jedním z rozšiřovacích pluginů je podpora pro formát STL, který v základu nepodporuje. [23]

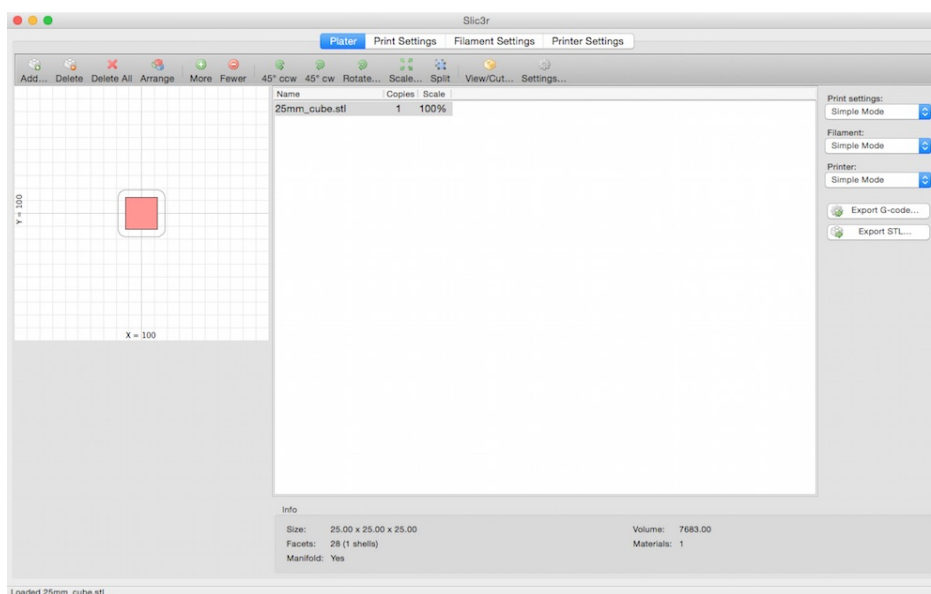


Obr. 2.3 Uživatelské rozhraní SketchUp Make.

2.2 Pro tvorbu řezů a generování G-kódu

3D tiskárny jsou obdobou CNC stroje, proto je nutné vytvořený model ve formátu STL převést na instrukce v G-kódu. Tohle zprostředkují CAM programy a v této podkapitole uvedena čtveřice nejvíce používaných. [6]

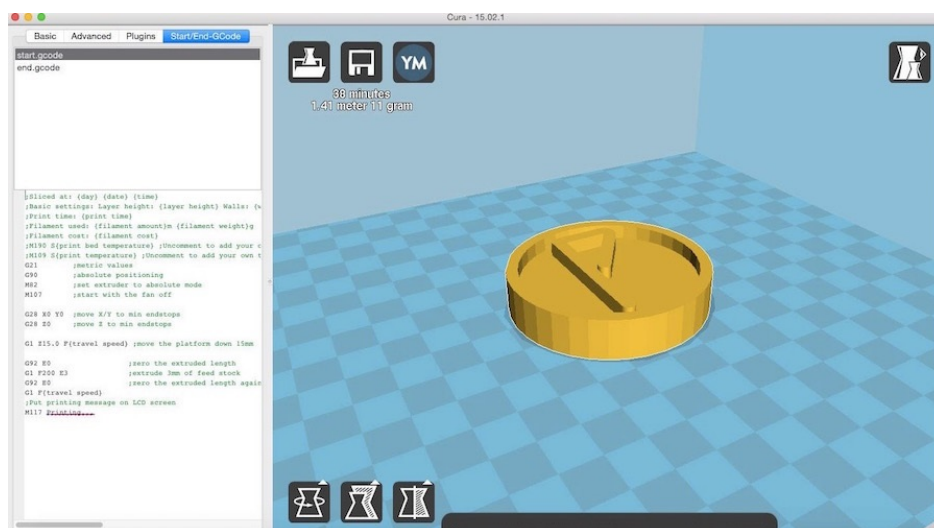
- **Skeinforge** je pravděpodobně nejstarší program pro tvorbu řezů. Je napsán v jazyce Python a býval výchozí program tiskárny Makerbot Replicator a mnoha tiskáren od RepRapu. V současné době je součástí prostředí MakerWare, kterým se řídí aktuální verze Makerbot tiskáren. Pro generování G-kódu je k dispozici několik Python skriptů, které ovlivňují rychlost výpočtu instrukcí, ale taktéž výslednou kvalitu řezů pro tisk. Skeinforge jako takový dnes používá málo uživatelů, ale jeho jádro pro tvorbu řezů se rozšířilo do většiny dnešních programů. [6]
- **Slic3r** je moderní open source program pro tvorbu řezů. V současnosti patří mezi nejoblíbenější. Nabízí velice přehledné uživatelské rozhraní a podrobné nastavení parametrů tisku. Podporuje tvorbu profilů s ohledem na typ tiskárny, materiálu a účelu. Slic3r dále umožňuje velmi pohodlnou přípravu tiskového plátu a využívá k výpočtu více vláken. [24]



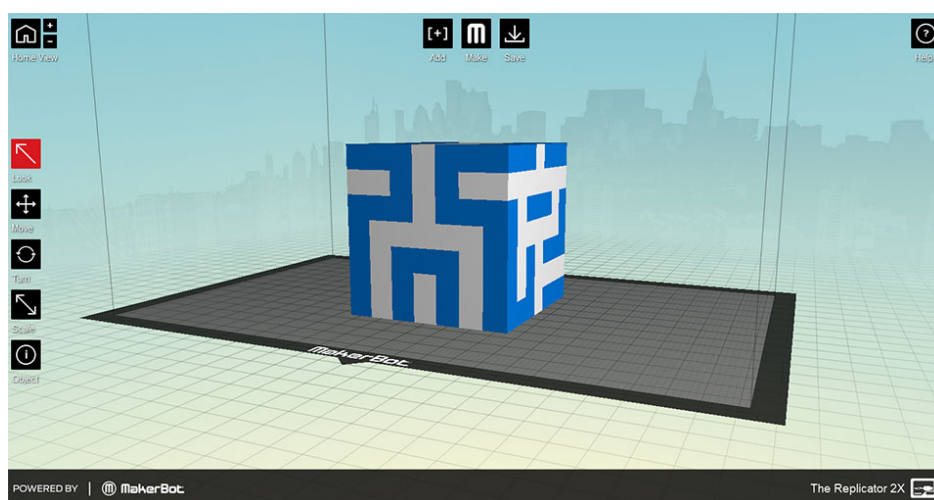
Obr. 2.4 Uživatelské rozhraní Slic3r.

- **Cura** je produktem společnosti Ultimaker, která si klade za cíl udělat z 3D tisku maximálně jednoduchou a efektivní činnost. Cura je plně předkonfigurována pro 3D tiskárny společnosti Ultimaker, ale lze ji použít i u tiskáren společnosti RepRap aj. Obsahuje veškeré funkce, které jsou spjaté s přípravou 3D modelu a následným tiskem. Pro tvorbu řezů využívá jádro ze Skeinforge, takže je zachována stejná kvalita výsledku.

- **MakerWare** jedná se o jednoduchý, ale zároveň robustní software, který je vyvíjen pro 3D tiskárny společnosti Makerbot. Nabízí moderní uživatelské prostředí s možnostmi pro načtení modelů ve formátu STL, jejich úpravu včetně náhledů ve 3D a nechybí možnost podrobného nastavení. Pro řezy používá vlastní engine, který je optimalizován pro silnější, rychlejší a více konzistentní výstupy. [25]



Obr. 2.5 Uživatelské rozhraní Cura.



Obr. 2.6 Uživatelské rozhraní MakerWare [25].

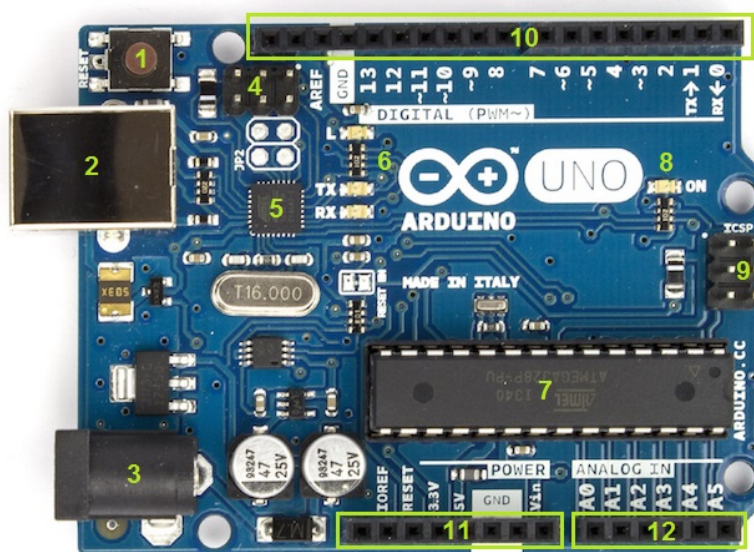
3 MIKROKONTROLÉR ARDUINO UNO

Arduino Uno [26] je mikrokontrolér založený na čipu ATmega328. Obsahuje 14 digitálních vstupů/výstupů, kde 6 z nich mohou být použity k generování impulzů pulzně šířkové modulace PWM (Pulse Width Modulation). Dále poskytuje 6 analogových vstupů, keramický rezonátor o frekvenci 16 MHz, USB rozhraní, konektor pro napěťový zdroj, ICSP konektor pro externí programování a tlačítko reset.

Napájení mikrokontroléru je možné za použití USB rozhraní nebo externího napájení. Deska je schopna pracovat při napětí 6 až 20 V. Pokud je ale dodáváno méně než 7 V, dochází k nestabilnímu chování. Výstupní napěťový pin pro 5 V by nebyl schopen toto napětí dodávat. Naopak při napětí vyšším než 12 V, dochází k přehřívání napěťového regulátoru, což může vést k poškození. Doporučené napětí je v rozsahu 7 až 12 V.

3.1 Popis mikrokontroléru

Osazený čip ATmega328 disponuje pamětí o velikosti 32 kB (z toho 0.5 kB využívá bootloader). Dále obsahuje paměti SRAM (2 kB) a EEPROM (1 kB).



Obr. 3.1 Mikrokontrolér Arduino Uno.

Vyznačením (Obr. 3.1) odpovídají následující vysvětlivky [31]:

- 1 - Resetovací tlačítko. Slouží ke spuštění nahraného programu znovu od začátku.
- 2 - USB konektor typu B.
- 3 - Napájecí konektor (7 až 12 V).

- 4 - ICSP rozhraní pro externí programování pomocí USB-Serial převodníku.
- 5 - USB-Serial převodník. Stará se o komunikaci mezi hlavním čipem a PC. Plní roli překladatele.
- 6 - Indikační LED diody L, RX a TX. Dioda L je připojena k digitálnímu výstupu č. 13, čímž se testuje například blikání bez připojené externí LED diody. Diody LX a TX slouží pro signalizaci probíhající sériové komunikace.
- 7 - Hlavní čip celé desky. V tomto případě ATmega328.
- 8 - Indikační LED dioda pro signalizaci aktivního napájení desky.
- 9 - ICSP rozhraní pro externí programování hlavního čipu (využívají některé rozšiřující shieldy).
- 10 - Digitální piny. Ty s označením ~ (3, 5, 6, 9, 10 a 11) podporují pulzně šířkovou modulaci (PWM).
- 11 - Napájecí výstupy Arduina.
- 12 - Analogové vstupy pro měření analogových hodnot.

Všech 14 digitálních pinů mikrokontroléru lze použít jako vstup nebo výstup za použití funkcí **pinMode()**, **digitalWrite()** a **digitalRead()**. Pracuje při napětí 5 V a každý z těchto pinů může poskytnout nebo přijmout maximální proud 40 mA. [26] [27]

3.2 Programové vybavení

Arduino je možné programovat v jazyce C nebo C++. Nejjednodušší je nicméně používat knihovnu Wiring. Jedná se aktuálně o velmi rozšířenou knihovnu, kterou lze kvůli její komplexnosti považovat za samostatný programovací jazyk. Tento jazyk původně vznikl pro vývojovou desku podobnou Arduinu a vychází z jiného otevřeného projektu s názvem Processing. Syntakticky je ale jazyk téměř shodný s jazykem C a patří do skupiny tzv. „C-like“ jazyků. [28]

Každý nový program (Sketch) má na začátku následující kód.

```
void setup() {  
  // put your setup code here, to run once:  
}  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

Funkce **setup()** je inicializační částí programu a vykonává se pouze jednou po spuštění programu. To znamená buď po připojení napájení, vykonáním restartu stiskem tlačítka Reset nebo po nahrání kódu do mikrokontroléru z vývojového prostředí. Například se zde mohou nastavovat časovače pro PWM na jednotlivých digitálních pinech. [30]

Funkce **loop()** je část programu, která se vykonává neustále dokola až do chvíle, kdy je mikrokontrolér odpojen od napájení.

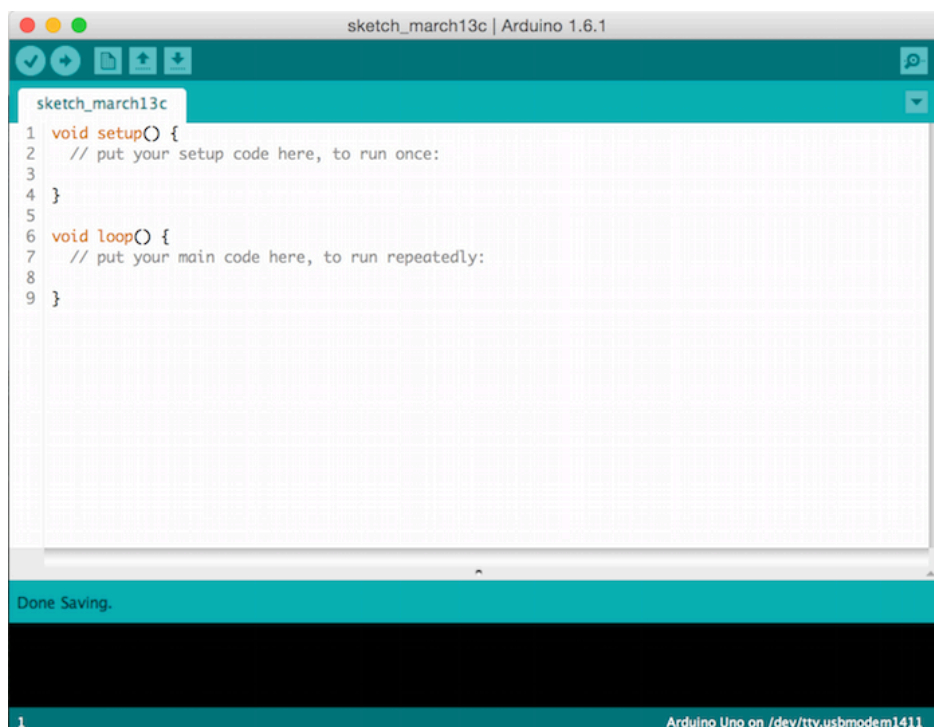
3.2.1 Datové typy

Každá proměnná musí mít svůj datový typ, kterým lze přesně určit co v proměnné bude a jak bude velká. Mezi základní datové typy patří číselný *int*, *double*, *long*, *float* nebo *byte* (velikost 8 bitů, rozsah 0 až 255). Dále nechybí logický datový typ *boolean* nebo znakový *char*.

Jazyk Wiring nabízí speciální datové typy. Například třída typu *String*, která umožňuje manipulaci s řetězcí mnohem komplexněji než se standardním polem znaků. Ať už se jedná o přímo implementované funkce pro nahrazení řetězců nebo hledání podřetězce v řetězci. [29]

3.3 Vývojové prostředí

Vývojové mikrokontroléry Arduino se programují v prostředí Arduino IDE (Obr. 3.2). Prostředí je v porovnání s jinými (Eclipse, Visual Studio aj.) relativně strohé a spíše připomíná rozšířený textový editor. Splňuje základní funkce jako zvýraznění syntaxe podporovaného jazyka nebo ověření implementovaného kódu.



Obr. 3.2 Vývojové prostředí Arduino IDE.

Postrádá funkce pro ladění programu nebo integrovanou dokumentaci. K dokumentaci lze přistupovat např. přes webový prohlížeč. Nahrání programu do mikrokontroléru

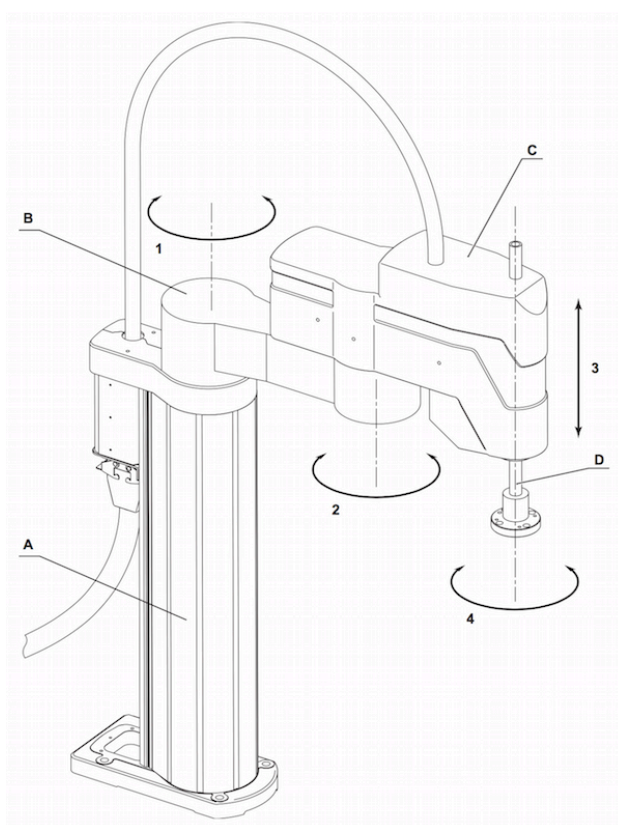
vyžaduje aktivní propojení pomocí USB s pracovní stanicí nebo lze využít jiné rozhraní určené pro externí programování. V prostředí se pouze zvolí odpovídající model desky a komunikační port, ke kterému je v dané chvíli připojen.

4 PRŮMYSLOVÝ ROBOT STÄUBLI RS40B

Jako prostředek, pro řízení polohy tiskové hlavy vůči pracovní ploše, byl zvolen průmyslový robot Stäubli RS40B. V následujících podkapitolách je popsáno robotické rameno Stäubli RS40, uspořádání SCARA a kontrolér CS8C včetně možností datové komunikace.

4.1 Robotické rameno RS40B

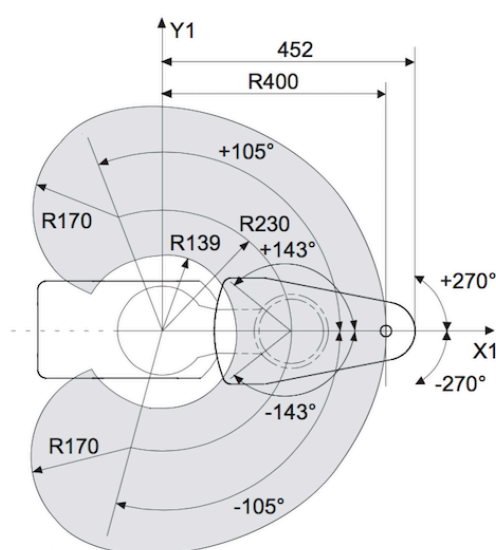
Pro pohyb v prostoru a mechanickou interakci s okolím slouží robotické rameno Stäubli RS40B [32]. Jedná se 4-osý robotický manipulátor typu SCARA, který umožňuje pohyb ve čtyřech stupních volnosti (4DOF). Na obrázku (Obr. 4.1) jsou vyznačeny jednotlivé články a klouby ramene. Počínaje podstavcem (A), následuje rameno (B), přední část ramene (C) a ložisko (D). Pohyby jednotlivých kloubů (1 až 4) jsou generovány servomotory. Pro osy 3 a 4 jsou servomotory vybaveny parkovací brzdou.



Obr. 4.1 Robotické rameno RS40B [32].

4.1.1 Rozsah pohybů, rychlost a hmotnostní zatížení

Na obrázku (Obr. 4.2) je znázorněn pracovní prostor průmyslového robota Stäubli RS40B.



Obr. 4.2 Pracovní prostor [32].

V tabulce (Tab. 4.1) jsou uvedeny číselné hodnoty pohybových rozsahů ramene.

Tab. 4.1 Rozsahy ramene [32].

Rozsah [mm]	400
Úhlový rozsah (osa 1) [°]	± 105
Úhlový rozsah (osa 2) [°]	± 143
Dráha (osa 3) [mm]	200
Úhlový rozsah (osa 4) [°]	± 500

Rychlosti uvedené v tabulce (Tab. 4.2) výrobce dělí na nominální a maximální. Maximální logická rychlost je 1.8 m/s. [32]

Tab. 4.2 Přehled rychlostí ramene [32].

	Nominální rychlost	Maximální rychlost
Osa 1 [°/s]	350	360
Osa 2 [°/s]	480	659
Osa 3 [mm/s]	1600	1923
Osa 4 [°/s]	1200	2020

Hmotnostní zatížení [32] má nominální hodnotu 2 kg a maximální 5 kg. Překročení nominálního zatížení je možné v určitém rozsahu při zavedení specifického omezení výkonů rychlosti a zrychlení ramene.

4.2 Kontrolér CS8C

Kontrolér CS8C [33] je součástí průmyslového robota Stäubli RS40B. Jeho činnost slouží k provádění veškerých výpočtových operací a pokynů pro řízení pohybu robotického ramena. Je složen z několika modulů. Na obrázku (Obr. 4.3) je tento kontrolér zobrazen a následně stručně popsán.



Obr. 4.3 Kontrolér CS8C [33].

Ukazatel (1) označuje procesorem řízené digitální zesilovače výkonu pro každou osu ramene. Napájení (2) RPS a (3) ARPS dodávají nezbytné napětí okolním zařízením pro správnou činnost. Nezbytné bezpečnostní funkce jsou uvedeny na RSI desce (4). K RSI desce je též připojen ruční programovací panel MCP a WMS panel pro změnu pracovního módu. Inteligentní část kontroléru se nachází v procesoru (5), který spravuje aplikace a řídí vstupy a výstupy. Uvedení kontroléru do chodu respektive jeho vypnutí je provedeno pomocí hlavního vypínače (6) a o přeměnu elektrické energie se stará modul napájení PSM (7).

4.2.1 Ruční programovací panel MCP

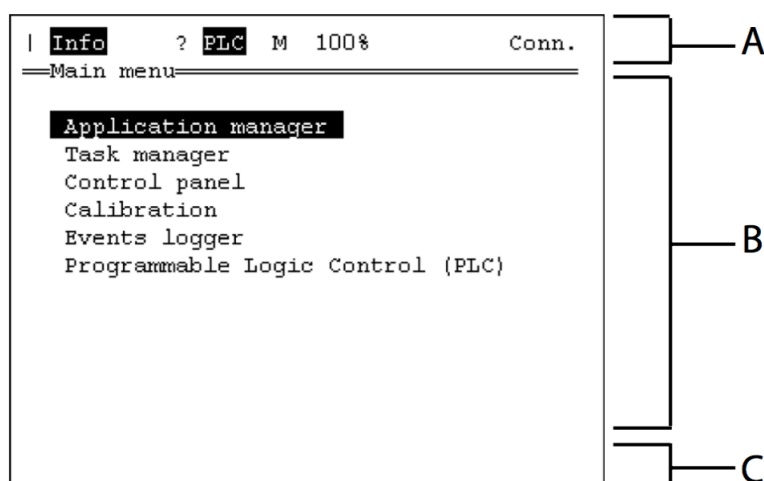
Pomocí programovacího panelu [33] připojeného ke kontroléru CS8C lze robota Stäubli RS40B manuálně ovládat, uvádět do provozu, konfigurovat jeho nastavení a programovat.

Součástí ovladače je monochromatický LCD displej a klávesnice se světelnou LED indikací u vybraných kláves. Mimo klasickou alfanumerickou klávesnici jsou k dispozici například klávesy pro volbu pracovního módu, napájení motorů, navigaci ad.

Displej panelu (Obr. 4.5) je složen ze tří hlavních oblastí, a to ze stavové lišty (A), pracovní stránky (B) a menu (C).



Obr. 4.4 Ruční panel MCP [33].



Obr. 4.5 Displej MCP [33].

Manuální panel rovněž obsahuje bezpečnostní prvky pro dosažení bezpečnosti práce s robotem.

4.2.2 Komunikační rozhraní

Kontrolér je vybaven čtyřmi USB porty na přední straně. Dalším rozhraním pro komunikaci jsou dva sériové porty RS232 a dva porty pro Ethernet.

Sériové porty slouží pro výměnu dat mezi aplikací VAL3 a vybavením buňky. Konfigurace sériové komunikace se provádí na displeji vstupů/výstupů na ručním programovacím panelu.

4.3 Programovací jazyk VAL3

Jazyk VAL3 [34] je vysokoúrovňový programovací jazyk vyvinutý pro řízení robotů společnosti Stäubli. Tento programovací jazyk kombinuje základní rysy standardních vysokoúrovňových jazyků (přebírá způsob zápisu neboli syntaxi) s funkcemi, které jsou specifické pro řízení průmyslových robotů:

- nástroje pro řízení robotů
- nástroje pro geometrické modelování,
- nástroje pro řízení vstupů/výstupů.

VAL3 se skládá z následujících prvků:

- | | |
|---------------|---|
| • aplikace | • konstanty |
| • programy | • proměnné (globální/lokální a parametry) |
| • knihovny | |
| • datové typy | • úlohy |

Aplikace ve VAL3 se skládají z programů obsahujících instrukce k vykonání, globálních proměnných a knihoven. Když je aplikace v běžícím stavu, obsahuje také sadu úloh (paralelní vykonání programů).

Výchozí aplikace vždy obsahuje programy **start()** a **stop()**, které nemohou mít žádné parametry a jsou vždy volány při spuštění a ukončení aplikace. V jednom okamžiku může běžet pouze jediná aplikace.

4.3.1 Datové typy

Proměnná nebo konstanta je charakteristická tím, že umožňuje systému možnost řídit aplikace a programy. Všechny proměnné a konstanty ve VAL3 mají specifický typ.

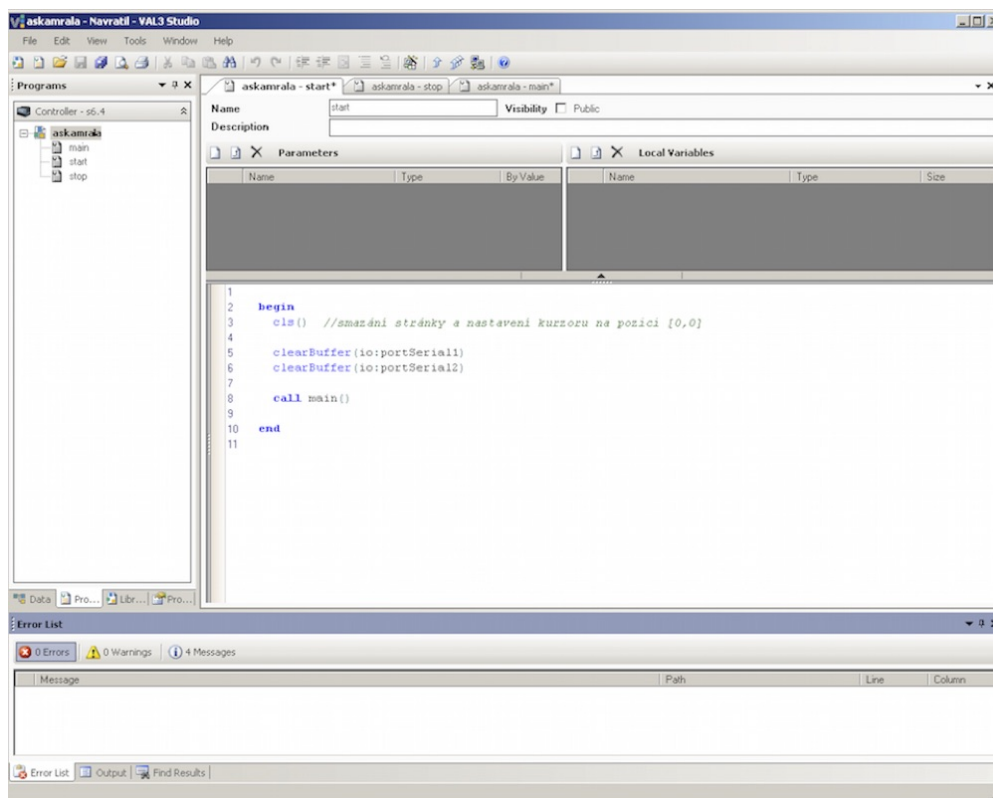
Mezi základní podporované typy patří logický *bool*, číselný *num*, řetězcový *string*, *dio* pro digitální vstupy/výstupy, *aio* pro analogové vstupy/výstupy a datový typ *sio*, který je pro vstupy/výstupy ze sériových portů a pro síťové (Ethernet) sokety.

Další skupinu tvoří strukturované typy, mezi které patří typ *trsf* (transformace v kartézském souřadném systému), *frame* (lokální souřadný systém), *tool* (nástroj upevněný na přírubě ramene), *point* (pozice nástroje v souřadném systému), *joint* (pozice kloubů), *config* (konfigurace ramene) a datový typ *mdesc* (parametry pohybu ramene).

4.3.2 Vývojové rozhraní

Programování ve VAL3 programovacím jazyku se provádí buď přímo v ručním programovacím panelu MCP, či v prostředí VAL3 studio, který je součástí prostředí Stäubli

Robotics Studio (SRS) běžící na počítači spojeném s řídicí jednotkou a hardwarovým klíčem. VAL3 Studio oproti programování ručním panelem poskytuje standardní funkce vývojového prostředí, k čemuž patří prostředky pro ladění nebo kontrola syntaxe.



Obr. 4.6 Vývojové prostředí VAL3 Studio.

II. PRAKTICKÁ ČÁST

5 HARDWARE

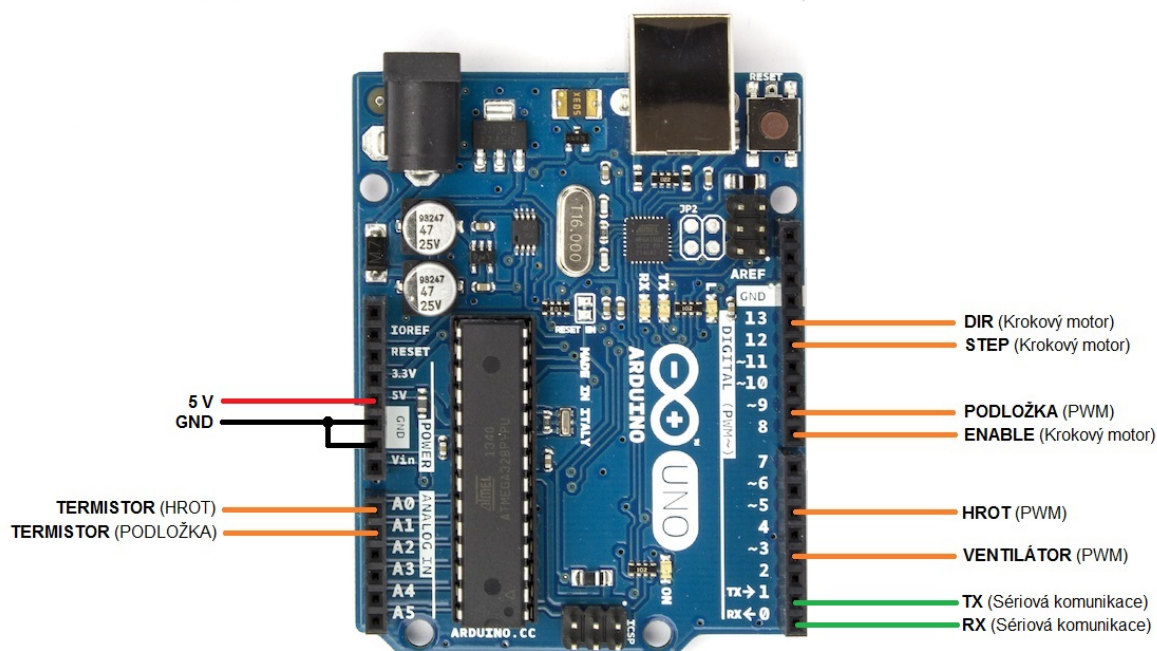
Prvním úkolem byl výběr vhodné tiskové hlavy a doplňujícího hardware k dosažení řízení a komunikace s dalšími zařízeními v dalších krocích řešení.

5.1 Arduino Uno

Jako řídicí prvek byl vybrán mikrokontrolér Arduino Uno, který nabízí dostatečné množství vstupů a výstupů.

Předpokladem je využití dvou analogových vstupů pro termistory, tří výstupů s PWM modulací pro regulaci teploty na hrotu, podložce a řízení ventilátoru umístěného na tiskové hlavě a tří digitálních výstupů pro krokový motor. Těmto předpokladům Arduino Uno pohodlně vyhovuje a bylo zbytečné volit vyšší model.

Na obrázku (Obr. 5.1) je znázorněno propojení jednotlivých vstupů a výstupů, výstupního napájení a komunikace.



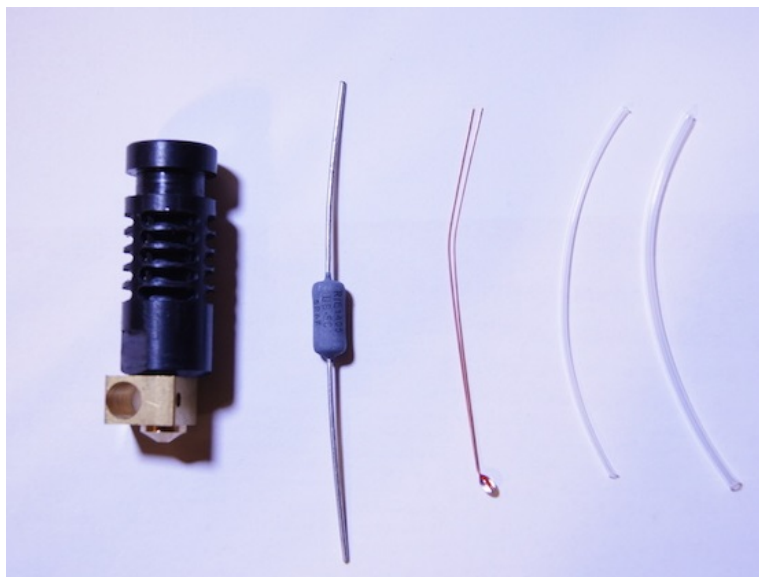
Obr. 5.1 Zapojení mikrokontroléru.

5.2 Tisková hlava

Výběr tiskové hlavy byl proveden na základě hledisek uvedených v teoretické části. Byl kladen nárok na kvalitní zpracování trysky. Dále bylo nutné zvolit materiál, kterým se bude tisknout. Byl volen termoplast ABS kvůli jeho vlastnostem, kterými jsou tvrdost a vyšší teplota tání. Tělo tiskové hlavy je dostupné ve formě vytištěných částí z ABS

plastu, které se ve výsledku upevní k sobě. Volbou nakonec byla tisková hlava složená z následujících součástí:

- **Hrot** - Jedná se o model J-Head MK V-B 0.35 mm [36]. Tento hrot je univerzální pro tisk termoplastem ABS i PLA. Na obrázku (Obr. 5.2) je zobrazena kompletní sestava. Mimo mosazného hrotu s PTFE trubicí byl součástí rezistor s odporem 5.6Ω pro vytápění trysky a $100k \Omega$ NTC termistor pro měření teploty.



Obr. 5.2 Zleva: Hrot, topné těleso, termistor, teflonová izolace pro termistor.

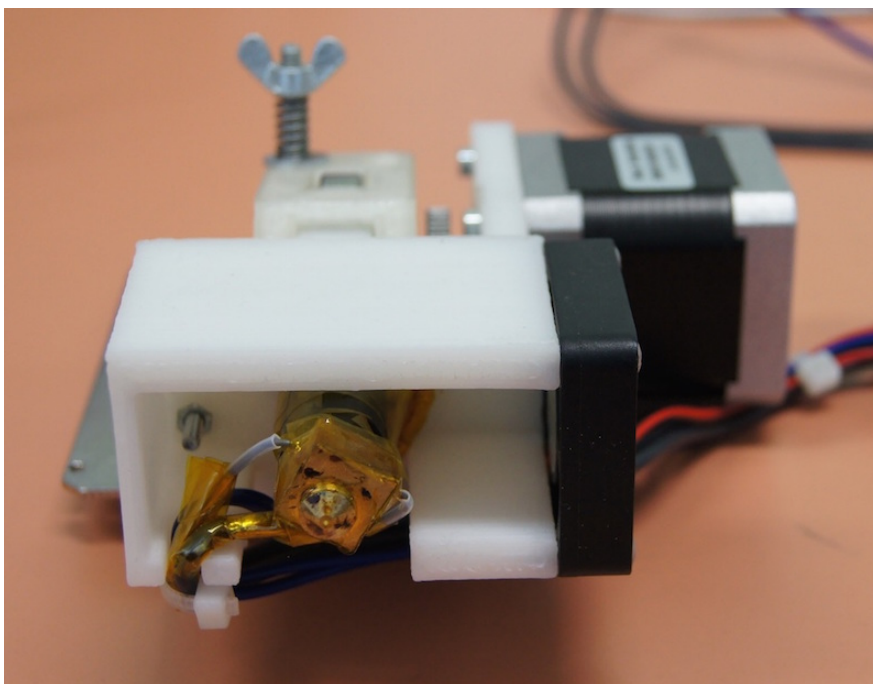
- **Krokový motor** - Jedná se o hybridní dvoufázový krokový motor Nema 17 [37] od společnosti Smart Automation.

Parametry motoru jsou:

Velikost kroku:	1.8°
Jmenovitý proud:	1.68 A/fáze
Odpor:	$1.65 \Omega/\text{fáze}$
Indukčnost:	2.8 mH/fáze
Počet vývodů:	4
Hmotnost:	0.35 kg

- **Pevná část tiskové hlavy** - Tělo je složeno ze tří částí, které se navzájem spojí a upevní šrouby. Tyto části jsou vytištěny z ABS plastu. Každý díl plní odlišnou funkci a jsou na ně upevněny příslušné komponenty.
- **Mechanické součástky** - Mezi mechanické součástky patří kompatibilní šrouby k uchycení a spojení všech částí a ozubená kola tvořící systém poháněný krokovým motorem pro tlačení materiálu.

Na obrázku (Obr. 5.3) je výsledná podoba tiskové hlavy po montáži. Kolem hrotu bylo nanесeno několik vrstev kaptonové pásky, aby se zamezilo uvolnění tepelného tělesa nebo termistoru. Tyto části nejsou v hrotu nijak vlepeny, ale pouze vloženy do příslušných bloků.



Obr. 5.3 Tisková hlava.

5.3 Vyhříváná podložka

Vyhříváná podložka je důležitou součástí tiskárny, bez které by tisk s ABS plastem byl praktický nemožný. Pokud by nebyla použita vyhříváná podložka, docházelo by při tisku k prudkému ochlazení, což by způsobilo deformaci, vnitřní pnutí v plastu a praskání.

Byla vybrána deska z tištěného spoje typu Mk2a [38], kterou je nutné napájet stejnosměrným napětím ve výši 12 V a její rozměry jsou 214×214 mm, přičemž pracovní plocha je 200×200 mm.

Součástí dodávky byl 100k Ω NTC termistor pro měření teploty.

5.4 Napájecí zdroj

Pro napájení byl použit průmyslový spínaný zdroj, typ CARSPA HS-350/12 s pevným výstupním napětím 12 V DC. Je vybaven ochranami vůči zkratu, přetížení, přepětí nebo přehřátí. Nárok byl kladen na vysokou proudovou zátěžnost, protože při zapnutém vytápění vyhřívací podložky do ní teče proud přes 10 A. Zvolený průmyslový zdroj má

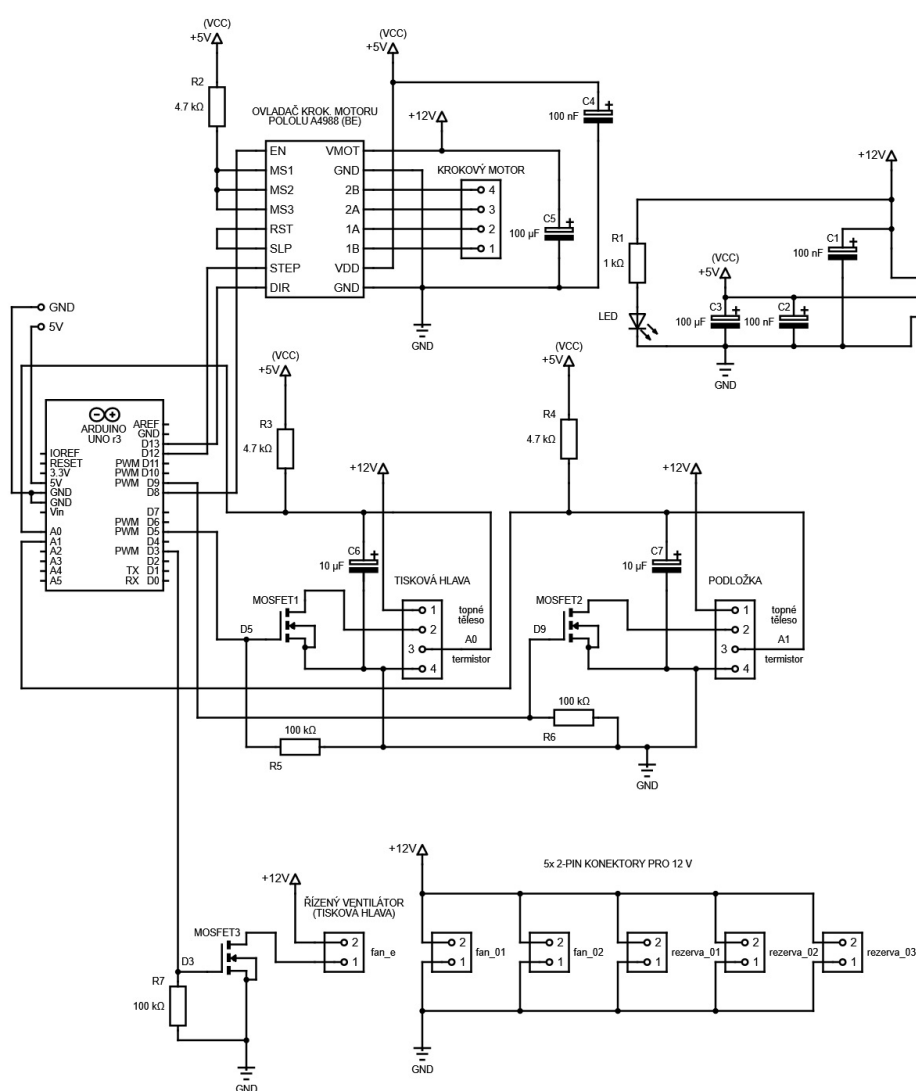
maximální proudové zatížení 29 A, které vyhovuje nárokům pro vytápění hrotu a desky a ostatní elektroniky.

5.5 Návrh desky

K řídicí jednotce bylo nutné navrhnut rozšiřující desku, která zajišťuje napájení komponent požadovanými hodnotami napětí (silová část) a řízení pomocí pulsně šířkové modulace ze strany mikrokontroléru (signálová část).

5.5.1 Schéma zapojení

Znázorněné schéma (Obr. 5.4) popisuje obvodové zapojení veškerých součástí.



Obr. 5.4 Schéma celkového zapojení.

Cílem návrhu desky bylo zajistit řízení příkonu vytápění hrotu, podložky a chladícího ventilátoru (umístěný na tiskové hlavě) ze strany mikrokontroléru, což zajistilo použití unipolárních tranzistorů. Dále jsou obsahem desky další integrované obvody, a to ovladač krokového motoru a převodník z logických úrovní TTL na standard RS232 pro sériovou komunikaci.

5.5.2 Přehled součástek

V následující tabulce (Tab. 5.1) jsou uvedeny použité součástky.

Tab. 5.1 Přehled elektrických součástek.

Označení	Hodnota	Název
C1, C2, C4	100 nF	Elektrolytický kondenzátor
C3, C5	100 μ F	Elektrolytický kondenzátor
C6, C7	10 μ F	Elektrolytický kondenzátor
R1	1 k Ω	Rezistor
R2, R3, R4	4.7 k Ω	Rezistor
R5, R6, R7	100 k Ω	Rezistor
LED	1.8 V / 20 mA	LED dioda
MOSFET(1, 2, 3)	-	Unipolární tranzistor

Veškeré kondenzátory v zapojení plní funkci filtru proti napěťovým špičkám. Tohle platí výhradně pro ovladač krokového motoru. Pro napětí 12 V je mezi VMOT a zemí paralelně umístěn kondenzátor C5 jako ochrana proti špičkám, které mohou přesáhnout maximální napětí a integrovaný obvod nevratně poškodit. Obdobné provedení platí i pro napětí 5V, tedy kondenzátor C4 mezi VDD a zemí.

Odpor R1 je předřadný odpor pro LED diodu signalizující aktivní napájení desky. Pro hodnotu předřadného odporu bylo nejprve vypočteno napětí na rezistoru odečtením napětí na diodě a napětí zdroje.

$$U_R = U_Z - U_D = 10.2 \text{ V} \quad (5.1)$$

Dále pomocí Ohmova zákona byl vypočten odpor rezistoru. Za U bylo dosazeno vypočtené napětí na rezistoru a za I byl dosazen proud, který smí diodou protékat.

$$R = \frac{U_R}{I_D} = 510 \text{ } \Omega \quad (5.2)$$

Vzhledem k vypočtenému odporu byl zvolen předimenzovaný 1 k Ω odpor.

Rezistory R2, R3 a R4 jsou tzv. pull-up rezistory. R2 slouží pro udržení logické 1

na vstupech MS1, MS2 a MS3, které jsou vzájemně propojeny. R3 a R4 jsou odpory využitě při výpočtech přepočtových tabulek mezi analogovými daty a °C.

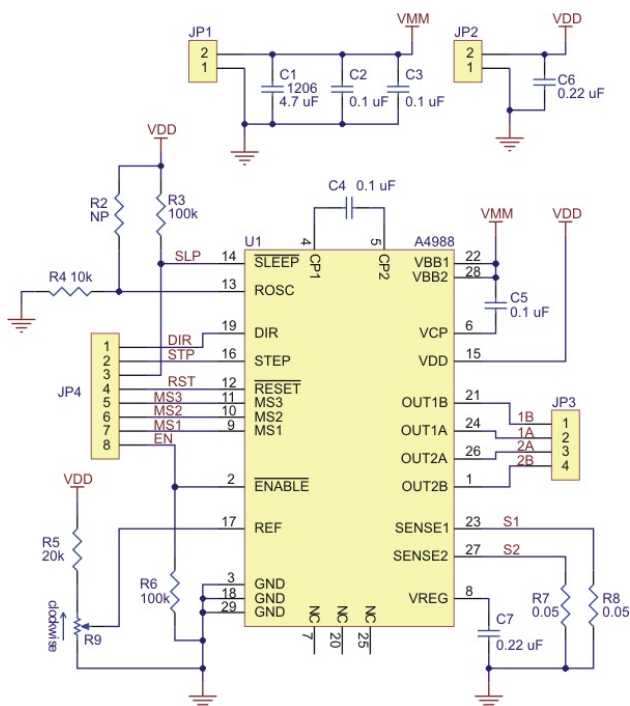
R5, R6 a R7 jsou pull-down rezistory, které jsou zapojeny mezi vstupem do hradla (D3, D5 a D9) a zemí. Slouží k udržení logické hodnoty 0 a díky hodnotě 100 kΩ do těchto vstupů prakticky neteče žádný proud.

Mosfety neboli unipolární tranzistory jsou typu N a slouží k zprostředkování řízení ze strany mikrokontroléru. Byl zvolen proudově předimenzovaný typ, aby se předešlo k nadměrnému přehřívání vzhledem k frekvencím PWM. Specifikace mosfetu jsou uvedeny v příloze P I [40].

5.5.3 Ovladač krokového motoru

Pro výkonové buzení krokového motoru byla použita integrovaná deska od výrobce Pololu. Jedná se o ovladač Pololu A4988, která má 16 pinů, z toho je 8 logických vstupů pro připojení k mikrokontroléru, 4 piny pro napájení a zbývající 4 piny pro připojení motoru.

Deska je osazena integrovaným obvodem A4988 dodávaným společností Allegro, potenciometrem, kterým se nastavuje efektivní hodnota proudu protékajícího motorem. Dále jsou součástí desky specifické odpory a kondenzátory, které zajišťují správnou funkci integrovaného obvodu. Na následujícím obrázku (Obr. 5.5) je schéma zapojení integrovaného obvodu.



Obr. 5.5 Schéma zapojení desky Pololu A4988 [35].

Čip A4988 je primárně určen pro bipolární řízení krokových motorů. Řízení je možné dosáhnout v plném, 1/2, 1/4, 1/8 a 1/16 kroku. Pro svoji činnost vyžaduje napájení 3.5 až 5 V pro logickou část a 8 až 35 V pro část výkonovou.

Logické napájení bude zprostředkováno mikrokontrolérem Arduino Uno a napájení motoru průmyslovým zdrojem.

Význam logických vstupů:

- **¬ENABLE** - Resetovací tlačítko. Slouží ke spuštění nahraného programu znovu od začátku.
- **MS1, MS2, MS3** - Kombinace jejich propojení určuje výsledné rozlišení kroku (viz Tab. 5.2).
- **¬RESET** - Resetování vstupního překladače, které je aktivní v log. 0.
- **¬SLEEP** - Úsporný režim, který je aktivní v log. 0.
- **STEP** - S každou přivedenou nástupnou hranou vykoná jeden krok.
- **DIR** - Určuje směr otáčení motoru.

Tab. 5.2 Tabulka pro jednotlivá rozlišení kroku [35].

MS1	MS2	MS3	Rozlišení kroku
0	0	0	Plný krok (1/1)
1	0	0	Poloviční krok (1/2)
0	1	0	Čtvrtinový krok (1/4)
1	1	0	Osminový krok (1/8)
1	1	0	Šestnáctinový krok (1/16)

5.5.4 Převodník RS232

Pro převod logických úrovní signálu na vyšší napětí potřebné pro sériovou komunikaci ve standardu RS232 byl původně zamýšlen převodník Pololu 23201a [39]. Tento adaptér umožňuje přístup ke všem signálním vodičům na DB9 konektoru (devátý je zem). Pro potřeby této realizace se vystačí se dvěma piny a to RX a TX. Ty jsou značeny z perspektivy dalšího zařízení, takže data vysílané z COM portu skrze pin TX jsou na převodníku přijímány na přijímacím pinu RX. Analogicky to platí naopak pro případ, kdy COM port jiného zařízení přijímá (RX) a z převodníku se vysílá (TX).

Převodník je osazen integrovaným obvodem MAX3238 od firmy Texas Instruments. Tento obvod obsahuje pět budičů a tři přijímače pro RS232. Tím je dosažen zmíněný převod plné sériové linky RS232 v 9ti linkovém provedení pro DB9 konektor. Obsahuje blok pro automatické vypnutí, který se aktivuje při odpojení komunikace nebo době

bez aktivní komunikace přesahující 30 sekund. Tím se obvod uvede do stavu nízké spotřeby.

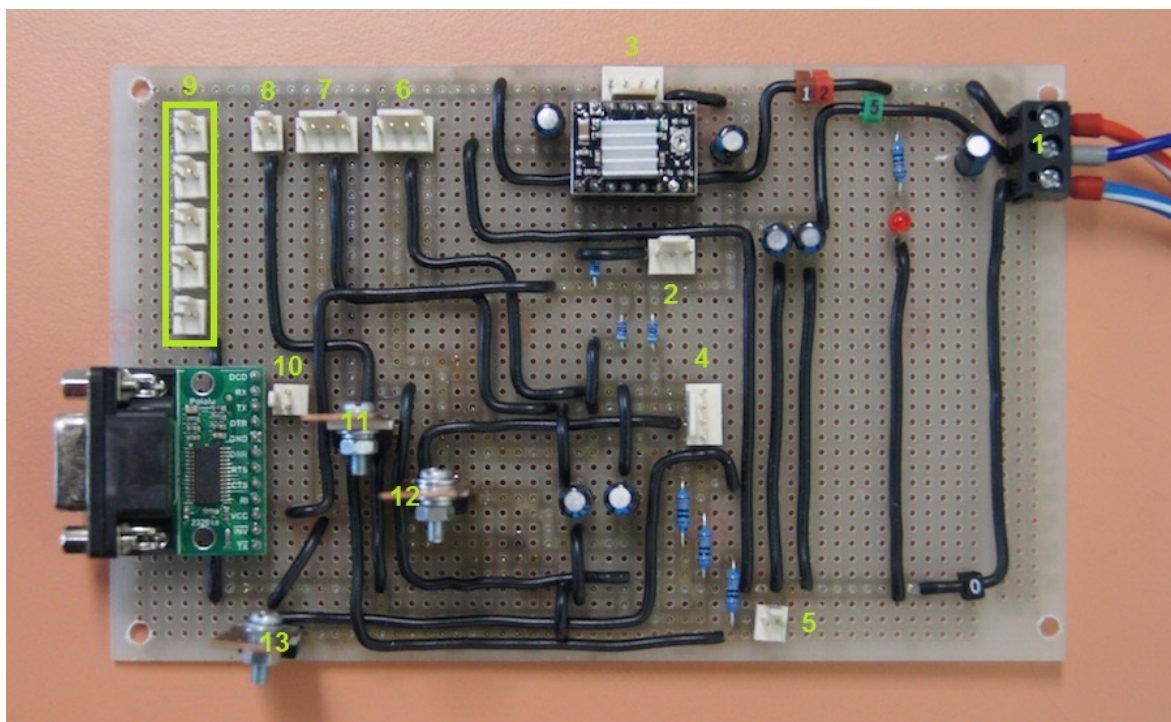
Během praktické realizace byl tento převodník nahrazen integrovaným obvodem, který byl na půdě fakulty dostupný. Převodník Pololu nevykazoval správnou činnost komunikace a náhradní řešení je osazeno shodným integrovaným obvodem MAX3238.

Vlastnosti obvodu: [41]

Napájení:	3 až 5.5 V
Datový přenos:	až 250 kb/s
Přepětová ochrana	15 kV

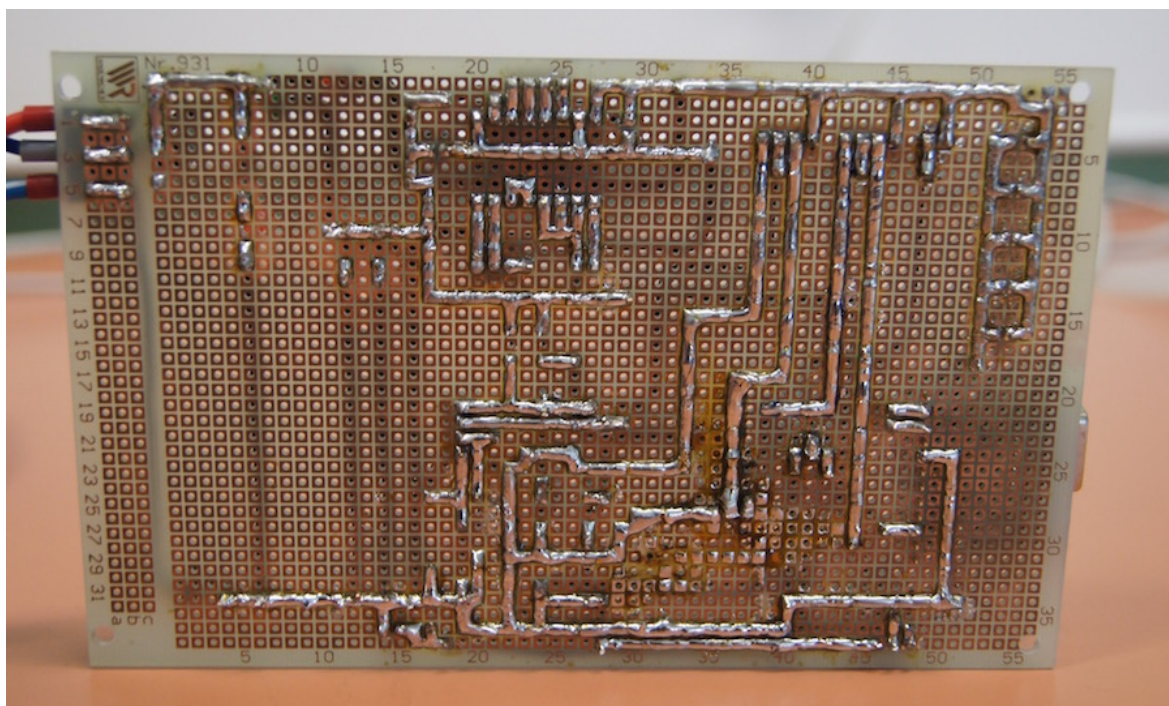
5.5.5 Výsledná realizace

Z uvedených součástek a integrovaných obvodů byla provedena realizace desky dle navrženého schématu zapojení. Na následujícím obrázku (Obr. 5.6 je pohled na hotovou desku shora dolů. Pro další propojení s ostatními součástmi jsou na desce vyvedeny konektory se zámkem.



Obr. 5.6 Realizovaná deska (pohled shora).

Na dalším snímku (Obr. 5.7) je pohled na spodní část realizované desky.



Obr. 5.7 Realizovaná deska (pohled zdola).

Vyznačením (Obr. 5.6) odpovídá následující:

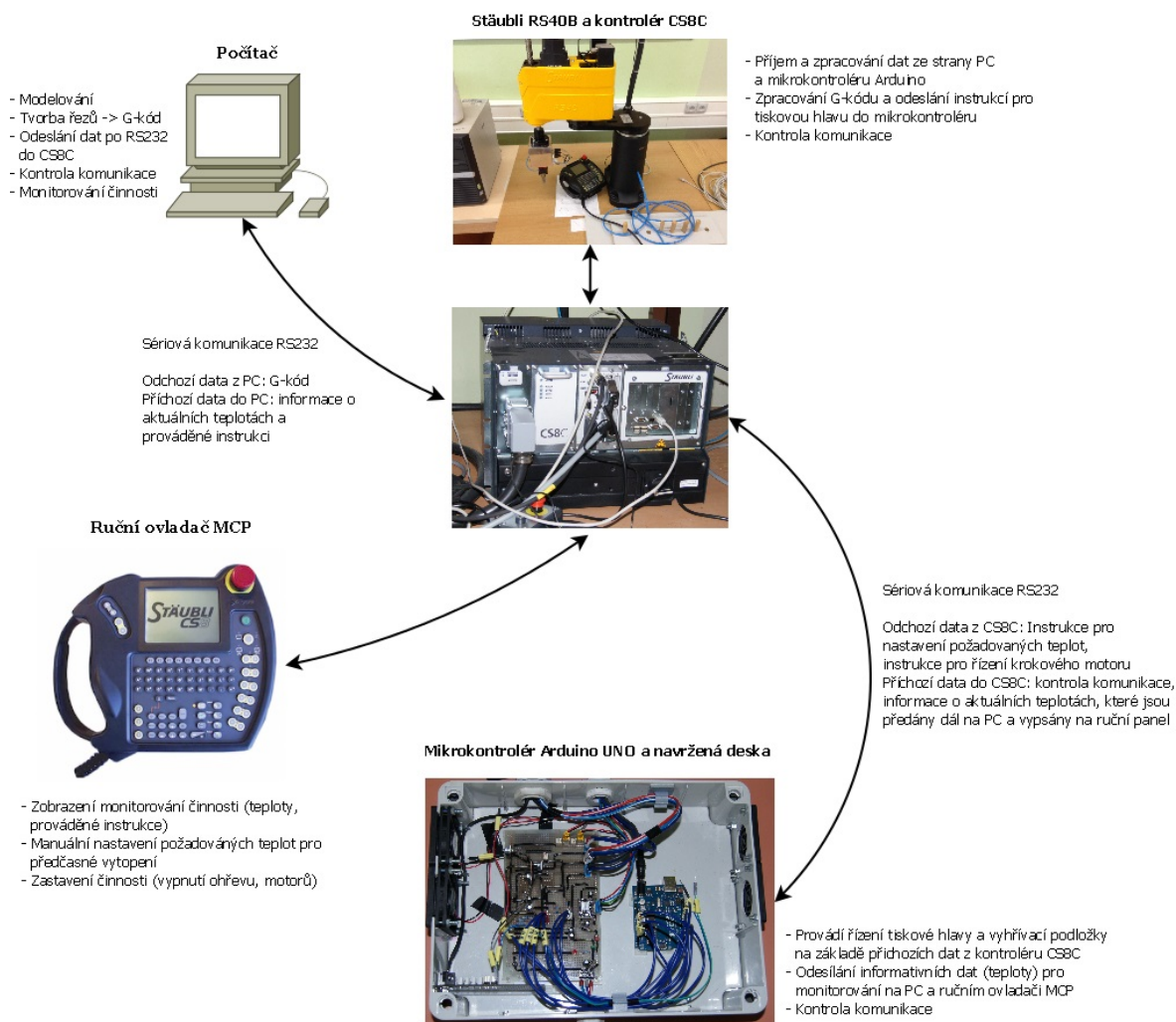
- **1** - Svorkovnice pro napájení. Červený vodič - 12 V (průmyslový zdroj), tmavě modrý - 5 V (mikrokontrolér), modrý s bílým pruhem - zem/GND (prům. zdroj).
- **2** - Konektor k ovladači krokového motoru ze strany mikrokontroléru. Připojují se digitální výstupy D8, D12 a D13.
- **3** - Výstupní konektor z ovladače krokového motoru. Připojuje se konektor krokového motoru a výstup odpovídá dvojicí 2B, 2A a 1A, 1B pro napájení dvou vinutí na motoru.
- **4** - Konektor ze strany mikrokontroléru pro připojení digitálních výstupů D5 a D9 pro PWM řízení vytápění topného tělesa v hrotu a podložky. Dále pro připojení dvou analogových vstupů A0 a A1, kterými je vedena analogová hodnota z termistorů.
- **5** - Konektor ze strany kontroléru, ke kterému je přiveden digitální výstup z pinu D3 pro řízení ventilátoru. Druhý pin je propojení země z Arduina.
- **6** - Konektor pro připojení tiskové hlavy k desce. 4 vodiče, dvojice pro termistor, další dvojice pro topné těleso.
- **7** - Konektor pro připojení vyhřívané podložky k desce. 4 vodiče, dvojice pro termistor, další dvojice pro topné těleso.

- **8** - Konektor pro připojení řízeného ventilátoru umístěného na tiskové hlavě.
- **9** - 5x 12 V konektory, např. pro napájení ventilátorů ke chlazení součástek
- **10** - Konektor pro propojení TX, RX mezi převodníkem a mikrokontrolérem Arduino.
- **11** - MOSFET pro řízený ventilátor na tiskové hlavě.
- **12** - MOSFET pro topné těleso.
- **13** - MOSFET pro vyhřívanou podložku.

Všechny MOSFETy byly mírně upraveny přidáním měděného plechu pro lepší odvod tepla. U tranzistorů k topnému tělesu a ventilátoru nebyly plně vyžadující, ale pro vyhřívanou podložku je nezbytný, protože pro výhřev podložky je nutné dodávat velice vysoký příkon, což se projeví tvorbou tepla v MOSFETu.

6 CELKOVÉ PROPOJENÍ JEDNOTLIVÝCH PRVKŮ SYSTÉMU

Na obrázku (Obr. 6.1) je znázornění zamýšleného propojení celého systému 3D tiskárny. Jednotlivé prvky obsahují popis předpokládané činnosti a propojení po sériové lince RS232 obsahují popis přenášených dat.



Obr. 6.1 Celkové propojení jednotlivých součástí.

7 NÁVRH REGULÁTORŮ

Nezbytnou částí pro realizaci řízení teploty hrotu tiskové hlavy a vyhřívací podložky bylo navržení vhodných parametrů PID regulátorů.

7.1 Identifikace

Nejprve bylo nutné provést identifikaci pro obě soustavy. Byl vytvořen program v řídicí jednotce Arduino k získání statických charakteristik. Toho bylo docíleno pomocí PWM modulace, kde hodnotě 0 odpovídá 0 % výkonu a hodnotě 255 odpovídá 100 % výkonu. Pro dosažení PWM modulace je použita funkce **analogWrite()**, která zapisuje analogovou hodnotu (PWM hodnotu) na příslušný pin.

V programu bylo důležité si nejprve definovat potřebné piny pro čtení teplot z termistorů a pro zápis PWM hodnot představující procentuální velikost dodávaného příkonu.

```
#define THERMISTOR_HOTEND_PIN 3
#define THERMISTOR_HEATBED_PIN 1
int heaterHotendPin = 5;
int heaterHeatbedPin = 9;
int fanPin = 3;
```

Dalším krokem bylo nastavení časovačů pro piny podporující PWM, které byly použity pro řízení. V základním nastavení mají jednotlivé skupiny pinů různé nastavení frekvencí, které mohou při nevhodném nastavení nevratně poškodit připojené tranzistory.

```
TCCR0B = TCCR0B & 0b11111000 | 0x04; //Casovac 0 (PWM piny 5 & 6)
TCCR1B = TCCR1B & 0b11111000 | 0x05; //Casovac 1 (PWM piny 9 & 10)
TCCR2B = TCCR2B & 0b11111000 | 0x07; //Casovac 2 (PWM piny 3 & 11)
```

Uvedený zdrojový kód udává nastavení pinů 5 a 6 na frekvenci 250 *Hz*, piny 9, 10 na frekvenci 32 *Hz* a piny 3, 11 na frekvenci taktéž 32 *Hz*. Hodnoty pro nastavení frekvencí byly čerpány ze zdroje [30]. Zvolené hodnoty frekvencí jsou zvoleny vzhledem k faktu, že při praktických experimentech došlo při vyšších hodnotách frekvence k poškození několika tranzistorů.

Další klíčovou částí programu bylo posílání určité úrovně PWM, což bylo provedeno v bloku **loop()**, který se stále opakuje.

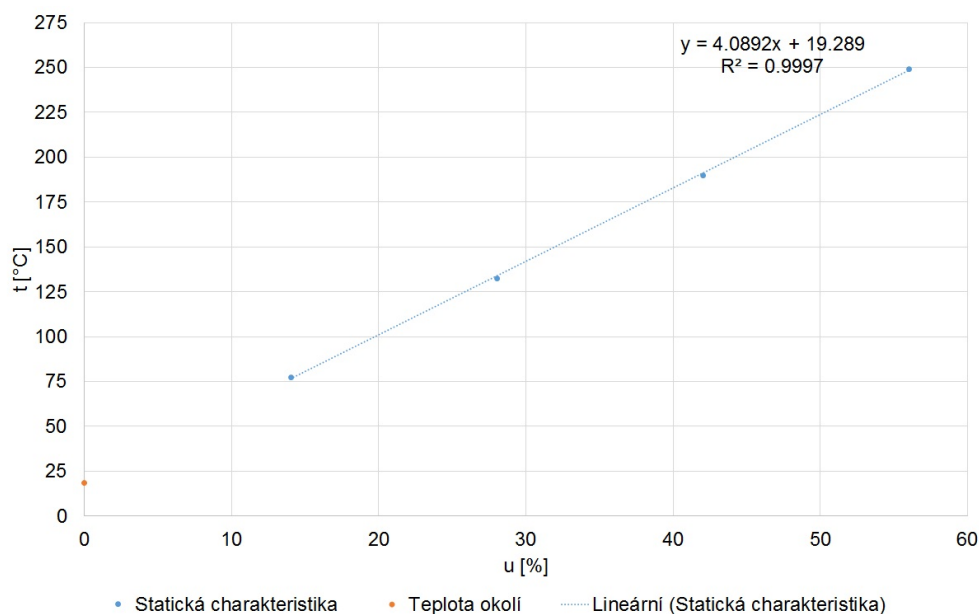
```
analogWrite(heaterHotendPin, 0); // Prikon heateru - hrot
analogWrite(heaterHeatbedPin, 0); // Prikon heateru - podlozka
analogWrite(fanPin, 127); // Prikon ventilatoru - 127/50 %
```

Pomocí těchto příkazů byla provedena identifikace kromě ventilátoru, který je v provozu po celou dobu činnosti při 50% příkonu.

7.1.1 Hrot

Byla naměřena statická charakteristika soustavy pro vytápění hrotu tak, že na vstup soustavy byla postupně přivedena 14 % (36 PWM), 28 % (72 PWM), 42 % (107 PWM) a 56 % (143 PWM) maximální hodnoty dodávaného příkonu.

Ustálené hodnoty výstupní veličiny byly zaneseny do následujícího grafu (Obr. 7.1).



Obr. 7.1 Statická charakteristika hrotu.

Ze statické charakteristiky je patrné, že soustava je lineární v celém požadovaném rozsahu. Předpokladem je vytápění hrotu na teplotu 230 °C, proto byla naměřena přechodová charakteristika při změně akční veličiny o 5 % v rozsahu z 50 % na 55 %.

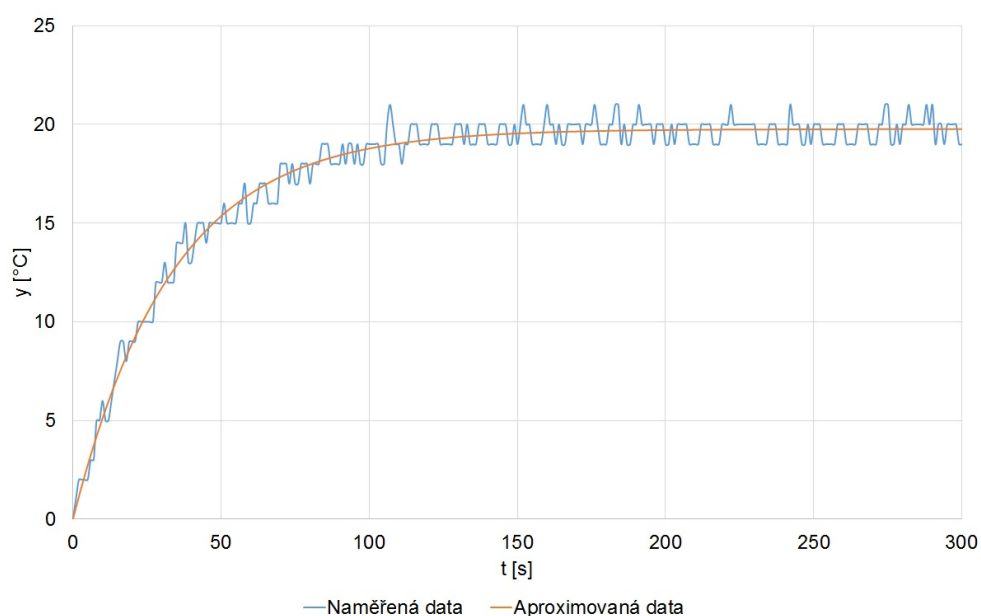
Získaná přechodová charakteristika byla patřičně upravena a následně identifikována (Obr. 7.2) v prostředí Matlab pomocí funkce *fminseach*, která hledá minimum vícerozměrné funkce bez použití gradientu. Minimum hledá na základě kritéria, které provádí rozdíl mezi simulovanou a reálnou veličinou.

Přenos aproximované soustavy je ve tvaru

$$G_s = \frac{19.763}{(33.62s + 1)(0.03431s + 1)} \quad (7.1)$$

Výsledná přenosová funkce je s ohledem na změny akční veličiny dle $k = \frac{\Delta y_{max}(t)}{\Delta u(t)} \left[\frac{^\circ\text{C}}{\%} \right]$ ve tvaru

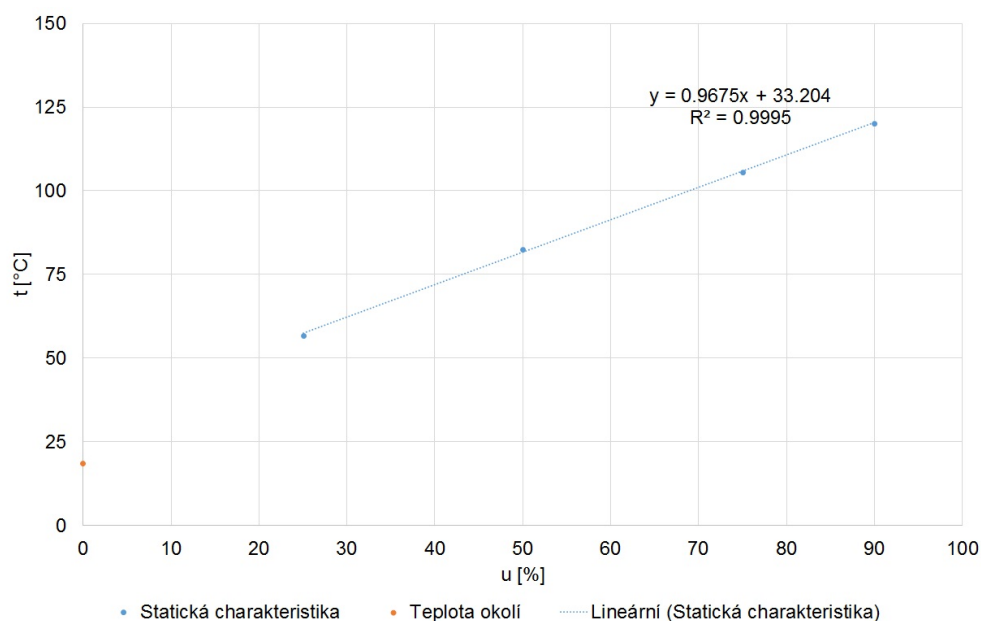
$$G_s = \frac{3.9526}{(33.62s + 1)(0.03431s + 1)} \quad (7.2)$$



Obr. 7.2 Srovnání naměřených dat s identifikovanou soustavou.

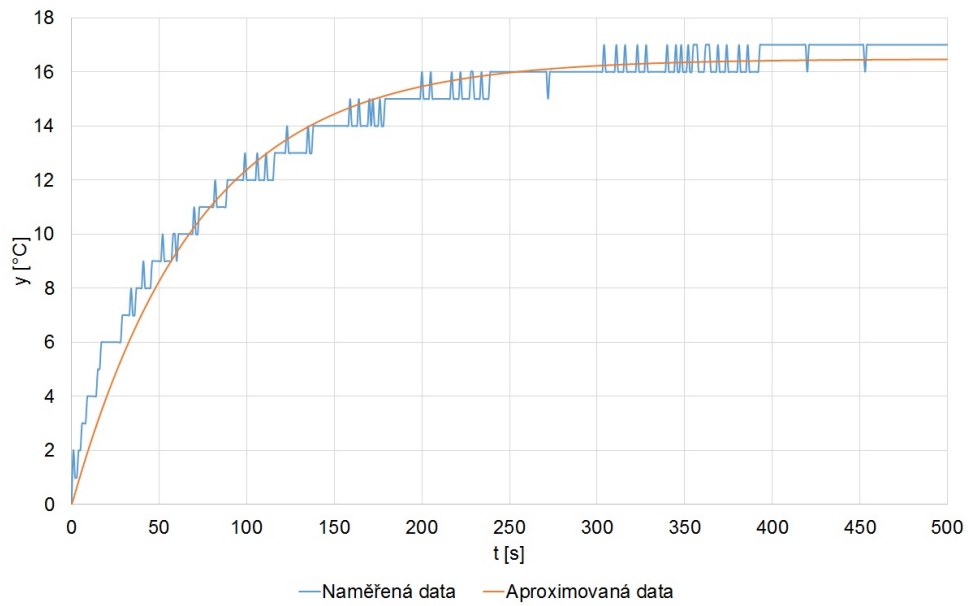
7.1.2 Vyhřívání deska

Analogickým postupem se provedla identifikace přechodové funkce pro vyhřívanou podložku. Na vstup soustavy byla postupně přivedena 25 % (64 PWM), 50 % (127 PWM), 75 % (191 PWM) a 90 % (230 PWM) maximální hodnoty dodávaného příkonu.



Obr. 7.3 Statická charakteristika vyhřívání podložky.

Dále byla naměřena přechodová charakteristika v rozsahu z 50 % se skokem na 55 % o 20 %.



Obr. 7.4 Srovnání naměřených dat s identifikovanou soustavou.

Přenos aproximované soustavy je ve tvaru

$$G_s = \frac{16.475}{(71.66s + 1)(0.174s + 1)} \quad (7.3)$$

Výsledná přenosová funkce je s ohledem na změny akční veličiny dle $k = \frac{\Delta y_{max}(t)}{\Delta u(t)} \left[\frac{^{\circ}C}{\%} \right]$ ve tvaru

$$G_s = \frac{3.9526}{(33.62s + 1)(0.03431s + 1)} \quad (7.4)$$

7.2 Výpočet parametrů

Návrh parametrů regulátoru bylo provedeno metodou požadovaného modelu (inverze dynamiky). Pro regulovanou soustavu

$$\frac{k}{(T_1s + 1)(T_2s + 1)} e^{-T_d s} \quad (7.5)$$

platí pro výpočet stavitelných parametrů PID regulátoru při nulovém dopravním zpoždění ($T_d = 0$) následující vztahy [42]

$$k_p = \frac{2T_I}{k(2T_w + T)} \quad (7.6)$$

$$T_I = T_1 + T_2 - T \quad (7.7)$$

$$T_D = \frac{T_1 T_2}{T_1 + T_2} - \frac{T}{4} \quad (7.8)$$

V programu mikrokontroléru je pro funkci PID regulace zvolena perioda $T = 100ms$, ostatní parametry se dosadí ze získaných přenosových funkcí.

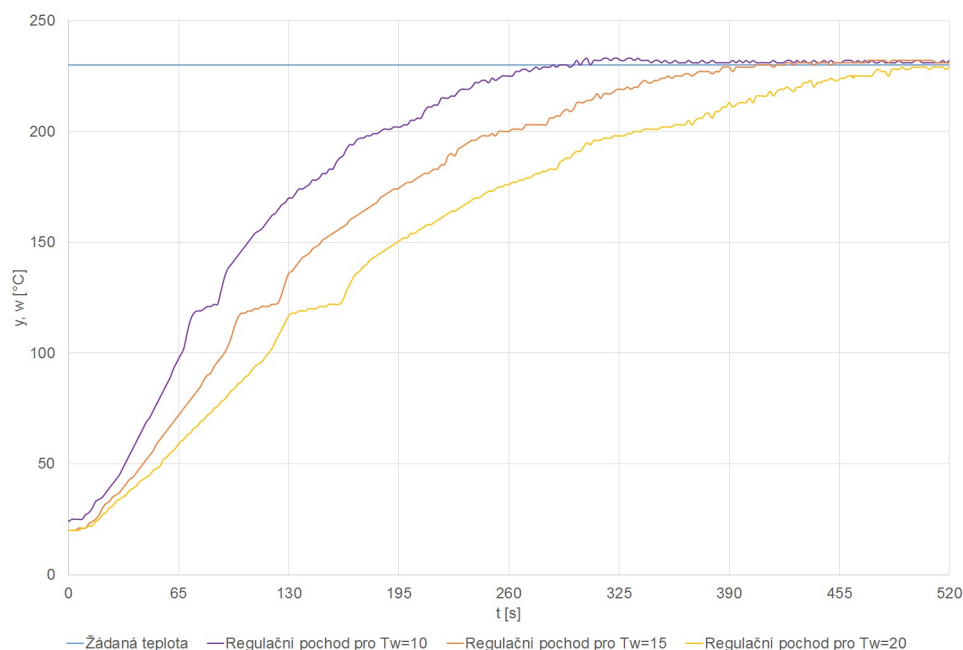
V následující tabulce (Tab. 7.1 je přehled vypočtených stavitelných parametrů pro oba PID regulátory při různých nastavení požadované časové konstanty T_w

Tab. 7.1 Vypočtené parametry PID regulátorů.

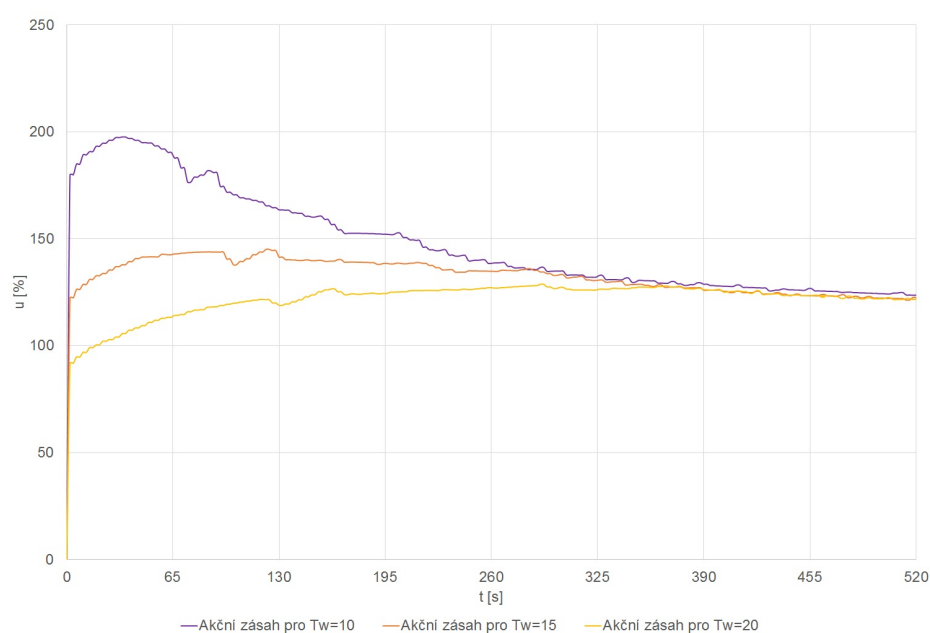
Topné těleso hrotu				Vyhřívaná podložka			
T_w	k_P	k_I	k_D	T_w	k_P	k_I	k_D
10	0.8513	0.0253	0.0292	10	8.6649	0.1208	1.2874
15	0.5676	0.0169	0.0194	20	4.3433	0.0605	0.6453
20	0.4257	0.0126	0.0146				

7.2.1 Regulační pochody

Pro vypočtené stavitelné parametry bylo provedeno měření regulačních pochodů pomocí programu mikrokontroléru. Měření proběhlo pro požadovanou teplotu $230\text{ }^{\circ}\text{C}$.



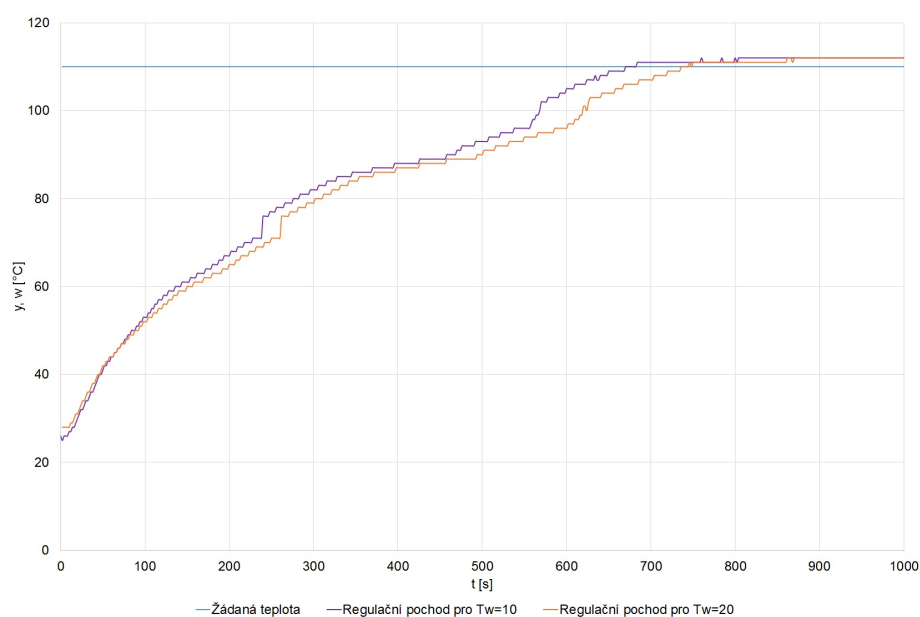
Obr. 7.5 Regulační pochody pro topné těleso hrotu.



Obr. 7.6 Průběhy akčních zásahů pro topné těleso hrotu.

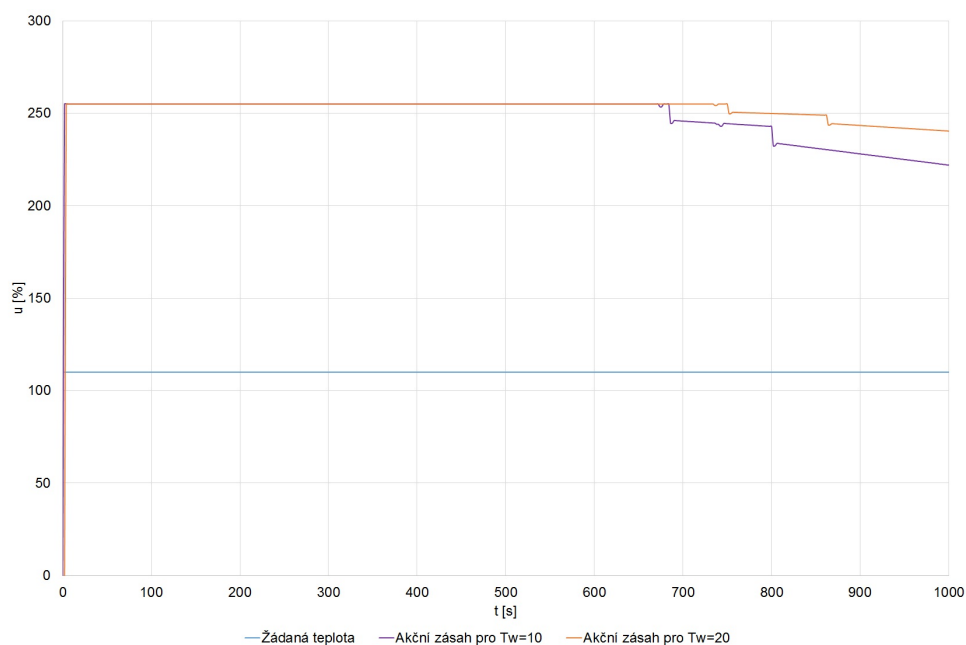
Z průběhů (Obr. 7.5 a 7.6) je zřejmé, že se zvyšující časovou konstantou T_w je regulační pochod pomalejší a ustálení na žádané teplotě trvá delší dobu. Bylo ale dosaženo nižšího akčního zásahu na začátku regulace.

Z praktického hlediska není důvod u regulace teploty hrotu používat nízkou hodnotu časové konstanty T_w , protože vyhřátí desky trvá značně dlouhou dobu. A je zbytečné, aby požadovaná teplota trysky byla dosažena za poloviční čas a následně se muselo čekat na vyhřátí podložky.



Obr. 7.7 Regulační pochody pro topné vyhřívanou podložku.

U další dvojice grafů (Obr. 7.7 a 7.8) jsou znázorněny průběhy pro vyhřívanou desku s požadovanou teplotou $110\text{ }^{\circ}\text{C}$ při $T_w = 10$ a $T_w = 20$. Je patrné, že podložka vyžaduje 100% příkon (255 PWM) a bez ohledu na zvolenou časovou konstantu jsou regulační pochody velice podobné. Volitelná časová konstanta prakticky nemůže nijak ovlivnit rychlost ohřevu. Jedině jejím zvýšením by se prodloužil regulační pochod, což je v této realizaci nežádoucí.



Obr. 7.8 Průběhy akčních zásahů pro vyhřívanou podložku.

Pro praktické řízení tedy budou použity stavitelné parametry PID regulátorů pro časovou konstantu $T_w = 15$ u regulace výhřevu tiskové hlavy a pro $T_w = 10$ u regulace vyhřívané podložky.

8 PROGRAMOVÁNÍ ARDUINO UNO

Po zpracování naměřených dat a získání stavitelných parametrů PID regulátorů následovala implementace programu pro mikrokontrolér Arduino Uno včetně protokolu, kterým bude přijímat instrukce po sériové lince. Vytváření aplikace probíhalo v prostředí Arduino IDE a aplikace byla pojmenována *program_mc*.

8.1 Řízení

Pro programové řízení byla použita knihovna *PID Library*, která je k dispozici pro mikrokontroléry Arduino.

Pro funkci **PID()** bylo nutné nejprve deklarovat nezbytné proměnné, na které se bude odkazovat.

```
double Setpoint_hotend, Input_hotend, Output_hotend, Setpoint_heatbed,
Input_heatbed, Output_heatbed;}
```

Následně byla provedena deklarace funkcí pro oba PID regulátory a konstantních proměnných definujících rychlost opakování algoritmu.

```
PID PID_hotend(&Input_hotend, &Output_hotend, &Setpoint_hotend, Kp_hotend,
Ki_hotend, Kd_hotend, DIRECT);
// Input_hotend - teplota Hot Endu z termistoru
// Output_hotend - u(t) pro PWM
// Setpoint_hotend - zadana teplota
```

```
PID PID_heatbed(&Input_heatbed, &Output_heatbed, &Setpoint_heatbed,
Kp_heatbed, Ki_heatbed, Kd_heatbed, DIRECT);
// Input_heatbed - teplota Heated Bedu z termistoru
// Output_heatbed - u(t) pro PWM
// Setpoint_heatbed - zadana teplota
```

```
const int sampleRate = 100; // rychlost PID smycky [ms]
const int sampleRate2 = 100; // rychlost PID smycky [ms]
```

Dále v části programu **setup()**, která slouží k inicializaci, byly nastaveny parametry pro funkce PID, kde **SetMode()** s hodnotou *AUTOMATIC* definuje automatický režim a **SetSampleTime()** přiřazuje zvolené doby pro opakování výpočtu algoritmu.

```
PID_hotend.SetMode(AUTOMATIC); // zapnutí PID smycky
PID_hotend.SetSampleTime(sampleRate); // nastavení rychlosti
PID_heatbed.SetMode(AUTOMATIC); // zapnutí PID smycky
PID_heatbed.SetSampleTime(sampleRate2); // nastavení rychlosti
```

Nakonec je nutné v opakující se smyčce programu definovat funkci **Compute()**, která provádí samotný výpočet dle nastavené frekvence v **SetSampleTime()**.

Funkcí **analogWrite()** se výsledek ve formě hodnoty PWM pošle na příslušný digitální pin.

```
PID_hotend.Compute(); // spusteni PID pro trysku
PID_heatbed.Compute(); // spusteni PID pro podlozku
// vystup z PID na 0-255 PWM trysky
analogWrite(heaterExtruderPin, Output_hotend);
// vystup z PID na 0-255 PWM podlozky
analogWrite(heaterHeatbedPin, Output_heatbed);
```

8.2 Komunikační protokol

Pro navázání sériové komunikace je nutné v inicializační části programu spustit funkci **Serial()** včetně nastavení rychlosti.

```
Serial.begin(9600); // spusteni seriového portu, rychlost 9600 baudu
Serial.println("Seriova komunikace aktivni."); // kontrolni zprava
```

Pak ve funkci **loop()** byly naprogramovány podmínky, kterými se kontrolují a porovnávají příchozí instrukce od dalšího zařízení v rámci aktivní sériové linky.

Pro ukázkou je zde jedna z podmínek, která kontroluje tvar *tXXX*, kde *XXX* musí být v rozsahu 201 až 244 °C, jinak není podmínka splněna a topné těleso v trysce nezačne topit.

V opačném případě se zadaná hodnota přiřadí k proměnné *Setpoint_hotend*, která tuto hodnotu nastaví jako žádanou a bude zahájen ohřev.

```
if (retezec[0] == 't') { // topeni trysky na tisk. hlave
    retezec.remove(0, 1);
    tep_t = retezec.toInt();
    // kontrola zadaneho rozsahu teplot
    if (tep_t > 245 || tep_t < 200)
    {
        tep_t = 0; // nezacne topit dokud neni v rozsahu
    }
    Setpoint_hotend = tep_t;
}
```

Pro každou takovou instrukci se kontroluje ukončovací znak, tzv. terminátor, který definuje konec přijímaného řetězce po sériové lince.

Kompletní zdrojový kód programu je uveden v příloze P II.

ZÁVĚR

Úkolem práce byla implementace 3D tisku na průmyslovém robotu Stäubli RS40B. Tato realizace se skládala z mnoha podproblémů, které bylo nutné řešit.

Úplným počátkem práce byl vhodný výběr komponent pro realizaci tisku. Jednalo se o volbu vyhovující tiskové hlavy a vzhledem ke zvolenému materiálu i vyhřívané podložky. Všechny tyto komponenty vyžadovaly určitý čas pro montáž. Potom byla na řadě elektrikářská činnost, kdy bylo potřeba zvolit minimálně lehce předimenzovanou kabeláž pro propojení s dalšími prvky. Například vyhřívanou podložkou může v danou chvíli protékat proud vyšší nežli 10 A, proto bylo nutné s ohledem na proudovou zatížitelnost kabelů zvolit ideální průřezy. Dále stojí za zmínku topné těleso hrotu, které vytápí trysku na teploty přesahujících 200 °C, což se může podepsat na izolaci kabeláže, která je v blízkosti topného odporu. K prevenci jakýchkoliv problémů byla v tomto případě použita kabeláž se silikonovou izolací, která má pracovní teploty až do výše 180 °C.

Potom následovala volba vhodného hardware. V práci byla provedena analýza předpokládaného počtu vstupů a výstupů, kterým pohodlně vyhovoval mikrokontrolér Arduino Uno. Pro něj bylo nutné navrhnout rozšiřující desku, neboť samotný mikrokontrolér není schopen řídit prvky, které vyžadují napájení vyšší než 5 V. Mimo návrhu desky bylo nutné zajistit dostačující zdroj. Nároky byly kladeny na vysokou proudovou zatížitelnost a výkon.

Pak přišla část návrhu desky, pro kterou bylo vynaloženo až moc vysoké množství času a úsilí. Během výroby desky vznikaly mnohé komplikace až to dospělo k vyčerpání některých součástí. Zásadní problém nastal s unipolárními tranzistory MOSFET, které jsou značně náchylné a vzhledem k tomu, že na desce jsou použity tři kusy, tak ve skutečnosti jich deskou prošlo minimálně deset. Práce probíhala tak, že byl vytvářen obvod po obvodu a důkladně testován dostupnými měřicími přístroji.

Další komplikace nastaly, když ze dne na den přestal pracovat ovladač krokového motoru. A aby toho nebylo málo, tak převodník logických úrovní TTL na RS232 od společnosti Pololu nefungoval vůbec.

Svou nekvalitní činnost předvedly i oba termistory. Jak je uvedeno v práci, jedná se o 100 kΩ NTC termistory, u kterých odpor klesá s rostoucí teplotou. Od výrobce jsou k dispozici přepočtové tabulky, které slouží pro konverzi analogových dat na lépe čitelné hodnoty ve °C. Ale i u termistorů se náhle vyskytl problém, že se jejich rozlišovací schopnost změnila a byla nutná vlastní kalibrace. Ta byla provedena pomocí multimetru s teplotní sondou a pro různé úrovně příkonu byl v výstupním analogovém hodnotám přiřazena naměřená teplota.

K čemu se dospělo nejdál je úplná realizace hardwarové části včetně naprogramování mikrokontroléru. Ten je schopen po sériové lince přijímat příkazy pro ohřev trysky na tiskové hlavě a podložky. Dále je možné instrukcemi ovládat krokový motor v obou směrech. Samozřejmostí jsou příkazy pro zastavení ohřevu (jak trysky, tak podložky) a pro vypnutí motoru.

SEZNAM POUŽITÉ LITERATURY

- [1] ROTOLO, By Isaac Budmen and Anthony. *The book on 3D printing*. 2013. ISBN 978-1489529442.
- [2] MAGAZINE, Make. *MAKE: Ultimate guide to 3D printing*. Sebastopol, CALIF: O'Reilly Media, 2012. ISBN 978-144-9357-375.
- [3] HORVATH, Joan. *Mastering 3D Printing*. New York: Apress, 2014, 224 s. ISBN 978-1-4842-0025-4.
- [4] PRŮŠA, Josef. *Základy 3D tisku* [online]. 2014. [cit. 2015-05-08]. Dostupné z: <http://www.prusa3d.cz/kniha-zaklady-3d-tisku-josefa-prusi/>
- [5] *Easycnc: Informace o technologiích 3D tisku* [online]. 2010. [cit. 2015-03-14]. Dostupné z: <http://www.easycnc.cz/inpage/informace-o-technologiich-3d-tisku/>
- [6] CANESSA, Enrique, Carlo FONDA a Marco ZENNARO. *Low-cost 3D Printing for Science, Education & Sustainable Development*. First Edition. Trieste: ICTP The Abdus Salam International Centre for Theoretical Physics, 2013. ISBN 92-95003-48-9. Dostupné z: <http://sdu.ictp.it/3D/book.html>
- [7] *Stratasys* [online]. 2014. [cit. 2015-05-01]. Dostupné z: <http://www.stratasys.com>
- [8] ZEMČÍK, O. a J. DVOŘÁČEK. *Rozložení teplotního pole v FDM tiskové hlavě*. In: *Vysoce přesné technologie obrábění: programování moderních CNC strojů*, 2011. pp.21-28. ISBN 978-80-214-4352-5.
- [9] ZUKAS, Victoria a Jonas A. ZUKAS. *An Introduction to 3D Printing*. Sarasota, FL: First Edition Design Publishing, Inc., 2015, 109 s. ISBN 978-1622-878-97.0.
- [10] *RepRap: Fused filament fabrication* [online]. 2014. [cit. 2015-05-06]. Dostupné z: http://reprap.org/wiki/Fused_filament_fabrication
- [11] *LiveScience: What is Selective Laser Sintering?* [online]. 2013. [cit. 2015-05-10]. Dostupné z: <http://www.livescience.com/38862-selective-laser-sintering.html>
- [12] *14220: 3D tisk-metody* [online]. 2013. [cit. 2015-05-08]. Dostupné z: <http://www.14220.cz/technologie/3d-tisk-metody/>
- [13] *RepRap: Heated Bed* [online]. 2015. [cit. 2015-05-10]. Dostupné z: http://reprap.org/wiki/Heated_Bed

-
- [14] *RepRap: PLA* [online]. 2015. [cit. 2015-05-10]. Dostupné z: <http://reprap.org/wiki/PLA>
- [15] *RepRap: Extruders* [online]. 2015. [cit. 2015-05-17]. Dostupné z: <http://reprap.org/wiki/Category:Extruders>
- [16] *3D printer list: J Head MKV-B Hot-End* [online]. [cit. 2015-05-13]. Dostupné z: <https://sites.google.com/site/3dprinterlist/home/hot-ends/j-head-mkv-b-hot-end>
- [17] SALINAS, Richard. *3d printing with reprap cookbook*. S.l.: Packt Publishing Limited, 2014. ISBN 978-1-78216-988-8.
- [18] *ProtoParadigm: The Difference Between ABS and PLA for 3D Printing* [online]. 2013. [cit. 2015-05-16]. Dostupné z: <http://www.protoparadigm.com/news-updates/the-difference-between-abs-and-pla-for-3d-printing/>
- [19] *CubeX 3D Printing Blog: PLA vs. ABS printing* [online]. 2013. [cit. 2015-05-16]. Dostupné z: <http://cubex3dprinting.blogspot.cz/2013/09/pla-vs-abs-printing.html>
- [20] *CAD Studio: Autodesk 123D* [online]. 2015. [cit. 2015-05-14]. Dostupné z: <http://www.cadstudio.cz/123d>
- [21] *AUTODESK: Autodesk 123D Design App* [online]. 2015. [cit. 2015-05-14]. Dostupné z: <http://apps.123dapp.com/design/>
- [22] *OpenSCAD: About OpenSCAD* [online]. 2015. [cit. 2015-05-16]. Dostupné z: <http://www.openscad.org/about.html>
- [23] *SketchUp: Why SketchUp?* [online]. 2015. [cit. 2015-05-16]. Dostupné z: <http://www.sketchup.com>
- [24] *Slic3r: About* [online]. 2015. [cit. 2015-05-13]. Dostupné z: <http://slic3r.org/about>
- [25] *Dream 3D: MakerBot MakerWare* [online]. 2015. [cit. 2015-05-14]. Dostupné z: <http://www.dream3d.co.uk/makerware/>
- [26] *Arduino: Arduino Uno* [online]. [cit. 2015-05-16]. Dostupné z: <http://www.arduino.cc/en/main/arduinoBoardUno>
- [27] EVANS, Brian. *Beginning Arduino Programming*. New York: Apress, 2011, 272 s. ISBN 978-1-4302-3778-5.

- [28] BANZI, Massimo. *Getting started with Arduino*. 1st ed. Make:Books/O'Reilly, 2009. ISBN 978-059-6155-513.
- [29] *Arduino: Language Reference* [online]. 2015. [cit. 2015-05-17]. Dostupné z: <http://www.arduino.cc/en/Reference/HomePage>
- [30] WARREN, John-David, Josh ADAMS a Harald MOLLE. *Arduino robotics*. New York, NY: Apress, c2011, xxiv, 601 p. Technology in action series. ISBN 978-1-4302-3184-4.
- [31] MARGOLIS, Michael. *Arduino cookbook*. 1st ed. Sebastopol: O'Reilly, 2011, 631 s. ISBN 978-0-596-80247-9.
- [32] *Arm - RS series 40B/60/80 family: Instruction manual*. Stäubli Faverges, 2008, 106 s.
- [33] *CS8C Controller: Instruction manual*. Stäubli Faverges, 2008, 190 s.
- [34] *VAL3 Reference Manual: Version 6*. Stäubli Faverges, 2008, 186 s.
- [35] *Pololu: A4988 Stepper Motor Driver Carrier, Black Edition* [online]. 2015. [cit. 2015-05-17]. Dostupné z: <https://www.pololu.com/product/2128>
- [36] *RepRap Source: J-Head Mk V-B Hot End 0.35 mm for 1.75 mm filament* [online]. 2015. [cit. 2015-05-17]. Dostupné z: <https://www.reprapsource.com/m/show/6893>
- [37] *RepRap Source: Stepper Motor SM42HT47-1684A with 60cm wires* [online]. 2015. [cit. 2015-05-17]. Dostupné z: <https://www.reprapsource.com/m/show/6910>
- [38] *RepRap Source: PCB Heatbed Mk2a* [online]. 2015. [cit. 2015-05-17]. Dostupné z: <https://www.reprapsource.com/m/show/6862>
- [39] *Pololu: Pololu 23201a Serial Adapter* [online]. 2015. [cit. 2015-05-18]. Dostupné z: <https://www.pololu.com/product/126>
- [40] Fairchild. *RFP50N06, N-Channel Power MOSFETs*. Data Sheet, July 1999.
- [41] Texas Instruments. *MAX3238*. Data Sheet, June 1999 [Revised March 2004]. Dostupné z: <http://www.ti.com.cn/cn/lit/ds/symlink/max3238.pdf>
- [42] NAVRÁTIL, Pavel. *Automatizace: Vybrané statě*. Zlín, 2011. ISBN 978-80-7318-935-8.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
FDM	Fused Deposition Modeling
FFF	Fused Filament Fabrication
SLA	Stereolitography
SLS	Selective Laser Sintering
PTFE	Polytetrafluorethylen
ABS	Acrylonitrile Butadiene Styrene
PLA	Polylactic Acid
STL	Formát používaný pro 3D tisk
PWM	Pulsně šířková modulace (Pulse Width Modulation)
SCARA	Selective Compliant Articulated Robot for Assembly
DOF	Degree of freedom
MCP	Manual Control Pendant
SRS	Stäubli Robotics Studio
°C	Stupně Celsia
mm	milimetr
LED	Elektroluminiscenční dioda (Light Emitted Diode)
AC	Označení pro střídavý proud (Alternating Current)
DC	Označení pro stejnosměrný proud (Direct Current)
PID	Proporcionálně integračně derivační regulátor

SEZNAM OBRÁZKŮ

Obr. 1.1	Princip technologie FDM [10].	12
Obr. 1.2	Princip technologie SLS [12].	13
Obr. 1.3	Princip technologie SLA [5].	13
Obr. 1.4	Popis tiskové hlavy [15].	14
Obr. 1.5	Popis hrotu [16].	15
Obr. 1.6	Vyhřívaná podložka [13].	16
Obr. 1.7	Porovnání ABS a PLA u vytištěného modelu [19].	17
Obr. 2.1	Uživatelské rozhraní Autodesk 123D [20].	18
Obr. 2.2	Uživatelské rozhraní OpenSCAD [6].	19
Obr. 2.3	Uživatelské rozhraní SketchUp Make.	19
Obr. 2.4	Uživatelské rozhraní Slic3r.	20
Obr. 2.5	Uživatelské rozhraní Cura.	21
Obr. 2.6	Uživatelské rozhraní MakerWare [25].	21
Obr. 3.1	Mikrokontrolér Arduino Uno.	22
Obr. 3.2	Vývojové prostředí Arduino IDE.	24
Obr. 4.1	Robotické rameno RS40B [32].	26
Obr. 4.2	Pracovní prostor [32].	27
Obr. 4.3	Kontrolér CS8C [33].	28
Obr. 4.4	Ruční panel MCP [33].	29
Obr. 4.5	Displej MCP [33].	29
Obr. 4.6	Vývojové prostředí VAL3 Studio.	31
Obr. 5.1	Zapojení mikrokontroléru.	33
Obr. 5.2	Zleva: Hrot, topné těleso, termistor, teflonová izolace pro termistor. .	34
Obr. 5.3	Tisková hlava.	35
Obr. 5.4	Schéma celkového zapojení.	36
Obr. 5.5	Schéma zapojení desky Pololu A4988 [35].	38
Obr. 5.6	Realizovaná deska (pohled shora).	40
Obr. 5.7	Realizovaná deska (pohled zdola).	41
Obr. 6.1	Celkové propojení jednotlivých součástí.	43
Obr. 7.1	Statická charakteristika hrotu.	45
Obr. 7.2	Srovnání naměřených dat s identifikovanou soustavou.	46
Obr. 7.3	Statická charakteristika vyhřívané podložky.	46
Obr. 7.4	Srovnání naměřených dat s identifikovanou soustavou.	47
Obr. 7.5	Regulační pochody pro topné těleso hrotu.	48
Obr. 7.6	Průběhy akčních zásahů pro topné těleso hrotu.	49
Obr. 7.7	Regulační pochody pro topné vyhřívanou podložku.	49

Obr. 7.8	Průběhy akčních zásahů pro vyhřívanou podložku.	50
----------	---	----

SEZNAM TABULEK

Tab. 4.1	Rozsahy ramene [32].	27
Tab. 4.2	Přehled rychlostí ramene [32].	27
Tab. 5.1	Přehled elektrických součástí.	37
Tab. 5.2	Tabulka pro jednotlivá rozlišení kroku [35].	39
Tab. 7.1	Vypočtené parametry PID regulátorů.	48

SEZNAM PŘÍLOH

- P I. MOSFET RFP50N06 TO220AB
- P II. Zdrojový kód mikrokontroléru
- P III. DVD-ROM médium

PŘÍLOHA P I. MOSFET RFP50N06 TO220AB



RFG50N06, RFP50N06, RF1S50N06SM

Data Sheet

July 1999

File Number 3575.4

50A, 60V, 0.022 Ohm, N-Channel Power MOSFETs

These N-Channel power MOSFETs are manufactured using the MegaFET process. This process, which uses feature sizes approaching those of LSI integrated circuits gives optimum utilization of silicon, resulting in outstanding performance. They were designed for use in applications such as switching regulators, switching converters, motor drivers, and relay drivers. These transistors can be operated directly from integrated circuits.

Formerly developmental type TA49018.

Ordering Information

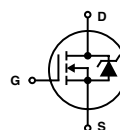
PART NUMBER	PACKAGE	BRAND
RFG50N06	TO-247	RFG50N06
RFP50N06	TO-220AB	RFP50N06
RF1S50N06SM	TO-263AB	F1S50N06

NOTE: When ordering, use the entire part number. Add the suffix, 9A, to obtain the TO-263AB variant in tape and reel, i.e. RF1S50N06SM9A.

Features

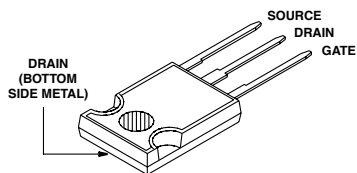
- 50A, 60V
- $r_{DS(ON)} = 0.022\Omega$
- Temperature Compensating PSPICE® Model
- Peak Current vs Pulse Width Curve
- UIS Rating Curve
- 175°C Operating Temperature

Symbol

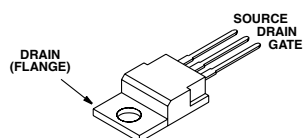


Packaging

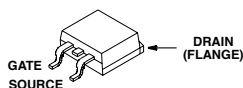
JEDEC STYLE TO-247



JEDEC TO-220AB



JEDEC TO-263AB



RFG50N06, RFP50N06, RF1S50N06SM

Absolute Maximum Ratings $T_C = 25^\circ\text{C}$, Unless Otherwise Specified

	RFG50N06, RFP50N06 RF1S50N06SM	UNITS
Drain to Source Voltage (Note 1)	V_{DSS}	V
Drain to Gate Voltage ($R_{GS} = 20\text{k}\Omega$) (Note 1)	V_{DGR}	V
Gate to Source Voltage	V_{GS}	V
Continuous Drain Current (Figure 2)	I_D	A
Pulsed Drain Current	I_{DM}	(Figure 5)
Pulsed Avalanche Rating	E_{AS}	(Figure 6, 14, 15)
Power Dissipation	P_D	W
Linear Derating Factor	0.877	$W/^\circ\text{C}$
Operating and Storage Temperature	T_J, T_{STG}	$^\circ\text{C}$
Maximum Temperature for Soldering		
Leads at 0.063in (1.6mm) from Case for 10s.	T_L	$^\circ\text{C}$
Package Body for 10s, see Techbrief 334	T_{pkg}	$^\circ\text{C}$

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

1. $T_J = 25^\circ\text{C}$ to 150°C .

Electrical Specifications $T_C = 25^\circ\text{C}$, Unless Otherwise Specified

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNITS
Drain to Source Breakdown Voltage	BV_{DSS}	$I_D = 250\mu\text{A}$, $V_{GS} = 0\text{V}$ (Figure 11)	60	-	-	V
Gate to Source Threshold Voltage	$V_{GS(TH)}$	$V_{GS} = V_{DS}$, $I_D = 250\mu\text{A}$ (Figure 10)	2	-	4	V
Zero Gate Voltage Drain Current	I_{DSS}	$V_{DS} = 60\text{V}$, $V_{GS} = 0\text{V}$	-	-	1	μA
		$T_C = 150^\circ\text{C}$	-	-	50	μA
Gate to Source Leakage Current	I_{GSS}	$V_{GS} = \pm 20\text{V}$	-	-	± 100	nA
Drain to Source On Resistance	$r_{DS(ON)}$	$I_D = 50\text{A}$, $V_{GS} = 10\text{V}$ (Figures 9)	-	-	0.022	Ω
Turn-On Time	t_{ON}	$V_{DD} = 30\text{V}$, $I_D = 50\text{A}$ $R_L = 0.6\Omega$, $V_{GS} = 10\text{V}$ $R_{GS} = 3.6\Omega$ (Figure 13)	-	-	95	ns
Turn-On Delay Time	$t_{d(ON)}$		-	12	-	ns
Rise Time	t_r		-	55	-	ns
Turn-Off Delay Time	$t_{d(OFF)}$		-	37	-	ns
Fall Time	t_f		-	13	-	ns
Turn-Off Time	t_{OFF}		-	-	75	ns
Total Gate Charge	$Q_{g(TOT)}$	$V_{GS} = 0$ to 20V	-	125	150	nC
Gate Charge at 10V	$Q_{g(10)}$	$V_{GS} = 0$ to 10V	-	67	80	nC
Threshold Gate Charge	$Q_{g(TH)}$	$V_{GS} = 0$ to 2V	-	3.7	4.5	nC
Input Capacitance	C_{ISS}	$V_{DS} = 25\text{V}$, $V_{GS} = 0\text{V}$ $f = 1\text{MHz}$	-	2020	-	pF
Output Capacitance	C_{OSS}	(Figure 12)	-	600	-	pF
Reverse Transfer Capacitance	C_{RSS}		-	200	-	pF
Thermal Resistance Junction to Case	$R_{\theta JC}$	(Figure 3)	-	-	1.14	$^\circ\text{C/W}$
Thermal Resistance Junction to Ambient	$R_{\theta JA}$	TO-247	-	-	30	$^\circ\text{C/W}$
		TO-220, TO-263	-	-	62	$^\circ\text{C/W}$

Source to Drain Diode Specifications

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNITS
Source to Drain Diode Voltage	V_{SD}	$I_{SD} = 50\text{A}$	-	-	1.5	V
Reverse Recovery Time	t_{rr}	$I_{SD} = 50\text{A}$, $dI_{SD}/dt = 100\text{A}/\mu\text{s}$	-	-	125	ns

PŘÍLOHA P II. ZDROJOVÝ KÓD MIKROKONTROLÉRU

```
#include <PID_v1.h>
#define THERMISTOR_HOTEND_PIN 0
#define THERMISTOR_HEATBED_PIN 1
#define STRLEN 16
char pole[STRLEN];
int poleIndex = 0;
int cislo;
int heaterExtruderPin = 5;
int heaterHeatbedPin = 9;
int fanPin = 3;
unsigned long time;
int data, teplota, data2, teplota2, x, tep_t, tep_d, pom;
String retezec;

// PID parametry
// HROT
//Tw = 10
//float Kp_hotend = 0.8513;
//float Ki_hotend = 0.0253;
//float Kd_hotend = 0.0292;
//Tw = 15
float Kp_hotend = 0.5676;
float Ki_hotend = 0.0169;
float Kd_hotend = 0.0194;
//Tw = 20
//float Kp_hotend = 0.4257;
//float Ki_hotend = 0.0126;
//float Kd_hotend = 0.0146;
// PODLOZKA
// Tw = 10
float Kp_heatbed = 8.6649;
float Ki_heatbed = 0.1208;
float Kd_heatbed = 1.2874;
// Tw = 20
//float Kp_heatbed = 4.3433;
//float Ki_heatbed = 0.0605;
//float Kd_heatbed = 0.6453;

double Setpoint_hotend, Input_hotend, Output_hotend, Setpoint_heatbed,
Input_heatbed, Output_heatbed;
PID PID_hotend(&Input_hotend, &Output_hotend, &Setpoint_hotend,
Kp_hotend, Ki_hotend, Kd_hotend, DIRECT);
// Input_hotend - teplota Hot Endu z termistoru
// Output_hotend - u(t) pro PWM
// Setpoint_hotend - zadana teplota

PID PID_heatbed(&Input_heatbed, &Output_heatbed, &Setpoint_heatbed,
Kp_heatbed, Ki_heatbed, Kd_heatbed, DIRECT);
// Input_heatbed - teplota Heated Bedu z termistoru
// Output_heatbed - u(t) pro PWM
// Setpoint_heatbed - zadana teplota

const int sampleRate = 100; // rychlost PID smycky [ms]
const int sampleRate2 = 100; // rychlost PID smycky [ms]

// Nastaveni pro Serial
const long serialPing = 10; // nastaveni frekvence pingu smycky
unsigned long now = 0; // Promenna pro hlidani stopy casu
```

```
unsigned long lastMessage = 0; // Promenna pro sledovani posledni
//zpravy seriove kom.
```

```
// Prepocetove tabulky z analog dat na degC
// Identifikovany termistor 100K 16/05/2015 pomoci tep. sondy a
//multimetrem METEX M-3890D
#define NUMTEMPS 20
short temptable[NUMTEMPS][2] = {
    {1, 841}, // datasheet
    {185, 258}, //
    {199, 251}, //
    {201, 235}, //
    {214, 220}, //
    {227, 203}, //
    {246, 195}, //
    {257, 183}, //
    {272, 177}, //
    {322, 158}, //
    {410, 135}, //
    {432, 122}, //
    {533, 117}, //
    {576, 101}, //
    {632, 91}, //
    {828, 60}, //
    {899, 45}, //
    {920, 41}, //
    {956, 32}, //
    {1008, 3} // datasheet
};
```

```
// Identifikovany termistor pro vyhr. desku 100K pomoci tep.
//sondy a multimetrem METEX M-3890D
#define NUMTEMPSHB 32
short temptablehb[NUMTEMPSHB][2] = {
    {1, 864}, // datasheet
    {54, 258}, // datasheet
    {107, 258}, // datasheet
    {160, 185}, // datasheet
    {213, 168}, // datasheet
    {266, 154}, // datasheet
    {319, 143}, // datasheet
    {372, 133}, // datasheet
    {425, 125}, // datasheet
    {478, 116}, // datasheet
    {532, 113}, // namereno
    {573, 110}, // namereno
    {597, 102}, // namereno
    {603, 96}, // namereno
    {632, 90}, // namereno
    {635, 89}, // namereno
    {648, 88}, // namereno
    {686, 85}, // namereno
    {695, 84}, // namereno
    {749, 75}, // namereno
    {741, 72}, // namereno
    {757, 70}, // namereno
    {776, 67}, // namereno
    {798, 63}, // namereno
    {839, 58}, // namereno
    {864, 53}, // namereno
    {901, 46}, // namereno
    {925, 41}, // namereno
    {943, 36}, // namereno
    {957, 32}, // namereno
    {986, 23}, // namereno
    {1008, 2} // datasheet
};
```

```

void setup()
{
    //Timer 0 (PWM pins 5 & 6) - Hlava
    TCCR0B = TCCR0B & 0b11111000 | 0x04;
    //Timer 1 (PWM pins 9 & 10) - Podlozka
    TCCR1B = TCCR1B & 0b11111000 | 0x05;
    //Timer 2 (PWM pins 3 & 11) - Ventilator
    TCCR2B = TCCR2B & 0b11111000 | 0x07;

    // krok motor
    pinMode(8,OUTPUT); // Enable
    pinMode(12,OUTPUT); // Step
    pinMode(13,OUTPUT); // Dir
    zapniMotor();

    Input_hotend = nacteni_teploty();
    Input_heatbed = nacteni_teploty_deska();
    Setpoint_hotend = 0; //zadana teplota
    Setpoint_heatbed = 0; //zadana teplota

    Serial.begin(9600);
    Serial.println("Seriova komunikace aktivni.");

    PID_hotend.SetMode(AUTOMATIC); // zapnuti PID smycky
    PID_hotend.SetSampleTime(sampleRate); // nastaveni rychlosti
    PID_heatbed.SetMode(AUTOMATIC); // zapnuti PID smycky
    PID_heatbed.SetSampleTime(sampleRate2); // nastaveni rychlosti

    Serial.println("");
    lastMessage = millis(); // timestamp
}

void loop()
{
    // provadet behem seriove komunikace
    while(Serial.available() > 0)
    {
        char ch = Serial.read();
        if (ch == '#') // terminator, priznak pro rozpoznani
        {
            pole[poleIndex] = 0;
            poleIndex = 0; // reset indexu pro dalsi string
            retezec = String(pole);
            poleIndex = 0;
        }
        else {
            pole[poleIndex++] = ch; // pridani znaku do pole
            break;
        }
        Serial.print(poleIndex);
        Serial.print('\n');

        // OHREV
        if (retezec[0] == 't') { // topeni trysky na tisk. hlave
            retezec.remove(0, 1);
        }
    }
}

```

```

        tep_t = retezec.toInt();
        // kontrola zadaneho rozsahu teplot
        if (tep_t > 245 || tep_t < 200)
        {
            tep_t = 0;    // nezacne topit dokud neni v rozsahu
        }
        Setpoint_hotend = tep_t;
    }
    else if (retezec[0] == 'd')    // topeni podlozky
    {
        retezec.remove(0, 1);
        tep_d = retezec.toInt();
        if (tep_d > 115 || tep_d < 90) // kontrola rozsahu
        {
            tep_d = 0; // nezacne topit dokud neni v rozsahu
        }
        Setpoint_heatbed = tep_d;
    }

    // MOTOR
    // smer do trysky
    else if (retezec[0] == 'm' && retezec[1] == 'h') {
        retezec.remove(0, 2);
        pom = retezec.toInt();           // pocet kroku
        motorH();
        vypis();
    }
    // smer od trysky
    else if (retezec[0] == 'm' && retezec[1] == 'l') {
        retezec.remove(0, 2);
        pom = retezec.toInt();           // pocet kroku
        motorL();
        vypis();
    }
}

// PRIKAZY PRO ZASTAVENI CINNOSTI
else if (retezec.length() == 5 && retezec == "stopt") {
    Setpoint_hotend = 0;
    Setpoint_heatbed = 0;
}
else if (retezec.length() == 5 && retezec == "stopm") {
    vypniMotor();
}
else if (retezec.length() == 6 && retezec == "startm") {
    zapniMotor();
}
else if (retezec.length() == 5 && retezec == "vypis")
{
    vypis();
}
else
{
    retezec = "";
}

// 0...255 prikon ventilatoru na extruderu -
// 127 default (~50%)
analogWrite(fanPin, 127);
Input_heatbed = nacteni_teploty_deska();
Input_hotend = nacteni_teploty();
PID_hotend.Compute(); // spusteni PID pro trysku
PID_heatbed.Compute(); // spusteni PID pro podlozku
// vystup z PID na 0...255 PWM Hot end
analogWrite(heaterExtruderPin, Output_hotend);
// vystup z PID na 0...255 PWM Heated bed

```

```

        analogWrite(heaterHeatbedPin, Output_heatbed);
    }
}

int motorH()
{
    PORTB |= _BV(PB5);
    //digitalWrite(13,HIGH); // smer na HIGH (material jde do hrotu)
    for(x = 0; x < pom; x++) // kroky
    {
        // 1 krok, delay urcuje rychlost otaceni
        // (obejde se vystup HIGH, pouze zapis primo na piny pro
        // rychlost
        // checksum u digitalWrite)
        PORTB |= _BV(PB4); //digitalWrite(12,HIGH); // vystup HIGH
        delayMicroseconds(100); // cekani [ms]
        PORTB &= ~_BV(PB4); // vystup LOW
        //digitalWrite(12,LOW); // vystup LOW
        delayMicroseconds(100); // cekani [ms]
    }
}

int motorL()
{
    PORTB &= ~_BV(PB5);
    // smer na LOW (material jde od hrotu zpatky ven)
    //digitalWrite(13,LOW);
    for(x = 0; x < pom; x++) // kroky
    {
        // 1 krok, delay urcuje rychlost otaceni
        PORTB |= _BV(PB4);
        //digitalWrite(12,HIGH); // vystup HIGH
        delayMicroseconds(500); // cekani [ms]
        PORTB &= ~_BV(PB4);
        //digitalWrite(12,LOW); // vystup LOW
        delayMicroseconds(500); // cekani [ms]
    }
}

int zapniMotor()
{
    digitalWrite(8,LOW);
}
int vypniMotor()
{
    digitalWrite(8,HIGH);
}

int nacteni_teploty()
{
    int th_data = analogRead(THERMISTOR_HOTEND_PIN);
    int aktual_teplota = 0;
    byte i;
    for (i=1; i<NUMTEMPS; i++)
    {
        if (temptable[i][0] > th_data)
        {
            int real_teplota = temptable[i-1][1] + (th_data -
            temptable[i-1][0]) * (temptable[i][1] -
            temptable[i-1][1]) / (temptable[i][0] -
            temptable[i-1][0]);
            if (real_teplota > 255)

```

```

        real_teplota = 255;
        aktual_teplota = real_teplota;
        break;
    }
}
// Podminka 0 degC
if (i == NUMTEMPS)
    aktual_teplota = 0;
return aktual_teplota;
}

```

```

int nacteni_teploty_deska()
{
    int th_data = analogRead(THERMISTOR_HEATBED_PIN);
    int aktual_teplota = 0;
    byte i;
    for (i=1; i<NUMTEMPSHB; i++)
    {
        if (temptablehb[i][0] > th_data)
        {
            int real_teplota = temptablehb[i-1][1] + (th_data -
            temptablehb[i-1][0]) * (temptablehb[i][1] -
            temptablehb[i-1][1]) /
            (temptablehb[i][0] - temptablehb[i-1][0]);
            if (real_teplota > 255)
                real_teplota = 255;
            aktual_teplota = real_teplota;
            break;
        }
    }
    // Podminka 0 degC
    if (i == NUMTEMPSHB)
        aktual_teplota = 0;
    return aktual_teplota;
}

```

```

void vypis() {
    Serial.print("Zadana teplota hrotu = ");
    Serial.print(Setpoint_hotend);
    Serial.print(" Aktualni teplota = ");
    Serial.print(Input_hotend);
    Serial.print(" Vystup PWM = ");
    Serial.print(Output_hotend);
    Serial.print("\n");
    Serial.print("Zadana teplota desky = ");
    Serial.print(Setpoint_heatbed);
    Serial.print(" Aktualni teplota desky = ");
    Serial.print(Input_heatbed);
    Serial.print(" Vystup PWM = ");
    Serial.print(Output_heatbed);
    Serial.print("\n");
}

```

PŘÍLOHA P III. DVD-ROM MÉDIUM

Na přiloženém médiu se nachází textová část práce ve formátu PDF a zdrojovém jazyce \LaTeX . Dále jsou zde uloženy veškeré naměřené data, Matlab skripty pro vyhodnocení a veškeré zdrojové kódy pro mikrokontrolér Arduino.