

Vzdálené řízení a monitoring měřicích přístrojů

Remote management and monitoring of measuring devices

Bc. Josef Trbušek

Diplomová práce
2011



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2010/2011

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Josef TRBUŠEK**

Osobní číslo: **A09492**

Studijní program: **N 3902 Inženýrská informatika**

Studijní obor: **Informační technologie**

Téma práce: **Vzdálené řízení a monitoring měřicích přístrojů**

Zásady pro vypracování:

1. Vytvořte rešerši na téma **SW nástroje pro řízení a monitoring měřicích přístrojů**.
2. Pomocí programovacího jazyka **C/CPP**, knihovny **wxWidgets** a dalších vhodných knihoven a **SW** technologií vytvořte **multiplatformní systém pro vzdálený monitoring a řízení měřicích přístrojů na platformách MS Windows a Linux**.
3. Systém by měl umožňovat **vzdálený monitoring měřených parametrů, vzdálené řízení měřicího přístroje, komunikaci s měřicími přístroji pomocí protokolů USB a TCP/IP (VXI) a dalších vhodných**.
4. Systém by měl být **plně konfigurovatelný a rozšiřitelný**.
5. Aplikace bude šířena pod licencí **GPL 2**.
6. Vytvořte **programovou a uživatelskou dokumentaci**.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. PRATA, S. **Mistrovství v CPP: Computer Press, 2007, 3. vydání, 1120s, ISBN 978-80-251-1749-1**
2. SMART, Julian, HOCK, Kevin. **Cross-Platform GUI Programming with wxWidgets, Prentice Hall, 2006, ISBN 0-13-147381-6**
3. BLIŽŇÁK, Michal. **Systémové programování. 1. vyd. Zlín: UTB ve Zlíně, 2005. 202 s. ISBN 80-7318-364-1**
4. BLIŽŇÁK, M, DULÍK, T, VAŠEK, V. **wxShapeFramework: An easy way for diagrams manipulation in CPP applications. WSEAS Transactions on Computers [online]. March 2010, Volume 9, Issue 3, [cit. 2010-12-01]. Dostupný z WWW: . ISSN 1109-2750**
5. KANISOVÁ, H. - MÜLLER, M. **UML srozumitelně. 2. akt. vyd. Brno: Computer Press, 2006. 176 s. ISBN 80-251-1083-4**

Vedoucí diplomové práce:

Ing. Michal Bližňák, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

24. února 2011

Termín odevzdání diplomové práce:

18. května 2011

Ve Zlíně dne 24. února 2011

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Tato diplomová práce je zaměřena na tvorbu multiplatformní SW aplikace umožňující vzdálený přístup k měřicím přístrojům a jejich monitoring. Pro komunikaci s přístroji je možné využívat připojení přes USB a TCP/IP. Teoretická část práce popisuje softwarové a hardwarové nástroje použité pro realizaci tohoto projektu a popis a srovnání s jinými obdobnými programy. Praktická část popisuje vytvořený program z hlediska uživatele a programátora.

Klíčová slova: wxWidgets, C++, měřicí přístroje, USB, TCP/IP, XML, wxShapeFramework.

ABSTRACT

This thesis aims at creating a program that allows communication and monitoring measuring instruments. For communication can be used connection via the USB and TCP/IP. The theoretical part of thesis describes a software and hardware tools used to implement this software and a description and comparison with other programs for same purpose. The practical part deals with a description of the program from the user's and programmer's points of view.

Keywords: wxWidgets, C++, measuring instruments, USB, TCP/IP, XML, wxShapeFramework

Úvodem bych rád poděkoval svému vedoucímu práce, panu Ing. Michalu Bližňákovi, Ph.D. za pomoc, rady a trpělivost při tvorbě této diplomové práce. Dále bych rád poděkoval svým rodičům a přítelkyni za velké množství trpělivosti a podpory. A taktéž mnoha dalším lidem, kteří se radou nebo jakýmkoliv jiným způsobem přispěli k vytvoření této práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	11
I TEORETICKÁ ČÁST	12
1 POUŽITÉ SOFTWARE A HARDWAROVÉ PROSTŘEDKY	13
1.1 WXWIDGETS.....	13
1.1.1 Stručná historie wxWidgets	13
1.1.2 Architektura.....	14
1.2 JAZYK C/C++.....	15
1.2.1 Stručná historie	15
1.2.2 Výhody objektově orientovaného programování	15
1.3 POUŽITÉ MĚŘICÍ PŘÍSTROJE	16
1.3.1 Agilent 33220A	16
1.3.2 Agilent DSO1004A	17
1.4 POUŽÍVANÉ STANDARDY	18
1.4.1 TCP/IP.....	18
1.4.2 USB	18
1.4.3 SCPI	19
1.4.4 GPL 2.....	19
2 JINÉ MOŽNOSTI PRO VZDÁLENÉ ŘÍZENÍ A MONITOROVÁNÍ MĚŘICÍCH PŘÍSTROJŮ	20
2.1 AGILENT VEE	20
2.2 OVLÁDÁNÍ POMOCÍ WEBOVÉHO ROZHRAŇÍ PŘÍSTROJE	21
2.3 MOŽNOSTI OVLÁDÁNÍ POD SYSTÉMEM LINUX	22
2.4 POROVNÁNÍ S VYTVÁŘENÝM PROGRAMEM.....	23
II PRAKTICKÁ ČÁST	24
3 UŽIVATELSKÁ DOKUMENTACE	25
3.1 SPUŠTĚNÍ PROGRAMU.....	25
3.2 HLAVNÍ OKNO PROGRAMU	25
3.3 MENU FILE	27
3.3.1 New	28
3.3.2 Open	28
3.3.3 Save	29
3.3.4 Exit	30
3.4 MENU EDIT	31
3.4.1 Undo.....	31
3.4.2 Redo	31
3.4.3 Align.....	32
3.5 MENU PROJECT	32
3.5.1 Run	33
3.5.2 Stop	33

3.6	MENU HELP.....	33
3.7	VKLÁDANÉ PRVKY	33
3.7.1	Bitmap	34
3.7.2	Edit Text.....	35
3.7.3	Check Box	36
3.7.4	Choice	37
3.7.5	Plot	38
3.7.6	Rectangle.....	41
3.7.7	Slider	41
3.7.8	Universal	42
3.8	PRACOVNÍ PLOCHA PROGRAMU.....	43
3.9	TABULKA VLASTNOSTÍ.....	44
3.9.1	Nastavení Instrument ID	44
4	PROGRAMÁTORSKÁ DOKUMENTACE.....	46
4.1	PROSTŘEDKY PRO TVORBU PROGRAMU	46
4.2	STRUKTURA PROGRAMU	46
4.3	SPRÁVA HLAVNÍHO OKNA	49
4.3.1	Menu	49
4.3.1.1	New	49
4.3.1.2	Open.....	50
4.3.1.3	Save.....	50
4.3.1.4	Exit.....	51
4.3.1.5	Undo.....	51
4.3.1.6	Redo.....	51
4.3.1.7	Align	52
4.3.1.8	Run.....	52
4.3.1.9	Stop	52
4.3.1.10	About.....	52
4.3.2	Nastavování vlastností prvků	52
4.4	SPRÁVA PROJEKTŮ	53
4.4.1	Manipulace se záložkami	53
4.4.2	Manipulace s prvky projektu.....	53
4.5	OVLÁDACÍ PRVKY	54
4.5.1	Bitmap	54
4.5.2	Edit Text.....	54
4.5.3	Check Box	54
4.5.4	Choice	55
4.5.5	Plot	55
4.5.6	Rectangle.....	55
4.5.7	Slider	55
4.5.8	Universal	56
4.6	KOMUNIKACE S PŘÍSTROJI.....	56
	ZÁVĚR.....	58
	ZÁVĚR V ANGLIČTINĚ	59

SEZNAM POUŽITÉ LITERATURY	60
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	61
SEZNAM OBRÁZKŮ	62
SEZNAM PŘÍLOH.....	64

ÚVOD

Diplomová práce se zabývá návrhem a tvorbou programu pro vzdálené ovládání měřicích přístrojů pomocí PC. Účelem vytvořeného programu bude mimo jiné jeho využití na virtuálních pracovních stanicích v rámci projektu VLAM [9] a tím zpřístupnění mnohdy velice drahých měřicích přístrojů uživateli, který nebude mít možnost s měřicími přístroji fyzicky manipulovat.

Jedním z požadavků na tento program je, aby byl funkční pro platformy MS Windows a Linux. Z tohoto důvodu byly pro vývoj použity jazyk C/C++ a knihovna wxWidgets, které jsou multiplatformními. U knihovny wxWidgets je navíc výhoda ve využití nativních prvků, tj. že knihovna používá primárně prvky dané platformy a až nejsou tyto k dispozici, využívá vlastní. Výhodné použití je i proto, že knihovna wxWidgets je primárně určena pro jazyk C++ a je šířena jako open source, tj. zdarma.

Komunikace pomocí různých protokolů je řešena pomocí pluginů. Pro každou platformu a protokol je vytvořen zvláštní plugin. Zde lze spatřovat velkou výhodu v možnosti budoucího jednoduchého rozšiřování celé aplikace pro komunikaci na jiných systémech a protokolech. To z důvodu, že všechny pluginy jsou vytvořeny na základě společného interface, kdy mají stejné funkce a pouze samotná implementace funkcí je rozdílná kvůli nejednotnému způsobu komunikace na různých platformách a protokolech. V základu programu je vytvořen plugin pro komunikaci na platformě MS Windows a Linux. V případě MS Windows jde o jeden plugin, kdy pomocí ovladače VISA je možné s přístroji komunikovat jak přes USB tak TCP/IP. V případě platformy Linux jsou dva samostatné pluginy pro komunikaci na jednotlivých protokolech USB a VXI (TCP/IP).

Samotné ovládání přístrojů v programu je řešeno pomocí vytvořených prvků, kterým lze nastavovat vlastnosti a pomocí nichž lze nastavovat parametry přístrojů.

I. TEORETICKÁ ČÁST

1 POUŽITÉ SOFTWARE A HARDWAROVÉ PROSTŘEDKY

Tato část práce popisuje hlavní prostředky, které byly použity pro realizaci programu, který byl cílem této práce.

1.1 wxWidgets

Jak již bylo zmíněno v úvodu, wxWidgets je open source multiplatformní knihovna využívající nativních prvků platformy. Tato knihovna má široké spektrum využití od jednoduchým konzolových aplikací přes GUI aplikace až po složité vícevláknové aplikace. Knihovna je vytvořena v jazyku C++, což však neznamená, že ji nelze používat při vývoji v jiných jazycích jako Java, C#, Ruby, Python aj.[1]

1.1.1 Stručná historie wxWidgets

Hlavním tvůrcem knihovny je Julian Smart, který jako zaměstnanec Univerzity v Edinburku vyvíjel nástroj pro vytváření diagramů s názvem Hardy. Jelikož se odmítal rozhodovat mezi jednotlivými platformami dané doby, rozhodl se vytvořit vlastní multiplatformní knihovnu. V září roku 1992 tedy vychází první verze tehdy ještě wxWindows 1.0. Zkratka wx na začátku názvu byla odvozena od w – windows a x – X Window system. [1]

Jelikož již první verze byla volně dostupná na FTP serveru a dokázala nabídnout lepší řešení než jiné komerční projekty, začala se kolem wxWindows vytvářet rozsáhlá komunita, ze které je vhodné jmenovat tyto tvůrce:

Marcus Holzem – vytvořil Xt port a v roce 1997 pomohl navrhnout API pro novou verzi wxWindows, v roce 1998 však odešel od verze 2.

Wolfram Gloger – navrhnul portování na GTK+, neboť GTK+ mělo hlavní problém ohledně toho, že bylo založeno pouze na jazyku C

Robert Roebling – hlavní vývojář portu wxGTK, který je dodnes hlavním portem pro systémy Unix/Linux

Vadim Zeitlin – přispěl značně k projektu návrhy a kódy

Stefan Csomor – v roce 1998 práce na portu pro Mac OS

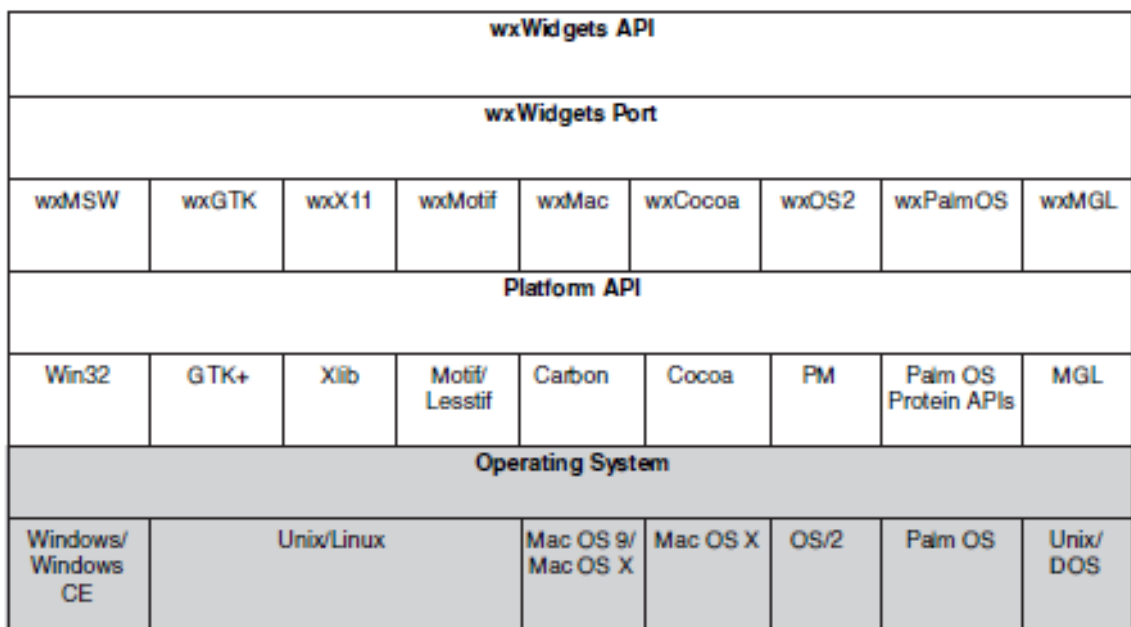
Václav Slavík – tvůrce třídy wxHTML a prohlížeče nápovědy ve formátu HTML (verze 2.1.11)

V roce 2002 J. Smart a R. Roebing vyvinuli port wxX11 pro Unix a X11 a v roce 2003 je již podporován systém Windows CE pro mobilní zařízení.

V roce 2004 dochází ke změně názvu z wxWindows na wxWidgets kvůli stížnostem společnosti Microsoft a přidána podpora Mac OS X portem wxMac a podpora pro Palm OS 6 portem wxPalmOS.

V roce 2005 vychází verze 2.6 vylepšující všechny porty, které v podstatě pokračuje do současnosti, kdy poslední stabilní verze je 2.8.12.[8]

1.1.2 Architektura



Obrázek 1: Porty wxWidgets [1]

Jak je možné vidět na obrázku výše, wxWidgets obsahuje základní vrstvu API, portaci, která je využívána pro jednotlivé platformy využívané operačními systémy.

Vnitřní struktura wxWidgets se dělí na: [1]

Společný kód – kód společný všem portům bez ohledu na jejich určení a použití.

Obecný kód – emuluje prvky, které jsou pro danou platformu neznámé a nestandardní (např. *wxCalendarCtrl*)

wxUniversal – implementuje prvky pro platformy, které nemají svoje vlastní.

Platformně závislý kód – je jedinečný pro každý port

Rozšiřující kód – třídy, vylepšující základní prvky, tvořící samostatný prvek. Jsou umístěny v adresáři *contrib*.

Kód třetích stran – třídy vytvořeny mimo wxWidgets nutné pro správnou funkci vlastností wxWidgets.

1.2 Jazyk C/C++

1.2.1 Stručná historie

Počátky jsou datovány mezi léty 1969 a 1973 do Bellových laboratoří AT&T. Hlavními tvůrci jsou Denis Ritchie, Brian Keninghan a Ken Thompson. Název je odvozen od staršího jazyka B.[1]

V roce 1973 je jazyk stabilní a je do něj přepsáno jádro systému Unix. Během 80. let probíhá normování jazyka C. V roce 1989 je vydána první norma známá jako ANSI C a o rok později i ISO norma. V roce 1999 je vydána ještě norma C99 rozšiřující původní normu o inline funkce, definice proměnných kdekoliv v kódu a další vlastnosti. Tato byla v roce 2000 přijata i jako ANSI norma.

Již od roku 1986 probíhal vývoj objektového jazyka C++. Název je odvozen od jazyka C a značí jakoby jeho vylepšení. Jazyk C++ je plně kompatibilní s jazykem C. Jeho hlavním rozšířením je podpora objektově orientovaného programování a zavedení knihovny STL. V roce 1998 byla přijata první norma jazyka C++ a další pak v roce 2003. Aktualizace norem proběhly v letech 2006 a 2007.

1.2.2 Výhody objektově orientovaného programování

Objektově orientované programování, ať už v kterémkoliv jazyku, jež objektově orientované programování podporuje, má tři základní principy, pro které je výhodné jeho použití:[1]

Zapouzdření – spojení dat a metod pro manipulaci s těmito daty do nové struktury – třídy. Třída je tedy popisem objektů se stejnými vlastnostmi a objekt je instancí třídy. S členskými daty třídy lze manipulovat jen pomocí členských metod třídy.

Dědičnost – umožňuje vytvářet z již existujících tříd třídy nové. Již existující třída je pak nazývána předek a nová potomek. Potomek získává od předka všechny všechna data a metody a může přidávat své vlastní nebo stávající upravovat. Směrem více k předkům jsou třídy tedy obecnější a opačně konkrétnější. Je možná i vícenásobná dědičnost, kdy potomek získává vlastnosti od více předků. Zde je však možné nebezpečí konfliktů.

Mnohotvárnost – umožňuje používat více verzí stejné funkce. Výběr mezi nimi je prováděn pomocí včasné nebo pozdní vazby. Včasná vazba je prováděna během překlady programu, kdežto pozdní během běhu programu. S pozdní vazbou se pojí i pojmy abstraktní třída a virtuální metoda. Abstraktní třída je společnou pro třídy využívající jejich virtuálních metod. Tyto metody jsou v abstraktní třídě pouze deklarovány a implementovány jsou v třídě potomka. Dnes se využívá především pozdní vazby.

1.3 Použité měřicí přístroje

1.3.1 Agilent 33220A



Obrázek 2: Agilent 33220A[3]

Jedná se o funkční generátor libovolných průběhů. Je schopný generovat signál s frekvencí 1 μ Hz až 20 MHz (podle typu signálu). Signál může být typu sinus, čtverec, rampa, šum a DC. Rovněž je schopen generovat pulsy s proměnnou hranou. Poskytuje velké množství typů modulace a logaritmické a lineární škálování. Připojení je možné pomocí USB, LAN nebo GPIB.[2]

1.3.2 Agilent DSO1004A



Obrázek 3: Agilent DSO1004A[3]

Jde o osciloskop s 60 MHz šířkou pásma poskytující 4 analogové kanály. Zobrazení probíhá na 5,7“ displeji. Dokáže automaticky měřit 23 veličin jako frekvenci, amplitudu apod. Sekvenční mód přístroje umožňuje záznam průběhu. Připojení je možné přes USB. Přes USB je možné i uložení dat na externí paměťové médium. [2]

1.4 Používané standardy

1.4.1 TCP/IP

Je hlavním protokolem sítě Internet a jde o sadu protokolů zajišťujících komunikaci v počítačové síti.[5]

Protokol TCP/IP se skládá ze 4 vrstev, kdy každá využívá služeb nižší a poskytuje své služby vyšší vrstvě. Podle úrovně jde o tyto vrstvy:

Aplikační vrstva – programy využívající síťového přenosu dat pro služby uživateli.

Transportní vrstva – umožňuje přizpůsobit chování sítě potřebám aplikace. Je pouze v koncových zařízeních.

Síťová vrstva – zajišťuje síťovou adresaci, směrování a předávání datagramů. Je ve směrovačích i v koncových zařízeních.

Vrstva síťového rozhraní – nejnižší vrstva přistupující k fyzickému médiu. Je specifická pro každou síť podle její implementace.

Stejně vrstvy dvou různých systémů mohou mezi sebou komunikovat při použití spojení vytvořeného sousední nižší vrstvou.

1.4.2 USB

Je univerzální sériová sběrnice, pomocí níž se v dnešní době připojuje k počítači (a nejen k tomu) většina externích zařízení. Výhodou je možnost připojení Plug & Play (tj. připojení zařízení k PC za jeho chodu). Na USB port je možné připojit 127 zařízení. [4]

Ve verzi USB 1.1 existují pomalá (Low-Speed) zařízení s rychlostí přenosu 1,5 Mb/s a rychlejší (Full-Speed) s rychlostí 12 Mb/s. V obou verzích je přenos realizován pomocí 4vodičového vedení (2 kroucené datové vodiče a 2 napájecí). Ve verzi USB 2.0 je počet vodičů zachován a rychlost navýšena na 480Mb/s (v režimu Hi-Speed). Je zachována i zpětná kompatibilita. V současné době je již verze USB 3.0. Ta navyšuje počet vodičů ze 4 na 8 a rychlost na 5Gb/s a zachovává zpětnou kompatibilitu.

U USB sběrnice všechny aktivity vycházejí z PC, tj. PC může požadovat data od zařízení, zařízení nemůže samo od sebe vysílat. Veškerý přenos dat je realizován v tzv. rámcích, které trvají přesně 1 ms a mohou obsahovat pakety pro několik zařízení. Pro

přenos se využívá metody NRZI, kdy nula v datech vede ke změně úrovně a jednička úroveň ponechává.

1.4.3 SCPI

Je standardem pro syntax, příkazy a datové formáty používané pro komunikaci s měřicími přístroji. [7]

Definice tohoto standardu proběhla v roce 1990 se specifikací IEEE 488.2. Díky tomu se sjednotil způsob komunikace s velkým množstvím měřících přístrojů.

SCPI příkaz je textový řetězec tvořený jedním nebo více klíčovými slovy, která jsou často doplněna o parametr. Existují hlavní příkazy, které jsou společné pro všechna zařízení, a příkazy, které jsou specifické podle typu zařízení (např. příkazy určené pro osciloskop nemají význam u generátorů apod.). Parametr se v příkazu vždy odděluje mezerou. Pro spojení více příkazů pro vykonání se tyto oddělují středníkem.

1.4.4 GPL 2

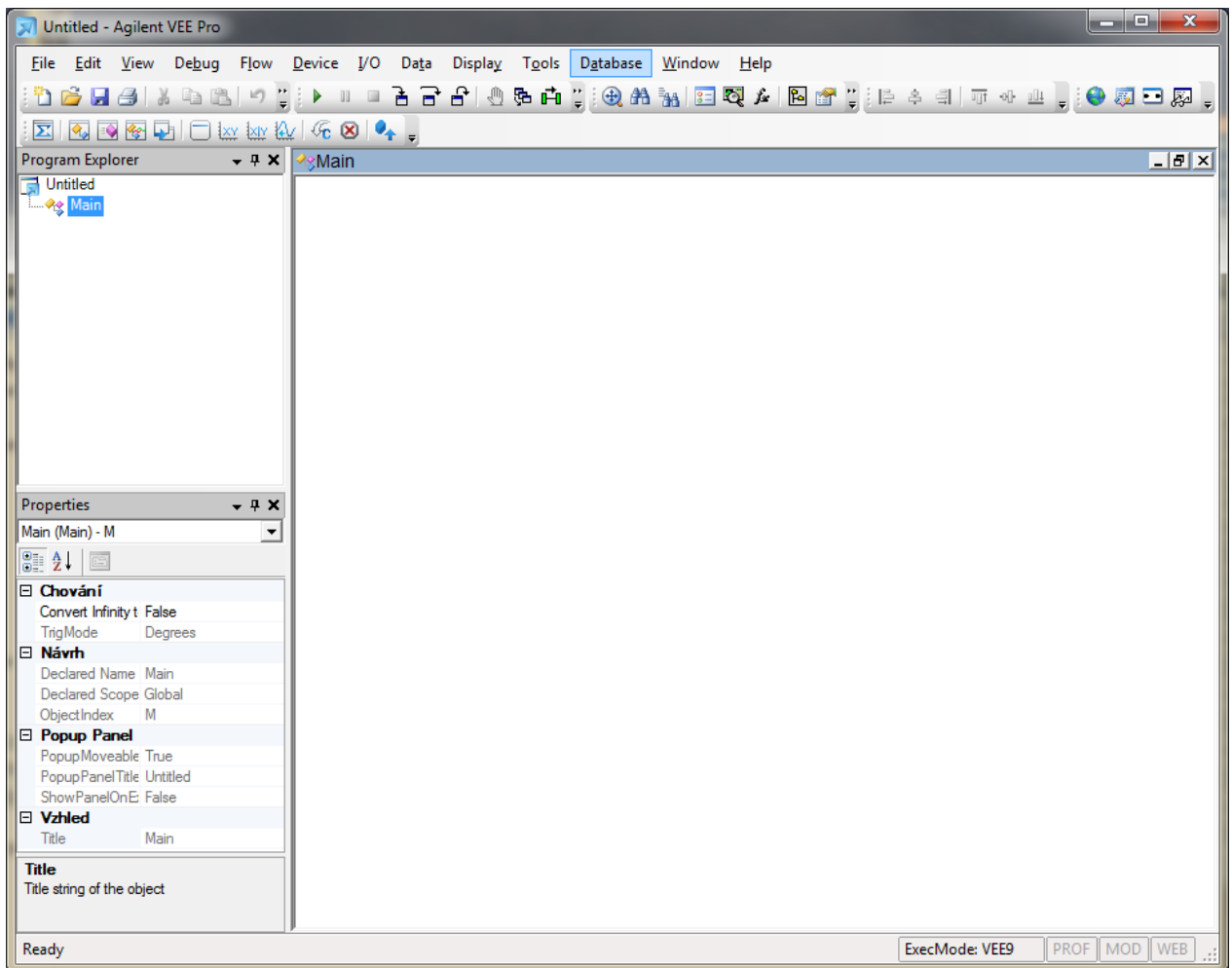
Je licencí pro svobodný software, která vyžaduje, aby odvozená díla byla dostupná pod stejnou licencí. Jde tedy o silně copyleftovou licenci, která uživateli programu poskytuje práva svobodného softwaru a pomocí copyleftu zajišťuje ochranu těchto svobod při změně softwaru nebo jeho přidání k jinému projektu.[6]

Ve verzi GPLv2 je hlavní změnou klauzule, která říká, že pokud uživatel přidá do softwaru omezení, které brání šíření softwaru pod licencí GPL, pak software nesmí být šířen vůbec.

2 JINÉ MOŽNOSTI PRO VZDÁLENÉ ŘÍZENÍ A MONITOROVÁNÍ MĚŘÍCÍCH PŘÍSTROJŮ

V této části jsou zmíněny nejznámější již existující programy, které mají stejný účel jako program vytvářený. Jsou zmíněny jejich kvality ale i nedostatky a jsou porovnány s vytvářeným programem.

2.1 Agilent VEE



Obrázek 4: Okno programu Agilent VEE Pro

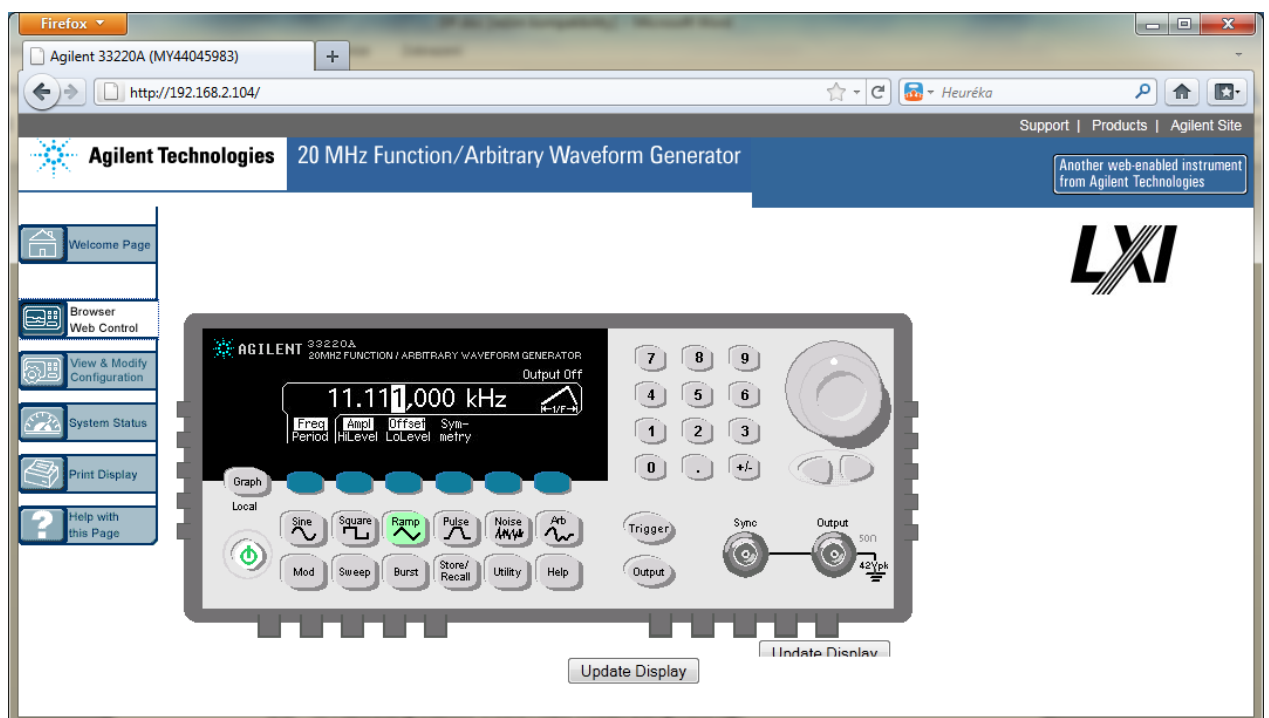
Jedná se o grafický programovací software od firmy Agilent. Jde o programovací jazyk vysoké úrovně využívající grafické zobrazení obdobné vývojovým diagramům. Prakticky to tedy znamená, že uživatel si pomocí grafických bloků a jejich propojení sestaví sekvenci, která může být vykonána.[2] Nutná je při tom znalost SCPI příkazů, která

je potřeba do některých bloků doplnit. Ačkoliv je výrobcem firma Agilent, daný software díky standardům podporuje zařízení téměř všech výrobců. Současná verze softwaru je Agilent VEE Pro 9.2. Od verze 9.0 software podporuje multithreading, multi-core programování, doplňování SCPI příkazů, podmíněné breakpointy a okno breakpointů, integrované databáze a integrované LXI. Hlavní změnou ve verzi 9.2 je podpora systému Microsoft Windows 7. [2]

Aplikace je tedy především díky možnosti grafického návrhu programu pro uživatele příznivá, nicméně má i své nevýhody. Mezi ně patří, že aplikace je pouze pro platformu Microsoft Windows, což je pro uživatele jiných platforem nevýhodné. Další z nevýhod je, že program není distribuován zdarma. Uživatel si může stáhnout 30denní zkušební verzi a poté, pokud chce program dále používat, si musí pořídit jednu z placených licencí.

Pro běh programu je nutná aplikace Agilent IO Libraries Suite, která je prostředkem pro zpřístupnění měřicích přístrojů a jejich případné konfiguraci. Tato aplikace je zdarma ke stažení z internetu.

2.2 Ovládání pomocí webového rozhraní přístroje



Obrázek 5: Webové rozhraní pro ovládání přístroje

Pokud je měřicí přístroj připojen pomocí LAN, pak je možné se na něj připojit pomocí internetového prohlížeče. Zde je možné přístroj vzdáleně ovládat pomocí prostředí skriptů samotného přístroje.

Samozřejmostí je, že samotný přístroj musí mít nastavenou svou IP adresu. Tu může získat pomocí připojení, které obsahuje DHCP a adresu přístroji přidělí. Dále je možné, pokud je přístroj připojen přímo k počítači, aby byla adresa přístroji přidělena pomocí programu Agilent Connection Expert, který je součástí Agilent IO Libraries Suite, při detekci připojených přístrojů. V krajní variantě je samozřejmě možné adresu přístroje nastavit přímo na přístroji ručně.

Výhodou zde tedy je, že veškeré prostředky pro ovládání jsou zdarma, jelikož webová stránka pro ovládání je uložena přímo na zařízení a odtud je načítána do webového prohlížeče. Tímto se tato možnost stává vlastně de facto multiplatformní, jelikož k ovládání je potřeba pouze systém s internetovým prohlížečem. Díky tomu je například možné ovládat přístroj i pomocí mobilního telefonu připojeného do LAN sítě pomocí bezdrátového připojení.

Nevýhodou zde je, že takto ovládat lze jen zařízení připojená do LAN sítě, což je problém u zařízení, která nemají možnost připojení pomocí LAN jako osciloskop Agilent DSO1004 uvedený výše.

2.3 Možnosti ovládání pod systémem Linux

Nejlepší možností je ovládání přes webové rozhraní přístroje pomocí prohlížeče tak, jak je uvedeno výše. Je to pravděpodobně nejlepší možnost grafického ovládání přístrojů na této platformě.

Jinými možnostmi jsou ovládání pomocí ovladačů pro daný protokol přes terminál. V tomto případě nejen, že je nutná znalost SCPI příkazů, ale i textové prostředí není pro všechny případy zrovna nejvhodnější (např. data křivky osciloskopu). Výhodou samozřejmě je, že se v tomto případě jedná o volně šiřitelný software a instalace je pro uživatele Linuxu rovněž jednoduchá.

2.4 Porovnání s vytvářeným programem

Vytvářený program je v podstatě kombinací všech výše uvedených programů, sloužících pro ovládání přístrojů.

Je multiplatformní jako webové rozhraní, ale na rozdíl od něj je umístěn v paměti počítače a umožňuje ovládání přes USB. Nicméně na druhou stranu neposkytuje natolik příjemné uživatelské prostředí a je zde nutná určitá znalost SCPI příkazů.

S Agilent VEE má vytvářený program společné vkládání grafických bloků, které jsou konfigurovatelné, nicméně oproti Agilent VEE nevytváří sekvence příkazů, které se po spuštění provádí automaticky. U vytvářeného programu po vytvoření a nakonfigurování grafických bloků po spuštění uživatel manuálně ovládá bloky a tím přístroje. Vytvářený program je navíc multiplatformní a není distribuován pod placenou licenci.

Ve srovnání s programy pro ovládání dodávanými s ovladači pro Linux má vytvářený program společné jen to, že je vyžadována určitá znalost SCPI příkazů uživatelem.

II. PRAKTICKÁ ČÁST

3 UŽIVATELSKÁ DOKUMENTACE

V této části je program popsán z hlediska uživatele, tj. návod pro obsluhu programu uživatelem, možnosti programu apod. Snímky programu v této kapitole jsou z prostředí systému Microsoft Windows 7. Pro jiné systémy je vzhled podobný.

3.1 Spuštění programu

Ještě před samotným spuštěním programu je potřeba mít nainstalovány některé programy. Pod systémem Microsoft Windows to je:

- Agilent IO Libraries Suite – software volně dostupný ze stránek výrobce [2]

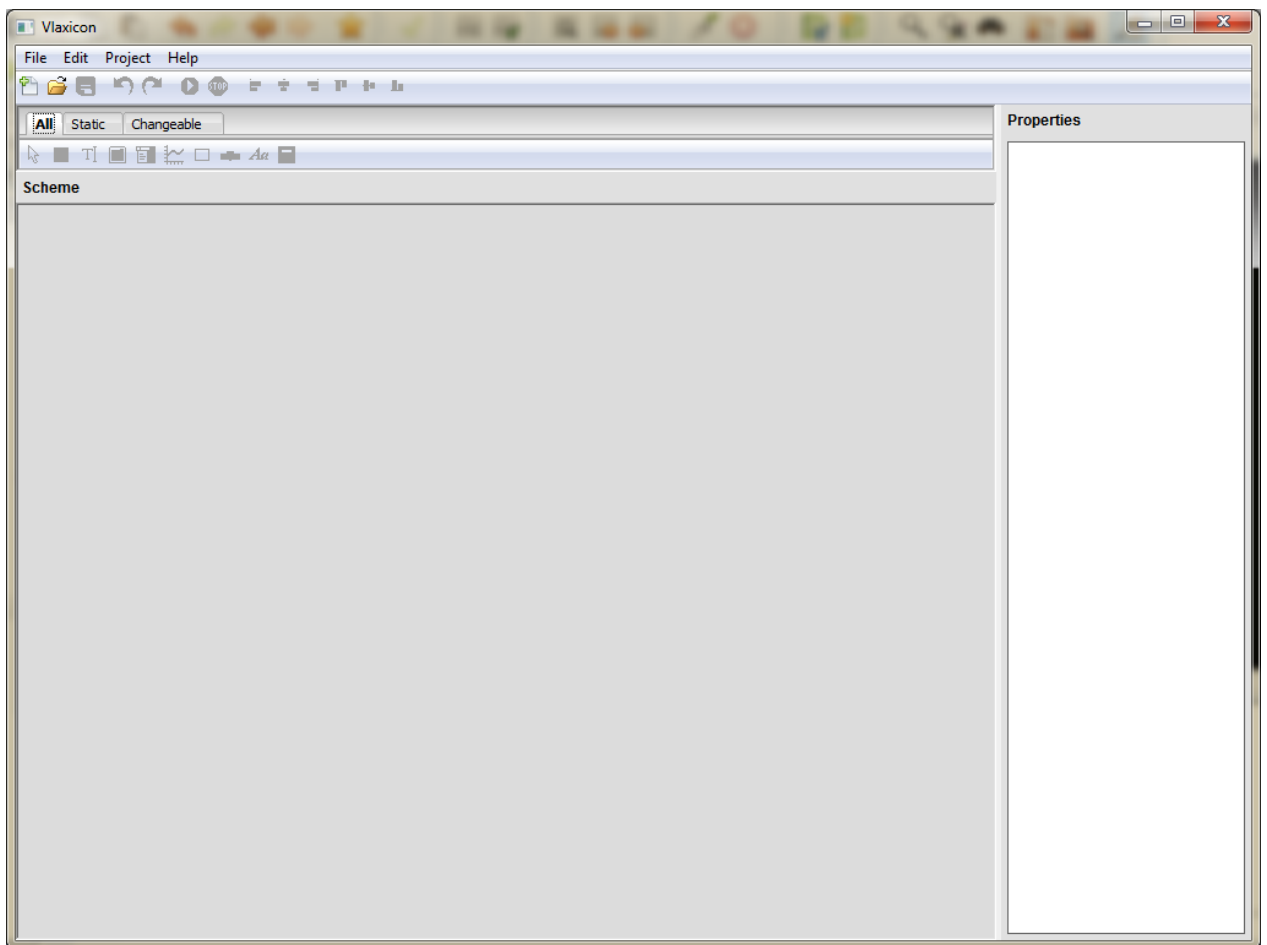
Pod systémem Linux je třeba:

- Stažení ovladače `usbtc` a připojení zařízení pomocí tohoto ovladače – stažitelný rovněž ze stránek Agilentu [2]

Program je distribuován ve formě spustitelných binárních souborů a se zdrojovými kódy. V adresáři `build` je umístěn soubor `create_build_files.bat` (resp. `create_build_files.sh` pro systém Linux). V tomto souboru je potřeba upravit cesty pro ovladače, tj. v systému Microsoft Windows parametr `visa-path` a pod systémem Linux parametr `usbtc-path`. Po úpravě potřebného souboru se spuštěním souboru vytvoří soubory pro překlad. Pomocí těchto lze program přeložit a vytvořit tak spustitelný soubor.

3.2 Hlavní okno programu

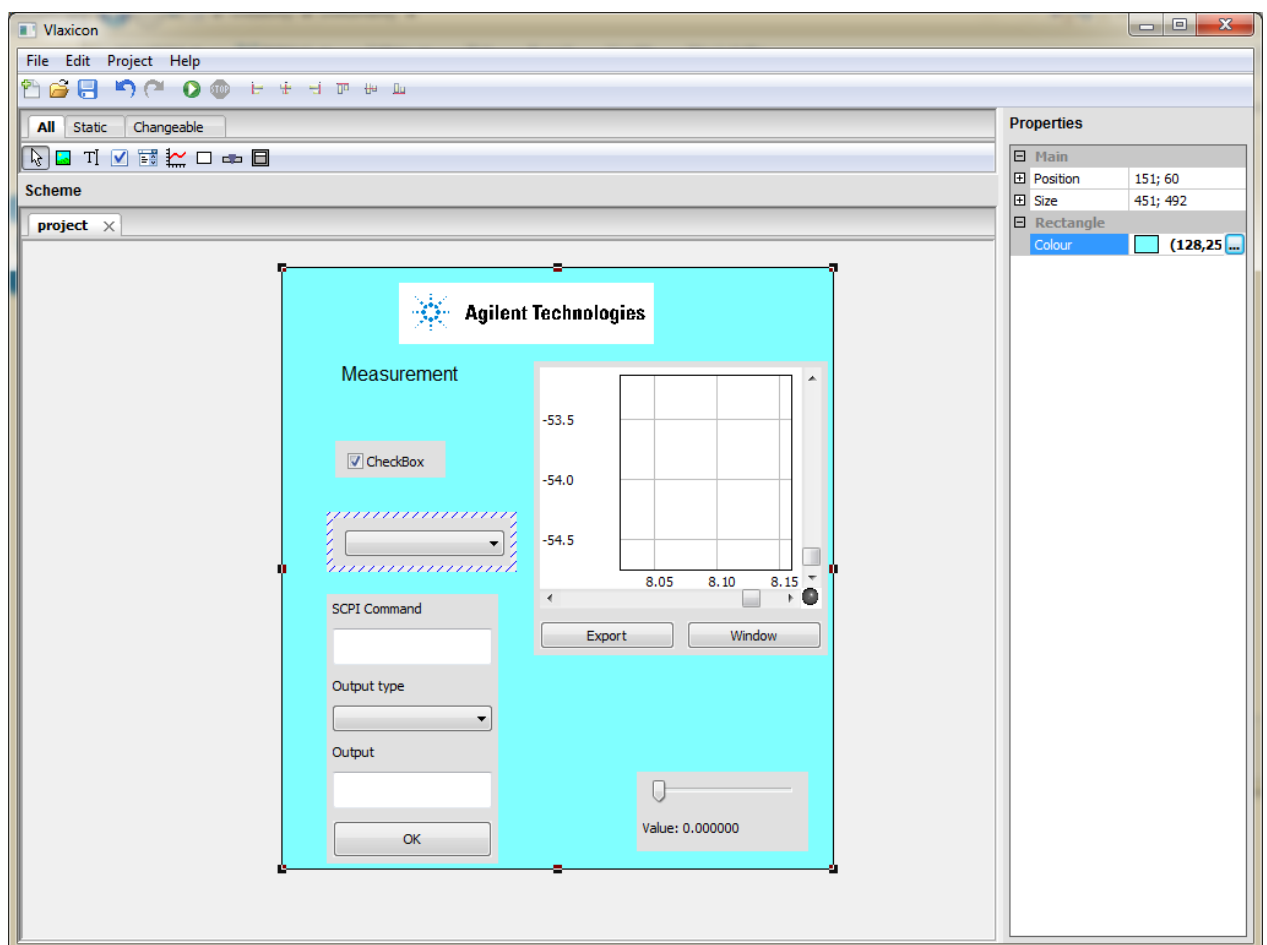
Po spuštění programu vypadá okno obdobně jako na obrázku níže. Jedinými aktivními prvky zde jsou prvky pro nový projekt (*New*) anebo otevření již uloženého projektu (*Open*).



Obrázek 6: Okno programu po spuštění

Na následujícím obrázku níže je taktéž okno programu s otevřeným projektem a vloženými prvky. Na tomto obrázku je rovněž již plně patrné rozmístění prvků programu.

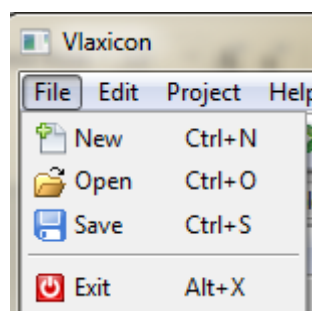
V horní části okna se nachází menu a hlavní lišta nástrojů pro ovládání programu. Pod hlavní lištou nástrojů jsou záložky, kde každá obsahuje určitý typ prvků, které je možné vložit na pracovní plochu projektu. Tato plocha je vytvořena při vytvoření nového nebo otevření již existujícího projektu jako záložka pod záložkami s prvky. V pravé části okna se pak nachází tabulka pro zobrazení vlastností vybraného vloženého prvku.



Obrázek 7: Okno s otevřeným projektem

3.3 Menu File

Toto menu obsahuje základní možnosti pro ovládání celého programu, tj. možnosti uložení, načtení, zavření programu apod.

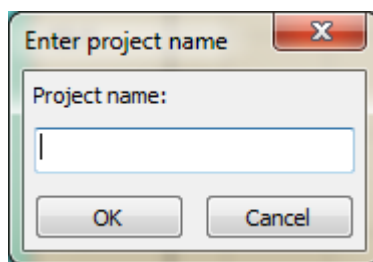


Obrázek 8: Menu File

3.3.1 New

Pomocí této volby lze vytvořit novou prázdnou pracovní plochu projektu. Tuto možnost lze rovněž vyvolat pomocí stejného tlačítka v hlavní liště nástrojů nebo pomocí klávesové zkratky CTRL + N.

Po zvolení této volby se zobrazí dialog (viz obrázek níže) pro zadání názvu projektu. Název musí být vyplněn jinak nelze nový projekt vytvořit.



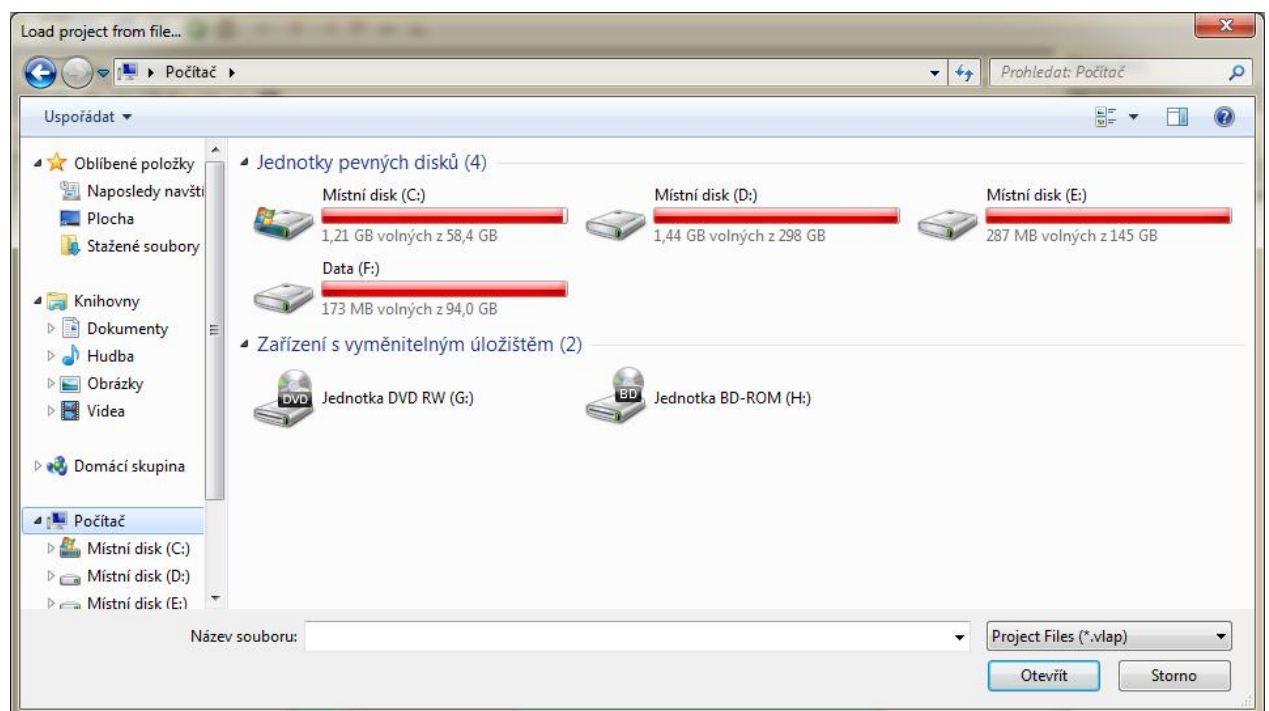
Obrázek 9: Dialog pro zadání názvu nového projektu

Jakmile je název projektu potvrzen, zobrazí se nová záložka s pracovní plochou pojmenovaná podle zadání. Zároveň se zpřístupní možnosti pro práci s touto plochou.

3.3.2 Open

Tato volba slouží pro načtení již dříve uloženého projektu. Volba je dostupná i přes položku v hlavní liště nástrojů a pomocí klávesové zkratky CTRL + O.

Při zvolení této možnosti se otevře standardní dialog pro volbu souboru (viz obrázek níže), ve kterém lze zvolit příslušný soubor pro načtení.



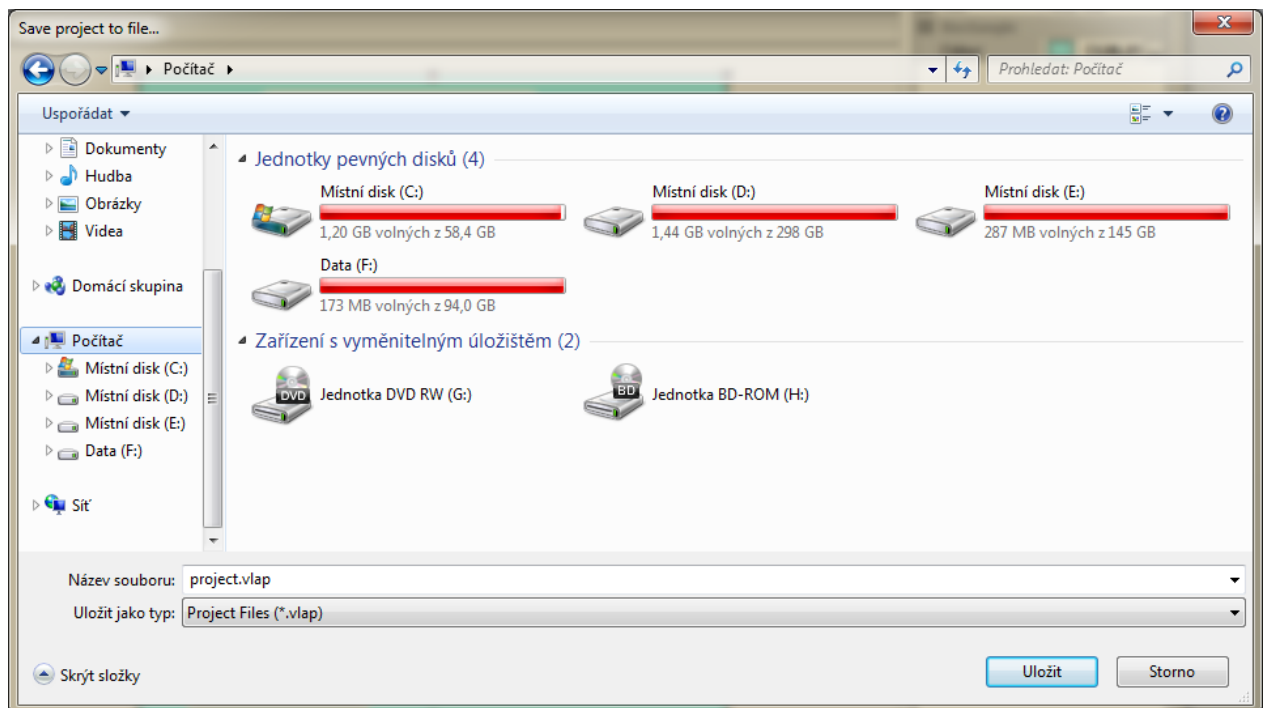
Obrázek 10: Dialog pro načtení projektu

Po zvolení souboru je obdobně jako při volbě *New* vytvořena pracovní plocha a v ní jsou vytvořeny ovládací prvky podle dat ze zvoleného souboru.

3.3.3 Save

Toto je volba pro uložení momentálně vybraného projektu do souboru. Dostupná je rovněž přes tlačítko v liště nástrojů a pomocí klávesové zkratky CTRL + S.

Při výběru této volby se zobrazí standardní dialog pro výběr názvu a cesty k souboru (viz obrázek níže).



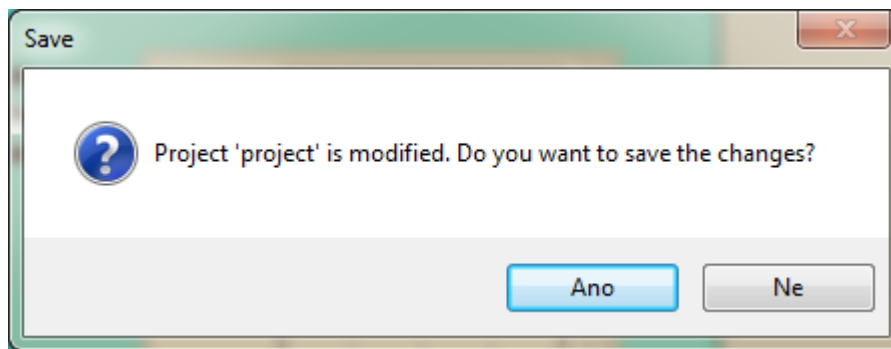
Obrázek 11: Dialog pro uložení souboru

Zde je primárně jako název souboru nastaven název shodný se zadaným názvem projektu, ale je možné tento název souboru změnit. Pokud soubor se zadaným názvem na zadané cestě již existuje, je zobrazen dotaz, zda se má tento soubor přepsat či nikoliv (v tom případě je potřeba změnit název nebo cestu souboru).

Při úspěšném zadání je na dané cestě vytvořen soubor se zadaným názvem a příponou *.vlap.

3.3.4 Exit

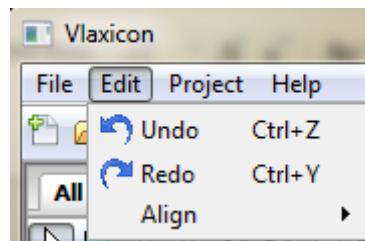
Tato volba slouží k ukončení programu. Stejně tak jako standardní tlačítko pro uzavření okna aplikace. A rovněž lze ke stejnému účelu použít klávesové zkratky ALT + X. Pokud před takovýmto ukončením některé z projektů nebyly uloženy, zobrazí program dotaz na uložení pro každý takovýto neuložený projekt zvlášť.



Obrázek 12: Hláška s dotazem na uložení projektu

3.4 Menu Edit

Toto menu obsahuje volby pro úpravy momentálně vybraného projektu. Jedná se především o úpravy vizuální.



Obrázek 13: Menu Edit

3.4.1 Undo

Tato volba je určena pro vrácení poslední změny v aktuálně vybraném projektu. Rovněž je dostupná z hlavní lišty nástrojů anebo pomocí klávesové zkratky CTRL + Z.

3.4.2 Redo

Volba určená pro obnovení poslední vrácené změny v aktuálním vybraném projektu. Dostupná je tato volba rovněž z hlavní lišty nástrojů a pomocí klávesové zkratky CTRL + Y.

3.4.3 Align

Slouží pro nastavení zarovnání při současném výběru více prvků na pracovní ploše aktivního projektu. Toto menu obsahuje další volby, kterými jsou:

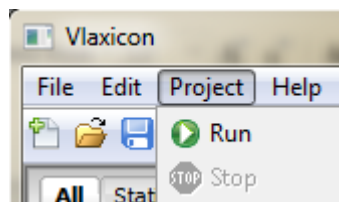
- *Left* – zarovnání prvků doleva
- *Center* – zarovnání prvků na střed vertikálně
- *Right* – zarovnání prvků doprava
- *Top* – zarovnání prvků nahoru
- *Middle* – zarovnání prvků na střed horizontálně
- *Bottom* – zarovnání prvků dolů

Všechny tyto možnosti jsou dostupné i z hlavní nástrojové lišty, ale neexistují pro ně klávesové zkratky.

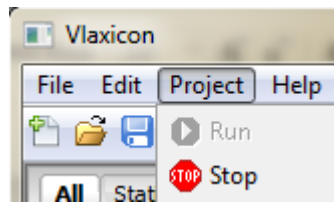
3.5 Menu Project

Toto menu slouží pro správu projektu, lépe řečeno pro přepínání mezi režimem editace a režimem ovládání.

V režimu editace je možné vkládat na pracovní plochu projektu ovládací prvky a tyto konfigurovat pomocí jejich vlastností. Tento režim je nastaven jako výchozí. V režimu ovládání jsou přístroje pomocí prvků ovládány.



Obrázek 14: Menu Project v režimu editace, kdy je možné přepnutí do režimu ovládání.



Obrázek 15: Menu Project v režimu ovládání, kdy je možné se přepnout do režimu editace.

3.5.1 Run

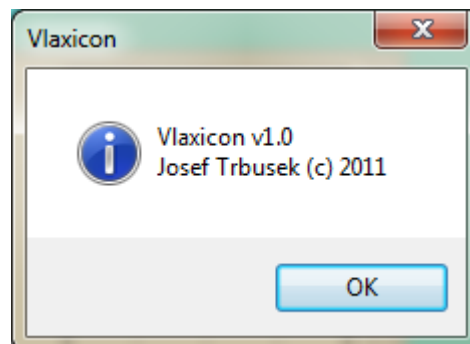
Tato položka slouží pro přepnutí do režimu ovládání. Po zvolení probíhá kontrola, zda byly na prvcích nastaveny všechny potřebné parametry a zda je možné se připojit k přístrojům. Pokud dojde k chybě je zobrazena chybová hláška a případně je označen prvek, kvůli kterému došlo k chybě.

3.5.2 Stop

Tato volba přepíná projekt do editačního režimu, kdy je zastavena komunikace s přístroji a je možné opět libovolně upravovat aktivní projekt

3.6 Menu Help

Toto menu obsahuje pouze volbu *About*, která zobrazuje dialog s informacemi o programu (viz obrázek níže).



Obrázek 16: Okno informující o programu.

3.7 Vkládané prvky

Jsou prvky v záložkách pod hlavní lištou nástrojů. Tyto záložky jsou členěny podle typu prvků. Jsou zde tyto záložky:

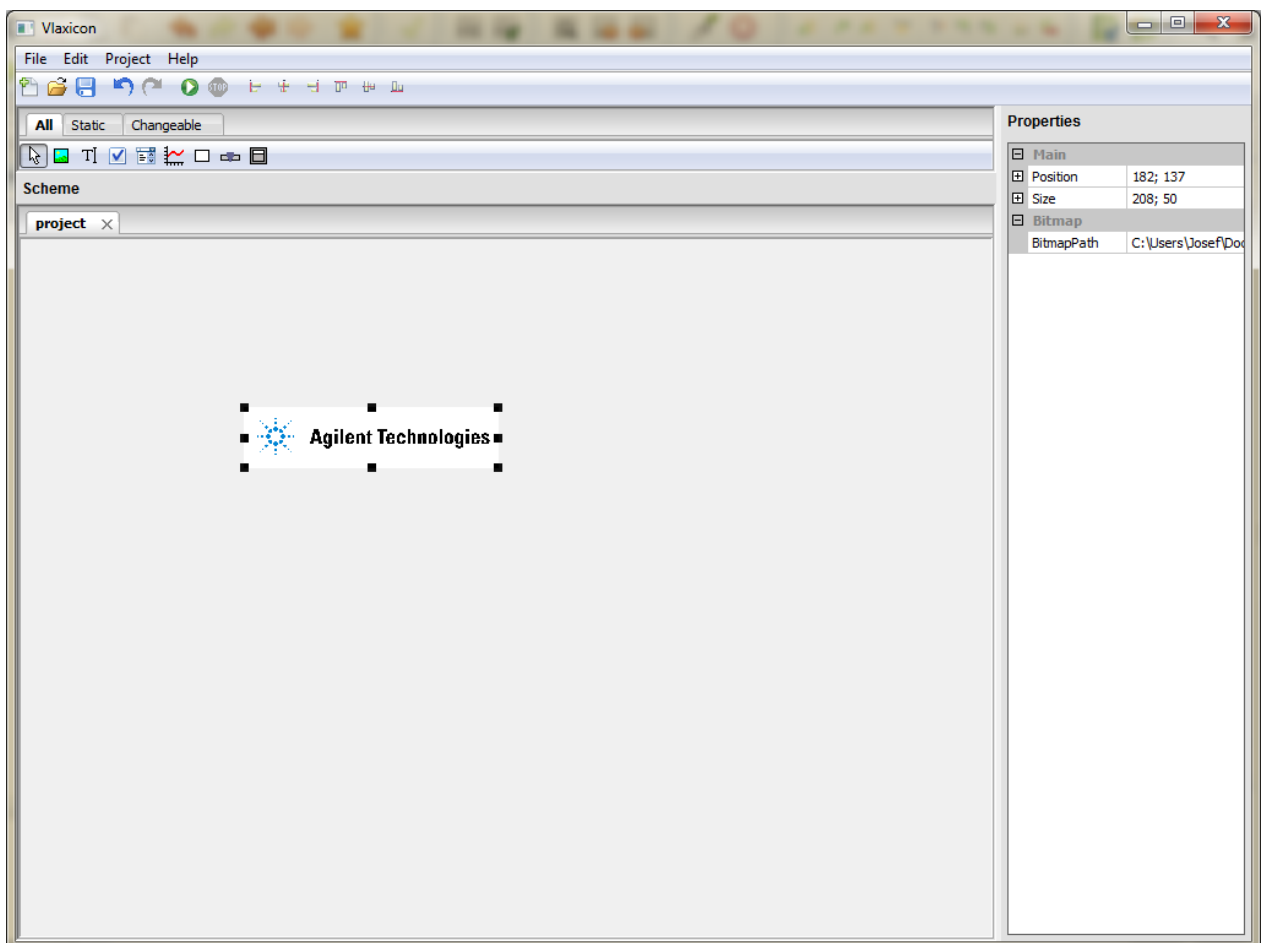
- *All* – obsahuje všechny možné prvky
- *Static* – obsahuje prvky, s kterými nelze během režimu ovládání manipulovat
- *Changeable* – obsahuje prvky, které lze během režimu ovládání nastavovat.

Prvek se vloží tak, že se vybere a levým tlačítkem myši se klikne na plochu projektu, kam se má prvek vložit. Pokud je při vkládání na pracovní plochu projektu navíc držena klávesa CTRL, pak se po vložení nástroj nepřepne na nástroj manipulace s prvky.

3.7.1 Bitmap

Je prvek zařazený mezi statické prvky. Jedná se o vložení bitmapového obrázku.

Při vkládání na pracovní plochu projektu se zobrazí dialog pro zadání cesty k bitmapovému souboru. Po jeho zadání se bitmapa zobrazí na pracovní ploše v místě, kde bylo kliknuto.



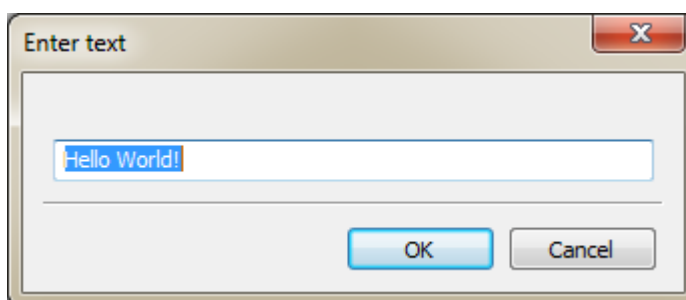
Obrázek 17: Okno programu s vloženým prvkem typu Bitmap

Po vložení prvku lze tento v editačním režimu upravit pomocí tabulky vlastností. Jak lze i vidět na obrázku výše, pomocí této tabulky lze editovat polohu a velikost prvku a dále obrázek zdrojový soubor obrázku.

3.7.2 Edit Text

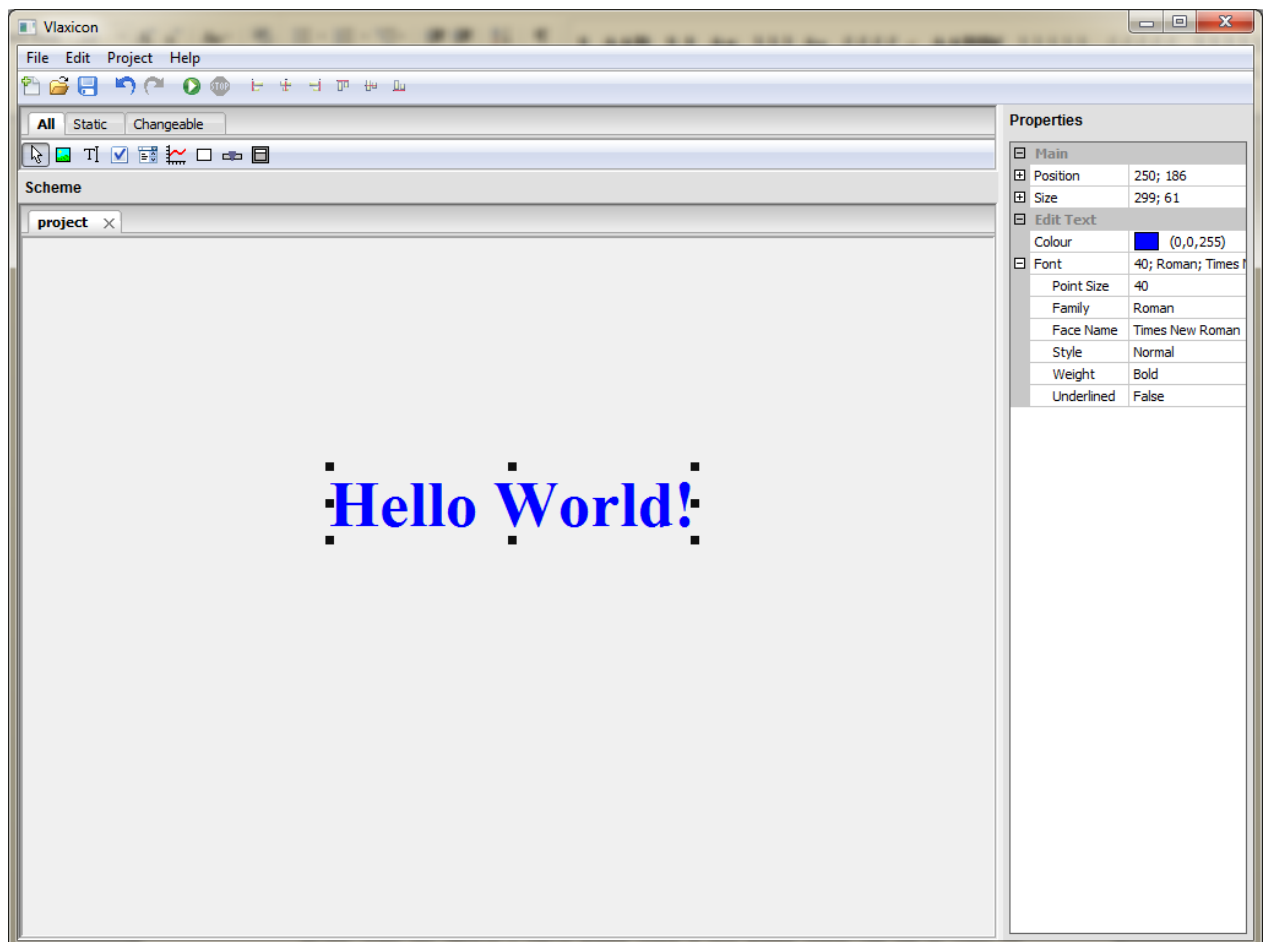
Jde o statický prvek textu, kdy text lze editovat dvojklikem na prvek.

Při kliknutí pro vložení na plochu se zobrazí dialog (obrázek níže) pro zadání výchozího textu prvku.



Obrázek 18: Dialog pro zadání textu prvku Edit Text

Po potvrzení textu prvku se prvek s tímto textem vloží na pracovní plochu projektu do místa, kde bylo kliknuto. Pomocí tabulky vlastností lze pak editovat polohu a velikost prvku. Dále lze upravit i barvu písma textu a styl písma textu, tak jak je vidět na následujícím obrázku.



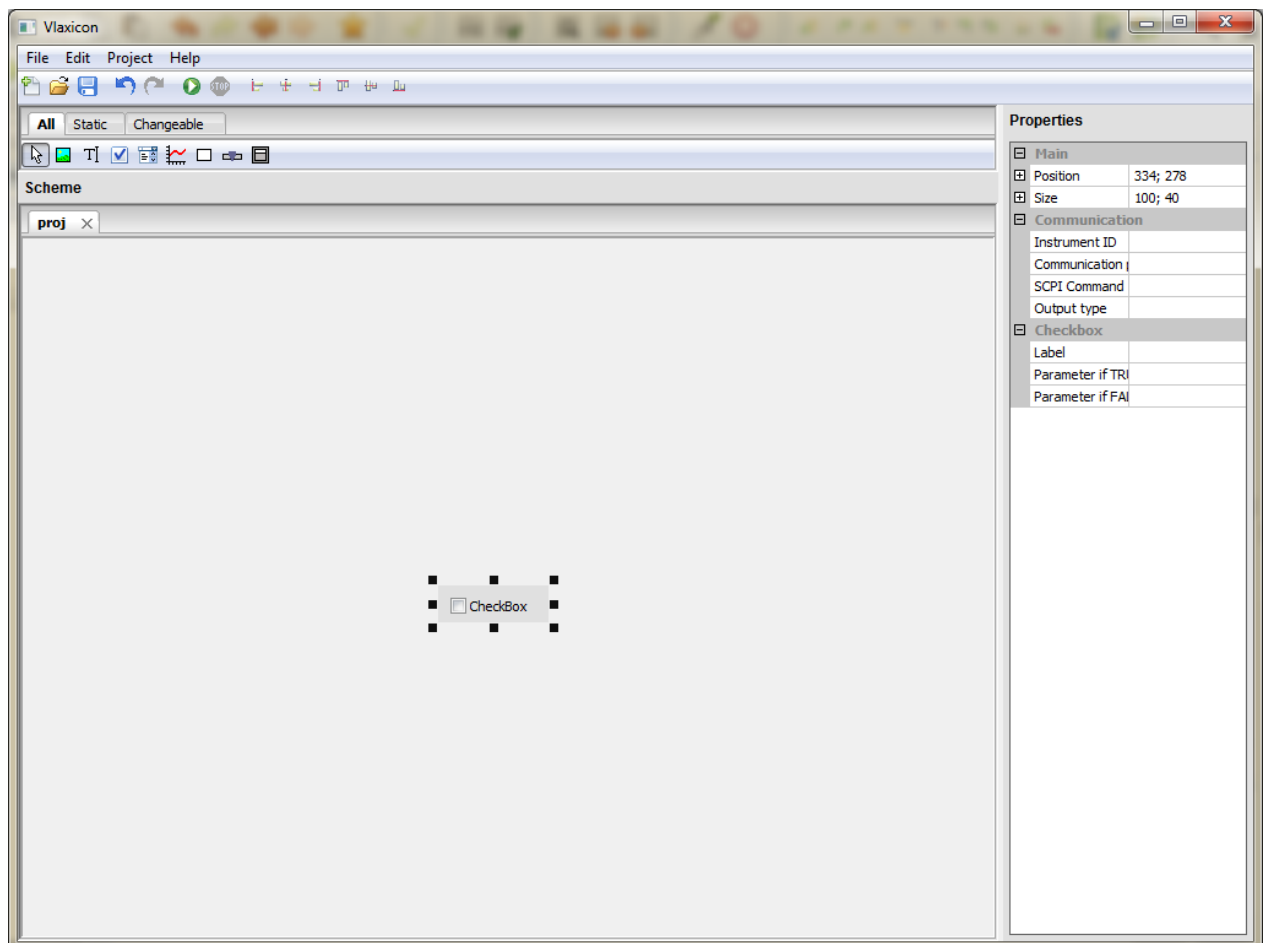
Obrázek 19: Okno programu s vloženým a upraveným prvkem Edit Text

3.7.3 Check Box

Jde o prvek zařazený mezi *Changeable*. Jde o klasické zaškrtnutí, kdy v tomto případě se pomocí něj přepíná mezi dvěma nastavenými parametry. Při vložení je tento prvek nastaven jako odškrtnutý.

Po vložení lze v tabulce vlastností nastavit kromě pozice a velikosti také *Instrument ID*, což je adresa přístroje pro připojení a ovládání (URL, IP adresa, VISA adresa, atd...), *Communication plugin*, což je plugin, pomocí kterého bude komunikace s přístrojem prováděna. Ten je potřeba volit podle toho pomocí jakého rozhraní je přístroj připojena a podle operačního systému. *Instrument ID* a *Communication plugin* jsou položky, které musejí být nastaveny, aby bylo možné s přístrojem pomocí prvku komunikovat. Dále je možné nastavit *SCPI Command*, což je příkaz, který bude prvek vykonávat. Pokud se bude jednat o dotaz do přístroje, pak je potřeba nastavit i *Output type*. U tohoto prvku je rovněž vhodné nastavit položky *Parametr if TRUE* a *Parametr if FALSE*, které udávají parametr

příkazu pro přístroj (zadaný jako *SCPI Command*) pro případ zaškrtnutí (TRUE) a odškrtnutí (FALSE). Posledním možným nastavením je *Label*, kterým se nastavuje popisek zaškrtnutí.

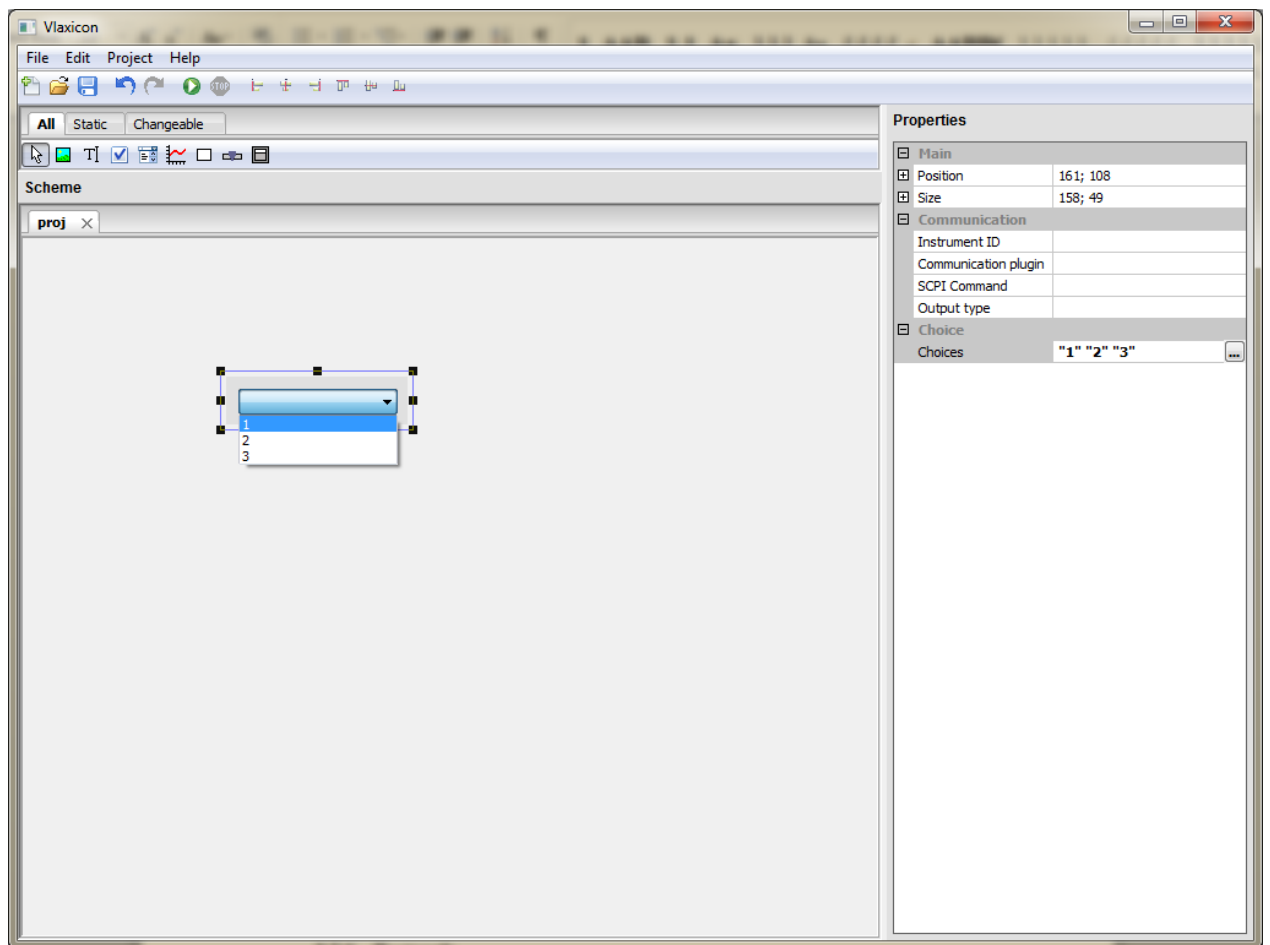


Obrázek 20: Okno s vloženým prvkem Check Box

3.7.4 Choice

Je také prvkem z kategorie *Changeable*, který je roletovým menu pro výběr jedné z nabízených možností, která je použita jako parametr příkazu.

Po vložení lze upravit vlastnosti pozice, velikosti, je potřeba zadat parametry pro komunikaci, tj. *Instrument ID*, *Communication plugin SCPI Command* a případně *Output type* obdobně jako je tomu u prvku Check Box. Pomocí položky vlastností *Choices* se pak nastavují samotné možnosti prvku pro výběr.

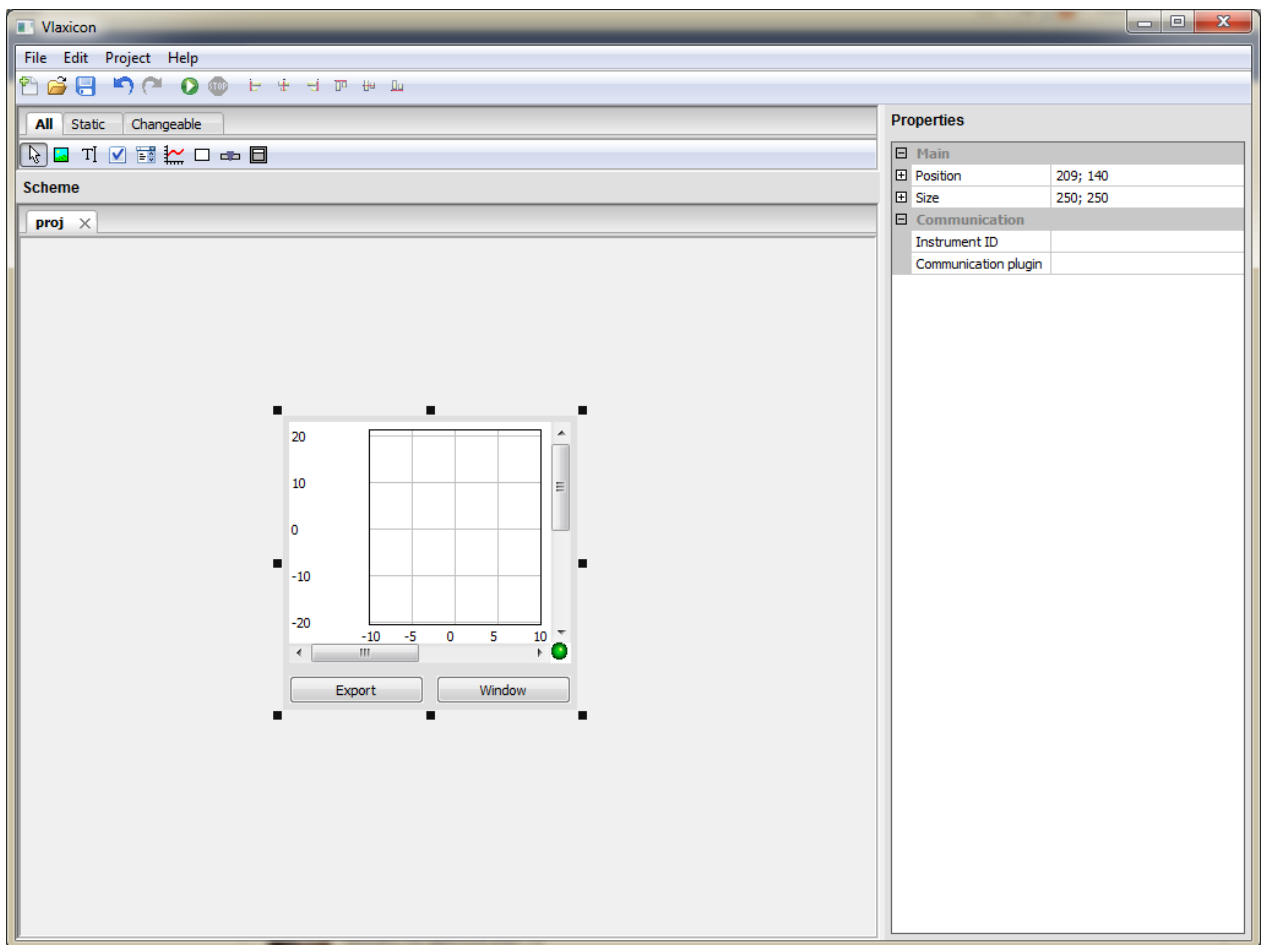


Obrázek 21: Okno programu s vloženým prvkem Choice s nastavenými volbami.

3.7.5 Plot

Ačkoliv tento prvek komunikuje s přístroji, je zařazen do kategorie *Static*, jelikož se mu nenastavují příkazy.

U tohoto prvku lze pomocí vlastností nastavit pouze polohu, velikost a komunikační parametry *Instrument ID* a *Communication plugin*.

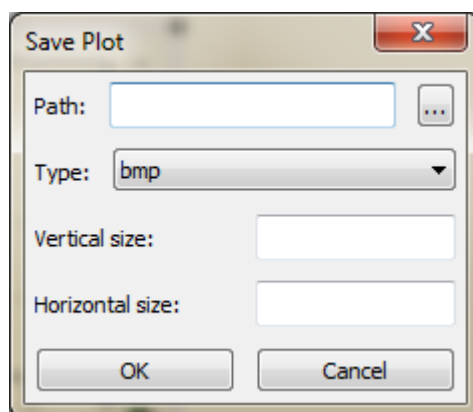


Obrázek 22: Okno programu s vloženým prvkem Plot

Tlačítko *Export*, které je součástí prvku, slouží pro uložení obsahu grafu do souboru. Při jeho stisku se objeví dialog níže, ve kterém se postupně nastavují tyto parametry pro uložení obrázku:

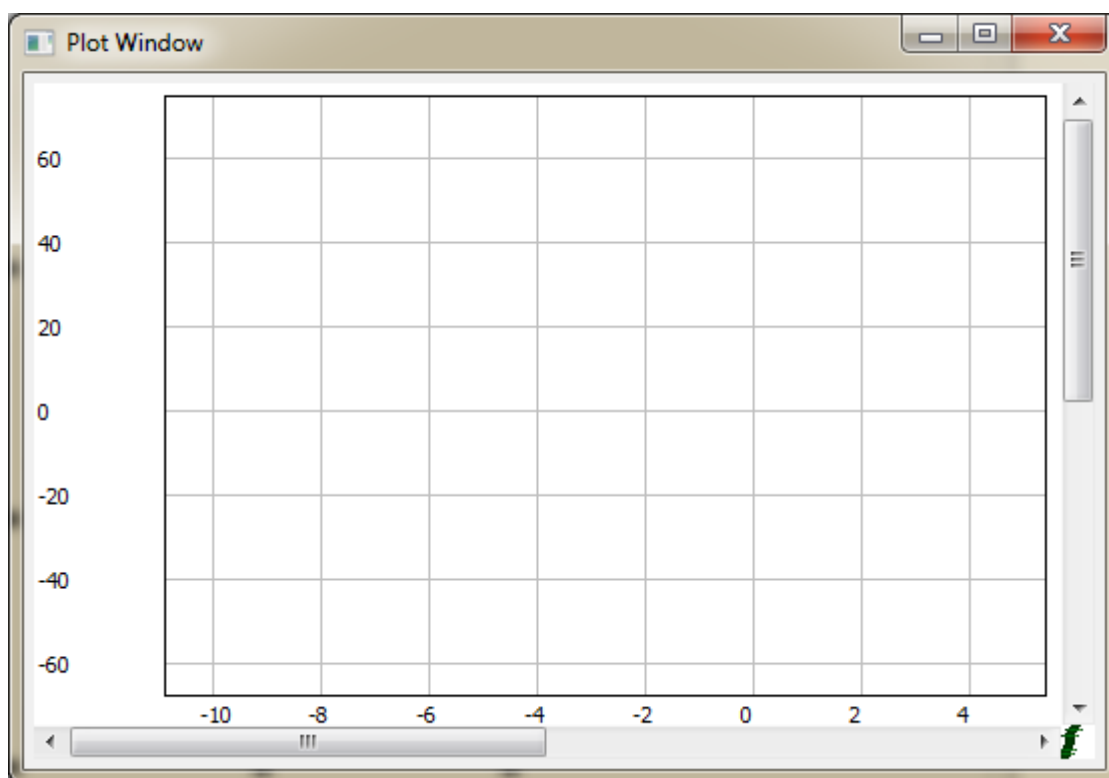
- *Path* – cesta a název k souboru
- *Type* – typ souboru. Jsou zde možnosti uložení do rastrového formátu bitmapového souboru (BMP) nebo vektorového formátu SVG souboru
- *Vertical size* – vertikální velikost ukládaného obrázku v pixelech
- *Horizontal size* – horizontální velikost ukládaného obrázku v pixelech

Po úspěšném uložení grafu do souboru program o tomto informuje zprávou.



Obrázek 23: Dialog pro uložení grafu do obrázkového souboru.

Tlačítko *Window*, které je rovněž součástí prvku, slouží pro vyvolání samostatného okna zobrazujícího graf. S tímto oknem lze volně manipulovat ve smyslu pozice a velikosti.

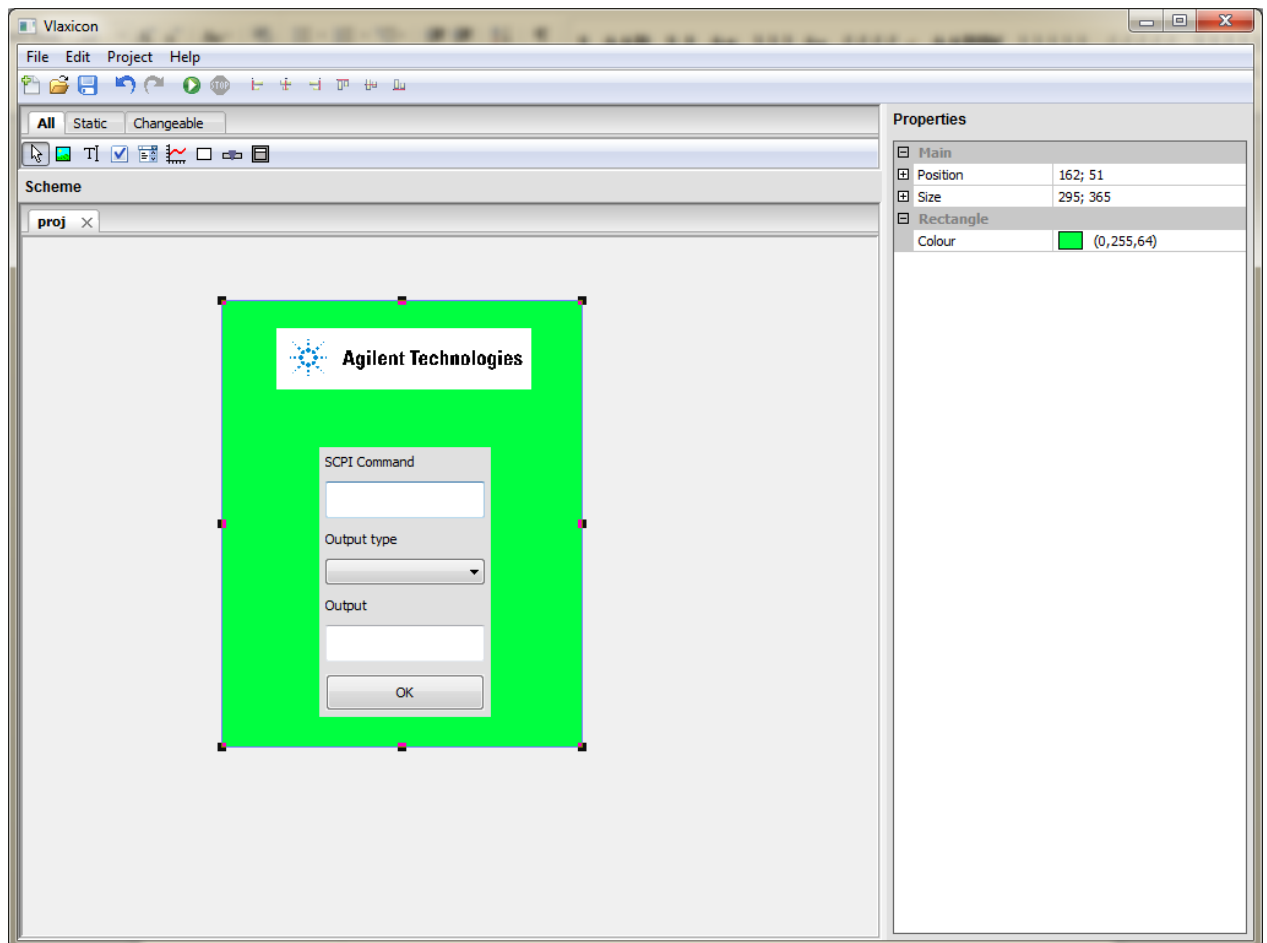


Obrázek 24: Samostatné okno pro vykreslování grafu

3.7.6 Rectangle

Tento statický prvek slouží pro vykreslení obdélníku, který je schopen seskupovat veškeré ostatní prvky. S těmito lze pak pomocí manipulace s tímto prvkem manipulovat hromadně.

U prvků tohoto typu lze nastavit, pozici, velikost a barvu.



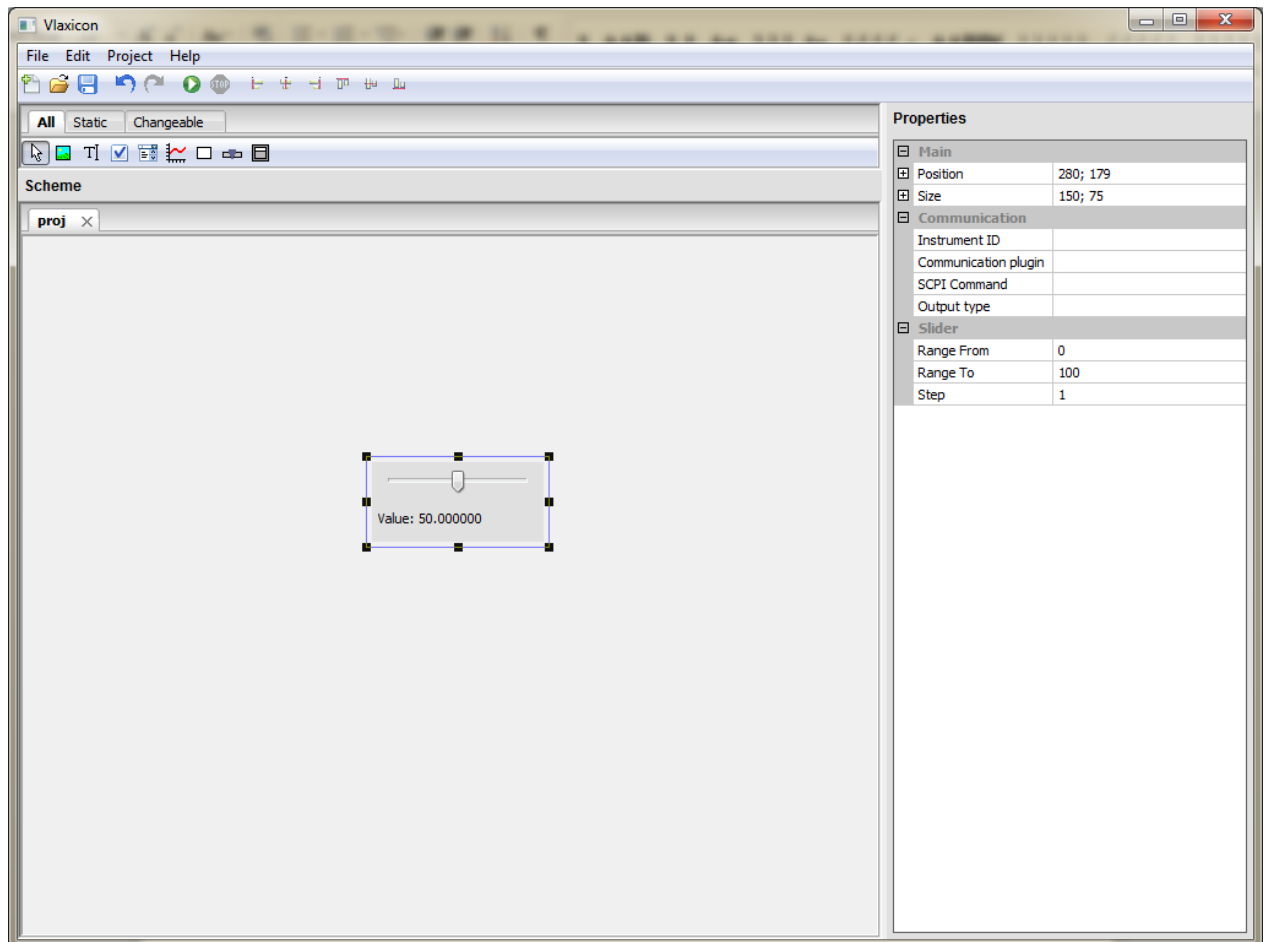
Obrázek 25: Okno programu s vloženým prvkem Rectangle s upravenými vlastnostmi a obsahujícího jiné ovládací prvky

3.7.7 Slider

Jde o prvek z kategorie *Changeable*, který slouží pro změnu parametru příkazu v určitém rozsahu číselných hodnot.

Po vložení tohoto prvku lze u něj nastavit vlastnosti polohy, rozměru, potřebných komunikačních parametrů *Instrument ID*, *Communication plugin*, *SCPI Command* a případného *Output type*. U tohoto prvku lze také nastavit *Range from*, což je minimální

bod rozsahu, *Range to* maximální bod rozsahu a *Step* který udává velikost jednoho kroku při tažení táhlem prvku.



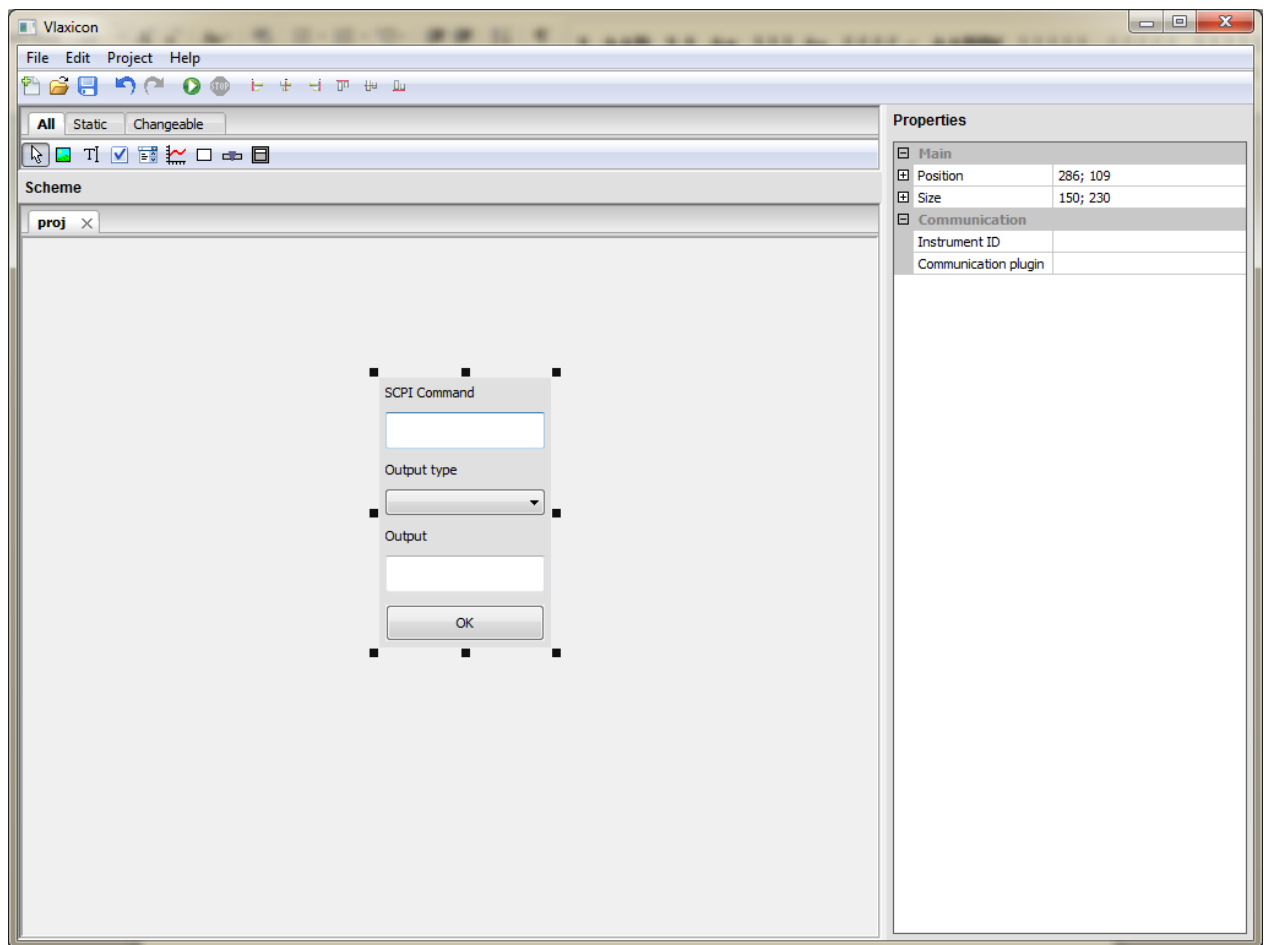
Obrázek 26: Okno programu s vloženým prvkem Slider

Po přepnutí do režimu ovládání prvek funguje tak, že odesílá do přístroje příkaz v pravidelném intervalu kvůli tomu, aby nebyl přístroj přetěžován při rychlých a častých změnách parametru posunem táhla.

3.7.8 Universal

Tento prvek z kategorie Changeable je jakýmsi univerzálem pro plnění veškerých příkazů nebo dotazů na přístroj.

Po vložení lze pomocí vlastností nastavit polohu velikost a komunikační parametry *Instrument ID* a *Communication plugin*. Zbývající parametry se nastavují přímo na prvku a lze je tedy měnit během režimu ovládání.



Obrázek 27: Okno programu s vloženým prvkem universal

Na prvku lze tedy už v režimu ovládání nebo editace nastavit parametry *SCPI Command* a *Output type*. V tomto případě však *SCPI Command* musí být kompletní včetně případného parametru příkazu.

V režimu ovládání se pak příkaz (nebo dotaz) odešle do přístroje pomocí tlačítka *OK*. Pokud jde o dotaz a vše proběhne správně, reakce přístroje se zobrazí v poli *Output*.

3.8 Pracovní plocha programu

Jak již bylo zmíněno, v programu může být více nezávislých pracovních ploch pro různé projekty. Každá pracovní plocha projektu je jednou záložkou, mezi kterými lze libovolně přepínat, měnit jejich pořadí, případně je i zavřít.

Hlavní manipulace na takovéto pracovní ploše probíhá pomocí myši:

- Kliknutím levým tlačítkem na prvek je prvek vybrán a lze jej upravovat pomocí vlastností.
- Kliknutím levým tlačítkem do prázdného prostoru se vloží zvolený prvek, pokud je nějaký vybrán.
- Tažením se stisknutým levým tlačítkem za vyznačené body prvku prvek mění velikost.
- Tažením se stisknutým levým tlačítkem za okraj prvku se prvek přesunuje.
- Dvojklikem lze aktivovat změnu textu u vloženého prvku Edit Text.

Kromě myši se využívá klávesa DELETE pro smazání zvoleného objektu.

Se záložkami jsou možné tyto manipulace, kromě vytvoření které bylo popsáno již dříve:

- Zavření pomocí stisknutí tlačítka pro zavření umístěného přímo na záložce. V takovém případě program kontroluje, jestli byl daný projekt uložen s posledními změnami a pokud ne zobrazí dialog s dotazem na uložení (obdobně jako tomu je při kontrole při uzavření celého programu).
- Změna pořadí uspořádání záložek je možná přetažením dané záložky na požadovanou pozici.

3.9 Tabulka vlastností

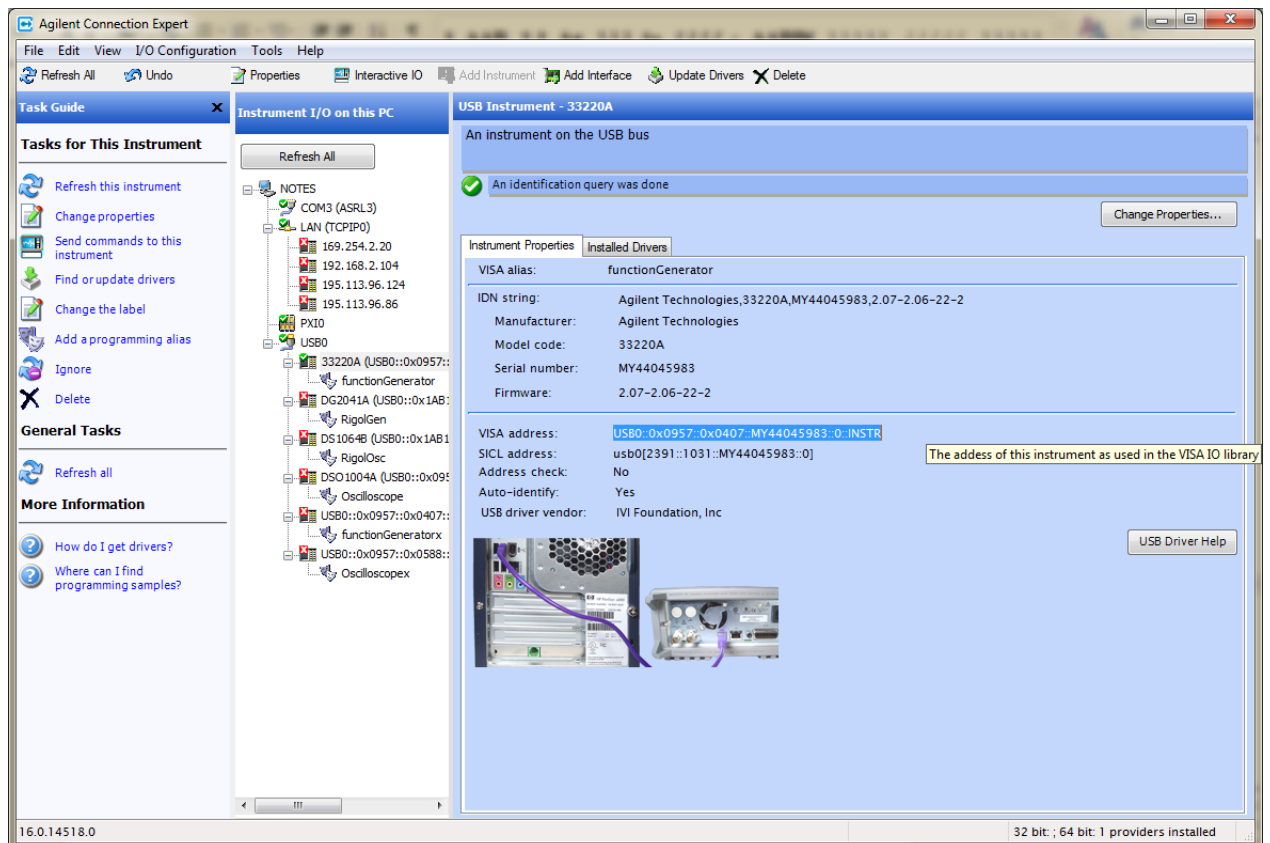
Je tabulka v pravé části okna programu, pomocí které se nastavují jednotlivé parametry prvků. U této tabulky lze manipulovat tažením myši za její levý okraj s její velikostí na úkor pracovní plochy.

Jednotlivé položky tabulky se načítají vždy po vybrání prvku podle typu tohoto vybraného prvku. Pokud není vybrán žádný prvek tabulka je prázdná. Jednotlivé vlastnosti, které lze u každého prvku nastavit byly popsány již výše.

3.9.1 Nastavení Instrument ID

Vzhledem k tomu, že nastavení tohoto parametru není zrovna jednoduché je mu věnována tato část.

Pod systémem Microsoft Windows je pro běh programu nutný software Agilent IO Libraries Suite. Součástí tohoto úprogramu je i Agilent Connection Expert. Zde lze nalézt potřebný parametr VISA Address. V tomto programu stačí jen vybrat patřičný přístroj kliknutím myši a ve vlastnostech již lze nalézt potřebný parametr. Lze též použít parametr VISA alias, který je však dostupný pouze pro zařízení připojená přes USB.



Obrázek 28: Okno programu Agilent Connection Expert se zvýrazněným parametrem VISA address.

Pod systémem Linux je pro komunikaci přes VXI (TCP/IP) pomocí pluginu LinuxVXI potřeba zadat IP adresu přístroje, kterou lze zpravidla zjistit přímo na přístroji pomocí Utility -> I/O -> LAN -> Current Config. Pro komunikaci přes USB (plugin LinuxUSB) je potřeba zadat port usbtmc ke kterému je zařízení připojeno. K dispozici je usbtmc1 až usbtmc10.

4 PROGRAMÁTORSKÁ DOKUMENTACE

Tato část je věnována popisu vytvářeného programu z hlediska programátora, tj. pomocí čeho a jak byl program vytvářen, jaký je princip činnosti jednotlivých funkcí.

4.1 Prostředky pro tvorbu programu

Program byl vytvářen v programovacím prostředí CodeLite s využitím překladače MinGW a multiplatformní knihovny wxWidgets. Pro tvorbu bylo využito taktéž komponent, které nejsou přímou součástí wxWidgets a to wxShapeFramework, wxPropertyGrid, wxPlotCtrl a wxSVG. Pro tvorbu pluginů pro komunikaci bylo pro systém Microsoft Windows využito knihoven VISA, které jsou součástí instalace Agilent IO Libraries Suite a jsou schopny komunikovat jak přes USB tak TCP/IP. Pro komunikaci pod systémem Linux byly použity ovladače USBTMC (pro USB) a VXI11 (přes TCP/IP).

4.2 Struktura programu

Obrázek níže znázorňuje diagram tříd, které byly v rámci projektu vytvořeny, a jejich vzájemných závislostí.

Na tomto diagramu lze strukturu programu dobře popsat. Ve spodní části diagramu je třída *MainFrame*, která zapouzdřuje hlavní okno programu. Tato třída dědí ze třídy *MainFrameBase*, která je generována pomocí programu wxFormBuilder a proto není vhodné do ní dělat zásahy. Tyto třídy vytváří design hlavního okna a třída *MainFrame* se kromě mnoha jiných ovládacích prvků skládá z pracovních ploch, kterých může být otevřených libovolné množství podle uživatelových potřeb.

Třídou pro pracovní plochu je *FrameCanvas*, která zapouzdřuje operace pro tuto plochu a dědí velkou část funkcionalitu z třídy *wxSFShapeCanvas*, která je součástí wxShapeFramework. Na tuto plochu jsou vkládány prvky odvozené od třídy *wxSFShapeBase* (taktéž součást wxShapeFramework). V tomto programu jsou to prvky tříd *wxSFBitmapShape*, *wxSFEditTextShape*, *wxSFRectShape* a *ObjectBaseShape*. Poslední zmíněná třída je vytvořena v tomto projektu, ostatní jsou standardní součástí wxShapeFramework.

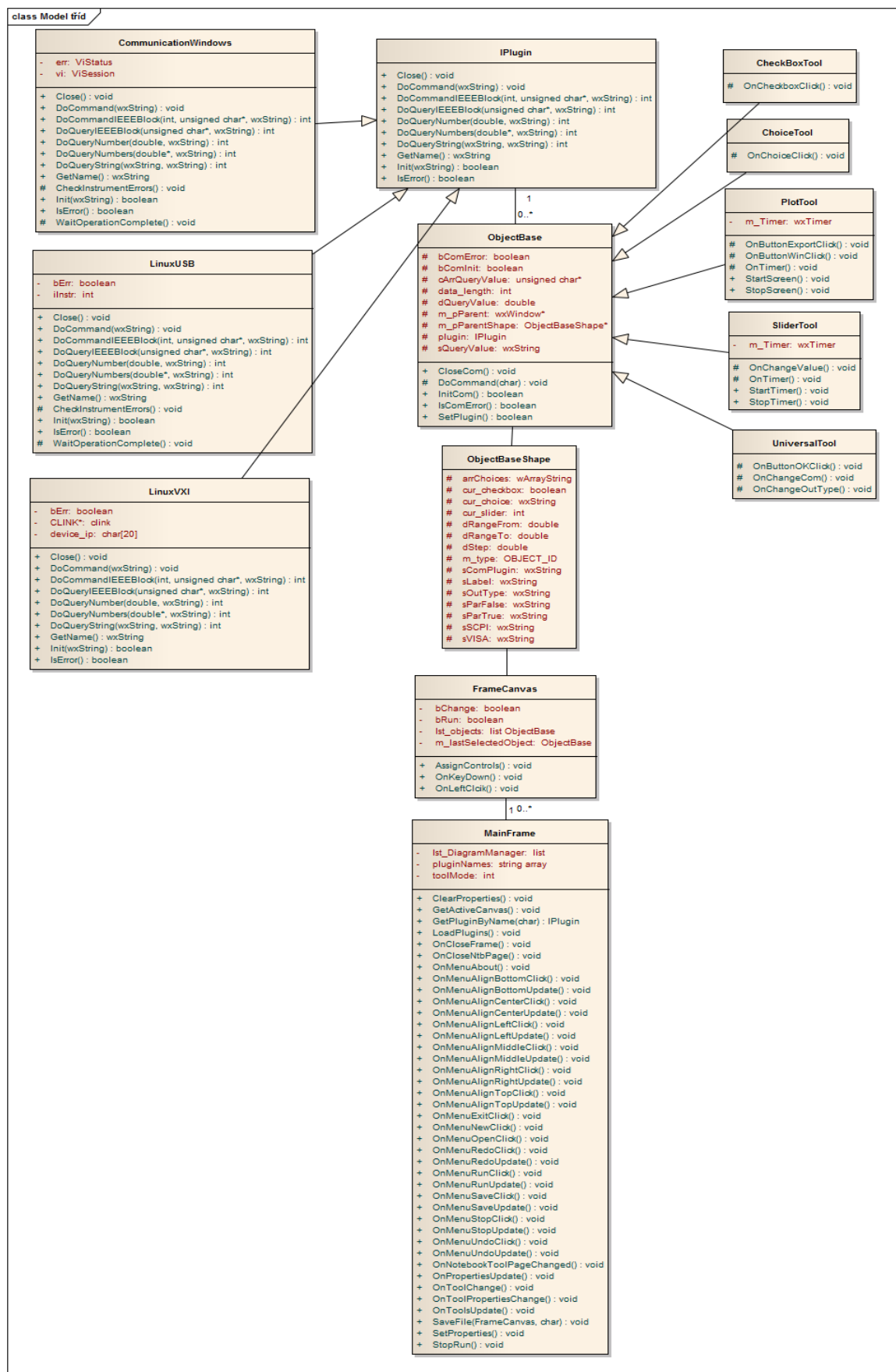
Třída *ObjectBaseShape* je vytvořena pro ukládání hodnot jednotlivých specifických kontrolních prvků, pro které jsou vytvořeny vlastní třídy a do tohoto shapu jsou vkládány jako jeho součást. Hlavním účelem této třídy je tedy uchovávání údajů těchto prvků pro ukládání a načítání ze souboru. Tato třída je založena na třídě *wxSFControlShape*.

Třídě *ObjectBaseShape* se nastaví kontrolní prvek podle typu, který je v ní udán. Základem pro všechny tyto prvky je třída *ObjectBase*, která obsahuje společné operace pro všechny prvky a tudíž jednotlivé konkrétní prvky jsou od ní odvozeny. Těmito třídami jednotlivých prvků jsou:

- *CheckboxTool* – prvek Check Box
- *ChoiceTool* – prvek Choice
- *PlotTool* – prvek Plot
- *SliderTool* – prvek Slider
- *UniversalTool* – prvek Universal

Třída *ObjectBase* pak obsahuje načtený plugin pro komunikaci s přístrojem. Pro pluginy je vytvořena interface třída *IPlugin*, což znamená, že tato třída pouze deklaruje společné funkce pro všechny pluginy, které jsou implementovány ve třídách jednotlivých pluginů, které mohou obsahovat i svoje další potřebné funkce a atributy. Od interface *IPlugin* jsou tedy odvozeny třídy pluginů:

- *Plugin* – plugin pro komunikaci pod systémem Microsoft Windows
- *LinuxVXIPlugin* – plugin pro komunikaci přes TCP/IP pod systémem Linux
- *LinuxUSBPlugin* – plugin pro komunikaci přes USB pod systémem Linux



Obrázek 29: Diagram znázorňující strukturu a závislost tříd projektu

4.3 Správa hlavního okna

Zásadními proměnnými hlavního okna pro celý projekt jsou:

- Proměnná *lst_DiagramManager*, která je typu `list<wxSFDDiagramManager*>` a je seznamem všech momentálně otevřených projektů v programu
- Proměnné *m_arrPluginNames* a *m_arrPluginAlias* (obě typu `wxArrayString`), kdy *m_arrPluginNames* shromažďuje cesty k dostupným pluginům a *m_arrPluginAlias* shromažďuje jejich názvy, pod kterými vystupují v programu. U těchto proměnných si odpovídá indexace položek, tj. pro jeden plugin je na stejném indexu pole cesta a název v programu.

V konstruktoru třídy hlavního okna je kromě funkcí upravujících počáteční hodnoty některých prvků volána funkce *LoadPlugins()*, která inicializuje právě položky *m_arrPluginNames* a *m_arrPluginAlias*.

V prvopočátku jsou dostupné pouze položky pro vytvoření nového projektu (*New*) nebo otevření již existujícího projektu (*Open*). Tyto jediné jsou také dostupné vždy. Pro všechny ostatní položky jsou vytvořeny funkce reagující na událost typu `wxEVT_UPDATE_UI`, která neustále kontroluje, zda nedošlo ke změně, která by upravovala daný prvek. Ve většině případů se v jednotlivých funkcích mění dostupnost prvku podle reimu ve kterém se program nachází a podle toho zda je otevřený nějaký projekt.

4.3.1 Menu

Zde jsou rozebrány funkčnosti jednotlivých položek menu, které jsou součástí hlavního okna programu. Funkce položek menu jsou společné zároveň pro odpovídající položky hlavní lišty nástrojů.

4.3.1.1 New

Položka menu *File* vytvářející nový projekt. Po jejím zvolení se před reagující událost *OnMenuNewClick* vyvolá dialog třídy *DialogNewMain*. Zde je od uživatele požadováno zadání názvu projektu. Pokud uživatel v tomto dialogu stiskne tlačítko *Cancel*, pak je dialog uzavřen a nic dalšího se neděje. Při stisku tlačítka *OK*, je nejdříve zkontrolováno, zda prvek typu `wxTextCtrl`, do kterého se název zadává, není prázdný.

Pokud tomu tak je dialog není uzavřen a uživatel musí zadat název. Pokud je název zadán a stisknuto *OK*, pak se dialog uzavře, je vytvořen nový prvek typu *wxSFDiagramManager*, kterému je přiřazen nový prvek typu *FrameCanvas*. Následně je nový *FrameCanvas* přidán jako nová záložka do *m_auinotebookScheme* (prvek záložek typu *wxAuiNotebook*, pro pracovní plochy projektů) a nový *wxSFDiagramManager* přidán do *lst_DiagramManager*.

4.3.1.2 Open

Položka menu *File* sloužící pro načtení již uloženého projektu. Využívá se zde komponenty *wxXmlSerializer* [10], která je součástí *wxShapeFramework* [11]. Tato komponenta po zavolání funkce *LoadCanvas* nad prvkem typu *FrameCanvas* provede deserializaci souboru typu *xml*, ze kterého je takto schopná načíst jednotlivé prvky a jejich vlastnosti, které byly přidány do serializace (viz třída *ObjectBaseShape* a funkce *XS_SERIALIZE*).

Prakticky toto znamená, že v reagující události *OnMenuOpenClick* je vyvolán dialog typu *wxFileDialog*, ve kterém si uživatel zvolí soubor, následně je vytvořena nová záložka *FrameCanvas* a k ní *wxSFDiagramManager* obdobně jako při vytváření nového projektu (viz výše) a nad touto záložkou je volána funkce *LoadCanvas*. Zadaný soubor je deserializován, jsou vytvořeny prvky typu *ObjectBaseShape* nebo případně jiných typů, které jsou uvedeny výše v části o struktuře programu. Jelikož samotný *ObjectBaseShape* neobsahuje část s kontrolními prvky je volána nad záložkou ještě funkce *AssignControls*, která prochází všechny prvky *ObjectBaseShape* vložené na záložce a podle typu, který je v nich uveden, dotváří kontrolní prvky.

4.3.1.3 Save

Tato položka v menu *File* je vlastně opakem položky *Open*. Provádí uložení zvoleného projektu do souboru.

Prakticky se provede to, že uživateli je zobrazen dialog *wxFileDialog* pomocí kterého zadá název souboru a cestu, kam se má soubor uložit. Následně je zjištěn momentálně vybraná záložka pomocí funkce hlavního okna *GetActiveCanvas*. Nad touto

záložkou je pak volána funkce *SaveCanvas*, která provede serializaci prvků na záložce a zápis do zvoleného souboru

4.3.1.4 *Exit*

Položka menu *File*, která ukončuje program. Před samotným ukončením je však nutné, aby bylo zkontrolováno, zda všechny otevřené projekty byly uloženy s posledními změnami. Toho je dosaženo tak, že třída *FrameCanvas* obsahuje proměnnou *bChange* typu *bool*, která značí, zda daný projekt byl po posledních změnách uložen či nikoliv. Tuto proměnná je nastavena na hodnotu *true* (tj. byly provedeny změny) ve všech funkcích, které jakkoliv mění vlastnosti prvků projektu. Na *false* tuto proměnnou nastavuje uložení projektu.

Kontrola je tedy provedena tak, že jsou procházeny všechny záložky a zjišťována hodnota proměnné *bChange*. Pokud tato je *true* pak je vyvolán dialog typu *wxMessageBox* dotazující se na uložení daného projektu. Pokud uživatel zvolí *Ano*, pak je projekt uložen stejně jako tomu je při možnosti *Save*.

Dále je před ukončením kontrolováno, zda program není v režimu ovládání. V tomto případě je potřeba opět projít všechny záložky, a pokud se daný projekt nachází v režimu ovládání, je volána funkce *StopRun*, která zajistí ukončení všech timerů prvků a uzavření spojení s přístrojem.

4.3.1.5 *Undo*

Položka menu *Edit*, která vrací zpět poslední změnu ve zvoleném projektu. O celou funkci se zde stará funkce *Undo*, která je volána nad získaným aktivním projektem pomocí *GetActiveCanvas*.

4.3.1.6 *Redo*

Položka menu *Edit*, která obnovuje poslední vrácenou změnu ve zvoleném projektu. O celou funkci se zde stará funkce *Redo*, která je volána nad získaným aktivním projektem pomocí *GetActiveCanvas*.

4.3.1.7 *Align*

Je položka menu *Edit*, která se stará o zarovnání prvků. Toto menu obsahuje další položky podle typu zarovnání.

Princip činnosti je ten, že je získán pomocí *GetActiveCanvas* aktivní projekt a pomocí funkce *AlignSelected* jsou prvky patřičně zarovnány. To jak jsou zarovnány, určují parametry funkce *AlignSelected* pro vertikální a horizontální zarovnání.

4.3.1.8 *Run*

Položka menu *Project*, která přepíná z režimu editace do režimu ovládání.

Princip činnosti v tomto případě je ten, že v aktivním projektu jsou procházeny postupně všechny prvky typu *ObjectBaseShape* a u těchto je kontrolováno, zda je nastaven plugin pro komunikaci. Pokud ano, pak je tento přímo přiřazen danému prvku. Dále je kontrolováno zadání identifikačního řetězce přístroje, s kterým má prvek komunikovat. Pokud je tento zadán, pak je pomocí již přiřazeného pluginu provedena inicializace spojení. Pokud cokoliv neprojde kontrolou, je zbytek nastavování přerušeno, již nastavené inicializace jsou vráceny zpět pomocí funkce *StopRun* a prvek u kterého došlo k chybě je označen a vyvolána zpráva typu *wxMessageBox* s danou chybou. Taktéž jsou zde u prvků, které obsahují timery, tyto timery spuštěny

4.3.1.9 *Stop*

Položka menu *Project*, která přepíná z režimu ovládání do režimu editace. Provádí se zde funkce *StopRun*, která projde všechny prvky a uzavře u nich komunikaci s přístrojem a případně zastaví u prvků, které mají timery, tyto timery

4.3.1.10 *About*

Položka menu *Help*. Vyvolá okna zprávy *wxMessageBox* s výpisem základních údajů o programu.

4.3.2 **Nastavování vlastností prvků**

Vlastnosti lze nastavovat v tabulce vlastností, která je tvořena komponentou *wxPropertyGrid*. Nastavení této tabulky se provádí voláním funkce *SetProperties()*, která

si zjistí z aktivního projektu momentálně vybraný prvek a podle jeho typu vytvoří potřebné položky pro jeho nastavení.

Uživatel poté může hodnoty těchto položek měnit. Na jakékoliv změny v tabulce vlastností reaguje funkce *OnToolPropertiesChange*, která podle názvu změněné položky provede potřebnou změnu nastavení vybraného prvku.

4.4 Správa projektů

4.4.1 Manipulace se záložkami

Jak již bylo zmíněno, projekty jsou zobrazeny pomocí prvku typu *wxAuiNotebook* v záložkách. Díky tomuto prvku je i možná manipulace se záložkami. Tento prvek samostatně ošetřuje možnost změny pořadí jednotlivých záložek pomocí přetažení myši.

Dále umožňuje i zavírání jednotlivých záložek, ovšem pro účel tohoto programu je třeba při této události provést ještě několik věcí. Je zde potřeba zkontrolovat zda byl projekt uložen a zda není v režimu ovládání. Toto je řešeno obdobně jako při uzavírání celého programu, které bylo popsáno výše. Pomocí proměnné *bChange* je zjištěno, zda byly provedeny změny projektu od jeho posledního uložení. Pokud ano, je vyvolán dotaz na uložení a případné uložení projektu. Při zjištěném režimu ovládání, je volána funkce *StopRun* hlavního okna, která režim ovládání zastaví.

Mimo toto je nutné i odstranění patřičného prvku *wxSFDiagramManager* ze seznamu těchto prvků v třídě hlavního okna.

4.4.2 Manipulace s prvky projektu

Kromě toho, že s prvky lze manipulovat pomocí tabulky vlastností, lze s nimi manipulovat i pomocí myši a klávesnice přímo v záložce projektu.

Třída *wxSFShapeCanvas*, ze které dědí *FrameCanvas* přímo ošetřuje události levého tlačítka myši pro změnu velikosti a polohy prvku a označení objektů. Ve třídě *FrameCanvas* jsou pak specifikovány možnosti pro vložení prvku na levé kliknutí myši a vybrání prvku levým kliknutím myši.

Princip je ten, že při levém kliku myši, je vyvolána událost, na kterou reaguje funkce třídy *FrameCanvas* *OnLeftClick*. V této funkci je pomocí funkce

GetShapeAtPosition zjištěno, zda se na místě kliknutí nachází prvek nebo ne. Podle toho se buď označí daný prvek anebo se vloží nový prvek. Pokud tedy na dané pozici prvek je, pak se tento označí a vyvolá se událost třídy *MainFrame SetProperties* pro nastavení tabulky vlastností prvku. Pokud se na pozici, kde bylo kliknuto, nic nenachází, pak je z třídy hlavního okna zjištěno, co za prvek je vybráno v záložkách prvků, je vložen prvek typu *ObjectBaseShape*, kterému je nastaven příslušný typ kontrolního prvku a je volána funkce *AssignControls*, která kontrolní prvek doplní.

4.5 Ovládací prvky

Jak již bylo výše uvedeno, jednotlivé prvky se vkládají v reakci na stisk levého tlačítka myši a případně se dotvářejí pomocí funkce *AssignControls*. Nutno dodat, že veškeré prvky jsou vkládány jako prvky patřícího *wxDiagramManageru*, který lze z *FrameCanvasu* získat pomocí funkce *GetDiagramManager*.

4.5.1 Bitmap

Je prvek obrázku vytvářený třídou *wxSFBitmapShape*, jež je součástí *wxShapeFramework*, kdy při vložení je vyvolán dialog typu *wxFileDialog*, kde uživatel vybere soubor. Pomocí funkce *CreateFromFile* je obrázek přiřazen do prvku.

4.5.2 Edit Text

Jde o prvek textu tvořený třídou *wxSFEEditText* z *wxShapeFramework*. Při vkládání je vyvolán dialog typu *wxTextEntryDialog*, ve kterém je uživatel vyzván k zadání textu prvku. Tento je pomocí funkce *SetText* nastaven do prvku.

U tohoto prvku je odchyťována událost na dvojkliku, kdy je možné přímo v ploše projektu upravit text prvku.

4.5.3 Check Box

Je potomkem *ObjectBase* ve třídě *CheckboxTool*. Jedná se o zaškrťavátko, u kterého je odchycena událost změny stavu zaškrtnutí. Obsluhou této události je funkce třídy *CheckboxTool OnCheckboxClick*, která mění hodnotu stavu prvku a pokud je

program v režimu ovládání, pak odesílá příkaz, který prvku byl nastaven spolu s parametrem, který je dán stavem prvku, do přístroje.

4.5.4 Choice

Je zapouzdřen v třídě *ChoiceTool*, která dědí z *ObjectBase*. Jde o roletové menu s výběrem možnosti, která bude použita jako parametr příkazu přístroje. Je zde zachycena událost na změnu volby, kterou zpracovává funkce *OnChoiceClick*. V této funkci je uložena hodnota volby a pokud je program v režimu ovládání pak je odeslán příkaz s parametrem do přístroje.

4.5.5 Plot

Je prvek, který vykresluje graf. Zapouzdřen je ve třídě *PlotTool*, která dědí z *ObjectBase*.

Pokud je program v režimu ovládání, pak je spuštěno vlákno třídy *screenThread*, ve kterém je prováděno průběžné načítání dat křivky z přístroje. Tato data jsou přepočítávána do proměnné *pActiveData*, která je typu *wxPlotData*. Pomocí funkce *SendRefreshRequest*, je v hlavním vlákne vyvolána událost funkce *OnRefreshRequest*, která data vykreslí do grafu. Pokud je zvoleno uživatelem tlačítko Export pro uložení snímku grafu, pak je vyvolán dialog třídy *SavePlotDialogMain*. V tomto se nastavují údaje pro uložení do souboru. Po potvrzení pomocí OK se další operace provádí podle zvoleného typu souboru, aby bylo dosaženo úspěšného uložení.

4.5.6 Rectangle

Je prvek čtyřúhelníku, jehož funkcionalitu zapouzdřuje třída *wxSFRectShape*, která je součástí *wxShapeFramework*. Této třídě je pomocí funkce *AcceptChild* nastaveno aby akceptovala možnost vložení jakéhokoliv prvku. To znamená, že lze jakýkoliv prvek přetažením umístit do vytvořeného čtyřúhelníku.

4.5.7 Slider

Je prvek sloužící k nastavování reálné číselné hodnoty ze zadaného rozsahu. Tento prvek je zapouzdřen ve třídě *SliderTool*, která dědí z *ObjectBase*. Tento prvek obsahuje

časovač, který pokud uplyne jeho časový interval, odešle příkaz spolu s parametrem hodnoty prvku do přístroje. Odchycena je zde událost na změnu polohy táhla. Ta zastavuje a znovu spouští časovač, aby neuplynul jeho časový interval. Takto je ošetřeno, aby se příkaz neodesílal s každou změnou táhla a tím se přístroj nezahltl.

Jelikož táhlo, které je typu *wxSlider*, je v základu pouze pro celočíselné hodnoty minima, maxima a kroku, dochází zde k přepočtu, kdy aktuální hodnota je $\text{minimum} + \text{krok} * \text{pozice táhla}$.

4.5.8 Universal

Je prvek zapouzdřený ve třídě *UniversalTool*, která dědí z *ObjectBase*. Jde o prvek pro zadání libovolného příkazu nebo dotazu pro přístroj. Jsou zde zachyceny změny nastavujících prvků pro uložení stavu a pomocí funkce *OnButtonOKClick* je ošetřena událost stisku tlačítka *OK*. Tato funkce zajišťuje, že pokud je program v režimu ovládní, pak je na tento stisk odeslán příkaz do přístroje. Pokud jde o dotaz (což je rozpoznáno podle toho, že zadaný příkaz končí znakem "?"), pak je do pole *Output* vypsána odpověď z přístroje. Ta je dle zvoleného typu umístěna v proměnné stejného typu, která je součástí třídy *ObjectBase*.

4.6 Komunikace s přístroji

Jak bylo již výše zmíněno, komunikace s přístroji probíhá pomocí pluginů. Ty jsou vytvořeny na základě společného interface. Společnými funkcemi jsou:

- *Init* – inicializuje spojení s přístrojem.
- *IsError* – značí, zda nastala chyba
- *Close* – uzavírá komunikaci s přístrojem
- *DoCommand* – provedení zadaného příkazu
- *DoQueryString* – provedení dotazu na přístroj s vrácením hodnoty typu *wxString*
- *DoQueryNumber* - provedení dotazu na přístroj s vrácením hodnoty typu *double*
- *DoQueryIEEEBlock* - provedení dotazu na přístroj s vrácením hodnoty typu *unsigned char**

- *GetName* – vrací název pluginu, pod kterým plugin vystupuje v programu

CommunicationWindows je plugin určený pro komunikaci s přístroji pod systémem Microsoft Windows. Pro komunikaci je zde použito ovladače VISA, díky kterému je možné komunikovat stejně jak přes USB tak TCP/IP. Připojený přístroj je zde uložen jako proměnná typu *ViSession*. Příkaz do přístroje je odeslán pomocí funkce *viPrintf* a případná odpověď přístroje je získána pomocí *viScanf*.

LinuxVXIPlugin je plugin určený ke komunikaci přes TCP/IP pod systémem Linux. Zde se využívá ovladače VXI11. Připojený přístroj je inicializován pomocí IP adresy a uložen v proměnné třídy *CLINK*. Komunikace zde probíhá tak, že do přístroje je příkaz zaslán pomocí funkce *vxi11_send* a data z přístroje získána pomocí *vxi11_receive*. Získaná data jsou vždy typu *char** a na patřičný formát jsou až posléze převedena.

LinuxUSBPlugin je pluginem pro komunikaci pod systémem Linux přes protokol USB. Využívá se zde *usbtmc* ovladače. Zde se přístroje tváří jakoby soubory, do kterých je zapisováno a z nich čteno. Zápis je pomocí funkce *write* a čtení pomocí funkce *read*. Připojený přístroj je uložen jako prvek typu *int*.

ZÁVĚR

Teoretickou část této práce lze rozdělit na dva větší celky, a to popis použitých prostředků pro vývoj programu, který je součástí práce, a popis jiných alternativ vytvářeného programu.

První část teoretické části se tedy zabývá popisem softwarových a hardwarových prostředků, které byly použity při tvorbě této práce. Je zde popsán jazyk C/C++ a knihovna wxWidgets. Dále jsou zde popsány standardy, které byly při tvorbě využity a to USB, TCP/IP, SCPI a GPL 2. Popsány jsou rovněž přístroje firmy Agilent, na kterých byl program testován.

Druhá část teoretické části se zabývá možnými programy, které mají stejný účel jako vytvářený program. Jde především o programy společnosti Agilent a to konkrétně Agilent VEE Pro a webové rozhraní přístrojů Agilent. Kromě popisu tato část obsahuje i srovnání jednotlivých programů s vytvářeným programem.

Praktická část popisuje samotný vytvořený program. Rozdělit ji lze opět na dvě části.

V první části je program popsán z hlediska uživatele. Jde v podstatě o manuál k používání programu. Popsány zde jsou programy potřebné pro spuštění programu, ovládání programu a jeho funkce.

V druhé části praktické části je program popsán z hlediska programátora. To znamená, že je zde vysvětlena struktura celého programu, na jakých komponentách je program založen a principy činnosti jednotlivých funkcí.

ZÁVĚR V ANGLIČTINĚ

The teoretical part of this work can be divided into two major part – description of meansn for developing a program that is part of this work and description of alternatives for the same purpose as the created program.

The first part of the teoretical part deals with a description of software and hardware tools that have been used for this work. There is a described C/C++ and wxWidgets library. It also describes the standards which were used to create the program – USB, TCP/IP, SCPI and GPL2. There is also describes Agilent instruments on which the program has been tested.

The second part of the teoretical part deals with the possible programs that serve the same purpose as the created program. I tis mainly about Agilent programs Agilent VEE Pro and Agilent web interface of the instruments. In addition, this section includes a comparison between created programs a porgrams desribed in this section.

The practical part describes developing of the program. It can also be split into two parts.

The first part of the practical part describes program in terms of users. This is essentially a manual for using the program. Here are describes programs needed to run the program, controlling the program and program functions.

In the second part of the practical part program is describes in terms of the programmer. This means that there is explained the structure of the program, components on which is the program based and the principles of individual functions.

SEZNAM POUŽITÉ LITERATURY

- [1] TRBUŠEK, Josef. *Nástroj pro tvorbu dialogových oken pomocí wxWidgets*. Zlín, 2009. 58 s. Bakalářská práce. UTB ve Zlíně.
- [2] *Czech Republic Home / Agilent* [online]. 2000 [cit. 2011-05-01]. Dostupné z WWW: <<http://www.home.agilent.com>>.
- [3] *TRIO Smartcal* [online]. 2010 [cit. 2011-05-01]. Dostupné z WWW: <<http://www.triosmartcal.com.au/>>.
- [4] BURKHARD, Kainka. *USB - měření, řízení a regulace pomocí sběrnice USB*. Česká republika : BEN - Technická literatura, 2002. 247 s. ISBN 80-7300-073-3.
- [5] COMER, Douglas. *Internetworking with TCP/IP - Principles, Protocols and architecture*. Prentice Hall : Prentice Hall, 2006. 688 s.
- [6] GPLv2. *GNU General Public License version 2*. USA : Free Software Foundation, Inc., 1991. 1 s. Dostupné z WWW: <<http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt>>.
- [7] SCPI. *Standard Commands for Programmable Instruments*. USA : SCPI Consortium, 1999. 819 s. Dostupné z WWW: <<http://www.jpacsoft.com/downloads/SCPI-99.pdf>>.
- [8] OLLIVIER, Kevin. *WxWidgets* [online]. 1997 [cit. 2011-05-01]. Dostupné z WWW: <www.wxwidgets.org>.
- [9] UNIS, UTIA, FIT. *VLAM* [online]. 2006 [cit. 2011-05-01]. Dostupné z WWW: <<http://www.vlam.cz/index.html.cz>>.
- [10] BLIŽŇÁK, M, DULÍK, T, VAŠEK, V. A persistent cross-platform class objects container for C++ and wxWidgets. *WSEAS Transactions on Computers* [online]. May 2009, Volume 8, Issue 5, [cit. 2011-05-01]. Dostupný z WWW: . ISSN 1109-2750
- [11] BLIŽŇÁK, M, DULÍK, T, VAŠEK, V. wxShapeFramework: An easy way for diagrams manipulation in CPP applications. *WSEAS Transactions on Computers* [online]. March 2010, Volume 9, Issue 3, [cit. 2010-12-01]. Dostupný z WWW: . ISSN 1109-2750

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ANSI	American National Standard Institute, americká norma
API	Application Programming Interface, programovací rozhraní aplikace
FTP	File Transfer Protocol, protokol přenosu souborů mezi dvěma počítači
GPIB	General Purpose Interface Bus
GTK	GIMP Toolkit
GUI	Graphical User Interface
ISO	International Organization for Standardization
HTML	HyperText Markup Language
LAN	Local Area Network
LXI	LAN eXtensions for Instruments
NRZI	Non Return to Zero Inverted
SCPI	Standard Commands for Programmable Instruments
TCP/IP	Transmission Control Protocol/Internet Protocol
USB	Universal Serial Bus
VISA	Virtual Instrument Software Architecture

SEZNAM OBRÁZKŮ

Obrázek 1: Porty wxWidgets [1].....	14
Obrázek 2: Agilent 33220A[3].....	16
Obrázek 3: Agilent DSO1004A[3].....	17
Obrázek 4: Okno programu Agilent VEE Pro	20
Obrázek 5: Webové rozhraní pro ovládání přístroje.....	21
Obrázek 6: Okno programu po spuštění	26
Obrázek 7: Okno s otevřeným projektem	27
Obrázek 8: Menu File	27
Obrázek 9: Dialog pro zadání názvu nového projektu.....	28
Obrázek 10: Dialog pro načtení projektu.....	29
Obrázek 11: Dialog pro uložení souboru	30
Obrázek 12: Hláška s dotazem na uložení projektu.....	31
Obrázek 13: Menu Edit.....	31
Obrázek 14: Menu Project v režimu editace, kdy je možné přepnutí do režimu ovládání.	32
Obrázek 15: Menu Project v režimu ovládání, kdy je možné se přepnout do režimu editace.....	33
Obrázek 16: Okno informující o programu.	33
Obrázek 17: Okno programu s vloženým prvkem typu Bitmap	34
Obrázek 18: Dialog pro zadání textu prvku Edit Text.....	35
Obrázek 19: Okno programu s vloženým a upraveným prvkem Edit Text	36
Obrázek 20: Okno s vloženým prvkem Check Box.....	37
Obrázek 21: Okno programu s vloženým prvkem Choice s nastavenými volbami.....	38
Obrázek 22: Okno programu s vloženým prvkem Plot.....	39
Obrázek 23: Dialog pro uložení grafu do obrázkového souboru.	40
Obrázek 24: Samostatné okno pro vykreslování grafu	40
Obrázek 25: Okno programu s vloženým prvkem Rectangle s upravenými vlastnostmi a obsahujícího jiné ovládací prvky	41
Obrázek 26: Okno programu s vloženým prvkem Slider	42
Obrázek 27: Okno programu s vloženým prvkem universal	43

Obrázek 28: Okno programu Agilent Connection Expert se zvýrazněným parametrem VISA address.....	45
Obrázek 29: Diagram znázorňující strukturu a závislost tříd projektu.....	48

SEZNAM PŘÍLOH

Příloha P I – Okno aplikace v Microsoft Windows 7 a Ubuntu

Příloha P II – Struktura přiloženého CD

PŘÍLOHA P II: STRUKTURA PŘILOŽENÉHO CD

- /dokumenty

Obsahuje dokumentace přístrojů s příkazy SCPI

- /dokumenty/doxygen

Obsahuje dokumentaci programu pomocí aplikace doxygen

- /práce

Obsahuje text diplomové práce ve formátech DOC a PDF

- /prezentace

Obsahuje prezentaci k diplomové práci

- /programy

Obsahuje programy potřebné pro spuštění vytvořeného programu

- /projekt

Obsahuje vytvořený program

- /projekt/install

Obsahuje soubory pro instalaci programu

- /project/program

Obsahuje spustitelný program

- /project/source

Obsahuje zdrojové soubory programu

- /příklad

Obsahuje vzorový projekt