VŠB - Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2009

Jakub Kvapil

VŠB - Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra měřící a řídící techniky

Absolvování individuální odborné praxe

Individual Professional Practise in the Company

2009 **Jakub Kvapil**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra měřicí a řídicí techniky

Akademický rok 2008/2009

Zadání bakalářské práce

Student: **Jakub Kvapil**

Studijní program: B2645 Elektrotechnika, sdělovací a výpočetní technika

Studijní obor: 2612R041 Řídicí a informační systémy

Téma: Absolvování individuální odborné praxe
Individual Professional Practise in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: **ELCOM, a.s.**
2. Struktura závěrečné zprávy:
   a. Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
   b. Úkoly zadané studentovi v průběhu odborné praxe
   c. Zvolený postup řešení zadaných úkolů
   d. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
   e. Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
   f. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **prof. Ing. Vilém Srovnal, CSc.**

Konzultant bakalářské práce: Ing. Jiří Komínek

Datum zadání: 30.11.2008

Datum odevzdání: 07.05.2009

prof. Ing. Vilém Srovnal, CSc.
vedoucí katedry

prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Prohlašuji, že

- jsem byl seznámen s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména §35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a §60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně ke své vnitřní potřebě bakalářskou/diplomovou práci užít (§35 ods. 3).
- souhlasím s tím, že jeden výtisk bakalářské práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO.
- beru na vědomí, že odevzdáním své 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 předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě dne Podpis studenta

Prohlášení spolupracující firmy

Souhlasím se zveřejněním této bakalářské práce dle požadavku čl. 26, odstavce 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava. Citlivá data uvedená v příloze této bakalářské práce nebudou veřejně dostupná a není povoleno jejich následné využití bez souhlasu majitele autorských práv, tedy firmy ELCOM a.s., Na Větrově 34, 142 00 Praha 4, a firmy ROHDE & SCHWARZ GmbH & Co. KG, Mühldorfstraße 15, 81671 Mnichov, Německo.

V Ostravě dne Podpis zástupce firmy

Poděkování

Rád bych poděkoval vedoucímu práce panu prof. Ing. Vilému Srovnalovi, CSc. za ochotu a pomoc při konzultacích, panu Ing. Jiřímu Komínkovi za odborné vedení a konzultace během mé praxe a panu doc. Ing. Janu Žídkovi, CSc. za umožnění absolvovat odbornou praxi ve firmě ELCOM a.s. Dále bych chtěl poděkovat výbornému kolektivu pracovníků ve firmě ELCOM a.s.

Abstrakt

Tato bakalářská práce popisuje mou studentskou odbornou praxi absolvovanou v jedné z divizí firmy ELCOM a.s. sídlící ve Vědecko-technologickém parku Ostrava. Zařazen jsem byl do softwarového oddělení, kde jsem se zabýval vývojem a testováním programů pro měřicí přístroje. Vytvořil jsem také program pro úpravu xml souborů.

Klíčová slova

Měřicí přístroj, přístrojový ovladač, ukázkový program, vzdálené ovládání přístroje, xml

Abstract

This Bachelor work describes my student professional practice in one division of the ELCOM Company, residing in the Ostrava Science and Technology park. I was working in software department where I was developing and testing programs for programmable instruments. I also developed program for editing xml files.

Keywords

Measurement instrument, instrument driver, example program, remote instrument control, xml

Seznam použitých symbolů a zkratek

GPIB (*General Purpose Interface Bus* - Komunikační sběrnice pro obecné využití) je jednou z nejpoužívanějších komunikačních sběrnic v oblasti měřící techniky

IEEE 488.2 je standard definující vyšší vrstvu komunikace, než je vrstva fyzická

IVI (*Interchangeable Virtual Instrument* - Zaměnitelný virtuální přístroj) je architektura přístrojových ovladačů, přinášející standardizaci a sjednocení ovládacích příkazů pro různé přístroje

LAN (*Local Area Network* – místní síť) je označení pro malé (lokální) počítačové sítě

SCPI (*Standard Comands for Programmable Instruments* - Standardní příkazy pro programovatelné přístroje) definuje jednotné ovládací příkazy programovatelných přístrojů

USB (*Universal Serial Bus* – Univerzální sériová sběrnice) – velmi používaná sběrnice pro propojení různých zařízení

VISA (*Virtual Instrument Software Architecture* - Softwarová architektura virtuálního přístroje) je standard pro nastavování a programování přístrojů

XPath je jazyk používaný pro práci s xml soubory

Obsah

1 Úvod

Tato bakalářská práce popisuje mé působení na odborné praxi studentů ve firmě ELCOM a.s. sídlící v Ostravě ve Vědecko-technologickém parku Vysoké školy Báňské – Technické univerzity Ostrava. Na praxi jsem nastoupil v říjnu 2008 a praxe pokračovala až do května 2009. V následujících kapitolách této bakalářské práce bude představena firma ELCOM a.s., vypracován teoretický rozbor problematiky, se kterou jsem se na odborné praxi setkal, popsány pracovní úkoly, které jsem na praxi dostal, rozebrány mé znalosti, které jsem měl před nástupem na praxi a které jsem během praxe získal, a celkové zhodnocení mého působení na praxi a její přínos.

Kapitola 2 představuje firmu ELCOM a.s., její jednotlivé části a jejich zaměření. Je zde také popsáno mé pracovní zařazení ve firmě. V kapitole 3 jsem zpracoval teoretický rozbor problematiky přístrojových ovladačů a zpracoval způsob komunikace přístroje a ovládacího počítače. Představil jsem také ukázkové ovládací programy, jejichž tvorbou jsem se na praxi zabýval. Kapitola 4 ukazuje konkrétní projekty, které jsem na praxi dělal. Popisuje ukázkové ovládací programy, program pro testování přístrojových ovladačů nebo program sloužící k editaci xml souborů. V kapitole 5 je zhodnocení mých znalostí, které jsem měl před nástupem na praxi, a popis znalostí, které jsem na praxi získal. Kapitola 6 hodnotí mé celkové působení na odborné praxi. V kapitole 7 je uvedena doporučená literatura a odkazy na zdroje, ze kterých jsem čerpal.

2 Popis odborného zaměření firmy a popis pracovního zařazení studenta

2.1 Firma ELCOM, a. s.

Firma ELCOM, a. s. podniká v oblasti silnoproudé elektrotechniky, elektroenergetiky, virtuální instrumentace a je organizačně rozdělena do pěti divizí:

2.1.1 Divize Aplikovaná elektronika

Divize se sídlem v Praze a v Brně je zaměřena na výzkum, vývoj a výrobu speciálních výkonových elektronických zařízení, zejména speciálních napájecích zdrojů pro České dráhy a městskou hromadnou dopravu. Většina zdrojů je však vyráběna jako speciální podle potřeb zákazníka. Podle doposud zrealizovaných zakázek se jednalo o zdroje pro zkušebny a laboratoře, popř. zdroje, které měly přizpůsobit napájení technologie pro jinou napěťovou soustavu (např. pro 60Hz). Součástí byly i stejnosměrné pulzní zdroje určené např. pro galvanizaci s vysokou přesností regulace.

2.1.2 Divize Pohony

Divize se sídlem v Praze je zaměřena na dodávky elektromotorů, hlavně v nevýbušném provedení např. pro petrochemický průmysl. V této oblasti zastupuje na českém trhu německou firmu LOHER. Dále dodává měniče kmitočtů ABB do výkonu 20 MW. V případech havárií velkých motorů nabízí rychlou dodávku náhradních motorů z evropské skladové databáze. Ve spolupráci s ostatními divizemi společnosti je pak možné dodat pohony „na klíč“, tedy včetně projektu, rozvodných zařízení, implementace do řídícího a vizualizačního systému technologie, ověření zpětných vlivů na napájecí síť podle standardů, atd.

2.1.3 Divize Realizace a inženýrink

Divize se sídlem v Brně a Ostravě se věnuje konkrétním dodávkám rozvoden, kompenzačních zařízení nízkého a vysokého napětí. Dále provádí rozbory technických stavů napájecích sítí hlavně z hlediska optimálního provozu a zajištění elektromagnetické kompatibility. Zabývá se hlavně projektovou činností, finálními dodávkami, autorskými a technickými dozory, komplexními a garančními zkouškami v oblasti elektrických průmyslových sítí zejména ve vodárenském odvětví, hutnickém a sklářském průmyslu.

2.1.4 Divize Výroba

Divize se sídlem v technologickém parku v Bystřici nad Pernštejnem slouží jako výrobní závod pro ostatní divize a dále pak jako materiálně-logistická centrála firmy. Výroba je zaměřena především na rozvaděče nízkého a vysokého napětí a na kompenzační rozvaděče. Důležitou roli hraje také výroba ochranných tlumivek do kompenzátorů. Velmi široká spolupráce je s Divizí aplikovaná elektronika a to v oblasti speciálních výkonových elektronických zařízení, zejména univerzálních napájecích zdrojů pro zabezpečovací zařízení pro koridory Českých drah.

2.1.5 Divize Virtuální instrumentace

Tato je nejmladší divizí firmy, která vznikla při Vědecko-technologickém parku (dále jen VTP) Ostrava. VTP je součástí areálu Vysoké školy báňské v Ostravě. Divize je úzce propojena s Katedrou měření a informatiky elektrotechnické fakulty. Zabývá se systémovou integrací, návrhem a dodávkami měřicích a testovacích pracovišť postavených na bázi virtuální instrumentace. Vyrábí speciální analyzátor (projekt BK) na měření energetických rušení v elektrických sítích pro rakouskou firmu DEWETRON. Větší část činnosti je zaměřena na vizualizaci technologických procesů. Jedno oddělení vytváří tzv. ovladače pro nové měřicí přístroje firem Hewlett-Packard, Rohde&Schwarz, Marconi, LeCroy aj. Celá divize úzce spolupracuje s firmou National Instruments se sídlem v USA. [7]

2.2 Popis pracovního zařazení studenta

Jak již bylo řečeno výše, v Ostravě se nachází divize Virtuální instrumentace. Tato divize se ještě dále dělí na dílčí oddělení podle svého zaměření. Finančním ředitelem divize je doc. Ing. Jan Žídek, CSc., který mi nabídl spolupráci s firmou formou absolvování odborné praxe. Svou praxi jsem absolvoval v Softwarovém oddělení, které se zaměřuje na vývoj software pro různé měřicí přístroje mnoha světových výrobců. Oddělení sestává z několika pracovníků pod vedením Ing. Jiřího Komínka, který byl i mým pracovním vedoucím a zároveň konzultantem mé bakalářské práce.

3 Teoretický rozbor úkolů zadaných studentovi v průběhu odborné praxe

Během své praxe jsem se podílel na několika úkolech v rámci oddělení, převážně na vývoji software. Pod pojmem vývoj software se skrývá vývoj ovladačů pro měřicí přístroje a vývoj aplikací pro automatizaci měření na těchto přístrojích. Nezbytnou součástí těchto softwarových produktů je i jejich dokumentace, jejíž tvorba byla také mým úkolem. Dále jsem vytvořil samostatný program pro úpravu xml souborů. Před samotným popisem konkrétních aplikací nejdříve nastíním problematiku přístrojových ovladačů a způsob komunikace s ovládaným přístrojem.

3.1 Přístrojové ovladače a jejich komunikace s přístrojem

Většinu přístrojů, nemusí to být jen přístroje měřicí, ale také například laboratorní zdroje nebo generátory signálů, lze ovládat dálkově. Ovládaný přístroj je propojen s ovládacím počítačem přes určité rozhraní a je tímto počítačem ovládán. Rozhraní propojení se může u jednotlivých přístrojů lišit a je závislé na vybavení konkrétního přístroje. K počítači lze na rozdíl od přístroje připojit téměř libovolné rozhraní při použití odpovídající převodníkové karty. Na počítači je pak spuštěna aplikace, která dává přístroji příkazy, co a jak má přístroj provádět. Tato aplikace využívá funkcí přístrojového ovladače, který v sobě integruje ovládací rutiny jednotlivých funkcí přístroje a posílá příkazy přes komunikační rozhraní do přístroje.

Přístrojový ovladač je souhrn programových funkcí, které ovládají měřicí přístroj. Každá funkce přístrojového ovladače odpovídá určité operaci, jako je například nastavování přístroje nebo čtení dat z přístroje. Přístrojové ovladače usnadňují ovládání přístroje a redukují čas potřebný k vývoji ovládacích aplikací, protože uživatel nemusí znát ovládací protokol pro každý použitý přístroj. Přístrojový ovladač využívá VXIplug&play nebo IVI standardu a VISA knihoven pro komunikaci se zařízením.

IVI (*Interchangeable Virtual Instrument* – Zaměnitelný virtuální přístroj) je architektura přístrojových ovladačů, přinášející standardizaci a sjednocení ovládacích příkazů pro různé přístroje. IVI rozděluje přístroje do různých skupin podle funkce (např. Zdroje ss napětí, Digitální multimetry, Generátory funkcí, Osciloskopy) a každé z těchto skupin přiřazuje soubor ovládacích příkazů - SCPI. SCPI (*Standard Comands for Programmable Instruments* - Standardní příkazy pro programovatelné přístroje) definuje jednotné ovládací příkazy. Tím je zajištěna přenositelnost ovládacích programů a/nebo přístrojových ovladačů mezi přístroji stejného a dokonce i různých výrobců.

VXIplug&play je standard přinášející zjednodušení dálkového ovládání přístrojů. Definuje normu pro přístrojové ovladače, která má myšlenku velmi podobnou s IVI a tou je přenositelnost. Dle VXIplug&play je možné používat jeden přístrojový ovladač pro různé přístroje anebo různé ovládací programy (různých výrobců) pro jeden přístroj.

SCPI jsou slova nebo zkratky tvořené ASCII znaky, které se posílají přes komunikační rozhraní do přístroje. Příkazy jsou rozděleny do skupin podle svého významu (např. konfigurace přístroje nebo měření hodnot) a jejich znění odpovídá příslušné funkci. Pro ukázku si můžeme vybrat spektrální analyzátor, kde se nastavuje měřené frekvenční pásmo svou počáteční a koncovou hodnotou SCPI příkazem:

FREQuency:STARt <hodnota>

FREQuency:STOP <hodnota>

Výsledné příkazy odesílané do přístroje mohou vypadat takto:

FREQ:STAR 3.0e6; FREQ:STOP 6.0e6

Tímto se nastaví počáteční frekvence na hodnotu $3*10^6$ Hz a koncové frekvence na $6*10^6$ Hz. Definice uvádí celé znění příkazu, ale pro komunikaci s přístrojem se používají pouze počáteční znaky psané velkými písmeny, proto konečný příkaz vypadá takto: FREQ:STAR. Příkazy se mohou samozřejmě řetězit, tzn. lze posílat několik příkazů oddělených středníkem najednou.

Dále je možné se přístroje dotázat na hodnoty, které má nastavené. Příkazy jsou obdobné jako pro nastavování jen s rozdílem, že místo číselné hodnoty je otazník:

FREQuency:STARt ?

Po položení dotazu je nutné vyčkat na odpověď a přečíst ji z komunikačního rozhraní.

Příkazy se liší podle funkce a vybavení konkrétního přístroje, ale stejné funkce různých přístrojů musí mít shodné příkazy, aby vyhovovaly SCPI specifikaci. Tím je také splněna myšlenka IVI. SCPI ovšem neřeší problematiku a definici komunikačních rozhraní, pouze definuje ovládací příkazy. Uživatel musí komunikaci programovat sám, což je ale velmi pracné a zbytečně to zpomaluje vývoj ovládacích programů měřících přístrojů, anebo využít již napsané knihovny, které se sami postarají o správnost komunikace. Příkladem může být VISA knihovna, používaná ve většině dnešních aplikací.

VISA (*Virtual Instrument Software Architecture* – Softwarová architektura virtuálního přístroje) je standard pro nastavování a programování přístrojů, který využívá různá komunikační rozhraní, jako je GPIB, LXI, PXI, sériové, síťové (ethernet) a/nebo USB připojení. VISA poskytuje programové rozhraní mezi hardwarem (přístroj, komunikační rozhraní) a vývojovým prostředím, ve kterém je napsán přístrojový ovladač a/nebo ovládací program. Toto řešení přináší mnohé výhody, protože není nutné znát komunikační protokoly jednotlivých rozhraní. VISA využívá SCPI příkazů jako formu komunikace, standard IEEE 488.2 pro správu komunikace a různá komunikační rozhraní jako fyzickou vrstvu komunikace.

IEEE 488.2 standard vznikl jako nadstavba standardu IEEE 488.1, což je poslední revize definice sběrnice GPIB. IEEE 488.2 zajišťuje správnost komunikace, ale nedefinuje fyzickou vrstvu komunikace. Ta byla definována již v IEEE 488.1, ovšem bez jakékoliv správy komunikace. IEEE 488.2 se proto stará o vyšší vrstvu komunikace než je fyzická vrstva, obstarává vzájemné propojení sousedních vrstev a zabezpečuje správné fungování komunikace za všech podmínek. Definuje formát zpráv - příkazů a dotazů posílaných přístroji, datové formáty a zpracování chyb komunikace. I když tato definice je doplněním IEEE 488.1, není nutné je používat dohromady. IEEE 488.1 funguje bez nadstavby IEEE 488.2 a také IEEE 488.2 funguje s jinými komunikačními rozhraními než GPIB. Proto VISA využívá tohoto standardu ve spojení s SCPI příkazy a různými fyzickými komunikačními rozhraními.

Tyto výše uvedené standardy vznikly společným úsilím mnoha předních světových výrobců měřicích přístrojů a společností vyvíjejících software pro měření se smyslem usnadnit, urychlit a zefektivnit vývoj aplikací pro měřicí přístroje. V dnešní době je většina těchto společností součástí IVI Foundation, což je asociace založená pro vytvoření jednotných specifikací v této oblasti. Pro představu o síle asociace uvedu několik jejich členů: Agilent Technologies, National Instruments, Rohde & Schwarz a Aeroflex Corporation. Součástí IVI Foundation se stalo i SCPI Consorcium, čímž se SCPI prosadilo jako celosvětově používaný standard. Dále IVI Foundation spravuje také VISA specifikaci.

3.1.1 Nejrozšířenější komunikační rozhraní používaná k ovládání přístrojů

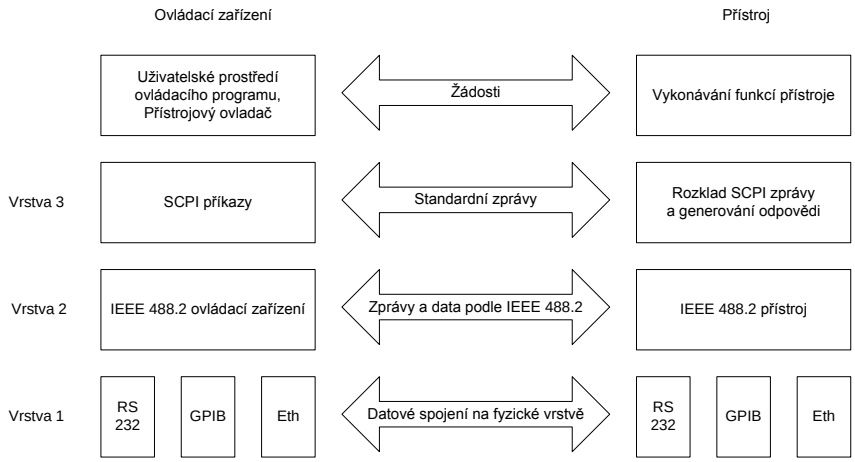
GPIB (*General Purpose Interface Bus* - Komunikační sběrnice pro obecné využití) - je jednou z nejpoužívanějších sběrnic v oblasti měřící techniky. Vyvinuta byla v roce 1972 společností Hewlett-Packard jako propojení několika laboratorních měřících přístrojů. Sběrnice umožňuje propojení až 15 přístrojů na vzdálenost do 25m. Sběrnice se skládá z 24 vodičů, z nichž 8 je datových a ostatní slouží pro řízení komunikace. Přenos informací probíhá paralelně po osmi bitech. Přenos je asynchronní a je řízen tzv. master (z angl. *master* = hlavní, vedoucí) přístrojem. Na sběrnici mohou být dále připojeni tzv. mluvčí – přístroje, které jsou schopné vysílat data, a/nebo tzv. posluchači – přístroje, které data pouze přijímají (např. tiskárny). Přenosová rychlost je nepřímo závislá na délce spojovacího kabelu, její nejvyšší možná hodnota je 8Mb/s. Při použití sběrnicových zesilovačů je možné zvýšit možnou délku propojení a také zvýšit počet možných připojených přístrojů. GPIB se od svého uvedení na trh stala určitým standardem a je hojně využívaná dodnes. Téměř každý přístroj, který obsahuje nějaké komunikační rozhraní, má kromě jiných i GPIB.

Ethernet je technologie počítačových sítí používaných pro lokální počítačové sítě LAN. Ethernet je standardizován normou IEEE 802.3. Standard definuje fyzickou spojovací vrstvu, prostředky síťového přístupu a adresovací formáty. Ethernet umožňuje propojování mnoha různých zařízení (počítačů, měřicích přístrojů, tiskáren, atd.) do společné sítě, aby mohli všichni navzájem komunikovat. Ethernet (LAN) lze propojit s internetovou sítí a tím je umožněna komunikace připojených zařízení na velmi dlouhé vzdálenosti. Měřicí přístroje využívají ethernetu kvůli jeho rozšířenosti, nízké ceně pro vytvoření propojení (není potřeba žádné převodníkové karty, protože téměř každý počítač má ethernetový port) a celkovým možnostem propojení (propojení mnoha přístrojů na neomezenou vzdálenost).

USB (*Universal Serial Bus* – Univerzální sériová sběrnice) je standard sériové komunikační sběrnice, které umožňuje připojit několik zařízení k ovládacímu počítači. USB vzniklo jako náhrada za již zastaralá komunikační rozhraní jako je sériové (RS232) nebo paralelní (LPT) rozhraní. Na rozdíl od nich dovoluje připojování za chodu počítače a také obsahuje napájecí vodiče. To ovšem nejsou důvody, proč se USB rozšířilo do oblasti měřicích přístrojů. Tím je jeho velká rozšířenost, kdy každý dnešní počítač má několik portů USB. Maximální délka USB kabelu je 5m a počet připojených zařízení je 127. Podle normy USB je maximální přenosová rychlost 12Mb/s (USB1.1) nebo 480Mb/s (USB 2.0).

Na následujících obrázcích jsou zobrazeny formy komunikace mezi jednotlivými bloky komunikace ovládacího programu a přístroje. Obrázek 3.1 zobrazuje blokové schéma komunikace začínající u ovládacího programu a přístrojového ovladače, který skrze VISA knihovnu a různá komunikační rozhraní komunikuje s přístroji. Obrázek 3.2 ukazuje jednotlivé vrstvy komunikace ovládacího zařízení (např. PC) s přístrojem a jejich vzájemné propojení. Také je zde popsána funkce a činnost všech vrstev komunikace na obou zařízeních.

Obr. 3.1: Blokové schéma komunikace mezi Ovládacím programem a přístrojem



Obr. 3.2: Schéma jednotlivých komunikačních vrstev a jejich vzájemné komunikace

3.1.2 Přístrojové ovladače v různých programovacích jazycích

Přístrojové ovladače a ovládací programy, které tyto ovladače využívají, jsou psány v programovacích jazycích a programových prostředích k tomu účelu vhodných nebo přímo k tomu účelu vytvořených. Nejvhodnější je použití nižších programovacích jazyků, díky jejich přímé komunikaci s hardwarem. Příkladem mohou být jazyky C a C++, které sice nejsou přímo nižší programovací jazyky, ale mají k nim velmi blízko a jsou velmi používané, tudíž práce s nimi je jednoduchá pro většinu programátorů. Těchto programovacích jazyků využívá prostředí LabWindows/CVI společnosti National Instruments (dále jen NI). Je to jedno z nejpoužívanějších programovacích prostředí pro vývoj přístrojových ovladačů a programů pro automatizaci měření. Další velmi používaným prostředím je LabVIEW, také od společnosti NI. Využívá grafického programování v jazyce G, což je velmi jednoduché a srozumitelné i bez znalosti jakékoliv syntaxe. Společnost NI poskytuje rozsáhlou podporu pro tato programovací prostředí jako například knihovny vstupních/výstupních komunikačních rozhraní nebo ovladače pro mnoho různých přístrojů, což velmi pomáhá při vývoji aplikací komunikujících s měřicím přístrojem.

Dalším vývojovým prostředím, ve kterém se vyvíjejí přístrojové ovladače, je Microsoft (dále jen MS) Visual Studio s programovacími jazyky .NET Frameworku, jako je C# a VB.NET. Toto prostředí, ani tyto programovací jazyky, nejsou primárně určeny k vývoji aplikací pro automatizaci měření. V této oblasti k nim ani není tak obsáhlá podpora jako u produktů NI. Podpora pro tyto jazyky závisí na jednotlivých výrobcích měřicích přístrojů, zda vyvinou příslušné ovladače a zajistí technickou podporu v těchto programovacích jazycích.

3.2 Ukázkové ovládací programy

Pro mnoho výrobců měřicích přístrojů je charakteristický vysoký standard nabízených služeb a uživatelské podpory. Součástí každého přístroje musí nutně být i uživatelská dokumentace ilustrující funkce a způsoby ovládání přístroje. Ta je samozřejmě ve formě textu, ale také jako několik ukázkových programů ukazujících různé funkce a možnosti přístroje.

Ukázkový ovládací program slouží k ovládání určité funkce nebo nastavení hodnot přístroje. Tyto ukázkové programy se mohou distribuovat společně s přístroji a slouží jako vzor pro uživatele k vytvoření vlastních ovládacích programů. Tyto programy většinou ukazují, jak přístroj správně nakonfigurovat pro určité měření nebo jak měření provést a vyčíst data z přístroje. Programy se dodávají ve formě zdrojových kódů ve všech programovacích jazycích, ve kterých existují ovladače pro konkrétní přístroj. Zdrojový kód ukázkového programu má často větší vypovídající hodnotu, je jednodušeji pochopitelný a srozumitelnější než textová forma specifikace komunikace a ovládání přístroje. Ukázkový program by neměl být příliš rozsáhlý, má za úkol ukázat pouze určitou funkci a neodvádět pozornost od daného problému. Dále také ukázkový program neřeší způsob propojení počítače a přístroje, pouze vyvolává funkce přístrojového ovladače, který již má obsluhu komunikace v sobě naprogramovanou.

V praxi jsou ukázkové programy velmi často využívány. Používají je koncoví uživatelé přístrojů jako základ svých vlastních programů, kdy použijí ukázkový program jako kostru programu svého. Tímto způsobem ušetří mnoho času pro vývoj vlastního programu. Nebo těchto ukázkových programů využívají lidé, kteří mají jen základní znalosti programování, a tvorba takového programu by pro ně byla velmi náročná. Ukázkové programy nejsou tedy bezúčelné, ale jsou uživateli velmi žádané.

Jedním z mých úkolů na praxi bylo vytvoření ukázkových programů pro několik přístrojů firmy Rohde&Schwarz, světového výrobce elektronických testovacích a měřicích přístrojů. Převážnou většinu těchto přístrojů tvořily spektrální analyzátory FSQ, FSL, FSU nebo vektorový analyzátor ZVL. Ovladače pro tyto přístroje existují v mnoha programovacích jazycích a vývojových prostředích, jako je jazyk C/C++ v prostředí NI LabWindows/CVI, jazyk G v LabVIEW, nebo jazyky .NET Frameworku (C#, VB.NET) v prostředí MS Visual Studio. Jak již bylo řečeno výše, ukázkové programy se vytvářejí ve všech jazycích, ve kterých existuje příslušný přístrojový ovladač. Toto jsem samozřejmě musel dodržet, a jazyky, které jsem neznal před svým nástupem na praxi, jsem se musel naučit. Součástí každého programu musí nutně být i textový soubor nápovědy, ve kterém je uživateli popsán samotný smysl ukázkového programu, jeho spuštění, nastavení a používání.

4 Řešení zadaných úkolů

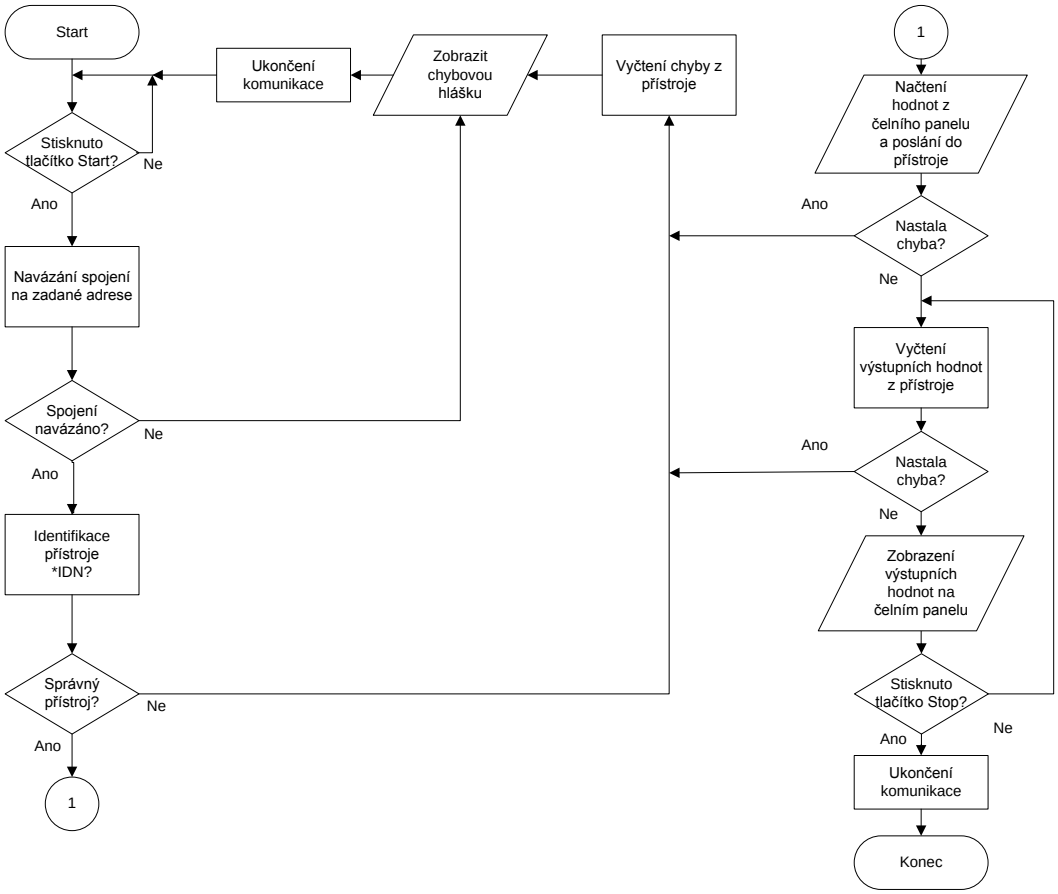
4.1 Seznámení s přístrojovými ovladači a komunikací s přístrojem

Ze začátku bylo nutné seznámit se s problematikou přístrojových ovladačů a způsobem komunikace mezi ovládaným přístrojem a ovládacím zařízením, v mém případě počítačem typu PC. Nejlepší způsob, jak tomuto porozumět, je zhlédnutí již hotových řešení. Dostal jsem tedy přístroj, kartu komunikačního rozhraní, přístrojový ovladač a ukázkový ovládací program. Konkrétním přístrojem byl regulovatelný stabilizovaný zdroj Rohde & Schwarz NGPQ 32/6, s počítačem propojen přes GPIB komunikační rozhraní a ovládán pomocí ukázkového programu využívajícího prostředí NI LabWindows/CVI. Ukázkový program je napsán v jazyce C a má grafické uživatelské prostředí, kde lze nastavovat parametry zdroje jako je výstupní napětí a proudové omezení, různé limitní a alarmní hodnoty nebo povolit zobrazení na displeji přístroje. Dále program vyčítá aktuální výstupní hodnoty a zobrazuje je ve svém okně.

Současně s ukázkovým programem je vhodné spustit program NI Spy (angl. *spy* = špion), který slouží ke sledování probíhající komunikace na komunikačním rozhraní, v tomto případě GPIB. Zobrazují se zde jak příkazy posílané ukázkovým programem (nebo spíše přístrojovým ovladačem), tak odpovědi přístroje.

Zkompiloval a spustil jsem tedy ukázkový program a zkusil nastavit několik parametrů zdroje. V NI Spy jsem sledoval vzájemnou interakci: po stisknutí tlačítka Start se program (přístrojový ovladač) pokusí navázat spojení s přístrojem na uživatelem specifikované adrese. V mém případě byla adresa „GPIB::20::INSTR“, což znamená, že přístroj je na sběrnici GPIB a má adresu 20. Pokud se připojení podaří, program pokračuje dále ve vykonávání svých funkcí, jinak je ohlášena chyba a uživatel vyzván k nápravě. Program tedy pokračuje dále a otázkou „*IDN?“ (SCPI příkaz pro identifikaci) se zeptá na typ přístroje. Přístroj vrací svůj název: „ROHDE&SCHWARZ,NGPQ32/…“, což je přístroj, pro který je ukázkový program určený. Teď se již mohou začít nastavovat hodnoty přístroje. Toto se děje taká pomocí SCPI příkazů a například příkaz pro nastavení rozsahu na 8V/6A vypadá takto: „SOUR:CURR:RANG 6.000000e0“, nebo povolení zobrazování na displeji přístroje: „DISP:ENAB ON“. Po nastavení všech parametrů následuje nekonečná smyčka vyčítání hodnot z přístroje, která se ukončí stiskem tlačítka Stop nebo nějakou chybou v komunikaci nebo v přístroji. Otázkou „MEAS:SCAL:VOLT:DC?;MEAS:SCAL:CURR:DC?“ se program zeptá na aktuální výstupní napětí a proud a čeká na odpověď přístroje. Po přijetí odpovědi program zkontroluje příkazem „*STB?“, zda se nevyskytla nějaká chyba. Pokud vše proběhlo bez chyby, může se pokračovat v nekonečné smyčce vyčítání výstupních hodnot. Vyčítání lze ukončit stisknutím tlačítka Stop. Po uzavření programu se musí zajistit uzavření navázaného komunikačního spojení mezi počítačem a přístrojem.

Na obrázku 4.1 je zobrazen vývojový diagram ukázkového programu. Jeho chování již bylo popsáno, jen nebyl zmíněn systém diagnostiky chyby. Po každém do přístroje odeslaném příkazu se kontroluje stav přístroje přečtením hodnoty Status registru (tzv. Status Byte – STB), zda nenastala nějaká chyba. Toto se provádí dotazem *STB?, po kterém přístroj vrátí hodnotu Status registru. Pokud je hodnota rovna nule, přístroj provedl předchozí operaci bez chyby a je připraven pokračovat. Pokud se ale hodnota od nuly liší, je třeba přečíst hodnotu chybového registru (Error registr) a zobrazit jeho text. Také se ukončí provádění programu a program se vrací na začátek.



Obr. 4.1: Vývojový diagram ukázkového programu

Stejný ukázkový program existuje také v jazyce G v programovacím prostředí LabVIEW. Funkce programu je naprosto stejná a okno programu je velmi podobné tomu v LabWindows/CVI.

Vyzkoušením tohoto programu a prostudováním jeho kódu jsem získal představu o fungování ovládacích programů a mohl jsem pokračovat v práci vlastní tvorbou.

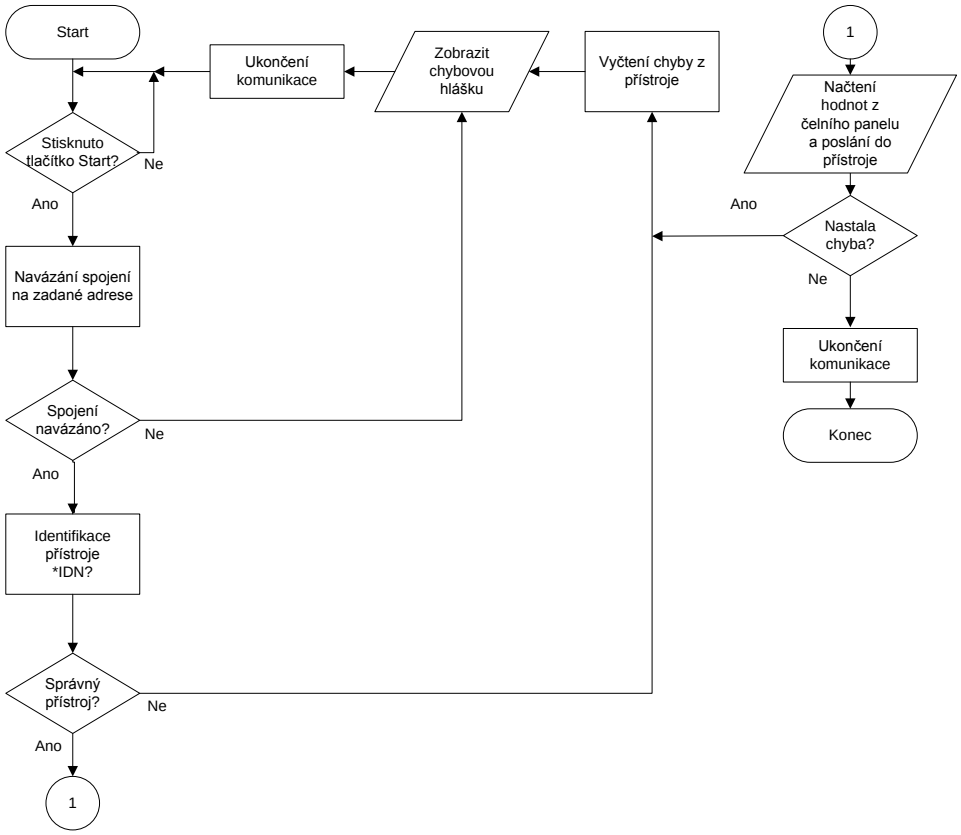
4.2 Tvorba ukázkových ovládacích programů

Převážnou část své odborné praxe jsem strávil tvorbou ukázkových ovládacích programů pro různé přístroje v různých programovacích jazycích a vývojových prostředích. Všechny přístroje byly od německého výrobce měřicích a testovacích přístrojů Rohde&Schwarz. Byly to konkrétně televizní, spektrální a vektorové signálové analyzátory. Zadání ukázkových programů navrhovala firma Rohde&Schwarz podle hlavních funkcí přístroje tak, aby bylo jednoduchou formou ukázáno, jak tyto funkce ovládat a jak je využít. Ukázkové programy byly psány v jazycích C/C++ v LabWindows/CVI, jazyce G v LabVIEW, a jazycích C# a VB.NET ve Visual Studiu. Svou funkcí a hlavně strukturou kódu si byly jednotlivé ukázkové programy velmi podobné a je zbytečné popisovat každý zvlášť. Proto zde popíši hlavní znaky všech podobných programů dohromady.

Ukázkové programy lze rozdělit podle jejich funkce na dvě skupiny: konfigurační, které pouze nastavují různé hodnoty přístroje, a měřicí, které mohou nastavovat hodnoty přístroje, ale hlavně vyčítají z přístroje změřená data.

4.2.1 Konfigurační ukázkové programy

Tato skupina ukázkových programů slouží pouze k nastavování požadovaných hodnot nebo k výběru požadovaného měřicího módu přístroje. Výběrem měřicího módu se myslí přenastavení přístroje pro různá měření, například měření parametrů rádiového, televizního, GSM nebo WiFi signálu. Dále se mohou nastavovat měřicí parametry přístroje, například měřený frekvenční rozsah, počet vzorků měření, nebo systémová nastavení přístroje jako povolení zobrazování na displeji přístroje během činnosti programu nebo nastavení délky prodlevy, kdy přístroj neodpovídá, než nastane reset a vyvolá se chybová hláška.



Obr. 4.2: Vývojový diagram konfiguračního ovládacího programu

Na obrázku 4.2 je vývojový diagram společný pro naprostou většinu konfiguračních ukázkových ovládacích programů, který je shodný s diagramem na obrázku 4.1 s tím rozdílem, že zde chybí vyčítání dat z přístroje. Ale to není účelem tohoto druhu ukázkových programů, na rozdíl od další skupiny.

4.2.2 Měřící ukázkové programy

Tyto ukázkové programy mohou nastavovat přístrojové hodnoty, ale hlavně jsou určeny pro vyčtení výsledků měření z přístroje a jejich následné zobrazení v okně programu. Zobrazují se buď křivky v grafech, nebo jen číselné hodnoty výsledků měření. Zobrazení v grafech se ovšem neprovádí v programovacích jazycích .NET Frameworku, protože by k tomu byly nutné externí grafické knihovny. V těchto situacích se prostě změřená data uloží do souboru.

Vývojový diagram příslušející k těmto ukázkovým programům je na obrázku 4.1. V příloze je uveden konkrétní příklad ukázkového ovládacího programu.

4.2.3 Testování ukázkových ovládacích programů

Všechny ukázkové programy se musí před svým zveřejněním otestovat na přístroji, zda doopravdy plní požadovanou funkci. Testování jsem prováděl na měřicích přístrojích Rohde&Schwarz FSQ, FSL a FSU s připojeným generátorem signálů SMU. Přístroje byly připojeny k počítači přes různá komunikační rozhraní (GPIB, Ethernet, Sériový port). Testování zabere většinou více času než samotný vývoj daného programu. Otestovat se musí všechny možné stavy (různé možnosti nastavení hodnot) programu, aby se zaručila jeho bezchybnost a funkčnost.

Protože není možné mít všechny přístroje fyzicky dostupné na pracovišti, musí se některé programy poslat do firmy Rohde&Schwarz na otestování. Toto řešení není ideální, protože čekání na odezvu může být dlouhé. S tímto jsem se ale setkal pouze jednou u ukázkových programů pro přístroj ETL.

Nabízí se ještě jiný způsob testování programů na přístroji, ale v tomto případě ne na fyzicky dostupném, nýbrž virtuálním přístroji. Mnoho měřicích přístrojů obsahuje standardní počítačový hardware, příslušnou měřicí elektroniku a operační systém MS Windows XP Embedded. V operačním systému je nahrán program - firmware měřicího přístroje a grafické uživatelské rozhraní přístroje, které zobrazuje například měřená data, nastavené hodnoty přístroje nebo popisky příslušných akčních tlačítek umístěných vedle displeje. V podstatě všechna funkčnost přístroje je naprogramovaná a funguje pod operačním systémem. Pouze měření zajišťuje měřicí hardware, který odpovídá typu daného přístroje. Pro otestování funkčnosti ukázkového programu ale nepotřebujeme měřicí hardware a proto stačí využít softwarového základu přístroje. Tento software je nainstalován na počítači typu PC propojeném přes komunikační rozhraní ethernet s ovládacím počítačem. Ovládaný počítač se teď tváří jako přístroj a lze na něm otestovat funkčnost ukázkového programu. Anebo lze software nainstalovat přímo do počítače, kde je spuštěn testovaný program, a komunikovat přes localhost spojení. Tímto způsobem jsem testoval ukázkové programy pro přístroj ZVL. Testování probíhá naprosto stejně jako s fyzickým přístrojem i bez přítomnosti měřící elektroniky. V simulaci lze totiž použít virtuálně vytvořený testovací signál.

4.2.4 Součásti ukázkových programů

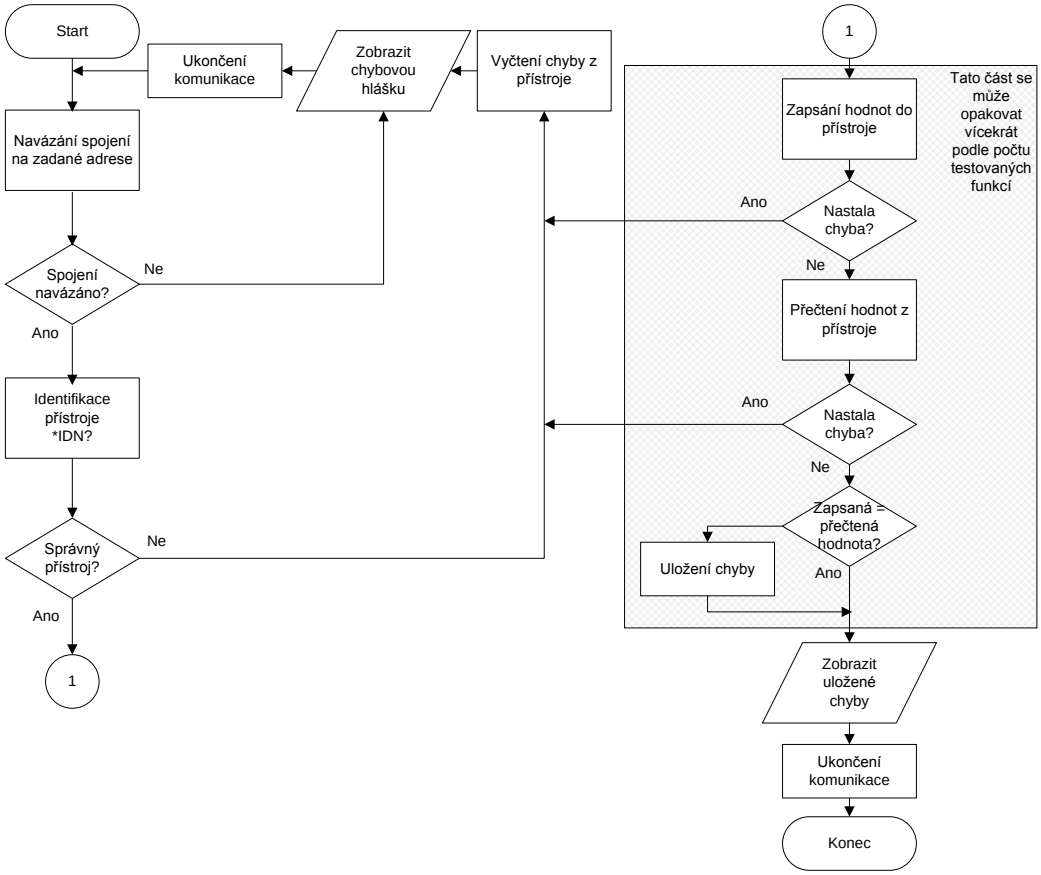
Nedílnou součástí každého programu, potažmo každého profesionálního softwarového produktu, by měla být i uživatelská dokumentace. V ní jsou obsaženy informace o programu, jak ho nainstalovat, nastavit a používat. Je v ní také popsáno grafické uživatelské prostředí programu a funkce jednotlivých tlačítek. Dále je uveden odkaz nebo kontakt na uživatelskou podporu pro případ vyskytnuvších se problémů. Nápověda obsahuje i seznam všech souborů distribuovaných v balíku ukázkového programu. A protože každý ukázkový program je popsán v anglickém jazyce, tak také soubory nápovědy jsou psány anglicky.

4.3 Tvorba programů pro testování přístrojových ovladačů

Přístrojové ovladače obsahují několik desítek nebo i stovek funkcí nastavujících parametry příslušného přístroje. Každá taková funkce zapíše hodnoty do přístroje, a pokud se nevyvolá chyba, zdá se být všechno v pořádku. Tato myšlenka je správná, ale až s použitím odzkoušeného a stabilního přístrojového ovladače. Při vývoji ovladače se ale může stát, že se hodnota nezapíše vůbec nebo se zapíše jinam. Proto se ovladače musí proti tomuto otestovat. Kontrolovat všechny funkce přístrojového ovladače ručně jednu po druhé by ale bylo velmi zdlouhavé, proto se vytvářejí testovací programy, které toto zautomatizují. Testovací program obsahuje posloupnost mnoha funkcí, které zapisují hodnoty do přístroje. Po vyvolání každé funkce zapisující data do přístroje, se musí zkontrolovat správnost zapsaných dat. Vyčtou se data z umístění, kam se měly zapsat, a porovnají se

s daty, které měly být zapsány. Pokud se shodují, je vše v pořádku. V opačném případě se chyba zaznamená a po vyzkoušení všech funkcí, tedy až program dojde na konec, se zobrazí výčet všech chyb.

Tyto testovací programy se ale také používají pro testování samotného přístroje. Pokud výrobce provede změnu nebo úpravu firmwaru přístroje, použije testovací program pro otestování firmwaru. Velká výhoda testovacích programů spočívá v možnosti opakování testů, kdy se jeden test může spustit mnohokrát za sebou bez jakékoliv vynaložené námahy. Toto u manuálně prováděných testů nelze. Testovací programy se také používají při předávání hotových přístrojových ovladačů firmě, která si je objednala. Při předání se ovladač nechá otestovat testovacím programem, čímž se dokáže jeho bezchybnost a funkčnost.



Obr. 4.3: Vývojový diagram programu pro testování přístrojových ovladačů

Vývojový diagram testovacího programu uvádím na obrázku 4.3. Začátek programu je shodný s ukázkovými programy, jejichž vývojové diagramy jsou uvedeny výše. Nejprve se musí navázat spojení s přístrojem na zadané adrese a pak se zjišťuje, zda je přítomný správný přístroj. Ověřování správnosti přístroje zde nutné není, protože tento testovací program používá pouze vývojář přístrojových ovladačů, ne uživatel, a ten je schopný si pohlídat správnost přístroje sám. Ale splést se může každý, hlavně v tom množství přístrojů a přístrojových ovladačů, a proto je lepší tuto část kódu nevynechávat. Následuje samotná testovací sekvence, kdy se zapíší hodnoty na požadované umístění v přístroji a ihned se přečtou zpátky do počítače. Hodnoty zapsané a vyčtené se porovnají, a pokud nejsou stejné, uloží se chyba do paměti. Tato testovací sekvence zápisu, čtení a porovnání se opakuje tolikrát, kolik je testovaných funkcí v testovacím programu. Po vyzkoušení všech funkcí se přečtou

z paměti uložené chyby a zobrazí se. Tím program dokončil svou funkci, může ukončit komunikaci s přístrojem a zavřít se.

Tyto programy slouží pouze pro testovací účely vývojářům softwaru, kteří tento druh programů používají velmi často, a proto k nim není nutné vytvářet obsáhlou nápovědu.

4.4 Vytvoření programu pro úpravu XML souborů

Přístrojové ovladače se skládají z jednotlivých funkcí. Ve vývojovém prostředí LabWindows/CVI má každá tato funkce svůj čelní panel (front panel). Každý čelní panel a také všechny jeho prvky mají textovou nápovědu. Tuto textovou nápovědu ze všech čelních panelů přístrojového ovladače lze uložit do jednoho xml souboru, čímž je umožněna její jednoduchá editace. Pro úpravu textů nápovědy není tedy nutné otevírat každý čelní panel, ale stačí editovat text v xml souboru. Upravovat xml soubory lze v každém textovém editoru. Ovšem ruční úprava souborů nápovědy není možná, kvůli rozsáhlosti nápovědy. Běžný xml soubor nápovědy přístrojového ovladače obsahuje několik tisíc různých elementů s texty a úprava těchto textů by byla bez použití nějakého programového editoru nad lidské síly. Většina dostupných editorů xml souborů je velmi komplexní a pro tyto účely tedy nevhodná.

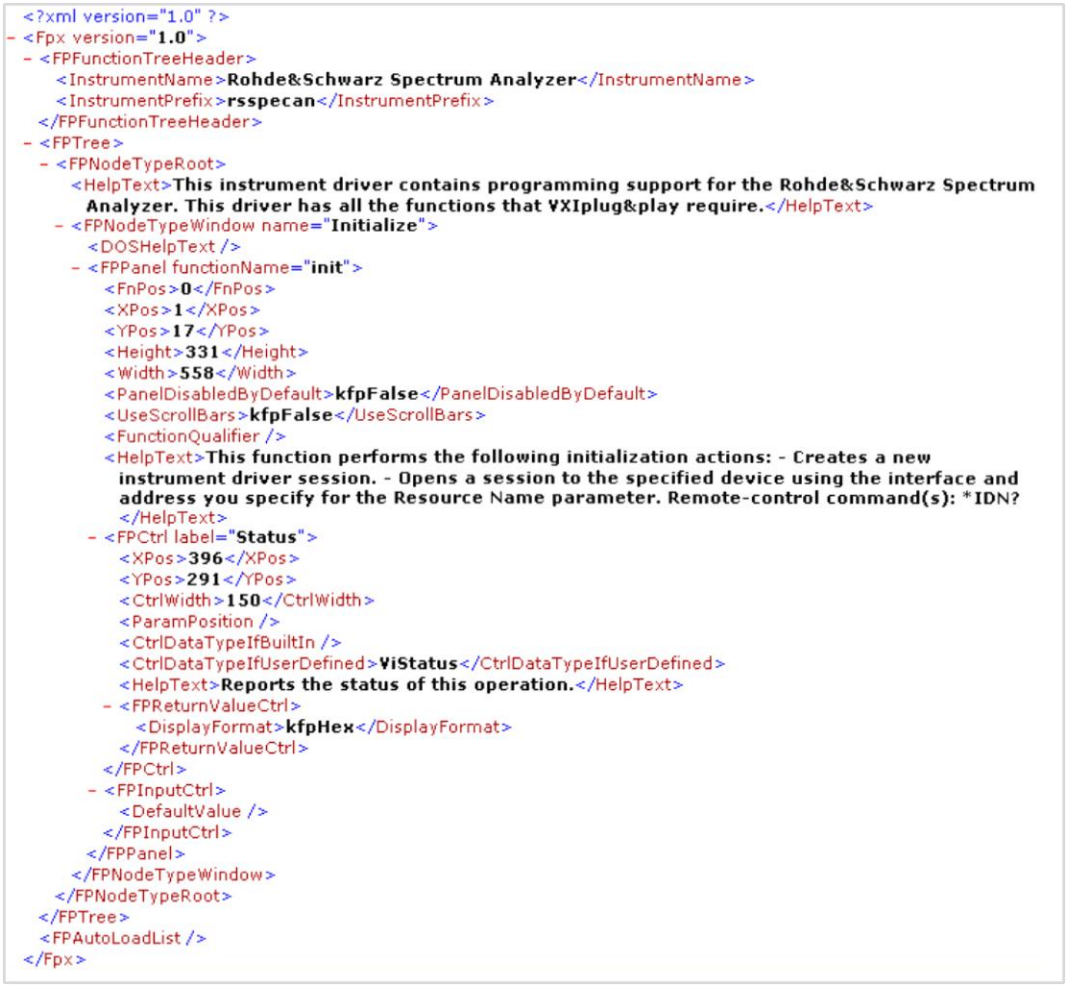
Mým úkolem bylo tedy vytvoření programu, který by umožňoval jednoduše editovat texty nápovědy uložených v xml souboru. Program musí umět vyhledat jednotlivé uzly ve struktuře xml souboru, vyhledat text v popisku konkrétního uzlu a popřípadě tento text nahradit jiným. Také program musí umožňovat načtení textu z textového souboru a jeho následné uložení do požadovaného uzlu xml souboru. Dalším požadavkem bylo, aby program byl spustitelný pod systémem Windows v grafickém okně, ale také bez okna v příkazové řádce. Dalším neméně důležitým bodem zadání bylo vytvoření nápovědy k programu, a to jak ve formátu textového souboru (.txt), tak i ve formátu souboru nápovědy (.chm).

Pro realizaci tohoto zadání jsem využil programovací jazyk C# ve vývojovém prostředí MS Visual Studio. Zde se dá jednoduše vytvořit celková funkčnost programu ve spojení s grafickým uživatelským prostředím nebo s možností ovládání přes příkazovou řádku. V samotném programu jsem použil pro zpracování xml souboru metodu XPath dotazu, protože nejvíce vyhovovala hlavnímu požadavku zadání programu. Tím je myšlena jednoduchost výběru libovolného uzlu ve struktuře xml souboru.

4.4.1 XPath dotaz

XPath je jazyk používaný pro práci s xml soubory. Jeho největší výhodou a předností je jednoduchost vyhledávání jednotlivých elementů, uzlů a/nebo atributů v libovolné struktuře xml souboru. XPath dotaz je vlastně způsob zápisu cesty k hledanému výrazu v xml struktuře. Cestu lze zapisovat mnoha způsoby, například pro vybrání jednoho konkrétního elementu zapíšeme cestu absolutně od elementu s nejvyšší důležitostí až po námi hledaný element, nebo můžeme zadat pouze název elementu a XPath dotaz vrátí všechny nalezené elementy tohoto názvu.

Pro ukázku uvedu příklad xml souboru (obr. 4.4), který zobrazuje data xml souboru, který byl vytvořen z přístrojového ovladače v LabWindows/CVI. Struktura xml souboru má kořenový element *Fpx*, který obsahuje mnoho vnořených elementů. Nás bude zajímat element *HelpText* vnořený v elementu *FPCtrl* s atributem *label="Status"*.



Obr. 4.4: Struktura ukázkového xml souboru

Podle definice syntaxe XPath dotazu [8] zadáme dotaz:

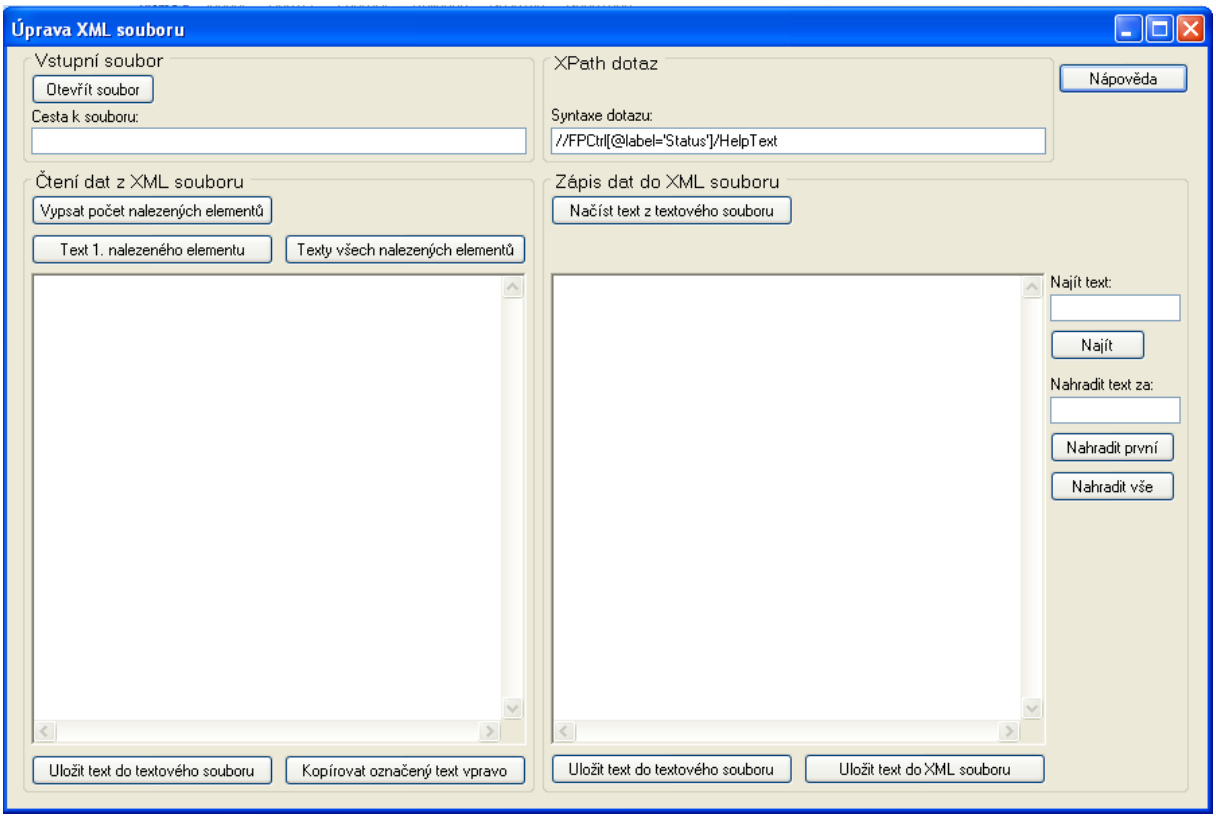
//FPCtrl[@label='Status']/HelpText

Tento dotaz vyhledá všechny elementy *HelpText*, které jsou vnořeny v elementu *FPCtrl* s atributem *label="Status"*. Vrátí tedy text: *Reports the status of this operation*

Tento příklad ukazuje jednoduchost a rychlost vyhledávání v xml souborech s pomocí XPath dotazu.

4.4.2 Popis grafického okna programu

Okno programu (obr. 4.5) je rozděleno do několika částí: volba vstupního souboru, zadání XPath dotazu, čtení z xml souboru a zápis do xml souboru.



Obr. 4.5: Grafické uživatelské rozhraní programu

V oddílu Vstupní soubor vybere uživatel xml soubor určený k editaci. Dále uživatel zadá XPath dotaz do příslušného textového pole. V tomto okamžiku je na výběr několik možných činností: program může vypsat počet nalezených elementů specifikovaných XPath dotazem, vypsat text prvního nalezeného elementu nebo vypsat texty všech nalezených elementů. Anebo se vyčítat ze souboru vůbec nemusí a uživatel může přímo zadat text, který se zapíše do elementu (nebo elementů) specifikovaným XPath dotazem. V editačních textových oknech lze text libovolně psát, upravovat, kopírovat, načítat z nebo ukládat do textového souboru. Je zde také možnost najít text a popřípadě nalezený text i nahradit za jiný. Stiskem tlačítka Nápověda se v novém okně otevře nápověda k tomuto programu.

4.4.3 Popis programu spuštěného v příkazové řádce

Program lze také spustit bez grafického uživatelského rozhraní v příkazové řádce. Pokud se program spustí bez parametrů, otevře se okno grafického rozhraní. Pro spuštění bez okna tedy musí uživatel zadat alespoň jeden parametr při spouštění programu. Jsou dostupné 4 parametry:

- **„-help“** – zobrazí se nápověda
- **„-filePath:“** - za dvojtečku se napíše cesta k XML souboru. Z tohoto souboru se bude načítat, ale bude se do něj i zapisovat
- **„-XPath:**“ – za dvojtečku se napíše syntaxe XPath dotazu
- **„-textFilePath:“** – za dvojtečku se napíše cesta k textovému souboru, ve kterém je uložen text pro zápis do XML souboru

Pro správné fungování programu je potřeba zadat poslední tři jmenované parametry. Pokud některý zadaný není nebo je zadaný špatně, program vyzve uživatele k nápravě zadání.

Program funguje tak, že otevře xml souboru specifikovaný parametrem „-filePath:“ a vyhledá v něm elementy podle zadaného XPath dotazu. Do těchto elementů vloží text ze zadaného textového souboru a výsledný xml soubor uloží – přepíše původní xml soubor.

Pro usnadnění spouštění programu v příkazové řádce jsem vytvořil dávkový soubor, který program spustí se zadanými parametry. Parametry se zapíší do dávkového souboru a nemusí se pořád vypisovat do příkazové řádky. Tím se vše podstatně urychlí, ulehčí a zautomatizuje.

4.4.4 Nápověda k programu

K programu bylo nutno vytvořit také nápovědu a to v několika formách: stručný popis instalace a ovládání programu v textovém formátu (.txt) a soubor nápovědy (.chm), který popíše celý program, jeho fungování a ovládání. Tentokrát je veškerá nápověda napsána v českém jazyce, protože program pro editaci xml souborů je vytvořen pouze pro potřeby zaměstnanců firmy ELCOM a.s.

Textová nápověda obsahuje pouze nejdůležitější body pro instalaci a ovládání programu, proto se zde o ní nebudu rozepisovat.

Zajímavější je nápověda ve standardním formátu pro soubory nápovědy - .chm soubor. Nápověda se vytváří v programu Microsoft HTML Help Workshop. Zde se vytváří jednotlivé stránky nápovědy jako samostatné html stránky a pomocí zde vytvořené stromové struktury se stránky sjednotí do jednoho souboru nápovědy. Použitím html se usnadní vytváření jednotlivých stránek, zajistí se přiměřená grafická úprava a vzhled stránek se sjednotí. Také lze používat kaskádové styly pro zjednodušení úprav html stránek. Po vytvoření všech stránek a nastavení všech potřebných hodnot v programu se soubory zkompilují do jednoho souboru .chm. Tento soubor lze jednoduše otevřít, protože operační systémy Windows obsahují prohlížeč souborů nápovědy.

5 Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe a znalosti či dovednosti scházející studentovi v průběhu odborné praxe

Odbornou praxi jsem absolvoval v softwarovém oddělení firmy, kde jsem se zabýval vývojem programů pro automatizaci měření. Uplatnil jsem znalosti programování, programovacích jazyků a algoritmizace získaných během studia svého oboru na vysoké škole. Užitečné byly také znalosti elektrických měření a měřicích přístrojů.

Ve škole jsem se naučil programování v jazycích C a C#. Tyto znalosti posloužily jako základ pro mou praxi, i když to byly znalosti pouze základní a pro samostatnou tvorbu profesionálních programů byly nedostatečné. Posloužily mi jako základ pro další studium těchto programovacích jazyků.

Jazyk C a také i C++ jsem se učil prohlížením a rozborem již hotových programů. Syntaxi jazyka jsem znal z výuky ve škole, ale práci s objekty nebo zpracovávání událostí jsem viděl poprvé. Tyto jazyky jsem využil při programování ukázkových ovládacích programů ve vývojovém prostředí LabWindows/CVI, které jsem také vůbec neznal. Toto prostředí jsem nastudoval z doporučené literatury [3]. Tato kniha, jako naprostá většina dalších, byla v anglickém jazyce.

Využil jsem tedy i znalosti anglického jazyka. Většina materiálů, které jsem studoval, byly v angličtině. Materiály myslím knihy, nápovědy k jednotlivým programům a vývojovým prostředím, různé texty k dané problematice ať už ve formě odborných článků zveřejněných na internetu, nebo ve formě specifikací standardů, které definovaly například komunikační rozhraní. Mé znalosti anglického jazyka jsem dosáhl již před nástupem na vysokou školu, kdy jsem byl schopen přečíst a porozumět odbornému textu bez větších problémů. Výuka cizích jazyků na vysoké škole se nezaobírá technickým slovníkem, ale snaží se naučit cizojazyčně nejméně zdatné jedince základnímu porozumění danému jazyku. Absolvování 4 semestrů cizího jazyka jsem bral spíše jako procvičení cizího jazyka a také procvičení mozku učením krkolomných písemkových chytáků. Odborné výrazy, a dá říct i cizí jazyk všeobecně, jsem se naučil sám studiem cizojazyčné literatury. Ale jak jsem již zmínil, před nástupem na vysokou školu jsem měl velmi dobrou znalost anglického jazyka, díky mému předcházejícímu učiteli.

Vrátím se zpět k programovacím jazykům. Dalším jazykem, který jsem znal ze studia na vysoké škole, byl jazyk C#. Tento jazyk jsme měli pouze v jednom předmětu, tedy jen jeden semestr, ale během toho jsem se naučil ty nejpodstatnější věci. Na tyto základy se dalo lehce navazovat a učit se novým věcem bez větších problémů. Tento programovací jazyk jsem využíval v prostředí MS Visual Studia pro programování ukázkových ovládacích programů a také pro vytvoření programu pro úpravu xml souborů. Pro studování problematiky tohoto programovacího jazyku jsem využil literaturu [1]. Z této knihy jsem také čerpal informace ohledně způsobu zpracování xml souborů, které jsem využil při realizaci programu upravujícího xml soubory. Jazyk C# jsem se začal učit teprve před necelým rokem, ale velmi jsem si ho oblíbil kvůli jeho jednoduché syntaxi a způsobu práce s jednotlivými objekty, třídami atd.

Druhým programovacím jazykem používaným ve Visual Studiu byl VisualBasic (VB.NET). VisualBasic jsem také viděl poprvé, ani s jeho staršími verzemi (Basic) jsem se nikdy nesetkal. Naštěstí programy v něm vytvořené se velmi podobají programům vytvořeným v C#, jediný rozdíl je

jen v syntaxi jazyka VB.NET. Podklady pro nastudování tohoto programovacího jazyku jsem čerpal v nápovědě k Visual Studiu. Ale studoval jsem to jen málo, protože tento jazyk je velmi jednoduchý k naučení při znalosti jiného programovacího jazyku, který mu je blízký (např. C#). Tímto programovacím jazykem jsem tvořil ukázkové ovládací programy.

Dalším jazykem, který jsem na odborné praxi používal, byl jazyk G ve vývojovém prostředí LabVIEW. Je to grafický programovací jazyk, založený na principu dataflow (tok dat). Nejvíce se využívá pro sběr dat, ovládání přístrojů a průmyslovou automatizaci. Běh programu je udáván strukturou jednotlivých grafických prvků a jejich vzájemným propojením. Každý grafický prvek plní – vykonává určitou funkci, ať už to je funkce elementární (např. sčítání čísel, práce s textovým řetězcem) nebo funkce složitější, která může být a většinou také je složena z několika jiných funkcí. Každý takovýto program obsahuje čelní panel – grafické uživatelské rozhraní, kde uživatel zadává vstupní hodnoty a kde se zobrazují výstupy. LabVIEW jsem při své odborné praxi využil k tvorbě ukázkových ovládacích programů. Současně s mým nástupem na praxi jsem začal ve škole navštěvovat lekce Virtuální instrumentace I, které se zabývaly výukou tohoto jazyka. Současně s výukou ve škole jsem ale musel studovat doporučenou literaturu [4] [5] [6], abych se jazyk naučil rychleji a více do hloubky. Začátky s tímto programovacím jazykem nebyly příliš obtížné díky skvělé výuce ve škole a celkové jednoduchosti programování v tomto jazyce. Práce s tímto jazykem mě baví, je jednoduchá a zábavná.

Znalosti programovacích jazyků jsem již popsal a teď bych se zaměřil na mé znalosti problematiky přístrojových ovladačů. V předmětu Elektrická měření vyučovaném v prvním ročníku jsem se s přístrojovými ovladači a komunikací s přístrojem setkal. Předmět ovšem nebyl zaměřen na tuto problematiku, ta byla pouze nastíněna. A navíc jsem naprostou většinu znalostí přístrojových ovladačů získaných v tomto předmětu zapomněl, protože jsem je až do nástupu na svou odbornou praxi nevyužil. Začínal jsem, dá se říci, od začátku. Studoval jsem tedy doporučenou literaturu [2] a předložené hotové programy. Začínal jsem rozborem hotového ukázkového ovládacího programu (viz. kapitola 4.1), který komunikoval s přístrojem, čímž jsem si mohl prohlédnout a vyzkoušet způsob a formu komunikace mezi ovládacím počítačem a ovládaným přístrojem. Po pochopení tohoto programu jsem již vytvořil podle předloženého zadání svůj první ukázkový program. Tvorbou dalších ukázkových programů jsem se dostával hlouběji do dané problematiky a začínal jsem rozumět novým věcem. Vytvořil jsem také program pro testování přístrojových ovladačů, čímž se mé znalosti přístrojových ovladačů znásobily. Toto byla hlavní náplň mé odborné praxe a dá se říci, že jsem se propracoval z prakticky nulových znalostí této problematiky až ke schopnosti samostatně vytvořit program ovládající měřicí přístroj, potažmo i jednoduchý přístrojový ovladač, v různých programovacích jazycích.

Dále jsem vytvořil program pro úpravu xml souborů. Program jsem tvořil ve vývojovém prostředí MS Visual Studio. Je psaný programovacím jazykem C#. S tvorbou grafického uživatelského rozhraní jsem problémy neměl, protože to jsme dostatečně procvičili ve škole při výuce programování v tomto jazyce. Ovšem na cvičeních jsme se nezabývali xml soubory a jejich programovým zpracováním, to jsem musel nastudovat sám v literatuře [1]. Součástí tohoto projektu bylo také vytvoření souboru nápovědy. Se soubory nápovědy (.chm) jsem se setkal jako jejich uživatel, ale nikdy jsem je sám nevytvářel. Tyto soubory se tvoří v programu MS HTML Help Workshop jako seskupení html stránek. Html jazyk jsem znal již před studiem na vysoké škole, proto jsem s tímto neměl větší problémy. Pokud bych html jazyk již neznal, musel bych se ho naučit, protože při výuce na vysoké škole jsem se s ním nesetkal.

Tato kapitola je zaměřena na mé znalosti získané během studia na vysoké škole a proto jsem ještě nezmiňoval pomoc mého konzultanta během odborné praxe. Zmínit se ovšem musím, protože pan Komínek byl nápomocný při každé mé nevědomosti. Problematiku, kterou jsem neznal, mi podrobně vysvětlil, ukázal různé příklady použití nebo uvedl zdroj informací pro mé další studium. Většinu doporučené literatury uvedené v této bakalářské práci jsem dostal zapůjčenou z firemní knihovny domů k prostudování. To mi také velmi pomohlo, protože takové materiály bych velmi těžce někde jinde získával.

6 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Během své odborné praxe ve firmě ELCOM a.s. jsem se naučil mnoho nových a užitečných věcí. Získal jsem náhled do reálné firmy, seznámil se s jejím fungováním, způsobem zpracování zakázek a projektů. Naučil jsem se vytvářet konečný produkt se všemi náležitostmi. Získal jsem nové zkušenosti jak v programování různými programovacími jazyky tak také v problematice přístrojových ovladačů a měřicích přístrojů.

Na odborné praxi jsem se nejvíce zabýval tvorbou ukázkových ovládacích programů. K vlastní tvorbě těchto programů jsem dospěl přes porozumění problematice komunikace ovládacího počítače a ovládaného přístroje. Studoval jsem hotový ukázkový program, přístrojový ovladač a způsob komunikace s přístrojem. Postupně jsem tomu začínal rozumět a mohl jsem tvořit své programy. Ty jsem tvořil v mnoha programovacích jazycích, z nichž jsem znal pouze dva. Zbývající tři jsem se musel doučit. Ke všem těmto ukázkovým programům jsem vypracoval textovou nápovědu, která byla samozřejmě v angličtině. Procvičil jsem si tedy i cizí jazyk.

Dále jsem vytvořil testovací program. Byl psán v jazyce C/C++ v LabWindows/CVI. Tento program slouží jak pro testování přístrojových ovladačů, tak také pro testování samotných přístrojů.

Dostal jsem také zadání vytvořit program pro úpravu xml souborů nápovědy. Jako prostředek programování jsem si vybral jazyk C# v prostředí Visual Studia. Program jsem úspěšně vytvořil i se souborem uživatelské nápovědy ve formátu .chm.

Během své odborné praxe ve firmě ELCOM a.s. jsem se naučil mnoha novým věcem v oblasti programování a hlavně přístrojové techniky a komunikace s přístroji. Ve firmě pracuje mnoho odborníků, z nichž jedním je i pan ing. Jiří Komínek, můj konzultant na odborné praxi. Díky němu jsem se naučil mnoha věcem, pomohl mi porozumět dané problematice a rozkryl způsoby řešení jednotlivých úkolů. Jsem rád, že jsem mohl být součástí tohoto kolektivu profesionálů, a určitě využiji zkušenosti zde získané při svém dalším studiu i v profesním životě.

7 Doporučená literatura

[1] **NAGEL, Christian.** *C#2005 Programujeme profesionálně*. [překlad] Jakub Mikulaštík a Petr Dokoupil. 1. vydání. Computer press, 2006. ISBN 80-251-1181-4.

[2] **PIEPER, John M.** *Automatic Measurement Control – A tutorial on SCPI and IEEE 488.2*. Rohde&Schwarz.

Firemní literatura:

[3] *LabWindows/CVI – Getting started with LabWindows/CVI*. National Instruments, říjen 2007

[4] *LabVIEW – Getting Started with LabVIEW*. National Instruments, srpen 2006

[5] *LabVIEW Basics I – Development Course Manual*. National Instruments, říjen 2005

[6] *LabVIEW Basics II – Development Course Manual*. National Instruments, říjen 2005

Elektronické zdroje:

[7] URL: <http://www.elcom.cz/o-spolecnosti-elcom/> [cit. 2009-05-01]

[8] URL: <http://www.w3.org/TR/xpath20/> [cit. 2009-05-01]