

# 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 .....	11
4.3.	Class diagram .....	12
4.4.	Sequence diagram .....	13

# 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 späťne otvoriť pomocou aplikácie.

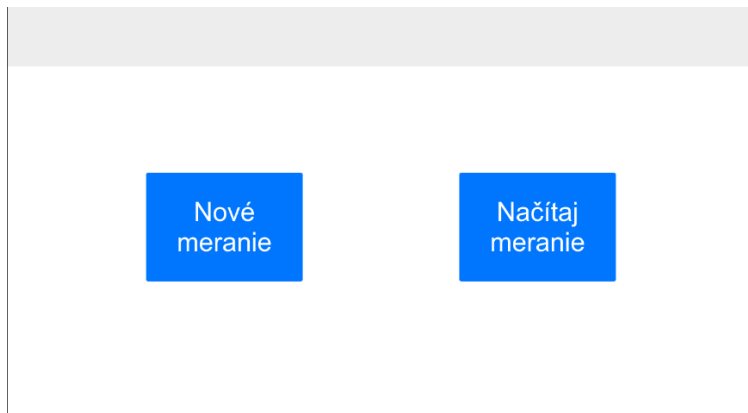
Pickle je modul používaný na serializáciu a deserializáciu pythonovských štruktúr. Pomocou tohto modulu jednoducho skonvertujeme celé meranie na prúd bajtov, ktorý sa dá jednoducho uložiť. Neskôr sa opäť vieme vrátiť do pôvodnej objektovej hierarchie. Na používanie tohto modulu netreba nič inštalovať,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

### 2.2. Exportovanie meraní

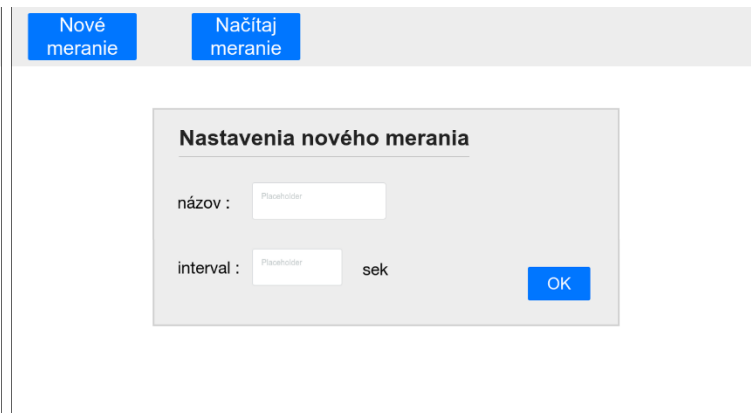
Žiaci si budú môcť exportovať namerané údaje vo forme tabuľky a grafu do Excelu vo formáte .xlsx.

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.

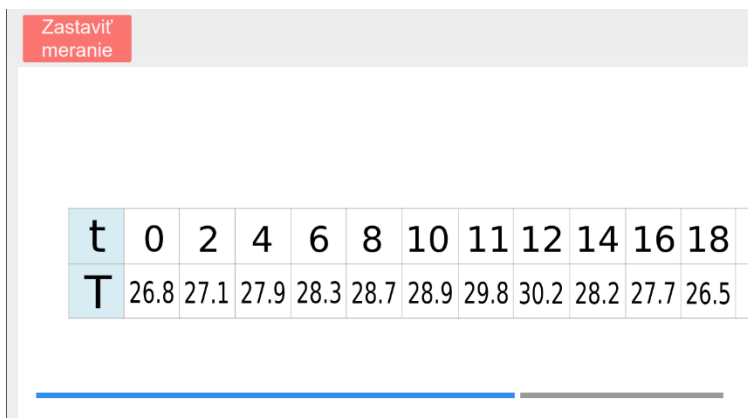
### 3. Návrh používateľského rozhrania



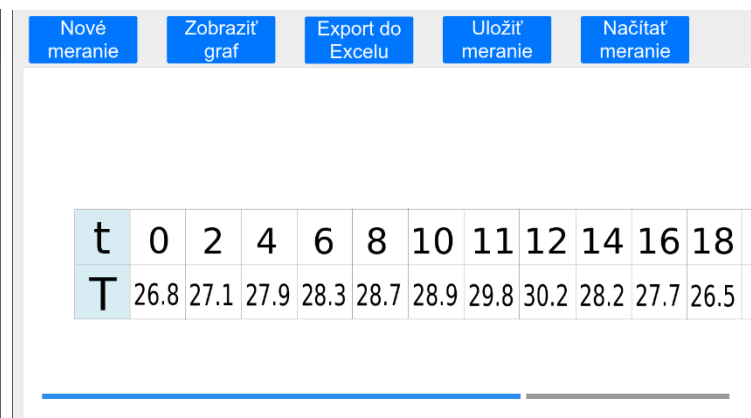
obr. 1 - štart programu



obr. 2 – nové meranie



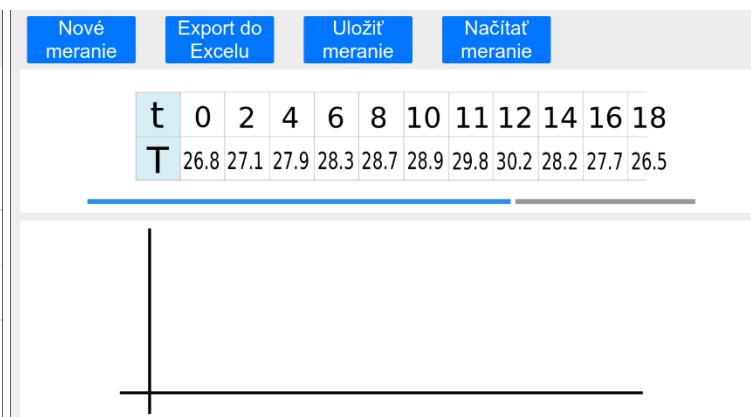
obr. 3 – počas merania



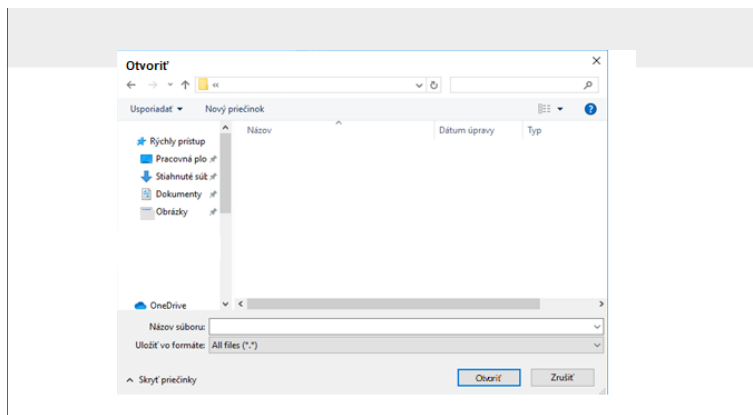
obr. 4 – po meraní / načítané meranie



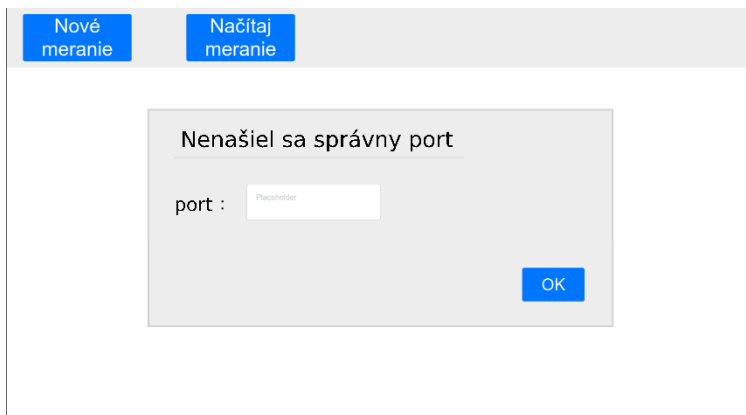
obr. 5 – možnosti exportu



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



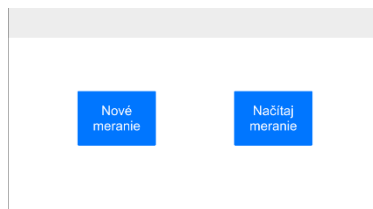
Každý súbor reprezentuje zobrazenie niektorej časti okna, ktorá je rozdelená na panely. Panely sa zobrazujú v závislosti od stavu okna a kliknutých tlačidiel.

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 vytváraní panelov a ich zobrazovania a iba sa prekresluje obsah tohto okna. Tým 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).

***start\_up.py***



Ekvivalentom prvej obrazovky z GUI.

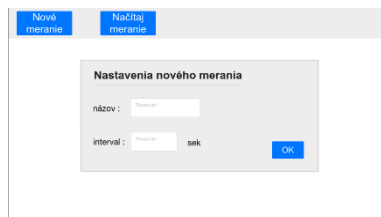
Obsahuje 1x triedu **Frame** a 1x triedu **Panel**.

Teda inicializuje Panel Buttonov, hoci niektoré Buttony sú zatiaľ z logických dôvodov skryté. Zobrazuje 2x Button.

Button „*Nové meranie*“ spúšťa súbor `set_measur.py` .

Button „*Načítaj meranie*“ otvára prieskumníka súborov a po výbere súboru spúšťa súbor `table_measur.py` .

***set\_measur.py***



Ekvivalentom druhej obrazovky z GUI.

V prípade, že sa nenašlo pripojené meracie zariadenie vyhadzuje chybové hlásenie s možnosťou dopísať správny port. Toto sa opakuje, kým sa nenájde správny port.

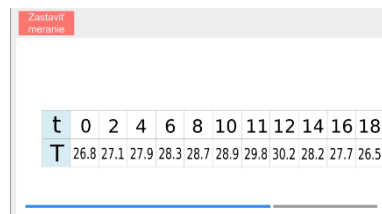
Obsahuje 1x triedu Frame a 2x triedu Panel.

Panel Buttonov sa dedí. Niektoré Buttony ostávajú skryté.

Zobrazuje Panel, ktorý ponúka vpisovacie okná pre názov, interval merania a tlačidlo OK.

Button „OK“ spúšťa súbor measur.py .

***measur.py***



t	0	2	4	6	8	10	11	12	14	16	18
T	26.8	27.1	27.9	28.3	28.7	28.9	29.8	30.2	28.2	27.7	26.5

Ekvivalentom tretej obrazovky z GUI.

Obsahuje 1x triedu Frame a 2x triedu Panel.

Panel Buttonov sa dedí. Zobrazuje sa len jeden Button pre zastavenie merania.

Druhý Panel vykresľuje postupne dopĺňajúcu sa tabuľku.

Button „Zastaviť meranie“ spúšťa súbor table\_measur.py .

***table\_measur.py***



t	0	2	4	6	8	10	11	12	14	16	18
T	26.8	27.1	27.9	28.3	28.7	28.9	29.8	30.2	28.2	27.7	26.5

Ekvivalentom štvrtej obrazovky z GUI.

Obsahuje 1x triedu Frame a 2x triedu Panel.

Panel Buttonov zedený. Už je viditeľných všetkých 5 Buttonov :

1. Button „Nové meranie“ spúšťa súbor set\_measur.py .
2. Button „Zobraziť graf“ spúšťa súbor tablegraph\_measur.py .
3. Button „Export do Excelu“ ponúka výber čo si používateľ žiada exportovať a následne posíla úlohu Handler-u, ktorý export vykoná.
4. Button „Uložiť meranie“ posíla úlohu Handler-u, ktorý vytvorí save file.
5. Button „Načítať meranie“ otvára prieskumníka súborov a po výbere súboru spúšťa súbor table\_measur.py .

***tablegraph\_measur.py***

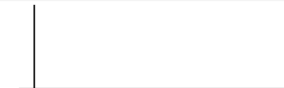
Nové meranie

Export do Excelu

Uložiť meranie

Načítať meranie

t	0	2	4	6	8	10	11	12	14	16	18
T	26.8	27.1	27.9	28.3	28.7	28.9	29.8	30.2	28.2	27.7	26.5



Ekvivalentom šiestej obrazovky z GUI.

Obsahuje 1x triedu Frame a 3x triedu Panel.

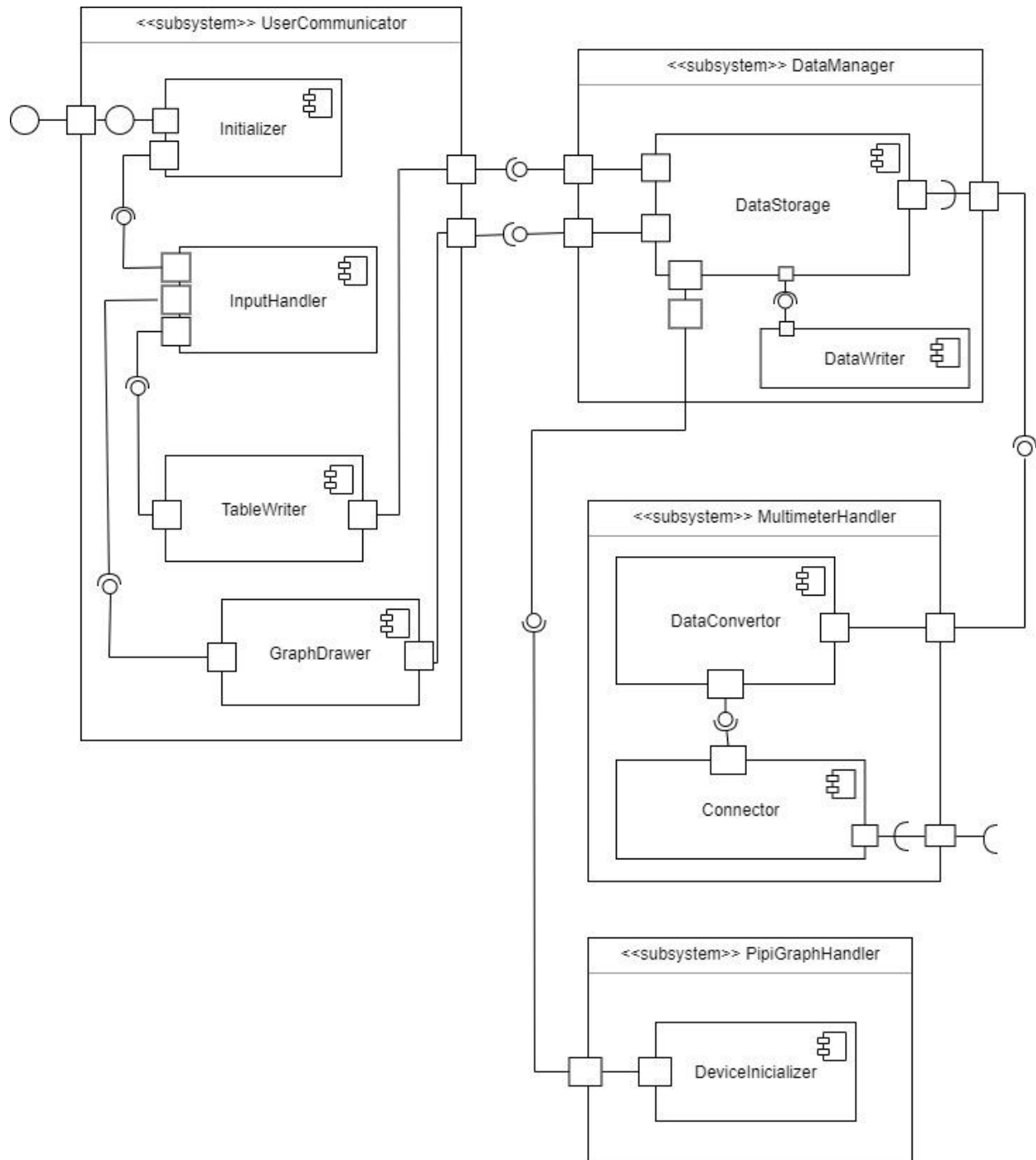
Panel Buttonov zedený. Button pre zobrazenie grafu je skryté

Panel Tabuľky je zdedený.

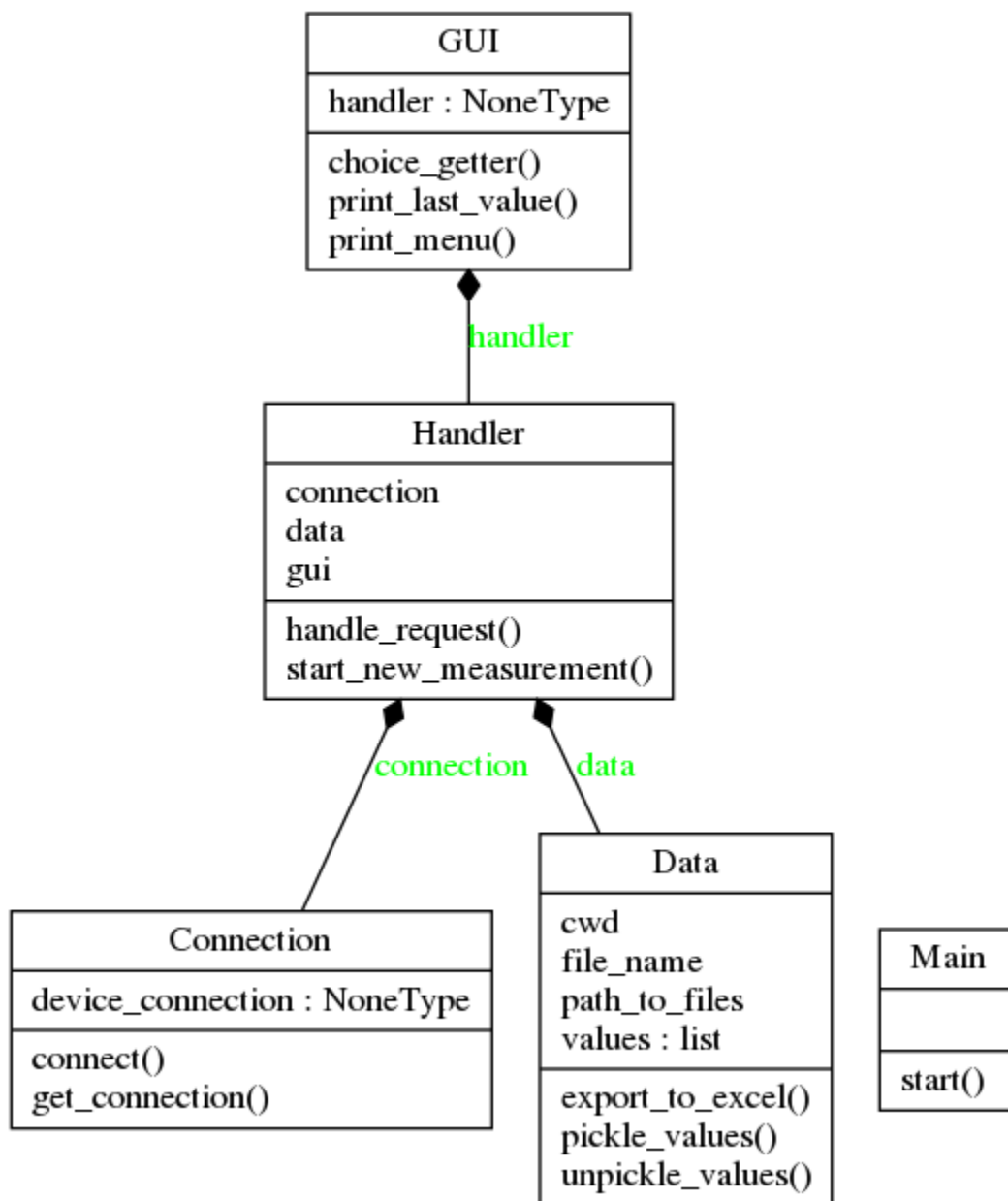
Pribúda Panel, ktorý obsahuje Graf.

Splitter rozdeľuje Frame horizontálne na 3 časti v poradí : 1. Button-y, 2. Tabuľka, 3. Graf

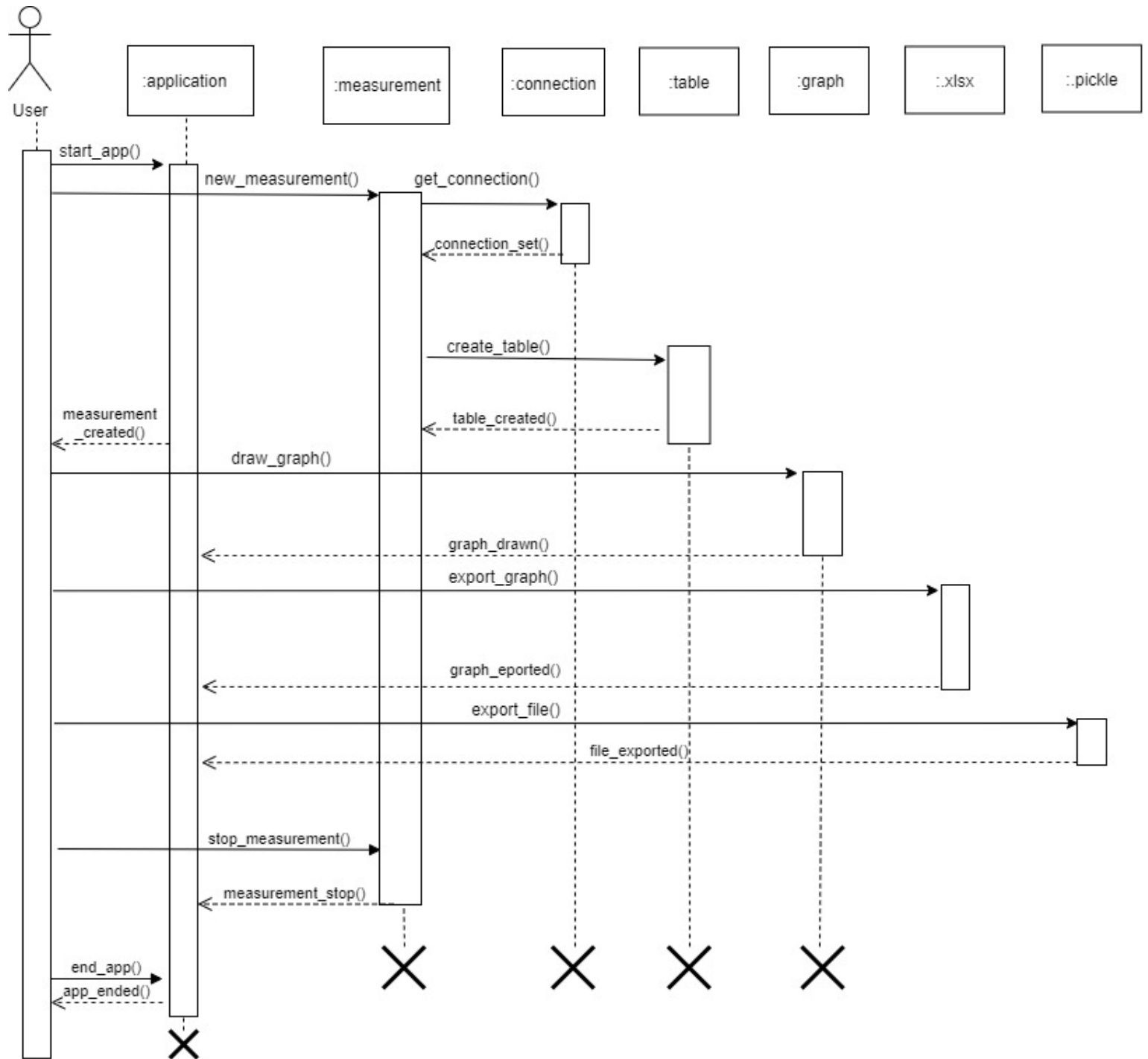
## 4.2. Component diagram



### 4.3. Class diagram



## 4.4. Sequence diagram



## 4.5. Popis jednotlivých súbtorov

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

***button\_panel.py***

***connection.py***

***data\_parser.py***

***graph\_panel.py***

***handler.py***

***input\_panel.py***

***measurement\_data.py***

***messagebox.py***

***panel\_handler.py***

***pipigraph.py***

***splitter.py***

***table\_panel.py***

***window.py***