



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA BIOMEDICÍNSKÉHO INŽENÝRSTVÍ
Katedra biomedicínské informatiky

**Ovládací software pro nezávislé ovládání
pěti pneumatických svalů v robotické
struktuře**

**Control software for independent control of
five pneumatic muscles in a robotic structure**

Bakalářská práce

Studijní program: Informatika a kybernetika ve zdravotnictví

Studijní obor: Biomedicínská informatika

Autor bakalářské práce: Marek Darsa

Vedoucí bakalářské práce: Ing. Martin Bejtík

Kladno 2024

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Darsa** Jméno: **Marek** Osobní číslo: **510218**
Fakulta: **Fakulta biomedicínského inženýrství**
Garantující katedra: **Katedra informačních a komunikačních technologií v lékařství**
Studijní program: **Informatika a kybernetika ve zdravotnictví**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Ovládací software pro nezávislé ovládání pěti pneumatických svalů v robotické struktuře

Název bakalářské práce anglicky:

Control software for independent control of five pneumatic muscles in a robotic structure

Pokyny pro vypracování:

Hlavním cílem této práce bude vytvořit sofistikovaný a uživatelsky přívětivý software, který bude sloužit k ovládání a kalibraci pneumatických svalů, přičemž důraz bude kladen na přesnost, bezpečnost a adaptabilitu systému. Software bude napsán v programovacím jazyce Python a bude zahrnovat různé moduly pro správu uživatelských rolí (admin/uživatel), nastavení nulových hodnot pro každodenní spuštění systému a integraci bezpečnostních omezení pro zachování provozního rozsahu pneumatických svalů. Vytvořené softwarové dílo bude zejména umožňovat: • Přihlášení uživatele nebo admina • Ovládání jednotlivých pneumatických svalů • Podle typu svalu nastavit bezpečnostní limity • Transformaci požadované translace a rotace robotické struktury na odpovídající tlaky v jednotlivých pneumatických svalech Kromě programování a testování softwaru bude důležitou součástí práce také vytvoření podrobné dokumentace, která poslouží budoucím uživatelům pro snadnější orientaci v ovládání systému a jeho možnostech.

Seznam doporučené literatury:

- [1] Soft robotics, Transferring theory to application, 2015, ISBN 978-3-662-44505-1
- [2] Koshima, Hideko, Mechanically responsive materials for soft robotics, Weinheim, Germany: Wiley-VCH, 2020, ISBN 978-3-527-34620-2
- [3] Sciavicco, L. a Bruno Siciliano, Modelling and control of robot manipulators, Springer, 2000, ISBN 1-85233-221-2

Jméno a příjmení vedoucí(ho) bakalářské práce:

Ing. Martin Bejtík

Jméno a příjmení konzultanta(ky) bakalářské práce:

Ing. Jan Kauler, Ph.D.

Datum zadání bakalářské práce: **14.02.2024**

Platnost zadání bakalářské práce: **20.09.2025**

doc. Ing. Karel Hána Ph.D.
vedoucí katedry

prof. MUDr. Jozef Rosina, Ph.D., MBA
děkan

PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci s názvem „Ovládací software pro nezávislé ovládání pěti pneumatických svalů v robotické struktuře“ vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k bakalářské práci. Prohlašuji, že tato práce je přesnou kopií odevzdané elektronické verze.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

V Kladně 15.5.2024

.....

Marek Darsa

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu své práce Ing. Martinu Bejticovi, který mi byl velice nápomocen při jakémkoliv dotazu a problému.

ABSTRAKT

Ovládací software pro nezávislé ovládání pěti pneumatických svalů v robotické struktuře

Tato bakalářská práce se zabývá vývojem ovládacího softwaru pro pneumatickou soustavu s pěti svaly. Cílem práce bylo vytvořit sofistikovaný a uživatelsky přívětivý software umožňující ovládání a kalibraci svalů s důrazem na přesnost, bezpečnost a adaptabilitu systému. Software byl vyvinut v programovacím jazyce Python. Disponuje moduly pro správu uživatelských rolí, nastavení hodnot na technickou nulu a integraci bezpečnostních omezení, jako je validace vstupů a omezení maximálního tlaku v pneumatických svalech. Funkce softwaru zahrnují přihlašování uživatelů, individuální ovládání svalů, nastavení bezpečnostních limitů a transformaci požadovaných pohybů na odpovídající výstupy v různých jednotkách. Mezi ty patří, jednotky krokových motorů – kroky, tlakové jednotky – mbar a také elektrické výstupy – mV. Vedle implementace softwaru je klíčovou součástí práce také tvorba podrobné dokumentace, která usnadní uživatelům orientaci v systému a jeho využitelnosti. Softwarové řešení představuje kombinaci přívětivého uživatelského rozhraní a robustních funkcí, umožňující efektivní ovládání pneumatické soustavy v robotických aplikacích.

Klíčová slova

Pneumatické svaly

Ovládací software

Robotická struktura

Uživatelská dokumentace

ABSTRACT

Control software for independent control of five pneumatic muscles in a robotic structure.

This bachelor thesis deals with the development of control software for a pneumatic system with five muscles. The aim of the thesis was to develop sophisticated and user-friendly software that allows control and calibration of the muscles with emphasis on accuracy, safety, and adaptability of the system. The software was developed in the Python programming language and has modules for managing user roles, setting values to technical zero and integrating safety constraints such as input validation and maximum pressure limitations in the pneumatic muscles. Software features include user login, individual muscle control, setting safety limits, and transforming desired movements into corresponding pressures in different units. These units include stepper motor step units, mbar pressure units and mV. The key part of the work, in addition to the software implementation, is the creation of detailed documentation to help users navigate the system and its capabilities. The software solution is a combination of a user-friendly interface and robust features, enabling efficient control of the pneumatic system in robotic applications.

Keywords

Pneumatic muscles

Control software

Robotic structure

User documentation

Obsah

Seznam a zkratk	5
1 Úvod	6
2 Přehled současného stavu	7
2.1 Motivace pro vznik projektu	7
2.2 Současná pneumatická soustava a její fungování	8
2.2.1 Operační systém pro řízení soustavy	8
2.2.2 Komponenty pneumatické soustavy	9
2.3 Pneumatické svaly a jejich typy	10
2.4 Python.....	14
3 Cíle práce	17
4 Návrh aplikace	19
4.1 Funkční specifikace	19
4.1.1 Knihovna pro GUI.....	19
4.1.2 Databázová knihovna	24
4.1.3 Hashovací knihovna	27
4.1.4 Knihovna time	28
4.1.5 Knihovna openpyxl	28
4.1.6 Knihovna pathlib	29
4.1.7 Knihovna re	29
4.1.8 Knihovna PneumoCVUTFBML.DeviceLoader	29
5 Implementace	30
5.1 Pneumatická soustava	30
5.2 DeviceLoader	31
5.3 Main	32
5.4 AdminWindow	35
5.5 UserWindow.....	42
6 Uživatelská dokumentace	48
6.1 Dokumentace pro uživatele	48
6.2 Dokumentace pro administrátora	52
7 Testování	57

8	Diskuse	63
9	Závěr	65
	Seznam použité literatury	66
	Seznam použitých obrázků	68
	Seznam použitých tabulek.....	69
	Příloha A: Zdrojový kód	70
	Příloha B: Seznam přiložených souborů	71

Seznam a zkratek

Seznam zkratek

Zkratka	Význam
SW	Software (Program)
GUI	Graphical User Interface (Grafické uživatelské rozhraní)
DB	Database (Databáze)
SQL	Structured Query Language (Strukturovaný dotazovací jazyk)
CMD	Command Prompt (Příkazový řádek)
MVCC	Multi-Version Concurrency Control (Kontrola konkurence více verzí)
CLI	Command Line Interface (Rozhraní příkazového řádku)
NoSQL	Not Only SQL (Nejen SQL)
MVCC	Multi-Version Concurrency Control (Řízení konkurence více verzí)
MySQL	My Structured Query Language (Můj Strukturovaný Dotazovací Jazyk)

1 Úvod

Tato bakalářská práce, navazuje na projekt, který má za cíl vytvořit pneumatickou soustavu, který bude schopen ovládat pět pneumatických svalů nezávisle na sobě. Zmíněný projekt se primárně věnuje mechanické stránce problému, avšak postrádá softwarovou implementaci. Předchozí softwarové řešení, se kterým tento projekt pracuje, umožňuje pouze otáčení krokového motoru na základě zadané funkce a počtu kroků. Tímto způsobem se motor posouvá v určeném směru a interaguje s ventilem, který reguluje tlak v systému. Takovéto řešení ovšem není dostatečně robustní, neboť nastavení tlaku v systému je možné pouze prostřednictvím neintuitivních jednotek krokového motoru, což může být pro uživatele obtížné a neefektivní. Navíc by bylo nutné ovládání softwaru provádět přímo z vývojového prostředí, což by znamenalo velkou možnost chyby. Nedostatek pokročilých funkcí v softwaru by omezoval efektivitu systému, neboť uživatel by musel provádět všechny činnosti ručně a opakovaně.

Zde prezentovaná práce představuje komplexní softwarové řešení pro eliminaci nedostatků současného systému. Zahrnuje vytvoření grafického uživatelského rozhraní s využitím moderních knihoven, které umožňuje ovládání až pěti pneumatických svalů současně, a to s možností volby různých ovládacích jednotek. Kromě toho je implementována funkce kalibrace soustavy a tvorba nových převodních vzorců. Důležitou součástí je také přístup do databáze obsahující veškeré potřebné informace pro správný provoz systému, včetně uvedených převodních vzorců a informací o aktuálně nastaveném výchozím převodním vzorci.

2 Přehled současného stavu

2.1 Motivace pro vznik projektu

Soustava vzniká v rámci studentského grantového fondu s cílem řešit problematiku ochrany anesteziologů při katetrizačních procedurách, během nichž jsou vystaveni radiačnímu záření. Tento projekt reaguje na potřebu minimalizovat riziko expozice anesteziologů tomuto typu záření, které se vyskytuje při provádění těchto zákroků. V současné době spočívá monitorování expozice anesteziologů radiačnímu záření na speciálních prstenech, které sledují, jaká dávka záření je absorbována během určitého časového intervalu.

Zdravotničtí pracovníci, včetně anesteziologů, jsou podrobena přísným normativům, které stanovují maximální přípustné úrovně expozice záření. I přes dodržování těchto limitů jsou vystaveni riziku škodlivých účinků radiačního záření. Hlavním cílem navrhované soustavy je eliminovat nutnost přímé přítomnosti anesteziologa během katetrizačních procedur, čímž by se minimalizovala jeho expozice radiačnímu záření.

Dalším z hlavních cílů navrhované soustavy je poskytnout možnost využití při procedurách v prostoru magnetické rezonance. Práce v těchto prostorech často čelí výzvám spojeným s omezeným prostorem, což může ztížit provádění zdravotnických procedur. Zvláště problematická je manipulace s pacientem a přístupnost k určitým částem těla, které jsou umístěny v těsné blízkosti magnetického pole.

Dále se setkáváme s výzvou zachování kvality obrazových dat získaných během magnetické rezonance. Existuje riziko, že pohyby rukou lékaře mohou způsobit artefakty na snímcích, což může vést ke snížení jejich kvality nebo dokonce k úplné znehodnocení. V této situaci by navržená soustava poskytovala možnost operátorovi přesně nastavit polohu pacientovy hlavy do požadované pozice, a to bez rizika vnesení rušivých prvků do snímku.

Využití navrhované soustavy není omezeno pouze na původní účel, pro který byla vyvíjena. Existují další perspektivy využití, které zahrnují aplikace v oblasti fyzioterapie krku a dalších odvětvích, kde je zapotřebí manipulace s polohou pacienta a generování nakloněných rovin.

2.2 Současná pneumatická soustava a její fungování

Jedná se o pneumatickou soustavu, která je schopna v jednu chvíli ovládat pět na sobě nezávislých pneumatických svalů, tyto svaly ovládá pomocí samostatných komponent, proto je možné ovládat soustavu stylem, že jeden sval se bude tlakem plnit a druhému bude tlak ubírán. Pneumatická soustava funguje na principu přeměny tlakové energie vzduchu na mechanickou energii určitého média, které vykonává pohyb. V rámci tohoto projektu se jedná o pneumatické svaly.

Hlavní výhodou pneumatické soustavy je čistota. Pokud by došlo k poruše potrubí, které vede stlačený vzduch z kompresoru do pneumatického svalu, a začal by vzduch unikat. Tento problém je ovšem jednoduše řešitelný, stačí k tomu pouze jednoduchá výměna zmíněné hadičky. Po výměně je přístroj opět schopen znovu fungovat bez jakýchkoliv problémů. Čistota spočívá v tom, že když uniká do okolí stlačený vzduch tak to nemá žádné negativní účinky na prostředí, do kterého uniká. Naproti tomu například pokud by soustava byla řešena pomocí hydraulického pohonu, jenž má ve svých trubičkách natlakovanou hydraulickou kapalinu, která v případě úniku udělá velké škody a znečištění. Po tomto úniku nejen že by se musela veškerá kapalina vyčistit z místa kam unikla, ale musela by se také vyměnit hadička. Po výměně hadičky by se celá soustava musela znovu natlakovat a odvzdušnit, aby fungovala správně. [1]

Hlavní nevýhodou pneumatické soustavy je ten že díky stroji, který vytvoří stlačený vzduch – kompresoru, je prostředí velice hlučné. Pokud se ovšem tento kompresor přemístí z přímé blízkosti operátora, tento nedostatek se odstraní.

2.2.1 Operační systém pro řízení soustavy

Pro tuto soustavu byl zvolen open source operační systém LXQt. Jedná se o rychlé a efektivní desktopové prostředí postavené na základech frameworku Qt. Jeho hlavní výhodou oproti ostatním prostředím je poskytnout uživatelům snadno ovladatelné prostředí, které zároveň spotřebovává minimální množství systémových prostředků.[2]

Pro tyto své vlastnosti je tento operační systém oblíbený nejen mezi lidmi, kteří potřebují systém s co největším výkonem, bez zbytečných ztrát na pokročilém grafickém uživatelském rozhraní. Je také oblíbený u uživatelů, kteří nedisponují zařízeními s velkým výpočetním výkonem, kterým tento operační systém umožňuje pracovat na zařízeních, které by například velice náročné Windows již nemuseli zvládnout. [2]

2.2.2 Komponenty pneumatické soustavy

2.2.2.1 Řídící počítač

Jako řídící počítač byl zvolen PAC-STM32 od výrobce Prokyber s.r.o. Tento počítač poskytuje široké možnosti komunikace a ovládání díky integrovaným komunikačním rozhraním, včetně USB, RS-485, UART, SPI a I2C. Díky svým expanzním slotům a podpoře připojení rozšiřujících modulů je možné tento systém snadno rozšiřovat a přizpůsobovat konkrétním potřebám projektu.

2.2.2.2 Krokový motor

V tomto projektu se využil krokový motor PK245-01A. Krokový motor je typem elektrických motorů, které se využívají pro přesné pohyby v robotice, strojírenství a mnoho dalších oborech.

Krokový motor disponuje možností pohybovat jak ve směru, tak v protisměru hodinových ručiček.

Krokové motory jsou často zaměňovány se servomotory, hlavním rozdílem mezi těmito dvěma součástkami je ten, že krokový motor se může točit kolem své osy podle libosti, jelikož nemá omezený pohyb v úhlech. Na rozdíl od toho servomotor má nastavenou úhlovou výseč, ve které se může pohybovat.

Dalším velkým rozdílem mezi těmito dvěma motory je ten, že servomotor obsahuje zpětnou vazbu, která mu umožňuje zjistit v jaké poloze se přesně nachází a s touto informací dále pracovat. Krokový motor tuto schopnost většinou nemá, a tudíž je pouze schopen reagovat na impulzy a ty vykonat.

2.2.2.3 Tlakový snímač

V soustavě se využívá také tlakový snímač PSE 303. Tento snímač slouží k tomu, aby administrátor mohl při vytváření nových převodních vzorců vytvořit také možnost zadávat požadovanou tlakovou změnu i v tlakových jednotkách.

V soustavě se nachází pouze jeden tento tlakový snímač, proto se nevyužívá při běžném běhu soustavy.

2.2.2.4 Pneumatický regulátor

V tomto projektu byl jako regulátor tlaku využit pneumatický regulátor IR1010-01BG.

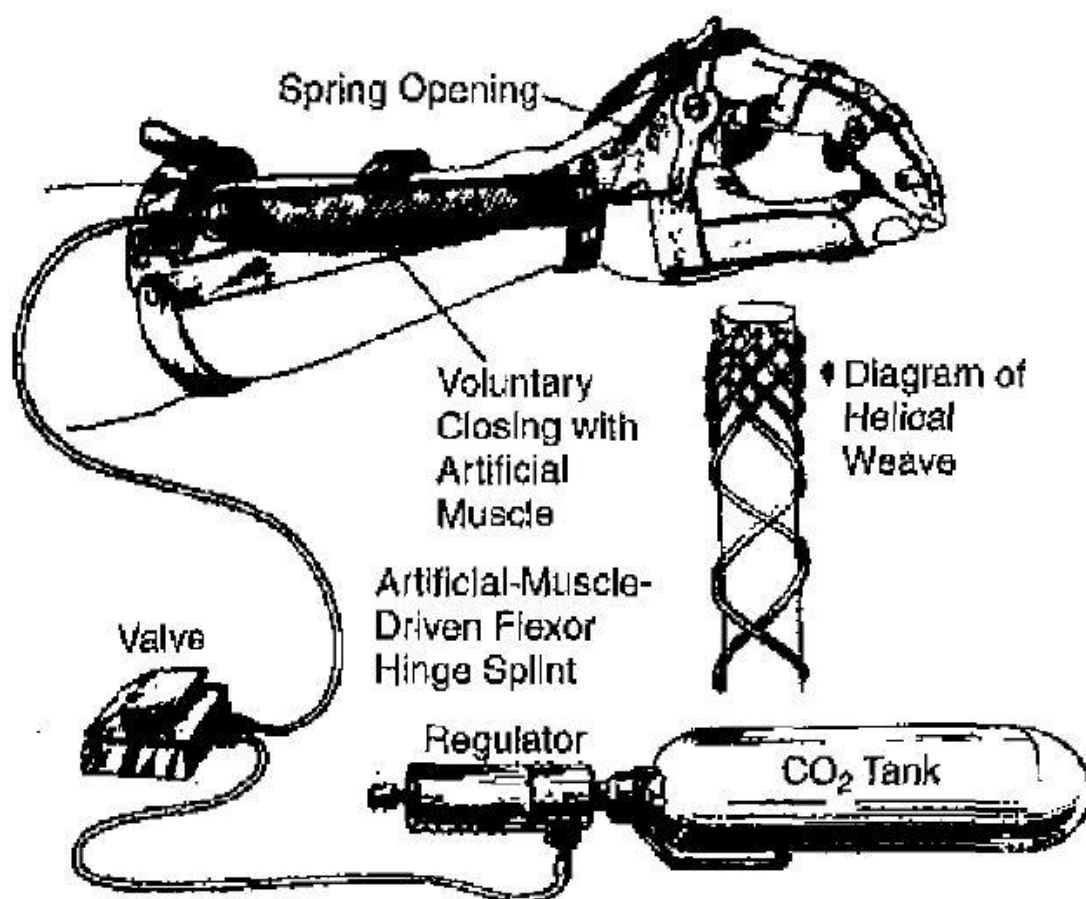
Pneumatický regulátor je zařízení používané k řízení tlaku v pneumatických systémech. Jeho hlavní funkcí je udržovat požadovaný tlak vzduchu v určeném rozsahu, aby zajistil správné fungování pneumatických zařízení a komponent.

2.3 Pneumatické svaly a jejich typy

Pneumatické svaly představují významnou inovaci v oblasti biomechanických systémů a jejich vznik sahá až do 30. let 20. století. Ruský fyzik S. Garasiev se věnoval zkoumání a konstrukci pneumatických svalů s cílem poskytnout alternativní pohon pro bio protézy. Jeho koncept vycházel z potřeby vytvořit umělý sval, který by mohl interagovat s lidským organismem podobně jako přirozené svalové tkáně.[3; 4]

Centrálním prvkem lidských svalů je jejich schopnost kontrakce a expanze. Při konstrukci pneumatického svalu pro bio protézu byl klíčovým prvkem elastický materiál, který utvářel měchýřovitou strukturu. Tato struktura byla schopna se rozpínat a stahovat v závislosti na tlaku vzduchu uvnitř měchýřku. Pro správnou expanzi a kontrakci byly implementovány paralelní pevné vlákna, která zajistila, že měchýřek se rozšíří do šířky a zkrátí svoji délku, simulující tak pohyb lidského svalu.

Další, kdo se také zabýval využitím pneumatických svalů pro řízení biologických protéz Obrázek 2.1 byl Joseph L. McKibben. Jeho přístup spočíval v konstrukci pneumatických svalů, které omezují pohyb svalu v různých směrech, čímž dosáhl požadovaného typu pohybu. McKibbenovy svaly jsou charakterizovány opletením z drátů, kde dva dráty jsou křížené, vytvářející pletení Obrázek 2.2, které transformuje pneumatickou energii na mechanickou energii.[4]



Obrázek 2.1 Návrh McKibbenovy biologické protézy pomocí pneumatické svaly

Zdroj: <https://dwindra.id/blog/assets/images/pam/pam.jpg>



Obrázek 2.2 Oplet McKibbenova pneumatického svaly Zdroj:

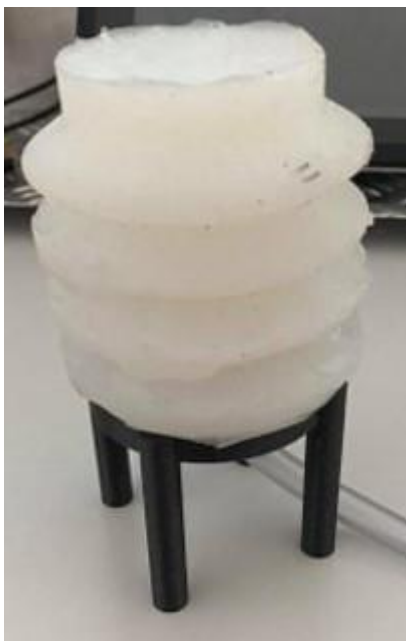
https://web.archive.org/web/20070927065220im_/http://www.shadowrobot.com/images/airmuscle-rubber-tube-braid.jpg

Hlavní výhodou těchto pneumatických svalů spočívá především v jejich jednoduchosti. Svaly mohou být vyráběny v různých délkách a velikostech a mohou být rozpínány různým způsobem v závislosti na stylu opletení. Například rozdíl mezi Yarlovým a McKibbenovým svaly spočívá v tom, že Yarlov sval se při expanzi roztahuje do sférického tvaru, zatímco McKibbenův sval spíše do tvaru válce.[3; 4]

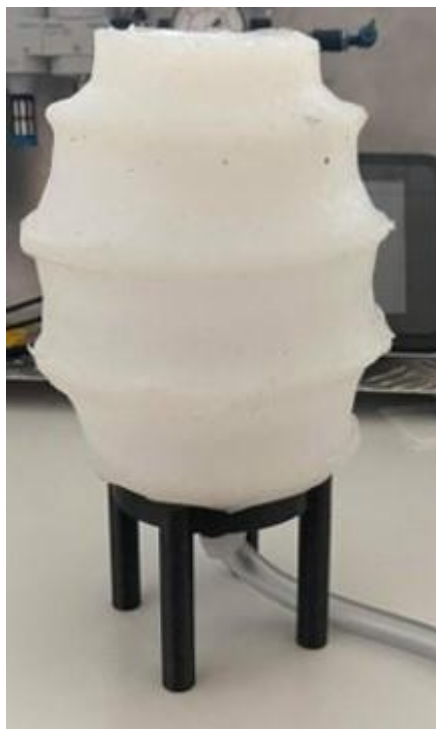
Dalším typem pneumatického svalu je Kukoljův sval, který se odlišuje v způsobu opletení pneumatického měchýřku. U Kukoljova svalu je toto opletení provedeno pomocí spojených drátů, které zůstávají pevně spojené i při nafukování a vyfukování pneumatického měchýřku. Tento princip se liší od McKibbenova svalu, který využívá pletení, kde dochází k posunu jednotlivých drátů při nafukování a vyfukování pneumatického měchýřku. [3; 4]

Dalším rozdílem mezi Yarlovými, Kukoljovi a McKibbenovými svaly a je v hustotě pletení mřížky, zatímco McKibbenový sval patří do pletených pneumatických svalů, tyto svaly se vyznačují hustě pleteným opletem. Yarlovův a Kukoljovův sval patří do skupiny takzvaných síťovaných svalů, tyto svaly mají mnohem řidčeji proveden svůj oplet.[3; 4]

V této bakalářské práci jsou využívány pneumatické svaly speciálně vytvořené pro tuto soustavu. Tyto pneumatické svaly jsou nejbližší podobné pneumatickým svalům navrženým vědeckou skupinou Greer a spol. Tento typ pneumatického svalu se vyznačuje tím, že na jedné straně je přiváděn tlak a na druhé straně je sval uzavřen. Konstrukce svalu je navržena tak, aby na rozdíl od McKibbenových pneumatických svalů, které se smršťují a rozšiřují při přivedení tlaku, tento pneumatický sval prodlužoval svůj tvar. V této soustavě slouží tento sval jako zvedací mechanismus, kde je potřeba plynule regulovat výšku. Dalším významným rozdílem mezi tímto svalovým mechanismem a McKibbenovým pneumatickým svalem je absence opletu kolem svalu, který by zajišťoval směr jeho pohyb. Směr pohybu je zajištěn tvarem svalu samotného. Sval je odléván ze silikonu pomocí formy, která byla vyrobena speciálně pro tento projekt. Tvar tohoto svalu Obrázek 2.3 je navržen tak, aby se napínal v jeho slabých místech a tímto napínáním zvětšoval svoji délku Obrázek 2.4. [4]



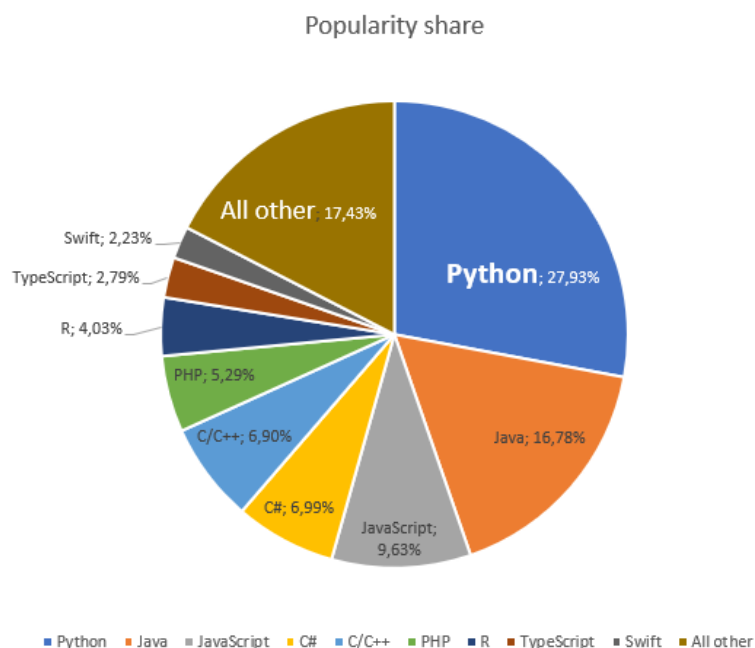
Obrázek 2.3 Vyfouknutý pneumatický sval Zdroj: Vlastní



Obrázek 2.4 Nafouknutý pneumatický sval Zdroj: Vlastní

2.4 Python

Programovací jazyk python patří k jednomu z nejpobulárnějších programovacích jazyků v dnešní době obrázek 2.3.



Obrázek 2.5 Graf popularity programovacích jazyků v roce 2023 Zdroj: <https://hitech-us.com/wp-content/uploads/2023/01/image-2.png>

Python je vysokoúrovňový programovací jazyk s dynamickým typováním, který byl vyvinut v roce 1991. Vysokoúrovňový charakter tohoto jazyka představuje vyšší úroveň abstrakce, která umožňuje programátorům psát kód na základě konceptů blízkých lidskému myšlení, spíše, než aby se zabývali detaily implementace na úrovni hardware, jako je manipulace s pamětí, která je typická pro nízkoúrovňové jazyky, například assembler.

Dalším velice populárním jazykem je programovací jazyk C#. Tento programovací jazyk je často využíván při vývoji her, například v rámci herního enginu (herní jádro) Unity, který se rozšířil jako platforma pro vytváření her pro různá zařízení, včetně počítačů, herních konzolí, mobilních zařízení a v dnešní době velice rozšířené a populární virtuální realitě.

JavaScript široce využíván, zejména v oblasti webového vývoje. Node.js představuje platformu založenou na JavaScriptu, která umožňuje provádět JavaScriptový kód mimo prostředí webového prohlížeče, což poskytuje ideální prostředí pro tvorbu serverových aplikací a síťových služeb. Knihovna React, je framework pro tvorbu uživatelských rozhraní, který se často využívá ve spojení s JavaScriptem pro vytváření interaktivních webových stránek a aplikací.

Každý z uvedených programovacích jazyků má své vlastní výhody a nevýhody, které mohou ovlivnit rozhodnutí vývojáře při výběru vhodného nástroje pro konkrétní úkol.

Například, C# je obecně považován za jazyk s vysokou úrovní abstrakce, což umožňuje vývojářům psát čistý a čitelný kód. Je také silně typovaný, což může vést k menšímu množství chyb za běhu programu. Nicméně, v některých případech může být vyžadována vyšší úroveň znalostí a zkušeností s platformou, na které se vyvíjí (například s Unity), což může zvýšit obtížnost vývoje.

Na druhou stranu JavaScript je populární pro svou univerzálnost a širokou podporu ve webovém prostředí. Jeho flexibilita umožňuje vývojářům psát kód pro frontend (uživatelské rozhraní) i backend (serverová část) aplikace, a to prostřednictvím frameworků jako je React nebo Node.js. Nicméně, dynamická povaha jazyka může vést k obtížím s udržováním a laděním větších projektů, a některé jeho vlastnosti, jako například nedostatečná kontrola typů, mohou způsobit nechtěné chyby za běhu.

Při výběru mezi technologiemi jako Blazor a React je rozhodnutí často ovlivněno konkrétními potřebami a znalostmi vývojáře. Blazor, který umožňuje psát webové aplikace s použitím C# a .NET, může být atraktivní pro vývojáře s již existujícími znalostmi a zkušenostmi v těchto technologiích. Na druhou stranu, React je často preferován pro svou jednoduchost a popularitu ve webovém vývoji, což může vést k rychlejšímu vývoji a snazšímu nalezení podpory a zdrojů.

Python získal především na své popularitě díky své komunitě, která vytvořila pro python nesčetně knihoven, díky kterým je python jedním z nejuniverzálnějších programovacích jazyků.

„Professional programmers often quickly discover that they like Python – They appreciate its expressive power, readability, conciseness and interactivity. They like world of open-source software development that’s generating an ever-growing base of reusable software for an enormous range of application areas“[5].

Pythonovské knihovny jako NumPy, Pandas a Matplotlib poskytují programátorům mnoho mocných nástrojů pro manipulaci s čísly, tabulkami a vizualizací dat. Díky nim odpadá potřeba využívat náročného softwaru jako je Matlab, který je sice ve svém oboru špičkový, avšak jeho použití v běžné praxi je omezené především kvůli své vysoké ceně. Naopak Python je kompletně zdarma.[5]

V oblasti umělé inteligence je Python rovněž silným hráčem. Knihovna PyTorch slouží pro vývoj umělé inteligence a je považována za jeden z nejmocnějších nástrojů v tomto směru. Jedním z konkrétních příkladů je použití PyTorch pro vytváření MIAgents v Unity. MIAgents jsou agenti schopní operovat v prostředí vytvořeném programátorem, provádět definované akce, a dokonce se učit na základě interakce s prostředím. Tato interakce může být pozitivně či negativně ohodnocena, což umožňuje trénovat agenty k dosahování určitých cílů.

Dále, i v oblasti webového vývoje, Python nabízí silné nástroje. Knihovna Django umožňuje vytváření webových stránek s vysokou úrovní flexibility a škálovatelnosti. S pomocí Django je možné rychle a efektivně vytvářet webové aplikace s komplexními funkcionalitami.

Herní vývoj v Pythonu rovněž není problematický. Knihovna PyGame poskytuje herním vývojářům prostředí pro vytváření her s různými úrovněmi složitosti. PyGame umožňuje tvorbu 2D her s intuitivním rozhraním a širokými možnostmi pro implementaci herní logiky a grafiky.

3 Cíle práce

Hlavním cílem této práce je vytvořit sofistikovaný a uživatelsky přívětivý software, který bude sloužit k ovládání a kalibraci pneumatických svalů, přičemž důraz bude kladen na přesnost, bezpečnost a adaptabilitu systému. Tento jeden hlavní cíl se skládá z mnoha malých podcílů.

Vyhledat knihovnu pro tvorbu grafického uživatelského rozhraní:

- Primárním kritériem je moderní estetika. Knihovna by měla nabízet moderně vypadající prvky a možnost konfigurace barevného schématu.
- Preferovaná knihovna by měla být jednoduchá a optimálně komplexní. Měla by se jednat spíše o kolekci widgetů a nástrojů pro rychlé vytvoření uživatelského rozhraní než o rozsáhlý framework optimalizovaný pro vytváření grafiky pro hry.

Vyhledat vhodnou databázi pro ukládání dat:

- Lokálnost je primárním požadavkem na databázi. Systém by měl být schopen fungovat i bez přístupu k internetu, což vyžaduje vyřazení databázových systémů pracujících na principu klient-server. Proto je vhodné zvolit databázi, která ukládá své záznamy lokálně.
- Databáze musí být schopna provádět základní operace pomocí SQL jazyka nebo jeho nadstavb. Vybraný databázový systém by neměl být příliš komplexní a měl by být schopen základního odkazování mezi tabulkami pomocí vedlejších klíčů.

Implementovat GUI aplikaci pomocí vybrané knihovny:

- Navržení jednoduchého a intuitivního rozhraní komponent pro snadnou práci uživatele.
- Implementace mechanismu automatické validace vstupů na základě definovaných pravidel.
- Přidání možnosti potvrzování vstupů pomocí klávesy "Enter" na vhodných místech uživatelského rozhraní.
- Rozšíření funkcionalit tak, aby uživatel mohl interagovat s aplikací nejen prostřednictvím tlačítek, ale také klávesovými zkratkami.

Implementovat autentizaci a oprávnění uživatelů:

- Přidat funkcionalitu, která přesune uživatele do uživatelské části po zadání uživatelského hesla a administrátory do administrátorské části po zadání administrátorského hesla.
- Implementovat do systému autentizace hashovací funkci, která zabezpečí integritu dat v databázi v případě hackerského útoku.

Kalibrace pneumatických svalů:

- Navrhnout a implementovat funkce pro kalibraci pneumatických svalů pro administrátory.
- Zabezpečit ukládání výsledků kalibrace, ve formátu .xlsx na domovskou obrazovku, pro další analýzu.

GUI pro správu databáze a převodních vzorců:

- Navrhnout uživatelské rozhraní pro správu databáze, které zobrazí hodnoty převodních vzorců pro každý sval.
- Implementovat možnost úpravy převodních vzorců a volbu aktuálního vzorce pro použití v aplikaci.

4 Návrh aplikace

4.1 Funkční specifikace

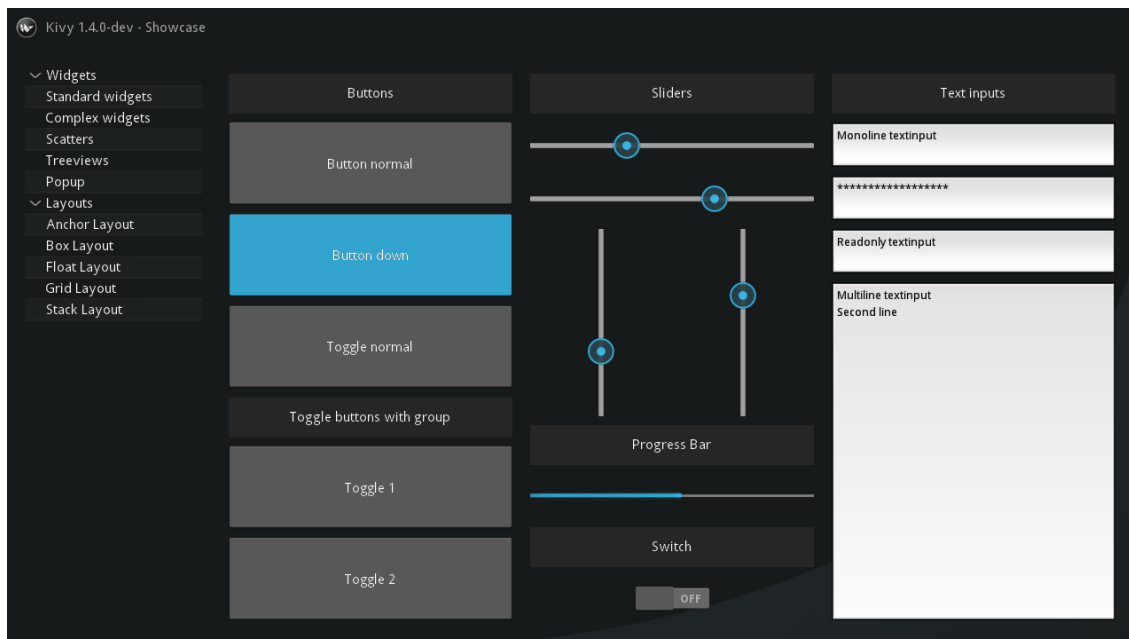
4.1.1 Knihovna pro GUI

Programovací jazyk Python obsahuje jednu integrovanou knihovnu pro grafické uživatelské rozhraní ta je rovněž i nejpoužívanější a jedná se o knihovnu tkinter[6]. Dalšími velice populárními knihovnami jsou Kivy a PyQt. Tyto knihovny nalézají uplatnění zejména u uživatelů, kteří preferují modernější vzhled jejich uživatelského rozhraní. Nicméně, Tkinter zůstává stále nejpoblárnější volbou. Tento fakt je zčásti důsledkem skutečnosti, že uživatelé, kteří usilují o vytvoření výkonné a moderní okenní aplikace, často volí jiný programovací jazyk než Python. Proto se v Pythonu často využívá Tkinter pro jednoduché formy grafických uživatelských rozhraní.

1.1.1.1 Kivy

Kivy je knihovna pro programovací jazyk python. Tato knihovna je pod licencí open-source. „Jako open source se označují programy, jejichž zdrojový kód je volně přístupný široké veřejnosti“[7]. Hlavní výhodou Kivy je její multiplatformnost, umožňující spouštění aplikací na různých operačních systémech včetně Windows, Linuxu, macOS, a také na mobilních zařízeních jako jsou Android a iOS. Kivy vyniká zejména v oblasti vývoje 2D her, kde poskytuje pokročilé možnosti tvorby simulací a grafiky. Samotná knihovna je napsaná v jazyce Python.[8]

Knihovna Kivy představuje mocný nástroj, vhodný pro mnohé účely, nicméně v případě tohoto konkrétního projektu je knihovna Kivi příliš komplexní. Ukázka GUI pomocí knihovny Kivy Obrázek 4.1.



Obrázek 4.1 GUI vytvořené pomocí knihovny Kivy Zdroj:

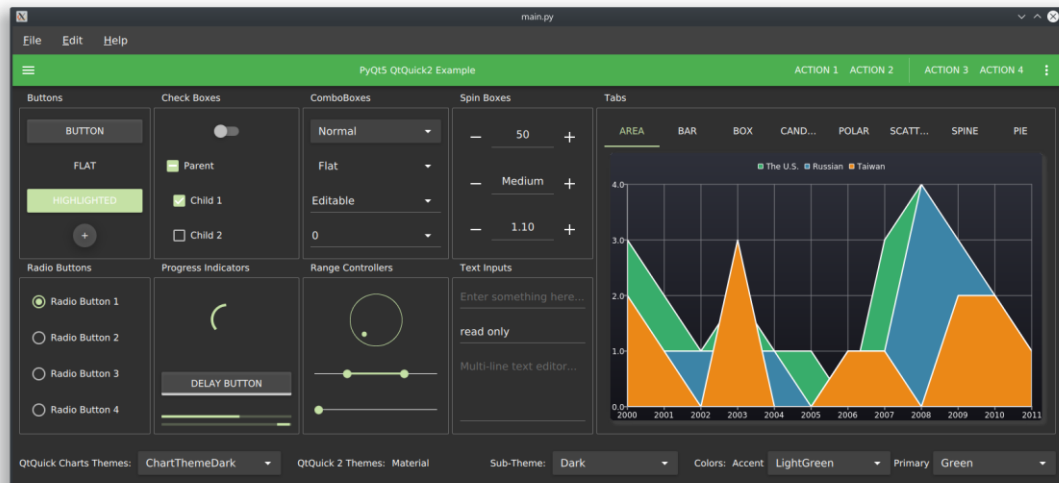
<https://i.stack.imgur.com/y6Hmq.png>

1.1.1.2 PyQt

PyQt představuje další významnou knihovnu pro tvorbu grafických uživatelských rozhraní v jazyce Python. Tato knihovna umožňuje využívat Qt aplikační framework. „Framework (Programovací rámec) jedná se o sadu hotových bloků. Programátorovy ulehčí práci především v tom, že nemusí své programy začínat vždy od nuly“ [9]. Qt framework je implementován v jazyce C++ a umožňuje psaní multiplatformních aplikací, které jsou kompatibilní s hlavními operačními systémy pro počítače i mobilní zařízení.

PyQt knihovna poskytuje výhodu v tom, že není nutné využívat celý Qt framework pro konkrétní projekt. Místo toho se může využít pouze grafické vlastnosti frameworku a případně kombinovat s jinými frameworky nebo pracovat bez nich. [10]

Knihovna PyQt je velice komplexní, a ještě sofistikovanější je, když se využívá přímo framework Qt. Ovšem v tomto projektu není takto komplexní framework potřeba proto se ani tato knihovna nevyužila. Ukázka grafického uživatelského rozhraní v pythonu pomocí knihovny PyQt Obrázek 4.2.

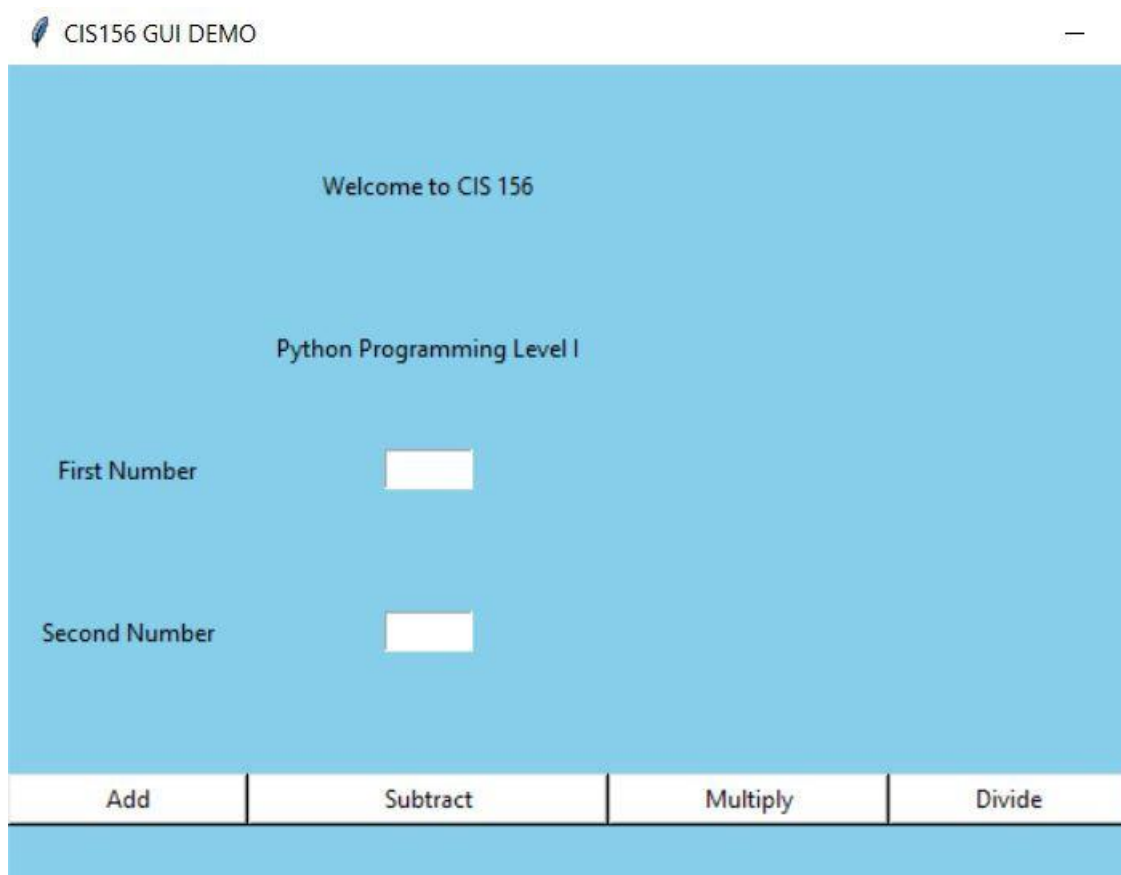


Obrázek 4.2 GUI vytvořené pomocí knihovny PyQt

Zdroj: <https://i.imgur.com/QN8YUZW.png>

1.1.1.3 Tkinter a CustomTkinter

Tkinter je knihovna pro tvorbu grafických uživatelských rozhraní, která je zabudována přímo do distribuce programovacího jazyka Python. Tím pádem je tato knihovna automaticky dostupná při stažení Pythonu z oficiálních zdrojů a není nutné ji instalovat samostatně. Pro programátora stačí pouze inicializovat Tkinter a může s ním ihned začít pracovat. Díky těmto vlastnostem je Tkinter zvláště populární mezi začátečníky a programátory, kteří upřednostňují jednoduchost před moderním vzhledem grafických uživatelských rozhraní, které nabízejí ostatní knihovny jako Kivy nebo PyQt. Ukázka GUI vytvořeného pomocí knihovny tkinter Obrázek 4.3.[11]

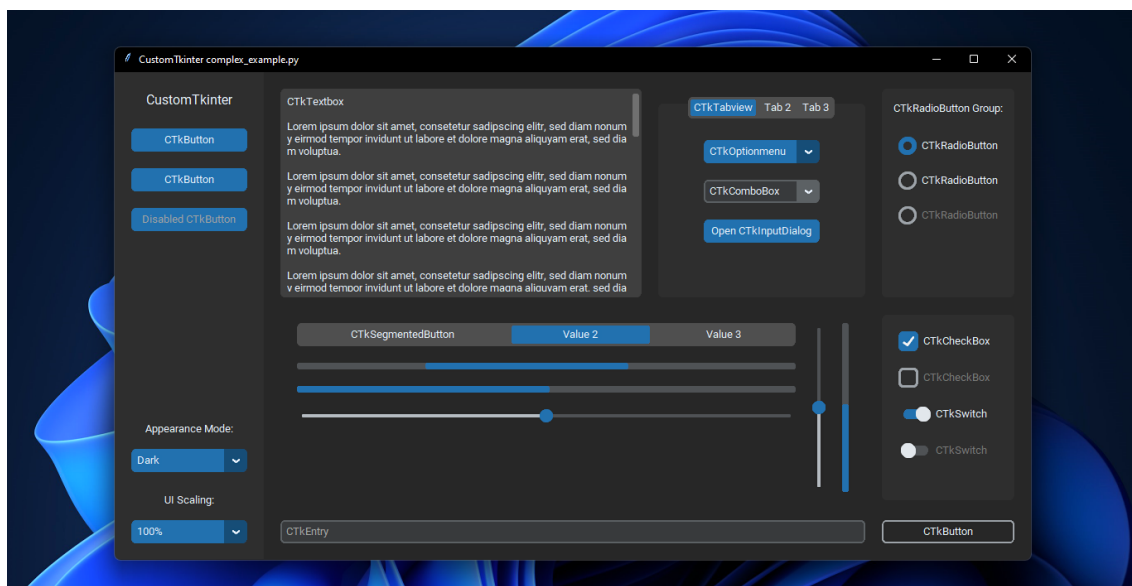


Obrázek 4.3 GUI vyhotovené pomocí knihovny Tkinter Zdroj:

https://static.wixstatic.com/media/90b6f2_3041927f0a11481a995dde96b3e2b01d~mv2.jpg/v1/fill/w_703,h_547,al_c,q_90/90b6f2_3041927f0a11481a995dde96b3e2b01d~mv2.jpg

Knihovna CustomTkinter je uživatelská nadstavba pro knihovnu tkinter. Tato knihovna byla vytvořena pod licencí MIT – jedná se o open-source knihovnu. Tuto knihovnu vytvořil programátor jménem Tom Schimansky. Knihovna vznikla v závislosti na tkinteru přebírá některé její vlastnosti a upravuje je do moderního vzhledu. Knihovny spolu mohou pracovat, tudíž je možnost pracovat s oběma knihovnami zároveň a z každé si vzít některé její vlastnosti, což se ovšem i doporučuje v oficiální dokumentaci knihovny customtkinter. Hlavními přednostmi této knihovny je například možnost vybrání si vlastního režimu ('světlý', 'tmavý'). Tento režim se dá nastavit v závislosti nastavení vašeho počítače nebo se dá i ručně měnit.[12]

Pro všechny pozitivní vlastnosti se vybrala tato knihovna jako nástroj, pomocí kterého se bude vytvářet grafické uživatelské rozhraní a jeho nedostatky budou doplněny knihovnou tkinter. Ukázka moderního grafického uživatelského rozhraní vytvořeného pomocí knihovny CustomTkinter Obrázek 4.4.



Obrázek 4.4 GUI vyhotovené pomocí knihovny customtkinter Zdroj:

https://raw.githubusercontent.com/TomSchimansky/CustomTkinter/master/documentation_images/complex_example_dark_Windows.png

4.1.2 Databázová knihovna

„Databáze je organizovaný soubor strukturovaných informací neboli dat, které se obvykle ukládají v elektronické podobě v počítačovém systému“ [13].

Existuje velké množství různých databází a každá tato databáze má svoje vlastní vhodné využití, jedním z nejkompexnějších typů databázových systémů jsou distribuované databáze.

4.1.2.1 Distribuované databáze

Distribuované databáze představují významný typ databázových systémů, který je vhodný pro velké databázové systémy. Jednou z hlavních vlastností distribuovaných databází je možnost distribuce dat mezi více serverů umístěných na různých místech. Tato schopnost umožňuje efektivní zpracování a správu velkého objemu dat prostřednictvím víceuživatelské sítě, kde každý uzel může provádět operace na své části databáze.[13]

V rámci bakalářské práce nebyl použit tento databázový systém z důvodu jeho značné komplexity a schopnosti pracovat s velkým objemem dat distribuovaných na více serverech. Požadavky na databázový systém v daném projektu se soustředily především na lokální databázi s omezeným počtem tabulek.

4.1.2.2 Databáze typu NoSQL

NoSQL databáze představují moderní přístup k ukládání a manipulaci s daty, který se odlišuje od tradičních relačních databází. Zkratka NoSQL znamená "Not only SQL" (Nejen SQL), což naznačuje, že tyto databáze nejsou omezeny pouze na strukturovaný SQL jazyk. Jedním z hlavních rysů NoSQL databází je schopnost pracovat s nestrukturovanými daty a umožňovat ukládání různých typů informací, včetně binární reprezentace multimediálního obsahu, jako jsou obrázky, videa nebo zvukové stopy.

Díky této flexibilitě jsou NoSQL databáze ideální pro aplikace, které vyžadují škálovatelnost a možnost pracovat s různými typy obsahu. Například pro sociální sítě, e-commerce (elektronický obchod) platformy nebo systémy pro správu obsahu, kde je potřeba efektivně pracovat s velkým objemem dat a zároveň udržovat vysoký výkon systému.[13]

Tento databázový systém poskytuje optimální řešení pro aplikace vyžadující ukládání rozmanitých typů dat, včetně textových, číselných a multimediálních souborů, jako jsou obrázky nebo celé soubory. Nicméně, v rámci tohoto konkrétního projektu není nutné implementovat takovou rozsáhlou funkcionalitu. Z daných požadavků vyplývá, že databáze bude využívána primárně pro uchovávání textových a číselných dat.

4.1.2.3 Relační databáze:

Relační databáze představují jeden z nejrozšířenějších typů databází, který se opírá o strukturu tabulek s řádky a sloupci. Jejich klíčovým prvkem jsou primární klíče, které jednoznačně identifikují každý záznam v tabulce. Tento typ databáze je ideální pro ukládání strukturovaných dat s definovanými vztahy mezi nimi, což umožňuje efektivní správu informací v rámci organizace.

Jednou z klíčových výhod relačních databází je možnost propojování tabulek pomocí vedlejších klíčů. To znamená, že v jiné tabulce může být uložen například úplný profil pacienta včetně adresy, a na tuto adresu se může odkazovat pomocí vedlejšího klíče v tabulce, kde jsou uloženy informace o pacientech. Tento přístup umožňuje efektivní a flexibilní správu dat, jelikož adresu může mít jak pacient, tak i lékař, nebo může být sdílena mezi různými tabulkami v rámci databáze.[13]

Další výhodou relačních databází je schopnost jedné tabulky sloužit pro více účelů. Například, místo trvalého bydliště pacienta může být definováno jako trvalé bydliště, ale také jako přechodné bydliště, a tato informace může být využita ve více kontextech, což zvyšuje flexibilitu a použitelnost dat v databázi.

Tento databázový systém byl vybrán pro účely této bakalářské práce s přihlédnutím k jeho jednoduchému uživatelskému rozhraní a dostatečnému pokrytí funkcionalit. Specifikace požadavků explicitně stanovují nutnost ukládání výhradně textových a číselných dat.

4.1.2.4 MySQL

Rozhodnutí o typu databáze pro implementaci v bakalářské práci nyní přechází k specifikaci konkrétního relačního databázového systému. Jedním z nejvyužívanějších a dobře etablovaných systémů v této oblasti je MySQL.

MySQL je open-source databázový systém postavený na jazyce SQL, což jej činí vhodným pro široké spektrum aplikací. Jeho architektura byla navržena tak, aby nebyla omezena na žádnou konkrétní platformu, což znamená, že je schopný provozu na různých operačních systémech. Primárním cílem jeho vývoje bylo vytvoření databázového systému optimalizovaného pro webové aplikace.[13]

MySQL se vyznačuje vysokou úrovní výkonu a škálovatelnosti, což mu umožňuje efektivně zpracovávat velké množství dotazů a obsluhovat tisíce paralelních procesů. Tyto vlastnosti jsou klíčové zejména pro webové obchody, kde je nezbytné zajistit rychlé a spolehlivé odezvy databázových operací při zpracování transakcí a udržování stability systému. Z tohoto důvodu je MySQL často preferovanou volbou pro implementaci databázového řešení v prostředí webových aplikací.

MySQL databáze, s ohledem na svou vysokou komplexitu a využitelnost v oblastech vyžadujících online přístup k datům, se vyznačuje prominentním postavením na poli relačních databázových systémů. Nicméně, v kontextu našeho specifického projektu, který postrádá potřebu pro tuto charakteristickou vlastnost, byla tato platforma záměrně vyloučena z uvažovaných variant.

4.1.2.5 PostgreSQL

PostgreSQL je relační databázový systém, který je vyvíjen komunitou, nikoli velkou korporací, což přispívá k jeho popularitě. Jedná se o klient-server architekturu, kde veškerá data jsou uchovávána na serveru a klienti se k němu připojují, často prostřednictvím příkazové řádky (CLI). PostgreSQL disponuje širokou škálou pokročilých funkcí, které převyšují možnosti MySQL, avšak na druhou stranu není tak rychlý jako zmíněný konkurenční systém.

Hlavní výhodou PostgreSQL je jeho využití metody MVCC (Multi-Version Concurrency Control, Řízení konkurence více verzí). Tato technika opravňuje databázi umožnit přístup k datům více uživatelům současně, aniž by docházelo k blokaci. To znamená, že čtenáři neblokují zapisovatele a zapisovatelé neblokují čtenáře. Tento efekt je dosažen tím, že databáze udržuje informace o tom, která data byla poskytnuta klientům a tyto informace udržuje až do konce transakce. Tímto způsobem se minimalizují konflikty a umožňuje se efektivní manipulace s daty.[14]

Vzhledem k vysoké komplexitě a rozmanitosti možností, které relační databázový systém PostgreSQL nabízí, není vhodnou volbou pro tento projekt. Hlavními kritérii výběru databázového systému jsou jednoduchost a lokalita, což vylučuje nasazení PostgreSQL pro tuto konkrétní aplikaci.

4.1.2.6 SQLite3

SQLite3 je relační databázový systém, který se odlišuje od tradičních server-klient modelů tím, že veškerá data jsou uložena lokálně na disku v jediném souboru, což představuje podobnost s běžnými soubory, jako jsou například textové soubory ve svém formátu .docx, které můžeme otevírat a následně s nimi pracovat například pomocí aplikace „Word“. Databázové systémy mají také svůj vlastní formát .db, s tímto formátem se dá pracovat například v aplikaci „SQLite Database Browser“. SQLite3 je schopen ukládat a manipulovat se strukturovanými daty, což jej činí mocným nástrojem pro práci s databázemi.[15]

Tento typ databáze je zahrnut v základní distribuci jazyka Python. To znamená, že programátoři, kteří používají Python, nemusí instalovat žádné dodatečné knihovny pro práci s SQLite3, stačí pouze inicializovat tuto knihovnu a mohou začít vytvářet vlastní databáze.

Díky své jednoduchosti je SQLite3 vhodný pro širokou škálu uživatelů, od začátečníků, kteří se teprve seznamují s prací s databázemi, až po zkušené programátory hledající výkonnou a jednoduchou databázi pro lokální projekty.

Je však třeba poznamenat, že kvůli své povaze lokální databáze a absenci serverového modelu, SQLite3 neposkytuje pokročilé mechanismy jako je například MVCC (Multi-Version Concurrency Control). Avšak, tato jednoduchost neznamena, že SQLite3 je omezený pouze na základní operace ukládání a načítání dat. Databázový systém je také schopen pracovat se složitějšími procesy, včetně triggerů (spouštěč), což umožňuje vytvářet dynamické a komplexní aplikace. „Některé aplikace mohou používat SQLite pro interní ukládání dat. Je také možné vytvořit prototyp pomocí SQLite a pak portovat kód do větší databáze, jako je PostgreSQL nebo Oracle“ [15].

Pro účely této bakalářské práce byl jako databázový systém zvolen SQLite3. Tato volba byla motivována především její jednoduchostí a vysokou rychlostí. Vzhledem k tomu, že soustava, na níž je tato bakalářská práce vypracovávána, nenabízí přístup k internetu, bylo nevhodné zvolit databázový systém založený na klient-server architektuře. Proto byla vlastnost SQLite3 umožňující lokální uložení dat na disku považována za vhodnou volbu.

4.1.3 Hashovací knihovna

Hashovací knihovna je nedílnou součástí této bakalářské práce, kvůli správné autentizaci uživatelů. V kontextu zabezpečení je běžnou praxí ukládat hesla do databází ve formě za hashovaného textu. Tento proces slouží k ochraně hesel uložených v databázi před hackerským útokem. Hash je řetězec o pevné délce, který vzniká z libovolně dlouhého vstupního řetězce. Důležitou vlastností této procedury je, že je jednosměrná – z hashové hodnoty není možné zpětně získat původní heslo. Dokonce i prázdný řetězec má svou jedinečnou hashovou hodnotu, "e3b0c44298fc1c149afb f4c8996fb92427ae41e4649b934ca495991b7852b855". Tento fakt výrazně komplikuje útoky na hesla, neboť útočník nemá jinou možnost, než zkoušet náhodná hesla a provádět bruteforce (hrubá síla) útoky.

Tato knihovna nabízí spoustu různých hashovacích algoritmů jako jsou „sha384, Whirlpool, sha256 atd.“. V tomto projektu se využívá hashovací funkce sha256. „Důvodem pro použití SHA-256 je, že se jedná o jeden z nejznámějších a nejbezpečnějších hashovacích algoritmů, které se v současné době používají, a zároveň nabízí méně času potřebného k výpočtu hashe“ [16].

4.1.4 Knihovna time

Tato knihovna je nástrojem primárně určeným pro manipulaci s časem v programovacím prostředí. Poskytuje rozsáhlou funkcionalitu pro práci s časem, včetně získání aktuálního času, manipulace s kompletními daty ve formátu "%Y-%m-%d %H:%M: %S", a umožňuje řízení časových prodlev pomocí funkce „time.sleep(X)“, kde programátor může nastavit hodnotu proměnné X na požadovanou dobu spánku programu v sekundách.[17]

V tomto projektu byla knihovna „time“ využita zejména pro regulaci časové odezvy na změny tlaku v pneumatických svalech. Pomocí funkce „time.sleep(X)“ bylo možné definovat dobu čekání programu mezi jednotlivými čtecími časy, což umožnilo citlivě reagovat na dynamiku změn tlaku v daném systému.

4.1.5 Knihovna openpyxl

Vzhledem k tomu, že tato knihovna není součástí základního balíčku Pythonu, je nutné ji doinstalovat, aby s ní mohl uživatel pracovat. Tato knihovna je navržena k manipulaci se soubory s příponami „.xlsx“, „.xlsm“, „.xltx“ a „.xltm“, což jsou typy souborů spojené s programem Excel od společnosti Microsoft. Nicméně tato označení jsou poněkud zavádějící, neboť „Excel“ je pouze název softwaru vyvíjeného společností Microsoft, který je schopen pracovat s těmito soubory.[18]

Je třeba si uvědomit, že mnoho uživatelů nemusí používat program Excel, zejména kvůli nemožnosti pracovat off-line bez zakoupení drahé licence. Existují však alternativy, jako například „Google Sheets“ od společnosti Google, které poskytují podobné funkce a jsou zdarma k dispozici online, nebo z celkového balíčku LibreOffice aplikace „Calc“ tento sw je dostupný zdarma i pro off-line práci.

Vzhledem k této problematice je vhodné použít obecnější termín pro označení tohoto typu softwaru, a to „Tabulkový editor“. Tento termín lépe odráží povahu programů, které umožňují vytvářet, editovat a analyzovat tabulková data bez ohledu na konkrétní platformu nebo poskytovatele softwaru.

Tato knihovna představuje významný nástroj díky svým mnohostranným funkcím, které umožňují nejen základní manipulaci s daty, jako je jejich vkládání na specifická místa a následné načítání, ale také jejich formátování. Jednou z významných a pokročilých vlastností této knihovny je schopnost vytvářet grafy z poskytnutých dat.

4.1.6 Knihovna pathlib

Tato knihovna je součástí standardní distribuce programovacího jazyka Python a zaujímá klíčové postavení při manipulaci s cestami v Python programování. Její využití je obzvláště efektivní díky široké škále předdefinovaných cest, jako je například funkce „Path.home()“, která umožňuje snadný přístup k domovské složce uživatele a ukládání souborů na tuto lokaci. Tato knihovna se stává mocným nástrojem v programování díky své schopnosti poskytnout uživatelům flexibilitu při určování umístění uložených souborů. Výhodou je, že každý uživatel má možnost volby umístění a uložení souborů dle vlastní preference, což by nebylo možné s pevně stanovenou cestou, která by neodpovídala individuálním potřebám uživatelů. [19]

4.1.7 Knihovna re

Jedná se o další knihovnu ze základní distribuce programovacího jazyka Python. Tato knihovna slouží především pro práci s regulárními výrazy. „Regulární výrazy, známé také jako regex nebo regexp, jsou nástroji pro práci s textem, které umožňují vyhledávání, nahrazování a manipulaci s textovými řetězci na základě definovaného vzoru (patternu)“ [20].

V rámci této bakalářské práce byla knihovna použita k validaci vstupních dat, kde byl specifikován regulární výraz, akceptující pouze čísla a znaménko mínus. Veškeré jiné znaky byly automaticky označeny jako neplatné a tímto způsobem byla implementována efektivní ochrana proti nevalidním vstupům.[21]

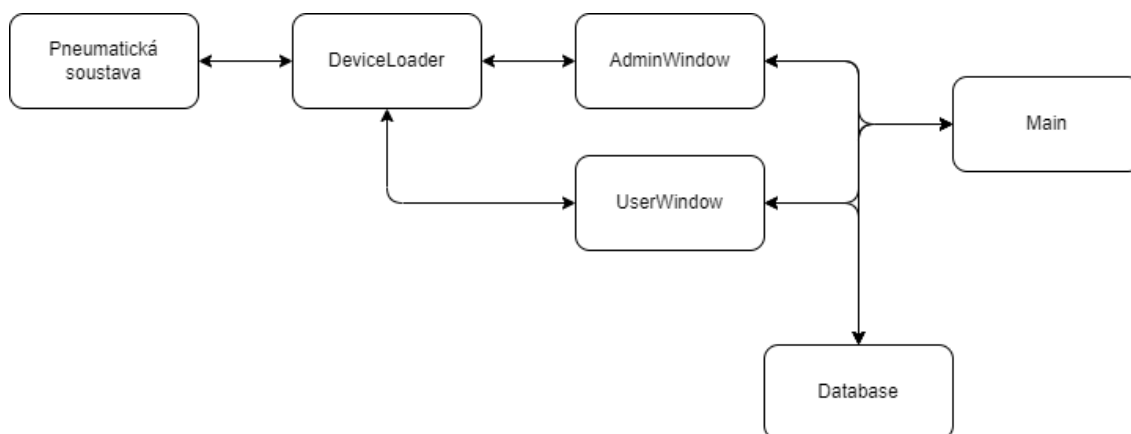
4.1.8 Knihovna PnemoCVUTFBMI.DeviceLoader

Tato knihovna byla speciálně vytvořena se soustavou, pro niž je v rámci této práce navržen grafický uživatelský rozhraní (GUI). Účelem této knihovny je usnadnit komunikaci s dalšími knihovnami, jako je sériová sběrnice či krokové motory, v rámci dané mechanické soustavy.

Tato knihovna představuje klíčový spojovací prvek mezi GUI a jednotlivými mechanickými součástmi daného systému. Její návrh a implementace jsou klíčovými kroky pro efektivní a intuitivní ovládání a monitorování soustavy pomocí uživatelsky přívětivého rozhraní.

5 Implementace

Tento projekt byl rozčleněn do jednotlivých komponent (modulů) s cílem zvýšit přehlednost a usnadnit správu výsledného softwarového produktu. Tento přístup byl zvolen zejména s ohledem na rozsah a komplexnost projektu, stejně jako na potřebu spolupráce více jednotlivců, kteří se na projektu podílejí. Je třeba zdůraznit, že tým pracoval na této soustavě již před zahájením této bakalářské práce, a bude i nadále po jejím ukončení, přičemž se plánuje neustálé udržování a vylepšování softwarového produktu. Proto je klíčové, aby byla struktura projektu správně navržena s ohledem na snadnou budoucí údržbu a rozšiřitelnost. Projekt byl rozdělen do šesti hlavních komponent, jak je znázorněno na Obrázku 5.1.



Obrázek 5.1 Komponent diagram soustavy této bakalářské práce vyhotoven v jazyce UML. Zdroj: Vlastní

5.1 Pneumatická soustava

Pneumatická soustava není předmětem této bakalářské práce. Jedná se o autonomní systém, který umožňuje nezávislé ovládání pěti krokových motorů, jež řídí ventily a tím mění tlak na jednotlivých výstupech. Tyto výstupy jsou navrženy tak, aby umožňovaly připojení pneumatických svalů, což jsou zařízení schopná konvertovat tlakovou energii vzduchu na mechanický pohyb. Tento systém poskytuje robustní řízení a flexibilitu při manipulaci s pneumatickými svaly a dalšími příslušenstvími v rámci příslušné aplikace.

5.2 DeviceLoader

Tato komponenta je navržena tak, aby mohla efektivně komunikovat se zbytkem softwarového systému a řídit jednotlivé krokové motory prostřednictvím interakce se systémem. Tato třída slouží jako rozhraní pro komunikaci se soustavou, přičemž její hlavní funkcí je ovládání krokových motorů. Zbytek softwarového systému interaguje s touto komponentou jako prostředníkem pro komunikaci se soustavou. Přestože se tato komponenta skládá z řady funkcí, pro potřeby projektu je využívána především jedna základní funkce, která umožňuje řízení krokových motorů.

getBoardX()

X - > číslo desky. Příklad `getBoard1()` nám dovolí přistupovat k jednotlivým funkcím ovládací desky. Mezi tyto funkce patří:

- `on()` – Tato funkce, která je zásadní pro aktivaci a přípravu desky pro další operace, musí být deklarována na začátku každého programu, ve kterém je tato soustava používána. Její účelem je inicializovat desku a nastavit ji do aktivního stavu, aby bylo možné provádět další operace v rámci programu. Tato inicializační funkce zajišťuje správnou funkci celé soustavy a připravuje ji k provádění požadovaných úkolů.
- `readA0()` – Tato funkce je určena k odečtení hodnoty v mV z integrovaného voltmetru v rámci soustavy. S každou deskou je spojen svůj vlastní voltmetr, proto je nutné specifikovat desku, pro kterou se měření bude provádět. Každé měření vyžaduje tedy specifikaci cílové desky, aby bylo zajištěno správné odečtení hodnoty z odpovídajícího voltmetru. Tímto způsobem se zajišťuje přesnost a spolehlivost měření v rámci celé soustavy.
- `go_forward(speed, steps)` – Tato funkce představuje základní pracovní mechanismus, který umožňuje softwaru nastavit požadovaný posun krokového motoru. Její parametry zahrnují dvě proměnné: `speed` (rychlost) a `steps` (kroky).

Proměnná `speed` určuje rychlost, s jakou se provede jednotlivá akce. Například, pokud je hodnota `speed` nastavena na 20, znamená to, že se krokový motor se bude točit rychlostí 20 kroků za sekundu.

Druhá proměnná, `steps`, určuje, o kolik kroků se má krokový motor posunout. Pokud tedy bude funkce volána s parametry `go_forward(20, 100)`, znamená to, že se krokový motor bude pohybovat vpřed po dobu 5 sekund, přičemž jedna operace bude odpovídat 100 krokům.

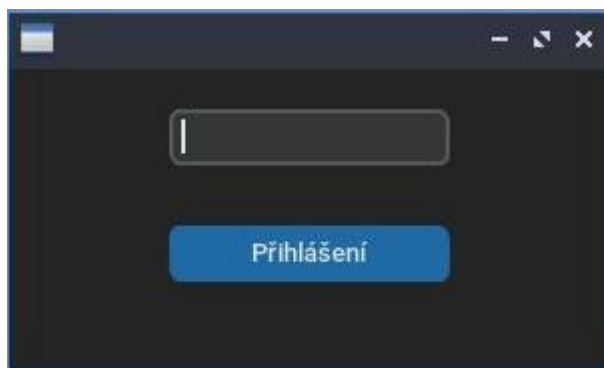
- `get_steps_from_start()` – Funkce je schopna v sobě uchovávat aktuální hodnotu počtu kroků jednotlivých kroků od začátku měření. Tato funkce byla v této bakalářské práci využita jako kontrolní funkce, pomocí které si šlo ověřit, zda se krokový motor posunul o správný počet kroků.

- `go_backward(speed, steps)` – Tato funkce je analogií předchozí funkce, avšak s tím rozdílem, že umožňuje provádět jednotlivé kroky pozpátku. Stejně jako předchozí funkce, i tato má schopnost přijímat záporné hodnoty, což umožňuje flexibilitu v použití. V některých případech může být jedna z těchto funkcí redundantní, protože obě mají podobnou funkčnost. Nicméně, v závislosti na konkrétním použití může být jedna z funkcí preferována před druhou.

5.3 Main

Tato první komponenta, vyvinutá v rámci této bakalářské práce, představuje základní kámen celého grafického uživatelského rozhraní. Právě v této komponentě probíhá inicializace a vytváření grafického uživatelského rozhraní (GUI). Komponenta navíc komunikuje s dalšími třemi komponentami: AdminWindow (administrátorské okno), UserWindow (uživatelské okno) a databází.

V této komponentě se nachází základní třída nazvaná App. Tato třída slouží k inicializaci hlavní okenní aplikace, jak je znázorněno na Obrázku 5.2. Jejím úkolem je zahájit proces vytváření uživatelského rozhraní a umožnit interakci s ním prostřednictvím různých funkcionalit poskytovaných dalšími komponentami.



Obrázek 5.2 Přihlašovací okno, dark mode Zdroj: Vlastní

```

# Zjištění šířky obrazovky
screen_width = self.winfo_screenwidth()

# Zjištění výšky obrazovky
screen_height = self.winfo_screenheight()

# Vypočet prostředku obrazovky
screen_x = (screen_width - width) // 2
screen_y = (screen_height - height) // 2

# Nastavení pozice okna
self.geometry(f"{width}x{height}+{screen_x}+{screen_y}")

```

Kód 5.1: Výpočet pozice okna

Prvním oknem, které se zobrazí při spuštění aplikace, je zároveň autentizačním oknem. Toto okno je vytvořeno pomocí grafické knihovny CustomTkinter. Jeho pozice, stejně jako pozice všech ostatních oken, je vypočítávána dynamicky. Kód 5.1 obsahuje parametry `width` (šířka) a `height` (výška), které slouží k nastavení velikosti okna.

Na Obrázku 5.2 v pravém horním rohu je zobrazena trojice základních tlačítek, která slouží k manipulaci s oknem, jako je minimalizace na lištu, přizpůsobení obrazovce a zavření okna. Tyto funkce jsou automaticky vytvářeny prostřednictvím grafické knihovny CustomTkinter, a proto se v kódu nenachází žádný specifický kód spojený s těmito tlačítky. Tato funkčnost je standardní součástí grafického uživatelského rozhraní a umožňuje uživatelům snadnou interakci s oknem aplikace.

Oknu obrázek 5.2. dominují dvě hlavní widgety (prvky). Prvním widgetem je `CTkEntry` (zadávací) pole. Jedná se o políčko, do kterého může uživatel zapisovat znaky. Následně je možné pomocí funkce `get()` získat tyto znaky jako řetězec a pracovat s nimi v rámci programu. Tato komponenta má dva parametry.

Prvním z nich je `placeholder_text` (drží místo), což je text, který je uvnitř pole a slouží jako nápověda pro uživatele, co má do daného pole zadat. Tento placeholder zmizí poté, co uživatel začne vyplňovat hodnoty. Druhým parametrem je parametr `show`. Tento parametr je především vhodný v případě, kdy je potřeba skrýt, co uživatel do pole zadává, jako například při zadávání hesla. V tomto případě nahradí jednotlivá písmena hesla hvězdičky Obrázek 5.3.



Obrázek 5.3 Přihlašovací GUI, kde je heslo nahrazeno hvězdičkami, light mode Zdroj: Vlastní

Druhým widgetem je `CTkButton` (tlačítko), což je tlačítko, které má dva parametry. Prvním parametrem je `text`, který umožňuje napsat do tlačítka libovolný text. Druhým parametrem je `command` (příkaz), který slouží k přiřazení specifického příkazu k tlačítku.

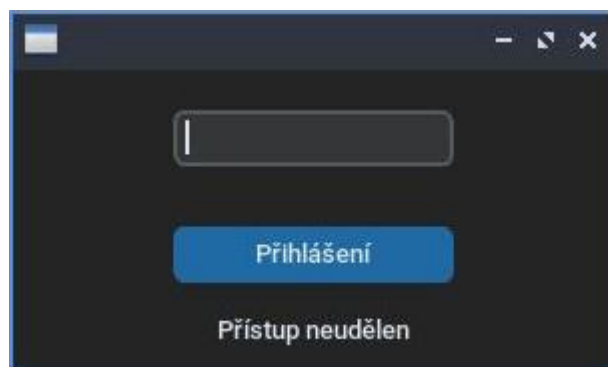
Například v tomto případě je tlačítko použito k provedení série akcí, které autentizují uživatele. Tato funkce je současně přiřazena speciální bindovací (vazební) funkci, která umožňuje určit, jakou klávesovou zkratku použijeme pro vyvolání tohoto úkonu. V tomto kódu byla zvolena standardní klávesa Enter Kód 5.2.

```
self.bind("<Return>", lambda event: self.login())
```

Kód 5.2: přiřazení funkce `login()` fyzické klávese enter.

Autentizační mechanismus byl navržen jednoduše: pokud pracovník zadá uživatelské heslo, dostane se na uživatelské okno; pokud zadá heslo pro administrátora, dostane se do administrátorské části. Tato hesla jsou ověřována v databázi a pokud databáze najde shodu, uživateli je umožněn přístup na základě informace, kterou mu databáze vrátí.

Pro zajištění bezpečnosti uložených hesel je heslo v databázi hashováno. To znamená, že když uživatel potvrdí své heslo, nejprve se toto heslo převede na hash a až poté je posláno na ověření do databáze. Pokud uživatel zadá špatné přístupové heslo, obdrží odpovídající hlášku, jak je znázorněno na Obrázku 5.4.



Obrázek 5.4 Přihlašovací okno přístup neudělen Zdroj: Vlastní

Pokud uživatel zadá správné heslo, program vytvoří nové okno typu `CTkToplevel`, buď pro administrátorskou nebo uživatelskou část. Toto nové okno se otevře a současné okno, ve kterém probíhalo zadávání hesla, bude automaticky zavřeno.

`CTkToplevel` je speciální typ okna, který uchovává informace o svém nadřazeném okně. Díky tomu je schopno se k nadřazenému oknu vrátit, aniž by muselo být vytvořeno nové okno. To umožňuje efektivněji spravovat okna v aplikaci a poskytuje uživateli plynulý zážitek z používání.

5.4 AdminWindow

Komponenta `AdminWindow` je přístupná pouze po zadání správného administrátorského hesla. Administrátor by měl být osobou, které lze svěřit náročnější úkoly bez takové míry kontroly jako u normálních uživatelů. Tato premisa byla zohledněna i v tomto projektu, což umožňuje administrátorovi provádět změny v systému.

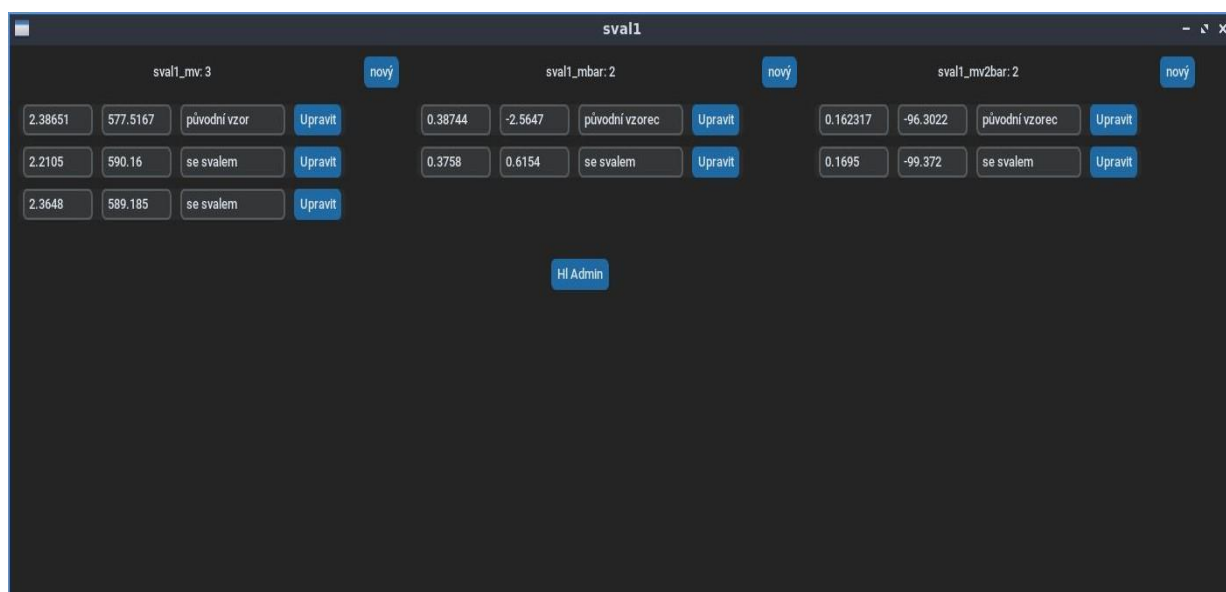
Po úspěšném přihlášení pomocí správného hesla uvidí administrátor první obrazovku, kde bude moci vybrat mezi dvěma úkony, jak je znázorněno na Obrázku 5.5. Tato obrazovka mu poskytuje možnost provést specifické úkony v rámci svých administrátorských práv.



Obrázek 5.5 Úvodní okno administrátorské části, light mode Zdroj: Vlastní

V nejspodnější části okna se nachází tlačítko s názvem "Hlavní Stránka". Toto tlačítko umožňuje návrat zpět na autentizační obrazovku. Je vhodné především pro administrátory, kteří chtějí například otestovat nový převodní vzorec po jeho nastavení. Tlačítko jim poskytuje možnost vrátit se na autentizační obrazovku, aniž by museli celou aplikaci vypínat a znovu spouštět. Po stisknutí tlačítka se aktivuje funkce, která zavře aktuální okno a znovu otevře autentizační obrazovku.

V prostřední části okna se nachází 5 tlačítek. Tyto tlačítka umožňují administrátorovi přistoupit k databázovému oknu, jak je znázorněno na Obrázku 5.6. Byla zvolena metoda vytvoření tlačítka pro každý sval z důvodu, že jméno tohoto tlačítka je předáno do dalšího okna, jak je uvedeno v Kódu 5.3, a následně s těmito hodnotami lze pracovat.



Obrázek 5.6 GUI reprezentace databáze pro sval 1 Zdroj: Vlastní


```
command=lambda: self.databaseWindow("sval1")
```

Kód 5.3: Nastavení parametru tlačítka `command` na lambda funkci, která je schopna předávat string (řetězec písmen)

Hodnota svalu je následně využívána k vybrání požadovaných tabulek z databáze, jak je znázorněno v Kódu 5.4. V jazyce Python je k dispozici funkce f-string, která umožňuje vkládat proměnné přímo do řetězců pomocí složených závorek `{}`. Tím se kód stává přehlednější, než je tomu u klasického zápisu, kdy se musí jednotlivé řetězce a proměnné spojovat pomocí logického operátoru plus `+`. V tomto případě byla tato metoda použita pro všechny čtyři převodní vzorce, což usnadňuje čitelnost a údržbu kódu.

```
self.load_table(f"{sval}_mm", 0)
```

Kód 5.4: Zavolání funkce pro načtení požadované tabulky z databáze

Toto okno (Obrázek 5.6) je rozděleno do pěti hlavních částí. První část obsahuje tlačítko pro návrat na hlavní administrátorskou obrazovku, kde si uživatel může vybrat jiný sval pro práci s jeho databází nebo provést jinou akci.

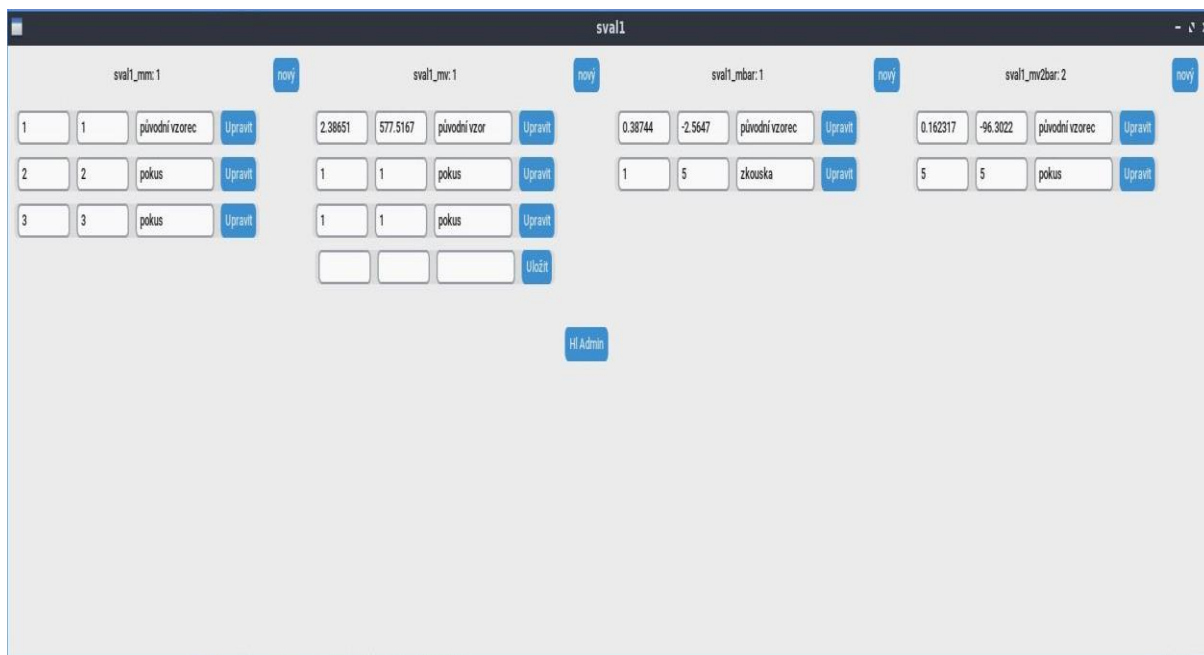
Dále jsou zde tři téměř identické části, lišící se pouze tím, z jaké tabulky v databázi načítají data. Význam jednotlivých převodních vzorců je dále vysvětlen v kapitole 5.5 UserWindow. Tyto části slouží k práci s databází a umožňují administrátorovi provádět různé operace v závislosti na zvoleném svalu.

Vedle názvu tabulky se nachází číslo, které indikuje, který převodní vzorec je zrovna aktivní. Například na obrázku 5.6, vedle vzorce pro sval1 pro převodní vzorec mezi kroky a mV je číslo 3. To značí, že pokud se uživatel rozhodne využít výpočet, který bude počítat s tímto vzorcem, bude použit právě tento převodní vzorec z databáze.

Každá část okna obsahuje tři prvky. Prvním prvkem jsou vstupní pole, ve kterých se nachází sklon, posun a popis funkce. Tento prvek byl záměrně zvolen jako vstupní pole, aby administrátor nemusel složitě spouštět editování jednotlivých vzorců. Stačí pouze změnit libovolnou část převodního vzorce na požadovanou hodnotu a následně využít druhý prvek, kterým je tlačítko "upravit".

Toto tlačítko slouží k uložení provedených změn do databáze. Zároveň disponuje funkcí, pomocí které se volí aktuální převodní vzorec. V prostředí, kde soustava obsahuje mnoho převodních vzorců pro různě těžké hlavy pacientů, je rychlá změna mezi těmito vzorci nezbytná. Stačí jednoduše zmáčknout tlačítko "upravit" a tím se nastaví nový aktuálně používaný převodní vzorec. Tímto způsobem je administrátor schopen rychle a efektivně pracovat s různými převodními vzorci v systému.

Posledním prvkem je tlačítko "nový". Pomocí tohoto tlačítka je administrátor schopen vytvářet nové převodní vzorce. Po stisknutí tlačítka se objeví nová vstupní pole spolu s tlačítkem "uložit", jak je znázorněno na Obrázku 5.7. Zde musí administrátor vyplnit vstupní pole příslušnými hodnotami. Po dokončení vyplnění stiskne tlačítko "uložit" a tím se nový převodní vzorec uloží do databáze. Po vložení do databáze se ovšem vzorec nenastaví jako aktivní. Pro to, aby ho administrátor z aktivoval musí zmáčknout tlačítko „upravit“. Tímto způsobem je umožněno administrátorovi snadno a rychle přidávat nové převodní vzorce do systému.



Obrázek 5.7 GUI databáze, když administrátor vytváří nový převodní vzorec.

Zdroj: Vlastní



Obrázek 5.8 GUI Uživatele Zdroj: Vlastní

Druhé pracovní okno administrátorské části slouží k vytváření měření pro následnou práci na vytvoření nových převodních vzorců Obrázek 5.8. Toto okno je rozděleno do dvou rámců, což umožnilo vytvoření přehledného uživatelského prostředí s jasně rozdělenými prvky.

V levé části okna se nachází CtkRadioButton (přepínač), což je speciální typ tlačítka, který kombinuje vlastnosti tlačítka a výběrového pole. Uživatel má možnost vybrat jednu možnost ze seznamu, ale na rozdíl od běžného výběrového pole může vybrat vždy pouze jednu možnost. Tato komponenta je velmi vhodná pro případy, kdy je třeba uživateli umožnit vybrat pouze jednu možnost, jako například v tomto případě, kde je nutné zajistit, aby uživatel mohl ovládat pouze jeden sval.

Ve spodní části tohoto rámu se nachází komponenta CtkOptionMenu (rozhodovací menu). Tato komponenta představuje rozbalovací seznam, který zobrazuje pouze aktivní prvek. Když je tato komponenta rozkliknuta, zobrazí všechny možnosti nastavení, které jsou "Dark, Light, Systém". Tato nastavení určují barevné schéma aplikace. Díky CustomTkinteru je možné nastavit tmavý nebo světlý režim manuálně nebo automaticky podle systému. V tomto projektu byla vybrána možnost automatického barevného režimu podle systému, což umožňuje aplikaci automaticky přizpůsobit nastavení systému, na kterém je spuštěna.

V nejspodnější části levé části okna se nachází tlačítko „Hl admin“ toto tlačítko slouží k tomu, aby se admin mohl dostat zpět na první administrátorskou obrazovku.

V druhém fragmentu se nachází hlavní pracovní část tohoto okna. Zde jsou umístěna tři vstupní pole, do kterých uživatel musí zadat číselnou hodnotu.

Prvním vstupním polem je pole speed, kam administrátor musí zadat požadovanou rychlost, kolik kroků chce, aby soustava vykonala.

Druhým vstupním polem je pole steps, kam administrátor musí zadat, kolik kroků má soustava vykonat.

Posledním vstupním polem je pole s placeholder textem "HW Value" (hodnota), kam administrátor musí ručně zadávat hodnoty z externího tlakoměru v jednotkách mbar.

Tato vstupní pole také disponují validačním mechanismem, který zamezuje zadání jiného vstupu než čísla. Tím je zajištěna integrita dat a správnost vstupů od uživatele.

Tlačítko "Krok" je schopno potvrdit celý proces a provést měření provedením požadovaného postupu na krokovém motoru.

S tímto tlačítkem "Krok" je spojeno také tlačítko "Pokračovat". Toto tlačítko slouží jako bezpečnostní prvek pro ověření, zda administrátor připojil všechny potřebné části a je připraven provádět měření na novém svalu. Po stisknutí tlačítka "Pokračovat" je provedena kontrola správnosti nastavení a připojení součástí a teprve poté je umožněno provést krokové měření. Tímto způsobem je zajištěna bezpečnost a spolehlivost celého procesu měření.

Proces měření dat je navržen tak, aby administrátor měl k dispozici jeden externí tlakoměr, pomocí kterého může měřit tlak v jednotlivých svalech soustavy. Proto soustava potřebuje určité zajištění, aby se zabránilo situacím, kdy by administrátor začal provádět měření na svalu, aniž by například již měl přepojený tlakoměr na nový sval.

Toto tlačítko slouží jako pojistka: když se administrátor rozhodne měřit jiný sval, soustava v programu ověří, zda je současný sval jiný než minulý sval, a tím spustí sérii akcí. Soustava aktivuje tlačítko pro potvrzení, že vše je připraveno k měření. Dokud toto tlačítko nebude stisknuto, soustava nedovolí administrátorovi pokračovat v práci na soustavě, čímž zajišťuje bezpečnost a správnost prováděných operací.

Další akcí, která se spustí, je automatické navrácení původního měřeného svalu na technickou nulu. Program si pamatuje, kolik kroků od začátku udělal, a poté, když uživatel zvolí jiný sval, program začne automaticky sjíždět na technickou nulu. Až poté, co soustava sjede na technickou nulu, bude administrátorovi umožněno stisknout tlačítko "Pokračovat" a provést následné měření.

Posledním tlačítkem tohoto grafického uživatelského rozhraní je „Technická nula“. Jedná se o tlačítko s funkcionalitou, která zajistí, že veškeré svaly budou ve svých nulových hodnotách. Tato funkcionalita má za úkol zajistit co největší přesnost měření při kterém se bude vytvářet převodní vzorec.

Tímto způsobem je zajištěno, že soustava se před každým novým měřením vrátí do výchozí polohy, což je důležité pro správnost a konzistenci prováděných měření. Administrátor je také chráněn před nechtěnými chybami, protože soustava automaticky provádí proces nastavení technické nuly, než mu umožní pokračovat v další práci.

Hlavní funkcí tohoto měření je zaznamenávání jednotlivých hodnot měření do souboru .xlsx. Tuto funkcionalitu zajišťuje knihovna openpyxl. Program do tohoto souboru ukládá následující informace:

- Počet kroků motoru od začátku měření
- Aktuální hodnota interního voltmetru v mV
- Hodnota fyzického tlakoměru v mbar
- Rychlost, jakou se v danou chvíli krokový motor pohyboval

Všechny tyto hodnoty jsou ukládány do souboru, dokud administrátor nezmění sval. Poté soustava provádí všechny kroky spojené s tímto úkonem a uloží tento soubor na plochu počítače, aby byl snadno dostupný pro další práci. Tím je zajištěno, že veškerá data jsou systematicky zaznamenána a uložena pro budoucí použití.

Tento program je navržen tak, aby fungoval pro 5 jednotlivých svalů, přičemž každý z nich funguje úplně stejným způsobem, s výjimkou jejich názvu. Existuje však riziko vážné redundance kódu, pokud by se každý sval musel řešit individuálně, což by mohlo vést k duplikaci kódu a zvýšení náročnosti na údržbu.

Aby se tomu předešlo, vytvořil se slovník, ve kterém jsou definovány jednotlivé desky, které komunikují s jednotlivými svaly. Poté se ze slovníku vybírá ta správná deska podle čísla svalu. Tímto způsobem se minimalizuje duplikace kódu a zajišťuje se, že veškeré operace se svaly jsou prováděny efektivně a jednotně.

Kód 5.5 obsahuje tuto logiku, která umožňuje programu dynamicky pracovat s více svaly a minimalizuje riziko chyb spojených s duplikací kódu.

V kódu 5.5 je vidět, že se vybírá ten sval, který odpovídá hodnotě minulého svalu. Tento kód byl vyjmut z části programu, kde program pohyboval minulým svalem na technickou nulu.

```
b_objects = {1: b1, 2: b2, 3: b3, 4: b4, 5: b5}

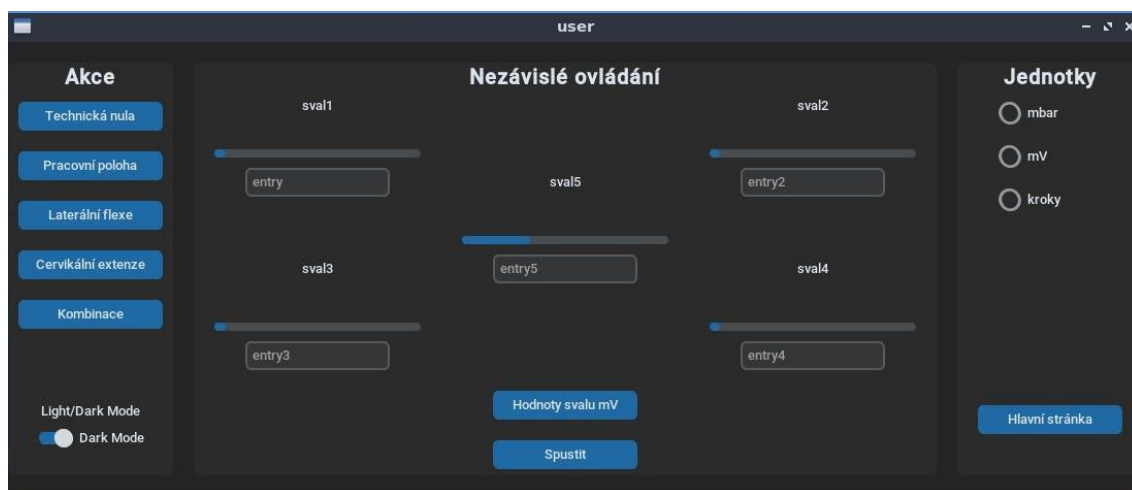
b_objects[Minuly_Sval].go_backward(Speed, Steps)
```

Kód 5.5: Využití slovníku pro omezení redundance kódu

5.5 UserWindow

UserWindow představuje poslední část této bakalářské práce a je určena pro uživatele (user). Tato komponenta je spuštěna pouze v případě, že se uživatel přihlásí pomocí správného uživatelského hesla, které odpovídá heslu uloženému v databázi a je označeno jako uživatelské. UserWindow komunikuje s hlavním modulem (Main), databází a modulem pro načítání zařízení (deviceloader).

Toto okno se skládá ze tří rámců (framů), které jsou zobrazeny na Obrázku 5.9. Každý z těchto rámců slouží k organizaci a přehlednému zobrazení různých funkcí a informací, které jsou uživateli k dispozici. UserWindow je navrženo tak, aby uživatel mohl snadno a efektivně pracovat s aplikací a využívat všechny její funkcionality v souladu s jeho oprávněními a potřebami.



Obrázek 5.9 GUI uživatel Zdroj: Vlastní

První frame jménem „Akce“ obsahuje tlačítka pro jednotlivé akce. V této bakalářské práci se vytvořilo 5 akcí, které si uživatel může volit podle svých preferencí.

První akcí, která se skrývá pod tlačítkem „Technická nula“, je funkcionality, která se nachází i v jiných částech programu a slouží k tomu, aby si uživatel mohl. Sousta nastavit do výchozí polohy a začít s novou prací svalů.

Druhou akce je „Pracovní poloha“ jedná se o akci pomocí které veškeré svaly se dostanou do své základní polohy tato poloha. Pracovní poloha se vyznačuje tím, že veškeré svaly se dotýkají podložky, se kterou pohybují. V programu je tato hodnota nastavena napevno na 650 mV.

Soustava musí vypočítat kolik kroků musí provést, aby se dostala na požadovaný tlak to se dělá pomocí převodní rovnice, kterou si program načte z databáze a následně načte aktuální hodnotu v mV a požadovanou hodnotu v mV obě tyto hodnoty převede na kroky a následně tento výsledek vloží do funkce která pohybuje s krokovým motorem.

Třetí akcí a zároveň první akcí která provádí pohyby hlavou je akce „Laterální flexe“. Jedná se o akci, která má za úkol natočit pacientovu hlavu první do levé a poté do pravé strany, a nakonec vrátit hlavu zpět do původní polohy. Zde jsou také nastaveny pevné hodnoty, do kterých se mají svaly dostat. Po testování jako vhodná poloha vyšla tak že svaly s vyšším zdvihem, a tudíž i tlakem budou nastaveny na hodnotu 700 mV a svaly s nižším zdvihem budou nastaveny na hodnotu 630 mV.

Čtvrtá akce „Cervikální extenze“ je analogie třetí akce jenom s tím rozdílem, že hlava pacienta se nepohybuje zleva doprava, ale shora dolů (pacientovy se zvedá čelo a klesá brada a obráceně), proto se tato akce jmenuje „Nahoru Dolu“.

Poslední pátou akcí je akce jménem „Kombinace“ jedná se o akci, která doslova kombinuje postupně dvě předchozí akce. Z důvodu zamezení redundance a požadavků na soustavu bylo možné v poslední funkci pouze volat dvě předchozí funkce a tím provést požadované úkony Kód 5.6.

```
def kombinace(self):  
    self.levo_pravo()  
    time.sleep(5)  
    self.nahoru_dolu()
```

Kód 5.6: Volání jednotlivých funkcí ve funkci kombinace

Druhým typem widgetu je CTkSwitch (přepínač), což je komponenta umožňující uživateli zapnout nebo vypnout určitou funkci. V tomto případě slouží k přepínání mezi light (světlý) mode a dark (tmavý) mode. Když uživatel poprvé spustí toto okno, automaticky se mu zapne režim, který je předvolený podle systému. Avšak uživatel má možnost tuto volbu změnit podle svých preferencí.

CTkSwitch poskytuje uživateli snadný způsob, jak přizpůsobit zobrazení aplikace svým potřebám a vkusu. Tímto způsobem může uživatel přizpůsobit prostředí aplikace tak, aby mu co nejlépe vyhovovalo a usnadňovalo práci či interakci s aplikací.

Druhým rámcem je "Jednotky". Tato část obsahuje radiobuttony pro výběr jednotek, které uživatel může vybrat podle svých preferencí ohledně ovládání pneumatické soustavy. Radiobuttony byly zvoleny, protože je nutné zajistit, aby uživatel vybral pouze shodnou jednotku pro ovládání. Výběr pomocí checkboxů (zaškrťovací políčko) by vyžadoval další funkce, které by zajistily, že každý nový výběr uživatele automaticky zruší předchozí výběr. Radiobuttony řeší tuto potřebu automaticky tím, že uživatel může vybrat pouze jednu možnost ze seznamu.

Tento radiobutton má také vlastní validaci, která zajišťuje, že uživatel, který nevybere žádnou jednotku, bude upozorněn na tuto skutečnost a zároveň mu nebude umožněno pokračovat dále, dokud neodklikne vyskakovací okno. Tím je zajištěno, že uživatel bude informován o nutnosti vybrat jednotky pro zadávání a že nebude možné pokračovat v procesu, dokud tato podmínka není splněna.

Druhým prvkem v tomto rámu je tlačítko "Hlavní stránka". Pomocí tohoto tlačítka se uživatel může vrátit zpět na první okno, kde zadával heslo pro přihlášení. Toto tlačítko je vhodné především pro administrátory, kteří současně pracují jako uživatelé a mohli by zapomenout změnit převodní vzorce. Díky tomuto tlačítku se eliminuje potřeba vypnout a znovu spustit aplikaci, což uživatelům ušetří čas a zlepší jejich uživatelskou zkušenost.

Hlavním prvkem okna je "Nezávislé ovládání". V tomto prvku bude uživatel schopen zadávat hodnoty, které ovlivní tlak pneumatické soustavy. Každé entry pole je označeno názvem příslušného svalu jako placeholder. Pro zajištění bezpečnosti a správného zadávání čísel byla provedena validace vstupu. Program je schopen přijímat pouze číselné hodnoty, protože neví, na jakou hodnotu by měl převést text. Stejný typ validace se používá i v dalších částech programu pomocí regulárních výrazů. Další částí validace je validace desetinných míst. Program disponuje také validací, že uživatel je schopen zadávat pouze celá čísla.

Dalším bezpečnostním prvkem, který tento rámec obsahuje, je CTKProgressBar (pruh průběhu), který se nachází nad entry polem. Tato komponenta zobrazuje aktuální stav tlaku pneumatického svalu. Slouží jako informační prvek pro uživatele, který jim umožňuje určit, jaký je momentální stav natlakování soustavy a kolik je ještě možné přidat nebo ubrat tlaku. Tento prvek není pouze vizuální kontrolou, ale je také nastaven tak, aby upozornil uživatele, pokud se pokusí překročit povolenou hranici tlaku svalu. To je důležité zejména pro ochranu pacienta, který je v blízkosti těchto svalů.

Horní hranice je nastavena tak, aby se zabránilo možnému nevratnému poškození pneumatického svalu a tím i ohrožení pacienta. Spodní hranice je stanovena tak, aby se zabránilo poškození ventilu v případě, že by uživatel neustále snižoval tlak, i když by byl ventil již uzavřen. Tím se minimalizuje riziko poškození ventilu a zajišťuje bezpečnost a spolehlivost systému.

Tato ochrana není ovšem pouze vizuální, ale je také implementována v sw. Tato ochrana je navržena tak že v se vždy nejdříve vypočítá, do jaké polohy se sval dostane, to se porovná s maximální povolenou mezí. Pokud ovšem u jednoho svalu tato validace neproběhne úspěšně a program objeví, že by došlo k překročení povolené hranice ukončí operace bez provedení jakékoliv akce a uživatele o této skutečnosti bude informovat.

Tento rámec obsahuje jediné tlačítko, které spouští celkový krok celé soustavy najednou. Soustava je nakonfigurována tak, že pokud by měla být posunuta o 0 kroků, tento krok se vůbec neuskuteční a přeskočí se.

V UserWindow bylo nejdůležitější a zároveň nejsložitější částí navrhnout výpočet pro správné přepočty mezi jednotlivými jednotkami. Důležité bylo zaměřit se na omezení redundance, která hrozila zejména ve dvou případech. V této soustavě se nachází 5 svalů, pro její správné fungování jsou potřeba 3 převodní vzorce.

Nejjednodušší jednotkou pro práci jsou kroky, protože se nemusí nijak upravovat. Stačí pouze přenést jejich přesnou hodnotu do funkce `go_forward` nebo `go_backward`.

V tomto projektu byla možnost zadávání záporných hodnot řešena tak, že byla nastavena podmínka, která rozděluje tyto hodnoty na kladné a záporné. Pokud se jednalo o zápornou hodnotu, která měla vstoupit do funkce `go_backward()`, byla tato hodnota vložena jako záporná, čímž se její zápornost zneplatnila a hodnota tak vstupovala do funkce jako kladná. Tímto způsobem byla zachována konzistence a správné fungování funkce pro pohyb krokových motorů.

Redundance v této části kódu byla elegantně vyřešena pomocí jednoho slovníku a jednoho pole, jak je patrné z Kódu 5.6. Tímto způsobem je program schopen na jednom řádku dosáhnout stejného výsledku, kterého by jinak dosáhl opakovaným psaním téže funkce pro každý sval. Stačí použít jeden `for` cyklus, který provede akci pro každý sval, což výrazně zjednodušuje a zkracuje kód.

```
svaly[i].go_forward(10, results[i])
```

Kód 5.6: Omezení redundance

Pro změnu tlaku za pomoci milivoltů je nutné použít převodní rovnici z databáze. Vzhledem k tomu, že v databázi není pouze jedna převodní rovnice, program musí zjistit, jaký převodní vzorec má načíst z dané tabulky. K tomu slouží Kód 5.7. Tento kód omezuje redundanci tím, že načítá informace pro každý index, který může nabývat hodnot od 1 do 5. Poté se tyto informace spojí se stringem "sval" a dále se spojí s příslušnou jednotkou, kterou získáme z radiobuttonu. Následně se může provést výběr celého požadovaného řádku s příslušným svalem. Tímto způsobem je zajištěno efektivní a dynamické načítání převodních vzorců bez zbytečné redundance v kódu.

```

cur.execute(f"SELECT sval{index}_{jednotka} FROM sval{index}")

# Spojení navráceného SQL příkazu do stringu
cislo_vzorcu = cur.fetchone()

cur.execute(f"SELECT * FROM sval{index}_{jednotka} WHERE id IN
({int(cislo_vzorcu[0])})")

```

Kód 5.7: Výběr příslušné rovnice z databáze

Po načtení příslušného převodního vzorce program provádí výpočet konečné hodnoty, na kterou má do iterovat. Toho dosahuje pomocí aktuální hodnoty v mV načtené z tlakoměru, ke které přičítá posun, který si uživatel sám volí.

Následně je nutné převést startovní a konečné hodnoty v mV na jednotky, které je krokový motor schopný přijmout. Pro toto je potřeba spočítat startovní kroky, tedy kolik kroků bylo provedeno od začátku procesu. Alternativní možností by bylo uchovávat tuto informaci v programu, ale kvůli přesnosti nastavení hodnot je tato hodnota počítána pokaždé znovu.

Poté se vypočítá, kolik kroků motor musí provést, aby se dostal do této pozice z technické nuly. Nakonec se tyto dvě hodnoty odečtou a program si tento výsledek uloží do pole, do kterého se ukládají veškeré hodnoty svalů. Po naplnění tohoto pole program začne provádět pohyb krokových motorů, čímž dosáhne požadovaného tlakového stavu.

Vyřešení jednotek, které nemají žádné referenční hodnoty ze soustavy, bylo zásadním úkolem. Pro milivoltovou jednotku je soustava schopna odečíst hodnoty z integrovaných voltmetrů, což je relativně snadné. Ovšem pro jednotky tlaku tato možnost neexistuje. Jednou z možností by bylo přidání senzorů, které by dodaly referenční hodnoty, ale to by nejen bylo ekonomicky náročné, ale také by vyžadovalo přepracování celé soustavy.

Proto byla zvolena metoda, kdy se jako referenční hodnoty využívají hodnoty v mV, které se následně převedou na příslušné jednotky pomocí převodní funkce. Pomocí referenční hodnoty se vypočítají počáteční a konečný krok, které se následně odečtou a získá se výsledek. Pouze u jednotek mbar je potřebný ještě jeden mezistupeň, kdy se aktuální hodnota z mV převede na požadovanou jednotku, čímž se vytvoří referenční hodnota.

Po provedení veškerých úprav tlaků se přepočítají hodnoty progressbarů na nové hodnoty, aby se zajistilo, že při dalším opakování měření nedojde k jejich překročení. Tímto způsobem je zajištěno, že proces měření probíhá bezpečně a spolehlivě.

Posledním prvkem, který se nachází v uživatelské části je prvek „Hodnoty svalu mV“, který je schopný vypsát aktuální hodnoty konkrétních voltmetrů u jednotlivých svalů Obrázek 5.11. Tato funkcionality slouží především k tomu, aby měl uživatel číselnou reprezentaci, v jaké poloze se aktuálně konkrétní sval nachází.



Obrázek 5.10 GUI uživatele s hodnotami svalů v mV

6 Uživatelská dokumentace

Tento dokument slouží jako uživatelský manuál pro práci se systémem. Je rozdělen do dvou částí: jedna je určena uživatelům a druhá administrátorům. Pro optimální fungování systému je zásadní provést kontrolu všech komponent a zajistit jejich správné spuštění. Zvláště je důležité věnovat pozornost kompresoru, který není přímou součástí systému, avšak hraje klíčovou roli v jeho činnosti.

6.1 Dokumentace pro uživatele

1. Spuštění programu:

- Metoda 1: Přes ikonu programu
 - Najed'te kurzorem myši na ikonu programu.
 - Dvakrát zmáčkněte levé tlačítko myši.
- Metoda 2: Alternativní možnost
 - Klikněte pravým tlačítkem myši na ikonu programu.
 - Z nabídky vyberte možnost "Otevřít program" a klikněte levým tlačítkem myši.

2. Autentizační okno:

- Po úspěšném spuštění programu se zobrazí autentizační okno.
- Toto okno slouží k ověření uživatele/administrátora.

3. Zadání hesla:

- Do pole s textem „Heslo“ vložte své heslo.
- Pro zadání hesla do tohoto pole:
 - Klikněte levým tlačítkem myši na toto pole.
 - Nebo použijte klávesu "Tab" pro zaměření na toto pole a poté zadejte heslo.

4. Potvrzení hesla:

- Pro potvrzení a ověření zadaného hesla klikněte na tlačítko "Přihlášení".
- Potvrzení lze provést dvěma způsoby:
 - Klikněte levým tlačítkem myši na tlačítko "Přihlášení".
 - Použijte klávesu "Enter" na klávesnici. (Poznámka: Numerický Enter není pro tuto činnost naprogramován.)

5. Zpráva o neúspěšném přihlášení:

- Pokud se přihlášení nezdaří, budete o této skutečnosti informováni hláškou „Přístup neudělen“ pod tlačítkem.

6. Úspěšné přihlášení:

- Při úspěšném přihlášení se ověřovací okno zavře.
- Otevře se nové okno, které je zároveň hlavním pracovním oknem uživatelské části.

7. Volba akcí v levé části:

- V levé části rozhraní můžete vidět tlačítka akcí.
- Tyto akce můžete vybírat podle svého uvážení.
- „Technická nula“ pomocí tohoto tlačítka nastavíte soustavu do výchozí nulové pozice. Tato pozice je vhodná provést před začátkem práce a po ukončení práce na soustavě
- „Pracovní poloha“ jedná se o polohu, kdy se svaly nafouknou do takové polohy kdy se všechny budou lehce dotýkat podložky, se kterou budou následně pohybovat.
 - Z této polohy se doporučuje vždy začínat jednotlivé úkony
- „Laterální flexe“ tato akce provede pohyb hlavy pacienta nejdříve vlevo a poté vpravo. Po provedení těchto akcí se svaly vrátí automaticky do pozice „Pracovní poloha“.
- „Cervikální extenze“ tato funkce je analogie předchozí funkce pouze s tím rozdílem že tato funkce jako první akci provede pohyb čela vzhůru a brady dolů a následně obráceně. Na konci procedury se opět vrátí do „Pracovní poloha“.

8. Přepínání mezi módy v levé části:

- Nejspodnější komponentou v levé části je switch.
- Tento switch slouží k přepínání mezi "light" a "dark" mode.

9. Volba jednotek v pravé části:

- V pravé části se nacházejí Radiobuttony pro výběr jednotek, ve kterých chcete zadávat hodnoty.
- Můžete si vždy vybrat pouze jednu jednotku, ve které budete měnit tlak v jednotlivých svalech soustavy.
- Pokud se rozhodnete změnit jednotky, pouze klikněte na novou jednotku, kterou si přejete používat, a program automaticky vybere novou a zruší výběr minulý.

10. Návrat na hlavní obrazovku:

- Pod radiobuttony se nachází tlačítko "Hlavní stránka."
- Pomocí tohoto tlačítka se můžete vrátit zpět na hlavní obrazovku.
- Například můžete použít toto tlačítko k návratu na hlavní obrazovku pro zadání hesla pro administrátora, pokud jste administrátor.

11. Hlavní pracovní jednotka:

- Hlavní pracovní oblastí programu je prostřední část, kde jsou zobrazeny 5 různých skupin komponentů.
- Tyto skupiny komponentů jsou identické, liší se pouze názvem svalu.

12. Rozdělení svalů:

- Svaly jsou rozděleny podle funkce:
 - Sval 1 až Sval 4 zvedají hlavu.
 - Sval 5 provádí pohyby od těla k tělu.
- Svaly jsou dále rozděleny v GUI:
 - Svaly 1 a 2 jsou umístěny dále od pacientova těla
 - Svaly 3 a 4 jsou umístěny blíže k pacientovým ramenům.

13. Funkce komponenty progress bar:

- Všechna políčka fungují na stejném principu, proto se postup práce vysvětlí na jednom z nich. Pro ilustraci práce je zvolen Sval 1. Pod označením svalu se nachází komponenta progress bar, která slouží jako signalizace aktuálního procentuálního tlaku ve svalech.
- Svaly jsou nastaveny tak, aby měly jak horní, tak dolní možnou hranici tlaku.
 - Horní hranice je nastavena tak, aby nedošlo k přetlakování svalu a jeho následnému prasknutí.
 - Dolní hranice je zde pro zabránění přetočení ventilu regulujícího tlak v jednotlivých svalech.
 - Tato mez není pouze optická, ale je také softwarově nastavená. To znamená, že pokud se rozhodnete změnit tlak tak, že by tato hodnota překročila tuto mez, budete o této skutečnosti informováni.
- Pokud se pokusíte provést sérii akcí a tlak by překročil určenou mez v jednom nebo více svalech, program vás informuje o tom, že série akcí nemohla být provedena z důvodu překročení limitu tlaku.

14. Zadávání hodnot do entry polí:

- Hodnoty, které chcete zadávat jednotlivým svalům, se zadávají do jednotlivých entry polí, která se nachází pod již zmíněnými progress bary.
- Do těchto polí zadáváte vždy kladnou číselnou hodnotu, pokud chcete tlak zvyšovat. Pokud chcete tlak v jednotlivých svalech snižovat, zadáte do požadovaného entry pole hodnotu s mínusem na začátku.
- Jednotlivé svaly můžete ovládat nezávisle na sobě, což znamená, že klidně dva svaly můžete nechat zvýšit svůj tlak a u dalších dvou tento tlak snížit.
- Jednotlivá políčka obsahují validaci, což znamená, že pokud se rozhodnete zadat hodnotu, kterou soustava není schopna zapsat do svalu (například písmeno nebo speciální znak), budete upozorněni. Totéž platí, pokud necháte pole prázdné.
 - Soustava je nastavena tak, aby bylo vždy nutné vyplnit všechna pole. Pokud nechcete měnit tlak na určitém svalu, zadáte do pole jednoduše nulu
- Pokud se programu nebude líbit vaše zadaná hodnota, budete na tuto skutečnost upozorněni buď ihned po kliknutí mimo pole nebo poté, co zmáčknete tlačítko "Spustit".
- V soustavě není možné zadávat desetinná místa.
- Mezi jednotlivými entry poli můžete normálně přecházet pomocí levého tlačítka myši nebo jednoduše stisknutím klávesy "Tab", což vás automaticky přesune na další entry pole.

15. Spouštění akce "Spustit":

- Poté, co jste spokojeni se všemi zadanými hodnotami, můžete zmáčknout tlačítko "Spustit". Toto tlačítko nejprve znovu zvaliduje všechna entry pole, aby nedošlo k chybnému fungování soustavy.
- Poté se provedou veškeré potřebné výpočty a soustava změní tlaky v jednotlivých svalových skupinách.

16. Hodnoty svalu v mV

- Pokud si budete chtít zjistit jakou hodnotu v mV má jaký krokový motor. Lze to udělat pomocí tlačítka „Hodnoty svalu mV“.
 - Tato funkcionality je zde především kdybyste chtěli například vyrovnat dva svaly do stejné výšky.

6.2 Dokumentace pro administrátora

1. Spuštění programu:
 - Metoda 1: Přes ikonu programu
 - Najed'te kurzorem myši na ikonu programu.
 - Dvakrát zmáčkněte levé tlačítko myši.
 - Metoda 2: Alternativní možnost
 - Klikněte pravým tlačítkem myši na ikonu programu.
 - Z nabídky vyberte možnost "Otevřít program" a klikněte levým tlačítkem myši.
2. Autentizační okno:
 - Po úspěšném spuštění programu se zobrazí autentizační okno.
 - Toto okno slouží k ověření uživatele/administrátora.
3. Zadání hesla:
 - Do pole s textem „Heslo“ vložte své heslo.
 - Pro zadání hesla do tohoto pole:
 - Klikněte levým tlačítkem myši na toto pole.
 - Nebo použijte klávesu "Tab" pro přesun na toto pole a poté zadejte heslo.
4. Potvrzení hesla:
 - Pro potvrzení a ověření zadaného hesla klikněte na tlačítko "Přihlášení".
 - Potvrzení lze provést dvěma způsoby:
 - Klikněte levým tlačítkem myši na tlačítko "Přihlášení".
 - Použijte klávesu "Enter" na klávesnici. (Poznámka: Numerický Enter není pro tuto činnost naprogramován.)
5. Zpráva o neúspěšném přihlášení:
 - Pokud se přihlášení nezdaří, budete o této skutečnosti informováni hláškou „Přístup neudělen“ pod tlačítkem.
6. Úspěšné přihlášení:
 - Při úspěšném přihlášení se ověřovací okno zavře. Otevře se nové okno, které umožní administrátorovi vybrat si další možnost práce.
7. Tlačítko "Hlavní stránka":
 - Tlačítko "Hlavní stránka" slouží administrátorovi k návratu zpět na hlavní přihlašovací okno.

8. Tlačítko "Měření":

- Po zvolení tlačítka "Měření" se dostanete do okna měření.
- Toto okno slouží k provádění měření, a na základě tohoto měření se automaticky vytváří soubor typu .xlsx.
- Tento soubor se ukládá na plochu pro snadnou manipulaci.
- Vždy se vytváří soubor s odpovídajícím jménem svalu.
- Jednotlivé soubory se při vícenásobném měření na jednom svalu přepisují, proto je důležité buď tyto soubory zpětně nepožadovat, nebo je ukládat na jiné místo, kde budou dlouhodobě uchovány.

9. Výběr svalu pro měření:

- V levé části se nacházejí Radiobuttony, pomocí kterých můžete vybrat sval, na kterém budete provádět měření.
- Vybrán může být vždy pouze jeden sval, tato podmínka je již zabudovaná v Radiobuttonech, protože se jedná o komponentu, která automaticky nastaví předchozí výběr do výchozího nevybraného stavu, pokud je vybrán jiný sval

10. Komponenta Option Menu:

- Prostřední prvek na levé straně je komponenta Option Menu.
- Po kliknutí se rozbalí možnosti, ze kterých si můžete vybrat vaši preferovanou možnost.
- Základní možností je možnost "Systém", která je nastavena ze základu. Tato možnost je zvolena tak, aby ladila s nastavením systému a přizpůsobila se tématu nastavenému na počítači.
- Dalšími možnostmi jsou možnosti pevného nastavení vašeho preferovaného stylu.

11. Návrat na rozhodovací okno administrátora:

- Tlačítko "HL admin" slouží k návratu na rozhodovací okno administrátora

12. Zadávání hodnot do entry polí:

- V pravé části se nacházejí tři entry pole, která slouží k zadávání jednotlivých potřebných hodnot.
- Mezi těmito entry poli můžete proklikávat pomocí levého tlačítka myši nebo klávesou "Tab".
- Všechny pole disponují validační schopností, která se aktivuje poté, co z označeného pole kliknete jinam.
- Do pole je nutné zadávat pouze kladné celočíselné hodnoty. Veškeré ostatní možnosti jsou chybné, a na tuto skutečnost budete upozorněni pomocí vyskakovacího okna.
- První hodnotou je rychlost, která určuje, kolik kroků se krokový motor bude pohybovat, což je tedy úhlová rychlost krokového motoru. Doporučená hodnota je 10.
- Druhou hodnotou, která se v tomto měření využívá, je hodnota externího tlakoměru. Tato hodnota se zadává ručně. Tuto hodnotu odečtete z displeje na soustavě (viz Obrázek 6.13). Na displeji se ukazují hodnoty ve formátu .XXX. Pro co nejpresnější měření je důležité zadávat celé číslo, ale bez tečky. Například, pokud tlakoměr ukáže hodnotu .023, do pole pro tlak zadáte pouze 23
- Poslední hodnotou jsou kroky, které určují, kolik kroků provede krokový motor kolem své osy. Hodnoty jsou volitelné, ale pro co nejpresnější měření se doporučuje nezadávat příliš velké skoky.

13. Tlačítko "Krok":

- Tlačítko "Krok" slouží k provedení měření.
- Toto tlačítko obsahuje validační funkci, která se aktivuje po stisknutí tlačítka.
- Tlačítko provádí dvě základní funkce:
 - Zapisuje do souboru všechny definované hodnoty.
 - Posouvá vámi nastavenou rychlostí a počtem kroků motor, který otáčí s tlakovým ventilem, a tím zvyšuje tlak.
- Toto tlačítko lze ovládat jak pomocí numerického Enteru, tak Enteru na klávesnici.

14. Tlačítko "Pokračovat":

- Tlačítko "Pokračovat" se aktivuje pouze v případě, že se rozhodnete využít jiný sval než předtím.
- Slouží jako pojistka, abyste potvrdili, že vše je připraveno na měření na jiném svalu.
- Pokud provádíte první měření, potvrdíte tlačítkem "Pokračovat" a poté znovu tlačítkem "Krok".
- Po provedení měření se automaticky smaže hodnota tlaku a vy zadáte pouze aktuální hodnotu a potvrdíte.
- Poté, co se rozhodnete měřit nový sval, zvolíte nový sval, který chcete měřit, a stisknete tlačítko "Krok". Automaticky se spustí snížení tlaku na předchozím měřeném svalu. Poté, co tento tlak klesne na hodnotu technické nuly, se vám zpřístupní tlačítko "Pokračovat". Nyní přepojíte tlakoměr na nový sval, stisknete tlačítko "Pokračovat" a můžete opět provádět měření pomocí tlačítka "Krok".

15. Výběr svalu pro měření:

- Druhou možností na hlavní stránce je výběr jednoho z pěti svalů v okně.
- Všechny tlačítka fungují na stejném principu s jediným rozdílem, že každé pracuje s databází příslušného svalu

16. Uživatelské rozhraní databáze pro sval 1:

- Po kliknutí na tlačítko s názvem "Sval 1" se zobrazí grafické uživatelské rozhraní databáze.
- Zde se zobrazují tři základní vzorce, přičemž se budeme věnovat možnosti "sval1_mv".
- Tato možnost slouží k převodu mezi kroky a milivoly a obráceně.
- Nacházejí se zde tři hlavní entry.
 - Prvním polem je sklon.
 - Druhým polem je posun.
 - Posledním polem je pole, ve kterém si můžete dát nějaký popis, například pro jak těžkou hlavu pacienta je tento vzorec vhodný.

- Pole jsou záměrně volena jako entry pole, aby nebylo potřeba řešit složité nastavení. Stačí přepsat hodnotu, kterou chcete, a zmáčknout tlačítko "Upravit". Tím se tyto hodnoty uloží do databáze a zároveň se změní aktuálně používaný převodní vzorec označený červeným obdélníkem, který se nachází vedle názvu vzorce.
- Tento údaj slouží především k tomu, abyste viděli, jaký je aktuálně využívaný převodní vzorec. Pokud například posledním pacientem byl dospělý muž a nyní na proceduru má být dětský pacient, stačí pouze kliknout na tlačítko "Upravit", a tím se zvolí nový převodní vzorec.
- Pokud chcete zadat nový převodní vzorec, stačí kliknout na tlačítko "Nový" v příslušném sloupci a následně zadat veškeré potřebné informace. Pozor, tento vzorec se ovšem po zmáčknutí tlačítka "Uložit" nenastaví automaticky jako aktuální. To je potřeba udělat znovu tlačítkem "Upravit"

7 Testování

Tato kapitola se zaměřuje na testování celkové funkčnosti soustavy. Nezaměřuje se na jednotlivé části soustavy jako je validace vstupů nebo ochrana pomocí koncových dorazů. Všechny tyto funkcionality byly otestovány a fungují podle předpokladů.

První testem bylo zjistit dolní napěťovou hodnotu krokového motoru. Jak již bylo zmíněno dříve v práci, krokový motor má napětí, které je schopna soustava odečítat pomocí integrovaného voltmetru. Dále bylo zjištěno, pokud se tlakový ventil příliš utáhne je nutné ho pak znovu povolit což je časově náročné, a i nebezpečné, jelikož by mohlo dojít k přílišnému utažení ventilu a tím k jeho zničení. Po testování se určila jako vhodná výchozí pozice pro technickou nulu soustavy na 595–615 mV. Tato hodnota rovněž představuje dolní mez, pod kterou se soustava nemůže dostat. Toto bylo ošetřeno podmínkou v programu.

Následující test, který byl proveden bylo testování maximálního rozsahu silikonových pneumatických svalů. Soustava je stále ve fázi vývoje, a proto se neustále testují jednotlivé druhy silikonů ze kterých se svaly odlévají, proto každý sval má odlišné maximální možnosti natlakování. Z důvodu bezpečné práce se zvolili dané hodnoty.

Sval 3 při měření například byl natlakován na tlak 28 mbar, ovšem při dalších testech tento sval prasknul již při tlaku 19 mbar.

Během měření bylo rovněž zjištěno že na integrovaných voltmetrech značně kolísá napětí. Proto bylo také provedeno měření, které se skládalo ze sta jednotlivých ihned po sobě jdoucích měřeních, které zjistily procentuální kolísání napětí na jednom svalu přes devět procent Tabulka 7.1.

Tabulka 7.1: Shrnující tabulka jednotlivých měření

Column1	Max Hodnota voltmetru	Min Hodnota voltmetru	Max odchylka	Min hod svalu	Max hod svalu	rozsah	Procentuální kolísání
Sval 1	659.5956421	648.5715942	11.0240479	595	730	130	8.480036809
Sval 2	654.0118408	644.2129517	9.79888916	595	760	160	6.124305725
Sval 3	664.0670166	649.7564697	14.3105469	595	750	155	9.540364583
Sval 4	657.2838135	645.8790894	11.4047241	595	770	175	6.708661248

Další testování bylo provedeno při pohybu jednotlivých svalů pomocí tlakových jednotek – mbar, tyto jednotky byly hlavní jednotkou pomocí, které se soustava ovládala, ovšem test byl nastaven tak, aby soustava poskytovala i data v napěťových jednotkách – mV, tato data se následně analyzovala společně s tlakovými jednotkami.

Měření probíhalo tak, že se vždy měřil pouze jeden sval. K tomuto svalu se připojil externí tlakoměr. Tento tlakoměr se ke svalu připojil z důvodu zjištění přesnosti měření. Bylo provedeno 10 náhodných měření kde se buď přidávalo nebo ubíralo 3 a 5 mbar.

Měření má povolenou chybu 1 mbar a 10 mV. Tyto hodnoty byly zvolené z důvodu tabulky 7.1. V této tabulce je minimální odchylka voltmetru na krokovém motoru 10 mV.

Tato měření byla provedena dvě pro sval 1 a pro sval 4.

Hodnoty v zeleném poli jsou hodnoty, které jsou v mezi přípustné odchylky, hodnoty v červeném poli jsou již mimo tuto oblast, a proto se považují za nepřesné posunutí soustavy.

Všechny tabulky s výsledky jsou vytvořené na stejném principu první hodnota je počáteční hodnota, ze které soustava vychází ve druhém sloupci je hodnota, kterou soustava bere jako konečnou a chce se do ní dostat. A poslední hodnota je výsledek, do kterého se soustava dostala po provedení veškerých kroků.

Posun krokových motorů byl zároveň testován pomocí funkce `get_steps_from_start()`. Pomocí této funkce bylo ověřeno, zda krokový motor provedl správný počet kroků. Tato funkce potvrdila při každém měření, že posun krokových motorů byl správný.

Na svalu 1 bylo provedeno 10 měření tabulka 7.2. a na svalu 4 bylo provedeno 6 měření.

Tabulka 7.2 Tabulka přesnosti měření svalu 1

Počáteční hodnota v mbar	Hodnota, která by měla být v mbar	Hodnota, která je v mbar	Počáteční hodnota v mV	Hodnota, která by měla být v mV	Hodnota, která je v mV	Posun v krocích
14	17	17	659	677	682	8
17	14	15	682	665	670	-8
15	12	13	670	652	656	-8
13	10	9	656	638	641	-8
9	12	12	641	659	653	8
12	15	15	653	671	669	8
15	18	18	669	687	689	8
18	13	13	689	661	658	-13
13	8	9	658	629	628	-13
9	14	13	628	656	656	13

Tabulka 7.3 Tabulka přesnosti měření svalu 4

Počáteční hodnota v mbar	Hodnota, která by měla být v mbar	Hodnota, která je v mbar	Počáteční hodnota v mV	Hodnota, která by měla být v mV	Hodnota, která je v mV	Posun v krocích
14	17	17	661	679	677	7
17	20	21	677	695	693	7
21	18	19	693	675	681	-7
19	16	17	681	663	670	-7
17	12	14	670	639	646	-12
14	19	19	646	676	654	12

Jako poslední měření se provedlo měření na dvou předem vytvořených funkcí. První akcí je akce „Laterální flexe“ při této akci se nejdříve svaly 1 a 3 nastaví na hodnotu 700 mV a svaly 2 a 4 nastaví na hodnotu 630 mV. Toto nastavení hodnot se provede 2x. Krok nastavení hodnoty se rozhodlo duplikovat po provedení experimentálním měření, které ukázalo, že lepší časové i přesnosti výsledky má tato metoda. Po první skupině iterací se změnil požadované hodnoty na jednotlivých svaích. Pro svaly 1 a 3 je to 630 mV a pro svaly 2 a 4 to je 700 mV. Tento krok se také dvakrát opakoval a jako poslední krok této akce je nastavení svalů do výchozí pracovní pozice 650 mV, rovněž se 2x opakuje. Výsledky pro jednotlivé svaly jsou v tabulkách 7.4 až 7.7.

Tabulka 7.4 : Tabulka přesnosti měření svalu 1 během akce Laterální flexe

Počáteční hodnota v mV	Hodnota, která by měla být v mV	Hodnota, která je v mV	Posun v krocích
634	650	652	7
652	700	673	21
673	700	691	12
691	630	674	-26
674	630	628	-20
628	650	639	10
639	650	653	5

Tabulka 7.5 Tabulka přesnosti měření svalu 2 během akce Laterální flexe

Počáteční hodnota v mV	Hodnota, která by měla být v mV	Hodnota, která je v mV	Posun v krocích
696	650	649	-28
649	630	639	-12
639	630	629	-6
629	700	646	43
646	700	699	32
699	650	695	-30
695	650	641	-27

Tabulka 7.6 Tabulka přesnosti měření svalu 3 během akce Laterální flexe

Počáteční hodnota v mV	Hodnota, která by měla být v mV	Hodnota, která je v mV	Posun v krocích
648	650	647	0
637	700	664	23
664	700	706	16
706	630	685	-52
685	630	645	-20
645	650	642	8
642	650	647	8

Tabulka 7.7 Tabulka přesnosti měření svalu 4 během akce Laterální flexe

Počáteční hodnota v mV	Hodnota, která by měla být v mV	Hodnota, která je v mV	Posun v krocích
661	650	649	-5
649	630	637	-8
637	630	625	-3
625	700	690	29
690	700	700	4
700	650	654	-20
654	650	642	-2

Poslední měření, které se v rámci testování provádělo byla „Cervikální extenze“. Při tomto měření je první poloha pro svaly 1 a 2 700 mV, pro svaly 3 a 4 630 mV. V druhé poloze se tyto hodnoty změní na pro svaly 1 a 2 na 630 mV a pro svaly 700 mV. Po těchto dvou polohách se sval vrátí zpět do původní polohy. Výsledky pro jednotlivé svaly jsou v tabulkách 7.8 až 7.11.

Tabulka 7.8 Tabulka přesnosti měření svalu 1 během akce Cervikální extenze

Počáteční hodnota v mV	Hodnota, která by měla být v mV	Hodnota, která je v mV	Posun v krocích
642	650	649	4
649	700	669	23
669	700	705	14
705	630	687	-25
687	630	625	-26
625	650	641	11
641	650	645	4

Tabulka 7.9 Tabulka přesnosti měření svalu 2 během akce Cervikální extenze

Počáteční hodnota v mV	Hodnota, která by měla být v mV	Hodnota, která je v mV	Posun v krocích
656	650	652	-3
649	700	651	29
651	700	698	29
698	630	679	-41
679	630	636	-30
636	650	635	8
635	650	641	9

Tabulka 7.10 Tabulka přesnosti měření svalu 3 během akce Cervikální extenze

Počáteční hodnota v mV	Hodnota, která by měla být v mV	Hodnota, která je v mV	Posun v krocích
639	650	633	-2
633	630	636	-2
636	630	635	-3
635	700	655	28
655	700	693	19
693	650	691	-19
691	650	658	-18

Tabulka 7.11 Tabulka přesnosti měření svalu 4 během akce Cervikální extenze

Počáteční hodnota v mV	Hodnota, která by měla být v mV	Hodnota, která je v mV	Posun v krocích
639	650	648	4
648	630	643	-7
643	630	639	-5
639	700	671	23
671	700	693	11
693	650	693	-17
693	650	647	-17

8 Diskuse

Pro tento projekt byl zvolen programovací jazyk Python, což je obvyklá volba pro projekty, kde je potřeba spolupracovat s větším množstvím externích součástí. Pokud by se například zvolil programovací jazyk C# tak programovací část by byla stejně náročná ovšem by se muselo psát mnohem více ovládacích knihoven.

Výběr jednotlivých knihoven byl ovlivněn především tím, zda jsou tyto knihovny zahrnuty v základní distribuci jazyka python. Všechny ostatní knihovny kromě knihoven pro tvorbu grafického uživatelské rozhraní a databázové knihovny nemají jiné možnosti, které by bylo programátory hojně využíváno.

Na druhou stranu databázové knihovny a knihovny pro tvorbu mají velké množství alternativ. Proto u výběru těchto knihoven rozhodovalo především jejich specifikace. U databáze nebylo potřeba mít obrovskou distribuovanou síť tabulek které budou obsahovat i jiné zápisy než textová data. Proto byl zvolen druh databáze, který vytváří pouze lokální soubor a do něho ukládá veškeré zápisy.

Stejný princip rozhodování byl i u knihovny pro tvorbu GUI. Nebylo nutné vytvářet pokročilé GUI, které by mělo pokročilou fyziku. Pro tento důvod se zvolila základní knihovna integrovaná v základní distribuci Pythonu, tato knihovna ovšem nedisponuje moderním vzhledem a z tohoto důvodu se zvolila knihovna která tyto nedostatky neguje a je schopna vytvořit se stejnou jednoduchostí jako knihovna tkinter moderně vypadající GUI.

Soustava disponuje velice přesným voltmetrem, který je schopný měřit s přesností na nano volty. Toto pozitivum ovšem neguje kolísání napětí na krokovém motoru které, může dosahovat i přes devět procent což při konstrukci soustavy která je schopna zvládnout stovky mbar není problém, ovšem silikonové pneumatické svaly využívané v této soustavě ovšem tento rozsah značně zmenšují.

Jak ukázalo měření tyto svaly snížily rozsah na desítky milivoltu a necelých 30 mbar. Tato vlastnost svalu je dosti omezující pro soustavu.

Druhá měření na jednotlivých svalech 1 a 4 ukázaly předchozí nedostatek z kolísání soustavy. Tlakové jednotky, které nejsou tak přesné odpovídají ve většině případů povolené chybě, u milivoltů se dosáhlo stejně pozitivních výsledků. Podle měření se ukázalo že sval 1 byl schopen se dostat pokaždé do správného rozsahu jak v tlakových, tak napěťových jednotkách.

Testovaný sval 4 již zaznamenal během měření chyby ovšem tyto chyby byly zaznamenány vždy pouze v jedné z měřených jednotek jednou to bylo v mbar kdy byl sval mimo požadovanou hodnotu v mbar o 2, přípustná chyba byla nastavena na 1. Ovšem tlaková hodnota se lišila pouze o 7 mV a přípustná chyba byla nastavena na 10.

Další chyba se vyskytla ihned v následujícím měření, kdy naopak tlak pneumatického svalu byl přesně na odpovídající hodnotě 19 mbar ovšem hodnota voltmetru se lišila od požadované hodnoty o 13 mV.

Posledním druhem měření, které bylo testování bylo testování přesnosti akcí, které byly vytvořeny. První akcí bylo Laterální flexe. Během prvotních testů této akce se zjistilo že soustava má problém s posunem o velké množství kroků, kde krokový motor se posunul o správný počet kroků ovšem soustava se dostala pouze na poloviční hodnotu tlaků. Z tohoto důvodu se po konzultaci s vedoucím rozhodlo, že se zkusí tato hodnota nastavit 2x. Toto prodloužilo minimální nutnou dobu, která je nutná pro dosažení přesného tlaku, ovšem zároveň to velice zpřesnilo měření, jako je například v tabulce 7.4. Tlak svalu 1 se nikdy na první pokus nedostal na požadované tlaky, ovšem při druhé iteraci se mu již povedlo nastavit požadovaný tlak. Při stejném měření sval 4 dosáhl výsledků, kde mu na dosažení přesného tlaku stačila pouze jedna iterace.

Druhou měřenou akcí byla Cervikální extenze. Při této akci se rovnou na základě předchozí zvolilo nastavení požadované hodnoty dvakrát po sobě. Opět se jako referenční hodnoty použily svaly 1 a 4 neboť na nich je nejlépe vidět proměnlivost soustavy. Sval 1 byl schopen při iteraci ze 630 mV na 650 mV tento krok provést během jedné iterace soustavy toto ovšem sval 4, který v předchozí akci byl schopen nastavit správné hodnoty již při prvním posunu krokového motoru, nyní toto nedokázal a potřeboval k tomu, aby se dostal na správné kroky 2 iterace soustavy.

Soustava dosahovala velice přesných výsledků, ovšem vyskytly se i chyby nejvíce problematický byl sval 3. U toho svalu během měření vzniklo podezření na poškození, které se bude muset prošetřit techniky z firmy co tuto soustavu navrhla.

9 Závěr

Tato bakalářská práce navazovala již na běžící projekt pneumatické soustavy s 5 na sobě nezávisle ovládanými svaly. Soustava poskytovala dostatek funkcí pro všechny potřebné úkony jako jsou: posunutí krokového motoru, zjištění, v jaké poloze se krokový motor nachází, nebo funkci pomocí které lze odečíst z integrovaného voltmetru napětí.

Projekt byl již veden v programovacím jazyce Python, který se potvrdil jako správná volba, díky svým obsáhlým možnostem knihoven. Především v ohledu grafického uživatelského rozhraní, kde customtkinter poskytl jednoduchost integrované Pythonovské knihovny tkinter a moderní vzhled rozsáhlých frameworků jako je Qt.

Soustava je konstruována velice robustně neboť má i přes svůj velký tlakový rozsah velice přesné pohyby. Tento rozsah je ovšem omezen konstrukcí pneumatických svalů, které nejsou schopné snést tak vysoký tlak, a tudíž soustavu omezují.

Softwarové řešení bylo navrženo s ohledem na to, že soustavu budou obsluhovat zdravotničtí pracovníci, kteří nemusí mít odborné znalosti v technologii nebo počítačích. Z toho důvodu bylo grafické uživatelské rozhraní navrženo tak, aby uživatele případně administrátora vedlo ale především kontrolovalo, jestli do polí zadává správné hodnoty, což je ošetřeno pomocí validačních polí. Další bezpečnostním prvkem soustavy, který pomáhá uživatelům k bezpečnějšímu ovládání je rozsah, ve kterém může operátor pracovat. Pomocí této výše se nemusí uživatel strachovat, že příliš utáhne ventil a tím ho zničí nebo příliš natlakuje sval a tím ho praskne.

Pro uživatele a administrátory, kteří by si nebyli jisti jakoukoliv svojí akcí, tak je pro ně vytvořen podrobný uživatelský manuál, který se věnuje veškerým situacím, do kterých se může operátor dostat.

Zároveň během této bakalářské práce se povedlo splnit veškeré stanovené cíle.

Seznam použité literatury

- [1] Úvod do pneumatiky. Online. In: . S. 4. Dostupné z: https://coptel.cz/pluginfile.php/15746/mod_resource/content/1/tkm-uvod_do_pneumatiky.pdf. [cit. 2024-05-08].
- [2] BORISOV, Bobby. *LXQt 1.0.0 Lightweight Desktop Arrived, Here's How it Looks Like*. Online. Linuxiac. 2021. Dostupné z: <https://linuxiac.com/lxqt-1-0-0/>. [cit. 2024-05-08].
- [3] KOPEČNÝ, Lukáš. *MCKIBBENŮV PNEUMATICKÝ SVAL - MODELOVÁNÍ A POUŽITÍ V HMATOVÉM ROZHRANÍ*. Doktorská práce. Brno: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2009.
- [4] KALITA, Bhaben; LEONESSA, Alexander a DWIVEDY, Santosha K. A Review on the Development of Pneumatic Artificial Muscle Actuators: Force Model and Application. Online. *Actuators*. 2022, roč. 11, č. 10. ISSN 2076-0825. Dostupné z: <https://doi.org/10.3390/act11100288>. [cit. 2024-03-31].
- [5] DEITEL, P. J. a DEITEL, H. M. *Intro to Python: for computer science and data science : learning to program with AI, big data and the cloud*. Deitel. United States of America: Pearson Education, 2020. ISBN 978-0-13-540467-6.
- [6] BANSAL, Rishabh. *Python GUI – tkinter*. Online. GeeksforGeeks. 2023. Dostupné z: <https://www.geeksforgeeks.org/python-gui-tkinter/>. [cit. 2024-03-15].
- [7] *Co je to Open source? Zdroj: https://it-slovník.cz/pojem/open-source/?utm_source=cp&utm_medium=link&utm_campaign=cp*. Online. IT SLOVNÍK. Dostupné z: <https://it-slovník.cz/pojem/open-source>. [cit. 2024-03-15].
- [8] ALI, Francis. *Začínáme s Kivy pro vývoj grafického uživatelského rozhraní*. Online. Python GUIs. 2023. Dostupné z: <https://www.pythonguis.com/tutorials/getting-started-kivy/>. [cit. 2024-03-15].
- [9] *Co je to Framework?* Online. A star search. 2021. Dostupné z: <https://a-starsearch.cz/blog/co-je-to-framework>. [cit. 2024-03-15].
- [10] VIKTORIN, Petr a HRONČOK, Miro. *GUI v Pythonu: PyQt5*. Online. Nauč se Python!. 2017. Dostupné z: <https://naucese.python.cz/lessons/intro/pyqt/>. [cit. 2024-03-15].

- [11] GEON. *Co je to Tkinter?* Online. PY.CZ. 2005. Dostupné z: <https://www.py.cz/Tkinter>. [cit. 2024-03-15].
- [12] SCHIMANSKY, Tom. *CustomTkinter*. Online. Git Hub. 2021. Dostupné z: <https://github.com/TomSchimansky/CustomTkinter/tree/master>. [cit. 2024-03-15].
- [13] *Co je to databáze?* Online. Oracle. Dostupné z: <https://www.oracle.com/cz/database/what-is-database/>. [cit. 2024-03-16].
- [14] MIHALCEA, Vlad. *How does MVCC (Multi-Version Concurrency Control) work*. Online. Vlad Mihalcea. 2022. Dostupné z: <https://vladmihalcea.com/how-does-mvcc-multi-version-concurrency-control-work/>. [cit. 2024-03-16].
- [15] *Rozhraní DB-API 2.0 pro SQLite databáze¶*. Online. Python. Dostupné z: <https://docs.python.org/3/library/sqlite3.html>. [cit. 2024-03-16].
- [16] VASUDEV4. *Modul hashlib v Pythonu*. Online. GeeksforGeeks. 2023. Dostupné z: <https://www.geeksforgeeks.org/hashlib-module-in-python/>. [cit. 2024-03-16].
- [17] *Časový přístup a převody*. Online. Python. Dostupné z: <https://docs.python.org/3/library/time.html>. [cit. 2024-03-16].
- [18] *Openpyxl 3.1.2*. Online. Pypi. Dostupné z: <https://pypi.org/project/openpyxl/>. [cit. 2024-03-16].
- [19] IHRITIK. *Modul Pathlib v Pythonu*. Online. GeeksforGeeks. Dostupné z: <https://www.geeksforgeeks.org/pathlib-module-in-python/>. [cit. 2024-03-16].
- [20] SEOPRAKTICKY.CZ. *Co je to regulární výraz a k čemu se používá*. Online. Collabim. 2023. Dostupné z: <https://www.collabim.cz/akademie/knihovna/co-je-to-regularni-vyraz-a-k-cemu-se-pouziva/#:~:text=Už%20víme%2C%20že%20se%20regulární%20výrazy%20skládají%20ze,3%20závorky%20se%20používají%20pro%20seskupování%20množin%20znaků..> [cit. 2024-03-16].
- [21] *Operace s regulárními výrazy*. Online. Python. Dostupné z: <https://docs.python.org/3/library/re.html>. [cit. 2024-03-16].

Seznam použitých obrázků

Obrázek 2.1 Návrh McKibbenovy biologické protézy pomocí pneumatické svaly	11
Obrázek 2.2 Oplet McKibbenova pneumatického svaly Zdroj: https://web.archive.org/web/20070927065220im_/http://www.shadowrobot.com/images/airmuscle-rubber-tube-braid.jpg	11
Obrázek 2.3 Vyfouknutý pneumatický sval Zdroj: Vlastní	13
Obrázek 2.4 Nafouknutý pneumatický sval Zdroj: Vlastní	13
Obrázek 2.5 Graf popularity programovacích jazyků v roce 2023 Zdroj: https://hitech-us.com/wp-content/uploads/2023/01/image-2.png	14
Obrázek 4.1 GUI vytvořené pomocí knihovny Kivy Zdroj: https://i.stack.imgur.com/y6Hmq.png	20
Obrázek 4.2 GUI vytvořené pomocí knihovny PyQt	21
Obrázek 4.3 GUI vyhotovené pomocí knihovny Tkinter Zdroj: https://static.wixstatic.com/media/90b6f2_3041927f0a11481a995dde96b3e2b01d~mv2.jpg/v1/fill/w_703,h_547,al_c,q_90/90b6f2_3041927f0a11481a995dde96b3e2b01d~mv2.jpg	22
Obrázek 4.4 GUI vyhotovené pomocí knihovny customtkinter Zdroj: https://raw.githubusercontent.com/TomSchimansky/CustomTkinter/master/documentation_images/complex_example_dark_Windows.png	23
Obrázek 5.1 Komponent diagram soustavy této bakalářské práce vyhotoven v jazyce UML. Zdroj: Vlastní	30
Obrázek 5.2 Přihlašovací okno, dark mode Zdroj: Vlastní	32
Obrázek 5.3 Přihlašovací GUI, kde je heslo nahrazeno hvězdičkami, light mode Zdroj: Vlastní	34
Obrázek 5.4 Přihlašovací okno přístup neudělen Zdroj: Vlastní	35
Obrázek 5.5 Úvodní okno administrátorské části, light mode Zdroj: Vlastní	36
Obrázek 5.6 GUI reprezentace databáze pro sval 1 Zdroj: Vlastní	36
Obrázek 5.7 GUI databáze, když administrátor vytváří nový převodní vzorec.	38
Obrázek 5.8 GUI Uživatel Zdroj: Vlastní	39
Obrázek 5.9 GUI uživatel Zdroj: Vlastní	42
Obrázek 5.10 GUI uživatele s hodnotami svalů v mV	47

Seznam použitých tabulek

Tabulka 7.1: Shrnující tabulka jednotlivých měření.....	57
Tabulka 7.2 Tabulka přesnosti měření svalu 1	59
Tabulka 7.3 Tabulka přesnosti měření svalu 4	59
Tabulka 7.4 : Tabulka přesnosti měření svalu 1 během akce Laterální flexe.....	60
Tabulka 7.5 Tabulka přesnosti měření svalu 2 během akce Laterální flexe.....	60
Tabulka 7.6 Tabulka přesnosti měření svalu 3 během akce Laterální flexe.....	60
Tabulka 7.7 Tabulka přesnosti měření svalu 4 během akce Laterální flexe.....	61
Tabulka 7.8 Tabulka přesnosti měření svalu 1 během akce Cervikální extenze	61
Tabulka 7.9 Tabulka přesnosti měření svalu 2 během akce Cervikální extenze	62
Tabulka 7.10 Tabulka přesnosti měření svalu 3 během akce Cervikální extenze ...	62
Tabulka 7.11 Tabulka přesnosti měření svalu 4 během akce Cervikální extenze ...	62

Příloha A: Zdrojový kód

Veškeré zdrojové kódy této bakalářské práce se nacházejí na GitHub. V době řádného termínu odevzdání této bakalářské práce jsou v repozitáři nahrané všechny potřebné soubory a po termínu odevzdání již nedojde k úpravám repozitáře, které by ovlivnily funkčnost soustavy.

Odkaz na GitHub:

https://github.com/IronKnee22/Ovladaci_software_pro_nezavisle_ovladani_peti_p_neumatickych_svalu_v_roboticke_strukture

Příloha B: Seznam příložených souborů

Ovladaci_sw	Zdrojový kód pro ovládání pneumatické soustavy
-------------	--