

Analýza technológií

1 Úvod

Tento dokument má slúžiť na popis technológií, ktoré sú potrebné na plnú funkčnosť ISVOS-y. Taktiež opisuje komponenty, ktoré budú v našom systéme a taktiež zobrazuje dátový model a triedny diagram.

2 Analýza technológií

Ako bolo stanovené v dokumente "Katalóg požiadaviek", v sekcii "2.1.1 Používateľské rozhrania" projekt bude písaný v programovacom jazyku Python.

2.1 Používané technológie

2.1.1 Možnosti technológií pripojenia SVOSA k SA:

1. LAN - (Local Area Network) systém vzdialeného ovládania poskytuje dve metódy :
 - SICL-LAN – v ovládacom systéme používajúcom SICL-LAN server, komunikácia medzi vonkajším ovládačom (klientom) a SA (serverom) je uskutočnená použitím SICL-LAN protokolu. Samotná komunikácia je uskutočnená pomocou SICL (Standard Instrument Control Library). Užívateľ môže ovládať SA programovaním pomocou SICL alebo VISA v jazyku C pod operačným systémom Linux, alebo Visual C++, Visual Basic a VEE pod operačným systémom Windows.
 - Telnet – v ovládacom systéme cez telnet server, komunikácia je uskutočňovaná cez pripojenie medzi zásuvnými modulmi poskytovanými procesmi vonkajšieho ovládača a SA na nadviazanie sieťového spojenia medzi nimi. Ovládací systém cez telnet môže komunikovať dvoma portami a to:
 - port 23 – používa sa na dialógové ovládanie používajúce telnet (užívateľské rozhranie pre TELNET protokol)
 - port 5025 – používa sa na ovládanie z programu
2. GBIP – (General Purpose Interface Bus) je štandardné rozhranie pre pripojenie počítačov a periférnych zariadení, ktoré podporuje nasledujúce medzinárodné štandardy : IEEE 488.1, IEC-625, IEEE 488.2, a JIS-C1901 . GPIB rozhranie umožňuje ovládať Agilent SA z externého počítača. Počítač odosiela príkazy a pokyny na SA a prijíma dáta odoslané z SA cez

GPIB .

Výber technológie pripojenia SVOSA k SA

Na základe špecifikácie požiadaviek sme sa rozhodli uprednostniť LAN technológiu, keďže má byť SA ovládaný cez lokálnu sieť, s použitím metódy SICL-LAN pretože ovládací počítač má bežať pod operačným systémom Linux.

2.1.2 Hardwarové technológie

1. Náš systém bude používať technológiu TCP/IP na spojenie sa s koncovým zariadením. Očakáva sa že to bude väčšinou notebook alebo PC. Prípadné rozšírenie je zapojenie zariadenia do sieťového prvku (napríklad rozbočovač).

2.1.3 Softwarové technológie

Náš systém bude používať rôzne knižnice pre dosiahnutie konkrétnych požiadaviek

1. Na vykresľovanie grafov budeme používať knižnicu "Gnuplot", ktorá dokáže vykresľovať rôzne typy grafov. Všeobecnejšie sa dá používať aj ako nástroj v Linuxe na kreslenie grafov cez príkazový riadok.

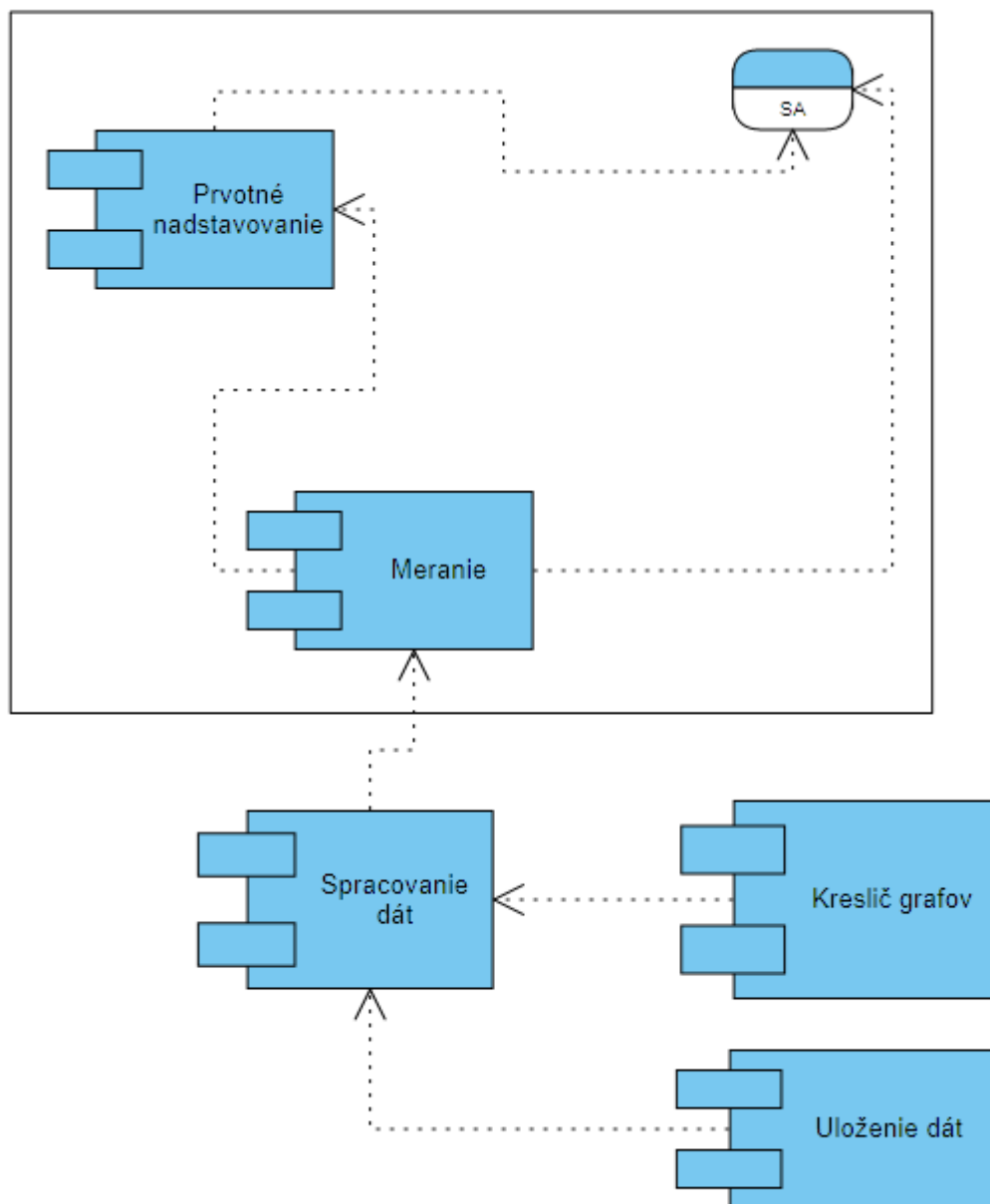
2. Taktiež na nízko úrovňovú sieťovú komunikáciu s SA budeme potrebovať aj knižnicu menom "socket". Budeme vedieť pomocou nej sa pripojiť na konkrétnu IP adresu na konkrétny port nášho SA.

Konkrétne sa najprv inicializuje socket . Vyberie sa adresnú rodinu konkrétne : "socket.AF_INET" ,ktorá je používaná pre IP adresovaný socket. Skratka AF_INET je skratka od Adress Family InterNET. Taktiež si pri inicializácii akým spôsobom má komunikovať. My sme si zvolili " socket.SOCK_STREAM " čo nám poskytuje sekvenčný, spoľahlivý, obojsmerný , spojitý tok byte-ov.

Potom sa napojíme na ip adresu zariadenia v sieti (konkrétne táto : 192.168.1.102) a na príslušný port (konkrétne 5025). Potom už len posielame správy SA v textovej forme alebo si od neho pýtame určité množstvo textu (konkrétne cyklicky 1024 znakov pokiaľ uznáme za vhodné). A na koniec spojenie ukončujeme príkazom close(), ktorý štandardne ukončuje spojenie medzi spojenými zariadeniami.

3. Knižnica "math" budeme používať na výpočet matematických vzorcov a volanie `math.atan2()` funkcie, ktorá ráta arkus tangens a vracia výsledok v radiánoch. Pričom táto funkcia má 2 parametre, ktoré musia byť čísla.

3 Komponentný diagram



Obrázok 3.1

3.1 Komponent prvotné nastavovanie

Komponent slúži na nastavenie všetkých potrebných parametrov potrebných na napojenia sa pomocou siete na SA. Okrem iného sa v ňom nastavujú dôležité konštanty ako napríklad adresa

zariadenia na sieť či príslušný port, ktoré užívateľ ďalej nemôže meniť. Tu sa budú dať aj nastaviť ostatné parametre ako bolo už definované v katalógu požiadaviek.

3.2 Komponent meranie

Komponent slúži na posielanie pokynov na meranie a ich spätné prijímanie v nastavenom formáte. Dáta sa ukladajú do internej štruktúry. Spracovanie dát zo štruktúry je už spracúvaný iným systémom, ktorý dátam rozumie.

3.3 Komponent spracovanie dát

Komponent prečíta dáta z internej štruktúry a preloží do ďalších potrebných štruktúr a počítajú sa potrebné veličiny z dodaných dát pre ich ďalšie využitie.

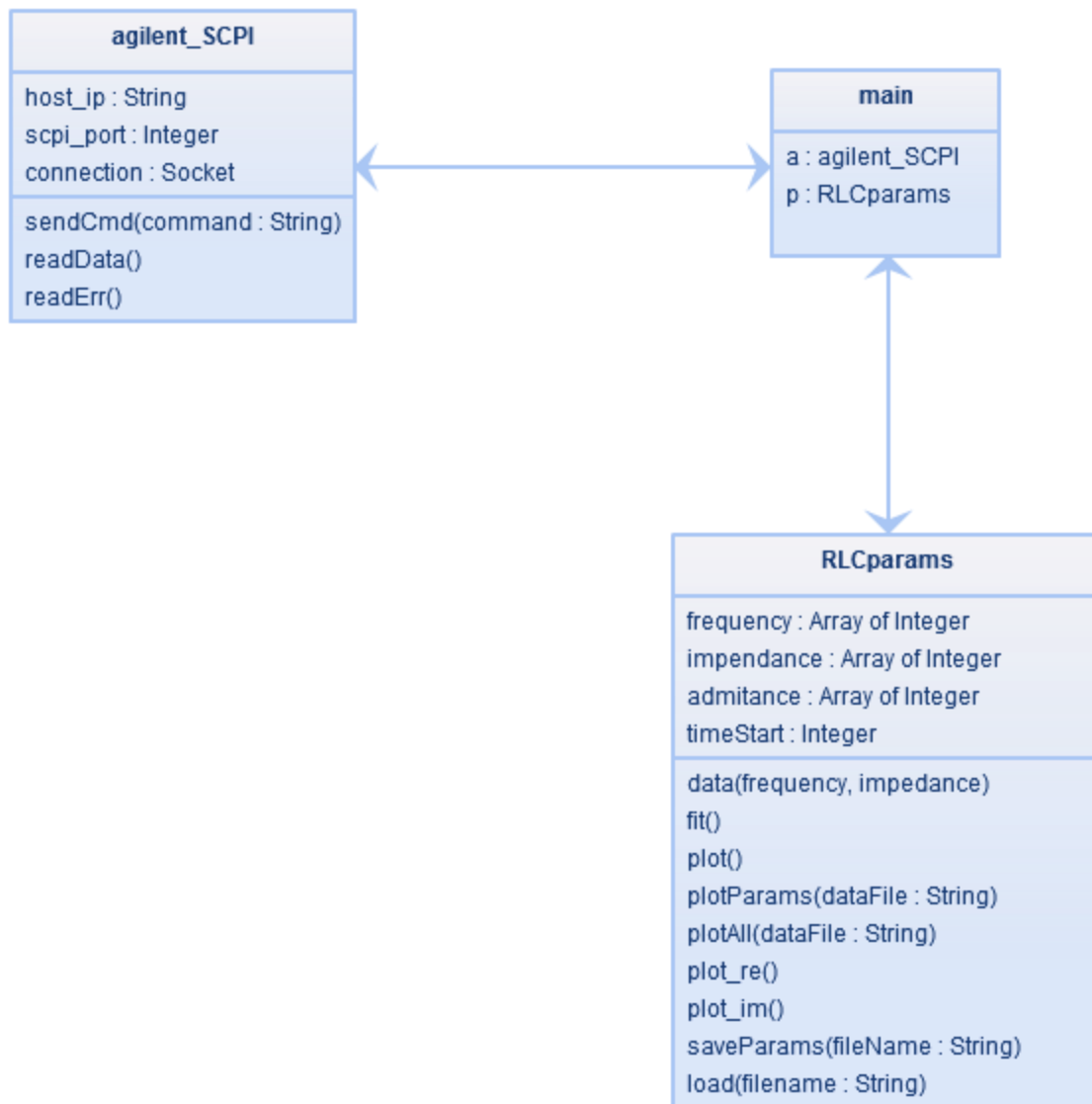
3.4 Komponent uloženie dát

Dáta sa ukladajú do prehľadnej tabuľky, ak sa užívateľ rozhodne inak sa dáta nebudú ukladať.

3.5 Komponent kreslič grafov

Z poskytnutých spracovaných a vyrátaných dát a veličín dokáže tento komponent zobrazíť potrebné grafy.

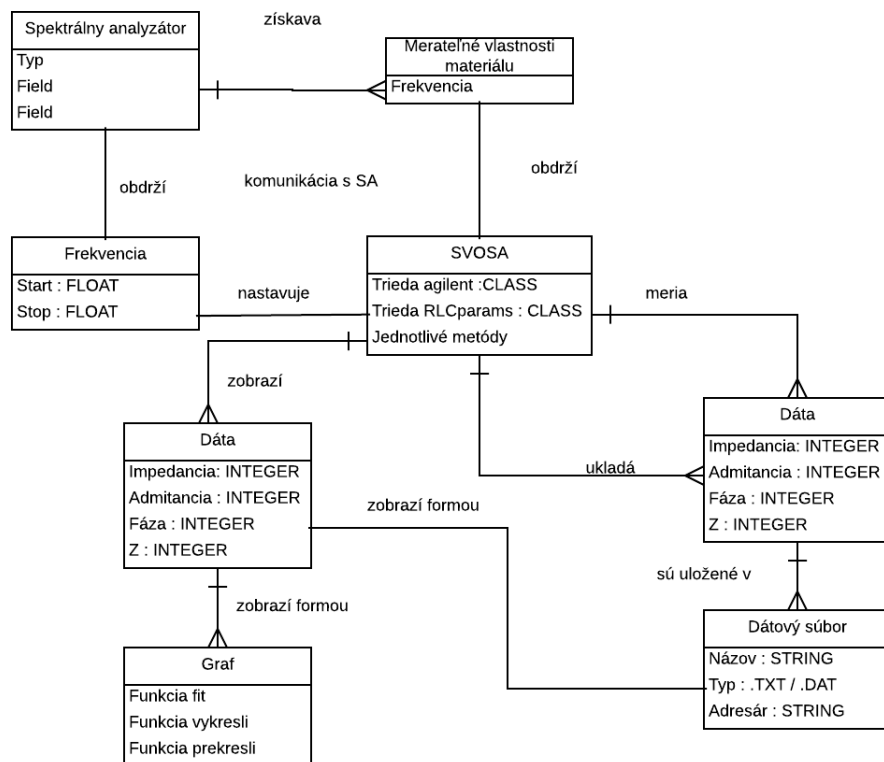
4 Triedny diagram



Obrázok 4.1

Triedny diagram znázorňuje štruktúru projektu, popisuje jeho triedy a ich vzájomné prepojenie. Hlavný program main spravuje a inicializuje triedy agilent_SCPI a RLCparams. SVOSA pomocou triedy agilent_SCPI nadviaže spojenie zo SA, a následne pomocou tejto triedy aj prijíma dáta, ktoré po úspešnom prijatí analyzuje a graficky znázorní pomocou triedy RLCparams.

5 Dátový model zobrazený pomocou entitno-relačného diagramu



Obrázok 5.1

Podrobnejší entitno-relačný diagram znázorňujúci dátový model Informačného systému.

SVOSA pracuje a analyzuje získané dáta od SA a tie následne dokáže graficky znázorňovať, alebo ich dokáže ukladať (exportovať) do textového, alebo dátového súboru s ktorým môžu pracovníci ďalej pracovať.