

# Návrh

Meranie fyzikálnych veličín pomocou meracieho prístroja

Dáša Keszeghová, Anna Rebeka Sojka, Matúš Gál, Jakub Švorc  
10-31-2019

# Obsah

1.	Špecifikácia vonkajších interfejsov.....	2
1.1.	Komunikácia s inými zariadeniami.....	2
1.1.1.	Pripojenie prístroja .....	2
1.1.2.	Komunikácia prístroja s počítačom.....	2
1.1.3.	Používané technológie .....	3
2.	Formáty súborov.....	4
2.1.	Ukladanie meraní.....	4
2.2.	Exportovanie meraní.....	4
3.	Návrh používateľského rozhrania .....	5
4.	Návrh implementácie.....	7
4.0.	Implementácia vnútorného mechanizmu - popis.....	7
4.1.	Implementácia grafiky – popis.....	7
4.2.	Component diagram .....	9
4.3.	Class diagram .....	10
4.4.	Sequence diagram.....	11
4.5.	Popis jednotlivých súborov .....	12

# 1. Špecifikácia vonkajších interfejsov

## 1.1. Komunikácia s inými zariadeniami

### 1.1.1. Pripojenie prístroja

Aplikácia komunikuje s multifunkčným meracím prístrojom (MT-1820), ktorý má užívateľ pripojený k počítaču pomocou USB kábla a cez USB port. Aplikácia si bude automaticky detegovať port, v prípade keď sa nepodarí automaticky detegovať port, aplikácia požiada používateľa o zadanie portu ručne.

Prístroj sa pripája pomocou COM portov. V prípade, že COM port nie je nainštalovaný aplikácia si doinštaluje potrebný COM port pomocou inštalera (CP210x\_VCP\_Windows\CP210xVCPInstaller\_x64.exe) .

### 1.1.2. Komunikácia prístroja s počítačom

Na to aby prístroj mohol komunikovať s počítačom a taktiež aj s aplikáciou musí byť v móde REL/RS232 („Relative Value Measurement mode“), tento mód sa dosiahne podržaním tlačidla RS232/REL na prístroji. V prípade, že tento mód nie je zapnutý počítač nevie komunikovať s prístrojom.

Prístroj komunikuje posielaním 14 bitov pričom každý bit má svoj vlastný význam: (význam jednotlivých bitov je možné nájsť na nasledujúcom odkaze - <https://github.com/drahoslavzan/Proskit-MT1820-Probe/blob/master/proskit.cc>)

Význam jednotlivých bitov:

- 00 - znamienko (+/-)
- 01 – 04 hodnota, ktorú prístroj nameral
- 05 – medzera
- 06 – desatinná čiarka (za ktorú cifru z nameranej hodnoty sa má napísať desatinná čiarka)
- 07 – 08 – príznaky
- 09 – 10 – jednotka, v ktorých prístroj meria (fyzikálna veličina)
- 11 – stupnica na dolnej časti displeja prístroja
- 12 – 13 – nový riadok

Jednotky, v ktorých prístroj meria (09-10 bit):

- UNIT\_hFE = 0x0010
- UNIT\_mV = 0x4080 (napätie vo voltoch)
- UNIT\_V = 0x0080 (napätie v milivoltoch)
- UNIT\_Ohm = 0x0020 (odpor v Ohmoch)
- UNIT\_kOhm = 0x2020 (odpor v kilo-Ohmoch)
- UNIT\_MOhm = 0x1020 (odpor v mega-Ohmoch)
- UNIT\_DIODE\_V = 0x0480
- UNIT\_F = 0x0004 (teplota v stupňoch Fahrenheit)
- UNIT\_Hz = 0x0008
- UNIT\_DUTY\_Hz = 0x0200
- UNIT\_C = 0x0002 (teplota v stupňoch Celzia)
- UNIT\_uA = 0x8040 (elektrický prúd v mikroampéroch)
- UNIT\_mA = 0x4040 (elektrický prúd v miliampéroch)
- UNIT\_A = 0x0040 (elektrický prúd v ampéroch)

### 1.1.3. Použité technológie

- Python – Jadro aplikácie bude napísané v tomto jazyku, kvôli veľkej podpore knižníc a modulov na riešenie projektu.
- wxPython – Modul na tvorbu interface-u a interaktívnej časti aplikácie. Nakoľko zabudovaný pythonovský modul tkInter nepodporuje čítanie obrazovky, je tento nástroj najvhodnejší, aj vzhľadom na fakt, že čítač obrazovky NVDA je napísaný aj pythone.
- NVDA – program na čítanie obsahu obrazovky. Interaktívna aplikácia ktorá tvorí zvukový výstup na opísanie označeného prvku obrazovky alebo na opísanie prvku, na ktorý ukazuje myš počítača. Program používajú žiaci, pre ktorých je výsledok projektu určený. Jeho používanie a prístup k nemu je zadarmo.
- JAWS – ďalší čítač obrazovky, ktorý je tiež používaný žiakmi avšak jeho licencia a prístup k nemu je platený.

## 2. Formáty súborov

Aplikácia bude pracovať s dvomi formátmi súborov:

### 2.1. Ukladanie meraní

Merania sa budú dať uložiť vo formáte .pickle, tieto súbory sa budú dať späťne otvoriť pomocou aplikácie.

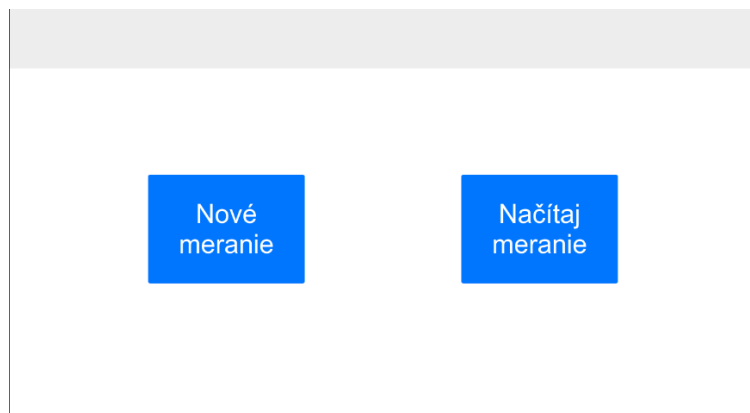
Pickle je modul používaný na serializáciu a deserializáciu pythonovských štruktúr. Pomocou tohto modulu jednoducho skonvertujeme celé meranie na prúd bajtov, ktorý sa dá jednoducho uložiť. Neskôr sa opäť vieme vrátiť do pôvodnej objektovej hierarchie. Na používanie tohto modulu netreba nič inštalovať, stačí použiť import z knižnice – „import pickle“. Na serializáciu budeme používať pickle.dumps() a deserializáciu pickle.loads().

### 2.2. Exportovanie meraní

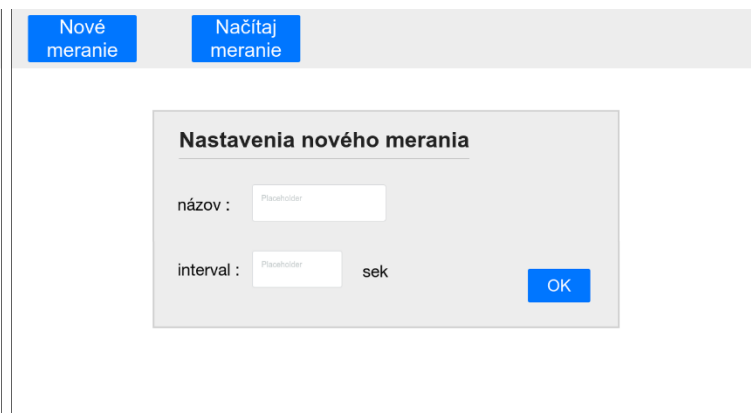
Žiaci si budú môcť exportovať namerané údaje vo forme tabuľky a grafu do Excelu vo formáte .xlsx.

Namerané údaje budú v Exceli zapísane v dvoch riadkoch, pričom prvý bude obsahovať čas a druhý nameranú hodnotu. Graf bude čiarový, na x-ovej osi bude čas a na y-ovej bude nameraná hodnota. Namerané dvojice hodnôt budú vykreslené na grafe a spojené spojovacou čiarou kontrastnej farby.

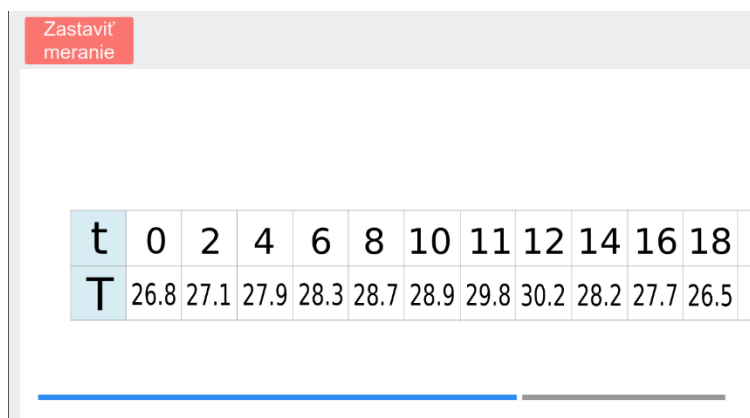
### 3. Návrh používateľského rozhrania



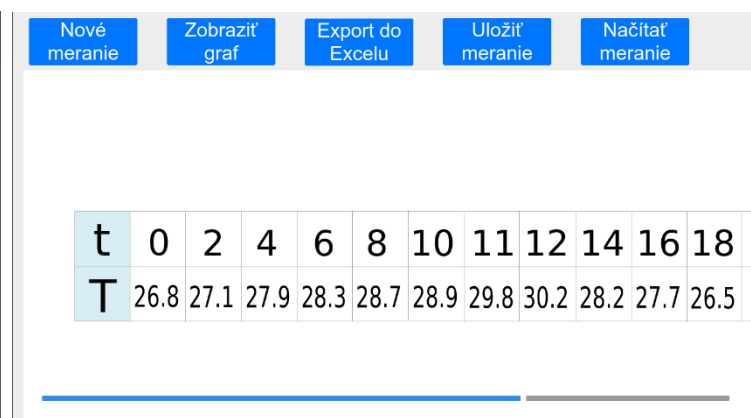
obr. 1 - štart programu



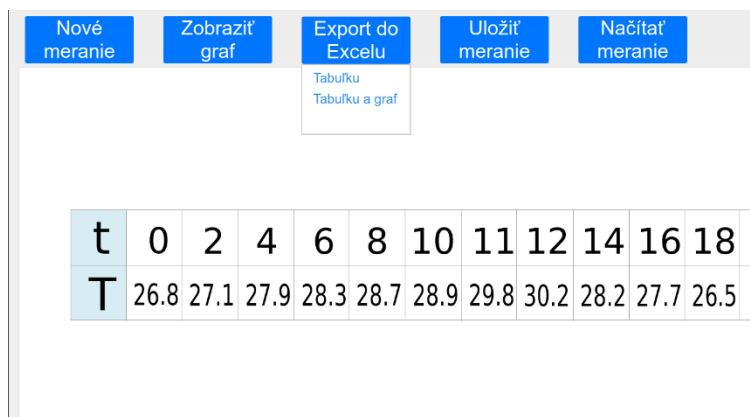
obr. 2 – nové meranie



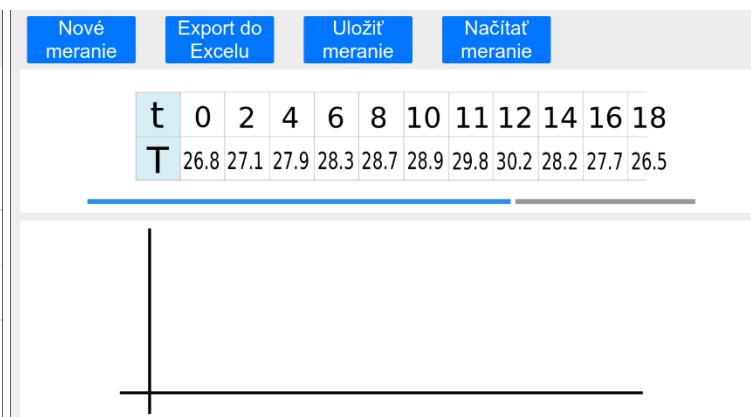
obr. 3 – počas merania



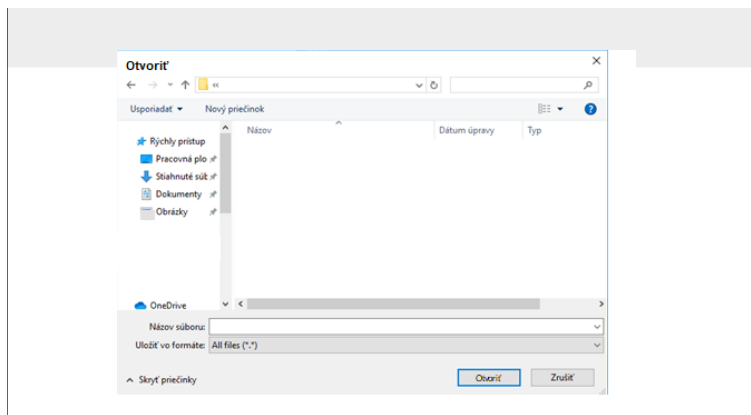
obr. 4 – po meraní / načítané meranie



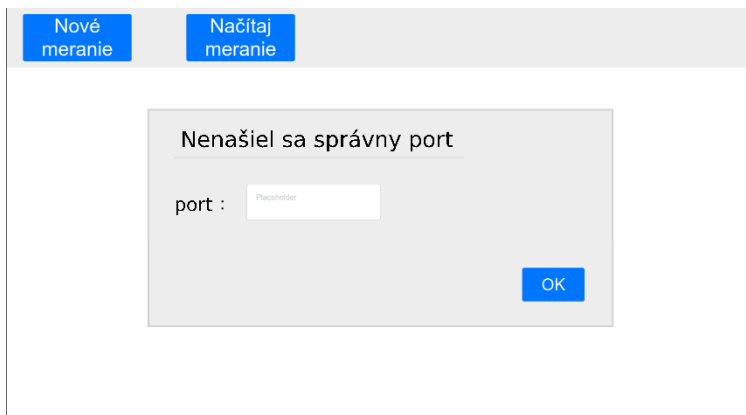
obr. 5 – možnosti exportu



obr. 6 – zobrazenie grafu



**obr. 7 – načítanie merania**



**obr. 9 – chybové hlásenie o porte**

## 4. Návrh implementácie

### 4.0. Implementácia vnútorného mechanizmu - popis

O správny chod aplikácie sa budú starať nástroje : **Handler**, **Parser** a **Data Manager**.

- **Handler** (naprogramovaný v súbore handler.py) spracúva všetky udalosti prichádzajúce z grafickej plochy, teda čo sa má vykonať : po kliknutí na konkrétne tlačidlo, po stlačení klávesovej skratky apod.  
Teda udalosť na tlačidlo volá Handler a pomocou dohodnutého kódu informuje o aké tlačidlo ide. Rovnako to je aj s udalosťou stlačenia klávesovej skratky.  
Ďalej má na starosti vyhľadanie portu. V prípade, že sa to nepodarí, podáva hlásenie a pýta správne číslo portu od používateľa.
- **Parser** spracováva a prekladá prichádzajúce dáta z prístroja. Tie posiela priamo Data Manager-u alebo prípadne Handler-u podľa situácie.
- **Data Manager** si pamätá dáta a prijíma pokyny od Handler-a. Namerané hodnoty ukladá, exportuje, ponúka na zobrazenie apod.

### 4.1. Implementácia grafiky – popis

Grafická časť aplikácie bude rozdelená do ôsmich súborov nasledovne :

- button\_panel.py
- start\_up.py
- graph\_panel.py
- input\_panel.py
- messagebox.py
- panel\_handler.py
- table\_panel.py
- window.py

Každý súbor reprezentuje zobrazenie niektorej časti okna, ktorá je rozdelená na panely. Panely sa zobrazujú v závislosti od stavu okna a kliknutých tlačidiel.

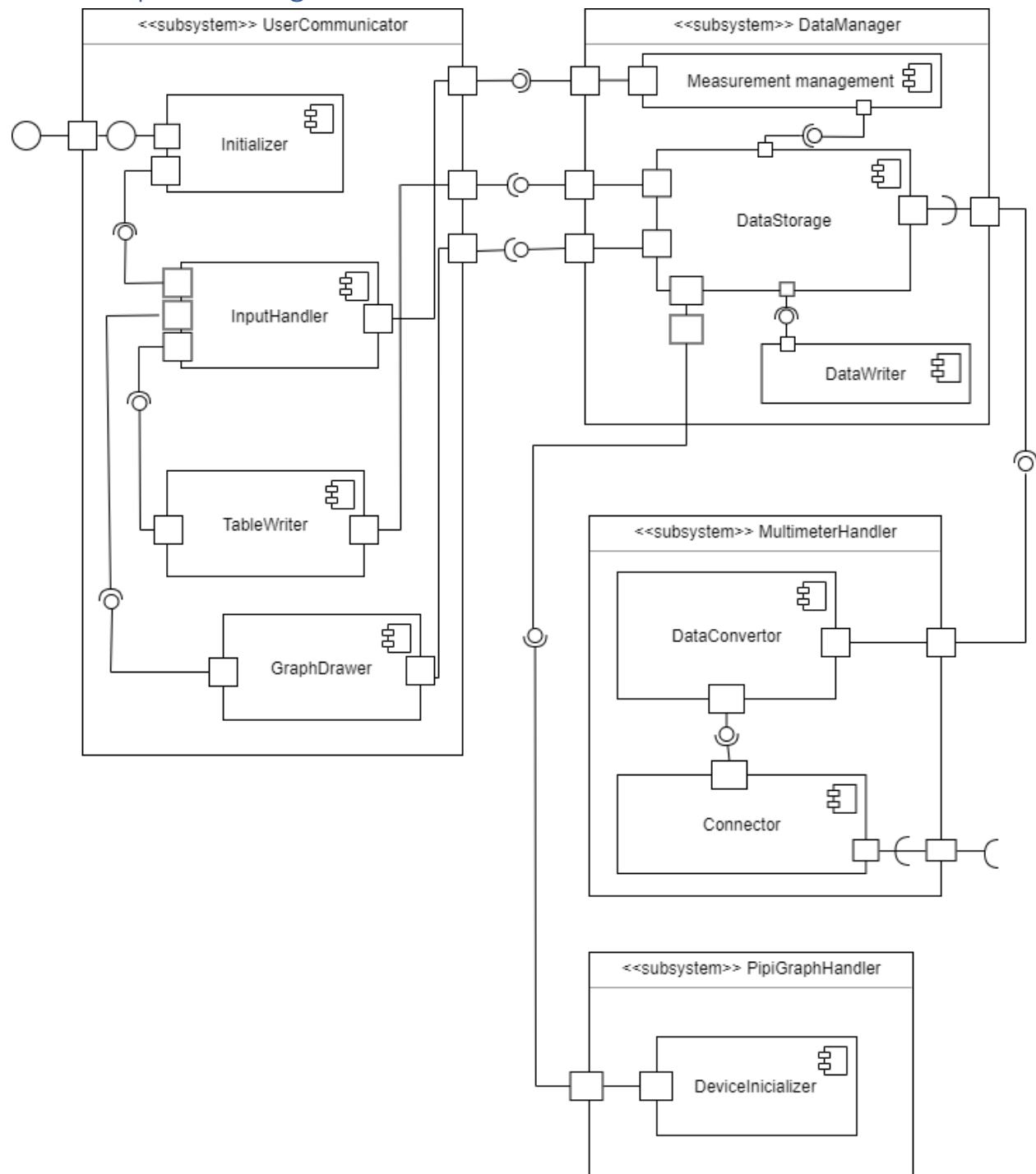
Samotná vykreslovacia plocha – **MainWindow**, využíva funkcie (dedí od) triedy **Frame**. Rozloženie a vyobrazené prvky v danom frame sa prispôbia akciám užívateľa.



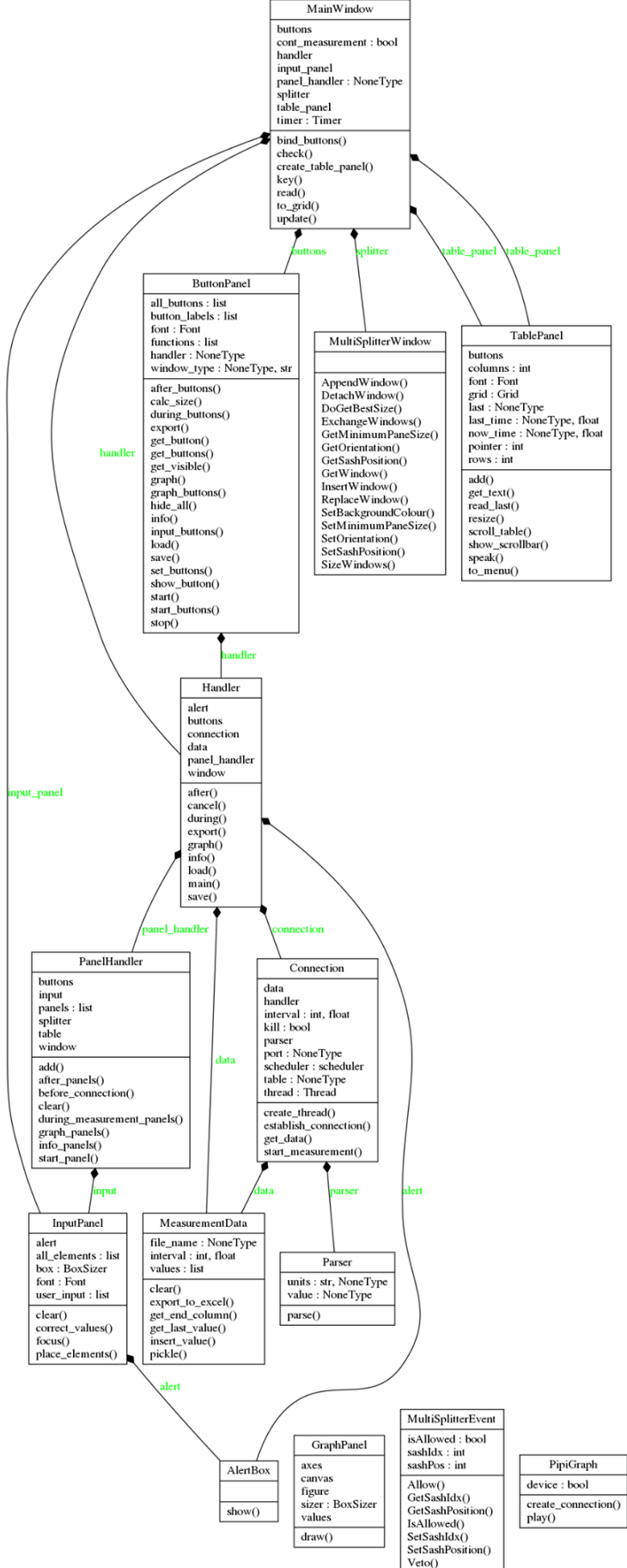
Inicializuje sa jedno okno, ktoré funguje počas celého behu aplikácie. Toto okno sa predáva ako argument, pri výtvarní panelov a ich zobrazovania a iba sa prekresľuje obsah tohto okna. Tým pádom sa zabráni duplikovaniu kódu a pre každú obrazovku sa zachová rovnaká funkcionálna okna.

Panely sa zobrazujú do Frame-u pomocou triedy typu **Splitter**. Ktorá Frame **horizontálne** rozdelí na viaceré Panely podľa potreby (napr. niekedy sa graf zobrazuje, inokedy nie).

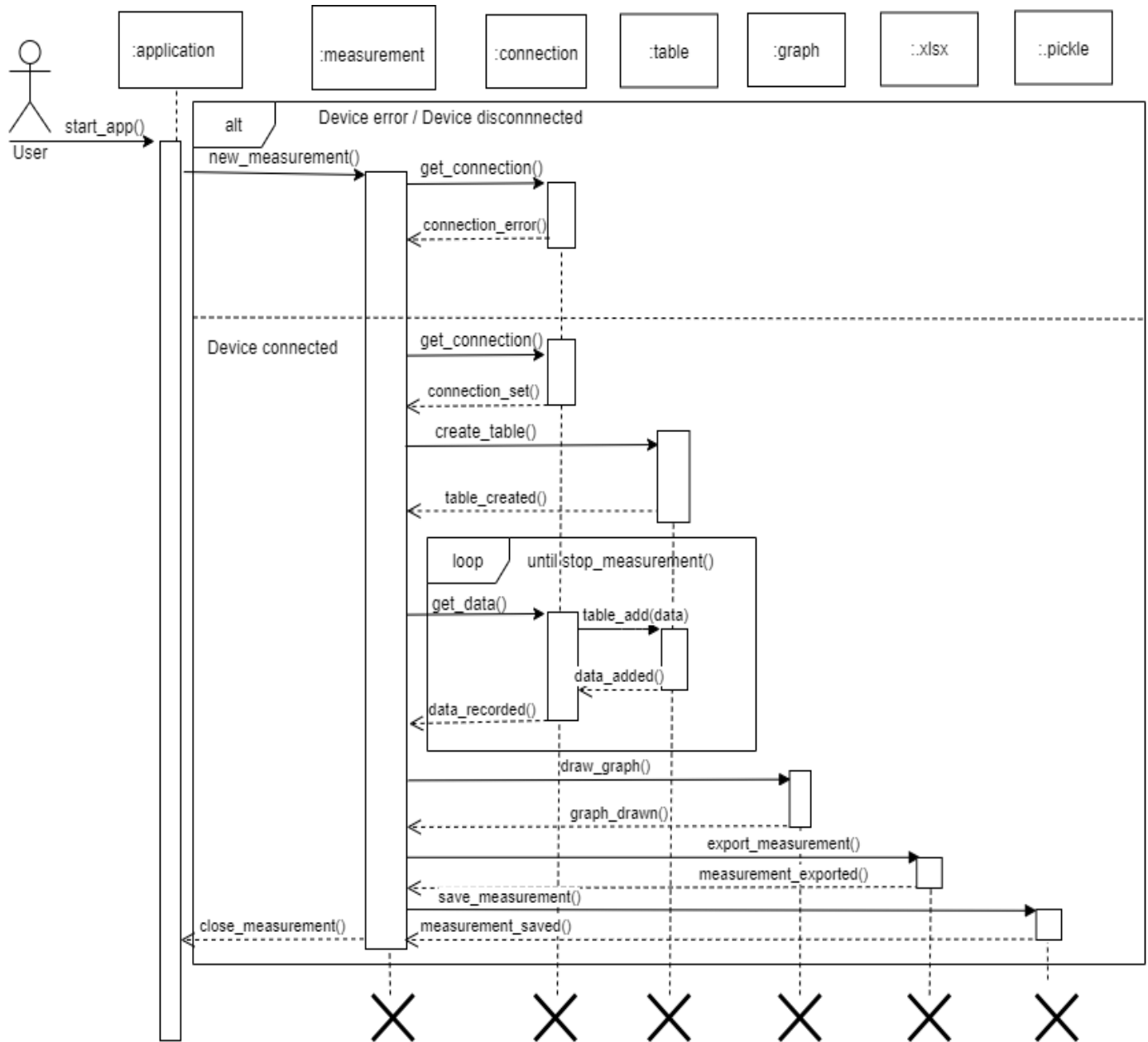
## 4.2. Component diagram



### 4.3. Class diagram



#### 4.4. Sequence diagram



## 4.5. Popis jednotlivých súborov

- ***\_\_init\_\_.py***

Tento súbor je inicializačný súbor. Slúži ako „entry point“ pri kompilovaní do spúšťateľného exe súboru. Vytvára sa v ňom hlavné okno a inicializuje sa v ňom *handler* pre prácu a spracovávanie dát z prístroja.

- ***button\_panel.py***

V súbore sa vytvárajú všetky tlačidlá potrebné počas celého behu aplikácie, ktoré sa neskôr priradia oknu. Teda inicializujú sa iba raz a neskôr sa len prispôsobuje ich pozícia a viditeľnosť.

Obsahuje rôzne funkcie, ktoré nám zabezpečujú zobrazovanie a skrývanie tlačidiel, ich komunikáciu s handlerom a iné pomocné funkcie, ktoré selektujú konkrétne tlačidlá. Celú túto funkcionality zastrešuje trieda *ButtonPanel*.

- ***connection.py***

Obsahuje triedu *Connection*, ktorá vytvára spojenie s prístrojom, číta údaje z prístroja a spravuje vlákno, v ktorom beží samotná komunikácia prístroja, teda meranie. Dáta z prístroja posielajú Parseru, ktorý bity premení do žiakov známej a zrozumiteľnej podoby. Tieto údaje uskladňuje trieda *MeasurementData* a zaznamenané údaje sú pridávané do tabuľky (*TablePanel*).

- ***data\_parser.py***

V tomto súbore trieda *Parser* zastrešuje premenu bitov na hodnotu viditeľnú aj na meracom prístroji. Z prúdu bitov je vyčítaná nameraná hodnota a veličina. Keďže jednotky merania sa môžu počas merania meniť, nameraná hodnota je vždy prepočítaná na základné jednotky. Význam jednotlivých bitov takéhoto bitového prúdu je popísaný v časti 1.1.2.

Táto trieda namerané hodnoty vráti triede *Connection*, ktorá ich následne pošle objektu, ktorý obsluhuje tabuľku a doplní ich tam.

- ***graph\_panel.py***

V súbore je trieda *GraphPanel*, ktorá vykresľuje z nameraných hodnôt čiarový graf. Namerané hodnoty sa predávajú pri inicializácii triedy *GraphPanel*. Graf je umiestnený pod tabuľkou.

- ***handler.py***

Trieda *Handler* v tomto súbore obsluhuje dopyty od používateľa - klávesové udalosti, stláčanie tlačidiel a klikanie myšou. Okrem toho, má na starosti aj volanie metód, ktoré prispôsobujú obsah okna. Reaguje aj na pripojenie zariadenia, či je pripojené, či sa s ním dá komunikovať a či sa na danom porte nachádza.

Spravuje aj súbory – ich ukladanie, načítavanie a export údajov do excelu.

Taktiež využíva triedu *AlertBox*, pomocou ktorej informuje používateľa o prípadných chybách a vykonaných úkonoch.

- ***input\_panel.py***

Reprezentuje úvodnú obrazovku. Trieda *InputPanel* vytvára 2 prázdne Textfieldy, do ktorých užívateľ zadá meno súboru a interval, v ktorom sa majú hodnoty zaznamenávať. Následne po potvrdení údajov používateľom kontroluje dané vstupy – teda či nie je číslo záporné a zadané korektne, či je zadaný názov súboru. V prípade zlých vstupov zobrazí okno so správou o chybe.

- ***measurement\_data.py***

Obsahuje triedu *MeasurementData*, ktorá spravuje namerané údaje, stará sa o exportovanie nameraných údajov vo forme tabuľky a grafu do Excelu a ukladá meranie do formátu .pickle, ktorý je používaný pri načítavaní už vykonaných meraní v aplikácii.

- ***messagebox.py***

Obsahuje triedu *AlertBox*, ktorá má za úlohu vytvoriť vyskakovanie okno s upozornením. Toto dialógové okno je využívané pri chybových hláškach alebo pri úspešne vykonaných udalostiach ako napr. exportovanie meranie.

- ***panel\_handler.py***

Obsahuje triedu *PanelHandler*, ktorá obsluhuje výmenu panelov, keď je potrebné zmeniť rozloženie okna napríklad pri kliknutí na niektoré tlačidlo alebo pri použití niektorej klávesovej skratky.

Celá aplikácia beží v jednom frame, na základe stavu, v ktorom sa aplikácia nachádza (napr. počas merania, po meraní, s vykreslením grafom, ...) sú panely pridávané alebo odoberané z hlavného framu.

- ***pipigraph.py***

Obsahuje triedu *PipiGraph*, ktorá pracuje s prístojnom „ππgraf“. Trieda *PipiGraph* je inicializovaná s nameranými hodnotami z triedy *MeasurementData* a vytvára spojenie s prístrojom. Po úspešnom nadviazaní spojenia s prístrojom vydáva zvuku vo forme pípania.

Vydávanie zvuku funguje nasledovne: Posuvným potenciometrom sa používateľ virtuálne posúva po čiarovom grafe z nameraných hodnôt. Na základe pozície posuvného potenciometra sa určí aká x-ová súradnica grafu predstavuje túto hodnotu a následne trieda *PipiGraph* vydá zvuk zodpovedajúci y-ovej súradnici príslušnej pre danú x-ovú súradnicu.

Výška tónu sa mení na základe hodnoty aké je reprezentovaná v grafe. Čím vyššia hodnota na vstupe príde, tým vyšší bude tón.

- ***splitter.py***

Tento súbor zobrazuje zmodifikovanú originálnu verziu wx.MultiSplitterWindo. V súbore sa nachádza trieda *MultiSplitterWindow*. Táto trieda zaisťuje rozmiestňovanie panelov v okne. Taktiež sa stará a obsluhuje vymieňanie panelov, ich zobrazovanie a skrývanie

podľa toho, v akom stave merania sa používateľ nachádza a podľa toho, aké akcie používateľ robí.

- ***table\_panel.py***

Obsahuje triedu *TablePanel*, ktorá vytvára dvojriadkovú tabuľku a umožňuje ovládanie tabuľky. Zaobstaráva pridávanie hodnôt do tabuľky, posúvanie (scrollovanie) tabuľky, zmenu veľkosti tabuľky. Taktiež, má metódu, ktorá umožňuje čítanie hodnôt v tabuľke, táto metóda potrebuje k fungovaniu internet. Nakoľko NVDA screen reader nevie čítať tabuľku, tak čítanie je zabezpečené pomocou tejto metódy.

- ***window.py***

Trieda *MainWindow* v tomto súbore predstavuje základné hlavné okno, do ktoré sa zobrazujú panely a všetky ovládacie prvky. Toto okno existuje počas celého behu aplikácie, je iba jedno a toto okno sa ďalej predáva ako argument na zobrazovanie panelov pomocou splitter-a. Stará sa aj o vytvorenie klávesových skratiek a ich bindovanie na príslušné funkcie.