

# Dokumentácia

Univerzálny program na snímanie optických spektier

Projekt z predmetu Tvorba informačných systémov

Sebastian Horňák, Lucia Korbeľová, Martin Purgát, Dávid Smolák

24. novembra 2022

<b>1. Úvod</b>	<b>4</b>
1.1. Účel tohto katalógu požiadaviek	4
1.2. Rozsah využitia systému	4
1.3. Slovník pojmov	4
1.4. Odkazy na dokumenty od zadávateľa	4
1.5. Prehľad nasledujúcich kapitol	4
<b>2. Všeobecný popis</b>	<b>5</b>
2.1. Perspektíva systému	5
2.2. Funkcie systému	5
2.3. Charakteristika používateľa	6
2.4. Všeobecné obmedzenia	6
2.5. Predpoklady a závislosti	6
<b>3. Špecifické požiadavky</b>	<b>6</b>
3.1. Funkčné požiadavky	7
3.1.1. Komunikácia s Lockinom	7
3.1.2. Komunikácia s krokovým motorom	8
3.1.3. Meranie a spracovanie dát	8
3.1.4. Požiadavky na rozhranie	9
3.2. Kvalitatívne požiadavky	9
<b>4. Úvod</b>	<b>10</b>
4.1. Účel dokumentu	10
4.2. Zameranie a rozsah	10
4.3. Prehľad nasledujúcich kapitol	10
<b>5. Návrh systému</b>	<b>10</b>
5.1. Použité technológie	10
5.2. Dátový model perzistentných údajov	11
5.2.1. Formát súborov	11
5.2.2. komunikačných protokolov	12
5.3. Návrh používateľského rozhrania	12
<b>6. Implementácia</b>	<b>16</b>
6.1. UML component diagram	16
6.2. UML class diagram	17
6.3. UML state diagram	18
<b>7. Rozdelenie na moduly</b>	<b>19</b>
7.1. Modul krokový motor	19
7.2. Modul Disperzný prvok	19
7.3. Modul lockin	19
7.4. Modul čítanie, zápis a spracovanie dát	19
7.5. Modul GUI	19

7.6. Interfejsy medzi modulmi	20
<b>8. Plán implementácie</b>	<b>21</b>
<b>9. Testovacie scenáre</b>	<b>22</b>
9.1. Čítanie napätia	22
9.2. Krokový motor jednoduchý posun o 1 krok dopredu	22
9.3. Krokový motor jednoduchý posun o 1 krok dozadu	22
9.4. Krokový motor posun o 201 krokov dopredu	22
9.5. Krokový motor posun o 201 krokov dozadu	22
9.6. Kalibrácia merania 1	23
9.7. Kalibrácia merania 2	23
9.8. Meranie s jednotkovým krokom	23
9.9. Meranie s nie jednotkovým krokom	23
9.10. Automatické vyplnenie formuláru informácií o meraní pri spustení aplikácie	23
9.11. Automatické uloženie vyplnených informácií hlavičky pri spustení merania	24
9.12. Pokus o spustenie merania s chýbajúcimi údajmi vo formulári o meraní	24
9.13. Vytvorenie nového súboru a vypísanie hlavičky do súboru	24
9.14. Automatické pridanie číselného postfixu do mena súboru	24
9.15. Vytvorenie kópie súboru na užívateľom určenom mieste	25
9.16. Vytvorenie kópie súboru na užívateľom určenom mieste pred spustením merania	25
9.17. Vyrátanie vlnovej dĺžky na základe uhla krokového motora	25
9.18. Zápis údajov z merania do súboru	25
9.19. Správne vykreslenie údajov v grafe	25
9.20. Načítanie údajov merania zo staršieho súboru	26
9.21. Pokus o načítanie údajov merania zo staršieho súboru v zlom formáte	26
9.22. Správne vybratie súboru na porovnanie s aktuálnym meraním	26
9.23. Zmena adresára, z ktorého chceme vybrať súbor s porovnávaným meraním	26
9.24. Nahradenie vybraného súboru na porovnanie iným súborom	26
9.25. Zrušenie porovnavania	27
9.26. Zvolenie používaného disperzného prvku	27
9.27. Spustenie merania	27
9.28. Jeden krok merania	27
9.29. Zmena jednotky z Angstrom na uhol	27
9.30. Zmena jednotky z uhol na Angstrom	27
9.31. Inicializovanie polohy motora	28
9.32. Logovanie	28

# 1. Úvod

## 1.1. Účel tohto katalógu požiadaviek

Tento dokument slúži ako súhrn všetkých požiadaviek pre univerzálny systém na snímanie optických spektier pomocou spektrometra, ktorý vznikol ako semestrálny projekt v rámci predmetu Tvorba informačných systémov na Fakulte matematiky, fyziky a informatiky Univerzity Komenského v Bratislave. Tento dokument je založený a spísaný na základe požiadaviek, ktoré boli diskutované a dohodnuté na osobných stretnutiach so zadávateľom. Je určený pre všetky osoby, ktoré sú aktívne zapojené do vývoja, správy a užívania daného systému. Slúži predovšetkým ako záväzná dohoda o funkcionalite systému medzi zadávateľom a vyvojármi systému (tím Merači).

## 1.2. Rozsah využitia systému

Hlavným cieľom je vyvinúť systém, ktorý bude ovládať spektrometer pomocou krokového motora a následne zbierať namerané dáta z prístroja, ukladať ich do súboru a graficky ich vizualizovať formou 2D bodového grafu. Systém zjednoduší prácu s prístrojom a nahradí aktuálny zastaralý spôsob komunikácie so spektrometrom ( poprípadе s viacerými rôznymi spektrometrami) tým, že umožní konfigurovanie jednotlivých parametrov merania.

## 1.3. Slovník pojmov

- TSV - tabulátorom oddelené hodnoty (tab-separated values)
- lockin - skratka pre lock-in amplifier čo je typ zosilňovača, ktorý dokáže získať signál z prostredia, ktoré má veľké množstvo šumu
- disperzný prvok - súčiastka, ktorá rozkladá svetlo na jednotlivé zložky (mriežka alebo hranol)

## 1.4. Odkazy na dokumenty od zadávateľa

Repozitar projektu: <https://github.com/TIS2022-FMFI/spektra>

Všetky externé informácie, dokumenty, manuály a pod. poskytnuté zadávateľom:

- [informácie o mriežkach, ktoré sa budú implementovať do systému \(preložená google sheets verzia\)](#)
- manuál pre picaxe :  
[link 1](#)  
[link 2](#)
- [manuál pre lockin SR510](#)
- [hodnoty citlivosti lockinu](#)

## 1.5. Prehľad nasledujúcich kapitol

V nasledujúcej kapitole bude stručne a všeobecne popísaná funkcionalita systému, jeho obmedzenia, charakteristiku používateľov systému a konkrétne rozhrania

systému s jeho okolím a ich vlastnosti. Následne v tretej kapitole definujeme kompletný zoznam všetkých požiadaviek zadávateľa na systém.

## 2. Všeobecný popis

### 2.1. Perspektíva systému

Aplikácia bude slúžiť študentom alebo učiteľom FMFI na analýzu spektier svetla pomocou spektrometra. Systém umožní komunikáciu a ovládanie spektrometra pomocou krokového motora, ktorý ovláda uhol otočenia mriežky v spektrometri a nahradí tak zastaraný informačný systém, ktorý bol doteraz používaný.

### 2.2. Funkcie systému

Aplikácia bude umožňovať komunikáciu s lockinom a taktiež nakonfigurovať všetky potrebné parametre lockinu pre plánované meranie pomocou spektrometra a následne ho spustiť, prerušiť a zastaviť. Systém takisto obsluhuje krokový motor zabezpečujúci ovládanie uhla otočenia mriežky vo vnútri spektrometra. Systém potom spracováva merané dáta, ktoré sú tiež v reálnom čase zobrazované do 2D bodového grafu a zapisované do súboru.

Existuje tiež možnosť paralelne zobraziť predošlé meranie zo súboru a zároveň vykreslovať aktuálne meranie pre účely porovnania.

Užívateľské prostredie bude rozdelené na viacero celistvých častí- časť monitorujúca aktuálny stav systému, časť vykresľujúca graf, časť zobrazujúca súborový systém pre načítanie dát pre porovnanie s aktuálnym meraním, časť pre nastavenia monochromátora a milivoltmetra (lockinu) a nakoniec časť pre ovládanie a nastavenie motora. Používateľ bude mať tiež možnosť upraviť zobrazenie grafu napr. zväčšovať graf zobrazujúci merané dáta.

Funkciou systému je aj možnosť kalibrovať krokový motor v závislosti od použitej mriežky (disperzný prvok). Kalibrácia sa musí robiť z časti manuálne. Používateľ najskôr pomocou tlačidla v UI posúva motor po atomických krokoch motora na štartovaciu kalibračnú pozíciu, tam odčíta hodnotu na monochromátore a zapíše ju do aplikácie. Potom rovnako posúva motor na cieľovú kalibračnú hodnotu a odčíta hodnotu na monochromátore a zapíše ju do aplikácie. Na základe systému stanoví lineárnu závislosť medzi počtom krokov a prejdenou vzdialenosťou v priestore vlnových dĺžok. Kalibrácia sa uloží v config file a používateľ ju bežne používa pri meraniach. Treba poznamenať, že samotný uhol otočenia krokového motora v jednom atomickom kroku je pevný a predstavuje 1.5 stupňa.

Meranie prebieha nasledovne:

1. vyberieme mriežku
2. nastavíme polohu krokového motor do presnej pozície (vid stupnica monochromátora) - pomocou tlačidiel v UI

3. užívateľ potvrdí, že pre vybranú mriežku je to nastavené v danej polohe a on z toho vie aký posun urobí vo vlnovej dĺžke na jeden krok - to sme nazvali inicializácia
4. pred samotným meraním musíme nastaviť počiatočnú vlnovú dĺžku a konečnú vlnovú dĺžku a po potvrdení začne meranie = ak nie sme na počiatočnej polohe tak sa krokovým motorom presunieme na danú polohu
5. pri samotnom meraní nastáva posun krokového motora dopredu a pri každom posune modul zodpovedný za lockin overí aktuálnu pozíciu krokového motora, ak je pozícia iná ako tá predchádzajúca, modul pre lockin prečíta nameranú hodnotu a krokový motor sa posunie ďalej, pokiaľ nepríde na koniec merania, tam lockinu pošle vopred preddefinovaný signál, že meranie je ukončené.
6. zároveň celé merania sa zapisuje do súboru a simultánne vykresľuje do grafu
7. po ukončení merania sa krokový motor presunie do inicializačnej pozície.

## 2.3. Charakteristika používateľa

Systém bude rozpoznávať iba jeden typ používateľa. Tento bežný používateľ bude mať prístup ku celej funkcionalite aplikácie a bude mať všetky prístupové práva.

## 2.4. Všeobecné obmedzenia

Kvôli obmedzeniam dostupného hardwaru je potrebné pri vývoji myslieť na to, že nie každý lockin je programovateľný, a preto je nutné v prípade potreby umožniť robiť zmeny ako nastavenie citlivosti alebo zadanie potrebných parametrov prístroja manuálne. Tieto nastavené hodnoty sa následne zas manuálne zadajú v aplikácii a budú použité pri interpretácii nameraných hodnôt. Systém taktiež nedisponuje žiadnym senzorom podľa, ktorého by vedel pri zapnutí určiť polohu krokového motora čiže treba inicializovať krokový motor. To znamená zadať jeho počiatočnú polohu manuálne.

## 2.5. Predpoklady a závislosti

Predpokladá sa, že systém sa bude spúšťať na zariadeniach s operačným systémom Windows 10. Taktiež na komunikáciu s lockinom a ovládanie krokového motora je potrebné, aby malo zariadenie minimálne 2 USB porty.

# 3. Špecifické požiadavky

Predstavujú konkrétne požiadavky zadávateľa na výsledný produkt, rozdelené do viacerých kategórií/podkategórií. Ak sa pri požiadavke vyskytne v zátvorka "voliteľné" znamená, že daná požiadavka nepredstavuje nutnú súčasť výsledného produktu.

## 3.1. Funkčné požiadavky

Poskytujú informácie o nárokoch zadávateľa projektu na funkcionality z pohľadu komunikácie s lockinom a krokovým motorom a z pohľadu samotného programu.

### 3.1.1. Komunikácia s Lockinom

- 3.1.1.1. Komunikácia s lockinom bude prebiehať cez usb-serial convertor (napr. COM16).
- 3.1.1.2. Používateľ bude mať možnosť nastaviť parametre lockinu (ak to zariadenie umožňuje): uhol (int), rozsah (int), referencia (int), časová konštanta (int). Tieto parametre sú platné pre typ milivoltmetra SR510.
- 3.1.1.3. Aplikácia bude po pripojení lockinu automaticky načítavať nastavené parametre lockinu (ak to dané zariadenie umožňuje).
- 3.1.1.4. Aplikácia bude podporovať typy mriežky M456039, M465645 a M455931.
- 3.1.1.5. Používateľ bude mať možnosť si pri každom meraní (aj merania za sebou) nastaviť nové parametre merania, alebo ponechať posledné nastavenia.
- 3.1.1.6. Nastavenie citlivosti sa automaticky mení na základe aktuálnej sily signálu. Ak je sila signálu menšia ako 10% nastavenej citlivosti tak sa citlivosť zvýši ak je väčšia ako 85% tak sa citlivosť zníži. (naraz o jeden krok podľa tabuľky na str.17 SR510 manuál) Zároveň citlivosť automaticky nepôjde pod 20 $\mu$ V.
- 3.1.1.7. Aplikácia bude podporovať komunikáciu s lockinom SR510, preto aplikácia bude môcť čítať aktuálne parametre zariadenia a prijímať namerané hodnoty v reálnom čase.
- 3.1.1.8. Používateľ nebude mať možnosť nastaviť komunikačné parametre s lockinom (baud rate, stop bits, data bits...) v aplikácii, budú pevne určené v kóde aplikácie.
- 3.1.1.9. Aplikácia pripraví užívateľské rozhranie na budúce jednoduché pridanie ďalšieho lockinu. Potrebne časti používateľského rozhrania sa pri jeho zvolení zmenia tak, aby sa odstránili nie relevantné časti rozhrania a uvoľnil sa priestor na komponenty, ktoré bude potrebné v budúcnosti doprogramovať. **(voliteľné)**.
- 3.1.1.10. Aplikácia umožní používateľovi vybrať typ lockinu zo zoznamu podporovaných lockinov.
- 3.1.1.11. V užívateľskom rozhraní bude možnosť výberu lockinu, ktorý nebudeme implementovať, avšak uľahčíme tým jeho implementovanie v budúcnosti **(voliteľné)**.

### 3.1.2. Komunikácia s krokovým motorom

- 3.1.2.1. Používateľ má možnosť ovládať krokový motor pomocou mikrokontroléru PICAXE cez sériový port napojený na usb-serial convertor a nastavenia komunikácie budú pevne uložené v kóde aplikácie.
- 3.1.2.2. Používateľ má možnosť posúvať krokový motor oboma smermi (dopredu - meranie, dozadu - kalibrácia na začiatok). Veľkosť jedného kroku motoru sa zistí pri kalibrácii tak, že pomocou aplikácie sa určí počet krokov na prejde a z displeja monochromátoru používateľ odčíta aktuálny uhol, ktorý manuálne zadá do programu. Podiel rozdielu počiatočného a konečného uhla s počtom prejdých krokov predstavuje veľkosť jedného kroku.
- 3.1.2.3. Možnosť inicializovať krokový motor (viac info v kap. 2.2, 2.4)
- 3.1.2.4. Krokový motor bude spomaľovať a zrýchľovať pri premiestňovaní sa do začiatkovej polohy merania na základe vzdialenosti od cieľovej polohy (**voliteľné**).

### 3.1.3. Meranie a spracovanie dát

- 3.1.3.1. Používateľ vie odštartovať automatické meranie dát a tiež od ktorého uhla meranie začína a v akom uhle otočenia mriežky meranie končí. Tieto údaje užívateľ zadá v užívateľskom rozhraní.
- 3.1.3.2. Aplikácia v reálnom čase číta namerané dáta prichádzajúce z lockinu (milivoltmetra) - aktuálnu nameranú hodnotu a z PIXCAX-e - poloha motora, uhol.
- 3.1.3.3. Aplikácia v reálnom čase vykresľuje namerané dáta do grafu. (údaje v grafe zostanú aj po skončení merania, kým sa nespustí nové meranie alebo sa nezruší celá aplikácia).
- 3.1.3.4. Aplikácia v reálnom čase ukladá namerané dáta do súboru.
- 3.1.3.5. Namerané dáta sa do súboru ukladajú v nasledujúcej štruktúre - informatívna hlavička určená zadávateľom (viď. prílohy) a oddeľovačom (tabulátor) oddelené namerané hodnoty.



- 3.1.3.6. Používateľ bude môcť vyplniť formulár dodatočných údajov merania, ktoré sa uložia v hlavičke súboru s údajmi nameranými z merania. Tieto údaje sú: typ milivoltmetra, typ disperzného prvku, typ zdroja svetla, počiatočná a konečná pozícia merania, nastavenia monochromátoru.
- 3.1.3.7. Používateľ má možnosť prerušiť meranie, následne zmeniť parametre lockinu a potom pokračovať v meraní.
- 3.1.3.8. Po obnovení merania používateľom sa merané dáta ukladajú do nového súboru. Jeho meno sa odvodí od pôvodného súboru pridaním čísla na koniec mena súboru (v prípade potreby to budú postupne čísla 1, 2, 3, ...).
- 3.1.3.9. Používateľ má možnosť načítať dáta z iného merania zo zvoleného súboru, ktoré sa dokreslia do aktuálneho grafu zobrazujúceho aktuálne meranie. Taktiež sa v GUI zobrazí aj legenda porovnávaného merania.
- 3.1.3.10. Legenda k aktuálnemu meraniu je vytvorená z nastavení merania zadávaných užívateľom v aplikácii alebo poprípade pri spustení systému z automaticky načítaných nastavení merania z posledného spusteného merania.

### 3.1.4. Požiadavky na rozhranie

- 3.1.4.1. Používateľské rozhranie bude obsahovať 2D graf zobrazujúci merania v reálnom čase.
- 3.1.4.2. UI obsahuje sekciu s informáciami o aktuálnom stave aplikácie a zobrazuje informácie o systéme v reálnom čase.
- 3.1.4.3. Sekcia grafu bude mať možnosti na prispôsobenie zobrazenia grafu (zväčšenie na celú obrazovku).  
**(voliteľné)**

## 3.2. Kvalitatívne požiadavky

Poskytujú informácie o nárokoch zadávateľa projektu na atribúty kvality, ktoré má systém nadobúdať poprípade iné požiadavky nesúvisiace s funkcionalitou systému.

- 3.2.1. Práca s aplikáciou bude intuitívna a jednoduchá na pochopenie pre niekoho so základným porozumením problematike, ktorú aplikácia rieši.

- 3.2.2. Systém bude navrhnutý natoľko flexibilne, aby bolo v budúcnosti v prípade potreby umožnené nenáročne pridanie nových typov periférií (spektrometrov, mriežok a pod.).
- 3.2.3. Aplikácia nebude disponovať žiadnou bezpečnostnou vrstvou navyše (chýba ochrana pred neautorizovaným prístupom, malware a pod.), keďže nebude používaná na prácu s citlivými údajmi.
- 3.2.4. Lokalizácia aplikácie bude slovenský jazyk.
- 3.2.5. Aplikácia bude podporovaná na zariadeniach s operačným systémom Windows 10.

## 4. Úvod

### 4.1. Účel dokumentu

Tento dokument slúži ako kompletný návrh systému a je určený pre vývojárov systému. Obsahuje všetky informácie potrebné pre pochopenie funkcionality a jej implementácie. Návrh je vytvorený s ohľadom na všetky požiadavky uvedené v katalógu požiadaviek.

### 4.2. Zameranie a rozsah

Pre prácu s týmto dokumentom sa predpokladá predošlá znalosť dokumentu Katalóg požiadaviek a všetkých jeho bodov. Tento dokument poskytuje podrobný návrh implementácie všetkých bodov uvedených v katalógu požiadaviek. Sú v ňom uvedené všetky technológie využité v systéme, popísané všetky rozhrania systému s externými systémami a súbormi. Okrem iného obsahuje tiež UML diagramy - state, component a triedny diagram aplikácie. Ďalej obsahuje kompletný návrh užívateľského rozhrania.

### 4.3. Prehľad nasledujúcich kapitol

V nasledujúcich kapitolách sa budeme venovať kompletnému návrhu systému, opísaného vizuálne pomocou diagramov, obrázkov aj slovne.

## 5. Návrh systému

### 5.1. Použité technológie

- Python - technológia, na základoch, ktorej bude celý systém vytvorený.
- Pyside - framework používaný na vytváranie grafického rozhrania pythonovských aplikácií - na tejto technológií bude postavené celé užívateľské rozhranie.

- PyQtGraph - je to pythonovská knižnica určená na tvorbu grafov v používateľskom prostredí - bude využitá na vytvorenie grafu s výsledkami merania.
- Picaxe - mikrokontrolér pomocou, ktorého budeme posielat príkazy krokovému motoru.
- Visual Basic - objektovo orientovaný programovací jazyk, pre potreby komunikácie s krokovým motorom

## 5.2. Dátový model perzistentných údajov

### 5.2.1. Formát súborov

Súbory, v ktorých sa budú ukladať dáta z jednotlivých meraní sa budú skladať z dvoch logických častí. Z legendy/hlavičky merania, v ktorej budú uložené všetky dodatočné informácie o meraní a jeho parametroch a v druhej časti budú namerané dáta.

Hlavička sa skladá z týchto častí:

- vzorka
  - názov
  - poznámka k technológií
  - popis vzorky (hrúbka)
  - meranie vzorky
  - teplota
- disperzný element
  - mriežka
  - názov konkrétnej mriežky
- monochromator
  - začiatok a koniec vstupnej štrbiny
  - začiatok a koniec výstupnej štrbiny
  - optický filter
- detektor
  - informacia o detektore PMT  
PMT (špecifikovať) + napätie
- budiace svetlo
  - informacia o jednom z nasledujúcich 2 druhov svetla podľa toho, ktorý sa vyberie:  
laser + názov  
lampa + názov
- meranie
  - dátum a čas merania
  - počiatočná vlnová dĺžka [ $\text{\AA}$ ] / počiatočný uhol [ $^\circ$ ]
  - koncová vlnová dĺžka [ $\text{\AA}$ ] / koncový uhol [ $^\circ$ ]
  - krok motora [v pulzoch]
  - počet integrácií
  - korekcia [ $\text{\AA}$ ]
- použitý milivoltmeter
  - typ lockinu

- informácia o tom, či sa citlivosť prispôsobuje automaticky (AUTO) alebo manuálne
- referencia [Hz] (súvisí s prerušovačom / modulatorom )
- rozsah
- fázový posun
- časová konštanta

Väčšina týchto dát sa získa z užívateľského prostredia od používateľa. Niektoré parametre lockinu sa prečítajú priamo z lockinu.

Zvyšok súboru bude obsahovať namerané dáta z prístroja. Každý riadok bude obsahovať tri údaje súvisiace s jedným konkrétnym meraním zaznamenaným počas celého procesu merania. Tieto tri údaje budú oddelené tabulátorom a budú vyzeráť nasledovne:

uhol Alfa [°]                      vlnová dĺžka [Å]                      intenzita[mv]

Kde intenzita je údaj získaný z lockinu. Alfa a vlnová dĺžka je údaj, ktorý vieme pomocou kalibrácie a z údajov o konkrétnej mriežke.

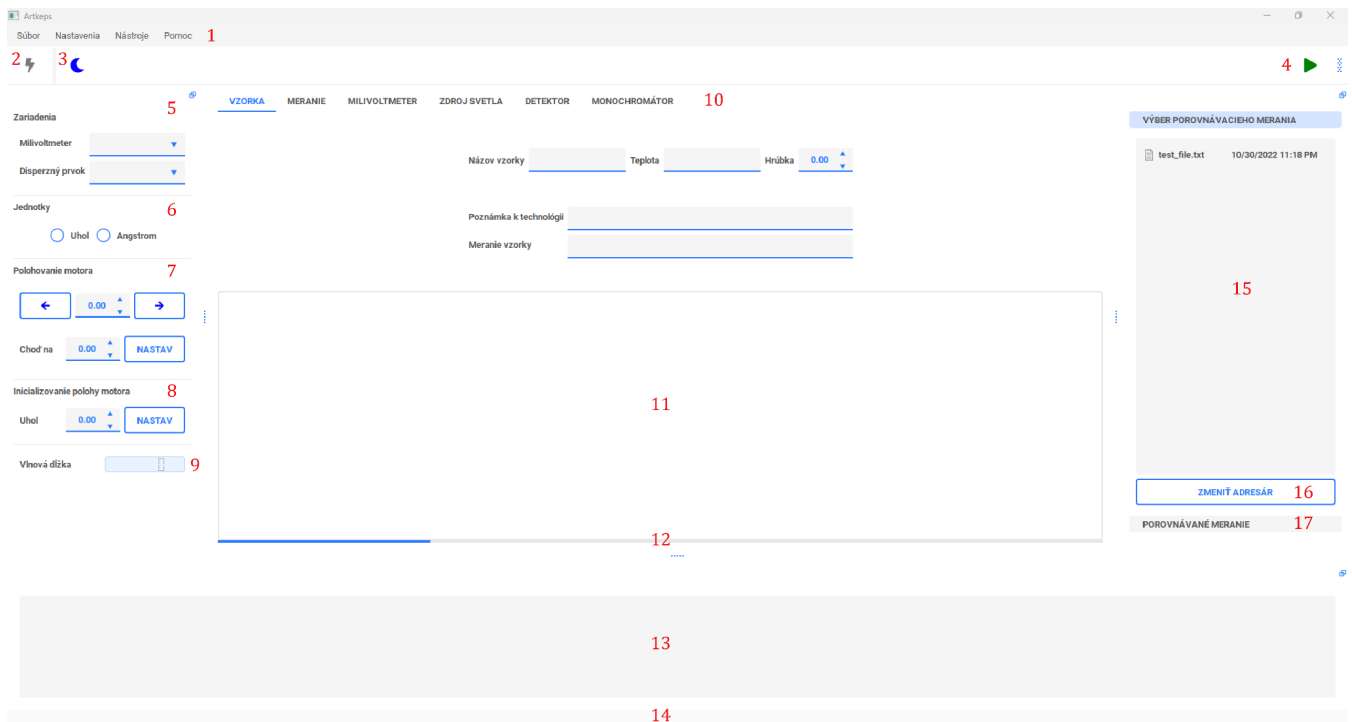
Tieto riadky priebežne pribúdajú počas toho ako prebieha meranie.

Meno súboru je taktiež údaj získaný od užívateľa v používateľskom prostredí.

#### 5.2.2. komunikačných protokolov

- aplikácia komunikuje s krokovým motorom cez mikrokontrolér PICAXE pripojená k PC cez sériový port COM16 s redukciou na USB-A konektor, komunikácia prebieha cez custom skript napísaný vo VISUAL BASIC
- krokový motor - komunikačný protokol:
  - reťazec vo formáte "[0-9]4[F|R]"
  - R - značí posun dozadu (Reverse)
  - F - značí posun dopredu (Forward)
  - Príklady komunikácie:
    - príklad 1: "0100F" značí 100 krokov dopredu
    - príklad 2: "0001R" značí 1 krok dozadu
- aplikácia komunikuje s lockinom SR510 cez sériový port
  - posiela sa reťazec v tvare "[A-Z] " + povinné alebo nepovinné parametre
  - popis všetkých príkazov je dostupný v manuáli SR510 na strane 17 ([manuál pre lockin SR510](#))

### 5.3. Návrh používateľského rozhrania



1. **Hlavné menu** aplikácie- poskytuje možnosti pre otvorenie porovnávacieho merania, ukončenie aplikácie, spustenie kalibrácie, zmenu aktuálneho režimu zobrazenia, zobrazenie dokumentácie, zobrazenie informácií o projekte
2. Indikátor **pripojenia milivoltmetra** - sivý pre odpojený milivoltmeter a žltý pre pripojený
3. Tlačidlo pre **prepnutie režimu zobrazenia** z denného na nočný
4. Tlačidlo pre **spustenie merania**
5. Menu pre **výber milivoltmetra a disperzného prvku**, ktorý má byť použitý pri nasledujúcom meraní
6. **Výber jednotiek** v akých sa má zobrazovať aktuálne meranie
7. Sekcia pre **ovládanie polohy motora**
8. **Potvrdenie aktuálnej polohy motora** užívateľom
9. **Aktuálna vlnová dĺžka** registrovaná milivoltmetrom
10. Menu pre **výber aktuálnych nastavení**, ktoré majú byť použité pri meraní
11. **Graf** zobrazujúci hodnoty aktuálneho a porovnávacieho merania
12. Zobrazenie **zostávajúceho počtu krokov** pre aktuálne spustené meranie (vyobrazená hodnota je iba ilustračná)
13. Zobrazenie **aktuálnych informácií o činnosti** aplikácie
14. Informačný panel pre zobrazenie **aktuálnych zmien stavu aplikácie**.
15. Zobrazenie **adresára pre výber porovnávacieho merania**
16. Tlačidlo pre **zmenu aktuálneho adresára** pre výber porovnávacieho merania
17. Tlačidlo pre **zobrazenie legendy** aktuálne porovnávaného merania

### 5.3.1. Karta pre nastavenia vzorky merania

VZORKA MERANIE MILIVOLTMETER ZDROJ SVETLA DETEKTOR MONOCHROMÁTOR

Názov vzorky

Teplota

Hrúbka 0.00

Poznámka k technológii

Meranie vzorky

### 5.3.2. Karta pre nastavenia merania

VZORKA MERANIE MILIVOLTMETER ZDROJ SVETLA DETEKTOR MONOCHROMÁTOR

Názov súboru

Počet integrácií 0

Korekcia 0.0000

Začiatok 0.00 Koniec 0.00

Krok motora 0

- 5.3.2.1. **Názov súboru**- označuje meno súboru do ktorého sa budú namerané hodnoty ukladať počas nasledujúceho merania
- 5.3.2.2. **Začiatok a koniec**- označuje počiatočnú a koncovú polohu motora pri nasledujúcom meraní
- 5.3.2.3. **Počet integrácií**- údaj použitý pri spracovaní signálu z milivoltmetra
- 5.3.2.4. **Korekcia**- údaj použitý pri spracovaní signálu z milivoltmetra
- 5.3.2.5. **Krok motora**- udáva krok motora použitý pri nasledujúcom meraní

### 5.3.3. Karta pre nastavenia vzorky merania

VZORKA MERANIE MILIVOLTMETER ZDROJ SVETLA DETEKTOR MONOCHROMÁTOR

Ref 135.60

Časová konštanta- pre 0.30

Časová konštanta- post 0.00

Rozsah 99.99 ☒ Auto

Fázový posun 42

*Pozn.* Tieto údaje nie sú relevantné pre milivoltmeter implementovaný v tomto projekte, keďže pre model SR510 je možné tieto informácie získať pomocou API poskytovanej týmto milivoltmetrom.

### 5.3.4. Karta pre nastavenia vzorky merania

VZORKA MERANIE MILIVOLTMETER ZDROJ SVETLA DETEKTOR MONOCHROMÁTOR

Typ ▼ Poznámka

### 5.3.5. Karta pre nastavenia vzorky merania

VZORKA MERANIE MILIVOLTMETER ZDROJ SVETLA DETEKTOR MONOCHROMÁTOR

PMT

Napätie

### 5.3.6. Karta pre nastavenia vzorky merania

VZORKA MERANIE MILIVOLTMETER ZDROJ SVETLA DETEKTOR MONOCHROMÁTOR

Optický filter

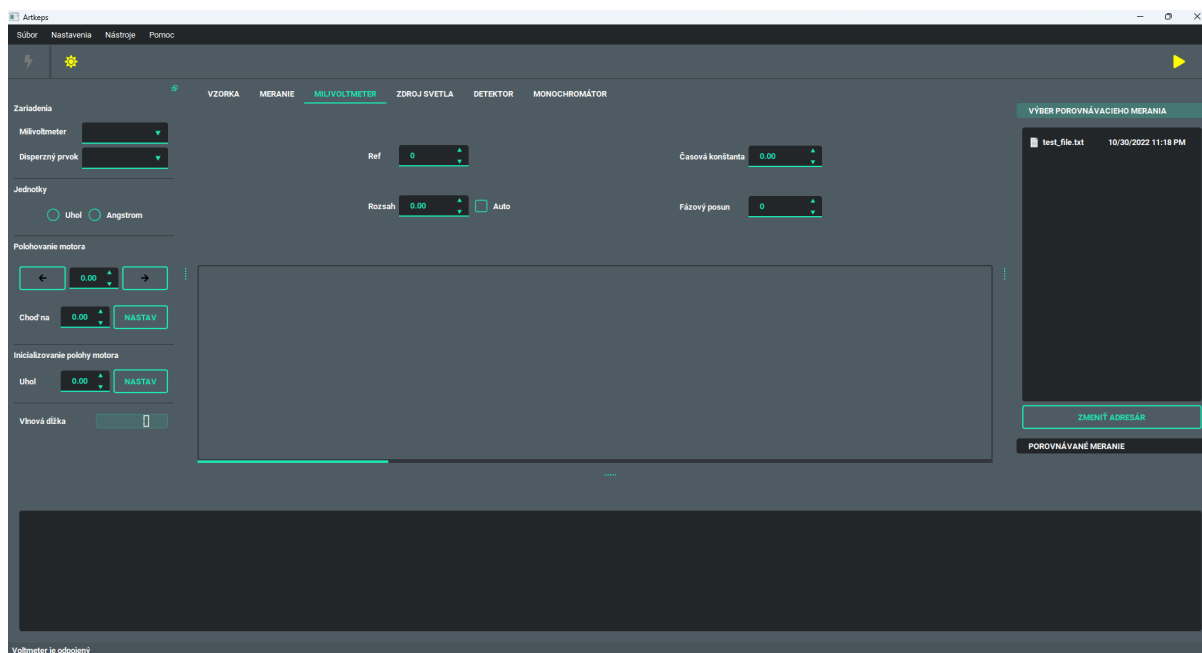
Vstupná štrbina

Začiatok 0.00  
Koniec 0.00

Výstupná štrbina

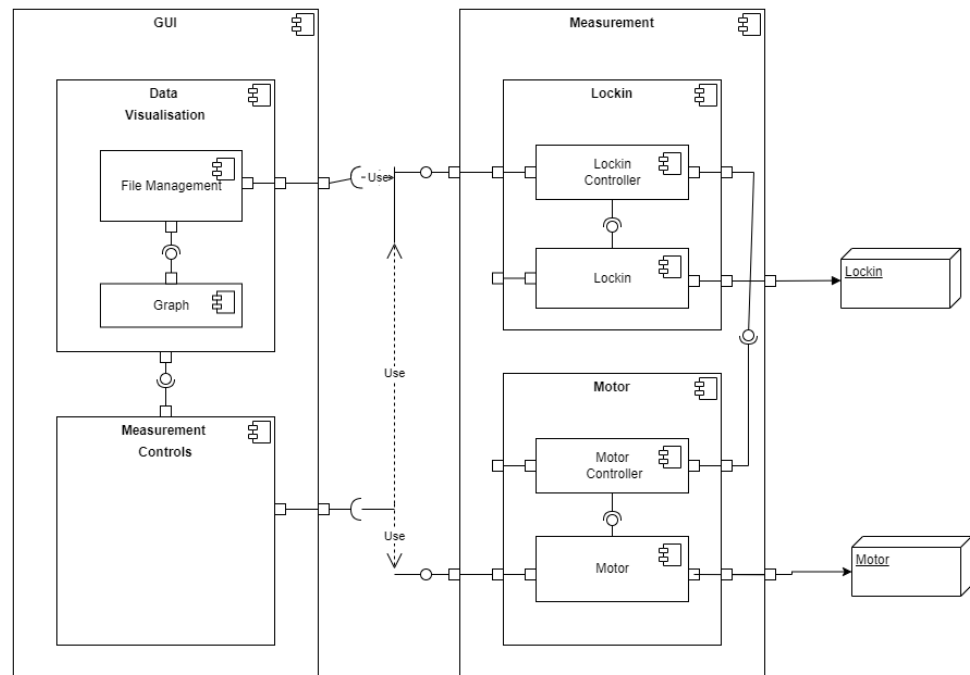
Začiatok 0.00  
Koniec 0.00

### 5.3.7. Zobrazenie v nočnom režime



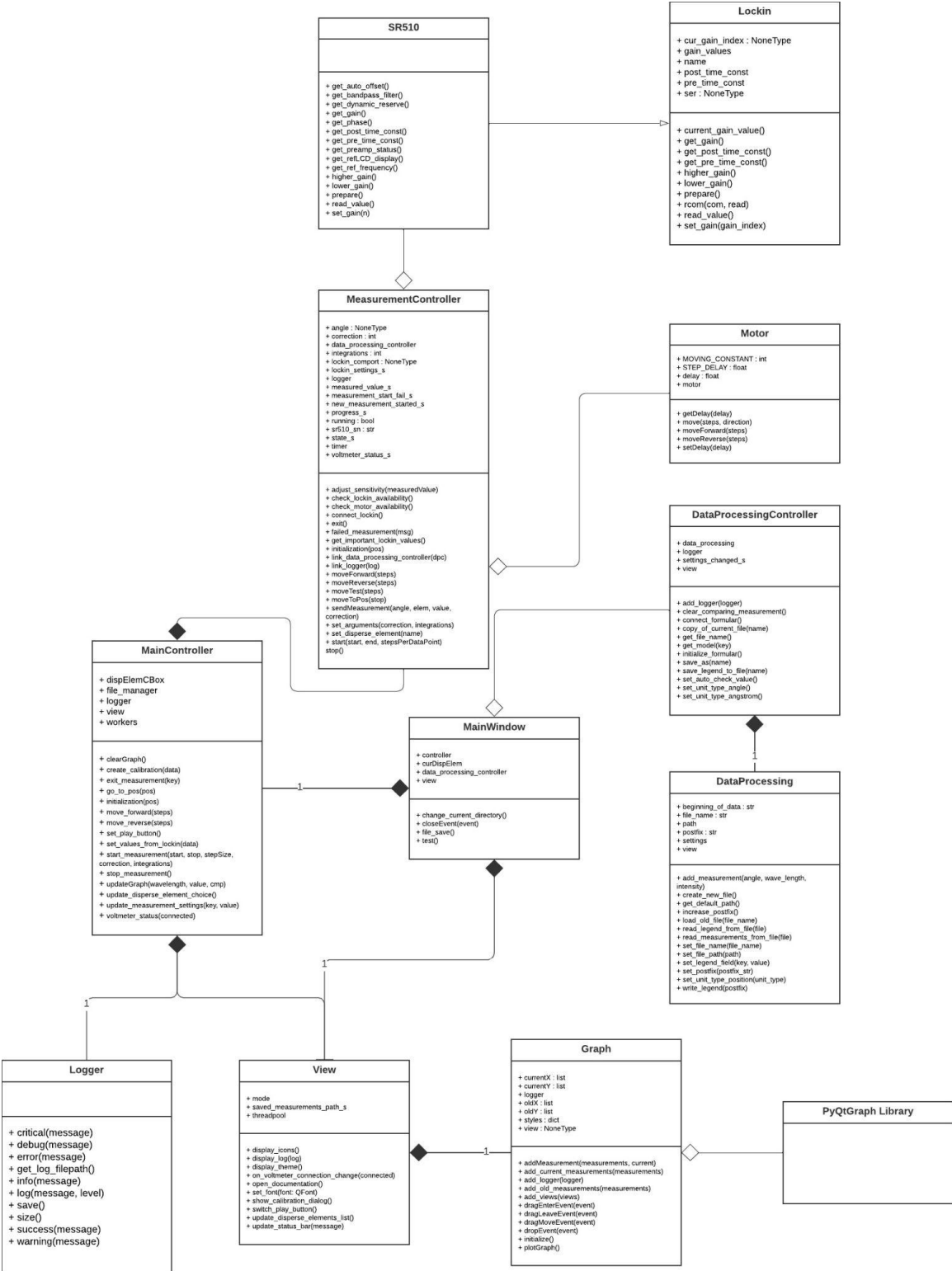
## 6. Implementácia

### 6.1. UML component diagram



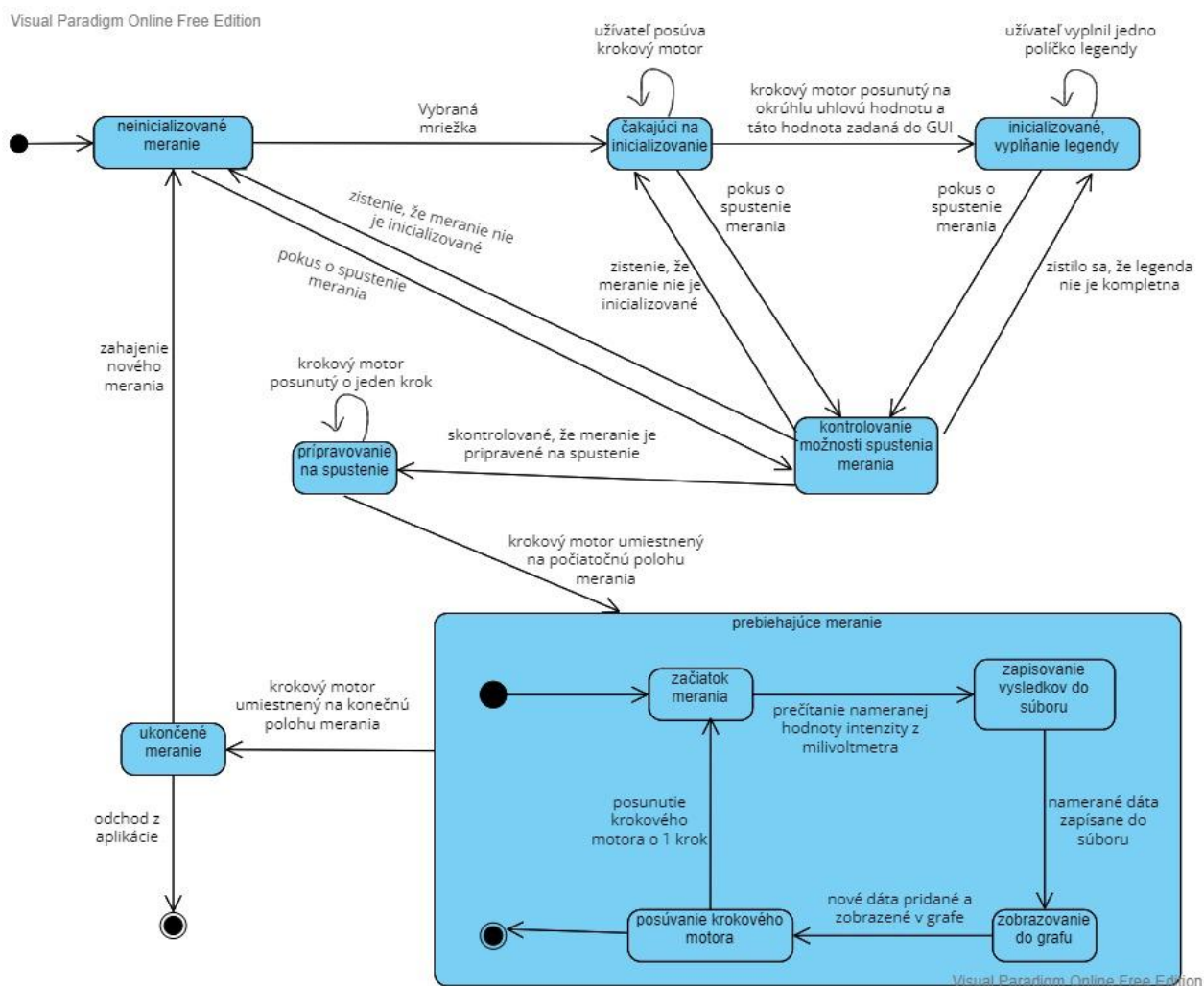


## 6.2. UML class diagram



### 6.3. UML state diagram

Na nasledujúcom obrázku je zobrazený stavový diagram pre entitu meranie.



## 7. Rozdelenie na moduly

### 7.1. Modul krokový motor

- zodpovedný za základnú komunikáciu s krokovým motorom
- obsahuje metódu na pohyb krokového motora smerom dopredu (v smere hodinových ručičiek) o presne daný počet krokov
- obsahuje metódu na pohyb krokového motora smerom dozadu (v proti smere hodinových ručičiek) o presne daný počet krokov
- implementuje zrýchľovanie či spomaľovanie posunu krokového motoru
- obsahuje kalibračnú metódu

### 7.2. Modul Disperzný prvok

- zodpovedný za ukladanie informácií o disperznom prvku
- modul uskutočňuje kalibráciu, resp. drží informácie o nej
- modul prepočítava kroky na uhol a opačne pre konkrétny disperzný prvok
- zabezpečuje dopočítanie aktuálnej vlnovej dĺžky na základe momentálneho uhla. Tento výpočet závisí od momentálne použitého typu mriežky.

### 7.3. Modul lockin

- zodpovedný za základnú komunikáciu s lockinom
- obsahuje metódy na prepínanie alebo čítanie citlivosti merania
- metódy na čítanie časových konštant alebo čítanie nameraného napätia

### 7.4. Modul čítanie, zápis a spracovanie dát

- zodpovedný za zápis práve nameraných hodnôt na koniec aktuálneho súboru
- zabezpečuje zapísanie legendy merania na začiatok súboru pred spustením merania. Informácie, ktoré sa zapisujú do legendy získava z gui a z lockinu.
- zodpovedá za prečítanie údajov zo staršieho súboru, tak aby sa dali údaje zo daného merania zakresliť do grafu paralelne s aktualne prebiehajúcim meraním.
- zodpovedá za prečítanie legendy zo staršieho súboru, tak aby sa dali údaje z legendy daného merania vypísať do gui.
- ukladá naposledy použité nastavenia merania a pri spustení aplikácie tieto nastavenia automaticky predvyplní do formuláru legendy v gui.

### 7.5. Modul GUI

- zodpovedný za správu GUI

- zabezpečuje zakreslovanie dát z aktuálneho merania do grafu. To aké hodnoty sa majú do grafu zakresliť sú posielané z modulu pre lockin a modulu pre disperzný prvok.
- zabezpečuje zakreslovanie dát zo staršieho ukončeného merania do grafu. To aké hodnoty sa majú do grafu zakresliť získava z modulu pre čítanie, zápis a spracovanie dát, ktorý z porovňovaného súboru získa zoznam dvojitých nameraných intenzít a vlnových dĺžiek.

## 7.6. Interfejsy medzi modulmi

- všetky moduly sú pod správou kontrolérov
- MainController zabezpečuje synchronizáciu práce medzi viacerými typmi kontrolerov:
  - Measurement Controller - zabezpečuje komunikáciu medzi modulmi KrokovýMotor, Lockin, DisperznýPrvok a modulom pre spracovanie dát.
    - zabezpečuje tlmočenie funkcií GUI kontroleru zo spomínaných modulov
    - ma na starosti ovládanie celého procesu merania. Teda dáva pokyny korešpondujúcim modulom, aby posuvali krokový motor, čítali namerané dáta z lockinu, zapisovali namerané dáta do súboru a aby sa zobrazovali do grafu namerané dáta.
    - komunikuje s modulom pre disperzný element, od ktorého zisťuje aktuálny uhol otočenia motoru a taktiež ho využíva na vypočítanie vlnovej dĺžky momentálneho merania (závisí od použitej mriežky).
  - Krokový Motor Controller - špecifikuje use case modulu Krokový motor, vrátenie sa na pôvodnú pozíciu a má implementované getter-y pre získavanie informácií o aktuálnej pozícii. Zabezpečuje napojenie a ovládanie krokového motora. Vie pohybovať krokovým motorom v smere aj v proti smere hodinových ručičiek, zrýchliť aj spomaliť pohyb otáčania.
    - Komunikuje s ním MeasurementController, ktorý mu posíla príkaz na posunutie sa krok ďalej počas merania.
    - Taktiež posúva motor o určený počet krokov požadovaným smerom pri prebiehajúcej inicializácii merania.
  - LockinController - špecifikuje use case modulu Lockin
    - zabezpečuje napojenie a ovládanie lockinu
    - ovláda zmenu citlivosti vzhľadom na intenzitu meraného signálu
    - číta nameranú intenzitu signálu z lockinu, keď MeasurementController počas merania zavolá túto funkcionality.

- FileController / DataProcessingController - špecifikuje use case modulu pre čítanie, zápis a spracovanie dát
  - komunikuje s MeasurementControllerom , ktorý mu počas merania posiela priebežne jednotlivé namerané údaje, ktoré sa majú zapísať do súboru (uhol, vlnovú dĺžku a intenzitu)
  - pri spustení aplikácie automaticky vyplní nastavenia/legendu merania podľa naposledy použitých nastavení.
  - pri vyplňaní legendy automaticky ukladá nové hodnoty do svojej vnútornej reprezentácie legendy v pamäti.
  - Komunikuje s GUI pri porovnávaní aktuálneho merania s iným starším meraním. Zo súboru prečíta legendu a načíta zoznam nameraných dát a tie pošle späť GUI na zobrazenie.
- GuiController - špecifikuje use case modulu GUI

## 8. Plán implementácie

- pripraviť dátové modely a vyjasniť technicko-implementačné otázky
- naprogramovať GUI prostredie v pyside a Qt (Martin)
- naprogramovať modul pre spracovanie dát merania a ich zápis a čítanie zo súboru (Lucia)
  - pripravenie metód, ktoré z lockinu a gui získavajú dáta potrebné na zápis do hlavičky súboru
  - pripravenie metód, ktoré zapíšu do súboru hlavičku
  - pripravenie metód, ktoré vedia prečítať hlavičku zo súboru
  - pripravenie metód, ktoré na koniec otvoreného súboru pridajú nové údaje získané z merania
  - pripravenie metód, ktoré prečítajú zo starého súboru všetky dáta z merania, aby sa mohli vykresliť do grafu
- naprogramovať vykresľovanie dát v grafe (Lucia)
  - príprava metódy, ktorá pridajú do grafu novu sadu údajov
  - príprava metódy, ktorá vykreslí údaje do grafu
- pripraviť Visual Basic skript na komunikáciu s Krokovým motorom (David)
- pripraviť modul a náväzný kontrolér pre krokový motor (David)
  - pripraviť ovládacie metódy
  - pripraviť kalibračné metódy
  - pripraviť getter metódy o polohe motora
  - definovať rozšírené správanie krokového motora v príslušnej Controller triede
- pripraviť modul pre lockin (Sebastian)
- každá z predchádzajúcich častí sa bude samostatne testovať a pripravovať paralelne a nezávisle od seba.
- Následne sa pripraví MeasurementController pomocou použitia a prepojenia modulov pre lockin a krokový motor (Sebastian, David)

- Príprava DataProcessingController za použitia modulu pre prácu so súbormi a jeho prepojenie s GUI aplikácie. (Lucia)
- následne bude potrebné otestovať či modul pre krokový motor a lockin vramci measurement controlleru spolupracujú tak ako by mali
- Pridanie komunikácie s data processing controllerom do measurement controlleru. Čiže pridanie spracovania a zobrazovania dát získaných z merania.
- Prepojenie jednotlivých modulov tak, aby spolu komunikovali tak ako by mali.
- ďalej sa otestuje či všetky komunikujúce moduly spolu navzájom správne pracujú
- Keď bude otestovaná celková funkcionálna aplikácia ako celku, výsledný produkt sa konzultuje so zadávateľom
- zapracujú sa prípadné pripomienky

## 9. Testovacie scenáre

### 9.1. Čítanie napätia

**Scenár:** Lockinu SR510 pošleme príkaz "Q\r"

**Očakávaný výstup:** `print(lockin.readline())` vypíše namerané napätie, ktoré vidno na displeji lockinu

### 9.2. Krokový motor jednoduchý posun o 1 krok dopredu

**Scenár:** Krokovému motoru zašleme príkaz o posune o 1 krok smerom dopredu, pričom do daného smeru je krokový motor natočený

**Očakávaný výstup:** Krokový motor sa posunie presne o 1 krok dopredu

### 9.3. Krokový motor jednoduchý posun o 1 krok dozadu

**Scenár:** Krokovému motoru zašleme príkaz o posune o 1 krok smerom dozadu, pričom do daného smeru je krokový motor natočený

**Očakávaný výstup:** Krokový motor sa posunie presne o 1 krok dozadu

### 9.4. Krokový motor posun o 201 krokov dopredu

**Scenár:** Krokovému motoru zašleme príkaz o posune o 201 krok smerom dopredu, pričom krokový motor je natočený dopredu.

**Očakávaný výstup:** Krokový motor sa posunie presne o 201 krok dopredu pričom posun bude zahrňovať zrýchlenie krokového motora v prvej fáze posunu (cca prvých 65 krokov) a spomalenie krokového motora v poslednej fáze posunu (cca posledných 65 krokov)

### 9.5. Krokový motor posun o 201 krokov dozadu

**Scenár:** Krokovému motoru zašleme príkaz o posune o 201 krok smerom dozadu

**Očakávaný výstup:** Krokový motor sa posunie presne o 201 krok dozadu pričom posun bude zahrňovať zrýchlenie krokového motora v prvej fáze

posunu (cca prvých 65 krokov) a spomalenie krokového motora v poslednej fáze posunu (cca posledných 65 krokov)

#### 9.6. Kalibrácia merania 1

**Scenár:** Kalibrujeme meranie prvýkrát - neexistuje žiadny údaj o predchádzajúcej kalibrácii. Krokový motor sa posunie o zadaný počet krokov od štartovacej hodnoty po koncovú pričom si pamätá zadaný počet krokov. Používateľ nakoniec zadá koncovú polohu, ktorú odčíta zo stupnice prístroja.  
**Očakávaný výstup:** Vypočítaná veľkosť kroku bude uložená v súbore.

#### 9.7. Kalibrácia merania 2

**Scenár:** Kalibrujeme meranie, pričom existuje údaj o predchádzajúcej kalibrácii. Krokový motor sa posunie o zadaný počet krokov od štartovacej hodnoty po koncovú pričom si pamätá zadaný počet krokov. Používateľ nakoniec zadá koncovú polohu, ktorú odčíta zo stupnice prístroja.  
**Očakávaný výstup:** Vypočítaná veľkosť kroku prepíše pôvodnú hodnotu kalibrácie pre danú mriežku.

#### 9.8. Meranie s jednotkovým krokom

**Scenár:** Pred meraním prebehla kalibrácia a inicializácia. Používateľ stanoví koncovú pozíciu merania, veľkosť kroku stanoví na 1.  
**Očakávaný výstup:** meranie prebehne od aktuálnej pozície presne po koncovú s informáciami o polohe merania (uhol natočenia mriežky) a nameranej hodnoty, jednotlivé hodnoty uhlu natočenia mriežky sa budú líšiť o jeden krok.

#### 9.9. Meranie s nie jednotkovým krokom

**Scenár:** Pred meraním prebehla kalibrácia a inicializácia. Používateľ stanoví koncovú pozíciu merania, veľkosť kroku stanoví na číslo v intervale 2 až 100.  
**Očakávaný výstup:** meranie prebehne od aktuálnej pozície presne po koncovú s informáciami o polohe merania (uhol natočenia mriežky) a nameranej hodnoty, jednotlivé hodnoty uhlu natočenia mriežky sa budú líšiť o veľkosť kroku vynásobenú s počtom krokov pripadajúcich na jedno meranie. Meranie je ukončené v presnej koncovej pozícii alebo ešte pred ňou, v závislosti od nastavených parametrov merania.

#### 9.10. Automatické vyplnenie formuláru informácií o meraní pri spustení aplikácie

**Scenár:** Užívateľ spustil aplikáciu.  
**Očakávaný výstup:** Systém nájde (na predpísanom mieste) súbor s údajmi o tom, ako bol pri poslednom meraní vyplnený formulár nastavení v gui. Ak takýto súbor v aplikácii neexistuje, všetky políčka formulára zostanú nevyplnené. V prípade, že sú v aplikácii uložené posledne nastavenia, systém tieto nastavenia načíta a vypíše do políčok vo formulári gui. Načíta

všetky údaje až na názov súboru a názov merania, aby sa zabránilo prípadným chybám.

#### 9.11. Automatické uloženie vyplnených informácií hlavičky pri spustení merania

**Scenár:** Užívateľ vyplnil všetky údaje hlavičky vo formulári gui a úspešne spustí meranie.

**Očakávaný výstup:** Systém vytvorí/prepíše konfiguračný súbor informácií hlavičky na predpísanom mieste. Zapiše doňho všetky ukladané údaje z formulára (až na meno súboru a názov merania) v formáte json.

#### 9.12. Pokus o spustenie merania s chýbajúcimi údajmi vo formulári o meraní

**Scenár:** Užívateľ nevyplnil všetky potrebné údaje vo formulári o meraní a stlačil tlačidlo na spustenie merania.

**Očakávaný výstup:** Systém zistí, že nejaké údaje chýbajú, bez ktorých nie je možné vytvoriť súbor o meraní (napr. názov súbor, nejaka súčasť hlavičky súboru,...) alebo správne nastaviť lockin a krokový motor. Meranie sa nespustí, v loggeri sa o tom napíše správa a bude sa čakať, kým užívateľ dodá potrebné informácie.

#### 9.13. Vytvorenie nového súboru a vypísanie hlavičky do súboru

**Scenár:** Užívateľ vyplnil všetky potrebné údaje vo formulári o meraní a stlačil tlačidlo na spustenie merania.

**Očakávaný výstup:** Systém z používateľského rozhrania načíta všetky údaje, ktoré sa majú objaviť v hlavičke súboru a názov súboru. Vytvorí súbor s požadovaným názvom a správne vyplnenou hlavičkou.

#### 9.14. Automatické pridanie číselného postfixu do mena súboru

**Scenár:** Užívateľ vyplnil všetky potrebné údaje vo formulári o meraní a stlačil tlačidlo na spustenie merania. Ako názov súboru určil meno x.txt, avšak v defaultnom priečinku (saved\_measurements) už existuje súbor s menom x.txt.

**Očakávaný výstup:** Systém zistí, že v defaultnom priečinku sa už nachádza súbor x.txt a pridá automaticky číselný postfix na koniec názvu súboru (začína od postfix 1. Ak sa nachádza v priečinku x.txt, x1.txt, x2.txt tak postfix bude 3). Systém z používateľského rozhrania načíta všetky údaje, ktoré sa majú objaviť v hlavičke súboru a názov súboru s novým postfixom. Vytvorí súbor s požadovaným názvom (x1.txt) a správne vyplnenou hlavičkou. Taktiež sa v legende v políčku s názvom súboru aktualizuje názov na nový názov s postfixom.



#### 9.15. Vytvorenie kópie súboru na užívateľom určenom mieste

**Scenár:** Užívateľ vyplnil všetky potrebné údaje vo formulári o meraní a stlačil tlačidlo na spustenie merania. Prebehlo cele meranie. Následne stlačil v menu možnosť na uloženie merania ako..

**Očakávaný výstup:** Systém zobrazí okienko na vybratie lokácie a pomenovanie kópie súboru aktuálneho merania. Po zvolení lokácie kópie, sa vytvorí nový súbor, do ktorého sa skopírujú údaje o poslednom meraní na zvolené miesto.

#### 9.16. Vytvorenie kópie súboru na užívateľom určenom mieste pred spustením merania

**Scenár:** Užívateľ vyplnil všetky potrebné údaje vo formulári o meraní (v legende), ale ešte nespustil meranie. Následne stlačil v menu možnosť na uloženie merania ako..

**Očakávaný výstup:** Systém zobrazí okienko na vybratie lokácie a pomenovanie kópie súboru aktuálneho merania. Po zvolení lokácie kópie, sa vytvorí nový súbor, do ktorého sa zapíšu údaje o legende na zvolené miesto. V tomto novom súbore bude len legenda a nebudú tam žiadne namerané údaje, keďže ešte meranie spustené nebolo.

#### 9.17. Vyrátanie vlnovej dĺžky na základe uhla krokového motora

**Scenár:** Prístroj sa posunul o jeden krok a namerá nový údaj. Pred poslaním na zapísanie do súboru, sa vyráta aktuálna vlnová dĺžka prístroja.

**Očakávaný výstup:** Na základe uhla (získa sa z krokového motoru aktuálny uhol natočenia prístroja) a momentálnej mriežky získa vlnovú dĺžku, ktorá sa pošle na zapísanie do súboru a na vykreslenie do grafu.

#### 9.18. Zápis údajov z merania do súboru

**Scenár:** Prístroj sa posunul o jeden krok a namerá nový údaj. Tento bol poslaný na zapísanie do súboru.

**Očakávaný výstup:** Komponent zabezpečujúci prácu so súbormi získa nameranú intenzitu. Následne získa momentálnu vlnovú dĺžku a uhol od ostatných komponentov systému. Tieto tri údaje sa zapíšu na nový riadok súboru v nasledujúcom formáte:

uhol Alfa [°]                      vlnová dĺžka [A°]                      intenzita[mv]

#### 9.19. Správne vykreslenie údajov v grafe

**Scenár:** Prístroj sa posunul o jeden krok a namerá nový údaj. Tento bol poslaný na vykreslenie do grafu.

**Očakávaný výstup:** Komponent zabezpečujúci prácu s grafom získa nameranú intenzitu. V grafe sa zobrazuje na y-ovej osi nameraná intenzita a na x-ovej vlnová dĺžka. Preto sa na grafe zobrazí nový modrý bod na suradnici [A, I], kde I sa rovná intenzite a A je vlnová dĺžka.

#### 9.20. Načítanie údajov merania zo staršieho súboru

**Scenár:** užívateľ vybral možnosť načítania údajov zo súboru staršieho merania na porovnanie s aktuálnym meraním.

**Očakávaný výstup:** Komponent zabezpečujúci prácu so súbormi zo súboru získa zoznam dát o intenzite a vlnových dĺžkach z daného merania a legendu daného merania. Tento zoznam dát sa vykreslí v grafe červenou farbou. Legenda staršieho merania sa zobrazí v komponente pre zobrazenie dodatočných informácií o porovnávanom meraní (v kapitole 5.3. je označený číslom 17).

#### 9.21. Pokus o načítanie údajov merania zo staršieho súboru v zlom formáte

**Scenár:** užívateľ vybral možnosť načítania údajov zo súboru merania z minulosti do grafu. Avšak vybral nevhodný súbor.

**Očakávaný výstup:** Komponent zabezpečujúci prácu so súbormi zistí, že formát súboru je nevhodný a vypíše o tom správu v logu. Do grafu sa nepridajú žiadne dáta a nevypíše sa žiadna legenda.

#### 9.22. Správne vybratie súboru na porovnanie s aktuálnym meraním

**Scenár:** Užívateľ si v okienku "výber porovnávacieho merania" vyberie želaný súbor a ten presunie draggovaním nad grafový komponent a tam ho dropne.

**Očakávaný výstup:** Komponent zabezpečujúci prácu so súbormi zo súboru získa zoznam dát o intenzite a vlnových dĺžkach z daného merania a legendu daného merania. Tento zoznam dát sa vykreslí v grafe červenou farbou. Legenda staršieho merania sa zobrazí v komponente pre zobrazenie dodatočných informácií o porovnávanom meraní (v kapitole 5.3. je označený číslom 17).

#### 9.23. Zmena adresára, z ktorého chceme vybrať súbor s porovnávaným meraním

**Scenár:** Používateľ stlačí tlačidlo "zmeniť adresár" a zobrazí sa okno zobrazujúce súborovú štruktúru. Užívateľ si vyberie priečinok, ktorého obsah sa zobrazí v okne nad daným tlačidlom.

**Očakávaný výstup:** Obsah vybraného priečinku sa zobrazí v časti "výber porovnávacieho merania" a z neho si môže užívateľ vybrať porovnavacie meranie.

#### 9.24. Nahradenie vybraného súboru na porovnanie iným súborom

**Scenár:** Bolo vybrané porovnavacie meranie a v gui sú zobrazené informácie o tomto meraní. Užívateľ si v okienku "výber porovnávacieho merania" vyberie želaný súbor a ten presunie draggovaním nad grafový komponent a tam ho dropne.

**Očakávaný výstup:** V grafe sa nahradia namerané dáta z doteraz porovnávaného merania dátami z novo vybraného súboru na porovnanie. Legenda sa taktiež nahradí novou.

#### 9.25. Zrušenie porovnania

**Scenár:** Bolo vybrané porovnavacie meranie a v gui sú zobrazené informácie o tomto meraní. Užívateľ stlačí tlačidlo “zruš”.

**Očakávaný výstup:** V grafe sa odstránia namerané dáta z doteraz porovnávacieho merania. Zobrazená legenda sa taktiež odstraní.

#### 9.26. Zvolenie používaného disperzného prvku

**Scenár:** Užívateľ vyberie jeden z možných disperzných prvkov.

**Očakávaný výstup:** Na základe vybraného disperzného prvku (mriežky) sa v systéme uloží informácia o používanom disperznom prvku. Následne sa počas merania na základe zvolenej mriežky používa správny vzorec na vypočítanie vlnovej dĺžky. taktiež inicializácia a kalibrácia sú prispôsobené aktuálnej mriežke.

#### 9.27. Spustenie merania

**Scenár:** Užívateľ inicializoval meranie, vyplnil všetky políčka formulára nastavujúceho parametre merania a stlačil tlačidlo start (komponent číslo 4 v kapitole 5.3.)

**Očakávaný výstup:** Kontroluje sa či boli vyplnené všetky prvky legendy merania. Keďže nastavenia sú kompletne, vytvorí sa súbor s názvom od užívateľa v defaultnom priečinku a doňho sa zapíše v hlavičke legenda merania. Krokový motor sa presunie na počiatočnú polohu určenú užívateľom a zaháji sa meranie.

#### 9.28. Jeden krok merania

**Scenár:** Užívateľ spustil meranie.

**Očakávaný výstup:** Krokový motor sa posunie na počiatočnú pozíciu určenú užívateľom. Vykonáva jeden krok merania až pokým sa neposunie na konečnú pozíciu meranie. Vtedy sa meranie ukončí. Jeden krok merania pozostáva z prečítania nameranej intenzity z milivoltmetra. Zapísanie a dopočítanie aktuálneho uhla, vlnovej dĺžky a intenzity do súboru. Zakreslenie intenzity a vlnovej dĺžky do grafu. Na záver kroku sa posunie krokový motor o jeden krok ďalej a vykoná sa celá táto sekvencia úkonov od začiatku až pokým sa meranie neskončí.

#### 9.29. Zmena jednotky z Angstrom na uhol

**Scenár:** Užívateľ klikne na radiobutton uhol.

**Očakávaný výstup:** Stlačením tlačidla “CHOĎ” so zadanou platnou hodnotou sa motor presunie na danú pozíciu v uhloch. Počiatočná a koncová pozícia meraného intervalu je teraz zadávaná v uhloch.

#### 9.30. Zmena jednotky z uhol na Angstrom

**Scenár:** Užívateľ klikne na radiobutton Angstrom.

**Očakávaný výstup:** Stlačením tlačidla “CHOĎ” so zadanou platnou hodnotou sa motor presunie na danú pozíciu v Angstromoch, ktorá podľa tabuliek pre danú mriežku zodpovedá určitému uhlu. Počiatočná a koncová pozícia meraného intervalu je teraz zadávaná v Angstromoch.

#### 9.31. Inicializovanie polohy motora

**Scenár:** Užívateľ vyberie jeden z možných disperzných prvkov a milivoltmeter. Užívateľ nastaví motor na žiadanú polohu pomocou šípok na polohovanie motora vpravo/vľavo (pozn. motor musí byť nasmerovaný dopredu v koncovej polohe). Používateľ nastaví hodnotu uhla podľa hodnoty zo stupnice spektrometra do políčka “Inicializovanie polohy merania” a stlačí tlačidlo “Inicializuj”.

**Očakávaný výstup:** Poloha motora je inicializovaná, následne funguje tlačidlo “CHOĎ”, ktoré posunie motor na zadanú platnú pozíciu, taktiež už funguje samotné meranie.

#### 9.32. Logovanie

**Scenár:** Užívateľ sa pokúsil spustiť meranie bez vyplnenia všetkých položiek legendy merania.

**Očakávaný výstup:** V logeri sa vypíše chybová hláška o tom, že nebola kompletne vyplnená legenda merania.