

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Software pro měření elektromagnetické
kompatibility

Michal Lauterbach

Bakalářská práce

2014

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal Lauterbach**
Osobní číslo: **I11119**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Software pro měření elektromagnetické kompatibility.**
Zadávací katedra: **Katedra informačních technologií**

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

Cílem bakalářské práce je vytvoření softwaru pro měření elektromagnetické kompatibility dle norem ČSN EN 55015 a ČSN EN 61000-3-2. Aplikace bude naprogramována v programovacím jazyce Java a bude zajišťovat ovládání měřících přístrojů připojených k PC, sběr naměřených dat a jejich následné vyhodnocení do měřicího protokolu.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

VACULÍKOVÁ, Polina a Emil VACULÍK. Elektromagnetická kompatibilita elektrotechnických systémů: Praktická příručka techniky omezení elektromagnetického vř rušení. Praha: Grada Publishing, spol. s r.o., 1998, 504 s. ISBN 80-716-9568-8.

HEROUT, Pavel. Učebnice jazyka Java. 5., rozš. vyd. České Budějovice: Kopp, 2010, 386 s. ISBN 978-80-7232-398-2.

HEROUT, Pavel. Java: grafické uživatelské prostředí a čeština. 2. vyd. České Budějovice: Kopp, 2007. ISBN 80-723-2328-8.

ČSN EN 61000. Elektromagnetická kompatibilita (EMC): Část 3-2: Meze pro emise proudu harmonických (zařízení se vstupním fázovým proudem $\leq 16A$). /2010. Praha: Úřad pro technickou normalizaci, meteorologii a státní zkušebnictví, 2010.

ČSN EN 55015 ed. 3:2007 + A1:2008 + A2:2009. Meze a metody měření charakteristik vysokofrekvenčního rušení způsobeného elektrickými svítilny a podobným zařízením. Praha: Český normalizační institut, 2007.

Vedoucí bakalářské práce:

Ing. Richard Capalini, CSc.

Steinel Technik Pardubice

Konzultant bakalářské práce:

Ing. Martin Lauterbach

Katedra elektrotechniky

Datum zadání bakalářské práce:

20. prosince 2013

Termín odevzdání bakalářské práce:

9. května 2014



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2014

Prohlášení autora

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 25.8.2014

Michal Lauterbach

Poděkování

Tímto bych rád poděkoval vedoucímu práce panu Ing. Richardu Capalini, CSc. za odborné vedení a užitečné rady v průběhu práce. Rád bych také poděkoval firmě Steinel Technik, kde jsem tuto bakalářskou práci realizoval.

Děkuji také mé rodině a všem, kteří mě podporovali po celou dobu studia.

Anotace

Cílem bakalářské práce je vytvoření softwaru pro měření elektromagnetické kompatibility dle norem ČSN EN 55015 a ČSN EN 61000-3-2. Aplikace bude naprogramována v programovacím jazyce Java a bude zajišťovat ovládání měřících přístrojů připojených k PC, sběr naměřených dat a jejich následné vyhodnocení do měřícího protokolu.

Klíčová slova

Měření, EMC, elektromagnetická kompatibilita, Java, programování, software

Title

Software for measuring the electromagnetic compatibility.

Annotation

The aim of this thesis is creation of the software for measuring the electromagnetic compatibility in accordance with norm CSN EN 55015 and CSN EN 61000-3-2. The application will be programmed in Java programming language and it will ensure controlling of measuring devices connected to the computer, collecting of measured data and preparing of protocol of measuring.

Keywords

Measurement, EMC, electromagnetic compatibility, Java, programming, software.

Obsah

| | |
|--|-----------|
| Seznam obrázků | 8 |
| Seznam tabulek | 9 |
| Seznam zkratk | 10 |
| Úvod | 11 |
| 1 Programovací jazyk Java | 12 |
| 1.1 Objekty | 12 |
| 1.2 Třídy | 12 |
| 1.3 Metody | 13 |
| 1.4 Grafické uživatelské rozhraní | 13 |
| 1.4.1 JLabel | 13 |
| 1.4.2 JTextField | 14 |
| 1.4.3 JButton | 14 |
| 1.4.4 Další komponenty | 14 |
| 1.5 Sériová komunikace | 14 |
| 1.6 Vykreslování grafů a PDF | 15 |
| 2 Elektromagnetická kompatibilita | 16 |
| 2.1 Elektromagnetické rušení | 16 |
| 2.1.1 Druhy rušení | 17 |
| 2.2 Měření rušivých signálů | 18 |
| 2.2.1 Měření s umělou sítí | 18 |
| 2.3 Měření vyšších harmonických složek | 20 |
| 3 Implementace aplikace | 22 |
| 3.1 Hlavní okno | 23 |
| 3.2 Aplikace pro měření vyšších harmonických | 24 |
| 3.2.1 Použití aplikace | 24 |

| | | |
|-------|---|-----------|
| 3.2.2 | Popis implementace | 27 |
| 3.3 | Aplikace pro měření rušivých signálů | 30 |
| 3.3.1 | Použití aplikace | 31 |
| 3.3.2 | Implementace aplikace | 32 |
| | Závěr | 37 |
| | Literatura | 37 |
| | Příloha A – Ukázkový výstupní protokol | 40 |
| | Příloha B – Ukázkový výstupní protokol | 41 |
| | Příloha C – Diagram aktivit | 43 |
| | Příloha D – Ukázka zdrojového kódu Spectran.java | 44 |
| | Obsah CD | 45 |

Seznam obrázků

| | | |
|---|--|----|
| 1 | Základní řetězec EMC a příklady jednotlivých oblastí. [8] | 16 |
| 2 | Uspořádání pracoviště pro měření rušivého napětí na síťových svorkách. [8] | 19 |
| 3 | Balíčková struktura aplikace. | 22 |
| 4 | Hlavní okno aplikace. | 24 |
| 5 | Okno aplikace pro měření vyšších harmonických. | 25 |
| 6 | Pracoviště pro měření vyšších harmonických signálů | 26 |
| 7 | Okno aplikace pro měření rušivých signálů. | 30 |
| 8 | Okno aplikace pro generování protokolů. | 32 |
| 9 | Pracoviště pro měření rušivých signálů | 33 |

Seznam tabulek

| | | |
|---|--|----|
| 1 | Meze rušivého napětí na napájecích svorkách [11] | 20 |
| 2 | Meze pro zařízení s výkonem do 25 W včetně | 21 |
| 3 | Meze pro zařízení s výkonem nad 25W | 21 |

Seznam zkratek

| | |
|--------------|---|
| ČSN | Česká technická norma |
| EMC | <i>Electromagnetic Compatibility</i> , elektromagnetická kompatibilita |
| EMI | <i>Electromagnetic Interference</i> , elektromagnetická interference |
| LISN | <i>Line Impedance Stabilizing Network</i> , umělá síť (umělá zátěž vedení) |
| AWT | <i>Abstract Windowing Toolkit</i> , grafická knihovna |
| GUI | <i>Graphical User Interface</i> , grafické uživatelské rozhraní |
| API | <i>Application Programming Interface</i> , rozhraní pro programování aplikací |
| SPARC | <i>Scalable Processor Architecture</i> , stupňovatelná architektura procesorů |
| PNG | <i>Portable Network Graphics</i> , formát rastrové grafiky |
| PDF | <i>Portable Document Format</i> , formát pro ukládání dokumentů |
| EPS | <i>Encapsulated PostScript</i> , formát pro ukládání vektorové grafiky |
| SVG | <i>Scalable Vector Graphics</i> , formát pro ukládání vektorové grafiky |
| AGPL | <i>Affero General Public License</i> , licence svobodného softwaru |
| GAE | <i>Google App Engine</i> , platforma pro vývoj aplikací |
| HTML | <i>HyperText Markup Language</i> , značkovací jazyk pro hypertext |
| USB | <i>Universal Serial Bus</i> , univerzální sériová sběrnice |

Úvod

Jakékoliv vyráběné a prodávané elektrické zařízení musí splňovat řadu norem. Mezi ně patří i normy, které určují přípustné hodnoty vyšších harmonických složek a rušivých proudů. Měření těchto hodnot jsou prováděna už při vývoji daného elektrického zařízení. Je ale finančně nevhodné tato měření svěřovat vždy specializované společnosti, protože je často třeba provádět je i několikrát za měsíc. Cílem této práce je tedy vytvoření komplexní aplikace pro měření elektromagnetické kompatibility elektrického zařízení a následné zpracování výsledných dat.

Aplikace bude umožňovat snadné a intuitivní ovládání a bude uživatele přehlednou formou informovat o aktuálním stavu měření, případných problémech, jako jsou náhlé poklesy výkonu měřeného zařízení, nebo při nastavení nevhodné měřicí metody pro daný výkon zařízení. Výsledkem měření bude protokol ve formátu PDF, který bude obsahovat informace o daném měření, jako je jméno uživatele, který měření provedl, datum měření a místo, kde bylo měření provedeno. Dále bude obsahovat přehlednou formou zpracovaná výsledná data. Hodnoty budou zobrazeny buď ve formě tabulek, ze kterých bude jednoznačně patrné, zda hodnoty vyhovují mezím, které jsou dány normou, případně ve formě přehledných grafů.

1 Programovací jazyk Java

Programovací jazyk Java je univerzálním programovacím jazykem, který můžeme, podobně jako např. programovací jazyk Basic, Pascal, C, nebo C++, zařadit mezi tzv. třetí generaci programovacích jazyků. Jedná se o jazyk objektově orientovaný. Za předchůdce Javy je považován jazyk C, respektive jazyk C++, který je, stejně jako Java, objektově orientovaný. Snahou vývojářů Javy bylo eliminovat chyby a zákeřnosti, které jsou typické pro jazyk C++. Vznikl tak jazyk „čistší“ a celkově lépe zvládnutelný. [1]

1.1 Objekty

Java se řadí mezi objektově orientované jazyky, pracuje tedy s objekty. Objekt je v podstatě abstrakcí reality, obsahuje určitá data (údaje, proměnné, datové atributy), se kterými jsou spojeny určité operace. Tato abstrakce by měla být vždy účelová, u každého reálného objektu sledujeme jen ty údaje, které jsou relevantní pro naši aplikaci. [2]

1.2 Třídy

Třída je obecný popis, ve kterém jsou deklarovány datové atributy, které popisují stav objektu, a metody, které umožňují provádět s objekty různé činnosti. Důležitým pojmem, souvisejícím s třídami, je *instance*. Instance je konkrétní datový objekt vytvořený v paměti počítače. K vytvoření instance třídy slouží konstruktor, což je specifická metoda, která může, ale nemusí obsahovat vstupní parametry. Konstruktor má vždy stejný název, jako třída. [2]

Třída s parametrickým konstruktorem

```
public class Trida
{
    int cislo;

    public Trida(int telefon)
    {
        this.cislo = telefon;
    }
}
```

1.3 Metody

Metody realizují určitou činnost (operaci) s daty. Jde o úsek programu, který je možno opakovaně použít, k tomu se využívá tzv. volání metod. V těle metody je typicky posloupnost určitých příkazů, které se provedou, jakmile je metoda zavolána. U každé metody je také uveden návratový typ. Tento typ říká, jakou hodnotu metoda vyprodukuje. Pokud metoda neprodukuje žádnou hodnotu, má návratový typ `void`. [1]

Metoda pro výpis daného počtu pozdravů

```
public void sayHello(int count)
{
    for (int i = 0; i < count; i++)
    {
        System.out.println("Hello!");
    }
}
```

1.4 Grafické uživatelské rozhraní

První implementací grafiky v Javě byla knihovna AWT, která se objevila hned v počátcích vývoje jazyka, ve verzi JDK 1.0 a přetrvává v Javě dodnes. Některé části jsou programátory stále využívány, i když nejčastěji ve věcech obecně souvisejících s grafikou, nikoli přímo s GUI. Filosofie knihovny AWT byla taková, že každá komponenta v Javě měla svůj nativní protějšek v operačním systému, tak jako tomu je u některých GUI knihoven v C/C++. Další dostupnou knihovnou je knihovna Swing. Ta je, na rozdíl od AWT, kompletně nezávislá na platformě a ve velké míře využívá dědičnosti, kompozice a rozhraní. V současné době jde o nejpoužívanější knihovnu pro tvorbu GUI v Javě, která obsahuje velké množství komponent. Některé z nich, které jsou použity i v této práci, jsou uvedeny níže. [3]

1.4.1 JLabel

JLabel je jednou ze základních komponent, která slouží pro zobrazení textu, případně obrázku. Umožňuje nastavení typu písma, jeho barvy, zarovnání a mnoha dalších para-

metrů. Nejpoužívanějšími metodami jsou `void setText(String text)` pro nastavení obsahu a `String getText()` pro získání obsahu `JLabelu`.

1.4.2 JTextField

Komponenta `JTextField` slouží pro zadávání, případně editování jednořádkového textu. Základními metodami jsou, stejně jako u komponenty `JLabel`, `void setText(String text)` a `String getText()`. Stejně jako u komponenty `JLabel` můžeme nastavovat typ písma, barvu písma a navíc můžeme nastavit např. počet zobrazovaných znaků.

1.4.3 JButton

Další ze základních komponent je `JButton`. Jedná se o klasické tlačítko, tak jak ho známe např. z webových formulářů. Toto tlačítko umožňuje vyvolávat různé akce. K tomu je třeba aby byl na tlačítku použit tzv. posluchač (listener), který události (typicky kliknutí myší) zachytává.

1.4.4 Další komponenty

Mezi další často používané komponenty patří `JTextArea`. Ta je podobná komponentě `JTextField`, s tím rozdílem, že umožňuje zobrazení obsahu na více řádcích, hodí se tak např. pro delší texty. Další oblíbenou komponentou je `JProgressBar`. Tato komponenta umožňuje grafické zobrazení stavu určité operace, jako je například stahování, generování souboru apod. Je možno nastavovat různé parametry, jako minimální a maximální hodnotu, aktuální hodnotu, barvu písma, pozadí apod. Nejpoužívanějšími metodami jsou `void setValue(int val)` pro nastavení hodnoty progress baru a `int getValue()` pro její získání.

1.5 Sériová komunikace

Standardní knihovnou, umožňující komunikaci přes sériový port, je knihovna `Java Communications API` (známá také jako `javax.comm`). Toto API nám umožňuje jednoduše získat seznam dostupných portů, nastavovat parametry připojení a samozřejmě také samotnou komunikaci, tedy přenos dat do zařízení a ze zařízení. Bohužel je toto API

použitelné jen v operačních systémech Solaris SPARC, Solaris x86 a Linux x86. [4] Pro operační systém Windows je dostupná alternativa, knihovna RXTX. Tato knihovna poskytuje prakticky ty samé možnosti práce se sériovým portem, jako Java Communications API. Při jejím použití je však třeba dávat pozor, zda se jedná o verzi pro 32-bit, nebo 64-bit verzi operačního systému. [5]

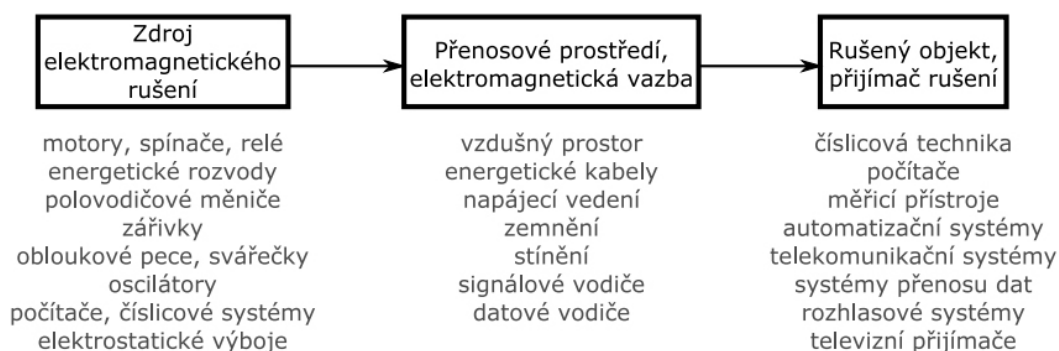
1.6 Vykreslování grafů a PDF

Pro vykreslování grafů je možno použít volně dostupnou knihovnu JFreeChart. Tato knihovna umožňuje snadnou tvorbu různých typů grafů, nastavování mnoha parametrů, jako je barva datové řady, jednotky, typ stupnice apod. Graf je možno zobrazit jako Swing komponentu, výstup je navíc možno uložit ve formě rastrového obrázku ve formátu PNG nebo JPEG, případně vektorového obrázku ve formátu PDF, EPS nebo SVG. [6]

Pro generování souborů ve formátu PDF je možno využít knihovnu iText, která je šířena pod licencí AGPL. Tato knihovna umožňuje snadnou tvorbu PDF souborů, které mohou obsahovat odstavce textu, tabulky, obrázky atd. Umožňuje do souborů přidávat záložky, čísla stránek, vodoznaky, umožňuje spojování, nebo naopak rozdělování již existujících PDF souborů apod. Knihovna je kromě Javy dostupná i pro .NET, Android a GAE. [7] Alternativou pro knihovnu iText je Apache PDFBox, tato knihovna však neposkytuje takové možnosti, jako iText.

2 Elektromagnetická kompatibilita

Elektromagnetická kompatibilita (EMC) je schopnost zařízení nebo systému fungovat vyhovujícím způsobem ve svém elektromagnetickém prostředí¹ bez vytváření nepřijatelného elektromagnetického rušení pro osoby nebo zařízení v tomto prostředí. Místo termínu elektromagnetická kompatibilita se někdy používají české ekvivalenty elektromagnetická slučitelnost a elektromagnetická snášenlivost. Při zkoumání EMC se často vychází z tzv. základního řetězce EMC uvedeného na obr. 1



Obrázek 1: Základní řetězec EMC a příklady jednotlivých oblastí. [8]

2.1 Elektromagnetické rušení

Elektromagnetické rušení (EMI) lze obecně posuzovat jako nežádoucí ovlivňování funkčnosti elektrických nebo elektronických zařízení elektromagnetickou energií vyzařovanou v kmitočtovém spektru od velmi nízkých frekvencí (jednotky Hz) až po desítky GHz. Rušení však nemá vliv jen na určité části kmitočtového spektra využívaného k přenosu signálu, ale jeho působení se projevuje i různými nepřímými účinky na životní prostředí, proto je dnes považováno za problém nejen technický, ale i ekologický. Vysokofrekvenční rušení lze definovat jako nežádoucí ovlivňování normální funkce elektrických nebo elektronických zařízení vysokofrekvenční energií, jejíž kmitočtové spektrum začíná od 10 kHz. Příčinou vzniku tohoto rušení bývá taková činnost elektrického zařízení, při které dochází k náhlé změně elektrického proudu procházejícího jeho obvodem. [9]

¹Elektromagnetické prostředí je souhrn elektromagnetických jevů existujících v daném místě.

2.1.1 Druhy rušení

Elektromagnetické rušení se podle vzniku dělí na dvě základní skupiny:

- technické rušení, tj. rušení vytvořené technickými prostředky,
- přirozené rušení.

Podle směru působení rozlišujeme rušení:

- vnější,
- vzájemné (mezisystémové),
- vnitřní (vnitrosystémové).

Rušivé signály generované technickými zdroji rušení lze podle povahy obecně rozdělit do tří základních skupin:

1. Šum (mezinárodní zkratka N, z anglického noise) – změny projevující se především změnou tvaru křivky. Šum má převážně periodický charakter. Typickými zdroji jsou motory nebo klasické (rotační) svářečky.
2. Impulsy (S, angl. spikes) – změny impulsové povahy, které jsou charakterizované velkým poměrem amplitudy² k době trvání. Tyto impulsy se projevují v napětí sítě jako kladné nebo záporné špičky. Příčinou vzniku impulsů jsou zpravidla spínací pochody. Typickými zdroji jsou veškeré kontaktní spínací přístroje.
3. Přechodné jevy (T, angl. transients) – náhodné jednorázové děje projevující se zpravidla v obálce křivky napětí, s dobou trvání od několika period sítě napětí až po několik sekund. Zpravidla jsou vyvolány náhlou změnou zatížení rozvodné sítě při zapínání a vypínání spotřebičů velkých výkonů, např. vysavačů, vrtaček apod. [9]

²Amplituda je maximální hodnota periodicky měnící se veličiny

2.2 Měření rušivých signálů

Měření rušivých signálů je jednou z nejdůležitějších částí problematiky EMC, představuje totiž praktické ověření dosaženého stupně EMC testovaného zařízení z hlediska jeho rušivého vyzařování. Protože dosáhnout naprosto dokonalé elektromagnetické kompatibility je prakticky nemožné, je nutno stanovit a měřením ověřit dodržení maximálních přípustných hodnot rušivých signálů. Veškeré metody, postupy, podmínky a použité měřicí přístroje pro měření elektromagnetické kompatibility jsou jednoznačně specifikovány příslušnými normami. Měření lze, dle provedení, rozdělit do několika kategorií:

- měření s umělou sítí,
- měření s napětovou sondou,
- měření s proudovou sondou,
- měření s absorpčními kleštěmi,
- měření pomocí antén,
- měření na volném prostranství,
- měření v elektromagnetických stíněných prostorech. [8]

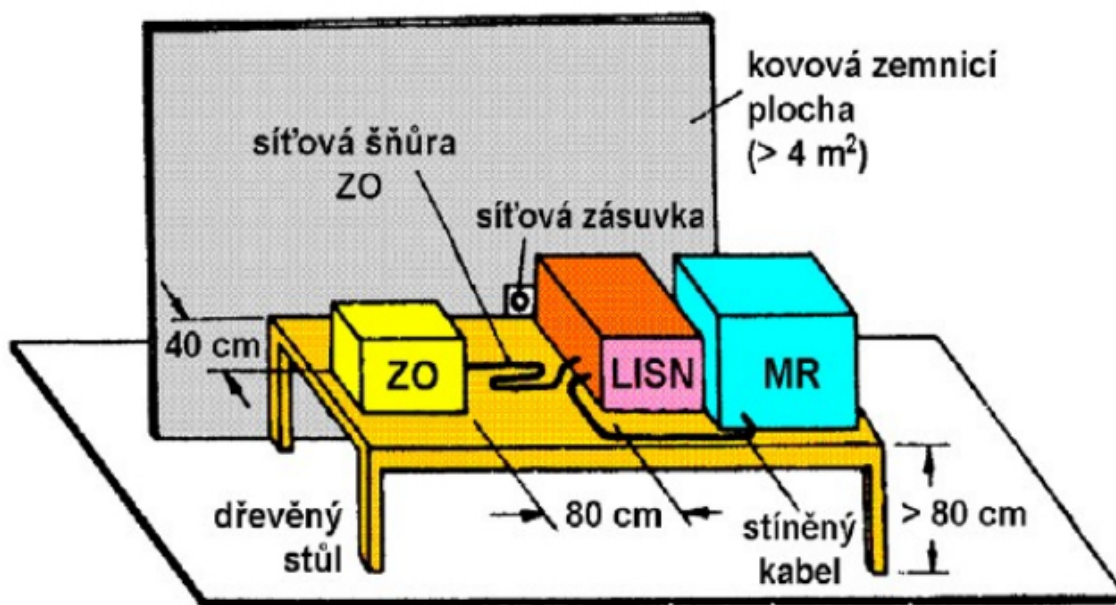
Cílem této bakalářské práce je vytvoření aplikace pro měření s umělou sítí, proto bude této problematice věnována následující kapitola.

2.2.1 Měření s umělou sítí

Umělá síť, někdy také označovaná jako umělá zátěž vedení LISN (Line Impedance Stabilizing Network) se užívá pro měření rušivých signálů, které vznikají v zařízení napájeném z rozvodné elektrické sítě a zároveň ze spotřebiče vystupují tímto napájecím vedením. Umělá síť plní tři funkce:

- připojení měřícího zařízení (měřiče rušení),
- zajišťuje, aby se na vstup měřícího zařízení dostaly jen rušivé signály ze zkoušeného zařízení, nikoli z vnější napájecí sítě,
- zajišťuje impedanční přizpůsobení měřícího zařízení k výstupním svorkám umělé sítě.

Dalším důležitým přístrojem, kterým jsou měřeny všechny druhy elektromagnetických signálů, je měřič rušení (MR), což je selektivní μV -metr, spektrální analyzátor, případně speciální měřicí zařízení měřící na požadovaných kmitočtech, obvykle od 9 kHz do 1 GHz. Pro správné a reprodukovatelné měření je důležité správné uspořádání měřicího pracoviště, které je popsáno normou. Zkoušený objekt (ZO) je společně se zařízením umělé sítě a měřicem rušení umístěn na dřevěném stole tak, aby jeho vzdálenost od nejbližšího okraje zařízení umělé sítě byla 80 cm. Je-li napájecí kabel zkoušeného objektu delší, než 1 m, musí být její nadbytečná část meandrovitě složena, přičemž délka takto vytvořeného svazku nesmí přesáhnout 40 cm. Má-li být zkoušený objekt za provozu uzemněn, musí být jeho zemnění připojeno ke vztaženému zemnímu bodu umělé sítě. Pokud nemusí být zkoušené zařízení uzemněné, musí být umístěno ve vzdálenosti 40 cm od umělé země tvořené svislou kovovou deskou s minimálními rozměry 2×2 m a vzdáleno minimálně 80 cm od každého jiného kovového předmětu, který není součástí měřeného zařízení. Obdobné zásady platí i pro jiná měření rušivých signálů, např. pomocí napěťových a proudových sond. Správné uspořádání pracoviště je vyobrazeno na obr. 2. [8]



Obrázek 2: Uspořádání pracoviště pro měření rušivého napětí na síťových svorkách. [8]

Hraniční úrovně rušivých signálů na jednotlivých frekvencích předepisuje norma ČSN EN 55015 a jsou uvedeny v tabulce 1:

Tabulka 1: Meze rušivého napětí na napájecích svorkách [11]

| Kmitočtový rozsah | Meze [dB μ V] |
|--------------------|-------------------|
| 9 kHz až 50 kHz | 110 |
| 50 kHz až 150 kHz | 90 až 80 |
| 150 kHz až 0,5 MHz | 66 až 56 |
| 0,5 MHz až 5,0 MHz | 56 |
| 5,0 MHz až 30 MHz | 60 |

2.3 Měření vyšších harmonických složek

Harmonické složky signálu jsou sinusová napětí nebo proudy o kmitočtech, které jsou celistvými násobky kmitočtu napájecí sítě (typicky 50 Hz). Harmonické proudy způsobují úbytky harmonického napětí na impedanci sítě. Mezi zdroje generující v síti význačné harmonické proudy patří např. zařízení s fázovým řízením a velkým výkonem nebo neřízené usměrňovače. Hlavními škodlivými účinky harmonických jsou:

- chybná činnost regulujícího zařízení,
- chybná funkce hromadného dálkového ovládání a jiných systémů síťové signalizace, ochran a jiných prostředků řízení,
- přídatné ztráty v kondenzátorech a rozvodech,
- přídatný hluk motorů a jiných přístrojů,
- telefonní interference. [9]

Zařízení jsou podle normy ČSN EN 61000-3-2 rozdělena do čtyř tříd, přičemž pro tuto bakalářskou práci je podstatná třída C – světelná zařízení. Podle této normy jsou také určeny meze pro proudy jednotlivých harmonických, které jsou pro zařízení s výkonem do 25 W včetně uvedeny v tabulce 2 a pro zařízení nad 25 W v tabulce 3:

Tabulka 2: Meze pro zařízení s výkonem do 25 W včetně

| Řád harmonické n | Největší dovolený harmonický proud na watt [mA/W] | Největší dovolený harmonický proud [mA/W] |
|---|--|--|
| 3 | 3,4 | 2,3 |
| 5 | 1,9 | 1,14 |
| 7 | 1,0 | 0,77 |
| 9 | 0,5 | 0,40 |
| 11 | 0,35 | 0,33 |
| $13 \leq n \leq 39$ (jen liché harmonické) | $3,85/n$ | $0,15 \times \frac{15}{n}$ |

Tabulka 3: Meze pro zařízení s výkonem nad 25W

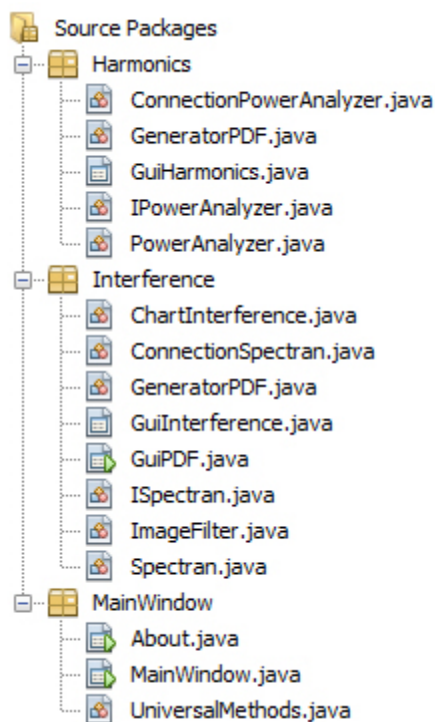
| Řád harmonické n | Největší dovolený harmonický proud [% z vstup. proudu] |
|---|---|
| 2 | 2 |
| 3 | $30 \times \cos \varphi$ |
| 5 | 10 |
| 7 | 7 |
| 9 | 5 |
| $11 \leq n \leq 39$ (jen liché harmonické) | 3 |

3 Implementace aplikace

Aplikace je napsána v programovacím jazyce Java. Tento jazyk jsem zvolil, protože s ním mám oproti jiným programovacím jazykům více zkušenosti a připojení k různým měřicím přístrojům přes sériový port jsem v něm řešil již dříve. Kvůli rozdílným knihovnám pro 32-bit a 64-bit operační systémy je aplikace vydána ve dvou verzích, které se liší právě knihovnami, uživatel musí tedy ke správné funkci zvolit verzi dle svého systému.

Aby byla aplikace a její otevřená okna snadno rozpoznatelná, byla změněna standardní ikona, kterou mají aplikace psané v Javě za ikonu vlastní.

Celá aplikace je rozdělena do tří balíčků. Balíček MainWindow obsahuje třídu s hlavním oknem, dále oknem „O aplikaci“ a také statickou třídu s univerzálními metodami pro celou aplikaci. Balíček Harmonics obsahuje aplikaci pro měření vyšších harmonických a s ní související třídy. Balíček Interference pak obsahuje aplikaci pro měření rušivých signálů a související třídy.



Obrázek 3: Balíčková struktura aplikace.

Adresářová struktura aplikace je pevně daná a pro správnou funkci musí být zachována. Kromě samotného spustitelného souboru s příponou .jar, se v adresáři s aplikací nachází několik dalších souborů a složek. V adresáři *fonts* se nachází soubor obsahující font Arial Unicode MS. Ten je důležitý pro správné zobrazení některých speciálních znaků ve výsledném protokolu. V adresáři *grafy* jsou uloženy tři soubory s výslednými naměřenými grafy. Do tohoto adresáře jsou nastaveny výchozí cesty pro načtení souborů s grafy při tvorbě protokolu měření rušivých signálů. Dále je zde adresář *img*, ten obsahuje různé ikony a loga, které jsou v aplikaci použity. V adresáři *lib* jsou pak uloženy knihovny používané pro tvorbu grafů, generování PDF souborů a sériovou komunikaci se zařízeními. Dočasné soubory jsou pak ukládány do složky *temp*. V hlavním adresáři se pak nachází další potřebné soubory knihoven – *rxtxParallel.dll* a *rxtxSerial.dll*.

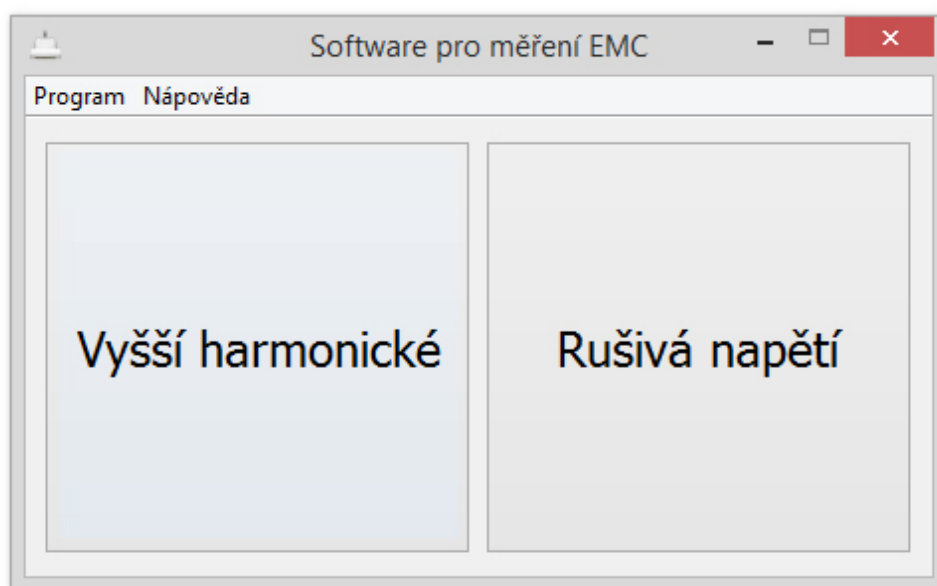
Protože na některých počítačích, na kterých byla aplikace testována, docházelo k problémům se spouštěním aplikace přímo přes .jar soubor, je v adresáři vytvořen ještě soubor *EMC.bat*. Tento soubor obsahuje skript, který umožňuje spustit aplikaci, aniž by muselo být zobrazeno okno s příkazovým řádkem jazyka Java.

Obsah .bat souboru pro spuštění programu

```
start javaw -jar EMC.jar
```

3.1 Hlavní okno

Hlavní okno aplikace je „rozcestníkem“ pro spouštění jednotlivých měřících aplikací – aplikace pro měření vyšších harmonických složek a aplikace pro měření rušivých signálů. Formulář hlavního okna tedy obsahuje dvě tlačítka, jejichž stiskem je vytvořena nová instance patřičného okna a samotné okno je poté zobrazeno. V hlavním okně se dále nachází menu bar pro otevření výběrových menu, které nám umožní ukončení aplikace a zobrazení okna „O aplikaci“. V tomto okně jsou pak zobrazeny informace o aktuální verzi programu a v případných dalších verzích zde bude zobrazován seznam změn (tzv. changelog). Pokud je hlavní okno ukončeno, jsou ukončena i další související okna (otevřené měřící aplikace, grafy apod.).



Obrázek 4: Hlavní okno aplikace.

3.2 Aplikace pro měření vyšších harmonických

Tato měřicí aplikace umožňuje měření vyšších harmonických složek signálu. K měření se využívá výkonového analyzátoru Voltech PM1000+ připojeného k sériovému portu počítače, případně ke konvertoru umožňujícího připojit zařízení se sériovým portem do USB portu počítače. Pro účely této práce byl použit konvertor ASIX UCAB232.

3.2.1 Použití aplikace

Aplikace se otevírá stisknutím tlačítka v hlavním okně (viz kapitola 3.1). Ihned po otevření okna aplikace jsou vyhledány dostupné porty a na základě toho je vytvořen obsah combo boxu³, který umožňuje jejich výběr. Po výběru portu, ke kterému je zařízení připojeno, je třeba stisknout tlačítko „Připojit“. Pokud je v tento moment port využíván (např. jinou aplikací), je na to uživatel upozorněn. V opačném případě se k němu aplikace připojí a uživatel je o tom informován ve stavovém řádku. Po připojení se aktivuje tlačítko pro spuštění měření. Uživatel před samotným měřením vyplní mezní

³Combo box je komponentou grafického uživatelského rozhraní. Umožňuje výběr jedné z položek, které se po kliknutí na něj zobrazují v rozbalovacím menu.

výkon zařízení. Ten je během celého měření kontrolován a pokud dojde k jeho odchylce o více než 1 W (typicky pokud se zařízení vypne, nebo je vypojeno), uživatel je na to upozorněn. Dále je možno změnit čas mezi jednotlivými měřeními. Celkově proběhne deset měření a jejich výsledky se průměrují. Na výběr je ze dvou metod měření – pro zařízení s výkonem do 25 W a pro zařízení s výkonem nad 25 W. Uživatel má dále možnost vyplnit své jméno, typové označení zařízení a místo měření, tyto údaje jsou poté automaticky vloženy do výsledného měřicího protokolu.

Měření vyšších harmonických

Vyberte COM port:

▼ **Připojit** **Odpojit**

Mezní výkon: W

Čas mezi měřeními: s

☒ Metoda pro výkon do 25W

☐ Metoda pro výkon nad 25W

Měřil:

Zařízení:

Místo měření:

MĚŘIT **STOP**

0%

Generovat PDF

Harmonické:

Akt. výkon: W

Prům. výkon: W **Účinnost:**

Čeká se na připojení

Obrázek 5: Okno aplikace pro měření vyšších harmonických.

Pokud jsou všechny údaje správně vyplněny, je možno spustit měření. V základním nastavení (s pětivteřinovou pauzou mezi měřeními) zabere celé měření přibližně jednu minutu. Aktuální průběh měření je možno sledovat na progress baru. Po celou dobu měření se v okně aplikace zobrazuje aktuální naměřený výkon. Ihned po dokončení měření jsou zobrazeny výsledné hodnoty vyšších harmonických a také průměrný výkon, případně průměrný účinník. Pokud je hodnota nižší, než hraniční hodnota, kterou udává norma, pak je zobrazena zeleným písmem. V opačném případě je zobrazena písmem červeným. Uživatel pak na první pohled vidí, zda naměřené hodnoty vyhovují normě, nebo nikoliv. Po dokončení měření je také aktivováno tlačítko pro generování výstupního protokolu ve formátu PDF. Po stisknutí tlačítka dojde k vygenerování protokolu a jeho zobrazení. Výsledný soubor ve svém názvu obsahuje typové označení měřeného zařízení a dále datum a čas měření, což později usnadňuje jeho hledání. Název souboru může tedy být např.

„protokol_mereni_harmonickych_ihf_lampa_10_08_2014_08_55_23.pdf“. Z tohoto názvu je patrné, že měření proběhlo 10. srpna 2014 v 8:55:23 a měřeno bylo zařízení s typovým označením „ihf lampa“.



Obrázek 6: Pracoviště pro měření vyšších harmonických signálů

3.2.2 Popis implementace

Aplikace pro měření vyšších harmonických je rozdělena do těchto tříd:

- *ConnectionPowerAnalyzer* – slouží k navázání spojení a komunikaci se zařízením,
- *GuiHarmonics* – grafické rozhraní aplikace,
- *IPowerAnalyzer* – rozhraní k třídě *PowerAnalyzer*
- *GeneratorPDF* – slouží ke generování protokolu ve formátu PDF
- *PowerAnalyzer* – samotná logika měření.

Pro sériovou komunikaci se zařízením je použita knihovna *RXTX*. K této knihovně byla k dispozici rozsáhlá dokumentace a řada ukázkových řešení. Navíc jsem s ní již měl zkušenosti, protože jsem ji využíval při programování několika předchozích aplikací. Veškerá konfigurace parametrů připojení k měřicímu zařízení a metody potřebné pro komunikaci s tímto zařízením jsou obsaženy ve třídě *ConnectionPowerAnalyzer*. Z parametrů je nejdůležitější přenosová rychlost (tzv. baud rate), počet start-bitů, stop-bitů a nastavení parity. Tyto parametry odpovídají údajům uvedeným v návodu k zařízení. Při nejvyšší možné rychlosti komunikace docházelo k občasným výpadkům komunikace a díky tomu byla data často nekompletní. Bylo tedy nutno použít nižší komunikační rychlost zařízení, po této změně nastavení už probíhá komunikace bez výpadků a data přichází kompletní.

————— Nastavení parametrů připojení —————

```
serialPort.setSerialPortParams(  
    19200,  
    SerialPort.DATABITS_8,  
    SerialPort.STOPBITS_1,  
    SerialPort.PARITY_NONE);  
  
serialPort.setFlowControlMode(3);  
serialPort.setConnected(true);
```

Jak již bylo zmíněno, třída obsahuje také řadu metod, které umožňují komunikaci se zařízením. Metoda `searchPorts()` umožňuje získání seznamu dostupných portů a jejich následné uložení do combo boxu. Pro samotné připojení k zařízení slouží metoda `void connect()`, pro odpojení pak metoda `void disconnect()`. Pro komunikaci se

zařazením je klíčová metoda `void sendData(byte[] data)`, která umožňuje odeslání dat ve formě bajtů do zařízení a metoda `void serialEvent(SerialPortEvent evt)`, která zpracovává data posílaná ze zařízení do počítače. Veškeré události jsou ošetřeny zachycováním výjimek. Případná chybová hlášení jsou pak zobrazována ve stavovém řádku okna aplikace.

Třída *PowerAnalyzer* obsahuje samotnou logiku měření. Důležitou metodou je metoda `void send(String data)`. Ta slouží pro převod textových příkazů na sérii bajtů a jejich následné odesílání do zařízení. Další podstatnou metodou je metoda `void measurement(int method)`, ve které je ověřeno, zda byl vybrán správný typ měření, odpovídající danému výkonu zařízení. Na základě toho je volána buď metoda `void measurementMethod1()`, nebo `void measurementMethod2()`. Tyto dvě metody jsou v podstatě nejzásadnější, proto se jim budu podrobněji věnovat.

V metodě `void measurementMethod1()` je nejprve definována hodnota mezního výkonu a času mezi měřeními, na základě údajů vyplněných v okně aplikace. Poté probíhá nastavení parametrů zařízení, toto se děje zasláním několika zpráv. Ty určují, jaké harmonické budou měřeny (sudé, nebo liché), jejich rozsah apod.

Nastavení parametrů měřicího zařízení

```
send(":SEL:CLR");  
send(":HMX:AMP:SEQ 1");  
send(":HMX:AMP:FOR 0");  
send(":HMX:AMP:RNG 39");  
send(":SEL:AHM");  
send(":SEL:WAT");
```

Poté následuje cyklus 10 měření. V každé iteraci je odeslána zpráva, po které následuje příjem dat ze zařízení. Data přichází ve formě dlouhého textového řetězce, přičemž jednotlivé hodnoty jsou odděleny čárkou. Tento řetězec je parsován (rozdělen) na jednotlivé hodnoty a ty jsou uloženy do jednorozměrného pole. V případě, že je hodnota neplatná, je v původním textovém řetězci zapsána jako tři pomlčky. Ty jsou poté nahrazeny nulou, tak aby bylo možné později hodnotu uložit jako číslo. Ve vnořeném cyklu jsou pak zpracovávány jen potřebné hodnoty a ty jsou ukládány do dvourozměrného pole. První dimenze pole udává pořadí měření, druhá pak pořadí hodnoty. Pokud po dobu měření nedošlo k odchylce od zadaného výkonu, následuje výpočet výsledných

průměrných hodnot. Protože požadavkem bylo, aby bylo na první pohled zřejmé, které hodnoty jsou vyhovující a které ne, bylo třeba zvolit takovou grafickou komponentu, která umožní snadné formátování jednotlivých částí vypisovaného textu, zejména pak barvu textu. Výpis proto probíhá do grafické komponenty *JTextPane*, v níž je pro formátování textu použit jazyk HTML. Hodnoty, které jsou nižší, než hraniční jsou vypsány zelenou barvou, vyšší hodnoty pak červenou barvou.

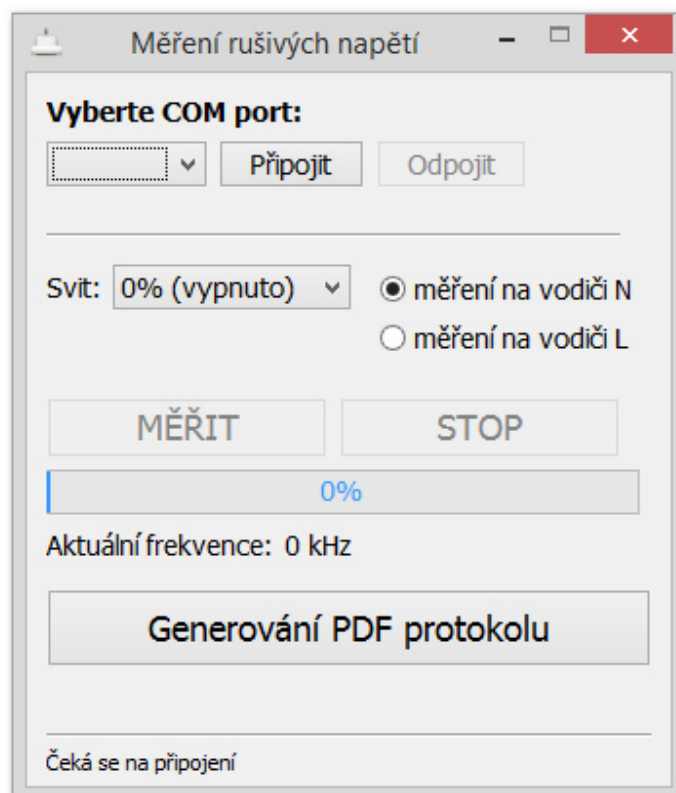
Metoda `void measurementMethod2()` je velmi podobná předchozí popisované. Hlavní odlišnosti jsou v nastavovaných parametrech zařízení, protože při této metodě měření jsou třeba jiné hodnoty, než v předchozím případě. Samotné měření pak probíhá opět v deseti opakováních, jsou vypočítány průměrné hodnoty a ty jsou porovnávány s hraničními hodnotami a vypsány, stejně jako v případě předchozí metody, buď zelenou, nebo červenou barvou. Kromě výše zmíněných metod obsahuje třída také metody pro výpočet hraničních hodnot, se kterými jsou naměřené hodnoty porovnávány a také metodu pro inicializaci prvků GUI.

Jak bylo zmíněno dříve, uživatel má po měření možnost zobrazit PDF protokol. Jeho generování zajišťuje třída *GeneratorPDF*. Ta obsahuje řadu metod pro tvorbu obsahu souboru. Základními metodami pak jsou `void createPdfUnder25W(...)` a `void createPdfOver25W(...)`. V parametrech těchto metod jsou předávány jednak údaje vyplněné v okně aplikace, tzn. jméno osoby provádějící měření, typové označení měřeného zařízení, místo měření, ale i naměřené a vypočítané hodnoty. Obě tyto metody v sobě obsahují volání dalších metod pro tvorbu záhlaví, odstavců textu a pro tvorbu tabulek s naměřenými hodnotami. Příprava skriptů pro tvorbu jednotlivých částí dokumentu byla velmi náročná, u každého prvku je totiž třeba nastavovat řadu parametrů, jako je např. fonty písma, zarovnání, tloušťka čar u tabulek apod. Zároveň bylo také třeba hlídat, aby text „nepřetékal“ přes okraj stránky a také, aby dokument vypadal prakticky stejně, jako vzorový protokol, který jsem měl k dispozici. Další menší překážkou byl fakt, že celý protokol měl být napsaný v německém jazyce, a tak bylo třeba věnovat zvýšenou pozornost správnému zápisu některých slov apod.

```
PdfPCell cellLogo;  
cellLogo = new PdfPCell(logo);  
cellLogo.setRowspan(3);  
cellLogo.setVerticalAlignment(Element.ALIGN_MIDDLE);  
cellLogo.setHorizontalAlignment(Element.ALIGN_CENTER);  
table.addCell(cellLogo);
```

3.3 Aplikace pro měření rušivých signálů

Tato aplikace slouží k měření rušivých signálů za použití spektrálního analyzátoru a zařízení umělé sítě. Pro účely této práce byl použit spektrální analyzátor Spectran NF-5035 X. Toto zařízení se k počítači připojuje přímo přes rozhraní USB a automaticky vytváří virtuální sériový port.

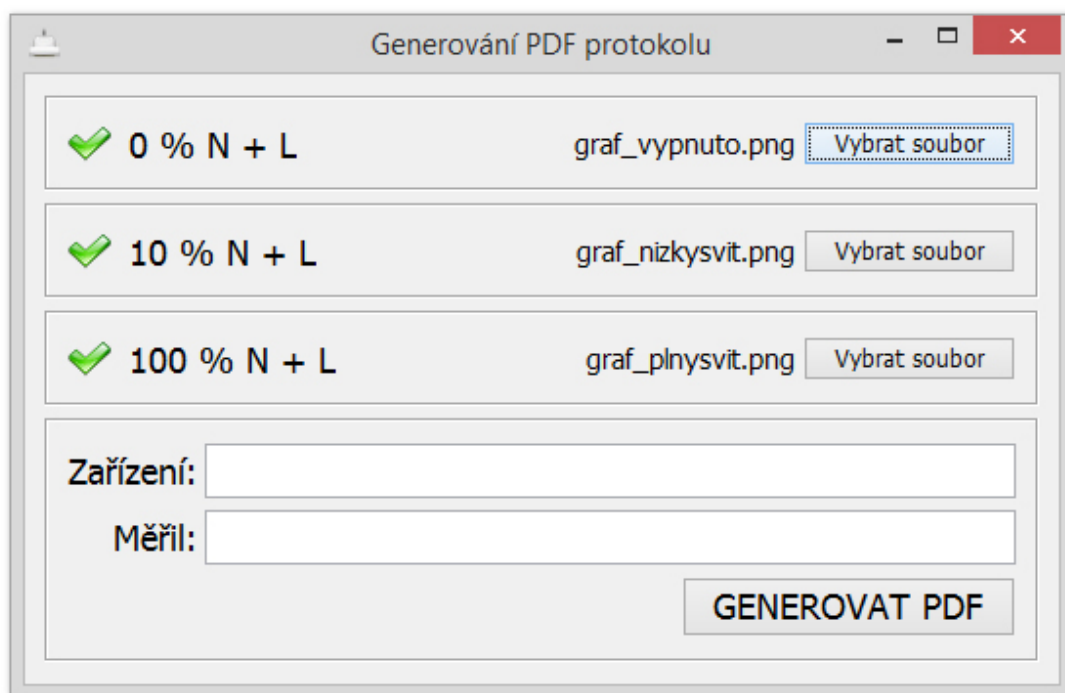


Obrázek 7: Okno aplikace pro měření rušivých signálů.

3.3.1 Použití aplikace

Aplikace se, stejně jako aplikace pro měření vyšších harmonických, otevírá stisknutím tlačítka v hlavním okně (viz kapitola 3.1). Ihned po spuštění jsou vyhledány dostupné porty, ze kterých uživatel vybere a stiskne tlačítko „Připojit“. Pokud je připojení k portu možné, spojení se zařízením je navázáno a ve stavovém řádku se objeví text „Připojeno“. V opačném případě je uživatel upozorněn, že je port již využíván. Před měřením je třeba zkontrolovat, zda nastavení svitu zařízení a nastavení zařízení umělé sítě odpovídá vyplněným údajům v okně aplikace. Poté je možno začít měření. Po stisknutí tlačítka „MĚŘIT“ se zobrazí okno s grafem měřených hodnot. V něm je možno v reálném čase sledovat hodnoty, zároveň se v okně aplikace zobrazuje procentuální průběh měření a aktuální měřená frekvence. Jedno měření trvá zhruba 10 minut. Po jeho dokončení je uživatel vyzván, aby změnil nastavení na zařízení umělé sítě, případě změnil nastavení svitu lampy. V okně aplikace jsou údaje změněny automaticky, uživatel tak nemusí nic měnit, jen opět stiskne tlačítko pro zahájení měření. Celkově proběhne šest měření, jejichž výsledkem jsou tři grafy. V každém grafu jsou zároveň naměřená data na vodiči N a L, která se liší barvou datové řady. Grafy jsou ukládány ve formátu PNG a je možno je zobrazit i po ukončení měření.

Po dokončení měření je možno vygenerovat PDF protokol. Aplikace pro jejich generování má automaticky nastavené cesty k souborům s naměřenými grafy. Uživatel tedy vyplní jen své jméno a typové označení zařízení a stiskne tlačítko „Generovat PDF“. Ihned po vygenerování protokolu dojde k jeho otevření. Aplikace pro generování je spustitelná nezávisle na předchozím měření, je tedy možno vygenerovat i protokoly s manuálně vybranými soubory např. z dřívějších měření. Výběr zmíněných souborů je indikován ikonkami. V případě, že soubor nebyl vybrán, nebo se jej nepodařilo načíst, je zobrazena ikonka s červeným křížkem. V opačném případě je zobrazena ikonka se zeleným symbolem.



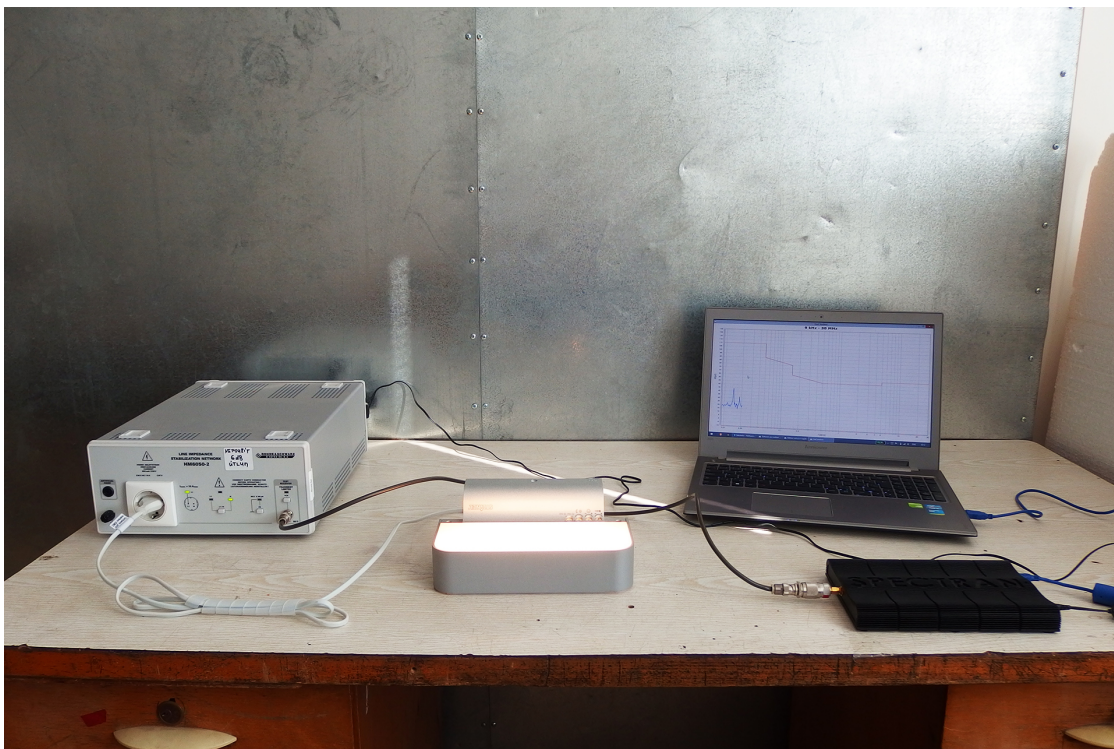
Obrázek 8: Okno aplikace pro generování protokolů.

3.3.2 Implementace aplikace

Aplikace pro měření rušivých signálů je rozdělena do těchto tříd:

- *ChartInterference* – zajišťuje zobrazení okna s grafem a vykreslování hodnot,
- *ConnectionSpectran* – slouží k navázání spojení a komunikaci se zařízením,
- *GeneratorPDF* – skripty, pomocí nichž je generován PDF soubor,
- *GuiInterference* – grafické rozhraní měřicí aplikace,
- *GuiPDF* – grafické rozhraní aplikace pro tvorbu PDF protokolů,
- *ISpectran* – rozhraní k třídě Spectran
- *ImageFilter* – filtr pro vymezení formátů otevíraných souborů
- *Spectran* – samotná logika měření.

Třída *ChartInterference* slouží k zobrazení okna s grafem a umožňuje zobrazování měřených hodnot v reálném čase. Pro tvorbu grafu je využito knihovny *JFreeChart*. Na internetu je k této knihovně dostupná rozsáhlá dokumentace a ukázkové příklady použití, ze kterých jsem vycházel. Graf obsahuje tři datové řady – *thresholdValues*, *seriesN* a *seriesL*. První slouží k zobrazení hraniční čáry, kterou by výsledné naměřené



Obrázek 9: Pracoviště pro měření rušivých signálů

hodnoty neměly přesáhnout. Zbylé dvě slouží pro vykreslení naměřených hodnot. Body do datové řady se přidávají metodou `void add(x, y)`, první parametr udává hodnotu bodu na ose X, druhý parametr na ose Y.

— Skript pro přidání datové oblasti —

```
thresholdValues.add(0.009, 110);  
thresholdValues.add(0.050, 110);  
thresholdValues.add(0.050, 90);  
thresholdValues.add(0.150, 80);  
thresholdValues.add(0.150, 66);  
thresholdValues.add(0.500, 56);  
thresholdValues.add(5, 56);  
thresholdValues.add(5, 60);  
thresholdValues.add(30, 60);
```

Dále jsou ve třídě nastaveny rozsahy jednotlivých os, logaritmické měřítko, barvy datových řad a tloušťka jejich čar. Třída také obsahuje metody `void addValueN(double x, double y)` a `void addValueL(double x, double y)`, které slouží pro přidání jednotlivých hodnot do příslušné datové řady.

Nejdůležitější třídou je třída *Spectran*. Ta obsahuje kromě jiného také definice kódů jednotlivých nastavitelných parametrů zařízení.

Tyto kódy i s popisem jednotlivých paketů jsou popsány v programovací příručce k zařízení. V této příručce ale bylo několik zcela zásadních chyb, které vývoj aplikace výrazně zdržovaly. Chyby byly jednak v popisech paketů, které bylo třeba v aplikaci správně identifikovat právě podle pořadí bajtů v nich, ale také v jednotkách, ve kterých jsou určité parametry nastavovány. Tato chyba byla objevena až po důsledném zkoumání odchozích paketů z aplikace MCS Spectrum Analyzer, která je k spektrálnímu analyzátoru výrobcem dodávána. Výrobce měřicího zařízení byl na chyby poté upozorněn a v nové verzi příručky jsou již chyby upraveny.

Třída dále obsahuje řadu metod, které zajišťují průběh měření. Jednou z nich je metoda `void showChart()`, která zajišťuje otevření okna s grafem a jeho následnou maximalizaci. Je volána přes započítím měření, ale jen v případě, že okno s grafem již není otevřeno – tedy v případě, kdy jde o první část měření při daném nastavení svitu měřeného zařízení. Metoda `void powerPeakMeas(int type)` zajišťuje inicializaci spektrálního analyzátoru a poté samotné měření. Inicializace probíhá odesláním dvou speciálních ověřovacích paketů. Poté už probíhá samotné měření. Pro každou část měření je následně volána metoda `cyclePowerPeak(float min, float max, float step, int type)` s různým nastavením počáteční a koncové frekvence, velikostí kroku mezi frekvencemi a typem měření (zda se jedná o měření na vodiči N, nebo L). V této metodě proběhne cyklus, který v každé iteraci zajišťuje získání maximální hodnoty signálu na dané frekvenci. Protože zařízení měří v jednotkách Volt, je třeba každou hodnotu převést na jednotky dBμV a poté je možno ji vykreslit do grafu.

Ve třídě je také použito několik pomocných metod pro přípravu paketů posílaných do zařízení a také pro zpracování přijatých dat. Protože data ze zařízení chodí jako série bajtů v endianitě⁴ typu Little Endian, je třeba je zpracovat a převést na jiný datový typ – v tomto případě na float. K tomu slouží metoda `float convertIncomingByteToFloat(byte[] bytes)`. Ta zajistí správné pořadí bajtů, jejich následný bitový součet a převod na datový typ float.

⁴Endianita je způsob uložení čísel v operační paměti. Definuje, v jakém pořadí jsou bajty uloženy.

Metoda pro převod přijatých dat na datový typ float

```
private float convertIncomingByteToFloat(byte[] bytes)
{
    int[] temp = new int[4];
    temp[0] = (bytes[5] & 0x000000FF) << 24;
    temp[1] = (bytes[4] & 0x000000FF) << 16;
    temp[2] = (bytes[3] & 0x000000FF) << 8;
    temp[3] = (bytes[2] & 0x000000FF);

    int bitsToConvert = (temp[3] | temp[2] | temp[1] | temp[0]);
    float ret = Float.intBitsToFloat(bitsToConvert);
    return ret;
}
```

Pakety se do zařízení posílají ve formě série bajtů v přesně daném pořadí. K jejich přípravě slouží metoda `byte[] prepareSetPacket(byte id, float value)`. V parametrech je této funkci předán kód nastavovaného parametru (ID) a hodnota, na kterou chceme požadovaný parametr nastavit. Návratovou hodnotou této metody je tedy pole bajtů, které je možno poslat do zařízení a nastavit pomocí něho např. počáteční frekvenci apod.

Metoda pro přípravu nastavovacího paketu

```
private byte[] prepareSetPacket(byte id, float value)
{
    int convertedToInt = Float.floatToIntBits(value);

    byte[] bdata = new byte[7];

    bdata[0] = 0x21;
    bdata[1] = id;
    bdata[2] = 0x00;
    bdata[3] = (byte) (convertedToInt);
    bdata[4] = (byte) (convertedToInt >> 8);
    bdata[5] = (byte) (convertedToInt >> 16);
    bdata[6] = (byte) (convertedToInt >> 24);

    return bdata;
}
```

Třída *ConnectionSpectran*, která zajišťuje spojení a komunikaci s měřícím zařízením je pak velice podobná třídě *ConnectionPowerAnalyzer*, která je popsána v kapitole 3.2. Zásadním rozdílem je, že v tomto případě nezáleží na nastavení rychlosti, počtu start-bitů a stop-bitů apod., tyto hodnoty tudíž není třeba vůbec nastavovat. Nastavování rychlosti zajišťuje samotné zařízení, nezářídka ale docházelo k nestabilitě spojení, zejména po odeslání dat do zařízení, pokud ihned po něm následoval příjem dalších dat. Toto je vyřešeno krátkou pauzou po každém odeslání dat. Po této úpravě probíhá již komunikace bez problémů.

Závěr

Hlavním cílem práce bylo vytvoření jednoduše použitelné aplikace, která umožní měření všech podstatných parametrů, které musí zařízení z hlediska elektromagnetické kompatibility splňovat. Původním záměrem bylo vytvoření aplikace jen pro měření vyšších harmonických signálů, protože na měření dalších parametrů jsem neměl potřebné zařízení. Toto zařízení bylo ale během vývoje zakoupeno, a tak bylo možno aplikaci rozšířit i o měření rušivých signálů. Původně měla aplikace umožňovat i měření na velmi vysokých frekvencích za použití vysokofrekvenčního spektrálního analyzátoru. Bohužel byl však dostupný jen analyzátor v nevhodné konfiguraci pro toto měření a bylo by třeba výrazně upravit jeho hardware.

Celý dosavadní vývoj aplikace zabral přibližně čtvrt roku, přičemž jsem aplikaci upravoval dle připomínek uživatelů, aby pro ně byla co nejsnadněji použitelná a celý proces měření elektromagnetické kompatibility byl co nejsnadnější. Největším zdržením způsobovaly chyby v programovací příručce pro spektrální analyzátor. Dlouhou dobu také zabrala příprava skriptů pro tvorbu PDF protokolů. Každý prvek bylo třeba umístit na správné místo a zároveň zajistit, aby neovlivňoval umístění jiných prvků. Pokud bych vytvářel další podobný dokument, určitě bych zvážil naprogramování aplikace, která by přípravu takového dokumentu umožňovala pomocí grafického rozhraní. Naopak překvapivě snadná byla práce s výkonovým analyzátozem, který je použit pro měření vyšších harmonických. Veškeré příkazy jsou posílány jako text, což jednak usnadňuje jejich pochopení a hlavně není třeba žádných složitých převodů na série bajtů, jako v případě komunikace se spektrálním analyzátozem. Další překvapivě snadnou věcí byla práce s knihovnou JFreeChart, která slouží pro vykreslování grafů. K této knihovně je dostupná opravdu perfektně zpracovaná dokumentace, navíc má i silnou uživatelskou základnu, proto je mnoho věcí probíráno v internetových diskuzích.

Do budoucna je počítáno s rozšířením aplikace o měření rušivých signálů na vysokých frekvencích, případně o další potřebná měření, která budou usnadňovat vývoj elektronických zařízení.


Samotná bakalářská práce byla, i přes prvotní obavy ze složitosti, napsána v sázecím systému L^AT_EX a tento systém budu rozhodně používat i pro další odborné práce.

Literatura

- [1] PITNER, Tomáš. *Java - začínáme programovat*. Vyd. 2. Praha: Grada, 2004, 222 s. ISBN 80-247-0295-9.
- [2] PAVLÍČKOVÁ, Jarmila a Luboš PAVLÍČEK. *Úvod do Javy*. Vyd. 1. Praha: Oeconomica, 2005, 227 s. ISBN 80-245-0963-6.
- [3] JELÍNEK, Lukáš. Java (24) - úvod do grafiky a GUI. Linuxsoft.cz [online]. 2006 [cit. 2014-08-13]. ISSN 1801-3805. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=1184
- [4] Java Communications API. *Oracle: Hardware and Software, Engineered to Work Together* [online]. 2012 [cit. 2014-08-13]. Dostupné z: <http://www.oracle.com/technetwork/java/index-jsp-141752.html>
- [5] RXTX for Java. *Mfizz: Software engineering expertise* [online]. 2008 [cit. 2014-08-13]. Dostupné z: <http://mfizz.com/oss/rxtx-for-java>
- [6] JFreeChart. *Jfree.org* [online]. 2014 [cit. 2014-08-13]. Dostupné z: <http://www.jfree.org/jfreechart/>
- [7] IText®, a JAVA PDF library. *SourceForge.net: Free, secure and fast downloads from the largest Open Source applications and software directory* [online]. 2014 [cit. 2014-08-13]. Dostupné z: <http://sourceforge.net/projects/itext/>
- [8] DŘÍNOVSKÝ, Jiří. *Elektromagnetická kompatibilita: přednášky*. Vyd. 1. V Brně: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2010, 229 s. ISBN 978-80-214-4202-3.
- [9] VACULÍKOVÁ, Polina a Emil VACULÍK. *Elektromagnetická kompatibilita elektrotechnických systémů: Praktický průvodce techniky omezení elektromagnetického vf rušení*. 1. vyd. Praha: Grada, 1998, 487 s. ISBN 80-716-9568-8.

- [10] ČSN EN 61000. *Elektromagnetická kompatibilita (EMC): Část 3-2: Meze pro emisi proudu harmonických (zařízení se vstupním fázovým proudem $\leq 16A$).* 3/2010. Praha: Úřad pro technickou normalizaci, meteorologii a státní zkušebnictví, 2010.
- [11] ČSN EN 55015 ed. 3:2007 + A1:2008 + A2:2009. *Meze a metody měření charakteristik vysokofrekvenčního rušení způsobeného elektrickými svídky a podobným zařízením.* Praha: Český normalizační institut, 2007.

Příloha A – Ukázkový výstupní protokol 1

| | | |
|---|--|-----------------------|
|  | Grenzwerte für Oberschwingungsströme EN 61000-3-2 unter 25W | Datum: 07.07.2014 |
| | | Ort/Abt.: Pardubice |
| | | Verfasser: Lauterbach |

Testobjekt: iHF 2


Eingangswirkleistung $\leq 25W$:

Beleuchtungseinrichtungen mit einer Eingangswirkleistung $\leq 25W$ müssen eine der beiden folgenden Anforderungen erfüllen
(Tabelle 3, Spalte 2 Grenzwerte für Geräte der Klasse D EN 61000-3-2)

Leistung in W = 13.12

| Oberschwingungsordnung n | Zulässiger Höchstwert je Watt | Gemessener Wert in mA [absolut] | Berechneter Höchstwert in mA | Zulässiger Höchstwert in mA | Norm erfüllt | |
|-------------------------------|-------------------------------|---------------------------------|------------------------------|-----------------------------|--------------|------|
| | | | | | Ja | Nein |
| 3 | 3.4 | 3.384 | 44.608 | 2300.0 | X | |
| 5 | 1.9 | 3.643 | 24.928 | 1140.0 | X | |
| 7 | 1.0 | 3.111 | 13.12 | 770.0 | X | |
| 9 | 0.5 | 4.807 | 6.56 | 400.0 | X | |
| 11 | 0.35 | 2.678 | 4.592 | 330.0 | X | |
| 13 | 0.296 | 4.159 | 3.8835 | 210.0 | | X |
| 15 | 0.257 | 1.68 | 3.3718 | 150.0 | X | |
| 17 | 0.226 | 0.623 | 2.9651 | 132.353 | X | |
| 19 | 0.203 | 1.602 | 2.6634 | 118.421 | X | |
| 21 | 0.183 | 0.652 | 2.401 | 107.143 | X | |
| 23 | 0.167 | 2.34 | 2.191 | 97.826 | | X |
| 25 | 0.154 | 0.592 | 2.0205 | 90.0 | X | |
| 27 | 0.143 | 0.336 | 1.8762 | 83.333 | X | |
| 29 | 0.133 | 0.786 | 1.745 | 77.586 | X | |
| 31 | 0.124 | 0.681 | 1.6269 | 72.581 | X | |
| 33 | 0.117 | 1.049 | 1.535 | 68.182 | X | |
| 35 | 0.11 | 0.16 | 1.4432 | 64.286 | X | |
| 37 | 0.104 | 0.973 | 1.3645 | 60.811 | X | |
| 39 | 0.099 | 0.381 | 1.2989 | 57.692 | X | |

Příloha B – Ukázkový výstupní protokol 2

| | | |
|---|------------------|-----------------------|
|  | Messprotokol EMV | Datum: 25.08.2014 |
| | | Ort/Abt.: Pardubice |
| | | Verfasser: Lauterbach |

Testobjekt: iHF 2

Prüfdaten

Prüfart:

Spektrummessungen

Prüfmittel:

Spectrum Analyzer HP8563E

Line Impedance Stabilisation Network HM6050-2 (das Band 10 kHz – 30 MHz)

CDN-M2/M3N (das Band 30 MHz – 300 MHz)

Notebook mit der Software Multimon

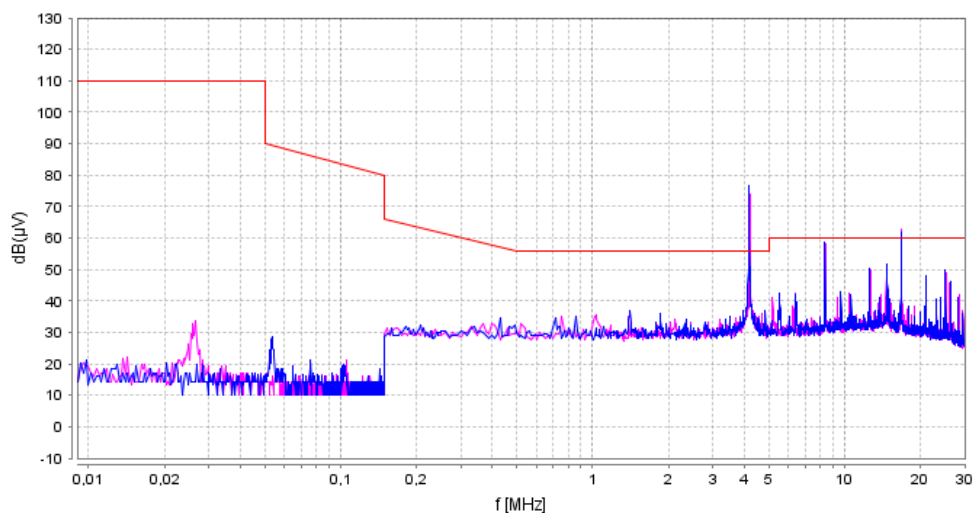
Prüfparameter:


Spektrum gemessene mit Spitzendetektor

Ergebnis

Leistung 0 % - N+L

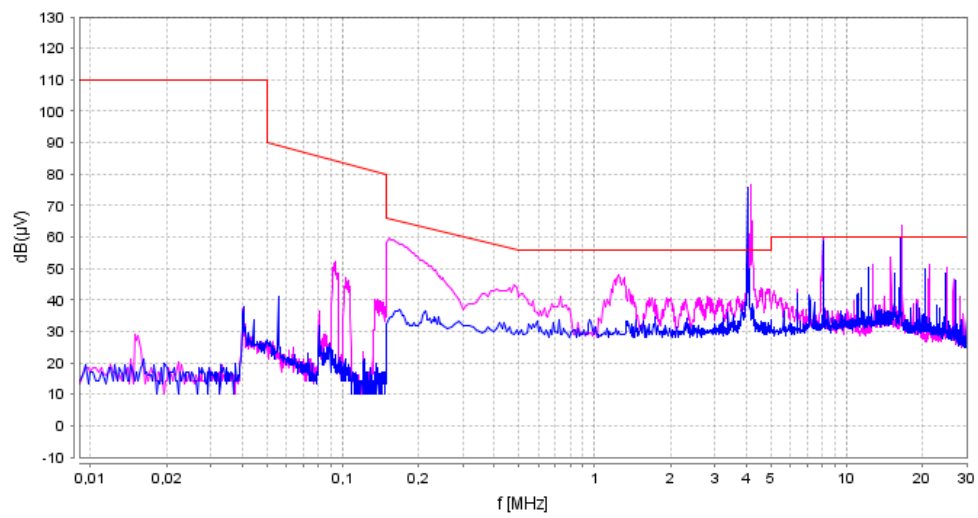
9 kHz - 30 MHz



| | | |
|---|------------------|-----------------------|
|  | Messprotokol EMV | Datum: 25.08.2014 |
| | | Ort/Abt.: Pardubice |
| | | Verfasser: Lauterbach |

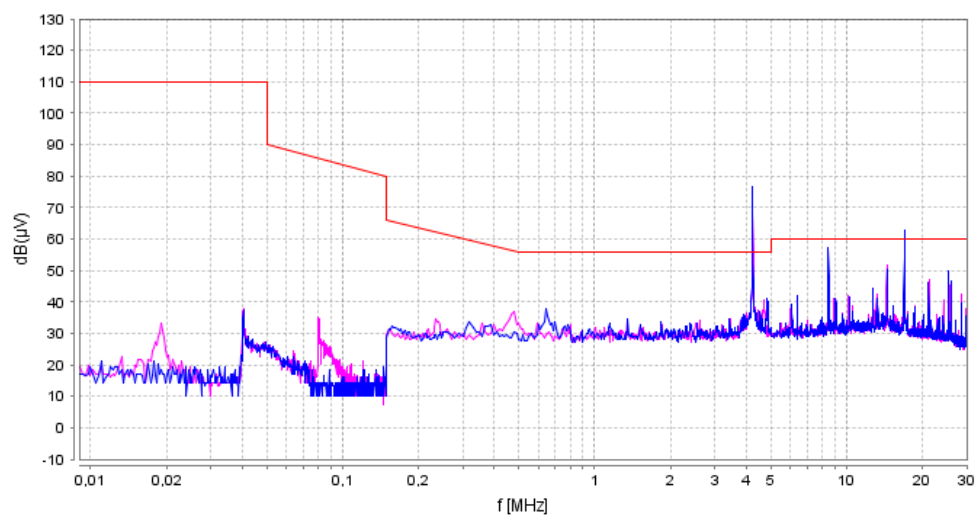
Leistung 10 % - N+L

9 kHz - 30 MHz



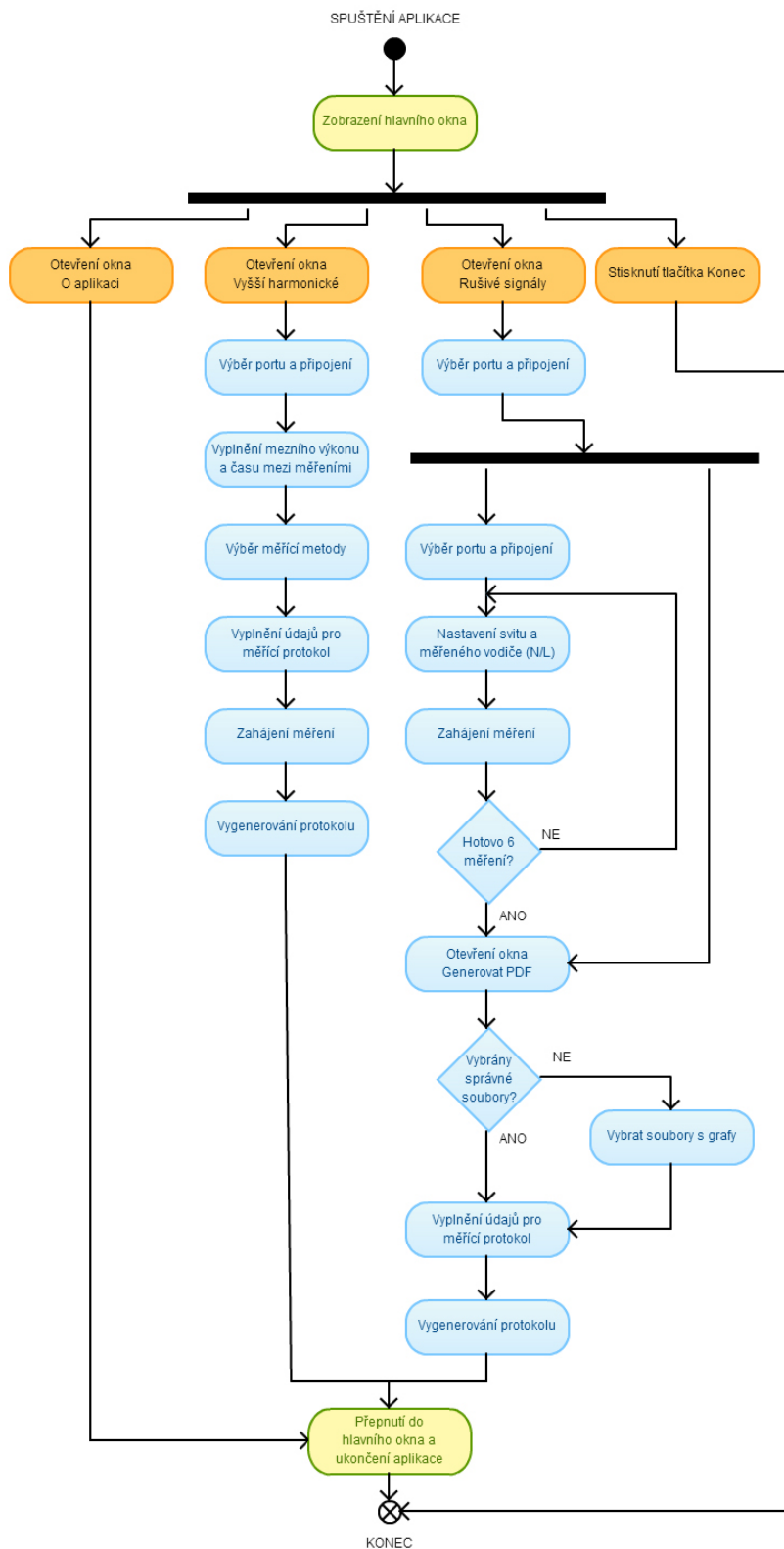
Leistung 100 % - N+L

9 kHz - 30 MHz



2/2

Příloha C – Diagram aktivit



Příloha D – Ukázka z kódu Spectran.java

```
private byte[] verifyPacket()
{
    byte[] bdata = new byte[5];

    bdata[0] = 0x01;
    bdata[1] = (byte) 0xA5;
    bdata[2] = 0x5A;
    bdata[3] = (byte) 0xF1;
    bdata[4] = 0x1F;

    return bdata;
}

private void beep(float time) throws IOException
{
    byte[] bdata = prepareSetPacket(STDONE, time);
    connection.sendData(bdata);
}

private byte[] prepareGetPacket(byte id)
{
    byte[] bdata = new byte[3];

    bdata[0] = 0x20;
    bdata[1] = id;
    bdata[2] = 0x00;

    return bdata;
}
```

Obsah CD

Obsah přiloženého CD je následující:

- soubor `LauterbachM_SoftwareMereni_RC_2014.pdf` – elektronická verze práce,
- adresář `tex` – zdrojové soubory pro sazbu bakalářské práce v \LaTeX u,
- adresář `zdojove_kody` – zdrojové soubory aplikace.