

Aplikace pro rychlé skalární měření výkonového přenosu v závislosti na kmitočtu

Application for fast scalar measurement of power transmission with
respect to frequency

Bc. Ondřej Růčka

Diplomová práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Ondřej RŮČKA**
Osobní číslo: **A11709**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**
Forma studia: **prezenční**

Téma práce: **Aplikace pro rychlé skalární měření výkonového přenosu v závislosti na kmitočtu**

Zásady pro vypracování:

1. Nastudujte programování grafických aplikací v Matlabu.
2. Nastudujte Instrument Control Toolbox v Matlabu.
3. S pomocí Instrument Control Toolboxu vytvořte program pro skalární měření frekvenčního přenosu mezi generátorem SMR20 a spektrálním analyzátozem FSP40, tak aby bylo dosaženo rychlosti alespoň 10 milisekund na měřící bod nebo lepší.
4. Výsledky měření prezentujte jednoduchou grafickou aplikací v Matlabu pro některý vybraný typ mikrovlnných měření.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. HERINGOVÁ, B., HORA, P. Matlab Díl 1. Práce s programem, Praha. Vydavatelství H-S 1995.
2. HERINGOVÁ, B., HORA, P. Matlab Díl 2. Popis funkcí, Praha. Vydavatelství H-S 1995.
3. Introduction to Graphical User Interface (GUI), Matlab 6.5. PDF document, dostupný z <http://www.ewh.ieee.org/r8/uae/GUI.pdf>.
4. RAUSCHER, C. Fundamentals of spectrum analysis. Munich, 2007. ISBN 978-3-939837-01-5. 219 stran.
5. Microwave generator SMR 20. Operating manual. RohdeSchwarz. 2009. 230 stran.
6. Spectrum analyzer FSP 40. Operating manual. RohdeSchwarz. 2010. 796 stran.

Vedoucí diplomové práce:

Ing. Stanislav Goňa, Ph.D.

Ústav elektroniky a měření

Datum zadání diplomové práce:

8. února 2013

Termín odevzdání diplomové práce:

3. června 2013

Ve Zlíně dne 8. února 2013



prof. Ing. Vladimír Vasek, CSc.
děkan



doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

ABSTRAKT

Cílem této práce je vytvořit program, který skalárně měří frekvenční přenos mezi generátorem SMR20 a spektrálním analyzátozem FSP40. Toho je docíleno vzdáleným ovládáním těchto přístrojů pomocí vhodných příkazů pro daný úkol. Program by měl být prezentován jako grafická aplikace vytvořená v Matlabu. Měření má probíhat rychle, cílem je dosáhnout rychlosti alespoň 10 ms na jeden měřený bod.

Klíčová slova: Frekvenční analýza, analyzátor, vzdálené ovládání, GPIB, SCPI, Matlab, FSP 40, SMR 20

ABSTRACT

The goal of this work is to create a program that measures the scalar frequency transmission between the generator SMR20 and spectrum analyzer FSP40. This is achieved by remote control of devices using commands for the task. The program should be presented as a graphical application created in Matlab. Measurement should be quick, measurement speed should be at least 10 ms for one measurement point.

Keywords: Frequency analysis, analyzer, remote control, GPIB, SCPI, Matlab, FSP 40, SMR 20

Poděkování

Děkuji vedoucímu mé diplomové práce Ing. Stanislavu Goňovi za ochotu, vstřícnost, kvalitní a odborné vedení a poskytnuté konzultace.

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 FREKVENČNÍ ANALÝZA	12
1.1 DRUHY SIGNÁLŮ	12
1.1.1 Periodické signály	13
1.1.2 Neperiodické signály.....	15
1.2 PŘÍKLADY SIGNÁLŮ A JEJICH FREKVENČNÍCH SPEKTER	15
1.2.1 Sinusový signál	15
1.2.2 Amplitudově modulovaný signál	16
1.2.3 Obdélníkový signál	16
1.2.4 Trojúhelníkový signál	17
1.2.4.1 Šum	17
1.3 VYUŽITÍ	18
1.4 DRUHY FREKVENČNÍCH ANALYZÁTORŮ	18
1.4.1 Analogové analyzátory.....	18
1.4.1.1 Analyzátor s přepínatelnými filtry	18
1.4.1.2 Analyzátor s přeladitelným filtrem	19
1.4.1.3 Heterodynní spektrální analyzátor	19
1.4.2 Spektrální analyzátory používající číslicové filtry.....	20
1.4.3 FTT analyzátory	20
1.5 ZOBRAZENÍ SPEKTRA.....	21
1.6 PARAMETRY	22
1.6.1 Rozsah zobrazovaných frekvencí.....	22
1.6.2 Rozsah zobrazovaného výkonu.....	23
1.6.3 Rozlišení frekvencí (RBW).....	23
1.6.4 Sweep time	23
1.6.5 Funkce stopy	23
1.6.6 Detektory	23
1.6.6.1 MAX PEAK.....	24
1.6.6.2 MIN PEAK	24
1.6.6.3 AUTO PEAK.....	24
1.6.6.4 SAMPLE.....	24
1.6.6.5 RMS	24
1.6.6.6 AVERAGE	25
1.6.6.7 QUASI PEAK.....	25
2 VZDÁLENÉ OVLÁDÁNÍ ANALYZÁTORU FSP 40	26
2.1 GPIB	26
2.2 ZÁKLADNÍ PARAMETRY	26
2.3 LINKY.....	27
2.3.1 Datové linky	27
2.3.2 Linky pro řízení přenosu	27
2.3.3 Linky pro řízení rozhraní	27
2.4 PŘENOS	28
2.4.1 Algoritmus přenosu dat	29

2.5	SCPI PŘÍKAZY	29
2.5.1	Struktura příkazu	29
2.5.2	Parametry	30
2.5.2.1	Číselné parametry	30
2.5.2.2	Boolovy parametry	31
2.5.2.3	Textové parametry	31
2.5.2.4	Znakové řetězce	31
2.5.2.5	Bloky dat	31
2.6	DĚLENÍ PŘÍKAZŮ	31
2.6.1	Běžné příkazy	32
2.6.2	Příkazy pro specifické zařízení	32
2.7	ZPRACOVÁNÍ PŘÍKAZŮ	33
2.7.1	Popis toku dat	33
2.7.1.1	Vstupní jednotka	33
2.7.1.2	Rozeznávání příkazu	34
2.7.1.3	Datová část a hardware	34
2.7.1.4	Signalizace stavu	34
2.7.1.5	Výstupní jednotka	34
2.8	SYSTÉM HLÁŠENÍ STAVU PŘÍSTROJE	34
2.8.1	Struktura SCPI registrů	35
2.8.1.1	Část CONDition	35
2.8.1.2	Část PTRansition	35
2.8.1.3	Část NTRansition	36
2.8.1.4	Část EVENT	36
2.8.1.5	Část ENABle	36
2.8.1.6	Sumarizační bit	36
2.8.2	Status byte a service request enable registr	36
2.8.3	Event status register	37
2.8.4	STATus:OPERation registr a STATus:QUEStionable registry	38
2.9	APLIKACE SYSTÉMU HLÁŠENÍ STAVU	38
2.9.1	Service Request	38
2.9.2	Zápis chybové řady	39
II	PRAKTICKÁ ČÁST	40
3	SKALÁRNÍ MĚŘENÍ VÝKONOVÉHO PŘENOSU V ZÁVISLOSTI NA KMITOČTU	41
3.1	FUNKCE PROGRAMU	41
3.2	STRUKTURA PROGRAMU	41
3.2.1	Připojení přístrojů	41
3.2.2	Nastavení přístrojů	42
3.2.3	Měřicí část	42
3.3	ALGORITMUS PROGRAMU	43
3.4	VZDÁLENÉ OVLÁDÁNÍ	44
3.5	GUI	45
3.5.1	Rozložení GUI	47
3.5.1.1	Nastavení parametrů přístrojů	47
	Generátor SMR - 20 :	47

3.5.1.2	Tlačítka	48
3.5.1.3	Graf	48
3.5.2	Menu	49
3.5.2.1	Load_defaults.....	49
3.5.2.2	Load	49
3.5.2.3	Save.....	49
4	MĚŘENÍ.....	51
4.1	ÚKOL MĚŘENÍ.....	51
4.2	PARAMETRY MĚŘENÍ.....	51
4.3	ZMĚŘENÝ SIGNÁL.....	53
4.4	POROVNÁNÍ.....	53
4.5	ÚTLUM	55
4.6	RYCHLOST MĚŘENÍ.....	56
4.6.1	Časování	56
4.6.2	Doba sweepu	56
4.6.3	Časovač	57
4.6.4	Průměrování	57
4.6.5	Rozhraní	57
4.6.6	Měření modulovaného signálu	57
4.7	VYHODNOCENÍ VÝSLEDKŮ	58
	ZÁVĚR	59
	ZÁVĚR V ANGLIČTINĚ.....	60
	SEZNAM POUŽITÉ LITERATURY.....	61
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	62
	SEZNAM OBRÁZKŮ	63
	SEZNAM TABULEK.....	65
	SEZNAM PŘÍLOH.....	66

ÚVOD

Hlavním cílem mé diplomové práce vytvořit aplikaci, která bude měřit závislost výkonu na frekvenci v programu Matlab.

První kapitola teoretické části mé práce je věnována problematice frekvenční analýzy. Na principu frekvenční analýzy pracují analyzátoři a pro jeden z těchto přístrojů jsem psal program. Popisuju zde pouze základy této problematiky jako typy signálů nebo druhy analyzátorů.

Druhá kapitola pojednává o vzdáleném ovládní přístrojů. Podrobněji jsem se zaměřil na nástroje, které využívám v praktické části své práce pro vzdálené ovládní zařízení. Jedná se o rozhraní GPIB, které využívám pro přenos dat mezi přístroji a SCPI příkazy. Tyto příkazy jsem používal pro zadávání instrukcí jednotlivým přístrojů jako například nastavení parametrů nebo měření. Dalo by se říct, že těmito příkazy se dá přístroj ovládat podobně jako tlačítka v normálním režimu.

V praktické části nejprve popisuju program, který jsem vytvořil. Je zde podrobně popsána struktura programu, jeho funkce a grafické rozhraní, které slouží k ovládní přístroje.

Poslední částí je pak ověření funkce programu a zhodnocení. Pro kontrolu správnosti naměřených výsledků jsem porovnal hodnoty naměřené programem s hodnotami nezávisle měřenými jiným přístrojem.

I. TEORETICKÁ ČÁST

1 FREKVENČNÍ ANALÝZA

Frekvenční analýza je komplikovaný proces, jehož cílem je zobrazení frekvenčního spektra signálu. Toto spektrum je vyjádření amplitudy měřené veličiny v závislosti na frekvenci. Spektrální analýzu v praxi provádíme pomocí spektrálních analyzátorů. Tyto přístroje fungují na různých principech, které jsou dále popsány.

1.1 Druhy signálů

Elektrické signály mohou být vyjádřeny v závislosti na čase (osciloskop), nebo v závislosti na frekvenci (spektrální analyzátor). Vztah mezi těmito dvěma zobrazeními definuje Fourierova transformace. Každá proměnná v časové doméně má své frekvenční spektrum. Vztah je definován těmito vzorci:

$$X_f(f) = F\{x(t)\} = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} dt \quad (1)$$

$$x(t) = F^{-1}\{X_f(f)\} = \int_{-\infty}^{\infty} X_f(f) \cdot e^{j2\pi ft} dt \quad (2)$$

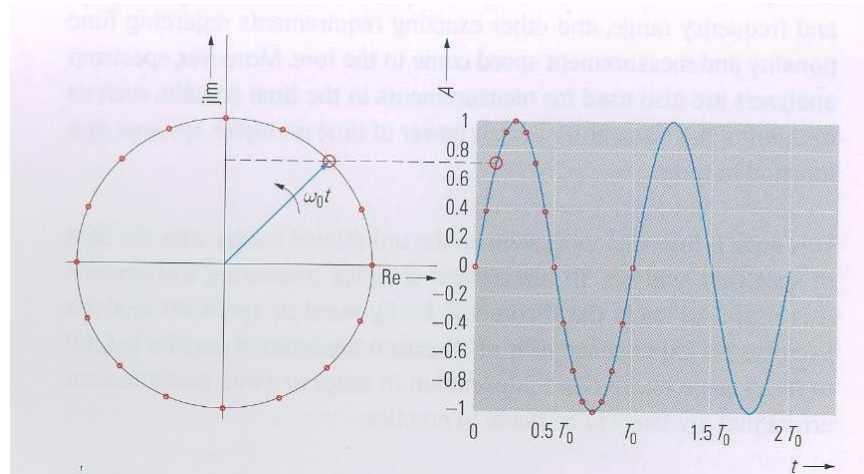
$X_f(f)$ - signál v závislosti na frekvenci

$x(t)$ - signál v závislosti na čase

$F\{x(t)\}$ - Fourierova transformace $x(t)$

$F^{-1}\{X_f(f)\}$ - převrácená hodnota Fourierovy transformace $X_f(f)$

Na obrázku 1 můžeme vidět vyjádření závislosti amplitudy na čase. Zobrazení signálu v čase využívají osciloskopy. Pro znázornění průběhu signálu je příhodné porovnání s vektorovým vyjádřením, kde amplituda signálu koresponduje s imaginární složkou vektoru. Toto porovnání je znázorněno na obrázku 2.



Obr. 1 Vektorové zobrazení signálu [4]

1.1.1 Periodické signály

Každý periodický signál podle Fourierova teoremu může být složen ze sinusových a kosinusových signálů o různé amplitudě a frekvenci. Sumu těchto signálů nazýváme Fourierova řada.

$$x(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} A_n \cdot \sin(n \cdot \omega_0 \cdot t) + \sum_{n=1}^{\infty} B_n \cdot \cos(n \cdot \omega_0 \cdot t) \quad (3)$$

Parametry A_0 , B_n a A_n jsou Fourierovy parametry a jsou definovány podle typu signálu:

$$A_0 = \frac{2}{T_0} \int_0^{T_0} x(t) dt \quad (4)$$

$$A_n = \frac{2}{T_0} \int_0^{T_0} x(t) \cdot \sin(n \cdot \omega_0 \cdot t) dt \quad (5)$$

$$B_n = \frac{2}{T_0} \int_0^{T_0} x(t) \cdot \cos(n \cdot \omega_0 \cdot t) dt \quad (6)$$

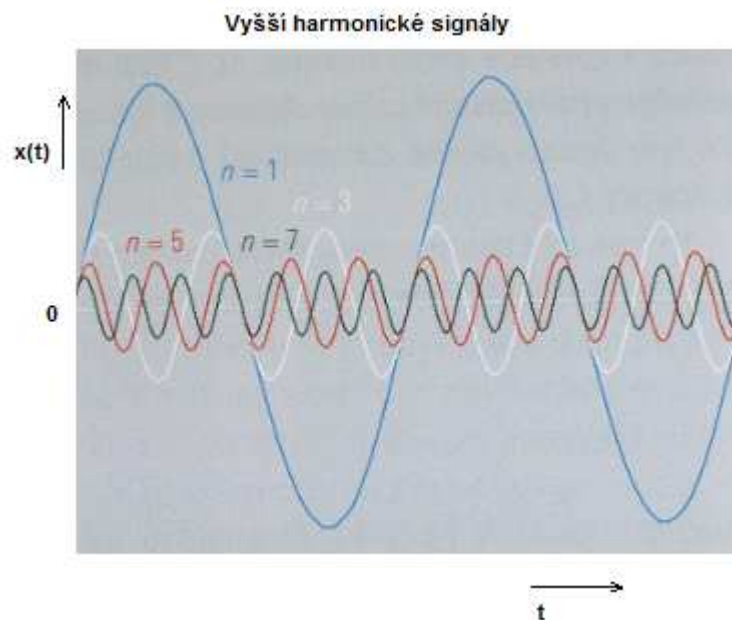
$\frac{A_0}{2}$ - stejnosměrná složka

n - stupeň vyšší harmonické frekvence

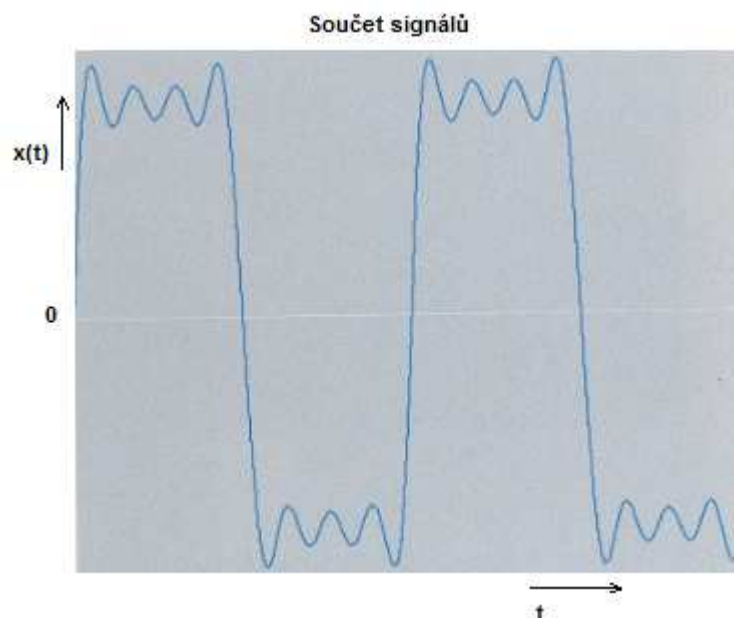
T_0 - perioda

ω_0 - úhlová frekvence

Na obrázcích níže je možné vidět signály nejprve jednotlivě a poté po jejich sloučení. Čím více je harmonických složek tvořících výsledný signál, tím více se jejich součet blíží obdélníkovému tvaru. [4]



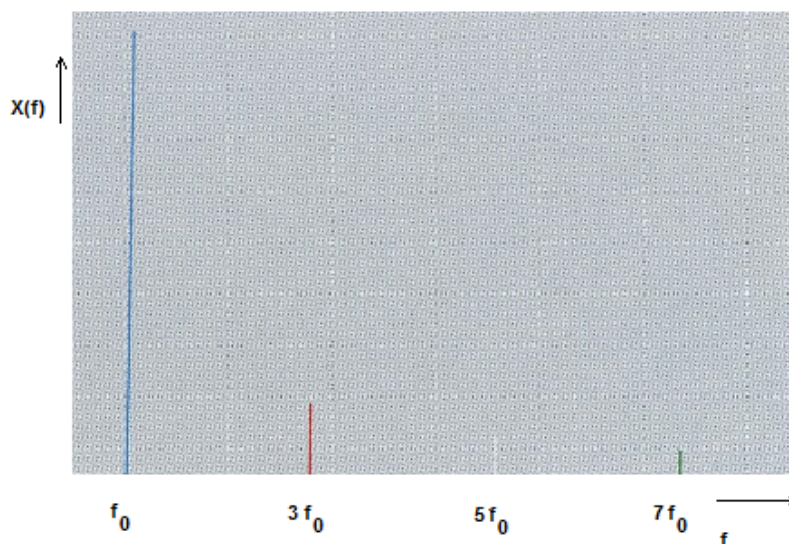
Obr. 2 Vyšší harmonické signály [4]



Obr. 3 Součet vyšších harmonických signálů [4]

Pro vypočtení frekvenčního spektra periodického signálu, který je vyjádřen pomocí Fourierovy řady, musí být každý prvek Fourierovy řady přetransformován. Z toho vyplývá, že frekvenční spektrum periodických signálů je diskrétní. [4]

Na obrázku níže můžeme vidět rozsah spektra signálu z obrázku 2. Je zde vidět, že pro každý signál náleží jedna hodnota spektra.



Obr. 4 Rozsah frekvenčního spektra [4]

1.1.2 Neperiodické signály

Neperiodické signály nemohou být v závislosti na čase popsány Fourierovou řadou. Jejich frekvenční spektrum tedy není diskrétní. Frekvenční spektrum těchto signálů je vyjádřeno pomocí Fourierovy transformace.

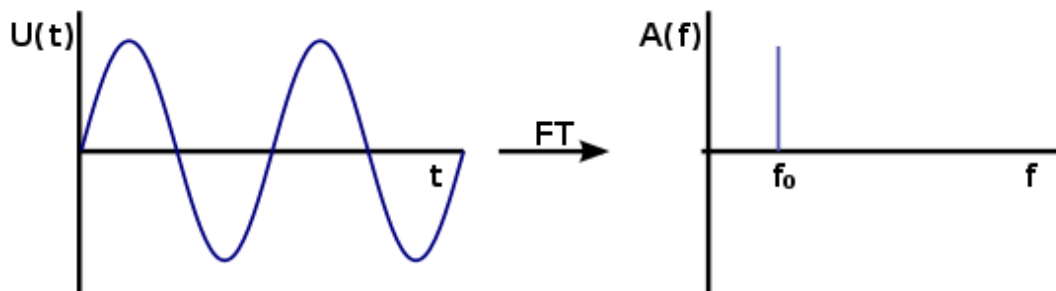
Pro signály, které se podobají sinusovým nebo kosinusovým signálům je možno spektrum přibližně spočítat. Pro signály s náhodnou charakteristikou (šum, náhodné bitové sekvence) tato možnost většinou není. [4]

1.2 Příklady signálů a jejich frekvenčních spekter

1.2.1 Sinusový signál

Průběh funkce sinus. Tento signál se též nazývá harmonický a v čase je definován:

$$U(t) = \sin(f_0 \cdot t) \quad (7)$$

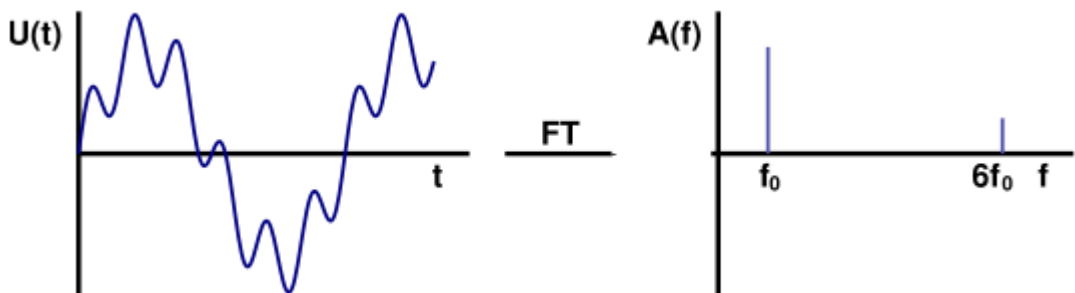


Obr. 5 Spektrum sinusového signálu [9]

1.2.2 Amplitudově modulovaný signál

Harmonický signál modulovaný pomocí jiného harmonického signálu. Jedná se o součet těchto signálů. V čase je definován:

$$U(t) = \sin(f_0 \cdot t) + \frac{\sin(6 \cdot f_0 \cdot t)}{4} \quad (8)$$

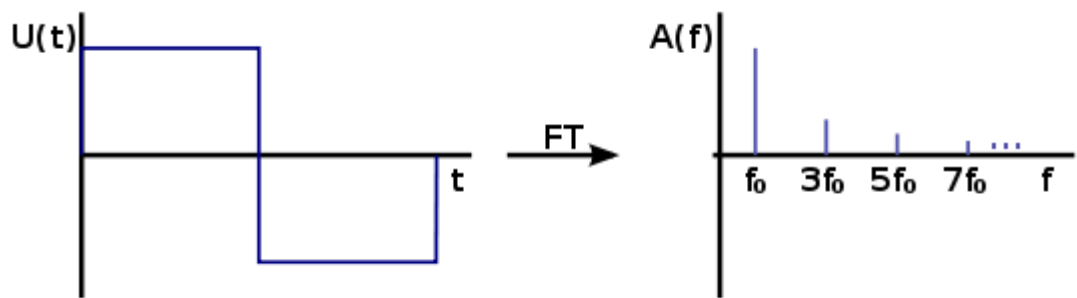


Obr. 6 Spektrum modulovaného signálu [9]

1.2.3 Obdélníkový signál

Ideální obdélníkový signál je složen z nekonečného množství vyšších harmonických signálů. Jeho frekvenční spektrum se tedy také skládá z nekonečně mnoha diskrétních hodnot. V čase je definován:

$$U(t) = \frac{\sin(f_0 \cdot t)}{1} + \frac{\sin(3f_0 \cdot t)}{3} + \frac{\sin(5f_0 \cdot t)}{5} + \dots \quad (9)$$

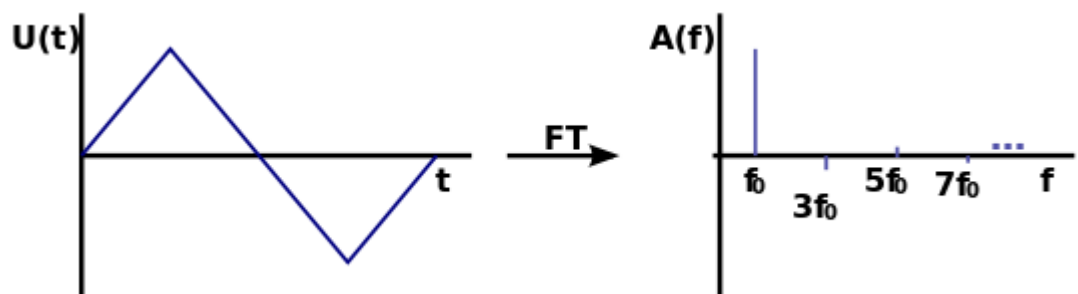


Obr. 7 Spektrum obdélníkového signálu [9]

1.2.4 Trojúhelníkový signál

Tento signál je stejně jako signál obdélníkový složen z nekonečného počtu signálů. V čase je definován:

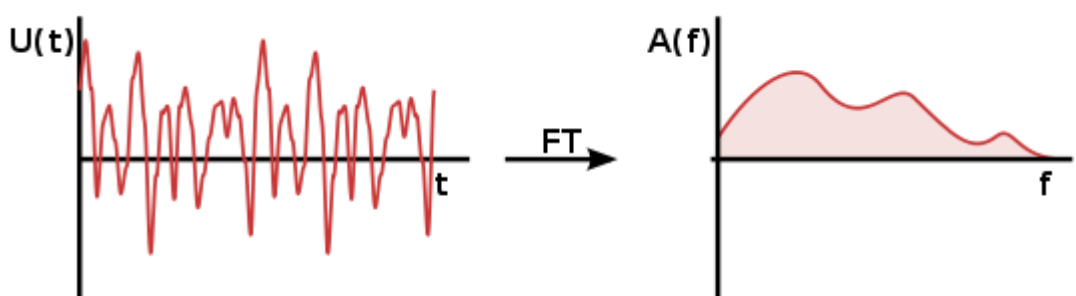
$$U(t) = \frac{\sin(f_0 \cdot t)}{1^2} - \frac{\sin(3f_0 \cdot t)}{3^2} + \frac{\sin(5f_0 \cdot t)}{5^2} - \frac{\sin(7f_0 \cdot t)}{7^2} + \frac{\sin(9f_0 \cdot t)}{9^2} - \dots \quad (10)$$



Obr. 8 Spektrum obdélníkového signálu [9]

1.2.5 Šum

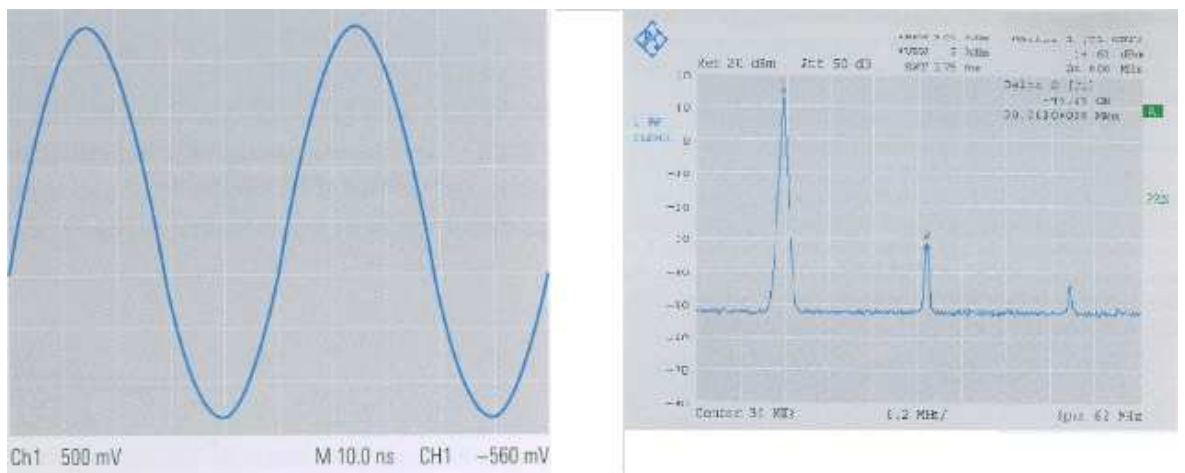
Šum je signál neperiodický, k jeho vyjádření v čase tudíž nelze použít žádný matematický vzorec.



Obr. 9 Spektrum šumu [9]

1.3 Využití

Zda-li je vhodnější měřit signál v závislosti na čase nebo v závislosti na frekvenci zaleží na tom co měříme. Pokud provádíme například měření přenosu digitálních dat, je vhodné použít měření v časové doméně na osciloskopu. Pokud chceme měřit harmonické signály, vhodnější je použít frekvenční analyzátor. Na obrázku 10 je vidět průběh signálu v závislosti na čase, který na první pohled vypadá jako sinusový signál o frekvenci 20 MHz, jehož frekvenční spektrum by se mělo skládat pouze z hodnoty 20 MHz. Provedeme-li však frekvenční analýzu, zjistíme, že frekvenční spektrum obsahuje další frekvence, které jsou při zobrazení v časové doméně neviditelné. [4]



Obr. 10 Frekvenční analýza [4]

1.4 Druhy frekvenčních analyzátorů

Analyzátoři mohou pracovat dvěma způsoby – pomocí filtrace nebo využitím diskrétní Fourierovy analýzy.

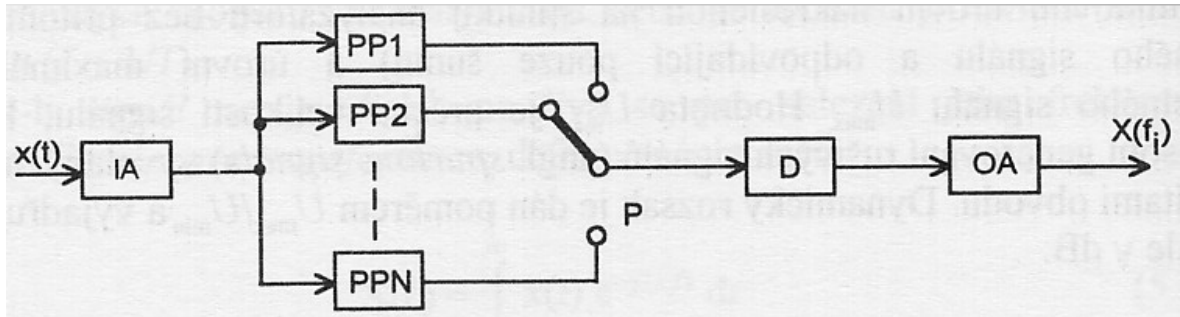
1.4.1 Analogové analyzátoři

Tyto analyzátoři využívají pro získání frekvenční analýzy analogové filtry. Analogový analyzátor měří amplitudové spektrum a jako filtry používá pásmové propusti. Tyto analyzátoři mají stálou šířku pásma, která je volitelná podle střední frekvence filtru (1 %, 3 %, 10 %, 33 % a 100 %). [10]

1.4.1.1 Analyzátor s přepínatelnými filtry

U těchto analogových analyzátorů můžeme přepínat ručně jednotlivé filtry. Tyto filtry mají stejné mezní frekvence, aby bylo možno měřit spektrum v celku. Přepínačem volíme ručně

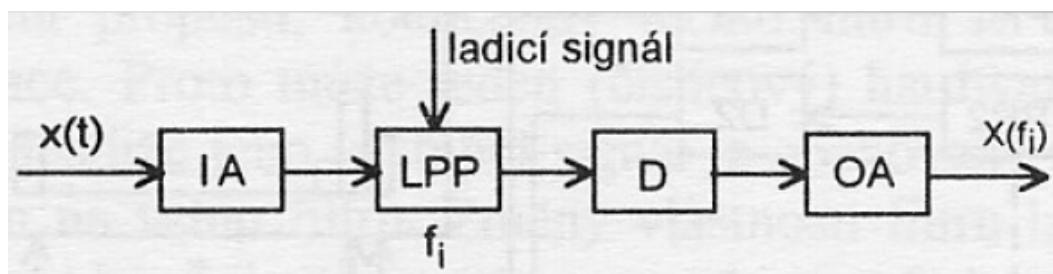
jednu z měřených složek spektra. Signál je nejprve zesílen pomocí vstupního zesilovače IA, poté je filtrována přes pásmovou propust', která je zvolena podle polohy přepínače P. Dále jde signál přes demodulátor D do výstupního zesilovače OA. [10]



Obr. 11 Analyzátor s přepínatelnými filtry [10]

1.4.1.2 Analyzátor s přeladitelným filtrem

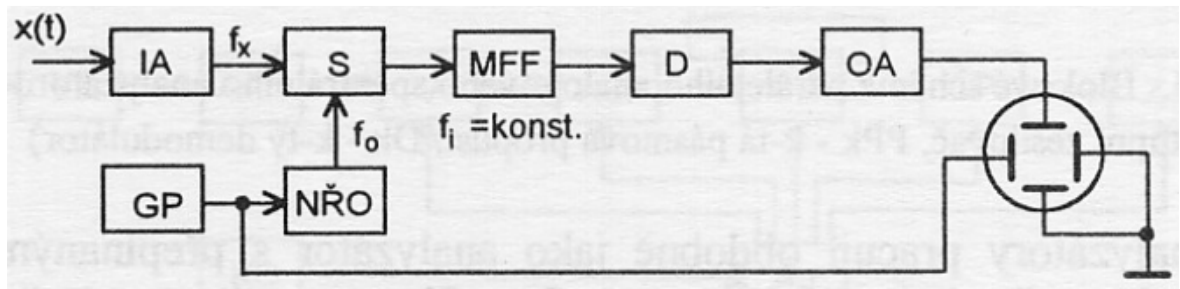
Tento analyzátor funguje na podobném principu jako analyzátor s přepínatelnými filtry s tím rozdílem, že filtry nepřepínáme ručně, ale pásmová propust' je jen jedna a je přelazována pomocí elektrického signálu. [10]



Obr. 12 Analyzátor s přeladitelnými filtry [10]

1.4.1.3 Heterodynní spektrální analyzátor

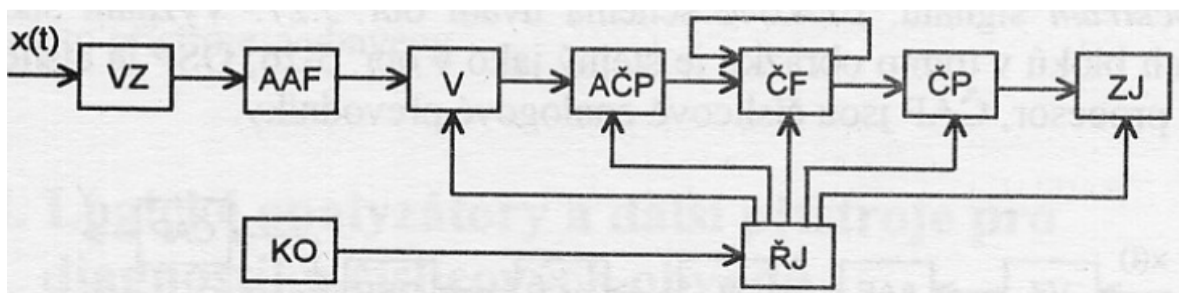
Tyto spektrální analyzátoři mají vstupní impedanci 50Ω , stejně jako většina ostatních telekomunikačních zařízení. Některé přístroje mohou mít i vstup s impedancí 75Ω , pro zapojení televizního systému (CATV). Za pomoci impedančního transformátoru lze tyto systémy měřit i se vstupem o impedanci 50Ω . Kvalita analyzátoru závisí na vstupu VSWR. Tento vstup je ovlivněn prvky atenuátor, vstupní filtr a směšovač. Atenuátor používáme při měření vysokých výkonů. Umožní nám nastavit úroveň výkonu signálu, který dále vstupuje do směšovače. Velikost atenuátoru se většinou nastavuje po 10 dB, analyzátoři pro měření se širokým dynamickým rozsahem můžou využívat atenuátory nastavitelné po 5 dB nebo i po 1 dB. Signál se dále ve směšovači smísí se signálem generovaným oscilátorem. [4]



Obr. 13 Heterodynní analyzátor [10]

1.4.2 Spektrální analyzátor používající číslicové filtry

Signál je nejprve zesílen vstupním zesilovačem VZ a filtrován antialiasingovým filtrem AAF. Poté je převeden do číslicové podoby pomocí vzorkovače V a A/D převodníku AČP. Signál je již v digitální podobě filtrován přes číslicový filtr ČF, uložený do paměti ČP a poslán na zobrazovací jednotku ZJ. [10]



Obr. 14 Analyzátor používající číslicové filtry [10]

1.4.3 FTT analyzátor

Frekvenční spektrum lze spočítat pomocí Fourierovy transformace. Pro její ideální zobrazení je však potřeba nekonečný počet dat, což je samozřejmě v praxi nerealizovatelné. Analogový signál je tedy nutno pomocí AD převodníku převést na digitální signál, který se bude skládat z určitého počtu vzorků.

Šířka pásma vstupního signálu je limitována vzorkovací frekvencí. Pokud by byla šířka pásma moc velká, došlo k překrývání signálů. [4]

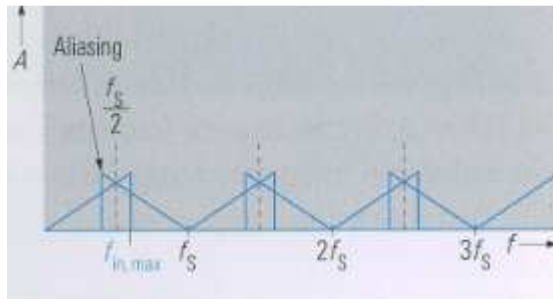
$$f_s \geq 2 \cdot B_{in} \quad (11)$$

f_s – vzorkovací frekvence

B_{in} – šířka pásma vstupního signálu

Pro signály filtrovány dolní propustí se vzorkovací frekvence určí podle maximální frekvence signálu.

$$f_s \geq 2 \cdot f_{in\ max} \quad (12)$$



Obr. 15 Překrývání signálu [4]

Pro vypočtení frekvenčního spektra z diskretního signálu využíváme diskretní Fourierovy transformace. Ta je popsána vzorcem:

$$X_k = \sum_{n=0}^{N-1} x(nT_s) \cdot e^{-2j\pi kn/N} \quad (13)$$

n – index vzorku

N – celkový počet vzorků

$x(nT_s)$ – vzorek v bodu nT_s

Výsledkem diskretní Fourierovy transformace je diskretní frekvenční spektrum. Toto spektrum je složeno z jednotlivých hodnot, které jsou vyjádřeny:

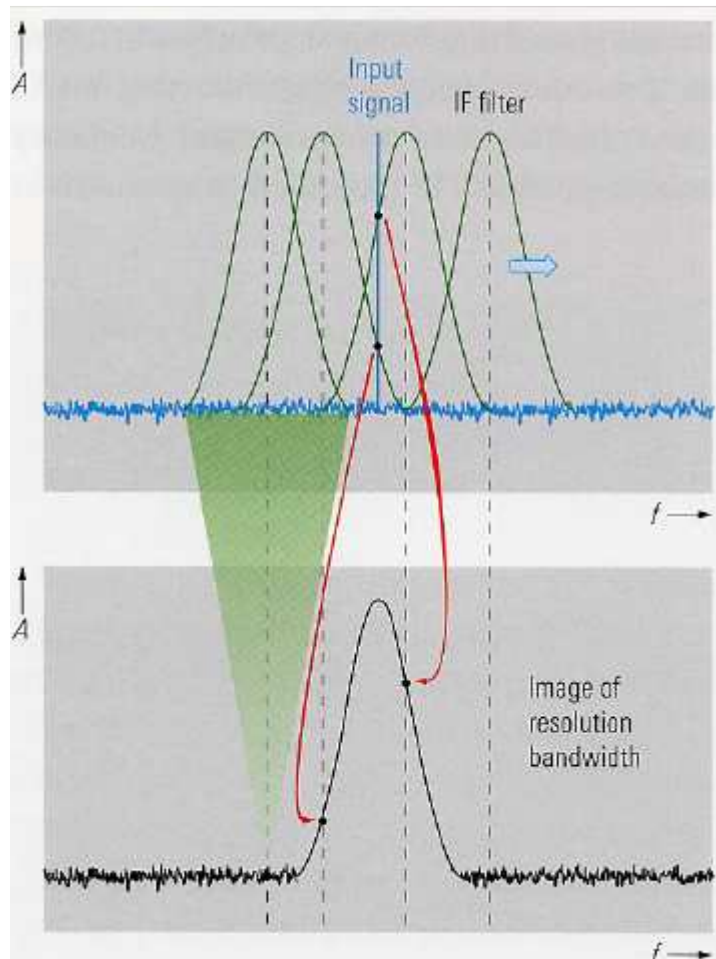
$$f(k) = k \cdot \frac{f_s}{N} = k \cdot \frac{1}{N \cdot T_s} \quad (14)$$

f_s – diskretní hodnota frekvence

k – index hodnoty frekvence

1.5 Zobrazení spektra

Proces zobrazení frekvenčního spektra se nazývá sweep. Sweepovací proces probíhá tak, že filtr (RBW – resolution bandwidth) je rozmítaný po frekvenčním pásmu o šířce definované hodnotou spanu. Střed tohoto frekvenčního pásma je určen střední frekvencí. Bod, ve kterém filtr protne vstupní signál je zobrazen na aktuální střední frekvenci filtru. Na obrázku níže je vidět proces jednoho sweepu.



Obr. 16 Zobrazení spektra [4]

Doba jednoho sweepu se nazývá sweeptime a je závislá na několika parametrech:

$$t_{sweep} = 5 \cdot \frac{1}{2\pi \cdot RBW} \cdot \frac{span}{RBW} = \frac{5}{2\pi} \cdot \frac{span}{RBW^2} \quad (15)$$

1.6 Parametry

Spektrální analyzátoři umožňují nastavit tyto základní parametry

1.6.1 Rozsah zobrazovaných frekvencí

Udává rozsah frekvencí, které bude přístroj zobrazovat. Tento rozsah je možno zadat pomocí maximální a minimální frekvence nebo zadáním střední frekvence a spanu. Moderní přístroje umožňují nastavení oběma způsoby.

1.6.2 Rozsah zobrazovaného výkonu

Tento rozsah závisí na maximální dosažené hodnotě výkonu, což je ovlivněno referenční úrovní výkonu a na spanu.

1.6.3 Rozlišení frekvencí (RBW)

Tento parametr udává šířku filtru.

1.6.4 Sweep time

Čas potřebný ke zpracování frekvenčního spektra.

1.6.5 Funkce stopy

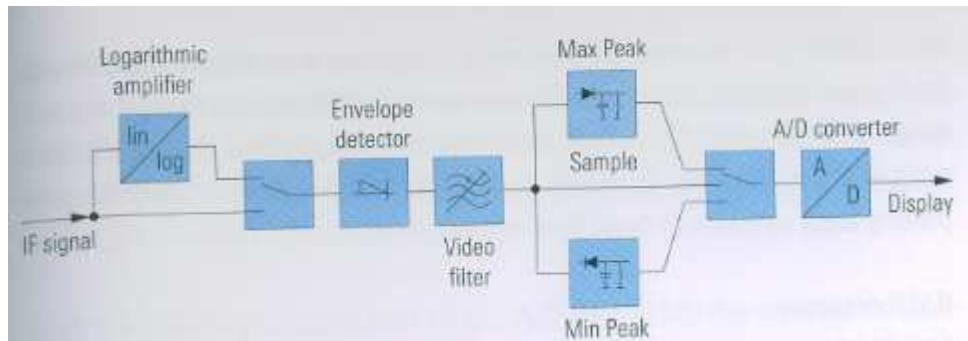
Analyzátor umožňuje vybrat funkci stopy z těchto možností:

- **CLEAR/WRITE** – tato funkce je nejjednodušší, každý sweep pouze přepíše sweep předchozí, tato funkce je nastavena defaultně
- **MAX HOLD** – je-li aktivovaná funkce max hold, hodnota se uloží do paměti pouze, když je nová hodnota větší než hodnota předchozí
- **MIN HOLD** – opak funkce maxMAX HOLD, je-li aktivovaná funkce min hold, hodnota se uloží do paměti pouze, když je nová hodnota menší než hodnota předchozí
- **VIEW** – zastaví průběh stopy a zobrazí ho (je-li stopa zmrazena, může se měnit nastavení přístroje, aniž by se měnila zobrazovaná stopa)
- **BLANK** – smaže zobrazenou stopu
- **AVERAGE** – spočítá průměrnou hodnotu pro každý bod stopy. Je-li aktivní mód single sweep, průměr se počítá ze všech sweepů v rámci jednoho single sweepu

1.6.6 Detektory

Na dnešních přístrojích je spektrum zobrazováno na LC displejích. Tyto displeje jsou značně limitovány rozlišením. Měřených hodnot bývá daleko, více než je počet pixelů, na kterých je spektrum zobrazováno. Zvláště při velkých hodnotách spanu obsahuje jeden pixel mnoho informací. Proto analyzátor nabízí možnost zvolit si, jakým způsobem bude přístroj hodnoty zobrazovat prostřednictvím detektorů. [4]

Základní typy detektorů jsou max peak, min peak, auto peak a sample.



Obr. 17 Blokové schéma detektorů [4]

1.6.6.1 MAX PEAK

Detektor max peak zobrazí největší hodnotu. Tento detektor vybere a zobrazí pixel s největší naměřenou hodnou. I v případě, že je signál zobrazen ve velmi malém rozlišení, nedojde ke ztrátě dat. Tento detektor je vhodný pro EMC měření.

1.6.6.2 MIN PEAK

Opak max peak detektoru – vybere a zobrazí nejmenší hodnotu

1.6.6.3 AUTO PEAK

Tento detektor měří největší i nejmenší hodnotu. Tyto hodnoty na displeji zobrazí a propojí čarou.

1.6.6.4 SAMPLE

Na každém pixelu se nachází více hodnot. Detektor sample vybere pouze jednu hodnotu vzorku, který se nachází na daném pixelu.

1.6.6.5 RMS

Detektor RMS vypočítá hodnotu pro každý pixel zobrazované stopy ze všech hodnot, které se na pixelu nachází. Výsledná hodnota se spočítá pomocí vzorce:

$$V_{RMS} = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^N v_i^2} \quad (16)$$

V_{RMS} – napětí

N – počet vzorků

v_i – vzorky

Výkon spočítáme pomocí referenčního odporu R:

$$P = \frac{V_{RMS}}{R} \quad (17)$$

1.6.6.6 AVERAGE

Tento detektor spočítá průměrnou hodnotu ze všech hodnot, které se nacházejí na daném pixelu. Využívá vzorce:

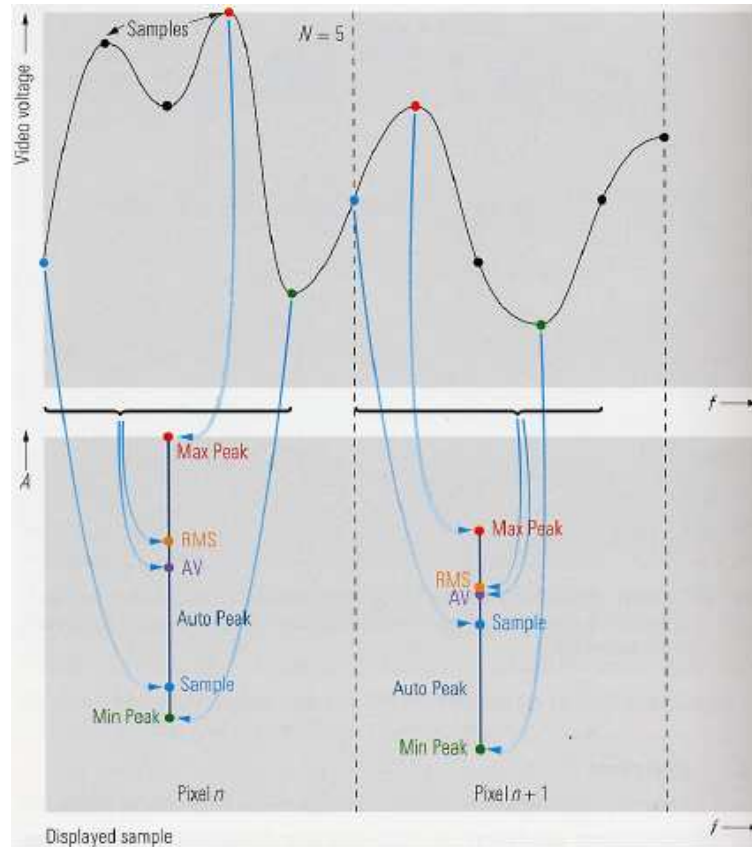
$$V_{AV} = \frac{1}{N} \cdot \sum_{i=1}^N v_i \quad (18)$$

Výkon se spočítá podle vzorce:

$$P = \frac{V_{RMS}}{R} \quad (19)$$

1.6.6.7 QUASI PEAK

Tento detektor se používá při aplikacích na měření rušení.



Obr. 18 Detektory [4]

2 VZDÁLENÉ OVLÁDÁNÍ

Pod pojmem vzdálené ovládání se rozumí možnost ovládat přístroje pomocí počítače. Přístroj je s počítačem spojen přes rozhraní, které je schopno přenášet data mezi těmito dvěma objekty. Ve své práci jsem používal rozhraní typu GPIB.

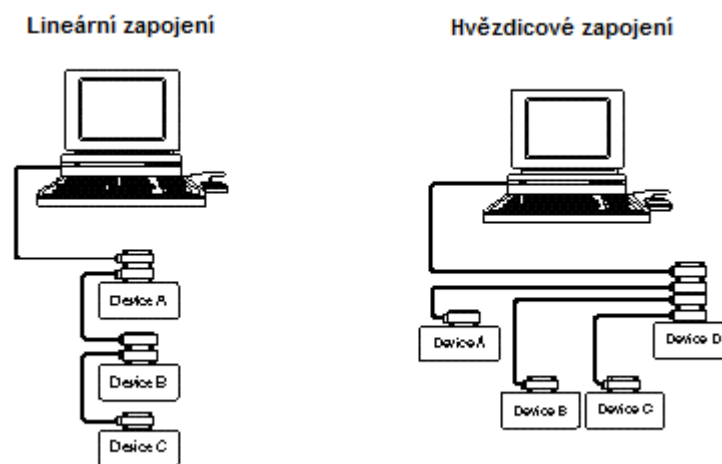
2.1 GPIB

GPIB (General Purpose Interface Bus) sběrnice slouží pro komunikaci zařízení na krátkou vzdálenost. GPIB rozhraní (jinak též IEEE-488) bylo vytvořeno v roce 1965 firmou Hewlett-Packard. Komunikace na GPIB sběrnici probíhá pomocí zpráv pro zařízení a zpráv pro rozhraní. Mezi zprávy pro zařízení (data) můžeme zařadit instrukce pro přístroje, výsledky měření nebo stav zařízení, zprávy pro rozhraní jsou příkazy pro sběrnici jako inicializace sběrnice, adresování přístrojů nebo nastavení vzdáleného ovládání. [7]

2.2 Základní parametry

Na jednu GPIB sběrnici lze připojit maximálně 15 zařízení najednou. Maximální rychlost přenosu je 1 MB/s (v praxi 250 – 500 kB/s). Maximální délka kabelu je 20 m a maximální vzdálenost mezi dvěma prvky jsou 2 m. GBPI využívá negativní TTL logiku, kde spodní napěťová úroveň (logická 1) je menší nebo rovna 0,8 V a horní napěťová úroveň (logická 0) je větší nebo rovna 2 V. [7]

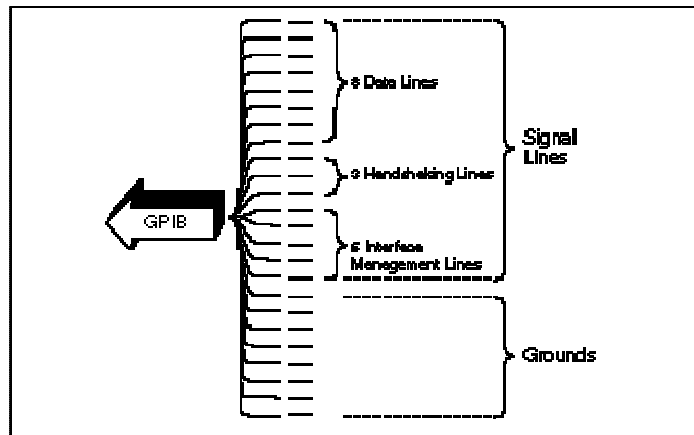
Zařízení lze propojit lineárně, hvězdicově nebo kombinací těchto dvou metod.



Obr. 19 Topologie [7]

2.3 Linky

Rozhraní GPIB se skládá z 24 vodičů. 8 vodičů je zemních, 8 datových, 5 vodičů pro řízení rozhraní a 3 vodiče pro řízení přenosu.



Obr. 20 GPIB linky [7]

2.3.1 Datové linky

Datové vodiče přenášejí data a příkazy. Zda se jedná o data nebo příkazy určuje stav linky ATN. Data a příkazy jsou většinou přenášena pomocí sedmi bitů DIO1 – DIO7, poslední bit DIO8 bývá nevyužit nebo udává paritu. [7]

2.3.2 Linky pro řízení přenosu

Linky pro řízení přenosu kontrolují správnost přenosu mezi zařízeními propojenými GPIB sběrnici.

NRFD (not ready for data) dává informaci o tom, zda zařízení je či není schopno přijmout data

NDAC (not data accepted) – indikuje, zda zařízení přijalo data či nikoliv

DAV (data valid) – udává, zda jsou data v takovém stavu, že je zařízení může bezpečně přijmout

2.3.3 Linky pro řízení rozhraní

Linky pro řízení rozhraní se starají o tok informací po sběrnici.

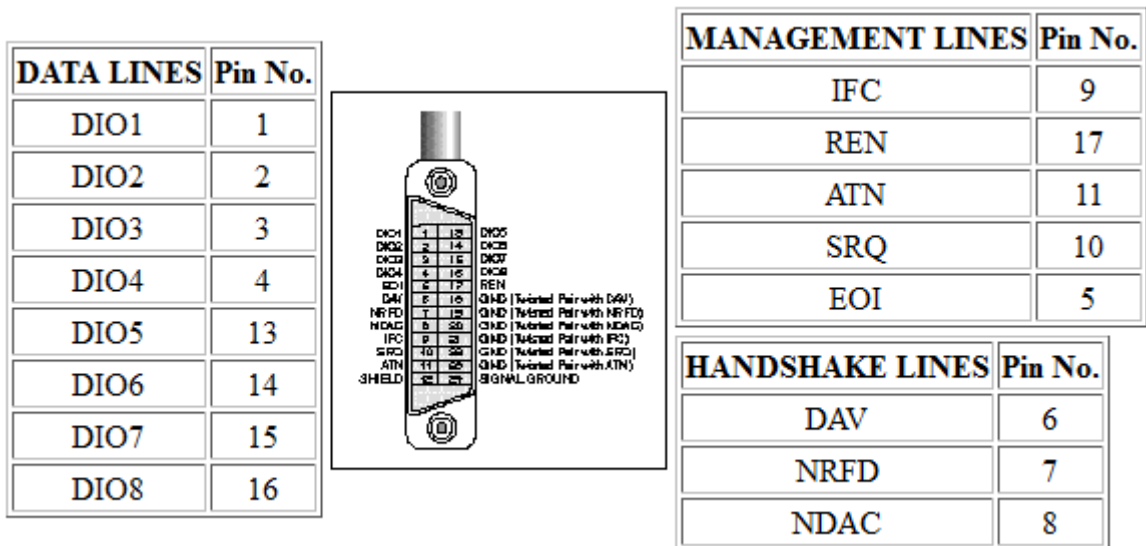
ATN (attention) – udává, zda se při přenosu jedná o data (FALSE) nebo o příkaz (TRUE)

IFC (interface clear) – linka je buzena při inicializaci sběrnice

REN (remote enable) - linka je buzena, když uživatel aktivuje místní nebo vzdálený mód ovládání zařízení

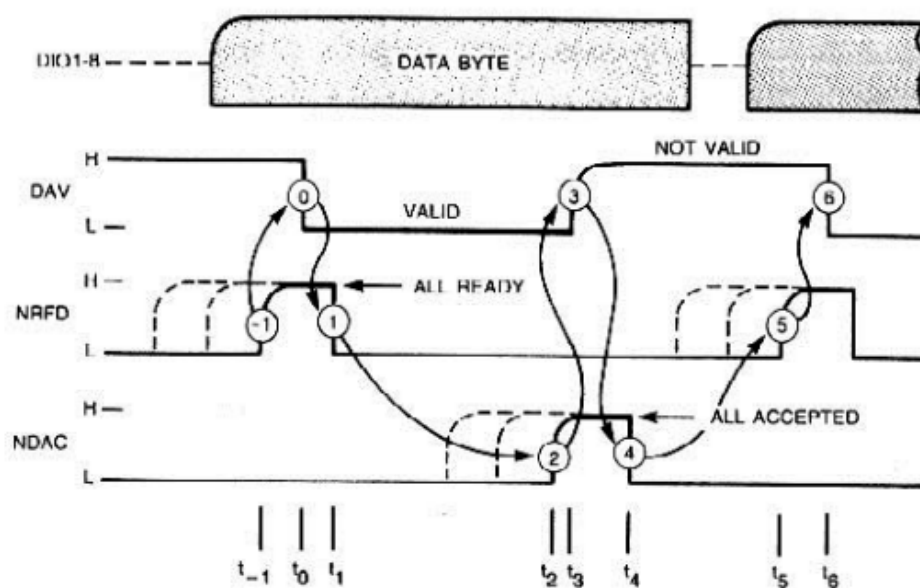
SQR (service request) – linka pro vyžádání obsluhy

EOI (end or identify) – tato linka slouží buď pro označení konce řetězce zprávy, nebo pro identifikaci hlášení



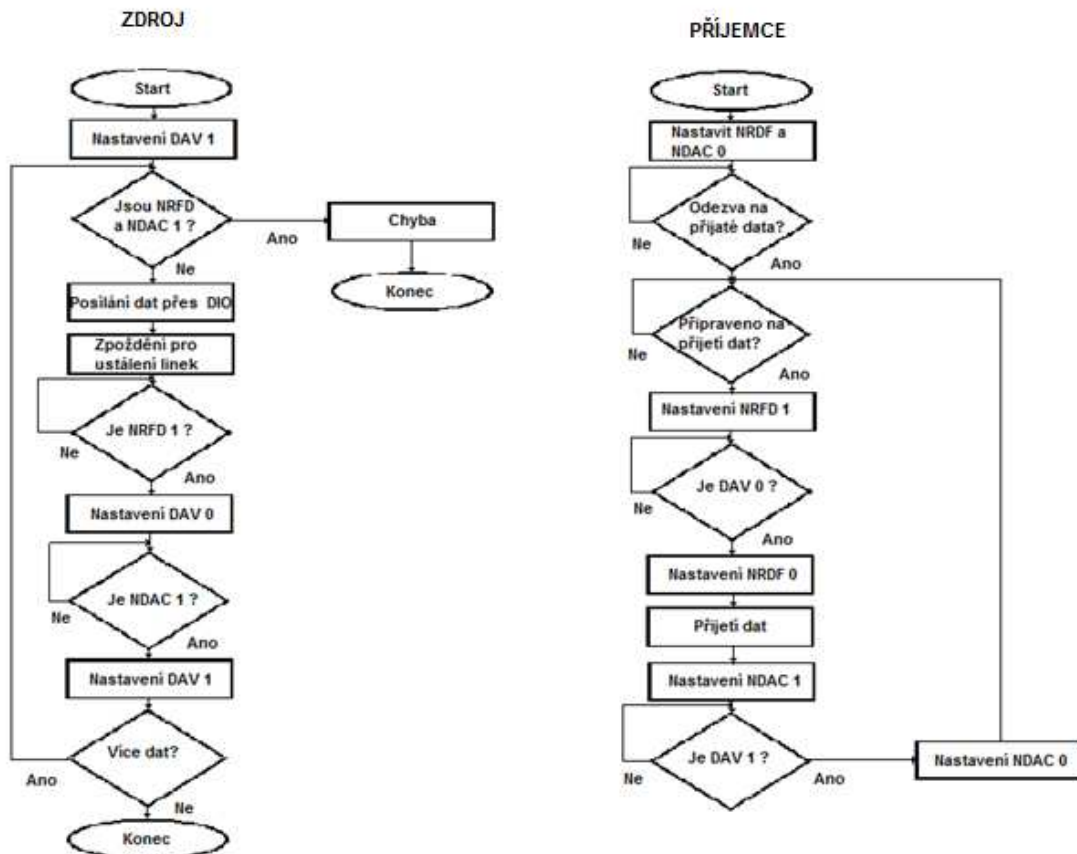
Obr. 21 Piny na GPIB [7]

2.4 Přenos



Obr. 22 Proces přenosu [8]

2.4.1 Algoritmus přenosu dat



Obr. 23 Algoritmus přenosu dat [8]

2.5 SCPI příkazy

Na přístroj jsou data posílána v podobě SCPI příkazů. Jedná se o sadu příkazů na programování přístrojů, přičemž nezáleží na typu přístroje nebo na výrobci přístroje.

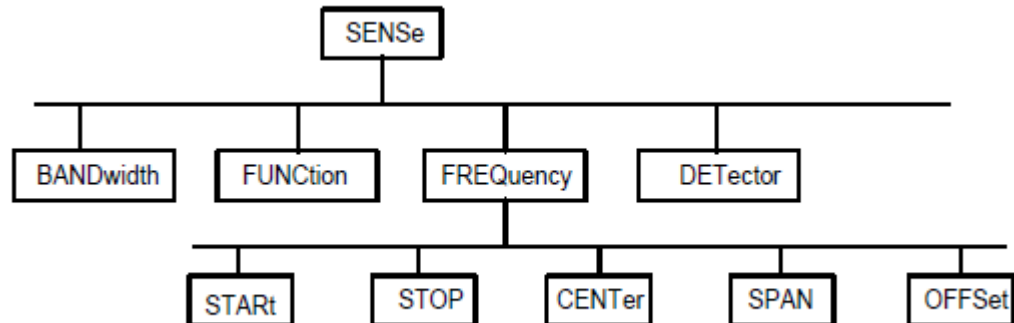
2.5.1 Struktura příkazu

SCPI příkaz se skládá z hlavičky a z parametrů (většinou jeden nebo dva), kdy hlavička a parametry jsou oddělené mezerou. Hlavička se většinou skládá z více částí. Jedná-li se o dotaz, je příkaz ukončen otazníkem.

Příkazy mohou být napsány v plné nebo ve zkrácené verzi. Zkrácená verze využívá pouze první čtyři písmena klíčového slova. [6]

Příklad:

SENS:FREQ:CENT 5GHZ



Obr. 24 Hierarchická struktura příkazů SCPI [6]

Parametr je od hlavičky oddělen mezerou. Nastane-li situace, že je parametrů více než jeden, jsou odděleny čárkou.

2.5.2 Parametry

Forma parametrů je pro každý příkaz určená zvlášť. Mohou být zadány jako text, číselná hodnota, řetězec znaků, Boolovy parametry nebo blok dat. Stejně jako forma parametrů, i rozpětí hodnot parametrů je specifické pro každý příkaz. [6]

2.5.2.1 Číselné parametry

Číselné parametry mohou být zadány v libovolném tvaru – desetinná tečka, vyjádření exponentem, se znaménkem. Skládá se z maximálně 255 znaků. Exponent je značen písmenem “E” a jeho hodnota se musí pohybovat v rozmezí -32000 až 32000. Hodnoty číselných parametrů také mohou být zadány s jednotkami K (kilo), G (giga), apod. [6]

Příklady:

SENS:FREQ:SPAN 5KHZ – zadání s jednotkou

SENS:FREQ:SPAN 5E3 – zadání s exponentem

Existují i speciální numerické parametry. Tyto parametry nejsou vyjádřeny číselně, ale textově. Patří mezi ně:

- MIN/MAX – maximální/minimální hodnota
- DEF – tento parametr načte přednastavenou hodnotu z paměti přístroje
- UP/DOWN – tyto parametry změni aktuální hodnotu o nastavený krok (délka kroku je zadána speciálním příkazem)
- INF/NINF – nekonečno/negativní nekonečno
- NAN – not a number, tento paramet je zasílán zařízením jako odpověď na neproveditelnou operaci (například dělení nulou)

2.5.2.2 *Boolovy parametry*

Tyto parametry reprezentují logické hodnoty 0 a 1 a jsou reprezentovány jako ON OFF.

2.5.2.3 *Textové parametry*

Textové parametry mohou být zadávány krátce či dlouze (podobně jako hlavička). Pokud se jedná o odezvu zařízení, je v krátké verzi.

2.5.2.4 *Znakové řetězce*

Znakové řetězce jsou ohraničeny uvozovkami (“ ” nebo ‘ ‘)

2.5.2.5 *Bloky dat*

Tato forma se používá, pokud je potřeba přenést velký objem dat. Parametr začíná znakem #. První číslice udává, kolik následujících číslic určuje velikost přenášených dat v bitech. Nakonec následuje blok dat. [6]

Příklad:

HEADer:HEADer #3584xxx.....

Počet číslic určujících délku datového bloku – 3

Délka datového bloku – 584 b

2.6 Dělení příkazů

Příkazy dělíme na dvě skupiny – běžné příkazy a příkazy pro specifické zařízení.

2.6.1 Běžné příkazy

Hlavička těchto příkazů začíná hvězdičkou, po které následuje jedno klíčové slovo a jeden nebo žádný parametr.

Příklady:

**RST* – tento příkaz resetuje zařízení

**ESE 244* – nastaví bity registru EVENT STATUS ENABLE

**ESR?* – pošle dotaz na obsah registru EVENT STATUS REGISTER

2.6.2 Příkazy pro specifické zařízení

Struktura těchto příkazů je hierarchického typu. Hlavička je složena z jednoho nebo více klíčových slov oddělených dvojtečkou. Klíčové slovo odpovídá vždy určité skupině příkazů – klíčové slovo *FREQuency* odpovídá skupině příkazů spojených s frekvencí. Pořadí klíčového slova reprezentuje jeho úroveň. Čím nižší je úroveň příkazu, tím je hlavička delší. [6]

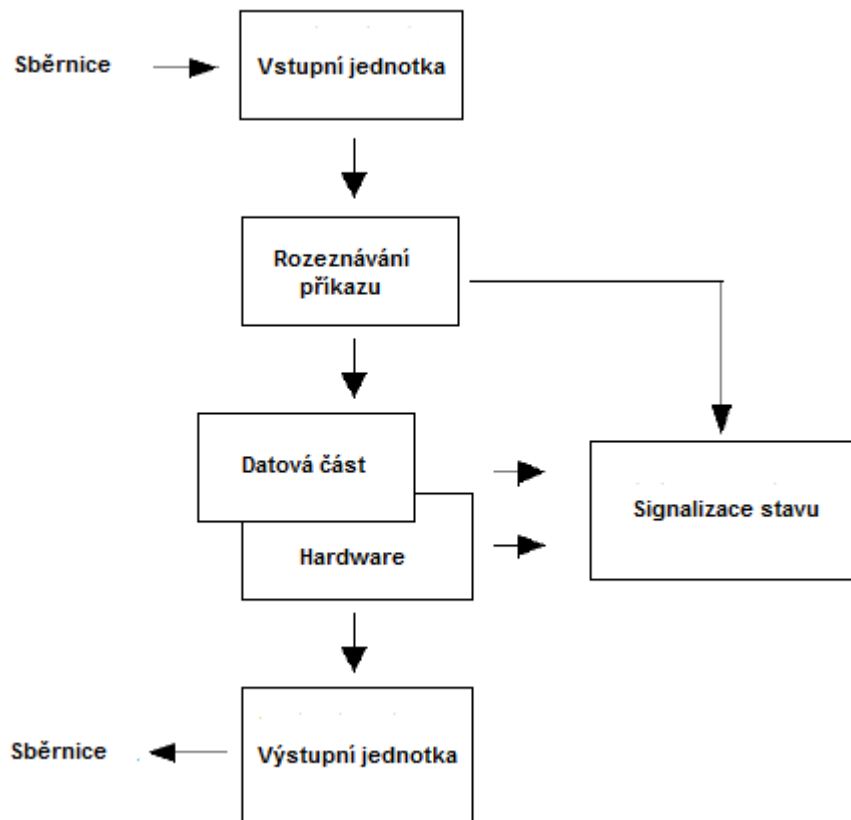
Příklad:

SENSe:FREQuency:CENTer 5GHZ - hlavička tohoto příkazu má tři úrovně.

Tab 1 Přehled znaků v SCPI příkazech

Znak	Funkce
:	odděluje klíčové slova hlavičky
;	odděluje dva příkazy v příkazovém řádku
,	odděluje dva parametry jednoho příkazu
?	příkaz je dotazem
*	označuje běžný příkaz
"	označuje parametr ve formě řetězce znaků
#	označuje začátek datového bloku

2.7 Zpracování příkazů



Obr. 25 Proces zpracování příkazu [6]

2.7.1 Popis toku dat

2.7.1.1 Vstupní jednotka

Vstupní jednotka přijímá znaky, co přicházejí po sběrnici. Tyto znaky se po jednom postupně ukládají do zásobníku. Pokud se zásobník naplní nebo pokud jednotka přijme znaky označující ukončení zprávy, vstupní jednotka pošle informace do bloku rozpoznávání příkazů. Pokud je zásobník plný, vstupní jednotka další data nepřijímá a musí se počkat, než se data zpracují. Obdrželi-li jednotka ukončení zprávy a zásobník plný není, může přijímat další data. [6]

2.7.1.2 Rozeznávání příkazu

Blok rozeznávání příkazu přijímá data od vstupní jednotky a analyzuje je. Rozpoznané příkazy jsou posílány na datovou základnu přístroje. Nalezne-li chybu v příkazu, zašle informaci na blok signalizace stavu a zbytek, je-li to možné, zpracuje. Blok je schopný okamžitě po odeslání příkazu na datovou základnu zpracovávat příkaz nový. S daty je pracováno v takovém pořadí, v jakém dorazily. [6]

2.7.1.3 Datová část a hardware

Datová část přijímá příkazy a preposílá je na hardwarovou část k vykonání (například nastavení frekvence). Předtím než jsou data poslána hardwaru, dochází k ověření jejich kompatibility. Dojde-li k závěru, že příkaz provést nelze, pošle hlášení o chybě na signalizaci stavu. Pokud je na blok zaslán dotaz, datová část zašle odpověď na výstupní jednotku. [6]

2.7.1.4 Signalizace stavu

Signalizace stavu sbírá informace o stavu zařízení z jiných bloků, a pokud je požádán, zašle tyto informace na výstupní jednotku. [6]

2.7.1.5 Výstupní jednotka

Výstupní jednotka sbírá data od datové základny a podle zadaných příkazů neukládá do zásobníku a posílá na sběrnici. Pokud přijatý příkaz nepožaduje výstupní hodnotu nebo pokud ještě čeká na data od datové části přístroje, jednotka zašle na signalizaci stavu zprávu "Query UNTERMINATED".[6]

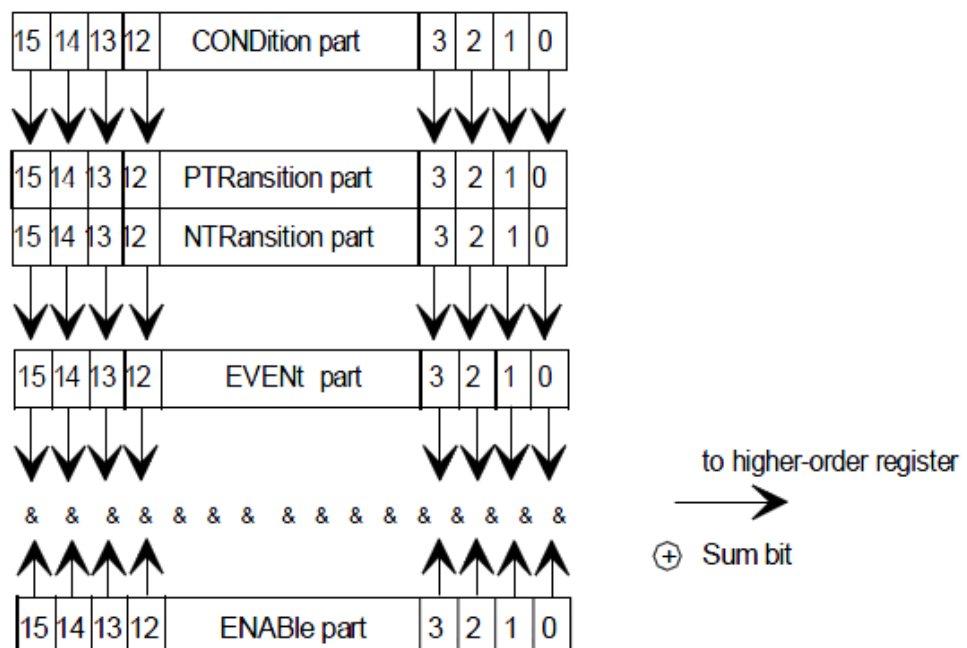
2.8 Systém hlášení stavu přístroje

Systém hlášení stavu podává informace o tom, v jakém stavu se momentálně nachází zařízení s nímž pracujeme. Informace o přístroji jsou zaznamenávány do status registrů. Tyto registry mají hierarchickou strukturu. Na vrcholu této struktury se nachází status byte (STB), jehož je maskou registr service request enable (SRE). Registr STB přijímá informace od event status registru (ESR), jehož maskou je registr event status enable (ESE) a od registrů STATus:OPERation a STATus:QUEStionable. Objekty STB, SRE, ESR a ESE jsou definovány standartem IEEE 488.2, registry STATus:OPERation a

STATUS:QUESTIONable jsou definovány SCPI. Celá hierarchická struktura status registrů je zobrazena v příloze P I. [6]

2.8.1 Struktura SCPI registrů

Každý SCPI status registr se skládá z pěti částí a každá tato část má 16 bitů a vykonává jinou funkci. Tyto bity jsou na sobě nezávislé a každý bit má svoji specifickou funkci, která je stejná pro každou část. Bit číslo 15 je pro každou část nastaven na 0. [6]



Obr. 26 Model status registru [6]

2.8.1.1 Část CONDition

Tato část je určena přímo hardwarem nebo sumarizačním bitem nižšího registru. Podává informace o stavu přístroje. Tato část je určena jenom pro čtení, nejde do ní zapisovat ani ji smazat. Po přečtení se nijak nemění. [6]

2.8.1.2 Část PTRansition

Část Positive-transition složí k tomu, že dojde-li ke změně bitu v části CONDition z 0 na 1, podle toho, jestli je nastaven příslušný bit v této části, nastaví se bit v části EVENT na 1. Z této části může být čteno a může se do ní i zapisovat. [6]

PTR bit = 1: EVENT je nastaven

PTR bit = 0: EVENT není nastaven

2.8.1.3 Část *NTR*ransition

Tato část funguje stejně jako část Positive-transition s tím rozdílem, že ke změně bitu v *EVENT* části (je-li nastavený příslušný bit v části Negative-transition) dojde v případě, že se změní bit v části *COND*ition z 1 na 0. [6]

NTR bit = 1: *EVENT* je nastaven

NTR bit = 0: *EVENT* není nastaven

2.8.1.4 Část *EVENT*

Tato část indikuje, jestli došlo ke změně od stavu posledního čtení, jedná se jakousi paměť části *COND*ition a je neustále přístrojem aktualizována. Slouží pouze ke čtení a pokud dojde k přečtení, data se vynulují. [6]

2.8.1.5 Část *ENABLE*

Bity této části určují, zda příslušné bity z části *EVENT* ovlivňují sumarizační bit. Ke každému bitu v části *EVENT* je přiřazen příslušný bit funkcí AND. Výsledek této logické operace je pak přiřazen k sumarizačnímu bitu funkcí OR. Z této části může být čteno a může se do ní i zapisovat. [6]

ENABLE-Bit = 0: *EVENT* bit neovlivní sumarizační bit

ENABLE-Bit = 1: pokud je *EVENT* bit 1 sumarizační bit je také nastaven na 1

2.8.1.6 Sumarizační bit

Slouží k nastavení *COND*ition části vyššího registru.

2.8.2 Status byte a service request enable registr

STB podává informace o stavu zařízení. Můžeme tedy říct, že se jedná o ekvivalent části *COND*ition u *SCPI* registrů. Informace o stavu zařízení shromažďuje od registrů nižších úrovní. Jeho šestý bit slouží jako sumarizační pro ostatní bity status bytu. Pro přečtení *STB* používáme příkaz "**STB?*" [6]

Funkce status enable registr je ekvivalentní *SCPI* části *ENABLE*. Každý bit *SRE* je přiřazen bitu *STB*, s výjimkou bitu číslo 6. Pokud je bit *SRE* nastaven a příslušný bit *STB* se změní z 0 na 1, na sběrnici se vygeneruje service request (*SRQ*). *SRE* může být nastaven příkazem "**SRE*", příkaz pro čtení je "**SRE?*". [6]

Tab 2 Popis bitů status byte [6]

Číslo bitu	Název bitu	Funkce
2	Error Queue not empty	Bit je nastaven, pokud dojde k zápisu do chybové řady. Pokud je tento bit povolen SRE, každý zápis do chybové řady generuje Service request. Každá chyba tak může být rozpoznána a blíže specifikována.
3	QUESTIONable status sum bit	Bit je nastaven, pokud je nastaven bit v EVENT části QUESTIONable status registru a bit části ENABLE je 1. Nastavený bit indikuje že něco není v pořádku s přístrojem. Pro bližší specifikování problému je třeba prozkoumat QUESTIONable status registr.
4	MAV bit	Tento bit je nastaven, pokud je na výstupním zásobníku zpráva, kterou je možno číst. Může být použit pro automatické čtení z přístroje.
5	ESB bit	Sumarizační bit event status registru. Je nastaven, pokud je nastaven jeden z bitů event status registru a je povolen v event status enable registru. Jeho nastavení značí chybu nebo událost. Pro bližší specifikování je třeba prozkoumat event status registr.
6	MSS bit	Master status summary bit je nastaven, pokud přístroj vygeneruje service request. To nastane pokud se nastaví jiný bit tohoto registru a je povolen maskou SRE.
7	OPERation status register sum bit	Bit je nastaven, pokud je nastaven bit v EVENT části OPERation status registru a bit části ENABLE je 1. Nastavený bit indikuje že přístroj vykonal akci. Pro bližší specifikování této činnosti je třeba prozkoumat OPERation status registr.

2.8.3 Event status register

Tento registr může být chápán jako část EVENT v SCPI registrech. Pro čtení z tohoto registru používáme příkaz "*ESR?". Jeho maskou je registr event status enable, který je tak ekvivalentem části ENABLE v SCPI registrech. Pro nastavení jeho bitů používáme příkaz "*ESE", pro čtení používáme příkaz "*ESE?". [6]

Tab 3 Popis bitů event status registru [6]

Číslo bitu	Název bitu	Funkce
0	Operation Complete	Tento bit je nastaven poté, co byly vykonány všechny předchozí příkazy.
1		Tento bit se nevyužívá.
2	Query Error	Bit se nastaví, pokud chce kontroler číst data z přístroje bez toho, aby poslal dotaz nebo pokud místo požadovaných dat dostane nové instrukce. Většinou bývá problém v příkazu, který je chybný a nemůže být vykonán.
3	Device-dependent Error	Bit se nastaví, pokud dojde k chybě zařízení. Do chybové řady je zaslána zpráva o chybě, kterou blíže specifikuje k ní přiřazené číslo.
4	Execution Error	Tento bit se nastaví, pokud je syntaxe přijatého příkazu v pořádku, ale nemůže být vykonán z jiného důvodu. Do chybové řady je zaslána zpráva o chybě, kterou blíže specifikuje k ní přiřazené číslo.
5	Command Error	Tento bit se nastaví, pokud je přijat příkaz, který nelze identifikovat nebo má špatnou syntaxi. Do chybové řady je zaslána zpráva o chybě, kterou blíže specifikuje k ní přiřazené číslo.
6	User Request	Bit je nastaven zmáčknutím klávesy LOCAL.
7	Power On	Bit se nastaví, pokud je přístroj napájen.

2.8.4 STATus:OPERation registr a STATus:QUEStionable registry

Jedná se o SCPI registry. STATus:OPERation registr obsahuje informace jaké akce se právě na přístroji provádějí a může být na ně dotazován. Registrů STATus:QUEStionable je mnoho a v každém je možno zjistit, zda-li nenastala určitá situace specifikovaná registrem a bitem (například provedené měření bylo nekalibrované). [6]

2.9 Aplikace systému hlášení stavu

Aby bylo možno efektivně nakládat s tímto systémem, je nutno informace v něm obsažené posílat na kontroler, kde mohou být dále zpracovány.

2.9.1 Service Request

Za určitých okolností může přístroj poslat na kontroler Service request. To ve většině případů vyvolá přerušení, na které kontrolní program reaguje danými akcemi. Service request je vždy vyvolán bitem číslo 2, 3, 4, 5 nebo 7 status bytu. Každý z těchto bitů obsahuje informaci od dalšího registru, chybové řady nebo výstupního zásobníku. [6]

Příklad:

`CALL IBWRT(analyzer%, "*ESE 1")` - nastaví bit 0 v ESE (operace dokončena)

*CALL IBWRT(analyzer%, "*SRE 32")* – nastaví bit 5 v SRE (ESB)

Po vykonání těchto příkazů přístroj vygeneruje Service request.

2.9.2 Zápis chybové řady

Pokaždé, když nastane chybový stav přístroje, dojde k zápisu do chybové řady. Detailní zprávu o chybě je možno prohlédnout pomocí příkazu "SYSTem:ERRor?". Při každém dotazu dostaneme odpověď z chybové řady. Pokud k žádné chybě nedošlo, zařízení pošle odpověď 0, "No error".[6]

II. PRAKTICKÁ ČÁST

3 SKALÁRNÍ MĚŘENÍ VÝKONOVÉHO PŘENOSU V ZÁVISLOSTI NA KMITOČTU

Cílem praktické části mé diplomové práce bylo vytvořit grafickou aplikaci pro skalární měření výkonového přenosu v závislosti na kmitočtu. Tuto aplikaci jsem vytvořil v programu MATLAB, verze 7.1.0.246 (R14) Service Pack 3, 2005.

3.1 Funkce programu

Hlavní funkce programu spočívá v ovládní dvou měřících přístrojů a to generátoru signálu SMR – 20 a frekvenčního analyzátoru FSP – 40. Přístroje je možno ovládat pomocí grafického rozhraní. Výsledky měření se v průběhu měření do grafu vykreslují do grafu. Zadané parametry měření je možno ukládat a načítat pomocí jednoduchého menu.

3.2 Struktura programu

Hlavní program se skládá ze dvou částí: Připojení přístrojů, nastavení hodnot parametrů přístrojů a měření.

3.2.1 Připojení přístrojů

Přístroje jsou propojeny pomocí rozhraní GPIB. Než dojde k samotnému připojení na přístroje, je třeba provést detekci adres, na kterých jsou oba přístroje. Tato část programu se aktivuje pomocí tlačítka **Detekce adres**. Program načte adresy pomocí funkce *instrhwinfo*. Tato funkce poskytuje podrobné informace o připojených rozhraních a v její této struktuře je možno najít adresy, kde jsou přístroje připojeny. Program rozezná, která adresa náleží generátoru a která analyzátoru a nakonec je uloží do proměnných *adresa1* a *adresa2* (*boardindex1* a *boardindex2*). Samotné připojení programu na přístroje se provede až po stisknutí tlačítka **Start**.

Nalezení adres přístrojů:

```
adr=instrhwinfo('gpi','agilent');  
s1=char(adr.ObjectConstructorName(1));  
s2=char(adr.ObjectConstructorName(2));
```

Připojení počítače na přístroj probíhá pomocí funkce *gpi*, který vytvoří objekt rozhraní. Tato funkce má tři argumenty – *vendor boardindex* a *primaryaddress*.

- **vendor** – výrobce připojeného rozhraní, v našem případě *agilent*

- **boardindex** – číslo GPIB desky
- **primaryaddress** – primární adresa samotného přístroje

Poté co je vytvořen objekt rozhraní, je třeba ho aktivovat příkazem *fopen()*.

Příklad:

```
g=gplib('agilent', handles.boardindex2, handles.adresa2);  
fopen(g);
```

3.2.2 Nastavení přístrojů

Nastavení jednotlivých parametrů měření je provedeno uživatelem pomocí grafického rozhraní nebo je automaticky zadáno programem. Program sám nastavuje automaticky tyto parametry:

Generátor SMR – 20: aktivace RF

Analyzátor FSP – 40: atenuátor, jednotky výkonu, nastavení displeje, hodnota VBW, TRACE, detektor, počet bodů sweepu

Parametry nastavené uživatelem:

Generátor SMR – 20: rozpětí měřených frekvencí, výkon

Analyzátor FSP – 40: referenční úroveň výkonu, doba jednoho sweepu, počet sweepů v rámci jednoho single sweepu, span, RBW

Nastavení přístrojů se provede v momentu stisknutí tlačítka **Start** po tom, co se aktivuje vzdálené ovládání přístroje. Toto nastavení je na zařízení doručeno ve formě SCPI příkazů.

3.2.3 Měřící část

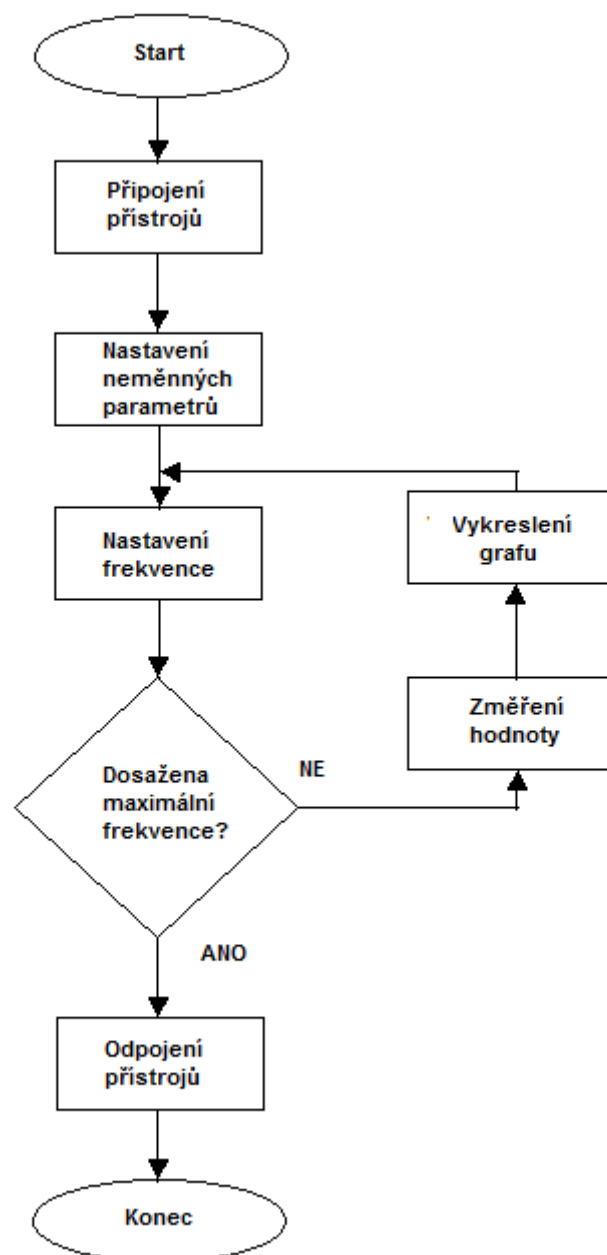
Tato část programu automaticky následuje po předchozí části nastavení parametrů přístrojů. Měření probíhá v nastaveném rozsahu frekvencí a cyklicky se opakuje v závislosti na délce kroku, která je určena uživatelem. Program nejprve pošle příkaz na generátor a nastaví jeho frekvenci. Poté zašle příkaz pro k tomu ekvivalentní nastavení střední frekvence analyzátoru. Analyzátor změří výkon tolikrát, kolikrát je nastaveno v poli počet průměrů. Z těchto naměřených hodnot poté vypočítá průměrnou křivku. Nastaví marker na maximální hodnotu křivky, hodnotu přečte a uloží do proměnné y. Tuto operaci cyklicky opakuje s hodnotu frekvence zvýšenou o velikost kroku až do dosažení zadané

maximální frekvence. Hodnoty frekvence (proměnná x) a výkonu (proměnná y) se průběžně vykreslují do grafu.

Na konci procesu měření je nutno oba přístroje odpojit a smazat námi vytvořené objekty.

```
fclose(g)
delete(g)
clear g
```

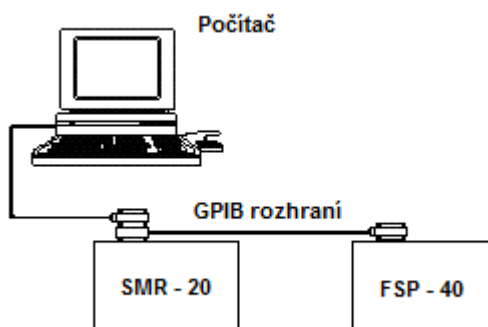
3.3 Algoritmus programu



Obr. 27 Algoritmus programu

3.4 Vzdálené ovládání

Oba přístroje jsou s počítačem propojeny pomocí GPIB rozhraní. Přes tuto sběrnici jsou přístroje ovládány pomocí příkazů SCPI (Standard Commands for Programmable Instruments). Přístroje aktivují režim vzdáleného ovládání v momentu, kdy se počítač programově připojí na přístroj - provede se příkaz *fopen()* a status přístroje se změní z *closed* na *open*.



Obr. 28 Zapojení

Jednotlivé příkazy na ovládání přístrojů jsou na zařízení posílány pomocí funkce *fprintf()*, čtení z přístroje se provádí pomocí funkce *fscanf()*.

Příklad:

```
fprintf(a, 'INP:ATT:AUTO ON');
```

Prvním argumentem funkce *fprintf* je název zařízení, na které je příkaz adresován (v tomto příkladu *a*), druhým argumentem je pak samotný příkaz (v tomto příkladu 'INP:ATT:AUTO ON'). Podobu příkazů je nutno vyhledat v operačním manuálu přístroje.

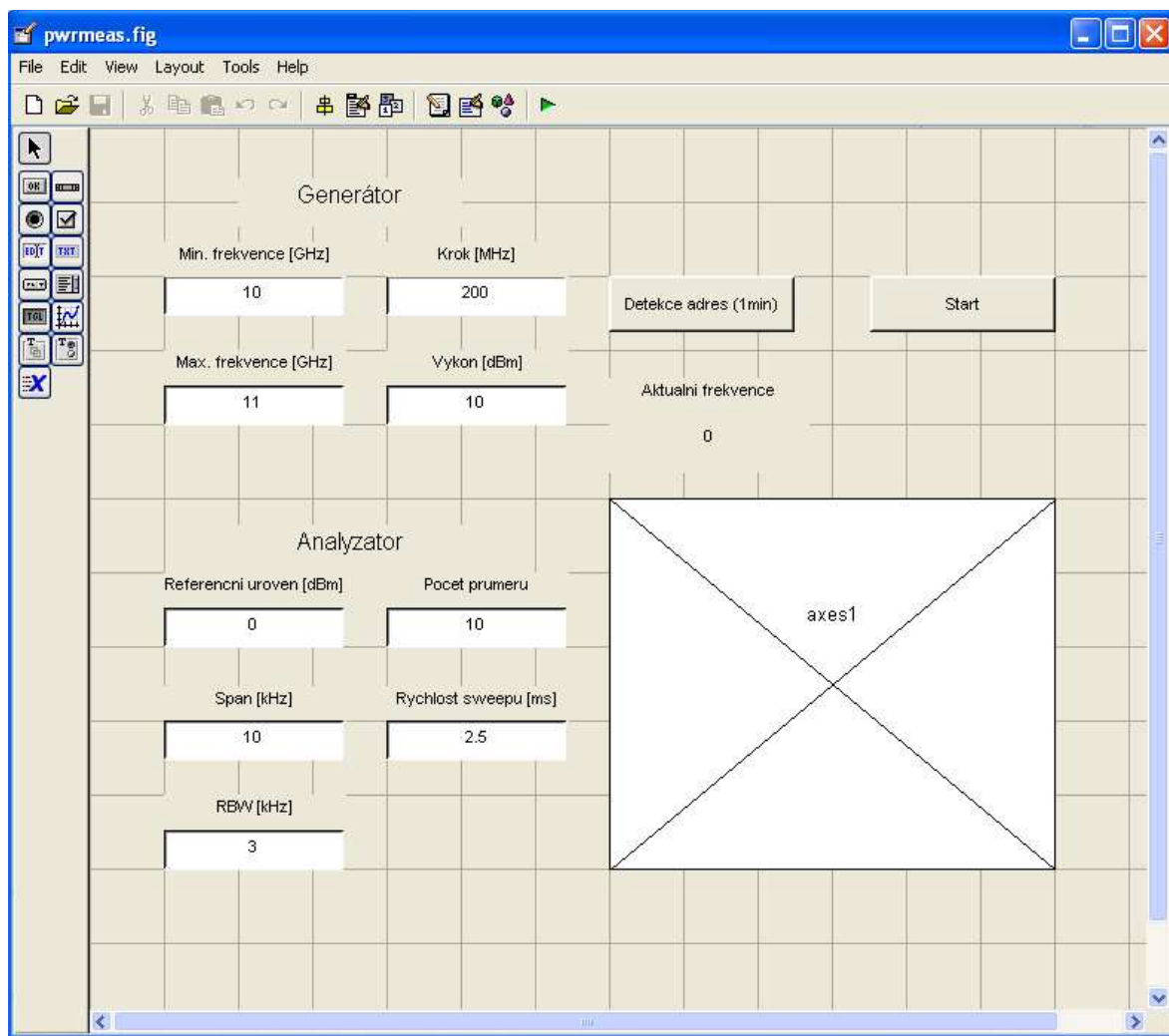
Seznam použitých SCPI příkazů:

Tab 4 SCPI příkazy

Příkaz	Popis
Analyzátor FSP - 40	
INP:ATT:AUTO ON	automatické nastavení atenuátoru
CALC:UNIT:POW dBm	jednotky výkonu
DISP:WIND:TRAC:Y:RLEV x dBm	nastavení referenčního výkonu
DISP:WIND:TRAC:Y:SPAC LOG	nastavení displeje
DISP:WIND:TRAC:Y 100 dB	
FREQ:SPAN X kHz	nastavení spanu
BAND:AUTO: OFF	automatické nastavení RBW
BAND X Khz	nastavení RBW
BAND:VID X kHz	nastavení VBW
SWE:TIME X ms	nastavení sweeptime
DISP:WIND:TRAC:MODE X	nastavení trace
DET X	nastavení detektoru
SWE:POIN X	nastavení počtu bodů sweepu
SW:COUN X	nastavení počtu sweepů
INIT:CONT:OFF	vypnutí kontinuálního sweepování
INIT:CONM	single sweep
FREQ:CENT: X	nastavení střední frekvence
CALC:MARK1:MAX	marker 1 na maximaální hodnotu
CALC:MARK1:Y?	načtení markeru
Generátor SMR - 20	
:OUTP ON	aktivace RF
:SOUR:POW X dBm	nastavení výkonu
:SOUR:FREQ X	nastavení frekvence

3.5 GUI

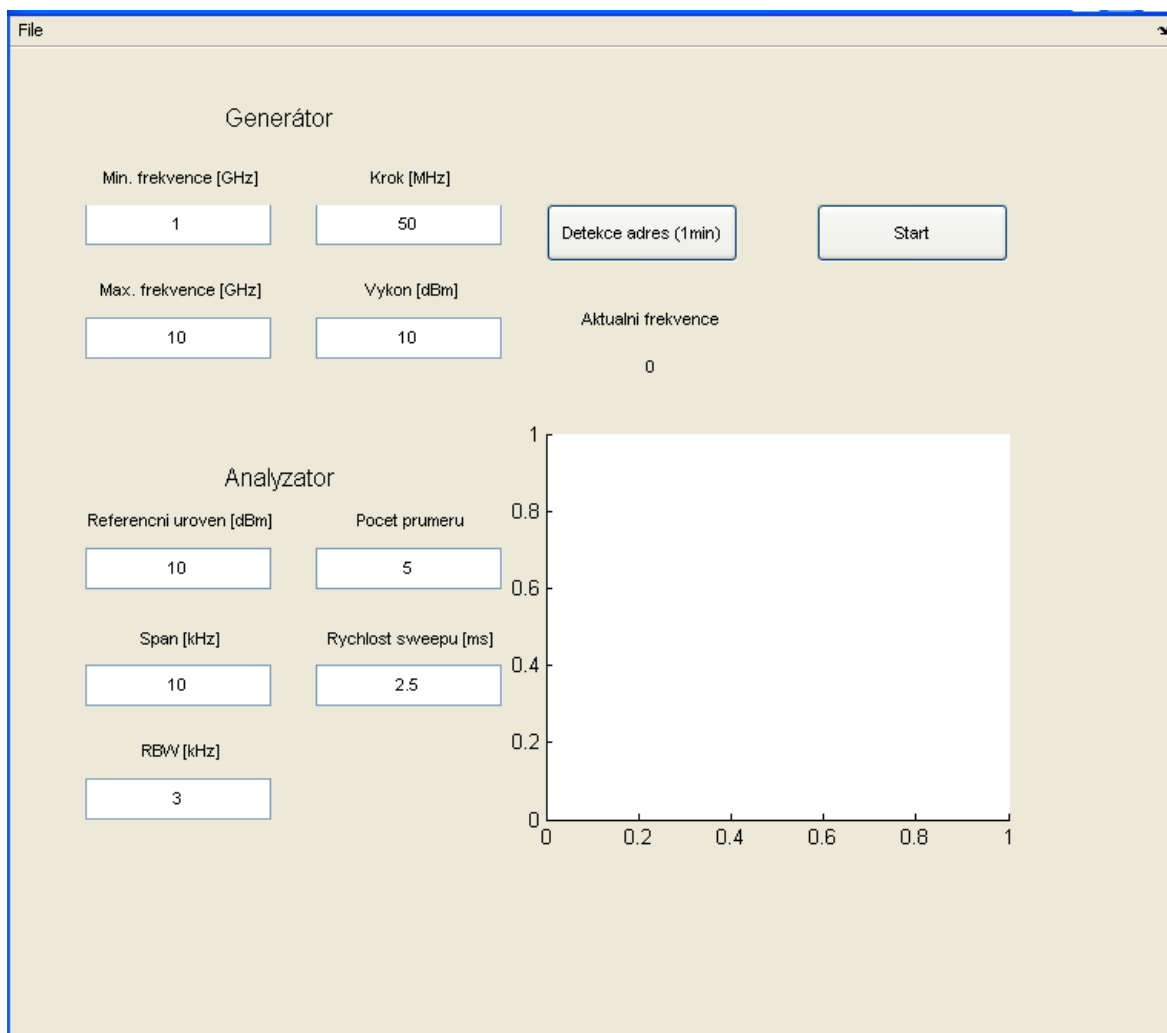
Grafické rozhraní jsem vytvořil pomocí nástroje GUIDE. Tento nástroj umožňuje vytvořit grafickou podobu GUI přímo v editoru. GUIDE také sám generuje callbacky pro grafické objekty.



Obr. 29 GUIDE

Na obrázku 29 je možno vidět zobrazení grafického rozhraní v editoru GUIDE. V tomto editoru je možno vytvářet objekty pomocí postranní lišty. Parametry těchto objektů je možno nastavit property inspector pomocí jako. Je zde možno učít tag objektu, zobrazovaný text, velikost textu, barvu objektu.

3.5.1 Rozložení GUI



Obr. 30 GUI

3.5.1.1 Nastavení parametrů přístrojů

V rámci GUI je možno manuálně zadávat hodnoty generátoru a analyzátoru. Tyto hodnoty zadáváme do oken na levé straně v příslušné sekci pro každý přístroj.

Generátor SMR - 20 :

Minimální frekvence – hodnota minimální frekvence udává, na které frekvenci začne proces měření, zadává se v jednotkách GHz

Maximální frekvence – hodnota maximální frekvence udává, na které frekvenci proces měření skončí, zadává se v jednotkách GHz

Krok – hodnota kroku udává, o kolik se změní frekvence na generátoru a střední frekvence na analyzátoru při proběhnutí jednoho cyklu měření, zadává se v jednotkách MHz

Výkon – výkon generátoru, zadává se v jednotkách dBm

Analyzátor FSP – 40:

Referenční úroveň výkonu – tato hodnota udává referenční výkon analyzátoru, zadává se v jednotkách dBm

Span - hodnota spanu analyzátoru, zadává se v jednotkách kHz

RBW – hodnota RBW udává šířku filtru (Resolution Bandwidth), zadává se v jednotkách kHz

Počet průměrů – tato hodnota udává tzv. sweepcount, což je počet, kolikrát proběhne měření jednoho sweepu při jednom cyklu měření

Rychlost sweepu – tato hodnota udává, za jak dlouho proběhne jeden sweep, zadává se v jednotkách ms

Pokud uživatel nezadá žádné hodnoty, budou použity defaultní hodnoty, které jsou načteny ze souboru *load_default.m*.

Se zadanými parametry se v rámci GUI pracuje pomocí struktury handles. Tato struktura umožňuje přenášet proměnné mezi callbacky různých objektů.

3.5.1.2 Tlačítka

V pravé části na hoře jsou dvě tlačítka. Tlačítko **Start** spustí celý proces měření, tlačítko **Detekce adres** slouží k načtení adres připojených přístrojů. Proces načtení adres trvá 1 – 2 minuty.

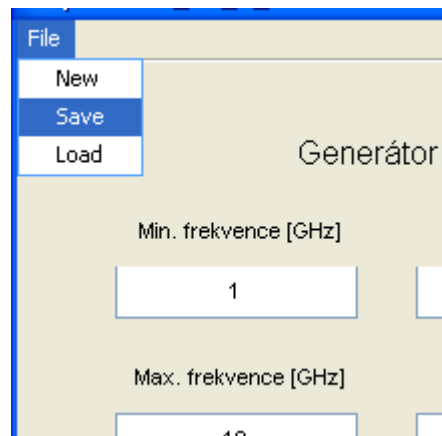
Pod těmito dvěma tlačítky je pole, které ukazuje aktuální hodnotu frekvence měřeného signálu v průběhu měření.

3.5.1.3 Graf

Vpravo dole se nachází graf, který vykresluje naměřené hodnoty výkonu v závislosti na měnící se frekvenci. Osa x grafu se přizpůsobí podle zadaných hodnot minimální a maximální frekvence, osa y je v rozsahu 0 až -100 dBm. Program hodnoty do grafu vykresluje průběžně v každém kole cyklu procesu měření.

3.5.2 Menu

V menu programu jsou možné tři volby: *New*, *Save* a *Load*. Toto menu slouží k ukládání a načítání parametrů, které jsme nastavili. Možnost *New* načte defaultní hodnoty ze souboru *load_default.m*. Možnost *Save* uloží parametry do libovolného souboru s příponou **.m* a možnost *Load* tyto parametry ze zvoleného souboru načte.



Obr. 31 Menu

3.5.2.1 *Load_defaults*

Funkce *load_defaults* slouží k nastavení defaultních parametrů měření. Využívá pouze dva funkce – *setappdata* a *set*. Pomocí funkce *setappdata* program uloží aplikační data do struktury handles. Funkce *set* poté zobrazí hodnotu v příslušném poli.

Příklad:

```
setappdata(main_handle, 'fmin', 1);  
set(findobj('Tag', 'freqmin'), 'String', num2str(1));
```

3.5.2.2 *Load*

Načtení parametrů probíhá stejně jako u funkce *load_defaults* pomocí příkazů *setappdata* a *set* s tím rozdílem, že hodnoty parametrů nejsou pevně dané, ale načítají se ze zvoleného souboru.

3.5.2.3 *Save*

Tato funkce ukládá nastavení parametrů do zvoleného souboru. Užívá k tomu dvou funkcí – *get* a *setappdata*. *Get* načte hodnoty z příslušných polí grafického rozhraní a tyto hodnoty jsou následně uloženy funkcí *setappdata*.

Příklad:

```
freqmin=get(findobj('Tag','freqmin'),'String');  
setappdata(main_handle, 'freqmin',freqmin);
```

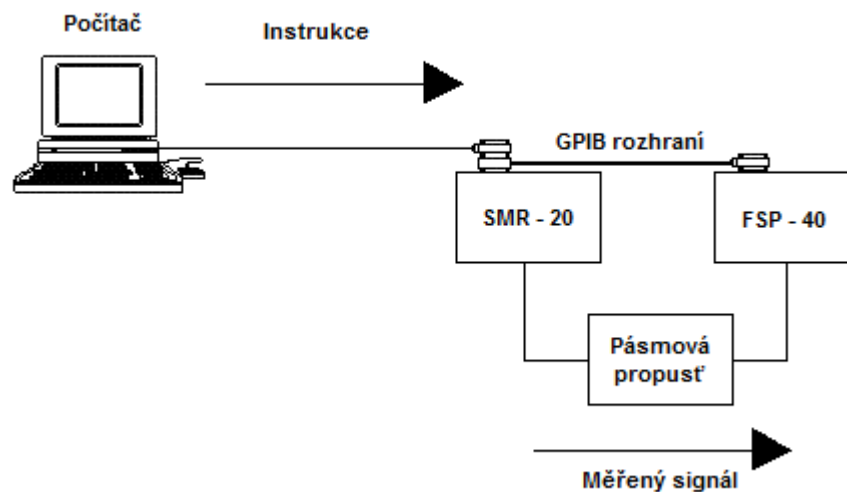
4 MĚŘENÍ

Abych mohl ověřit, že program správně funguje, provedl jsem kontrolní měření. Bylo zapotřebí zjistit, zda hodnoty načtené programem jsou korektní, a jestli program splňuje požadavky na rychlost měření.

4.1 Úkol měření

Generátor posílá signál o narůstající frekvenci na analyzátor. Z analyzátoru program načte data o výkonu a vynese je do grafu.

Pokusné měření jsem provedl přes pásmovou propust'. Pro kontrolu správnosti měření jsem výsledky porovnal s hodnotami naměřenými na vektorovém analyzátoru ZVB20, při použití stejné propusti i stejné kabeláže.



Obr. 32 Blokové schéma měření

4.2 Parametry měření

Parametry nastavené programem:

Generátor SMR – 20:

- RF - zapnuto

Analyzátor FSP – 40:

- atenuátor - automatické nastavení
- jednotky výkonu – dBm

- osa y – 100 dB
- hodnota VBW – 10 kHz
- TRACE – average
- detektor – RMS
- počet bodů sweepu - 501

Parametry nastavené uživatelem:

Generátor SMR – 20:

- minimální frekvence – 1 GHz
- maximální frekvence – 11 GHz
- výkon – 10 dBm
- krok – 10 MHz

Analyzátor FSP – 40:

- span – 10 kHz
- RBW – 3 kHz
- sweepcount - 5
- sweeptime – 2,5 ms
- referenční úroveň výkonu – 10 dBm

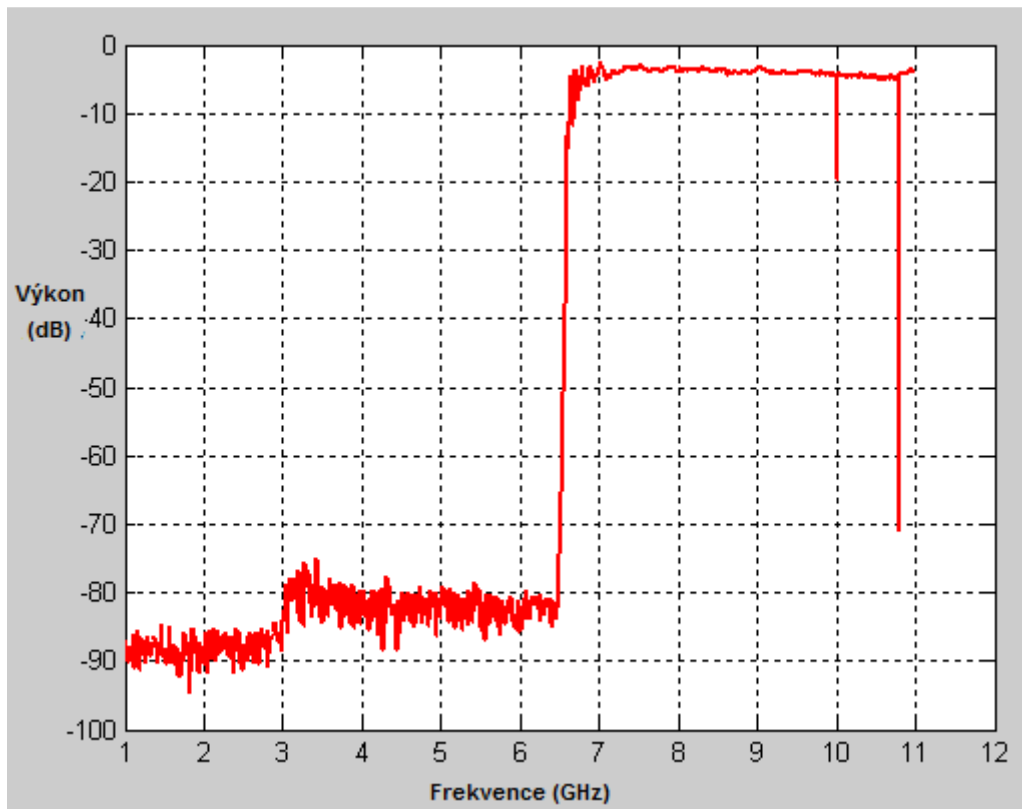
Pásmová propust' R100:

Hodnota mezní frekvence:

$$f_m = \frac{c}{\lambda_c} = \frac{3 \cdot 10^8}{2a} = \frac{3 \cdot 10^8}{2 \cdot 0,0228} \cong 6,5 \text{GHz}$$

Použitý typ kabeláže - koaxiální kabel

4.3 Změřený signál



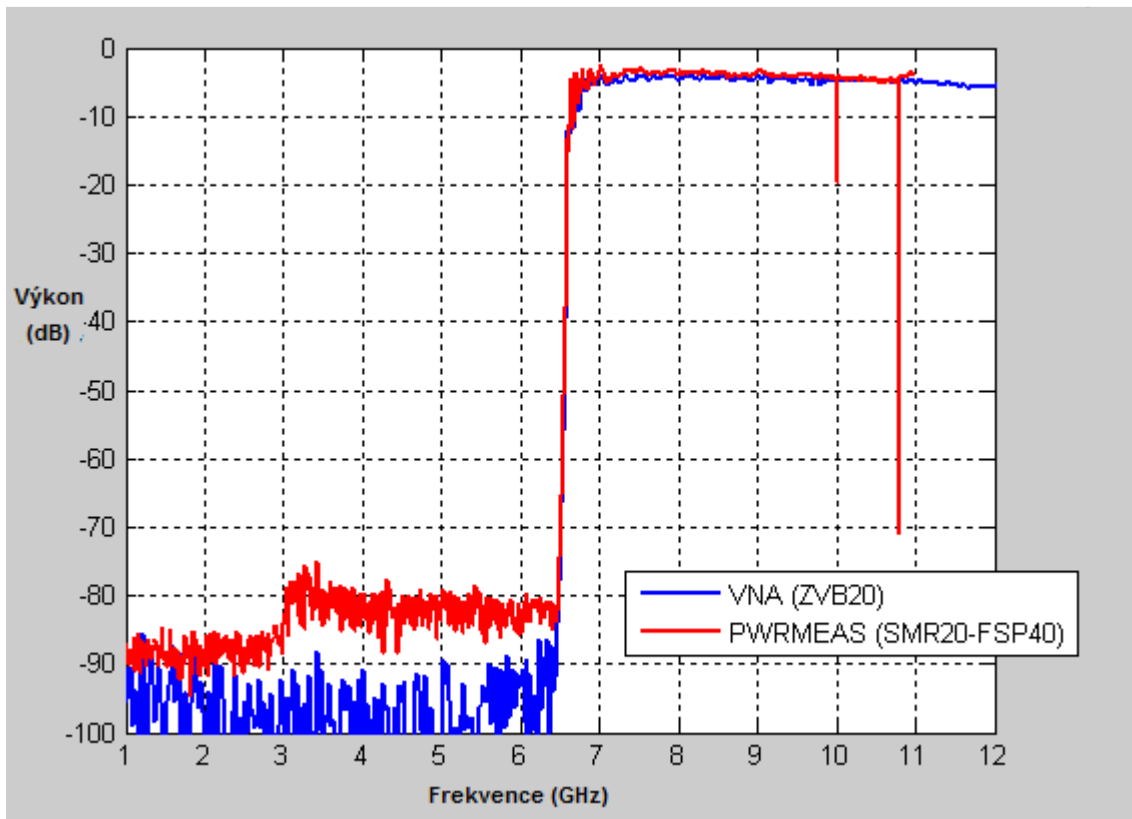
Obr. 33 Změřený signál

Na frekvenci 1 – 6,5 GHz pásmová propuť podle očekávání žádný signál nepropouští. V této oblasti je vidět pouze signál způsobený šumem. Na frekvenci kolem 3 GHz je vidět mírný nárůst způsobený změnou nastavení oscilátoru uvnitř frekvenčního analyzátoru FSP 40. V oblasti 6,5 GHz pak dochází k dramatickému nárůstu výkonu až na hodnotu -3,5 dB, na kterých se úroveň výkonu po několika výkyvech ustálí. V závěru jsou patrné propady výkonu. V těchto propadech dochází z neznámých důvodů ke ztrátě dat. I po několikanásobném zopakování, byly výsledky měření totožné i se zmíněnými výkonovými propady.

4.4 Porovnání

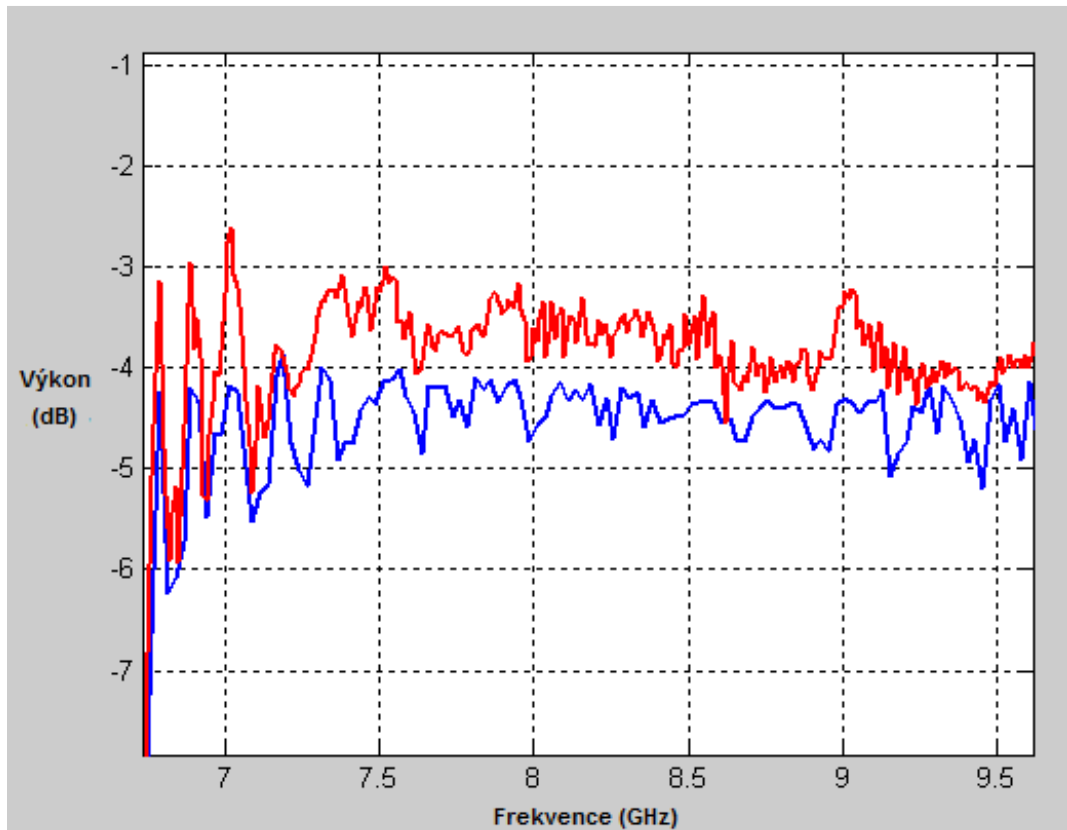
Výsledek měření jsem porovnal s výsledky měření na vektorovém obvodovém analyzátoru ZVB 20. Tento přístroj je nízkošumový a měří frekvenční spektrum vektorově na rozdíl od analyzátoru FSP 40, který měří spektrum skalárně. Analyzátor ZVB 20 je kvalitní a proto by měl být vhodný pro srovnání.

Pro srovnání měření bylo důležité, aby měření na obou přístrojích proběhlo s totožnou kabeláží a se stejnou pásmovou propustí, jelikož útlumy na vedení značně ovlivňují výsledky měření.



Obr. 34 Porovnání naměřených signálů

Z grafu můžeme vyčíst, že signály jsou si velmi podobné. První část grafu se sice liší, ovšem to je způsobeno rozdílnými přístroji. Přístroj ZVB 20 je nízkošumový, takže v této oblasti vykazuje daleko menší hodnotu šumu. Náběžná hrana na úrovni frekvence 6,5 GHz je identická v obou případech měření. Maximální hodnota výkonu se liší zhruba o jeden decibel. Propady výkonu v oblasti 10 a 11 GHz se na přístroji ZVB 20 neobjevily.



Obr. 35 Detailní zobrazení srovnání signálů

Na tomto detailním zobrazení je patrné, že oba signály jsou posunuty v průměru o jeden decibel, ovšem jejich tvar je téměř totožný a to i v malých výkyvech. Odchylka mezi oběma signály může být částečně způsobena chybovostí přístrojů. Odchylka na přístroji FSP 40 je 0,5 dB. Další možnou příčinou rozdílu mezi signály může být rozdílnost funkcí analyzátorů, kdy analyzátor FSP 40 měří spektrum pouze skalárně, kdežto analyzátor ZVB 20 měří spektrum vektorově, což znamená, že neměří jenom amplitudu, ale i fázi.

4.5 Útlum

Útlum výkonu změřený na analyzátoru ZVB 20 má hodnotu přibližně -4,5 dB. Je to způsobeno kabeláží, pásmovou propustí a přechody mezi nimi. Oba dva koaxiální kabely mají útlum o hodnotě přibližně 1 dB. Přechod mezi kabely a propustí má útlum kolem 0,5 dB a pásmová propust' samotná vykazuje také podobný útlum.

4.6 Rychlost měření

Co se týče rychlosti měření, nebyly výsledky příliš uspokojivé. V zadání byl požadavek na rychlost měření 10 ms na bod. Největší rychlost, které se mně podařilo dosáhnout, byla 75 ms na jeden bod grafu při spolehlivém měření, což je několikrát pomalejší, než bylo cílem.

4.6.1 Časování

V programu jsou použity příkazy na zpoždění *pause()*. Tento příkaz zastaví činnost programu na dobu definovanou argumentem této funkce. Tento příkaz jsem používal kvůli tomu, aby měl analyzátor čas na zpracování spektra signálu a aby bylo možno načíst marker. Délku pauzy jsem určil metodou pokus omyl a nastavil na nejnižší možnou hodnotu, aby program fungoval. Hodnota pauzy je 0,02 s, což už samo o sobě vylučuje, aby byla rychlost měření jednoho bodu 10 ms. Při použití kratší pauzy program neměří vůbec. Příčin, proč analyzátor není schopen zpracovat signál rychleji, může být několik. Jednou z nich je dlouhá doba jednoho sweepu.

4.6.2 Doba sweepu

Program funguje tak, že jde nastavit dobu trvání sweepu. Minimální hodnota, kterou je analyzátor FSP 40 schopen akceptovat je 2,5 ms. Program dále umožňuje zadat počet sweepů pro měření jednoho bodu, ze kterých se vypočítá průměrná hodnota. Pokud nastavíme 4 sweepů na bod, měl by být výsledný čas měření jednoho bodu 10 ms.

Doba jednoho sweepu se také dá spočítat podle vzorce:

$$t_{sweep} = \frac{5}{2\pi} \cdot \frac{span}{RBW^2}$$

Pokud je nastavení parametrů: span 10 kHz, RBW 3 kHz, poté bude čas na jeden sweep:

$$t_{sweep} = \frac{5}{2\pi} \cdot \frac{1 \cdot 10^4}{(3 \cdot 10^3)^2} = \frac{5}{2\pi} \cdot \frac{1 \cdot 10^4}{9 \cdot 10^6} = \frac{5}{2\pi \cdot 9 \cdot 10^2} = 0,885ms$$

Vypočtená doba je menší než nastavený čas, takže problém by zde být neměl.

Dokonce i při nastavení jednoho sweepu na bod, což by znamenalo dobu měření 2,5 ms, byl čas stále nejlépe 75 ms na bod.

4.6.3 Časovač

Zrychlit program by možná šlo použitím jiného způsobu časování. Příkaz *pause()* není ideální z hlediska toho, že zastaví celý běh programu, což může způsobovat problémy. Místo příkazu pauze by bylo možno použít časovač, který běží nezávisle na běhu programu.

4.6.4 Průměrování

Další možnou příčinou může být problém s počítáním průměru z naměřených hodnot signálů. V tomto případě průměrování zařizuje sám analyzátor, a to pomocí funkce stopy *average*. Abych ověřil, jestli tato funkce program opravdu zpomaluje, přepsal jsem program tak, že přístroj změřil pětát tu samou hodnotu frekvence po jednom sweepu. Program výsledky z každého měření průběžně uložil a následně sám zprůměroval. Rychlost měření bohužel byla 0,08 s, což je ještě nižší.

4.6.5 Rozhraní

GPIB rozhraní nepatří k nejrychlejším. Pro zrychlení měření jsem proto zkusil komunikovat s přístrojem pomocí LAN. Připojit přístroj se mi sice podařilo, ovšem úspěšné komunikace už jsem nedocílil.

4.6.6 Měření modulovaného signálu

Pro ověření, zda jsou přístroje vůbec schopny pracovat požadovanou rychlostí, jsem provedl další pokusné měření.

Generátor SMR 20 generoval amplitudově modulovaný signál. Tento signál byl zpracováván analyzátozem, ze kterého pak program četl výsledky. Frekvence zůstala neměnná, signál pouze měnil amplitudu. Analyzátor zobrazoval frekvenční spektrum, které měnilo svou amplitudu v závislosti na modulačním signálu. Modulační frekvence byla 100 Hz.

$$T_m = \frac{1}{f_m} = \frac{1}{100} = 10ms$$

V tomto případě program fungoval bez problémů. Rychlost zobrazení jednoho bodu tedy byla menší než 10 ms. Generátor i analyzátor byly oba ovládány pomocí TCPI příkazů přes GPIB rozhraní.

Program v tomto případě fungoval požadovanou rychlostí, což naznačuje, že by program mohl být zpomalován změnou frekvence měřeného signálu.

4.7 Vyhodnocení výsledků

Aplikace po stránce měření pracuje poměrně spolehlivě. Naměřené hodnoty výkonu v závislosti na frekvenci byly velmi podobné hodnotám neměřených na analyzátoru ZVB 20, takže lze předpokládat, že jsou tyto výsledky korektní.

Při zkoušení měření s různými parametry jsem narazil na zvláštní problém, který spočívá v tom, že pokud zadáme hodnotu kroku větší než 100 MHz, dojde k selhání měření. Program zobrazuje korektní hodnoty pouze do hodnoty 3 GHz a poté graf vykazuje pouze lineární charakteristiku. Bohužel se mi nepovedlo tento problém vyřešit. Měření selhává i při změně parametrů sweeptime či sweepcount. Po snížení kroku pod 100 MHz program opět funguje standardně.

Co se týká rychlosti měření, požadované doby 10 ms na jeden bod měření se mi podařit nedokázalo. Nejrychlejší dosažená doba měření byla sedmkrát větší, než by bylo potřeba, což není ideální. Otázkou je, zda-li je vůbec možno požadované rychlosti dosáhnout z důvodu limitace hardwarem přístroje.

Tab 5 Časy měření

požadovaný čas	10 ms
čas dosažený s automatickým průměrováním	75 ms
čas dosažený s manuálním průměrováním	80 ms

ZÁVĚR

Výsledky mé práce mohou být brány rozporuplně. Program funguje z hlediska měření správně. Naměřené hodnoty jsou korektní – v tomto ohledu se dá mluvit o úspěchu. Problém je však v rychlosti, s jakou program pracuje. I přes veškerou snahu se mně nepovedlo měření zrychlit natolik, aby vyhovovalo zadání. Pokusil jsem se o změnu typu rozhraní pro komunikaci mezi zařízeními, s čímž jsem bohužel neuspěl. Další mou snahou bylo komunikovat s přístroji prostřednictvím matlab instrument driveru. Ani toto se však s úspěchem nesesetkalo. Možným vysvětlením pomalého měření může být to, že přístroj není schopen takto rychle reagovat na změnu frekvence. Pro toto tvrzení ovšem nemám žádné důkazy, takže se jedná o pouhou domněnku. Každopádně z tohoto hlediska se o úspěšném výsledku příliš mluvit nedá.

Tato práce byla pro mě něčím novým. V průběhu jsem se musel naučit mnoho nových a zajímavých věcí. S programem Matlab jsem předtím moc zkušeností neměl. Nástroje jako Instrument control toolbox nebo GUIDE mi pomohly při učení se v tomto programu pracovat. Doposud jsem se nesesetkal s programováním grafických aplikací, z čehož jsem měl zpočátku obavy, ale nakonec se ukázalo, že nejde o věc složitou (alespoň v rozsahu mé práce). Práce v editoru GUIDE v tomto směru tuto věc opravdu velmi zjednodušila. S příkazy SCPI jsem taky zatím nepracoval. Tento typ příkazů je univerzální nezávisle na výrobci nebo typu přístroje a je poměrně rozšířený. Proto si myslím, že je dobré, že jsem se s tímto druhem programování seznámil.

ZÁVĚR V ANGLIČTINĚ

The results of my work can be taken inconsistent. The program works in terms of measuring correctly. The measured values are correct - in this regard, it is success. The problem is the speed with which the program operates. Despite all efforts I couldn't reduce measurement speed enough to suit the task. I tried to change the type of interface for communication between devices but I unfortunately did not succeed. Another of my effort was to communicate with devices using MATLAB instrument driver. Even this, however, did not succeed. A possible explanation for the slow measurement may be that the device is not able to respond so quickly to the change in frequency. For this argument, however, I have no evidence, so this is just a hypothesis. Anyway, from this point of view, this wasn't successful too much.

This work was something new for me. In the course I had to learn many new and interesting things. With Matlab I had not had much experience. Tools such as the Instrument Control Toolbox or GUIDE helped me with learning to work with this program. So far I have not met with programming graphics applications, of which I was initially worried, but it turned out that it is not a difficult thing (at least to the extent of my work). The editor GUIDE in this direction this thing really simplified. I have never worked with SCPI commands. This type of command is universal regardless of the manufacturer or device type and is relatively widespread. Therefore I think it is good that I have did this kind of programming.

SEZNAM POUŽITÉ LITERATURY

- [1] HERINGOVÁ, B., HORA, P. Matlab Díl 1. Práce s programem, Praha. Vydavatelství H-S 1995.
- [2] HERINGOVÁ, B., HORA, P. Matlab Díl 2. Popis funkcí, Praha. Vydavatelství H-S 1995.
- [3] Introduction to Graphical User Interface (GUI) – Matlab 6.5. PDF document – dostupný z <http://www.ewh.ieee.org/r8/uae/GUI.pdf>
- [4] RAUSCHER, C. Fundamentals of spectrum analysis. Munich, 2007. ISBN 978-3-939837-01-5. 219 stran.
- [5] Microwave generator SMR 20. Operating manual. RohdeSchwarz. 2009. 230 stran.
- [6] Spectrum analyzer FSP 40. Operating manual. RohdeSchwarz. 2010. 796 stran.
- [7] *GPIB Tutorial* [online]. [cit. 2013-05-20]. , dostupné na WWW: [<http://www.hit.bme.hu/~papay/edu/GPIB/tutor.htm>]
- [8] *Automatizované měřicí systémy se sběrnici IEEE 488* [online]. [cit. 2013-05-20]. , dostupné na WWW: [<http://www.hit.bme.hu/~papay/edu/GPIB/tutor.htm>]
- [9] *Spektrum signálu* [online]. [cit. 2013-05-20]. , dostupné na WWW: [http://cs.wikibooks.org/wiki/Praktick%C3%A1_elektronika/Spektrum_sign%C3%A1lu]
- [10] *Spektrální analyzátory* , Radioelektronická měření (MREM, LREM), Jiří Dřínovský, PDF document, 25 stran.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ESE	Event status enabled
ESR	Event status Register
GPIB	General Purpose Interface Bus
RBW	Resolution bandwidth
SCPI	Standard commands for programmable instruments
SRE	Service request enable
SRQ	Service request
STB	Status byte

SEZNAM OBRÁZKŮ

<i>Obr. 1</i> Vektorové zobrazení signálu [4].....	13
<i>Obr. 2</i> Vyšší harmonické signály [4]	14
<i>Obr. 3</i> Součet vyšších harmonických signálů [4].....	14
<i>Obr. 4</i> Rozsah frekvenčního spektra [4]	15
<i>Obr. 5</i> Spektrum sinusového signálu [9].....	16
<i>Obr. 6</i> Spektrum modulovaného signálu [9].....	16
<i>Obr. 7</i> Spektrum obdélníkového signálu [9]	17
<i>Obr. 8</i> Spektrum obdélníkového signálu [9]	17
<i>Obr. 9</i> Spektrum šumu [9].....	17
<i>Obr. 10</i> Frekvenční analýza [4].....	18
<i>Obr. 11</i> Analyzátor s přepínatelnými filtry [10]	19
<i>Obr. 12</i> Analyzátor s přeladitelnými filtry [10]	19
<i>Obr. 13</i> Heterodynní analyzátor [10]	20
<i>Obr. 14</i> Analyzátor používající číslicové filtry [10]	20
<i>Obr. 15</i> Překrývání signálu [4].....	21
<i>Obr. 16</i> Zobrazení spektra [4]	22
<i>Obr. 17</i> Blokové schéma detektorů [4]	24
<i>Obr. 18</i> Detektory [4].....	25
<i>Obr. 19</i> Topologie [7]	26
<i>Obr. 20</i> GPIB linky [7]	27
<i>Obr. 21</i> Piny na GPIB [7].....	28
<i>Obr. 22</i> Proces přenosu [8]	28
<i>Obr. 23</i> Algoritmus přenosu dat [8].....	29
<i>Obr. 24</i> Hierarchická struktura příkazů SCPI [6].....	30
<i>Obr. 25</i> Proces zpracování příkazu [6]	33
<i>Obr. 26</i> Model status registru [6]	35
<i>Obr. 27</i> Algoritmus programu	43
<i>Obr. 28</i> Zapojení.....	44
<i>Obr. 29</i> GUIDE.....	46
<i>Obr. 30</i> GUI.....	47
<i>Obr. 31</i> Menu	49
<i>Obr. 32</i> Blokové schéma měření	51

<i>Obr. 33 Změřený signál</i>	<i>53</i>
<i>Obr. 34 Porovnání naměřených signálů.....</i>	<i>54</i>
<i>Obr. 35 Detailní zobrazení srovnání signálů.....</i>	<i>55</i>

SEZNAM TABULEK

<i>Tab 1 Přehled znaků v SCPI příkazech</i>	<i>32</i>
<i>Tab 2 Popis bitů status byte</i>	<i>37</i>
<i>Tab 3 Popis bitů event status registru</i>	<i>38</i>
<i>Tab 4 SCPI příkazy</i>	<i>45</i>
<i>Tab 5 Časy měření</i>	<i>58</i>

SEZNAM PŘÍLOH

Příloha P I : Přehled status registrů

Elektronické přílohy

Příloha P II : Zdrojový kód programu pwrmeas

