

Univerzita Pardubice

Dopravní fakulta Jana Pernera

Aplikace pro dálkové ovládání přístrojů Rohde & Schwarz

Matěj Petkov

Diplomová práce

2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Matěj Petkov**
Osobní číslo: **D11760**
Studijní program: **N3708 Dopravní inženýrství a spoje**
Studijní obor: **Dopravní infrastruktura: Elektrotechnická zařízení v dopravě**
Název tématu: **Aplikace pro dálkové ovládání přístrojů Rohde&Schwarz**
Zadávající katedra: **Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě**

Z á s a d y p r o v y p r a c o v á n í :

Diplomová práce bude zaměřena na vytvoření aplikace pro dálkové ovládání přístrojového vybavení od firmy Rohde&Schwarz vypsanych níže. Ovládání přístrojů bude umožňovat obousměrnou komunikaci: čtení, zápis měřených či simulovaných dat a nastavení vybraných parametrů jednotlivých přístrojů a to s využitím USB i TCPIP protokolu (s případnou analýzou dalších možností GPIB, VISA, UDP a jiné). Aplikace bude kompatibilní s osciloskopy (RTO), obvodovými analyzátory (FSH, ZVL), spektrálními analyzátory (FSL,FSC), měřičem výkonu FSH-Z1, generátory signálu řady SM a přístrojem pro radiový monitoring PR100. Součástí DP bude dokumentace pro jednotlivé přístroje obsahující detailní nastavení potřebné pro ovládání přístrojů.

Dále bude DP obsahovat analýzu přenosových rychlostí pomocí jednotlivých druhů ovládání.

Seznam přístrojů:

- osciloskopy
- obvodové analyzátory
- spektrální analyzátory
- měřiče výkonu
- radiový monitoring
- generátory signálu

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:

[1] **PIEPER, John M. Automatic Measurement Control: A tutorial on SCPI and IEEE 488.2. München: Rohde & Schwarz GmbH & Co. KG, 2007. ISBN 3939837024.**

[2] Rohde & Schwarz [online]. 2013 [cit. 2013-02-27]. Dostupné z: <http://www.rohde-schwarz.com>

[3] National Instruments [online]. 2013 [cit. 2013-02-27]. Dostupné z: <http://www.ni.com>

Vedoucí diplomové práce:

Ing. Jan Pidanič

Katedra elektrotechniky

Datum zadání diplomové práce:

1. března 2013

Termín odevzdání diplomové práce:

23. května 2013



prof. Ing. Bohumil Culek, CSc.
děkan

L.S.



doc. Ing. Radovan Doleček, Ph.D.
vedoucí katedry

V Pardubicích dne 6. března 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 14. 5. 2013

Matěj Petkov

Poděkování

Rád bych zde poděkoval panu Ing. Janu Pidaničovi, Ph.D. za jeho ochotu a vstřícnost, a panu Ing. Tomáši Shejbalovi za pomoc a zpětnou vazbu při vývoji aplikace.

Anotace

Cílem této diplomové práce je vytvoření grafické aplikace (GUI), která umožňuje vzdálené řízení přístrojového vybavení od firmy Rohde & Schwarz. Za tímto účelem jsou podrobně rozebrány specifické vlastnosti a podmínky tohoto druhu komunikace. Na základě těchto poznatků je navržen takový způsob sestavení fungujícího ATE systému, aby jeho realizace byla jednoduchá a rychlá, a aby byly minimalizovány eventuální komplikace související s jeho zprovozněním.

Součástí práce je popis jednotlivých ovládaných přístrojů včetně přesných návodů k nastavení parametrů potřebných pro jejich dálkové řízení.

Klíčová slova

ATE systém, vzdálené řízení, SCPI, knihovna VISA, Matlab.

Title

Remote Control Software for Rohde & Schwarz Instruments

Annotation

The aim of this master thesis is to create a Graphical User Interface (GUI) which provides remote control of Rohde & Schwarz Instruments. For this purpose, characteristics of this kind of communication are discussed in detail. Based on this knowledge, a method of ATE system's assembly is designed with a focus on simple and quick implementation and elimination of potential complications.

Description of controlled instruments as well as instructions for settings of remote control parameters are included in this paper.

Keywords

ATE system, remote control, SCPI, VISA library, Matlab.

Obsah

ÚVOD.....	9
1 PLATNÉ STANDARDY A JEJICH VÝVOJ	10
1.1 AUTOMATIZOVANÉ SYSTÉMY MĚŘENÍ.....	10
1.2 IEEE 488.....	11
1.3 IEEE 488.2.....	12
1.4 ROZŠÍŘENÍ 488.2 – PŘÍKAZY SCPI.....	13
2 PŘÍKAZY SCPI.....	15
2.1 ÚVOD DO SCPI.....	15
2.2 STRUKTURA PŘÍKAZŮ A SYNTAXE	16
2.2.1 <i>Hlavička</i>	17
2.2.2 <i>Parametry</i>	18
2.3 TYPY ZPRÁV A JEJICH TVAR	20
2.3.1 <i>Command vs. Query</i>	20
2.3.2 <i>Zkracování příkazů</i>	20
2.3.3 <i>Číselné značení v SCPI zprávách</i>	22
2.4 PARAMETRY SCPI PŘÍKAZŮ.....	23
2.4.1 <i>Datový typ znak</i>	23
2.4.2 <i>Datový typ řetězec</i>	24
2.4.3 <i>Dekadický číselný datový typ</i>	24
2.4.4 <i>Ostatní číselné datové typy</i>	26
2.4.5 <i>Numerické hodnoty pomocí klíčových slov</i>	28
2.4.6 <i>Booleovský datový typ</i>	31
2.4.7 <i>Libovolný datový blok</i>	32
2.5 HIERARCHICKÝ STROM SCPI PŘÍKAZŮ	33
2.5.1 <i>Hierarchie uzlů</i>	33
2.5.2 <i>Skládání hlavičky příkazu</i>	34
2.6 PODSYSTÉMY SCPI PŘÍKAZŮ	36
2.6.1 <i>Model SCPI přístroje</i>	36
2.6.2 <i>Základní rozdělení SCPI přístrojů</i>	37
2.6.3 <i>Popis podsystémů</i>	38
3 KNIHOVNA VISA	41
3.1 ISO/OSI MODEL SÍŤOVÉ KOMUNIKACE	41
3.2 VISA A PROTOKOLY VXI-11	42
4 PODMÍNKY PRO VÝVOJ A BĚH APLIKACE.....	43
4.1 REALIZACE SÍŤOVÉHO SPOJENÍ	43
4.2 VÝVOJOVÉ PROSTŘEDÍ MATLAB.....	44
4.2.1 <i>Instrument Control Toolbox</i>	44
4.3 KOMUNIKACE POMOCÍ VISA.....	45
4.4 PŘÍSTROJOVÉ VYBAVENÍ.....	47
4.4.1 <i>Spektrální analyzátor R&S® FSL</i>	48
4.4.2 <i>Obvodový analyzátor R&S® ZVL</i>	49
4.4.3 <i>Spektrální analyzátor R&S® FSH</i>	50
4.4.4 <i>Signálové generátory R&S® SMx</i>	52
4.4.5 <i>Osciloskop R&S® RTO</i>	53
5 POPIS APLIKACE.....	55
5.1 SPOLEČNÉ PRVKY V GUI.....	56
5.1.1 <i>Tlačítka s grafikou</i>	56

5.1.2	<i>Možnosti exportu</i>	57
5.1.3	<i>Manual Data Cursor</i>	58
5.1.4	<i>Zavření GUI vs. ukončení aplikace</i>	58
5.2	NOVÉ PŘIPOJENÍ - MAINGUI	59
5.3	FSL-GUI	61
5.4	ZVL-GUI	62
5.5	FSH-GUI	64
5.5.1	<i>Spectrum Analyzer GUI</i>	65
5.5.2	<i>Network Analyzer GUI</i>	65
5.5.3	<i>Power Meter GUI</i>	66
5.6	SMx100A-GUI	67
5.6.1	<i>Modulation Generator</i>	68
5.6.2	<i>Modulation</i>	69
5.6.3	<i>RF</i>	70
5.7	RTO1014-GUI	71
	ZÁVĚR	73
	ZDROJE	75
	SEZNAM ZKRATEK	76
	SEZNAM OBRÁZKŮ	77
	SEZNAM TABULEK	79

Úvod

Měřicí zařízení jsou nedílnou součástí vědeckotechnických laboratoří, a ty, které náleží Katedře elektrotechniky na Fakultě elektrotechniky a informatiky, nejsou pochopitelně výjimkou. Jejich přístrojové vybavení čítá moderní zařízení disponující množstvím pokročilých funkcí. Při jejich pravidelném používání však vyvstává řada nedostatků, které může uživatel shledat jako omezující: ať už je to malý displej na předním panelu, složité hledání a nastavování parametrů přístroje, nebo zdlouhavé přenášení naměřených dat pomocí přenosného paměťového zařízení (zpravidla USB disku či SD karty). Potřeba automatizace úkonů a většího komfortu práce je tak tím větší, čím „schopnější“ daný přístroj je.

Uspokojení tohoto požadavku by měla přinést aplikace vyvinutá v rámci této diplomové práce – diskomfort při manipulaci s přístrojem bude odstraněn přenesením ovládacího rozhraní na obrazovku počítače, kde je obsluha programu přirozeně možná pomocí klávesnice a myši. Neopomenutelnou funkcí by měl být jednoduchý transport dat získaných měřením signálu tak, aby bylo okamžitě možné jejich další zpracování v systému Matlab.

V práci bude nejprve stručně popsán původ standardů, díky kterým je dnes možné provádět měření pomocí vzdáleného řízení přístrojů. Jejich vývoj zahrnuje i vznik speciálního programovacího jazyka, jehož potřeba se v kontextu historického vývoje jevila jako nevyhnutelná. Tento jazyk přinesl řád a systém do rychle se rozvíjející oblasti, jejíž neuspořádanost začala brzdit další rozvoj. Příkazům SCPI bude proto věnován dostatek pozornosti a prostoru, neboť se jedná o netradiční a nepříliš obvyklý způsob programování, jehož znalost je přitom klíčová pro vytvoření řídicí aplikace.

Pro popis principu přenosu informace mezi řídicím počítačem a měřicím přístrojem, který v ATE systému (automatizovaném systému měření) probíhá, bude využit vrstvý model síťové komunikace. Dále budou v práci uvedeny všechny podmínky nutné pro vývoj a provoz aplikace – spolu s popisem ovládaných přístrojů a detailními návody k nastavení parametrů potřebných pro úspěšné ustavení spojení bude tato část po úpravách použita také jako manuál a nápověda pro uživatele naprogramovaného řídicího softwaru.

V poslední části textu bude popsán praktický výsledek diplomové práce – grafická aplikace pro dálkové ovládání přístrojů Rohde & Schwarz, při jejímž vývoji bude využita rozsáhlá teoretická příprava. Cílem není pouze tvorba GUI, ale také na základě nabytých znalostí návrh nejvhodnějšího postupu sestavení ATE systému tak, aby splňoval požadavky na rychlost přenosu dat, a zároveň na jednoduchost jeho konstrukce.

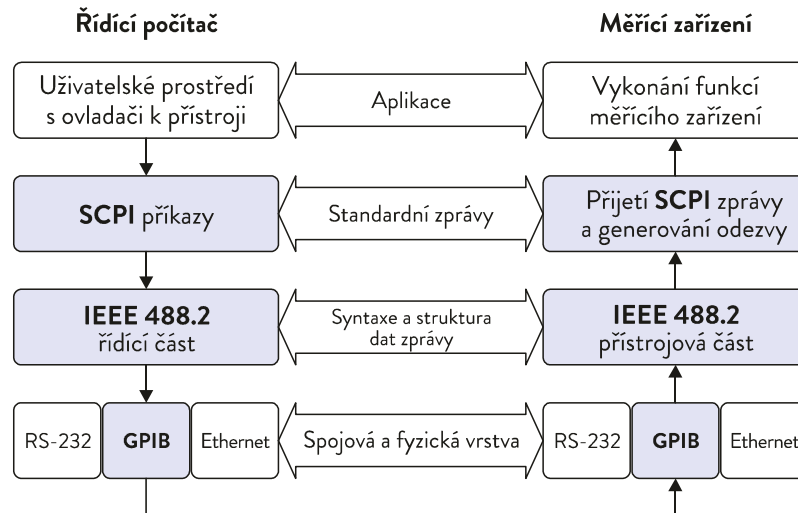
1 Platné standardy a jejich vývoj

1.1 Automatizované systémy měření

Měření, která byla dříve prováděna výhradně ručně a jejichž výsledky bylo nutno vyčítat přímo z přístrojů, prošla během druhé poloviny 20. století významným vývojem, který reagoval především na probíhající automatizaci technologií. Na začátku 70. let se na trhu objevilo mnoho zařízení umožňujících dálkové ovládání. Tyto přístroje (angl. *Instruments*) jsou kabelem spojeny s řídicím počítačem (nazývaným *Controller*), přičemž samo spojení je realizováno přes sběrnici neboli *Interface bus*. Takto je možné vzájemně propojit různé přístroje přes jedno rozhraní a spolu s řídicím počítačem tak vytvořit přístrojovou soustavu – *Instrumentation System*.

Jelikož je taková soustava sestavena za účelem testování a měření, je často nazývána *Test & Measurement System*, nebo také *ATE (Automatic Test Equipment) system*. Jednotky, které mohou být v těchto systémech připojeny, představují širokou škálu zařízení: od velkých, stolních přístrojů až po ty, které jsou instalovány na desce plošných spojů. Role jednotlivých členů ATE systému je zřejmá: přes interface bus jsou řídicím počítačem vysílány zprávy, které v sobě obsahují příkazy pro vzdálenou správu – *remote commands*. Těmi controller řídí činnost měřicího zařízení, jež na případný dotaz počítače (typicky vyčítání naměřených hodnot, ale také např. načítání nastavení či identifikace přístroje) odpoví stejnou cestou přes sběrnici.

Předtím, než budou chronologicky rozebrány standardy související s ATE systémy, je vhodné pro lepší představu o postavení jednotlivých norem uvést jejich pozici ve vrstevném modelu. To je systém používaný v teorii sítí pro znázornění procesu komunikace mezi dvěma uzly jedné sítě. Definiuje nezávislé „vrstvy“, přičemž každá z nich vykonává část funkcí, které jsou nutné pro uskutečnění spojení; jednotlivé vrstvy na sebe přitom logicky a především funkčně navazují. Na obr. 1.1 je zobrazen vrstevný model popisující komunikaci v přístrojové soustavě, a v něm jsou zvýrazněny vrstvy a standardy, které jsou předmětem dalších kapitol.



Obr. 1.1 Přehled standardů pro ATE a jejich role ve vrstevném modelu [1]

1.2 IEEE 488

Nedlouho poté, co se na trhu objevili první přístroje umožňující dálkové řízení, se velikost nabídky techniky tohoto typu začala rapidně zvyšovat, přičemž se samozřejmě jednalo o produkty různých výrobců. Všechna tato zařízení měla své vlastní způsoby realizace nejen samotného připojení hardwaru, ale také jeho softwarové ovládání – příkazy pro vzdálenou správu. Se stále rostoucí různorodostí vyvstala pochopitelně potřeba vytvořit standardy a normy, které by sjednotily podobu vzdáleného řízení. V roce 1973 se proto pod vedením *International Electrotechnical Commission (IEC)* sešlo několik nejvýznamnějších výrobců měřicí techniky a navrhli světový standard pro řízení přístrojů – později (v roce 1975) přijatý jako IEEE 488, často označovaný jako *General Purpose Interface Bus (GPIB)*. Tento standard se v průběhu následujících let rozšířil do naprosté většiny vyráběné měřicí a zkušební techniky a v podstatě všechny běžně užívané přístroje sběrnici GPIB disponují. Standard IEEE 488 definuje rozhraní z elektrického i mechanického hlediska, navíc poskytuje protokoly pro síťovou komunikaci mezi vysílačem (*talker*) a přijímačem dat (*listener*). Jedná se o protokoly se spojením, které zajišťují spolehlivé doručení dat a jejich synchronizaci pomocí *handshake*¹. Ve standardu je také stanoveno hierarchické rozdělení přístrojové soustavy, tzn. především nadřazenost řídicího počítače.

¹ Handshaking je proces, kterým je u síťových protokolů zajištěn spolehlivý přenos dat. Princip spočívá ve vzájemné komunikaci mezi odesílatelem a adresátem, která předchází samotnému odeslání užitečné informace, a během které dochází k vzájemnému potvrzení odesílání, resp. očekávání dat.

Je nutné poznamenat, že funkce spojové (linkové) a fyzické vrstvy síťového modelu, které zprvu zastávalo GPIB, byly postupem času přebírány jinými rozhraními, a to nejen těmi existujícími, jako je sériová linka RS-232, ale rovněž nově vyvinutými standardy. Tak je možné dnes využít například rozšíření sběrnice VMEbus pro automatizované měření *VMEbus Extensions for Instruments (VXI)*, technologii umožňující realizovat lokální síť – *Ethernet (IEEE 802.3)*, sériovou sběrnici *FireWire (IEEE 1394)* vyvinutou společností Apple, nebo její v dnešní době převládající alternativu *Universal Serial Bus (USB)*.

1.3 IEEE 488.2

Zavedení standardu GPIB bylo jistě významným krokem kupředu v technologii řízení měřicích přístrojů vzdálenými příkazy, brzy ovšem vyšla najevo potřeba další standardizace. Specifikace IEEE 488 se totiž věnovala mechanické a elektrické části přístrojové soustavy, přičemž popisovala pouze základní komunikační protokoly; nedefinovala však formát dat a přesný tvar příkazů, kterými by zařízení měla být obsluhována. Nutnost zavést jasná pravidla syntaxe a struktury dat svedla opět dohromady nejvýznamnější výrobce, jejichž aktivita vyústila v roce 1987 ve vznik nadstavby stávajícího standardu označované jako *IEEE 488.2 – Standard Codes, Formats, Protocols, and Common Commands*. Tato norma již definovala funkce vyšších vrstev síťového modelu, popisovala jejich roli ve využití služeb nižších vrstev (fyzické a spojové) a zároveň se věnovala rozhraní mezi těmito úrovněmi. V době vývoje IEEE 488.2 bylo uvažováno o výhradním využití GPIB, a proto je ve standardu explicitně popsáno pouze propojení s tímto typem sběrnice; nicméně hlavní část funkcionality nezávisí na typu použitého protokolu nižších vrstev a může dobře spolupracovat i s dalšími (výše uvedenými) protokoly.

Protokoly obsluhující jednotlivé vrstvy síťového modelu jsou typicky přítomné jak v řídicím počítači, tak v obsluhovaném přístroji. Funkce těchto protokolů, IEEE 488.2 nevyjímaje, jsou ovšem na obou jednotkách odlišné a velká většina těchto funkcí se vztahuje k měřicímu zařízení.

Jak vyplývá z názvu standardu, důležitou součástí normy jsou kódy a formáty dat, to znamená definice syntaxe zpráv odesílaných přes interface bus. IEEE 488.2 se však problematikou zabývá hlouběji; popisuje také principiálně koncept komunikace mezi řídicím počítačem a přístrojem. Za tímto účelem jsou zavedeny dvě rozdílné formy zpráv: příkaz – *command* a dotaz – *query*, které se liší svojí syntaxí. Dále je kladen velký důraz na řádné chování přístroje při narušení protokolu aplikací, která jej využívá – jde především o situace,

kteře mohou nastat při výměně zpráv na trase controller-instrument. Automatická náprava těchto chybových stavů je důležitá zejména proto, aby přístrojová soustava mohla spolehlivě a samostatně pracovat i bez lidského zásahu.

Další důležitou oblastí, kterou norma IEEE 488.2 pokrývá, je *Status Reporting* – rozlišení všech druhů možných stavů a událostí, které se mohou vyskytnout při činnosti měřicího zařízení. Tato vlastnost využívá funkcí GPIB protokolů, konkrétně *Service Request (SRQ)* a *Serial Poll* („sériový dotaz“). Jedná se o jediné prostředky, díky kterým má zařízení možnost převzít iniciativu a vyžádat si pozornost od řídicího počítače. Právě pro tuto schopnost je celý systém hlášení stavů a chyb napojen na tento mechanismus. Zvláštní předností standardu IEEE 488.2 je zavedení jasné a flexibilní struktury, která umožňuje přístroji přehledná hlášení všech událostí obslužné aplikaci. Tím také poskytuje uživateli příležitost vyžádat si jakékoli informace o stavu připojeného zařízení. Často opomíjeným faktorem, hrajícím však významnou roli při obdržení spolehlivých výsledků měření, je také odpovídající synchronizace činností mezi řídicím počítačem a měřicím zařízením. Technikami, které to umí zajistit, se zabývá samostatná část IEEE 488.2.

Součástí IEEE 488.2 jsou také definice tzv. *Common Commands*, příkazů, které řídí často užívané funkce přístroje. Jedná se o malé množství typických provozních příkazů, které tvoří nepatrnou množinu ze všech možných funkcí měřicího zařízení. Standard ovšem nedefinuje samotný programovací jazyk, ten přišel teprve se standardy vyšší vrstvy.

1.4 Rozšíření 488.2 – příkazy SCPI

Posledním krokem standardizace, který zajistil spolehlivou a především kompatibilní komunikaci s přístrojem bez ohledu na výrobce, bylo zavedení *Standard Commands for Programmable Instrumentation (SCPI)* – hierarchických příkazů, kterými lze univerzálně komunikovat s měřicími přístroji. V této části jsou uvedeny především kvůli svému významu v kontinuitě vývoje; detailnímu popisu těchto příkazů je věnována samostatná kapitola 2.

Vznik SCPI příkazů reprezentujících jednotný programovací jazyk měřicích přístrojů byl přirozeným vyústěním patnácti let vývoje v oboru vzdáleného řízení. Norma IEEE 488 z roku 1975 popsala používaný hardware, norma IEEE 488.2 z roku 1987 se věnovala komunikaci v přístrojové soustavě a posledním problémem tedy zůstala neexistence jasných pravidel pro příkazy, kterými se ovládala měřicí zařízení. Konkrétní tvar těchto příkazů se lišil nejen mezi výrobci, ale dokonce mezi jednotlivými modely jednoho přístroje. Odstranění této

komplikace bylo jen otázkou času a ještě na konci 80. let přišla se svým řešením firma Hewlett-Packard.

Celá myšlenka začala vytvářením seznamu mnemotechnických zkratk (*mnemonics*) – krátkých slov odvozených z příkazů používaných přístroji od HP. Vybrány byly takové zkratky, jejichž funkce byly snadno odvoditelné z významu původního nezkráceného slova, a které byly zároveň jedinečné a nezaměnitelné. Seznam těchto příkazů začal postupně získávat tvar; skupiny zpráv typických pro daný druh přístroje se vyvinuly až do určité podoby programovacího jazyka platného pro jistou oblast měřicích zařízení – osciloskopy, signálové generátory, voltmetry apod. V této fázi byl již rozeznán potenciál takového přístupu a byly učiněny kroky mající za cíl stanovit jednotný přístup ve vývoji obecně platného jazyka pro všechny měřicí přístroje. Tento koncept byl zpočátku nazýván TMSL (*Test and Measurement System Language*) a byl celosvětově přijat hlavními výrobci. Speciálně pro tuto příležitost ustavené konsorcium publikovalo v roce 1990 první vydání nového průmyslového standardu pojmenovaného SCPI (*Standard Commands for Programmable Instrumentation*). Od té doby je vývoj tohoto standardu kontinuální – každý rok vycházejí nové verze, a do zavedených pravidel tak bylo přidáno mnoho nových příkazů a dodatečných funkcí.

Hlavním cílem SCPI příkazů je zjednodušit programování ATE aplikací, čehož dosahují v první řadě svou univerzálností, jelikož SCPI je možné používat bez ohledu na typ přístroje a jeho výrobce, přičemž odpadá nutnost složitě zjišťovat seznam specifických příkazů. Z této vlastnosti plyne další výhoda, a to přenositelnost kódu – celé části programů se dají uplatnit univerzálně bez nutnosti úprav při každé změně měřicího zařízení.

Na síťovém modelu je znázorněna úroveň, na které pracují SCPI příkazy. Jedná se o úroveň nad protokolem IEEE 488.2, na jehož základech je programovací jazyk SCPI postaven, a z něhož přejímá základní principy a terminologii. Kvůli této úzké vazbě platí, že každé zařízení, které využívá SCPI příkazy, musí rovněž vyhovovat standardu IEEE 488.2. Dá se tedy říct, že přístroj kompatibilní s SCPI je také kompatibilní s IEEE 488.2. Nicméně přestože byly původně SCPI vyvinuty tak, aby na fyzické vrstvě pracovaly se sběrnici GPIB, je možné (a dnes už běžnější) spojit řídicí počítač a měřicí zařízení pomocí jiného rozhraní, např. VXI, RS232 nebo Ethernet, preferovaný při vývoji aplikace pro tuto diplomovou práci.

2 Příkazy SCPI

2.1 Úvod do SCPI

Zkratka SCPI je odvozena z anglického *Standard Commands for Programmable Instruments*, což lze do češtiny přeložit jako Standardizované příkazy pro programovatelné přístroje. Tyto příkazy jsou ve formě zprávy posílány řídicím počítačem do měřicích zařízení a umožňují jejich dálkové řízení. Jedním z důvodů vývoje těchto příkazů byla situace popsána již v kapitole 1.4, kdy se seznam platných řídicích zpráv lišil nejen mezi výrobci, ale také mezi přístroji jedné značky, a programování pro vzdálenou obsluhu se tím značně komplikovalo. Snaha o odstranění těchto překážek vedla ke vzniku nového programovacího jazyka, jehož hlavními vlastnostmi jsou univerzálnost a přenositelnost. Díky tomu je možné použít shodný tvar příkazu v kódu programu pro libovolné zařízení a jeho funkce bude vždy totožná.

Sjednocen byl také formát dat používaný SCPI zprávami odesílanými z přístroje do řídicího počítače jako odpověď na dotaz – *query*. Aby bylo možno efektivně dosáhnout takové uniformity, byl v roce 1990 přijat (jako součást normy IEEE 488.2) standard SCPI. Ten definuje seznam použitelných zpráv, odpovídající reakce přístroje, dále popisuje přesnou syntaxi a význam každého SCPI příkazu. Z pochopitelných důvodů se funkce měřicí techniky neomezují pouze aktuálně platným seznamem zpráv, ale naopak standard samotný je pravidelně každoročně aktualizován přidáním příkazů popisujících nové funkce, které se objevují jako důsledek dynamického rozvoje měřicích přístrojů.

Přestože jsou příkazy SCPI považovány za programovací jazyk a ve vrstevném modelu patří do nejvyšší, aplikační vrstvy, je třeba je odlišit od počítačových jazyků (C++, Java, Pascal atd.). Hlavní odlišnost je na první pohled patrná a spočívá v rozdílném adresátovi příkazu – na rozdíl od klasických programovacích jazyků není v případě SCPI cílem zprávy počítač, ale měřicí zařízení. Z toho plyne neschopnost počítačů rozumět SCPI příkazům, kteráž skutečnost ovšem nebrání řídicímu PC s těmito příkazy pracovat. Samotný controller je totiž řízen pomocí vlastního programovacího jazyka, kterému rozumí, zatímco SCPI zprávy, jež pro něj nemají význam, jsou mu prostě předány jako textové řetězce ASCII znaků. Program (resp. řídicí počítač jeho prostřednictvím) zajistí odeslání zprávy a je na jejím příjemci – měřicím přístroji, aby její obsah pochopil a příkaz vykonal.

Velkým kladem tohoto konceptu je vzájemná nezávislost programovacích jazyků, díky které je možné psát program pro ovládání PC v libovolném, programátorem zvoleném jazyce, a pokud je dodržena správná syntaxe SCPI příkazů v textových řetězcích, zařízení bude vždy odpovídajícím způsobem reagovat. Z téže vlastnosti však plyne také zásadní nevýhoda – kvůli neschopnosti počítače rozumět SCPI ve formě řetězce chybí programátorovi zpětná vazba, a to ať už se jedná o kontrolu syntaxe, či například o nastavení vhodných parametrů předávaných přístrojům. Správnost příkazů je tedy čistě v kompetenci autora programu, což může vést ke zdoluhavému hledání skryté drobné chyby v situaci, kdy řídící PC zahlásí v pořádku odeslanou zprávu a měřicí zařízení přitom nevykoná zadaný úkol.

2.2 *Struktura příkazů a syntaxe*

Základní rozdělení zpráv, kterými řídící počítač komunikuje s přístrojem, rozlišuje dva typy, a to v souvislosti se směrem, kterým je zpráva vysílána. Pokud je příkaz adresován měřicímu zařízení, je jeho zdrojem počítač, jenž se řídí kódem napsaným programátorem. Takovým zprávám se říká programové zprávy (*program messages*), a jelikož jsou generovány aplikací napsanou člověkem, je jejich struktura pro programátora dobře čitelná. V případě, že je v reakci na programovou zprávu od přístroje vyžadována odpověď (*response message*), je jejím zdrojem měřicí zařízení a příjemcem řídící počítač, čili jedná se o strojovou komunikaci. Data, která jsou přijata jako reakce na příkaz vyslaný počítačem, musí proto daleko přísněji splňovat pravidla daná standardem a umožnit tak jejich další výpočetní zpracování. Aby byla pro lidského uživatele alespoň do jisté míry zjednodušena manipulace s oběma druhy zpráv, je jako jejich formát zvolen řetězec znaků ASCII.

Následující odstavce se budou zabývat formální podobou SCPI zpráv a je proto vhodné pro názornost uvést ukázkou typického příkazu. V tomto případě se jedná o nastavení středu zobrazované frekvence na hodnotu 3 GHz; příkaz použitelný např. u spektrálních analyzátorů:

```
SENSe:FREQuency:CENTer 3e9
```

Z příkladu je patrné, že v syntaxi SCPI příkazů lze rozlišit dva základní prvky: hlavičku příkazu a parametry, přičemž parametrů se v závislosti na typu příkazu a požadované reakci přístroje může vyskytovat hned několik; často je ale také možné parametry ve zprávě úplně vynechat. Z formálního hlediska se tedy ve struktuře SCPI odlišuje oddělovač hlavičky (*header separator*, zpravidla mezera, je ale možné použít i jiné znaky – viz dále) a oddělovač

parametrů (*data separator*, představovaný čárkou). V jedné zprávě je navíc možné odeslat více kompletních SCPI příkazů, tj. každý se svou hlavičkou a se svými parametry. Je však nutné je oddělit třetím typem oddělovače nazývaným *message unit separator* (oddělovač zpráv), který se ve zprávě značí středníkem. Takto by pak vypadala zpráva obsahující dva příkazy – již uvedené nastavení středu zobrazované frekvence je třeba doplnit ještě zobrazovaným rozsahem (700 MHz):

```
SENSe:FREQuency:CENTer 3e9; :SENSe:FREQuency:SPAN 700e6
```

Pro úplnost je třeba uvést možnost spojení dvou SCPI příkazů typu query v jedné zprávě, po jejímž odeslání přístroj vrátí odpovědi na oba dotazy v jednom řetězci oddělené rovněž pomocí *message unit separator*. Při programování je ovšem takový postup značně nepraktický, neboť s návratovou hodnotou zpravidla programátor potřebuje dále pracovat, a proto je vhodnější uložit každou odpověď do samostatné proměnné; tzn. odeslat dotazy jako dvě nezávislé SCPI zprávy.

2.2.1 Hlavička

Hlavička příkazu je tvořena textovým řetězcem vzniklým složením řadou mnemotechnických zkratk oddělených dvojtečkou (významu klíčových slov a hierarchické struktury stromu SCPI příkazů se detailně věnují další kapitoly). Jedná se v podstatě o zkratky nebo fráze zvolené tak, aby samy o sobě vyjadřovaly smysl klíčového slova, přičemž jejich řazení ve zprávě respektuje hierarchii příkazů – od nejobecnějších kategorií až po konkrétní povely. Oddělovač hlavičky a parametrů se vkládá mezi řetězec mnemotechnických zkratk a první parametr, a umožňuje tak přístroji rychle a jednoduše rozlišit jednotlivé části zprávy. Nejběžnějším oddělovačem je mezera, resp. libovolný počet mezer; podle standardu je však možné použít jakýkoli netisknutelný znak z tabulky ASCII vyjma znaku „LF“ = *Line Feed*, který posouvá kurzor na další řádek (souřadnice 0A_{hex} v ASCII tabulce – viz obr. 2.1).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	-	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Obr. 2.1 ASCII tabulka [4]

Podle standardu IEEE 488.2 lze rozlišit tři typy hlaviček, které jsou používány jak pro příkazy, tak pro SCPI dotazy. Prvním druhem je **prostá hlavička**, tedy jedna samostatná mnemotechnická zkratka. Příkazů v takovém formátu není mnoho, za příklad mohou posloužit příkazy INITiate nebo ABORt.

Dalším typem je nejrozšířenější, klasická podoba SCPI příkazu – se **složenou hlavičkou**. Ta sestává z několika klíčových slov v podobě mnemotechnických zkratk, která jsou oddělena dvojtečkou, a která vystihují jednotlivé větve hierarchické struktury příkazů SCPI. Příkladem může být právě nastavení rozsahu frekvence SENSE:FREQUENCY:SPAN.

Poslední variantou je **hlavička obecných příkazů**, v angl. *Common Commands*. Tento typ je užit pouze pro obecné příkazy, což je skupina zpráv se specifickou syntaxí, kterými je řízen registr stavů, a které lze rovněž použít pro několik provozních funkcí přístroje. Používají se ve zprávách typu command i query. Z hlediska syntaxe mají hlavičky takových příkazů jednu zvláštnost – všechny common commands vždy začínají hvězdičkou. Typickým zástupcem takové této skupiny hlaviček je příkaz typu query *IDN?, kterým řídicí počítač požádá měřicí zařízení o identifikaci a jako odpověď dostane řetězec odpovídající platnému nastavení přístroje (obvykle obsahuje název přístroje a číslo modelu, příp. číslo verze firmware).

2.2.2 Parametry

Část zprávy vyhrazená parametrům reprezentuje hodnotu předávanou SCPI příkazem. Formát parametrů se může lišit v závislosti na typu předávané zprávy: rozlišujeme tak např. číselné, znakové nebo boolean parametry. Detailnímu popisu jednotlivých druhů parametrů je

speciálně věnována kapitola 2.4, zde jsou parametry uvedeny především v kontextu syntaxe SCPI zpráv.

Měřicímu zařízení lze předat jedním příkazem více parametrů a v takovém případě je třeba je oddělit – k tomu slouží *data separator*, z ASCII tabulky je pro tento účel použita čárka. Bylo zmíněno, že parametry mohou nabývat mnoha různých podob, existují však SCPI příkazy, které naopak parametry nemají – je to typ, který je popsán v kapitole 2.2.1 jako příkazy s prostou hlavičkou. Jedná se většinou o povely související s činností přístroje, jako jsou INITiate (= zahájit měření) nebo *WAI (= počkej na dokončení předchozí instrukce). Pro tento typ příkazů je navíc typické, že nemají svou variantu *query*.

Následující ukázka porovnává možné formy parametrů – jako první je bezparametrický příkaz, kterým lze přerušit právě probíhající operaci:

```
ABORT
```

Dalším je typický příkaz s jedním parametrem typu boolean – logická hodnota ON v tomto případě zapne jednorázové snímání průběhu:

```
INITiate:CONTinuous ON
```

Jako příklad vícenásobných parametrů může sloužit příkaz k přidání druhého měřeného průběhu – parametry jsou ve formátu řetězce a nastavují název nového průběhu na „Prubeh2“ a s-parametr na odrazový s_{11} .

```
CALCulate:PARAmeter:SDEFine "Prubeh2", "S11"
```

Výše popsaná pravidla platí pro zprávu typu „příkaz“ i „dotaz“ – *command* i *query*. Po odeslání zprávy *query*, kterou je možné se „dotázat“ přístroje, očekáváme příjem odpovědi, pro kterou v otázce syntaxe platí v podstatě opačná zásada. U tohoto typu je obvykle vynechána hlavička a přijatá data tvoří výhradně parametry v příslušném formátu. Takto by například vypadal dotaz na x-ovou souřadnici (frekvenci) Markeru 1 a odpověď na něj (3,2 GHz):

```
CALCulate:MARKer1:X?
```

```
>> 3.2e9
```

2.3 Typy zpráv a jejich tvar

2.3.1 Command vs. Query

V této práci bylo již několikrát zmíněno rozdělení SCPI příkazů na dva základní typy: *command* a *query*. Je tedy vhodné věnovat několik řádků jejich formálnímu popisu. Z pohledu programátora spočívá hlavní funkční rozdíl ve směru komunikace: příkazem je přístroji buď uložena nějaká činnost, nebo jsou jím nastaveny pomocí parametrů vlastnosti měření – přenos je tedy jednosměrný, zatímco po odeslání dotazu (ať už s parametry, či bez nich) je řídicím počítačem očekávána odpověď ve formátu řetězce – výměna dat v tomto případě probíhá obousměrně. Z hlediska syntaxe lze obě formy od sebe snadno odlišit – hlavička query je vždy zakončena otazníkem. Obě verze zpráv – command i query – mohou za hlavičkou obsahovat parametry.

Velmi výhodná vlastnost SCPI je snadný přechod mezi command a query. Při znalosti jednoho tvaru nemusí programátor složitě dohledávat jeho ekvivalent pro druhý typ zprávy, protože na samotném textovém řetězci, který tvoří hlavičku příkazu, se nic nemění, pouze je na konec hlavičky přidán otazník. K většině příkazů command tak existuje snadno sestavitelná varianta query, přičemž její podoba není (pro její jednoduché sestavení a zvýšení přehlednosti seznamů) konkrétně uváděna v tabulce příkazů, která bývá součástí manuálu každého přístroje. Namísto toho je u jednotlivých SCPI příkazů jen výslovně upozorněno na existenci jejich query formy. Snadnou konstrukci dotazu je možné demonstrovat na příkladu nastavení pozice Markeru 1, resp. jejího vyčtení.

Typ *command* (nastavení markeru na frekvenci 1 GHz):

```
CALCulate:MARKer1:X 1e9
```

A forma *query* (dotaz na pozici a odpověď přístroje):

```
CALCulate:MARKer1:X?
```

```
>> 1000000000
```

2.3.2 Zkracování příkazů

Nelze si nevšimnout, že všechny SCPI příkazy, které byly v předchozím textu jako ukázky uvedeny, jsou zapsány použitím zvláštní kombinace velkých a malých písmen. I když se jedná pouze o vizuální formátování znaků, které na funkčnost příkazů samotných nemá

vliv, není svědomité dodržování tohoto úzu samoúčelné. Jak již bylo uvedeno, klíčová slova, ze kterých se skládají hlavičky SCPI a některé parametry, představují jednotlivé uzly hierarchické struktury a jsou mnemotechnickými jednoslovnými zkratkami vyjadřujícími svou funkci. Tato hesla mají však ještě svou vlastní zkrácenou verzi, která je právě za účelem přehlednosti zvýrazňována při zápisu příkazu v jeho dlouhé formě; ta může nabývat maximální délky 12 znaků. Vytvoření krátké formy se řídí třemi jednoduchými **pravidly**:

1. Zkratka je vytvořena použitím prvních čtyř znaků dlouhé podoby slova.

CALCulate -> CALC
SENSe -> SENS ...

2. Pokud je čtvrtým znakem slova samohláska, jsou použity pouze první tři znaky.

PARAmeter -> PAR
REFerence -> REF ...

3. Jestliže je dlouhá forma tvořena čtyřmi znaky, je dlouhý i zkrácený tvar totožný.

SPAN -> SPAN
DATA -> DATA ...

Použití dlouhé či krátké formy zápisu nemá žádný vliv na význam příkazu či na jeho zpracování v měřicím zařízení, jde zkrátka jenom o formální podobu, přičemž záleží pouze na preferenci uživatele, kterou možnost zápisu si zvolí (což platí i pro použití velký/malých písmen). Pokud je upřednostňována rychlost a stručnost, použije programátor pravděpodobně zjednodušenou variantu; bude tím ovšem potlačena jedna z charakteristických vlastností klíčových slov – ta byla záměrně volena tak, aby mnemotechnicky napovídala vlastní funkci, což příkazy ve zkráceném tvaru ztrácí. Zároveň je však třeba dbát na přesný tvar příkazu – ať už je použita jedna nebo druhá verze, nelze použít jiný počet písmen, než je ve standardu SCPI předepsán. Pro zařízení, které je příjemcem zprávy, by tak vlastně vznikl nový příkaz, který nezná, a který by způsobil *Command Error*. Neexistuje proto možnost volitelného zkrácení, např:

CALCulate -> ~~CALCUL~~

2.3.3 Číselné značení v SCPI zprávách

V syntaxi SCPI příkazů je třeba ještě věnovat pozornost drobnosti, jejíž použití může být ovšem klíčové – číselné příponě, která může doplňovat některá klíčová slova v odesílané zprávě. V případě, že příkaz ovládá funkci přístroje, která je zastoupena několika shodnými alternativami, je možné a žádoucí pomocí tohoto numerického značení upřesnit cíl, který chceme ovládat. V praxi se může jednat o řízení hardwaru, např. rozlišení dvou vstupních portů, kterými zařízení disponuje:

```
INPut1; INPut2 atd.
```

Časté je také použití číselného dodatku ke klíčovému slovu za účelem softwarového odlišení adresáta příkazu; jako příklad může posloužit nutnost určení měněného průběhu v případě několika paralelně zobrazených *Trace*:

```
TRACe1; TRACe2 atd.
```

Dalším typicky opakovaně se vyskytujícím prvkem jsou *Markery*:

```
CALCulate:MARKer1:X 1e9; :CALCulate:MARKer2:X 1.2e9 atd.
```

Aby nedocházelo k nedefinovaným stavům (a zároveň jako ulehčení práce programátorům), je výchozí hodnota všech příkazů, které vyžadují číselnou příponu, nastavena na „1“. To znamená, že pokud by plná verze příkazu vypadala kupříkladu takto:

```
CALCulate1:MARKer1:REFerence:MODE CONTinuous
```

... pak lze vynechat číselné přípony při zachování stejného významu zprávy:

```
CALCulate:MARKer:REFerence:MODE CONTinuous
```

A pokud aplikujeme pravidla stanovená v teorii zkracování příkazů (kapitola 2.3.2), je možné zapsat celý příkaz následujícím způsobem, přičemž pro přístroj, který ho přijme, se budou všechny tři varianty jevit jako naprosto shodné.

```
CALC:MARK:REF:MODE CONT
```

2.4 Parametry SCPI příkazů

Důležitou oblastí standardu SCPI popsanou v normě IEEE 488.2 jsou parametry příkazů, kterým byla kvůli jejich významu a rozsahu věnována tato samostatná kapitola. V předchozích částech textu byla již stručně zmíněna role parametrů v syntaxi SCPI zpráv, stejně jako byly popsány jejich základní vlastnosti a charakteristiky. Jedním z uvedených rysů byla podoba dat přijatých v reakci na dotaz *query* vyslaný řídicím počítačem do měřicího zařízení. Tato odpověď v sobě narozdíl od příkazu *command* neobsahuje hlavičku, ale o to větší důraz je kladen na formálně správnou podobu dat ve formě parametrů. A protože příchozí parametry potřebujeme zpravidla dále zpracovat (buď reagovat na stav přístroje, nebo provádět analýzu přijatých dat) je podstatné pochopit definice a funkce parametrů, se kterými se v SCPI zprávách pracuje. Konečně, naměřený průběh bývá hlavním cílem použití měřicího přístroje.

V této práci už byly pro ilustraci uváděny různé druhy parametrů jako součást SCPI příkazů, takže je vhodné se nyní tomuto tématu věnovat systematicky a zavést formální kategorie. Datových typů, které mohou parametry vyjadřovat, je několik a některé příkazy dokonce nemusí mít parametr pouze jednoho datového typu. V následujících kapitolách budou jednotlivé možnosti popsány a na ukázkách bude předvedeno jejich využití v praxi.

2.4.1 Datový typ znak

Data označovaná jako typ „znak“ mají podobu mnemotechnických zkratk, ne nepodobných klíčovým slovům v hlavičkách SCPI příkazů. Platí pro ně také stejná pravidla zápisu, včetně zkracování. Stejně jako všechny parametry jsou od hlavičky odděleny pomocí *header separator*, tvořeným nejčastěji mezerou. Následující příkazy jsou uvedeny bez vzájemné souvislosti a slouží pouze jako názorné příklady takových parametrů (svislou čarou se obvykle odděluje více použitelných parametrů a lze ji vnímat v jejím logické smyslu jako OR = NEBO):

```
INPut:POLarity  NORMAl | INVerted
```

```
DISPlay:WINDow:TRACe:MODE WRITe | AVERage | MAXHold | MINHold
```

Zvláštní způsob využití tohoto datového typu představují uživatelem definovaná jména, která lze pomocí SCPI přiřadit průběhům *Trace*, souborům apod. V takovém případě pochopitelně tyto názvy nemají svou zkrácenou verzi; skládat se mohou z velkých a malých

písmen, číslic a podtržítka. Jako ukázka může posloužit příkaz zavádějící průběh s názvem „Muj_Prubeh“ (= datový typ znak), který sestává z 50 datových prvků.

```
TRACe:DEFine  Muj_Prubeh,  50
```

2.4.2 Datový typ řetězec

Tento datový typ je velmi podobný předchozímu znakovému, avšak přestože také představuje textová data, existuje důvod, proč jsou obě skupiny rozlišovány a tvoří samostatné kategorie. Z hlediska syntaxe spočívá rozdíl v množině znaků, které lze použít – řetězec je možné sestavit z takřka všech znaků tabulky ASCII (obr. 2.1). Odlišné je také použití při sestavování SCPI zprávy, kdy je nutné řetězec umístit buď do jednoduchých (, ‘), nebo dvojitých („ “) uvozovek. Takto vymezená data mohou tedy obsahovat i netisknutelné řídicí znaky, jako jsou „LF“ = *Line Feed*, který posouvá kurzor na další řádek, či „CR“ = *Carriage Return*, kterým je kurzor posunut na začátek řádku. Těmito speciálními znaky se dá textový řetězec naformátovat, čehož může být využito při tisku nebo při zobrazení takových dat na monitoru.

Standardem předepsané ohraničení uvozovkami může začít působit komplikace v případě, že součástí textu, který chceme zobrazit, mají být uvozovky jako obyčejný znak. Pro tento účel slouží právě dvojí verze uvozovek – pokud chceme v řetězci použít dvojitě uvozovky, celý text orámujeme jednoduchými uvozovkami, a naopak – viz ukázka:

```
DISPlay:TEXT  „ Textovy ‚Retezec‘ “  
DISPlay:TEXT  ‚ Textovy „Retezec“ ‘
```

Do tohoto způsobu formátování, který může díky nepozornosti snadno způsobit chybu v příkazu, ještě navíc vstupuje syntaxe počítačového vyššího programovacího jazyka, ve kterém je tvořena aplikace používající SCPI. Například ve vývojovém prostředí Matlab je celá SCPI zpráva odesílána jako řetězec, který je sám ještě ohraničen jednoduchými uvozovkami. Může tak nastat situace, kdy je třeba správně umístit až tři páry vnořených uvozovek.

2.4.3 Dekadický číselný datový typ

Pro zápis číselných parametrů v desítkové soustavě slouží dekadický datový typ. V tomto formátu jsou zapisována čísla třemi možnými způsoby: prvním je zápis **celočíselné**

hodnoty (v programátorské praxi též *integer*), která se skládá pouze z číslic a nemá desetinnou čárku, může navíc obsahovat jen znaménka plus nebo mínus. Dalším typem je **reálné číslo**, jehož desetinná část je oddělena tečkou; rovněž s možností použití znamének plus a mínus. A nakonec je to zápis v podobě **exponenciální**, tedy rozdělení čísla na mantisu a exponent, přičemž mantisa může být ve tvaru celočíselném či reálném. Obě části se mohou samozřejmě vyskytovat se znaménkem plus nebo mínus. Zajímavostí je, že při konstrukci SCPI zprávy je možné při uvedení parametru v exponenciálním tvaru oddělit mantisu a exponent libovolným počtem mezer.

Počet platných číslic závisí na přístroji, kterému data posíláme, ale v zásadě je možné se spolehnout na jednu důležitou vlastnost měřicích zařízení pracujících na standardu SCPI, a tou je **zaokrouhlování**. Číslo, které je takovému přístroji odesláno, a které – pokud jde o přesnost – přesahuje jeho schopnosti, není na platný počet číslic oříznuto, nýbrž zaokrouhleno.

Číselné parametry, které řídicí počítač předává příkazy *command*, často vyjadřují hodnotu nějaké fyzikální veličiny; např. při nastavování rozsahu zobrazované frekvence. SCPI kompatibilní zařízení umožňují v takovém případě do zprávy přidat jednotku dané veličiny, a to včetně **předpony soustavy SI** (viz následující tabulka).

Tabulka 2.1 Přehled předpon soustavy SI a jejich značení v SCPI [1]

10^n	Značka	Předpona
10^{12}	T	tera
10^9	G	giga
10^6	MA	mega
10^3	K	kilo
10^{-3}	M	mili
10^{-6}	U	mikro
10^{-9}	N	nano
10^{-12}	P	piko
10^{-15}	F	femto
10^{-18}	A	atto

Výhoda tohoto způsobu zadávání je evidentní – pro uživatele je snazší použít předponu, se kterou je zvyklý pracovat, než vypisovat mantisu a exponent. Následující dvě

podoby jedné zprávy jsou tedy ekvivalentní, a záleží jen na osobní preferenci, která z nich bude použita.

```
SENSe:FREQuency:START 3e9
```

```
<=>
```

```
SENSe:FREQuency:START 3GHz
```

V této souvislosti je třeba upozornit na použití předpony „mega“, resp. na její značení, které se liší od standardního „M“ soustavy jednotek SI. Ve SCPI příkazech jsou místo toho použita dvě písmena „MA“; důvodem je již dříve popsaná vlastnost SCPI měřicích zařízení, neboť tato nerozlišují velká a malá písmena. Pro rozlišení „mili“ a „mega“ bylo proto nutné zavést novou předponu, a například jednotka megahertz je tak značena jako [MAHz]. Problém vzniká u některých přístrojů, které – kvůli zažité praxi používat pouhé velké „M“ – označují jednotku megahertz běžným [MHz]. Pokud chce uživatel používat předpony SI spolu s jednotkami, je tedy vhodné předtím nahlédnout do manuálu příslušného měřicího zařízení.

Z pohledu přístroje, který odpovídá na dotaz *query*, takové problémy odpadají. Příchozí odezva je vždy v základních jednotkách SI, tzn. nejen v původním měřítku bez předpony, ale také ve standardních jednotkách dané fyzikální veličiny.

2.4.4 Ostatní číselné datové typy

Hodnoty, které chceme předat měřicímu zařízení jako číselné parametry příkazů, může být někdy vhodné zapsat v jiné než desítkové soustavě. Standard SCPI proto umožňuje vyjádření čísla nejen dekadicky, ale také v hexadecimální, osmičkové či binární soustavě. Aby přístroj dokázal rozlišit, že jsou použity tyto speciální číselné datové typy, je třeba v syntaxi parametrů před samotnou hodnotu čísla přidat jistou předponu (značku), která se skládá ze znaku „křížek“ = # a písmena určujícího, o kterou soustavu se jedná. Následující tabulka ukazuje přehled možností zápisu stejného čísla:

Tabulka 2.2 Přehled číselných soustav parametrů SCPI příkazů [1]

Číselná soustava	Značka	Příklad
Desítková	---	42
Hexadecimální	#H	#H2A
Osmičková	#Q	#Q52
Binární	#B	#B101010

Přístroj, který SCPI zprávu přijme, není citlivý na číselnou soustavu předaného parametru, čili použití konkrétního formátu čísel závisí jen na preferencích uživatele. Dobře lze tuto vlastnost ilustrovat na příkazech nastavujících registry, např. *Operation Event Enable* registr. První příkaz nastavuje jednotlivé bity registru na hodnoty 0000 0101 0101 0010 a využívá při tom snadno srozumitelného binárního čísla:

```
STATus:OPERation:ENABLE #B0000010101010010
```

Stejného nastavení registru lze ovšem dosáhnout při použití hexadecimální hodnoty čísla, případně jeho alternativy v dekadické či osmičkové soustavě, a na významu zprávy se přitom nic nezmění:

```
STATus:OPERation:ENABLE #H552
STATus:OPERation:ENABLE 1362
STATus:OPERation:ENABLE #Q2522
```

V případě dotazu *query* je ve výchozím nastavení použita v odpovědi desítková soustava, nicméně formát odezvy na query o obsahu stavového registru lze příslušným SCPI příkazem změnit, takže reakce přístroje by mohla být následující:

```
STATus:OPERation:ENABLE?
>> 1362

FORMat:SERGister HEXadecimal;

STATus:OPERation:ENABLE?
>> #H552
```

2.4.5 Numerické hodnoty pomocí klíčových slov

Číselné hodnoty, které chceme měřicímu zařízení předat pomocí parametrů, nemusí být vždy vyjádřeny pomocí konkrétního čísla. Často je vhodnější – a v některých případech dokonce žádoucí – vyjádřit tuto hodnotu pomocí mnemotechnické zkratky. Jedná se o standardem definovaná klíčová slova se specifickým významem; jejich seznam je uveden v tabulce níže.

Tabulka 2.3 Seznam klíčových slov reprezentujících numerické hodnoty [6]

Klíčové slovo	Význam
MAXimum	Nastavení maximální možné hodnoty
MINimum	Nastavení minimální možné hodnoty
DEFault	Automatická hodnota
UP	Zvýšení hodnoty o jeden krok
DOWN	Snížení hodnoty o jeden krok
INFINITY	Nekonečno (9.9E+37), pouze návratová hodnota
NINFINITY	Záporné INF (-9.9E+37), návratová hodnota
NaN	„Není číslo“ (9.91E+37), návratová hodnota

Obecně platí, že u všech příkazů, jejichž parametry se běžně vyjadřují numericky, je možné použít dvě klíčová slova: **MAXimum** a **MINimum**. Ta podle očekávání nastavují funkci, ve které jsou použity, na maximální, resp. minimální hodnotu. Rovněž varianty *query* takových příkazů mohou obsahovat parametry MAX a MIN, přičemž odpověď na ně bude podávat informaci o minimální nebo maximální dovolené hodnotě vzhledem k aktuálnímu nastavení přístroje. Jako příklad může posloužit nastavení programovatelné ořezové (cutoff) frekvence vstupního filtru některých zařízení. Následující SCPI příkazy názorně ukáží rozdíl mezi reakcí na query bez parametrů a na dotaz s přidáním parametru MAXimum – je to tedy také ukáзка situace, kdy je přímo nutností použít klíčové slovo ve smyslu číselné hodnoty.

Uvažujme měřicí zařízení s programovatelnou cutoff frekvencí vstupního filtru v rozsahu od 10 Hz do 8 kHz. Jako první odešleme příkaz typu *command*, kterým nastavíme vstupní filtr na cutoff frekvenci 3 kHz.

```
INPut:FILTer 3e3
```

Následují tři příkazy *query*, na nichž je dobře vidět vliv přidání parametru:


```
INPut:FILTer?
```

```
>> 3E+3
```

```
INPut:FILTer? MAXimum
```

```
>> 8E+3
```

```
INPut:FILTer? MINimum
```

```
>> 10
```

Možnost použití dalších příkazů je volitelná a tato klíčová slova nemusí být vždy platná u všech přístrojů. Je proto vhodné před jejich užitím ověřit jejich použitelnost v dokumentaci zařízení. Jednou z takových mnemotechnických zkratek je slovo **DEFault**, které představuje automaticky určenou hodnotu, jež je použita, pokud není konkrétním parametrem v příkazu daná vlastnost nastavena. V případě automatického nastavení přístroj sám vybere hodnotu, kterou považuje za vhodnou – lze tedy nechat záměrně výběr vhodného parametru na samotném SCPI zařízení. Jelikož většina příkazů nemá volitelné parametry, které lze jednoduše vynechat, je potřeba jejich hodnotu nějakým způsobem předat; to i přesto, že parametry buď nevíme, nebo jejich hodnota pro nás není důležitá – a k tomu slouží právě klíčové slovo DEFault. Měřicí zařízení si velikost takto označeného parametru samo zvolí, a to v závislosti na aktuální situaci. Z toho plyne, že nastavení parametrů do režimu „automaticky“ nepředstavuje použití konstantního čísla, ale je naopak proměnné jako důsledek snahy o vyhovění ostatním parametrům měření. Je tedy nutné rozlišovat automatickou hodnotu od hodnoty „reset“, která je vždy pevně daná, a již lze nalézt v dokumentaci přístroje.

Následující dva příkazy ukazují rozdílné reprezentace klíčového slova DEFault. Prvním je nastavení vnitřních hodin přístroje; jeho syntaxe je SYSTem:TIME <hodina>, <minuta>, <sekunda>. Při použití DEFault bude pravděpodobně takový čas nastaven na 0:00.00.

```
SYSTem:TIME DEFault, DEFault, DEFault
```

Naproti tomu u příkazu nastavujícího rozsah voltmetru bude slovem DEFault vybrána nejvyšší možná hodnota – jako nejbezpečnější automatická volba.

```
SYSTem:VOLTage:RANGe DEFault
```

Klíčovými slovy **UP** a **DOWN** se zvyšuje, resp. snižuje hodnota dané veličiny o jeden krok. Funkci těchto příkazů ilustruje další ukázka: nastavování napětí výstupního signálu signálového generátoru v krocích. Nejprve je nutné jedním příkazem nastavit velikost jednoho kroku, v tomto případě např. 2 dB.

```
SOURce:POWer:STEP:INCRement 2
```

Poté je třeba přístroj přepnout do módu pro uživatelem zadané krokování:

```
SOURce:POWer:STEP:MODE USER
```

Nyní je možné výstupní napětí nastavovat po krocích použitím příkazů UP a DOWN; zde konkrétně je jeho hodnota o jeden krok (tj. o nastavené 2 dB) zvýšena. Stejný princip samozřejmě platí i u klíčového slova DOWN – hodnota napětí by takto byla o 2 dB snížena.

```
SOURce:POWer:LEVel:IMMediate:AMPLitude UP
```

Kladné a záporné nekonečno v SCPI přístrojích je pochopitelně třeba chápat nikoli v matematickém smyslu, ale spíše jako vyjádření rozsahu konkrétních, teoreticky použitelných číselných hodnot, které lze při komunikaci s měřicím zařízením použít. Tyto hranice jsou představovány určitými čísly (viz tabulka 2.3); pro kladné nekonečno je to $9,9 \cdot 10^{37}$, záporné nekonečno je vyjádřeno jako $-9,9 \cdot 10^{37}$. Tyto hodnoty je možné dle SCPI standardu popsat klíčovými slovy **INFINITY**, resp. **NINFINITY**, a jakékoli číslo mimo jimi vymezený interval nebude přístroji předáno a bude vyhodnoceno jako chyba. Zároveň je nutné si uvědomit, že i hodnota, která je v pořádku z hlediska obecného rozsahu, nemusí nutně splňovat limit fyzikálních možností daného přístroje. Například generátoru pracujícímu v rozsahu frekvencí 9 kHz až 6 GHz bude příkazem předána frekvence $7 \cdot 10^9$ Hz, neboť je z intervalu (NINF; INF), ale přístroj nemůže takovou hodnotu nastavit – příkaz se tedy neprovede a zobrazí se varování „Value out of range“.

Poslední klíčové slovo – **NANumber** („není číslo“) – je podobně jako nekonečno také reprezentováno určitou číselnou hodnotou, a to $9,91 \cdot 10^{37}$. Jelikož je v zařízeních často možné vykonávat s naměřenými průběhy matematické operace, může se v SCPI zprávě objevit NANumber jako návratová hodnota např. při dělení nulou.

2.4.6 Booleovský datový typ

Oblast použití booleovské logiky je v SCPI příkazech skutečně rozsáhlá, velké množství funkcí každého přístroje je možné pomocí tohoto datového typu zapínat a vypínat; případně jím ve formě návratové hodnoty na dotaz *query* vyhodnocovat jejich stav. Zadání booleovských parametrů lze provést uživatelsky nejpříjemnějším vyjádřením: True = **ON**, False = **OFF**. Je ovšem možné použít také čísel, a to podle obvyklého smyslu True \neq 0, False = 0; neboli jakékoli číslo nerovné nule bude chápáno jako vyjádření True. V reakcích na *query* je v odpovědích na SCPI zprávy použita binární logika; tj. True = 1, False = 0.

Níže uvedené příkazy (jedná se o zapnutí obrazovky přístroje při vzdáleném řízení) předvádí různé možnosti zadání těchto parametrů. Nejprve pomocí klíčového slova ON:

```
SYSTem:DISPlay:UPDate ON
```

Také použitím libovolného čísla různým od nuly:

```
SYSTem:DISPlay:UPDate 4
```

Na *query* bude odpověď přístroje vždy číselná, a to buď 1 (True) nebo 0 (False):

```
SYSTem:DISPlay:UPDate?
```

```
>> 1
```

Další ukázkové příkazy jsou jenom vzorkem z mnoha možných aplikací parametrů booleovského datového typu. Sloužit mohou např. pro zapnutí/vypnutí markeru...

```
CALCulate:MARKer1:STATe ON (OFF)
```

... pro zapnutí/vypnutí plynulého snímání průběhu (Continuous Sweep)...

```
INITiate:CONTinuous ON (OFF)
```

... pro zapnutí/vypnutí amplitudové modulace...

```
SOURce:AM:STATe ON (OFF)
```

... a v mnoha dalších případech.

2.4.7 Libovolný datový blok

Pod tímto datovým typem se rozumí přenos celého souhrnu (bloku) libovolných informací, které jsou přenášeny v podobě datových bytů. Výhoda takového formátu vynikne především u transferu většího množství dat – 8bitově kódovaná data totiž nemusí dbát na srozumitelnost pro člověka (jsou určena pro další strojové zpracování), což umožňuje jejich přenos efektivním a kompaktním způsobem.

Existují dva možné tvary, kterých může libovolný datový blok nabývat. Odlišují se přístupem k délce datového bloku; konkrétně k její znalosti a explicitního uvedení. Ve struktuře libovolného datového bloku totiž samotným datům předchází hlavička, jež obsahuje pouze informace o struktuře dat, která jsou přenášena. A pokud je v těchto informacích uveden údaj o celkové (hlavička + data) velikosti bloku v bytech, pak tento typ označujeme jako datový blok s **určenou délkou**. Ukázka hlavičky tohoto druhu formátu předvádí tři části, ze kterých sestává, a které jsou popsány níže:

```
#214<B><B><B><B><B><B><B><B><B><B><B><B><B><B>
```

Znak „#“ označuje začátek datového bloku; číslo „2“ značí, kolik následujících číslic bude popisovat délku přenášených dat. V tomto případě je délka označena číslicemi „1“ a „4“, to znamená, že datový blok bude obsahovat 14 jednotlivých bytů.

Druhým typem tohoto formátu je libovolný datový blok s **neurčenou délkou**. I v tomto případě vysílaným bytům předchází hlavička, která obsahuje znak křížek = „#“, ovšem na následující pozici je vždy číslice „0“. Tím obsah hlavičky končí a následují byty přenášených dat. Následuje ukázka struktury stejného datového bloku jako v předchozím případě, nyní však s předem neurčenou délkou:

```
#0<B><B><B><B><B><B><B><B><B><B><B><B><B><B>LF^End
```

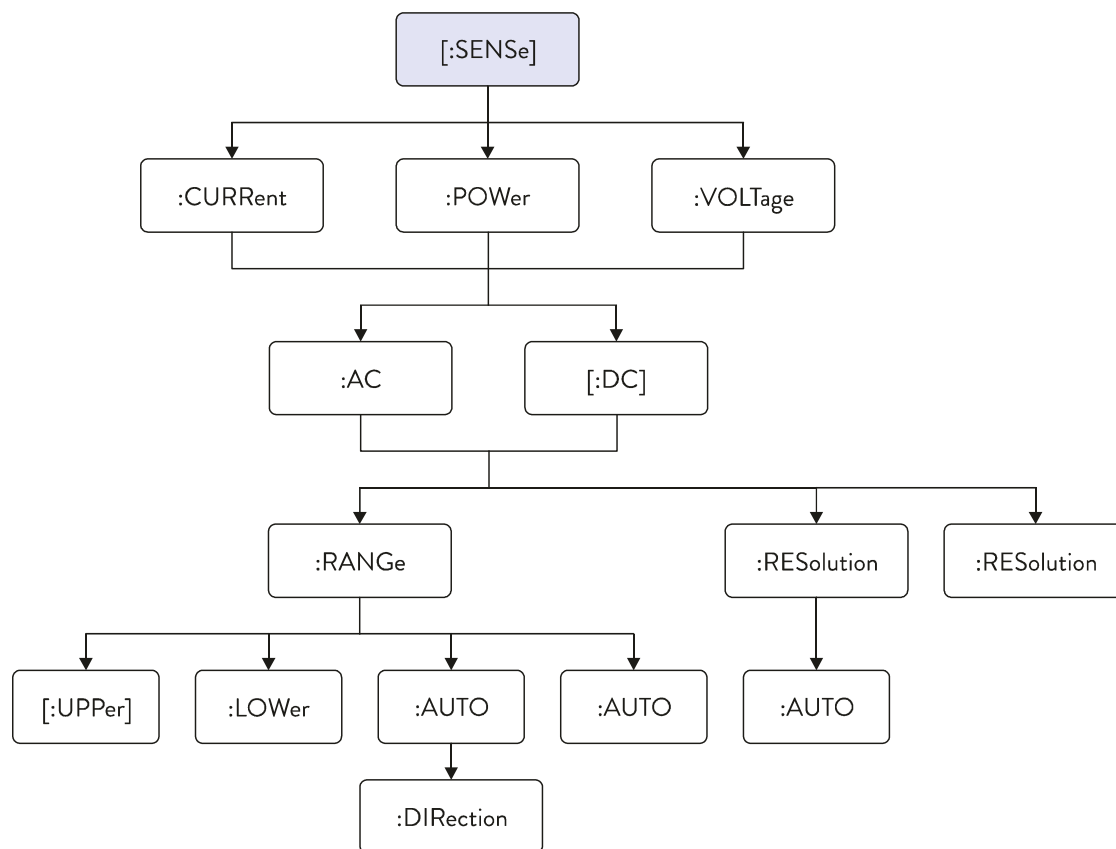
V porovnání s předchozím formátem vidíme, že na konci bloku přibýly dva nové znaky. Aby totiž bylo zřejmé, kde datový blok končí, je nutné na konec řetězce bytů připojit ještě znak „LF“ = *Line Feed*, (souřadnice 0A_{hex} v ASCII tabulce – viz obr. 2.1) a ukončení zprávy „End“. To však činí tento způsob velmi specifickým, a to ze dvou důvodů: první je spíše formální, neboť podstata funkce terminátoru „End“ spočívá v ukončení nejen bloku libovolných dat, ale také celé zprávy. To je třeba při programování kódu zohlednit, a případně další dotazy *query*, které přístroji chceme poslat v jedné zprávě, musí předcházet dotazu, na

který bude zařízení odpovídat právě pomocí datového bloku s neurčenou délkou. Druhý důvod je závažnější – zpráva typu „End“ je použitelná pouze u sběrnice GPIB a bloky s neurčenou délkou proto nelze posílat přes jiný typ komunikačního rozhraní. Přístroje pracující na standardu SCPI se však naopak snaží o nezávislost na konkrétní realizaci fyzické vrstvy; zařízení by mělo spolehlivě pracovat nezávisle na zvoleném způsobu spojení s řídicím počítačem (GPIB, Ethernet, USB aj.) – proto je zpravidla použit spíše datový blok s určenou délkou.

2.5 Hierarchický strom SCPI příkazů

2.5.1 Hierarchie uzlů

V předchozích kapitolách bylo již několikrát poukazováno na strukturovanost SCPI příkazů a na posloupnost uzlů, jejichž skládáním vznikají hlavičky SCPI. V této části se tedy podrobněji zaměříme na konkrétní podobu takového uspořádání. Skladba vystihující hierarchii jednotlivých uzlů je nazývána „Strom příkazů“ – toto označení dobře vystihuje povahu složení SCPI příkazů a názorně ho ilustruje ukázkové rozdělení uzlů na obr. 2.2. Každá mnemotechnická zkratka použitá v hlavičce SCPI příkazu představuje jeden uzel ve stromu příkazů, a podle očekávání tvar hlavičky koresponduje se strukturou stromu – první klíčové slovo v hlavičce odpovídá nejvyšší úrovni v hierarchii (na ní se nachází vždy jeden z podsystémů SCPI příkazů), přičemž každé další klíčové slovo je přidáno analogicky k cestě „sestupu“ ve stromu příkazů skrze uzly směrem k požadované funkci. Čím níže se tedy ve stromu nacházíme, tím více se vlastně zpřesňuje původně obecný podsystém až do podoby konkrétního příkazu.



Obr. 2.2 Hierarchický strom uzlů podsystému SENSE [1]

Na obr. 2.2 vidíme na nejvyšší pozici základní uzel – hlavní podsystém SCPI příkazů. Podsystémů rozlišujeme celkem 23; podrobněji jsou popsány v kapitole 2.6. V tomto případě byl zvolen systém SENSE (přesněji jeho část), ze kterého dále vybíhají větve uzlů příslušejících dostupným funkcím ovládaného přístroje. Uzly se mohou dále členit, v tomto případě až do 5. úrovně. Je patrné, že takto znázorněné schéma hierarchie příkazů popisuje soustavu sice přehledně, ovšem z praktického hlediska poněkud neobratně. Pokud bychom takovýmto stromem popisovali každý dostupný podsystém SCPI příkazů platných pro daný přístroj, byl by takový způsob značně rozmáchlý. V praxi se proto pro seznam příkazů používá výpis do tabulky, ve které jsou zároveň zpravidla uvedeny platné parametry a také existující varianty příkazů – command, query, příp. obojí. Samotný příkaz je zapsán ve tvaru složené hlavičky.

2.5.2 Skládání hlavičky příkazu

Hlavička SCPI příkazu byla již podrobně popsána v kapitole 2.2.1; nyní si uvedeme několik pravidel vyplývajících z podstaty její hierarchické struktury. Formálně lze v hlavičce

oddělit 2 části: cestu příkazu a koncový uzel. Na příkladu povelu nastavujícím aktuální úroveň výstupu signálového generátoru jsou tyto úseky názorně předvedeny.

Celá hlavička SCPI příkazu:

```
SOURce:POWer:LEVel:IMMediate:AMPLitude
```

Cesta příkazu:

```
SOURce:POWer:LEVel:IMMediate
```

Koncový uzel:

```
AMPLitude
```

Toto rozlišení jednotlivých segmentů hlavičky není úplně samoúčelné – cesta příkazu je totiž často pro několik koncových uzlů stejná, a lze ji proto v jedné SCPI zprávě uvést pouze u prvního příkazu, přičemž pro další příkazy zůstane tato cesta platná. Jedna z mnoha možností využití je například při nastavení rozsahu frekvence spektrálního analyzátoru. Pokud bychom chtěli nastavit počáteční frekvenci na 3 GHz a koncovou na 5 GHz, vypadala by SCPI zpráva v plné formě takto:

```
SENSe:FREQuency:START 3e9; :SENSe:FREQuency:STOP 5e9
```

Dvojtečka předcházející druhému příkazu přístroji říká, že tento příkaz bude uveden od základního uzlu a je v syntaxi povinná. Je však zřejmé, že cesta příkazu je v obou případech totožná, a obě instrukce se liší pouze koncovým uzlem. Není tedy nutné opakovaně vypisovat celý tvar – po vypuštění cesty příkazu zůstává v SCPI zprávě právě jen koncový uzel. Druhý příkaz přitom tentokrát přebírá cestu z prvního (včetně základního uzlu SENSe) a dvojtečka před něj tudíž nepatří:

```
SENSe:FREQuency:START 3e9; STOP 5e9
```

Pro popis další vlastnosti se ještě vrátíme ke stromu SCPI příkazů na obr. 2.2 – je možné si všimnout, že některé uzly obsahují klíčová slova, která jsou vymezena hranatými závorkami „[]“. Tak se v hierarchickém stromu či tabulce příkazů formálně značí uzly, které jsou přístrojem chápány jako výchozí, a které lze při odesílání SCPI vypustit. Platí potom rovnocennost obou podob:

```
SENSe:VOLTage:DC:RANGe:UPPer 200
```

```
<=>
```

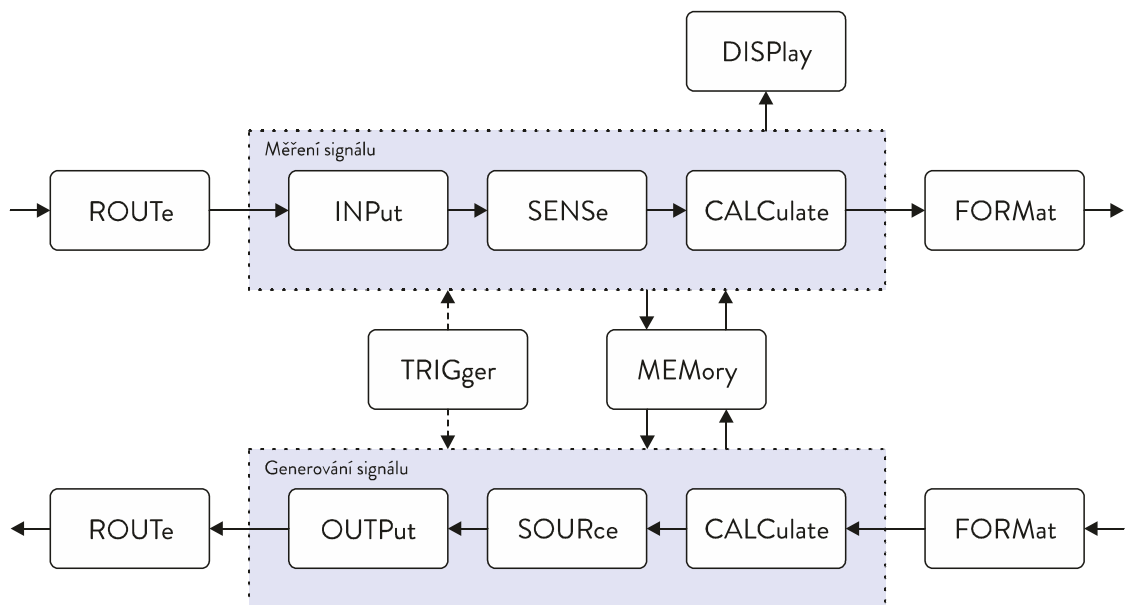
```
VOLTage:RANGe 200
```

Měřicí zařízení si výchozí klíčová slova v příkazové cestě samo doplní, což je třeba mít na paměti hlavně při výše zmíněném přebírání cesty druhým příkazem – pro druhý koncový uzel platí cesta včetně vynechaných slov. Stejně jako u zkracování příkazů (kapitola 2.3.2), i tentokrát je na uvážení programátora, v jaké míře je výhodné tuto vlastnost využít (především s ohledem na přehlednost a srozumitelnost zdrojového kódu).

2.6 Podsystemy SCPI příkazů

2.6.1 Model SCPI přístroje

Již v kapitole 2.5 byly zmíněny podsystemy SCPI rozdělující příkazy do velkých množin, které tvoří základní uzly v hierarchickém stromu. Nyní se soustavě subsystémů budeme věnovat blíže, protože toto téma konečně spojuje strukturu SCPI příkazů s funkcemi reálných měřicích zařízení. Příkazy, které jsou v podsystemech sdruženy, nejsou samozřejmě náhodné – spojuje je naopak jejich použití při obsluze funkcí souvisejících s určitou oblastí *Modelu SCPI přístroje* (viz obr. 2.3). Jejich účel se přitom (jak je ve standardu SCPI obvyklé) snaží vystihnout mnemotechnická zkratka, kterou je tvořen název podsystemu.



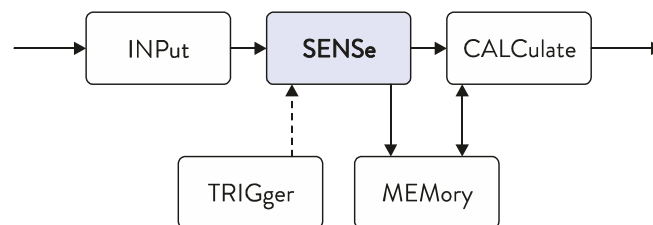
Obr. 2.3 Model SCPI přístroje [1]

Obecný model přístroje na obr. 2.3 je zobrazen jako blokové schéma subsystémů a jejich vzájemných vazeb, přičemž každý podsystém v tomto diagramu je základním uzlem vlastní, více či méně složité hierarchické struktury SCPI příkazů. Konkrétní úlohy, jež plní jednotlivé bloky, popisuje kapitola 2.6.3. Znázorněný model představuje všechny funkce, které může teoreticky měřicí zařízení poskytnout, a rovněž cesty, jimiž v takovém zařízení putují signál a data. Skutečné přístroje však zpravidla nenabízí všechny potenciální možnosti, a je tak pochopitelné, že např. spektrální analyzátor nebude obsahovat příkazy z oblasti „Generování signálu“, které budou naopak klíčovou částí signálových generátorů.

2.6.2 Základní rozdělení SCPI přístrojů

Na výše uvedeném modelu SCPI přístroje je patrné, že z hlediska funkcionality lze rozlišit tři hlavní skupiny měřicích zařízení; každou přitom vystihuje její vlastní subsystém².

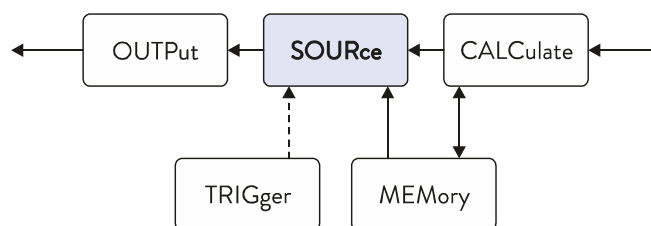
1. Přístroje pro **měření signálu** – mezi ně patří osciloskopy, spektrální analyzátoři, voltmetry apod. Klíčové příkazy pro zajištění jejich měřicí funkce jsou sdruženy v podsystému SENSe, který je doplněn dalšími subsystémy. Z obecného modelu SCPI přístroje tak (odpadnutím části určené generování signálu) získáme jednodušší schéma těchto zařízení (viz obr. 2.4).



Obr. 2.4 Blokové schéma přístroje pro měření signálu [1]

2. Jiný typ přístrojů je určen pro **generování signálu** – typickými zástupci jsou například signálové generátory, ale také napájecí zdroje aj. Hlavní podsystém těchto přístrojů je pojmenován SOURce, a jeho funkci mu opět pomáhají plnit další dodatečné podsystémy. Schéma zařízení generujícího signál na obr. 2.5 vychází z obecného modelu přístroje.

² Z pohledu syntaxe SCPI příkazů je pak název hlavního podsystému (v podobě mnemotechnické zkratky) automaticky považován za výchozí uzel a může být skládání cesty příkazů vynecháván.



Obr. 2.5 Blokové schéma přístroje pro generování signálu [1]

3. Třetí kategorie přístrojů má na starost **směrování signálu**. Jejich ovládací SCPI příkazy obsahuje subsystém ROUTE, jehož funkci lze odvodit od jeho názvu – přepíná a směřuje signál vstupního či výstupního portu, a pracuje proto často v kombinaci s bloky INPUT a OUTPUT. Jako příklad takového zařízení může sloužit např. multiplexer.

2.6.3 Popis podsystémů

Bloky použité ve schématu modelu SCPI přístroje na obr. 2.3 reprezentují nejdůležitější klíčové subsystémy, které spolu s naznačenými cestami signálu jasně vystihují základní možnosti měřicího zařízení. Funkcí, kterými je běžný přístroj vybaven, je ovšem mnohem více, a proto existuje také řada dalších podsystémů (celkem 23), ve kterých jsou tyto doplňující funkce sloučeny. Následující výčet popisuje stručně především smysl daných subsystémů; konkrétní příkazy v nich obsažené se totiž mohou v jednotlivých přístrojích lišit.

1. **CALCulate** – tento podsystém poskytuje nástroje pro *post-processing* snímaných dat (resp. *pre-processing* v případě přístrojů generujících signál). Pro tyto účely obsahuje příkazy pro matematické zpracování dat jako je derivace, integrace, výpočet průměru a další. Důležitou součástí tohoto systému jsou také příkazy obsluhující Markery.
2. **CALibration** – účel tohoto bloku jasně vystihuje jeho název: slouží k různým druhům kalibrací měřicího zařízení.
3. **CONTRol** – příkazy v tomto podsystému jsou specifické zejména v tom, že neovládají funkce pro zpracování signálu, ale jsou určeny pro ovládání „provozních“ součástí přístroje jako jsou ventilátory, kompresory apod.
4. **DIAGnostic** – název této skupiny příkazů je opět výstižný, neboť jsou zde sdruženy funkce zaměřené na diagnostiku, údržbu a opravu přístroje.

5. **DISPlay** – v tomto subsystému jsou obsaženy příkazy ovládající nejen všechny funkce související se způsobem zobrazování dat měření, ale také displej samotný, tj. například jeho jas či barvy. Umožňuje také úplně vypnout zobrazování měřeného průběhu na přístroji, což může být v případě vzdáleného řízení výhodné – pokud je průběh snímán aplikací v řídicím počítači, může pak vypnutí průběhu na přístroji ušetřit výpočetní výkon a urychlit tak měření. Struktura příkazů podsystému DISPlay může být poměrně rozsáhlá, protože některé přístroje mohou kromě vlastního displeje podporovat také zobrazení průběhů na externích monitorech.
6. **FORMat** – příkazy z tohoto subsystému nastavují formát, v jakém jsou přenášena data naměřeného průběhu, a to v rámci datových typů popsanych v kapitole 2.4 pojednávající o parametrech.
7. **HCOPY** – toto klíčové slovo je zkratka z anglického *Hard Copy*, a takto pojmenovaný subsystém má mnoho společného s podsystémem DISPlay – také se zabývá podobou naměřených průběhů, ovšem nikoli za účelem zobrazení, nýbrž pro následný tisk těchto dat. Součástí HCOPY jsou tedy příkazy nastavující formát stránky, umožňující výběr barev či styl zobrazení textu apod.
8. **INPut** – obsahuje funkce ovlivňující signál na vstupu do přístroje, a to ještě předtím, než je změřen AD převodníkem a zpracován pomocí systému SENSE. Typickými příkazy jsou např. zesílení či zeslabení signálu, filtrování, nastavení polarity signálu apod.
9. **INSTrument** – subsystém INSTrument je využit v případě, že jeden přístroj pracuje ve více módech (např. R&S[®]FSH obsahuje separátní režimy pro spektrální analyzátor, obvodový analyzátor, měření výkonu ad.). Tyto módy je pak možné přepínat pomocí příkazu INSTrument:SElect.
10. **MEMory** – ovládá interní paměť přístroje, to jest prostor určený pro přechodné ukládání binárních souborů, tabulek apod. Tomu odpovídají i obsažené příkazy zajišťující načítání uložených dat, kopírování, mazání či hlášení zbývajících volného místa v paměti.
11. **MMEMory** – kromě interní paměti je samozřejmě možné využít také velkokapacitní paměť (vnitřní nebo externí – např. USB disk). Tu pak ovládá subsystém MMEMory (*Mass Memory*), který má podobné funkce jako MEMory,

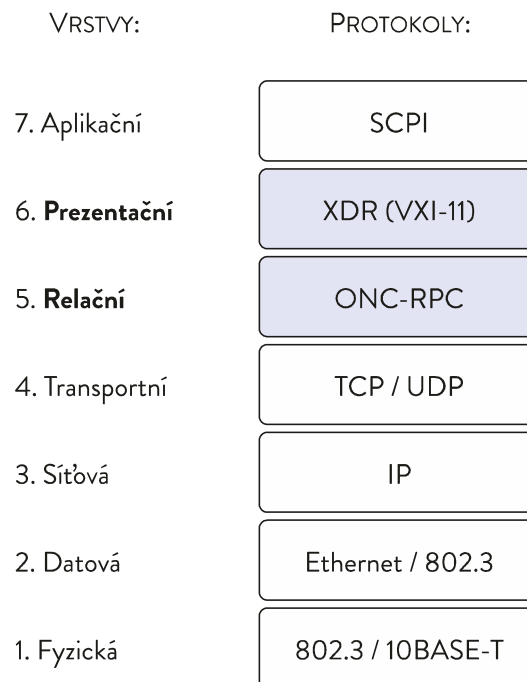
ale doplňuje je příkazy souvisejícími se správou souborů (především podpora klasické adresářové struktury ad.).

12. **OUTPut** – příkazy v tomto podsystému jsou analogické k bloku INPut, liší se jen směrem toku dat. Předtím než signál (vygenerovaný systémem SOURce) opustí přístroj, je možné ho právě pomocí OUTPut zeslabit, zesílit, filtrovat atd.
13. **PROGram** – podsystém určený k ovládání stažitelných programů uložených v paměti přístroje
14. **ROUTE** – viz kapitola 2.6.2
15. **SENSe** – viz kapitola 2.6.2
16. **SOURce** – viz kapitola 2.6.2
17. **STATus** – slouží pro obsluhu *Status Reporting*, rozlišení všech druhů možných stavů a událostí, které se mohou vyskytnout při činnosti měřicího zařízení.
18. **SYSTEM** – obsahuje příkazy, které mají spíše obecné provozní funkce – nápovědu, nastavení data a času apod.
19. **TEST** – sdružuje příkazy, kterými lze provádět autotestování přístroje
20. **TRACe** – důležitý subsystém, který poskytuje příkazy pro ovládání *Trace* – naměřeného průběhu. Poté, co je signál zpracován subsystémy INPut, SENSe a CALCulate, je zobrazen jako průběh na displeji. Z paměti přístroje ho pak lze stáhnout do řídicího počítače jako data, jejichž formát je nastaven pomocí FORMat. Další zpracování je pak již plně v rukou uživatele, čehož je využito v praktické části této DP – naprogramování ovládací aplikace pro vzdálené řízení.
21. **TRIGger** – pomocí tohoto podsystému je možné řídit generování či měření signálu tak, že jednotlivé činnosti přístroje jsou synchronizovány pomocí interní či externí události. Podmínkou pro spuštění akce může být například přítomnost signálu na vstupním portu, hrana signálu, případně daný počet pulzů na vstupu externího spouštěcího signálu aj.
22. **UNIT** – nastavení jednotek dané veličiny, např. u výkonu [W] či [dBm].
23. **VXI** – speciální skupina příkazů, která je přítomná pouze na přístrojích vybavených sběrnici VXI, a která slouží výhradně k zajištění funkcionality tohoto rozhraní.

3 Knihovna VISA

3.1 ISO/OSI model síťové komunikace

Po sestavení SCPI zprávy, při kterém byla dodržena všechna pravidla zevrubně popsaná v předchozích kapitolách, následuje další krok v komunikaci s přístrojem – je nutné příkazy nějakým způsobem předat z řídicího počítače (*controlleru*) cílovému zařízení, jemuž jsou určeny. Příkazy nelze posílat přímo do přístroje, neboť – jak již bylo v teorii SCPI uvedeno – jsou součástí vyššího programovacího jazyka, v němž jsou SCPI příkazy reprezentovány prostým textovým řetězcem. Pro popis přenosu SCPI do měřicího přístroje je vhodné se vrátit k vrstevnému modelu síťové komunikace, protože tuto problematiku nejnvýstižněji popisuje právě ISO/OSI model z teorie sítí [8].



Obr. 3.1 ISO/OSI model síťové komunikace v ATE systému [6]

Uvažujme ATE systém tvořený řídicím počítačem a měřicím zařízením, jejichž spojení je realizováno pomocí LAN. V ISO/OSI modelu pak jazyk SCPI (jako součást programovacího jazyka – např. Matlabu) patří do nejvyšší vrstvy – Aplikační, přičemž funkce vrstev 1 až 4 jsou zajištěny klasickými protokoly místní sítě, resp. Ethernetu. Při pohledu na vrstevný model (obr. 3.1) je zřejmé, že mezi Aplikační a Transportní vrstvou musí být na 5. a 6. vrstvě přítomny ještě protokoly, které „přeloží“ SCPI příkazy tak, aby mohly být přeneseny

sítí a přijaty přístrojem. Tyto protokoly (na obrázku zvýrazněny modře) se souhrnně nazývají VISA – *Virtual Instrument Software Architecture*. Protokoly TCP/IP pak s daty pracují běžným způsobem – rozdělení na segmenty, pakety, framy a nakonec bity umožní zprávu poslat přes přenosové médium (např. kroucenou dvojlinku Ethernetu), aby následně celý proces inverzně proběhl na straně přijímacího uzlu.

3.2 VISA a protokoly VXI-11

VISA se z pohledu informatiky řadí mezi *API (Application Programming Interface)*, to jest vstupně/výstupní rozhraní, které slouží pro programování aplikací. Představuje tedy knihovnu funkcí a protokolů, které jsou využívány pro komunikaci řídicího počítače se zařízením a naopak. Standardem VISA se řídí největší výrobci měřicí techniky, a vycházejí tak vstřícně hlavní myšlenky SCPI – zjednodušení vývoje softwaru kladením důrazu na univerzálnost a přenositelnost kódu. Knihovny nahané v přístrojích libovolné, běžně rozšířené značky (např. Rohde&Schwarz, Agilent Technologies či Tektronix) jsou díky tomu shodné, a tak kombinace obou norem SCPI a VISA významně zjednodušuje práci programátorům.

Výměnu zpráv mezi controllerem a přístrojem zajišťují protokoly standardu VXI-11, konkrétně se jedná o *ONC-RPC (Open Network Computing – Remote Procedure Calls)* na Relační vrstvě a *XDR (External Data Representation)* na Prezentační vrstvě [10]. Pozice těchto protokolů ve vrstevném modelu odpovídá jejich funkci – XDR je v kontaktu s Aplikační vrstvou a plní proto roli „překladače“, neboli přijímá SCPI zprávy z aplikace naprogramované pro dálkové řízení přístroje a přijatá data převádí do formátu nezávislého na programovacím jazyce. Samotný kód aplikace tak může být napsán např. v jazyce C, či – jako v případě této diplomové práce – v Matlabu, nicméně po zpracování protokolem XDR bude podoba SCPI zprávy již univerzální. V té chvíli může být předána protokolu ONC-RPC, který obstarává přenos zpráv; za tímto účelem spolupracuje se standardními TCP/IP protokoly sítě LAN. Z popsaných vlastností plyne nutnost využití knihovny VISA při vzdáleném ovládní měřicích zařízení, a proto je její instalace na řídicím počítači nutnou podmínkou fungující komunikace v ATE systému.

4 Podmínky pro vývoj a běh aplikace

4.1 Realizace síťového spojení

Prvním krokem, který předcházela vývoji samotné aplikace pro dálkové řízení, bylo rozhodnutí, jakým způsobem bude realizováno síťové spojení počítače, ve kterém budou aplikace poběží, a měřicího zařízení, které bude řídicím PC ovládáno. Moderní přístroje nabízejí řadu možností, ze kterých lze volit – od nejstarší sběrnice GPIB, přes praktický Ethernet až po současné USB. Z dostupných variant byla pro své výhodné vlastnosti zvolena technologie *Ethernet*, a to mimo jiné z těchto důvodů:

- vysoká přenosová rychlost,
- snadné nastavení parametrů spojení (IP adresy a adresy podsítí v počítači i přístroji se řídí standardními pravidly LAN),
- možnost ovládat zařízení v rámci místní sítě (odpadá nutnost přímého spojení PC a přístroje),
- takřka libovolná vzdálenost mezi přístrojem a řídicím počítačem (dosah je v podstatě omezen pouze rozlohou místní sítě),
- při zapojení přístroje v LAN možnost připojení několika PC najednou,
- jako přenosové médium je použita běžná kroucená dvojlinka,
- z hlediska programování praktická identifikace přístroje pomocí IP adresy, ad.

Spojení přístroje s počítačem má v takovém případě všechny rysy klasického sestavování místní sítě – měřicí zařízení se chová jako běžná stanice (ve většině přístrojů byl dokonce operační systém Windows) a jeho možnosti nastavení tomu odpovídají. Při nastavení IP adresy uzlu (a tím je jak přístroj, tak řídicí počítač) je možné zvolit buď automatické získání z DHCP serveru – tuto eventualitu uživatel využije, pokud je přístroj součástí místní sítě LAN, anebo ruční zadání statické IP adresy – to uživatel zvolí pro přímé spojení typu peer-to-peer. S IP adresou je pochopitelně úzce spjata maska podsítě, kterou je při manuální konfiguraci také nutné vyplnit. Jak vyplývá z pravidel podsítování, oba uzly se musí nacházet ve stejné podsíti a zároveň nesmí docházet ke konfliktu IP adres. Nezbytné nastavování těchto parametrů a s ním spojená byť základní znalost počítačových sítí mohou být vnímány jako drobná nevýhoda tohoto způsobu spojení, nicméně klady využití Ethernetu tento nedostatek výrazně převyšují. Samozřejmostí (i když snadno podceňitelnou) by měl být antivirem zabezpečený operační systém řídicího počítače – po sestavení LAN je přístroj v pozici běžné stanice v síti a jako takový může být potenciálně vystaven všem obvyklým hrozbám.

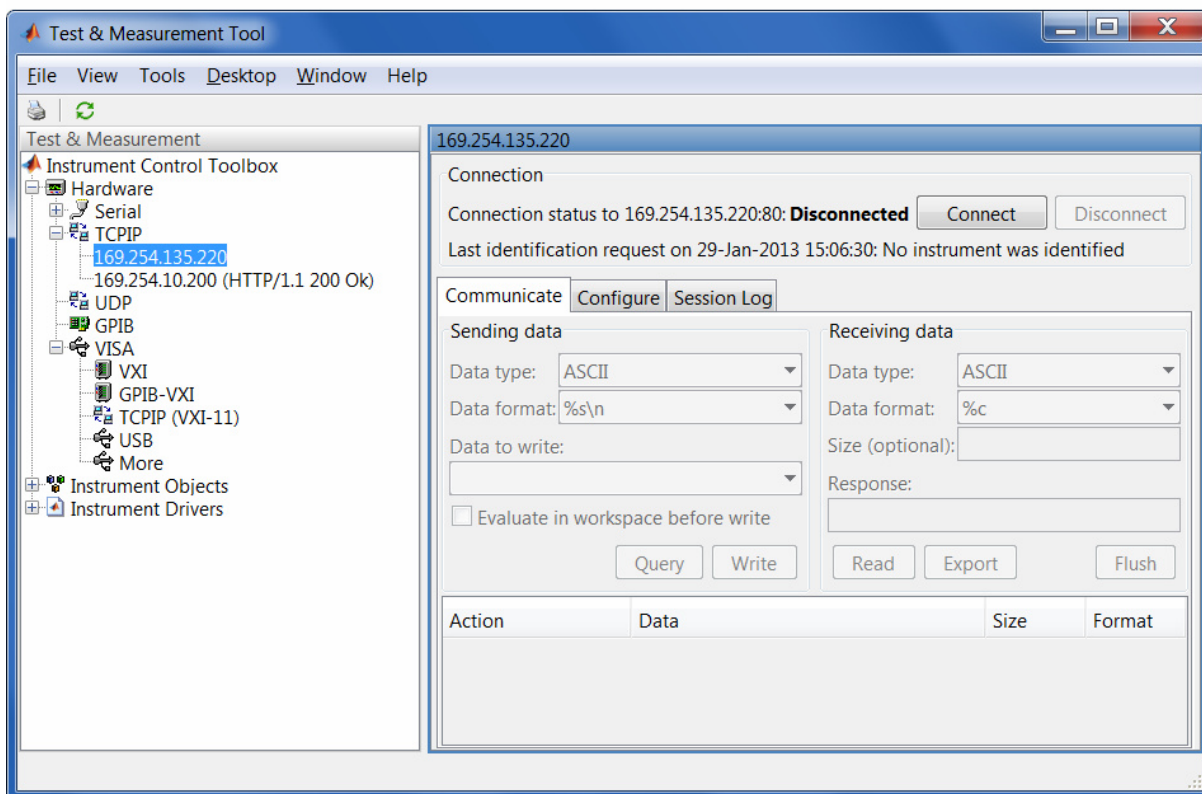
4.2 Vývojové prostředí Matlab

Aplikace pro vzdálené řízení přístrojů Rohde&Schwarz, která je výstupem této diplomové práce, byla naprogramována ve vývojovém prostředí MATLAB. To bylo zvoleno s ohledem na jeho charakteristické vlastnosti, kterými disponuje – předně je to prostředí zaměřené primárně na vědeckotechnické výpočty, a jako takové poskytuje velké množství implementovaných nástrojů a funkcí určených pro měření, zpracování a analýzu signálů. Z tohoto důvodu je velmi výhodné, aby průběhy naměřené přístroji byly již načítány aplikací běžící v Matlabu, neboť právě to umožní využití zmíněných funkcí (bez nutnosti eventuálního komplikovaného přenášení hodnot ze samostatné aplikace naprogramované např. v jazyce C).

Kromě výrazného usnadnění případného dalšího zpracování dat je třeba také zmínit uživatelsky přívětivý způsob objektivě orientovaného programování. Při vývoji aplikace bylo použito prostředí pro tvorbu programů s grafickým rozhraním (GUIDE), které uživateli intuitivně vychází vstříc do té míry, že v něm lze – v kombinaci s využitím funkcí z vestavěných knihoven – vytvořit základní grafickou aplikaci rychleji a přehledněji než v jazycích Java či C#. Možným argumentem proti použití prostředí Matlab by mohla být obtížná přenositelnost kódu do počítačů, v nichž není Matlab nainstalován. Narozdíl od jazyků C, Basic či Fortran, které jsou kompilované, a tedy snadno spustitelné v libovolném PC, je jazyk Matlabu *interpretovaný*, a programy v něm napsané tudíž ke svému běhu potřebují interpreta, který zdrojový kód počítači „tlumočí“. V oblasti odborných inženýrských výpočtů je ovšem Matlab natolik rozšířeným nástrojem, že problém nepřenositelnosti v praxi odpadá.

4.2.1 Instrument Control Toolbox

Význam knihoven funkcí – v Matlabu nazývaných *toolboxy* – byl již zmíněn. V souvislosti se strukturou samotného prostředí je možné je zjednodušeně považovat za složky obsahující soubory (*m-file*), ve kterých jsou jako skripty či funkce popsány všechny operace, které daný toolbox vykonává. Při volání těchto funkcí je nalezen soubor obsahující kód, který se následně vykoná. Tento princip je možné ilustrovat například u Image Processing Toolbox, což je knihovna funkcí, které mají na starosti široké spektrum úkonů souvisejících se zpracováním, analýzou či rekonstrukcí obrazu. Nejedná se již o prosté matematické funkce, nýbrž o složité procesy související s prací s obrazovými daty, které by bez instalace toolboxu nebylo možné provést.

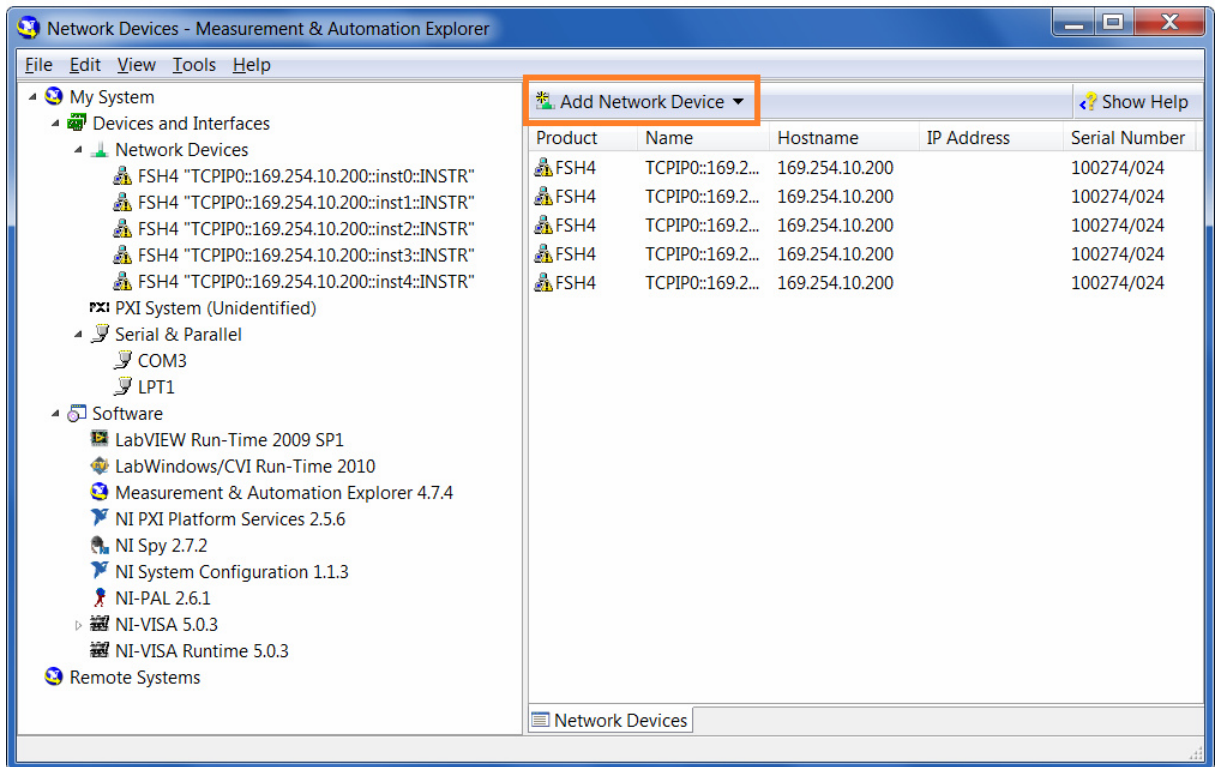


Obr. 4.1 Grafické rozhraní Instrument Control Toolboxu

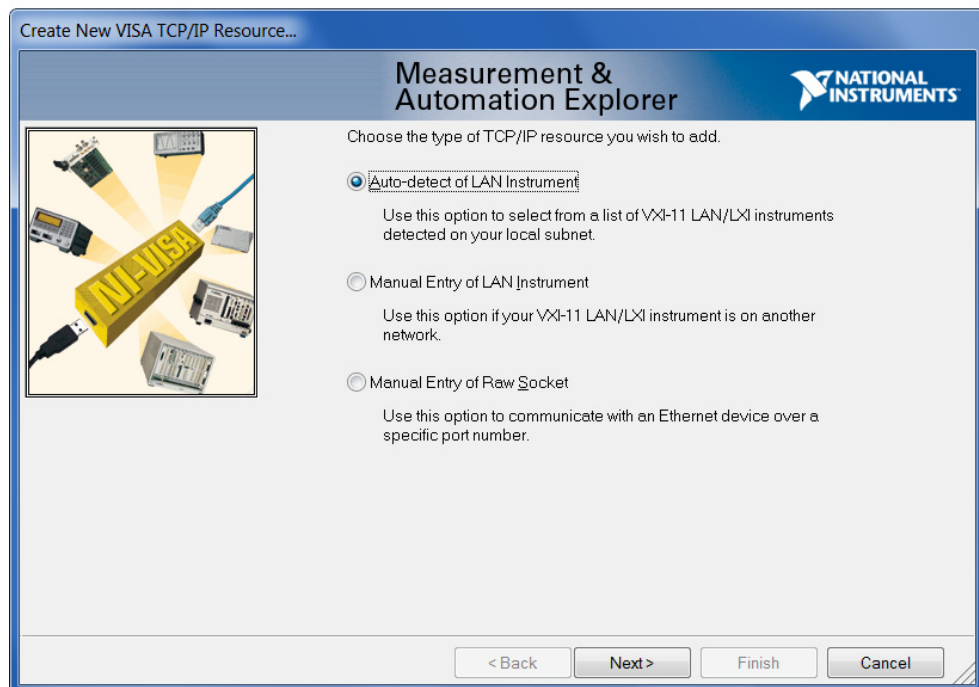
Pro komunikaci mezi PC a měřicím zařízením musí proto v Matlabu být nainstalován *Instrument Control Toolbox*, jehož funkce umožňují spravovat spojení pomocí knihovny VISA, a do připojeného přístroje je díky nim možné odesílat SCPI příkazy. Funkce, které jsou v podobě m-filů s tímto toolboxem nahrány do Matlabu, lze ovládat také z jednoduchého grafického rozhraní Test & Measurement Tool (obr. 4.1); jeho možnosti jsou však značně omezené. Pro účely této práce je navíc důležitá možnost algoritmizace úkonů tak, aby se prováděly automaticky bez potřeby uživatelského zásahu.

4.3 Komunikace pomocí VISA

Ke zprovoznění vzdáleného řízení chybí nyní poslední krok – nainstalovat knihovnu VISA, která je pro komunikaci mezi řídicím PC a přístrojem nepostradatelná (viz kapitola 3). Instalaci aktuální verze je možné stáhnout na stránkách společnosti National Instruments: www.ni.com/visa, v sekci *Downloads*. Spolu s ovladači se nainstaluje několik aplikací, z nichž asi nejdůležitější je *Measurement & Automation Explorer* (obr. 4.2) – grafické rozhraní pro připojení nového přístroje (obr. 4.3), které zároveň poskytuje přehled přístrojů již připojených.



Obr. 4.2 NI-VISA – Measurement & Automation Explorer



Obr. 4.3 Dialogové okno připojení nového přístroje

Stejně jako v případě toolboxu Matlabu, i tentokrát je tato aplikace vhodná spíše pro kontrolu spojení a řešení možných problémů s ním spjatých, konkrétní přínos pro tuto diplomovou práci ovšem nemá žádný.

To neplatí o knihovně VISA, ta naopak ve spolupráci s Instrument Control Toolboxem umožňuje Matlabu připojit měřicí zařízení a řídit ho pomocí SCPI příkazů. Pro komunikaci PC a přístroje musí být vytvořeno I/O rozhraní – to je představováno proměnnou třídy „objekt VISA“, která je zavedena po volání funkce `visa('NI',VISA_adresa)`. Písmena NI označují poskytovatele knihovny = National Instruments, VISA adresa je textový řetězec ve speciálním tvaru, jenž specifikuje způsob spojení (v našem případě přes TCP/IP), a jehož klíčovou část tvoří IP adresa připojovaného přístroje (řetězec může vypadat například takto: `'TCPIP0::169.254.10.200::inst0::INSTR'`). Pokud je v síti zařízení, které odpovídá této adrese, je ve *Workspace* Matlabu vytvořen objekt VISA. Přes tento objekt/rozhraní jsou pak odesílány SCPI typu příkaz (funkcí `fprintf`) a typu query (funkcí `query`), a to jako textové řetězce.

Pokud jsou provedeny všechny kroky popsané v kapitole 4 a po testu SCPI příkazem `'*IDN?'` vrátí přístroj jako odpověď svůj identifikátor, byl ATE systém úspěšně sestaven. Jsou tím splněny podmínky, které umožňují vývoj aplikace, a které je rovněž nutné dodržet pro použití této aplikace v praxi.

4.4 Přístrojové vybavení

Než přistoupíme k popisu aplikace pro dálkové ovládání, je vhodné se ještě zdržet u přístrojů, jimž je tento software určen. Jedná se o přístroje značky Rohde&Schwarz – moderní měřicí zařízení vybavená operačním systémem (většinou některá z verzí OS Windows) a komunikačním rozhraním Ethernet. Typově jsou zastoupeny spektrální analyzátory, obvodové analyzátory, osciloskopy, měřiče výkonu, signálové generátory a přístroje pro rádiový monitoring. Kromě stručného výpisu jejich funkcí je u každého zařízení uveden především návod, jak nastavit jeho parametry připojení k místní síti (tj. IP adresu a masku podsítě), které jsou nezbytné pro spojení s řídicím počítačem.

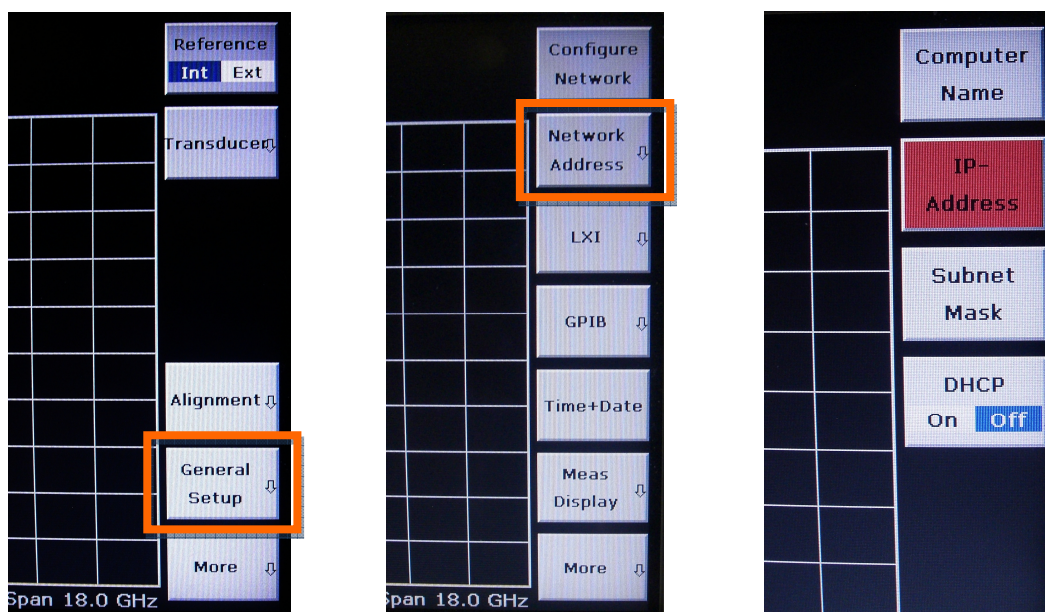
4.4.1 Spektrální analyzátor R&S® FSL



Obr. 4.4 R&S® FSL [11]

Hlavním úkolem spektrálních analyzátorů je měření a analýza signálu ve frekvenční oblasti, tj. měření výkonu spektra vstupního signálu – výsledkem je tudíž průběh závislosti amplitudového spektra na frekvenci. Typ FSL je stolní analyzátor vyráběný ve třech variantách lišících se zejména svým rozsahem: FSL3, FSL6 a FSL18 s rozsahy 9 kHz až 3, resp. 6 a 18 GHz [11]. Aplikace byla vyvíjena a testována na všech třech verzích tohoto přístroje. Vysokou přesnost tohoto přístroje, kterou výrobce deklaruje v dokumentaci, dokládá možnost nastavení až 32 001 vzorků snímaného signálu.

Nastavení parametrů LAN



Obr. 4.5 FSL – nastavení parametrů místní sítě

Pro komunikaci s PC pomocí LAN je třeba nastavit odpovídající síťové parametry, přičemž dostupné jsou obě běžné varianty, tj. a) přidělení IP adresy DHCP serverem, b) zadání statické IP adresy. S ohledem na pravidla podsítování je nutné v případě statické IP adresy zkontrolovat, zda jsou přístroj i řídicí počítač ve stejné podsíti (definované její maskou). Postup pro nastavení přístroje je následující:

1. Tlačítkem „SETUP“, nacházejícím se nalevo od displeje na předním panelu, vyvoláme menu pro nastavení přístroje.
2. V pravé části displeje se zobrazí nabídka možností, zvolíme „General Setup“.
3. Další volbou je položka „Network Address“.
4. Nyní jsou konečně dispozici klíčové položky:
 - „DHCP On/Off“ pro nastavení dynamického/statického přidělování IP adresy,
 - „IP Address“ pro nastavení IP adresy,
 - „Subnet Mask“ pro nastavení masky podsítě.

4.4.2 Obvodový analyzátor R&S® ZVL



Obr. 4.6 R&S® ZVL [11]

Vektorový obvodový analyzátor nabízí oproti spektrálnímu analyzátoru několik výhod – neměří totiž pouze skalární hodnotu amplitudy, ale snímá zvláště také vlastnosti fáze. Data signálu jsou proto v komplexním formátu, což umožňuje vykreslovat průběh v polárním grafu či Smithově diagramu, případně zobrazit průběh fáze nebo skupinového zpoždění. ZVL navíc na měřený obvod nahlíží jako na jednobran, resp. dvojbran; nabízí proto dva vstupní porty, které je možné využít pro měření rozptylových s-parametrů. Obvodový analyzátor ZVL je

také vyráběn ve třech modelech: ZVL3, ZVL6 a ZVL13 s rozsahy 9 kHz až 3, resp. 6 a 13,6 GHz [11].

Nastavení parametrů LAN

Aby bylo možné ovládat ZVL vzdáleně přes LAN, musí být opět nastaveny vhodné vlastnosti připojení k místní síti. Postup nastavení síťových parametrů je shodný s FSL (viz obr. 4.5):

1. Tlačítkem „SETUP“, nacházejícím se nalevo od displeje na předním panelu, vyvoláme menu pro nastavení přístroje.
2. V pravé části displeje se zobrazí nabídka možností, zvolíme „General Setup“.
3. Další volbou je položka „Network Address“.
4. Nyní jsou konečně dispozici klíčové položky:
 - „DHCP On/Off“ pro nastavení dynamického/statického přidělování IP adresy,
 - „IP Address“ pro nastavení IP adresy,
 - „Subnet Mask“ pro nastavení masky podsítě.

4.4.3 Spektrální analyzátor R&S® FSH



Obr. 4.7 R&S® FSH a sonda R&S® FSH-Z1 [11]

Spektrální analyzátor FSH v sobě spojuje schopnosti několika přístrojů – v závislosti na zakoupených licencích nabízí kromě spektrálního a obvodového analyzátoru například také měření výkonu. Při něm musí být FSH doplněn o sondu *Power Sensor* (viz obr. 4.7). Jeho předností je navíc přenosnost – svými kompaktními rozměry a akumulátorem vychází vstříc

případnému použití v terénu. Pro ty příležitosti, kdy není možné průběh stáhnout a zpracovat rovnou v PC, nabízí také jednoduché ukládání naměřených dat na SD kartu či USB disk.

Nastavení parametrů LAN



Obr. 4.8 FSH – nastavení parametrů místní sítě

I pro FSH platí, že přesné nastavení místní sítě je nutnou podmínkou pro komunikaci přes Ethernet. Postup je u tohoto typu přístroje odlišný od předchozích:

1. Tlačítkem „SETUP“, nacházejícím se na pravém kraji, vyvoláme menu pro nastavení přístroje.
2. Klávesou „F2“ zvolíme možnost „Instrument Setup“ (viz obr. 4.8).
3. Šipkou dolů nebo kolečkem vybereme požadovanou položku:
 - „DHCP“ pro nastavení dynamického/statického přidělování IP adresy,
 - „IP Address“ pro nastavení IP adresy,
 - „Subnet Mask“ pro nastavení masky podsítě.

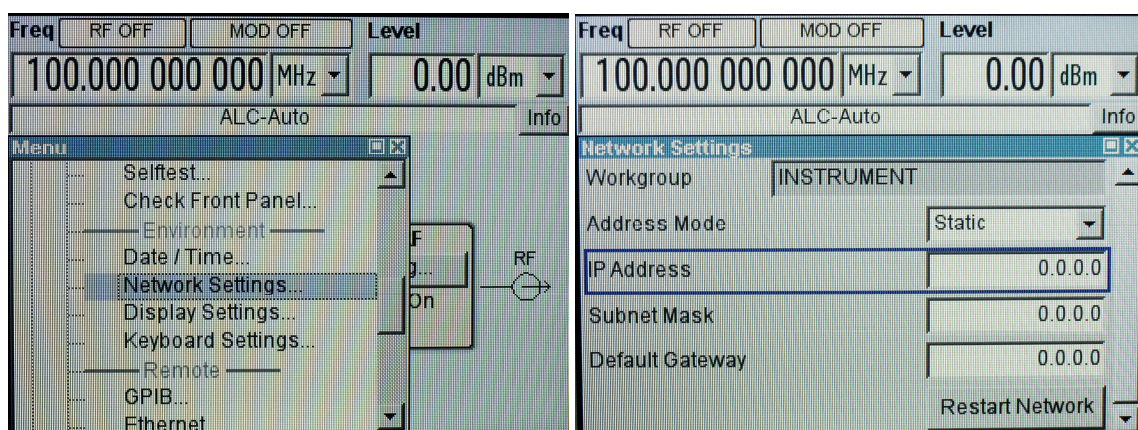
4.4.4 Signálové generátory R&S® SMx



Obr. 4.9 R&S®SMB100A (vlevo), SMC100A (vpravo) a SMBV100A (dole) [11]

Signálové generátory řady SM mohou sloužit jako zdroj mikrovlnného signálu (SMB 100A je vyráběno v modelech s rozsahem až od 40 GHz), umí pracovat jako vektorový signálový generátor (SMBV 100A) a poskytují možnost modulace signálu amplitudově, frekvenčně či fázově (a v případě SMC 100A navíc ještě modulací pulzní).

Nastavení parametrů LAN



Obr. 4.10 SMx – nastavení parametrů místní sítě

Způsob nastavení místní sítě je pro všechny tři typy generátorů stejný:

1. Tlačítkem „SETUP“, nacházejícím se nalevo od displeje na předním panelu, vyvoláme menu pro nastavení přístroje.
2. V zobrazeném menu je oddělení „Environment“ a v něm položka „Network Settings...“ – po potvrzení se otevře dialogové okno síťového nastavení.
3. Možností „Address Mode“ lze nastavit:
 - „Auto (DHCP)“ = IP adresa a maska podsítě budou přiděleny dynamicky DHCP serverem,
 - „Static“ = zpřístupní kolonky „IP Address“ a „Subnet Mask“, ve kterých můžeme nastavit IP adresu a masku podsítě ručně.

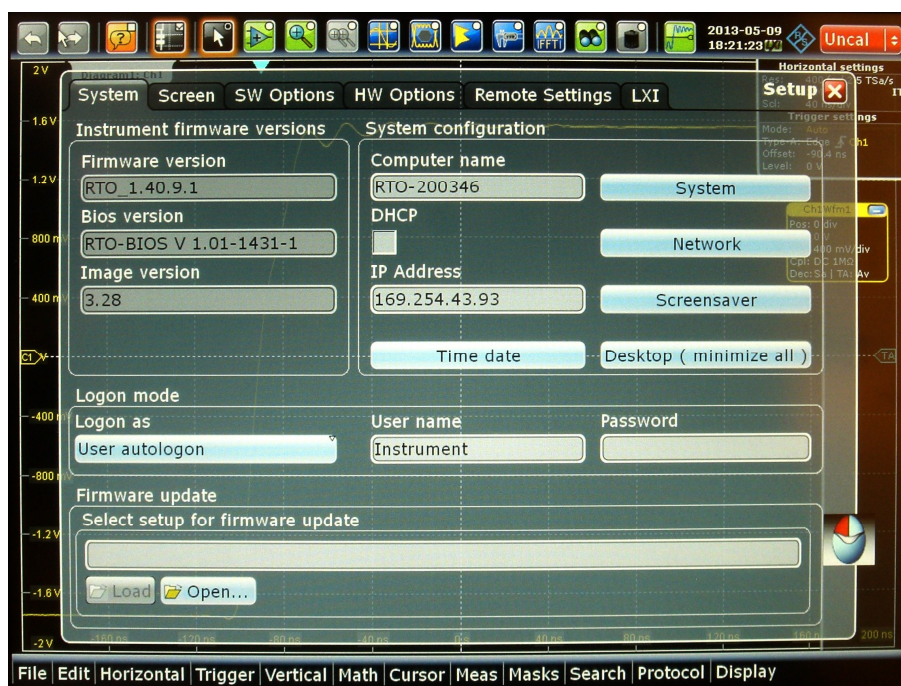
4.4.5 Osciloskop R&S® RTO



Obr. 4.11 R&S® RTO [11]

Osciloskop RTO 1014 je moderní zařízení vybavené výkonným procesorem, který mu umožňuje zobrazovat (a přenášet do PC) signál s velkým rozlišením – a to díky vysoké vzorkovací frekvenci. Pro velmi přesné měření signálu může osciloskop vykreslit průběh pomocí až 40 milionů vzorků – ty lze využitím aplikace z této diplomové práce následně stáhnout do Matlabu. Nadto je přístroj vybaven velkým dotykovým displejem (s možností připojit přes USB port myš) – manipulace s tímto osciloskopem je tak velmi komfortní. Model č. 1014, na němž byla aplikace vyvíjena, je vybaven čtyřmi vstupními kanály a měří na šířce pásma 1 GHz.

Nastavení parametrů LAN



Obr. 4.12 RTO – nastavení parametrů místní sítě

Způsob nastavení místní sítě je pro všechny tři typy stejný:

1. Tlačítkem „SETUP“, nacházejícím se nalevo od dotykového displeje na předním panelu, vyvoláme menu pro nastavení přístroje.
2. Vybereme záložku „System“.
3. Zde (viz obr. 4.12) nalezneme zaškrťovací pole „DHCP“ a nastavení IP adresy a masky podsítě (tlačítkem „Network“ je možné přístroj přepnout do prostředí Windows a postup nastavení místní sítě je pak analogický s PC).

Nastavení rozlišení (počet vzorků staženého průběhu)

Počet vzorků, které budou tvořit data signálu po stažení průběhu do PC, lze v RTO1014 jednoduše nastavit:

1. Na dolní liště hlavní obrazovky klikneme na možnost „Horizontal“; v rozbalené nabídce vybereme položku „Resolution“.
2. Druhou možností je poklepnutí na informační okénko nadepsané „Horizontal settings“ v pravém horním rohu displeje.
3. Zobrazí se okno nastavení rozlišení: v něm jsou na ukázce vysvětleny možnosti, jak lze rozlišení změnit. Výsledný počet vzorků je uveden v poli „Record length“.

5 Popis aplikace

Výsledkem této diplomové práce je aplikace pro vzdálené řízení vybraných přístrojů značky Rohde & Schwarz; v případě běžného použití nahrazující komplikovanou manipulaci s tlačítky a malým displejem těchto zařízení. Všechny přístroje byly při vývoji aplikace k dispozici, díky čemuž mohl být ovládací software *průběžně testován* – tím bylo umožněno vyladění funkcí aplikace „na míru“ jejímu budoucímu uplatnění v praxi. Jedním z cílů při návrhu aplikace byla minimalizace eventuálních komplikací plynoucích z jejího zprovoznění na PC – z tohoto důvodu bylo také zvoleno přímé použití SCPI příkazů jako spolehlivého způsobu programování, který s sebou nese významnou výhodu: *nevyžaduje instalaci ovladačů* každého přístroje zvlášť. Přesto existuje několik podmínek, které musí být pro úspěšné spuštění aplikace splněny.

1. **Sít'ové připojení Ethernet** – pro svou rychlost a rozšířenost bylo jako komunikační médium využito rozhraní Ethernet; spojení mezi řídicím počítačem a měřicím zařízením je proto realizováno jako LAN.
2. **MATLAB** – nutnou podmínkou pro provoz aplikace je také přítomnost výpočetního systému Matlab (doporučena je verze 2012 a výše), přičemž instalace musí obsahovat **Instrument Control Toolbox**, jehož příkazy jsou klíčovou součástí zdrojového kódu.
3. **Knihovna VISA** – v řídicím PC musí být rovněž přítomny ovladače „překládající“ měřicímu přístroji textové řetězce z Matlabu do srozumitelné podoby SCPI příkazů. Instalaci VISA je možné stáhnout na stránkách www.ni.com/visa, v sekci „Downloads“.

LAN poskytuje navíc teoretickou možnost nepřímého spojení PC a přístroje – oba uzly mohou být připojeny do místní sítě a DHCP serverem jim může být přidělena dynamická IP adresa. Řídicím počítačem by tak bylo možné ovládat přístroj z libovolné fyzické vzdálenosti – dosah by byl omezen pouze rozlohou místní sítě. Z praktického hlediska je ovšem jednodušší zbavit se nutnosti vypořádat se s překážkami síťového zabezpečení a je doporučeno spojení typu *peer-to-peer* pomocí *statické IP adresy a masky podsítě*.





5.1 Společné prvky v GUI

Pro každý přístroj, který je pomocí této aplikace ovládán, bylo navrženo samostatné GUI, jež pokrývá specifické požadavky a možnosti daného zařízení. Liší se tedy nejen funkčně z pohledu zdrojového kódu, ale i v grafické podobě, která musí těmto funkcím odpovídat. Přesto je možné v aplikaci nalézt společné prvky, které mají naopak vždy stejnou funkci a podobu – uživatel se tak snadno zorientuje a práce s aplikací tím usnadní. Následující objekty se vyskytují ve většině GUI – aby se tedy předešlo redundantnímu, opakujícímu se popisu v dalších kapitolách, jsou zde nejprve uvedeny samostatně.

5.1.1 Tlačítka s grafikou

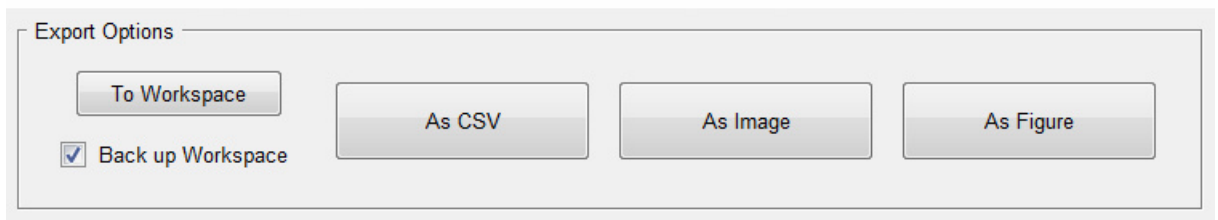
V každém GUI je množství tlačítek zastupujících buď přímo funkce a nastavení přístroje, anebo provádějící speciální operace v rámci zpracování stažených dat. Jednu skupinu tlačítek je však možné nalézt v ovládacím okně každého přístroje. Vzhledem k jejich opakovanému výskytu jsou pro zjednodušení použity piktogramy namísto popisku tlačítka. Tabulka 5.1 uvádí podobu těchto tlačítek a jejich funkce – jedná se především o „provozní“ úkony samotné aplikace doplněné tlačítkem *Preset*. To je fyzicky přítomné na každém zařízení a vrací jeho parametry na výchozí hodnoty dané výrobcem. Po stisknutí v GUI je odeslán SCPI příkaz, který má stejný účinek jako zmáčknutí tlačítka na panelu přístroje.

Tabulka 5.1 Přehled společných tlačítek v GUI

Tlačítko	Funkce
	Nové připojení = ukončí spojení s přístrojem, zavře aktuální GUI, a zavolá zpět úvodní okno pro připojení.
	Informace o GUI = otevře soubor PDF s popisem aplikace a jejích funkcí.
	Ukončit aplikaci = ukončí spojení s přístrojem, ukončí celou aplikaci zavřením aktuálního GUI i úvodního okna.
	Výchozí nastavení = má stejnou funkci jako fyzické tlačítko na přístroji – nastaví parametry na výchozí hodnoty.

5.1.2 Možnosti exportu

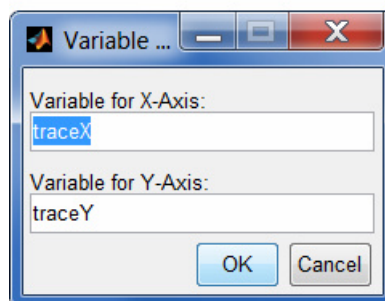
Klíčovou součástí aplikace je funkce exportu dat představujících průběh naměřený přístrojem. Panel *Export Options* lze tudíž nalézt v každém GUI, jehož cílem je stažení a vykreslení měřeného signálu. Na obr. 5.1 je zobrazen příklad takového panelu následovaný popisem činnosti tlačítek.



Obr. 5.1 Skupina tlačítek pro export dat

Workspace

V zadání diplomové práce byl požadavek zejména na export do Workspace kvůli možnosti dalšího zpracování stažených dat. Po stisknutí tlačítka *To Workspace* je uživatel vyzván k uvedení názvu proměnných, do kterých budou data exportována.



Obr. 5.2 Zadání názvu proměnných ve Workspace

Zaškrtnuté pole *Back up Workspace* řídí automatickou zálohu proměnných, které jsou kopírovány do Workspace. Tento proces slouží jako ochrana ztráty dat při eventuálním pádu Matlabu. Pro zálohy se vytváří samostatná složka v kořenovém adresáři aplikace (ve složce, kde je umístěn mainGUI.m). V názvu složky je uveden typ přístroje a datum měření pro snadnou identifikaci; ve složce jsou soubory typu *.mat obsahující jednotlivé proměnné zálohovaných Workspace.

CSV

Doplněna byla volba *As CSV* pro uložení do souboru ve formátu *.csv – vhodné např. pro přenesení naměřených dat do tabulkového procesoru Excel.

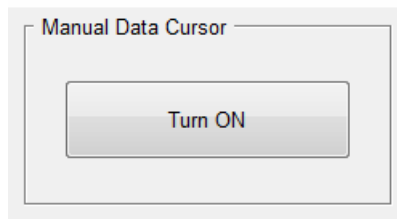
Obrázek

Tlačítko *As Image* zavolá funkci, která uživateli umožní sejmout screenshot buď celého GUI, nebo vybraného „výstřižku“ okna.

Samostatné okno „Figure“

Poslední možnost *As Figure* nabízí vykreslení staženého průběhu v samostatném okně. Uživatel si pak může graf buď zcela libovolně přizpůsobit svým potřebám, či jej rovnou uložit v některém z mnoha dostupných formátů.

5.1.3 Manual Data Cursor



Obr. 5.3 Tlačítko pro zapnutí funkce Data Cursor

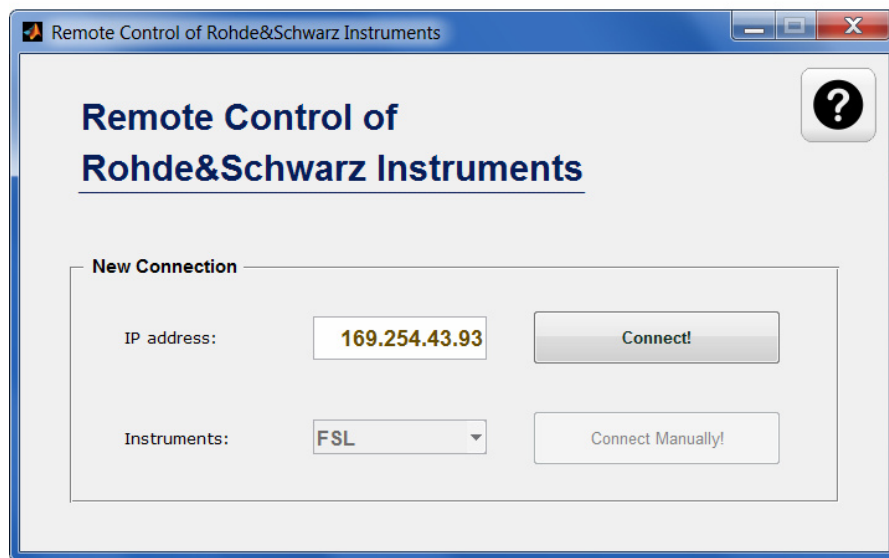
Kromě markerů, které budou popsány v dalších kapitolách, lze hodnoty z grafu vyčítat také zapnutím funkce `datacursormode`. Tlačítko pro její spuštění (viz obr. 5.3) je přítomné v každém GUI, který obsahuje souřadnicový systém pro zobrazení průběhu měřeného signálu.

5.1.4 Zavření GUI vs. ukončení aplikace

Další společnou vlastností všech GUI (kromě úvodního pro nové připojení) je rozdílný způsob zavření aktuálního okna a ukončení celé aplikace. V každém GUI je naprogramována reakce na stisk klávesy *Escape* – v takovém případě je zavřeno pouze momentálně běžící okno, spojení s přístrojem je ukončeno a je zavoláno zpět úvodní okno pro připojení nového zařízení. Naproti tomu kliknutí myši na *křížek* ukončí celou aplikaci včetně hlavního okna a v něm uložených proměnných.

5.2 Nové připojení - mainGUI

Aplikace *Remote Control of Rohde & Schwarz Instruments* se skládá z jednotlivých GUI – každé příslušející jednomu konkrétnímu přístroji. Centralizovaný přístup k těmto ovládacím rozhraním přitom zprostředkuje okno nového připojení. Tomuto způsobu odpovídá i struktura samotných souborů se zdrojovým kódem – aplikace se spouští pouze souborem *mainGUI.m* umístěným v kořenovém adresáři; ostatní soubory typu *.m a *.fig obsluhující další okna, samostatné funkce apod. jsou umístěny ve zvláštní složce *files*, která je při otevření *mainGUI.m* automaticky přidána do vyhledávacích cest Matlabu.

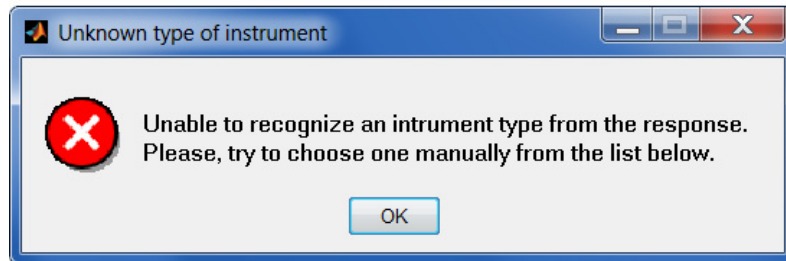


Obr. 5.4 Úvodní okno aplikace s nabídkou nového připojení

GUI pro nové připojení je vlastně jednoduchou „branou“ obsahující pouze položky, které umožní spojení s měřicím zařízením pomocí LAN. V kolonce *IP address* uživatel zadá IP adresu³ přístroje, ke kterému se chce připojit (návody na nastavení/zjištění této adresy u daných zařízení jsou uvedeny v kapitole 4.4), odeslání adresy pak proběhne po stisku tlačítka *Connect!*. V případě úspěšného připojení se aplikace sama pokusí identifikovat typ přístroje, a nabídne potvrzení automatického spuštění odpovídajícího GUI. Uživateli může být nicméně nabídnuta možnost ručně z rozbalovacího menu vybrat, které rozhraní chce spustit – určení typu přístroje je totiž výsledkem rozboru textového řetězce identifikátoru vráceného

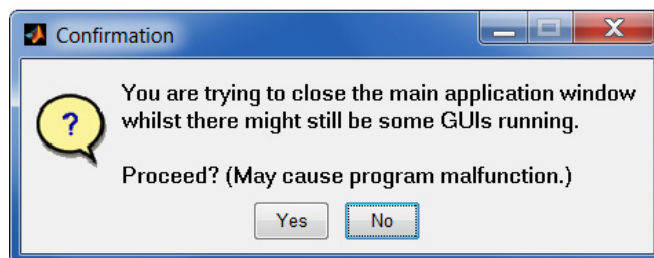
³ Zde je vhodné opět připomenout pravidla počítačového síťování: obecně všechny uzly v síti, které spolu chtějí komunikovat (a tedy i řídicí počítač a přístroj Rohde & Schwarz), musí být umístěny ve *stejně podsíti* (tzn. mít nastavenou stejnou masku podsítě), přičemž jejich IP adresy se musí v rámci této podsítě lišit. Nejjednodušší a obvyklý způsob, jak toho dosáhnout, je nastavení *rozdílu IP adres o 1* v posledním oktetu.

zařízením na dotaz '**IDN?*', přičemž tento řetězec je možné v přístroji změnit. V takovém případě by pak aplikace zařízení nerozpoznala (viz obr. 5.5), ovšem po manuálním výběru GUI by spuštěné rozhraní bylo samozřejmě plně funkční.



Obr. 5.5 Hláška při nerozpoznání typu připojeného přístroje

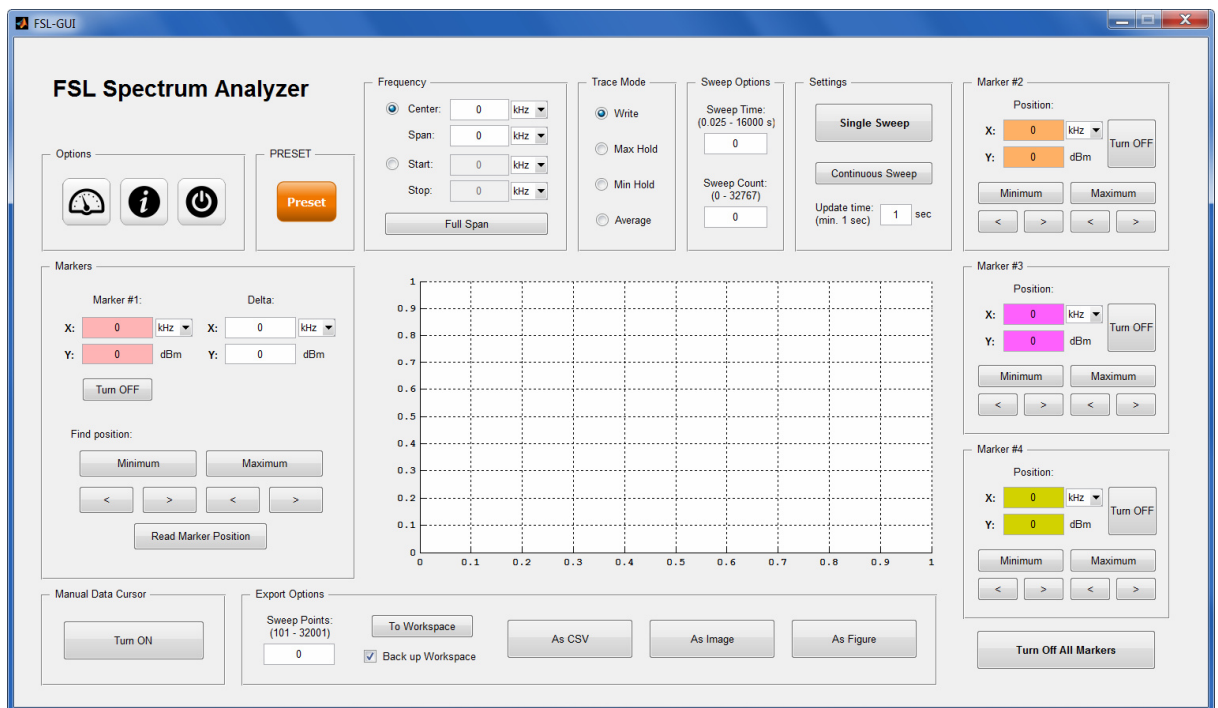
Okno mainGUI zůstane po připojení otevřené, aby se k němu uživatel mohl vrátit, pokud se ve spuštěném GUI rozhodne pro nové připojení. MainGUI zároveň slouží pro ukládání proměnných, které jsou sdíleny skrze jednotlivá grafická rozhraní (objekt VISA umožňující komunikaci s přístrojem, řídicí a pomocné proměnné apod.) – nemělo by proto být zavřeno dříve, než ostatní GUI. S touto situací počítá funkce naprogramovaná tak, aby uživatele varovala, že zavření okna způsobí chybu programu (obr. 5.6).



Obr. 5.6 Varování při pokusu o zavření mainGUI

Drobností, kterou se aplikace snaží vyjít vstříc komfortu uživatele, je ukládání poslední použité IP adresy do souboru *lastIP* v adresáři „files“. Při spuštění aplikace je tento záznam nahrán do pole „IP address“. V pravém horním rohu okna je také tlačítko pro nápovědu – po jeho stisknutí se v internetovém prohlížeči zobrazí upravená kapitola 4 této diplomové práce (převedená do formátu PDF), která by měla uživatele v případě potíží navést správným směrem.

5.3 FSL-GUI



Obr. 5.7 FSL-GUI

Primárním cílem grafického rozhraní pro ovládání přístroje R&S[®]FSL je stažení a vykreslení průběhu naměřeného signálu s možností exportu těchto dat do *Workspace* Matlabu (a několika dalšími variantami). V okně tohoto GUI můžeme poznat prvky z kapitoly 5.1 věnované společným funkcím všech GUI, jejich popis je tedy možné přeskočit. Ústředním prvkem okna je souřadnicový systém, v němž je vykreslováno **spektrum měřeného signálu** a tlačítka pro export nacházející se pod ním. Aktualizace grafu se provádí pomocí tlačítka *Single Sweep* pro jednorázové sejmутí průběhu, případně *Continuous Sweep* pro opakované vykreslení se zadanou periodou obnovování. Způsob snímání signálu je možné změnit v nabídce přepínačů *Trace Mode*: normální, zaznamenávání pouze maximálních či minimálních hodnot signálu, a zobrazení průměrné hodnoty.

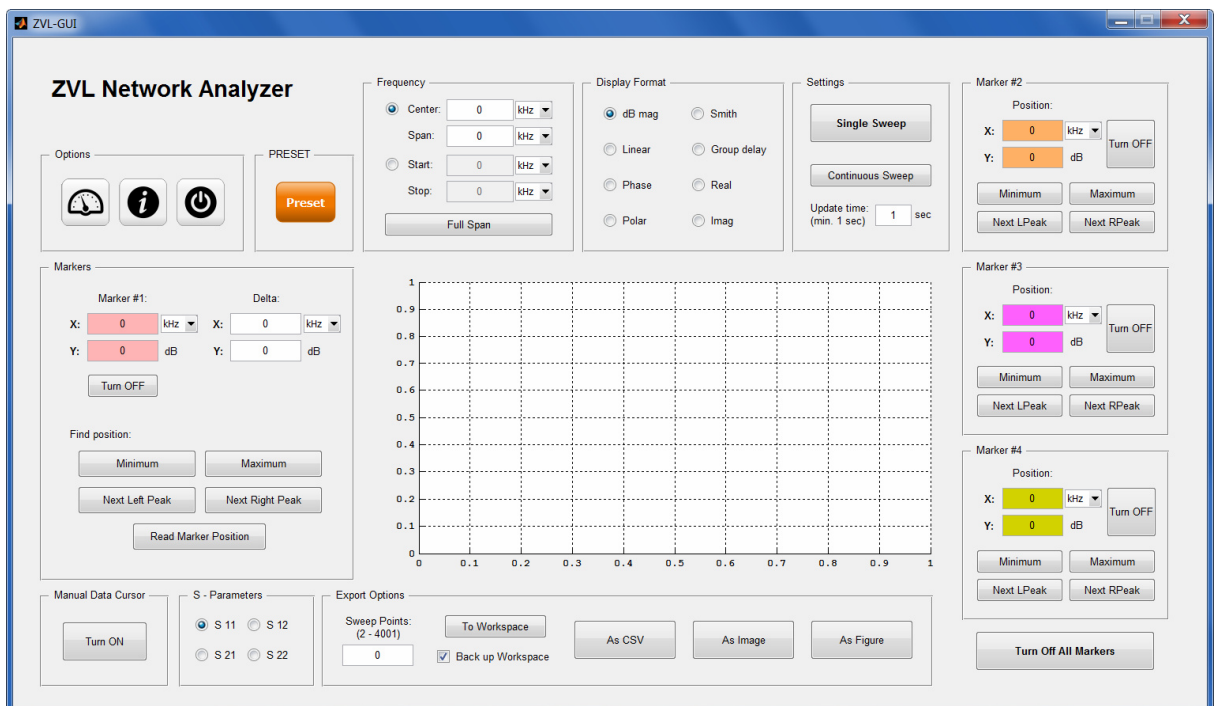
Funkcí, která se zaznamenaným průběhem úzce souvisí, jsou **markery** – práce při vyčítání hodnot je díky ovládání z PC rychlejší a uživatelský přívětivá. K dispozici jsou celkem 4 markery, které jsou kvůli snadné manipulaci barevně rozlišeny. Pozici markeru je možné zadat buď přímo pomocí souřadnice „X“ = frekvence, anebo lze využít funkce vestavěné v přístroji – operace nalezení *Minima* a *Maxima* jsou doplněny ještě hledáním *minim* a *maxim* nalevo či napravo od aktuální pozice markeru (tlačítka „<“ a „>“). Pokud je zapnut marker č. 1 a zároveň také marker č. 2, je navíc automaticky dopočítáván rozdíl mezi

jejich souřadnicemi X (frekvence) a Y (výkon v dBm) – výsledné hodnoty jsou uvedeny v polích *Delta*. U markeru č. 1 je ještě doplněna funkce *Read Marker Position* – tímto tlačítkem lze vyčíst pozici markeru ručně nastaveného na přístroji.

Aplikace „Remote Control of Rohde&Schwarz Instruments“ se ovšem snaží také usnadnit práci s přístrojem tak, aby nastavení nejdůležitějších parametrů nemusel uživatel provádět tlačítky na malém displeji přístroje, nýbrž pohodlně myší a klávesnicí na obrazovce monitoru. K dispozici je tedy možnost nastavení analyzované **frekvence**, a to dvěma způsoby: a) určením centrální frekvence + rozsahu, b) zadáním počáteční a koncové frekvence. K zobrazení celého rozsahu, na němž je FSL schopen snímat, slouží tlačítko *Full Span*.

Uživatel také může nastavit **vlastnosti snímání** signálu – pomocí *Sweep Time* lze určit dobu snímání, *Sweep Count* předepisuje počet opakování měření jednoho průběhu. Důležitým parametrem je však především *Sweep Points*, neboť ten udává počet vzorků, jimiž bude signál zaznamenán (a uložen v PC) – a tím tedy i přesnost měření.

5.4 ZVL-GUI

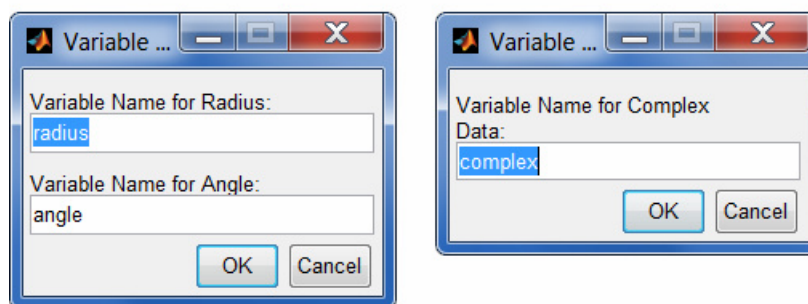


Obr. 5.8 ZVL-GUI

Grafické rozhraní pro ovládání obvodového analyzátoru R&S®ZVL se snaží respektovat unifikovanou podobu GUI této aplikace. Jednotný vzhled a možnosti programu tam, kde je to možné, pomáhají uživateli s rychlou orientací v okně při přechodu mezi

různými typy měřicích zařízení. S ohledem na rozdílnou funkcionalitu jednotlivých přístrojů bylo ovšem vždy nutné doplnit či změnit některé prvky daného GUI.

Oproti předešlému rozhraní je ZVL-GUI obsluhován jiným zdrojovým kódem, který grafickému oknu dává život. Odlišné nebyly pouze SCPI příkazy, kterými se obvodový analyzátor řídí, ale také přístup ke staženým datům při jejich zpracování. V popisu přístrojového vybavení (kapitola 4.4) byl zmíněn rozdíl mezi skalárním (FSL) a vektorovým měřením (ZVL), při kterém je odděleně zaznamenána amplituda a fáze signálu. Ze stažených dat bylo tedy nutno sestavit komplexní čísla a s těmi v kódu dále pracovat. Tento způsob práce s daty se projeví také při ukládání proměnných do *Workspace* – průběh polárního či Smithova diagramu není exportován jako vzorky osy X a osy Y, ale jako poloměr a fáze, resp. v komplexních číslech.



Obr. 5.9 Export proměnných do *Workspace*

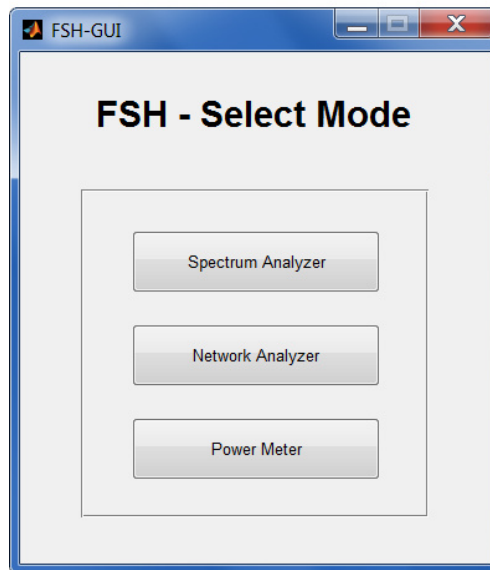
Tím se dostáváme k první významné odlišnosti od FSL-GUI – k panelu *Display Format*. Ten nabízí sadu přepínačů, jimiž lze určit **formát zobrazeného průběhu**, přičemž možnosti výběru odpovídají právě komplexní povaze měření: výkonové spektrum v [dB] a v [V], fáze, již zmíněný polární a Smithův diagram, skupinové zpoždění, a také reálná a imaginární složka.

Díky tomu, že ZVL nahlíží na měřený obvod jako na dvojbran (nabízí proto dva vstupní porty), je možné jej využít pro měření rozptylových **s-parametrů**. V panelu *S – Parameters* jsou k dispozici příslušné přepínače pro jednotlivé odrazové a přenosové koeficienty.

Systém **markerů** je shodný s FSL-GUI, jiná je však funkce automatického hledání pozice – hledání minim a maxim nalevo či napravo od aktuální pozice markeru musí na ZVL předcházet nejprve stisk odpovídajícího tlačítka *Minimum*, resp. *Maximum*. Teprve potom lze marker posouvat doleva (tlačítko *Next Left Peak*) či doprava (tlačítko *Next Right Peak*) po zvoleném druhu extrému.

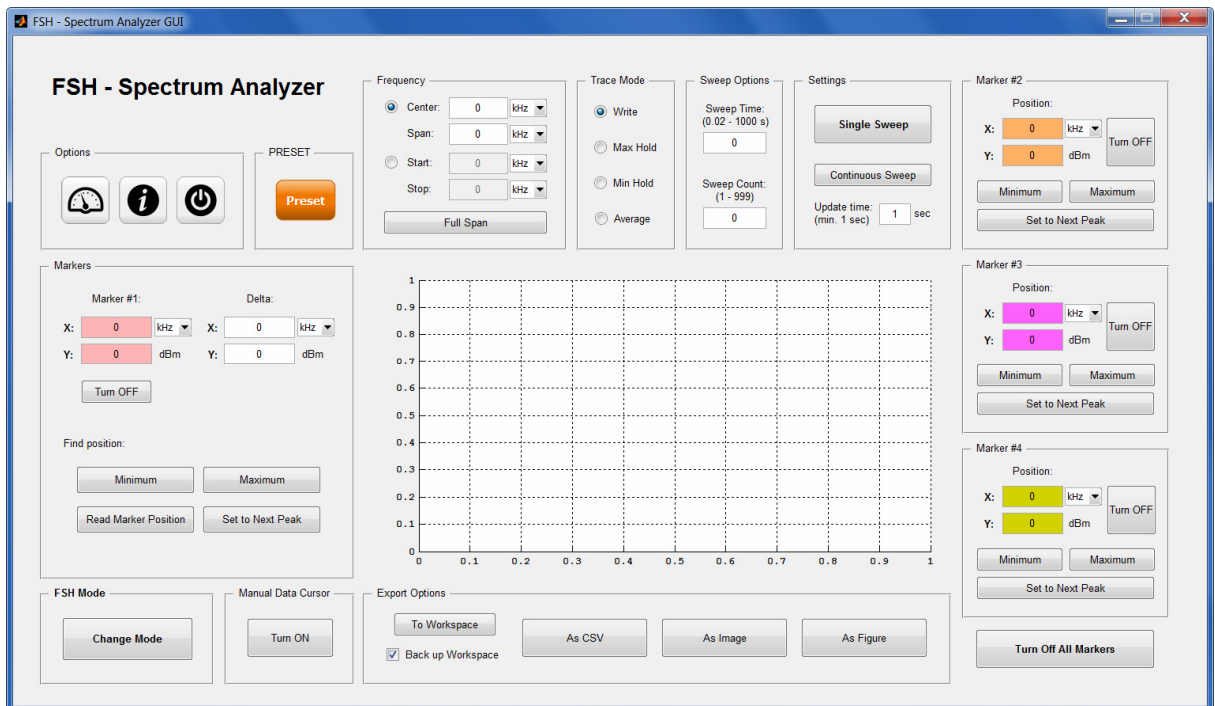
5.5 FSH-GUI

Předtím, než je možné otevřít GUI pro R&S®FSH, musí být uživatel vyzván k volbě módu (obr. 5.10), ve kterém chce analyzátor spustit – FSH totiž nabízí několik režimů měření; v této aplikaci je naprogramována obsluha spektrálního a síťového analyzátoru, a měřiče výkonu. Teprve po stisku některého tlačítka je otevřeno příslušné ovládací okno, a v případě potřeby se také do daného módu přepne samo měřicí zařízení.



Obr. 5.10 FSH-GUI = volba módu přístroje

5.5.1 Spectrum Analyzer GUI



Obr. 5.11 FSH – Spectrum Analyzer GUI

V režimu spektrálního analyzátoru je spuštěno GUI, které opět dodržuje pravidla jednotného vzhled a funkcí. Vykreslení průběhu do souřadnicového systému je znovu možné pomocí tlačítka *Single Sweep* a rovněž použití *Markerů* je tradiční. Rozdílné je automatické nastavení pozice markeru – tlačítko *Set to Next Peak* posouvá marker na následující menší maximum.

Specifikem přístroje FSH je absence nastavení počtu vzorků tak, jak bylo zvykem u FSL či ZVL. Namísto toho je signál vždy vykreslen pomocí 631 bodů, a v takovém množství vzorků je také možné ho exportovat do Matlabu či CSV.

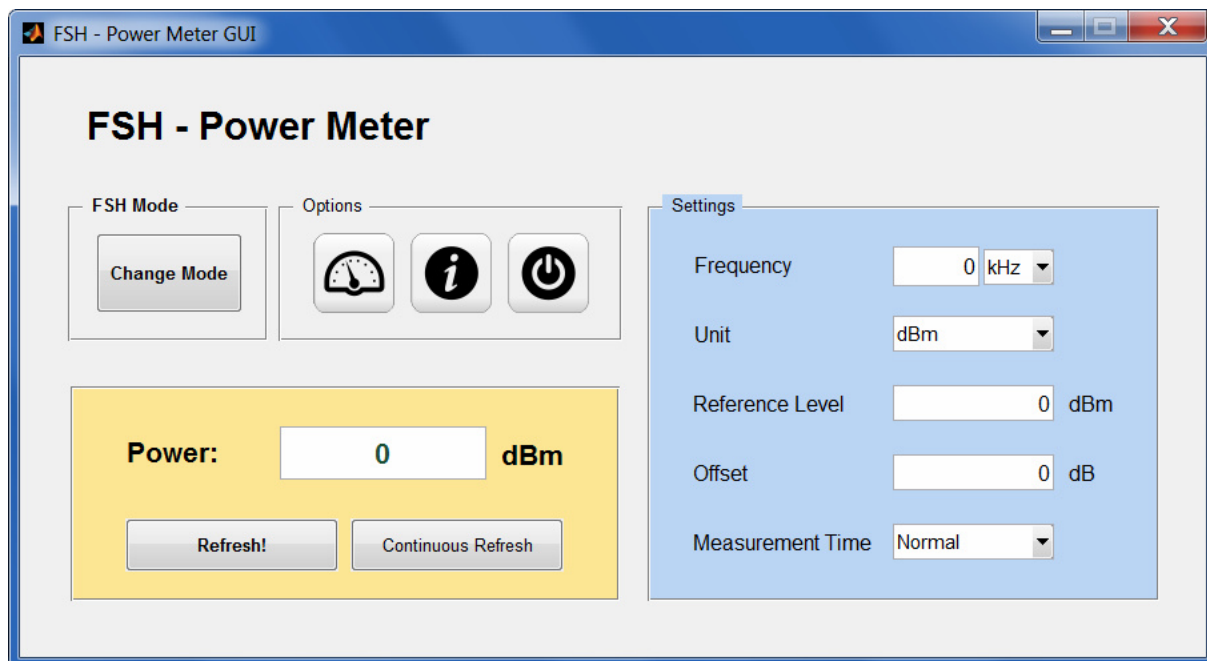
Tlačítkem, které je přítomné výhradně v GUI určených jednotlivým režimům FSH, je *Change Mode*. Po kliknutí na něj je zavřeno aktuální rozhraní a uživateli je nabídnuta zpět volba módu v FSH-GUI (obr. 5.10).

5.5.2 Network Analyzer GUI

Rozdíl mezi režimem obvodového analyzátoru a spektrálního analyzátoru (a jejich GUI) opět spočívá ve skalárním vs. vektorovém způsobu měření signálu. Network Analyzer GUI má proto v nabídce panelu *Display Format* adekvátní možnosti: výkonové spektrum v [dB], fáze, skupinové zpoždění, měření poměru stojatých vln VSWR, koeficient odrazu vlny

ad. Přítomná je rovněž volba odrazových a přenosových koeficientů rozptylových s-parametrů, a to pomocí přepínačů v panelu *S – Parameters*.

5.5.3 Power Meter GUI



Obr. 5.12 FSH – Power Meter GUI

Power Meter GUI má osobitou strukturu okna – jelikož je výsledkem měření výkonu číslo a nikoli průběh, není v GUI přítomen souřadnicový systém, a pochopitelně ani části věnované markerům. **Naměřený výkon** je zobrazen v poli *Power*, a to buď po každém stisknutí tlačítka *Refresh!*, nebo periodicky po přepnutí tlačítka *Continuous Refresh!*.

Parametry měření lze nastavit v panelu *Settings* – funkce položek je z jejich popisu zřejmá, za zmínku stojí rozbalovací menu *Measurement Time*. Tím je v FSH nastaven čas mezi jednotlivými měřeními: výběrem z možností „Short“, „Normal“ a „Long“ je také ovlivněna perioda aktualizace výsledku při zapnutí „Continuous Refresh!“.

Tlačítko *Change Mode*, společně ve všech GUI určených FSH, je možné nalézt i v Power Meter GUI. Po kliknutí na něj je zavřeno aktuální rozhraní a uživateli je nabídnuta zpět volba módu v FSH-GUI (obr. 5.10).

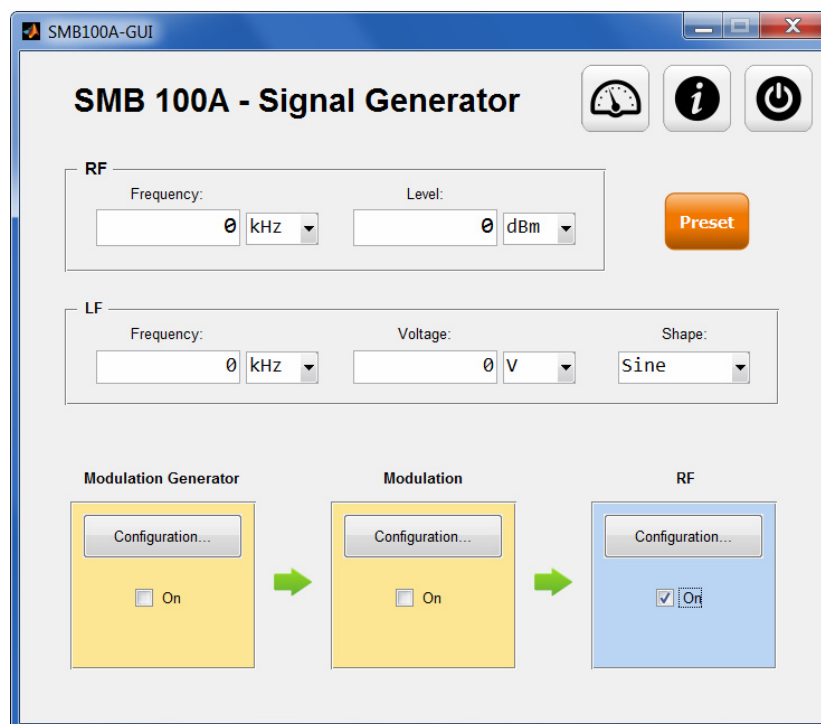
Pokud se při měření v režimu Power Meter nezobrazuje na přístroji výsledek, je pravděpodobné, že k němu není připojena sonda *Power Sensor* (např. FSH-Z1 na obr. 4.7). Na takovou situaci aplikace pamatuje a uživateli zobrazí následující hlášku:



Obr. 5.13 FSH – Power Meter GUI – upozornění na připojení výkonové sondy

5.6 SMx100A-GUI

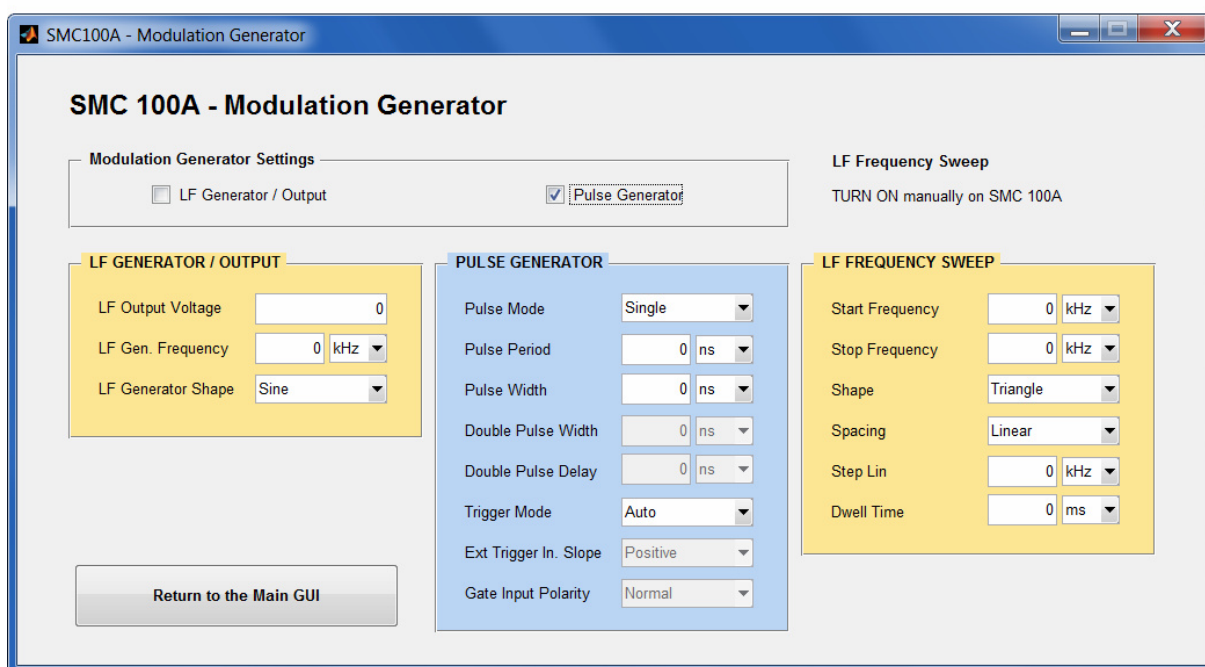
Velká podobnost ovládacích rozhraní pro signálové generátory řady SM umožňuje popsat společně všechny tři typy GUI, které jsou součástí aplikace pro dálkové ovládání. Jedná se o *SMB100A-GUI*, *SMBV100A-GUI* a *SMC100A-GUI*, přičemž každé z těchto grafických rozhraní ještě obsahuje tři další podokna poskytující možnost detailního nastavení vlastností generátoru. Struktura a grafická podoba GUI pro generátory kopíruje uspořádání ovládacích prvků z displeje přístrojů – uživatel tak může snadno nalézt požadovanou funkci, přičemž její nastavení či spuštění bude na PC nesrovnatelně pohodlnější než na panelu generátoru.



Obr. 5.14 Hlavní GUI signálových generátorů

Po připojení a rozpoznání přístroje je spuštěno hlavní okno – v něm jsou dvě skupiny editačních polí pro možnost rychlého nastavení radiofrekvenčního (*RF*) a nízkofrekvenčního (*LF*) generátoru. Nacházejí se zde také tři barevně odlišené bloky, které představují základní funkce generátorů (jejich popis následuje v dalších kapitolách). Po kliknutí na tlačítko *Configuration...* se zobrazí příslušné podokno s vlastnostmi, zaškrtačacím políčkem *On* lze funkci jednoduše zapnout nebo vypnout. Aby bylo pro uživatele na první pohled patrné, který blok je v chodu, změní se po zapnutí jeho barva na modrou. Modré označení aktivních funkcí platí ve všech GUI a jejich podoknech.

5.6.1 Modulation Generator

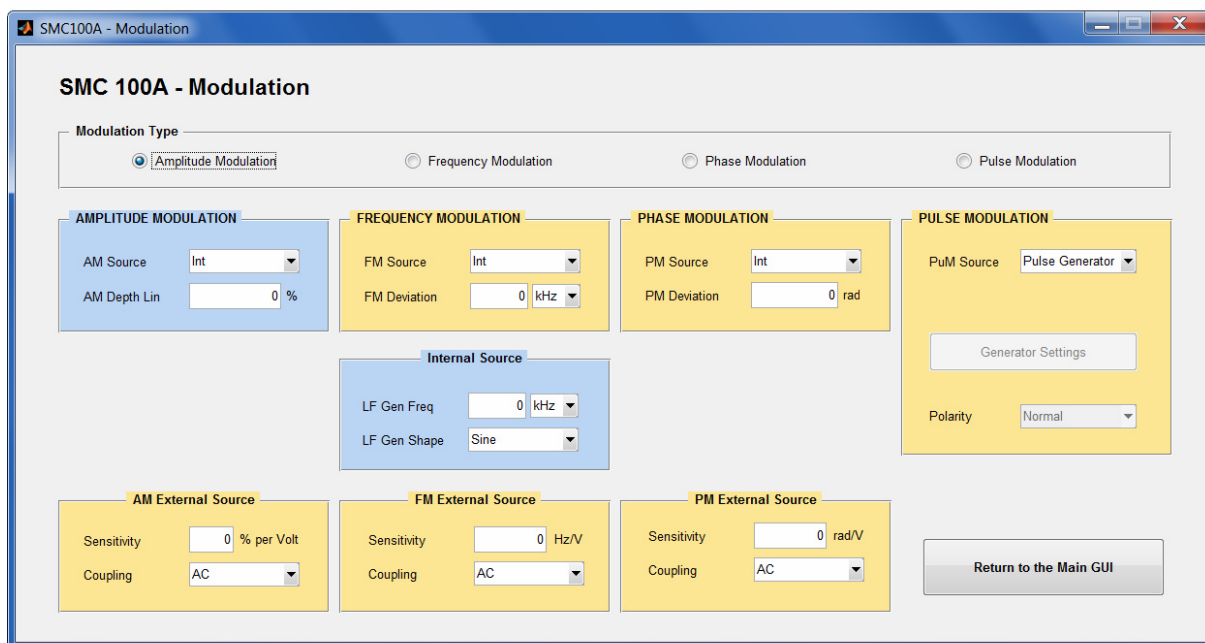


Obr. 5.15 Modulation Generator GUI

Na obrázku je zachyceno okno „Modulation Generator“ pro SMC100A. Ten je narozdíl od zbylých dvou typů vybaven navíc pulzním generátorem, který je možné zapnout zaškrtnutím *Pulse Generator*. Pro všechny typy platí, že tento GUI slouží k nastavení a zapnutí nízkofrekvenčního generátoru, který při zapnutí některé z modulací slouží také jako modulační signál. Kromě výstupního napětí, frekvence a tvaru signálu je možné pohodlně nastavit všechny vlastnosti *LF Frequency Sweep*, plynulé změny frekvence výstupního signálu v rozmezí daném hodnotami v polích *Start Frequency* a *Stop Frequency*. Zapnutí tohoto frekvenčního posunu je však nutné provést na samotném přístroji – v nastavení bloku „Modulation Generator“ zvolíme položku „LF Frequency Sweep“.

Všechna podokna nastavení obsahují tlačítko *Return to the Main GUI*, které uživatele vrátí do hlavního okna grafického rozhraní.

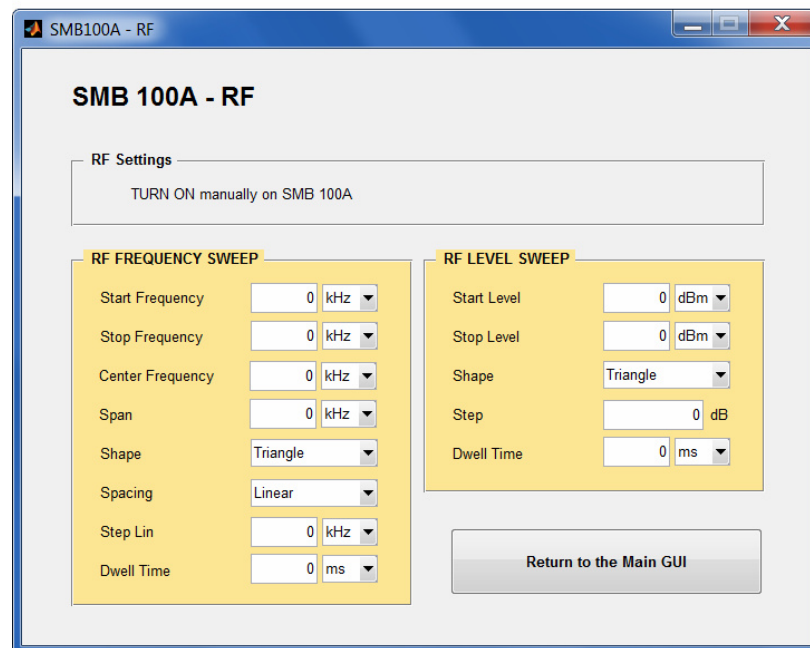
5.6.2 Modulation



Obr. 5.16 Modulation GUI

V okně „Modulation“ je dostupné nastavení amplitudové, frekvenční, fázové a v případě SMC100A také pulzní modulace. Při nastavení zdroje modulace na *Int* (vniřní) slouží nízkofrekvenční (*LF*) oscilátor jako modulační signál o tvaru zvoleném v menu *LF Gen Shape*. Opět platí, že aktivní bloky změní svou barvu na modrou. Režim modulace lze jednoduše vypnout v hlavním okně odškrtnutím políčka *On*.

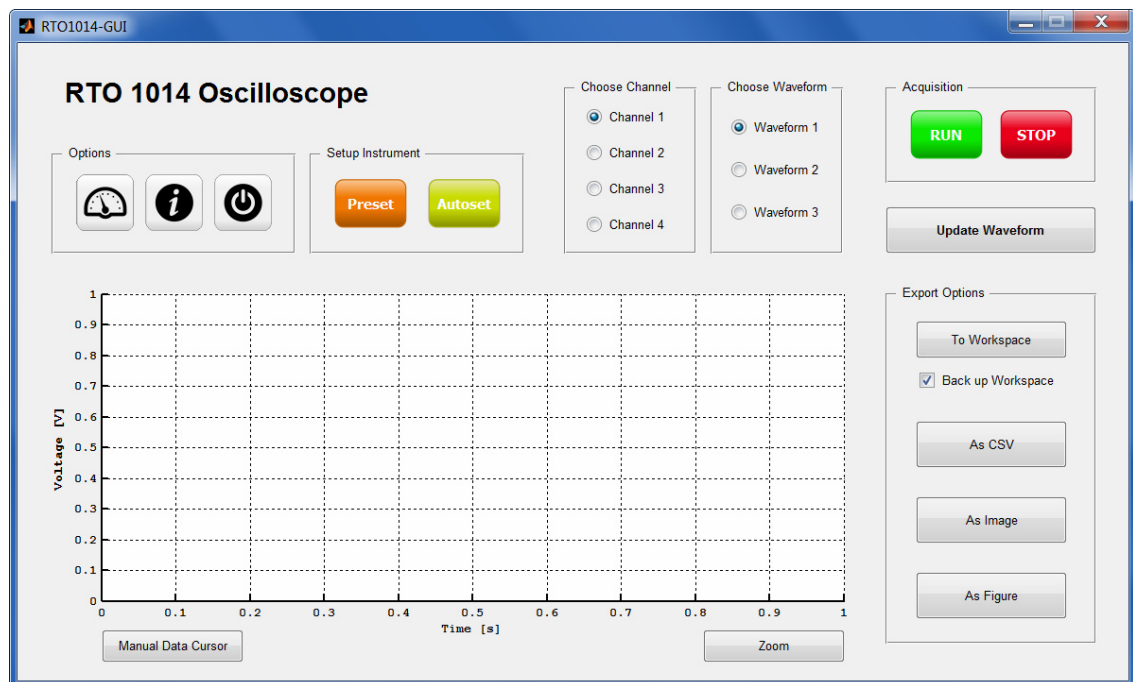
5.6.3 RF



Obr. 5.17 RF GUI

Vlastnosti plynulého posunu („sweep“) frekvence a úrovně radiofrekvenčního generátoru lze nastavit v okně „RF“. Stejně jako u LF generátoru, i tentokrát je nutné zapnout sweeping ručně na přístroji: v nastavení bloku „RF“ zvolíme položku „RF Frequency Sweep“, resp. „RF Level Sweep“.

5.7 RTO1014-GUI



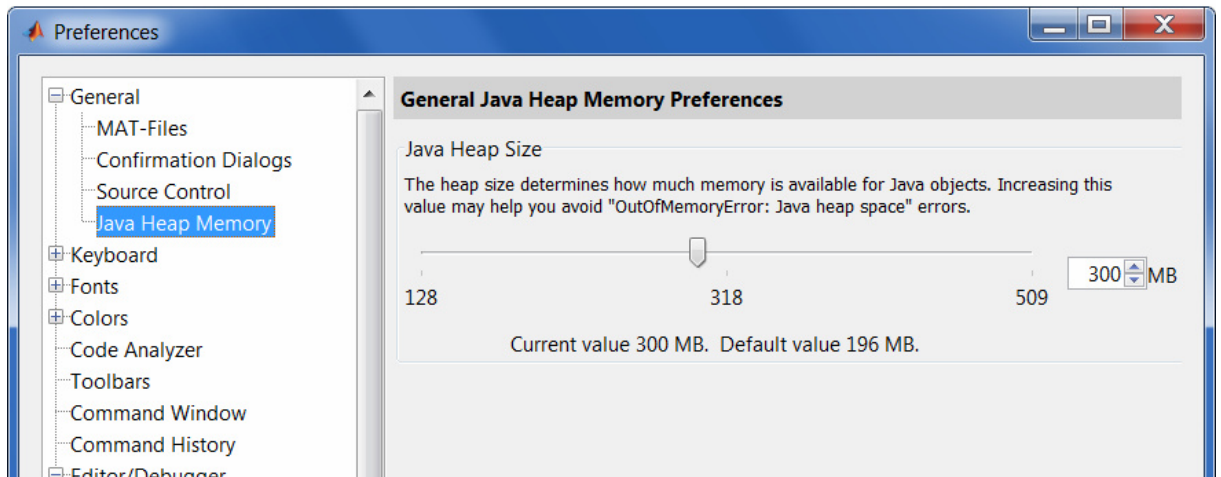
Obr. 5.18 RTO1014-GUI

Při programování GUI pro R&S[®] RTO1014 byla pozornost věnována především problematice přenosu dat, jejichž objem může být v závislosti na nastaveném rozlišení mnohonásobně větší než u předchozích GUI. Osciloskop poskytuje značné množství funkcí, které jsou však snadno a přehledně přístupné pomocí velkého dotykového displeje – nastavení vlastností měření proto nebylo nutné zanášet do ovládacího software.

V grafickém rozhraní můžeme najít prvky typické pouze pro ovládání osciloskopu. Tlačítko *Autoset* (vedle tlačítka *Preset*) zastupuje stejnou funkci, jako jeho fyzický zástupce přímo na panelu vedle displeje – přístroj se pokusí sám nalézt vyhovující zobrazení adekvátní ke snímanému signálu. Panely *Choose Channel* a *Choose Waveform* umožňují uživateli určit, který z měřených průběhů bude aplikací zpracován. Tlačítka *RUN* a *STOP* lze spustit anebo zastavit plynulé měření signálu. Přestože je v GUI přítomný souřadnicový systém určený k vykreslení snímaného průběhu po stisku tlačítka *Update Waveform*, nemusí být pro uživatele výhodné tuto funkci využít – kvůli velkému objemu přenášených dat může tato operace totiž trvat poměrně dlouhou dobu (záleží na výkonu PC). Na osciloskopu samotném je přitom průběh zobrazen v dostatečné kvalitě, a uživatel tedy může použít GUI k exportu dat ve vybraném formátu bez čekání na vykreslení grafu.

Kromě obvyklého *Manual Data Cursor* doplňuje navíc graf funkce *Zoom*, která uživateli umožní využít přednosti vysokého rozlišení zobrazeného průběhu.

Extrémní množství vzorků s sebou přináší také jisté komplikace – grafické rozhraní GUI využívá pro svůj běh *Java Heap Memory*, přičemž stahovaná data jsou načítána právě do této paměti. Její velikost závisí na výkonu počítače (především na operační paměti RAM) a je tedy omezená. V Matlabu ji lze nastavit v nabídce „Preferences“, záložka „General“, možnost „Java Heap Memory“.



Obr. 5.19 Nastavení Java Heap Size v Matlabu

I při nastavení maximální dostupné hodnoty se může stát, že paměti nebude dostatek. Na testovaném PC při nastavení Java Heap Size na 1000 MB činil největší možný počet stažených vzorků 2 miliony (2 MSa). Pokud byla na osciloskopu zvýšena hodnota na následný krok 4 MSa, Matlab vypsal tuto chybu:

```
Error using icinterface/query (line 120)
Java exception occurred:
java.lang.OutOfMemoryError: Java heap space
```

Pokud se při práci s RTO1014-GUI objeví podobné varování i po zvětšení Java Heap Size, je třeba v osciloskopu snížit rozlišení podle návodu z kapitoly 4.4.5.

Závěr

V úvodních kapitolách práce byly popsány standardy, které definují pravidla pro vzdálené řízení měřicích zařízení, včetně detailního popisu příkazů SCPI – programovacího jazyka, který je speciálně určen právě pro dálkové ovládání přístrojů. Jak se později ukázalo, právě tato teoretická příprava pomohla pochopit poněkud nezvyklou problematiku, a získané znalosti mohly být následně úspěšně využity při tvorbě řídicí aplikace.

Důležitou epizodu při vývoji software, která předcházela programování zdrojového kódu grafických rozhraní, představovalo hledání a zprovoznění vhodného způsobu spojení. Z možností, které se nabízely, se nakonec pro svou rychlost a jednoduchost prosadila realizace místní sítě LAN pomocí technologie Ethernet (využívající klasického TCP/IP protokolu), přičemž spojení a komunikaci obstarávají systém Matlab s Instrument Control Toolboxem ve spolupráci s knihovnou VISA. Velkou výhodou takového přístupu je především jeho univerzálnost – odpadá totiž nutnost instalace ovladačů každého přístroje zvlášť. Ethernetová komunikace má navíc teoretické předpoklady k tomu být ze všech variant nejrychlejší. Původně zamýšlená analýza přenosových rychlostí, která by tuto hypotézu potvrdila, se však v praxi ukázala bezpředmětná – vliv výkonu samotných měřicích zařízení a řídicího PC na dobu přenosu byl nakonec daleko větší, než druh použité komunikační sběrnice.

Aplikace vznikala osobitým způsobem, jelikož byla vytvářena na míru svému budoucímu využití. Každé měřicí zařízení, pro které byl tvořen software dálkového řízení, nabízí samozřejmě celou řadu speciálních funkcí – účelem aplikace tak nebylo podchytit všechny dostupné možnosti, nýbrž pouze často používané funkce důležité pro obvyklé měření. Programování proto provázely opakované konzultace a následné úpravy tak, aby všechny aspekty splňovaly požadavky zadání.

Struktura aplikace byla navržena po pečlivém rozvážení, neboť na jedné straně musí být uživateli poskytnuto přehledné, pro každý přístroj individuální grafické rozhraní, jehož funkce odpovídají typu zařízení, přitom by však aplikace neměla zbytečně obtěžovat svým vlastním komplikovaným ovládáním. Nejvhodnějším řešením se nakonec ukázala být kombinace jednoho společného hlavního okna, v němž je uskutečněno připojení pomocí IP adresy, a následné spouštění samostatných oken z tohoto centrálního GUI.

Tento způsob se při testování aplikace osvědčil – nejenže poskytuje uživateli dostatek prostoru pro pohodlné zadávání vstupních dat a vyčítání průběhů, ale také umožnil z pohledu programátora efektivní a průhlednou správu síťového spojení. Nezávislost jednotlivých oken zároveň významnou měrou usnadňuje další rozšiřování aplikace naprogramováním GUI pro libovolný typ přístroje a jeho snadnou implementací do úvodního okna pro nové připojení.

Neúspěchem skončil pokus vytvořit řídicí GUI pro přenosný přijímač PR100 sloužící k rádiovému monitoringu. Záznam dat tímto přístrojem funguje na zcela jiném principu než u ostatních zařízení – ukládaná data nemají předem daný počet vzorků, nahrávání naopak probíhá jako UDP stream, jehož délku si určuje uživatel sám. Vyčítání dat ze struktury tohoto datového toku vyžaduje zřejmě rozbor UDP datagramů, tzn. oddělení bajtů hlavičky a rozpoznání bajtů nesoucích informaci o snímaném signálu. Dálkové ovládání PR100 tedy zůstává příležitostí pro budoucí rozšíření tématu této diplomové práce.

Zdroje

- [1] PIEPER, John M. *Automatic Measurement Control: A tutorial on SCPI and IEEE 488.2*. München: Rohde & Schwarz GmbH & Co. KG, 2007. ISBN 3939837024.
- [2] IEEE-488. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, [cit. 2013-03-07]. Dostupné z: <http://en.wikipedia.org/wiki/IEEE-488>
- [3] Standard Commands for Programmable Instruments. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-03-07]. Dostupné z: http://en.wikipedia.org/wiki/Standard_Commands_for_Programmable_Instrumentation
- [4] ASCII. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-03-07]. Dostupné z: <http://cs.wikipedia.org/wiki/ASCII>
- [5] ROHDE&SCHWARZ. *ZVL Vector Network Analyzer: Operating Manual* [PDF]. 2009 Dostupné z: <http://www.rohde-schwarz.com>
- [6] ROHDE&SCHWARZ. *FSH4/8 Remote Control via LAN or USB: Software Manual* [PDF]. 2012. Dostupné z: <http://www.rohde-schwarz.com>
- [7] ROHDE&SCHWARZ. *Remote emulation modes with R&S signal generators* [PDF]. Dostupné z: <http://www.rohde-schwarz.com>
- [8] BIGELOW, Stephen J. *Mistrovství v počítačových sítích: správa, konfigurace, diagnostika a řešení problémů*. Vyd. 1. Překlad Petr Matějů. Brno: Computer Press, 2004, 990 s. ISBN 80-251-0178-9.
- [9] Virtual Instrument Software Architecture. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-04]. Dostupné z: http://en.wikipedia.org/wiki/Virtual_Instrument_Software_Architecture
- [10] AGILENT TECHNOLOGIES. *Using Linux to Control LXI Instruments Through VXI-11* [PDF], 2007. Dostupné z: <http://cp.literature.agilent.com/litweb/pdf/5989-6716EN.pdf>
- [11] ROHDE & SCHWARZ. [online]. 2013. Dostupné z: <http://www.rohde-schwarz.com>

Seznam zkratek

API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ATE	Automatic Test Equipment
GPIB	General Purpose Interface Bus
GUI	Graphical User Interface
GUIDE	GUI Development Environment
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
NI	National Instruments
ONC-RPC	Open Network Computing – Remote Procedure Calls
R&S	Rohde & Schwarz
SCPI	Standard Commands for Programmable Instrumentation
USB	Universal Serial Bus
VISA	Virtual Instrument Software Architecture
VMEbus	VersaModular Eurocard bus
VXI	VMEbus Extensions for Instruments
XDR	External Data Representation

Seznam obrázků

Obr. 1.1	Přehled standardů pro ATE a jejich role ve vrstevném modelu [1].....	11
Obr. 2.1	ASCII tabulka [4]	18
Obr. 2.2	Hierarchický strom uzlů podsystému SENSE [1].....	34
Obr. 2.3	Model SCPI přístroje [1]	36
Obr. 2.4	Blokové schéma přístroje pro měření signálu [1].....	37
Obr. 2.5	Blokové schéma přístroje pro generování signálu [1].....	38
Obr. 3.1	ISO/OSI model síťové komunikace v ATE systému [6].....	41
Obr. 4.1	Grafické rozhraní Instrument Control Toolboxu.....	45
Obr. 4.2	NI-VISA – Measurement & Automation Explorer	46
Obr. 4.3	Dialogové okno připojení nového přístroje	46
Obr. 4.4	R&S®FSL [11].....	48
Obr. 4.5	FSL – nastavení parametrů místní sítě	48
Obr. 4.6	R&S®ZVL [11].....	49
Obr. 4.7	R&S®FSH a sonda R&S®FSH-Z1 [11].....	50
Obr. 4.8	FSH – nastavení parametrů místní sítě.....	51
Obr. 4.9	R&S®SMB100A (vlevo), SMC100A (vpravo) a SMBV100A (dole) [11].....	52
Obr. 4.10	SMx – nastavení parametrů místní sítě.....	52
Obr. 4.11	R&S®RTO [11].....	53
Obr. 4.12	RTO – nastavení parametrů místní sítě.....	54
Obr. 5.1	Skupina tlačítek pro export dat.....	57
Obr. 5.2	Zadání názvu proměnných ve Workspace.....	57
Obr. 5.3	Tlačítko pro zapnutí funkce Data Cursor	58
Obr. 5.4	Úvodní okno aplikace s nabídkou nového připojení	59
Obr. 5.5	Hláška při nerozpoznání typu připojeného přístroje.....	60
Obr. 5.6	Varování při pokusu o zavření mainGUI	60
Obr. 5.7	FSL-GUI.....	61
Obr. 5.8	ZVL-GUI.....	62
Obr. 5.9	Export proměnných do Workspace	63
Obr. 5.10	FSH-GUI = volba módu přístroje	64
Obr. 5.11	FSH – Spectrum Analyzer GUI.....	65
Obr. 5.12	FSH – Power Meter GUI	66
Obr. 5.13	FSH – Power Meter GUI – upozornění na připojení výkonové sondy.....	67

Obr. 5.14	Hlavní GUI signálových generátorů.....	67
Obr. 5.15	Modulation Generator GUI.....	68
Obr. 5.16	Modulation GUI.....	69
Obr. 5.17	RF GUI	70
Obr. 5.18	RTO1014-GUI.....	71
Obr. 5.19	Nastavení Java Heap Size v Matlabu.....	72

Seznam tabulek

Tabulka 2.1	Přehled předpon soustavy SI a jejich značení v SCPI [1].....	25
Tabulka 2.2	Přehled číselných soustav parametrů SCPI příkazů [1]	27
Tabulka 2.3	Seznam klíčových slov reprezentujících numerické hodnoty [6]	28
Tabulka 5.1	Přehled společných tlačítek v GUI.....	56