**Návrh a realizace grafického rozhraní v Pythonu pro řízení a monitorování 5 pneumatických svalů s integrovaným senzorickým datovým sběrem**

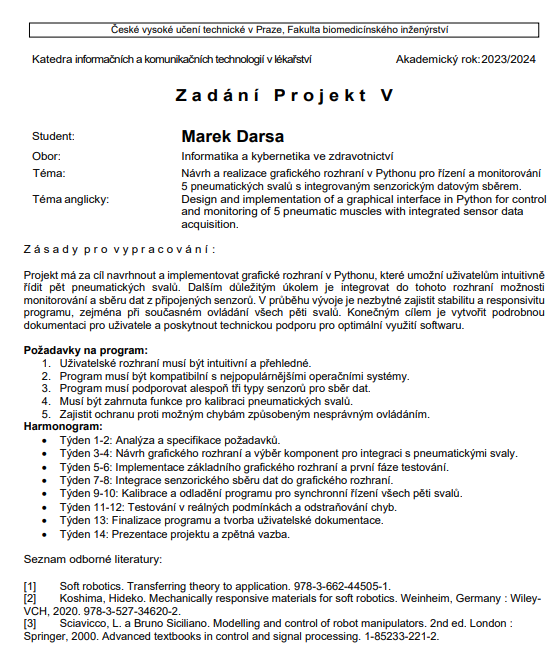
**Design and implementation of a graphical interface in Python for control and monitoring of 5 pneumatic muscles with integrated sensor data acquisition.**

Projekt V

Studijní program: Informatika a kybernetika ve zdravotnictví

Autor práce: Marek Darsa

Vedoucí práce: Ing. Martin Bejtic



Obsah obrázku text, Písmo, snímek obrazovky, dopis

Popis byl vytvořen automaticky

**PROHLÁŠENÍ**

Prohlašuji, že jsem práci s názvem „Vytvoření vhodného grafického uživatelského prostředí a aplikace pro ovládání pneumatických svalů“ vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k práci.

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).

V Kladně 25.06.2023 …...….………...………………...

Marek Darsa

**PODĚKOVÁNÍ**

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

**Návrh a realizace grafického rozhraní v Pythonu pro řízení a monitorování 5 pneumatických svalů s integrovaným senzorickým datovým sběrem**

Návrh a implementace grafického rozhraní v jazyce Python je hlavním cílem tohoto projektu, který se zaměřuje na intuitivní řízení pěti pneumatických svalů. Součástí projektu je také integrace monitorovacích funkcí a sběru dat z připojených senzorů do tohoto uživatelsky přívětivého rozhraní. Během vývoje je klíčové zajistit stabilitu a rychlou odezvu programu, zejména při simultánním ovládání všech pěti svalů. Konečným cílem je vytvořit podrobnou uživatelskou dokumentaci a poskytnout technickou podporu s cílem optimalizovat využití softwaru.

**Klíčová slova**

GUI

Python

Tkinter

Kalibrace

Pneumatický sval

**ABSTRACT**

**Design and implementation of a graphical interface in Python for control and monitoring of 5 pneumatic muscles with integrated sensor data acquisition.**

The project aims to design and implement a graphical interface in Python that allows users to intuitively control five pneumatic muscles. Another crucial objective is to integrate monitoring capabilities and data collection from connected sensors into this interface. Throughout the development, ensuring program stability and responsiveness, especially during simultaneous control of all five muscles, is essential. The ultimate goal is to create detailed user documentation and provide technical support to optimize software utilization.

**Keywords**

GUI

Python

Tkinter

Calibration

Obsah

[Seznam zkratek 8](#_Toc155024675)

[1 Úvod 9](#_Toc155024676)

[1.1 Přehled současného stavu 9](#_Toc155024677)

[1.2 Cíle práce 9](#_Toc155024678)

[1.3 Volba vhodné knihovny 12](#_Toc155024679)

[1.4 Korektura chyby soustavy 17](#_Toc155024680)

[1.5 Pneumatické svaly 18](#_Toc155024681)

[1.5.1 Úvod do Pneumatických Svalů: 18](#_Toc155024682)

[1.5.2 Principy Fungování Pneumatických Svalů: 19](#_Toc155024683)

[1.5.3 Rozdělení Pneumatických Svalů: 20](#_Toc155024684)

[1.5.4 Materiály a Konstrukce Pneumatických Svalů: 21](#_Toc155024685)

[1.5.5 Výhody a Nevýhody Pneumatických Svalů: 22](#_Toc155024686)

[1.5.6 Aplikace v Průmyslu a Robotice: 24](#_Toc155024687)

[1.5.7 Ovládání a Řízení Pneumatických Svalů: 24](#_Toc155024688)

[1.6 Kalibrace 26](#_Toc155024689)

[1.7 GUI pro ovládání pneumatických svalů 31](#_Toc155024690)

[1.8 GUI pro kalibraci 32](#_Toc155024691)

[2 Výsledky 33](#_Toc155024692)

[3 Diskuse 34](#_Toc155024693)

[4 Závěr 35](#_Toc155024694)

[5 Bibliografie 36](#_Toc155024695)

# Seznam zkratek

#### Seznam zkratek

|  |  |
| --- | --- |
| Zkratka | Význam |
| GUI  CMD  Front-end  Back-end  API  HW  SW  PAM  PLC | Graphical user interface (grafické uživatelské rozhraní)  Command Prompt (příkazový řádek)  vývoj grafického uživatelského rozhraní  vývoj firmwaru  Application programming interface  Hardware  Software  Pneumatic artificial muscles  Programmable Logic Controller |

# Úvod

Tato práce navazuje na předchozí projekt číslo 4, kde byl kladen důraz na vytvoření adekvátního grafického rozhraní pro ovládání tlakové soustavy. V rámci této pokračující práce byly implementovány další funkce pro řízení pneumatického systému, přičemž hlavním cílem bylo umožnit řízení všech pěti pneumatických svalů současně.

Jedním z klíčových úkolů této práce bylo vytvoření programu umožňujícího celkové ovládání všech pěti pneumatických svalů synchronizovaně. Značnou část úkolu tvořilo integrování monitorovacích a datových sběrných funkcí pro připojené senzory. Stabilita a responzivita byly zásadními aspekty tohoto programu.

Dále bylo požadováno rozšíření kalibračního softwaru o možnost přímého výběru měřeného svalu přímo v uživatelském rozhraní. Uživatel by měl mít schopnost nastavit počet kroků krokového motoru a jeho rychlost během měření. Tato rozšíření mají za cíl poskytnout uživateli větší kontrolu nad procesem měření a kalibrace, což přispívá k celkové flexibilitě a efektivitě systému.

## Přehled současného stavu

V aktuální fázi vývoje projektu, který probíhá již nějaký čas, je cílem této práce implementace klíčových funkcionalit, které dosud chyběly a jsou nezbytné pro optimální provoz systému. V současné době disponujeme grafickým uživatelským rozhraním (GUI) určeným pouze pro ovládání jednoho pneumatického svalu. Kromě toho máme vytvořený program pro kalibraci, který dokáže zaznamenávat kalibrační data do Excel souboru.

Je však třeba zdůraznit, že v současném stavu projektu existuje omezení v možnostech nastavení svalu přímo v programu. Týká se to i možnosti specifikovat počet kroků krokového motoru a jeho rychlost. Tato omezení znamenají, že uživatel má pouze přímou možnost konfigurace vlastností svalu v samotném programu. V rámci této práce se bude klást důraz na rozšíření těchto možností, což přinese zvýšenou flexibilitu uživatelského rozhraní a umožní efektivnější manipulaci s parametry systému.

## Cíle práce

Cílem této práce je koncipovat a implementovat grafické uživatelské rozhraní v programovacím jazyce Python, jehož hlavním posláním je poskytnout uživatelům intuitivní možnost řízení pěti pneumatických svalů. Kromě toho je klíčovým úkolem integrovat do tohoto rozhraní sofistikované funkce pro monitorování a sběr dat z připojených senzorů, přičemž je nezbytné zajistit vysokou stabilitu a rychlost programu, zejména při současném ovládání všech pěti svalů.

V rámci vývoje bude kladen zvláštní důraz na estetický design rozhraní a jeho uživatelskou přívětivost. Implementace funkcí monitorování a sběru dat vyžaduje sofistikované algoritmy pro efektivní zpracování a prezentaci naměřených hodnot. Pro dosažení optimální stability a responzivity programu bude provedena pečlivá optimalizace kódu a detailní testování, s důrazem na eliminaci potenciálních chyb.

Dále je součástí cílů této práce vytvořit podrobnou uživatelskou dokumentaci, která bude obsahovat komplexní informace o funkcích, ovládání a nastavení programu. Tato dokumentace bude sloužit jako klíčový prvek pro uživatele při správném využívání softwaru a bude doplněna návody a tipy na řešení případných otázek či problémů.

Python  
V současném dynamickém a neustále se vyvíjejícím světě softwarového inženýrství se jedním z nejvýznamnějších aktérů stal programovací jazyk, který kombinuje eleganci, čitelnost a výkonnost. Tímto jazykem není nic menšího než Python. Python, který byl poprvé uveden do světa v druhé polovině devadesátých let dvacátého století svým tvůrcem Guidem van Rossumem, rychle získal na popularitě a stal se klíčovým nástrojem pro širokou škálu aplikací od jednoduchých skriptů po rozsáhlé softwarové projekty.

Srdcem úspěchu Pythonu je jeho jednoduchá a čitelná syntaxe, která dává programátorům možnost psát kód srozumitelný a elegantní způsobem. Tato vlastnost, spolu s dynamickou typovou kontrolou, umožňuje rychlý vývoj a intuitivní psaní kódu, což činí Python ideální volbou pro začátečníky i zkušené vývojáře.

V Pythonu není omezením pouze syntaxe; jeho základní filozofie, nazývaná "Zen of Python," podporuje principy jednoduchosti, jasnosti a čitelnosti. Tato filozofie nejenže usnadňuje psaní kódu, ale také podporuje udržitelnost a spolupráci mezi vývojáři, což vytváří prosperující a přívětivou komunitu.

Nejenže se Python vyniká v jednoduchosti, ale také nabízí rozsáhlou standardní knihovnu, která poskytuje bohaté možnosti pro řešení různorodých problémů. Tato knihovna, spolu s bohatou sadou nástrojů pro správu závislostí, umožňuje vytvářet robustní a komplexní aplikace s minimálním úsilím.

Důležitým aspektem Pythonu je jeho univerzálnost a schopnost být používán ve široké škále odvětví, od webového vývoje po datovou vědu a umělou inteligenci. Jeho flexibilita a adaptabilita z něj činí nástroj, který si našel místo nejen mezi začátečníky a vzdělávacími institucemi, ale také v průmyslovém prostředí a vědeckém výzkumu.

Python je programovací jazyk s řadou vlastností, které přispívají k jeho oblíbenosti a širokému využití. Jednou z klíčových charakteristik je jeho jednoduchá a čitelná syntaxe, což umožňuje psát kód, který je krátký, srozumitelný a snadno udržovatelný. Tato vlastnost usnadňuje práci s Pythonem jak pro začátečníky, tak i pro zkušené vývojáře.

Další významnou vlastností je dynamická typová kontrola, což znamená, že typ proměnné je určen až za běhu programu. To poskytuje větší flexibilitu při programování, ale vyžaduje opatrnost při zacházení s daty. Python také podporuje objektově orientované programování (OOP), což umožňuje modelování dat a funkcí pomocí objektů, díky čemuž jsou kódy modulární a opětovně použitelné.

Jedním z klíčových pilířů Pythonu je jeho rozsáhlá standardní knihovna, která obsahuje mnoho užitečných modulů a balíčků pro různé účely. Tato knihovna usnadňuje práci s různými aspekty vývoje, včetně manipulace se soubory, síťové komunikace, matematických výpočtů a mnoha dalších.

V oblasti vývojových prostředí existuje několik možností pro psaní, spouštění a ladění kódu v Pythonu. IDLE je základní vývojové prostředí dodávané s instalací Pythonu, poskytující jednoduché nástroje pro začátečníky. PyCharm od společnosti JetBrains je komplexní integrované vývojové prostředí s pokročilými funkcemi, které usnadňují vývoj komplexních projektů.

Jupyter Notebook je interaktivní prostředí, které kombinuje kód, text a vizualizace, často používané pro datovou analýzu a vědecký výzkum. Vašem projektu však přispívá především Visual Studio Code (VSCode), lehké a rozšířitelné vývojové prostředí od Microsoftu. VSCode podporujemnoho programovacích jazyků, včetně Pythonu, a nabízí bohaté možnosti rozšíření a konfigurace, což ho činí atraktivní volbou pro vaše vývojové potřeby.

Pokud jde o interprety, existují různé možnosti, ale CPython je obvykle první volbou pro běžné použití. Další implementace, jako Jython pro integraci s platformou Java a IronPython pro integraci s platformou .NET, poskytují alternativy pro specifické potřeby a prostředí. Celkově vzato, Python a jeho ekosystém nástrojů nabízejí flexibilitu a efektivitu při vývojisoftwaru.

## Volba vhodné knihovny

V tomto projektu byl Python vybrán jako hlavní programovací jazyk z důvodu jeho jednoduchosti a širokého spektra použití. I přes to, že Python není tradičně považován za ideální volbu pro tvorbu grafického uživatelského rozhraní (GUI), rozhodli jsme se ho v tomto projektu využít.

Obvykle není Python primárně volbou pro vytváření GUI kvůli nedostatku výborných GUI knihoven. V důsledku toho se ve velkých projektech často používají specializované jazyky pro back-end a front-end. Toto oddělení napomáhá při návrhu multiplatformních aplikací. V případě sdíleného back-endu, například v Pythonu, můžeme napsat kód pouze jednou, což eliminuje potřebu používat odlišné jazyky pro různé platformy, jako například pro mobilní aplikace. Toto řešení se obvykle uplatňuje, pokud je požadována multiplatformnost.

V tomto projektu, který má konkrétní využití pouze na počítači, není nutné rozdělovat aplikaci na back-end a front-end části. Dalším důvodem pro využití front-endu a back-endu je možnost použít vysokorychlostní jazyk pro samotný vykonávací proces a pro okenní rozhraní použít jazyk optimalizovaný pro tento účel. Rychlé jazyky, jako je například C++ nebo C, umožňují dosáhnout vysoké optimalizace programu. Tyto jazyky se od Pythonu liší především tím, že používají kompilátor, který překládá kód přímo do strojového kódu, což šetří výpočetní čas počítače. Na rozdíl od toho Python nejprve překládá do jazyků, jako jsou C++ nebo C, a tento překládaný kód se následně znovu kompiluje do strojového kódu.

V rámci tohoto projektu však není potřeba multiplatformnost, extrémně pokročilé GUI ani vysoká rychlost programu, která by se zde neprojevila. Rychlost jazyka C by byla patrná zejména při použití třídících algoritmů nebo při výpočtu velmi vysokých čísel. Z tohoto důvodu je pro obě části projektu dostatečné využít programovací jazyk Python.

Hlavním důvodem, proč se nepoužívá vždy přístup back-end a front-end, spočívá v existenci API, které slouží jako komunikační prostředek mezi front-endem a back-endem. Tím pádem je nutná existence další komponenty. Python přináší univerzálnost, což je v našem konkrétním případě výhodné v porovnání s jazyky z rodiny C, jako jsou C a C++. Jazyky C mohou působit méně vhodně zejména kvůli omezeným možnostem rychlého vývoje a flexibilitě, kterou nabízí jazyky s vyšší úrovní abstrakce, například Python.

Programování v jazycích C vyžaduje pečlivou správu paměti a nízkoúrovňovou kontrolu, což může zpomalit vývoj a zvýšit náročnost na programátora. Naopak Python nabízí jednodušší syntaxi, dynamickou typovou kontrolu a rozsáhlou standardní knihovnu, což usnadňuje rychlý vývoj a zvyšuje čitelnost kódu. V aplikacích, kde není klíčová maximální úroveň výkonu, ale spíše flexibilita, rychlost vývoje a jednoduchost kódu, je Python často lepší volbou.

Větší abstrakce a automatizace, kterou Python nabízí, mohou přispět k efektivnějšímu vývoji a snížit pravděpodobnost chyb spojených s nízkoúrovňovou správou paměti. Tyto faktory představují klíčové důvody, proč jsme se rozhodli pro použití Pythonu v rámci našich konkrétních vývojových potřeb, zejména s ohledem na integraci do vývojového prostředí Visual Studio Code (VSCode).

**Výběr Grafických Knihoven v Pythonu**

Mezi nejznámější grafické knihovny Pythonu patří PyQt5, Tkinter a Kivy. V rámci tohoto projektu jsme provedli testování všech tří technologií s cílem nalézt a následně použít nejvhodnější technologii.

**PyQt5: Komplexnost a Intuitivnost**

PyQt5 je výkonná knihovna pro vytváření grafických uživatelských rozhraní (GUI) v jazyce Python. Tato knihovna poskytuje spojení mezi Pythonem a Qt frameworkem, což umožňuje vytváření moderních a interaktivních aplikací s profesionálním vzhledem.

PyQt5 obsahuje bohatou sadu nástrojů pro vytváření různorodých GUI aplikací, včetně okenních aplikací, editorů, nástrojů pro vizualizaci dat a dalších. Hlavní komponenty knihovny zahrnují moduly pro práci s grafikou, správu událostí, manipulaci s widgety, a další.

Jedním z hlavních prvků PyQt5 je modul pro práci s widgety, které jsou základními stavebními kameny GUI. PyQt5 poskytuje širokou škálu standardních widgetů, jako jsou tlačítka, textová pole, seznamy, tabulky a mnoho dalších. Těmito widgety lze snadno manipulovat a přizpůsobit jejich chování a vzhled.

Dále, PyQt5 podporuje správu událostí, což umožňuje reagovat na interakce uživatele, jako jsou stisky kláves, kliknutí myší nebo změny hodnot ve formuláři. Tím se zajišťuje dynamická a interaktivní povaha aplikací vytvořených pomocí této knihovny.

Dalším významným prvkem PyQt5 je jeho schopnost pracovat s grafikou a kreslit na plátno. To umožňuje vytvářet vlastní grafické prvky, vykreslovat grafy a diagramy nebo pracovat s obrazovými daty.

Celkově vzato, PyQt5 je silným nástrojem pro vývojáře, kteří chtějí vytvářet moderní a uživatelsky přívětivé aplikace v Pythonu. Jeho integrace s Qt frameworkem poskytuje širokou funkcionalitu a zároveň zachovává jednoduchost používání díky Pythonu.

**Kivy: Komplexní Framework pro GUI**

Kivy představuje významný framework pro vývoj multiplatformních mobilních a desktopových aplikací s grafickým uživatelským rozhraním (GUI) v jazyce Python. Jeho nejvýraznějšími rysy jsou podpora pro více operačních systémů, zaměření na dotyková rozhraní, deklarativní syntaxe pro popis GUI a využívání akcelerace OpenGL pro optimalizaci vizuálních prvků.

Jedním z hlavních přínosů Kivy je jeho schopnost poskytovat konzistentní uživatelské zážitky napříč různými platformami, což je zajištěno jeho multiplatformní podporou. To umožňuje vývojářům nasazovat aplikace na širokou škálu zařízení, včetně mobilních telefonů, tabletů a desktopů.

Zaměření na dotyková rozhraní reflektuje potřebu moderních mobilních aplikací, a Kivy v tomto směru vyniká. Jeho deklarativní syntaxe umožňuje snadnou specifikaci vzhledu a chování uživatelského rozhraní, což usnadňuje vývoj komplexních aplikací.

Využití akcelerace OpenGL přispívá k výkonu a plynulosti grafických prvků vytvořených pomocí Kivy. Tato technologie je klíčová pro efektivní zpracování a vykreslování grafických prvků, což je zásadní pro atraktivní uživatelské rozhraní.

Kromě toho, jako open-source projekt s licencí MIT, Kivy nabízí vývojářům možnost upravit a modifikovat zdrojový kód dle vlastních potřeb. To poskytuje komunitě široký prostor pro inovace a přizpůsobení frameworku podle specifických požadavků.

Kivy vynikajícím nástrojem pro vývoj moderních mobilních aplikací s důrazem na dotyková rozhraní, a to vše s využitím jednoduché a elegantní syntaxe jazyka Python.

**Tkinter**

Tkinter, jakožto standardní knihovna pro tvorbu grafických uživatelských rozhraní (GUI) v jazyce Python, přináší jednoduché a přístupné prostředky pro vytváření interaktivních aplikací. Tato knihovna je integrována s Tk, GUI frameworkem napsaným v jazyce Tcl, a poskytuje nástroje pro konstrukci okenních aplikací a dialogů.

Základní vlastnosti Tkinter zahrnují rozmanitou sadu widgetů, od tlačítek a vstupních polí po seznamy, což umožňuje vytvářet různorodá uživatelská rozhraní. Tkinter podporuje událostní model, což umožňuje efektivní zpracování interakcí uživatele, jako jsou stisky kláves, kliknutí myší nebo změny v hodnotách widgetů.

Pro usnadnění rozmístění widgetů v okně aplikace poskytuje Tkinter geometrické správce, které zjednodušují návrh a organizaci uživatelského rozhraní. Tkinter je k dispozici na všech hlavních platformách, což zajišťuje křížovou platformní kompatibilitu pro aplikace vytvořené s tímto frameworkem.

Navzdory své jednoduchosti a základní sadě funkcí je Tkinter vhodné pro tvorbu jednodušších aplikací a představuje příjemný vstup pro začátečníky v oblasti vývoje GUI v Pythonu. Jeho přehledná syntaxe a snadná použitelnost přispívají k rychlému vývoji aplikací s uživatelským rozhraním bez nutnosti komplexních postupů.

**CustomTkinter**, jako vývojová iniciativa v oblasti Pythonu, představuje sofistikované rozšíření standardní knihovny Tkinter, které zavádí pokročilé možnosti přizpůsobení vzhledu a chování uživatelského rozhraní. Cílem tohoto rozšíření je poskytnout vývojářům nástroje pro manipulaci s widgety na vyšší úrovni abstrakce a umožnit jim vytvářet komplexní uživatelská rozhraní s vysokým stupněm individualizace.

CustomTkinter redefinuje koncept widgetů, rozšiřuje standardní sadu o nové prvky a umožňuje vývojářům vytvářet plně přizpůsobené widgety podle konkrétních potřeb. Tato flexibilita umožňuje vytvářet specifické grafické prvky, přičemž se posouvá nad rámec standardních nabídek Tkinteru.

CustomTkinter přináší do hry pokročilé možnosti pro definici a aplikaci vlastních témat, což umožňuje programátorům důkladně kontrolovat vizuální identitu svých aplikací. Konfigurovatelné barvy, fonty a další stylistické prvky poskytují vývojářům prostředky pro konzistentní a esteticky přizpůsobený design.

CustomTkinter zdokonaluje událostní model pro správu interakcí uživatele. Programátoři mohou definovat vlastní obslužné funkce pro specifické události, což umožňuje detailní kontrolu nad interaktivitou aplikace a programovatelné reakce na různé uživatelské akce.

S rozšířením CustomTkinter je možné vytvářet vlastní dialogová okna a komplexní uživatelská rozhraní, která reflektují specifické požadavky projektu. Tím se vytváří prostředí pro konstrukci aplikací s vysokým stupněm personalizace.

CustomTkinter klade důraz na kvalitní dokumentaci a poskytuje platformu pro komunitní spolupráci, diskuzi a výměnu zkušeností. Tato komunitní podpora zajišťuje, že vývojáři mají přístup k potřebným informacím a mohou efektivně využívat rozšíření CustomTkinter ve svých projektech.

CustomTkinter reprezentuje významný krok vpřed v oblasti vývoje GUI aplikací v Pythonu. Poskytuje pokročilé nástroje pro vytváření sofistikovaných uživatelských rozhraní s důrazem na flexibilitu, individualizaci a programovatelnou interaktivitu. Tím zvyšuje úroveň abstrakce pro vývojáře a posiluje pozici Tkinteru jako klíčového nástroje pro tvorbu uživatelských rozhraní v ekosystému Pythonu.

**Openpyxl**

je knihovna pro práci s Excel soubory ve formátu xlsx. Je napsaná v jazyce Python a umožňuje snadnou manipulaci s daty v Excel tabulkách jako jsou čtení, zápis a úprava dat. Openpyxl lze použít pro vytváření nových Excel souborů nebo pro úpravu existujících souborů. Openpyxl poskytuje širokou škálu funkcí pro práci s Excel soubory, včetně čtení a zápisu buněk, vkládání a mazání řádků a sloupců, vytváření grafů a diagramů, formátování buněk a dalších funkcí. Openpyxl také podporuje funkce Excelu, jako jsou pivot tabulky a filtrování dat. Jedním z nejdůležitějších aspektů knihovny Openpyxl je snadnost použití. Knihovna je dobře zdokumentovaná a poskytuje přehledné rozhraní, které umožňuje rychlé a efektivní zpracování dat. Openpyxl je také open source projekt, což znamená, že je k dispozici zdarma pro použití a lze ho upravit podle vlastních potřeb.

Další výhodou Openpyxl je jeho multiplatformnost. Knihovna je podporována na různých operačních systémech, jako jsou Windows, Linux a macOS. Openpyxl také podporuje různé verze Pythonu, včetně Pythonu 2 a Pythonu 3. Vzhledem k tomu, že Excel soubory jsou stále velmi běžné v oblasti podnikových aplikací, Openpyxl může být užitečnou knihovnou pro mnoho různých aplikací. Například může být použit pro automatické generování reportů a analýz dat, vytváření databází nebo pro automatizaci procesů v různých oblastech, jako jsou finance, marketing, nebo logistika. V současné době je Openpyxl jednou z nejrozšířenějších knihoven pro práci s Excel soubory v jazyce Python. Jeho popularita se neustále zvyšuje díky jeho snadnosti použití, efektivitě a rozsáhlým funkcím.

Další významnou vlastností Openpyxl je podpora čtení a zápisu vzorců a funkcí. To umožňuje uživatelům automatizovat výpočty a manipulace s daty přímo v Excelových souborech, což je velmi užitečné zejména pro rozsáhlé datové analýzy a reporty.

Vývojáři mohou využít také pokročilé funkce Openpyxl, jako jsou například možnosti nastavení ochrany listů, filtrování dat, sledování změn a mnoho dalších. To poskytuje široké možnosti pro tvorbu sofistikovaných a interaktivních Excelových souborů.

## Korektura chyby soustavy

Během měření se vyskytla dost zásadní chyba. Touto chybou bylo natlakování pneumatického svalu na vyšší nežli požadovaný tlak. Tato chyba nastane z důvodu nepřesnosti krokového motoru. Tuto chybu nelze odstranit hardwarově, ale pouze softwarově. Tuto chybu nelze opravit pomocí HW, neboť HW, který se používá je kvalitní a chyba která vzniká je způsobená nedokonalým návrhem krokového motoru. Chyba vzniká při požadavku o nastavení požadovaného tlaku. Senzor, který nám snímá aktuální tlak svalu, vyšle signál po dosažení požadovaného tlaku. Tento signál se zpracuje řídící jednotkou a ta dá pokyn krokovému motoru, aby přestal dodávat stačený vzduch. Ovšem krokový motor není schopen zastavit přesně na požadovaném místě, ale pouze v jeho předvolených polohách. Další chyba vzniká při zpoždění komunikace mezi senzorem a krokovým motorem. Tato chyba se může rovnat i 20%. Tento problém se dá řešit hned dvěma způsoby. Prvním způsobem je soustavě zpětně zmenšit tlak na požadovanou hodnotu. Druhým možným řešením je vypnout soustavu předem a soustava by tak nabyla také požadovaných hodnot. Každá z těchto metod má své výhody i nevýhody. První možnost má výhodu přesnosti. U této možnosti jsme schopni zpětně přesně nastavovat požadovanou hodnotu a toto jsme schopni provádět jak dlouho budeme chtít nebo dokud se nenastaví požadovaný tlak. Hlavní nevýhodou této metody je rychlost. Pokaždé budeme muset čekat než se nastaví tlak, pak nechat program vyhodnotit hodnotu tlaku soustavy a zase počkat, než se nastaví na požadovanou hodnotu. Druhá metoda má výhodu v rychlosti, jelikož program již dopředu z předchozích měření spočítá statisticky chybu a dá pokyn krokovému motoru pro zastavení s předstihem. Hlavní nevýhodou metody je, že se spoléhá na předchozí měření u počítání chyby měření tudíž nebude tak přesná jako metoda 1.

## Pneumatické svaly

### Úvod do Pneumatických Svalů:

Pneumatické svaly představují specifickou třídu pneumatických aktuátorů, které slouží jako základní stavební bloky v oblasti robotiky a mechatroniky. Tato zařízení, známá také jako pneumatické pohony, přetvářejí tlak stlačeného vzduchu na pohybový výkon, což umožňuje rozmanité aplikace v automatizovaných systémech.

**Definice Pneumatických Svalů:**

Pneumatické svaly jsou zařízení poháněná stlačeným vzduchem, která konvertují pneumatickou energii na mechanický pohyb. Jejich konstrukce je obvykle založena na pružných elastických membránách nebo pístech, které reagují na změny tlaku vzduchu a generují odpovídající lineární nebo rotační pohyb.

**Použití v Robotice a Mechatronice:**

Pneumatické svaly nalezly významné uplatnění v robotických a mechatronických systémech díky své jednoduchosti, rychlosti a schopnosti generovat vysoký výkon. Jsou využívány ve strojích a zařízeních, kde je potřeba rychlá a silná mechanická akce, například v montážních linkách, manipulátorech a jiných automatizovaných zařízeních.

**Historie Vývoje Pneumatických Svalů:**

Historie pneumatických svalů sahá do počátků průmyslové revoluce, kdy byly používány jako jednoduché pístové mechanismy pro pohyb různých prvků. Postupem času se technologie pneumatických svalů vyvíjela paralelně s rozvojem pneumatiky a průmyslové automatizace. Moderní pneumatické svaly využívají pokročilé materiály a konstrukční principy, aby splnily nároky současných technologických standardů.

**Technologický Pokrok:**

Vývoj moderních materiálů, jako jsou polymery a kompozity, přispívá k lehkosti, odolnosti a výkonnosti pneumatických svalů. Navíc se výzkum zaměřuje na optimalizaci tvarů a konstrukce, což umožňuje dosažení vyšších úrovní efektivity a přesnosti. Integrace elektroniky do řídicích systémů pneumatických svalů poskytuje možnost sofistikovanějšího a preciznějšího ovládání, čímž rozšiřuje jejich aplikační možnosti v moderní technologické krajině.

### Principy Fungování Pneumatických Svalů:

Pneumatické svaly operují na základě základních principů pneumatiky a pneumatických aktuátorů, které tvoří základní mechanismy pro přeměnu stlačeného vzduchu na mechanický pohyb. V tomto kontextu je také důležité provést srovnání s jinými typy aktuátorů, zejména s elektrickými a hydraulickými, aby byly zdůrazněny výhody a omezení pneumatických systémů.

**Základní Principy Pneumatiky a Pneumatických Aktuátorů:**

- Princip Stlačitelnosti Plynu: Pneumatické svaly využívají schopnost plynu (v tomto případě vzduchu) stlačit se a expanzí, což umožňuje přenos energie.

- Zákon Boyle-Mariott: Změny tlaku a objemu plynu jsou přímo úměrné za předpokladu konstantní teploty. To je klíčovým prvkem při návrhu a řízení pneumatických systémů.

- Elastické Membrány a Písty: V pneumatických svalích jsou často používány elastické membrány nebo písty, které reagují na změny tlaku vzduchu a generují pohyb.

Srovnání s Elektrickými a Hydraulickými Aktuátory:

- Rychlost a Odezva: Pneumatické svaly jsou obvykle rychlejší a mají rychlejší odezvu než hydraulické aktuátory, ale mohou být pomalejší než elektrické.

- Výkon a Hmotnost: Pneumatické svaly jsou schopny generovat vysoký výkon při relativně nízké hmotnosti, což je výhodné pro mobilní aplikace. Elektrické aktuátory jsou efektivní, ale často těžší.

- Přesnost a Jemné Ovládání: Elektrické aktuátory jsou obecně přesnější a umožňují jemnější ovládání než pneumatické. Hydraulické aktuátory se často používají tam, kde je potřeba vysoká přesnost a síla.

- Náklady a Údržba: Pneumatické systémy mají často nižší počáteční náklady a jsou méně náročné na údržbu než hydraulické. Elektrické aktuátory mohou být nákladnější, ale jejich údržba je obvykle jednodušší.

### Rozdělení Pneumatických Svalů:

Pneumatické svaly, jako aktuátory pohybu využívající stlačený vzduch, lze klasifikovat podle dvou základních kritérií: typu pohybu, který generují (lineární versus rotační) a konkrétních variant, které existují v rámci těchto kategorií.

**Lineární vs. Rotační Pneumatické Svaly:**

1. Lineární Pneumatické Svaly:

- Mechanismy generující lineární pohyb jsou často reprezentovány písty nebo válci, které pohybují lineárně.

- Aplikace zahrnují lineární manipulátory, zdvižné mechanismy a brzdy, kde je požadován přímý posun nebo fixace v lineárním směru.

2. Rotační Pneumatické Svaly:

- Tato kategorie zahrnuje mechanismy, které indukují rotační pohyb a mohou být implementovány jako rotační válce nebo pneumatické motory.

- Používají se v otáčecích mechanismech, jako jsou otáčecí stoly, ventily, a další aplikace vyžadující rotaci.

**Příklady Konkrétních Typů Pneumatických Svalů a Jejich Aplikace:**

1. Dvouválcové Pneumatické Svaly:

- Tyto svaly obsahují dva válcové písty, což zvyšuje stabilitu a umožňuje rovnoměrný lineární pohyb.

- V průmyslových robotických systémech jsou využívány například pro zdvíhání a manipulaci s nákladem.

2. Válcové Pneumatické Svaly:

- Standardní lineární svaly využívají válcové písty a nacházejí uplatnění v širokém spektru aplikací, včetně manipulace s materiály a brzdových mechanismů.

3. Rotační Pneumatické Motory:

- Pneumatické motory generující rotační pohyb jsou integrovány do průmyslových ventilů a otáčecích zařízení.

4. Měniče Lineárního na Rotační Pohyb:

- Mechanismy, které transformují lineární pohyb na rotační, jsou využívány například v klapkách, ventilech nebo otáčecích ventilech.

### Materiály a Konstrukce Pneumatických Svalů:

Materiály používané při výrobě pneumatických svalů:

Při konstrukci pneumatických svalů jsou voleny materiály s ohledem na jejich fyzikální vlastnosti, odolnost vůči opotřebení a schopnost efektivní přenos tlaku vzduchu na mechanický pohyb. Výběr materiálů hraje klíčovou roli ve zajištění spolehlivosti a výkonnosti pneumatických systémů.

1. Ocelové slitiny:

- Válcové těla a písty pneumatických svalů jsou často vyrobeny z oceli nebo ocelových slitin. Tyto materiály poskytují dostatečnou pevnost a odolnost vůči tlakovým změnám.

2. Hliníkové slitiny:

- Pro části, kde je důležitá nízká hmotnost, jako jsou těla válců nebo přírub, se používají hliníkové slitiny. Hliník poskytuje dobrou odolnost vůči korozi a nízkou hmotnost, což zlepšuje dynamiku pohybu.

3. Polyuretany a Pryžové Materiály:

- Elastické těsnění a membrány pneumatických svalů jsou často vyrobeny z polyuretanu nebo pružných pryžových materiálů, které jsou odolné vůči opotřebení a mají schopnost těsněného spojení.

4. Nerezová ocel:

- Při provozu v agresivním prostředí, kde hrozí korozivní vlivy, se používá nerezová ocel pro ochranu před oxidací a prodloužení životnosti pneumatických svalů.

Konstrukce a Design s Ohledem na Odolnost, Hmotnost a Efektivitu:

1. Odpory Proti Tlaku:

- Vnitřní komponenty pneumatických svalů musí být navrženy tak, aby odolávaly vysokým tlakům vzduchu bez deformace nebo trvalého poškození.

2. Ochrana Před Opotřebením:

- Pohybové části, jako jsou písty a těsnění, jsou konstruovány s ohledem na minimální opotřebení a tření, což zvyšuje životnost a spolehlivost zařízení.

3. Optimalizace Hmotnosti:

- Použití lehkých materiálů, jako jsou hliníkové slitiny, pomáhá optimalizovat hmotnost pneumatických svalů, což je zvláště důležité v mobilních aplikacích.

4. Úpravy Pro Zvýšení Efektivity:

- Moderní konstrukce zahrnují optimalizace tvarů a proudových průřezů, aby byla dosažena co nejlepší efektivita přenosu tlaku na pohybový výkon.

### Výhody a Nevýhody Pneumatických Svalů:

Pneumatické svaly představují klíčovou součást v oblasti automatizace a průmyslového řízení. Jejich použití přináší několik výhod, ale současně se setkává i s určitými omezeními.

**Výhody Pneumatických Svalů:**

1. Rychlost:

- Pneumatické svaly jsou schopny generovat rychlý pohyb, což je ideální pro aplikace vyžadující dynamické a promptní reakce na změny v prostředí.

2. Jednoduché Ovládání:

- Ovládání pneumatických svalů je často jednoduché a přímé. Mnoho systémů může být řízeno pomocí jednoduchých ventilů a regulátorů tlaku vzduchu.

3. Nízké Náklady:

- V porovnání s některými alternativními pohonnými mechanismy, jako jsou hydraulika nebo elektromechanické systémy, jsou pneumatické svaly ekonomické, což zvyšuje jejich dostupnost pro široké spektrum aplikací.

4. Odolnost v Průmyslovém Prostředí:

- Pneumatické svaly jsou často odolné vůči prachu, vlhkosti a agresivním průmyslovým prostředím, což zvyšuje jejich spolehlivost v náročných podmínkách.

5. Vysoký Výkon na Hmotnostní Poměr:

- Pneumatické svaly mají schopnost generovat vysoký výkon při relativně nízké hmotnosti, což je výhodné zejména v mobilních aplikacích a robotice.

**Nevýhody Pneumatických Svalů:**

1. Možné Úniky Vzduchu:

- Jednou z hlavních nevýhod pneumatických systémů je možnost úniku vzduchu, což může způsobit ztrátu tlaku a ovlivnit výkon systému.

2. Potřeba Kompresoru:

- Pneumatické svaly vyžadují externí zdroj stlačeného vzduchu, což znamená potřebu kompresoru. To může ovlivnit mobilnost a náročnost na infrastrukturu systému.

3. Omezená Přesnost:

- Přesnost pohybu pneumatických svalů je obecně nižší ve srovnání s některými elektromechanickými aktuátory, což může být kritické v aplikacích vyžadujících vysokou přesnost.

4. Omezený Rozsah Silových Aplikací:

- Při srovnání s hydraulickými systémy jsou pneumatické svaly obvykle omezeny ve své schopnosti generovat velké síly.

5. Hluk:

- Pneumatické systémy mohou generovat zvuk v důsledku rychlých pohybů vzduchu a mechanických částí, což může být nežádoucí v určitých aplikacích.

### Aplikace v Průmyslu a Robotice:

**Použití Pneumatických Svalů v Průmyslových Robotických Systémech:**

Pneumatické svaly hrají klíčovou roli v průmyslových robotických systémech, kde jsou využívány pro řadu úkolů, které vyžadují rychlý a spolehlivý pohyb. Jejich schopnost generovat vysoký výkon při relativně nízké hmotnosti a jednoduché ovládání je činí ideální volbou pro specifické aplikace v průmyslu a robotice.

**Příklady Konkrétních Průmyslových Odvětví:**

1. Automobilový Průmysl:

- Pneumatické svaly se používají v montážních linkách a robotických ramenech pro rychlé manipulace s díly během výroby vozidel. Jsou klíčovým prvkem pro zvyšování efektivity a rychlosti výrobních procesů.

2. Potravinářský Průmysl:

- V potravinářském průmyslu jsou pneumatické svaly využívány pro balení, třídění a manipulaci s potravinářskými výrobky. Jejich odolnost vůči vlhkosti a jednoduchá údržba jsou v tomto odvětví klíčovými faktory.

3. Logistika a Manipulace Materiálů:

- V průmyslových systémech pro manipulaci s materiály a logistiku jsou pneumatické svaly využívány pro pohyb konvejorů, zdvižných zařízení a manipulačních robotů.

### Ovládání a Řízení Pneumatických Svalů:

**Metody Ovládání Pneumatických Svalů:**

1. Manuální Ovládání:

- Manuální ovládání je jednoduchou a tradiční metodou, kde operátor pomocí ručních ventilů nebo spínačů manuálně reguluje přítok vzduchu do pneumatických svalů. Tato metoda se často používá v jednoduchých systémech nebo při údržbě.

2. PLC (Programmable Logic Controller):

- V průmyslovém prostředí se pneumatické svaly často ovládají pomocí PLC, což jsou programovatelné logické automaty. PLC umožňují programátorům vytvářet sofistikované řídící algoritmy a scénáře, což zlepšuje automatizaci a kontrolu nad pohybem pneumatických svalů.

3. Počítačové Řízení:

- V moderních průmyslových a robotických aplikacích se často využívá počítačové řízení, kde jsou pohony pneumatických svalů řízeny prostřednictvím počítačových systémů. To umožňuje složité a přesné pohyby s možností vzdáleného monitorování a řízení.

**Srovnání Různých Technik Řízení:**

1. Manuální Ovládání:

- Výhody:

- Jednoduché a přímé ovládání.

- Nižší náklady na zařízení.

- Nevýhody:

- Omezená automatizace.

- Nižší úroveň přesnosti.

- Omezená možnost vzdáleného ovládání.

2. PLC

- Výhody:

- Programovatelnost a flexibilita.

- Možnost komplexního řízení více svalů a senzorů současně.

- Možnost přizpůsobení podle konkrétních potřeb výrobního procesu.

- Nevýhody:

- Vyšší náklady na pořízení a údržbu PLC.

- Vyšší nároky na školení personálu.

3. Počítačové Řízení:

- Výhody:

- Vysoká úroveň přesnosti a opakovatelnosti.

- Schopnost řídit složité pohyby a interakce s dalšími zařízeními.

- Vzdálené řízení a monitorování.

- Nevýhody:

- Vyšší náklady na počítačový hardware a software.

- Potřeba pokročilých znalostí v oblasti automatizace a řízení.

## Kalibrace

Kalibrace je proces, kterým se zajišťuje, že měřicí zařízení nebo systém poskytuje výsledky odpovídající známým standardům. Cílem kalibrace je eliminovat či minimalizovat systematické chyby a zajistit přesnost a spolehlivost měření. Tento proces je klíčový pro různá odvětví, včetně vědy, průmyslu, lékařství a techniky.

Ve svém základním smyslu znamená kalibrace upravit měřicí zařízení tak, aby jeho výstupy byly v souladu s referenčními hodnotami. To je nezbytné, protože mnoho zařízení s časem může ztratit přesnost kvůli opotřebení, změnám prostředí nebo jiným faktorům. Bez pravidelné kalibrace by docházelo k akumulaci nepřesností a nespolehlivosti měření.

Kalibrace může být aplikována na různá zařízení, od jednoduchých teploměrů po složité analytické přístroje. V případě sofistikovanějších technologií, jako jsou laserové měřicí systémy, chemické analyzátory nebo lékařské diagnostické přístroje, je kalibrace klíčovým prvkem zajištění správné funkce a spolehlivých výsledků.

V průběhu kalibrace jsou měřicí zařízení porovnávána s kalibračními standardy, což jsou referenční hodnoty s vysokou přesností. Jakékoli odchylky jsou identifikovány a následně upraveny tak, aby bylo dosaženo co největší přesnosti měření. Výsledkem je kalibrační certifikát, který obsahuje informace o provedené kalibraci, použitých standardních hodnotách a případně opravách provedených během procesu.

kalibrace představuje klíčový krok v udržování přesnosti a spolehlivosti měření, což má důležité důsledky pro kvalitu vědeckých výzkumů, bezpečnost průmyslových procesů a přesnost lékařských diagnostik.

Kalibrace je zásadním prvkem v mnoha odvětvích, protože poskytuje řadu klíčových výhod a zajišťuje správné fungování měřicích zařízení a systémů. Níže jsou uvedeny hlavní důvody proč je kalibrace důležitá:

1. Zajištění přesnosti měření:

- Jedním z hlavních důvodů pro kalibraci je zajištění přesnosti měření. Měřicí zařízení mohou vykazovat odchylky v důsledku opotřebení, vlivu prostředí nebo technologických změn. Kalibrace umožňuje identifikovat a korigovat tyto odchylky, což vede k přesnějším a spolehlivějším výsledkům.

2. Splnění normativních požadavků:

- V mnoha průmyslových odvětvích a oblastech je kalibrace povinná a je stanovena normami a regulacemi. Pravidelná kalibrace měřicích zařízení je nezbytná pro dodržování předepsaných standardů a normativů. Tento krok je klíčový pro získání certifikací a povolení.

3. Minimalizace chyb:

- Kalibrace hraje klíčovou roli při minimalizaci systematických a náhodných chyb v měřicích systémech. Identifikací a opravou odchylek od referenčních hodnot je možné eliminovat systematické chyby, zatímco pravidelná kalibrace pomáhá sledovat a minimalizovat náhodné chyby.

4. Zajištění spolehlivosti výsledků:

- Spolehlivost výsledků měření je klíčovým faktorem v mnoha odvětvích, včetně lékařství, vědy a průmyslu. Pravidelná kalibrace zaručuje konzistentní a spolehlivé výsledky, což má zásadní význam pro kvalitu produktů, bezpečnost procesů a vědecké výzkumy.

6. Výrobní efektivita:

- V průmyslových prostředích hraje kalibrace klíčovou roli v udržování přesných měření a zajištění konzistentních výsledků ve výrobním procesu. To přispívá k efektivitě výroby, minimalizaci odpadu a zvyšování celkové kvality produktů.

3. Typy kalibrace:

Statická kalibrace:

- Definice: Statická kalibrace představuje kalibrační proces, během něhož je měřicí zařízení kalibrováno v ustáleném a neměnném stavu, nereagujícím na vnější podmínky či změny.

- Využití: Statická kalibrace nachází uplatnění zejména u zařízení, která vykazují minimální změny během svého provozu a nepodléhají vlivům pohybu nebo proměnných podmínek.

Dynamická kalibrace:

- Definice: Dynamická kalibrace je proces, při kterém se měřicí zařízení kalibruje za pohybu nebo v reálném čase, zohledňující dynamiku prostředí, ve kterém pracuje.

- Využití: Tento typ kalibrace je klíčový pro zařízení vystavená dynamickým podmínkám, jako jsou vibrace, pohyb nebo změny prostředí, a zajistí tak přesné měření i při těchto proměnných okolnostech.

Interní kalibrace:

- Definice: Interní kalibrace spočívá v kalibraci měřicího zařízení pomocí jeho vlastních interních referenčních standardů nebo funkcí.

- Využití: Interní kalibrace je vhodná pro zařízení s integrovanými standardy, které jsou pravidelně využívány k ověření a nastavení přesnosti měření.

Externí kalibrace:

- Definice: Externí kalibrace zahrnuje využití externích kalibračních standardů nebo specializovaných laboratoří k ověření a nastavení přesnosti měřicího zařízení.

- Využití: Externí kalibrace je nezbytná tam, kde vyžadovaná nezávislá certifikace či vysoké standardy přesnosti, a poskytuje externí referenční body pro porovnání s výstupy měřicího zařízení.

Kalibrace v reálném provozu (On-the-Fly):

- Definice: Tento typ kalibrace probíhá během běžného provozu zařízení, bez jeho přerušení či zastavení.

- Využití: Kalibrace v reálném provozu je praktická tam, kde není možné nebo efektivní zastavit zařízení z důvodu kalibrace, typicky ve výrobních linkách s nepřetržitým chodem.

5. Proces kalibrace:

1. Příprava zařízení:

- Definice: Prvním krokem procesu kalibrace je důkladná příprava měřicího zařízení. To zahrnuje zajištění, že zařízení je ve stabilním stavu, vyčištěné od případných nečistot, a že je dostatečně aklimatizováno na pracovní podmínky.

2. Nastavení kalibračních standardů:

- Definice: Kalibrační standardy, což jsou referenční hodnoty s vysokou přesností, jsou pečlivě vybrány a nastaveny v souladu s normami a specifikacemi. Tyto standardy slouží jako měřítko pro porovnání výstupů měřicího zařízení.

3. Provedení měření:

- Definice: Během této fáze je měřicí zařízení podrobeno sérii kalibračních měření. Naměřené hodnoty jsou porovnány s předem stanovenými kalibračními standardy, a jakékoli odchylky jsou zaznamenány.

4. Opravy a úpravy (pokud jsou nutné):

- Definice: V případě, že měřicí zařízení vykazuje odchylky od kalibračních standardů, jsou provedeny nezbytné opravy a úpravy. To může zahrnovat kalibraci nebo výměnu částí zařízení, aktualizaci softwaru nebo jiné korekce.

6. Záznam a sledování:

- Definice: Všechny kroky kalibrace jsou pečlivě zdokumentovány a zaznamenány. To zahrnuje údaje o přípravě zařízení, nastavení standardů, provedených měřeních, opravách, a výsledcích. Tato dokumentace je důležitá pro sledování historie kalibrace a auditovatelnost procesu.

7. Pravidelnost a opakování:

- Definice: Kalibrace by měla být prováděna pravidelně podle specifikací výrobce, normativních požadavků nebo interních standardů organizace. Pravidelnost kalibrace zajišťuje udržení přesnosti měření v čase a minimalizaci odchylek.

# Metody

## GUI pro ovládání pneumatických svalů

V tomto projektu se pro GUI použil Tkinter spolu s kombinací jeho nástavby CustomTkinter. Během této práce se využily tři knihovny. Nejdůležitější knihovnou je Tkinter. Tato knihovna slouží k celkové funkčnosti programu a využívají se v ní základní prvky, jako je rozvržení gridu a nastavení základních parametrů okenní aplikace. Pro moderní vzhled se použily prvky z knihovny CustomTkinter. Na začátku celého programu bylo potřeba zvolit si barevný režim. Pro tento projekt se používá tmavě modrá barva. V této aplikaci lze měnit mezi světlým a tmavým módem. Pro tuto akci se vybrala komponenta Optionmenu. Tato komponenta je schopna v sobě uchovávat možnosti a vždy ukazuje pouze vybranou a následně rozbalit a zobrazit celý výčet možností. Další komponentou, která se využila byl Radiobutton. Tato komponenta slouží pro výběr z několika možností, pokud chceme, aby byla zvolena pouze jedna možnost. Pro zadání hodnot je zde zvolena komponenta Entry s automaticky mazaným textem, který uživateli oznamuje, co má zadat. Je zde jedno funkční tlačítko, které celý program spouští. Tímto tlačítkem jsem schopni nastavit tlak soustavy. Nad tímto tlačítkem se nachází pole s přesnou hodnotou aktuálního tlaku soustavy/požadovaný tlak. (2)

## GUI pro kalibraci

Grafické uživatelské rozhraní je v tomto projektu dosti podobné již existujícímu GUI, kterým je GUI pro ovládání pneumatických svalů. Podobnost převzalo především z důvodu jednoduchosti pro uživatele, který nebude muset se pokaždé orientovat v jiném grafickém uživatelském prostředí. Tento program také obsahuje Optionmenu, ve kterém si může uživatel vybrat jaký barevný motiv se mu líbí, ovšem jeho hlavní výhodou oproti minulému programu je, že zde byla dodělána funkcionalita, která dokáže rozpoznat jaký motiv má uživatel nastaven defaultně v operačním systému, a ten nastavit. Tato funkcionalita se do projektu přidala především pro pohodlí uživatele, aby nemusel neustále měnit mód. Další komponentou v tomto grafickém uživatelském rozhraní mají Radiobuttony. Ty slouží k výběru konkrétního svalu. V tomto projektu se ovšem využily jako takové bezpečnostní prvky. Pokud si uživatel zvolí jiný sval, přístroj dá varování, že měřený sval se změnil. Dále uloží soubor s naměřenými hodnotami do souboru, a ten uloží na požadované místo v uložišti počítače. To ovšem není jediná akce po uzavření změně svalu. Další funkce je že program nastaví krokový motor do základní polohy a tím umožní jeho opětovné využití. Soubor se taktéž ukládá při zavření aplikace natvrdo. To znamená, pokud uživatel zavře aplikaci křížkem v pravém horním rohu aplikace pouze zavře svoje GUI, ale na pozadí běží dále dokud se nedokončí celý skript. Tento skript po zavření GUI má dvě funkce. První z nich je již zmíněné ukládání aktuálně upravovaného souboru, aby nedošlo ke ztrátě dat. Druhou funkcionalitou je, že pokud zavřeme GUI program automaticky nám vrátí měřený sval do základní polohy, aby byl schopen nové akce bez zkreslení. Posledním souborem komponent je input políčko, které se nazývá v Tkinter entery a tlačítko neboli button. Tyto dvě komponenty jsou spolu spojeny. Do input políčka uživatel zapisuje hodnotu, kterou odečte z tlakoměru a pomocí tlačítka tuto hodnotu potvrdí a program ví, že má tuto hodnotu uložit do souboru. Input políčko disponuje funkcí, která zajišťuje, že pokaždé potvrzení se políčko promaže, aby to nemusel provádět uživatel. Jedna z možností ovládaní, která tu chybí je možnost potvrzování hodnoty pomocí enteru. Tato možnost funguje pouze v čistě Tkinter programu, ale v GUI založeném na CustomTkinter tato funkce bohužel nefunguje i když je zahrnuta v dokumentaci jako funkční.

# Výsledky

Program byl otestován na soustavě a funkcionalita programu byla správná. Práce vytvořit grafické uživatelské rozhraní pro kalibraci pneumatické soustavy o pěti svalech se podařila.

# Diskuse

V tomto projektu se čerpalo již z hotových knihoven, které jsou sice vhodné pro práci s grafickým uživatelským rozhraním, ovšem práce s nimi nebyla stejně snadná jako práce například práce s technologií WPF, kterou disponuje C#. Hlavním rozdílem práce je jednoduchost. I když se python knihovna CustomTkinter tváří jako uživatelsky přívětivá, je velice náročná. Proto souhlasím s názory jiných programátorů. Tyto názory se shodují v malé využitelnosti speciální grafické knihovny přímo pro python, ale spíše doporučují použít speciální jazyk pro GUI. To znamená rozdělit aplikaci na front end a back end. Vytvoření kalibračního programu na rozdíl od vytvoření samotného GUI je mnohem jednoduší a časově méně náročné, jelikož právě k těmto účelů je python navržen.

# Závěr

V tomto projektu bylo stěžejní částí seznámení se s vhodnými knihovnami pro tvorbu uživatelského rozhraní v pythonu. Pro tuto práci se nakonec zvolila knihovna Tkinter, která zachovává jednoduchost, ale již má také modernizovaný vzhled oproti své základní verzi Tkinter. CustomTkinter je na Tkinter pouze nadstavbou. Byli jsme schopni sestrojit funkční uživatelské rozhraní, které umožní uživateli vybrat požadovaný sval, a také nastavit požadovaný tlak na tomto konkrétním svalu. Navrhnout a naprogramovat kalibrační program, který bude schopen nejen načítat data ze soustavy z konkrétního svalu, který si předtím vybereme, ale také je schopen přijímat data od uživatele a všechna tato data dát do společného souboru, který se jmenuje podle používaného svalu. Všechny cíle byly splněny.

# Bibliografie

|  |  |
| --- | --- |
| (1) | Https://docs.python.org/3/library/tkinter.html. *Docs.python*. |
| (2) | SCHIMANSKY, Tom. Https://github.com/TomSchimansky/CustomTkinter. *Github*. 2022. |
| (3) | Pneumatic artificial muscles. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-12-28]. Dostupné z: https://www.wikiwand.com/en/Pneumatic\_artificial\_muscles |
| (4) | *Pneumatic Artificial Muscles* [online]. [cit. 2022-12-28]. Dostupné z: https://softroboticstoolkit.com/book/pneumatic-artificial-muscles |
| (5) | PHILLIPS, Bill, Chris STEWART a Kristin MARSICANO. *Android programming: the Big nerd ranch guide*. Third edition. Atlanta, GA: Big nerd ranch, 2017. ISBN 01-347-0605-6. |
| (6) | M. DOUMIT, A. Http://softroboticstoolkit.com/publications/analytical-modeling-and-experimental-validation-braided-pneumatic-muscle. *Softroboticstoolkit*. 2009. |
| (7) | TONDU, B. Http://softroboticstoolkit.com/publications/modelling-mckibben-artificial-muscle-review. *Softroboticstoolkit*. 2012. |
| (8) | PHD, Micgekke. Https://www.labmanager.com/insights/how-to-make-a-calibration-curve-28411. *Labmanager*. 2022. |

**"Fluent Python: Clear, Concise, and Effective Programming"** - Autor: Luciano Ramalho.

* Knížka nabízí hluboký vhled do pokročilých funkcí Pythonu a jeho idiomů.

**Dám tam dokumentace všech knihoven**