

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

DOTYKOVÝ PROFILOMĚR

Bc. Martin Knížek

Diplomová práce
2017

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2016/2017

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Martin Knížek**
Osobní číslo: **I15180**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Řízení procesů**
Název tématu: **Dotykový profiloměr**
Zadávací katedra: **Katedra řízení procesů**

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

Cílem diplomové práce je návrh a realizace dotykového profiloměru. Snímání profilu bude prováděno ve dvou souřadných osách. K určení polohy doteku budou použity snímače translačního a rotačního pohybu (například optické nebo magnetické). Lineární posuv snímače (hrotu) bude zajištěn například pomocí krokového motoru. Konstantní velikost přtlaku měřícího doteku bude realizována pomocí posuvu pneumatického válce. Nadřazený řídicí systém bude součástí návrhu a bude umožňovat přenos měřených dat do PC přes zvolený typ komunikačního rozhraní (např. USB port). Pro nadřazený systém-osobní počítač bude realizován vizualizační a ovládací software ve zvoleném programovacím jazyce. Software bude umožňovat jak vizualizaci snímaného profilu a následnou analýzu naměřených dat. Součástí diplomové práce bude rešerše řešené problematiky, popis návrhu a funkce jednotlivých částí konstrukčního řešení zařízení, kompletní výrobní dokumentace, včetně okomentovaných zdrojových kódů vzniklého software.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

MATOUŠEK, D. Práce s mikrokontroléry ATMEL AVR - 3. díl - edice uP a praxe 2. vydání, BEN - technická literatura, 2006, ISBN 80-7300-209-4

ZÁHLAVA, V. Návrh a konstrukce DPS, BEN-technická literatura, 2010, ISBN 978-80-7300-266-4

HARNA, Z. Přesná mechanika, VUT, Brno, 1996, ISBN 8021407948.

Vedoucí diplomové práce:

Ing. Libor Havlíček, Ph.D.

Katedra řízení procesů

Datum zadání diplomové práce:

17. října 2016

Termín odevzdání diplomové práce:

17. května 2017



Ing. Zdeněk Němec, Ph.D.
děkan



L.S.



Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2016

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne

Martin Knížek

ANOTACE

Tato práce se zabývá problematikou kontaktního měření profilů. Byl proveden návrh a konstrukce dotykového profiloměru. Profiloměr provádí snímání ve 2 souřadných osách, pro snímání polohy jsou použity optické inkrementální snímače. Posuv v první ose zajišťuje krokový motor, jehož rotační pohyb je převeden na translační. V druhé ose pohyb zajišťuje pneumatický válec. Nadřazený řídicí systém je realizován pomocí mikročipu Atmel, který je součástí vývojového kitu Arduino. Tento systém přenáší naměřená data do PC přes USB port.

KLÍČOVÁ SLOVA

profiloměr, krokový motor, pneumatický válec, snímač polohy

TITLE

ROUGHNESS MEASURING INSTRUMENT

ANNOTATION

This work deals with Roughness Measuring Instruments. Roughness Measuring Instrument was designed and constructed. Device performs scanning in two coordinate axes, optical incremental encoders are used for sensing the position of head. Movement of first axis is provided by stepper motor, his rotational movement is converted to translational. The movement of second axis is provided by pneumatic cylinder. The control system is realized via microchip Atmel, which is a part of the Arduino development kit. This system transmits measurement data to PC via the USB port.

KEYWORDS

Roughness Measuring Instrument, Stepper motor, Pneumatic cylinder, Optical incremental encoders

OBSAH

	SEZNAM ZKRATEK A ZNAČEK	11
	SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ.....	13
	SEZNAM ILUSTRACÍ	14
	SEZNAM TABULEK.....	16
	ÚVOD	17
1	ÚVOD DO PROFILOMĚŘŮ	18
1.1	REŠERŠE LITERATURY	18
2	KONSTRUKČNÍ PRVKY PROFILOMĚŘU	19
2.1	OPTICKÝ INKREMENTÁLNÍ SNÍMAČ.....	19
2.1.1	Princip optického inkrementálního snímače.....	19
2.2	KROKOVÝ MOTOR	20
2.2.1	Hybridní krokový motor	21
2.3	DRIVER KROKOVÉHO MOTORU	23
2.4	SENZORY RELATIVNÍHO TLAKU.....	25
2.4.1	Princip monokrystalického snímače tlaku	25
2.4.2	Zapojení senzorů.....	26
2.4.3	Kalibrace senzorů.....	27
2.5	PNEUMATICKÉ PRVKY	28
2.5.1	Pneumatický válec	28
2.5.2	Ventily.....	29
2.5.3	Kompresor.....	30
2.5.4	Ostatní pneumatické prvky	30
2.6	LINEÁRNÍ VEDENÍ.....	30
2.6.1	Podepřené tyče	31
2.6.2	Prizmatické vedení	31
2.6.3	Kuličkový šroub.....	32
2.7	ZAPOJENÍ PNEUMATICKÉ ČÁSTI.....	32
3	NADŘAZENÝ ŘÍDICÍ SYSTÉM – HARDWARE	33
3.1	ČÁSTI MIKROPOČÍTAČE	34
3.1.1	Základní části mikropočítače	34
3.1.2	USART.....	35
3.1.3	SPI.....	36

3.1.4	Sběrnice I ² C	37
3.1.5	Čítač/časovač	37
3.1.6	Digitální vstup/výstup mikropočítače	38
3.2	ATMEGA 2560.....	38
3.3	ARDUINO	39
3.4	LCD DISPLEJ 16×4	40
3.5	DESKA SPÍNACÍCH RELÉ	42
3.6	TLAČÍTKA.....	43
3.7	SPÍNANÝ ZDROJ.....	45
3.8	ZAPOJENÍ ELEKTRONICKÉ ČÁSTI	47
4	NADŘAZENÝ ŘÍDICÍ SYSTÉM – SOFTWARE	48
4.1	ARDUINO IDE.....	49
4.2	PROGRAM V MIKROPOČÍTAČI	50
4.2.1	Přerušení programu	51
4.2.2	Inicializační sekvence	52
4.2.3	Obsluha periférií	53
4.2.4	Nastavení parametrů	54
4.2.5	Měřicí sekvence	55
4.2.6	Obsluha přístroje	56
5.1	JAVA	58
5.1.1	Základní vlastnosti Javy.....	58
5.2	NETBEANS.....	59
5.3	APLIKACE PRO VYHODNOCENÍ.....	59
5.3.1	Karta Data	60
5.3.2	Metody pro načtení dat	61
5.3.3	Karta Vizualizace	62
5.3.4	Karta Komunikace	63
6	KONSTRUKCE.....	64
6.1	AUTODESK INVENTOR.....	65
6.2	VIZUALIZACE	65
6.2.1	Posuv v horizontální ose	66
6.2.2	Posuv ve vertikální ose	67
6.3	VLASTNÍ KONSTRUKCE.....	67
6.3.1	Sváření v ochranné atmosféře	68

6.3.2	Soustružení.....	68
6.3.3	Uvedení do provozu.....	69
7	VYHODNOCENÍ.....	70
7.1	KOMPENZACE MÍSTA DOTEKU	70
7.2	MĚŘENÍ	73
7.2.2	Vliv tlaku na měření.....	74
7.3	VÝSLEDNÉ PARAMETRY	76
8	ZÁVĚR	77
	POUŽITÁ LITERATURA	79
	PŘÍLOHY	81

SEZNAM ZKRATEK A ZNAČEK

ACK	acknowledge
ASCII	american standard code for information interchange
CLK	clock
CNC	computer numerical control
COM	communication port
A/D	analog/digital
CS	chip select
DIR	direction
E	enable
EEPROM	electrically erasable programmable read-only memory
EPROM	erasable programmable read-only memory
FET	field effect transistors
FLASH	paměť typu FLASH
GND	ground
GNU	GNU's Not Unix
GSM	groupe spécial mobile
H	high
HTML	hypertext markup language
HW	hardware
I ² C	inter-integrated circuit
IDE	integrated development environment
JVM	Java virtual machine
L	low
LCD	liquid crystal display
LED	light emitter diode
MISO	master-input slave-output
MOSI	master-output slave-input
NC	not connected
PC	personal computer
PHP	hypertext preprocessor
PWM	pulse width modulation
R/W	read/write

ROM	read only memory
RISC	reduced instruction set computing
Rx	receive
SCL	synchronous clock
SCLK	synchronous clock
SDA	synchronous data
SPI	serial peripheral interface
SS	slave select
SW	software
TTL	transistor to transistor logic
Tx	transmit
USART	universal synchronous/asynchronous receiver/transmitter
USB	universal serial bus

SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ

α	úhel nakloněné roviny, °
φ	úhel natočení, °
I	elektrický proud, A
f	frekvence, Hz
n	počet zubů
n_f	počet fází
n_p	počet pólů
P	výkon, W
p	tlak, bar
R	elektrický odpor, Ω
S	plocha, m ²
t	čas, s
U	elektrické napětí, V
x	poloha v ose x, mm
y	poloha v ose y, mm

SEZNAM ILUSTRACÍ

Obr. 2.1 – Princip optického inkrementálního snímače.....	20
Obr. 2.2 – Konstrukce hybridního krokového motoru.....	21
Obr. 2.3 – Stator hybridního krokového motoru	22
Obr. 2.4 – Kroky hybridního krokového motoru.....	22
Obr. 2.5 – Driver s TB6600	24
Obr. 2.6 – Snímač tlaku Honeywell.....	25
Obr. 2.7 – Monokrystalický tenzometr	25
Obr. 2.8 – Schéma zapojení senzorů tlaku.....	26
Obr. 2.9 – Výstup senzoru	27
Obr. 2.10 – Pneumatický válec	28
Obr. 2.11 – Ventilový blok	29
Obr. 2.12 – Otevřený vozík s ložiskem a podepřená tyč	31
Obr. 2.13 – Vozík a kolejnice prizmatického vedení	32
Obr. 2.14 – Kuličkový šroub	32
Obr. 2.15 – Schéma pneumatické části.....	33
Obr. 3.1 – Arduino Mega 2560.....	40
Obr. 3.2 – LCD HD44780 16×4	41
Obr. 3.3 – Zapojení LCD	42
Obr. 3.4 – Deska spínacích relé	43
Obr. 3.5 – Rozložení klávesnice	43
Obr. 3.6 – Připojení tlačítka.....	44
Obr. 3.7 – Blokové schéma spínaného zdroje	45
Obr. 3.8 – Použité spínané zdroje	47
Obr. 4.1 – Algoritmus hlavního programu	50
Obr. 4.2 – Algoritmus inicializační sekvence.....	53
Obr. 4.3 – Algoritmus nastavení parametrů měření.....	55
Obr. 4.4 – Algoritmus měřicí sekvence	56
Obr. 4.5 – Ovládací panel	57
Obr. 5.1 – Karta Data.....	61
Obr. 5.2 – Přijatý řetězec	61
Obr. 5.3 – Rozdělení řetězce.....	62
Obr. 5.4 – Karta Vizualizace	63

Obr. 5.5 – Karta Komunikace	64
Obr. 6.1 – Pohled na menší rám.....	66
Obr. 6.2 – Pohled na hlavní rám – zvýrazněný posuv osy y.....	67
Obr. 7.1 – Kompenzace dotyku	70
Obr. 7.2 – Měření rozptylu	73
Obr. 7.3 – Měření s přítlakem 1,8 baru.....	74
Obr. 7.4 – Měření s přítlakem 1,6 baru.....	75

SEZNAM TABULEK

Tab. 2.1 – Parametry snímačů	19
Tab. 2.2 – Konektor snímače	20
Tab. 2.3 – Parametry krokového motoru	23
Tab. 2.4 – Zapojení konektoru ML10 pro driver	24
Tab. 2.5 – Zapojení konektoru ML10 na desce senzorů.....	26
Tab. 2.5 – Zapojení konektoru ML10 na desce senzorů – pokračování.....	27
Tab. 2.6 – Výsledky kalibrace	28
Tab. 2.7 – Parametry ventilů.....	29
Tab. 2.8 – Ovládání dvoucestného ventilu	30
Tab. 2.9 – Parametry kompresoru.....	30
Tab. 3.1 – Zapojení digitálních pinů ATMegy 2560	39
Tab. 3.2 – Zapojení konektoru ML10 na desce LCD	41
Tab. 3.3 – Zapojení konektoru ML10 na Relé desce.....	42
Tab. 3.4 – Zapojení konektoru ML10 pro klávesnici	43
Tab. 3.4 – Zapojení konektoru ML10 pro klávesnici – pokračování	44
Tab. 3.5 – Zapojení konektoru ML10 pro koncové mikrospínače	44
Tab. 3.6 – Použité spínané zdroje	46
Tab. 7.1 – Výsledky měření.....	76

ÚVOD

Obrábění se využívá k výrobě různých částí strojů a přístrojů, pomocí obrábění se dosahuje lepší přesnosti požadovaných rozměrů, tvaru a kvality povrchu. V dnešní době je kladen stále větší důraz na přesnost výsledných produktů. Výsledná přesnost má vliv na funkci zařízení, složeného z těchto dílů a také na jeho životnost. Je tedy třeba tvar a rozměry výrobků měřit a ověřit tak zda bylo dosaženo požadované kvality.

Dle účelu měření existují různé měřicí přístroje. Méně přesná měření se provádí například pomocí posuvného měřidla. Vyšších přesností už dosahují třmenové mikrometry nebo páčkové úchylkoměry. Tvar a rozměry také lze měřit opticky, ale zde vzniká problém se specifickými požadavky na typ povrchu obrobku. Například pro nerez je problém s odrazovými plochami a měření pomocí laseru či kamery je zde nepoužitelné. Jako lepší řešení se jeví použití kontaktního měření.

Přesná kontaktní měření je možno provést například třísouřadnicovým měřicím přístrojem, ale ten je určen spíše pro laboratoře a nehodí se pro velké provozy. Ve velkých provozech jsou nejčastěji využívány profiloměry. Profiloměry dosahují rozlišení až 0,1 μm , což je dostačující přesnost pro obráběné výrobky. Tato práce se zabývá právě návrhem a konstrukcí specifického profiloměru.

Ke konstrukci vlastního profiloměru mě zavedla praxe v kalibrační laboratoři, v několika společnostech jsem se dostal k problémům s nedostatečným rozsahem měření konvenčně vyráběných profiloměrů a také jejich horší přenositelnost. Rozhodl jsem se tedy pro konstrukci vlastního profiloměru.

Snímání polohy ve dvou měřených osách bude zajištěno optickými inkrementálními snímači. Jejich výstup bude zpracován pomocí vývojového kitu Arduino. Tento řídicí systém bude také zajišťovat řízení krokového motoru a ovládání ventilů pneumatické části přístroje. Dále bude posílat naměřená data do osobního počítače, a to skrze USB port. V počítači budou naměřená data zpracována a zobrazeny výsledky měření. Ke zpracování dat bude sloužit software napsaný v jazyce Java.

Práce se bude velkou částí zabývat také návrhem a samotnou konstrukcí přístroje. Ten bude zkonstruován ze zakoupených dílů, jako jsou motory, elektronické součástky, pneumatické prvky a snímače, a také z dílů, které budou vyrobeny v menší dílně. Většina dílů je běžně dostupných, některé je však třeba objednat u specializovaných dodavatelů.

Hlavními požadavky na přístroj jsou stálá přesnost při opakovaném měření, spolehlivost, lehká manipulovatelnost a kompaktní rozměry.

1 ÚVOD DO PROFILOMĚRŮ

Profiloměry spadají do kategorie souřadnicových měřicích přístrojů. Ty se využívají k měření tvaru plochy. Souřadnicové měřicí přístroje se v průmyslové výrobě používají více jak 50 let. I když technologie snímačů jde stále dopředu, princip těchto přístrojů se příliš nemění. Přístroj je zpravidla tvořen rameny, počet ramen udává počet snímaných os, obvyklý počet ramen jsou dvě (osy x a y) nebo tři (osy x, y a z).

Existuje několik základních konstrukcí souřadnicových měřicích přístrojů a to: sloupové, konzolové, portálové, mostové a přístroje s horizontálním nebo kloubovým ramenem. Každý typ konstrukce je vhodný pro jiný druh měření a přístroj se tedy volí dle konkrétního případu.

Profiloměr kterým se zabývá tato práce, spadá do kategorie přístroje s horizontálním ramenem. Je zde však jeden rozdíl oproti běžným přístrojům s horizontálním ramenem a to ten, že vertikální posuv je veden pomocí čtyř podepřených tyčí, a to z důvodu větší stability horizontálního ramena.

1.1 REŠERŠE LITERATURY

Diplomová práce Bc. Jana Opaty se zabývá metodikou měření tvarů a deformací ploch. Tato práce popisuje jak kontaktní, tak i bezkontaktní metody měření tvarů. Jsou zde popsány různé druhy souřadnicových měřicích přístrojů a jejich výhody a nevýhody. Práce se také zabývá rekonstrukcí nasnímané plochy. Rekonstrukce může být provedena pomocí základních geometrických tvarů nebo v případě složitějších útvarů s použitím po částech spojitých ploch (Opat, 2015).

Dalším podkladem zabývajícím se souřadnicovými měřicími přístroji jsou učební skripta z katedry výrobních systémů a automatizace Technické Univerzity Liberec. Obsahem těchto skript jsou různé typy konstrukcí souřadnicových snímačů, principy odměřovacích systémů, druhy snímacích hlavic, možnosti kalibrace těchto přístrojů a způsoby zpracování naměřených dat (Keller, 2013).

Podobnou tematikou se také zabývá bakalářská práce Mateje Harčaríka z fakulty strojního inženýrství Vysokého učení technického v Brně. Náplní této práce je hodnocení textury povrchu a jde opět o srovnání kontaktní a bezkontaktní metody měření. I když je měření textury mnohem přesnější než měření tvaru profilu, zpracování dat a metody měření jsou velice podobné (Harčarík, 2014).

2 KONSTRUKČNÍ PRVKY PROFILOMĚRU

V této kapitole budou popsány základní součásti konstruovaného profiloměru. U každé součásti bude vysvětlena její funkce a popsány její vstupy/výstupy.

2.1 OPTICKÝ INKREMENTÁLNÍ SNÍMAČ

Charakteristické vlastnosti optických inkrementálních snímačů jsou jejich rozlišovací schopnosti, malé rozměry a nízká hmotnost. Snímače jsou dvojího typu, dle pohybu, který snímají, pohyb rotační nebo translační. V této práci budou použity pouze snímače translačního pohybu, typu JCXE od výrobce TIGRE.

Optický inkrementální snímač bude sloužit pro snímání polohy jak v ose X, tak v ose Y. Jeho výstupem jsou vzájemně posunuté TTL signály, které budou dále vyhodnoceny pomocí mikročipu ATmega 2560. Napájecí a výstupní vodiče jsou přivedeny na standardní konektor COM 9. V tab. 2.1 jsou uvedeny parametry použitých snímačů.

Tab. 2.1 – Parametry snímačů

Osa snímače	Rozsah	Rozlišení
X	0 ÷ 100 mm	5 μm
Y	0 ÷ 250 mm	5 μm

2.1.1 Princip optického inkrementálního snímače

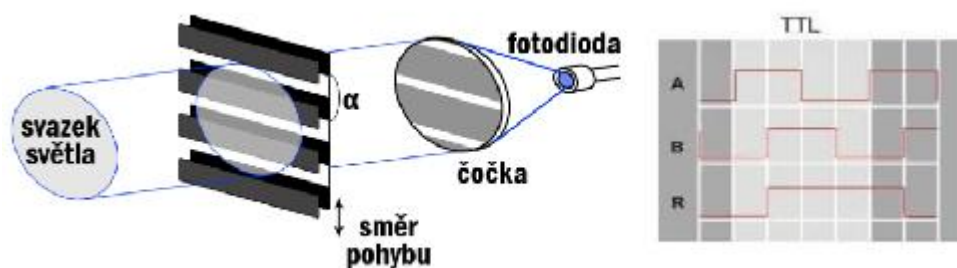
Optický inkrementální snímač se skládá zpravidla ze zdroje světla, čočky, stínítka a fotocitlivého prvku (například fotodiody). Základním principem snímače je clonění světelného toku mezi zdrojem světla a fotocitlivým prvkem. Stínítko (pravítko) je rozděleno na propustné a nepropustné části. Posuv vede k přerušení a opětovnému obnovení světelného spojení mezi zdrojem a fotocitlivým prvkem. Tato událost je převedena na impulsy, jak ukazuje obr. 2.1.

Použitý optický snímač má 3 výstupní kanály A, B a R. Výstupy A, B jsou vůči sobě posunuty o ¼ kroku senzoru. Tento posun indikuje směr posuvu snímače. Signál R je generován při přejetí referenční značky (Fejt, 2012; Rajsén, 2017).

V následující tab. 2.2 je zapojení konektoru COM 9.

Tab. 2.2 – Konektor snímače

Zapojení COM 9	
Číslo pinu	Signál
1	NC
2	0 V
3	NC
4	GND
5	NC
6	A
7	5 V
8	B
9	R



Obr. 2.1 – Princip optického inkrementálního snímače

2.2 KROKOVÝ MOTOR

Krokový motor vykonává pohyb po krocích, tyto kroky se spojují v otáčivý pohyb. Motor se stejně jako ostatní elektrické motory skládá ze statoru a rotoru. Stator se většinou skládá z ocelových lamel s drážkami, ve kterých jsou uložena měděná vinutí. Rotor bývá tvořen železným jádrem (variabilní reluktance), permanentním magnetem nebo kombinací permanentního magnetu vloženého do laminovaného železného jádra (tzv. hybridní krokové motory).

Krokový motor s železným jádrem má rotor vyrobený z měkkého železa s řadou zubů. Po přivedení proudu na vinutí statoru se aktivuje magnetické pole v jednom páru pólů, pokud je proud k prvnímu páru pólů přerušen a je zapojen druhý pár pólů rotor se pootočí o jeden

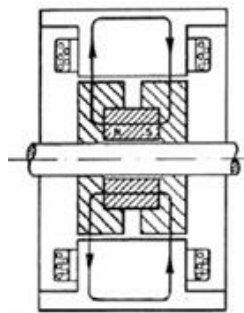
krok. Typ motoru s variabilní reluktancí dosahuje vysokých otáček při malém točivém momentu. V dnešní době není příliš využíván.

Motor s permanentními magnety má rotor tvořený permanentním magnetem bez zubů. Při správném napájení a střídání cívek statoru se bude také otáčet o krocích. Oproti motoru s železným jádrem je levnější a dosahuje většího točivého momentu je tedy využíván například v tiskárnách, počítačích, klimatizacích atd.

Hybridní krokový motor byl zvolen jako nejvhodnější pro profiloměr a je tedy podrobněji popsán v následující kapitole.

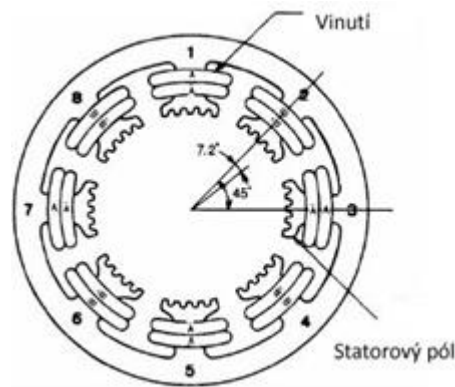
2.2.1 Hybridní krokový motor

Hybridní krokový motor (viz obr. 2.2) je kombinací motoru se železným jádrem a motoru s permanentními magnety. Dosahuje tedy relativně vysokých otáček při zachování značného točivého momentu.



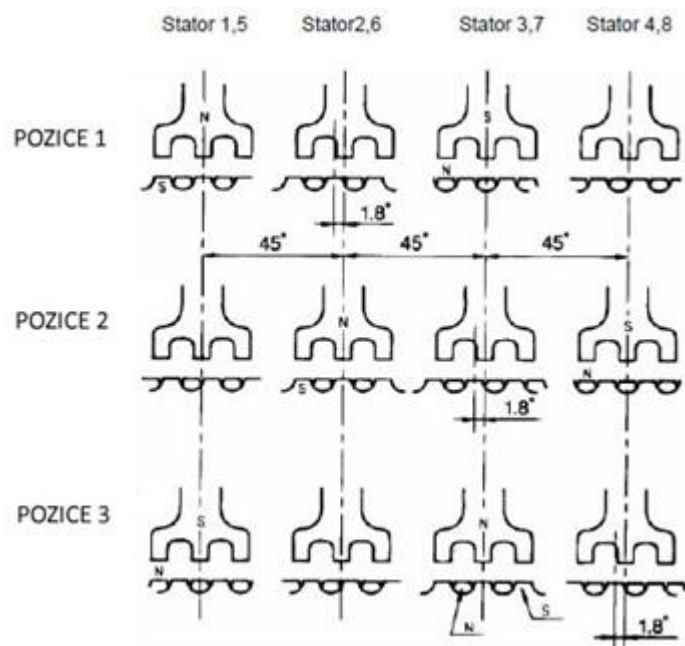
Obr. 2.2 – Konstrukce hybridního krokového motoru

Stator (viz obr. 2.3) hybridního krokového motoru je obvykle složen z několika částí. Bývá tvořen osmi statorovými póly, které jsou osazeny zuby. Na každém pólu jsou dvě vinutí, to každému pólu umožňuje být severním nebo jižním pólem na základě směru protékajícího proudu (Pohonnatechnika.cz, 2007).



Obr. 2.3 – Stator hybridního krokového motoru

Rotor hybridního krokového motoru je složen ze dvou laminátových kol, která jsou podobná ozubeným kolům. Uvnitř těchto dvou kol je uložen neodymový magnet. Obě části jsou vzájemně pootočené tak, že zuby na jedné straně odpovídají zubům na straně druhé a tvoří vzájemně opačné póly (Pohonnatechnika.cz, 2007).



Obr. 2.4 – Kroky hybridního krokového motoru

V pozici 1 jsou zuby rotoru přímo proti zubům statoru (viz obr. 2.4, Stator 1,5), které jsou severním pólem. Zuby statoru 3,7 jsou jižním pólem a jsou ve středu oproti severnímu pólu rotoru.

V pozici 2 jsou napájeny zuby 2,6 a 4,8 to způsobí posun rotoru o $\frac{1}{2}$ zubu, což znamená výsledné pootočení o $1,8^\circ$. Takto rotující magnetické pole statoru způsobí, že rotor se přesouvá mezi stabilními pozicemi.

V pozici 3 se opakuje celý proces s tím rozdílem, že póly rotoru a statoru jsou nyní obrácené.

Úhel φ jednoho kroku je vyjádřen vztahem

$$\varphi = \frac{360}{z \cdot n_f \cdot n_p}, \quad (2.1)$$

kde z je počet zubů,

n_f – počet fází,

n_p – počet pólů rotoru.

V této práci je použit motor s padesáti zuby ($z = 50$), dvěma fázemi ($n_f = 2$) a dvěma póly rotoru ($n_p = 2$). Po dosazení do vztahu (2.1) vychází, že velikost jednoho kroku je $1,8^\circ$. Jedna otáčka je 360° takže na jednu otáčku máme 200 kroků. Použitý motor nese označení Nema23 LDO-57STH51 a jeho parametry jsou uvedeny v tab. 2.3 (ldomotors.com, 2017).

Tab. 2.3 – Parametry krokového motoru

Parametr	Hodnota
Moment	$1 \text{ N}\cdot\text{m}^{-1}$
Maximální proud	2,8 A
Hmotnost	0,68 kg
Délka těla	51 mm
Počet fází	2

2.3 DRIVER KROKOVÉHO MOTORU

Jak vyplývá z předchozí kapitoly, pro řízení krokového motoru jsou třeba přesně generované impulzy. O generování těchto impulzů se stará tzv. driver. Pro tuto práci bude použit Driver s integrovaným obvodem Toshiba TB6600 (viz obr. 2.5). Tento driver je přímo navržen pro řízení dvoufázových krokových motorů. Driver využívá standardní 5 V TTL logiku.

Driver má několik vstupů prvním z nich je vstup ENABLE, pomocí toho vstupu se ovládá spuštění a vypnutí činnosti driveru. Dalším vstupem je DIR tento vstup slouží k určení

směru rotace motoru. Posledním vstupem je CLK, každý impuls přivedený na tento vstup způsobí pootočení motoru o jeden krok.

Výstupy driveru jsou A+, A-, B+ a B-, což jsou fáze motoru. Na ně jsou poté přiváděny budící signály motoru, které driver generuje na základě signálu CLK.

Důležitou funkcí driveru je možnost použití tzv. mikrokroků, kde se využívá současného napájení obou fází, ale různými proudy a tím se dosáhne dalších poloh. Kroky lze dělit až na 32 mikrokroků, což nám dává na jednu otáčku celkem 6400 mikrokroků.

V poslední řadě je driver také třeba napájet stejnosměrným napětím 12 V ÷ 24 V, maximální proud, který driver umožňuje, jsou 4 A. Driver je s nadřazeným řídicím systémem propojen konektorem ML10, jehož zapojení je uvedeno v tab. 2.4.



Obr. 2.5 – Driver s TB6600

Tab. 2.4 – Zapojení konektoru ML10 pro driver

Číslo pinu konektoru	Označení signálu
1	5 V
2	GND
3	NC
4	NC
5	NC
6	PUL
7	NC
8	DIR
9	NC
10	ENABLE

2.4 SENZORY RELATIVNÍHO TLAKU

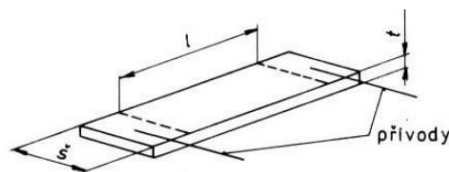
Jelikož budou v zařízení pneumatické prvky, je třeba snímat tlak v systému. Hlavním důvodem použití senzorů je zabezpečení minimálního tlaku v systému, není tedy vyžadována vysoká přesnost. Proto bude použit senzor relativního tlaku Honeywell ABPDANN010BG2A3 (viz obr. 2.6). Tento snímač má rozsah 0 bar ÷ 10 bar. Jeho výstupem je dvou bajtové slovo, vysílané po sériové sběrnici I²C.



Obr. 2.6 – Snímač tlaku Honeywell

2.4.1 Princip monokrystalického snímače tlaku

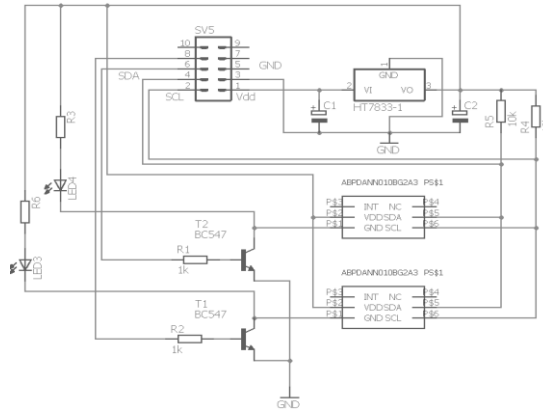
Základem tohoto typu snímačů je monokrystalický lepený tenzometr (viz obr. 2.7), je to destička z monokrystalu křemíku. Ten je tvořen páskem vyříznutým z křemíku znečištěného difuzí jiného materiálu. Vlivem mechanického namáhání krystalické mřížky je ovlivněna pohyblivost nosičů náboje a tím i odpor tenzometru. Tloušťka pásku je řádově v setinách mm a délka v jednotkách mm. Nevýhodou těchto snímačů je významná teplotní závislost.



Obr. 2.7 – Monokrystalický tenzometr

2.4.2 Zapojení senzorů

Při objednání těchto specifických senzorů od firmy Honeywell, nastal problém při konfiguraci. Český dodavatel je dodává pouze v jediné konfiguraci, a to pro logiku 3,3 V a s jedinou HW I²C adresou. Bylo tedy třeba navrhnout řešení tohoto problému.



Obr. 2.8 – Schéma zapojení senzorů tlaku

Na obr. 2.8 je zobrazeno řešení problému se senzory. Aby bylo možno komunikovat po sběrnici I2C ve 3,3 V logice, je třeba nejdříve vypnout vnitřní pull-up rezistory ATMegy 2560. Po jejich odpojení lze použít externí pull-up rezistory, konkrétně R4 a R5, které jsou připojeny na výstup pevného stabilizátoru napětí HT7833. Ten upravuje vstupní napětí, v tomto případě 5 V, na 3,3 V. Tímto zapojením je dosaženo toho, že vysoká úroveň signálu na sběrnici je nyní 3,3 V.

Dalším problémem byly shodné HW adresy senzorů. Proto je třeba mít zapojený vždy jen jeden senzor, to je realizováno pomocí bipolárních tranzistorů BC547, na schématu označených jako T1 a T2. Po přivedení řídicího signálu na bázi jednoho z tranzistorů je propojen pin čidla na zemnicí vodič a tím je uzavřen elektrický obvod. Softwarově je ošetřeno, aby bylo zapnuto vždy jen jedno čidlo, navíc je mezi přepnutí nutno vložit zpoždění 100 ms, aby se právě zapojený senzor stihl inicializovat a fungoval tak správně. V následující tab. 2.5 je uvedeno zapojení konektoru ML10.

Tab. 2.5 – Zapojení konektoru ML10 na desce senzorů

Číslo pinu konektoru	Signál
1	VDD
2	SCL
3	GND

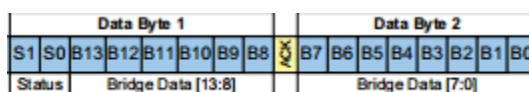
Tab. 2.5 – Zapojení konektoru ML10 na desce senzorů – pokračování

Číslo pinu konektoru	Signál
4	SDA
5	NC
6	T1
7	NC
8	T2
9	NC
10	NC

2.4.3 Kalibrace senzorů

Výstupem senzoru jsou dva bajty, které dohromady tvoří 16bitové slovo (viz obr. 2.9). V tomto slově je hodnota vnitřního převodníku senzoru a také dva stavové bity. Stavové bity udávají 4 stavy senzoru. První stav je normální, který signalizuje, že odesílaná data jsou v pořádku. Druhý stav slouží pro indikaci příkazového režimu senzoru, tento režim slouží pro prvotní nastavení, které provádí výrobce. Třetí stav nastane ve chvíli, kdy je započato nové čtení, ale výstup z vnitřního převodníku je stále stejný. A poslední stav indikuje diagnostický stav senzoru, v tomto stavu je možno zapisovat do vnitřní EEPROM paměti senzoru.

Nejdůležitější informace je obsažena ve spodních čtrnácti bytech výstupního 16bitového slova. Těchto 14bitů vyjadřuje výstup z vnitřního převodníku senzoru. Výstup může nabývat hodnot 0 (výstup na 0 %, minimální tlak) až 16383 (výstup na 100 %, maximální tlak). Pro srozumitelnější práci s touto hodnotou tlaku je třeba ji převést do nějakých standartních jednotek. V pneumatické oblasti se s oblibou používá jednotka bar



Obr. 2.9 – Výstup senzoru

(odpovídá 100 kPa), proto bude použita (honeywell.com, 2017).

Aby byl převod na jednotky barů možný, bylo použito etalonové měřidlo tlaku DTM10, které měří s přesností na 0,001 baru. Dle naměřených hodnot v celém rozsahu senzorů byly sestaveny kalibrační přímky. Pomocí rovnic kalibračních přímek se pak výstupní hodnota převodníku převádí na jednotky bar.

Po kalibraci bylo provedeno kontrolní měření, aby bylo možné zjistit správnost provedené kalibrace. Výsledky jsou v následující tab. 2.6.

Tab. 2.6 – Výsledky kalibrace

Etalon, bar	Senzor 1, bar	Senzor 2, bar
0,000	0,01	0,00
1,000	1,01	1,00
4,000	4,02	4,01
6,000	6,00	5,99
8,000	8,01	8,00
10,000	10,02	9,99
Průměrná odchylka	0,0086	-0,0014

2.5 PNEUMATICKÉ PRVKY

Pneumatická zařízení využívají stlačeného plynu jako pohonu, pro vykonávání pohybu. Nejčastěji používaným plynem je vzduch odebraný přímo z okolní atmosféry a stlačený pomocí vhodného kompresoru.

2.5.1 Pneumatický válec

Pro posuv a udržení stálého přtlaku měřicího hrotu bude použit pneumatický válec, tento pneumatický prvek je mechanické zařízení, které zajišťuje převod síly stlačeného vzduchu na mechanický pohyb. Byl zvolen dvojčinný pneumatický válec FESTO DSNU-16, vyobrazen na obr. 2.10. Tento válec má dvě oddělené komory, každá komora zajišťuje jeden směr výsuvu, oba směry jsou vzájemně protichůdné. Válec má délku výsuvu 8 cm a je určen pro maximální tlak 10 bar, což lze přepočítat na sílu, která pro maximální tlak činí přibližně 200 N.



Obr. 2.10 – Pneumatický válec

2.5.2 Ventily

Pro řízení přívodu vzduchu budou použity FESTO elektromagnetické ventily, umožňují snadné otevření/uzavření pomocí 24 V. Při sepnutí cívky pomocí jmenovitého napětí dojde vlivem elektromagnetického pole k mechanickému otevření pilotního (pomocného) ventilu. Pilotní vzduch potom otevře/zavře příslušný pracovní ventil. Výchozí stav ventilu je dán pomocí mechanické pružiny, která ventil udržuje buď zavřený, nebo otevřený. Ventily FESTO jsou prodávány v podobě bloků, které obsahují několik vstupů a výstupů mezi kterými lze přepínat.



Obr. 2.11 – Ventilový blok

V této práci je použit ventilový blok FESTO CPE10-M1BH-5/3G (viz obr. 2.11), obsahuje dvě elektromagnetické cívky, které otvírají pilotní ventily. Pomocí pilotních ventilů je dovnitř veden pilotní vzduch, který otvírá pracovní ventily. Tlak pilotního vzduchu musí být minimálně 2 bary. Tento ventil je přímo vhodný pro práci s dvojčinným pneumatickým válcem, protože má dva pracovní výstupy. Když je jeden z nich sepnut a je do něj přiveden vzduch, tak se druhý výstup spojí s výpustí a tím je umožněno vypuštění natlakované komory pneumatického válce. Ovládání ventilu je v tab. 2.8. Uvolněním komory je možné napustit komoru druhou a pohnout tak s válcem požadovaným směrem. Druhý elektromagnetický ventil nese označení MSEB-3-24V, je to ventil, který je ve výchozí poloze uzavřený a při přivedení napětí se otevře. Parametry ventilů jsou v tab. 2.7.

Tab. 2.7 – Parametry ventilů

Ventil	Ovládací napětí	Potřeba pilotního vzduchu	Rozsah tlaku	Počet poloh
CPE10-M1BH-5/3G	24 V	ano	0 ÷ 10 bar	3
MSEB-3-24V	24 V	ne	0 ÷ 16 bar	2

Tab. 2.8 – Ovládání dvoucestného ventilu

Cívka 1	Cívka 2	Výstup 1	Výstup 1
1	0	1	0
1	1	X	X
0	0	0	0
0	1	0	1

2.5.3 Kompresor

Jako zdroj stlačeného vzduchu bude použit běžný 12 V kompresor určený pro huštění pneumatik osobních automobilů, parametry v tab. 2.9. Jde o pístový kompresor, který stlačuje vzduch pomocí elektromotoru, jehož točivý pohyb je převeden na kmitavý, a to pomocí klikového mechanismu. Kmitavý pohyb potom vykonává píst ve válci a přes zpětnou klapku je tak možné stlačovat plyn (v tomto případě vzduch). Tento pístový kompresor je schopný vyvinout tlak až 6 barů, což je více než dostatečné, zároveň je velice kompaktní. Aby se předešlo možným problémům s vibracemi během měření, tak bude před zahájením měření vzduchem naplněna tlaková nádoba a kompresor tak během měření nebude v provozu.

Tab. 2.9 – Parametry kompresoru

Parametr	Hodnota
Napájecí napětí	12 V
Výkon	75 W
Maximální tlak	6 bar

2.5.4 Ostatní pneumatické prvky

Vzhledem k vysoké ceně profesionálních tlakových nádob je jako její ekvivalent použita tlaková nádoba práškového 1 kg hasicího přístroje. Tato nádoba je schopná vydržet tlak až 24 bar. Dalšími použitými komponenty jsou různé redukce, hadice a spojky. Jejich propojení bude popsáno v oddílu 2.7.

2.6 LINEÁRNÍ VEDENÍ

Lineární vedení je důležitým prvkem pro stavbu profiloměru, zajišťuje stabilní a plynulý posuv v obou osách. Hlavními požadavky na vedení jsou malý třecí odpor a odolnost

vedení proti pohybu mimo osu. Druhů vedení je několik, ale v práci budou popsány jen použité typy vedení, tzn. podepřené tyče a prizmatické vedení.

2.6.1 Podepřené tyče

Podepřené tyče se skládají z kalené ocelové tyče, která je jemně zbrošena, aby umožňovala hladký pohyb vozíku s kuličkovým ložiskem, vyobrazeno na obr. 2.12. Jelikož je tyč kulatá, a nemá žádné vodící drážky, je třeba použít alespoň dvě tyto tyče souběžně, aby se zamezilo nechtěnému rotačnímu pohybu vozíku. Dalším problémem může být, že je vozík používaný pro podepřené tyče, otevřený což může mít za následek jeho rozlomení při větší zátěži, je proto vhodné použít více vozíků za sebou nebo prodloužené vozíky.



Obr. 2.12 – Otevřený vozík s ložiskem a podepřená tyč

2.6.2 Prizmatické vedení

Na obr. 2.13 je prizmatické vedení, profesionální a velice robustní vedení. Při minimálním lineárním odporu nabízí velmi vysokou přesnost a jeho důmyslně profilovaný tvar zabraňuje pohybu v jiných směrech, než je směr dané osy. Další výhodou je, že i při relativně malých rozměrech snáší vysokou zátěž.



Obr. 2.13 – Vozík a kolejnice prizmatického vedení

2.6.3 Kuličkový šroub

Další důležitou součástí vedení je pohon lineárního vedení. Pomocí kuličkového šroubu a pružné spojky bude převeden rotační pohyb motoru na translační. Obě tyto součástky jsou na obr. 2.14. Druhý konec šroubu bude osazen ložiskem a uložen v ložiskovém domku. Kuličkový šroub zaručuje plynulý a hladký chod, vysokou přesnost a také se příliš neopotřebovává. Tato varianta je nejpoužívanějším pohonem profesionálních lineárních posuvů.



Obr. 2.14 – Kuličkový šroub

2.7 ZAPOJENÍ PNEUMATICKÉ ČÁSTI

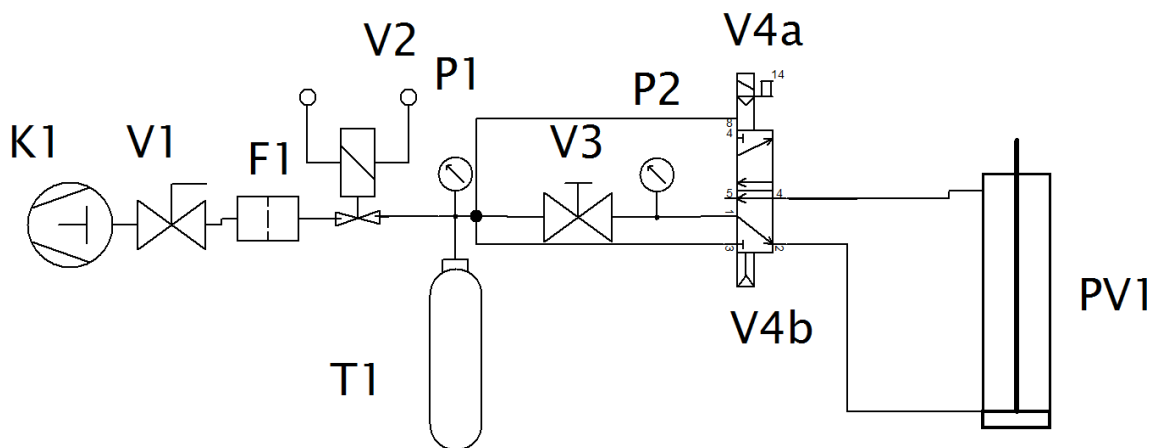
V této kapitole bude popsána pneumatická část měřicího zařízení, která slouží pro obsluhu pneumatického válce. Propojení pneumatických prvků je znázorněno na obr. 2.15, veškerá označení prvků popsanych v této kapitole se vztahují právě k tomuto schématu.

Základem celého systému je kompresor K1 (viz pododdíl 2.5.3), pomocí něhož je do systému přiváděn vzduch. Pro uchování dostatečného množství stlačeného vzduchu slouží

tlaková nádoba T1, mezi nádobou a kompresorem je manuální ventil V1, který slouží pro servisní účely.

Čidlo tlaku P1 (popsáno v oddílu 2.4) sleduje tlak v systému a na základě jeho výstupu je řízena prvotní sekvence, kdy je systém natlakován na požadovaný tlak. Před nádobou je FESTO vstupní blok, ten se skládá z manuálního ventilu V1, filtru F1 (odděluje případné pevné částice a kapalinu od stlačeného média) a elektromagnetického ventilu V2.

Za vstupním blokem následuje manuální regulační ventil V3, kterým se nastavuje tlak jdoucí do válce PV1. V závislosti na tomto tlaku se odvíjí příslušná síla, kterou válec vyvine. Tlak vycházející z redukčního ventilu je měřen pomocí čidla tlaku P2, tento tlak je dále distribuován pomocí dvojcestného ventilu V4. Přepnutí ventilu V4 do jedné nebo druhé polohy umožňuje zasunutí nebo vysunutí pneumatického válce. Jeho ovládání probíhá pomocí dvou elektromagnetických pilotních ventilů V4a a V4b.



Obr. 2.15 – Schéma pneumatické části

3 NADŘAZENÝ ŘÍDICÍ SYSTÉM - HARDWARE

Řídicí systém je programovatelné zařízení schopné ovládat různé technologické procesy. Je kombinací několika zařízení, jako jsou převodníky, čítače a časovače. K jeho funkci je dále zapotřebí vhodného obslužného programu. Výhodou je úplný přehled nad řízeným procesem, jelikož je možné sledovat hodnoty veličin zpracovaných převodníky. Tyto hodnoty, které jsou uchovány v digitální formě, je možno libovolně zpracovávat, zobrazovat, ukládat a zasílat přes různé komunikační sběrnice.

Pro tuto práci byl do úlohy řídicího systému zvolen jednočipový mikropočítač Atmel ATmega 2560, který je integrovaný na vývojové desce typu Arduino MEGA2560.

3.1 ČÁSTI MIKROPOČÍTAČE

Mikropočítač lze rozdělit na několik základních částí. Jsou to procesor, paměť programu, paměť dat a periferní obvody. První tři části jsou součástí každého mikropočítače, ale skladba periferních obvodů se liší dle typu mikropočítače. Skladba periferních obvodů se vybírá podle druhu aplikace, pro kterou bude mikropočítač použit. Jedinou vždy přítomnou periférií jsou vstupní a výstupní obvody. Při použití v automatizaci jsou také obvyklé obvody jako: převodníky, čítače/časovače, výkonové výstupy a galvanicky oddělené vstupy a výstupy (Pinker, 2004).

3.1.1 Základní části mikropočítače

Vzájemnou komunikaci jednotek v mikropočítači zajišťuje struktura sběrnic. Použití sběrnic umožňuje snadné rozšiřování mikropočítače o další jednotky, a to beze změn vnitřního zapojení. Sběrnice má však nevýhodu a tou je, že v každém okamžiku může být připojen pouze jeden zdroj dat. Nelze tak vysílat data z více zařízení ve stejnou chvíli. Komunikace po sběrnici je řízena vždy jednou jednotkou, zpravidla procesorem, ale sběrnici může kontrolovat i jiná jednotka. Sběrnice se dále dělí na tři menší sběrnice: adresovou, datovou a řídicí.

Adresová sběrnice je nezbytná pro adresování paměti a pro rozlišování mezi jednotkami, připojenými na datovou sběrnici. Šířka (počet vodičů) určuje maximální počet adres (tedy i maximální počet zařízení). U osmibitových mikropočítačů mívá adresová sběrnice šířku 16 bitů, pro šestnáctibitové mikropočítače je šířka 20 bitů minimum.

Po datové sběrnici se předávají data, její šířka bývá násobkem čísla osm. Jednotky připojené na tuto sběrnici mohou být jak zdrojem, tak příjemcem dat. Pokud je jednotka zdroj, pak je na datovou sběrnici připojena pomocí třístavových členů.

Zápis, čtení a další činnosti jednotek jsou řízeny pomocí signálů řídicí sběrnice. Většina těchto signálů je generována procesorem, ale mohou je generovat i jiné jednotky, které tímto prostřednictvím mohou zasahovat do chodu procesoru. Jednotky, které generují řídicí signály (včetně procesoru) jsou na řídicí sběrnici připojeny přes členy s otevřeným kolektorem.

Procesor se stará o řízení činnosti celého mikropočítače, zabezpečuje správné provádění instrukcí uložených v paměti programu, zpracovává data v paměti, řídí tok dat ze vstupních obvodů do mikropočítače a tok dat z výstupních obvodů mikropočítače. Řízení probíhá na základě instrukcí, ty vždy začínají operačním kódem (definice činnosti při provádění instrukce) a mohou pokračovat adresou operandu nebo přímými daty. Procesor nejdříve přečte operační kód a následně vyvolá všechny potřebná data pro provedení instrukce.

Paměť programu je soubor instrukcí, jejichž postupným zpracováváním se realizuje cílová činnost mikropočítače. Dále také může obsahovat různé konstanty a tabulky. Obvykle je typu ROM (EPROM, EEPROM, FLASH), aby nebylo možné program za běhu měnit.

Paměť dat poskytuje dočasné úložiště pro data získaná ze vstupních obvodů, pomocné proměnné atd. Tato paměť je vždy typu RAM, je zde důležitá rychlost a trvalé uložení informací zde ani není žádoucí (Pinker, 2004).

3.1.2 USART

Synchronní / asynchronní sériové rozhraní je zařízení pro sériovou komunikaci, pracuje buď v režimu asynchronním (SCI, používá se například pro RS232 nebo RS485) nebo v synchronním režimu (SPI).

Asynchronní přenos vysílá data na pinu označeném jako Tx (transmit) a přijímá data na pinu označeném jako Rx (receive). V případě klidového stavu, neprobíhá vysílání, ani příjem, je úroveň signálu logická „1“. Vysílání se zahajuje pomocí start bitu, tzn. změnou hodnoty na logickou „0“ po dobu jednoho bitu. V případě 8bitového přenosu následuje osm datových bitů vysílaných od nejnižšího po nejvyšší, po kterém následuje stop bit, který má úroveň logické „1“. Synchronizace je znaková, znaky tedy za sebou mohou následovat s libovolnými mezerami.

Asynchronní komunikace umožňuje jednoduchou, ale jen částečnou kontrolu přijatých dat. Ke kontrole přijatých dat se využívají bity ve stavovém registru, rozlišují se čtyři druhy chyb:

- chyba bitu (rozdíl vzorků při vícenásobném vzorkování),
- chyba parity (paritní bit nesouhlasí s vypočtenou paritou přijatých dat),
- chyba rámce (na pozici stop bitu je opačný stav),
- chyba ztráty paměti (nastane ve chvíli, kdy nebyla včas přečtena přijatá data).

V případě synchronního přenosu je jako další vodič přidán hodinový signál, který zajišťuje synchronizaci všech připojených zařízení. V případě využití synchronního přenosu je třeba definovat Master a Slave zařízení. (Pinker, 2004)

Tato práce využívá toto rozhraní pro komunikaci s PC, a to přes převodník FT232, který rozhraní UART převádí na standard USB.

Rychlost komunikace se určuje pomocí Baudů, udává počet změn stavu přenosu za sekundu. Pro tuto aplikaci je použita přenosová rychlost 9600 Bd.

3.1.3 SPI

Serial Peripheral Interface (SPI) je sériové periferní rozhraní používané pro komunikaci mezi mikropočítači a dalšími integrovanými obvody. Tento druh rozhraní není příliš odolný proti rušení, a proto se používá pro komunikace na malé vzdálenosti, většinou mezi obvody na jedné desce. Komunikace probíhá na jedné sběrnici společné pro všechny zařízení. Zařízení jsou adresována pomocí vodiče (SS nebo CS), který při logické „0“ zahajují příjem a vysílání.

Zařízení na sběrnici SPI, jsou rozdělena na typy Master a Slave. Master řídí komunikace skrze hodinový signál. Dále také Master vysílá signál SS (Slave select) nebo CS (Chip select), kterým určuje, s jakým zařízením bude probíhat komunikace. Slave pak na základě hodinového signálu vysílá data, pokud předtím byl aktivován signálem SS (CS).

Komunikaci začíná Master tím, že nastaví logickou „0“ na adresačním vodiči zařízení s kterým je třeba komunikovat. Následně začne generovat hodinový signál na vodiči SCLK, zároveň obě zařízení začínají vysílat data. Master přes pin MOSI (Master Out, Slave In) a Slave pak přes MISO (Master In, Slave Out). Po odeslání dat je možno dále komunikovat do doby ukončení komunikace. Komunikace se ukončuje zastavením vysílání hodinového signálu a nastavením signálu SS (CS) daného zařízení na logickou „0“ (Freescale Semiconductors, 2004).

3.1.4 Sběrnice I²C

Tato sériová sběrnice se používá pro připojení nízkorychlostních periférií. Sběrnice rozděluje připojená zařízení na řídicí (master) a řízené (slave). Master určuje zahájení a konec komunikace a generuje hodinový signál SCL. Slave je zařízení adresované masterem a na jeho pokyn odesílá požadovaná data.

Sběrnice I²C umožňuje připojení až 128 zařízení, k tomu je zapotřebí dvou vodičů. Vodič SCL (Synchronous Clock) přenáší hodinový signál a druhý vodič SDA (Synchronous Data) přenáší data. Pro oba kanály se využívá zapojení s otevřeným kolektorem. Základním pravidlem pro komunikaci je, že změna stavu SDA je povolena pouze když je SCL v úrovni L. Pokud je toto pravidlo porušeno znamená to vysílání zprávy START nebo STOP, které slouží pro zahájení a ukončení komunikace.

Maximální frekvence pro I²C je 400 kHz, pro frekvenci signálu SCL je určena minimální doba setrvání signálu SCL na úrovních H (high – logická 1) a L (low – logická 0). Při synchronizaci jednotlivých stanic je využíván právě signál SCL, jehož frekvence je měřena vnitřním časovačem každé stanice. Sběrnice neumožňuje duplexní přenos, tzn. v jednu chvíli, může vysílat pouze jedno zařízení. Každé zařízení připojené na sběrnici musí mít vlastní jedinečnou adresu délky 7 nebo 10 bitů.

Pro kontrolu spojení je za každým vyslaným bytem vyslán bit ACK. Zařízení, které vysílá, jej vysílá na úrovni H a příjemce jeho přijetí potvrzuje připojením vodiče SDA na úroveň L. V případě, že vysílající zařízení neobdrží potvrzení o přijetí bitu ACK je komunikace ukončena pomocí zprávy STOP (NXP Semiconductors, 2014).

Tato sběrnice je použita pro komunikaci se senzory tlaku popsané v oddílu 2.4.

3.1.5 Čítač/časovač

Čítač/časovač je jednou z nejvíce používaných součástí mikropočítačů, vyhrazených pro činnost v reálném čase. Zpracovává časové funkce, generuje signály s přesným časováním, počítá vnější signály a může sloužit pro synchronizaci programu. Pokud je v mikropočítači obsažen čítač/časovač, lze spoustu úloh realizovat s jeho pomocí a není třeba tvořit různé zpoždovací podprogramy apod. Použití čítače/časovače tedy značně zvyšuje efektivitu mikropočítače a šetří jeho výkon. V úkonech, kde je třeba plně využívat výpočetní výkon mikropočítače, jsou čítače/časovače nezbytné.

K ovládání čítače/časovače slouží sada vnitřních registrů mikropočítače. Některé se nastavují již na začátku programu, jako třeba počet pulzů potřebných pro přetečení nebo jaký

úkon se má pro přetečení provést. Jiné registry se potom používají i průběžně, například povolání a blokování čítání.

V této práci bude použit pouze režim časování, proto bude popsán detailněji. Funkce časovače se dospěje tak, že na vstup čítače je přiveden přesný kmitočet z generátoru, obvykle řízeného krystalem. V tomto případě je použit hodinový signál mikropočítače, který je udávám externím krystalem (12 MHz). Před přivedením signálu do čítače je ještě obvyklý dělicí obvod, který je schopen vstupní signál dělit (obvyklé dělení v mocninách dvou: 2, 4, 8, 16) a umožňuje tak odměření delšího časového okamžiku (Pinker, 2004).

Funkce časovače bude využita pro spuštění přerušení a následnou obslužnou rutinu tlačítek.

3.1.6 Digitální vstup/výstup mikropočítače

Digitální výstupy v mikropočítači jsou většinou řešeny dvěma základními způsoby: dvojčinný (Push-pull) výstup a otevřený kolektor (v případě že je místo tranzistoru použit MOSFET je tento vstup/výstup označován jako „open drain“, ale v chování je totožný s otevřeným kolektorem).

Dvojčinný výstup je tvořený dvojicí tranzistorů, které představují spínač vůči napájecímu napětí a spínač vůči zemi. Tranzistory jsou zapojené tak, že v ustáleném stavu je vždy otevřený pouze jeden z tranzistorů a tím se definuje logická hodnota výstupu.

Vstupy mikropočítače jsou řešeny pomocí rezistorů buď pull-up (rezistor připojený na úroveň logické „1“) nebo pull-down (rezistor připojený na úroveň logické „0“).

3.2 ATMEGA 2560

ATMega 2560 je jednočipový mikropočítač, který v sobě zahrnuje vše potřebné, aby byl schopen obsloužit potřebné úkony. Tento 8bitový mikropočítač má 256 kB vnitřní flash paměti, 8 kB RAM, 4 kB EEPROM, 54 digitálních vstupů/výstupů (z nich může být 15 použito pro generování PWM signálu), 16 analogových vstupů, dva 8bitové čítače/časovače, čtyři 16bitové čítače/časovače a 4 USART porty. Pomocí externího krystalu lze dosáhnout taktovací frekvence až 16 MHz. Sada instrukcí toho mikropočítače je typu RISC.

Důležitou funkcí, kterou mikropočítač umožňuje je externí přerušení, to je možné přes piny 2, 3, 21, 20, 19, 18. Při nastavené úrovni vstupního signálu na jednom z těchto pinů je možné přerušení právě prováděného programu a vykonání příslušného obslužného programu (ATmega2560, 2013).

V následující tab. 3.1 jsou uvedena zařízení připojená k mikropočítači.

Tab. 3.1 – Zapojení digitálních pinů ATMegy 2560

Připojené zařízení	Čísla digitálních pinů	Odpovídající signály
Driver	42, 44, 46	PUL, DIR, ENABLE
Klávesnice	41, 43, 45, 47, 48, 49	CANCEL, ENTER, DOWN, UP, LEFT, RIGHT
Koncové spínače	34, 36, 38, 40	ENDy0, ENDx1, ENDy1, ENDx0
LCD	23, 25, 27, 29, 30, 32	RS, E, d4, d5, d6, d7
Relé deska	31, 33, 35, 37, 39	relé 1, relé 2, relé 3, relé 4, relé 5
Senzory tlaku	18, 19, 20, 21	SCL, SDA, T1, T2
Optické snímače	2, 3, 22, 24, 26, 28	A1, A2, B1, B2, R1, R2

3.3 ARDUINO

Projekt Arduino vznikl v Itálii v roce 2005 a jeho cílem bylo vytvořit jednoduchou platformu pro studenty, která bude umožňovat snadný a rychlý vývoj. Jde o malé jednodeskové počítače postavené na základě mikropočítačů typu ATMega od výrobce Atmel. Tyto počítače jsou určeny pro řízení jednoduchých procesů a zařízení. Během let existence Arduina, bylo využito k tisícům projektů, a to od jednoduchých zařízení (řízení motoru, měření teploty, blikání LED, obsluha displeje...) po komplexní vědecké projekty.

Vývojová deska zvolená pro tuto práci je Arduino Mega 2560 (viz obr 3.1) s již výše zmíněným jednočipem ATMega 2560. Pro řízení profiloměru by měla být dostatečně rychlá a také má dostatečný počet vstupů/výstupů a periférií.

Arduino Mega 2560 obsahuje kromě jednočipového mikropočítače také převodník FT-232 a díky němu možnost komunikace s PC přes USB port. Tento USB port slouží také k nahrávání programu do mikropočítače. K tomuto účelu je nutný Bootloader, malý program, který uchovává informaci o umístění operačního systému, aby bylo možné program nahrát.

Pro psaní programů slouží programovací jazyk Arduino, jde o jazyk podobný jazyku C++. Jako vývojové prostředí se používá Arduino IDE, umožňuje psaní, kompilaci a nahrání programu do Arduino vývojové desky (Arduino, 2017).



Obr. 3.1 – Arduino Mega 2560

3.4 LCD DISPLEJ 16×4

Jako naprostá většina dnes používaných menších displejů je i zvolený čtyřřádkový displej (viz obr. 3.2) založen na integrovaném obvodu HD44780. Jde o kontrolér pro LCD od výrobce Hitachi. Kontrolér zahrnuje veškeré potřebné elementy pro činnost LCD, jako je paměť RAM, znakový generátor, analogová elektronika pro buzení displeje. Řízení libovolného displeje probíhá pomocí čtyř nebo osmibitové sběrnice. Jeden tento kontrolér je schopen řídit dvouřádkový znakový displej, pro zvolený displej se čtyřmi řádky jsou zapotřebí dva tyto kontroléry.

Pro připojení k mikropočítači je datová sběrnice rozšířena o piny E, R/W a RS. Pin E (Enable) slouží pro aktivaci zařízení, pinem R/W (Read/Write) se volí mezi čtením a zápisem a pinem RS (Register Select) se volí mezi instrukčním a znakovým módem. Instrukční mód slouží například pro úpravu integrované znakové sady a umožňuje tak tvorbu vlastních znaků nebo je skrze něj možno mazat aktuální zobrazená data a přesouvat kurzor. Ve znakovém módu je na displeji zobrazen příslušný znak, a to na současné pozici kurzoru (Learningaboutelectronics.com, 1999).

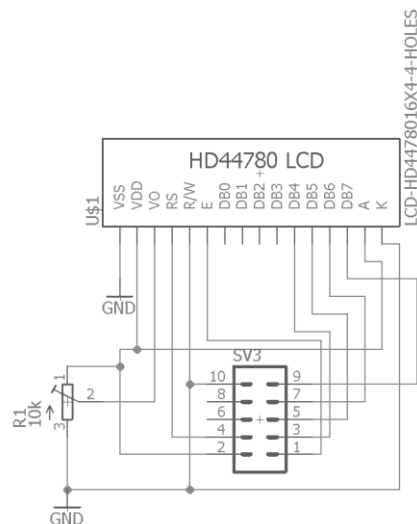


Obr. 3.2 – LCD HD44780 16×4

Z důvodu jednodušší instalace a práce s displejem, je displej ještě připojen na nosnou desku (viz obr. 3.3), která zjednodušuje jeho připojení. Napájení displeje a nastavení intenzity podsvícení je integrováno přímo na této desce, slouží k tomu trimr o hodnotě 10 k Ω . Dále je zde také trvale připojen pin R/W, a to na GND. Funkce čtení z displeje nebude totiž v programu využita. Zbylé vývody jsou přivedeny na konektor typu ML10 pro plochý kabel, který usnadňuje propojení s mikropočítačem. Zapojení konektoru ML10 je v tab. 3.2.

Tab. 3.2 – Zapojení konektoru ML10 na desce LCD

Číslo pinu konektoru	Označení signálu
1	E
2	Vdd
3	DB4
4	RS
5	DB5
6	NC
7	DB6
8	NC
9	DB7
10	R/W



Obr. 3.3 – Zapojení LCD

3.5 DESKA SPÍNACÍCH RELÉ

Jelikož maximální proud jedním pinem ATMegy 2560 je 40 mA, je pro spínání prvků, které potřebují vyšší proud (kompresor a ventily), nezbytné použít rozšiřující desku (viz obr. 3.4). Tato deska pomocí NPN tranzistoru a signálu z jednočipového mikropočítače spíná relé (čtyři FTR-F3 a jedno NT72-2C-S10), FTR-F3 je použito pro spínání 24 V umožňuje proud kontaktem až 5 A, což je pro ventily více než dostačující. Relé NT72-2C-S10 je určeno pro spínání kompresoru, jehož napájecí napětí je 12 V. Bylo tedy třeba jej oddělit, navíc kompresor má příkon 75 W, což při napájecím napětím 12 V znamená proud 6,25 A. Deska obsahuje celkem 5 relé bloků. Pro snadnější propojení jsou vstupy desky opět na konektoru ML10, odpovídající čísla relé jsou v tab. 3.3.

Tab. 3.3 – Zapojení konektoru ML10 na Relé desce

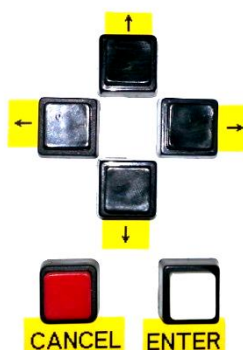
Číslo relé	Čísla pinů konektoru
1	10, 9
2	8, 7
3	6, 5
4	4, 3
5	2, 1



Obr. 3.4 – Deska spínacích relé

3.6 TLAČÍTKA

Pro ovládání profiloměru bude k dispozici jednoduchá klávesnice složená z 6 tlačítek (viz obr. 3.6). Další tlačítka, popřípadě mikrospínače, budou použity pro určení limitní polohy posuvu os. V tab. 3.4 je zapojení jednotlivých tlačítek klávesnice.



Obr. 3.5 – Rozložení klávesnice

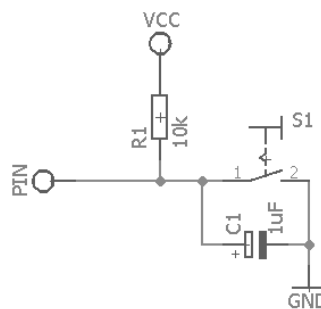
Tab. 3.4 – Zapojení konektoru ML10 pro klávesnici

Číslo pinu konektoru	Označení signálu
1	5 V
2	GND
3	NC
4	UP
5	NC
6	DOWN
7	RIGHT

Tab. 3.4 – Zapojení konektoru ML10 pro klávesnici – pokračování

Číslo pinu konektoru	Označení signálu
8	ENTER
9	LEFT
10	CANCEL

Na obr. 3.6 je příklad zapojení jednoho z tlačítek na pin mikropočítače. Použité zapojení je velmi jednoduché, ale také velice účelné. Rezistor R1 plní funkci pull-up rezistoru, tedy ve stavu, kdy je tlačítko rozepnuté udržuje na vstupu mikropočítače vyšší logickou úroveň. Druhou důležitou součástí je kondenzátor C1, jeho úlohou je potlačení zákmitů vzniklých při mechanickém spojení kontaktů uvnitř tlačítka. Tab. 3.5 obsahuje informace o zapojení koncových mikropsínačů.



Obr. 3.6 – Připojení tlačítka

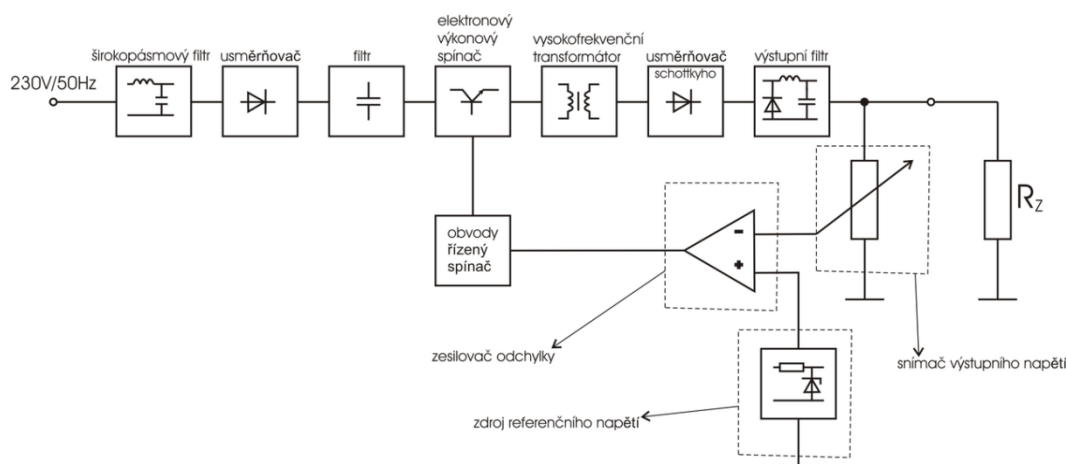
Tab. 3.5 – Zapojení konektoru ML10 pro koncové mikropsínače

Číslo pinu konektoru	Označení signálu
1	5 V
2	GND
3	NC
4	ENDy0
5	NC
6	ENDx1
7	NC
8	ENDy1
9	NC
10	ENDx0

3.7 SPÍNANÝ ZDROJ

Protože navrhovaný profiloměr je zamýšlen jako přenosné zařízení tak byl zvolen namísto síťového transformátoru spínaný zdroj, který při shodném výkonu má mnohem menší rozměry. Nevýhodou spínaných zdrojů může být rušení, které vzniká při spínání výkonových tranzistorů. Ale tato nevýhoda se projeví spíše v audio technice, takže pro profiloměr je zdroj vhodný.

Nejčastěji používaným zdrojem bez síťového transformátoru, je zdroj s vysokofrekvenčním transformátorem na obr. 3.7. Síťové napětí s frekvencí 50 Hz se usměrní a filtruje. Ze vzniklého stejnosměrného napětí spínač vytvoří střídavé napětí o frekvenci přesahující 20 kHz. Za spínačem následuje výše zmíněný vysokofrekvenční transformátor, po transformaci je napětí usměrněno rychlými Schottkyho diodami a filtrováno. Toto výstupní napětí se ještě pomocí zpětné vazby porovná s referenčním napětím a vlivem rozdílu dochází k regulaci pomocí spínače.



Obr. 3.7 – Blokové schéma spínaného zdroje

Pro filtraci síťového napětí se využívá širokopásmové filtru, který odstraňuje rušení způsobené spínací součástkou. Takto vyfiltrované napětí se usměrní, nejčastěji standardním zapojením Gretzův můstek. Následuje filtr, jeho účel je vyhladit průběh usměrněného napětí. Filtr tvoří elektrolytický kondenzátor, který musí vydržet poměrně velké závěrné napětí a zároveň mít malý efektivní stejnosměrný odpor. Tyto požadavky musí být splněny, aby byl kondenzátor schopen generovat krátké, avšak velké proudové impulzy.

Nejproblematičtějším a velice náročným prvkem je spínač, je tvořen spínacími tranzistory typu FET. Musí být schopen (respektive tranzistory musí být schopny) spínat

vysoké frekvence, až 1 MHz, a to pro vysoké proudy (jednotky až desítky ampér). Dalším požadavkem je závěrné napětí, které dle typu měniče, může dosahovat až 1000 V.

Nejdůležitější součástí spínaného zdroje je vysokofrekvenční transformátor. Je tvořen feritovým jádrem, jehož přesné složení se mění vzhledem k danému frekvenčnímu pásmu. Kvalita tohoto transformátoru do velké míry určuje vlastnosti celého spínaného zdroje.

Na usměrňovač za vysokofrekvenčním transformátorem je nutno použít rychlé diody, které mají velice krátké zotavovací časy. Nejlepší a nejpoužívanější jsou Schottkyho diody, ty splňují dané požadavky, a navíc v propustném směru mají menší úbytek napětí než běžné diody.

Funkce výstupního filtru je snížit vlnění výstupního napětí na takovou úroveň, aby bylo použitelné pro napájení daného zařízení. Kondenzátory v tomto filtru jsou většinou pro tento účel speciálně vyráběny nebo se nahrazují paralelním spojením dvou a více menších kapacit.

Poslední důležitou částí je zpětná vazba s rozdílovým zesilovačem, který zesiluje rozdíl mezi referenčním a výstupním napětím. Tento rozdíl slouží jako vstup pro obvod řízení spínače, jeho úkolem je spínat tranzistory. Spínání se realizuje za pomoci PWM (pulsní šířková modulace), její princip je v rozdílné době časů, kdy je vypínač sepnutý, a kdy je vypnutý. Různým poměrem těchto časů se vytváří výstupní napětí.

V této práci konkrétně jsou použity tři spínané zdroje, pro tři různá napětí a to: 5 V, 12 V a 24 V. Spínaný zdroj s výstupem 5 V slouží pro napájení řídicí části zařízení, jeho maximální proud jsou 2 A. Druhý spínaný zdroj s výstupem 12 V je určen pro napájení kompresoru, jeho maximální proud při 12 voltech je 16 A. Bylo třeba použít tohoto zdroje, protože použití prvku pro převod napětí z 24 V na 12 V, je při vysokém proudu kompresoru (proud se pohybuje kolem 6 A), velice neekonomické. Poslední 24 V spínaný zdroj slouží pro napájení driveru krokového motoru a spínání elektromagnetických ventilů, jeho maximální proud je 10 A. Kompletní soupis zdrojů je v tab. 3.6 a jejich fotografie jsou na obr. 3.8.

Tab. 3.6 – Použité spínané zdroje

Typ	Výstupní napětí	Maximální proud	Účel napájení
S-10-5	5 V	2 A	Řídicí elektronika
ATX	12 V	16 A	Kompresor
SYS-24250	24 V	10 A	Ventily, krokový motor



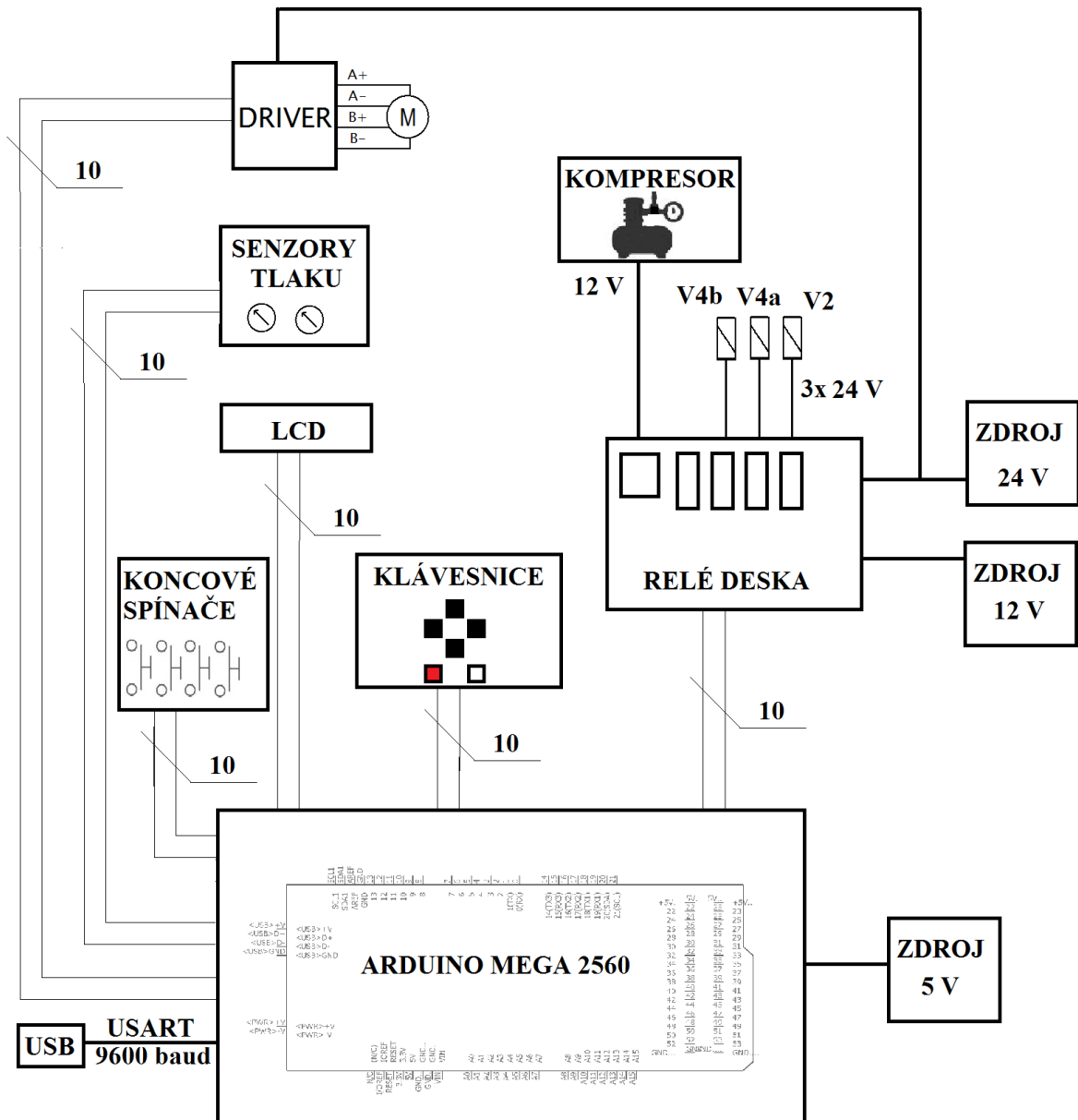
Obr. 3.8 – Použité spínané zdroje

3.8 ZAPOJENÍ ELEKTRONICKÉ ČÁSTI

Na obr. 3.9 je zobrazeno zjednodušené schéma zapojení elektronické části profiloměru. Jsou zde zobrazeny pouze hlavní propojení jednotlivých prvků. K propojení mezi deskami je využito plochých deseti žilových kabelů a příslušných konektorů ML10. I když nejsou vždy využity všechny vodiče kabelu, je toto spojení použito všude.

Veškeré kontrolní signály vychází z hlavní desky Arduino Mega2560, na její sběrnici I²C jsou připojeny dvě čidla tlaku P1 a P2. Mikropočítač řídí také spínání relé desky, která zajišťuje otvírání/zavírání ventilů a běh kompresoru. Ventil V4 je dvoucestný a je tedy ovládán pomocí dvou elektromagnetických ventilů V4a a V4b.

Driver krokového motoru je řízen za pomoci signálů ENABLE, DIR a CLK. Poslední částí je ovládací panel s tlačítky a LCD displejem. Čtyři tlačítka umístěná tak, že tvoří vrcholy kosočtverce, budou plnit podobnou funkci jako šipky na klávesnici, budou sloužit k pohybu v menu displeje. Další dvě tlačítka budou sloužit pro potvrzovací/spouštěcí a rušící/návratové příkazy.



Obr. 3.9 – Schéma elektronické části

4 NADŘAZENÝ ŘÍDICÍ SYSTÉM - SOFTWARE

Pro programování mikropočítačů lze použít libovolný jazyk, který je možné převést do strojového kódu cílového mikropočítače. Výrobce mikropočítačů, firma Atmel, poskytuje vlastní vývojové prostředí, a to AVR Studio nebo novější Atmel Studio. Při použití Arduino vývojového kitu je ale nejvhodnější použít Arduino IDE.

4.1 ARDUINO IDE

Vývojové prostředí Arduino IDE je psáno v jazyce Java, tento jazyk umožňuje běh programu na velkém množství platform a bude mu věnován samostatný oddíl 5.1. Toto prostředí obsahuje textový editor pro psaní kódu, textovou konzoli, okno zpráv a další funkce. Samozřejmostí je i podpora více než třiceti jazyků, mezi které spadá i čeština.

Programy psané v tomto vývojové prostředí se nazývají skeče (sketch), tyto skeče psané v textovém editoru jsou ukládány v souborovém formátu „.ino“. Editor podporuje funkce pro vyjmutí, kopírování, hledání a nahrazování textu. Okno zpráv zobrazuje zpětnou vazbu při ukládání a exportu, dále také zobrazuje chybová hlášení. Konzole zobrazuje výstup Arduino IDE, jako jsou chybové zprávy a další informace. V hlavním okně programu je také informace o připojené desce a číslu COM portu, na který je deska připojena.

Předtím, než začne samotné nahrávání napsaného programu, je třeba vybrat příslušné připojené zařízení a číslo COM portu, na kterém je připojeno. Při správném zadání těchto informací je možno spustit nahrávání programu. Jako první proběhne kompilace a kontrola chyb programu. Spuštění nahrávání automaticky spustí RESET připojené Arduino desky. Nahrávání lze pozorovat blikáním LED diod na vývojové desce, které mají označení Tx a Rx.

Při nahrávání programu se využívá již výše zmíněného bootloaderu, tento malý program byl již nahrán výrobcem přímo do mikropočítače. Umožňuje nahrávání nového software bez přídavných hardwarových zařízení tzv. programátorů. Bootloader je aktivní jen pár sekund právě po vyvolané funkci RESET, poté začne nahrávání zvoleného skeče.

Aby bylo programování snadnější, prostředí obsahuje některé základní knihovny. Tyto knihovny jsou již předpřipravené programy sloužící pro obsluhu různých periférií. Knihovny lze najít téměř na libovolnou periférii nebo je možné napsat si vlastní. Arduino IDE obsahuje něco přes deset základních knihoven, například: knihovna pro obsluhu paměti EEPROM, knihovna pro LCD displeje, knihovna pro GSM moduly, knihovna pro obsluhu servo motorů a další.

Další pomocnou funkcí je integrovaný monitor sériového portu, není tedy třeba další pomocný software. Tento monitor zobrazuje data přijatá z Arduino desky, lze skrze něj i data odesílat. Nastavení je jednoduché je jen třeba vybrat příslušný COM port a rychlost komunikace (Arduino, 2017).

4.2 PROGRAM V MIKROPOČÍTAČI

Program v mikropočítači začne ve chvíli, kdy bude připojen napájecí zdroj na desku Arduino. Po startu programu budou následovat podprogram inicializační sekvence, kde bude zařízení uvedeno do výchozí polohy a připraví se na měření.

Pro automatické měření bude nutné také zadat délku posuvu ve vertikální ose, tedy rozsah měření. Toto nastavení je možno provést přímo na ovládacím panelu zařízení.

Dalším krokem už je samotná měřicí sekvence, přístroj začne měření v nastaveném rozsahu. Tato sekvence může trvat i několik minut v závislosti na nastaveném kroku měření. Během měření budou průběžně odesílány naměřené hodnoty, tedy výstupy z lineárních snímačů polohy, do paměti PC, kde budou následně zpracovány pomocí vyhodnocovacího softwaru. Základní algoritmus je na obr. 4.1.



Obr. 4.1 – Algoritmus hlavního programu

4.2.1 Přerušení programu

Přerušení je způsob, kterým mikropočítač reaguje na neočekávanou událost a spustí její obsluhu. Toto přerušení (neočekávaná událost) může být spuštěna buď vnitřními obvody mikropočítače, nebo se může jednat o vnější signál. Událost může být vyvolána například periferními obvody, doběhnutím časovače, příjmem znaku na sériové lince nebo dokončeným převodem A/D převodníku.

Událost vyvolá přerušení současného programu a následuje skok na obslužný podprogram (rutinu). Po dokončení podprogramu se mikropočítač vrací na místo v přechozím programu. Při obsluze periférií je obsluha většinou jednoduchá a obvykle spočívá v přečtení datového registru periferního obvodu. Načtená data jsou poté pouze uložena do paměti pro další zpracování.

Pokud je třeba pracovat v reálném čase, kde je důraz na přesné dodržení času pro operace, je třeba počítat s latencí (zpožděním) přerušení. Je to doba, která uplyne od výskytu události do vyvolání obsluhy přerušení.

Událostí vyžadujících přerušení je často více v jednom okamžiku, k jejich zpracování slouží řadič přerušení, který je integrován v mikropočítači.

Jeho základní úlohy jsou následující

- eviduje aktivní žádosti na přerušení,
- pokud je aktivních žádostí více, tak je řadí dle stanovené priority,
- podává procesoru informaci o původu žádosti, identifikuje zdroj žádosti,
- dává procesoru informaci o existenci aktivní žádosti.

Důležitou více zmíněnou funkcí řadiče přerušení je řazení více současných požadavků přerušení do fronty, a to dle předem dané priority. Nejvyšší prioritu má nemaskovatelné přerušení (přerušení, které není možné blokovat), toto přerušení je často vyvedené na pin mikropočítače a slouží v případě havárie nebo zaseknutí programu mikropočítače. Druhé v řadě je přerušení od diagnostických obvodů (watchdog timer apod.). Třetí v prioritní řadě je vnější požadavek na přerušení (periferní obvody, senzory). Čtvrtou prioritu mají pak čítače/časovače. Předposledním členem řady jsou sériové vstupy/výstupy. A nejmenší prioritu mají pak A/D převodníky. (Pinker, 2004)

Při psaní programu, ve kterém se pracuje s přerušáním, je nutné nejdříve definovat, která přerušení budou použita. Přidělení externího přerušení na piny mikropočítače je dáno jeho HW konfigurací od výrobce. Pro obsluhu optických inkrementálních snímačů budou využity dvě externí přerušení, a to na pinech 2 a 3. Pomocí funkce

attachInterrupt(pin,rutina,událost) je přerušení definováno. Parametr *pin* určuje, který pin mikropočítače bude požadavek na přerušení vyvolávat. Dalším parametrem je *rutina*, tento parametr je název příslušného podprogramu, který chceme při přerušení vykonat. Posledním parametrem je *událost*, zde se nastavuje příslušná spouštěcí událost.

Funkce umožňuje reakci na několik událostí

- LOW – spustí přerušení, ve chvíli, kdy je pin na nízké logické úrovni,
- CHANGE – přerušení se spustí ve chvíli, kdy dojde ke změně stavu na pinu,
- RISING – reakce na změnu z nízké logické úrovně na vysokou úroveň,
- FALLING – změna opačná k RISING (z vysoké na nízkou log. úroveň),
- HIGH – přerušení se vyvolá, když je pin na vysoké log. úrovni.

V případě snímačů optických snímačů vzdálenosti bude použita událost RISING a bude snímán výstup snímače A, který signalizuje přejetí značky na pravitku. Při přejetí značky bude tedy vyvoláno přerušení a proveden podprogram. Obsah podprogramu je jednoduchý, v závislosti na stavu výstupu B se provede inkrementace nebo dekrementace proměnné (viz obr. 2.1), v které je uložena současná poloha snímače, který přerušení vyvolal.

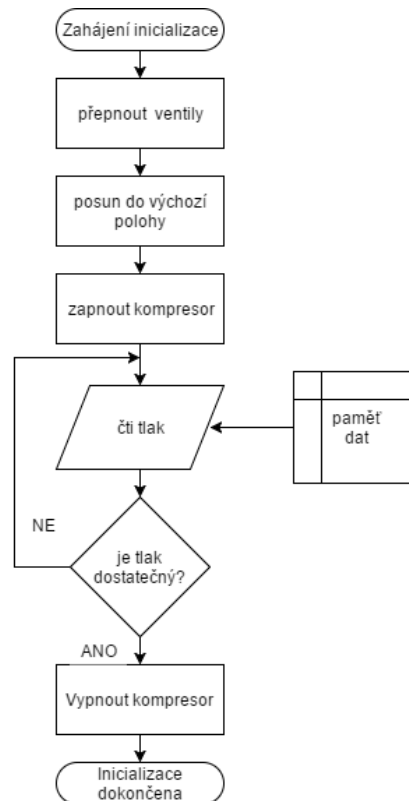
Když je příslušné přerušení pomocí funkce přesně definováno, stačí pak příkazem *interrupts()* povolit (odmaskovat) přerušení

4.2.2 Inicializační sekvence

Při startu měřicího zařízení je třeba všechny prvky uvést do výchozích stavů a poloh, kompletní algoritmus inicializační sekvence je na obr. 4.2. Jako první jsou tedy obslouženy ventily pneumatického vedení, jedná se o ventil u filtru a také dvojcestný ventil u pneumatického válce. Ventil u filtru se nastaví do polohy otevřeno. Dvojcestný ventil bude otevřen tak, aby stlačený vzduch zasouval pneumatický válec. Ovládání ventilů je realizováno pomocí relé na spínací desce (viz oddíl 3.5), která spínají 24 V přivedených na elektromagnetický ventil.

Nyní je na řadě uživatel pomocí klávesy ENTER spustí kompresor, který začne plnit celý systém vzduchem až do přednastaveného tlaku 3 bary. V tuto chvíli bude nějakou dobu trvat, než bude dosažen požadovaný tlak, proto je tento čas využit pro obsluhu krokového motoru (respektive driveru krokového motoru), který pohání posuv vertikální osy. Vertikální osa je přesunuta také do výchozí horní polohy. Výchozí poloha vertikální osy je snímána pomocí koncového mikrospínače s ramínkem.

Následuje cyklické dotazování, jestli už je v systému dostatečný tlak, ve chvíli, kdy je tohoto tlaku dosaženo, je zavřen ventil za kompresorem a kompresor se vypíná. Během tohoto procesu jsou zobrazovány na displeji oba měřené tlaky, jakmile je dosaženo příslušného tlaku na jednom nebo druhém okruhu, je to indikováno na displeji symbolem „OK“. Nyní je zařízení připraveno pro další činnost. Uživatel je vyzván ke stisku klávesy ENTER a program se přesune na nastavení.



Obr. 4.2 – Algoritmus inicializační sekvence

4.2.3 Obsluha periférií

Během běhu programu je třeba obsluhovat několik periférií, a to LCD displej, tlačítka a senzory tlaku. Obsluha displeje je prováděna skrze základní knihovnu LiquidCrystal.h, tato knihovna umožňuje snadné ovládání displeje. Stačí pouze zadat na jaké piny je displej připojen, pak pomocí příkazů lze na displeji zobrazovat znaky, posouvat kurzor nebo mazat aktuální zobrazené znaky.

Stisknutí tlačítka je kontrolováno každých 20 ms, k odměření tohoto času je využit vnitřní čítač/časovač timer 1. Při odměření zadaného času je vyvoláno přerušení a je proveden příslušný podprogram. V tomto případě jsou přečteny hodnoty pinů mikropočítače, na kterých

jsou tlačítka připojena. K nastavení čítače/časovače je použita knihovna `TimerOne`, které obsahuje metody pro přímé nastavení parametrů časovače. Pomocí příkazu `Timer1.initialize(perioda)` je v jednotkách mikrosekund nastavena požadovaná doba pro vyvolání přerušení. A dalším příkazem `Timer1.attachInterrupt(rutina)` je při přerušení obsloužena připravená rutina ve které se přečtou stavy jednotlivých tlačítek.

Při opakovaném vykonávání úlohy na základě stisknutého tlačítka je třeba přidat určité zpoždění, aby se například nastavovaný parametr neměnil příliš rychle a bylo tak možné jeho pohodlné nastavení.

Tlakové senzory jsou obslouženy při každém průchodu smyčkou, respektive při každém průchodu smyčkou je přečten tlak z jednoho z nich a při dalším průchodu z druhého senzoru.

K ovládání driveru krokového motoru byla vytvořena funkce `posuv_y(směr)`. Funkce dle vstupního parametru generuje signály pro driver, který na jejich základě pootočí krokovým motorem o jeden krok požadovaným směrem.

Signály, které metoda generuje, jsou:

- DIR (na základě úrovně tohoto signálu je určen směr rotace),
- PUL (impulz k otočení, driver reaguje na náběžnou hranu tohoto signálu).

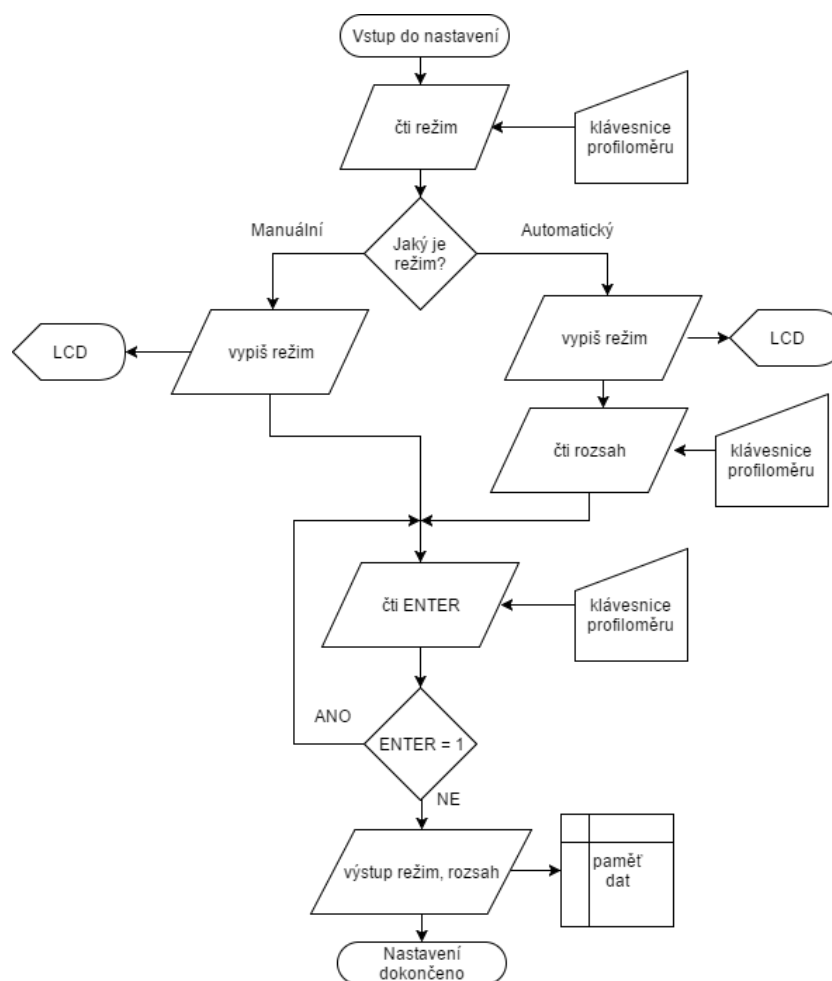
Signál PUL je nejdříve nastaven na logickou „1“ a po uplynutí 200 μ s (minimální šířka pulzu pro buzení driveru) je nastaven zpět na logickou „0“.

4.2.4 Nastavení parametrů

V menu nastavení (viz obr. 4.3) lze nastavit několik parametrů, hlavní funkcí k nastavení je režim. K dispozici jsou dva režimy:

- automatický (při měření se automaticky vykoná měření v předem zadaném rozsahu),
- manuální (měření bude provedeno manuálně pomocí kláves na ovládacím panelu).

V případě vybrání automatického režimu je zobrazen další parametr, a to je, rozsah měření v ose Y. Rozsah je zadáván v jednotkách centimetrů pomocí kláves na ovládacím panelu. Po potvrzení rozsahu měření, nebo vybrání manuálního režimu, je nastavení hotové a potvrzením uživatele, pomocí klávesy ENTER, se přechází do měřicí sekvence.



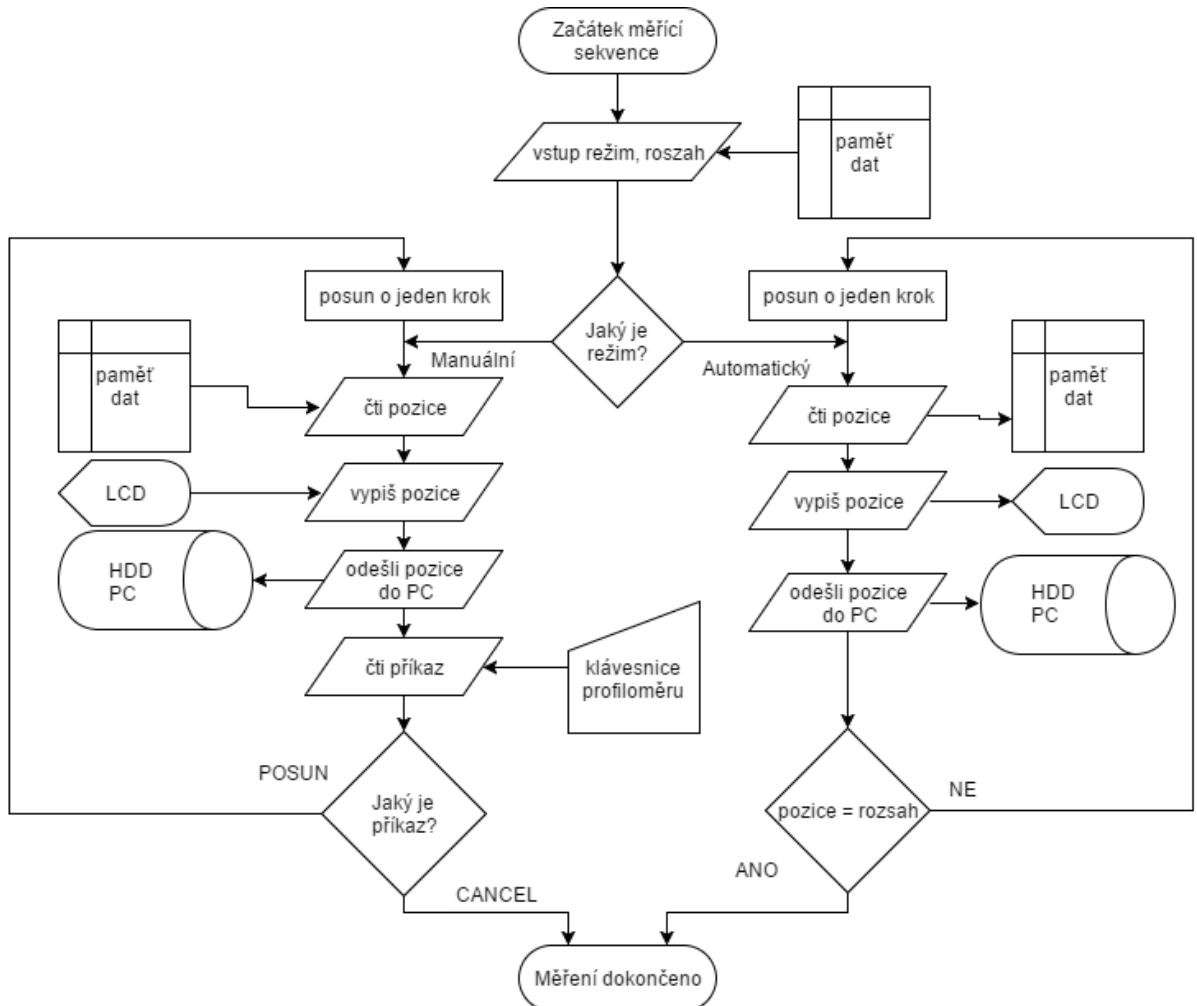
Obr. 4.3 – Algoritmus nastavení parametrů měření

4.2.5 Měřicí sekvence

Měřicí sekvence je hlavním programem profiloměru, v této sekvenci je provedeno samotné měření a jsou odeslány měřené hodnoty do osobního počítače. V závislosti na předchozím nastavení je zde buď ruční posuv pomocí kláves, nebo automatický režim, který sám zajistí měření v definovaném rozsahu. Algoritmus měřicí sekvence je na obr. 4.4.

Během měření jsou neustále odesílány současné polohy jednotlivých snímačů. K odesílání je využívána sériová komunikace, ta využívá piny Tx a Rx a logické úrovně TTL logiky (v tomto případě 5 V). Sériovou komunikaci lze využít ke komunikaci mezi dalšími mikropočítači nebo jako v tomto případě s osobním počítačem. Nastavení sériové komunikace se provádí pomocí příkazu *Serial.begin(rychlost)*, parametrem je pak datová rychlost komunikace. Datová rychlost se udává v bitech za sekundu (jednotka baud), byla zvolena standartní rychlost 9600 baud. Naměřené hodnoty polohy jsou pak odesílány pomocí příkazu

Serial.print(data), tato funkce posílá data jako lehce čitelný ASCII text. Před každým odesláním je odeslán identifikační znak, který spojuje měřená data s příslušnou osou (x nebo y).



Obr. 4.4 – Algoritmus měřicí sekvence

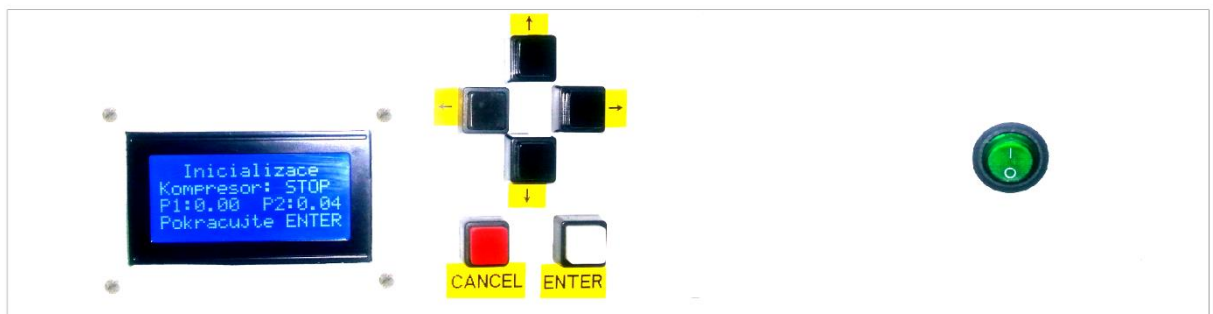
4.2.6 Obsluha přístroje

Ovládací panel je zobrazen na obr. 4.5, zapnutí přístroje se provede pomocí zeleného tlačítka vpravo. Jeho přepnutím do polohy zapnuto dojde k přivedení síťového napětí do všech třech napájecích zdrojů. V tu chvíli začíná běh programu a je možné spustit inicializační sekvenci. V této sekvenci je možné rozběhnou nebo zastavit kompresor pomocí kláves ENTER a CANCEL. Po natlakování na přednastavený tlak a zmáčknutí tlačítka ENTER se uživatel dostane do nastavení parametrů.

V nastavení parametrů si uživatel pomocí šipek vybere režim měření a při zvoleném automatickém režimu si opět pomocí šipek, nastaví požadovaný rozsah měření. Po potvrzení klávesou ENTER je spuštěna měřicí sekvence.

Pokud byl zvolen manuální režim, je v měřicí sekvenci umožněn pohyb pomocí šipek, a to ve směrech, které znázorňují. Pomocí tlačítka CANCEL je možný návrat do nastavení parametrů.

Popis všech sekvencí a navigace v nich je zobrazen na LCD displeji.



Obr. 4.5 – Ovládací panel

5 PROGRAM PRO VYHODNOCENÍ V PC

Pro vyhodnocení dat přijatých z profiloměru, byla vytvořena desktopová aplikace pro osobní počítač Profilemeter reader. Tuto aplikaci bylo možno napsat v různých jazycích, nejběžnější jsou: C++, C# nebo Java. V této práci bude použita právě Java, a to hlavně pro svou přenositelnost, která umožňuje funkci programu pod různými operačními systémy.

5.1 JAVA

Programovací jazyk Java je jedním z nejvíce používaných jazyků na světě. Je velice populární a nejširší uplatnění najde u programů, které vyžadují běh na různých platformách. Od roku 2007 je Java vyvíjena jako open source.

Tento jazyk je objektově orientovaný, to znamená, že výkonný kód je přidružen k datům. Toto přidružení (metody jsou uvnitř objektů) usnadňuje přenos kódu mezi projekty. Snadné propojení zavedlo tzv. dědičnost, která umožňuje, aby odvozené třídy měly stejné atributy jako základní třídy.

5.1.1 Základní vlastnosti Javy

Syntaxe jazyka Java vychází z jazyků C a C++, je jejich upravenou a v mnoha směrech zjednodušenou verzí. Byly zrušeny některé nízkoúrovňové konstrukce a také například ukazatelé, bezznaménkové číselné datové typy nebo příkaz „goto“.

Implementace podpory síťových aplikací je samozřejmostí, Java podporuje různé úrovně síťového připojení, práci se vzdálenými soubory, umožňuje vytvářet distribuované klientské aplikace a servery. Má také vlastnosti, které slouží pro ochranu zařízení v síťovém prostředí, a to před nebezpečnými operacemi nebo napadením systému.

Další předností je její robustnost, je určena pro psaní velice spolehlivého a stabilního softwaru. Aby bylo těchto vlastností dosaženo, jsou zakázány některé programátorské konstrukce, jakou je přímá zpráva paměti, příkaz „goto“ nebo ukazatelé. Tyto příkazy jsou nejčastější příčiny chyb. Pro zvýšení robustnosti musí mít všechny proměnné definovány svůj datový typ.

Pro jednodušší psaní programu je tento jazyk tzv. interpretovaný, to znamená, že místo reálného strojového kódu, uživatel vytváří mezikód. Mezikód je nezávislý na architektuře počítače nebo zařízení. Toto je velkou výhodou, neboť napsaný program může pracovat na libovolném zařízení, na kterém je k dispozici JVM (Java Virtual Machine).

V posledních verzích Javy se pro zrychlení využívá kompilace již před prvním spuštěním programu. Mezikód je tak zkompileován přímo pro dané zařízení a není nutno jej kompilovat za běhu programu. Programu se sice výrazně zvýšila doba startu programu, ale běh programu se významně zrychlil. Další variantou je jakýsi kompromis mezi těmito dvěma způsoby a tím je připravení kompilace jen nepoužívanějších částí kódu. (Oracle, 2017)

5.2 NETBEANS

Pro programování v jazyce Java je vhodné používat vývojové prostředí, nejvyužívanější jsou v současné době Eclipse a NetBeans. Pro tvorbu vyhodnocovacího SW jsem si zvolil NetBeans, a to z důvodu osobních zkušeností s tímto softwarem.

NetBeans je integrované vývojové prostředí za jeho počátkem stojí čeští vývojáři, dnes toto prostředí vlastní firma Oracle. I když firma Oracle financuje z velké části jeho vývoj, je prostředí distribuováno zdarma. Na jeho vývoji se dnes také podílí okolo 800 tisíc přispěvatelů, kteří především upozorňují na chyby a tvoří různé zásuvné moduly, čímž rozšiřují funkce programu. Prostedí samotné je psáno v jazyce Java, což umožňuje běh na různých operačních systémech.

Toto prostředí je primárně určeno pro jazyk Java, ale podporuje i jiné programovací jazyky (C/C++, PHP, Java FX, Java EE, HTML5...). Pro vývoj v Javě NetBeans podporuje různé technologie, základní jsou: grafický debugger, obsluha automatizačních nástrojů (Ant, Maven, Gradle), syntaktická analýza zdrojového kódu a nápověda, zobrazení dokumentace JavaDoc a podpora vývoje na základě standardů definovaných Java Enterprise Edition.

Velmi důležitou pomůckou při tvorbě vyhodnocovacího SW byla také vysoká podpora GUI aplikací. Nástroj pro jejich tvorbu, který prostředí obsahuje, umožňuje sestavit rozličné typy aplikačních a dialogových oken.

5.3 APLIKACE PRO VYHODNOCENÍ

Hlavní požadavky na aplikaci jsou jednoduchost, spolehlivost a rychlost zobrazení. Aplikaci, nazvanou Profilemeter reader, tedy tvoří jedno okno se třemi kartami. Jsou to karty Data, Vizualizace, a Komunikace. První dvě karty mají celkem jasný účel už dle jejich názvu, poslední karta je zde hlavně z hlediska vývoje, aby bylo možno sledovat tok dat na příslušném COM portu. V následujících oddílech budou popsány nejdůležitější funkce a uvedeny ukázky kódů, rozsáhlejší okomentované kódy obsahuje příloha C.

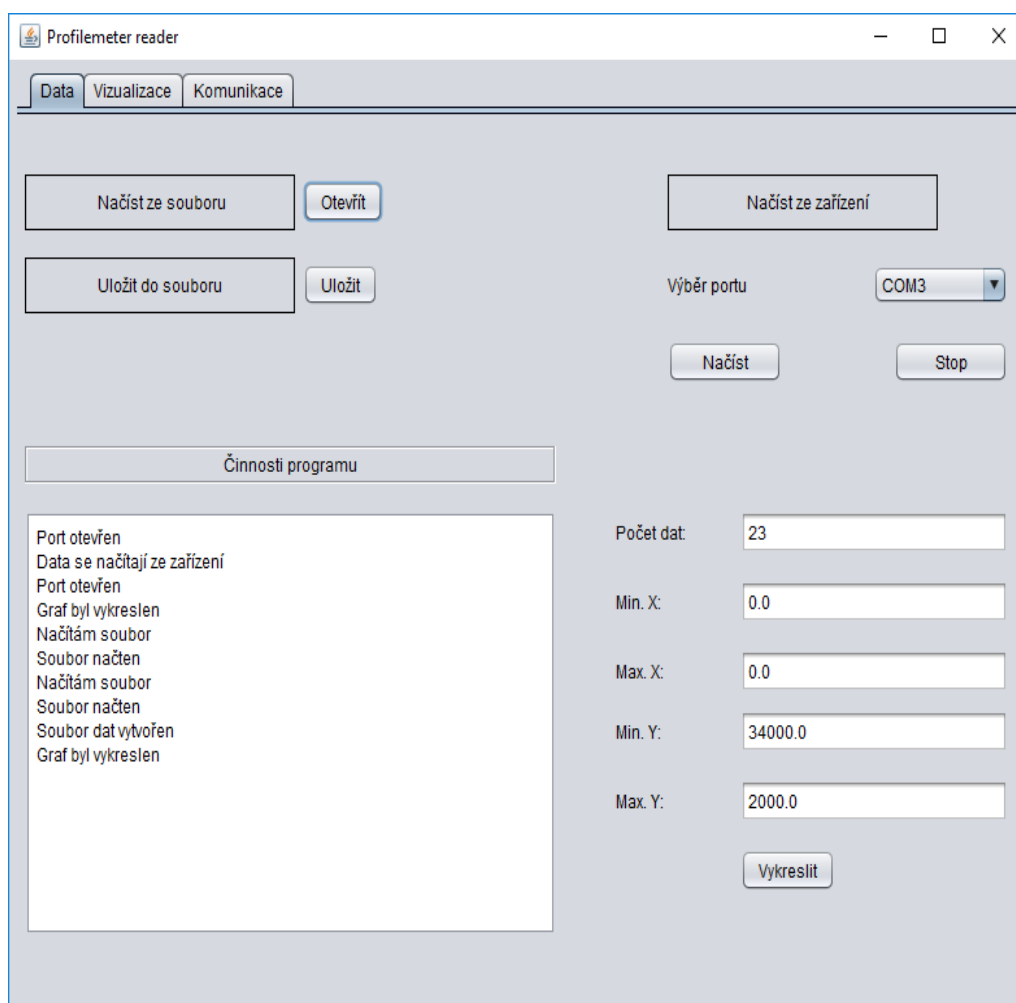
5.3.1 Karta Data

První karta nese název Data (viz obr. 5.1), je vybrána ihned po spuštění programu a jejím účelem je snadné načtení naměřených dat, jak z profiloměru, tak z externího souboru (uložená data z předchozího měření). Pro práci s daty v souborech jsou zde dvě funkce: *Načíst ze souboru* a *Uložit do souboru*. Ke spuštění funkcí slouží příslušná tlačítka *Otevřít/Uložit*.

Pro načtení dat z profiloměru, je nutno vybrat správný sériový port, na který je profiloměr připojen. Po výběru správného portu je možno zahájit načítání dat pomocí tlačítka *Načíst*. Data jsou načítána pomocí několika metod, které budou popsány v samostatném pododdílu 5.3.2.

Další prvky na kartě slouží k informativním účelům, je zde textový panel, ve kterém se vypisují jednotlivé úkony, které program právě dělá. Pak je zde soubor několika základních informací o datech, počet naměřených hodnot a výčet maxima a minima.

Když jsou načtena požadovaná data je možné je pomocí tlačítka *Vykreslit* vynést do grafu, který je umístěn na kartě Vizualizace (viz obr. 5.4).



Obr. 5.1 – Karta Data

5.3.2 Metody pro načtení dat

Na základě vybraného portu a navázaného spojení jsou přijímána data z profiloměru, ten po přepnutí do měřicí sekvence neustále odesílá údaje o poloze pomocí sériové komunikace. Data jsou přijata jako řetězec na obr. 5.2. Identifikační znaky *X* a *Y*, na pozicích 1 a 4, slouží k rozlišení obou os. Informace *poloha_x* a *poloha_y*, na pozicích 2 a 5, jsou celá čísla, která udává počet dílků (impulzů) vygenerovaných optickým snímačem. Znak „D“ (pozice 3) slouží jako dělicí znak, aby bylo možné rozpoznat, kde končí jedna informace a začíná druhá. Na pozici 6 jsou umístěny funkční znaky, které při výpisu tvoří nový řádek a

X	(pozice_x)	D	Y	(pozice_y)	"\n\r"
1	2	3	4	5	6

Obr. 5.2 – Přijatý řetězec

také značí konec řetězce.

Přijaté řetězce jsou plněny do zásobníku, kde čekají na další zpracování. Ve chvíli, kdy program nepřijímá nová data, začínají se zpracovávat data uložená v zásobníku. Zpracovává se vždy poslední přijatý prvek. Nejprve se odstraní znaky, které vytváří nový řádek a poté se řetězec rozdělí na dva samotné, které se uloží do pole a slouží jako vstup pro metodu *pridej_data*. Podrobnější vysvětlení je v komentářích u kódu na obr. 5.3.

Následuje metoda *pridejdata(pole)*, jejím vstupem je pole o dvou prvcích. Každý z nich obsahuje informaci o jedné ose a identifikační znak. Při vstupu do metody jsou znaky odstraněny a obě hodnoty uloženy do pomocných proměnných. Dalším krokem je převod z textového řetězce na číslo, k tomu slouží funkce *parseInt(řetězec)*. Následně je

```
if (!buffer.isEmpty()) { //dotaz na obsah zásobníku
    String stream = ""; //vynulování pomocné proměnné stream
    size = buffer.size() - 1; //zjištění velikosti zásobníku
    stream = buffer.get(size); //uložení posledního řetězce
    JTextArea1.append(stream); //zobrazení řetězce na COM monitoru
    if (nacist) {
        stream = stream.replace("\r", ""); //odstranění new-line znaku
        stream = stream.replace("\n", ""); //odstranění new-line znaku
        xy = stream.split("D"); //rozdělení řetězce pomocí znaku D
        pridej_data(xy); //volání metody pro přidání dat do listu
    }
    buffer.remove(size); //odstranění již zpracovaného řetězce
}
```

Obr. 5.3 – Rozdělení řetězce

zkontrolována duplicita dat. Je porovnáván soubor naměřených dat, respektive poloha osy y, s nově příchozí hodnotou osy y. Jelikož posuv pro jedno měření bude pouze jedním směrem, je pak hodnota v jednom bodě osy y dostatečná a není jich potřeba více, protože by byly totožné. Pokud nejsou data vyhodnocena jako duplicitní, jsou přidány nakonec souboru dat.

Z takto zpracovanými daty se již zachází snadno, pomocí několika metod je tedy spočítáno maximum, minimum pro každou osu a také počet uložených měřených bodů.

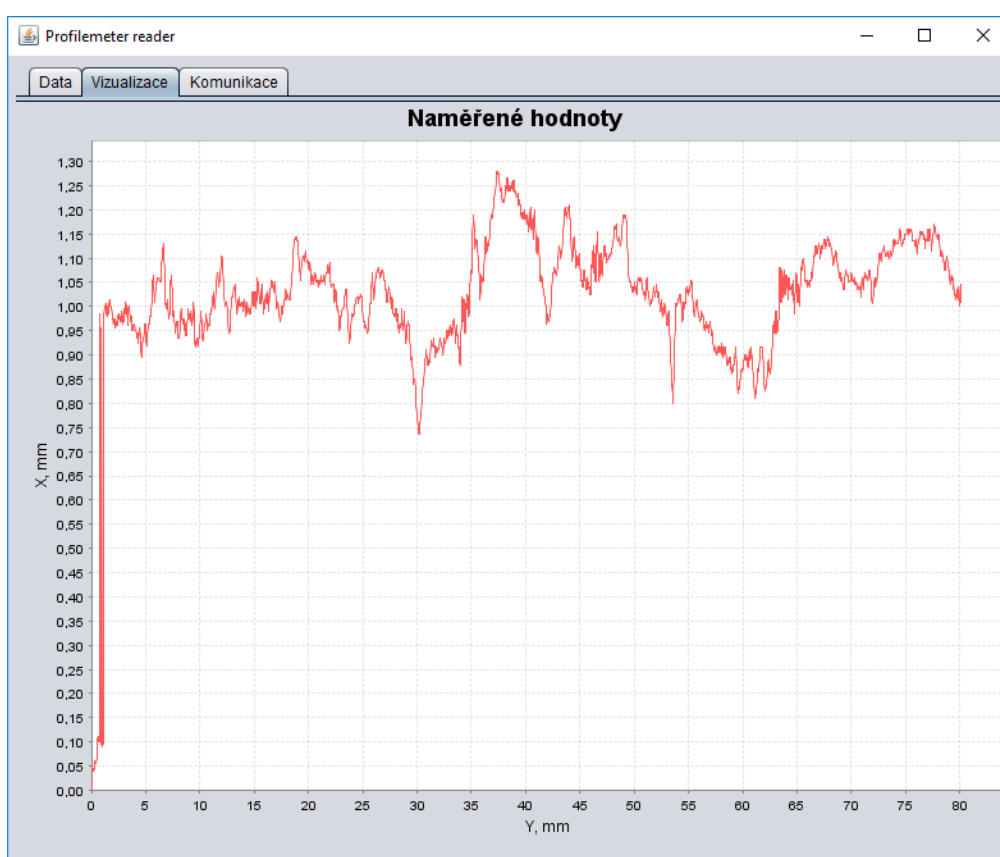
5.3.3 Karta Vizualizace

Pokud uživatel použije tlačítko *Vykreslit* na kartě Data, potom se na kartě Vizualizace vykreslí souřadný systém a v něm naměřené body pospojované lomenou čarou, pro lepší zobrazení snímaného profilu. K vykreslení grafu je použita knihovna JFreechart, která nabízí snadné řešení v oblasti grafů.

JFreechart je knihovna pro programovací jazyk Java, je zcela zdarma pod licenci GNU. Umožňuje snadné řešení na profesionální úrovni. Knihovna podporuje velké množství grafů, jako jsou XY grafy, koláčové, sloupcové nebo Ganttovi digramy.

Aby mohla být data vykreslena, je třeba z nich vytvořit specificky strukturovanou datovou sérii. V tuto chvíli jsou také data přepočtena z počtu impulzů na jednotky milimetrů. Jeden impulz optického snímače znamená posuv o 5 μm , pro přepočet na milimetry je tedy třeba impulzy dělit číslem 200 (jfree.org, 2017).

Pro vykreslení byla použita metoda *creatXYLineChart*, která ze zadaných dat a parametrů vytvoří souřadný systém a do něj vykreslí naměřená data (viz obr. 5.4).

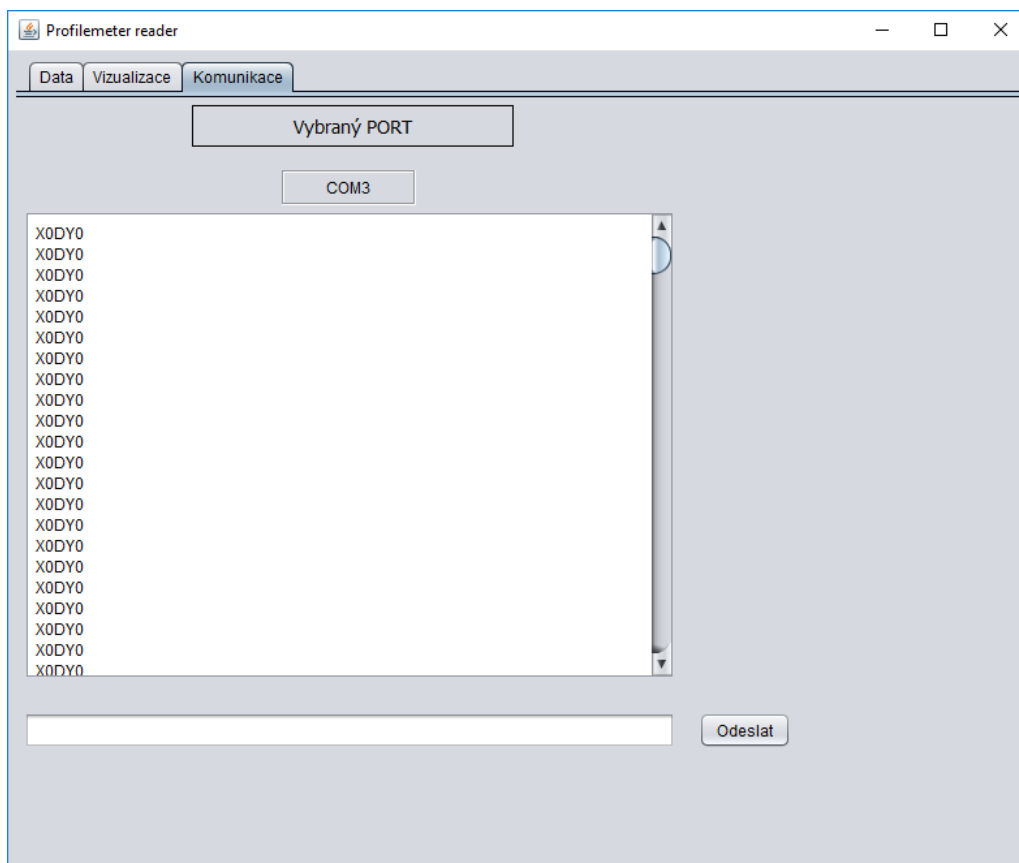


Obr. 5.4 – Karta Vizualizace

5.3.4 Karta Komunikace

Karta na obr. 5.5 je v podstatě jednoduchý monitor sériového portu, je zde vypsán název portu, se kterým se v současné chvíli komunikuje, výpis komunikace, pole pro zadání dat na odeslání a také tlačítko, kterým se data odešlou.

Programová část této karty má několik základních úloh, a to sledovat a zobrazovat příchozí tok dat, a na popud uživatele odeslat zadaný řetězec. Přijímání dat je realizováno pomocí Listeneru. Listener má za úkol informovat v případě nastalé události a také identifikovat k jaké události došlo. V případě sériové komunikace je Listener připojen na sériový port, a ve chvíli kdy jsou k dispozici data pro přijetí, vyvolá událost. V této události se veškerá přijatá data naplní do pomocné proměnné (Sintes, 2017)



Obr. 5.5 – Karta Komunikace

6 KONSTRUKCE

V této kapitole je popsána samotná realizace profiloměru, a to od jeho návrhu (vizualizace) po samotnou konstrukci. Návrh byl proveden ve studentské verzi programu Autodesk Inventor 2017. Konstrukce samotná potom v menší dílně. Při konstrukci bylo třeba využít několik dílenských strojů, jako jsou například soustruh nebo CO₂ svářečka.

6.1 AUTODESK INVENTOR

Vizualizační software Autodesk Inventor je parametrický, adaptivní 3 D modelovací program od firmy Autodesk. Je to nejprodávanější strojírenská 3 D CAD aplikace.

Elementárním prvkem konstruování v Inventoru jsou součásti, tvar součástí může být odvozen pomocí parametrických 2 D náčrtů. Jednotlivé součásti pak mohou tvořit větší celky, součásti jsou kombinovány a vázány různými typy vazeb. Takto nakombinované celky tvoří sestavy. Velikou výhodou je, že pokud dojde ke změně kóty nebo geometrie součásti, je automaticky přegenerována a aktualizována celé soustava, včetně následně vygenerované výkresové dokumentace.

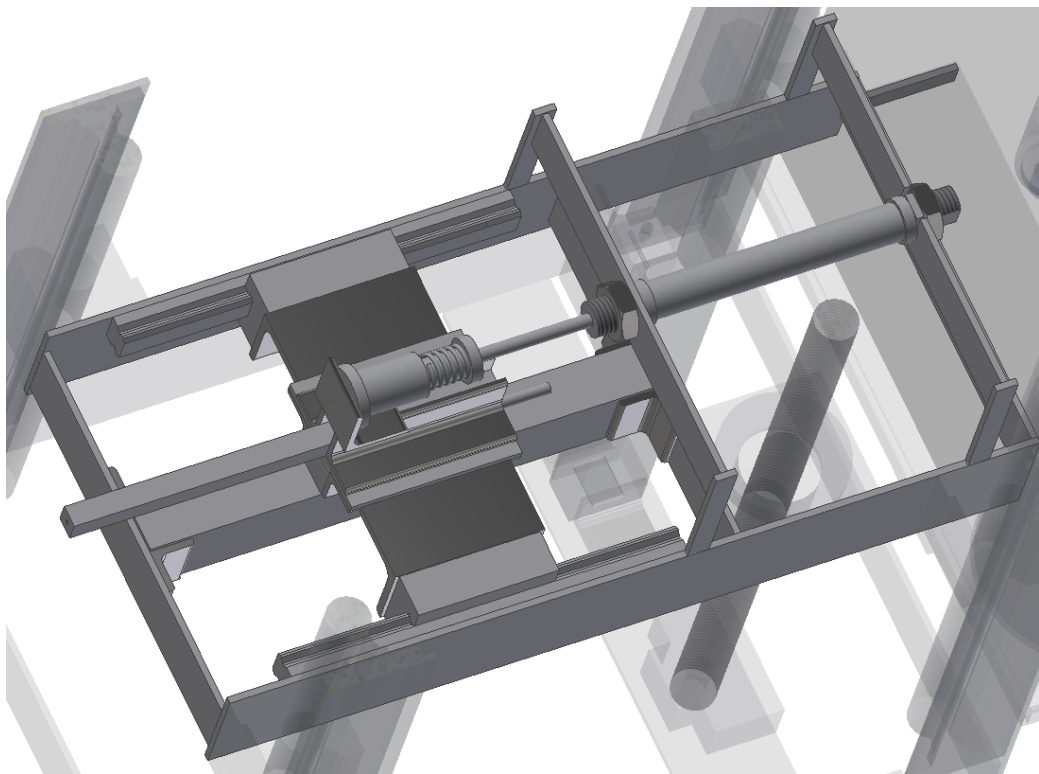
Mezi další nástroje, které Inventor obsahuje, patří rovněž funkce pro modelování plechových částí, svařenců a ocelových konstrukcí. Prostředí také nabízí rendering a pokročilé animace. Integrovaná SQL databáze (Obsahové centrum) nabízí statisíce normalizovaných součástí, což výrazně urychluje a usnadňuje vyvážení sestav (Autodesk, 2017).

6.2 VIZUALIZACE

Před započítím samotné konstrukce bylo třeba vytvořit přesný návrh v již výše zmíněném softwarovém nástroji Autodesk Inventor. Návrh byl proveden ve skutečných rozměrech. Vizualizace zobrazuje nejvíce důležité konstrukční části, jako je hlavní rám, vnitřní rám, vozíky, lineární vedení, vzduchotechnika a rozvaděč. Nezabývá se například šrouby nebo elektronickým vedením (kabely, konektory). Základem návrhu bylo umístění posuvů, dle jejich rozměrů a umístění se odvíjí celá konstrukce. V následujících kapitolách budou popsány posuvy, kompletní vizualizace bude součástí přílohy.

6.2.1 Posuv v horizontální ose

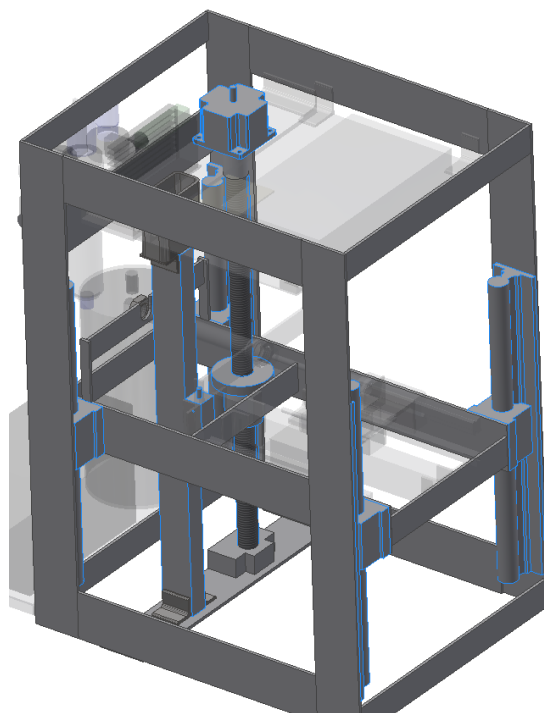
Posuv v horizontální ose (osa x) je umístěn na menším rámu (viz obr. 6.1). Je tvořen dvěma prizmatickými vedeními s vozíky umístěnými naproti sobě uvnitř rámu. Použití dvou vozíků je výhodnější z hlediska větší stability a odolnosti proti pohybu v jiném směru, než je pohyb lineárního vedení. Na vozících je připevněna nosná konstrukce pro měřicí hrot, jeho pohyb a přítlak je zajištěn pomocí pneumatického válce. Přítlak válce je tlumen také pružinou. Zesponu nosné konstrukce je potom připevněn vozík optického snímače vzdálenosti. Díky pružině dlouhé 30 mm byl rozšířen rozsah posuvu. Píst umožňuje výsuv 80 mm, ale rozsah snímače této osy je 100 mm a nebyl by tak efektivně využit. Přidaná pružina nabízí další prodloužení a snímač je pak plně využit. Délka pružiny v plně stlačeném stavu je 12 mm, takže zbývá rezerva, aby nedošlo k mechanickému poškození snímače v případě posunu mimo rozsah.



Obr. 6.1 – Pohled na menší rám

6.2.2 Posuv ve vertikální ose

Vertikální osa (osa y), respektive její posuv, je uložen v hlavním rámu spolu se snímačem (viz obr. 6.2). Posuv je veden po čtyřech podepřených kalených tyčích, jejich umístění tvoří vrcholy obdélníku. Jako pohon slouží krokový motor, jehož rotace je přeměněna na translaci pomocí kuličkového šroubu, jehož matice je připevněna k malému rámu. Rozsah posuvu této osy je 310 mm, což je více, než je rozsah snímače této osy, který je 250 mm. Aby nedošlo k poškození snímače, byl přidán limitní mikrosplínač s ramínkem. Tento mikrosplínač zároveň slouží k nastavení výchozí polohy.



Obr. 6.2 – Pohled na hlavní rám – zvýrazněný posuv osy y

6.3 VLASTNÍ KONSTRUKCE

Konstrukční postup vycházel z vizualizačního návrhu, nejdříve byly svařeny oba rámy a nosná konstrukce pro měřicí hrot. Vše bylo postupně sestavováno v jeden celek spolu se zakoupenými díly. Oproti vizualizaci jsou v konstrukci drobné změny, například zjednodušení držáků optických snímačů. Nebylo třeba je dávat z obou stran snímače, proto jsou jen ze strany jedné. Byly přidány a stabilizační nožičky a malý ovládací panel s displejem.

Aby nedošlo k poškození vodičů připevněných na pohyblivé části, tak byly použity kabelové řetězy, které umožňují bezpečné uložení vodičů a pneumatického potrubí. Vedení

spolu s kuličkovým šroubem a krokovým motorem byly zakoupeny u firmy 4ISP. Firma má dlouholetou praxi v oboru návrhů a prodeje CNC strojů (4isp.cz, 2017).

Při konstrukčním postupu bylo třeba přesného orýsování součástí, k tomu byl použit výškoměr s rýsovací jehlou a rovinná deska. Teplo vzniklé při sváření mělo negativní vliv na konstrukci, některé části se mírně prohuly, a to vedlo k menším výchylkám os. Tyto výchylky byly částečně odstraněny upravením uložení posuvů těchto os.

Konstrukce byla realizována v menší dílně vybavené základním nářadím, stojanovou vrtačkou, svářečkou a soustruhem.

6.3.1 Sváření v ochranné atmosféře

Cílem svařování je vytvořit trvalý nerozebíratelný spoj dvou a více součástí. Požadavkem při svařování je vytvoření takových termodynamických podmínek, aby bylo umožněno vzniknutí nových meziatomárních vazeb. Těchto vazeb je za běžných podmínek obtížné dosáhnout, proto se při svařování působí tlakem nebo teplem. Svařovat lze kovové i nekovové materiály, svařované materiály nemusí mít stejné vlastnosti. Ale typ spoje a typ materiálu ovlivňuje metodu jakou je třeba pro sváření použít. Při svařování vždy musí dojít ke změně fyzikálních nebo mechanických vlastností v okolí spoje. Další metody nerozebíratelného spojování jsou například pájení nebo lepení.

Existuje několik způsobů svařování, způsob použitý při výrobě profiloměru bylo obloukové svařování tavící se elektrodou v ochranné atmosféře. Tento způsob je typem tavného svařování, při kterém se přivádí energie ve formě tepla, v místě svařování se vytváří svárová lázeň. Teplo do místa svařování přivádí elektrický oblouk, hořící v ionizovaném plynu. Problémem při svařování je, že roztavený kov má tendenci reagovat s prvky obsaženými v atmosféře, zejména s kyslíkem a dusíkem. Proto se vytváří ochranná atmosféra pomocí aktivního plynu, nejpoužívanější jsou: oxid uhličitý, argon nebo helium (Brož, 2001).

6.3.2 Soustružení

Soustružením se rozumí třískové obrábění vnitřních nebo vnějších rotačních ploch, většinou pomocí jednobřitého nástroje. Hlavním řezným pohybem při soustružení je rotace obrobku kolem své osy. Vedlejším pohybem se pak rozumí pohyby, které vykonává řezný nástroj. Tento pohyb je podélný posuv ve směru otáčení obrobku.

Stroj, který se k soustružení používá, se nazývá soustruh, obrobek je upnut ve vřetenu, které vykonává rotační pohyb. Pracovní nůž se posouvá a postupně odebírá jednotlivé vrstvy

materiálu. Tím vzniká tříška jako odpad. Podstatná část mechanické energie je přenášena z elektromotoru soustruhu na obrobek, kde se mění na teplo, které zahřívá nůž i obrobek. Aby se zabránilo přehřátí nože, či obrobku používá se vhodná chladicí kapalina k odvodu tepelné energie.

Soustruhem je možné obrábět vnitřní i vnější válcové plochy, zarovnávat čela, provádět zápichy, upichování a řezání závitů. Lze také obrábět kuželové plochy s pomocí dalších nástrojů.

6.3.3 UVEDENÍ DO PROVOZU

Když byla konstrukce dokončena, zařízení bylo připraveno na uvedení do provozu, byly namontovány všechny důležité součásti, připojeny kabely a vzduchotechnika. Před započítím zkušebních měření bylo třeba správně seřídit přítlak. Pomocí siloměru byla změřena síla, která je třeba pro posunutí horizontální osy (x). Tato síla byla průměrně 28 N. Pokud tuto sílu přepočteme na tlak, potřebný pro pneumatický válec, aby tuto sílu vyvinul, dle vztahu (6.1) vyjde tlak 1,6 bar. A ten byl pomocí ručního redukčního ventilu nastaven. Přítlak je tak dostatečně silný na to, aby posunul vozík po vedení, ale zároveň je schopen se zasunout pokud, je na snímaném předmětu část, která vystupuje.

Výpočet tlaku vhodného pro přítlak je dán následujícím vztahem

$$p = \frac{F}{S} = \frac{28}{0,0075^2 \cdot \pi} = 158447 \text{ Pa} \cong 1,6 \text{ bar}, \quad (6.1)$$

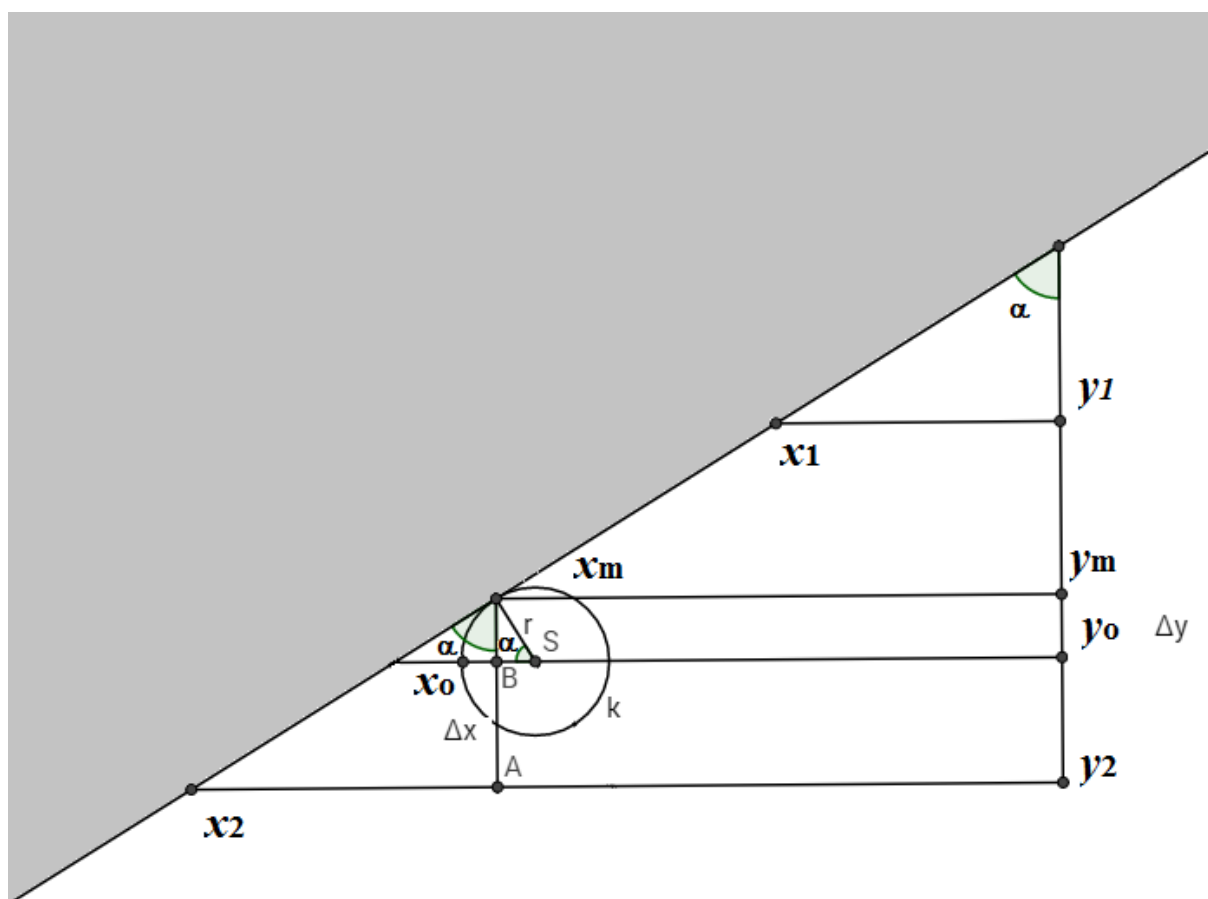
kde p je tlak potřebný pro přítlak,
 F – síla potřebná pro posunutí,
 S – je obsah pístu pneumatického válce.

7 VYHODNOCENÍ

Tato kapitola bude pojednávat o měření pomocí dotyku. Bude provedeno kontrolní měření na předem známém profilu. Dle výsledků bude vyhodnocena přesnost zkonstruovaného profiloměru.

7.1 KOMPENZACE MÍSTA DOTEKU

K měření je používán kuličkový měřicí dotek, jeho nevýhodou je jeho tvar, který v případě jiného tvaru plochy, než je kolmá rovina, způsobuje zkreslení měřených dat. Jeho velkou výhodou je velmi snadné odvalování. Na následujícím obr. 7.1 je náčrtek možného místa dotyku. Jsou zde vyznačeny tři měřené body, s pomocí polohy dvou krajních je možno vypočítat kompenzaci prostředního bodu.



Obr. 7.1 – Kompenzace dotyku

Na základě geometrického zobrazení a měřených bodů, lze sestavit rovnice pro kompenzaci. Ta je založená na podobnosti trojúhelníků, a to konkrétně trojúhelníků ΔX_2AX_m a ΔBSX_m . Jejich úhly α jsou shodné, zároveň tento úhel vyjadřuje sklon snímané roviny. Tento úhel lze snadno vypočítat pomocí dvou krajních bodů, jejich rozdíl tvoří strany trojúhelníka, tudíž pomocí goniometrické funkce arctg lze dopočítat úhel α . Výpočet je změny pozice osy x je

$$\Delta x = x_2 - x_1, \quad (7.1)$$

kde Δx je změna pozice osy x,
 x_1 je horní krajní bod v ose x,
 x_2 je dolní krajní bod v ose x.

Analogicky pak změna pozice osy y je

$$\Delta y = y_2 - y_1, \quad (7.2)$$

kde Δy je změna pozice osy y,
 y_1 je horní krajní bod v ose y,
 y_2 je dolní krajní bod v ose y.

Z geometrického náčrtku vyplývá, že úhel nakloněné roviny lze vypočítat takto

$$\alpha = \arctg\left(\frac{\Delta x}{\Delta y}\right), \quad (7.3)$$

kde α je úhel nakloněné roviny.

K výpočtu korekce měření v ose y můžeme použít vztah

$$e_y = \sin \alpha \cdot r, \quad (7.4)$$

kde e_y je korekce osy y,
 r – poloměr kuličky doteku.

Korekci měření v ose x je potom možné vyjádřit jako

$$e_x = r - \sqrt{r^2 - e_y^2}, \quad (7.5)$$

kde e_x je korekce osy x.

Pomocí vypočítaných korekcí je možno vypočítat skutečnou polohu v závislosti na nakloněné rovině. A to je dáno vztahy

$$x_o = x_m + e_x, \quad (7.6)$$

kde x_o je skutečná poloha měřicího doteku v ose x,
 x_m je naměřená poloha v ose x.

Pro osu y analogicky

$$y_o = y_m + e_y, \quad (7.7)$$

kde y_o je skutečná poloha měřicího doteku v ose y,
 y_m je naměřená poloha v ose y.

Pokud chceme tento výpočet použít pro programovací jazyk, je lepší vše shrnout do jednoho vztahu, pro opravení hodnoty x

$$x_o = x_m + \left(r - \sqrt{r^2 - \left(\sin \left(\arctg \left(\frac{x_2 - x_1}{y_2 - y_1} \right) \right) \cdot r \right)^2} \right) \quad (7.8)$$

a pro hodnotu y

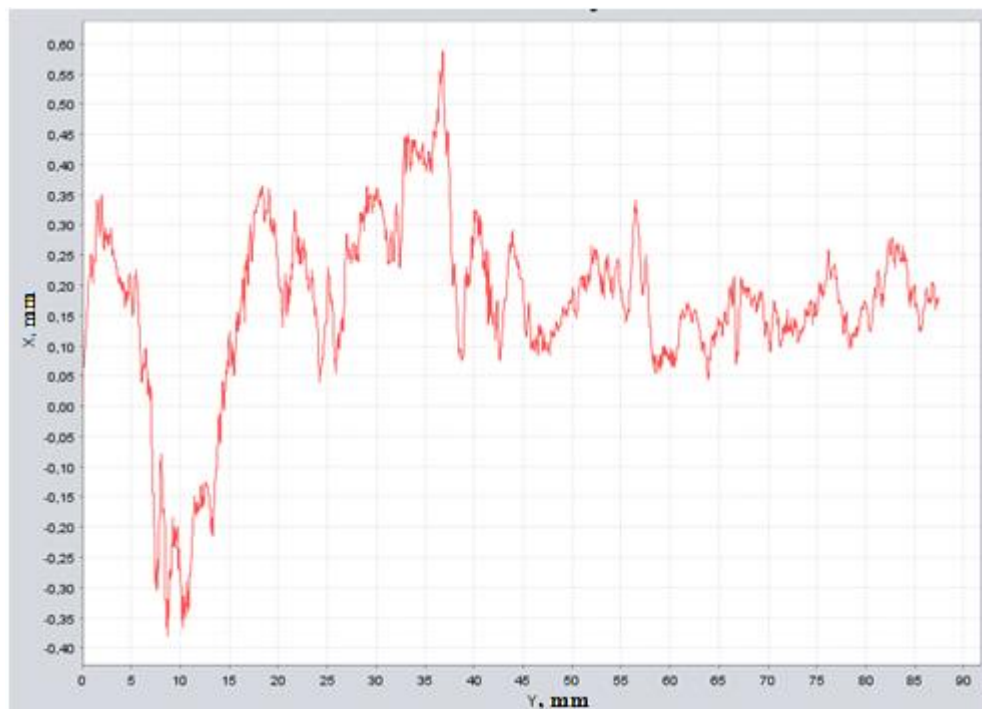
$$y_o = y_m + \sin \left(\arctg \left(\frac{x_2 - x_1}{y_2 - y_1} \right) \right) \cdot r \quad (7.9)$$

7.2 MĚŘENÍ

Pro měření byl vybrán ocelový obrobek s jednou rovnou stranou (přesnost na 0,01 mm), aby mohla být ověřena stabilita měřicího doteku. Byl použit automatický režim a bylo změřeno 90 mm povrchu.

Na obr. 7.2 je osa y v rozsahu 0 mm až 95 mm a ukazuje, jakým způsobem dotek kmitá při snímání, je to způsobeno nedostatečným odvalováním měřicího dotyku po měřeném profilu.

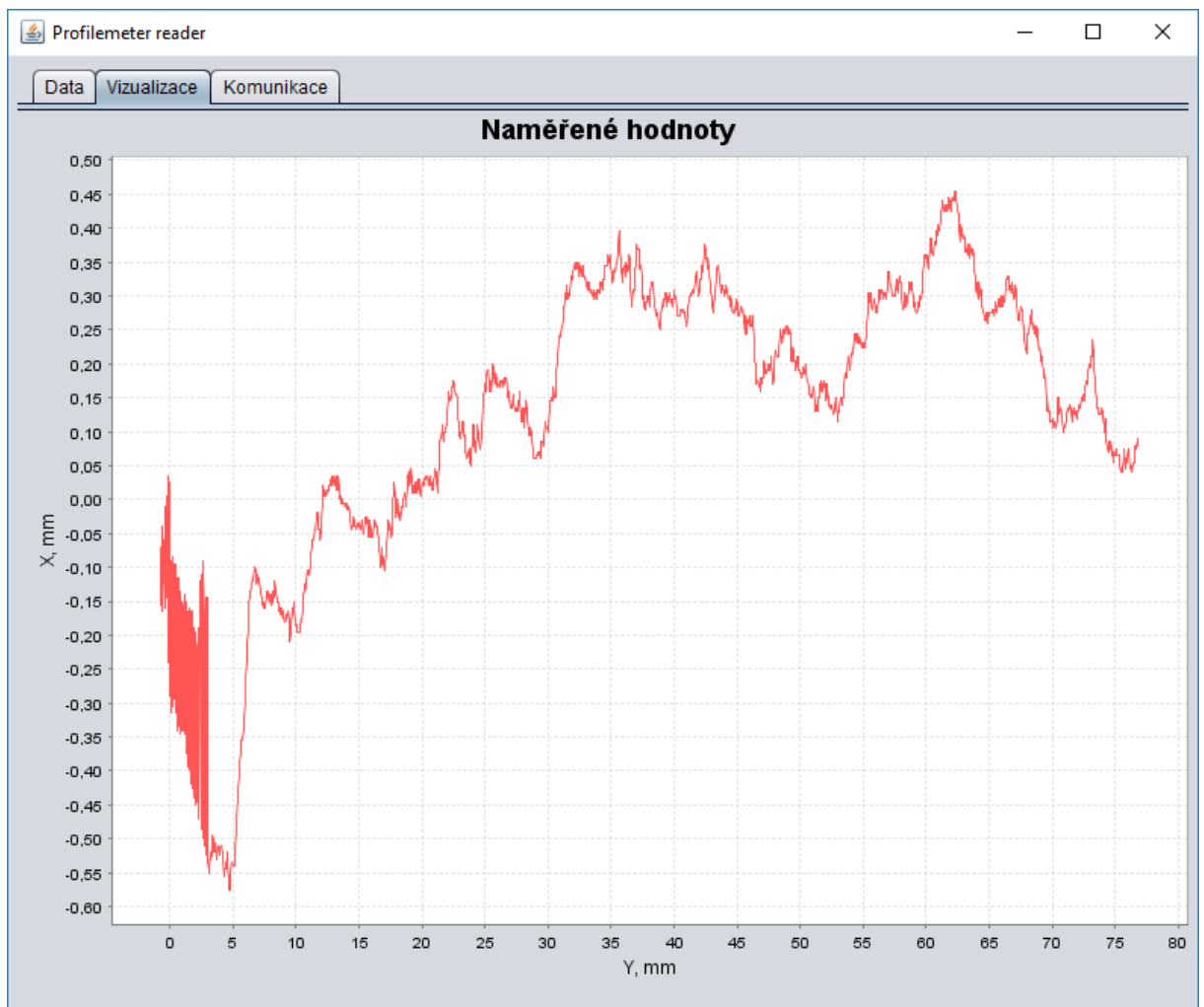
Rozptyl měření je od -0,35 mm do 0,55 mm, takže měření je přesné spíše na jednotky milimetrů. Tento rozptyl je způsoben příliš malou přesností konstrukce. V datech není znatelný trend, odchylky oscilují kolem pomyslné přímky, proto lze usoudit že ustavení předmětu proti profiloměru je dostatečně kolmé.



Obr. 7.2 – Měření rozptylu

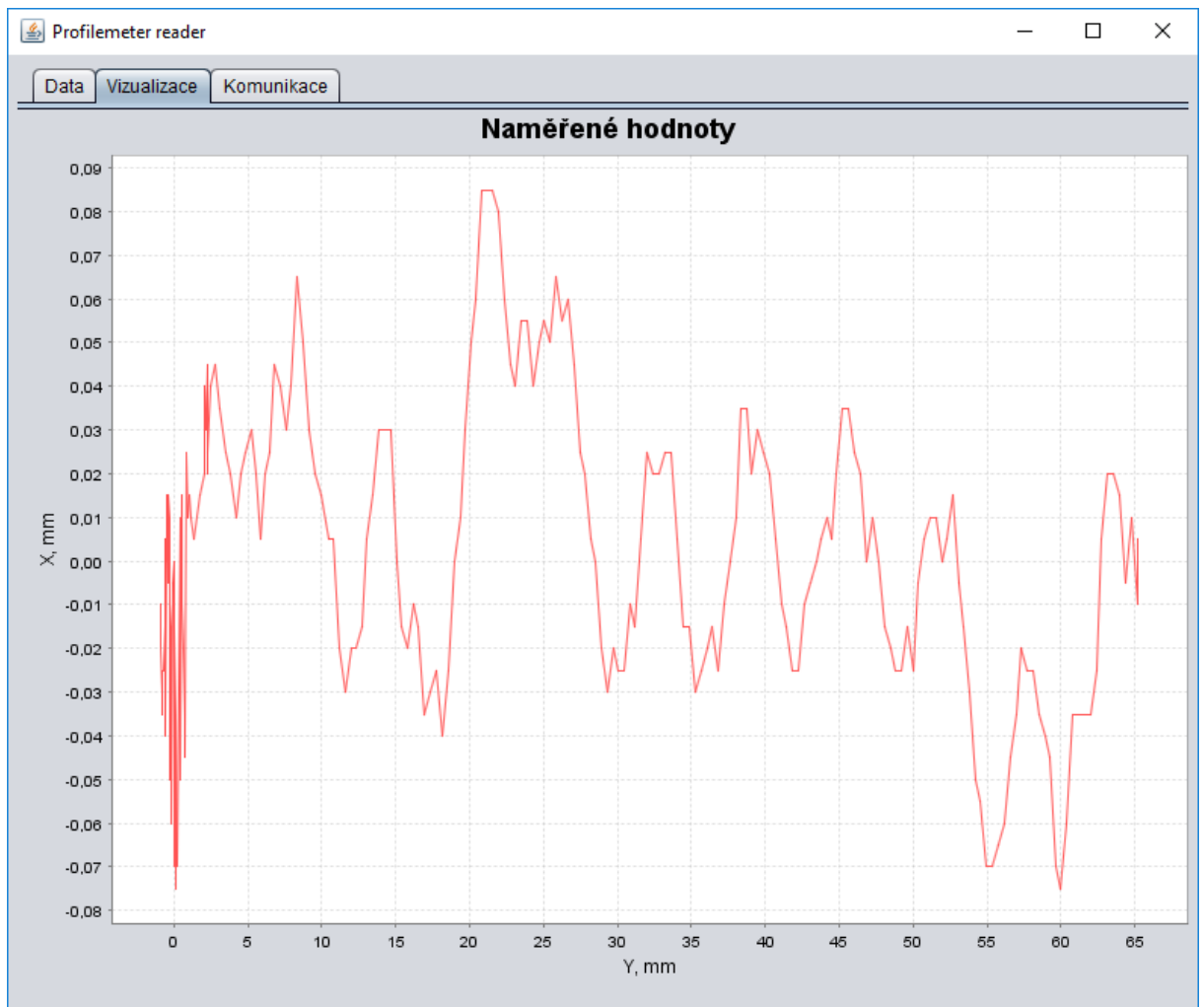
7.2.2 Vliv tlaku na měření

První měření v oddílu 7.2 bylo provedeno s vypočteným přítlakem 1,6 baru. Další měření bude provedeno se zvýšeným přítlakem na 1,8 baru. Na obr. 7.3 je zobrazen výsledek tohoto měření, ze zobrazených hodnot můžeme usoudit, že zvýšení přítlaku o 0,2 baru nemá významný vliv na kolísání.



Obr. 7.3 – Měření s přítlakem 1,8 baru

Při dalším měření byl naopak tlak snížen pod vypočtenou hodnotu a to na 1,4 baru. Výsledky jsou na obr. 7.4. Může se zdát, že kolísání se rapidně snížilo, nicméně takto výrazná změna signalizuje nedostatečný přítlak měřicího doteku. Dotek při tomto tlaku nedokáže kopírovat povrch a není tedy dostatečně přitlačen na měřenou plochu. Zobrazené kolísání je tedy důsledkem vibrací vzniklých rotací krokového motoru.



Obr. 7.4 – Měření s přtlakem 1,6 baru

7.3 VÝSLEDNÉ PARAMETRY

V následující tab. 7.1 jsou popsány výsledné parametry sestrojeného profiloměru.

Tab. 7.1 – Výsledky měření

Parametr	Hodnota
Měřicí rozsah (x, y)	(250×100) mm
Rozlišení	0,005 mm
Přesnost měření	±0,5 mm
Nejvyšší možný sklon snímání	30°
Maximální rychlost polohování	10 mm/s
Výkon kompresoru	75 W
Pohon vertikální osy	Krokový motor Nema 23, 57 mm
Řídící mikropočítač	ATMega2560

8 ZÁVĚR

První část práce pojednává o problematice měření výrobků a jejich částí, jsou zde popsány různé způsoby měření a dostupné přístroje. Práce pokračuje detailním popisem jednotlivých dílů, použitých při stavbě profiloměru. Jsou zde vysvětleny obecné principy vybraných dílů a specifikovány vybrané díly.

Cíle diplomové práce byly splněny. Byl navržen a zkonstruován dotykový profiloměr. Snímání je prováděno ve dvou souřadných osách, a to pomocí lineárních optických snímačů. Posun vertikální osy zajišťuje krokový motor a přítlak horizontální osy vytváří pneumatický válec. Zařízení přenáší naměřená data do osobního počítače pomocí USB portu. Data přijatá osobním počítačem jsou zpracována vytvořeným softwarem Profilemeter reader. Software umožňuje vizualizaci a uložení naměřených dat.

Před započítím konstrukce byl vypracován 3 D návrh pro realizaci dotykového profiloměru, na základě této vizualizace byl profiloměr zkonstruován. Vizualizace byla technickou předlohou pro výrobu všech základních stavebních dílů. Všechny její součásti i kompletní sestava jsou součástí přílohy. Konstrukce byla realizována v menší vybavené dílně, avšak pro reálné použití přístroje nebylo dosaženo dostatečné konstrukční přesnosti. Nejvhodnějším řešením by bylo zadat výrobu základních dílů specializované firmě. Nicméně s ohledem na finanční stránku věci a počáteční nejistoty ohledně funkčnosti celého zařízení, bylo využito vlastních dílenských dovedností vyhovujícím řešením.

Největší problém vytvořilo sváření, vzniklé teplo mělo vysokou tendenci deformovat materiál a způsobovalo tak nemalé problémy. Řešením by mohlo být použití silnějších materiálů a sváření omezit pouze na hlavní nosné rámy. Pokud, by se zbylé díly přidělávaly pomocí šroubů, značně by to omezilo deformace. Tento způsob by vyžadoval kompletní přestavbu profiloměru, protože deformace jsou takřka nevratné.

Dalším problémem, se kterým jsem se při konstrukci setkal, bylo upevnění pneumatického válce a připojení jeho výsuvné části na prizmatické vedení a senzor. Přidaná pružina, která zlepšuje reakci na prudkou změnu tvaru je umístěna na trnu našroubovaném na pneumatickém válci, tento trn je pak umístěn v pouzdru, které je pevně spojeno s vozíkem. Výroba trnu a pouzdra přinesla mnoho nových poznatků o soustružení. Soustružení bylo také využito v pneumatické části, tlaková nádoba měla veliký vnitřní průměr hrdla (31 mm) a bylo ji třeba napojit na rychlospojku pro vzduchotechniku s průměrem M10. Ani specializované firmy nebylo schopny dodat přímou redukci nebo alespoň složit různé redukce, respektive,

pokud bylo možno použít více redukcí, jejich počet byl okolo deseti, což je nepříjemné. Proto byl upraven původní manuální ventil. Což se ukázalo jako nejrychlejší a velice dobré řešení.

Problematika v elektronické části nastala spíše v silové sekci, musel být přidán další zdroj, protože odběr kompresoru byl příliš vysoký. Kompresor byl z ekonomických důvodů použit jednoduchý pístový bez mazání. Absence mazání má za nevýhodu to, že se kompresor při delším používání extrémně zahřívá. Tento problém se dá částečně odstranit občasným manuálním namazáním pístu. Trvalým řešením by bylo pořízení dražšího a kvalitnějšího kompresoru.

Řešení přítlaku se ukázalo jako realizovatelné a do jisté míry použitelné. Jeho spolehlivost by se dala zlepšit zpřesněním celé konstrukce. Pneumatický válec se ukázal jako možné řešení, jeho nevýhodou je určitá nestálost v přítlačné síle. Jednorázové nastavení tlaku manuálním redukčním ventilem se ukazuje jako nedostatečné. Tento problém by se dal vyřešit přidáním tenzometrického snímače, mezi hrot a válec, a výměnou manuálního redukčního ventilu za automatický. Tenzometrický snímač by pak poskytoval zpětnou vazbu a na základě získaných údajů by byl regulován tlak. Toto řešení by opět znamenalo přepracování konstrukce a zakoupení nových dílů. Nicméně pokud by nebylo třeba měřit přesněji, než na desetiny milimetru je toto řešení při přesné konstrukci dostačující.

Pro řízení byla použita vývojová deska Arduino MEGA2560, použití této desky se ukázalo jako vhodné. Nevýhodou by mohla být nedostatečná rychlost mikropočítače, v případě velmi rychlého pohybu obou snímačů, se může stát, že nebude obslouženo přerušení dostatečně rychle a dojde tak ke ztrátě informace o poloze. Tento problém z určité části řeší referenční značky na pravítku uvnitř snímače, ale pokud by došlo k přeskočení přerušení právě ve chvíli přejetí referenčního bodu, došlo by ke ztrátě informace o poloze také.

Vytvořený program v jazyce Arduino splňuje svůj účel, program dobře ovládá všechny součásti a také spolehlivě odesílá naměřená data do osobního počítače. V počítači pak komunikuje s vytvořeným softwarem Profilemeter reader. Tento program psaný v jazyce Java přijímá naměřená data a umožňuje je zpracovat nebo exportovat. Poskytuje také vizualizaci nasnímaného profilu.

POUŽITÁ LITERATURA

- 4isp. 2014. *CO2 lasery, vláknové lasery, laserové gravírky, CNC frézky, CNC stroje a komponenty* [online]. [cit. 2017-05-18]. Dostupné z: <http://cnc.inshop.cz/>
- Autodesk. 2017. *Inventor Overview* [online]. [cit. 2017-15-05]. Dostupné z: <https://www.autodesk.com/products/inventor/overview>
- Arduino. 2017 ARDUINO. *Arduino.cc* [online]. [cit. 2017-15-05]. Dostupné z: <https://www.arduino.cc>
- BROŽ, O.; KANDUS, B.; KUBÍČEK, J. 2001. *Technologie svařování a zařízení: učební texty pro kurzy svářečských inženýrů a technologů*. Ostrava: ZEROSS. ISBN 80-85771-81-0
- Datasheet ATmega2560. 2013. ATMEL CORPORATION. *Atmel.com* [online]. [cit. 2017-15-05]. Dostupné z: http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
- Fejt, R. 2012. *Optický inkrementální senzor* [online]. [cit. 2017-20-05]. Dostupné z: <http://senzory.aspone.cz/ois.htm>
- Freescale Semiconductor, Inc. 2004. *MC9S12XDP512 Device User Guide* [online] [cit. 2017-15-05]. Dostupné z: <http://www.nxp.com/assets/documents/data/en/datasheets/9S12XDP512DGV1.pdf>
- HARČARÍK, M. 2014. *Hodnocení textury povrchu kontaktní a bezkontaktní metodou* [online]. Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů. [cit. 2017-15-05]. Vedoucí práce: R. Jankových. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=84290
- Sensing.honeywell.com. 2017. *I²C Communications with Honeywell Digital Output Force Sensors* [online]. [cit. 2017-20-05]. Dostupné z: https://sensing.honeywell.com/honeywell-sensing-i2c-communications-digital-output-force-technical-note_008291-1-en.pdf
- Jfree.org, 2017. *JFreeChart!* [online]. [cit. 2017-20-05]. Dostupné z: <http://www.jfree.org/jfreechart/>
- KELLER, P. 2013. *Souřadnicové měřicí stroje* [online]. Učební materiál. Liberec: Technická univerzita v Liberci, Katedra výrobních systémů a automatizace. [cit. 2017-15-05]. Dostupné z: <http://www.ksa.tul.cz/getFile/id:2522>
- Ldomotors.com. 2017. *57mm Hybrid Stepper Series* [online]. [cit. 2017-20-05]. Dostupné z: <http://ldomotors.com/products/show/57mm-hybrid-stepper-series>
- Learningaboutelectronics.com. 1999. *LCD – Register select* [online]. [cit. 2017-15-05]. Dostupné z: <http://www.learningaboutelectronics.com/Articles/HD44780-LCD-register-select-RS-pin>
- NXP Semiconductors. 2014. *I²C-bus specification and user manual* [online]. [cit. 2017-15-05]. Dostupné z: http://www.nxp.com/documents/user_manual/UM10204.pdf
- OPAT, J. 2015. *Vybrané metody měření tvaru a deformace ploch a jejich konstrukce* [online]. Bakalářská práce. Praha: České učení technické v Praze, Fakulta stavební, Katedra geodézie a kartografie. [cit. 2017-15-05]. Vedoucí práce: P. Novák. Dostupné z: <http://geo.fsv.cvut.cz/proj/dp/2015/jan-opat-dp-2015.pdf>

- Oracle. 2017. *Java* [online]. [cit. 2017-15-05]. Dostupné z:
<http://www.oracle.com/technetwork/java/javamagazine/index.html>
- PINKER, J. 2004. *Mikroprocesory a mikropočítače*. [cit. 2017-15-05]. Praha: BEN –
technická literatura. ISBN 80-7300-110-1.
- Rajsen Mechatronics. 2017. *Dro Digital Readout System Miling PKG* [online]. [cit. 2017-20-
05]. Dostupné z: <http://www.sinpolinearscale.com/search.html?ss=JCXE>
- SINTES, T. 2000. *Events and listeners* [online]. [cit. 2017-20-05]. Dostupné z:
<http://www.javaworld.com/article/2077351/java-se/events-and-listeners.html>

PŘÍLOHY

A - CD

B - Technický popis

C - Zdrojové kódy

Příloha k diplomové práci

Dotykový profiloměr

Bc. Martin Knížek

CD

Obsah CD

- 1 Text diplomové práce ve formátu PDF
- 2 Kompletní soubory vizualizace dotykového profiloměru
- 3 Úplné zdrojové kódy aplikace pro Arduino MEGA2560 i aplikace Profilemeter reader
- 4 Desky plošných spojů ve formátu PDF

Příloha k diplomové práci

Dotykový profiloměr

Bc. Martin Knížek

TECHNICKÝ POPIS

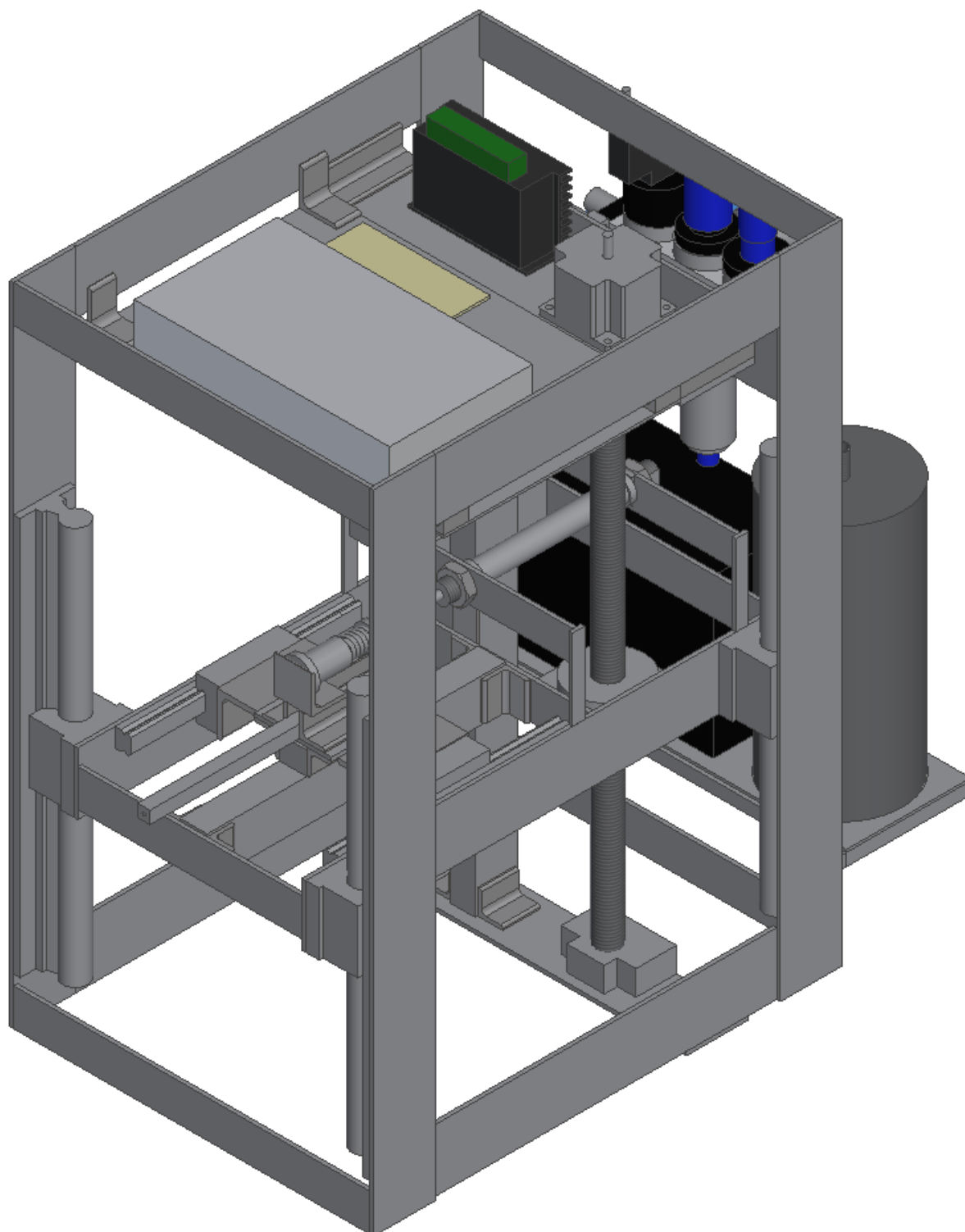
Obsah

	SEZNAM OBRÁZKŮ	2
1	VIZUALIZACE	3
2	DESKY PLOŠNÝCH SPOJŮ	9
3	FOTOGRAFIE REALIZOVANÉHO ZAŘÍZENÍ	12

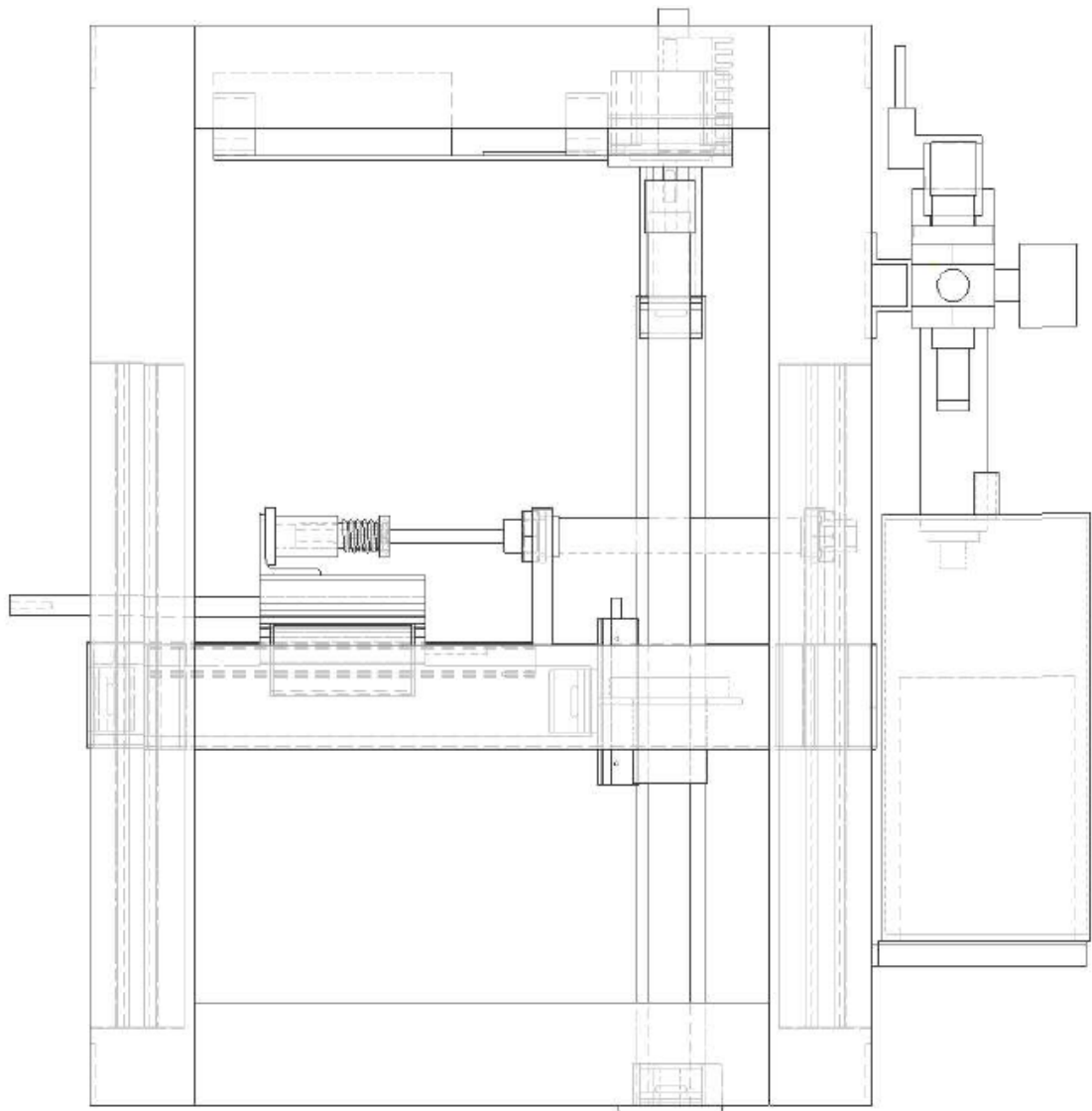
SEZNAM OBRÁZKŮ

Obr. B1.1 – 3 D vizualizace.....	3
Obr. B1.2 – Drátový model, pohled z boku.....	4
Obr. B1.3 – Drátový model, pohled zepředu.....	5
Obr. B1.4 – Drátový model, pohled zezadu.....	6
Obr. B1.5 – Drátový model, pohled shora.....	7
Obr. B1.6 – Drátový model, pohled zespodu.....	8
Obr. B2.1 – Redukce pro Arduino.....	9
Obr. B2.2 – Relé deska.....	9
Obr. B2.3 – Redukce pro LCD.....	10
Obr. B2.4 – Deska pro senzory tlaku.....	10
Obr. B2.5 – Deska pro optické snímače.....	11
Obr. B2.6 – Deska pro driver.....	11
Obr. B3.1 – Realizované zařízení.....	12
Obr. B3.2 – Realizované zařízení, vzduchotechnika.....	13

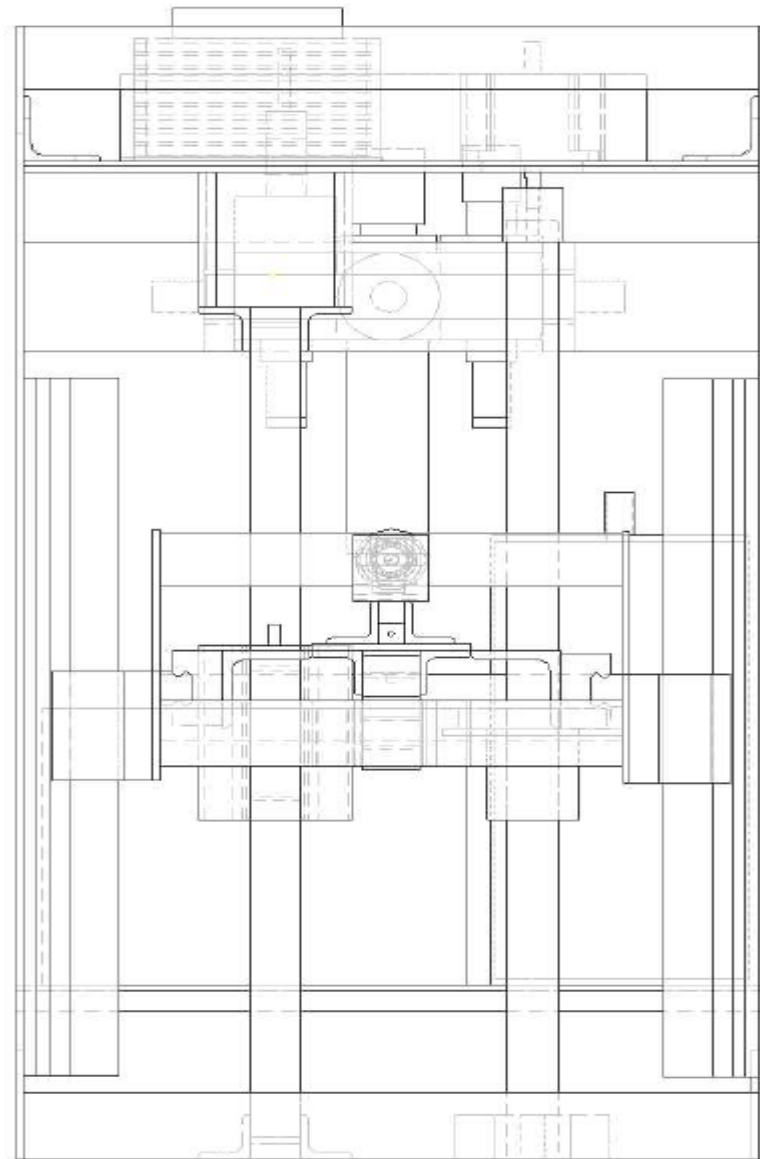
1 VIZUALIZACE



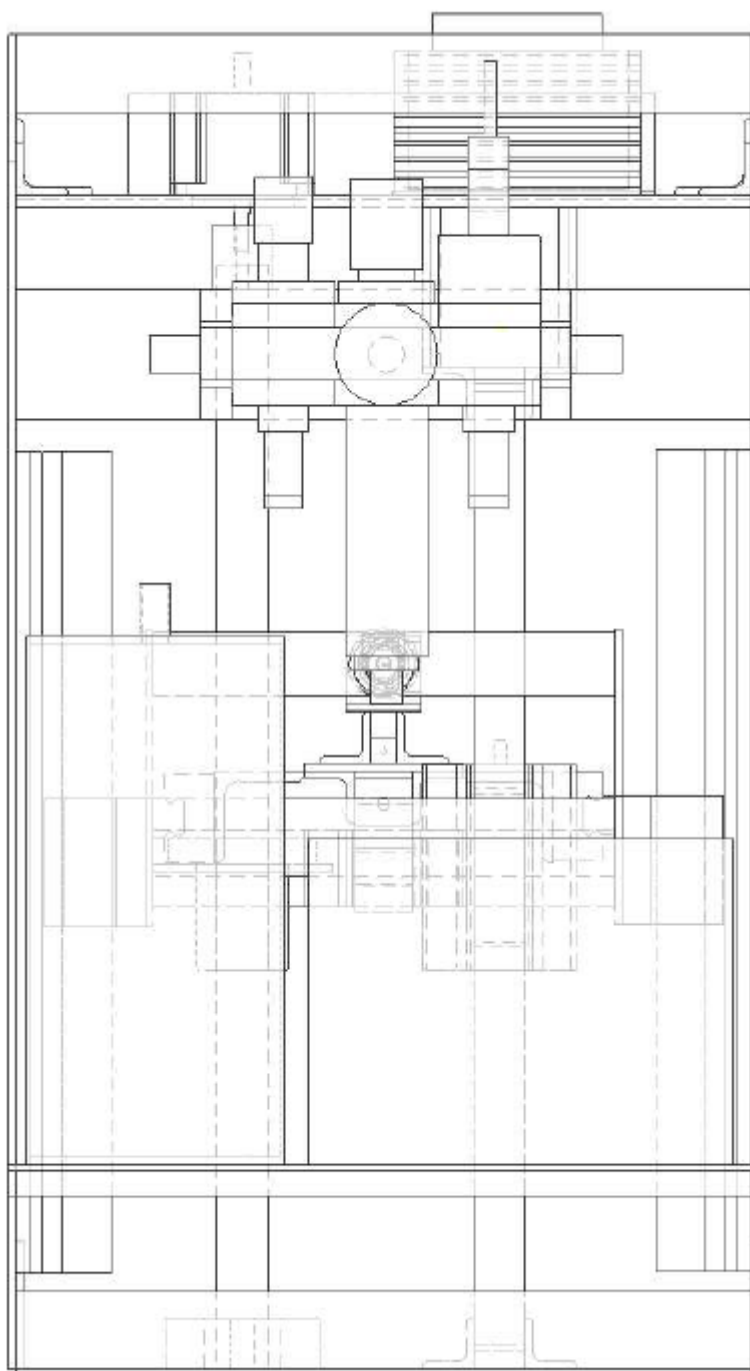
Obr. B1.1 – 3 D vizualizace



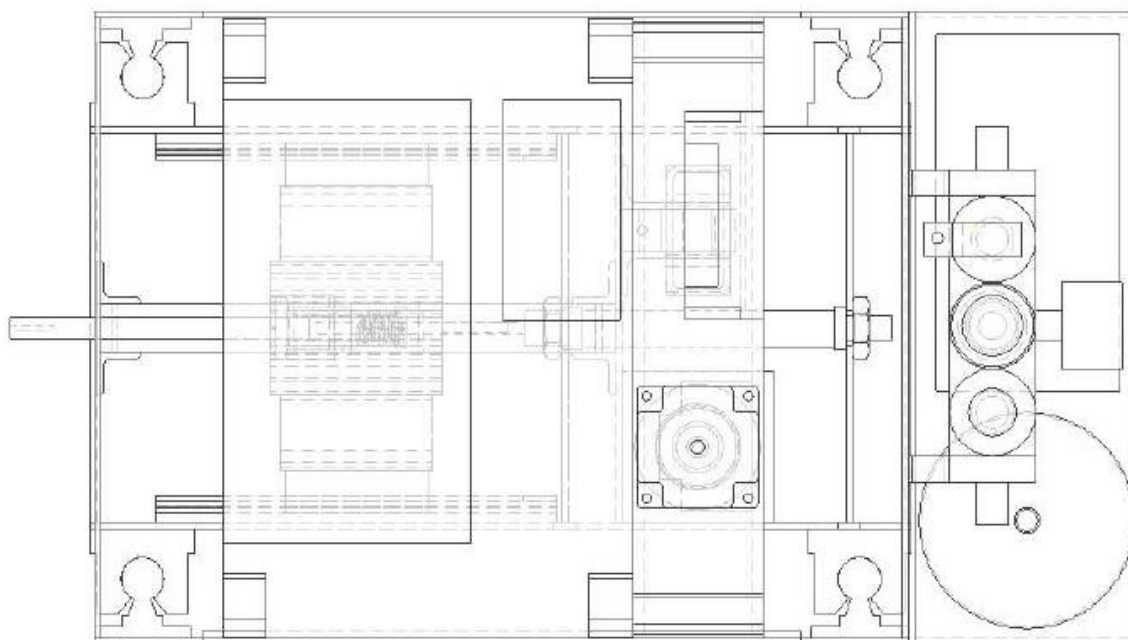
Obr. B1.2 – Drátový model, pohled z boku



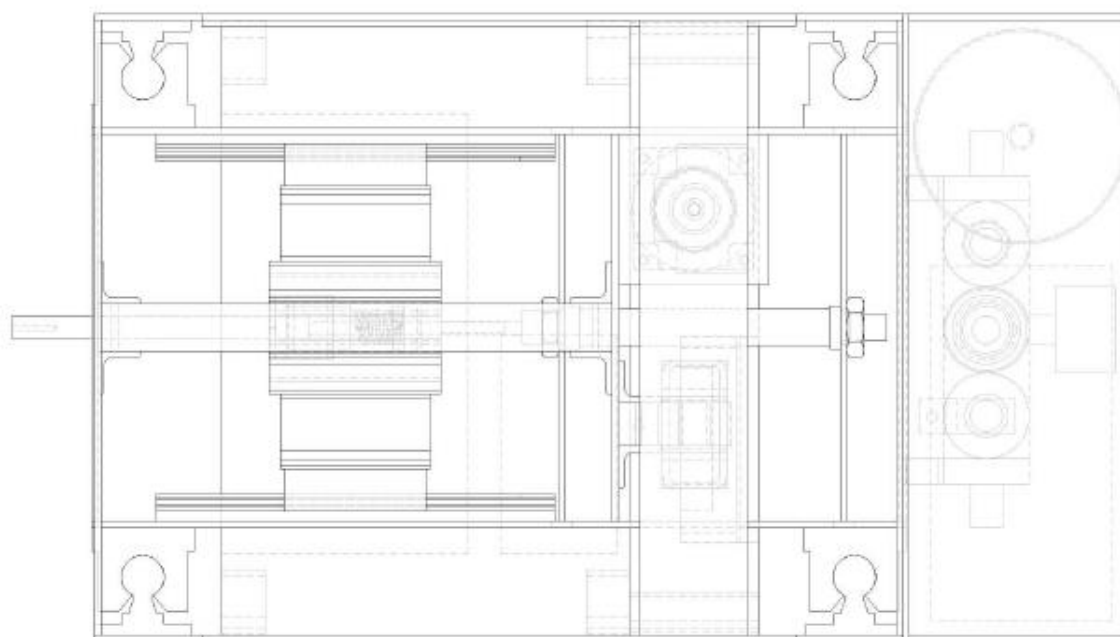
Obr. B1.3 – Drátový model, pohled zepředu



Obr. B1.4 – Drátový model, pohled zezadu

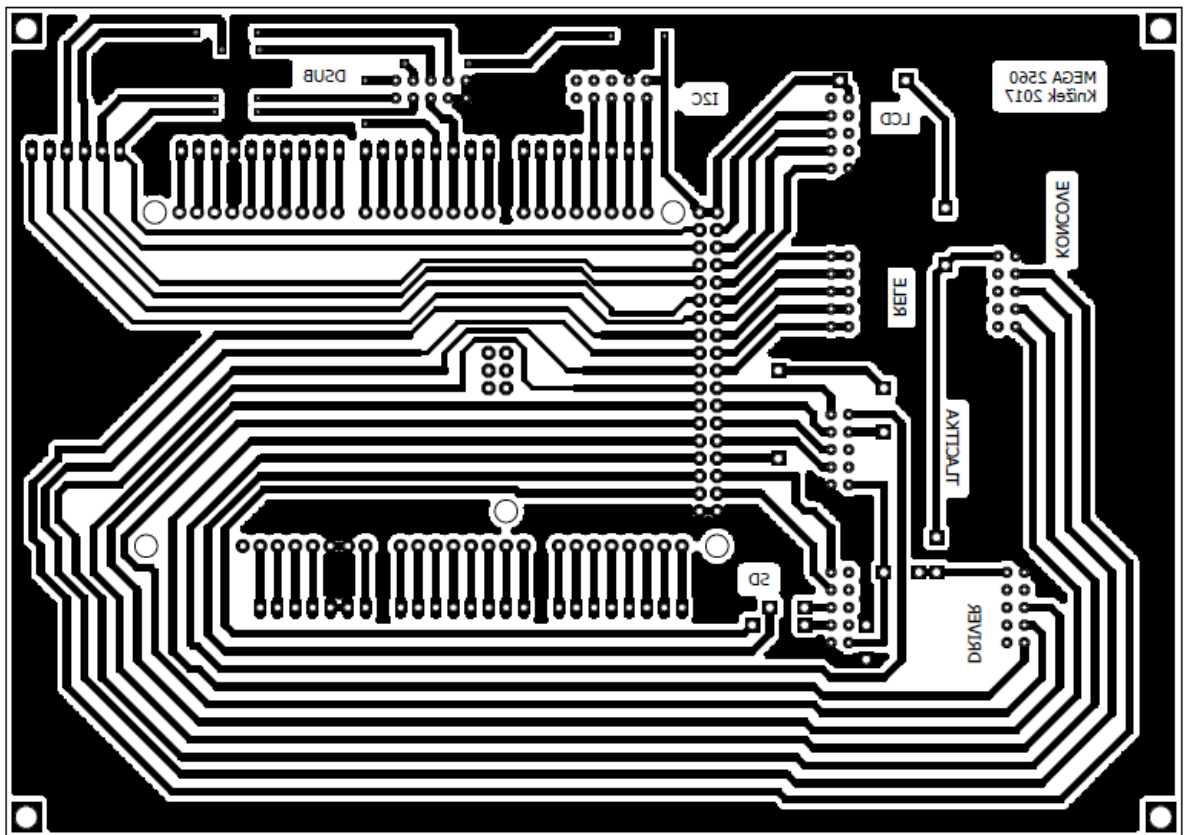


Obr. B1.5 – Drátový model, pohled shora

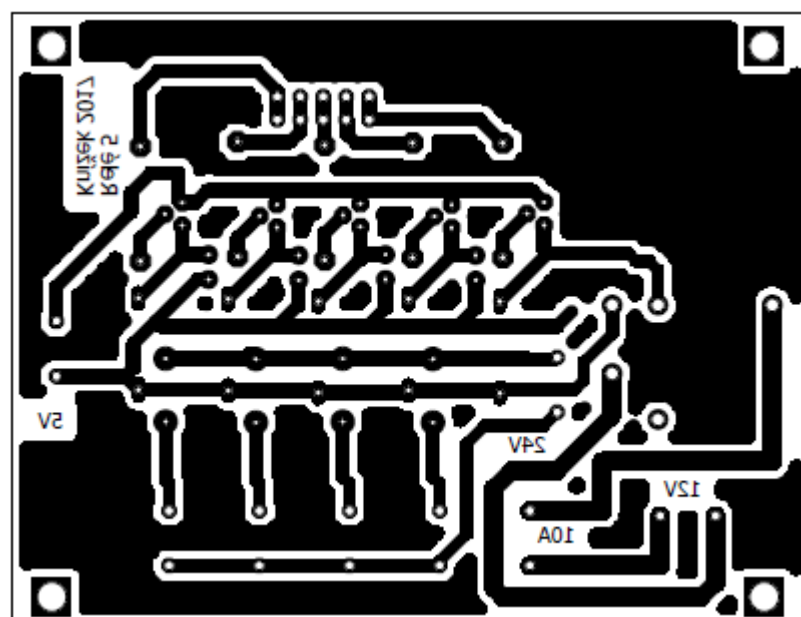


Obr. B1.6 – Drátový model, pohled

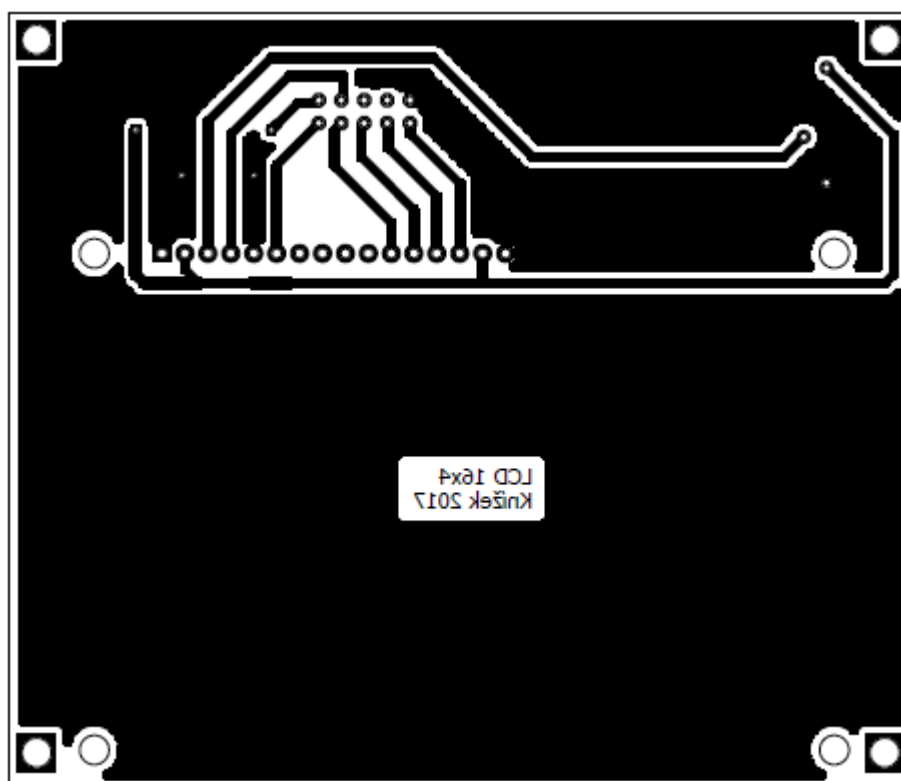
2 DESKY PLOŠNÝCH SPOJŮ



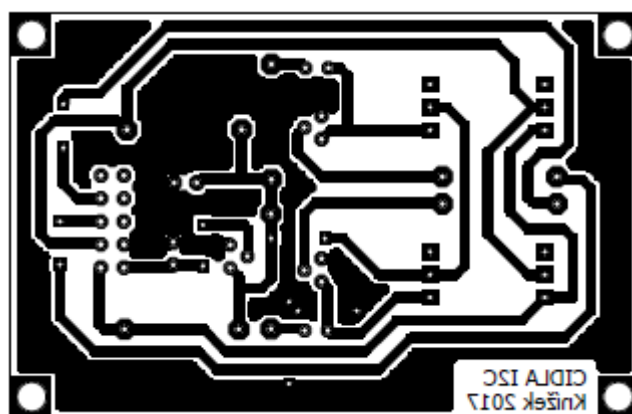
Obr. B2.1 – Redukce pro Arduino



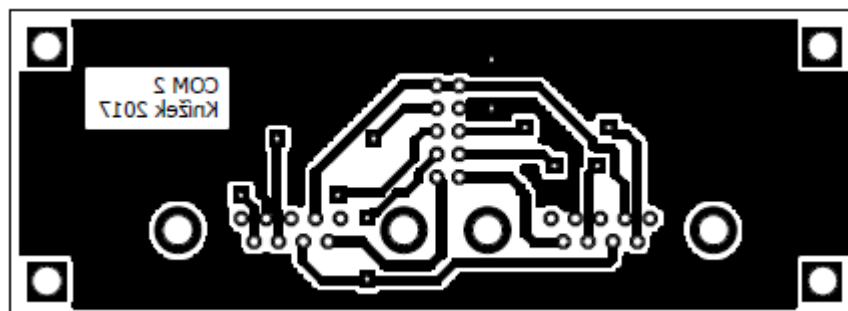
Obr. B2.2 – Relé deska



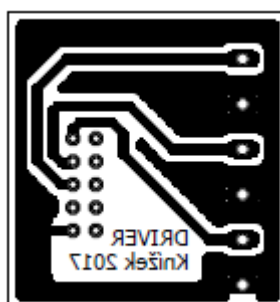
Obr. B2.3 – Redukce pro LCD



Obr. B2.4 – Deska pro senzory tlaku



Obr. B2.5 – Deska pro optické snímače



Obr. B2.6 – Deska pro driver

3 FOTOGRAFIE REALIZOVANÉHO ZAŘÍZENÍ



Obr. B3.1 – Realizované zařízení



Obr. B3.2 – Realizované zařízení, vzduchotechnika

Příloha k diplomové práci

Dotykový profiloměr

Bc. Martin Knížek

ZDROJOVÉ KÓDY

Obsah

	SEZNAM OBRÁZKŮ	2
1	ZDROJOVÉ KÓDY PRO DESKU ARDUINO	3
2	ZDROJOVÉ KÓDY APLIKACE PROFILEMETER READER.....	7

SEZNAM OBRÁZKŮ

Obr. C1.1 – Počáteční nastavení	3
Obr. C1.2 – Metoda pro měření tlaku	4
Obr. C1.3 – Metoda pro automatický režim měření	5
Obr. C1.4 – Obsluhy externích přerušení	5
Obr. C1.5 – Metody pro posuv v ose x a y	6
Obr. C2.1 – Inicializace a obsluha sériové komunikace	7
Obr. C2.2 – Cyklus čtení ze zásobníku	7
Obr. C2.3 – Zpracování přijatých naměřených dat	8
Obr. C2.4 – Metoda pro přípravu naměřených dat	8
Obr. C2.5 – Metoda pro vykreslení grafu	8

1 ZDROJOVÉ KÓDY PRO DESKU ARDUINO

```
void setup() {
  Serial.begin(9600);
  Wire.begin();

  //časovač 1
  Timer1.initialize(10000);
  Timer1.attachInterrupt(tlacitka);

  //tlacitka
  pinMode(41, INPUT);
  pinMode(43, INPUT);
  pinMode(45, INPUT);
  pinMode(47, INPUT);
  pinMode(48, INPUT);
  pinMode(49, INPUT);
  pinMode(40, INPUT);
  pinMode(38, INPUT);
  pinMode(36, INPUT);
  pinMode(34, INPUT);
  tlacitka();

  //piny LCD
  lcd.begin(16, 4);
  lcd.clear();
  lcd.setCursor(0, 0);

  //vypnutí pull-up rezistorů na I2C
  digitalWrite(20,0);
  digitalWrite(21,0);
  //zapnutí 2 čidla
  digitalWrite(18,0);
  digitalWrite(19,1);

  //piny ventilů a kompresoru
  pinMode(vent1, OUTPUT);
  pinMode(vent2, OUTPUT);
  pinMode(vent3, OUTPUT);
  pinMode(komp, OUTPUT);
  digitalWrite(vent1,0);
  digitalWrite(vent2,0);
  digitalWrite(vent3,0);
  digitalWrite(komp,0);
}
```

Obr. C1.1 – Počáteční nastavení

```

void mereni_tlak() {

    byte firstbyte;
    byte secbyte;

    Wire.beginTransaction(addr);
    Wire.write(0);
    Wire.endTransmission();
    Wire.requestFrom(addr, 2);

    while(Wire.available())
    {
        firstbyte = Wire.read();
        secbyte = Wire.read();
    }
    if(cidlo==true){
        digitalWrite(19,0);
        digitalWrite(18,1);
        tlak2 = (8070-float(word((firstbyte<B00111111),secbyte)))/1000*1.013;
    }
    else {
        digitalWrite(18,0);
        digitalWrite(19,1);
        tlak1 = (7500-float(word((firstbyte<B00111111),secbyte)))/1000*1.0273;
    }
    cidlo = !cidlo;
}
}

```

Obr. C1.2 – Metoda pro měření tlaku

```

//měření automat
if(automat){
lcd.clear();
lcd.print(" Auto. rezim ");
posun_x(1);
while(!CANCEL){
  ab = ab+1;
  if (ab == 10){
    ab = 0;
    lcd.setCursor(0, 1);
    lcd.print("X: ");
    lcd.print(float(vzdalenostx)/200);
    lcd.print(" mm ");
    lcd.setCursor(0, 2);
    lcd.print("Y: ");
    lcd.print(float(vzdalenosty)/200);
    lcd.print(" mm ");
    lcd.setCursor(0,3);
    lcd.print("Navrat >CANCEL");
  }
  if (float(vzdalenosty/200)>-(y_auto*10)){
    posun_y(1);
  }else{
    lcd.setCursor(0,0);
    lcd.print("Mereni dokonceno");
  }
}
}
mereni = false;
}
}

```

Obr. C1.3 – Metoda pro automatický režim měření

```

void mereni_vzdalenost1() {
  By = digitalRead(26);
  vzdalenosty = vzdalenosty-1+2*By;
}

void mereni_vzdalenost2() {
  Bx = digitalRead(22);
  vzdalenostx = vzdalenostx-1+2*Bx;
}

```

Obr. C1.4 – Obsluhy externích přerušení

```

void posun_y(int smer) {
  switch(smer) {
    case 0:
      if(ENdY0) {
        digitalWrite(DIR, 0);
        digitalWrite(PUL, 1);
        delayMicroseconds(200);
        digitalWrite(PUL, 0);
      }
      break;
    case 1:
      if(ENdY1) {
        digitalWrite(DIR, 1);
        digitalWrite(PUL, 1);
        delayMicroseconds(200);
        digitalWrite(PUL, 0);
      }
      break;
  }
}

void posun_x(int smer) {
  switch(smer) {
    case 1:
      if(ENdX0) {
        digitalWrite(vent3, 0);
        delay(200);
        digitalWrite(vent2, 1);
      }
      break;
    case 0:
      if(ENdX1) {
        digitalWrite(vent2, 0);
        delay(200);
        digitalWrite(vent3, 1);
      }
      break;
  }
}

```

Obr. C1.5 – Metody pro posuv v ose x a y

2 ZDROJOVÉ KÓDY APLIKACE PROFILEMETER READER

```
//inicializace komponent
initComponents();
//zjistění dostupných portů
cteni_portu();
//pripojení dle vybraného portu
vyber_portu();
//nastavení rychlosti sériové komunikace
serialPort.setBaudRate(9600);
//listener pro sériovou komunikaci
serialPort.addDataListener(new SerialPortDataListener() {
    @Override
    public int getListeningEvents() {
        return SerialPort.LISTENING_EVENT_DATA_AVAILABLE;
    }

    @Override
    public void serialEvent(SerialPortEvent spe) {
        data = serialPort.getInputStream();
        try {
            nacti_data(data);
        } catch (IOException ex) {
            Logger.getLogger(MK.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
});
```

Obr. C2.1 – Inicializace a obsluha sériové komunikace

```
//zpoždění začátku programu aby byla umožněna inicializace desky arduino
Thread.sleep(3000);
int size = 0;
//smyčka pro čtení ze zásobníku sériové komunikace
while (true) {
    if (!buffer.isEmpty()) { //dotaz na obsah zásobníku
        String stream = ""; //vynulování pomocné proměnné stream
        size = buffer.size() - 1; //zjistění velikosti zásobníku
        stream = buffer.get(size); //uložení posledního řetězce
        jTextArea1.append(stream); //zobrazení řetězce na COM monitoru
        if (nacist) {
            stream = stream.replace("\r", ""); //odstranění new-line znaku
            stream = stream.replace("\n", ""); //odstranění new-line znaku
            xy = stream.split("D"); //rozdělení řetězce pomocí znaku D
            pridej_data(xy); //volání metody pro přidání dat do listu
        }
        buffer.remove(size); //odstranění již zpracovaného řetězce
    }
}
```

Obr. C2.2 – Cyklus čtení ze zásobníku

```

private static void pridej_data(String[] xy) {
    //inicilaizace pomocných proměnných
    boolean duplicita = false;
    float[] pom = new float[2];
    //rozdělení vstupních dat
    xy[0] = xy[0].substring(1);
    xy[1] = xy[1].substring(1);
    //konverze ze string na int
    float x = Integer.parseInt(xy[0]);
    float y = Integer.parseInt(xy[1]);
    //ověření duplicity
    for (int i = 1; i < namerena_data.size(); i++) {
        if (namerena_data.get(i)[1] == y) {
            duplicita = true;
        }
    }
    pom[0] = x;
    pom[1] = y;
    if (!duplicita) {
        namerena_data.add(pom);
    }
}

```

Obr. C2.3 – Zpracování přijatých naměřených dat

```

//metoda pro naplnění dat do datasetu
private static XYDataset createDataset() {
    //pojmenování datasetu
    XYSeries datag = new XYSeries("Data");
    for (int i = 0; i < namerena_data.size(); i++) {
        //přidání dat dle os
        datag.add(namerena_data.get(i)[1] / 200, namerena_data.get(i)[0] / 200);
    }
    //vytvoření kolekce dat
    final XYSeriesCollection dataset = new XYSeriesCollection();
    //přidání datasetu do kolekce
    dataset.addSeries(datag);
    //metoda nyní vrátí vytvořená dataset
    return dataset;
}

```

Obr. C2.4 – Metoda pro přípravu naměřených dat


```

//metoda pro vykreslení grafu
private static void vykreslit_graf() {
    //ověření pro nové vykreslení grafu
    if (podruhe) {
        jPanel12.remove(CP);
        jPanel12.revalidate();
    }
    //vytvoření grafu
    JFreeChart xylineChart = ChartFactory.createXYLineChart(
        "Naměřené hodnoty",
        "Y",
        "X",
        createDataset(),
        PlotOrientation.VERTICAL,
        false, true, false);
    //vytvoření panelu s grafem
    CP = new ChartPanel(xylineChart);
    //přidání panelu do panelu Vizualizace
    jPanel12.add(CP, BorderLayout.CENTER);
    //nastavení viditelnosti
    CP.setVisible(true);
    //nastavení velikosti
    CP.setSize(780, 580);
    //aktualizace panelu vizualizace
    jPanel12.validate();
    podruhe = true;
}

```

Obr. C2.5 – Metoda pro vykreslení grafu