



# ŘÍZENÍ TUHOSTI ORTOPEDICKÉ MATRACE

## Diplomová práce

*Studijní program:* N2301 – Strojní inženýrství  
*Studijní obor:* 2301T049 – Výrobní systémy a procesy  
*Autor práce:* **Bc. Tomáš Novotný**  
*Vedoucí práce:* Ing. Radek Votrubec, Ph.D.



## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš Novotný**  
Osobní číslo: **S14000363**  
Studijní program: **N2301 Strojní inženýrství**  
Studijní obor: **Výrobní systémy a procesy**  
Název tématu: **Řízení tuhosti ortopedické matrace**  
Zadávací katedra: **Katedra výrobních systémů a automatizace**

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

1. Seznamte se s pneumatickými prvky s proměnnou tuhostí.
2. Na základě použitých čidel tlaku a ventilů proveďte výběr vyhovující řídicí desky Arduino.
3. Vytvořte potřebné převodníky pro připojení akčních členů na vstupy a výstupy mikrokontroléru.
4. Realizujte řídicí obvod pomocí desky Arduino a naprogramujte algoritmus řízení.

Rozsah grafických prací: **podle potřeby**  
Rozsah pracovní zprávy: **35 stran**  
Forma zpracování diplomové práce: **tištěná/elektronická**  
Seznam odborné literatury:

- [1] Snail Instruments. *Začínáme s Arduinem*. [www.hobbyrobot.cz](http://www.hobbyrobot.cz) [online]. 2014 [cit. 2015-01-09]. Dostupné z: <http://www.snailshop.cz/literatura/1537-zaciname-s-arduinem-prirucka.html>  
[2] Arduino. *Arduino Learning: Getting Started with Arduino*. [online]. 2014 [cit. 2015-01-09]. Dostupné z: <http://arduino.cc/en/Guide/HomePage>  
[3] BALÁTEĚ, J. *Automatické řízení*. 1. vyd. Praha: BEN - technická literatura, 2003. ISBN 978-80-247-4116-1.  
[4] OLEHLA, M., S. NĚMEČEK. *Automatické řízení*. vyd. 1. Liberec: Technická univerzita v Liberci, 2013. ISBN 978-807-3729-721.

Vedoucí diplomové práce: **Ing. Radek Votrubec, Ph.D.**  
Katedra výrobních systémů a automatizace

Datum zadání diplomové práce: **15. listopadu 2015**  
Termín odevzdání diplomové práce: **15. února 2017**

prof. Dr. Ing. Petr Lenfeld  
děkan



Ing. Petr Zelený, Ph.D.  
vedoucí katedry

V Liberci dne 15. listopadu 2015

## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 25.5.2016

Podpis:



## Poděkování

Chtěl bych poděkovat svému vedoucímu diplomové práce Ing. Radku Votrubcovi, Ph.D. za možnost zpracovávat velice zajímavé téma, kterému se tato diplomová práce věnovala. Dále pak za jeho cenné rady, ochotu a vstřícnost v průběhu řešení celé práce.

# Řízení tuhosti ortopedické matrace

## Anotace

Diplomová práce se zabývá možným řešením, návrhem a realizací aktivního řízení tuhosti ortopedické matrace. Cílem bylo vyvinout uživatelsky snadno obsluhovatelný pneumatický systém ovládání tuhosti v klíčových oblastech ramen a hýždí pro matraci Sára od firmy JELÍNEK. Jako řídicí jednotka bylo použito Arduino Micro s vlastním programovacím jazykem na bázi C++ knihovny Wiring.

*Klíčová slova: kvalita spánku, ortopedická matrace, řízení tuhosti, Arduino, programovací jazyk Wiring, stlačený vzduch, regulace tlaku*

# Control of Stiffness of Orthopedic Mattress

## Annotation

The diploma thesis deals with possible solution, design and own realization of active control of stiffness of orthopedic mattress. Goal was to develop user friendly pneumatic system for control stiffness in areas of shoulders and hips for mattress Sara offered by company JELINEK. As control unit was used Arduino Micro which uses own programming language based on C++ library Wiring.

*Keywords: sleep quality, orthopedic mattress, control of stiffness, Arduino, programming language Wiring, compressed air, pressure regulation*

## Obsah

Seznam použitých zkratk a symbolů.....	9
1 ÚVOD .....	10
2 MATRACE A JEJÍ VLIV NA KVALITU SPÁNKU.....	11
2.1 Vliv tvaru těla na volbu matrace .....	11
2.2 Poloha při spánku a její vliv na páteř a rozložení tlaku .....	12
2.2.1 Nejčastější polohy při spánku .....	12
2.2.2 Vliv polohy těla při spánku na tlak od matrace .....	14
2.2.3 Typy matrací .....	15
2.2.4 Ortopedická matrace SÁRA a její využití pro DP .....	17
3 ŘÍDICÍ PRVKY POUŽITÉ PRO REGULACI.....	19
3.1 Řídicí jednotka Arduino .....	19
3.1.1 Historie Arduina .....	19
3.1.2 Typy Arduina .....	20
3.1.3 Přídavné moduly pro Arduino tzv. „shieldy“ .....	24
3.1.4 Programovací jazyk Arduina .....	25
3.1.5 Prostředí pro programování Arduina .....	25
3.1.6 Popis použitých funkcí .....	27
3.2 Obvod pro spínání 12 VDC a regulátor napětí.....	34
3.3 Membránový kompresor .....	37
3.4 Senzor tlaku .....	38
3.5 Solenoidový ventil .....	39
3.6 Ostatní komponenty systému .....	40
4 METODA MĚŘENÍ TUHOSTI .....	42
4.1 Návrh měřicího zařízení.....	42
4.2 Realizace a ověření pohyblivé části .....	44
4.3 Ověření způsobilosti měřidla .....	46
5 SYSTÉM REGULACE TUHOSTI.....	49
5.1 Schéma pneumatického obvodu .....	49
5.2 Schéma elektrického obvodu .....	50
5.3 Řídicí algoritmus a vývojový diagram .....	51
5.4 Popis zdrojového kódu .....	53

6	NÁVRH VARIABILNÍHO ČLENU MATRACE .....	67
6.1	Stanovení cílů .....	67
6.1.1	Tuhost sestavy žlutý nosník + růžová vložka.....	67
6.1.2	Tuhost sestavy žlutý nosník + zelená vložka .....	68
6.1.3	Shrnutí požadovaného cíle .....	70
6.2	Návrh variant řešení.....	70
6.2.1	VARIANTA „A“ – pogumování .....	70
6.2.2	VARIANTA „B“ – Vzduchový vak.....	72
6.2.3	VARIANTA „C“ – Hadice uvnitř molitanové vložky .....	73
6.2.4	VARIANTA „D“ – Hadice skrz nosník (verze 1) .....	75
6.2.5	VARIANTA „E“ – Hadice skrz nosník (verze 2).....	79
6.3	Výsledné srovnání dosažených výsledků .....	83
7	PRAKTICKÁ REALIZACE .....	84
7.1	Nosník s proměnlivou tuhostí.....	84
7.2	Zhotovení boxu pro systém regulace .....	86
7.3	Závěrečný praktický test.....	88
7.3.1	Vyhodnocení praktického testu.....	89
8	ZÁVĚR .....	90
	Seznam použité literatury.....	91
	Seznam obrázků.....	93
	Seznam grafů.....	95
	Seznam tabulek.....	96
	Seznam příloh.....	97
	Obsah přiloženého CD .....	98



## Seznam použitých zkratek a symbolů

USB	univerzální sériová sběrnice
PUR	chemická sloučenina - polyuretan
LED	dioda emitující světlo
IDE	integrovaná elektronika zařízení
I/O	vstupy a výstupy
OEM	originální výrobce zařízení
I <sup>2</sup> C	integrovaná datová sběrnice
SDA	hodinový signál (I <sup>2</sup> C sběrnice)
SCL	datový kanál (I <sup>2</sup> C sběrnice)
ATX	standardizované označení typu počítačových desek
Wiring	programovací jazyk předchůdce Arduina
Arduino	otevřená elektronická prototypová platforma
VDC	stejnoseměrné napětí
KiCad	software pro tvorbu elektronických schémat a layoutů

# 1 ÚVOD

Většina z nás to zná a každodenně zažívá. Shon, stres, vysoké pracovní nasazení, stále se zvyšující nároky okolí, či zaměstnavatele. To vše jsou důsledky moderního věku, ve kterém žijeme. Ať už je naše zaměstnání kancelářského typu nebo se jedná a fyzicky namáhavou pozici, jednu věc máme všichni společnou a tou je potřeba dostatku kvalitního spánku, během kterého tělo regeneruje, uvolňuje napětí, a čerpá síly na následující den.

Vynakládáme nemalé finanční prostředky do aut, moderní elektroniky, exotické dovolené, ale kolik jsme ochotni investovat do kvalitního spánku? Málokdo si uvědomuje, že přibližně třetinu svého života strávíme právě spánkem. A i přes tento fakt velké procento z nás odsouvá investice spojené s kvalitním spánkem až na poslední příčky svých priorit.

Problém nemusí být jen v lhostejnosti. Z osobní zkušenosti znám několik případů, kdy člen rodiny, či blízký známý si za účelem lepšího spánku pořídil drahé několikavrstvé a několikazónové matrace bohužel bez ohledů na to, co je nejvhodnější pro jeho typ postavy. Každý z nás je ve své podstatě unikát, především co se kosterní stavby týče a s ohledem na to by měla být volena i matrace. Trh s matracemi umožňujícími regulaci tuhosti je velice úzký a zaměřený je především na ruční výměnu variabilních částí. Výrobky umožňující regulaci tuhosti pomocí tlaku jsou spíše celoplošně nafukovatelné matrace neberoucí ohled na fyziologii těla.

Diplomová práce se tedy nejprve zabývá matracemi, jejich vlivem na náš spánek, popisuje co se děje s páteří během spánku a jaká by měla být její optimální poloha. Dále je podrobně vysvětlen navrhovaný systém řešení regulace tuhosti, včetně popisu všech členů, řídicí jednotky i samotného algoritmu. Pro objektivní posuzování dosažených výsledků, bylo nutné navrhnout a sestavit zařízení pro měření tuhosti.

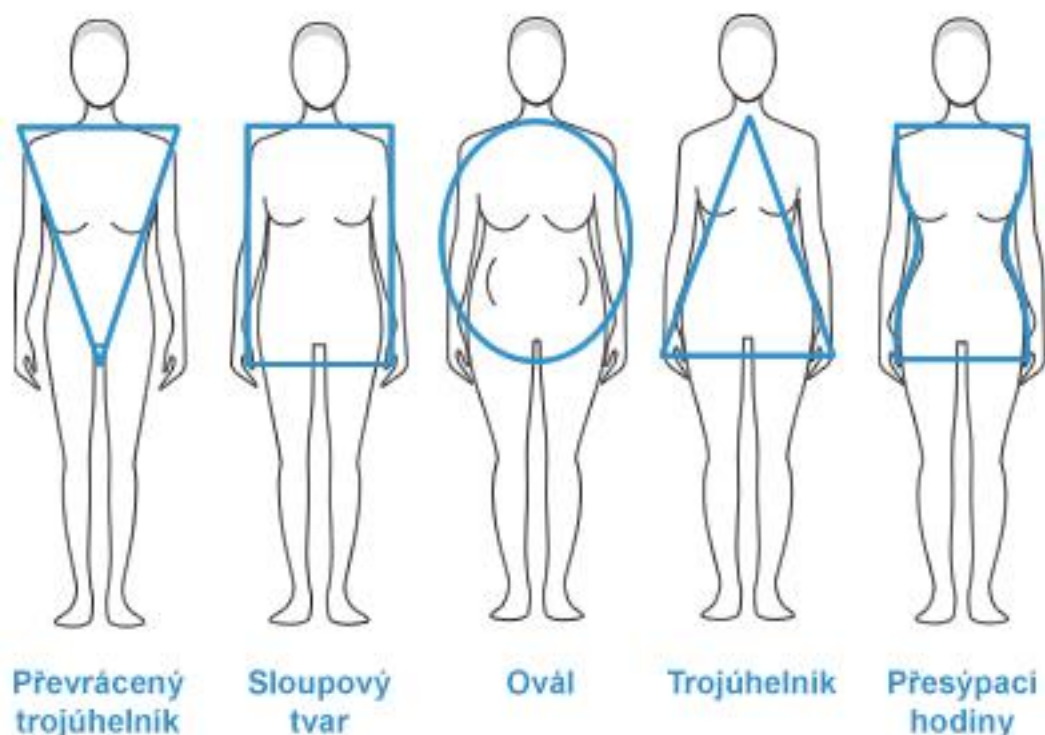
Poté byly navrženy a vyhodnoceny jednotlivé varianty způsobu regulace motanové vložky, nejprve na základě funkčnosti, posléze byla ověřena i jejich účinnost změřením samotné tuhosti.

Cílem diplomové práce bylo navrhnout a zrealizovat způsob uživatelsky snadno ovladatelné změny tuhosti matrace v klíčových místech (ramen a hýždí), tak aby mohla být optimálně nastavena pro jakýkoli typ postavy.

## 2 MATRACE A JEJÍ VLIV NA KVALITU SPÁNKU

### 2.1 Vliv tvaru těla na volbu matrace

Správně zvolená matrace je základem pro kvalitní spánek. Neexistuje nejlepší a nejhorší matrace. Vše se odvíjí od potřeb jednotlivce, způsobu spánku, tělesné výšky, hmotnosti a stavby těla. Genetickou stavbu těla, ale i tvar těla ovlivněný naší životosprávou lze charakterizovat základními mírami ramena / pas / boky.



**Obrázek 2.1 - 1:** Typy postavy

Jak je z obrázku patrné, tvar postavy má pak přímý vliv na zatížení matrace. Zatímco tělesnému typu s širokými rameny a úzkou pánví „převrácený trojúhelník“ bude vyhovovat matrace s měkčí pěnou v oblasti ramen a tužší v oblasti pánve, pro tělesný typ „trojúhelník“ by byla naprosto nevhodná. Měli bychom se snažit dosáhnout ideálního stavu, kdy tlak od matrace na tělo je ve všech místech vyvazován rovnoměrně.

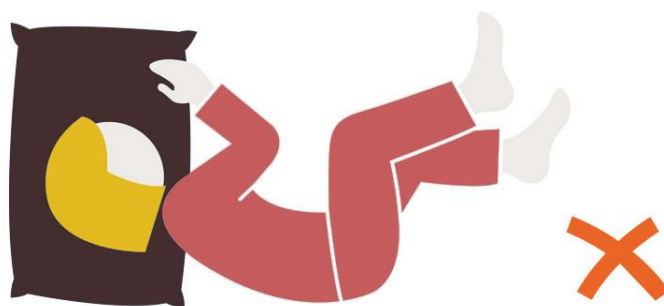
## 2.2 Poloha při spánku a její vliv na páteř a rozložení tlaku

Tlak od matrace, který je vyvozovaný na naše tělo, není ale ovlivněn jen tvarem postavy. Druhý zásadní vliv má i poloha těla při spánku. Platí však jedno zásadní pravidlo. Páteř by měla být v neutrální, vodorovné pozici. Existuje nesčetné množství poloh, které při spánku zaujímáme, přesto je možné je ve většině případů zařadit do čtyř základních kategorií. Seřazeny jsou chronologicky od nejběžnější, bohužel ne nejoptimálnější.

### 2.2.1 Nejčastější polohy při spánku

#### Spánek na boku v klubíčku

Patří mezi nejčastější polohy a je nazývaná také jako poloha „plodu“. Může se zdát pohodlná, ale bohužel je to jedna z nejméně vhodných poloh. Dochází při ní k nepřirozenému zakřivení páteře a nerovnoměrnému tlaku na meziobratlové ploténky, kdy na jedné straně jsou v tenzi a na druhé stlačovány. V důsledku stlačování hrudníku může docházet i ke ztíženému dýchání. Mezi negativní stránky patří u citlivých jedinců i výskyt refluxu (stav, kdy se žaludeční šťávy dostávají do jícnu a způsobující nepříjemné pálení žáhy), zejména při spánku na pravé straně. Výhodou je menší tendence k chrápání a je jednou z mála poloh, při které se pohodlně vyspí i těhotné ženy. Doporučeno je spát především na levém boku z důvodu lepšího prokrvení páteře a prevence proti případnému refluxu.

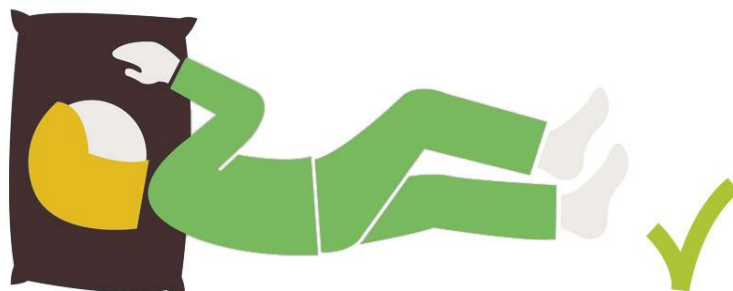


**Obrázek 2.2.1 - 1:** Poloha spánku na boku v klubíčku

#### Spánek na boku v rovné poloze

Může se zdát, že se tato poloha moc neliší od spánku v klubíčku, ale je zde zásadní rozdíl v zakřivení páteře. Pokud je zvolena vhodná matrace, páteř je tak

v přirozené pozici, nedochází ke stlačování meziobratlových plotének a je plně podpořeno jejich vyživování. Měli bychom volit větší adekvátní polštář, tak aby hlava byla v rovnoběžné pozici s páteří, stejně jako u předchozí varianty.



**Obrázek 2.2.1 - 2:** Poloha spánku na boku v rovné poloze

### Spánek na zádech

Není tolik preferovaný, ale jedná se o fyziologicky nejvhodnější pozici pro spánek. Výhody jednoznačně převládají. Při správně zvolné matraci a polštáři je páteř v ideální poloze ve všech rovinách, tělo je symetricky uloženo a tlak od matrace je tak distribuovaný na největší možnou plochu. S lehce podepřenou hlavou zabráníme problémům s refluxem. Ženy ocení i menší tvorbu vrásek a alergikové na roztoče menší problémy s dýcháním z důvodu, že obličej není v kontaktu s polštářem. Nevýhodou je větší tendence k chrápání.

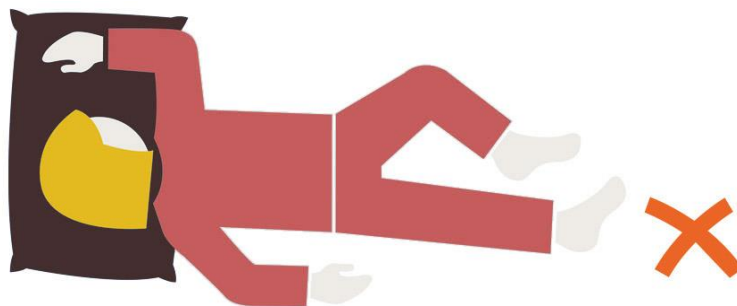


**Obrázek 2.2.1 - 3:** Poloha spánku na zádech

### Spánek na břiše

Patří mezi úplně nejhorší polohu ze všech zmiňovaných a měli bychom se jí vyhnout, nebo alespoň v ní nesetrvávat po delší dobu. Páteř je v krční oblasti nepřírodně zkroucená, což má nepříjemné následky po probuzení jako je zablokovaný krk, bolest krčních svalů, bolest hlavy, ale například i ramenních kloubů. Většina tělesné váhy je koncentrována do místa břicha a není tak možné dosáhnout neutrální

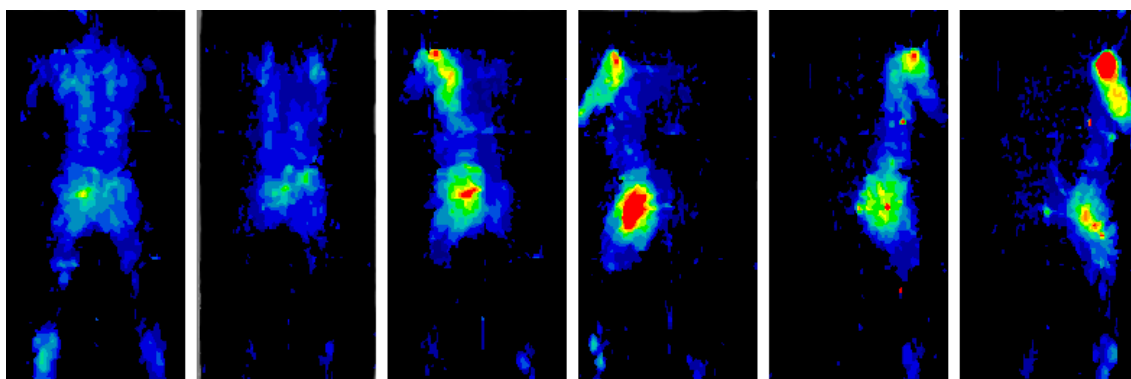
pozice páteře. Většina těchto faktorů vede k častějšímu probouzení během spánku a tím i snížení jeho kvality.



**Obrázek 2.2.1 - 4:** Poloha spánku na zádech

### 2.2.2 Vliv polohy těla při spánku na tlak od matrace

Sílu působící od matrace na lidské tělo je možné i přesně měřit pomocí sítě tenzometrů. Existuje několik firem specializujících se na vývoj takových podložek pro účely měření tlaku u matrací a automobilových sedadel. Důvodů může být hned několik. Komfort uživatele, zatížení, odolnost materiálů nebo rozložení sil z důvodu ergonomie. Počet obsažených tenzometrů se liší podle aplikace, zpravidla od tisíců až po několik desítek tisíc. Zařízení bývá prodáváno včetně vyhodnocovacího softwaru a konektivita je přes běžně dostupné USB rozhraní. Test velikosti zatížení dle polohy při spánku může vypadat následovně.



**Obrázek 2.2.2 - 1:** Poloha spánku na zádech a na boku

Z prvních dvou snímků je patrné, že poloha spánku na zádech značně snižuje tlak vyvozený na oblast ramen a hýždí. U spánku na boku je velice individuální, která část je zatěžována více, viz v předchozí kapitole popsání typy postavy.

### 2.2.3 Typy matrací

Výrobci matrací se na trhu předbíhají, kdo nabídne sofistikovanější provedení, nebo představí nový typ revolučního materiálu. Do jaké míry je to přínos pro zákazníka, či jen marketingový tah, jak přimět spotřebitele k nákupu nové matrace, by mohlo být předmětem zkoumání. Matrace je však možné rozdělit do několika základních skupin.

#### Molitanové

Od molitanových (polyuretanových) matrací s homogenní pěnou v celém svém objemu, tak jak je známe z dřívějších dob, se dnes už upouští. Především kvůli příliš velké tuhosti. Nahrazují je matrace sendvičové, kdy je celek tvořen z více typů pěn o různé tuhosti, případně střední vrstva bývá z pojeného lehčeného polyuretanu (RE pěna), který matraci výrazně zpevňuje a zvyšuje její tvrdost. Výhodou molitanových matrací je jejich cenová dostupnost a možnost na každé straně matrace mít jinou tuhost. Životnost matrace se pohybuje okolo 4 - 5 let.



**Obrázek 2.2.3 - 1:** Sendvičová PUR matrace

#### Pružinové

Elasticita je zajištěna pomocí pružin, které jsou umístěny buďto volně, nebo u dražších typů v jednotlivých taštičkách. Kvalita se odvíjí nejen od počtu pružin a způsobu vinutí drátu, ale také od složení a výšky vrchní vrstvy. Matrace má velice dobré větrací schopnosti a je tak vhodná pro potíci se osoby. Mezi hlavní výhody se řadí i dlouhá životnost. Předností je nízká cena a srovnatelná kvalita s dražšími typy matrací. Pružinové matrace bohužel nemívají odnímatelný potah a směřují se pokládat pouze na pevný podklad, při použití roštu by mohlo dojít k poškození pružin.

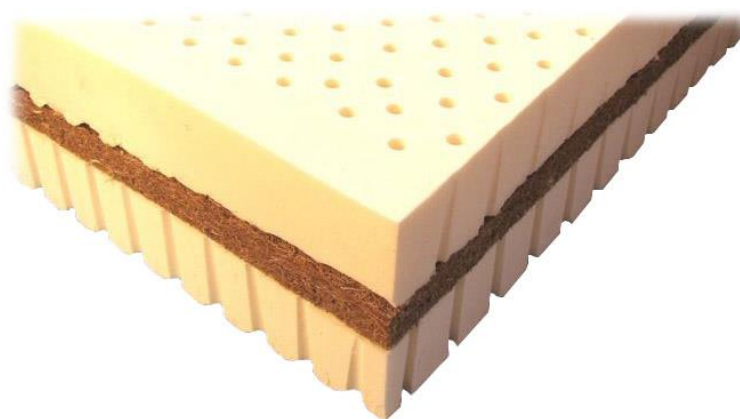




**Obrázek 2.2.3 - 2:** Vlevo matrace pružinová, vpravo soudečková

### Latexové

Matrace z latexu se v současnosti řadí mezi špičku. Jsou vyrobeny z přírodního materiálu a vynikají výbornou bodovou elasticitou, což znamená, že se deformují pouze tam, kde působí síla a nedochází k ovlivnění okolí. Matrace má taktéž výbornou prodyšnost, která je u dražších typů podpořena i středem z kokosového vlákna. Má i velice dobré i antibakteriální a antialergické vlastnosti. Životnost matrace by měla být minimálně 5 – 7 let, nevýhodou je vyšší pořizovací cena.



**Obrázek 2.2.3 - 3:** Latexová matrace s kokosovým jádrem

### Lamelové

Vrchní a spodní část matrace je tvořena lamelovým systémem, mezi který je umístěno jádro ze třech pružných nosníků z PUR pěny. Vrchní lamelová vrstva je dále krytá roznášecí pěnovou nebo latexovou matrací a vše je uzavřeno do snímatelného potahu. Nosníky z pravidla bývají nabízeny ve více tvrdostních variantách dle váhy postav. Navíc, pro dokonalou optimalizaci polohy páteře během spánku,



bývají opatřeny vyměnitelnými vložkami v oblasti ramen a pánve. Jedná se spíše o specializované ortopedické matrace, které nejsou na trhu tolik rozšířené. Na druhou stranu správně nastavená ortopedická matrace splní svůj účel daleko lépe, než většina „univerzálních“ matrací popsaných v předešlé části. Mezi výhody patří již zmiňovaná individuální nastavitelnost tvrdosti, výborné odvětrávací vlastnosti, čímž je vhodná pro alergiky a astmatiky, snímatelný potah a nahraditelnost všech částí systému. Z důvodu, že správně nastavená matrace splňuje veškeré požadavky na tuhost, neměla by být tato matrace kombinována s lamelovým roštem postele.



**Obrázek 2.2.3 - 4:** Ortopedická matrace s lamelovými rošty [6]

#### 2.2.4 Ortopedická matrace SÁRA a její využití pro DP

Před samým začátkem návrhu řízení tuhosti bylo nutné rozhodnout, zda bude využita a modifikována některá z komerčně nabízených matrací, nebo se vydáme cestou kompletního návrhu nové matrace, kde by byla snazší aplikace zvolených metod. Finančně by ale byla mnohem nákladnější. Po pečlivém uvážení došlo ke zvolení varianty modifikace komerčního výrobku a následovalo definování kritérií, které by tato matrace měla splňovat a to především odnímatelný potah a možnost modifikace členů ovlivňujících tuhost soustavy. Při volbě se vycházelo z následující matice.

**Tabulka 2.2.4 – 1:** Matice pro volbu vhodné matrace

	<i>Molitanová matrace</i>	<i>Pružinová matrace</i>	<i>Latexová matrace</i>	<i>Lamelová matrace</i>
<i>Odnímatelný potah</i>	<b>ANO</b>	<b>NE</b>	<b>ANO</b>	<b>ANO</b>
<i>Modifikovatelná tuhost</i>	<b>NE</b>	<b>NE</b>	<b>NE</b>	<b>ANO</b>

Po rozhodnutí jaký typ matrace bude zvolen pro úpravu, bylo nutné vybrat prodejce a konkrétní typ. V předchozí podkapitole byla již zmíněna matrace SÁRA od české rodinné firmy JELÍNEK, která se především díky možnosti a dostupnosti nákupu jednotlivých dílů sestavy jevila jako nejvhodnější. Tělo matrace je složeno ze dvou roštů a třech bavlněných vaků pro vkládání podélných nosníků, kde každý nosník má v oblasti ramen a kyčlí vykrojený otvor pro vyměnitelný člen o různé tuhosti. Podélné PUR nosníky ovlivňující tuhost v celé ploše jsou nabízeny dle váhové kategorie.

- do 60 kg – tmavě šedé nosníky
- 60 – 80 kg – modré nosníky
- 80 – 110 kg – růžové nosníky
- nad 110 kg – oranžové nosníky

Variabilní vyměnitelné členy do nosníků jsou vyráběny ze stejných tvrdostních typů jako nosníky, ale do konkrétního typu nosníky je vždy použita vložka s o stupeň nižší tuhostí.

- do 60 kg – zelená vložky
- 60 – 80 kg – tmavě šedá vložka
- 80 – 110 kg – modrá vložka
- nad 110 kg – růžová vložka

Cílem práce tedy bylo navrhnout takovou možnost regulace tuhosti, abychom zaprvé rozšířili rozmezí, ve kterém je možné tuhost regulovat, tzn. docílit tuhosti menší, než má nejměkčí člen a zároveň větší, než má člen nejtužší. A zadruhé umožnit plynulou regulaci v celém tomto rozsahu.

## 3 ŘÍDICÍ PRVKY POUŽITÉ PRO REGULACI

Při návrhu systému byly použity základní pneumatické prvky, jakými jsou solenoidové ventily, sensor tlaku, propojky, rozbočky a redukce. Po elektronické stránce pak LCD displej, regulátor napětí, optočleny, potenciometr a zdroj, řídicím členem byla zvolena jednotka s mikroprocesorem známá jako Arduino. Byl brán ohled na jednoduchost systému tak, aby nedocházelo ke zvyšování nákladů za prvky, které pro zákazníka nepřidávají hodnotu.

### 3.1 Řídicí jednotka Arduino

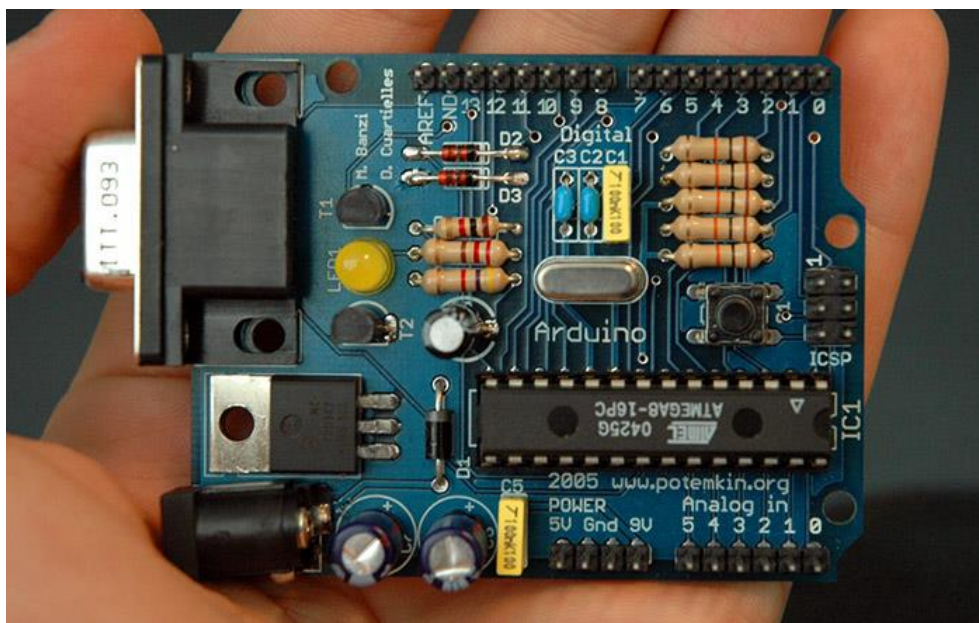
Arduino je otevřená prototypová platforma (open-source) pro každého uživatele, který ji chce používat nebo vylepšovat. Svou oblibu získalo především díky své jednoduchosti po hardwarové i softwarové stránce. Vývojové desky Arduina jsou schopné zpracovávat vstupní informace např. data ze senzorů, přímé zadání uživatele jako je text nebo stisk tlačítka, či třeba email zaslaný přes internet. Výstupem naopak může být sepnutí motoru, LED diody nebo i zaslání SMS zprávy. Konektivita většiny verzí Arduina je přes klasické USB nebo Micro USB port, pomocí kterého je zároveň i napájeno. Je však možné ho napájet i externě, napětím v doporučeném rozsahu 7 – 12 VDC. Například buď použitím 9V baterie nebo běžného zdroje do elektrické zásuvky. Pro programování je využíván speciální, volně distribuovatelný software Arduino Software (IDE), dostupný přímo na stránkách výrobce [www.arduino.cc](http://www.arduino.cc). Oblibu široké veřejnosti si také získalo především díky své kompatibilitě se všemi nejpopulárnějšími operačními systémy jako je Windows, Mac OS X, či 32bit i 64bit verze Linux.

#### 3.1.1 Historie Arduina

Základy pro Arduino položil kolumbijský student Hernando Barragán, který v roce 2004 ve své diplomové práci pod vedením Massima Banziho pro Interaction Design Institute Ivrea v Itálii vyvinul platformu Wiring. Cílem bylo vytvořit nízkonákladové zařízení pro realizování digitálních projektů, kde řídicím členem, pro který byl napsán i zdrojový kód, byl mikroprocesor ATmega128.

O rok později Massimo Banzi a jeho spolupracovníci přidali do platformy Wiring podporu levnějšího mikroprocesoru ATmega8. Nicméně namísto pokračování v práci na jazyce Wiring, okopírovali doposud vyvinutý zdrojový kód a nastavovali separátní projekt Arduino.

Tentýž rok bylo první Arduino i představeno veřejnosti. Cílem bylo vytvořit jednoduchou platformu pro rychlý vývoj prototypů. Projekt zaznamenal velký úspěch a postupně začaly vznikat další modifikované verze. Do února roku 2010 bylo údajně prodáno více jak 120 000 kusů.



**Obrázek 3.1.1 – 1:** První Arduino z roku 2005 osazené procesorem ATmega8 [8]

### 3.1.2 Typy Arduina

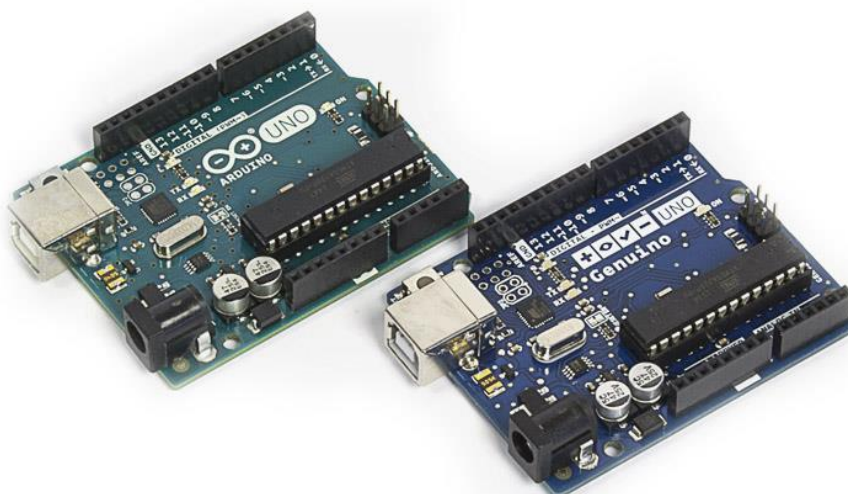
Od představení prvního hromadně vyráběného modelu Arduina uplynulo více jak deset let, a za tu dobu zaznamenalo značný vývoj. Na trhu je aktuálně k dispozici 18 typů, které se liší v celé škále základních parametrů jako je velikost zařízení, konektivita, počet vstupů a výstupů, velikost vnitřní paměti, použitý procesor a další. Následně jsou popsány nejběžněji používaná provedení.

### Arduino UNO

Jedná se nejrozšířenější typ Arduina. Vyráběn je pod dvěma obchodními názvy Arduino UNO (USA verze) a Genuino UNO (mimo USA). Má největší kompatibilitu s přídatnými zařízeními, takzvanými „shields“, které jsou podrobněji popsány v kapitole 3.1.3. Je ideální volbou pro začátečníky, protože má procesor uložený v patci a je tak umožněna jeho snadná výměna v případě, že dojde vinou uživatele k jeho poškození, cena náhradního procesoru se pohybuje okolo 100 Kč. Nevýhodou je malá vnitřní paměť, což omezuje maximální možnou velikost programu nahranou do procesoru, a pokud by bylo potřeba ukládat data, jako jsou obrázky, či audiové záznamy vnitřní paměť 2KB pravděpodobně nebude postačující. Cena za kompletní jednotku se pohybuje od 600 do 700 Kč.

**Tabulka 3.1.2 – 1:** Technické parametry Arduina UNO

<i>Mikroprocesor</i>	ATmega328P	<i>Max. zatížení na jeden I/O</i>	20 mA
<i>Operační napětí</i>	5V	<i>Flash paměť</i>	32 KB
<i>Vstupní napětí</i>	7-12V	<i>SRAM</i>	2 KB
<i>Digitální I/O</i>	14	<i>Takt procesoru</i>	16 MHz
<i>Analogové vstupy</i>	6	<i>Délka</i>	68.6 mm
<i>Analogové výstupy</i>	-	<i>Šířka</i>	52.4 mm



**Obrázek 3.1.2 – 1:** Arduino UNO (vlevo), Genuino UNO (vpravo)

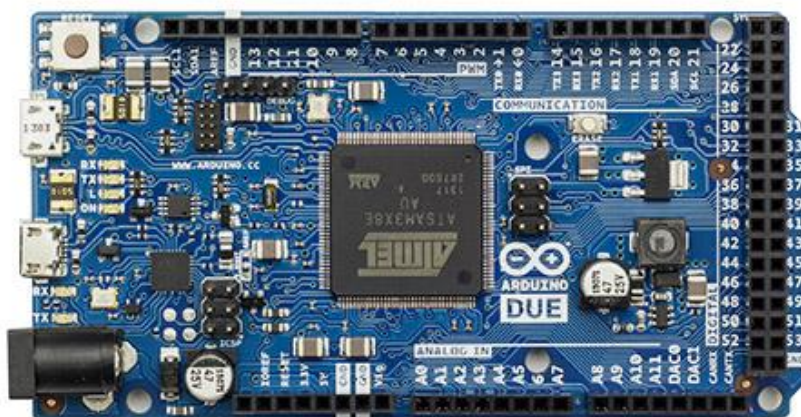


### Arduino Due

Je jedním z nejnovějších modelů rozšiřujících řadu Arduino zaměřený především na náročnější projekty, kde je kladen důraz na dostatečný výkon procesoru, vnitřní paměť a velké množství vstupů a výstupů. V porovnání s verzí UNO je značně větší a téměř dvojnásobně drahý, proto je nutné dobře posoudit, zda takové naddimenzování je pro projekt nutné. Rozdílem oproti většině verzí Arduino je velikost operačního napětí 3.3 voltů, které je rozdílné od běžnějších 5 voltů. Částečně je tím limitována možnost využití přídatných zařízení, které vysílají signál o velikosti 5 V a mohli by tak poškodit procesor Arduino. V takovém případě je vhodné zvolit variantu Arduino Mega. Cena za kompletní jednotku je 1 200 Kč.

**Tabulka 3.1.2 – 2:** Technické parametry Arduino Due

<i>Mikroprocesor</i>	AT91SAM3X8E	<i>Max. zatížení všech I/O</i>	130 mA
<i>Operační napětí</i>	3.3V	<i>Flash paměť</i>	512 KB
<i>Vstupní napětí</i>	7-12V	<i>SRAM</i>	96 KB
<i>Digitální I/O</i>	54	<i>Takt procesoru</i>	84 MHz
<i>Analogové vstupy</i>	12	<i>Délka</i>	101.5 mm
<i>Analogové výstupy</i>	2	<i>Šířka</i>	53.3 mm



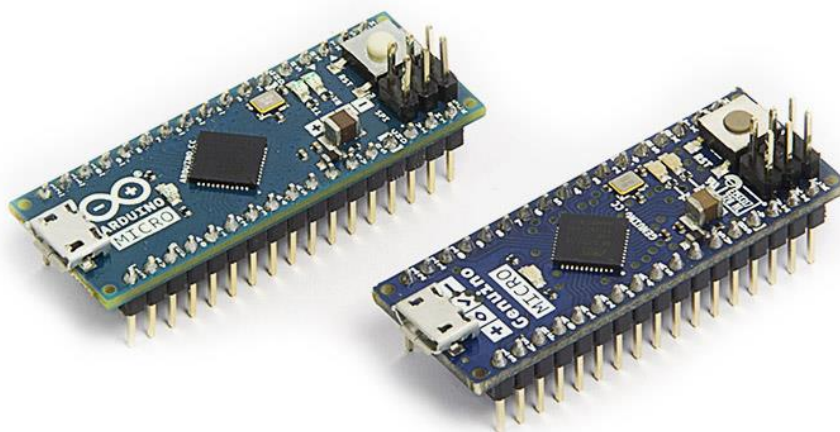
**Obrázek 3.1.2 – 2:** Arduino Due

### Arduino MICRO

Pro projekty, kde jsou základním rozhodujícím faktorem rozměry, je Arduino Micro (případně Nano nebo Mini) ideální volba. I přes své malé rozměry obsahuje úctyhodný počet I/O digitálních pinů a nechybí mu ani funkce dražších a větších modelů. Je tak vhodnou volbou pro většinu projektů. Z důvodu malých rozměrů bohužel není kompatibilní s většinou přídatných zařízení „shieldů“, ale na druhou stranu je oproti jiným variantám osazený samčími piny a je tak možné celou desku usadit do nepájivého pole. Stejně jako u varianty UNO platí, že pro USA je použito obchodní označení Arduino MICRO a pro ostatní státy Genuino MICRO. Ze zvyš zmiňovaných důvodů, především vlastní velikosti a možnosti usazení do nepájivého pole, bylo Arduino MICRO zvoleno pro projekt této diplomové práce. Cena za kompletní jednotku je 600 Kč.

**Tabulka 3.1.2 – 3:** Technické parametry Arduina MICRO

<i>Mikroprocesor</i>	ATmega32U4	<i>Max. zatížení na jeden I/O</i>	20 mA
<i>Operační napětí</i>	5V	<i>Flash paměť</i>	32 KB
<i>Vstupní napětí</i>	7-12V	<i>SRAM</i>	2.5 KB
<i>Digitální I/O</i>	20	<i>Takt procesoru</i>	16 MHz
<i>Analogové vstupy</i>	12	<i>Délka</i>	48 mm
<i>Analogové výstupy</i>	-	<i>Šířka</i>	18 mm



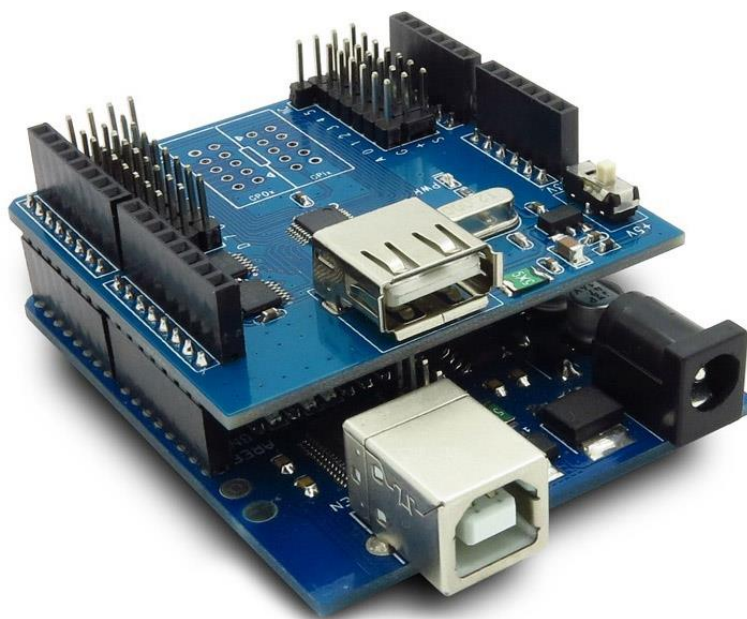
**Obrázek 3.1.2 – 3:** Arduino MICRO (vlevo) a Genuino MICRO (vpravo)

### 3.1.3 Přídavné moduly pro Arduino tzv. „shieldy“

Shieldy (česky štíty) jsou volitelné hardwarové rozšíření pro jednotky typu Arduino UNO. Mají přesné rozložení pinů jako samotné Arduino, s kterým jsou kompatibilní (tento údaj je udáván výrobcem shieldu) a ke spojení dochází jednoduchým přitlačením k sobě, takto na sebe lze navrstvit i více modulů naráz. Volné piny, které nejsou využity pro komunikaci mezi shieldem a Arduinem, bývají přemostěny a jsou dále k dispozici na jednotce Shieldu. Dodatečných modulů na trhu existuje celá řada. K dispozici jsou jak originální (OEM) moduly nabízené přímo výrobcem, tak i velké množství modulů s různými funkcemi od neoriginálních výrobců především z Číny.

Mezi základní nabízené OEM moduly patří například Arduino GSM Shield umožňující připojení k internetu, zahajování a přijímání hovorů i odesílání a přijímání SMS. Dále Arduino WiFi Shield, který nabízí možnost plnohodnotného WiFi modulu, či Arduino Ethernet Shield pro připojení k internetu pomocí ethernetového kabelu.

Zajímavé shieldy je možné sehnat i od neoriginálních výrobců. Mezi nejvíce prodávané patří například modul s relé pro spínání vyšších proudů, než jaké jsou povoleny pro samotné Arduino. Velice oblíbený je i dvouřádkový LCD displej se sadou tlačítek nebo jen základní shield umožňující připojení přes klasické USB.



**Obrázek 3.1.3 – 1:** Příklad spojení Arduina UNO a USB Shieldu



### 3.1.4 Programovací jazyk Arduina

Jak již bylo v kapitole 3.1.1 o historii Arduina zmíněno, programovací jazyk Arduina vychází z „programovacího jazyku“ Wiring. Ve většině dostupných informačních zdrojích o Arduinu, kolujících po internetu, je možné se dočíst, že právě Wiring byl samotným základem pro programovací jazyk Arduina. Málokdo už ale ví, že ve skutečnosti nejde o programovací jazyk, nýbrž o knihovnu C++. Uživatelé tak programují v C++ a ani o tom nevědí. Protože ale v případě Arduina jde o procesory, které oproti klasickým PC mají velice limitovanou paměť a je pravděpodobné, že uživatelé do ní budou chtít implementovat i komplikované programy, bylo potřeba najít cestu, jak stále opakující se části C++ kódu obejít. Zatímco C++ pracuje se samotnými registry a nezáleží na tom, jaký procesor zvolíme, v případě Arduina je zdrojový kód jazyka napsán na konkrétní typy procesoru. Pomocí zmiňované knihovny C++ lze v případě programování Arduina docílit i 80 % redukci výsledného kódu programu, oproti případu, kdy by vše bylo napsáno v čistém C++.

### 3.1.5 Prostředí pro programování Arduina

Před samotným stažením programu ze stránek výrobce je nejprve nutno zvolit pro jaký typ OS program požadujeme. V případě Windows je pak k dispozici i přenosná varianta bez nutnosti instalace fungující například i z USB flash paměti. Samotný software má přibližně 800 MB, takže se bez problému vejde na většinu paměťových médií. Samotný program je velice intuitivní a ihned po spuštění se zobrazí okno, ve kterém je možno ihned začít pracovat na kódu.

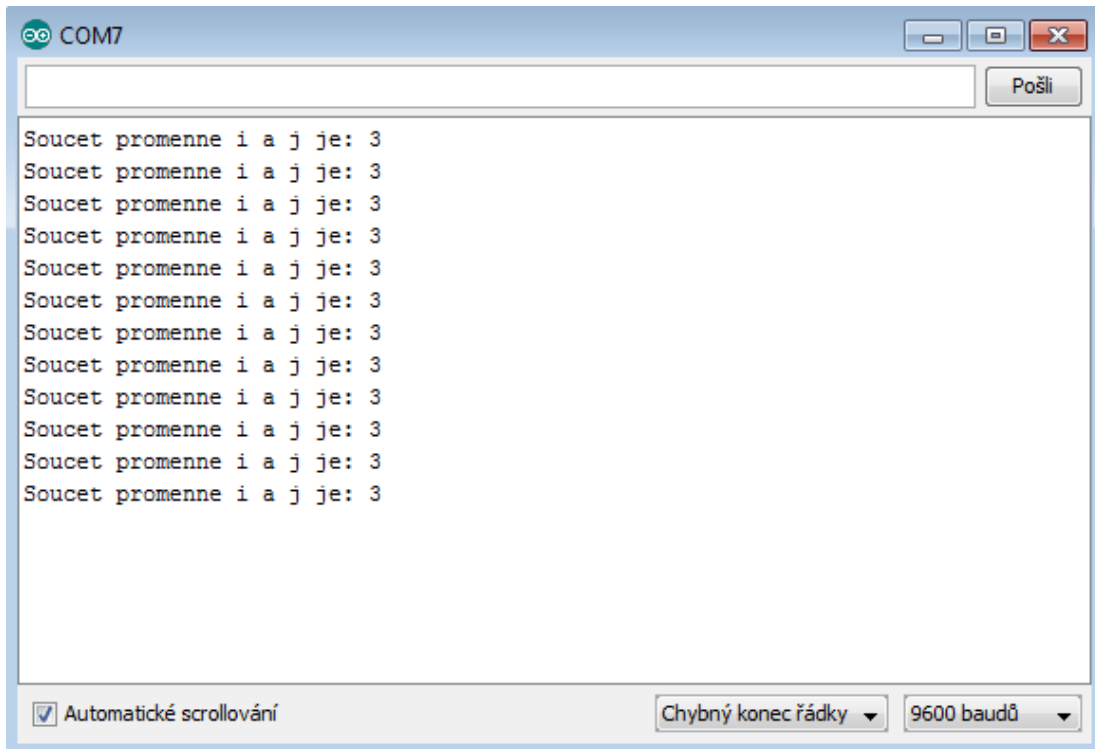
Před samotným začátkem, avšak již po připojení Arduina k PC, je nutné přes záložku „Nástroje -> Port -> COM (x)“ zvolit na jakém PC USB portu je Arduino připojeno a vybrat použitý typ vývojové desky „Nástroje -> Vývojová deska“. Před začátkem editace kódu je vhodné rozpracovaný projekt nejprve uložit. Pro drtivou většinu prováděných úkonů si uživatel vystačí s rychle přístupnými funkcemi nacházejících se ihned pod hlavní navigační lištou. Popsáno zleva:

- Ověřit (zkontroluje kód a ověří přítomnost chyb ve struktuře programu, v případě zjištění chyby vypíše nápovědu v dolní části obrazovky s odkazem na řádek a pozici, bez 100 % korektnosti není možné kód do Arduina nahrát)

- Nahrát (přenesení zdrojového kódu do jednotky Arduino)
- Nový (start nového projektu)
- Otevřít (otevření uloženého projektu)
- Uložit (uložení aktuálního projektu)
- Sériový monitor (v případě aktivování sériového monitoru uvnitř kódu programu, dojde po nahrání programu do Arduino k výpisu hodnot na obrazovku sériového monitoru při každém proběhnutém cyklu)



**Obrázek 3.1.5 – 1:** Programovací prostředí a příklad krátkého vzorového kódu



**Obrázek 3.1.5 – 2:** Ukázka výpisu na obrazovku při spuštění vzorového programu

### 3.1.6 Popis použitých funkcí

Na samém začátku je potřeba nejprve zmínit strukturu programu, která je rozdělena do třech, respektive do dvou základních částí, kde `setup()` je počáteční nastavení programu a `loop()` vykonávání cyklu programu. Před část `setup()` je možné začlenit i základní prvky kódu jako je deklarování globálních proměnných, či začlenění přídatných knihoven majících za úkol umožnit komunikaci s přídatnými zařízeními jako je např. LCD displej. Pro definování začátku a konce funkčních bloků a příkazů jsou využívány složené závorky `{ }`, jedná se párový element.

```
void setup()
{
    příkazy;
}

void loop()
{
    příkazy;
}
```

### setup()

Název funkce `setup()` vystihuje přesně její význam. Je vyvolána pouze jednou na začátku programu a využívá se například k nastavení vstupů a výstupů na jednotlivých pinech nebo aktivování sériové komunikace. Je potřeba zdůraznit, že i v případě, kdy by její obsah je prázdný, funkce musí být obsažena v kódu.

```
void setup()
{
    pinMode(digital_1, INPUT); // nastaví 'digital_1' jako vstup
}
```

### loop()

Po vykonání funkce `setup()` program přechází na funkci `loop()`, která je vykonávána cyklicky s nekonečným počtem opakování. Umožňuje tak změny v programu, odezvu na vstupní podněty a řízení desky Arduino.

```
void loop()
{
    digitalWrite(digital_1, HIGH); // aktivuje 'digital_1'
    delay(500); // vyčká 500 ms
    digitalWrite(digital_1, LOW); // deaktivuje 'digital_1'
    delay(500); // vyčká 500 ms
}
```

### : středník

Použití středníku je nutné na konci každého příkazu, má význam jako „oddělovač“ jednotlivých kroků programu.

### // řádkové a /\*...\*/ blokové komentáře

Pro přehlednost kódu je velice vhodné psát strukturovaně, odsazovat jednotlivé vnoření funkce a popisovat je stručnými výstižnými komentáři. Zejména při rozsáhlejších programu je snadné se ve změti proměnných a funkcí ztratit, nemluvě o tom, když se snaží v kódu zorientovat někdo jiný, než autor. Samotná lomítka se píší na konci řádku a vše následující za nimi je vyčleněno z programu. Lomítka s hvězdičkami pak slouží pro rozsáhlejší komentáře na více řádku, kdy je vyčleněno vše,

co se nachází uvnitř. V programovacím prostředí Arduina takto označený text zároveň pro lepší odlišitelnost změni automaticky barvu na šedou.

```
// příklad řádkového komentáře

/* z programu je vyčleněno vše, co se
nachází mezi lomítky s hvězdičkou */
```

### Proměnné

Proměnné jsou využívány pro pojmenování a ukládání číselných hodnot pro pozdější použití v programu. Jak jejich název napovídá, proměnné jsou čísla, která mohou být v průběhu cyklu pravidelně měněna. Existují čtyři základní typy byte, int, long a float. Jejich rozdíl je především v jejich rozsahu. Volit bychom je měli dle potřeb a uvážení, neboť proměnné s větším rozsahem vyžadují také více paměti, která v případě Arduina bývá dost limitována.

```
int vstupniPromenna = 10;          /* deklaruje proměnnou vstupniPro-
                                     menna a přiřadí jí hodnotu 10 */
vstupniPromenna = digitalRead(1)   /* přiřadí proměnné vstupniPromenna
                                     hodnotu z digitálního pinu 1 */
```

### Proměnná typu „byte“

Hodnoty jsou ukládány v 8-bitovém formátu jako celá čísla bez desetinných míst s číselným rozsahem 0 až 255.

### Proměnná typu „int“

Hodnoty jsou ukládány v 16-bitovém formátu jako celá čísla bez desetinných míst s číselným rozsahem -32 768 až 32 767.

Proměnná typu „long“

Hodnoty jsou ukládány v 32-bitovém formátu jako celá čísla bez desetinných míst s číselným rozsahem -2 147 483 648 až 2 147 483 647. Zvláštní vlastností všech celočíselných datových typů je jejich „nekonečnost“, kdy například v případě přičtení hodnoty 1 k číslu 32 767 je výsledkem -32 768.

Proměnná typu „float“

Datový typ float nalezne využití všude, kde není dostatečný rozsah předešle zmíněných datových typů nebo v případě potřeby desetinného místa. Pracuje s rozsahem -3.4028235E+38 až 3.4028235E+38. Hodnoty jsou ukládány v 32-bitovém formátu.

Konstanty

Programovací jazyk Arduina má předdefinováno několik základních konstant pro usnadnění orientace v programu.

Konstanty typu „true/false“

Konstanta FALSE je definována jako 0 (nula), zatímco TRUE jako 1, ale platí i pro jakoukoli jinou hodnotu, vyjma nuly.

Konstanty typu „high/low“

HIGH a LOW jsou velice často používané konstanty pro definování úrovní na digitálních pinech při čtení a zápisu. HIGH je možné chápat jako nastavení logické 1, aktivování, nastavení napětí na 5 V, zatímco LOW má logickou úroveň 0, deaktivuje a nastavuje napětí na 0 V.

Konstanty typu „input/output“

Definuje chování digitálních pinů, zda se jedná o vstup - INPUT nebo výstup - OUTPUT.

Aritmetické operace

Funkce pro sčítání, odčítání, dělení a násobení fungují klasickým použitím základních znamének + - \* /.

### Složené přiřazení

Pro usnadnění výpočtu, zejména pro podmínky ve smyčkách se využívají kombinace aritmetické operace a přiřazení hodnoty proměnné.

```
x ++           // totožné jako x = x + 1, nebo navýšení x o +1
x --           // totožné jako x = x - 1, nebo snížení x o -1
x += y         // totožné jako x = x + y, nebo navýšení x o +y
x -= y         // totožné jako x = x - y, nebo snížení x o -y
x *= y         // totožné jako x = x * y, nebo x vynásobené y
x /= y         // totožné jako x = x / y, nebo x vydělené y
```

### Porovnávací operátory

Logické operátory slouží k porovnávání dvě hodnoty (výrazy) a vrací výsledek TRUE nebo FALSE. Existují tři používané logické operátory AND (a), OR (nebo) a NOT (není), nejčastěji využívané v podmínce if.

#### Logické AND

```
if (x > 0 && x < 5)    /* vyhodnotí jako TRUE pouze
                        pokud obě podmínky jsou splněné */
```

#### Logické OR

```
if (x > 0 || y > 0)    /* vyhodnotí jako TRUE pokud
                        alespoň jedna z podmínek je splněná */
```

#### Logické NOT

```
if (!x > 0)            /* vyhodnotí jako TRUE pokud
                        x není větší než nula */
```

### Podmínka „if“ a „if... else“

Při každém cyklu kontroluje, zda je dosaženo zadané podmínky, pokud ano, spustí soubor příkazů obsažených ve složených závorkách spadajících pod podmínku IF, pokud ne, program příkazy ve složené závorce přeskočí. Podmínku IF je možno rozšířit i o ELSE, je pak do dalších složených závorek možné zapsat příkazy, které mají být vykonány v případě nesplnění podmínky.

```

if (x == 3)
{
    provedPrikazA;           // pokud x je rovno 3, provede příkaz A
}
else
{
    provedPrikazB;           // pokud x není rovno 3, provede příkaz B
}

```

Pozn: Je důležité pamatovat na rozdíl mezi dvojitým rovnítkem == (funguje jako porovnávací operátor dvou hodnot) a = (slouží k přiřazování hodnot).

### Cyklus „while“

V předkladu je možno chápat jako „opakuj, dokud platí“. Příkaz ve složených závorkách je opakován do té doby, dokud podmínka obsažená v hlavičce cyklu je vyhodnocena jako TRUE. K vyhodnocení podmínky dochází vždy na začátku cyklu.

```

while (x > 50)
{
    provedPrikazA;           // opakuje příkaz A dokud x je větší než 50
}

```

### Cyklus „do... while“

Je obdobou přechozího cyklu while, jen s tím rozdílem, že podmínka je vyhodnocována až na konci cyklu. Z toho důvodu i při nesplnění podmínky cyklus proběhne alespoň jednou.

```

do
{
    provedPrikazA;           // provede příkaz A
}
while (x > 50);             // pokud je x větší než 50, spustí cyklus znovu

```



Funkce „max(x, y)“ a „min (x, y)“

Vyhodnotí, které ze dvou čísel jakéhokoli datového typu je dle zadané funkce menší, či větší.

```
hodnotaA = min(2, 10); // po vyhodnocení je hodnotaA rovna 2
```

Funkce „map“

Přepočítá vstupní rozsah na požadované rozmezí. Například vstupní hodnotu z potenciometru v rozsahu 0-1023 na 0-100 % nebo 0-5 V.

```
hodnotaB = abs(map(hodnotaA, 0, 1023, 0, 100));
           /* Přemapuje vstupní rozsah proměnné hodnotaA
            * z 0-1023 na 0-100 v absolutní hodnotě */
```

pinMode(pin, mód)

Digitální piny Arduina jsou defaultně nastaveny jako vstupní, nicméně je vhodné každý (i vstupní) pin deklarovat v úvodní části programu void setup(). Do hodnoty „pin“ se zapisuje číslo pinu, k nalezení je na popisu desky, případně na stránkách výrobce. Mód určuje, zda se jedná o vstup - INPUT nebo o výstup - OUTPUT.

```
pinMode(5, INPUT); // nastaví digitální pin 5 jako vstup
```

digitalRead(pin)

Přečte hodnotu ze zadaného digitálního pinu a vrátí výsledek ve formě HIGH nebo LOW. Pin může být zadán buďto jako proměnná nebo číslo.

```
A = digitalRead(5); // přiřadí proměnné A hodnotu z digit. pinu 5
```

digitalWrite(pin, hodnota)

Nastaví zadaný pin na hodnotu HIGH (5 V) nebo LOW (0 V).

```
digitalWrite(5, LOW); // nastaví digitální výstup 5 jako LOW
```

Pozn.: Pozor na využití 5 V u digitálních výstupů a jejich zatížení, u většiny jednotek bývá bezpečné zatížení udávané výrobcem 20 mA, což stačí například pro rozsvícení LED, ale je nutno do serie s LED umístit odpor.

### analogRead(pin)

Přečte hodnotu ze zadaného analogového pinu v 10-ti bitovém rozlišení. Vráť hodnotu jako integer (celočíslo) v rozsahu 0-1023.

```
A = analogRead(5);           // přiřadí proměnné A hodnotu z analog. pinu 5
```

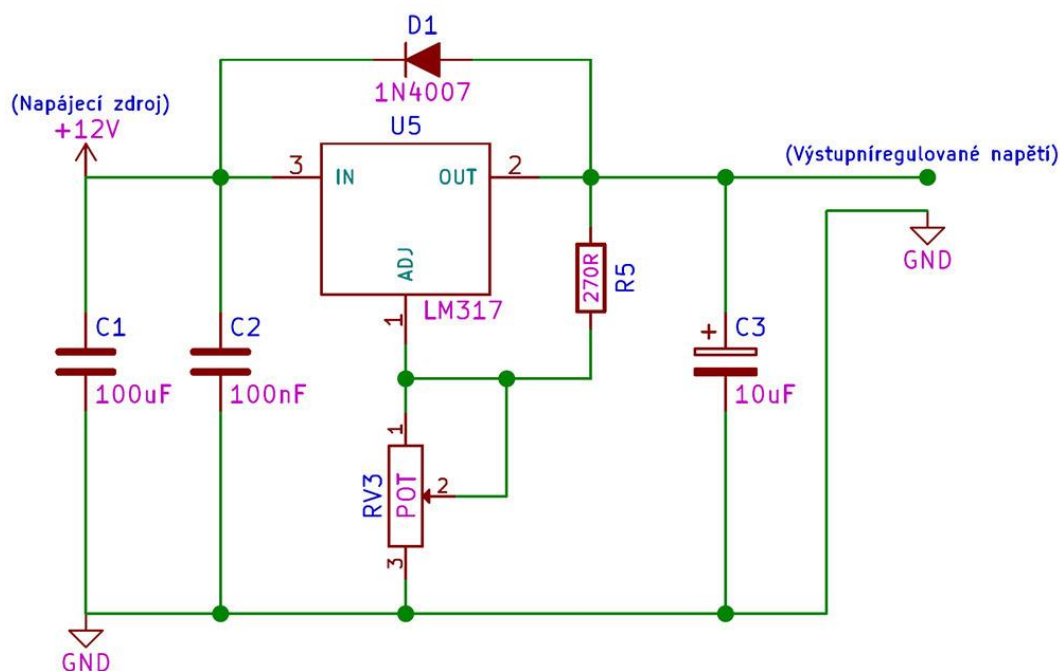
### Prodleva

Častým požadavkem programů bývá přidání prodlevy, například pro blikající LED. Hodnota se vkládá v milisekundách, tedy 1000 jednotek odpovídá 1 sekundě.

```
delay(1000);                  // vyčká jednu sekundu
```

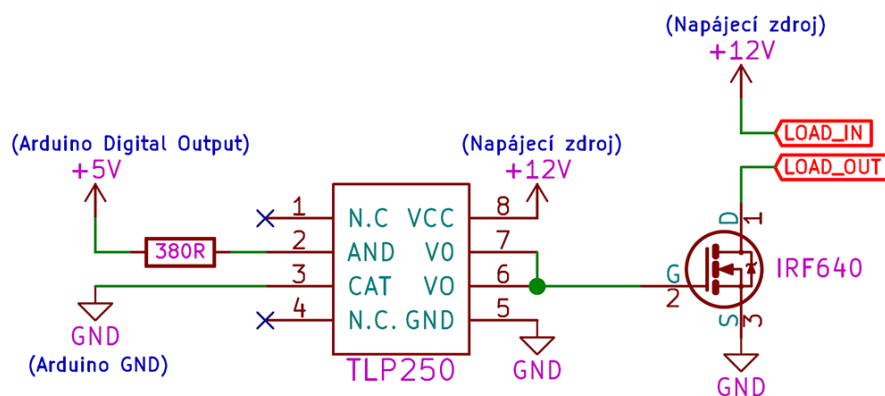
## 3.2 Obvod pro spínání 12 VDC a regulátor napětí

Záměrem práce bylo vytvořit kompaktní systém, který bude ke svému provozu potřebovat pouze jeden zdroj napájení. Jelikož napájení aktivních prvků jako jsou ventily a kompresor vyžadují napětí 12 V, které je pro napájení Arduina limitní a v případě výkyvů by tak mohlo dojít k poškození mikroprocesoru. Rozhodl jsem se sestavit jednoduchý regulovatelný zdroj nastavitelný pomocí potenciometru – pro naše potřeby nastaveno na optimálních 9 V. Jako stabilizátor napětí byl použit integrovaný obvod LM 317 s výstupním napětím 1,25 až 37 V (v závislosti na vstupním napětí), součástí jsou dále filtrační kondenzátory na vstupu a výstupu a nastavitelný odporový dělič.



**Obrázek 3.2 - 1:** Schéma regulovatelného zdroje

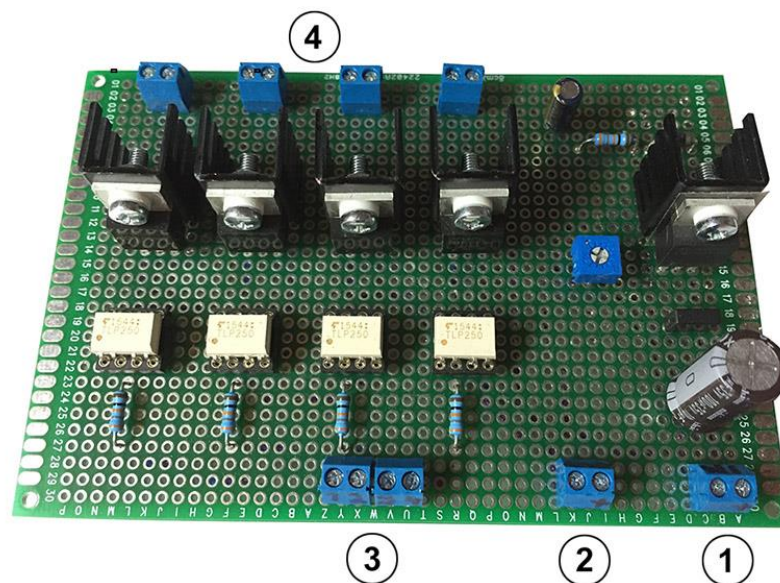
Z důvodu omezení zatížení digitálních výstupů Arduino bylo nutné sestavit dodatečné obvody pro spínání napětí na kompresoru a tlakových ventilech. Ty jsou sestaveny z optočlenů, které se skládají z LED diody a fotocitlivé polovodičové součástky. Jejich přední výhodou je možnost oddělení vstupního a výstupního obvodu a nízká náročnost na proudový odběr při spínání (LED). Pro naše potřeby se jako ideální volbou jevil optočlen TLP 250 s proudovým odběrem 11 mA. Jelikož i TLP 250 jsou limitovány maximálním zatížením 1,5 A jako koncový prvek byl použit tranzistor IRF 640 opatřený chladičem. Jako LOAD IN a LOAD OUT je označeno místo pro připojení spínaného zařízení. Kompletní schéma elektrického obvodu je z důvodu velkých rozměrů, umístěno na konci diplomové práce, v příloze 1.



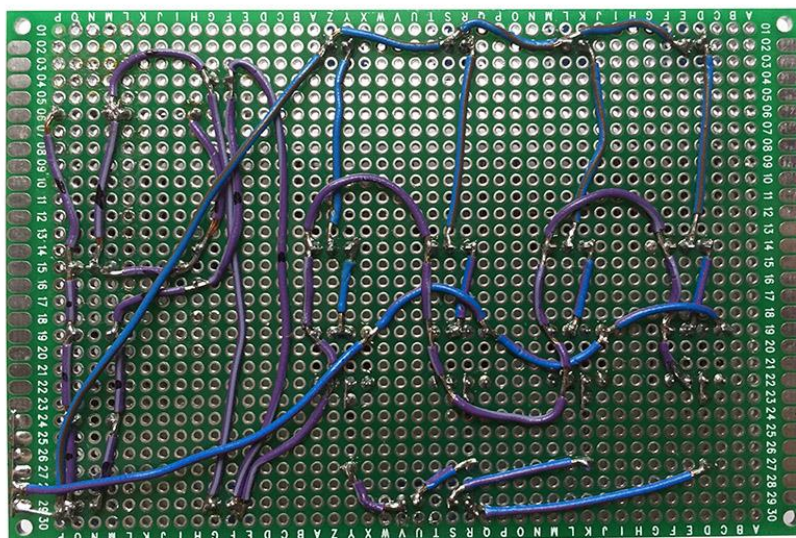
**Obrázek 3.2 - 2:** Schéma obvodu pro spínání napětí kompresoru a ventilů

Pro realizace obvodu byla využita oboustranně pájitelná prototypová deska o rozměrech 8 x 12 cm s pocínovanými otvory v počtu 30 x 42, samotné propojení cest bylo provedeno pomocí drátků. Z důvodu snadné manipulace a rozpojitelnosti systému jsem volil pro vstupy i výstupy terminály s aretací pomocí šroubu.

- 1) Vstupní napětí 12 V
- 2) Výstupní stabilizované napětí 9 V
- 3) Vstupy (1, 2, 3, 4) spínané signálem s Arduina 5 V
- 4) Spínané výstupy (1, 2, 3, 4) 12 V



**Obrázek 3.2 – 3:** Sestavený obvod – přední strana

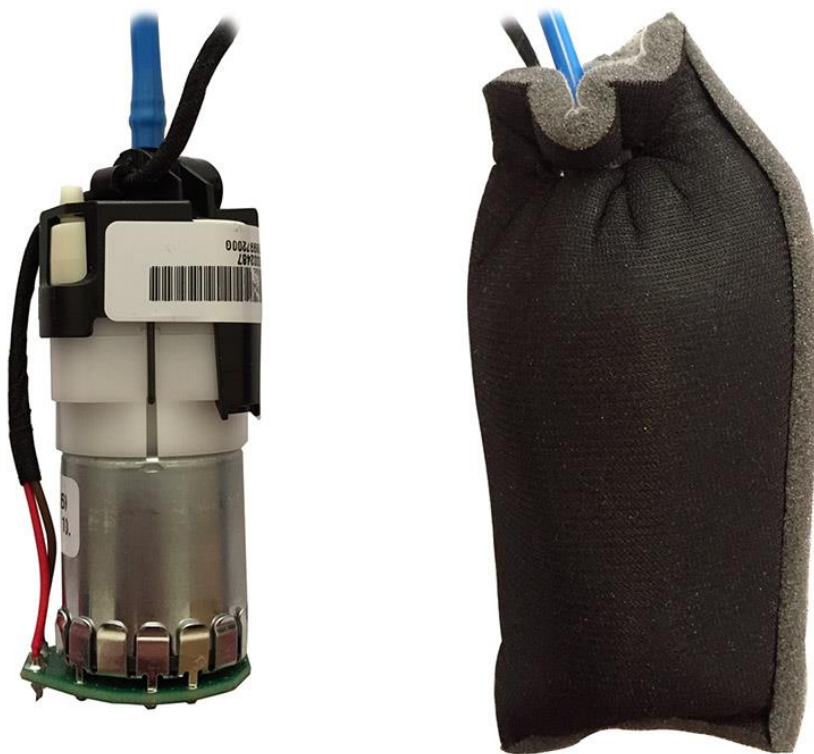


**Obrázek 3.2 – 4:** Sestavený obvod – zadní strana

### 3.3 Membránový kompresor

Volba generátoru vzduchu byla jednou za základních otázek při sestavování celého systému. Problémem byly zejména protichůdné požadavky na kompaktnost systému a zároveň nízký hluk, který by mohl ovlivňovat spánek spící osoby. Proto bylo původně zamýšleno (i na úkor kompaktnosti), alespoň pro náš prototypový model, použití kompresoru se vzdušníkem, tedy zásobníkem tlaku, který by bez obtíží a v tichosti dokázal vyrovnat případné výkyvy bez vyrušení i během noci a k doplnění vzdušníku by docházelo v denních hodinách.

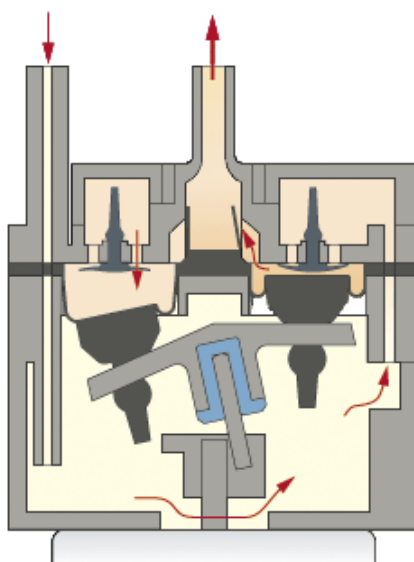
Avšak od mého zaměstnavatele, výrobce sedadel do automobilů, se naskytla příležitost využití membránových kompresorů používaných k nafukování vaků bočního vedení sedáků s napájením na 12 V. Tyto kompresory jsou velice kompaktní, přibližně 8 x 4 cm a s ohledem na své rozměry vynikají výbornými vlastnostmi v maximálním generovaném tlaku, životností dosahující až 3 000 hodin a nízkém hluku (přibližně 40 dB), který v případě zmiňovaných kompresorů je ještě snížen díky protihlukovému látkovému pouzdra se silnou laminací. Membránový kompresor nemá ani žádné omezení na polohu, ve které bude uložen.



**Obrázek 3.3 – 1:** Membránový kompresor, vpravo včetně protihlukového pouzdra



Princip membránového kompresoru je založen na přeměně otáčivého pohybu vykonávaného motorem na pohyb přímočarý vratný vykonávaný pryžovými písty. Při pohybu pístu směrem dolů dochází podtlakem k otevření silikonové klapky umožňující nasávání vzduchu. Po dosažení dolní pozice a následném stlačování dojde k uzavření klapky a vzduch je vytlačován hlavním vývodem ven.



**Obrázek 3.3 – 2:** Princip membránového kompresoru, šipkami je označen tok [15]

### 3.4 Senzor tlaku

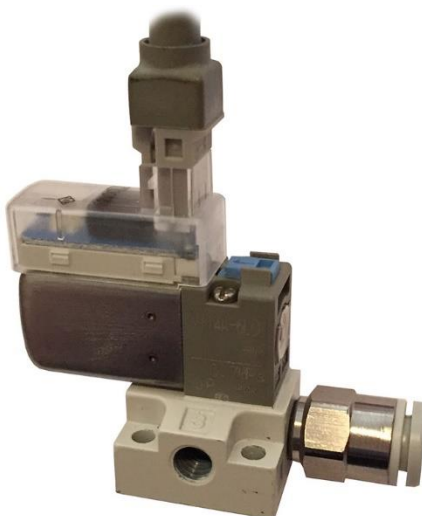
Aby bylo možné řídit hodnotu tlaku v systému, bylo nutné začlenit prvek k jeho měření v reálném čase. Na volbu sensoru nebyly kladeny žádné vysoké nároky. Postačující byl jakýkoli typ, který dovede měřit v rozsahu alespoň do 50 kPa a umožňuje napájení v rozmezí 5 – 12 V. Využit tedy byl z katedry dostupný tlakový sensor DRM0D-I2C-R4B od firmy B+B Thermo-Technik, měřící v rozsahu 0 – 160 kPa. Doporučené operační napětí je 6 – 15 V, které je v případě analogového výstupu usměrněno na 5 V. K dispozici je také digitální přenos pomocí I<sup>2</sup>C rozhraní, které však pro naše účely nebylo nezbytné. Změřený tlak je přímo úměrný hodnotě na analogovém výstupu, 0 – 5 V odpovídá tlaku 0 – 160 kPa.



**Obrázek 3.4 – 1:** Tlakový sensor DRMOD-I2C-R4B

### 3.5 Solenoidový ventil

Obdobně, jako v případě tlakového sensoru, nároky na solenoidové ventily měli dvě základní kritéria. Napájení pomocí 12 V a schopnost uzavřít tlak o velikosti alespoň 50 kPa. Taktéž v tomto případě byly použity z katedry dostupné solenoidové ventily s označením V114A-6LOU od firmy SMC, které jsou zároveň vybaveny rychlospojkami pro snadné připojení hadiček o vnějším průměru 6mm. Vyžadují napájení 12 VDC a umožňují uzavírat tlak až do 0,7 MPa. Princip solenoidového ventilu spočívá v jádru z feromagnetického materiálu uvnitř cívky, které je v klidovém stavu tlačeno pružinou a uzavírá tak cestu pro průchod kapaliny nebo vzduchu z jednoho vstupu do druhého. Po přivedení napětí, které prochází cívkou uvnitř ventilu, vznikne elektromagnetické pole, jehož působením na feromagnetické jádro dojde k překonání protisíly pružiny a uvolnění cesty pro průchod vzduchu, či kapaliny.



**Obrázek 3.5 – 1:** Elektromagnetický ventil V114A-6LOU

### 3.6 Ostatní komponenty systému

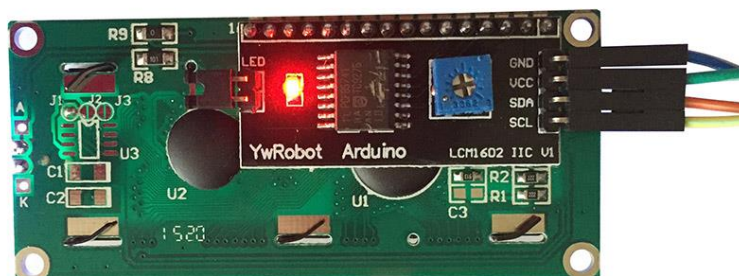
#### LCD displej

Ač se nejednalo o nezbytně nutný prvek pro funkčnost systému, tak v dnešní době, kdy je trh přesycen a konkurence značná, je z hlediska marketingu přinejmenším vhodné, soustředit se i na detaily, které koncového zákazníka dovedou přesvědčit právě pro náš výrobek. Použit tedy byl dvouřádkový LCD displej s atraktivním modrým podsvícením s možností zápisu 16 znaků na každý řádek. Dle pozice potenciometru bude na řádcích zobrazováno aktuální nastavení tuhosti pro konkrétní zónu matrace.

Ke komunikaci mezi Arduinem a LCD displejem se využívají podpůrné knihovny dle čipové sady LCD panelu, k dispozici jsou například na oficiálních stránkách Arduina. Přímé propojení displeje s jednotkou Arduina je poměrně náročné na počet potřebných digitálních pinů a tak bylo využito jednoduché připojení přes sériovou sběrnici I<sup>2</sup>C pomocí převodníku implementovaného na LCD panelu. Náročnost na digitální piny se tak snížila na pouhé dva, konkrétně pin 2 (SDA – sériová datová linka) a pin 3 (SCL – sériová linka hodinového signálu).



**Obrázek 3.6 – 1:** LCD displej – přední strana

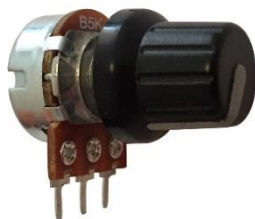


**Obrázek 3.6 – 2:** LCD displej (včetně I<sup>2</sup>C převodníku - černý) – zadní strana



### Potenciometry

Nabízely se dvě jednoduché varianty způsobu regulace uživatelem. První by zahrnovala použití tlačítek + a – pro navyšování a snižování požadované hodnoty, druhá použití potenciometru s hmatníkem. Vydal jsem se raději cestou intuitivnějšího ovládání, které bez většího přemýšlení lze používat i poslepu. Rozsah potenciometru pak odpovídá minimálnímu a maximálnímu možnému nastavení tuhosti. Pro náš účel byl zvolen 5k lineární potenciometr.



**Obrázek 3.6 – 3:** Lineární potenciometr s hmatníkem

### Napájecí zdroj

Volba zdroje vychází především z již zmíněných prvků, jako je kompresor a ventily, které vyžadují stejnosměrné napětí 12 V. Kombinované zatížení všech prvků systému nepřesáhne 3 A, tudíž by byla postačující například nabíječka od notebooku. Avšak v našem případě jsem měl k dispozici počítačový ATX zdroj, který byl jednoduchou úpravou přeměněn na 12 V zdroj. Úprava spočívá v propojení (např. odporem) zeleného a černého (zem) pinu na 24 pinovém konektoru, po kterém dojde ke spuštění zdroje. Poté už je možné využít 4 pinové napájecí kabely jako zdroje napětí, žlutý drát 12 V, červený drát 5 V, černý drát zem.



**Obrázek 3.6 – 4:** Počítačový ATX zdroj jako zdroj stejnosměrného napětí

## 4 METODA MĚŘENÍ TUHOSTI

Před začátkem návrhů způsobu regulace tuhosti pěny, bylo nejprve nutné zvolit a realizovat metodu pro ověřování dosažených výsledků, neboť posuzování dle subjektivních dojmů je v případě tuhosti pěny sice potřebné, ale spoléhat se pouze na něj může být zavádějící. Firmy, zabývající se vývojem a výrobou molitanových výrobků jako jsou matrace, či pěny do automobilových sedadel, ve většině případů využívají experimentální metodu vtlačování tělesa, jehož tvar se může lišit v závislosti na tvaru a velikosti měřené pěny. Existují dva základní principy měření. Při prvním je těleso vtlačováno konstantní rychlostí a odečítána je hodnota zatížení v průběhu celé dráhy, druhou metodou je vtlačování tělesa lineárně zatěžující silou, odečítána je pak v tomto případě hloubka vtlačení.

### 4.1 Návrh měřicího zařízení

Zařízení uzpůsobená k měření tuhosti pěn jsou velice nákladná a vlastní je pouze specializované firmy. Proto bylo nutné vymyslet a sestavit obdobné měřící zařízení, které splní dané požadavky a nebude finančně nákladné. V tomto ohledu mi bylo umožněno využití kalibrovaného laboratorního vybavení mého zaměstnavatele zabývajícího se montáží automobilových sedadel, především digitálního siloměru opatřeného tenzometrem využívaného k měření ovládacích sil během produktových auditů a modulárního upínacího systému ALUFIX od firmy Witte.



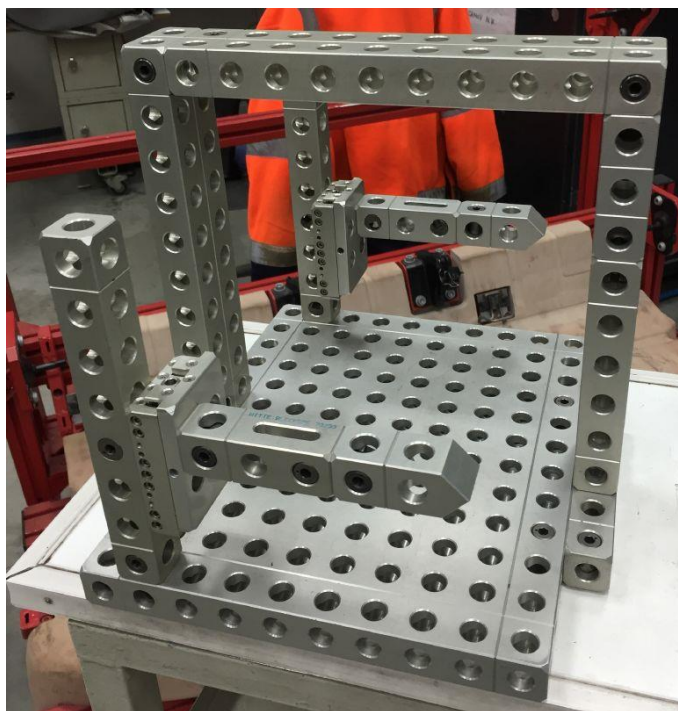
**Obrázek 4.1 – 1:** Kalibrovaný tenzometr Zemic

Tenzometr je konstrukčně dimenzovaný na maximální zatížení 50 kg. Pro naše účely předpokládáme zatížení maximálně 100 N, což odpovídá přibližně 10 kg. Není proto nutné se obávat přetížení a případného poškození tenzometru.



**Obrázek 4.1 – 2:** Kalibrovaný siloměr Mecmesin

Stavebnice ALUFIX je velice přesný (na setiny milimetru) modulární systém pro tvorbu měřicích přípravků, upínacího zařízení, či nosných konstrukcí. V nabídce existují stovky komponent pro různé druhy použití, které je možno dokupovat jednotlivě nebo v předpřipraveném setu. V našem případě bylo využito především podélných nosníků a základové desky pro vytvoření pevné konstrukce měřicího zařízení.



**Obrázek 4.1 – 3:** Sestavená konstrukce měřicího zařízení

## 4.2 Realizace a ověření pohyblivé části

Pohyblivou část bylo nutné navrhnout tak, aby byl zajištěn plynulý pohyb v ose „y“, ale nebylo umožněno tenzometru rotovat kolem této osy. Vtlačované těleso vůči měřené pění tak bude mít pouze jeden stupeň volnosti. Plynulý posuvný pohyb byl zajištěn použitím závitové tyče se závitem M12 se stoupáním 1,75 mm/ot., ke které byla přivařena tyč pro snadné ovládání. Součástí jsou dále vodící tyče z hlazené kulatiny o průměru 12 mm mající za úkol zabránit rotaci kolem osy „y“. Z pásoviny pak byly vytvořeny spojovací a zbylé vodící prvky. Pro plynulejší chod byly použity i dvě axiální ložiska mající funkci podložky v místě, kde je závitová tyč upevněna k profilu z pásoviny. Většina spojů je realizována pomocí šroubů a celý systém je tak snadno demontovatelný. V případě potřeby byly použity pojistné matice.

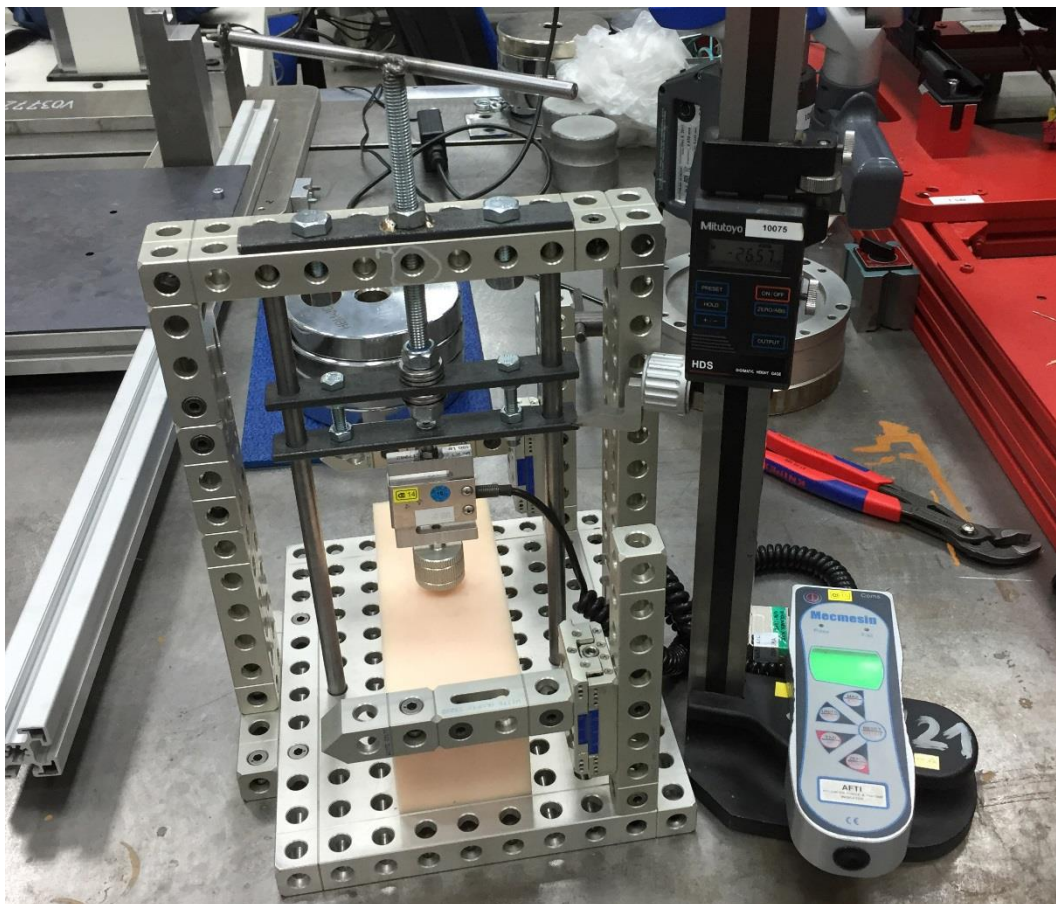
Hlazená tyč a pásovina je za přijatelnou cenu dostupná v kterémkoli obchodě s hutním materiálem. Po nařezání na požadovanou délku došlo k obroušení hran a vyvrtání potřebných děr.



**Obrázek 4.2 – 1:** Příprava prvků pro pohyblivou část



Jakmile byly veškeré prvky připraveny, mohlo dojít k sestavení kompletního měřicího zařízení a ověření základních požadovaných funkcí, především tedy plynulého posuvu v ose „y“.

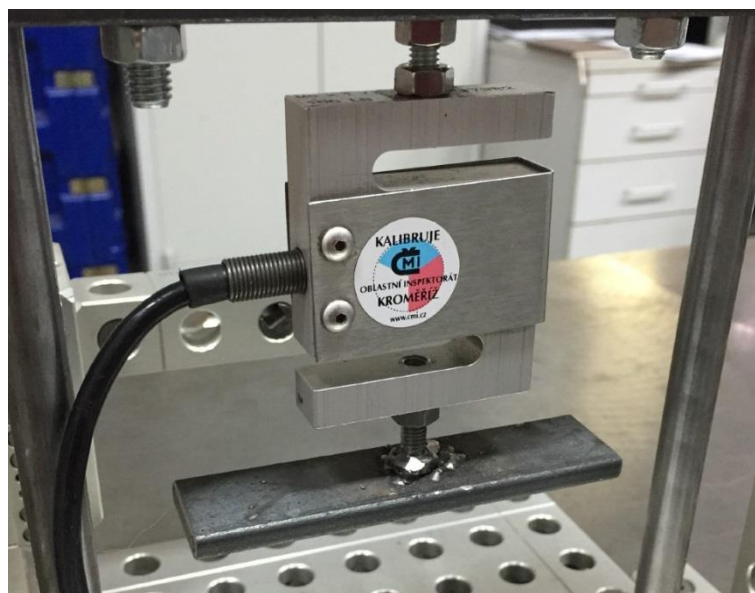


**Obrázek 4.2 – 2: Sestavené měřicí zařízení**

Po prvním testu pohyblivosti byl posuv vyhodnocen jako dobrý, avšak pro mírné zlepšení kluzných vlastností došlo dodatečně k namazání vodicích tyčí. Pro ověření posuvu 1,75 mm/ot. byl použit výškoměr, kde při následné kontrole nebyly zaznamenány žádné odchylky od očekávaných výsledků. Po mechanické stránce, tak bylo zařízení shledáno jako vyhovující.

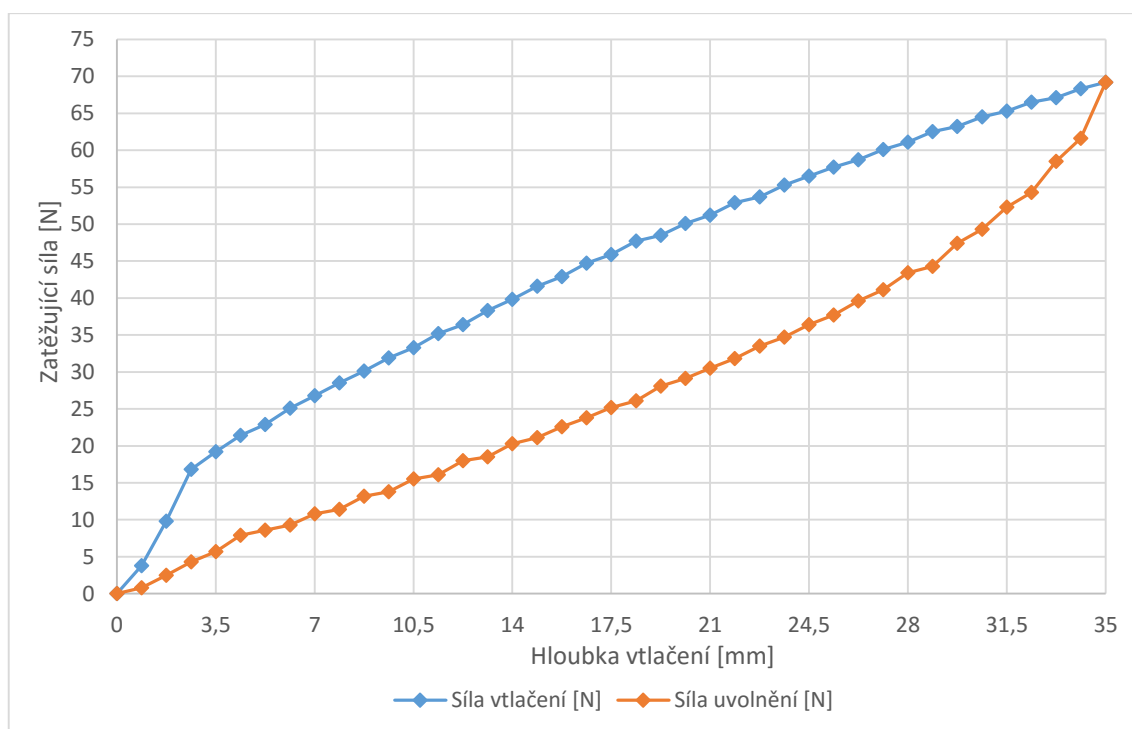
### 4.3 Ověření způsobilosti měřidla

Abychom měřené výsledné hodnoty mohly považovat za relevantní, bylo zapotřebí ověřit způsobilost měřicího zařízení, což se dá definovat jako schopnost zařízení podávat stejné výsledky i při opakovaném měření. K tomuto účelu byl použit jednolitý blok z růžové pěny bez výřezů (obr. 4.2 – 2). Očekávaným výsledkem tak byla hysterezní křivka se skokovým náběhem na začátku zatěžování a částečně na začátku uvolňování. Tento jev je způsobený nutností překonat počáteční vnitřní napětí uvnitř pěny. Jako vtlačované těleso byl použit prvek o rozměrech 100 x 25 mm mající za úkol simulovat vlastnosti chování lamely roštu.

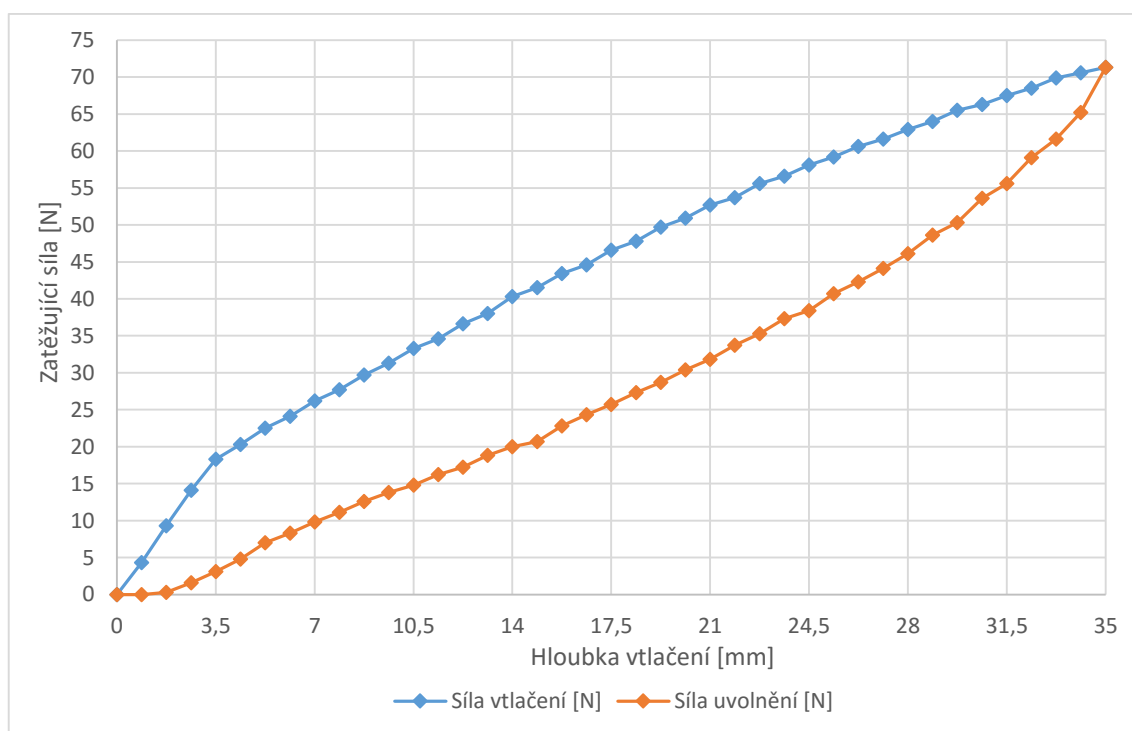


**Obrázek 4.3 – 1: Tenzometr a vtlačované těleso**

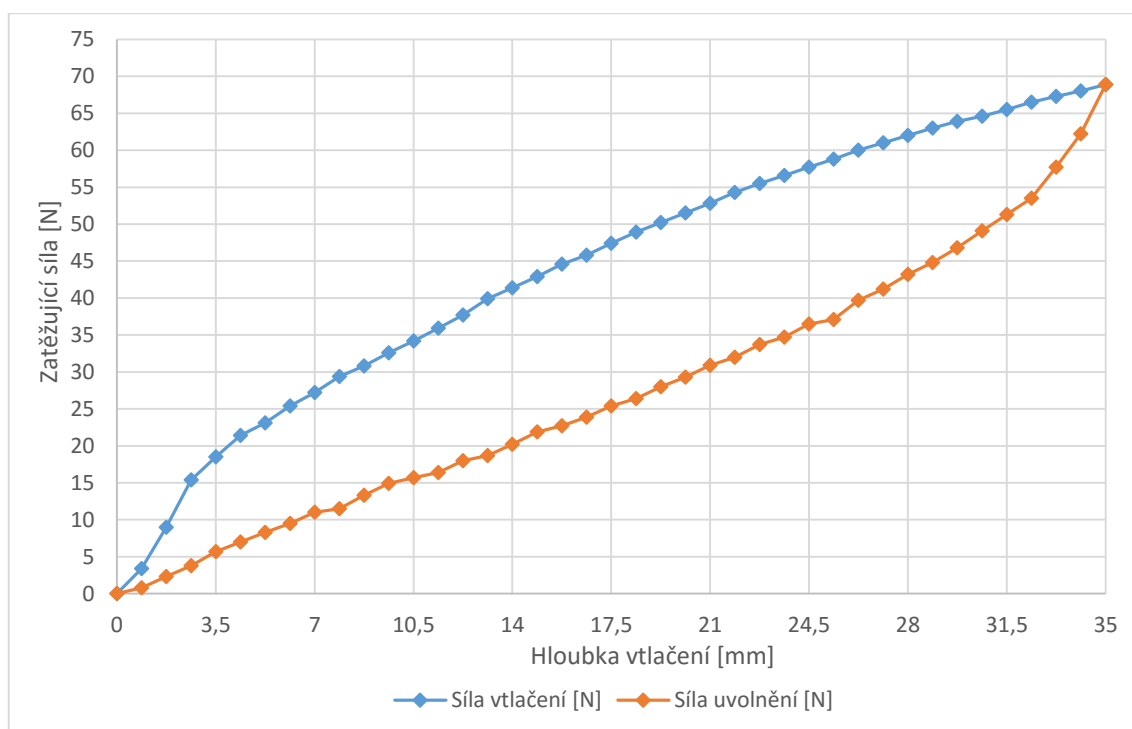
Ověření způsobilosti bylo prováděno na třech po sobě jdoucích měřeních, kdy při každém měření došlo k pootočení nebo částečnému posunutí pěny, aby se zamezilo případnému ovlivnění výsledku z důvodu stlačování během předchozího měření. Samotné měření probíhá ručním pootáčením kliky šroubovice při snaze provádět otáčení konstantní rychlostí. Během každé půlotáčky je zaznamenána hodnota z měřidla pomocí fotoaparátu, aby nebyl narušen plynulý chod. Celková vtlačovaná hloubka je 35 mm, což odpovídá 40 naměřeným hodnotám při stlačování a 40 hodnotám při uvolňování. Hodnoty jsou po skončení zaznamenány do tabulky a je vygenerován graf.



Graf 4.3 - 1: Verifikační měření č. 1



Graf 4.3 - 2: Verifikační měření č. 2



**Graf 4.3 – 2: Verifikační měření č. 3**

Dosažené výsledky naznačují, že měření proběhlo dle očekávání. Až na drobné odchylky způsobené ručním měřením a případným napozicováním pěny je možné považovat výsledky za shodné. Velice dobrého výsledku bylo dosaženo především také při maximálním odporu v koncové fázi stlačení v hloubce 35 mm, kde hodnoty dle jednotlivých měření dosahovaly 69,2 N, 71,3 N a 68,9 N. Maximální odchylka tak činila pouze 2,4 N, přibližně tedy 3 %. Z uvedených zjištění je tak možné považovat měřidlo pro naše účely za způsobilé.



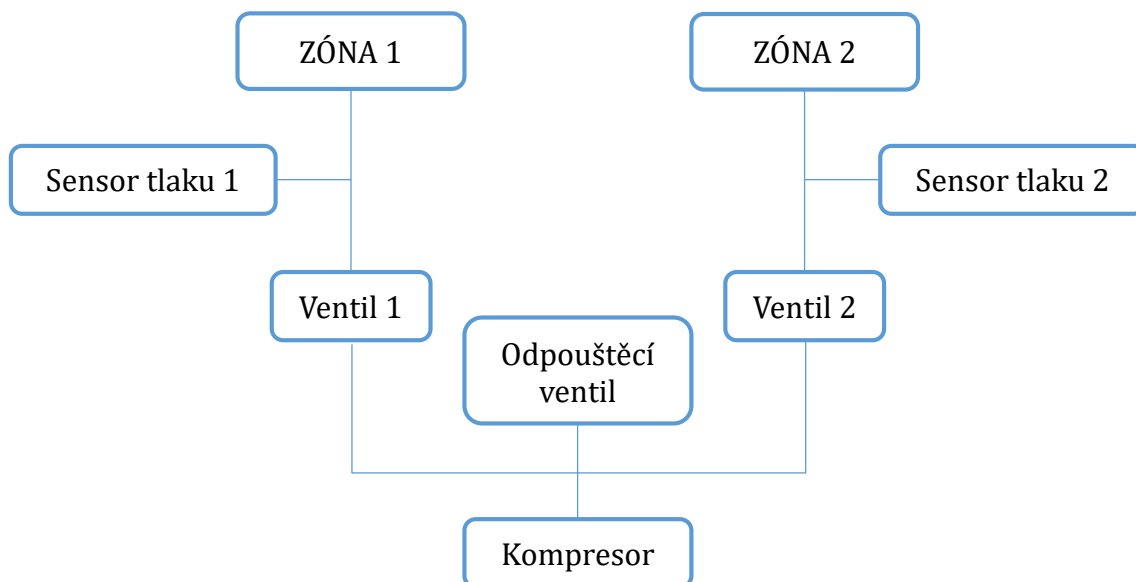
## 5 SYSTÉM REGULACE TUHOSTI

Jak již bylo zmíněno, cílem bylo navrhnout systém řízení, který bude složený jen z nezbytného počtu komponent nutného pro plnohodnotné fungování. Neboť jakékoli prodražování, které zákazníkovi nepřinese hodnotu, je z hlediska marketingu a očekávaného výnosu přinejmenším nevhodné.

### 5.1 Schéma pneumatického obvodu

Systém je tedy tvořen dvěma oddělenými částmi pro nezávislé ovládání jak ramenní, tak kyčelní oblasti. Každá z těchto částí má pro své fungování vlastní solenoidový ventil pro udržení tlaku, sensor tlaku pro informaci o aktuální hodnotě tuhosti (tlaku) a potenciometr pro plynulou regulaci v konkrétní zóně.

Sdílenými prvky pro obě zóny je řídicí jednotka Arduina - typ Micro, kde došlo k využití čtyř analogových pinů a šesti digitálních pinů, membránový kompresor dodávající dle potřeby tlak do soustavy a ventil pro odpouštění tlaku ze soustavy. Zjednodušené schéma pneumatické soustavy je znázorněno na následujícím grafu.



**Graf 5.1 - 1:** Schéma pneumatické soustavy

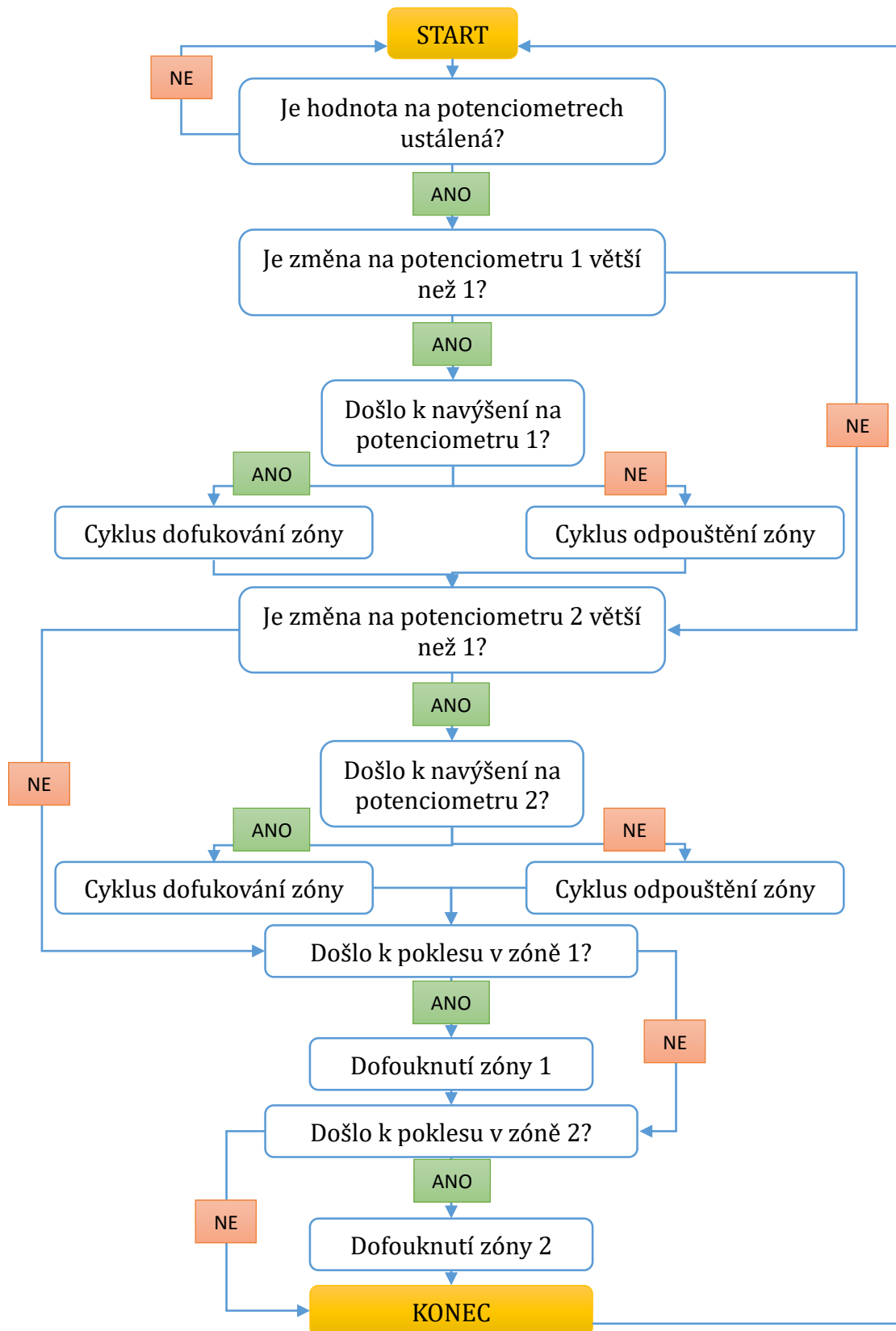


### 5.3 Řídicí algoritmus a vývojový diagram

Programování zdrojového kódu probíhalo přímo ve vývojovém prostředí Arduina, kde je možnost jak okamžité odezvy na případnou chybu, tak možnost uploadu kódu a ověření jeho funkčnosti.

Samotný kód je dělen do jednotlivých sekcí a podsekcí, u kterých nechybí krátký komentář pro snadnější orientaci, především pro ty, kteří kód nepsali. Jelikož je v kódu velké množství pasáží, kde dochází k opakování příkazů, jako jsou aktualizace proměnných v případě čtení a přepočtu hodnot z potenciometrů a tlakových snímačů nebo výpis hodnot na LCD displej, bylo využito definování vlastních vlastních void funkcí.

Pro lepší porozumění řídicího algoritmu byl vypracován zjednodušený vývojový diagram, popisující jeden pracovní cyklus funkce void loop(), která je cyklicky opakována. Opakující se funkce například pro aktualizaci hodnot na senzorech nebo potenciometru jsou z důvodu své početnosti pro přehlednost vynechány.



**Graf 5.3 - 1:** Zjednodušený vývojový diagram znázorňující pracovní cyklus

## 5.4 Popis zdrojového kódu

Zdrojový kód je rozčleněn do celkem tří základních skupin – definování proměnných, jednorázový cyklus při startu programu a pracovní cyklus. V těchto skupinách se postupně nachází čtrnáct částí, případně jejich pod-částí, kdy každá z nich je podrobně vysvětlena. Aktivní část kódu, případně s komentářem, je umístěna do ohraničeného bloku.

*Komentář // 0 //:*

*Knihovny obsahují rozšiřující řadu příkazů, pomocí kterých má Arduino umožněno komunikovat s LCD panelem. Musejí být umístěny ve složce programu Arduina pod složkou libraries a příslušnou pod-složkou, jinak by nedošlo ke správnému načtení.*

*// 0 // Přiřazení knihoven pro ovládání LCD panelu*

```
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
```

*Komentář // 1 //:*

*Dochází k deklarování, na jaké analogové piny jsou potenciometry připojeny a deklarování případných dalších proměnných potřebných do následujících kroků. Pokud jim bude hodnota přiřazena až později, není třeba ji zatím zadávat. Význam jednotlivých proměnných je popsán za lomítky.*

*// 1.1 // Potenciometr 1 a deklarace proměnných*

```
int potentiometer_1 = A0; // Potenciometr na pinu Analog 0
int potRaw_1; // Hodnota na potenciometru (0-1023)
int potPercent_1; // Hodnota na potenciometru převedená do stupnice (0-100)
int lastPotPercent_1=0; // Proměnná pro záznam hodnoty potenciometru z předchozího kroku
```

```
// 1.2 // Potenciometr 2 a deklarace proměnných
```

```
int potentiometer_2 = A2; // Potenciometr na pinu Analog 2
int potRaw_2; // Hodnota na potencimetru (0-1023)
int potPercent_2; // Proměnná pro hodnotu z potenciometru převedenou do
procent
int lastPotPercent_2=0; // Proměnná pro záznam hodnoty potenciometru z
předchozího kroku
```

*Komentář // 2 //:*

*Dochází k deklarování, na jaké analogové piny jsou senzory tlaku připojeny a deklarování případných dalších proměnných potřebných do následujících kroků. Pokud jim bude hodnota přiřazena až později, není třeba ji zatím zadávat. Význam jednotlivých proměnných je popsán za lomítky. V případě proměnné „pressure“ bylo nutné použít datový typ long, neboť int by svým rozsahem nebyl dostačující.*

```
// 2.1 // Senzor tlaku 1 a deklarace proměnných
```

```
int sensor_1 = A1; // Senzor tlaku na pinu Analog 3
int pressureRaw_1; // Proměnná pro hodnotu ze sensoru (0-1023)
long pressure_1; // Proměnná pro tlak převedený na Pascaly
int pressurePercent_1; // Proměnná pro tlak převedený do procent
```

```
// 2.2 // Senzor tlaku 2 a deklarace proměnných
```

```
int sensor_2 = A3; // Senzor tlaku na pinu Analog 3
int pressureRaw_2; // Proměnná pro hodnotu ze sensoru (0-1023)
long pressure_2; // Proměnná pro tlak převedený na Pascaly
int pressurePercent_2; // Proměnná pro tlak převedený do procent
```

*Komentář // 3 //:*

*Deklarace proměnných a přiřazení digitálních pinů pro kompresor a ventily.*

*// 3 // Kompresor, ventily a deklarace proměnných*

```
int compressor_1 = 4; // Kompresor (1) pro dofukování na pinu Digital 4
int valveBleed_2 = 5; // Odpouštějící ventil (2) na pinu Digital 5
int valve_3 = 6; // Ventil (3) pro udržení tlaku v zóně 1 na pinu Digital 6
int valve_4 = 7; // Ventil (4) pro udržení tlaku v zóně 2 na pinu Digital 7
```

*Komentář // 4 //:*

*Deklarované proměnné „value“ později slouží jako rozhodující faktor pro cyklus dofukování a odpouštění, „i“ je pak pomocná proměnná pro kontrolu ustálení hodnoty na potenciometrech.*

*// 4 // Pomocné proměnné*

```
int value_1; // Proměnná pro vyhodnocení zda má dojít k nafukování, či odpouštění
int i_1; // Proměnná pro kontrolu, zda došlo k ustálení hodnoty na potenciometru 1 a má se spustit kalibrace tlaku v soustavě
int value_2; // Proměnná pro vyhodnocení zda má dojít k nafukování, či odpouštění
int i_2; // Proměnná pro kontrolu, zda došlo k ustálení hodnoty na potenciometru 2 a má se spustit kalibrace tlaku v soustavě
```

*Komentář // 5 //:*

*Deklarování pinů mezi I<sup>2</sup>C převodníkem a LCD panelem a definování adresy pro tok dat. Popis zkratk: en (enable), rw (read/write), rs (reset).*

*// 5 // Nastavení LCD*

```
LiquidCrystal_I2C lcd (0x27,2,1,0,4,5,6,7); // Nastavení pinů pro komunikaci mezi I2C chipem a LCD (I2C adresa, en, rw, rs, d4, d5, d6, d7)
```

*Komentář // 6 //:*

*Začátek jednorázového cyklu při spuštění programu označeného jako void setup a definování vstupů a výstupů na zvolených pinech a k nim přiřazených proměnných.*

*// 6 // Nastavení vstupů a výstupů*

```
void setup()
{
  pinMode (potentiometer_1, INPUT); // Nastavení Analog 0 jako vstup
  pinMode (sensor_1, INPUT);        // Nastavení Analog 1 jako vstup
  pinMode (potentiometer_2, INPUT); // Nastavení Analog 2 jako vstup
  pinMode (sensor_2, INPUT);        // Nastavení Analog 3 jako vstup
  pinMode (compressor_1, OUTPUT);   // Nastavení Digital 4 jako výstup
  pinMode (valveBleed_2, OUTPUT);   // Nastavení Digital 5 jako výstup
  pinMode (valve_3, OUTPUT);        // Nastavení Digital 6 jako výstup
  pinMode (valve_4, OUTPUT);        // Nastavení Digital 7 jako výstup
}
```

*Komentář // 7 //:*

*Aktivování displeje – definování jeho rozměrů, pinu podsvícení, polarizace a aktivování podsvícení.*

*// 7 // Aktivace LCD displeje*

```
lcd.begin (16,2); // Start displeje pro 16 x 2 LCD displej
lcd.setBacklightPin (3, POSITIVE); // Nastavení podsvícení (pin
podsvícení, polarizace podsvícení)
lcd.setBacklight (HIGH);
```

*Komentář // 8 //:*

*Aby v následující části programu (krok 10), kde dochází ke kontrole ustáleného stavu na potenciometru, mohlo dojít k vyhodnocení (porovnávání aktuálního stavu potenciometru oproti stavu předchozímu), je nutné tuto hodnotu uložit již při startu programu. Hodnota potenciometru je pomocí funkce „map“ přemapována z rozsahu 0-1023 na požadovaných 0-100 (procenta), funkce „abs“ zajistí, že hodnoty budou pouze v absolutní hodnotě. Tímto krokem také končí smyčka void setup.*



```
// 8 // Uložení aktuálního stavu nastavení potenciometru do paměti jako
„lastPotPercent“ - nutné pro krok 13
```

```
potRaw_1 = analogRead (potentiometer_1);
potPercent_1 = abs(map(potRaw_1, 0, 1023, 0, 100)); // Přemapování
hodnoty z potenciometru z 0-1023 na 0-100
lastPotPercent_1 = potPercent_1;

potRaw_2 = analogRead (potentiometer_2);
potPercent_2 = abs(map(potRaw_2, 0, 1023, 0, 100)); // Přemapování
hodnoty z potenciometru z 0-1023 na 0-100
lastPotPercent_2 = potPercent_2;
}
```

*Komentář // 9 //:*

*Jelikož určité části kódu se v průběhu programu několikrát opakují, bylo by nevhodné je pokaždé vypisovat. Ať už z důvodu přehlednosti nebo kvůli omezené paměti jednotky Arduina. V tomto kroku došlo k vytvoření funkcí pro aktualizaci hodnot z potenciometrů a převedení do požadovaného formátu, jejich vyvolání poté provedeme jednoduchým příkazem. Surová hodnota z tlakového sensoru je nejprve převedena do hodnoty tlaku v pascálech dle stupnice rozsahu sensoru 0-160 000 Pa, poté vydělena 300, aby-  
chom při tlaku označeného jako 100 % dosáhli požadovaného maxima 30 000 Pa. Proměnná i\_1 a i\_2 jsou pomocné proměnné pro vyhodnocování ustálení potenciometru během kroku 12.*

```
// 9.1 // Definování funkce pro aktualizaci hodnot pro zónu 1
```

```
void value_refresh_1()
{
  potRaw_1 = analogRead(potentiometer_1);
  potPercent_1 = abs(map(potRaw_1, 0, 1023, 0, 100));
  pressureRaw_1 = analogRead(sensor_1);
  pressure_1 = abs(map(pressureRaw_1, 0, 1023, 0, 160000));
  pressurePercent_1 = pressure_1/300;
  i_1 = 0;
}
```

```
// 9.2 // Definování funkce pro aktualizaci hodnot pro zónu 2
```

```
void value_refresh_2()
{
    potRaw_2 = analogRead(potentiometer_2);
    potPercent_2 = abs(map(potRaw_2, 0, 1023, 0, 100));
    pressureRaw_2 = analogRead(sensor_2);
    pressure_2 = abs(map(pressureRaw_2, 0, 1023, 0, 160000));
    pressurePercent_2 = pressure_2/300;
    i_2 = 0;
}
```

*Komentář // 10 //:*

*Zejména zde je možné vidět, že vytvoření funkce pojmenované jako LCD\_refresh pro výpis aktuálních hodnot na LCD panel, je více než na místě. LCD panel funguje způsobem vypsání hodnoty dle příkazu a tato hodnota zůstane zobrazována až do té doby, kdy je přepsána hodnotou jinou nebo do doby promazání celé obrazovky specifickým příkazem, který ale pro náš účel není vhodný, neboť při požadavku na rychlou odezvu by docházelo k problikávání obrazovky. Jako příklad je možné uvést výpis hodnoty 100 a následné snížení a vypsání hodnoty 99. Pokud bychom neprovedli promazávací kroky, na obrazovce by se objevilo 990. Funkce tedy nejprve začíná sadou podmínek if, které mají za úkol promazat nechtěné hodnoty z předchozího výpisu, poté dochází k výpisu aktuálních hodnot. Na každém řádku je označení konkrétní zóny, poté hodnota z potenciometru následující hodnotou aktuálního tlaku v procentech. Uživatel tak má informaci o nastavené hodnotě a pro kontrolu i o hodnotě skutečné.*

```
// 10 // Definování funkce pro aktualizaci hodnot na LCD panelu
```

```
void LCD_refresh()
{
    if (potPercent_1 < 100)
    {
        lcd.setCursor(11,0); lcd.print(" ");
    }
    if (potPercent_2 < 100)
    {
        lcd.setCursor(11,1); lcd.print(" ");
    }
}
```

```

if (potPercent_1 < 10)
{
    lcd.setCursor(10,0); lcd.print(" ");
}
if (potPercent_2 < 10)
{
    lcd.setCursor(10,1); lcd.print(" ");
}
if (pressurePercent_1 < 100)
{
    lcd.setCursor(15,0); lcd.print(" ");
}
if (pressurePercent_2 < 100)
{
    lcd.setCursor(15,1); lcd.print(" ");
}
if (pressurePercent_1 < 10)
{
    lcd.setCursor(14,0); lcd.print(" ");
}
if (pressurePercent_2 < 10)
{
    lcd.setCursor(14,1); lcd.print(" ");
}

lcd.home (); // Nastaví kurzor na pozici 0, řádek 0
lcd.print ("ZONA(1): "); // Vypíše text v závorce
lcd.print (potPercent_1, 1); // Vypíše hodnotu potPercent_1
lcd.setCursor (12,0); // Nastaví kurzor na pozici 12, řádek 0
lcd.print ("/"); // Vypíše text v závorce
lcd.print (pressurePercent_1, 1); // Vypíše hodnotu pressurePercent_1
lcd.setCursor (0,1); // Nastaví kurzor na pozici 0, řádek 1
lcd.print ("ZONA(2): "); // Vypíše text v závorce
lcd.print (potPercent_2, 1); // Vypíše hodnotu potPercent_2
lcd.setCursor (12,1); // Nastaví kurzor na pozici 12, řádek 1
lcd.print ("/"); // Vypíše text v závorce
lcd.print (pressurePercent_2, 1); // Vypíše hodnotu pressurePercent_2
delay (50); // Vyčká 50 ms
}

```

*Komentář // 11 //:*

*Nyní začíná start samotného pracovního jádra programu – opakujícího se cyklu void loop. Jako první krok je aktualizace všech vstupních hodnot, tzn. hodnoty z potenciometrů a hodnoty ze sensorů tlaku. Vyvolání je provedeno pomocí krátkého odkazu na předešle definované funkce.*

*// 11 // Načtení hodnot na senzoru a potenciometru na začátku cyklu*

```
void loop()
{
  value_refresh_1(); // Vyvolá funkci definovanou v kroku 9.1
  value_refresh_2(); // Vyvolá funkci definovanou v kroku 9.2
```

*Komentář // 12 //:*

*Během následujícího cyklu while dojde k záznamu a sečtení třech po sobě načtených hodnot z potenciometru a vydělení poslední načtenou hodnotou. Pokud výsledek není roven třem (toho je dosaženo, pokud všechny hodnoty jsou totožné), cyklus nepustí program do další části. Dochází k vyhodnocování potenciometru 1 i 2 zároveň a stačí, že není dodržena podmínka alespoň na jednom z nich. Je možné si všimnout, že ke každé hodnotě potPercent je připočtena jednička. Tento krok nijak neovlivní funkčnost, ale je nezbytný pro případ, kdy hodnota na potenciometru je rovna nule, došlo by tak k zacyklení neboť by došlo na dělení nulou.*

*// 12 // Cyklus pro kontrolu, zda došlo k ustálení hodnoty na potenciometrech a má se spustit kalibrace tlaku v soustavě*

```
while ((i_1 / (potPercent_1 + 1) != 3) || (i_2 / (potPercent_2 + 1)
!= 3))

{
  potRaw_1 = analogRead(potentiometer_1);
  potPercent_1 = abs(map(potRaw_1, 0, 1023, 0, 100));
  i_1 = i_1 + (potPercent_1 + 1);

  potRaw_2 = analogRead(potentiometer_2);
  potPercent_2 = abs(map(potRaw_2, 0, 1023, 0, 100));
  i_2 = i_2 + (potPercent_2 + 1);
```

```

delay(100);

    if ((i_1 > 3 * (potPercent_1 + 1)) || (i_2 > 3 * (potPercent_2 +
1)))
    {
        i_1 = 0;
        i_2 = 0;
    }

    LCD_refresh(); // Výpis hodnot na obrazovku
}

```

*Komentář // 13.1 //:*

*Spustí se cyklus pro kalibraci vzduchu v zóně 1, pokud je zaznamenána změna na potenciometru 1 větší než 1. Proměnné value\_1 je přiřazena menší z hodnot - potenciometru nebo tlaku. Podmínka if dále pokračuje v kroku 13.1.1 a 13.1.2.*

*// 13.1 // Cyklus rozhodování pro ZÓNU (1), zda při změně hodnoty na potenciometru dojde k dofukování, či odpouštění*

```

if ((potPercent_1 - 1) > lastPotPercent_1 || (potPercent_1 + 1) <
lastPotPercent_1) // Spustí cyklus, pokud je změna na potenciometru
větší než 1
{
    value_1 = min(pressurePercent_1, potPercent_1); // Rozhodne a přiřadí
proměnné value_1 menší z hodnot v závorce
}

```

*Komentář // 13.1.1 //:*

*Pokud hodnota value\_1 je rovna hodnotě pressurePercent\_1 spustí se cyklus dofukování a opakuje se až do té doby, dokud tlak v soustavě 1 má menší hodnotu než hodnota nastavená na potenciometru 1.*

*// 13.1.1 // Cyklus DOFUKOVÁNÍ pro ZÓNU (1)*

```

if (value_1 == pressurePercent_1) // Spustí cyklus, pokud je menší
hodnotou hodnota tlaku
{

```

```

while (pressurePercent_1 < potPercent_1) // Opakuje, dokud hodnota
tlaku je menší než hodnota potenciometru
{
    digitalWrite(compressor_1, HIGH); // Sepne kompresor
    digitalWrite(valve_3, HIGH); // Otevře ventil okruhu 1

    // Aktualizace hodnot tlaku a potenciometru pro zónu 1 a 2 a
    aktualizace LCD
    value_refresh_1();
    value_refresh_2();
    LCD_refresh();
}

lastPotPercent_1 = potPercent_1; // Po dokončení cyklu zaktuali-
zuje porovnávací hodnotu pro potenciometr 1
digitalWrite(compressor_1, LOW); // Vypne kompresor
digitalWrite(valve_3, LOW); // Uzavře ventil okruhu 1
}

```

*Komentář // 13.1.2 //:*

*Pokud hodnota value\_1 je rovna hodnotě potPercent\_1, spustí se cyklus dofukování a opakuje se až do doby, dokud tlak v soustavě 1 má větší hodnotu než hodnota nastavená na potenciometru 1. Na konci bloku je podmínka z kroku 13.1 ukončena.*

*// 13.1.2 // Cyklus ODPOUŠTĚNÍ pro ZÓNU (1)*

```

if (value_1 == potPercent_1) // Spustí cyklus, pokud je menší hod-
notou hodnota potenciometru
{
    while (pressurePercent_1 > potPercent_1) // Opakuje, dokud hodnota
    tlaku je větší než hodnota potenciometru
    {
        digitalWrite(valveBleed_2, HIGH); // Otevře ventil pro odpouštění
        digitalWrite(valve_3, HIGH); // Otevře ventil okruhu 1

        // Aktualizace hodnot tlaku a potenciometru pro zónu 1 a 2 a
        aktualizace LCD
        value_refresh_1();
        value_refresh_2();
    }
}

```

```

    LCD_refresh();
}

lastPotPercent_1 = potPercent_1; // Po dokončení cyklu zaktualizuje porovnávací hodnotu pro potenciometr 1
digitalWrite(valveBleed_2, LOW); // Otevře ventil pro odpouštění
digitalWrite(valve_3, LOW); // Otevře ventil okruhu 1
}
}

```

*Komentář // 13.2 //:*

*Spustí se cyklus pro kalibraci vzduchu v zóně 2, pokud je zaznamenána změna na potenciometru 2 větší než 1. Proměnné value\_2 je přiřazena menší z hodnot - potenciometru nebo tlaku. Podmínka if dále pokračuje v kroku 13.2.1 a 13.2.2.*

*// 13.2 // Cyklus rozhodování pro ZÓNU (2), zda při změně hodnoty na potenciometru dojde k dofukování, či odpouštění*

```

if ((potPercent_2 - 1) > lastPotPercent_2 || (potPercent_2 + 1) <
lastPotPercent_2) // Spustí cyklus, pokud je změna na potenciometru
větší než 1
{
    value_2 = min(pressurePercent_2, potPercent_2); // Rozhodne a přiřadí
proměnné value_2 menší z hodnot v závorce
}

```

*Komentář // 13.2.1 //:*

*Pokud hodnota value\_2 je rovna hodnotě pressurePercent\_2 spustí se cyklus dofukování a opakuje se až do té doby, dokud tlak v soustavě 2 má menší hodnotu než hodnota nastavená na potenciometru 2.*

*// 13.1.1 // Cyklus DOFUKOVÁNÍ pro ZÓNU (1)*

```

if (value_2 == pressurePercent_2) // Spustí cyklus, pokud je menší
hodnotou hodnota tlaku
{
    while (pressurePercent_2 < potPercent_2) // Opakuje, dokud hodnota
tlaku je menší než hodnota potenciometru
    {

```

```

{
    digitalWrite(compressor_1, HIGH); // Sepne kompresor
    digitalWrite(valve_4, HIGH); // Otevře ventil okruhu 2

    // Aktualizace hodnot tlaku a potenciometru pro zónu 1 a 2 a
    aktualizace LCD
    value_refresh_1();
    value_refresh_2();
    LCD_refresh();
}

lastPotPercent_2 = potPercent_2; // Po dokončení cyklu zaktuali-
zuje porovnávací hodnotu pro potenciometr 2
digitalWrite(compressor_1, LOW); // Vypne kompresor
digitalWrite(valve_4, LOW); // Uzavře ventil okruhu 2
}

```

*Komentář // 13.2.2 //:*

*Pokud hodnota value\_2 je rovna hodnotě potPercent\_2 spustí se cyklus dofukování a opakuje se až do doby, dokud tlak v soustavě 2 má větší hodnotu než hodnota nastavená na potenciometru 2. Na konci bloku je podmínka z kroku 13.2 ukončena.*

*// 13.2.2 // Cyklus ODPOUŠTĚNÍ pro ZÓNU (1)*

```

if (value_2 == potPercent_2) // Spustí cyklus, pokud je menší hod-
notou hodnota potenciometru
{
    while (pressurePercent_2 > potPercent_2) // Opakuje, dokud hodnota
    tlaku je větší než hodnota potenciometru
    {
        digitalWrite(valveBleed_2, HIGH); // Otevře ventil pro odpouštění
        digitalWrite(valve_4, HIGH); // Otevře ventil okruhu 2

        // Aktualizace hodnot tlaku a potenciometru pro zónu 1 a 2 a
        aktualizace LCD
        value_refresh_1();
        value_refresh_2();
        LCD_refresh();
    }
}

```



```

    lastPotPercent_2 = potPercent_2; // Po dokončení cyklu zaktuali-
    zuje porovnávací hodnotu pro potenciometr 2
    digitalWrite(valveBleed_2, LOW); // Otevře ventil pro odpouštění
    digitalWrite(valve_4, LOW); // Otevře ventil okruhu 2
  }
}

```

*Komentář // 14 //:*

*Ačkoli jedna z primárních vlastností finálního výrobku určeného pro trh je dokonalá těsnost a nemělo by tak docházet k žádným únikům vzduchu, je dobré i tento případ v programu ošetřit. Aby byla nastavena určitá hystereze, rozmezí, ve kterém se požadovaný tlak má pohybovat, cyklus re-kalibrace se spustí při snížení tlaku v soustavě o více jak 2 % a k ukončení re-kalibrace dojde po dosažení tlaku o 2 % většího, než je hodnota nastavená na potenciometru.*

```

// 14.1 // Dofukování v případě poklesu tlaku v ZÓNĚ (1)
if (potPercent_1 > (pressurePercent_1 + 2)) // Spustí cyklus, pokud tlak
poklesne o víc jak 2% nastavené hodnoty
{
    while ((potPercent_1 + 2) > pressurePercent_1) // Dofukuje soustavu,
    dokud není dosaženo tlaku o 2% větší než hodnota potenciometru
    {
        digitalWrite(compressor_1, HIGH); // Sepne kompresor
        digitalWrite(valve_3, HIGH); // Otevře ventil okruhu 1

        // Aktualizace hodnot tlaku a potenciometru pro zónu 1 a 2 a
        aktualizace LCD
        value_refresh_1();
        value_refresh_2();
        LCD_refresh();
    }

    digitalWrite(compressor_1, LOW); // Vypne kompresor
    digitalWrite(valve_3, LOW); // Uzavře ventil okruhu 1
}

```

```
// 14.2 // Dofukování v případě poklesu tlaku v ZÓNĚ (2)

if (potPercent_2 > (pressurePercent_2 + 2)) // Spustí cyklus, pokud tlak
poklesne o víc jak 2% nastavené hodnoty
{
    while ((potPercent_2 + 2) > pressurePercent_2) // Dofukuje soustavu,
dokud není dosaženo tlaku o 2% větší než hodnota potenciometru
    {
        digitalWrite(compressor_1, HIGH); // Sepne kompresor
        digitalWrite(valve_4, HIGH); // Otevře ventil okruhu 2

        // Aktualizace hodnot tlaku a potenciometru pro zónu 1 a 2 a
        aktualizace LCD
        value_refresh_1();
        value_refresh_2();
        LCD_refresh();
    }

    digitalWrite(compressor_1, LOW); // Vypne kompresor
    digitalWrite(valve_4, LOW); // Uzavře ventil okruhu 2

}

```

*Komentář // 15 //:*

*Posledním krokem cyklu je vypsání aktuálních hodnot na LCD panel pro případ, kdy nedochází ke kalibraci soustavy, ani změně na potenciometru, ale pouze k případné změně tlaku v soustavě. Tímto končí cyklus void loop a vrací se zpět na začátek.*

```
// 15 // Aktualizace hodnot na displej

```

```
    LCD_refresh();
}

```

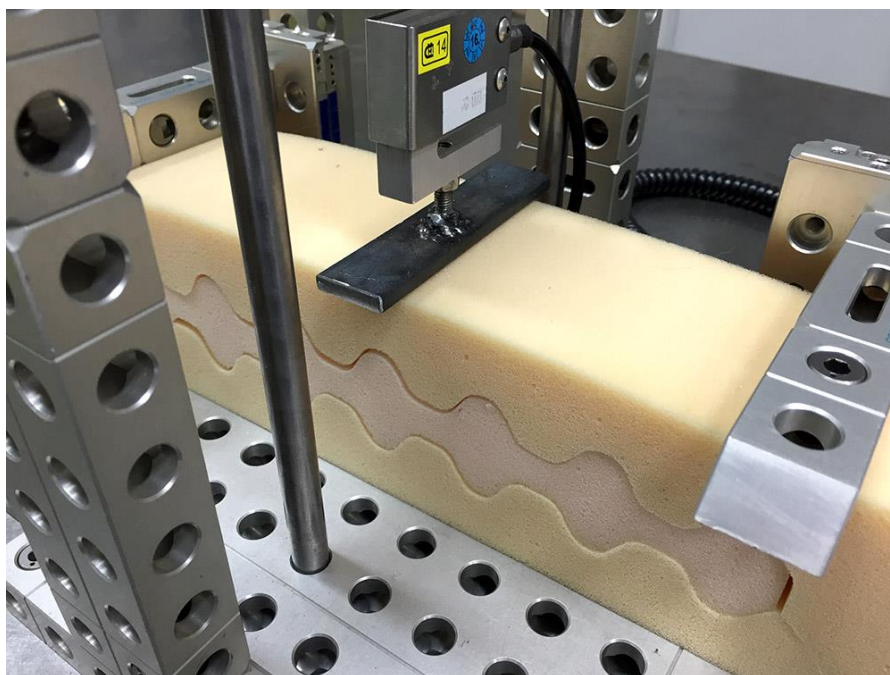
## 6 NÁVRH VARIABILNÍHO ČLENU MATRACE

### 6.1 Stanovení cílů

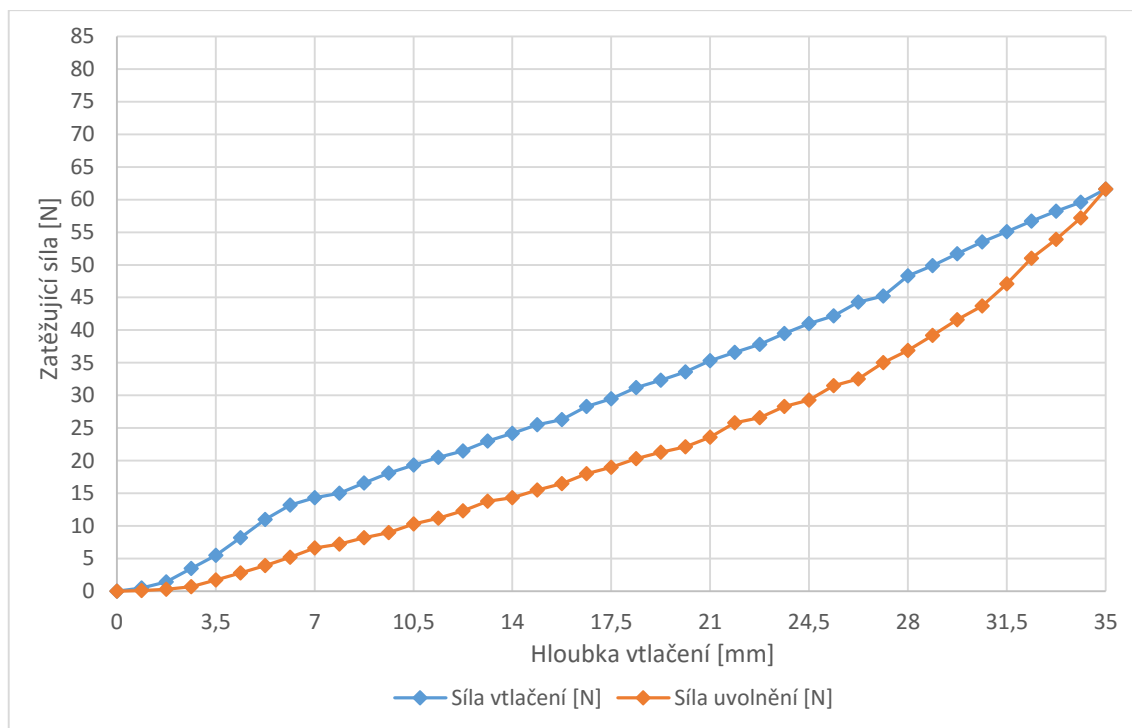
Aby bylo možné jednotlivé relevantní varianty hodnotit, bylo nejprve nutné stanovit požadované cíle, kterých chceme dosáhnout. Primárním a hlavním parametrem byla především tuhost a rozsah její nastavitelnosti. Abychom měli určitý reálný základ, zvolil jsem jako porovnávací hodnotu dostupný rozsah nastavení tuhosti pro nosníky matrace typu SARA. Jak již bylo zmíněno na začátku diplomové práce, molitanové vložky ovlivňující tuhost jsou dodávány v několika tuhostních variantách, přičemž růžová vložka je nejtvrdší a zelená naopak nejměkčí možná varianta. Typ nosníku jsem zvolil žlutý – nejtvrdší možný, bylo cílem dosáhnout co největšího extrému. Tyto dvě varianty (žlutý nosník s růžovou vložkou a žlutý nosník se zelenou vložkou) byly následně podrobeny zkoušce na sestaveném měřicím zařízení a výsledky poslouží jako referenční hodnoty pro návrh variabilních členů.

#### 6.1.1 Tuhost sestavy žlutý nosník + růžová vložka

V tomto případě došlo k měření nejtvrdší možné varianty, tedy žlutého PUR nosníku a růžové PUR vložky. Měření probíhalo stejným způsobem a za stejných podmínek jako testovací měření v rámci ověření způsobilosti měřicího zařízení.



**Obrázek 6.1.1 – 1:** Měřená soustava žlutý nosník + růžová vložka

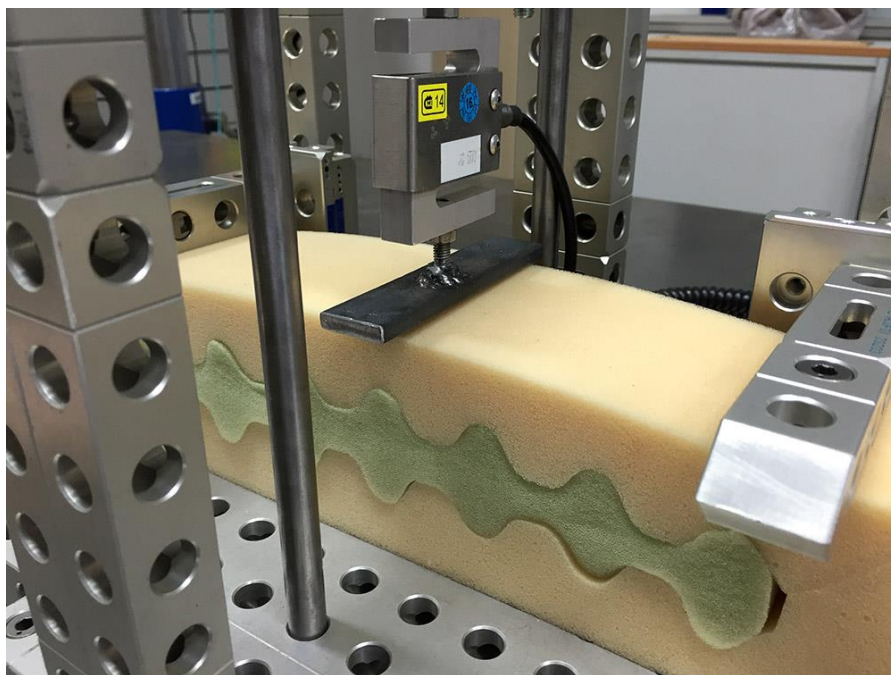


**Graf 6.1.1 – 1:** Naměřená tuhost soustavy žlutý nosník + růžová vložka

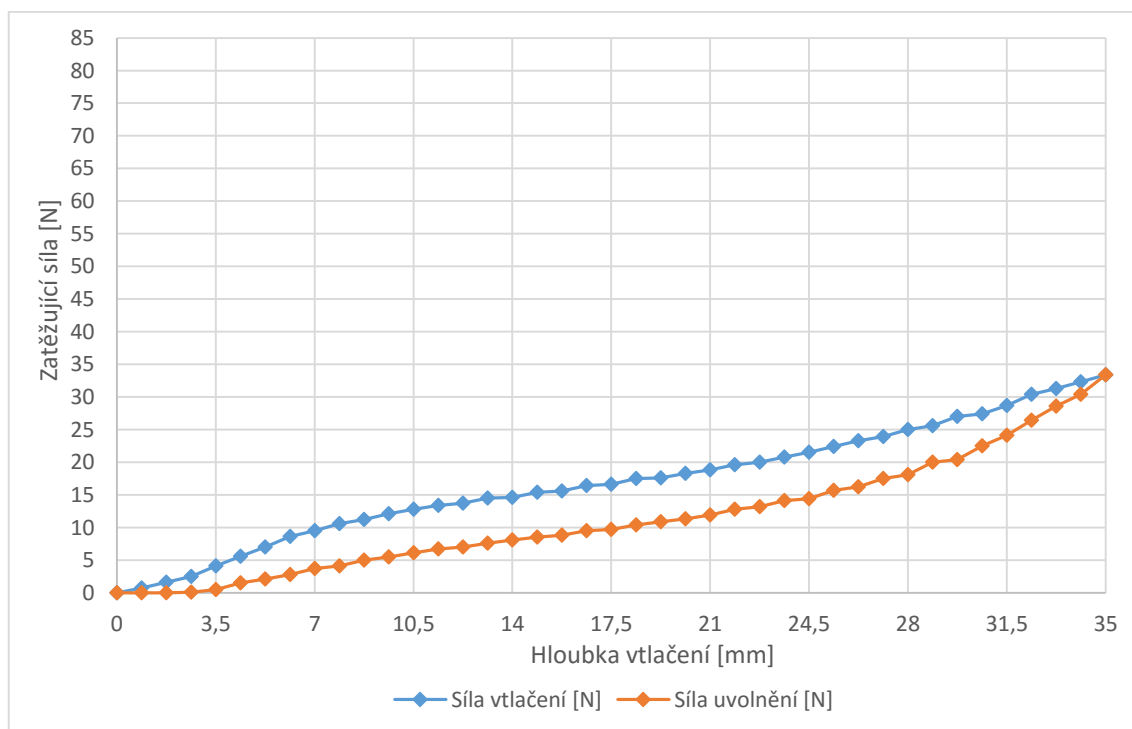
Jak výsledky naznačují, průběh tuhosti se oproti měření na celistvé pěně více zlinearizoval a hysterezní křivka již není tak viditelná, jako v případě verifikačního měření měřicího zařízení. Tento jev je způsoben především narušením celistvosti pěny tvorbou výřezu pro vložku, kdy dojde k přerušení vazeb uvnitř pěny. Zatěžující síla až na mírný počáteční skok plynule stoupá na naměřené maximum 61,6 N.

### 6.1.2 Tuhost sestavy žlutý nosník + zelená vložka

Kombinace žlutého nosníku a zelené vložky není zcela běžná, neboť se jedná o kombinaci nejtvrďšího a nejměkčího členu a výrobcem není ani primárně standardně nabízena. Vložky doporučené výrobcem se používají s tuhostí o řád, maximálně dva nižší. V případě zelené vložky jsou to řády tři. Nicméně pro náš záměr, dosažení extrému je dobré tyto hodnoty znát.



**Obrázek 6.1.2 - 1:** Měřená soustava žlutý nosník + zelená vložka



**Graf 6.1.2 - 1:** Naměřená tuhost soustavy žlutý nosník + zelená vložka

V porovnání s grafem tvrdé varianty má zelená vložka znatelné menší strmou průběhu naměřených hodnot. Dosažené maximum je 33,4 N, což je téměř poloviční tuhost. Stoupání je relativně konstantní s lehkým náznakem hysterezní křivky na začátku zátěže a dále na začátku uvolňování.

### 6.1.3 Shrnutí požadovaného cíle

Naměřené hodnoty měkké a tvrdé sestavy poslouží jako výchozí bod pro návrh optimálního rozsahu nastavitelnosti tuhosti. Měkká varianta s maximem 33,4 N by měla být chápána skutečně jako hraniční, protože se jedná o extrémně nízkou tuhost, které v reálném provozu pravděpodobně nikdy nebude vyžadována. Není tedy nezbytné, aby navržená regulace umožňovala dosahovat tak nízké hodnoty, nebo ji dokonce i překonávat. Naměřené hodnoty tuhé sestavy s maximem 61,6 N je naopak vhodné brát v úvahu jako velice relevantní, protože to se jedná o variantu, která je výrobcem primárně nabízena. Určitě by tedy bylo vhodné, dosáhnout i lehce vyšších hodnot pro případné nastavení velmi tvrdé matrace.

## 6.2 Návrh variant řešení

Při návrhu způsobu regulace tuhosti molitanového členu jsem vycházel z běžně dostupných materiálů a zásady, že řešení by nemělo být nijak extrémně drahé a především by mělo být snadno aplikovatelné a například i odolné pro transport matrace ve srolovaném stavu.

### 6.2.1 VARIANTA „A“ – pogumování

Inspiraci pro první návrh jsem čerpal z běžně prodávaných pogumovaných rukavic, kdy část rukavice, dlaň a prsty jsou potaženy vrstvou gumy zabraňující kontaktu pokožky s tekutinou, či propustnost prachu. Záměrem bylo potáhnout touto gumou nejměkčí zelenou vložku a umožnit tak zvyšování její tuhosti nafukováním. První problém nastal se sháněním tekuté gumy s obdobnými vlastnostmi, trh je v tomto ohledu velice slabý a volba po zvážení padla na tekutou gumu ve spreji PlastiDip. Výrobcem udávaná pružnost je 300 až 400 %, kterou si zachovává i při extrémních teplotních podmínkách -34 až +93 °C. Pořizovací cena spreje v černé barvě byla 299 Kč.



**Obrázek 6.2.1 – 1:** Guma ve spreji PlastiDip

Před nanášením na vložku matrace došlo nejprve k otestování požadovaných vlastností na drobném molitanovém kusu.

Při testovacím nanášení se bohužel projevila nečekaná, pro náš účel negativní vlastnost, kterou je příliš nízká viskozita gumy ve spreji. I při několikanásobné vrstvě docházelo k zatečení gumy do pórů pěny. Nepodařilo se tak vytvořit požadovanou neprodyšnou vrstvu na povrchu. Řešením by možná bylo nanášení tekuté gumy štětcem, nicméně tyto výrobky jsou poměrně drahé a s ohledem na nejistý výsledek bylo od této varianty upuštěno.



**Obrázek 6.2.1 – 2:** Testování pogumování

Vyhodnocení varianty „A“: **Nevyhovující**



### 6.2.2 VARIANTA „B“ – Vzduchový vak

Při návrhu vyhovujícího řešení byla snaha využít již vytvořených výřezů uvnitř nosníku. Jako další možné řešení bylo umístit vzduchový vak do měkké vložky, který při dofukování bude zvyšovat její tuhost. Bohužel umístit vak do původní zelené vložky nebylo z důvodu jejích tenkých žeber možné, použit byl tedy náhradní, také měkký molitan obdélníkového tvaru. Prototyp nafukovacího vaku byl vytvořen ze silnostěnné fólie, která byla po obvodu slepená lepidlem a spoje pro jistotu přelepeny pevnou textilní páskou. Případné mikro netěsnosti byly eliminovány pomocí závěrečného nanesení gumové vrstvy.



**Obrázek 6.2.2 – 1:** Gumový vak a vyřezaná molitanová vložka



**Obrázek 6.2.2 – 2:** Nafouknutý vak uvnitř molitanové vložky





**Obrázek 6.2.2 – 3:** Test vaku uvnitř nosníku

Jak fotka č. 6.2.2 - 3 naznačuje, při testu chování vložky uvnitř molitanového nosníku bohužel došlo k nepříjemnému vyboulení střední části vaku. Minimálně s ohledem na vizuální dojem postele, jistě není přijatelné, aby v určitých místech vznikaly jakékoli nerovnosti. Tento jev je možné pozorovat už na samotné vložce na fotce 6.2.2 – 2. Možným řešením by bylo svaření stěn vaku bodově, či podélně s určitým rozestupem. Avšak i tak by byl výsledek nejistý. Proto bylo rozhodnuto od této varianty upustit.

Vyhodnocení varianty „B“: **Nevyhovující**

### 6.2.3 VARIANTA „C“ – Hadice uvnitř molitanové vložky

I přes předchozí neúspěšné varianty byla stále vůle zachovat a pouze upravit dosavadní řešení pomocí vyměnitelných vložek. Možným způsobem se nabízelo navrtání molitanové vložky, což má za následek snížení její tuhosti a do navrtaných otvorů umístit silikonovou tenkostěnnou hadici, která při zvyšujícím tlaku bude vyplňovat otvory a zvyšovat tak tuhost samotného členu.

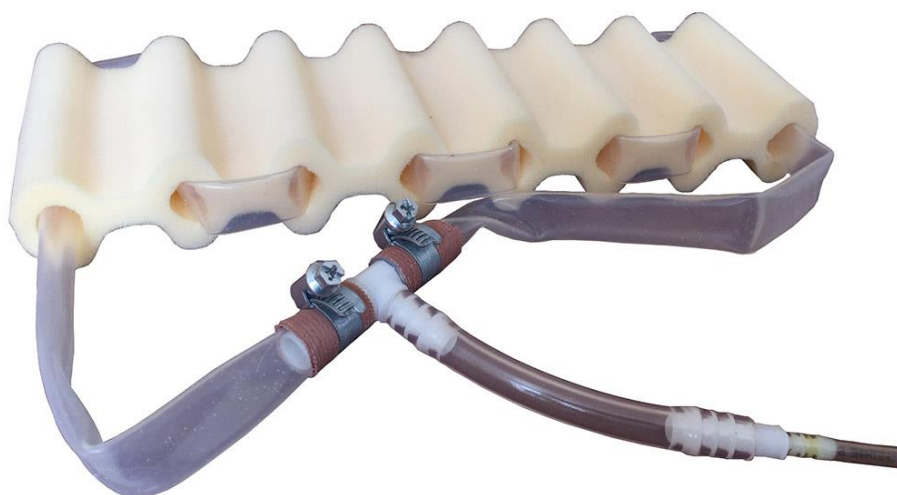
Hlavní problém, který bylo nutné vyřešit, byl způsob vrtání otvoru, neboť použití vrtáku nepřicházelo v úvahu. Docházelo by k zakousnutí a namotání pěny na vrták. Proto jsem zvolil metodu malého brusného kotouče o průměru 14mm v kombinaci s ruční vysokorychlostní bruskou. Vysoké otáčky a jemný brusný kotouč měly

zajistit hladký průnik skrz pěnu. Protože se jednalo o první experiment s vybrušováním pěny, zvolil jsem vložku ze žluté pěny, u které byly předpokládány nejmenší, či žádné komplikace při vybrušování.



**Obrázek 6.2.3 – 1:** Vybrušování pěnové vložky

Zvolena byla transparentní hadice ze silikonové pryže o rozměrech 14 mm pro vnější průměr a 12 mm pro průměr vnitřní, která sama o sobě dle pocitového dojmu působila velice měkce.



**Obrázek 6.2.3 – 2:** Vložka osazená hadicí



**Obrázek 6.2.3 – 3:** Test vložky osazené hadicí uvnitř nosníku

První dojmy z testované varianty C byly velice pozitivní, hadice plnila svůj účel tak, jak se předpokládalo, nedocházelo k nevzhledné deformaci nosníku a vložka při plně nafouknutém stavu působila skutečně tvrdě. Bohužel podstatným negativem byla tuhost vložky při zcela vyfouknutém stavu. Ačkoli hadice sama o sobě je velice dobře poddajná, tak v místech, kde je hadice ohýbána, vznikají tvrdé přechody mající značný dopad na celkovou tuhost. Při zcela vypuštěném stavu vložka dosahuje odhadem podobné tuhosti, jako vložka růžová, což je velice špatný výsledek. Nicméně bylo ověřeno, že použití hadice by mohlo být správnou cestou, jen je nutné udělat určité změny.

Vyhodnocení varianty „C“: **Nevyhovující**

#### 6.2.4 VARIANTA „D“ – Hadice skrz nosník (verze 1)

V předchozím kroku navrženou variantou byly vyčerpány možnosti využití pěnové vložky a hotového výřezu, nicméně bylo ověřeno, že cesta formou nafukování hadice, která je součástí vložky, či nosníku, by mohla být správnou cestou. Dalším návrhem tedy bylo probroušení samotného nosníku. Na otvory jsem zvolil brusný kotouč o průměru 32 mm a jako hadice byla použita bezešvá cyklistická duše s průměrem v nafouknutém 30 mm. Duši jsem zvolil především díky její cenové dostupnosti a obdobným vlastnostem jako má silikonová hadice, což je dobrá poddajnost a síla stěny 1 mm. Pro zkušební variantu byly vybroušeny čtyři otvory.



**Obrázek 6.2.4 – 1:** Probroušené otvory o průměru 32 mm



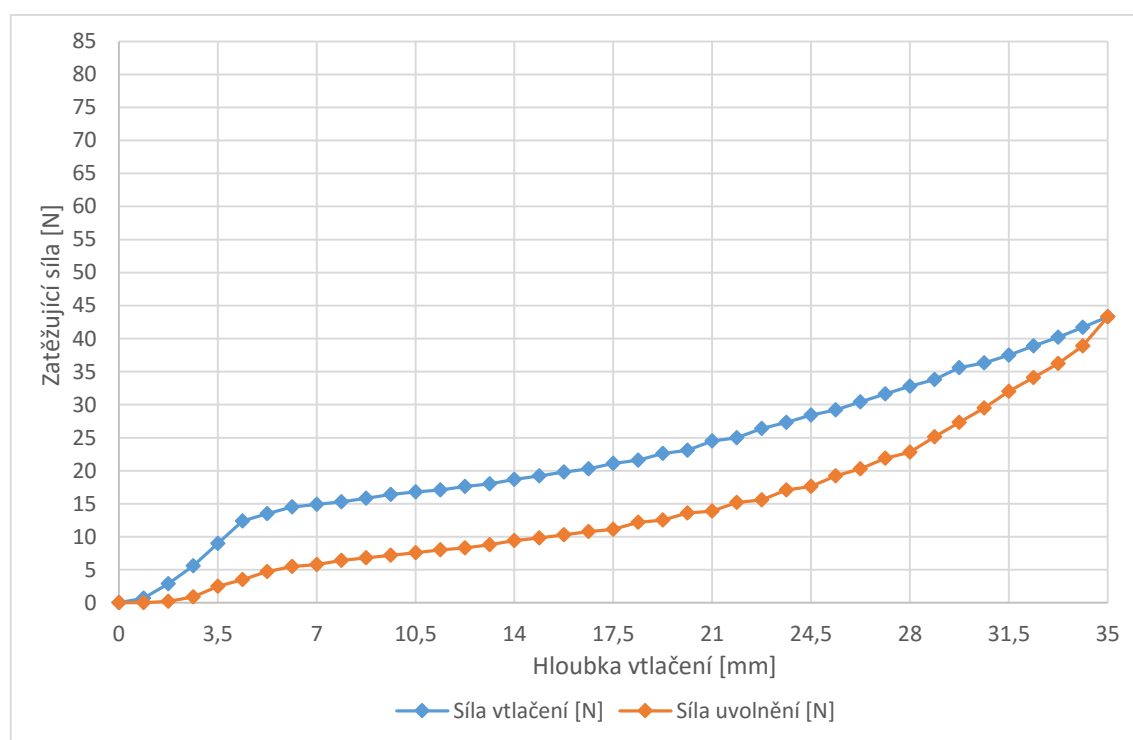
**Obrázek 6.2.4 – 2:** Nosník osazený duší



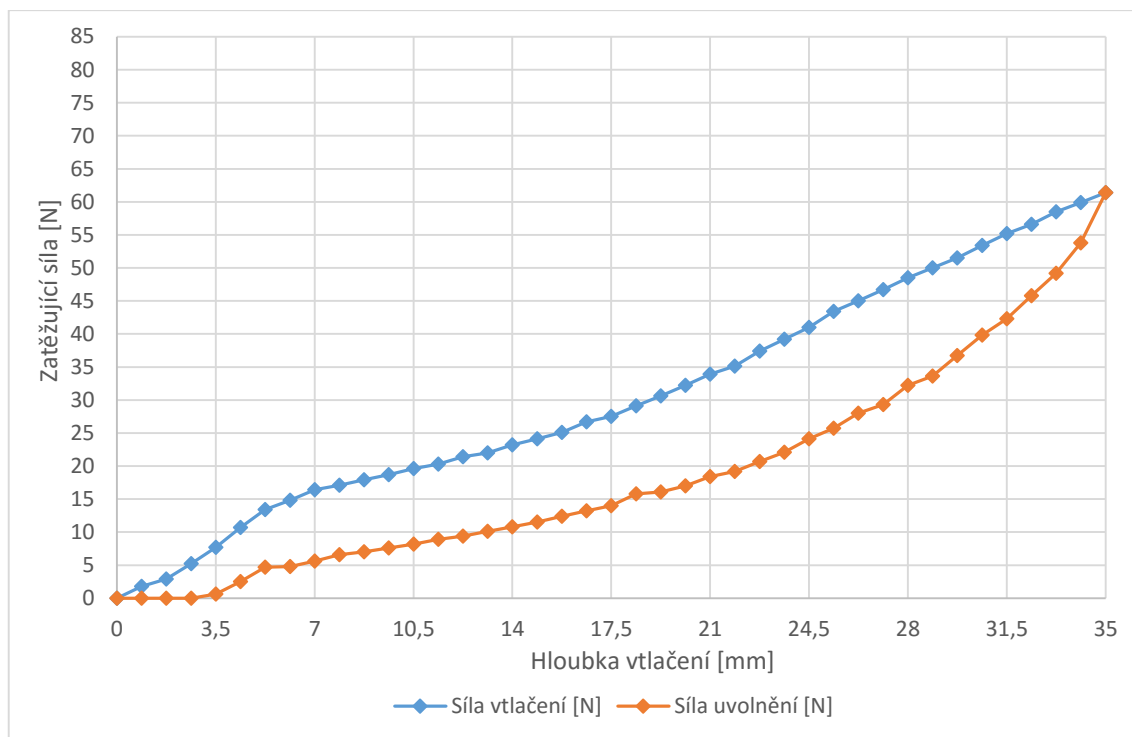
**Obrázek 6.2.4 – 3:** Nosník s duší v nafouklém stavu

První dojmy varianty D, kde nafukovaným členem byla duše o průměru 30 [mm], byly velice dobré. Neprojevila se žádná deformace pěny, nosník byl při vyfouknutém stavu znatelně měkčí, než v ostatních částech. Naopak při plném nahuštění, regulované místo působilo solidně a tuze. Jediné co by se dalo vytknout, je přílišné vyboulení na boku pěny z důvodu, že je použita duše o velkém průměru.

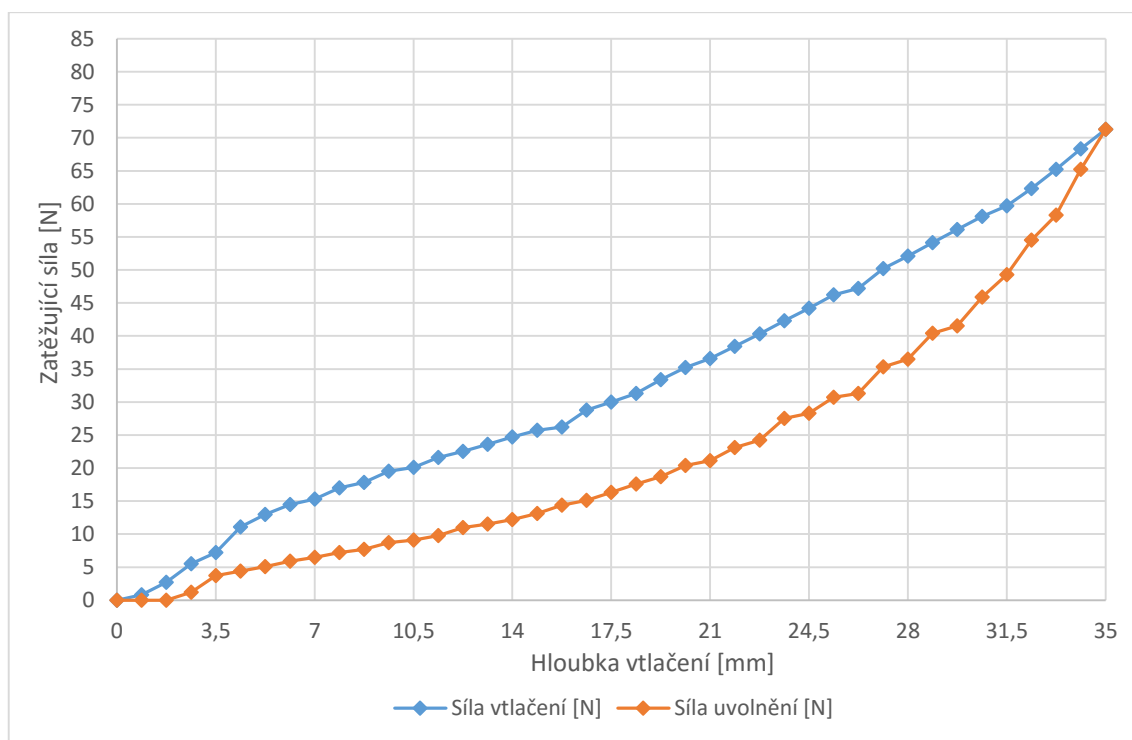
Abychom ověřili výše zmíněné poznatky, bylo nutné provést na navržené variantě kontrolní měření tuhosti. Mezní hranice pro měření byly zvoleny na 0, 50 a 100 % nastavení tuhosti, tedy 0, 15 000 a 30 000 Pa.



**Graf 6.2.4 - 1:** Kontrolní měření varianty D – [tuhost 0 %]



**Graf 6.2.4 - 2: Kontrolní měření varianty D - [tuhost 50 %]**



**Graf 6.2.4 - 3: Kontrolní měření varianty D - [tuhost 100 %]**



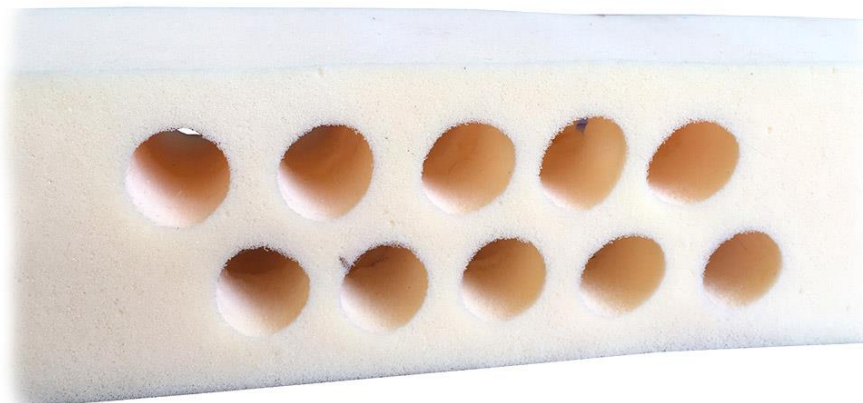
Grafy potvrzují očekávané výsledky v rozdílu tuhosti u nastavení 0 a 100 %. Při zcela odlehčeném stavu, v konečné fázi zatížení, bylo dosaženo zatížení 43,3 N, naopak horní hranice při maximálním tlaku dosáhla úrovně 71,3 N. Výsledný rozdíl maximální a minimální úrovně tak je 28 N. Průběh křivky ve všech případech vykazuje na počátku mírný skok, poté následuje průběh lehce progresivní. Tyto výkyvy mohou být způsobeny nehomogenitou regulované oblasti, neboť byla použita pouze jedna řada otvorů s velkým průměrem. Za zmínku také stojí, všimnout si rozdílu mezi tuhostí 0 a 50 %, kde rozdíl pro konečnou fázi činí 18,1 N, zatímco v případě 50 až 100 % je to pouhých 9,9 N. Z toho lze usuzovat, že zvýšení tuhosti není přímo úměrné navyšování tlaku.

V případě maximální tuhosti lze zhodnotit variantu D jako úspěšnou, neboť se podařilo dosáhnout požadovaného cíle a dokonce ho i mírně překonat. Ve spodní oblasti nastavení tuhosti (0 – 50 %) jsou výsledky bohužel stále relativně vysoké, ale i tak by tato varianta mohla být použita.

Vyhodnocení varianty „D“: **Vyhovující s výhradou**

### 6.2.5 VARIANTA „E“ – Hadice skrz nosník (verze 2)

Poznatky z varianty D nám umožnily posunout se opět o kus dál a tak cílem varianty E bylo odstranit zmiňovaná negativa a případně ještě vylepšit vlastnosti pozitivní. Hlavním problémem, který provázel variantu D, byla nehomogenita řízené oblasti, proto jsem se nyní rozhodl použít hadici o menším průměru a vytvořit dvě řady otvorů, které vůči sobě budou částečně posunuté. Otvory byly vrtány brusným kotoučkem o průměru 23 mm a jako tuhostní člen jsem zvolil opět silikonovou hadici o vnějším průměru 22 mm a vnitřním 20 mm. Hadice je v černém provedení a dle specifikace výrobce, by se měla vyznačovat jednou z nejlepších poddajností ze sortimentu prodávaných silikonových hadic.



**Obrázek 6.2.4 – 1:** Probroušené otvory o průměru 23 mm



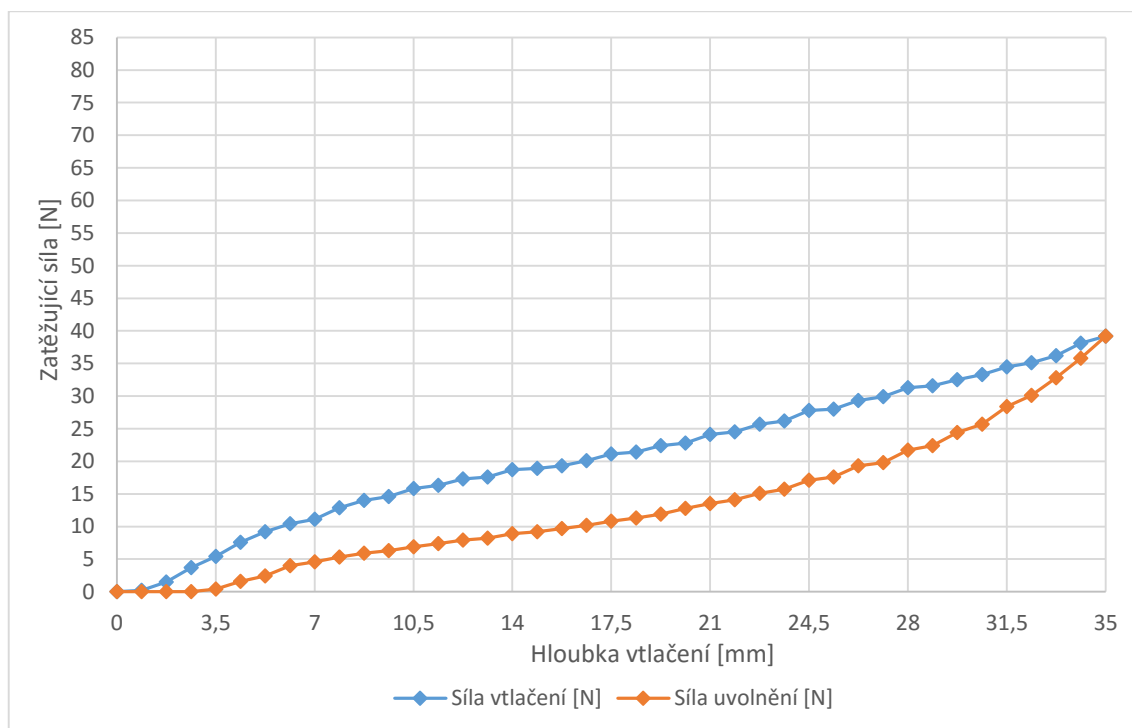
**Obrázek 6.2.4 – 2:** Nosník osazený hadicí



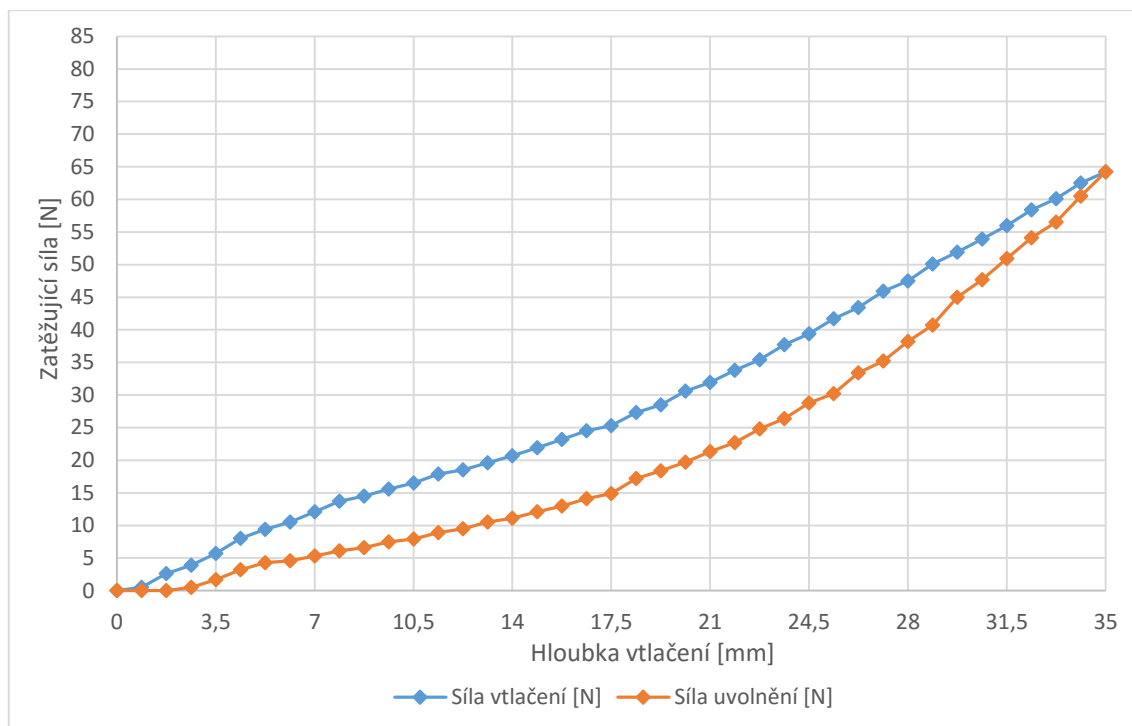
**Obrázek 6.2.4 – 3:** Nosník s hadicí v nafouklém stavu



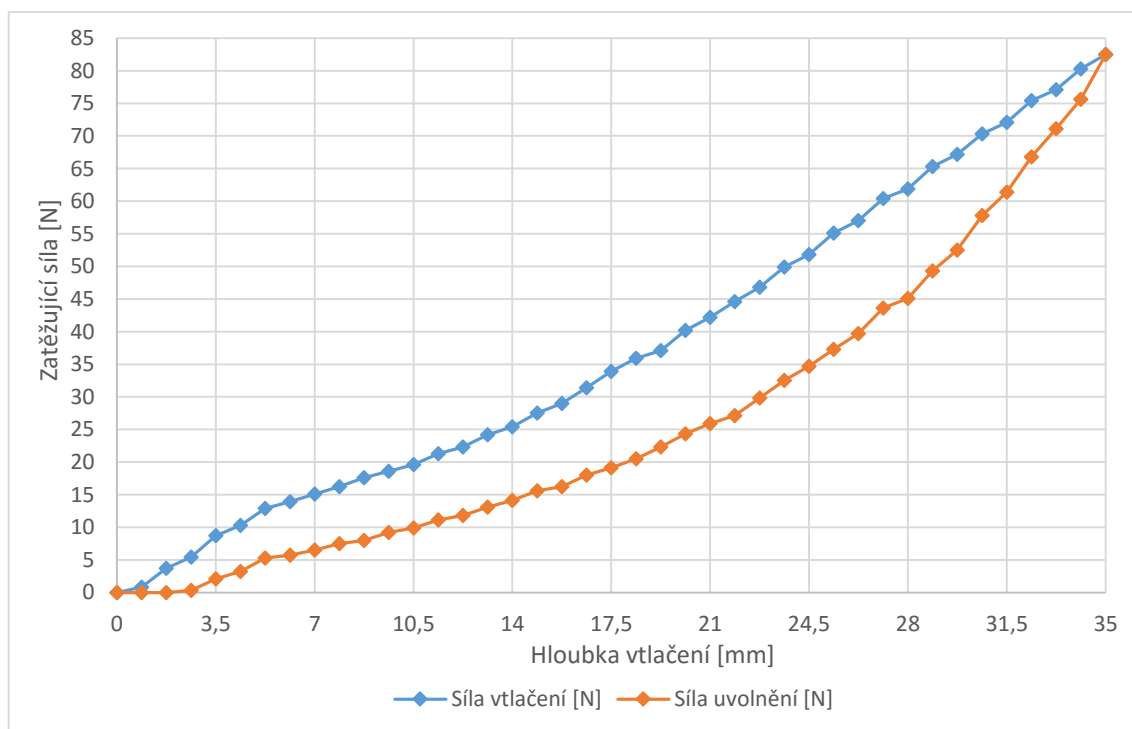
Prvotní test varianty E proběhl bez jakýchkoli komplikací. Hadice se nafukovala ve všech místech rovnoměrně a již při samotném pocitovém ozkoušení, byl velice znatelný rozdíl v tuhosti mezi vyfouknutým a plně nahuštěným stavem. Neprojevyly se ani žádné problematické deformace a rovněž vyboulení hadice na boku pěny bylo pouze zhruba poloviční, oproti variantě D. Přesné hodnoty tuhosti bylo následně nutné ověřit provedením měření.



**Graf 6.2.5 – 1: Kontrolní měření varianty E – [tuhost 0 %]**



**Graf 6.2.5 - 2: Kontrolní měření varianty E - [tuhost 50 %]**



**Graf 6.2.5 - 3: Kontrolní měření varianty E - [tuhost 100 %]**

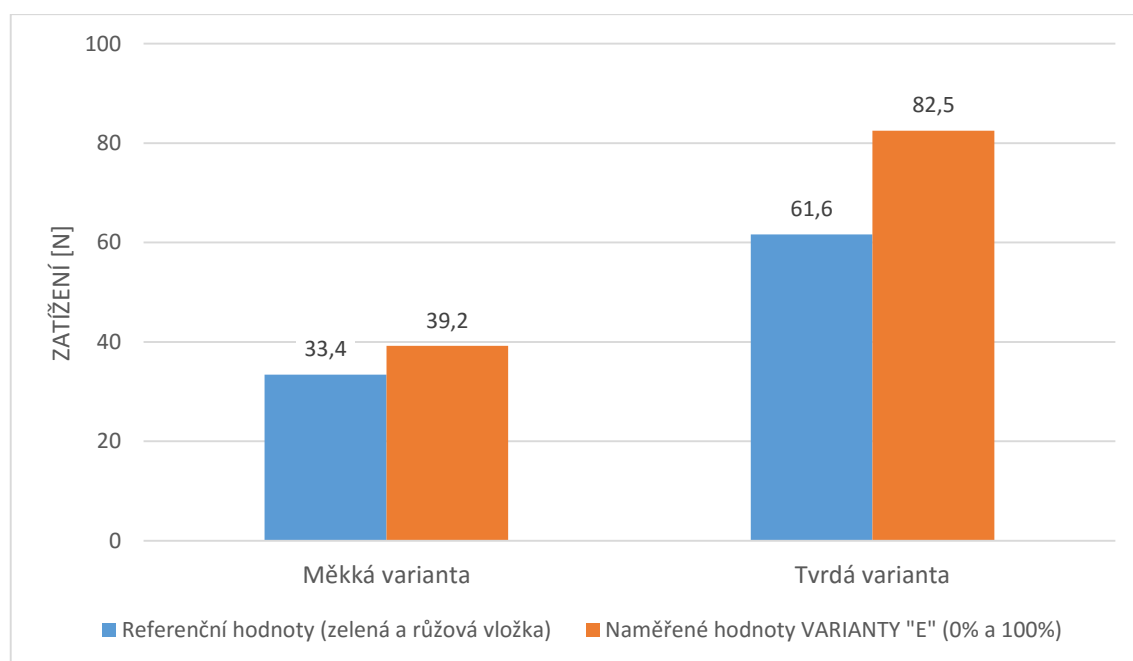
Z kontrolního měření je vidět znatelný pokrok oproti předchozí variantě D. V počáteční fázi se podařilo z velké části eliminovat skok v hodnotě zatížení. Hodnota nejnižší tuhosti při nastavení na 0 % klesla na hranici 39,2 N, naopak horní hranice při nastavení tuhosti 100 % byla posunuta až na 82,5 N. Rozdíl extrému je tedy 43,3 N, což je více než polovina. Samotné průběhy tuhosti se velice podobají naměřeným referenčním hodnotám a až na mírnou progresivitu jsou velice plynulé.

Variantu E hodnotím jako úspěšnou, neboť se podařilo dosáhnout všech stanovených cílů, jak v hranici maximální tuhosti, tak v minimální, která klesla pod očekávaných 40 N. Varianta E je tak vhodným řešením pro regulaci tuhosti matrace.

Vyhodnocení varianty „E“: **Vyhovující**

### 6.3 Výsledné srovnání dosažených výsledků

Pro jednoznačné grafické porovnání stanovených cílů s výsledky dosaženými u zvolené varianty „E“, byl vyhotoven následující graf. Porovnávána je hodnota tuhosti referenční měkké verze s použitím zelené vložky oproti nejměkčímu nastavení varianty E – hodnota tuhosti 0 %. A dále hodnota tuhosti nejtvrďší referenční verze s růžovou vložkou oproti maximální tuhosti 100 %.



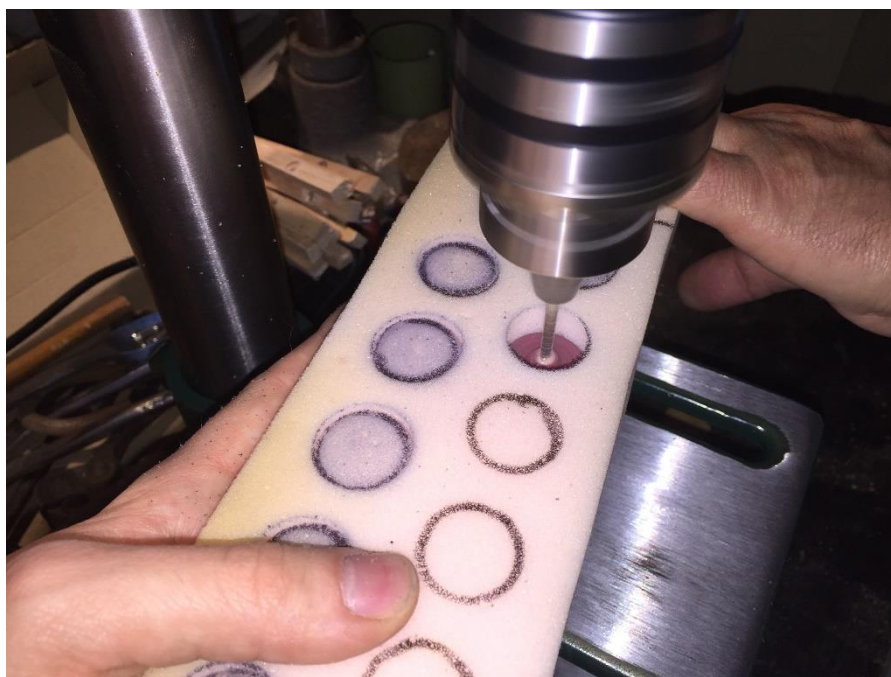
**Graf 6.3 – 1:** Porovnání referenčních hodnot a zvolené VARIANTY "E"

## 7 PRAKTICKÁ REALIZACE

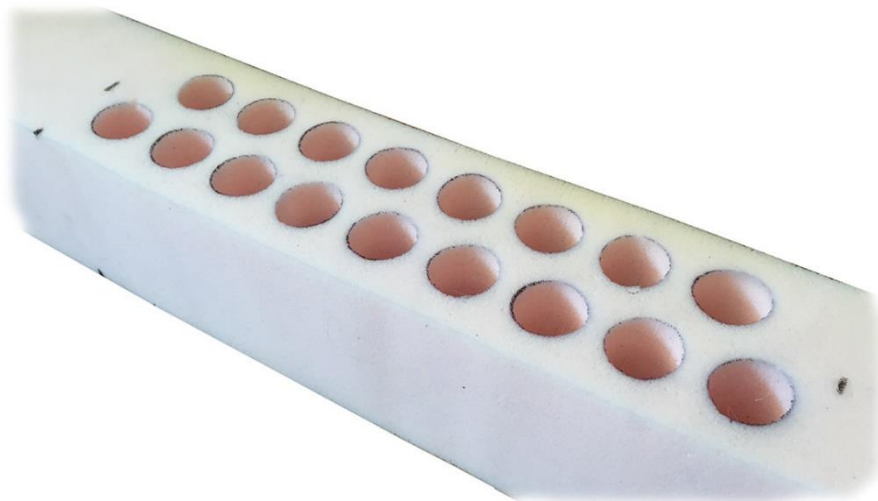
Po zvolení vhodné varianty k regulaci tuhosti zbývalo sestavit matraci do plně funkčního stavu. Regulovaným nosníkem jsem zvolil pouze střední část, která je namáhána a prohýbána nejvíce, krajní nosníky byly ponechány s neměnnou tuhostí. Závěrečný ostrý test byl plánován s ležící osobou, aby bylo možné jednoznačně demonstrovat vliv proměnlivé tuhosti.

### 7.1 Nosník s proměnlivou tuhostí

Jelikož nebyl k dispozici jednotlý dvoumetrový pěnový blok, byl jsem nucen ho sestavit (slepit) ze dvou částí vyřezaných ze standardně dodávaných nosníků. Otvory byly vybrušeny v oblastech ve stejném rozestupu a šířce jako bylo umístění otvorů pro pěnové vložky. Vybrušení probíhalo stejným způsobem jako u zkoušených variant. Nyní jsem ale využil stojanovou vrtačku s rovným vedením pro dosažení přesného vybrušení.



**Obrázek 7.1 – 1:** Přesné vybrušování na stojanové vrtačce



**Obrázek 7.1 – 2:** Díry po vybroušení



**Obrázek 7.1 – 3:** Finální nosník osazený hadicemi v regulovaných zónách

Na obrázku 7.1 – 3 je možné si všimnout, že pro druhou zónu jsem byl nucen použít jiný typ hadice, než byl původně testován. Důvodem byla nedostupnost hadice u dodavatele, kdy očekávaná doba dodání byla odhadována na 6 týdnů. Jako alternativa byla použita silikónová hadice o stejných rozměrech. Rozdíl byl pouze v barvě (čirá) a v o něco menší poddajnosti. Nejednalo se ovšem o jinak razantní rozdíl, proto bylo předpokládáno, že výsledný test by neměl být tímto řešením nijak výrazně ovlivněn.

Každá zóna byla na konci opatřena rychlospojkou pro snadné připojení hadičky. Nosník byl následně umístěn do středního vaku lamelové konstrukce a matrace tak bylo zkompletována.

## 7.2 Zhotovení boxu pro systém regulace

Jak již bylo zmíněno v předchozí části diplomové práce, nároky byly kladeny na kompaktnost, jednoznačnost a jednoduchost ovládacího zařízení. Z tohoto důvodu jsem se snažil volit pokud možno co nejmenší rozměry plastové krabice, do které se veškeré prvky regulace tuhosti podaří vměstnat. Po uvážení odhadu byl zvolen box o rozměrech 150(D) x 200(Š) x 75(V) mm. Jelikož se jednalo o univerzální box, bylo nutné vyfrézovat otvory pro ovládací prvky, ventily a obdelníkové okno pro displej.

Zařízení bylo zamýšleno jako plně funkční samostatná jednotka, proto bylo opatřeno konektorem pro přívod 12 V stejnosměrného napětí, tlačítkem pro vypnutí a zapnutí, tlačítkem reset pro případ zamrznutí a rychlospojkami pro snadné připojení hadic, všechny tyto prvky jsou umístěny na zadní straně. Přední strana obsahuje pouze prvky pro uživatele relevantní k ovládání matrace, tedy displej a potenciometry pro jednotlivé zóny.

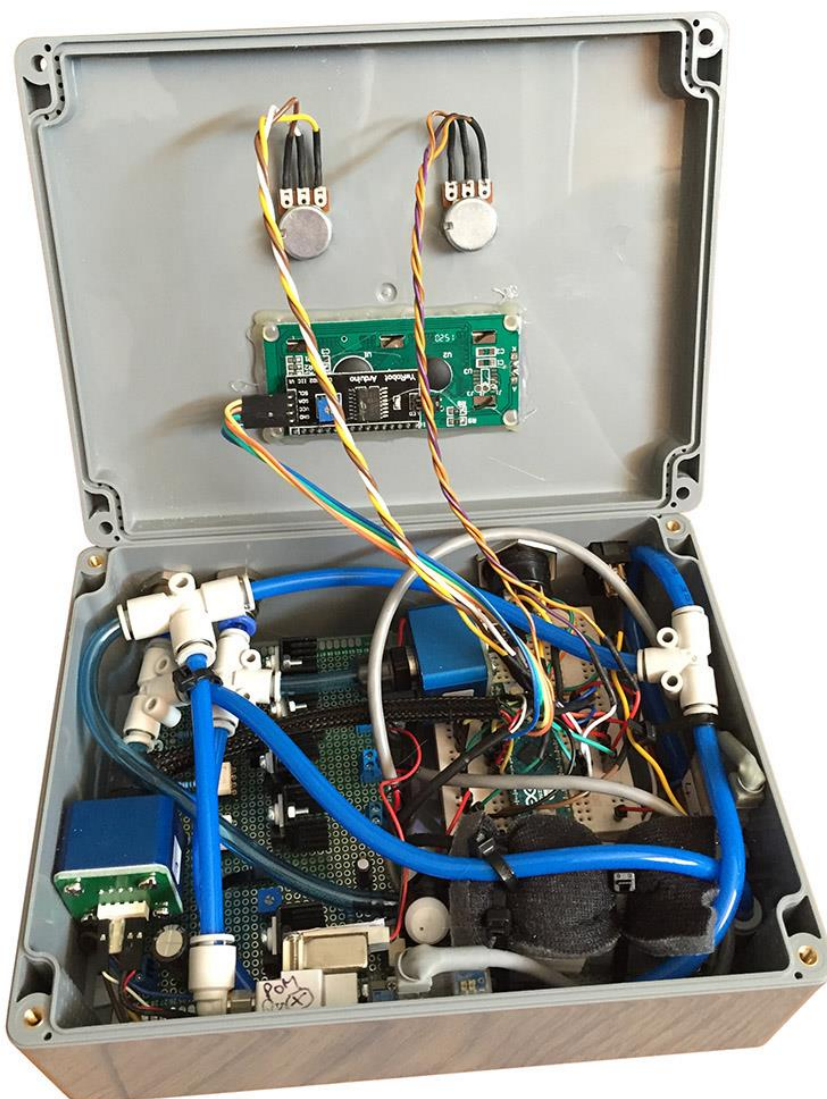


**Obrázek 7.2 – 1:** Regulační systém v boxu – přední (ovládací) panel





Obrázek 7.2 – 2: Regulační systém v boxu – zadní panel

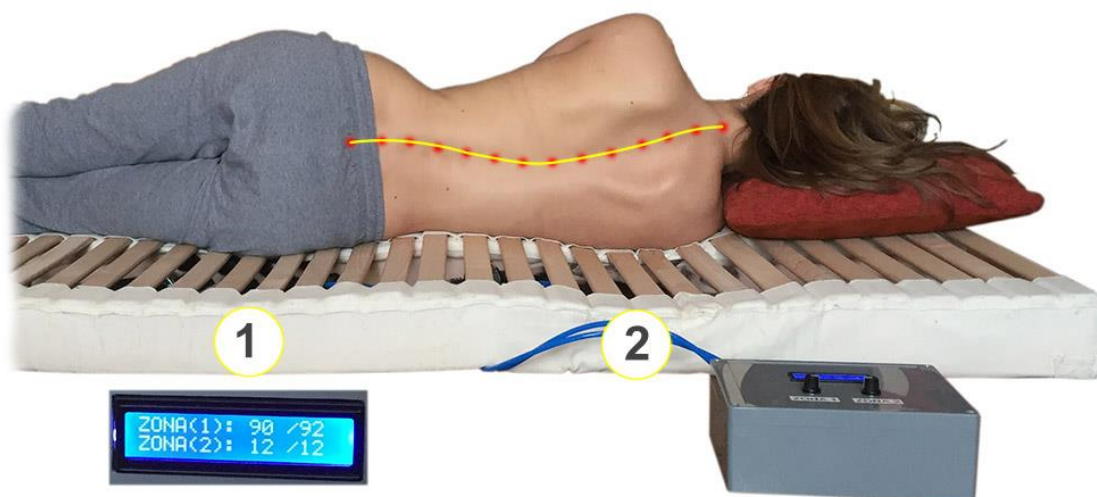


Obrázek 7.2 – 3: Regulační systém v boxu – pohled do vnitřní části

### 7.3 Závěrečný praktický test

Smyslem závěrečného praktického testu bylo ověřit funkčnost jednotlivých částí a jejich vzájemnou spolupráci při propojení. Za tímto záměrem byly nasimulovány dvě rozdílné situace, které měly demonstrovat nesprávné a správné nastavení matrace.

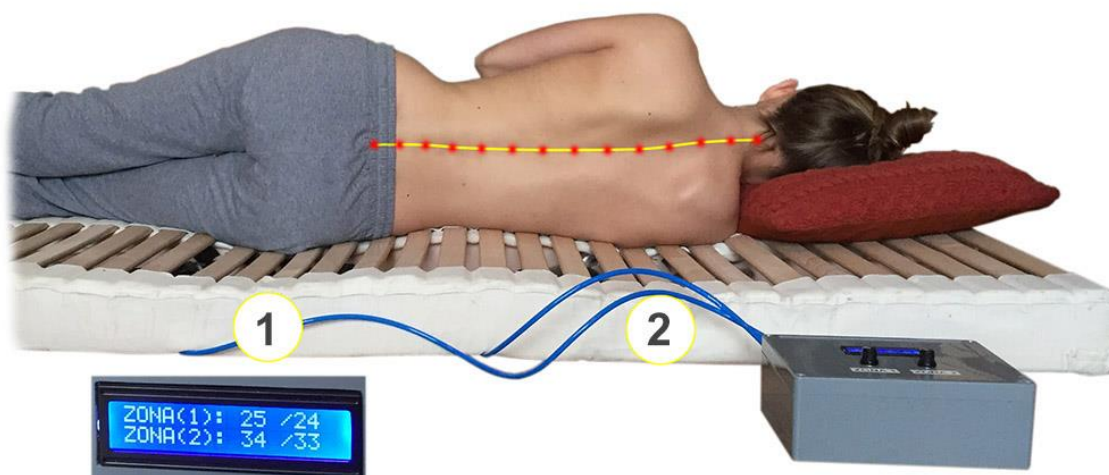
V prvním (nesprávném) případě došlo k nastavení tuhosti pro kyčelní oblast (ZONA 1) na velmi tvrdou (hodnota 90), naopak v oblasti ramen (ZONA 2) byla nastavena tuhost velice měkká (hodnota 12). Vliv nastavení matrace je možné pozorovat na zakřivení páteře figurantky na následujícím obrázku. Zakřivení páteře má nepřírozený esovitý průběh způsobený příliš velkým propadem v oblasti ramen a vysokou tvrdostí matrace v oblasti pánve.



**Obrázek 7.3 – 1:** Test matrace – demonstrace nesprávného nastavení

V rámci druhé varianty byla snaha využít potenciálu navrženého systému regulovatelné matrace a předvést ideální nastavení matrace pro postavu a tělesné proporce figurantky. Vzhledem k nízké tělesné hmotnosti a úzké postavě byla tuhost volena na spodní hranici. Příliš tvrdá varianta by vyvozovala nerovnoměrný tlak do klíčových oblastí kyčlí a ramen. Postupným nastavováním nejvhodnější tuhosti, byla nastavena hodnota 25 pro ZONU 1 a hodnota 34 pro ZONU 2. Na druhém obrázku je viditelné značné vyrovnaní páteře a srovnání osy kyčlí do kolmé pozice vůči zemi.





**Obrázek 7.3 – 2:** Test matrace – demonstrace správného nastavení

### 7.3.1 Vyhodnocení praktického testu

Výsledky závěrečného testu dopadly nad očekávání pozitivně. Systém jako celek (matrace + řídicí jednotka) fungovaly na první zapojení a bez jakýchkoli zjevných problémů. Podařilo se jednoznačně předvést rozdíl mezi správně a nesprávně nastavenou matrací a jako pozitivní hodnotím i fakt, že funkčnost nebyla nijak znatelně ovlivněna nutností nahradit původně testovanou černou hadici za méně poddajný typ.

## 8 ZÁVĚR

V první části diplomové práce byl popsán vliv matrace na kvalitu spánku a důvod, proč bychom tuto každodenní část našeho života neměli zanedbávat. Souhrnným přehledem jsou zmíněny typy komerčně prodávaných matrací, a jaký typ matrace byl dle požadovaných kritérií zvolen pro naši aplikaci. Následoval popis řídicí jednotky Arduino a zbylých klíčových prvků použitých pro řízení tuhosti.

Praktická část je zaměřena nejprve na návrh a zkonstruování zařízení pro měření tuhosti, včetně kontroly jeho způsobilosti dosahovat relevantních výsledků. Následoval návrh systému řízení tuhosti pomocí stlačeného vzduchu. Popsán a vysvětlen je zdrojový kód programu řídicí jednotky Arudino, poté schéma zapojení jednotlivých prvků a průběh pracovního cyklu programu.

Způsoby regulace tuhosti pěny matrace byly rozčleněny do jednotlivých variant. Každá varianta obsahuje vyhodnocení včetně komentáře, proč je, či není vhodná. Přestože většina z počátku navrhovaných variant nesplňovala požadované vlastnosti, poznatky získané při jejich realizování byly cenné pro další vývoj.

Po návrhu vhodného způsobu regulace tuhosti pěny došlo ke zkompletování celé matrace, zabudování regulačního systému do kompaktní krabičky a závěrečnému reálnému testu za pomoci ležící figurantky. Aby bylo možné ověřit funkčnost matrace, demonstrativně došlo ke správnému a nesprávnému nastavení tuhosti, kdy cílem bylo sledovat zakřivení páteře figurantky. Tyto dvě fotky jsou k nahlédnutí na předchozích stranách.

V rámci diplomové práce bylo dosaženo všech vytyčených cílů, jediná drobná komplikace, která se vyskytovala v průběhu celého řešení a je nutné se na ni před případnou ostrou aplikací zaměřit, je pozvolné klesání tlaku z důvodu drobných netěsností.

## Seznam použité literatury

### Materiálové listy:

- [1] OLEHLA, M., S. NĚMEČEK. *Automatické řízení*. 1. vyd. Liberec: Technická univerzita v Liberci, 2013. ISBN 978-807-3729-721.
- [2] KLOBOUČEK J. *Snímače v průmyslu*. Liberec: Technická univerzita v Liberci, 2012. ISBN 978-80-7372-828-1
- [3] PLÍVA Z., JINDRA. D., KOPRNICKÝ J., PETRŽÍLKA L. *Metodika zpracování bakalářských a diplomových prací*. Liberec: Technická univerzita v Liberci, 2014. ISBN 978-80-7494-049-1

### Elektronické zdroje informací:

- [1] *Typy postavy* [online]. 2014-12-17 [cit. 2016-04-01]. Dostupné z: <http://fannielucille.com/2014/12/17/what-is-your-body-type>
- [2] *Nejzdravější polohy při spaní* [online]. © 2016 [cit. 2016-04-01]. Dostupné z: <http://www.jysk.cz/loznice/jaka-je-nejzdravejsi-poloha-pri-spani-na-za-dech-na-brise-nebo-na-boku>
- [3] *Je špatné spát na břiše?* [online]. © 2015 [cit. 2016-04-02]. Dostupné z: <http://www.healthline.com/health/is-it-bad-to-sleep-on-your-stomach>
- [4] *Systém měření tlaku těla* [online]. 2016-02-16 [cit. 2016-04-02]. Dostupné z: <https://www.tekscan.com/products-solutions/systems/body-pressure-measurement-system-bpms-research>
- [5] *Jaký typ matrace je nejlepší?* [online]. 2014-12-17 [cit. 2016-04-02]. Dostupné z: <http://www.bydleni.cz/zprava/Ktera-matrace-je-lepsi-Penova-pruzinova-nebo-latexova>
- [6] *Ortopedická matrace SÁRA* [obrázek] [online]. 2014-03-17 [cit. 2016-04-03]. Dostupné z: <http://www.jelinek.eu/matrace-vhodne-pro-alergiky-a-astmatiky>
- [7] *Co je Arduino?* [online]. © 2016 [cit. 2016-04-03]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>

- [8] *Arduino* [obrázek] [online]. © 2016 [cit. 2016-04-04]. Dostupné z:  
<https://en.wikipedia.org/wiki/Arduino>
- [9] *Produkty Arduino* [online]. © 2016 [cit. 2016-04-04]. Dostupné z:  
<https://www.arduino.cc/en/Main/Products>
- [10] *Průvodce nejpoužívanějšími deskami Arduino* [online]. [cit. 2016-04-05].  
Dostupné z:  
<http://www.tested.com/tech/robots/456466-know-your-arduino-guide-most-common-boards>
- [11] *Arduino – programování v čistém C++* [online]. [cit. 2016-04-05].  
Dostupné z:  
[http://www.linuxsoft.cz/article.php?id\\_article=1877](http://www.linuxsoft.cz/article.php?id_article=1877)
- [12] *Introduction to Arduino: A piece of cake!* [online]. 09-30-2011  
[cit. 2016-04-05]. Dostupné z:  
<http://www.introtoarduino.com/downloads/IntroArduinoBook.pdf>
- [13] *Arduino - Manuals and Curriculum* [online]. © 2016 [cit. 2016-04-07].  
Dostupné z:  
<http://playground.arduino.cc/Main/ManualsAndCurriculum>
- [14] *Regulovatelný zdroj s LM317* [online]. © 2013 [cit. 2016-04-08]. Dostupné z:  
<http://www.hobbyelektro.eu/zdroje-menice-zdroj-s-lm317.php>
- [15] *OKENSEIKO – rolling pumps* [obrázek] [online]. [cit. 2016-04-10].  
Dostupné z:  
<http://www.okenseiko.com/en/product/rollingpump>
- [16] *LiquidCrystal\_I2C Class Reference* [online]. [cit. 2016-04-12]. Dostupné z:  
[http://volthauslab.com/datasheets/Arduino/Libraries/LiquidCrystal/LiquidCrystal/docs/html/class\\_liquid\\_crystal\\_i2\\_c.html](http://volthauslab.com/datasheets/Arduino/Libraries/LiquidCrystal/LiquidCrystal/docs/html/class_liquid_crystal_i2_c.html)
- [17] *LCD Displays (Blue and Yellow) with I2C/TWI Interface* [online].  
[cit. 2016-04-16]. Dostupné z:  
<http://arduino-info.wikispaces.com/LCD-Blue-I2C>

## Seznam obrázků

Počet: 60

- Obrázek 2.1 - 1: Typy postavy
- Obrázek 2.2.1 - 1: Poloha spánku na boku v klubíčku
- Obrázek 2.2.1 - 2: Poloha spánku na boku v rovné poloze
- Obrázek 2.2.1 - 3: Poloha spánku na zádech
- Obrázek 2.2.1 - 4: Poloha spánku na zádech
- Obrázek 2.2.2 - 1: Poloha spánku na zádech a na boku
- Obrázek 2.2.3 - 1: Sendvičová PUR matrace
- Obrázek 2.2.3 - 2: Vlevo matrace pružinová, vpravo soudečková
- Obrázek 2.2.3 - 3: Latexová matrace s kokosovým jádrem
- Obrázek 2.2.3 - 4: Ortopedická matrace s lamelovými rošty [6]
- Obrázek 3.1.1 – 1: První Arduino z roku 2005 osazené procesorem ATmega8 [8]
- Obrázek 3.1.2 – 1: Arduino UNO (vlevo), Genuino UNO (vpravo)
- Obrázek 3.1.2 – 2: Arduino Due
- Obrázek 3.1.2 – 3: Arduino MICRO (vlevo) a Genuino MICRO (vpravo)
- Obrázek 3.1.3 – 1: Příklad spojení Arduina UNO a USB Shieldu
- Obrázek 3.1.5 – 1: Programovací prostřední a příklad krátkého vzorového kódu
- Obrázek 3.1.5 – 2: Ukázka výpisu na obrazovku při spuštění programu
- Obrázek 3.2 – 1: Schéma regulovatelného zdroje
- Obrázek 3.2 – 2: Schéma obvodu pro spínání napětí kompresoru a ventilů
- Obrázek 3.2 – 3: Sestavený obvod – přední strana
- Obrázek 3.2 – 4: Sestavený obvod – zadní strana
- Obrázek 3.3 – 1: Membránový kompresor, včetně protihlukového pouzdra
- Obrázek 3.3 – 2: Princip membránového kompresoru, šipkami je označen tok
- Obrázek 3.4 – 1: Tlakový sensor DRM0D-I2C-R4B
- Obrázek 3.5 – 1: Elektromagnetický ventil V114A-6LOU
- Obrázek 3.6 – 1: LCD displej – přední strana
- Obrázek 3.6 – 2: LCD displej (včetně I<sup>2</sup>C převodníku - černý) – zadní strana
- Obrázek 3.6 – 3: Lineární potenciometr s hmatníkem

Obrázek 3.6 – 4:	Počítačový ATX zdroj jako zdroj stejnosměrného napětí
Obrázek 4.1 – 1:	Kalibrovaný tenzometr Zemic
Obrázek 4.1 – 2:	Kalibrovaný siloměr Mecmesin
Obrázek 4.1 – 3:	Sestavená konstrukce měřicího zařízení
Obrázek 4.2 – 1:	Příprava prvků pro pohyblivou část
Obrázek 4.2 – 2:	Sestavené měřicí zařízení
Obrázek 4.3 – 1:	Tenzometr a vtlačované těleso
Obrázek 5.2 – 1:	Propojený obvod v testovací fázi
Obrázek 6.1.1 – 1:	Měřená soustava žlutý nosník + růžová vložka
Obrázek 6.1.2 – 1:	Měřená soustava žlutý nosník + zelená vložka
Obrázek 6.2.1 – 1:	Guma ve spreji PlastiDip
Obrázek 6.2.1 – 2:	Testování pogumování
Obrázek 6.2.2 – 1:	Gumový vak a vyřezaná molitanová vložka
Obrázek 6.2.2 – 2:	Nafouknutý vak uvnitř molitanové vložky
Obrázek 6.2.2 – 3:	Test vaku uvnitř nosníku
Obrázek 6.2.3 – 1:	Vybrušování pěnové vložky
Obrázek 6.2.3 – 2:	Vložka osazená hadicí
Obrázek 6.2.3 – 3:	Test vložky osazené hadicí uvnitř nosníku
Obrázek 6.2.4 – 1:	Probroušené otvory o průměru 32 mm
Obrázek 6.2.4 – 2:	Nosník osazený duší
Obrázek 6.2.4 – 3:	Nosník s duší v nafouklém stavu
Obrázek 6.2.4 – 1:	Probroušené otvory o průměru 23 mm
Obrázek 6.2.4 – 2:	Nosník osazený hadicí
Obrázek 6.2.4 – 3:	Nosník s hadicí v nafouklém stavu
Obrázek 7.1 – 1:	Přesné vybrušování na stojanové vrtačce
Obrázek 7.1 – 2:	Díry po vybroušení
Obrázek 7.1 – 3:	Finální nosník osazený hadicemi v regulovaných zónách
Obrázek 7.2 – 1:	Regulační systém v boxu – přední (ovládací) panel
Obrázek 7.2 – 2:	Regulační systém v boxu – zadní panel
Obrázek 7.2 – 3:	Regulační systém v boxu – pohled do vnitřní části
Obrázek 7.3 – 1:	Test matrace – demonstrace nesprávného nastavení
Obrázek 7.3 – 2:	Test matrace – demonstrace správného nastavení

## Seznam grafů

Počet: 14

Graf 4.3 – 1:	Verifikační měření č. 1
Graf 4.3 – 2:	Verifikační měření č. 2
Graf 4.3 – 2:	Verifikační měření č. 3
Graf 5.1 - 1:	Schéma pneumatické soustavy
Graf 5.3 - 1:	Zjednodušený vývojový diagram znázorňující pracovní cyklus
Graf 6.1.1 – 1:	Naměřená tuhost soustavy žlutý nosník + růžová vložka
Graf 6.1.2 – 1:	Naměřená tuhost soustavy žlutý nosník + zelená vložka
Graf 6.2.4 – 1:	Kontrolní měření varianty D – [tuhost 0 %]
Graf 6.2.4 – 2:	Kontrolní měření varianty D – [tuhost 50 %]
Graf 6.2.4 – 3:	Kontrolní měření varianty D – [tuhost 100 %]
Graf 6.2.5 – 1:	Kontrolní měření varianty E – [tuhost 0 %]
Graf 6.2.5 – 2:	Kontrolní měření varianty E – [tuhost 50 %]
Graf 6.2.5 – 3:	Kontrolní měření varianty E – [tuhost 100 %]
Graf 6.3 – 1:	Porovnání referenčních hodnot a zvolené VARIANTY "E"

## Seznam tabulek

Počet: 4

Tabulka 2.2.4 – 1: Matice pro volbu vhodné matrace

Tabulka 3.1.2 – 1: Technické parametry Arduina UNO

Tabulka 3.1.2 – 2: Technické parametry Arduina Due

Tabulka 3.1.2 – 3: Technické parametry Arduina MICRO



## Seznam příloh

Počet: 1

Příloha 1: Schéma elektrického obvodu pro systém řízení tuhosti

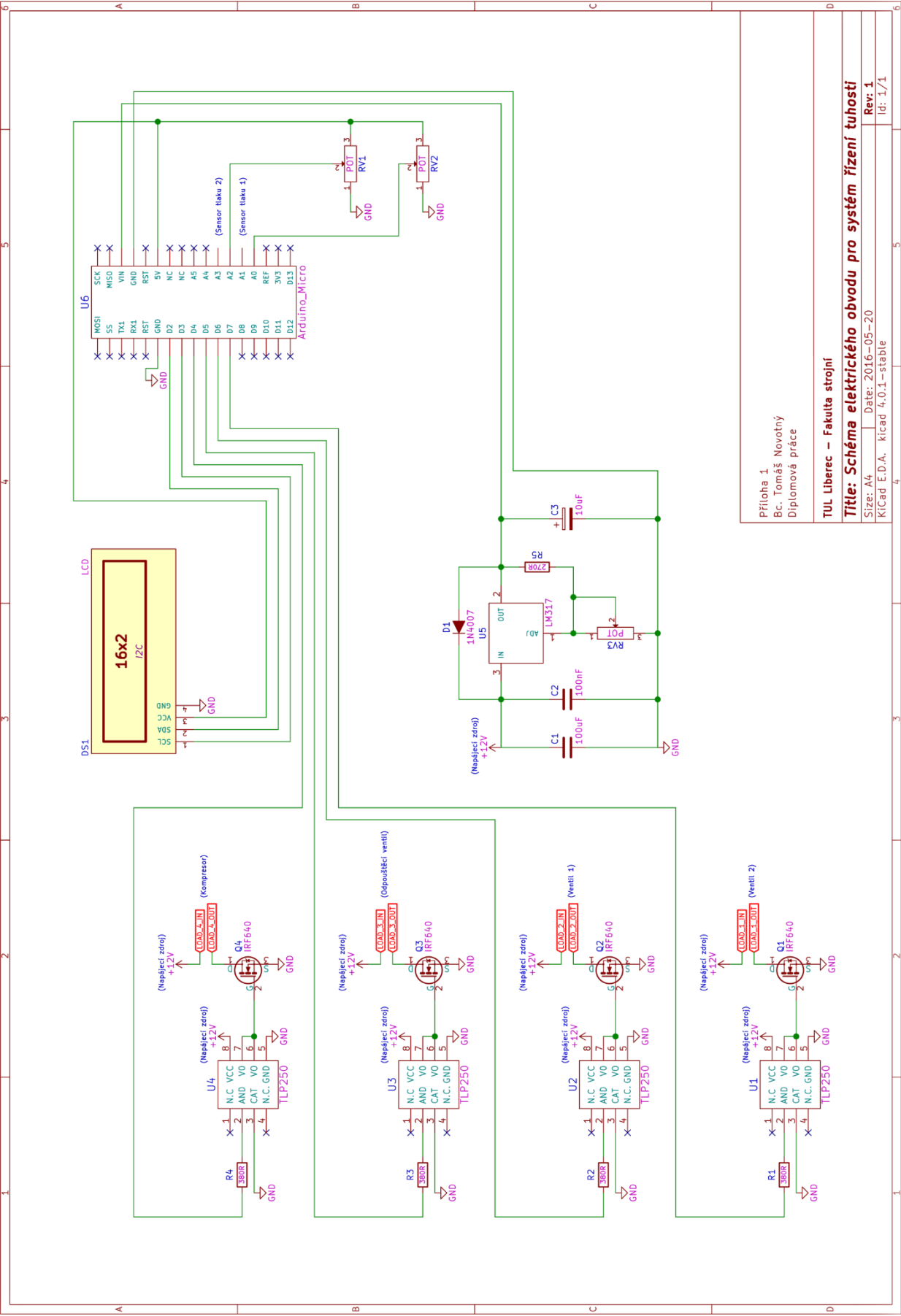
## Obsah přiloženého CD

Diplomová práce ve formátu PDF

Zdrojový kód programu ve formátu INO umístěný ve složce

Zdrojový kód programu ve formátu TXT

Příloha 1: Schéma elektrického obvodu pro systém řízení tuhosti



Příloha 1  
Bc. Tomáš Novotný  
Diplomová práce

TUL Liberec – Fakulta strojní

Title: Schéma elektrického obvodu pro systém řízení tuhosti

Size: A4  
Date: 2016-05-20  
Kicad E.D.A. kicad 4.0.1-stable

Rev: 1

Id: 1/1