**PythonPlotter**

**Front-end dokumentace**Chart, line chart

Description automatically generated



Jak již bylo zmíněno v obecném shrnutí přiloženém k projektu, aplikace slouží k vynesení vybraných hodnot do grafu/ů.

K vykreslení grafů a dalším funkcionalitám zvýrazněným v obrázku červeným podtržením, slouží knihovna: Matplotlib, dostupná jako balíček, nebo v podobně git adresáře. Tyto jednotlivé funkcionality umožňují zvětšit jednotlivé části grafu (symbol lupy), pohybovat se po grafu (symbol křížku), vrátit náhled grafu do původního stavu (symbol domečku), popřípadě odkrokovat jednotlivé změny dopředu či zpět (symbol šipek). Dále je zde možnost upravovat velikost grafů nebo mezery mezi jednotlivými grafy (symbol posuvníčku).

Aktuální stav aplikace je možno vyčíst ze „status labelu“ v dolní části okna. Na našem vzorovém obrázku reprezentovaném textem „Idling“.

Aplikace dále umožňuje zahájit či pozastavit vykreslování dat pomocí tlačítek „start“ a „stop“, odstranění/vyčištění hodnot všech grafů tlačítkem „clear“.

Program dále umožňuje upravovat viditelnost jednotlivých křivek v grafech pod menu View. Jednotlivé křivky mají své označení (X,Y,Z) a svoji barvu (červená, modrá, zelená). Pokud chce uživatel vidět pouze vybraná data, může ve zmíněném menu vypnout či zapnout vykreslování jednotlivých křivek. Možností „show full graph“ pak může vypnout omezení na počet bodů v jednotlivém grafu.

Tlačítka „record“ a „load“ slouží pro práci s daty, kdy po zmáčknutí tlačítka „record“ aplikace začne ukládat všechna data do textového souboru a tlačítkem „load“ můžeme z textových souborů načíst data zpět do grafu. Aby toto mohlo fungovat, je nutné vybrat v nastavení cesty k souborům, do kterých se má ukládat/ze kterých se má načítat. V nastavení dále lze upravit port ze kterého se čtou hodnoty, vyvolat prohledání všech dostupných portů a nastavit maximální počet bodů aktuálně zobrazených v grafu pro každou křivku.

O práci se sériovými porty se stará port\_reader.py. V nastavení programu je také možnost vybrat port "test", který generuje náhodné hodnoty a slouží jako simulace existujícího USB portu.

Hodnoty v textových souberech jsou uloženy ve formátu:# Plot:<plotnumber 0-2>;X:;Y:;Z: Každá z jednotlivých hodnot X,Y,Z představuje jeden z bodů křivek x,y,z.

Po stisknutí tlačítka „abort“ dojde k vypnutí aplikace.

**Back-end dokumentace**

V back-end dokumentaci rozebereme jednotlivé třídy a funkce popřípadě jednotlivé proměnné.

**Třídy**:

* Context – třída reprezentující aktuální stav aplikace. Jsou zde uložena všechna relevantní data, která jsou v průběhu upravována či nahrazována, dle aktuálního stavu aplikace. Funkce jednotlivých proměnných v této třídě je popsána v komentářích (viz obrázek)

Text

Description automatically generated

* SettingsWindow – v této třídě se nastavují nejdůležitější parametry aplikace uložené v Context. Vyvolání okna spustí funkce „settings\_widget\_show\_event(ctx)”, která reaguje na stisk tlačítka settings. V inicializaci třídy se z Context načtou hodnoty pro SettingsWindow, které se použijí jako výchozí. Všechny tyto položky se následně vloží do tabulky, která slouží jako náš „widget“. Součástí settings je i funkce „scan\_ports“, která s využitím „port\_reader.py“ načte všechny aktuálně dostupné sériové porty a zobrazí je v nabídce. Pokud byly v průběhu této funkce nalezeny nějaké porty, změní se text „stavového labelu“ na „devices were founded“ v opačném případě na „no devices were founded“. Další a poslední funkcí je „save\_and\_quit“, která slouží jako „destruktor“. V průběhu této funkce dojde k uložení všech nastavitelných hodnot a k ukončení okna.

Graphical user interface, application

Description automatically generated

**Funkce:**

* ***load\_data\_from\_file*** – na vstupu této funkce je předán kontext obshaující cestu k soubor, který má být načten. Nejprve fuknce zkontroluje, zda je tato cesta validní a pokud ano, zda je soubor dostupný pro čtení. Pokud ano, začne načítat jednotlivé řádky a parsovat vstup dle formátu zmíněného výše. Pokud jsou data jakýmkoliv způsobem poškozena, či neodpovídají formátu, dojde k přerušení zpracovávání dat a do „status labelu“ se přiřadí text „invalid data“. V opačném případě „data loaded.“
* ***read\_data*** – na vstupu této funkce je opět předán kontext ze kterého funkce vyčte aktuální port. Pokud je port nastavený na test, začnou se generovat náhodná data. V opačném případě se zavolá „port\_reader.py“. Pokud selže čtení z „port\_reader.py“, nastaví se status „permission denied“. Součástí této funkce je i kontrola maximálního počtu bodů vykreslovaných do grafu.
* ***write\_data\_to\_rc*** – funkce přebírá na vstupu kontext a text, kdy z kontextu opět načte cestu k souboru, do kterého má být text uložen a text je jedna řádka dat již v daném formátu.
* ***plot\_graf*** – funkce přebírá na vstupu kontext a graf ID, kdy v Context na základě graf ID zaindexuje do listu „axs“ a „graphs“ čímž vybere příslušný graf na základě ID a příslušná data, která vykreslí. V této funkci také probíhá kontrola, zd má být daná křivka vykreslována (viz view) a je zde možné (i když zatím pouze v kódu) nastavit barvu a název jednotlivých křivek.
* ***generate\_timestamps*** – z kontextu vyčte čas posledního vykreslení a přičte k němu 50 ms. Tento čas je nyní nastaven jako čas posledního vykreslení. Toto slouží jako generování x-ové osy.
* ***animate*** – tato funkce projde všechny grafy a zavolá funkce pro zobrazení hodnot.
* ***set\_animator*** – přečte data pomocí funkce „read\_data“ a zavolá funkci „animate“. Potom se uspí na stanovený čas (toto slouží místo aktivního čekání). Součástí této funkce je i kontrola zda má aplikace v tuto chvíli vykreslovat data.
* ***discard\_overflowing\_points*** – v případě, že chceme po stisknutí „show\_full\_graph“ opět redukovat body v grafu na původně nastavený limit, zavolá se tato funkce, která odstraní z grafu všechny přebytečné body.

**Events:**

Tyto funkce se aktivují pouze v případě, že byla splněna podmínka pro jejich spuštění (např.: stisknutí tlačítka). Tyto eventy pouze volají již výše zmíněné funkce a proto je zde nebudeme příliš rozvádět.

* ***stop\_animator\_event*** – zastaví animaci
* ***start\_animator\_event*** – spustí animaci
* ***clear\_graph\_event*** – nastaví grafy do výchozího stavu
* ***set\_recorder\_event*** – spustí ukládání dat
* ***load\_data\_event*** – inicializuje načítání dat ze souboru
* ***setting\_widget\_show\_event*** – inicializuje spuštění SettingsWindow

**Inicializační funkce:**

* ***init\_reader*** – při načítání aplikace prohledá všechny dostupné sériové porty a zobrazí je v nabídce.
* ***init\_graph\_data*** – inicializuje datové struktury s grafovými daty
* ***init\_gui*** – inicializace grafického vzhledu aplikace