

Návrh informačného systému

Krokový motor (BREWSTER)



Konfigurácia, čítanie a vizualizácia údajov z meracieho prístroja riadeného pomocou krokového motora.

4. 12. 2020

1. Úvod	3
1.1 Účel dokumentu	3
1.2 Zameranie a rozsah	3
1.3 Prehľad nasledujúcich kapitol	3
2. Špecifikácia vonkajších interfejsov	3
3. Dátový model	4
3.1 Formáty súborov	4
3.2 Komunikačné protokoly	5
4. Návrh používateľského rozhrania	5
4.1 Grafické používateľské rozhranie	6
5. Návrh implementácie	7
5.1 Stavový diagram	7
5.2 Diagram komponentov	7
5.3 Class diagram	9
5.4 Využitie technológií	10
5.5 Plán implementácie	10
6. Testovacie scenáre	10
6.1 Meranie šumu	10
6.2 Kontrola pripojených zariadení	11
6.3 Meranie aktuálneho spektra	11
6.4 Spustenie série meraní a vykreslenie grafu minimalných intenzít	11
6.5 Pohyb krokového motora a nastavenie počtu pulzov posunu	11
6.6 Zvolenie uhlovej jednotky	11
6.7 Nastavenie expozičnej doby spektrometra	12
6.8 Nastavenie parametrov a poznámky pre sériu meraní a kontrola konfiguračného súboru	12
6.9 Bezpečnostné tlačidlo STOP	12
6.10 Kalibrácia a aktuálny uhol	12
6.11 Posúvanie na zadaný uhol	12

1. Úvod

1.1 Účel dokumentu

Tento dokument slúži ako kompletný a detailný návrh informačného systému Brewster (`krokovy_motor`). Obsahuje všetky informácie potrebné pre vysvetlenie a pochopenie funkcionality ako aj spôsobu implementácie systému. Tento dokument je primárne určený pre vývojárov. Obsah v tomto dokumente pokrýva všetky požiadavky z katalógu požiadaviek.

1.2 Zameranie a rozsah

Pre prácu s týmto dokumentom sa vyžaduje oboznámenie sa s dokumentom katalóg požiadaviek, s ktorým návrh úzko súvisí. Tento dokument obsahuje kompletný a detailný návrh implementácie všetkých požiadaviek z katalógu požiadaviek. Ďalej špecifikuje vonkajšie interfejsy, formáty súborov a komunikačné protokoly. Obsahuje taktiež celkový návrh používateľského prostredia vrátane vizualizácie. Návrh implementácie opisujú diagramy, konkrétne stavový diagram, komponentný diagram a class diagram. Okrem toho sú v dokumente opísané využité technológie a cieľové prostredie nasadenia do prevádzky.

1.3 Prehľad nasledujúcich kapitol

Nasledujúce kapitoly sa venujú vonkajším interfejsom, dátovému modelu, používateľskému rozhraniu a jeho vizualizácií. Posledná kapitola opisuje detailný návrh implementácie systému.

2. Špecifikácia vonkajších interfejsov

Aplikácia bude spustená lokálne v zariadení a s používateľom bude komunikovať pomocou používateľského grafického rozhrania. Jadro systému bude komunikovať s dvoma externými zariadeniami, so spektrometrom a mikročipom pre ovládanie krokového motora. Program vie komunikovať so spektrometrom RedTide 650 UV Fiber Optic Spectrometer od spoločnosti Ocean Optics (ktorý je do zariadenia pripojený pomocou USB portu) pomocou drivera OmniDriver. Spektrometer do jadra systému posiela údaje o intenzite odrazeného svetla. Mikročip Picaxe 18M2 je do zariadenia pripojený a komunikuje cez virtuálny sériový port (ktorý sa vytvorí po pripojení prevodníka do konektora USB) pomocou nižšie uvedeného komunikačného protokolu. Jadro systému bude tento port detekovať.

3. Dátový model

3.1 Formáty súborov

Aplikácia uloží po každom meraní do novo-vytvoreného priečinku textové súbory. Priečinok bude mať názov vo formáte 'yyyy-MM-dd-HH-mm-ss' (dátum a čas začiatku merania). V tomto priečinku sa vytvorí 1 konfiguračný súbor s názvom config.txt a pre každý uhol natočenia ramena sa vytvorí súbor obsahujúci namerané spektrum s názvom vo formáte 'XXX-YYYY' v nameraných uhlových jednotkách zaokrúhlená na 4 desatinné miesta pričom symbol '-' oddeľuje desatinné miesta vpravo od celej časti vľavo. Celá časť vľavo bude doplnená nulami na 3 miesta zľava.

3.1.1 Textový súbor config.txt

Obsahuje nasledujúce konfiguračné údaje rozdelené po riadkoch.

- uhlové jednotky: "GRAD" / "DEG"
- mód: "CURRENT" / "LONG-TIME-AVG('pocet merani')"
- rozsah uhlov: "START 'pociatocny uhol' END 'konečny uhol' "
- expozičná doba: (ms) 'hodnota'
- veľkosť kroku: 'hodnota'
- pomer uhol : 1 krok: 'hodnota'
- rozsah vln. dĺžok (nm): " 'dolná hranica' - 'horná hranica' "
- odčítanie šumu: "TRUE" / "FALSE"
- lampa: 'parametre lampy'
- poznámka: 'poznámka k meraniu'

3.1.2 Textový súbor 'XXX-YYYY'.txt

Obsahuje n riadkov dvojíc " 'hodnota vlnovej dĺžky' - 'intenzita' ". Pričom n je rozsah vlnových dĺžok medzi dolnou a hornou hranicou s intervalom 1nm.

3.1.2 Textový súbor minimalValues.txt

Obsahuje n riadkov dvojíc " 'hodnota vlnovej dĺžky' - 'uhol pre ktoré nastalo minimum' ". Pričom n je rozsah vlnových dĺžok medzi dolnou a hornou hranicou s intervalom 1nm.

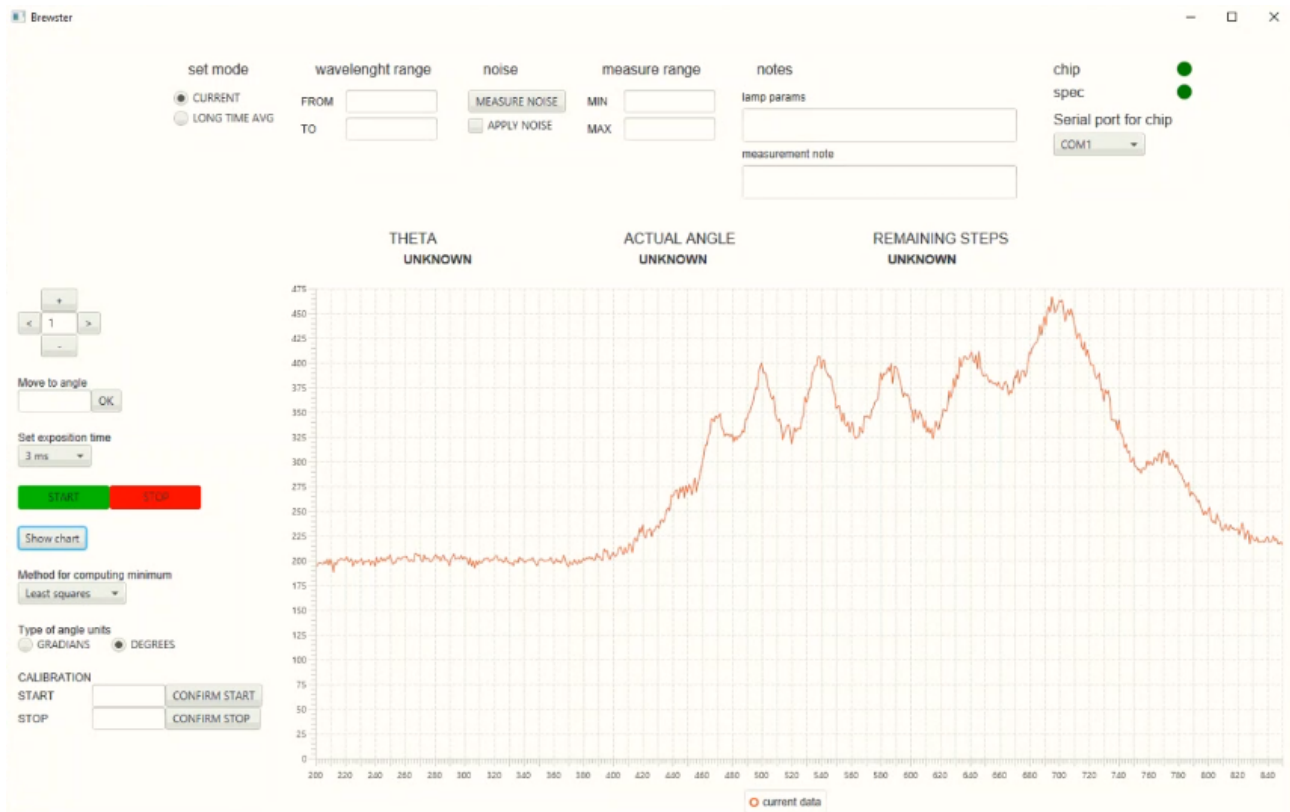
3.1.2 Textový súbor matica.txt

Obsahuje maticu so všetkými nameranými hodnotami, kde stĺpce reprezentujú vlnové dĺžky, riadky reprezentujú uhly a v poliach sú príslušné namerané intenzity.

3.2 Komunikačné protokoly

- **Komunikačný protokol pre zariadenie Picaxe 18M2** - zariadenie Picaxe po prijatí znaku '+' otočí motorom jedným smerom o jeden krok a po prijatí znaku '-' otočí motorom opačným smerom. Nasledujúci znak sa spracuje až vtedy, keď je motor pripravený vykonať ďalší krok. Na otvorenie komunikácie posielame na port znak '!', ak motor odpovie znakom '@', tak je úspešne pripojený. Hocikeď počas behu programu môže Picaxe dostať znak 'n', na základe čoho sa zastaví pritekanie prúdu do motora (opatrenie proti prehrievaniu). Komunikáciu medzi programom a zariadením Picaxe zabezpečuje knižnica JSerialComm.
- **Komunikácia so spektrometrom** prebieha pomocou drivera Omnidriver-2.56. Spektrometer sa ovláda cez inštanciu triedy Wrapper. Metódou `openAllSpectrometers()` zistíme, či je nejaký spektrometer pripojený k počítaču, prípadne, či ich je viac súčasne. Pomocou metódy `setIntegrationTime(spectrometerIndex, integrationTime)` vieme zadanému spektrometru nastaviť expozičnú dobu, pomocou metódy `setScansToAverage(spectrometerIndex, numberOfScansToAverageTogether)` vieme spektrometru nastaviť, aby každé meranie (získanie spektra) zopakoval viackrát a potom ich spriemeroval a pomocou metódy `getSpectrum()` vieme zo spektrometra získať aktuálne hodnoty spektra.

4. Návrh používateľského rozhrania



4.1 Grafické používateľské rozhranie

- **Tlačidlo START** - spustí sériu meraní s nastavenými parametrami.
- **Tlačidlo STOP** - bezpečnostné tlačidlo, ktoré všetko zastaví a vypne celý program.
- **Radiobuttony GRADIANS a DEGREES** - slúžia na zvolenie jednotky, v ktorej sa budú určovať stupne
- **Label Calibration START** - zobrazuje nastavenú začiatočnú pozíciu kalibrácie, ktorú používateľ nastaví tlačidlami + a - na stranách labelu.
- **Label Calibration STOP** - zobrazuje nastavenú koncovú pozíciu kalibrácie, ktorú používateľ nastaví tlačidlami + a - na stranách labelu.
- **Tlačidlo CONFIRM START** - potvrdí začiatočnú pozíciu kalibrácie
- **Tlačidlo CONFIRM STOP** - potvrdí koncovú pozíciu kalibrácie
- **Checkbox CURRENT** - nastaví mód merania na meranie aktuálneho spektra
- **Checkbox LONG TIME AVG** - nastaví mód merania na priemerované meranie
- **Label count of measures** - zobrazí počet meraní priemerovaného merania, ktorú používateľ nastaví tlačidlami + a - na stranách labelu
- **Label FROM** - zobrazí dolnú hranicu vlnových dĺžok pre ktoré sa uskutočňuje meranie, ktorú používateľ nastaví tlačidlami + a - na stranách labelu
- **Label TO** - zobrazí hornú hranicu vlnových dĺžok pre ktoré sa uskutočňuje meranie, ktorú používateľ nastaví tlačidlami + a - na stranách labelu
- **Button MEASURE NOISE** - spustí meranie šumu
- **CheckBox APPLY NOISE** - nastaví odčítanie posledného nameraného šumu pre nasledujúce merania
- **Label MIN** - zobrazí dolnú hranicu rozsahu merania v uhlových jednotkách pre ktoré sa uskutočňuje meranie, ktorú používateľ nastaví tlačidlami + a - na stranách labelu
- **Label MAX** - zobrazí hornú hranicu rozsahu merania v uhlových jednotkách pre ktoré sa uskutočňuje meranie, ktorú používateľ nastaví tlačidlami + a - na stranách labelu
- **TextArea Lamp params** - zadaný text v textboxe sa uloží do pamäte ako parametre lampy
- **TextArea Measure note** - zadaný text v textboxe sa uloží do pamäte ako poznámka k sérii meraní
- **Šípky na otáčanie motora** - tlačidlá + a - pridávajú a ubierajú počet pulzov pre jeden posun motora. Pomocou tlačidiel <- a -> sa rameno motora posunie o zadanú hodnotu pulzov dopredu/dozadu. Label v strede medzi tlačidlami sa zobrazuje aktuálnu hodnotu pulzov jedného posunu.
- **Combobox SET EXPOSITION TIME** - nastavenie expozičného času

- **Textbox MOVE TO ANGLE** - slúži na načítanie uhlu, na ktorý sa po stlačení tlačidla OK vedľa tohto textboxu rameno presunie
- **Kontrolky chip a spec** - zobrazujú, či sú spektrometer a čip Picaxe pripojené (zelená farba) alebo nie (červená farba)
- **Label ACTUAL ANGLE** - zobrazuje aktuálny uhol goniometra
- **Label THETA** - zobrazuje aktualnu hodnotu thety
- **Label REMAINING STEPS** - zobrazuje aktuálny počet zostávajúcich krokov v danej sérii meraní
- **Combobox pre výber komunikačného portu** - ponúka na výber COM porty pre pripojenie čipu.
- **Combobox pre výber metódy pre počítanie miním** - ponúka na výber danú metódu pre výpočet miním
- **Tlačidlo na načítanie a zobrazenie grafu miním zo súboru** - po stlačení tlačidla sa zobrazí možnosť výberu konkrétneho súboru na základe ktorého sa vykreslí graf miním v novom okne

5. Návrh implementácie

5.1 Stavový diagram

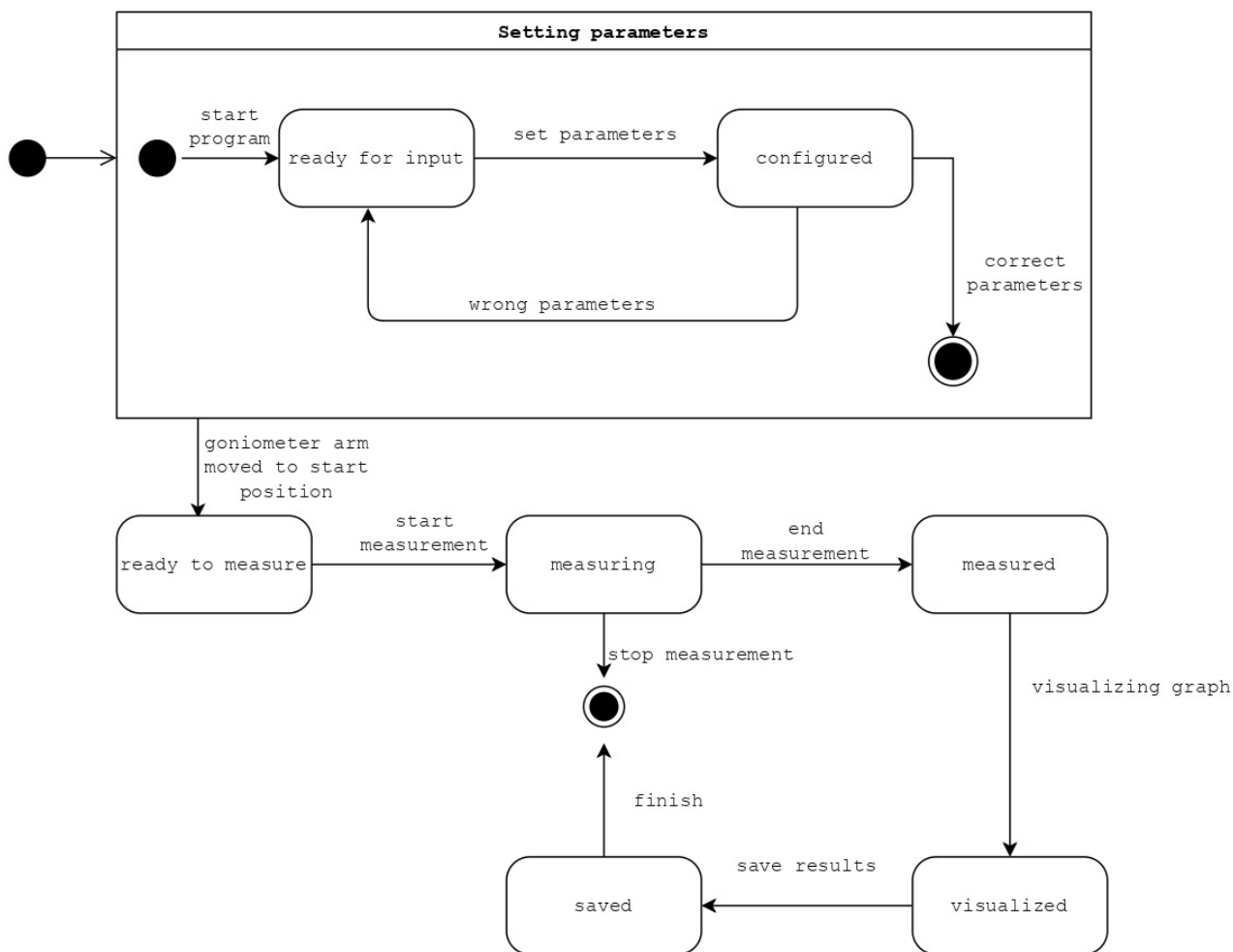
Po spustení programu je séria meraní pripravená na nastavenie jej parametrov. Keď bude nejaký parameter zadáný v nepovolenom formáte, séria meraní sa nespustí, ale nastavenia budú musieť byť najprv opravené. Po nastavení parametrov sa rameno goniometra posunie na pozíciu, z ktorej bude meranie začínať. Postupne sa namerajú všetky hodnoty a nakoniec sa vykreslí výsledný graf. Nakoniec sa dáta uložia do súboru. Môže sa stať, že používateľ predčasne ukončí vykonávanie série meraní už počas jej behu. Vtedy sa hodnoty nevykreslia ani neuložia.

5.2 Diagram komponentov

Používateľ bude môcť interagovať s komponentom GUI settings editor, pomocou ktorého vykoná všetky potrebné nastavenia. Tieto údaje pošle komponentu Settings handler, ktorý ich uchová. Ďalej ich bude využívať komponent Measurement handler pri meraniach. Measurement handler môže namerané dáta exportovať do súboru pomocou Data exporter alebo ich pošle komponentu Graph visualisation na zobrazenie v grafe. Taktiež bude využívať komponenty Spectrometer communication a Picaxe communication na komunikáciu so zariadeniami. Komponent GUI picaxe control slúži na posielanie príkazov z používateľského rozhrania čipu Picaxe. Na to využije komponent Picaxe communication.

TIS - Krokový motor

State diagram for entity measurement series



Stavový diagram

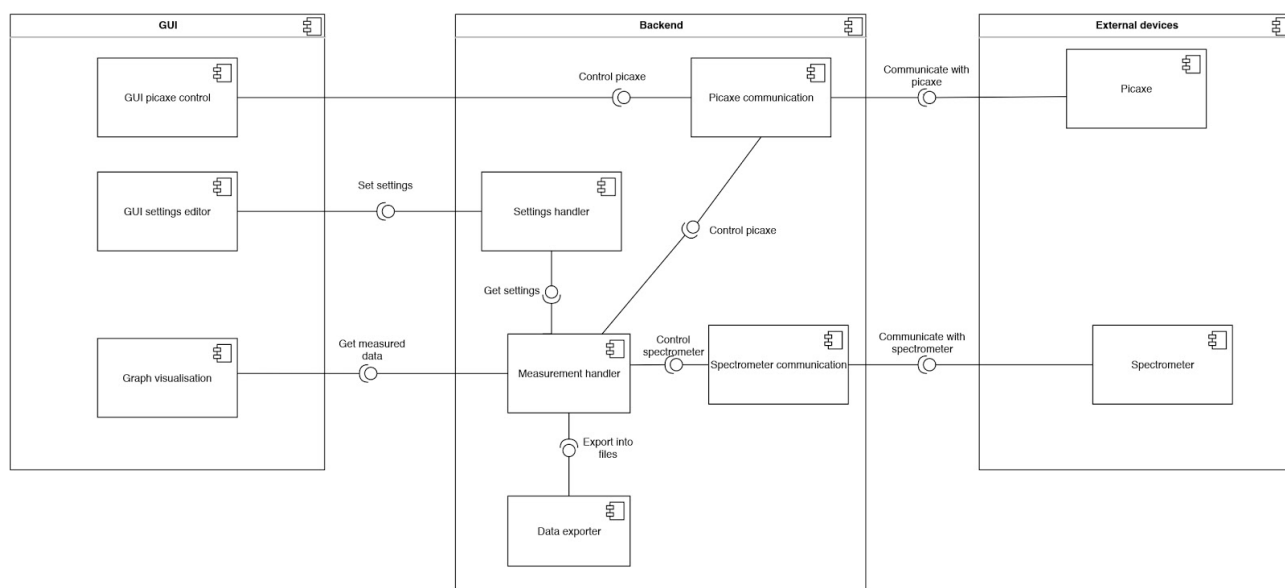
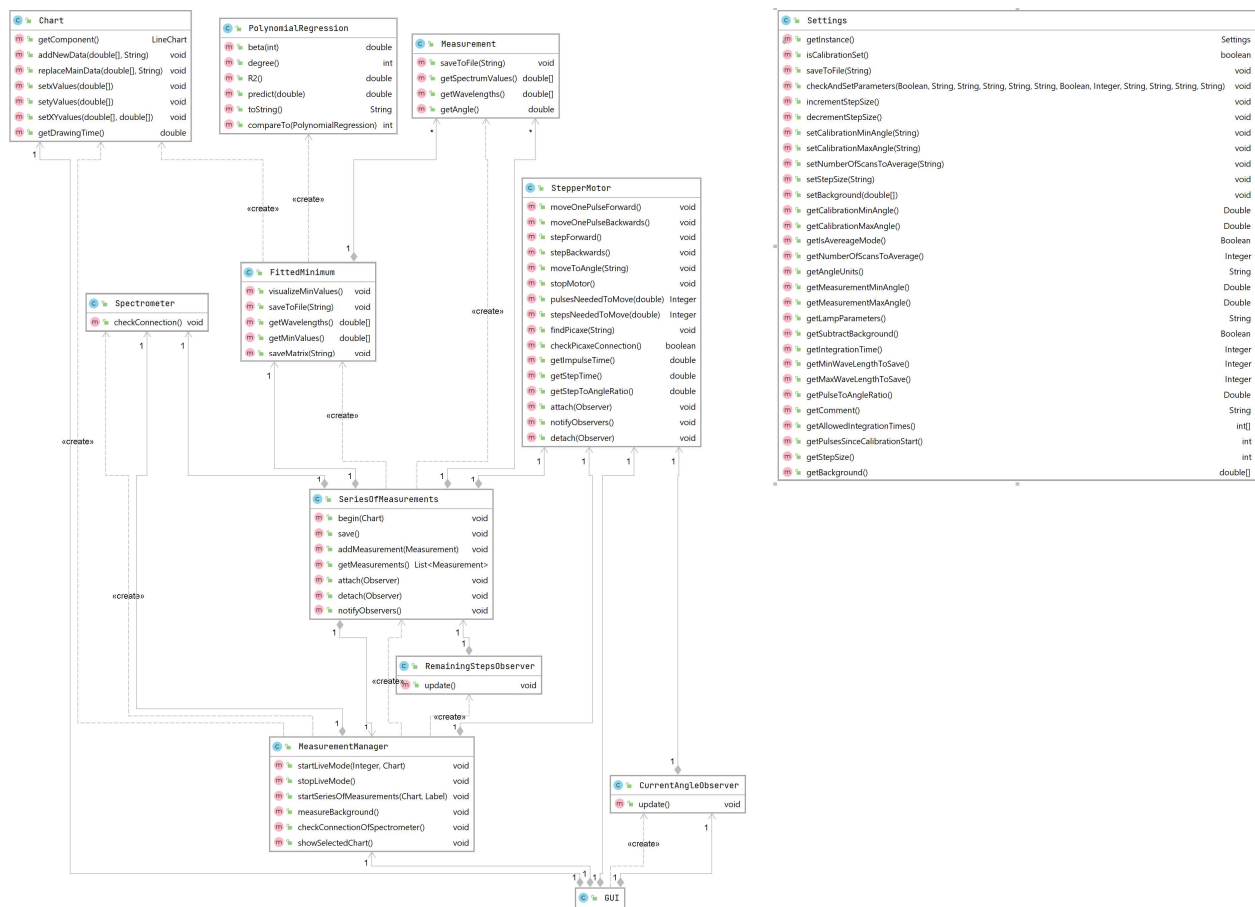


Diagram komponentov

5.3 Class diagram

Trieda GUI predstavuje používateľské rozhranie. Nastavenia zadané v používateľskom rozhraní sa uložia pomocou triedy SettingsManager, v ktorej sa vykonávajú aj kontroly správnosti vstupov. Ďalej ich využíva MeasurementManager pri spustení merania. GUI využíva aj triedu ChartManager na zobrazenie grafu a triedu SerialCommManager na pre komunikáciu s čipom picaxe v prípade, že používateľ chce pohnúť ramenom goniometra priamo v gui.

Trieda SpectrometerWrapper je navrhnutá podľa vzoru singleton a uchováva objekt wrapper, ktorý slúži na ovládanie spektrometra. Pomocou tried Measurement a ConfigurationFile sa dajú uchovať údaje merania a konfiguračný súbor. Merania a konfiguračný súbor sú súčasťou série meraní a preto tieto triedy využíva trieda SeriesOfMeasurements.



Class diagram

5.4 Využitie technológie

Javafx Linechart - používame na vykresľovanie 2D grafu.

- Linechart<Number,Number> - celý graf
- NumberAxis - jedna z osí grafu
- XYChart.Series() - predstavuje dvojicu x-ových a y-ových súradníc, ktoré sa zobrazia v grafe
- XYChart.Data() - predstavuje jednu dvojicu x-ovej a y-ovej súradnice

JSerialComm - používame na sériovú komunikáciu so zariadením Picaxe.

- SerialPort.getCommPorts() - získa všetky dostupné porty
- SerialPort.openPort() - otvorí daný port
- SerialPort.writeBytes(data, n) - do daného portu zapíšeme bajty (z data)
- SerialPortEvent.getReceivedData() - vráti prijaté data z portu

5.5 Plán implementácie

- Reprezentácia merania a série meraní v pamäti
- Reprezentácia nastavení merania v pamäti
- Zachytávanie nameraného spektra zo spektrometra
- Vykresľovanie grafu z nameraných hodnôt
- Používateľské rozhranie
- Ovládanie krokového motora - posúvanie so šípkami z GUI, posun na zadaný uhol
- Implementácia kalibrácie
- Implementácia série meraní
- Implementácia ukladania meraní do súboru
- Hľadanie minima z nameraných hodnôt a vykreslenie výsledných hodnôt

6. Testovacie scenáre

6.1 Meranie šumu

- A. Scenár** - Po stlačení tlačidla meraj šum a zobrazení vyskakovacieho okna zmeriame šum. Za rovnakých podmienok spustíme sériu meraní bez použitia šumu a následne s použitím nameraného šumu. Výsledky porovnáme.
- B. Výstup** - Výsledne grafy série meraní sa budú líšiť práve o nameraný šum.

6.2 Kontrola pripojených zariadení

- A. **Scenár** - Spustíme sériu meraní bez toho aby bol zapojený spektrometer. To isté zopakujeme s krokovým motorom.
- B. **Výstup** - Séria meraní sa nespustí, bodky sa okraji okna budú zafarbené na červeno pre spektrometer resp. krokový motor.

6.3 Meranie aktuálneho spektra

- A. **Scenár** - Do počítača zapojíme spektrometer. Do spektrometra skúsime zasietiť alebo ho prípadne zakrijeme rukou a budeme sledovať zmeny v grafe v okne aplikácie.
- B. **Výstup** - Graf bude v reálnom čase reagovať na zmeny intenzít nameraného svetla zo spektrometra.

6.4 Spustenie série meraní a vykreslenie grafu minimalných intenzít

- A. **Scenár** - Spustíme sériu meraní s ľubovoľným nastavením a čakáme na vykreslenie grafu minimalných intenzít.
- B. **Výstup** - Séria meraní sa úspešne ukončí a vykreslí sa graf, ktorý bude meraniu zodpovedať.

6.5 Pohyb krokového motora a nastavenie počtu pulzov posunu

- A. **Scenár** - Do počítača pripojíme krokový motor a následne pomocou tlačidiel v grafickom rozhraní aplikácie budeme pohybovať motorom, nastavovať rôzny počet impulzov (veľkosť jedného kroku) a pozorovať pohyb krokového motora.
- B. **Výstup** - Krokový motor sa bude pohybovať presne podľa príkazov používateľa.

6.6 Zvolenie uhlovej jednotky

- A. **Scenár** - Ako uhlovú jednotku zvolíme stupne a spustíme sériu meraní pre určitý rozsah. Uhlovú jednotku následne zmeníme na gradiany a znovu spustíme sériu meraní pre ten istý rozsah. V uložených dátach potom porovnáme číselné hodnoty uhlových jednotiek.

- B. Výstup** - V uložených dátach sa uhlové jednotky budú líšiť podľa vzťahu stupeň / gradián.

6.7 Nastavenie expozičnej doby spektrometra

- A. Scenár** - Pri meraní aktuálneho spektra budeme meniť hodnotu expozičnej doby a pozorovať zmeny v grafe.
- B. Výstup** - Graf bude v reálnom čase reagovať na zmeny expozičnej doby.

6.8 Nastavenie parametrov a poznámky pre sériu meraní a kontrola konfiguračného súboru

- A. Scenár** - Nastavíme všetky parametre a vyplníme poznámku pre sériu meraní, ktorú následne spustíme. Po skončení merania skontrolujeme konfiguračný súbor.
- B. Výstup** - Všetky údaje v konfiguračnom súbore sa budú zhodovať so zadanými parametrami pre danú sériu meraní.

6.9 Bezpečnostné tlačidlo STOP

- A. Scenár** - Spustíme sériu meraní s ľubovoľnými parametrami. Počas merania stlačíme tlačidlo STOP.
- B. Výstup** - Séria meraní ako aj pohyb krokového motora sa okamžite zastavia. Výstupné dáta zo série meraní sa neuložia.

6.10 Kalibrácia a aktuálny uhol

- A. Scenár** - Urobíme kalibráciu podľa definovaného postupu (posunieme motor na nulovú pozíciu ktorú potvrdíme. Následne motor posunieme na ľubovoľnú inú koncovú pozíciu, ktorú taktiež potvrdíme.). Po úspešnej kalibrácii posúvame motor a sledujeme hodnoty aktuálneho uhla.
- B. Výstup** - Po úspešnej kalibrácii sa zobrazí hodnota aktuálneho uhla, ktorá sa po posúvaní motora mení.

6.11 Posúvanie na zadaný uhol

- A. Scenár** - Po vykonaní úspešnej kalibrácie zadáme uhol, na ktorý chceme aby sa motor automaticky posunul.
- B. Výstup** - Motor sa automaticky posunie na najbližšiu možnú polohu (v rámci možností krokového motora) zodpovedajúcu zadanému uhlu.