

System pro zpracování dat z měřících přístrojů

System for processing data from measuring devices

Bc. Petr Bobek

Diplomová práce
2010



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2009/2010

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Petr BOBEK**
Osobní číslo: **A08457**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Systém pro zpracování dat z měřících přístrojů**

Zásady pro vypracování:

1. Seznamte se s principy radiových komunikačních technologií používaných v mobilních zařízeních.
2. Analyzujte struktury dat získaných měřeními radiových parametrů mobilních zařízení a navrhnete jejich databázovou reprezentaci.
3. Vytvořte systém (web. aplikaci) pro zpracování dat získaných měřeními radiových parametrů mobilních zařízení.
4. Naměřená data prezentujte jak ve formě grafů, tak ve formě statistik, s možností srovnání vlastností všech měřených zařízení.
5. Veškeré vytvořené zdrojové soubory umístěte v elektronické podobě do přílohy diplomové práce na CD a výsledný systém zveřejněte pod open source licenci GPL2.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. GARG, Vijay. WIRELESS COMMUNICATIONS AND NETWORKING. 1st edition. Morgan Kaufmann Publishers, 2007. 931 s. ISBN 978-0-12-373580-5
2. GORANSSON, Paul, GREENLAW, Raymond. Secure Roaming In 802.11 Networks. 1st edition. Newnes, 2007. 369 s. ISBN 978-0-7506-8211-4
3. YACOUB, Michel Daoud. WIRELESS TECHNOLOGY – Protocols, Standards, and Techniques. 1st edition. CRC PRESS, 2002. 547 s. ISBN 0-8493-0969-7
4. CHAK, Dan. Enterprise Rails. 1st edition. O'Reilly, 2008. 352 s. ISBN 978-0-596-51520-1
5. FISHER, Timothy. Ruby on Rails Bible. 1st edition. Wiley Publishing, Inc., 2008. 627 s. ISBN 978-0-470-25822-4
6. BLACK, DAVID. The Well-Grounded Rubyist. 1st edition. MANNING, 2009. 519 s. ISBN 978-1-933988-65-8
7. SCHWARTZ, Baron, et al. High Performance MySQL. 2nd rev. edition : O'Reilly, 2008. 710 s. ISBN 978-0-596-10171-8

Vedoucí diplomové práce:

Ing. Tomáš Dulík

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

19. února 2010

Termín odevzdání diplomové práce:

8. června 2010

Ve Zlíně dne 19. února 2010


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Cílem této diplomové práce je vytvoření systému (webové aplikace), naprogramovaného pomocí programovacího jazyka PHP s využitím libovolného frameworku, který bude analyzovat struktury dat získané měřením radiových parametrů mobilních zařízení. Získané výsledky se budou uživatelům zobrazovat pomocí grafů, dále jim aplikace umožní jejich vzájemné porovnání a sdílení s jinými uživateli.

Klíčová slova: IEEE 802.11, PHP, Zend Framework, Doctrine, jQuery, WLAN, Rohde & Schwarz, EVM, QAM, IQ Imbalance

ABSTRACT

The main aim of this thesis is to create a system (web application), programmed in PHP using arbitrary Framework, which will analyze the structure of data obtained by measuring the radio parameters from mobile devices. The results will be displayed to users using charts. An application also allow users to compare and share the results with other users.

Keywords: IEEE 802.11, PHP, Zend Framework, Doctrine, jQuery, WLAN, Rohde & Schwarz, EVM, QAM, IQ Imbalance

Zde bych chtěl zdůraznit mou vděčnost lidem, kteří mně při této práci podpořili.

Rád bych poděkoval Ing. Tomáši Dulíkovi za jeho rady, připomínky, návrhy, doporučení a také za volnost, kterou mi dal při vypracovávání celého projektu.

Speciální poděkování mé rodině a přítelkyni za celkovou podporu jak při vypracovávání této práce, tak i při studiu.

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.

Ve Zlíně

.....
Podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 RADIOVÉ PARAMETRY MOBILNÍCH ZAŘÍZENÍ	11
1.1 SPEKTRÁLNÍ MASKA	11
1.2 VYSÍLACÍ VÝKON EIRP	11
1.3 RADIOVÉ PARAMETRY PROTOKOLŮ Z RODINY STANDARDŮ 802.11.....	12
1.3.1 802.11 Specifikace	12
1.3.2 Konstelační diagram.....	13
1.3.3 QAM (Quadrature Amplitude Modulation)	14
1.3.4 EVM (Error Vector Magnitude).....	14
1.3.5 IQ Imbalance	16
2 METODIKY MĚŘENÍ	18
2.1 METODIKA ČTÚ	18
2.2 VLASTNÍ NAVRŽENÁ METODIKA	18
2.3 POROVNÁNÍ OBOU METOD	21
3 POUŽITÉ PŘÍSTROJE	22
3.1 SPEKTRÁLNÍ ANALYZÁTOR ROHDE & SCHWARZ FSL	22
3.1.1 Nastavení Spektrálního Analyzátoru.....	22
3.2 MULTIFUNKČNÍ WI-FI ZAŘÍZENÍ ZCOMAX WA2204	24
3.3 KARTA MIKROTIK R52.....	25
4 ZEND FRAMEWORK	26
4.1 HISTORIE ZF	26
4.2 MVC.....	27
4.3 FRONT CONTROLLER.....	28
4.3.1 Routes.....	28
4.3.2 Dispatcher	29
4.3.3 Konvence v pojmenovávání	31
4.4 NÁSTROJE	31
4.4.1 Apache.....	31
4.4.2 MySQL.....	31
4.4.3 PHP	32
4.4.4 ZF	32
4.4.5 Vývojové prostředí – Netbeans, Zend Studio	32
4.4.6 Zend Tool	33
4.4.7 Doctrine ORM.....	33
4.4.7.1 Instalace	34
4.4.7.2 Tvorba modelů.....	35
4.4.7.3 Asociace mezi modely	37
4.4.7.4 DQL	38
4.5 PSANÍ KÓDU.....	39
II PRAKTICKÁ ČÁST	40
5 MĚŘENÍ VE FREKVENČNÍ OBLASTI	41

5.1	NASTAVENÍ VÝKONU AP NA 1DBM.....	41
5.2	ZÍSKÁNÍ HODNOT Z ANALYZÁTORU	42
5.3	DALŠÍ MĚŘENÍ VE FREKVENČNÍ OBLASTI.....	42
6	MĚŘENÍ V ČASOVÉ OBLASTI	43
6.1	NASTAVENÍ VÝKONU AP NA 1 DBM	43
6.2	ZÍSKÁNÍ HODNOT Z ANALYZÁTORU	44
6.3	DALŠÍ MĚŘENÍ V ČASOVÉ OBLASTI	44
7	TVORBA APLIKACE.....	45
7.1	PŘIHLAŠOVACÍ SYSTÉM	45
7.2	REZERVACE.....	47
7.3	NOVÉ MĚŘENÍ.....	49
7.4	DOKONČENÁ MĚŘENÍ	50
7.5	POROVNÁVÁNÍ MĚŘENÍ	51
7.6	ZPŘÍSTUPNĚNÍ MĚŘENÍ	51
7.7	ADMIN SEKCE.....	52
8	MANUÁL PRO POUŽITÍ APLIKACE.....	53
8.1	ÚVODNÍ OBRAZOVKA A REGISTRACE.....	53
8.2	CHYBOVÉ HLÁŠKY	54
8.3	EDITACE ÚDAJŮ.....	54
8.4	REZERVACE TERMÍNU	54
8.5	NOVÉ MĚŘENÍ.....	56
8.6	UKONČENÁ MĚŘENÍ.....	56
8.7	ADMIN SEKCE.....	57
	ZÁVĚR	59
	CONCLUSION	61
	SEZNAM POUŽITÉ LITERATURY.....	62
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	64
	SEZNAM OBRÁZKŮ	66
	SEZNAM TABULEK.....	68
	SEZNAM PŘÍLOH.....	69
	PŘÍLOHA P I: SCREEN WEBOVÉ APLIKACE	70

ÚVOD

S bezdrátovými technologiemi se v dnešním světě setkal téměř každý z nás, ať už v podobě mobilních telefonů, různých wi-fi zařízení nebo GPS. S tím, jak roste jejich obliba, rostou také nároky na přenosovou rychlost, spolehlivost, dosah, dostupnost a s tím souvisí i vyšší nároky na modulaci dat, latenci atp.

Ten, kdo se již pokusil o instalaci wi-fi nebo jiného bezdrátového zařízení, určitě narazil na problémy spojené s maximálními rychlostmi udávanými výrobci vs. reálně dostupnými. Nejsou to jen rychlosti, kde někteří výrobci nedodržují normu. Mezi další problémy fyzické vrstvy 802.11 na straně vysílače může být například nedodržení předepsaného tvaru spektrální masky, nedodržení předepsaných hodnot EVM, nedostatečné potlačení nosné a postranního pásma ve vysílači nebo kolísání frekvence nosné. Na straně přijímače to může být nedodržení předepsané minimální úrovně vstupního signálu, potlačení sousedních a vzdálených kanálů nebo clear channel assessment sensitivity. [12]

Cílem této diplomové práce je vytvoření webové aplikace ke spektrálnímu analyzátoru pořízenému z grantu Fondu rozvoje Cesnet č. 351/2009, který je schopný analyzovat probíhající komunikaci až na úroveň paketů ze všech mobilních zařízení. Spektrální analyzátor je v současnosti vybaven pouze SW pluginem pro analýzu protokolů 802.11 (Wi-Fi), ale v budoucnu se počítá s možným dokoupením dalších pluginů.

Vytvořená aplikace naměřené charakteristiky pomůže zpracovat (uložením naměřených výsledků do databáze) a následně poskytne uživateli možnost naměřená data prezentovat jak ve formě grafů, tak ve formě statistik s možností srovnání vlastností všech měřených zařízení.

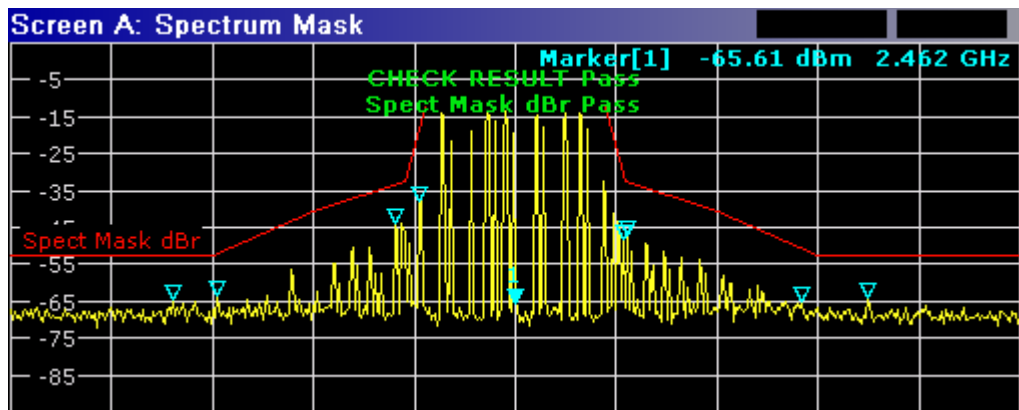
I. TEORETICKÁ ČÁST

1 RADIOVÉ PARAMETRY MOBILNÍCH ZAŘÍZENÍ

Tato kapitola se zabývá obecným popisem radiových parametrů mobilních zařízení, mezi které můžeme počítat např. Bluetooth, 3G, 4G, Wi-Fi, GPS, ale i spoustu dalších.

1.1 Spektrální maska

Spektrální maska vyznačuje maximální hranice vysílacího výkonu, které může naměřený signál v daném rozsahu frekvencí dosáhnout. Na obrázku níže (Obr. 1) je vyznačena červenou barvou spektrální maska definovaná standardem 802.11a, naměřené spektrum signálu žlutou. Žlutý průběh by nikdy neměl přesáhnout červenou linii.



Obr. 1: Spektrální maska

1.2 Vysílací výkon EIRP

EIRP (Equivalent Isotropically Radiated Power) je celkový výkon, který by bylo nutné vyzářit izotropní anténou (vyzařuje do všech směrů prostoru stejně) tak, aby bylo v daném směru dosaženo jisté intenzity záření. Používá se k vyjádření intenzity rádiového záření vysílaného směrem, kterým je anténa (typicky směrová) namířena. EIRP lze vypočítat dle vztahu (1), kde P_T je výstupní výkon karty, L_c jsou útlum anténního kabelu ostatních konektorů nebo pigtailů a G_a je zisk externí antény. [24]

$$EIRP [dBm] = P_T - L_c + G_a \quad (1)$$

1.3 Radiové parametry protokolů z rodiny standardů 802.11

Wi-Fi je standard pro lokální síť (taktéž známá pod zkratkou WLAN), který je definován specifikací IEEE 802.11. Uživatelé s bezdrátovým adaptérem např. notebookem, mobilním telefonem, PDA, PC – označovaným obvykle jako „stanice“ umožňuje v blízkosti přístupového bodu - tzv. access pointu, dále jen AP připojení k internetu. Oblast, která je pokryta jedním nebo více AP se říká hotspot.

Velkou výhodou a úspěch wi-fi přineslo právě nelicencované rádiové pásmo, které využívá. Pro koncového uživatele to znamená, že se nemusí ptát příslušného úřadu na povolení k používání této frekvence, ale na druhou stranu je zde možnost rušení od jiných zařízení pracujících na stejných frekvencích, které mohou zapříčinit chybné fungování, popřípadě zařízení úplně vyřadit z provozu.

1.3.1 802.11 Specifikace

Standard 802.11 zahrnuje mnoho druhů modulací. Nejpoužívanější z nich uvedeny v tabulce (Tab.1).

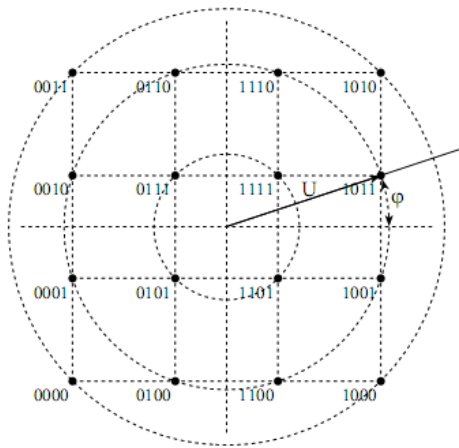
Standard	Rok vydání	Pásmo [GHz]	Maximální rychlost [Mbit/s]	Fyzická vrstva
původní IEEE 802.11	1997	2,4	2	DSSS a FHSS
IEEE 802.11a	1999	5	54	OFDM
IEEE 802.11b	1999	2,4	11	DSSS
IEEE 802.11g	2003	2,4	54	OFDM
IEEE 802.11n	2009	2,4 nebo 5	600	MIMO
IEEE 802.11y	2008	3,7	54	

Tab. 1: Přehled standardů IEEE 802.11

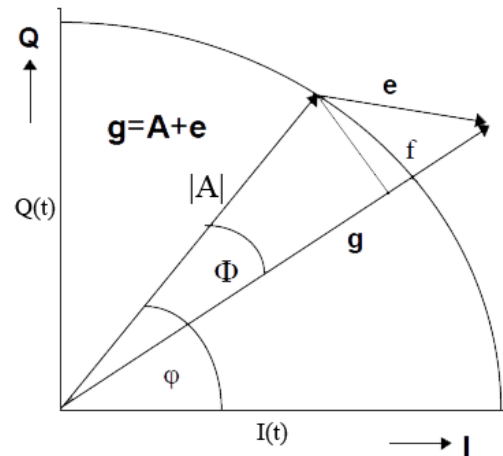
Maximální rychlosti udávané výrobci jsou oproti realitě trochu odlišné. Reálná propustnost IP vrstvy u 802.11b je menší než 5 Mbit/s, u 802.11g je propustnost okolo 20 Mbit/s. Nakonec porovnejme 802.11n, kdy je teoretické maximum 600 Mbit/s, ale technicky je na fyzické vrstvě možno přenést pouze 300 Mbit/s a reálná hodnota propustnosti na IP vrstvě se pohybuje okolo 130 Mbit/s. [16]

1.3.2 Konstelační diagram

Pro grafické znázornění modulace se používá IQ rovina (In-phase – soufázová složka, Quadrature – kvadrurní složka) viz. obrázek (Obr. 2), do které se zakreslují body odpovídající jednotlivým stavům nosné. Koncové body vektorů reprezentující jednotlivé stavy nosné se nacházejí pouze na kružnici s poloměrem rovnajícím její amplitudě.



Obr. 2: Konstelační diagram [17]



Obr. 3: Detail ideálního a naměřeného signálu s parametry, které se dají vypočítat [25]

Toto zobrazení je založeno na skutečnosti, že signál o úhlové rychlosti ω_c , časové fázi $\varphi(t)$ a amplitudě $A(t)$ je možné zobrazit v komplexní rovině jako fázor (neboli rotující vektor), složený ze dvou kvadrurních složek (IQ) se stejnými frekvencemi a vzájemnou fází 90 viz. (Obr. 3). Každému amplitudově fázovému stavu dané modulace odpovídá určitá velikost těchto složek.[18]

Parametry z obrázku (Obr. 3) se dají vypočítat dle níže uvedených vztahů (2) - (5):

$$A(t) = \sqrt{I^2(t) + Q^2(t)} \quad (2)$$

$$\varphi(t) = \arctg(Q(t)/I(t)) \quad (3)$$

$$I(t) = A(t)\cos\varphi(t) \quad (4)$$

$$Q(t) = A(t)\sin\varphi(t) \quad (5)$$

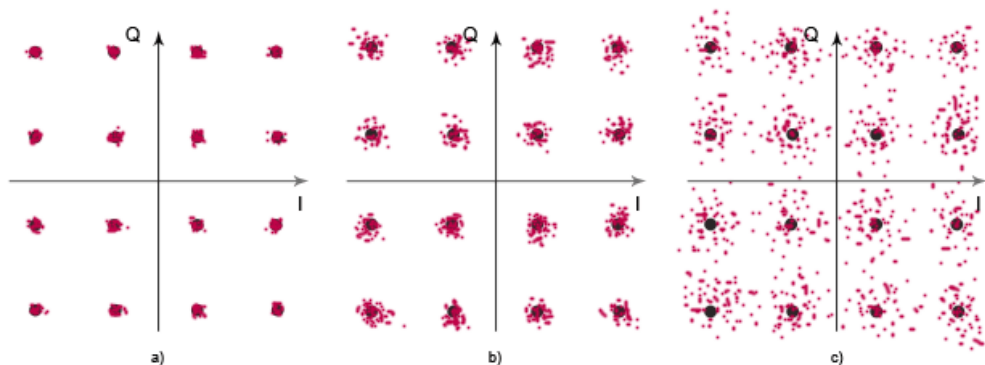
1.3.3 QAM (Quadrature Amplitude Modulation)

Kvadrurní amplitudová modulace je vícecestavová modulace, která využívá amplitudového a fázového klíčování. Klíčováním se rozumí ovlivňování nosné vlny pouze v rámci diskrétních stavů.

Výhodou vícecestavové modulace je, že umožňuje přenášet n bitů pomocí m symbolů (v jeden okamžik můžeme přenést více bitů najednou), čímž je možné ušetřit šířku pásma nebo naopak se stejnou šířkou pásma zvýšit přenosovou rychlost. Počet stavů může být různý podle typu QAM modulace.

Na následujících dvou příkladech lze vidět, jak se vypočítá počet stavů. Pro přenos 4 bitů v 1 symbolu, je potřeba 2^4 (16) stavů (tedy 16 QAM). 64 QAM bude znamenat, že amplituda a fáze nabývá 64 různých hodnot a dohromady přenáší 6 bitů. Obecná závislost mezi počtem stavů (m) a počtem bitů (n) je: $m = 2^n$.

S tím, jak se zvyšují počty stavů použitých modulací vzrůstají taktéž požadavky na přijímač, který musí rozlišit mnohem menší změny amplitudy, kmitočtu nebo fáze. V komunikačním kanálu působí taktéž šum a různé rušení, které chybovost výrazně zvyšují. [17] Jak se tyto závislosti projevují si můžete prohlédnout na obrázku (Obr. 4).



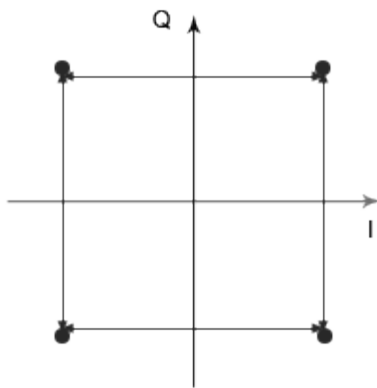
Obr. 4: Konstelační diagramy s různými druhy šumů a chyb

1.3.4 EVM (Error Vector Magnitude)

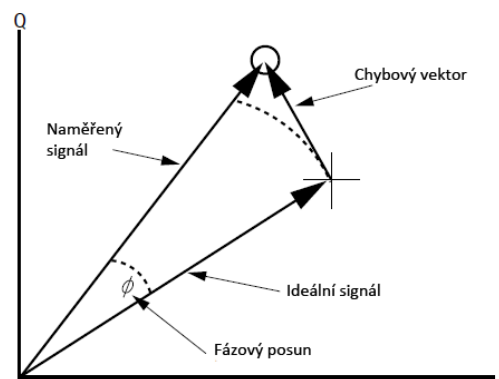
Error vector magnitude neboli amplituda chybového vektoru (dále jen EVM) je měření, které určuje jak přesný je digitální rádiový vysílač nebo přijímač (jinak řečeno, indikuje přesnost odlišení naměřeného od ideálního signálu).

Výstupní signál u ideálního vysílače má všechny konstelační body v ideálních polohách (Obr. 5), ale ve skutečnosti je situace odlišná a pozice aktuálního signálu může být unášena pryč od ideální polohy (Obr. 6).

Předpokládá se, že přijímač bude tolerovat určitou úroveň chyby. Norma udává, že by EVM mělo být menší nebo rovno např. 5,62% (liší se podle typu modulace). Na obrázku (Obr. 6) je ideální stav označen křížkem a naměřený signál označen kolečkem, ostatní parametry jsou popsány přímo v obrázku. [9]



Obr. 5: Ideální stav konstelačních bodů



Obr. 6: Detail ideálního a naměřeného signálu s parametry, které se dají určit [25]

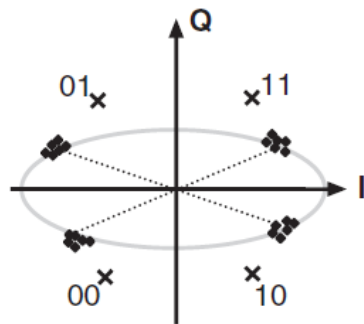
Matematicky je EVM definováno jako střední kvadratická hodnota chybových hodnot jednotlivých stavů v čase vzorkování. Lze vypočítat dle vztahu (6). Kde P_{error} je efektivní hodnota výkonu chybového vektoru a $P_{reference}$ je ideální amplituda signálu. [9]

$$EVM(\text{dB}) = \left(\frac{P_{error}}{P_{reference}} \right) = 10 \log_{10} \left(\frac{\frac{1}{N} (\sum_{k=1}^N \Delta I_k^2 + \Delta Q_k^2)}{\frac{1}{N} (\sum_{k=1}^N \Delta I_k^2)} \right) \quad (6)$$

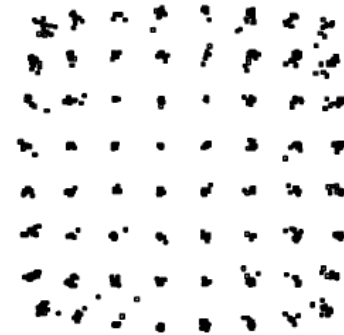
Chybový signál může být dále zkoumán jak v časové, tak i ve frekvenční oblasti. Pokud se podaří signál správně vyhodnotit, může být dobrým zdrojem informací o šumu, zkruslení a taktéž pomoci při identifikaci zdrojů degradace signálů. [18]

1.3.5 IQ Imbalance

Dalším z parametrů, který je potřeba pohlídat je IQ imbalance (rozdíly parametrů cest modulačních signálů I/Q), který způsobuje různá zkreslení konstelačního diagramu viz (Obr. 7) – kde ideální body jsou zakresleny křížkem a naměřené signály tečkami. IQ imbalance lze rozpoznat i pouhým okem a to tak, že konstelační diagram není přesný čtverec viz. (Obr. 8). Posunutí může nastat ve dvou rovinách (horizontální nebo vertikální). Posun v horizontální rovině nastane tehdy, když je amplituda I menší než amplituda Q a posun ve vertikální rovině tehdy, když je amplituda Q menší než amplituda I. [18]



Obr. 7: IQ Imbalance ($Q < I$) [9]



Obr. 8: Konstelační diagram s viditelnou IQ imbalance [9]

Výpočet odchylek amplitud, lze vypočítat podle následujících vztahů (7), (8), (9), (10), (11) a to buď jako efektivní nebo špičkové hodnoty.

$$e_i = (e_i)_I + (e_i)_Q \tag{7}$$

$$(e_i)_I = \frac{1}{N} \sum_{k=1}^N \delta I_k \tag{8}$$

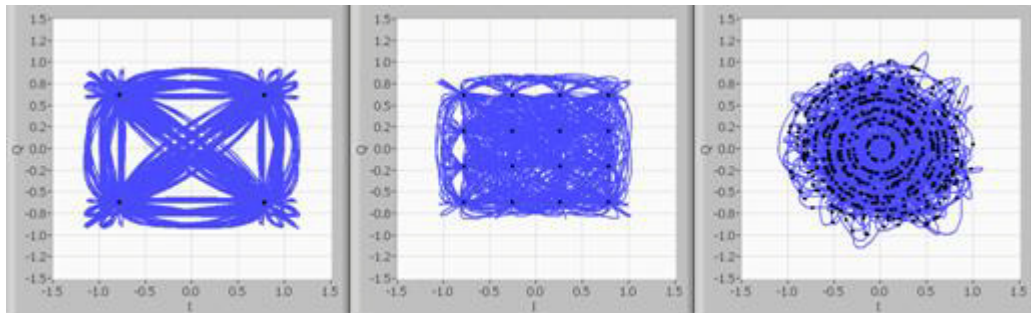
$$(e_i)_Q = \frac{1}{N} \sum_{k=1}^N \delta Q_k \tag{9}$$

$$V_Q = \frac{1}{M} \sum_{i=1}^M \frac{Q_i + (e_i)_Q}{Q_i} \tag{10}$$

$$\tag{11}$$

$$V_I = \frac{1}{M} \sum_{i=1}^M \frac{I_i + (e_i)_I}{I_i}$$

Vektorové diagramy se používají pro zobrazení cest, kterými prochází nosná při přechodu z jednoho stavu do druhého. Tyto diagramy se využívají i pro zobrazení IQ imbalance (Obr. 9) i přesto, že chyby v jednotlivých stavových polohách nejsou zcela zřejmé.



Obr. 9: IQ Imbalance 4-QAM, 16-QAM a 64-QAM [19]

2 METODIKY MĚŘENÍ

Měření skutečného nastaveného vysílacího výkonu karty může probíhat podle dvou metodik. První z nich je metodika českého telekomunikačního úřadu (dále jen ČTÚ) a druhá dle vlastní navržené metodiky. Více v následujících kapitolách.

2.1 Metodika ČTÚ

Metodika měření českého telekomunikačního úřadu (dále jen ČTÚ) je velmi přímočará a postupuje podle těchto kroků:

1. Nastavení módu spektra na “RMS“ (root mean square – efektivní hodnota)
2. Vybrání pevně dané šířky kanálu (např. 20 MHz)
3. Nastavení režimu zpracování signálu stopy na “max hold“ (zobrazuje jen maximální hodnotu v jednotlivých frekvencích po dobu měření)

Přesto, že je tato metoda uznávaná ČTU jako správná pro měření vysílacího výkonu, takto získaná data neodpovídají reálnému vyzářenému vysílacího výkonu karty. Metodika ČTÚ pouze udává číslo, které odpovídá součtu maximálních špiček výkonu na jednotlivých frekvencích v delším časovém horizontu.

Podrobnější informace o metodice a měření ČTÚ jsou uvedeny v literatuře [11].

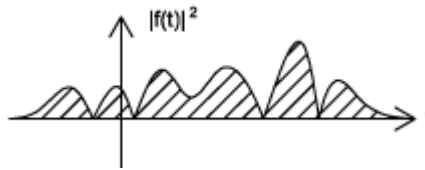
2.2 Vlastní navržená metodika

Mnohem přesnější a reálnější výsledky mohou být dosaženy pomocí měření v časové oblasti, ale jen za předpokladu, že výsledky z frekvenční oblasti (metodika ČTÚ) nejsou zapotřebí.

Spektrální analyzátor se v módu spektra nastaví na PVT (power vs. time), čímž se přepneme do časové oblasti (Obr. 15). V této oblasti lze měřit dvěma způsoby:

1. Energii signálu jednoho burstu

Energie signálu v časové oblasti se definuje jako velikost skalárního součinu signálu se sebou samým. Jinými slovy je to integrál nebo suma kvadrátů magnitud (absolutních hodnot) signálu v daném čase viz. (Obr. 11). Lze vypočítat dle vztahu (12). [17]

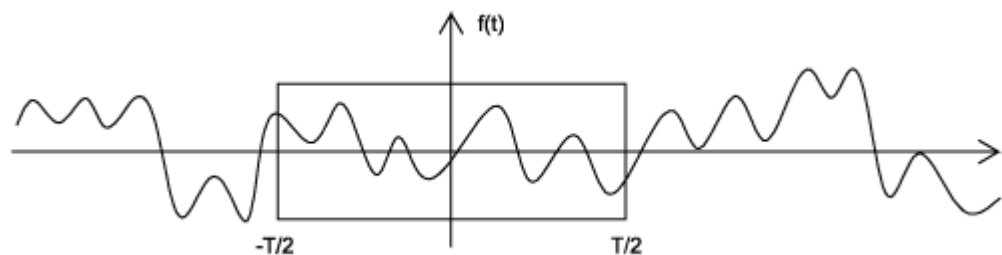


Obr. 10: Energie signálu v časové oblasti [21]

$$E = |\langle f, f \rangle| = \int_{-\infty}^{\infty} |f(t)|^2 dt \quad (12)$$

2. Výkon signálu v libovolném okně

Druhou možností je spočítat výkon signálu v libovolném okně o pevně dané šířce (Obr. 11). Lze jej vypočítat dle vztahu (13).



Obr. 11: Výkon signálu v daném okně [21]

$$E = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} |f(t)|^2 dt \quad (13)$$

Měřicí soustava (Obr. 12) se skládala z wi-fi zařízení (nastaveného do operačního módu AP), spektrálního analyzátoru R&S FSL a osobního počítače s wi-fi kartou – Mikrotik R52. AP a wi-fi karta jsou připojeny pomocí redukce pigtail ke vstupnímu konektoru na spektrálním analyzátoru. Počítač a spektrální analyzátor jsou připojeni na switch, který je pak napojený na internet. Přes vzdálenou plochu je pak možno přistupovat k měřicí soustavě i z domova nebo jiných místností na univerzitě.

Na spektrálním analyzátoru byly nastaveny tyto hodnoty: (Center Frequency – 2,462 GHz, Span 100 MHz, Sweep Time – 2,5ms, Reference Level – 20 dBm, Input Attenuation – 30 dB) – přesný popis nastavení v kapitole 4.1.1 – Nastavení spektrálního analyzátoru. Na měřicí soustavě se uměle vyvolával provoz pomocí příkazů v příkazové řádce (např. ping 192.168.1.254 -f -s 1300). Pomocí SCPI příkazů (Tab. 2) se data stahovala do programu VEE (verze 9.0) od společnosti Agilent.



Obr. 12: Schéma zapojení měřicí soustavy

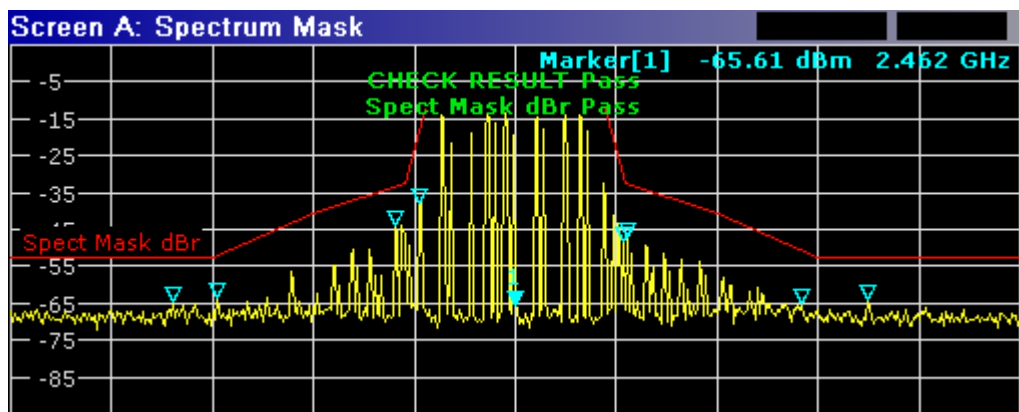
"FREQ:CENT 2.462GHz"	Nastavení Centr. Freq. na 2.462 GHz
"FREQ:SPAN 100MHz"	Span na 100 MHz
"DISP:WIND:TRAC:Y:RLEV 20dBm"	Ref. Level na 20 dBm
"INP:ATT 30 dB"	Input Attenuation na 30dB
"CALC:MARK:FUNC:POW:MODE WRIT MAXH"	Režim zpracování signálu stopy na "max hold"
"CALC1:MARK1:FUNC:POW:RES? CPOW"	Typ power managementu
"TRAC? TRACE1"	Vypsání všech hodnot stopy 1

Tab. 2: Příklady SCPI příkazů

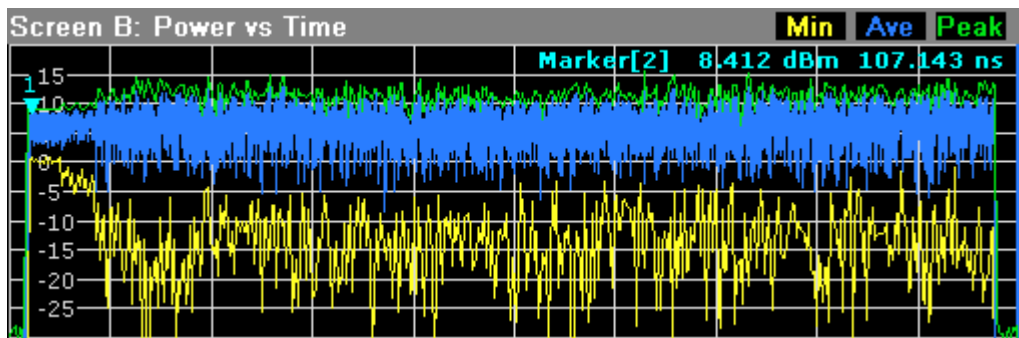
2.3 Porovnání obou metod

Měření v časové oblasti oproti frekvenční přináší výhody, jednou z nich je i menší výpočetní náročnost. Ve frekvenční oblasti je zapotřebí provést rychlou Fourierovu transformaci, kdežto v časové oblasti se bez takového výpočtu obejdeme.

Ukázky výsledků z frekvenční a časové oblasti jsou na obrázcích (Obr. 13), (Obr. 14).



Obr. 13: Frekvenční oblast měření



Obr. 14: Časová oblast měření

3 POUŽITÉ PŘÍSTROJE

3.1 Spektrální analyzátor Rohde & Schwarz FSL



Obr. 15: Spektrální analyzátor R&S FSL

Technické údaje:

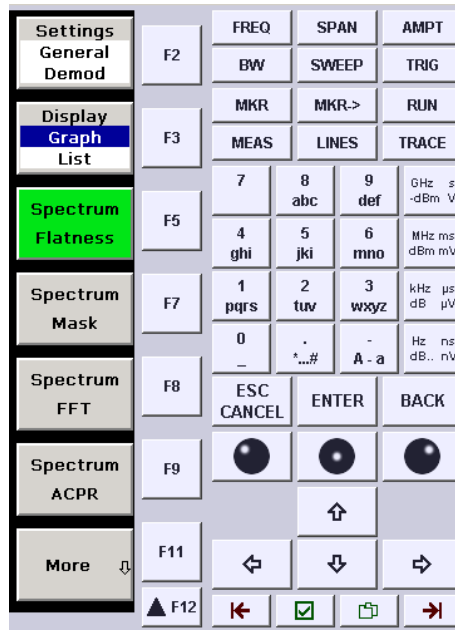
Kmitočtový rozsah:	9 kHz až 6 GHz
Rozlišení kmitočtu:	1 Hz
Doba rozmitání:	2,5ms až 16000s
Konektory:	BNC, USB, RJ-45, PS-2
Úrovně zobrazení:	Log. (10dB – 100 dB), Lin. (0% - 100%)
Módy spektra zobrazení:	Kmitočť v závislosti na čase (FM), amplituda v závislosti na čase (AM), fáze v závislosti na čase (jM), vf výkon v závislosti na čase, vf spektrum (FFT), nf spektrum (FFT), tabulky číselných hodnot pro: kmitočťový zdvih (.pičkový, ef. hodnota), modulační kmitočť, posuv nosné, výkon nosné

3.1.1 Nastavení Spektrálního Analyzátoru

Tak, aby spektrální analyzátor měřil správně a údaje měly vypovídající hodnotu, měl by projít před samotným měřením alespoň patnácti minutovým zahřívacím procesem při pokojové teplotě.

Jak už bylo uvedeno v tabulce (Tab. 2), může se analyzátor nastavit buď pomocí SCPI příkazů nebo druhou možností je využití tlačítek na samotném přístroji (Obr. 16). Díky

velkému grafickému displeji je ovládání analyzátoru pomocí tlačítek mnohem pohodlnější než používání SCPI příkazů. Ty byly v tabulce uvedeny jen jako další funkční možnost.



Obr. 16: Grafický displej FSL s tlačítka nastavení

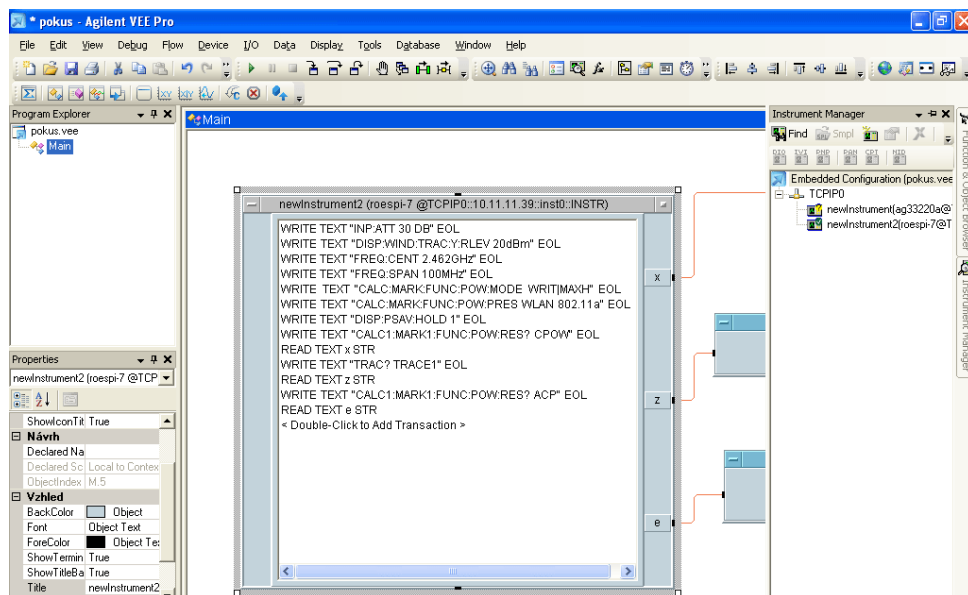
Nejprve je potřeba změnit centrální frekvenci (Center Frequency) pomocí tlačítka FREQ a pak tlačítka Center Freq. Na numerické klávesnici zadáme požadovanou hodnotu. Měření probíhalo na 11 kanálu, proto zadáváme 2462 MHz (2,462 GHz). Dalším parametrem je SPAN, což je rozsah frekvencí okolo centrální frekvence, kterou bude analyzátor zobrazovat. Provedeme stisknutím tlačítka SPAN a nastavením příslušné hodnoty (v našem případě 100 MHz).

Dalším krokem je nastavení detektoru a módu spektra. Přes tlačítka TRACE a pak DETECTOR nastavíme detektor na hodnotu Power versus Time (PVT). Mód spektra se nastaví taktéž stisknutím tlačítka TRACE a TRACE MODE. Z možností vybereme MAX HOLD.

Dobu rozmítání (Sweep Time) nastavíme stisknutím tlačítka SWEEP a nastavením požadované hodnoty (2,5 ms až několik sekund). Posledními kroky jsou nastavení referenční úrovně (20 dBm) a popřípadě atenuátoru (30 dB).

Jakmile jsou všechny hodnoty nastaveny je potřeba simulovat provoz, tak aby bylo možno něco naměřit. Pomocí SSH klienta (program putty) jsme se flood pingem o velikosti packetu 1300 bytů snažili takto provoz vyvolat.

Samotné naměřené hodnoty se pak pomocí SCPI příkazů a programu VEE viz. (Obr. 17) exportují do souboru pro případné další použití.



Obr. 17: Prostředí programu VEE

3.2 Multifunkční wi-fi zařízení ZCOMAX WA2204

Zařízení WA-2204 je multifunkční bezdrátový Accesspoint / Bridge / Router / Klient pro nasazení v pásmu 2,4 GHz dle standardu IEEE 802.11b/g.

Základní funkce:	AP, bridge, router, klient
Splňuje standardy:	802.11b/g
Výstupní výkon:	17dBm (+-3dBm) @ 54Mbps
Bezpečnost, Enkrypcce:	WEP 40,128-bit ; WPA, WPA2, port filtering, IP filtering, filtr MAC adres, port forwarding, DMZ hosting.
Modulace:	CCK(802.11b), OFDM(802.11g)
Počet kanálů:	13 Evropa (ETSI)

3.3 Karta Mikrotik R52

Duální 2,4 + 5 GHz miniPCI adaptér MikroTik R52 využívající chip Atheros AR5414. Karta podporuje mód Super G (108Mbps pro 802.11g).

Rozhraní:	miniPCI
Operační mód:	AP, Client, Ad-HOC
Frekvence:	2.4, 5 GHz
Normy:	802.11b, 802.11g, 802.11a
Max. výstupní výkon:	a: 17 dBm, b: 19dBm, g: 18 dBm
Modulace:	13 Evropa (ETSI)
Spotřeba:	max. 400 mW, průměrně 300 mW



Obr. 18: Mikrotik R52 [22]

4 ZEND FRAMEWORK

Zend Framework (dále jen ZF) je open source, objektově orientovaný, webový aplikační framework implementovaný v PHP 5 a licencovaný pod New BSD licenci. [13]

Hned na úvod by se slušelo zmínit proč vlastně použít pro vývoj aplikací tzv. „Framework“ a nepoužívat klasické PHP resp. objektově orientované PHP. Odpověď na tuto otázku je snadná a velice dobrým příkladem nám může být komunita okolo programovacího jazyka RUBY a jejich Ruby On Rails Frameworku, která se velkou měrou postarala o popularizaci a užívání frameworků jako takových.

Velikou výhodou ZF je implementace Model-View-Controller (MVC) architektury s podporou layoutů a šablonovacího systému, autentizace, autorizace a implementací cache, validátorů nebo filtrů. Tím, že tyto často využívané funkce jsou už sepsány, programátor nemusí ztrácet čas psaním tohoto kódu a může se plně soustředit jen na psaní funkčních celků aplikace.

Za zmínku by mohl stát tzv. router (slouží pro směrování požadavků), který je taktéž převzat z již výše zmíněného Ruby On Rails frameworku nebo “built-in“ jazykové komponenty, které usnadňují práci s lokálním formátem data a také s překladem stránek do jiných jazykových mutací. Výhod oproti klasickému psaní aplikací je mnohem více a budou zmíněny v dalších kapitolách.

4.1 Historie ZF

O Zend frameworku se první informace vynořily na ZendCon (říjen 2005) v San Franciscu. Cílem frameworku bylo poskytnout standardizovanou cestu tvorby PHP aplikací a asistence při RAD (Rapid Application Development) vývoji aplikací.

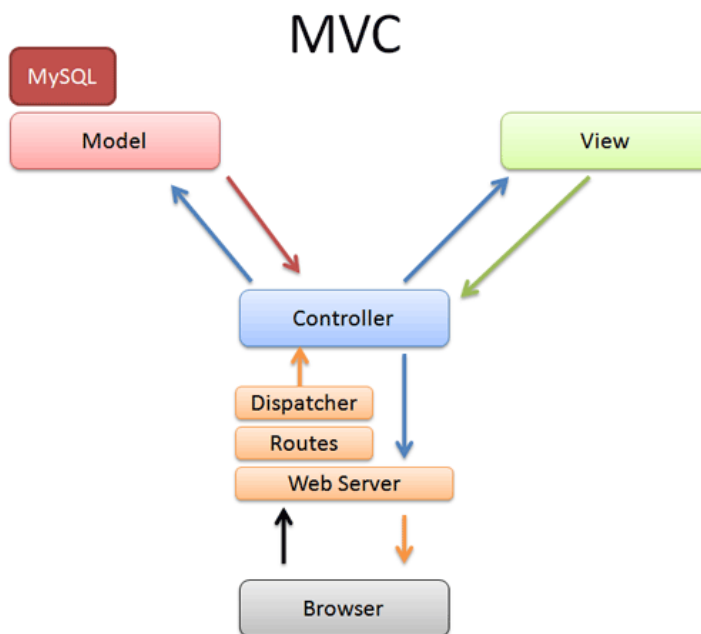
První ostrá verze byla uvolněna v červenci 2007 a již ta obsahovala MVC architekturu, přístupy do DB, vyhledávací engine Lucene, I18N podporu, autentizaci, autorizaci, atp.

Aktuální verze frameworku k současnému datu – květen 2010 je 1.10.4. Stáhnout je možno i verzi 2.0 beta, která ale není vhodná pro nasazení na production server. V nejaktuálnější verzi je potřeba externí komponenty jako např. ORM Doctrine složitě nastavovat a konfigurovat, v budoucí verzi ZF 2.0 by tyto problémy měly být odstraněny.

4.2 MVC

MVC je návrhový vzor (Obr. 19), v kterém je aplikace rozdělena na 3 odlišné části.

- **Datový model aplikace (Model):** Má na starosti správu dat používaných aplikací.
- **Uživatelské rozhraní (View):** Data z modelu jsou uživateli renderována pomocí view.
- **Řídící logika (Controller):** Reaguje na akce uživatele (např. stisknutí tlač.) a zajišťuje změny v modelu a ve view.



Obr. 19: MVC Architektura [20]

Životní cyklus požadavku pomocí MVC návrhového vzoru:

1. Prohlížeč po akci uživatele (např. kliknutí na odkaz) vytvoří požadavek pomocí URL na web server.
2. Jakmile server obdrží požadavek, použije Routes viz. (kapitola 1.3.1) aby zjistil, který controller použít. Dispatcher se pak postará o zavolání příslušné action a předání parametrů.

3. Controller analyzuje uživatelské požadavky, poslané data, cookies a nebo sessions.
4. Pokud je potřeba získat data z databáze (dále jen DB) controller posílá požadavek do modelu, který tyto požadavky vyřizuje. Taktéž se stará o validaci dat.
5. Jakmile controller dostane zpět data z modelu posílá je do view.
6. Nakonec controller vrací odpověď (HTML, XML, atp.) a metadata na server, který je pak zobrazí uživateli.

MVC není jen doménou ZF, ale je implementován i v řadě jiných frameworků, pro ukázkou jmenujme tyto: Symphony, CakePHP, ZNF, Ruby On Rails, Django, Nette nebo ASP.NET MVC. [7]

4.3 Front Controller

Front Controller se stará o samotnou inicializaci MVC vzoru (včetně Routeru a Dispatcheru), ale taktéž o posílání parametrů pomocí standardních superglobálních proměnných (`$_GET`, `$_POST`, `$_SERVER`) do systému pomocí strukturovaných objektů.

4.3.1 Routes

Jakmile Routes dostane objekt od Front Controlleru snaží se najít vhodný controller a action ke spuštění. Otázkou zůstává jak routes pozná který controller, resp. action použít. ZF defaultně používá strukturu ukázanou na obrázku (Obr. 20). Implementovat se samozřejmě dá i pomocí vlastního routeru, kterým si programátor definuje svojí vlastní strukturu.

Jakmile router naplnil své poslání namapováním požadavku k danému controlleru a action, přidá tyto informace k objektu, který poslal Front Controller a pošle jej dále Dispatcheru.



Obr. 20: Routování požadavků do controllerů a actions

4.3.2 Dispatcher

Dispatcher má za úkol (jak už jeho název napovídá) doručit objekt a spustit řídicí logiku, tím že inicializuje metodu v daném controlleru, jehož název si nese s sebou.

Spolu s každou action musí projít další kroky, které dispatcher spouští:

1. Inicializace controlleru (např. eshopController)
2. Volá se metoda `init()`
3. Volá se metoda `preDispatch()`
4. Volá se požadovaná action (např. `produktyAction()`)
5. Volá se metoda `postDispatch()`

Požadovaná action musí být v kódu implementována tak, aby mohla být zavolána na rozdíl od metod `init()`, `preDispatch()` a `postDispatch()`, které nejsou pro bezchybný chod potřebné. Slouží jen pro potřeby programátora a záleží jen na něm jestli je využije. [7]

V metodě `init()` se může např. inicializovat `flashMessenger` (tab. 3), který se stará o zobrazování jak chybových, tak i oznamovacích nebo potvrzovacích zpráv uživateli.

```
3 public function init(){
4     $this->flashMessenger =
5         $this->_helper->getHelper('FlashMessenger');
6 }
```

Tab. 3: Metoda init()

V metodě preDispatch, prováděnou před samotným spuštěním požadované action, si můžeme např. ověřit zda je uživatel přihlášen.

```
3 public function preDispatch() {
4     $auth = Zend_Auth::getInstance();
5
6     //POKUD UŽIV. NENÍ ZALOGOVANÝ -> LOGIN
7     if (!$auth->hasIdentity()) {
8         $this->flashMessenger->addMessage('Zalogujte se!');
9         $this->_redirect('/');
10    }
11 }
```

Tab. 4: Metoda preDispatch()

Zdrojový kód jednoduchého ukázkového eshopControlleru lze vidět v tabulce (Tab. 5). Vypíše jen textovou hlášku “Můj první controller“, ale předtím ještě inicializuje flashMessenger a zkontroluje, zdali je uživatel zalogovaný.

```
2 class EshopController extends Zend_Controller_Action {
3     #viz. init()
4     #viz. preDispatch()
5
6     public function indexAction(){
7         echo "Můj první controller";
8     }
9 }
```

Tab. 5: eshopController

4.3.3 Konvence v pojmenovávání

Pojmenovávání controllerů a actions se řídí striktními pravidly, což umožňuje jejich dynamické spuštění dispatcherem.

Jméno controlleru musí být ve formátu „EshopController“, kde název samotného controlleru je Eshop doplněný o suffix Controller. Název musí začínat velkým písmenem.

Jméno action musí být ve formátu „kosikAction“. Na rozdíl o jména controlleru začíná jméno action malým písmenem.

Například pokud je URL v tomto formátu *http://mojestranky.cz/user/update* musí se zavolat `UserController::updateAction()` [5]

4.4 Nástroje

Pro samotné fungování ZF, je potřeba nainstalovat vývojové prostředí, server, MySQL, PHP a samotný ZF tak, abychom si mohli naprogramovat první “hello world“ aplikaci.

V kapitolách 1.4.1 až 1.4.6 není popsán průběh samotných instalací, jelikož se jedná jen o defaultní nastavení a pro správnou instalaci stačí základní znalosti.

V následující kapitole (1.4.7 Doctrine) se instalaci ORM Doctrine budeme věnovat podrobněji, jelikož nastavení není intuitivní a je potřeba pokročilých znalostí.

4.4.1 Apache

Začneme z instalací úplně nejzákladnějšího softwaru, bez kterého bychom nemohli komunikovat ze světem a to bez serveru.

Apache je open-source HTTP server, vyvíjený již od roku 1996, který lze stáhnout z internetové adresy (<http://httpd.apache.org/>).

4.4.2 MySQL

Poté co se podaří nainstalovat server, další z komponenta, kterou bude potřeba doinstalovat bude databáze k uložení dat potřebných pro aplikaci.

Je možno nainstalovat mnoho různých databázových balíčků, ale MySQL jde s PHP ruku v ruce a práce s ním je pomocí GUI (např. phpMyAdmin) velice jednoduchá. MySQL lze stáhnout z následující adresy (<http://www.mysql.com/>). Popřípadě phpMyAdmin z adresy (<http://www.phpmyadmin.net>)

4.4.3 PHP

Dalším krokem pro vytváření aplikací pomocí ZF je instalace PHP. Z důvodu většího zabezpečení, výkonu, ale taky nových objektově orientovaných vlastností, které v nižších verzích chyběly je potřeba stáhnout, alespoň verzi 5.2.4. Aktuální verzi PHP lze stáhnout z adresy: (<http://www.php.net/>) [14]

4.4.4 ZF

Poslední důležitou součástí, která je potřeba nainstalovat je ZF. Lze jej stáhnout ze stránek (<http://www.zend.com>). Poté co se balíček knihovny stáhne, stačí jej rozbalit do rootu aplikace. Pokud si např. ve složce htdocs/ vytvoříme projekt helloworld, tak by adresářová struktura nově vytvořeného projektu spolu se ZF mohla vypadat následovně:

```
APACHE_HOME/htdocs/helloworld/library/Zend
```

4.4.5 Vývojové prostředí – Netbeans, Zend Studio

Každý pro vývoj aplikace preferuje jiné vývojové prostředí. Někomu stačí pouhý textový editor, někdo preferuje kompletní IDE přizpůsobené pro vývoj aplikací.

Jedním z mnoha open-source editorů je např. Netbeans. Od verze 6.9 i s podporou pro Zend Framework. Lze jej stáhnout z adresy: (<http://netbeans.org/>).

Dalším z řady editorů pro vývoj Zend aplikací je Zend Studio. Toto prostředí je postaveno na Eclipse platformě a jeho největší nevýhodou je, že není zdarma. Trial verzi lze stáhnout z adresy (<http://www.zend.com/products/studio/>).

4.4.6 Zend Tool

Dalším nástrojem pro rychlejší tvorbu aplikací je Zend Tool, který slouží pro generování tzv. kostry projektu, controllerů, modelů a dalších úkonů, které se neustále opakují. Zend tool je součástí ZF a jeho použití je velice jednoduché. Např. v rootu serveru si vytvoříme složku „zf“ do ní nakopírujeme složky *bin* a *library* ze stáhnuté poslední verze ZF.

Poté přes příkazovou řádku spustíme `zf.bat` (OS Windows) nebo `zf.sh` (Linux). Následující tabulka (Tab. 6) obsahuje několik základních příkazů:

Akce:	Příkaz:
Vytvoří nový projekt s adresářovou str.	<code>zf create project helloworld</code>
Vytvoří nový controller v daném projektu	<code>zf create controller name eshop</code>
Vytvoří novou action v daném controlleru	<code>zf create action name show</code>
Projekt bude využívat layout	<code>zf enable layout</code>

Tab. 6: Přehled příkazů Zend Tool

Příkazů, které se dají použít je nepřeborné množství. Zobrazit nápovědu v Zend Tool je možno příkazem: `zf ?`, ten zobrazí možnosti, které může uživatel zadat. Pokud budu potřebovat vytvořit nový controller a nevím jaký příkaz zadat, tak mohu začít hledat např. takto: `zf create ?`, a Zend Tool zobrazí všechny možnosti i s příkladem použití.

4.4.7 Doctrine ORM

Doctrine je Object Relational Mapper (ORM), který nabízí dodatečnou funkcionalitu k DBAL (database abstraction layer). Jednou z hlavních výhod je možnost psát dotazy do DB pomocí DQL (Doctrine Query Language), což je objektově orientovaný SQL dialekt. Své opodstatnění má hlavně při složitějších dotazech, kdy se spojují data více tabulek do jednoho dotazu. Používáním SQL dotazů není v Doctrine nijak limitováno - při jednodušších dotazech lze použít i jen samotný SQL dotaz bez použití DQL. [8]

4.4.7.1 Instalace

Jelikož instalace Doctrine není úplně jednoduchou záležitostí, ukážeme si jak je potřeba postupovat. Nejdříve si na [www stránkách](http://www.doctrine-project.org/) (<http://www.doctrine-project.org/>) musíme stáhnout Doctrine (Květen 2010 – v.1.2.2). Po rozbalení stačí nakopírovat složky Doctrine, vendor a soubor Doctrine.php do složky library v projektu. Ve složce /application/configs/data vytvoříme dvě nové složky sql a fixtures.

Nyní je potřeba nakonfigurovat ZF. Ve složce /application/configs budeme editovat soubor application.ini viz. tabulka (Tab. 7).

Nastavíme ke které DB se má doctrine připojit:	doctrine.dsn = "mysql://root:heslo@localhost/zf_db"
Nastavíme cesty k souborům, které doctrine vyžaduje	doctrine.data_fixtures_path = APPLICATION_PATH "/configs/data/fixtures" doctrine.sql_path = APPLICATION_PATH "/configs/data/sql" doctrine.migrations_path = APPLICATION_PATH "/configs/migrations" doctrine.yaml_schema_path = APPLICATION_PATH "/configs/schema.yml" doctrine.models_path = APPLICATION_PATH "/models"

Tab. 7: Nastavení application.ini

ZF má vlastní autoloader a Doctrine taktéž, proto musíme v souboru Bootstrap.php ve složce /application přidat funkci `_initDoctrine()` z tabulky (Tab. 8), tak aby bylo zajištěno, že oba dva autoloadery budou spolu korektně komunikovat. [15]

```
14 protected function _initDoctrine() {
15
16     $this->getApplication()->getAutoloader()
17         ->pushAutoloader(array('Doctrine',
18                                 'autoload'));
19     spl_autoload_register(array('Doctrine',
20                                 'modelsAutoload'));
21
22     $manager = Doctrine_Manager::getInstance();
23     $manager->setAttribute(Doctrine::
24         ATTR_AUTO_ACCESSOR_OVERRIDE, true);
25     $manager->setAttribute(
26         Doctrine::ATTR_MODEL_LOADING,
27         Doctrine::MODEL_LOADING_CONSERVATIVE
28     );
29     $manager->setAttribute(Doctrine::
30         ATTR_AUTOLOAD_TABLE_CLASSES, true);
31
32     $doctrine = $this->getOption('doctrine');
33
34     //naloadujeme všechny modely korektně
35     Doctrine::loadModels($doctrine['models_path']);
36
37     $conn = Doctrine_Manager::connection(
38         $doctrine['dsn'], 'doctrine');
39     $conn->setAttribute(Doctrine::
40         ATTR_USE_NATIVE_ENUM, true);
41     return $conn;
42 }
```

Tab. 8: Konfigurační nastavení pro Doctrine

4.4.7.2 Tvorba modelů

Jakmile je Doctrine nakonfigurovaná tak, že se korektně připojuje k DB, je potřeba ve složce /application/configs vytvořit soubor schema.yml, ve kterém si vytvoříme celou DB. Jelikož konfigurační soubor je YAML soubor, nesmí se k oddělování kódu používat tabulátor (pouze mezery).

```
1 #schema.yml.  
2 ---  
3 options:  
4   type: INNODB  
5   collate: utf8_general_ci  
6   charset: utf8
```

Tab. 9: Základní nastavení schema.yml

Tvorba modelů je velmi přímočará. Pokud budeme chtít vytvořit model “uživatelé” a přiřadit tomuto modelu sloupce (id, jméno, heslo a email), tak použijeme kód z (Tab. 10).

```
1 #schema.yml  
2 Users:  
3   columns:  
4     id:  
5       type: integer  
6       primary: true  
7       autoincrement: true  
8     username: string(70)  
9     password: string(30)  
10    email: string(100)
```

Tab. 10: Vytvoření modelu Uživatelé

Z výše uvedeného příkladu (Tab. 10) je ekvivalentní zápis v SQL následující:

```
1 CREATE TABLE Users (  
2   id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
3   username VARCHAR(70),  
4   password VARCHAR(30),  
5   email VARCHAR(100)  
6 ) TYPE=innodb;
```

Tab. 11: SQL zápis vytvoření tabulky

Zápis tvorby modelu v YAML souboru se může zdát na první pohled nemotorný, ale hned po vygenerování DB si uvědomíme, že se vygenerovaly i odvozené třídy, které jsou taktéž potřeba. Pro uložení nového uživatele do DB využijeme kód z tabulky (Tab. 12).

```
1 <?php
2     $user = new User();
3     $user->username = "petr";
4     $user->password = "SupertajneHESLO979";
5     $user->email = "petr@domena.cz";
6     $user->save();
7 ?>
```

Tab. 12: Uložení uživatele do DB

Pomocí příkazu *save()* se vytvořený záznam uloží do DB tak, že si Doctrine samo vytvoří spojení s DB a zavolá INSERT dotaz.

4.4.7.3 Asociace mezi modely

Asociace 1:1 je jednou z nejzákladnějších. Máme dvě třídy Uživatel a Email. Ke každé třídě se přidá parametr *relations* s názvem asociované třídy, lokálním a cizím klíčem.

```
1 # schema.yml
2 User:
3 # ...
4 relations:
5   Email:
6     local: id
7     foreign: user_id
8
9 # ...
10 Email:
11   columns:
12     user_id: integer
13     address: string(150)
14   relations:
15     User:
16       local: user_id
17       foreign: id
```

Tab. 13: Asociace 1:1

Asociace 1:M nebo M:1 jsou velmi podobné 1:1. V tomto případě máme dvě třídy Uživatel a Telefon. Předpokládáme, že by uživatel mohl mít více telefonů proto asociace 1:M.

```

1  # schema.yml
2  [-] User:
3  # ...
4  [-] relations:
5  [-]   Phonenumbers:
6         type: many
7         class: Phonenumbers
8         local: id
9         foreign: user_id
10
11 [-] Phonenumbers:
12 [-]   columns:
13         user_id: integer
14         phonenumbers: string(50)
15 [-]   relations:
16 [-]     User:
17         local: user_id
18         foreign: id

```

Tab. 14: Asociace 1:M

Pro získání všech telefonních čísel daného uživatele stačí zavolat `$user->PhoneNumbers`, čímž dostaneme pole se všemi telefonními čísly. Další asociace může být M:N a nebo asociace odkazující sama na sebe. Asociaci odkazující sama na sebe můžeme využít v případech, když chceme vytvořit vazby mezi uživateli (přátelé, potomci, předci, atp.) [8].

4.4.7.4 DQL

V následujících příkladech je pro práci s DQL vyžadována struktura, která je vytvořena v předchozí kapitole. V tabulce (Tab. 15) jsou vypsány ty nejjednodušší a nejzákladnější DQL příkazy. V následující tabulce (Tab. 16) jsou uvedeny složitější příklady. Více příkladů a informací v literatuře [8].

```

//výpis všech sloupců z tabulky User
$query = Doctrine::getTable('User')->findAll();

//výběr jen uživ. jména všech uživ.
$query = Doctrine_Query::create()->select('u.name')->from('User u');

```

Tab. 15: Základní DQL příkazy

```
//výběr uživ. a počtu vláken
$q = Doctrine_Query::create()
    ->select('u.id, COUNT(t.id) AS num_threads')
    ->from('User u, u.Threads t')
    ->groupBy('u.id');

//kombinace left a inner joinů
$q = Doctrine_Query::create()
    ->select('u.id')
    ->from('User u')
    ->leftJoin('u.Groups g')
    ->innerJoin('u.Phonenumber p WITH u.id > 3')
    ->leftJoin('u.Email e');

//SQL ekvivalent výše uvedeného příkladu
SELECT
u.id AS u__id
FROM user u
LEFT JOIN user_group u2 ON u.id = u2.user_id
LEFT JOIN groups g ON g.id = u2.group_id
INNER JOIN phonenumber p ON u.id = p.user_id
AND u.id > 3
LEFT JOIN email e ON u.id = e.user_id
```

Tab. 16: Složitější DQL příkazy [8]

4.5 Psaní kódu

Nedílnou součástí každé aplikace i psaní programového kódu. Programátoři by měli dodržovat tzv. anglickou zkratkou DRY, pod kterou se skrývá anglická věta (Don't Repeat Yourself), což v překladu znamená „Neopakuj se“.

Jak pro přehlednost, tak i pro šetření času i programátorových prstů je vhodné, aby se žádná část kódu v dané aplikaci neopakovala. Logika, která se stále opakuje by měla být konsolidována jen na jedno místo (např. je dobré vytvořit si v modelu funkce, které se dotazují do DB a pak controllerech a ve views jen volat vytvořenou funkci z modelu).

Takových doporučení je vícero, ale je dobré DRY dodržovat. Pokud na projektu pracuje více osob, tak je přímo nezbytné se těmito zásadami řídit a vyhnout se tak případným nepříjemnostem a komplikacím.

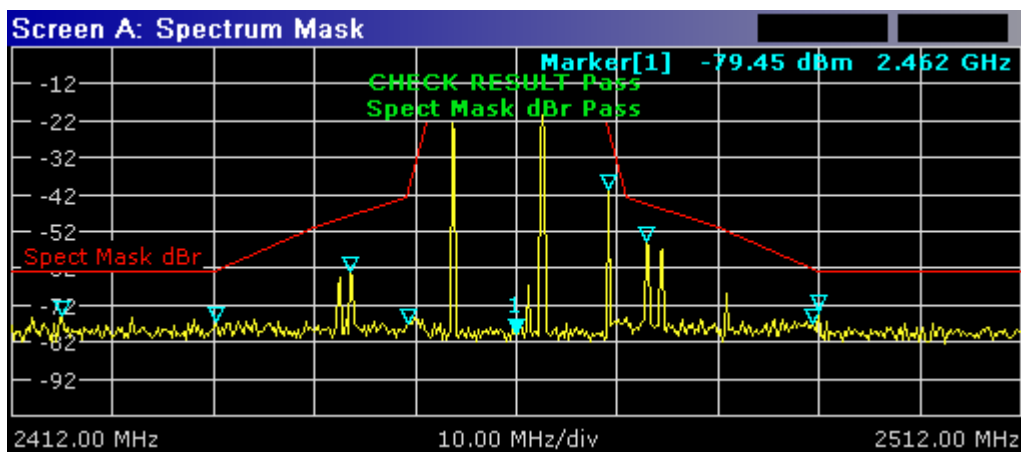
II. PRAKTICKÁ ČÁST

5 MĚŘENÍ VE FREKVENČNÍ OBLASTI

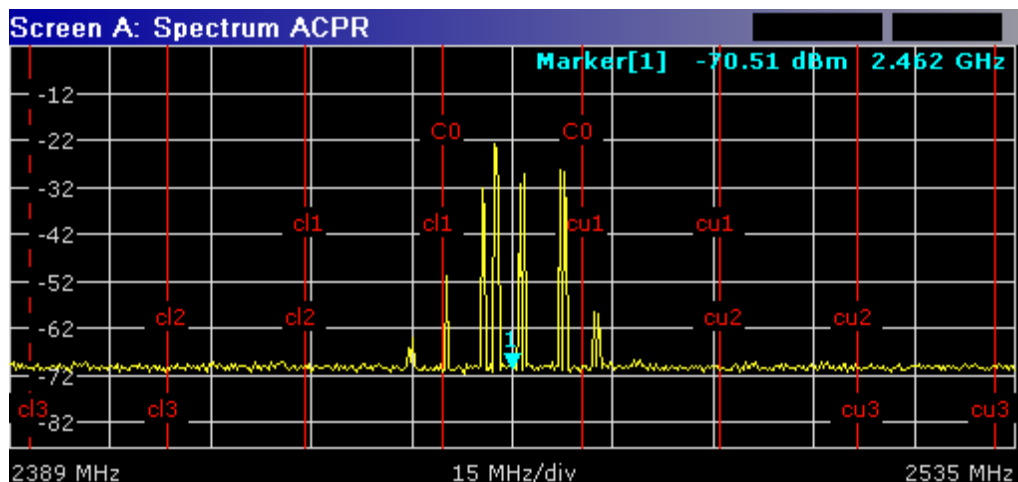
Ve frekvenční oblasti se měření striktně drží doporučení ČTÚ popsaného v kapitole (3.1 Metodika ČTÚ). Samotné měření probíhalo tak, že byl nastaven vysílací výkon karty na hodnotu v rozmezí 1 – 20 dBm (postup bude uveden níže), poté se vyčkalo na vykreslení grafů a pak se výsledné hodnoty uložily.

5.1 Nastavení výkonu AP na 1dBm

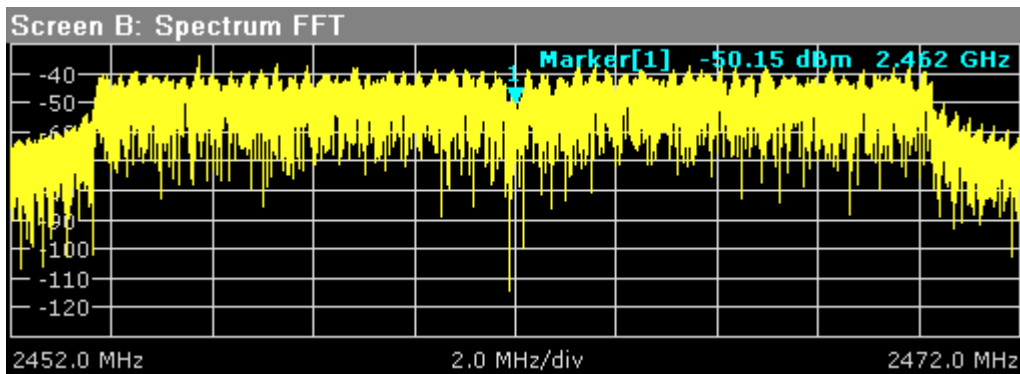
Pomocí příkazu “iwconfig r52 txpower 1“ byl změněn vysílací výkon karty na 1 dBm, bez toho, aniž by muselo být zařízení restartováno. Výsledky jsou uvedeny níže (Obr. 21), (Obr. 22), (Obr. 23), (Obr. 24).



Obr. 21: Spektrální maska



Obr. 22: Spektrum ACPR



Obr. 23: Spektrum FFT

Bursts:	1	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.		4.59	4.59	5.62	4.59	5.62	%
		-26.77	-26.77	-25.00	-26.77	-25.00	dB
EVM Data Carr.		4.60	4.60	5.62	4.60	5.62	%
		-26.74	-26.74	-25.00	-26.74	-25.00	dB
EVM Pilot Carr.		4.44	4.44	39.81	4.44	39.81	%
		-27.06	-27.06	-8.00	-27.06	-8.00	dB
IQ Offset		-31.55	-31.55	-15.00	-31.55	-15.00	dB
Gain Imbalance		-1.75	-1.75		-1.75		%
		-0.15	-0.15		-0.15		dB
Quadrature Err		-0.12	-0.12		-0.12		°
Freq. Err		-23363.79	-23363.79 ± 61550		-23363.79 ± 61550		Hz
Symb Clock Err		-9.55	-9.55 ± 25		-9.55 ± 25		ppm
Burst Power		11.45	11.45		11.45		dBm
Crest Factor		8.21	8.21		8.21		dB

Obr. 24: Tabulkové zobrazení spektra ACPR

5.2 Získání hodnot z analyzátoru

Jakmile je graf vygenerován je možné výsledné hodnoty frekvencí získat pomocí SCPI příkazu "TRAC? TRACE 1". Tento příkaz vrátí řetězec hodnot oddělených čárkami.

5.3 Další měření ve frekvenční oblasti

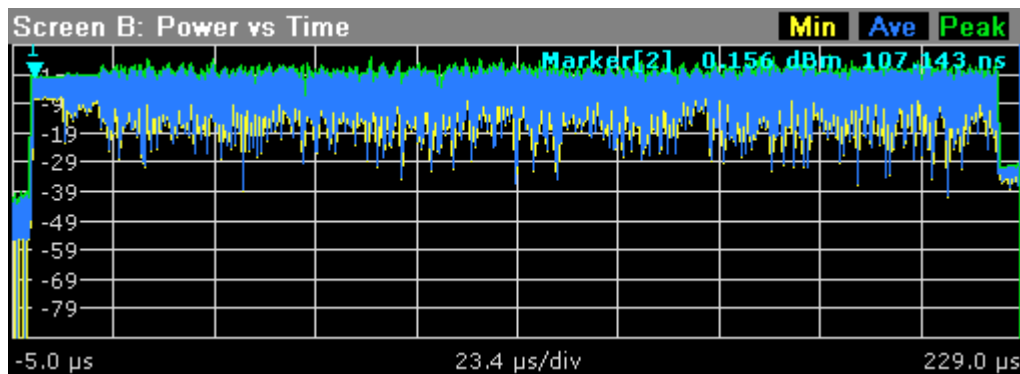
Pro názornost jsou uvedeny jen obrázky prvního měření, ostatní měření, kdy se hodnota vysílacího výkonu měnila až po 20 dBm jsou uloženy na příloženém CD.

6 MĚŘENÍ V ČASOVÉ OBLASTI

Měření v časové oblasti probíhalo naprosto stejně jako u frekvenční, změněn byl akorát mód spektra z RMS na PVT. Změna vysílacího výkonu karty taktéž probíhala pomocí SSH příkazů.

6.1 Nastavení výkonu AP na 1 dBm

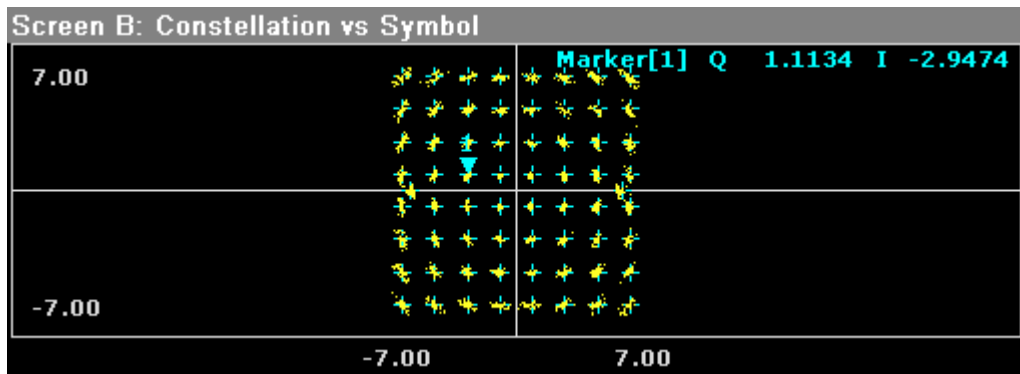
Pomocí příkazu “iwconfig r52 txpower 1“ byl změněn vysílací výkon karty na 1 dBm), bez toho, aniž by muselo být zařízení restartováno. Výsledky jsou uvedeny níže (Obr. 25), (Obr. 26), (Obr. 27), (Obr. 28).



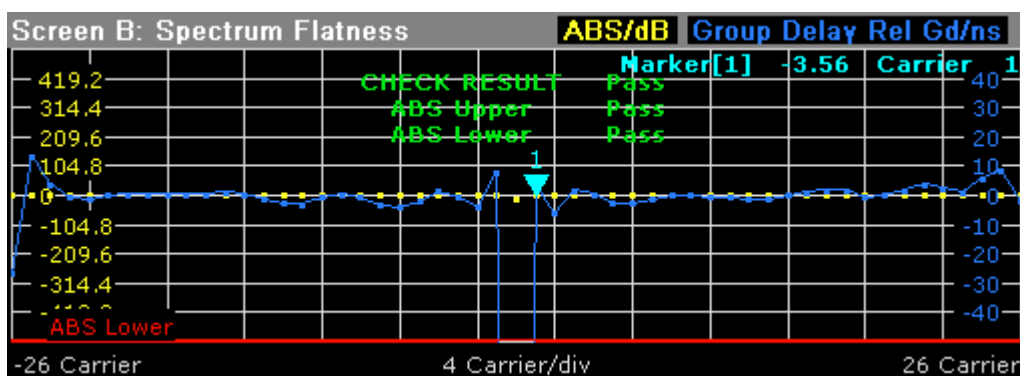
Obr. 25: Power vs. time

Result Summary							
Bursts:	1	Min	Mean	Limit	Max	Limit	Unit
EVM All Carr.		4.25	4.25	5.62	4.25	5.62	%
		-27.44	-27.44	-25.00	-27.44	-25.00	dB
EVM Data Carr.		4.23	4.23	5.62	4.23	5.62	%
		-27.48	-27.48	-25.00	-27.48	-25.00	dB
EVM Pilot Carr.		4.49	4.49	39.81	4.49	39.81	%
		-26.95	-26.95	-8.00	-26.95	-8.00	dB
IQ Offset		-30.19	-30.19	-15.00	-30.19	-15.00	dB
Gain Imbalance		-1.82	-1.82		-1.82		%
		-0.16	-0.16		-0.16		dB
Quadrature Err		-0.07	-0.07		-0.07		°
Freq. Err		-23268.51	-23268.51 ± 61550		-23268.51 ± 61550		Hz
Symb Clock Err		-9.33	-9.33 ± 25		-9.33 ± 25		ppm
Burst Power		-2.57	-2.57		-2.57		dBm
Crest Factor		8.92	8.92		8.92		dB

Obr. 26: Tabulková reprezentace EVM



Obr. 27: Konstelační diagram



Obr. 28: Plochost spektra

6.2 Získání hodnot z analyzátoru

Jakmile je graf vygenerován je možné výsledné hodnoty frekvencí získat pomocí SCPI příkazu “TRAC? TRACE <1|2|3>“. Například v zobrazení spektra PVT jsou k dispozici 3 stopy (Max, Average a Min) proto je nezbytné specifikovat, kterou stopu chceme uložit.

6.3 Další měření v časové oblasti

Pro názornost jsou uvedeny jen obrázky prvního měření, ostatní měření, kdy se hodnota vysílacího výkonu měnila až po 20 dBm, tak jako u frekvenční oblasti jsou uloženy na příloženém CD.

7 TVORBA APLIKACE

V této kapitole je popsána tvorba nejdůležitějších částí aplikace založené na ZF popsaného v kapitole - 2 Zend Framework. Ukázky kódů v této části jsou nekompletní a slouží jen k nastínění dané problematiky. Kompletní zdrojové soubory jsou přiloženy na CD.

7.1 Přihlašovací systém

Bezpečný přihlašovací systém je jednou z nejdůležitějších součástí systému, a proto je nutné věnovat tvorbě zvýšenou pozornost. Přihlašovací systém využívá tyto třídy:

1. **Zend_Form** – tvorba a validace formulářů
2. **Zend_Auth** – autentifikace uživatelů
3. **Zend_Session** – ukládání informací do session
4. **Zend_Mail** – posílání registračních emailů

V souboru schema.yml byla pomocí Doctrine modelů vytvořena databázová struktura (Tab. 17), pojmenovaná Users, pro ukládání informací o uživateli.

```
1 CREATE TABLE Users (  
2     id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
3     sname VARCHAR(70) NOT NULL,  
4     username VARCHAR(70) NOT NULL,  
5     password VARCHAR(30) NOT NULL,  
6     email VARCHAR(100) NOT NULL,  
7     activated BOOLEAN NOT NULL DEFAULT 0,  
8     is_admin BOOLEAN NOT NULL DEFAULT 0,  
9     group_id INT(7) NOT NULL DEFAULT 1  
10 ) TYPE=innodb;
```

Tab. 17: Vygenerovaná struktura tabulky Users

Jakmile je databázová struktura hotová, je potřeba vytvořit registrační a přihlašovací formulář. Ten je umístěn na úvodní stránce tak, aby se návštěvník mohl ihned přihlásit popřípadě kliknout na odkaz a zaregistrovat se. Validace uživatelského jména z registračního formuláře je uvedena v tabulce (Tab. 18).

```
2 $username = new Zend_Form_Element_Text('username');
3 $username->setLabel('Uživatelské jméno')->setRequired(true);
4 $username->addValidator('NotEmpty', true, array(
5     'messages' => "Uživ. jméno nesmí být prázdné!");
6
7 $usernameReg = new Zend_Validate_Regex
8 ('/^[a-zA-Z0-9\_\\-\\.]*$/');
9 $usernameReg->setMessage('Neplatné znaky');
10 $username->addValidator($usernameReg);
11
12 $usernameExists = new Zend_Validate_Db_NoRecordExists
13 ('user', 'username');
14 $usernameExists->setMessage('Uživ. jméno je používáno!');
15 $username->addValidator($usernameExists);
16
17 $usernameSize = new Zend_Validate_StringLength(3,25);
18 $usernameSize->setMessage('Délka 3 - 25 znaků!');
19 $username->addValidator($usernameSize);
20
21 $username->addFilter('StringToLower');
22 $username->addFilter('StringTrim');
23 $username->setAttrib('maxLength', 25);
```

Tab. 18: Ukázka validací registračního formuláře

S proměnnou username je vytvořeno nové textové pole s názvem username. Poté je nastaven popis a hodnota „setRequired“ na true, což znamená, že políčko je potřeba vyplnit. Následně se pak přidávají další validátory, které kontrolují zda zadaný řetězec vyhovuje zadaným podmínkám (nesmí být prázdné, musí být použity jen určité znaky, musí být volné a nebo řetězec musí splňovat určitou délku).

Pokud zadaná data projdou validací a formulář je odeslán pomocí POST metody - mohou být data uložena do DB. Uživateli se odešle email o úspěšné registraci a administrátorovi o novém požadavku ke schválení.

```
2 $user = new User();
3 /* přiřadíme hodnoty */
4 $user->save();
5
6 //odešleme email o nové registraci a potřebě jej autorizovat
7 Mailer::newRegNotice();
```

Tab. 19: Uložení uživatele a odeslání emailu

Jelikož se předpokládá hojně využívání aplikace i v mobilních telefonech, tak se při samotném procesu přihlašování ještě kontroluje zda prohlížeč uživatele podporuje javascript. Kontrola je velice jednoduchá. V samotném přihlašovacím formuláři je proměnná "js", která je defaultně nastavená na 0. Pokud je javascript povolený, tak ji přepíše na 1. Tato hodnota se zapíše do session a v aplikaci se kontroluje jak je nastavena. K této metodě bylo přistoupeno až poté co prohlížeče v mobilních telefonech tagy <noscript> nebraly v potaz.

7.2 Rezervace

Jakmile se uživatel přihlásí do aplikace nemůže hned začít měřit, ale musí si zarezervovat termín měření. K výběru preferovaného data v sekci rezervace slouží měsíční kalendář. Tento kalendář je generován za pomoci PHP třídy dostupné z [23]. Třída obsahovala několik drobných chyb, které bylo potřeba odstranit. Základní funkcionalita třídy kalendáře byla posléze rozšířena tak, aby plně vyhovovala požadavkům aplikace. Především se jednalo o filtraci těch dní v měsíci, které již nejsou aktuální nebo kontrolu správně zadaného data a jestli se o datum vůbec jedná. Tyto body nebyly v základní třídě zapracovány. Podmínky zajišťující správně zobrazená data lze vidět v tabulce (Tab. 20).

```
2  if ($cellNum >= $this->getFirstDayOfWeek() &&
3      $cellNum < ($this->getNumMonthDays() +
4      $this->getFirstDayOfWeek())) {
5      $html .= Zend_Locale_Format::toNumber($calDayNum,
6          array('locale' => $this->localeStr));
7      $actDay = $calDayNum;
8      $calDayNum ++;
9  }
10
11 //Můžeme zobrazit link na rezervaci?
12 if ($this->getRealMonthName() == $monthFromGet) {
13     if (($calDayNum-1 >= $today) &&
14         ($cellNum >= $this->getFirstDayOfWeek()
15         && $cellNum < ($this->getNumMonthDays()
16             + $this->getFirstDayOfWeek())) {
17         /*Zobrazíme link*/
18     }
19 }
```

Tab. 20: Podmínky zajišťující správně zobrazení dat v kalendáři

Poté co si uživatel vybere den měření, zobrazí se mu v tabulce jednotlivé hodiny a popisek zda je daná hodina již rezervovaná nebo volná. Před samotným vykreslením tabulky s daty se provede dotaz do DB na zjištění již rezervovaných hodin.

Výběr hodin lze provést dvěma způsoby (tažením myši nad volnými hodinami nebo výběrem pomocí select boxů). Tím prvním je výběr za pomoci javascriptu, resp. využitím knihovny jQuery a pluginu Selectable.

Bylo zapotřebí knihovnu stáhnout, nalinkovat do hlavičky aplikace a řádek s hodinami k rezervaci označit unikátním id atributem "selectable". Pak jen stačí zavolat kód z tabulky (Tab. 21), aby se choval řádek jako selectable (vybratelný).

```
//výběr datumů pomocí selectable
$( "#selectable" ).selectable(
  { filter: '.free' },
  { //jakmile máme vybráno, zobrazíme tlač. rezervovat
    selected: function(event,ui){ $("#res_btn").fadeIn(); }
  },{ //pak skryjeme
    unselected: function(event,ui){ $("#res_btn").fadeOut(); }
  });
```

Tab. 21: jQuery Selectable

Poté co si uživatel vybere hodinu, kterou by chtěl zarezervovat zobrazí se tlačítko "zarezervovat". A pomocí ajaxu se tyto hodnoty odešlou async controlleru a ten se postará o uložení dat do DB. V tabulce (Tab. 22) je uveden příklad takového volání.

```
$.post("/async/adddate",{ "ids" : cut_ids, "date" : date
}, function(response){
  if(response == "OK"){
    $('.res_mgs').fadeIn(500).delay(1000).fadeOut(500);
    $('.ui-selected').removeClass('free ui-selectee');
    $('.ui-selected').addClass('full');
    $('.ui-selected').text('xxx');
    $('.ui-selected').removeClass('ui-selected');
    $('.temp').fadeOut('slow', function() { $('.temp').remove(); });
    var echo_str = "<li>"+name+"</li>";
  }
});
```

Tab. 22: Ajaxové odeslání dat

7.3 Nové měření

V části “Nové měření“ je zobrazen přehled všech zarezervovaných termínů uživatele a popřípadě i link na spuštění měření, jestliže rozsah rezervovaných hodin odpovídá aktuálnímu času.

Výběr rezervovaných hodin uživatele se provádí voláním definovaných funkcí z modelu viz. (Tab. 23). Do proměnných *aktualne* a *veFronte*, které jsou dostupné ve view, přiřadíme aktuální rezervace, resp. rezervace na další dny. Samotnou funkci z Measure modelu, která vytváří DQL dotaz do databáze lze vidět v tabulce (Tab. 24).

```
//aktuální měření
$this->view->aktualne = Measure::findActual($user->id);

//získáme informace o rezervacích přihlášeného uživ.
$this->view->veFronte = Measure::findAllQueue($user->id);
```

Tab. 23: Uložení dat z databáze do proměnných

```
public function findAllQueue($user_id){
    return Doctrine_Query::create()
        ->from('Reservation r')
        ->Where('r.startdate > ?', date('Y-m-d H:i:s'))
        ->andWhere('r.user_id = ?', $user_id)
        ->orderBy('r.startdate')
        ->execute();
}
```

Tab. 24: DQL dotaz do databáze

Pro výpis aktuálních a budoucích rezervací voláme ve views kód z tabulky (Tab. 25). Kód v aplikaci je mnohem složitější a nepoužívá jen prostou funkci *echo* s výpisem toho co je v daných proměnných. Jde spíše o názornost jak se volají proměnné z controllerů ve views.

```
<?php echo $this->aktualne; ?>
<?php echo $this->veFronte; ?>
```

Tab. 25: Výpis rezervací ve view

7.4 Dokončená měření

Jakmile uživatel dokončí měření je přeměrován do části “Ukončená měření“, kde je přehled všeho co naměřil. Zobrazí se mu jen přehledová tabulka s daty a popisky měření, pro zobrazení detailů musí uživatel kliknout na odkaz “zobrazit detaily“.

Nejdříve je potřeba získat z databáze ukončená měření uživatele a ty pak PHP funkcí *foreach* postupně v tabulce vypíšeme.

Pro uživatele se může zdát, že jsou na stránce načteny jen základní hodnoty (již zmíněný datum a popis měření) a ostatní je načteno až po kliknutí na odkaz “zobrazit detaily“, ale není tomu tak. Vše je načteno, ale detaily měření jsou pomocí jQuery skryté.

Předpokládejme, že každý řádek s detaily je obalen divem, který má class atribut “toggle_container“ a id atribut “completed_45“, kde číslo udává id měření. Chybí už jen označit odkaz pro zobrazování a skrývání detailů, který dostane class atribut “trigger“ a id atribut s číslem měření (pro přesně danou identifikaci měření). Kód javascriptu, který skrývání obsluhuje je uveden v tabulce (Tab. 26).

```
//skryjeme containery s detaily o měření, při načtení stránky
$(".toggle_container").hide();

//rozbalíme a sbalíme detaily o měření
$("p.trigger").toggle(function() {
    $(this).text("").prepend("<a href='#'>- Skrýt Detaily</a>");
}, function () {
    $(this).text("").prepend("<a href='#'>+ Zobrazit Detaily</a>");
});
$("p.trigger").click(function() {
    var m_id = $(this).attr('id');
    $("#completed_"+m_id).slideToggle("slow");
});
```

Tab. 26: Skrývání detailů o měření

Dále je pak možné dané měření smazat (předá se jen parametr s ID měření a zavolá se DQL dotaz s požadavkem o smazání). Pokud má uživatel zapnutý javascript může si zobrazit vygenerovaný graf pomocí faceboxu (lightbox vytvoření pomocí jQuery knihovny) nebo pokud má javascript vyplý zobrazí se obrázek v novém okně prohlížeče.

7.5 Porovnávání měření

Uživatel má možnost porovnat měření s ostatními a získat tak lepší představu o průběžích měření. V detailech může kliknout na odkaz “přidat k porovnání“ a pokud má zapnutý javascript, ID měření se objeví ve formuláři pod výpisy již dokončených měření. Pokud uživatel zadá alespoň dvě, jsou vybrané měření porovnána a následně vygenerován i graf.

Pokud uživatel nemá zapnutý javascript jsou parametry předávány pomocí URL ve tvaru: */measure/completed?compare=9,7,8,2*

Protože si uživatel může libovolně měnit ID měření a prohlížet si měření druhých, provádí se v controlleru kontrola zadaných parametrů v adresním řádku viz. tabulka (Tab. 27). Vyhovují jen kladná celá čísla oddělená čárkou. Vše ostatní je vyfiltrováno pryč.

```
//pokud je vyplý js a snažíme se porovnávat měření
if(isset($_GET['compare'])) {
    //kontrola duplicit
    $compare_arr=""; $tmp_cmp = explode(",", $_GET['compare']);
    if(count($tmp_cmp) > 1) {
        $pattern = "/^[0-9]+$/";
        for( $i=0; $i<count($tmp_cmp); $i++ ) {
            $pos = strpos($compare_arr, $tmp_cmp[$i]);
            if(($pos === false) &&
                (preg_match($pattern, $tmp_cmp[$i])) {
                $compare_arr .= $tmp_cmp[$i].",";
            }
        }
        //$compare_arr = substr($compare_arr,0,-1);
    }
    else {
        $compare_arr = $_GET['compare'].",";
    }
}
```

Tab. 27: Kontrola vložených údajů z URL

7.6 Zpřístupnění měření

Další funkcí pro uživatele je možnost zpřístupnit svá měření jakémukoliv dalšímu registrovanému uživateli. Všichni uživatelé jsou vypsáni ve výběrové nabídce, kde si stačí jednoho vybrat a potvrdit odesláním formuláře.

Uživatelů bude postupně přibývat a na výběr v menu bude hodně uživatelů, proto ti, kteří jsou již vybráni se pomocí atributu "disabled" budou chovat jako ti, na které se nedá kliknout.

```
<select id="allusers" name="user">
  <option value="231" disabled="disabled">Petr Bobek</option>
</select>
```

Tab. 28: Použití atributu disabled

Jakmile je formulář odeslán je vytvořen záznam v tabulce přátelé (Friends), což je jen tabulka, která má dva primární klíče (ID uživatele, který zpřístupňuje měření a ID uživatele, kterému je měření zpřístupněno) a navíc ID měření. Jak se tyto asociace tvoří je uvedeno v tabulce (Tab. 29).

```
//vytvoříme nové uživatele
$petr = new User();
$petr->name = "Petr Bobek";

$pavel = new User();
$pavel->name = "Pavel Novák";

//přiřadíme pavla jako petrova kamaráda
$petr->Friends = $pavel;
```

Tab. 29: Uložení přátel do databáze

7.7 Admin sekce

Do sekce admin je možné přistupovat jen s administrátorskými právy. Tyto práva jsou při každém načtení stránky kontrolována pomocí *preDispatch()* funkce v těle controlleru viz. (Tab. 4) tak, aby se nestalo, že k nastavování práv skupin nebo uživatelů měl přístup někdo jiný.

Samotná tvorba této části není nijak složitá a jedná se spíše o výpisy uživatelských účtů, případné změny skupiny nebo tvorbu samotných skupin.

8 MANUÁL PRO POUŽITÍ APLIKACE

V této kapitole je popsána vytvořená webová aplikace a její jednotlivé části, jak s pohledu uživatele, tak i z pohledu administrátora. Mnoho funkcí bylo pro menší zatížení serveru řešeno pomocí AJAX požadavků, ale jelikož někteří z návštěvníků budou k aplikaci přistupovat z mobilních zařízení, kde prozatím podpora javascriptu ze strany prohlížečů chybí je funkčnost i bezpečnost realizována klasickými GET a POST metodami na straně serveru.

8.1 Úvodní obrazovka a registrace

Úvodní obrazovka aplikace (Obr. 29) slouží jen jako logovací stránka. Přístup na další stránky mají jen registrovaní uživatelé, které musí odsouhlasit někdo z administrátorů (uživatel je poté uvědomen e-mailem). Registrace uživatelů je podmíněna vyplněním základních údajů a opsáním bezpečnostního kódu (captcha) proti nekontrolovanému přílivu “mrtvých“ duší z řad robotů.

Systém pro zpracování dat z měřicích přístrojů

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Nepřihlášen!
[Login](#) | [Registrace](#)

Home | Informace

Login Formulář

 Uživ. jméno

[REGISTRACE](#)
[Zapomněl jsem heslo](#)

Heslo

Novinky:

Novinka 1 - poslední ale bude brzo nahrazena jinou ještě novější novinkou test

Novinka 2 - předposlední

Novinka 3 - starší novinky

Novinka 4 - ještě jedna starší novinka

Novinka 5 - jedna z těch nejstarších novinek

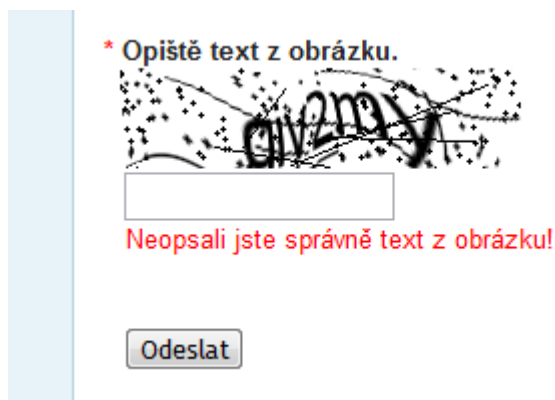
Jako diplomovou práci vytvořil pro UTB ve Zlíně [Petr Bobek](#)

Obr. 29: Úvodní obrazovka

Pokud již zaregistrovaný uživatel zapomene heslo ke svému účtu, má možnost si zaslat heslo nové a získat tak opět přístup ke svému účtu a ukončeným měřením.

8.2 Chybové hlášky

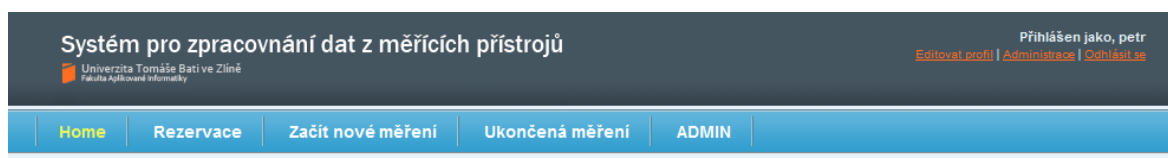
Jak už bylo zmíněno v úvodním odstavci této kapitoly, pokud uživatel zadá, některý z údajů špatně je na to upozorněn chybovou hláškou červené barvy pod daným textovým políčkem viz. (Obr. 30).



Obr. 30: Zobrazení chybové hlášky

8.3 Editace údajů

Po přihlášení má uživatel k dispozici rozšířenou nabídku v menu a navíc v pravém horním rohu možnost odhlášení, editace svých údajů (i svého hesla) a popřípadě přístup do administrátorské sekce.



Obr. 31: Hlavička po přihlášení uživatele

8.4 Rezervace termínu

Jedna z možností v menu je rezervace. Stisknutím tlačítka rezervace si může uživatel vybrat volný termín z kalendáře a tím si zablokovat na několik hodin analyzátor pro sebe. Rezervace je možná pro aktuální nebo následující měsíc a pokud jsou v daný termín ještě hodiny neobsazeny.

Rezervační Systém SPEKTROMETRU:

< květen 2010		červen 2010			červenec 2010 >	
neděle	pondělí	úterý	středa	čtvrtek	pátek	sobota
		1 zarezevuj	2 zarezevuj	3 zarezevuj	4 zarezevuj	5 zarezevuj
6 zarezevuj	7 zarezevuj	8 zarezevuj	9 zarezevuj	10 zarezevuj	11 zarezevuj	12 zarezevuj
13 zarezevuj	14 zarezevuj	15 zarezevuj	16 zarezevuj	17 zarezevuj	18 zarezevuj	19 zarezevuj
20 zarezevuj	21 zarezevuj	22 zarezevuj	23 zarezevuj	24 zarezevuj	25 zarezevuj	26 zarezevuj
27 zarezevuj	28 zarezevuj	29 zarezevuj	30 zarezevuj			

Obr. 32: Kalendář rezervací

V nabídce rezervace se uživateli zobrazí kalendář aktuálního měsíce v němž si může vybrat požadované datum. Jakmile si datum vybere má možnost si vybrat z dostupných hodin. Taktéž má možnost vidět, kdo a na jak dlouho si v daný den zarezoval analyzátor.

Rezervace: 27.Květen 2010

(Tažením myši v oblasti 'volno' vyberete rozsah hodin pro rezervaci.)

07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
volno	volno	xxx	volno	xxx	volno	volno	xxx	xxx	xxx	volno	volno	volno	volno	volno

Zarezevovat

Rezervace uživatelů:
 Petr Bobek: od 9:00 do 10:00
 Petr Bobek: od 11:00 do 12:00
 Veronika Bártová: od 14:00 do 17:00

Obr. 33: Rezervace uživatelů s povoleným js

Pokud má uživatel povolený javascript může využít příjemnějšího ovládání a výběru hodin pouze tažením myši nad preferovanými hodinami. Ti uživatelé, kteří mají javascript

vypnutý nebo jejich prohlížeč (např. v mobilních telefonech) javascript neumí, mohou využít výběr pomocí klasického výběrového menu s výčtem volných hodin.

8.5 Nové měření

Přehled rezervovaných termínů může uživatel vidět po kliknutí na tlačítko “Začít nové měření” (Obr. 34). Pokud je některá z rezervací aktuální, může uživatel začít s nastavováním požadovaných parametrů a provést měření.

Aktuálně Přístupné Měření:

Přístupné od 0:00 do 8:00 - (dostupné ještě 8 hodin) [ZAČÍT MĚŘIT](#)

Vaše Rezervace:

Měření je naplánováno na 26. květen od 15:00 do 16:00 - (přístupné za 15 hodin)
 Měření je naplánováno na 27. květen od 11:00 do 12:00 - (přístupné za 35 hodin)
 Měření je naplánováno na 31. květen od 13:00 do 14:00 - (přístupné za 5 dnů)

Obr. 34: Rezervace a aktuálně přístupné měření

8.6 Ukončená měření

V sekci ukončená měření (Obr. 35) se uchovávají měření, které uživatel ukončil. Řadí se vzestupně od nejaktuálnějšího po nejstarší.

Ukončená Měření:

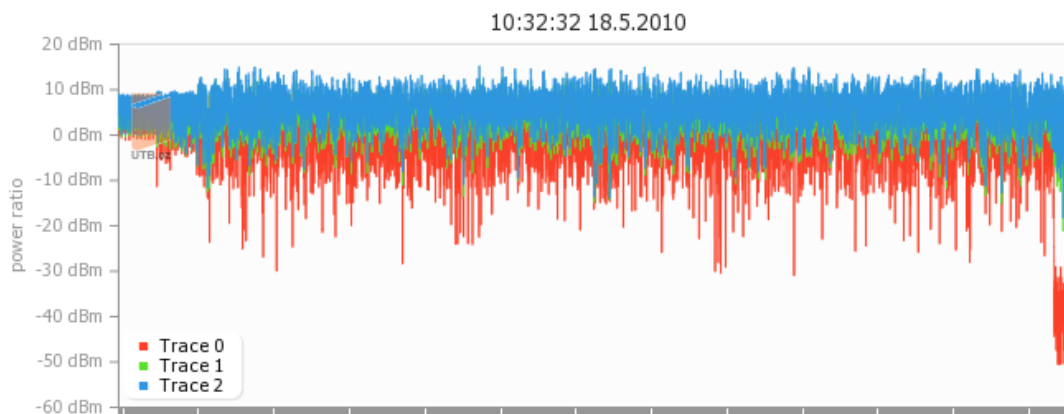
16.5.2010 - 11:15 ACP - ESPI 349 [+ Zobrazit Detaily](#)
 16.5.2010 - 11:16 FSL - měření praha - větší výkon [- Skryt Detaily](#)

Název a datum měření:	Použité nastavení:	Zobrazit graf:	Zpřístupnit uživ.:
Smazat měření Přidat k porovnání	Start: 5.11 GHz Stop: 5.29 GHz Ref Level: 20 dBm Span: 0.101GHz		Vyber uživ. <input type="button" value="Přidej"/> Viditelné pro: Franta Bolf ✗ Veronika Bártová ✗
18.5.2010 - 9:34 trace1			+ Zobrazit Detaily
18.5.2010 - 9:40 trace2			+ Zobrazit Detaily
18.5.2010 - 9:46 trace3			+ Zobrazit Detaily
30.11.1999 - 0:00 ESPI - 4.12			+ Zobrazit Detaily

Alespoň dvě měření musí být vybrána!

Obr. 35: Ukončená měření

U každého měření se zobrazuje jen datum a název měření, popřípadě si uživatel může kliknout na odkaz “více informací“ (možnost smazání měření, porovnání s ostatními, zobrazení grafu (Obr. 36), použitého nastavení a možnost zpřístupnit měření ostatním uživatelům).



Obr. 36: Porovnání více grafů

8.7 ADMIN sekce

Admin sekce je vyhrazena jen a pouze pro administrátory systému. Administrátor zde může povolovat nebo zamítat nově registrované uživatele, vyhledávat uživatele nebo spravovat skupiny, do kterých jsou uživatelé přidělováni.

Jméno skupiny:	Editovat:	Smazat:	Počet členů:	Zobrazit členy:
Nezařazen			7	Petr Bobek, Ada Admin, Janek Dlouhý, Veronika Bártová, Jajda majda, František Friedl, jejda,
uterni_14	Editovat	Smazat	1	Franta Bolt,

Obr. 37: Správa skupin

Může si taktéž zobrazit úplné informace (osobní údaje, počet měření, změnit skupinu, atp.) o vybraném uživateli. Nebo jej může delegovat na stejnou pozici a nastavit práva administrátora.

Informace O Uživateli:

ID: 4
Nickname: devian
Jméno: Veronika Bártová
Email: devi@gmail.com

Skupina:

Počet měření: 0

ADMIN?:

Obr. 38: Zobrazení úplných informací o uživ.

ZÁVĚR

Tato diplomová práce je zaměřena na popis radiových parametrů mobilních zařízení a jejím hlavním cílem bylo vytvoření webové aplikace pomocí programovacího jazyka PHP a libovolně zvoleného frameworku ke spektrálnímu analyzátoru, který je schopný měřit charakteristiky mobilních zařízení.

V teoretické části diplomové práce jsou popsány obecné radiové parametry mobilních zařízení. Dále je zde podrobněji rozebrána specifikace 802.11. V další kapitole je uživatel seznámen s použitými metodami měření (ČTÚ a vlastní metodika), použitými přístroji a jejich nastavením. V poslední části je uveden popis Zend frameworku a jeho nastavení pro správnou funkčnost aplikace. Další potřebné knihovny, aplikace nebo plugíny jsou zmíněny jen okrajově. Výjimku tvoří jen ORM Doctrine, který je popsán podrobněji.

V praktické části jsou uvedeny výsledné grafy jak z frekvenční, tak i z časové oblasti měření. Většina praktické části je ale věnována tvorbě samotné aplikace s ukázkami částí zdrojových kódů a popisem funkčnosti. Pro lepší názornost jsou k jednotlivým částem přiloženy i screeny obrazovky aplikace.

Součástí praktické části je také aplikace pro spektrální analyzátor vytvořená pomocí programovacího jazyka PHP a Zend frameworku. Tato aplikace umožňuje registrovaným uživatelům nastavit počáteční parametry měření a získaná data dále uložit ve formě dat nebo grafů. Jednotlivé grafy mohou být mezi sebou porovnány anebo sdíleny s ostatními uživateli. Programový kód aplikace byl psán s ohledem na další možnou rozšiřitelnost a univerzálnost. V potaz byly brány i možné problémy uživatelů přistupujících k aplikaci z mobilních zařízení, jejichž prohlížeče internetových stránek nepodporují javascript. V poslední kapitole je sepsán manuál použití aplikace, jak z pohledu uživatele, tak i z pohledu administrátora.

V budoucnu by se aplikace mohla rozšířit o několik vylepšení, které by mohly výslednou aplikaci posunout ještě o kousek dále. Uživatelé nejsou prozatím limitováni žádným konečným počtem zarezervovaných termínů, tudíž by se mohlo stát, že by si například jeden uživatel vyblokoval analyzátor na celý týden pro sebe. Jako nejlepší řešení se pak jeví naprogramování jednoduché funkce, která by kontrolovala vytíženost termínů za poslední týden. Pokud by zjistila, že nebyl analyzátor z poloviny času vytížený nezaváděly by se žádné restrikce a naopak pokud by byly všechny hodiny za poslední týden zaplněny, nastavil by se limitní počet rezervací pro jednotlivé uživatele.

Veškeré součásti diplomové práce, jako je aplikace, potřebné knihovny nebo screeny aplikace jsou uloženy na přiloženém CD. Dále je na CD přiložena složka s naměřenými výsledky všech měření jak z frekvenční tak i z časové oblasti – ve formátu PNG. Na CD je nahrána také elektronická podoba diplomové práce ve formátu PDF.

CONCLUSION

This thesis focuses on the characterization of radio parameters from mobile devices and its main aim was to develop web application using PHP programming language, and arbitrarily chosen PHP framework.

The theoretical part describes general parameters of wireless mobile devices. The 802.11 specification is discussed in more details. In the next chapter, the user is familiar with the measurement methods used (our own and CTU methodology), used devices and their configuration. The last section describes the Zend Framework and its settings to assure proper application functionality. Other required libraries, applications and plug-ins are mentioned only in a brief introduction. The only exception is ORM Doctrine, which is described in more detail.

The resulting charts are given in both frequency and from time-domain measurements. Most of the practical part is devoted to making application itself with examples of source codes and descriptions of functionality. The application code was written with a view for possible further expandability and versatility. Potential problems for users accessing applications from mobile devices (the web browser does not support javascript), were taken into account. The last chapter is written as a manual to application uses both a user's perspective and from the perspective of an administrator. To better illustrate the individual parts of the application the pictures of a screen were made.

In the future, application may extend for several improvements, which could shift final application a bit further. Users are not yet limited by any finite number of reserved hours, thus could happen that a user reserves analyzer for the entire week to themselves. The best solution then seems to have a simple function, which would control the hour occupancy in the last week. If found, that the analyzer is not busy half of the week would introduce no restrictions at all, and vice versa if the hours during last week were fully reserved, would set a limit to the number of possible bookings for each user.

All parts of this thesis, such as application, libraries or pictures of the application are stored on CD. There is also attached folder with the measured results of all measurements in both frequency and time domain – as a PNG file format. The CD contains the electronic version of this thesis in PDF format.

SEZNAM POUŽITÉ LITERATURY

Monografie:

- [1] POPE, Keith. *Zend Framework 1.8 Web Application Development*. 1st edition. BIRMINGHAM : Packt Publishing Ltd., 2009. 379 s. ISBN 978-1-847194-22-0.
- [2] PADILLA, Armando. *Beginning Zend Framework*. 1st edition. New York : Apress, 2009. 424 s. ISBN 978-1-4302-1825-8.
- [3] LYMAN, Forrest. *Pro Zend Framework Techniques : Built a Full CMS Project*. 1st edition. New York : Apress, 2009. 266 s. ISBN 978-1-4302-1879-1.
- [4] SHAFIK, Davey. *Zend PHP 5 Certification Study Guide*. 1st edition. Toronto : Nanobooks, 2006. 278 s. ISBN 0-9738621-4-9.
- [5] ALLEN, Rob; LO, Nick. *Zend Framework in Action*. 1st edition. New York : Manning Publications, 2008. 199 s. ISBN 1933988320.
- [6] MACINTYRE, Peter; MORSE, Ian. *Zend Studio™ for Eclipse Developer's Guide*. 1st edition. London : Sams, 2008. 216 s. ISBN 978-0-672-32940-1.
- [7] COGGESHALL, John; TOCKER, Morgan. *Zend Enterprise PHP Patterns*. 1st edition. New York : SPRINGER, 2009. 282 s. ISBN 978-1-4302-1974-3.
- [8] WAGE, Jonathan. *Doctrine ORM for PHP*. Nashville : SensioLabs, 2010. 373 s.
- [9] ROUPHAEL, Tony. *RF and Digital Signal Processing for Software-Defined Radio*. Oxford : Newnes, 2009. 396 s. ISBN 978-0-7506-8210-7
- [10] HANUS, Stanislav. *Bezdrátové a mobilní komunikace*. Brno : VUT, 2003. 135 s. ISBN 80-214-1833-8
- [11] BULA, Jan . *Měření rádiových parametrů WiFi karet*. Zlín, 2007. 59 s. Bakalářská práce. UTB.
- [12] DULÍK, Tomáš. *Měřicí pracoviště s bezpečným vzdáleným přístupem pro výzkum a vývoj aplikací a protokolů WLAN 802.11*, Zlín, 2009

Internetové zdroje

- [13] Zend Framework. In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 19.8.2007, last modified on 30.4.2010 [cit. 2010-05-25]. Dostupné z WWW: http://cs.wikipedia.org/wiki/Zend_Framework.
- [14] Zend.com [online]. 2009 [cit. 2010-05-25]. Zend Framework. Dostupné z WWW: <http://framework.zend.com/about/faq>.
- [15] Doctrine [online]. 2009 [cit. 2010-05-25]. Dostupné z WWW: <http://www.doctrine-project.org/projects/orm/1.2/docs/en>.
- [16] Intelek.cz [online]. 2009 [cit. 2010-05-25]. Intelek. Dostupné z WWW: http://www.intelek.cz/art_doc-5C56A0147621A13AC12575510053AE3E.html.
- [17] Internetprovsechny.cz [online]. 2009 [cit. 2010-05-25]. Internet pro všechny. Dostupné z WWW: <http://www.internetprovsechny.cz/blesk.php?cbl=321>.
- [18] Horevaj.com [online]. 2007 [cit. 2010-05-25]. VF Měřící technika. Dostupné z WWW: <http://www.horevaj.com/horevaj/K4.html>.
- [19] Ni.com [online]. 2007 [cit. 2010-05-25] Sources of Error in IQ Based RF Signal Generation. Dostupné z WWW: <http://zone.ni.com/devzone/cda/tut/p/id/5657>.
- [20] BhartiSoftland.com. 2009 [cit. 2010-05-25] Cake PHP Developments. Dostupné z WWW: <http://www.bhartisoftland.com/technologies-skill-sets/cake-php-developments.html>.
- [21] Cnx.org. 2004 [cit. 2010-05-25] Signal Energy vs. Signal Power. Dostupné z WWW: <http://cnx.org/content/m10055/latest/>.
- [22] Ipmedia.cz. 2009 [cit. 2010-05-25] miniPCI karta Mikrotik R52, Atheros AR5414 Dostupné z WWW: <http://www.ipmedia.cz/?cls=stoitem&stiid=630>.
- [23] Arietis-software.com [online]. 2009 [cit. 2010-05-25]. Arietis Software Innovations. Dostupné z WWW: http://www.arietis-software.com/index.php/2009/05/26/calendar-class-based-on-zend_date/.
- [24] EIRP. In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 22-07-2007, last modified on 05-02-2009 [cit. 2010-05-27]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/EIRP>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PHP	Hypertext Preprocessor
RAD	Rapid Application Development
DRY	Don't Repeat Yourself
MVC	Model View Controller
I18N	Internationalization
URL	Uniform Resource Locator
GUI	Graphic User Interface
ORM	Object-Relational Mapping
IDE	Integrated Development Environment
DQL	Doctrine Query Language
YAML	YAML Ain't Markup Language
SSH	Secure Shell
GPS	Global Position System
EVM	Error Vector Magnitude
PDA	Personal Digital Assistant
EIRP	Equivalent isotropically radiated power
AP	Access Point
QAM	Quadrature Amplitude Modulation
ČTÚ	Český Telekomunikační Úřad
RMS	Root Mean Square (Efektivní hodnota)
R&S	Rohde & Schwarz
PVT	Power versus Time
XML	Extensible Markup Language
HTML	HyperText Markup Language

GUI	Graphical User Interface
SCPI	Standard Commands for Programmable Instruments

SEZNAM OBRÁZKŮ

Obr. 1: Spektrální maska	11
Obr. 2: Konstelační diagram [17]	13
Obr. 3: Detail ideálního a naměřeného signálu s parametry [14]	13
Obr. 4: Konstelační diagramy s různými druhy šumů a chyb	14
Obr. 5: Ideální stav konstelačních bodů	15
Obr. 6: Detail ideálního a naměřeného signálu s parametry, které se dají určit	15
Obr. 7: IQ Imbalance ($Q < 1$) [9]	16
Obr. 8: Konstelační diagram s viditelnou IQ imbalance [9]	16
Obr. 9: IQ Imbalance 4-QAM, 16-QAM a 64-QAM [19]	17
Obr. 10: Energie signálu v časové oblasti [21]	19
Obr. 11: Výkon signálu v daném okně [21]	19
Obr. 12: Schéma zapojení měřicí soustavy	20
Obr. 13: Frekvenční oblast měření	21
Obr. 14: Časová oblast měření	21
Obr. 15: Spektrální analyzátor R&S FSL	22
Obr. 16: Grafický displej FSL s tlačítky nastavení	23
Obr. 17: Prostředí programu VEE	24
Obr. 18: Mikrotik R52 [22]	25
Obr. 19: MVC Architektura [20]	27
Obr. 20: Routování požadavků do controllerů a actions	29
Obr. 21: Spektrální maska	41
Obr. 22: Spektrum ACPR	41
Obr. 23: Spektrum FFT	42
Obr. 24: Tabulkové zobrazení spektra ACPR	42
Obr. 25: Power vs. time	43
Obr. 26: Tabulková reprezentace EVM	43
Obr. 27: Konstelační diagram	44
Obr. 28: Plochosť spektra	44
Obr. 29: Úvodní obrazovka	53
Obr. 30: Zobrazení chybové hlášky	54
Obr. 31: Hlavička po přihlášení uživatele	54
Obr. 32: Kalendář rezervací	55

Obr. 33: Rezervace uživatelů s povoleným js.....	55
Obr. 34: Rezervace a aktuálně přístupné měření	56
Obr. 35: Ukončená měření.....	56
Obr. 36: Porovnání více grafů.....	57
Obr. 37: Správa skupin	57
Obr. 38: Zobrazení úplných informací o uživ.....	58

SEZNAM TABULEK

Tab. 1: Přehled standardů IEEE 802.11	12
Tab. 2: Příklady SCPI příkazů	20
Tab. 3: Metoda init()	30
Tab. 4: Metoda preDispatch()	30
Tab. 5: eshopController	30
Tab. 6: Přehled příkazů Zend Tool	33
Tab. 7: Nastavení application.ini	34
Tab. 8: Konfigurační nastavení pro Doctrine	35
Tab. 9: Základní nastavení schema.yml	36
Tab. 10: Vytvoření modelu Uživatelé	36
Tab. 11: SQL zápis vytvoření tabulky	36
Tab. 12: Uložení uživatele do DB	37
Tab. 13: Asociace 1:1	37
Tab. 14: Asociace 1:M	38
Tab. 15: Základní DQL příkazy	38
Tab. 16: Složitější DQL příkazy [8]	39
Tab. 17: Vygenerovaná struktura tabulky Users	45
Tab. 18: Ukázka validací registračního formuláře	46
Tab. 19: Uložení uživatele a odeslání emailu	46
Tab. 20: Podmínky zajišťující správné zobrazení dat v kalendáři	47
Tab. 21: jQuery Selectable	48
Tab. 22: Ajaxové odeslání dat	48
Tab. 23: Uložení dat z databáze do proměnných	49
Tab. 24: DQL dotaz do databáze	49
Tab. 25: Výpis rezervací ve view	49
Tab. 26: Skrývání detailů o měření	50
Tab. 27: Kontrola vložených údajů z URL	51
Tab. 28: Použití atributu disabled	52
Tab. 29: Uložení přátel do databáze	52

SEZNAM PŘÍLOH

P I Screen webové aplikace

PŘÍLOHA PI: SCREEN WEBOVÉ APLIKACE

Systém pro zpracování dat z měřících přístrojů
Univerzita Tomáše Bati ve Zlíně
Fakulta Aplikované Informatiky

Přihášen jako: petr
[Editovat profil](#) | [Administrace](#) | [Ochlaňit se](#)

[Home](#) | [Rezervace](#) | [Začít nové měření](#) | [Ukončená měření](#) | [ADMIN](#)

VÍTEJTE,

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ut sem in felis molestie condimentum eget a erat. Nunc eros tortor, tincidunt vitae auctor tristique, ornare at est. Cras placerat nunc ut mauris cursus dignissim. Etiam id metus quis ligula consequat commodo non quis lorem. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

10:07 - 21.3.2010

Legend: Sample data, UTB.ac

Y-axis: power ratio (0 dBm, -11.7 dBm, -23.3 dBm, -35 dBm, -46.7 dBm, -58.3 dBm, -70 dBm)

X-axis: time (14, 29, 44, 59, 74, 89, 104, 119, 134, 149, 164, 179, 194, 209, 224, 239, 254, 269, 284, 299, 314, 329, 344, 359, 374, 389, 404, 419, 434, 449, 464, 479, 494)

Novinky:

- Novinka 1 - poslední ale bude brzo nahrazena jinou ještě novější novinkou test
- Novinka 2 - předposlední
- Novinka 3 - starší novinky
- Novinka 4 - ještě jedna starší novinka
- Novinka 5 - jedna z těch nejstarších novinek

Jako diplomovou práci vytvořil pro UTB ve Zlíně Petr Bobek