Návrh

Meranie fyzikálnych veličín pomocou meracieho prístroja

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

Obsah

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 ci ho 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 a však jeho licencia a prístup k nemu je platený.

2. Formáty súborov

Aplikácia bude pracovať s dvomi formátmi súborov:

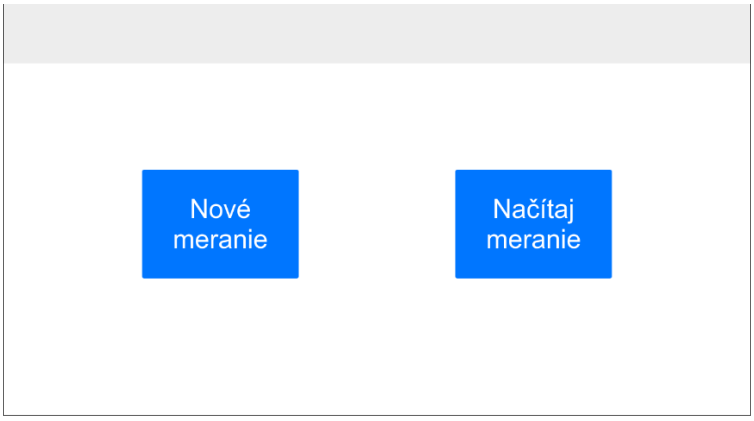
2.1. Ukladanie meraní

Merania sa budú dať uložiť vo formáte .pickle, tieto súbory sa budú dať spätne 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().

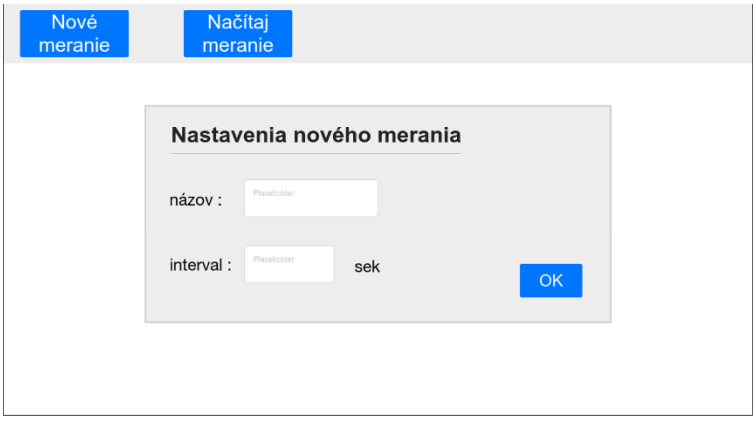
2.2. Exportovanie meraní

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.

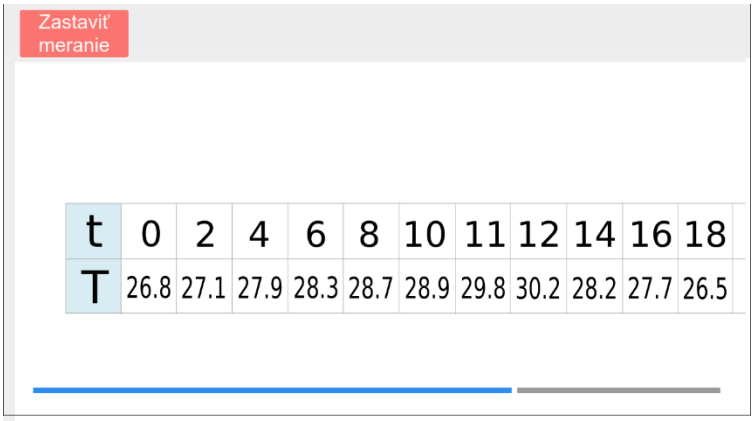
3. Návrh používateľského rozhrania



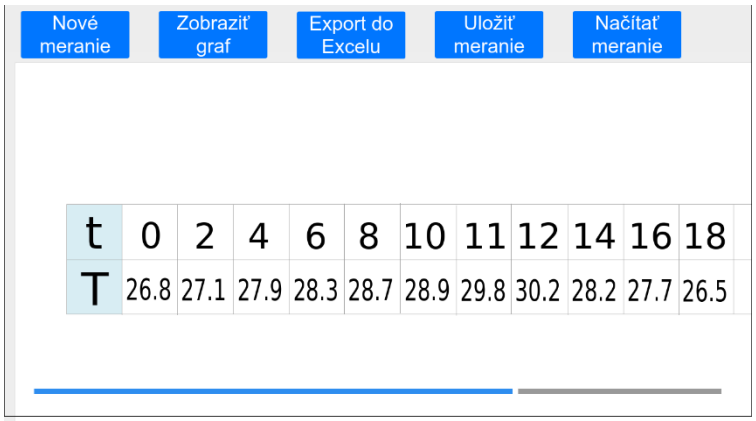
obr. 1 - štart programu



obr. 2 – nové meranie

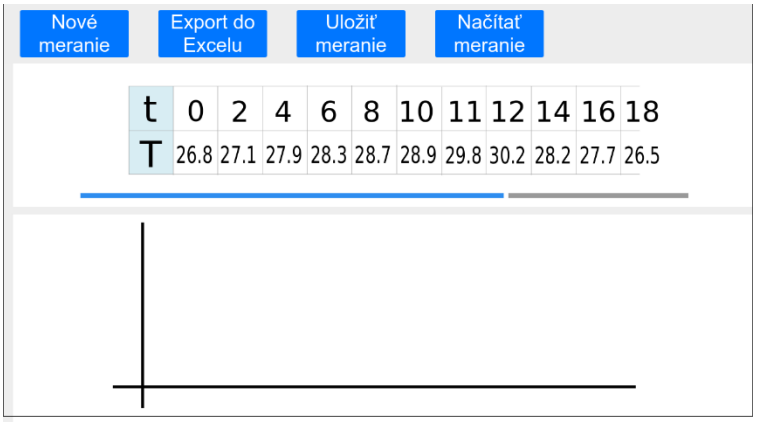


obr. 3 – počas merania

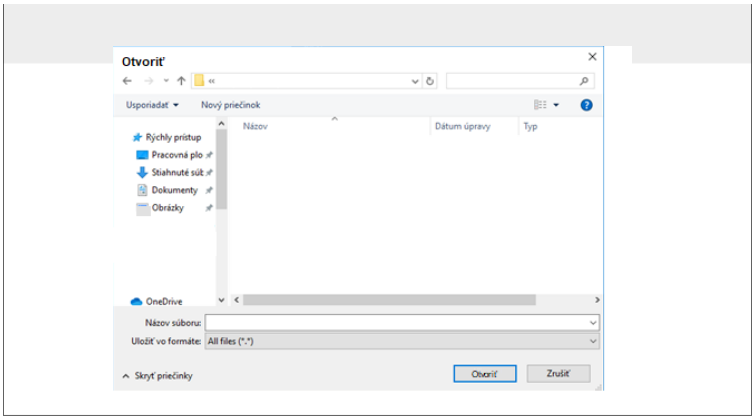


obr. 4 – po meraní / načítané meranie

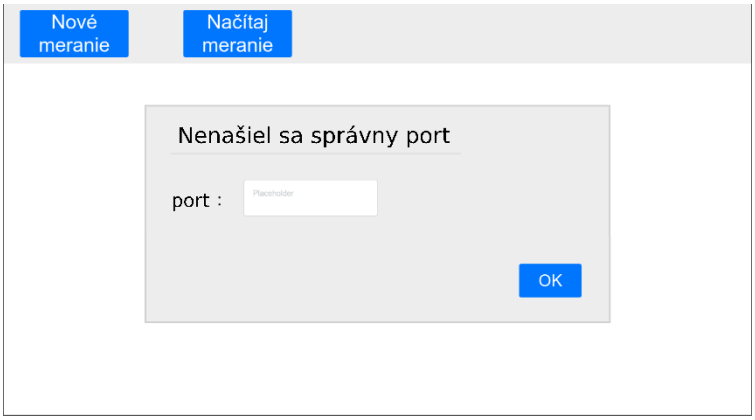
obr. 5 – možnosti exportu



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 tlačidle 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 aplikície bude rozdelená do ôsmych súborov nasledovne :

- button_panel.py
- start_up.py
- graph_panel.py
- input_panel.py
- messagebox.py
- panel_handler.py
- table_panel.py
- window.py

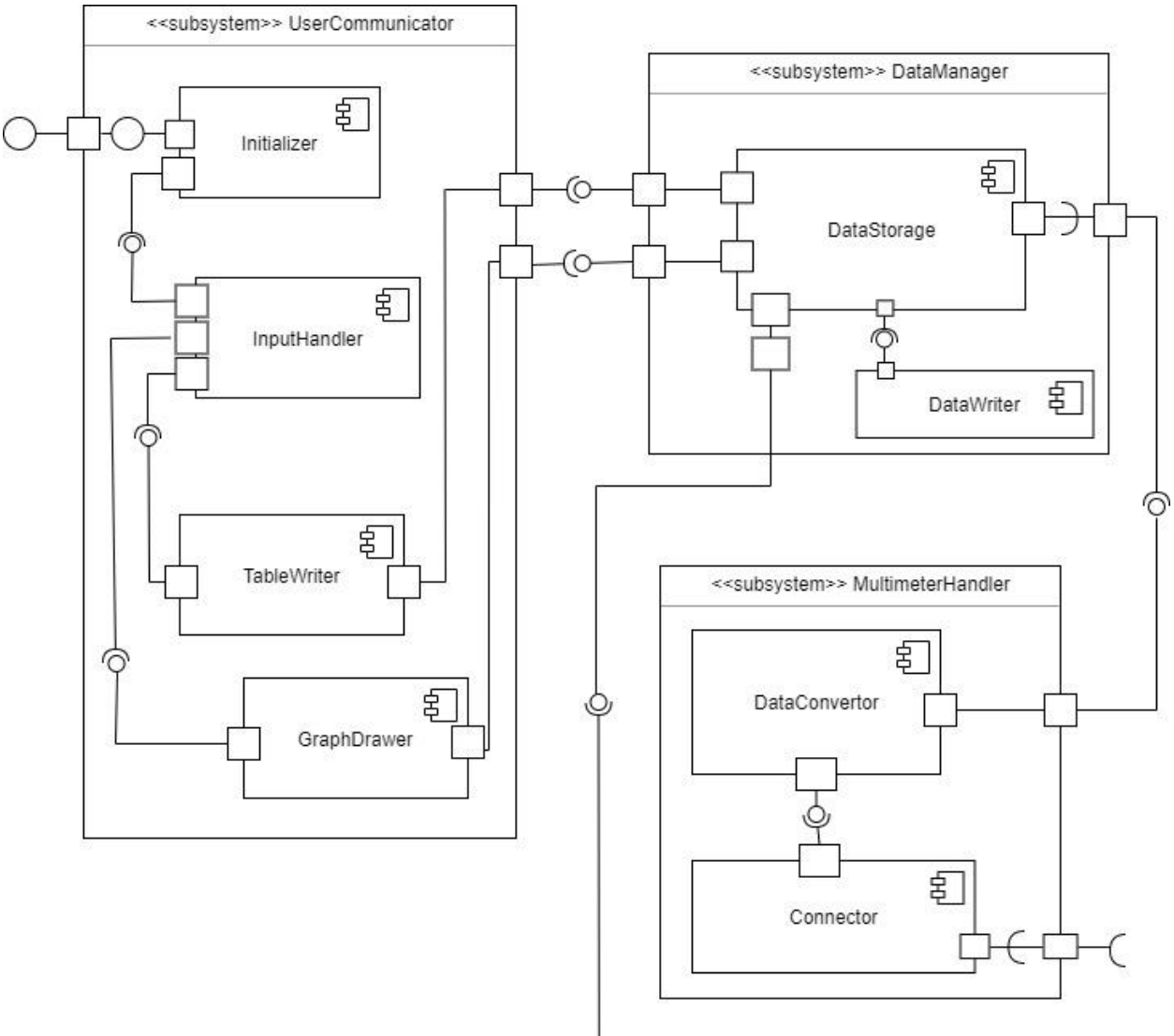
Každý súbor reprezentuje zobrazenie niektorej časti okna, ktorá je rozdelená na panely. Panely sa zobrazujú v závislosti od stavu okna a kliknutých tlačidiel.

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

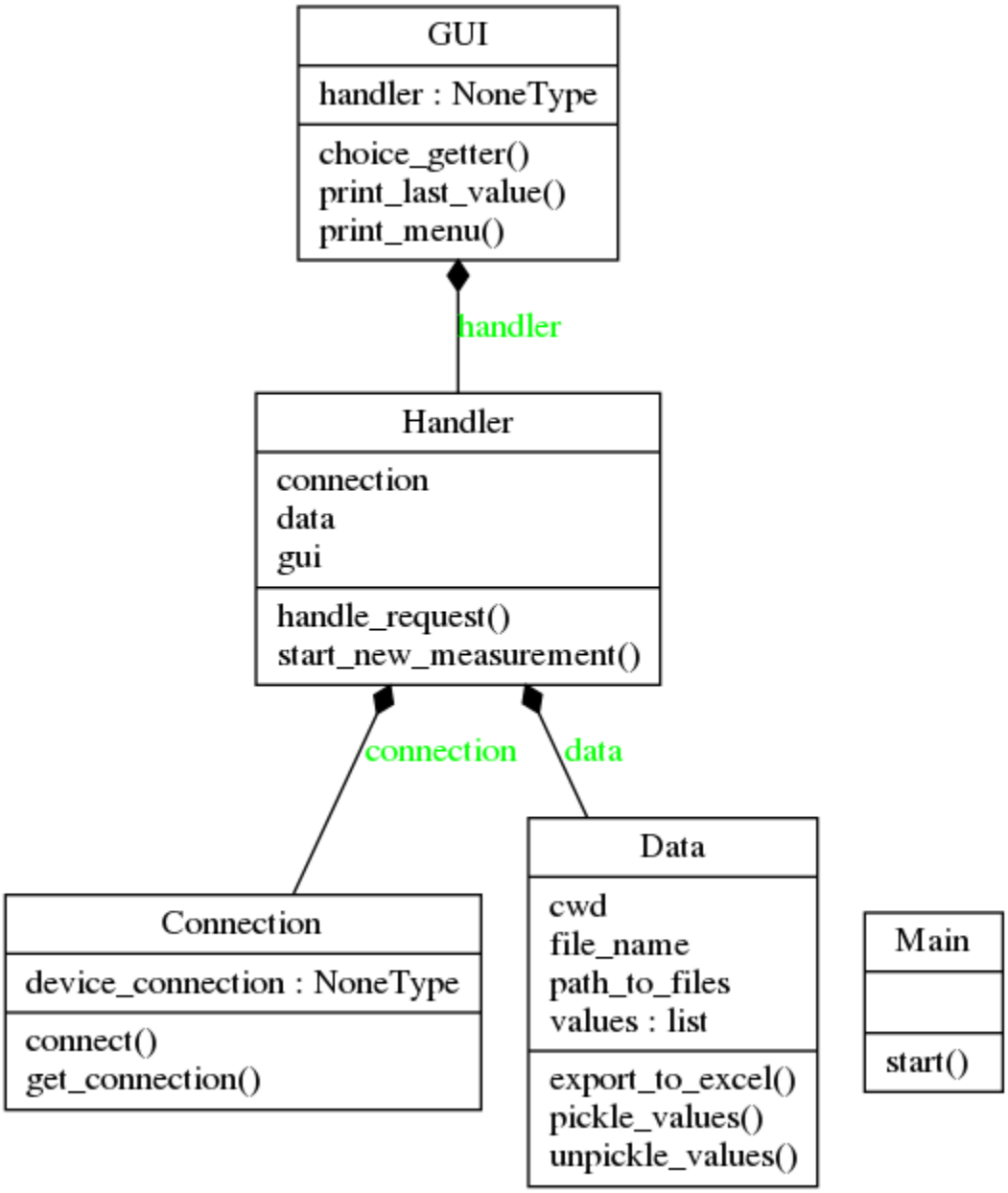
Inicializuje sa jedno okno, ktoré funguje počas celého behu aplikácie. Toto okno sa predáva ako argument, pri vytvarní panelov a ich zobrazovania a iba sa prekresluje obsah tohto okna. Tým sa zabraní duplikovaniu kódu a pre každú obrazovku sa zachová rovnaká funkcionalita okna.

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).

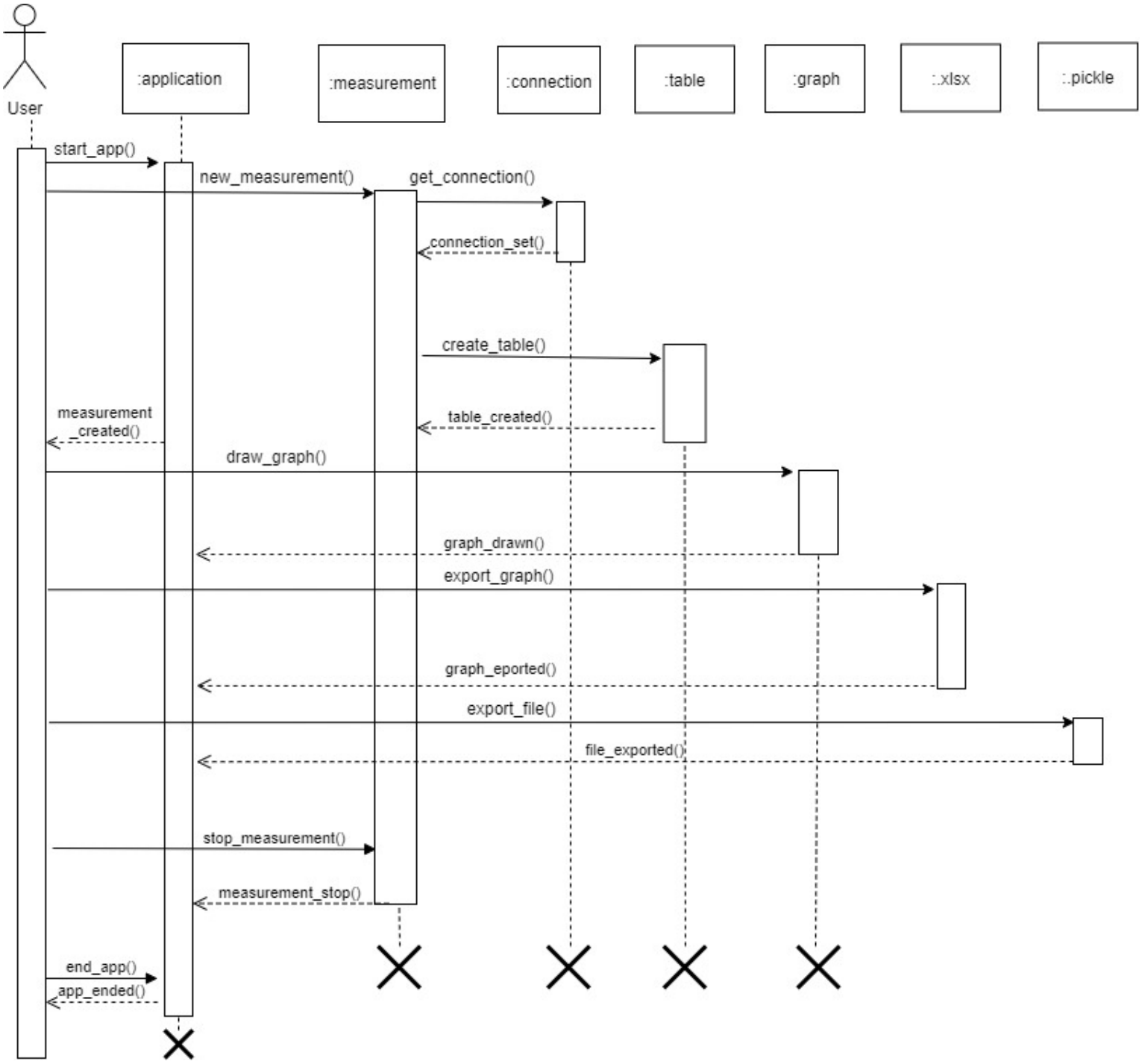
4.2. Component diagram



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 súšťateľného 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 tlačidlá, ktoré sa neskôr priradia oknu – teda inicializujú sa iba raz a neskôr sa len prispôsobuje ich pozícia a viditeľnost.

Obsahuje rôzne funkcie, ktoré nám zabezpečujú zobrazovanie a skrývanie tlačidiel, ich komunikáciu s handlerom a iné pomocné funkcie, ktoré selektujú konkrétne tlačidlá.

Celú túto funkcionalitu zastrešuje trieda **ButtonPanel.**

connection.py

Obsahuje triedu Connection, ktorá obsluhuje pripojenie prístroja, čítanie údajov z prístroja a spravuje vlákno, v ktorom beží samotná komunikácia prístroja, teda meranie. Dáta z prístroja pošle Parseru, ktorý bity premení do žiakom známej a zrozumiteľnej podoby.

data_parser.py

Trieda Parser v tomto súbore zastrešuje premenu bitov na čísla v desiatkovej sústave. Význam jednotlivých bitov takéhoto bitového prúdu je popísanie v časti 1.1.2.

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

graph_panel.py

V súbore je trieda GraphPanel, ktorá pridelí tabuľke miesto v okne,spravuje a dopĺňa údaje, ktoré jej pošle Connection do tabuľky.

handler.py

Handler trieda v tomto súbore obsluhuje klávesové udalosti a tlačidlá, ktoré používateľ stlačí. 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.

input_panel.py

Reprezentuje úvodnú obrazovku. Trieda InputPanel vytvára 2 prázdne boxy, do ktorých užívateľ zadá meno súboru a interval, v ktorom sa majú hodnoty zanamenávať, 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

V súbore je triedea MeasurementData, ktorá sa stará o exportovanie nameraných údajov do excelovského súboru, vytvorenie grafu v ňom a tvorbu tabuľky.

messagebox.py

Obsahuje triedu AlertBox, má za úlohu vytvoriť vyskakovacie okno s upozornením. Táto trieda sa používa napríklad pri zle zadaných vstupných údajoch.

panel_handler.py

Treba v tomto súbore – PanelHandler, obsluhuje výmenu panelov, keď je potrebné zmeniť rozloženie okna napríklad pri kliknutí na niektoré tlačidlo alebo pri použití niektorej klávesovej skratky.

pipigraph.py

splitter.py

table_panel.py

window.py