



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ ANALYZÁTORU JAKO WEBOVÁ APLIKACE

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie

Autor práce: **Ondřej Kubíček**
Vedoucí práce: Ing. Jan Kraus, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

GRAPHICAL USER INTERFACE FOR POWER ANALYZER AS A WEB APPLICATION

Bachelor thesis

Study programme: B2646 – Information Technology
Study branch: 1802R007 – Information Technology

Author: **Ondřej Kubíček**
Supervisor: Ing. Jan Kraus, Ph.D.



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ondřej Kubíček**
Osobní číslo: **M12000150**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Grafické uživatelské rozhraní analyzátoru jako webová aplikace**
Zadávající katedra: **Ústav mechatroniky a technické informatiky**

Zásady pro vypracování:

1. Seznamte se s požadavky na uživatelsky přívětivé grafické rozhraní analyzátoru kvality elektrické energie.
2. Vyberte vhodnou platformu pro návrh webových aplikací a navrhnete pro ni vlastní rozhraní pro analyzátor ARTIQ nebo SMC.
3. Otestujte spolehlivost a rychlost navrženého rozhraní.
4. Vyhodnoťte dosažené výsledky a diskutujte možnosti dalšího rozvoje aplikace.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **30–40 stran**

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

Seznam odborné literatury:

- [1] **Fulton, Steve a Jeff Fulton. HTML5 Canvas.: O'Reilly Media, 2011. ISBN 978-1449334987.**
- [2] **Pehlivanian, Ara a Don Nguyen. Jump start JavaScript.: SitePoint, 2013. ISBN 978-0987332189.**

Vedoucí bakalářské práce:

Ing. Jan Kraus, Ph.D.

Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: **10. října 2014**

Termín odevzdání bakalářské práce: **15. května 2015**



prof. Ing. Václav Kopecký, CSc.
děkan



doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2014

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.


Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 14.5 2015

Podpis: 

Poděkování

Tímto bych rád poděkoval Ing. Janu Krausovi, Ph.D. za vedení mé práce a za jeho věcné rady a připomínky. Dále bych také rád poděkoval své rodině za podporu při studiu.

Abstrakt

Tato bakalářská práce se zabývá návrhem webového rozhraní pro analyzátor kvality elektrické energie. Popisuje práci s HTML5 elementem canvas a komunikaci pomocí protokolu ModBus, který přístroj implementuje. Výsledkem práce je jednoduchá webová aplikace poskytující přehled o aktuálních datech získaných od přístroje. Aplikace zobrazuje harmonické napětí ve formě sloupcového grafu, ostatní data jsou zobrazena do tabulky. Graf lze zobrazovat více způsoby v závislosti na zadaných parametrech, které jsou v práci popsány. Mezi hlavní prvky grafu patří zobrazení krabicového grafu k jednotlivým sloupcům, které uživateli blíže naznačí informace o kvalitě napětí, či zobrazení limitů pro jednotlivé harmonické složky.

Klíčová slova: HTML5 canvas, JavaScript, kvalita elektrické energie, ModBus protokol, webová aplikace

Abstract

This thesis describes the design of a web interface for power quality analyzer. It describes communication with analyzer using ModBus protocol and also describes work with HTML5 canvas element. The result of this work is simple web application that provides overview of actual data received from the device. Application is able to show harmonic voltage as a bar graph and other types of data are displayed in the table. The graph can be displayed in several ways according to given parameters which are described in this thesis. The main elements of the chart include displaying a box plot for individual columns which indicates more informations for user about the power quality or display the limits for each harmonic component.

Keywords: HTML5 canvas, JavaScript, ModBus protocol, power quality, web application

Obsah

Úvod.....	12
1 Přehled současných implementací	13
1.1 Aplikace pro SMC 144.....	13
1.2 Vybrané desktopové implementace	13
1.2.1 Investigator od firmy Elspec	14
1.2.2 Program APPs od firmy Janitza	14
1.2.3 BK-TOUCH od Elcom	14
1.2.4 PQView	14
1.3 Vybrané mobilní aplikace	15
1.3.1 PSM-i pro analyzátory firmy PowerSight	15
1.3.2 HTAnalysis.....	15
1.3.3 METRIX ScopeNet	16
1.4 Souhrn	16
2 Analyzátor SMC 144	17
2.1 Vlastnosti.....	17
2.2 Protokol ModBus	17
3 Webové aplikace.....	19
3.1 JavaScript a AJAX	19
3.2 HTML5 element canvas	19
4 Cíl práce.....	21
4.1 Funkční požadavky	21
5 Návrh řešení.....	22
5.1 Struktura aplikace.....	22
5.2 Serverová část	23
5.2.1 Knihovna PHPModbus.....	24
5.3 Klientská část	24
5.3.1 Způsob zobrazování harmonických napětí	25
5.3.2 Zobrazení krabicového grafu.....	26
6 Implementace.....	27
6.1 Komunikace s analyzátozem	27
6.1.1 PHP funkce s knihovnou PHPModbus.....	28

6.1.2	Úprava a předání dat.....	29
6.2	Třída realizující graf.....	29
6.2.1	Vykreslení os a vodících čar.....	30
6.2.2	Předání dat do grafu (AJAX volání).....	31
6.2.3	Vykreslení aktuálních hodnot.....	32
6.2.4	Zakreslení limitů.....	33
6.2.5	Realizace krabicového grafu	34
6.2.6	Zobrazení dat do tabulky	35
6.2.7	Dostupné parametry nastavení	36
6.2.8	Výsledné zobrazení v závislosti na nastavení	37
6.3	Vyhodnocení rychlosti a spolehlivosti	40
7	Závěr.....	42
7.1	Další rozvoj aplikace	43
	Seznam použité literatury	44
A	Obrázkové přílohy	46
A.1	Webové rozhraní analyzátoru SMC 144	46
A.2	Program Investigator	46
A.3	GUI aplikace APPs.....	47
A.4	BK-TOUCH od Elcom.....	47
A.5	PQView	48
A.6	Program PSM-i.....	48
A.7	Mobilní aplikace HTAnalysis	49
A.8	METRIX ScopeNet	49
B	Obsah přiloženého CD.....	50

Seznam obrázků

Obrázek 1: Komunikační schéma aplikace.....	23
Obrázek 2: Hledání mediánů v množině dat.....	26
Obrázek 3: Graf harmonických napětí s krabicovým grafem.....	35
Obrázek 4: Výsledná tabulka s daty	36
Obrázek 5: Zobrazení lichých harmonických složek s krabicovým grafem.....	38
Obrázek 6: Zobrazení sudých harmonických pro dvě napětí s krabicovými grafy	38
Obrázek 7: Zobrazení lichých harmonických napětí s vyznačenými limitami.....	39
Obrázek 8: Zobrazení lichých složek v nepřiblíženém režimu bez krabicových grafů..	39
Obrázek 9: Tabulka lichých harmonických napětí analyzátoru.....	46
Obrázek 10: Realizace harmonických napětí v programu Investigator	46
Obrázek 11: Zobrazení harmonických napětí v aplikaci APPs	47
Obrázek 12: Harmonické složky v aplikaci BK-TOUCH	47
Obrázek 13: Graf harmonických napětí v programu PQView	48
Obrázek 14: Graf harmonických napětí a proudů v aplikaci PSM-i.....	48
Obrázek 15: GUI aplikace HTAnalysis	49
Obrázek 16: Ukázka aplikace METRIX ScopeNet	49

Seznam tabulek

Tabulka 1: Mapa registrů.....	18
Tabulka 2: Podpora elementu canvas v prohlížečích.....	20
Tabulka 3: Adresy uložení dat v paměti analyzátoru	24
Tabulka 4: Seznam parametrů pro práci s třídou grafu	36
Tabulka 5: Výsledky testování s pomalejším připojením.....	40

Seznam zdrojových kódů

Zdrojový kód 1: Nastavení konstant pro komunikaci.....	27
Zdrojový kód 2: Realizace čtení dat	28
Zdrojový kód 3: Výpis json formátu pomocí PHP funkce	29
Zdrojový kód 4: Vykreslení vodících čar	31
Zdrojový kód 5: Realizace AJAX volání.....	32
Zdrojový kód 6: For cyklus pro vykreslení limit.....	33

Seznam zkratek

- AJAX** Asynchronous JavaScript and XML, asynchronní vykonávání JavaScriptu
- API** Application Programming Interface, rozhraní pro programování aplikací
- CSS** Cascading Style Sheets, kaskádové styly
- DOM** Document Object Model, objektový model dokumentu
- GUI** Graphical User Interface, grafické uživatelské rozhraní (tlačítka, grafy, ...)
- HTML** HyperText Markup Language, značkovací jazyk pro tvorbu webových stránek
- LSB** Least Significant Bit, nejméně významný bit
- MSB** Most Significant Bit, nejvíce významný bit
- PLC** Programmable Logic Controller, průmyslový počítač používaný k automatizaci procesů v reálném čase
- PHP** Hypertext Preprocessor, programovací jazyk pro webové prostředí
- RMS** Root Mean Square, statistická hodnota měřící velikost měnící se veličiny

Úvod

Úkolem této práce je vytvoření jednoduché webové aplikace pro analyzátor měření kvality elektrické energie od firmy KMB systems, s.r.o., který nese název SMC 144. V textu nalezneme způsob, jakým lze s analyzátozem komunikovat pomocí protokolu ModBus a získávat tak aktuální data. Hlavní náplní práce je zpracování a zobrazení harmonického napětí [1] do podoby přehledného grafu. Graf je realizován pomocí HTML5 elementu canvas, pro který byly napsány metody zajišťující vykreslení jednotlivých prvků grafu (sloupce, osy, krabicové grafy, ...). V práci jsou tyto metody detailněji popsány a ukázány výsledné podoby grafu v závislosti na programátorem zadaných parametrech.

V první kapitole se seznámíme s již stávajícími aplikacemi, které se zabývají zpracováním a vyhodnocením kvality elektrické energie. Každá ze zvolených implementací je doplněna o screenshot toho, jak je zde realizováno zobrazení a zpracování harmonických napětí. Druhá kapitola je věnována analyzátoru SMC 144. Popíšeme si také protokol ModBus, který přístroj. Ve třetí kapitole se zaměříme na webové aplikace a s nimi související technologie, které byly v rámci implementace využity.

Kapitola Návrh řešení se zabývá popisem struktury aplikace a věcmi potřebnými pro realizaci jednotlivých částí aplikace. Nalezneme zde také blokové schéma aplikace, které znázorňuje způsob, jakým probíhá komunikace mezi jednotlivými stěžejními bloky. V kapitole Implementace je již detailně rozepsána realizace jednotlivých metod. Také zde najdeme finální vzhled několika variant vytvořeného grafu harmonických napětí. A nakonec také vyhodnocení rychlosti navrženého řešení v závislosti na rychlosti připojení uživatele.

1 Přehled současných implementací

V této kapitole se podíváme na již stávající způsoby implementace aplikací pro zobrazování harmonických napětí. Ke každé implementaci je přiložen screenshot ukazující způsob realizace. Nejdříve si ukážeme defaultní zobrazení pro analyzátor SMC 144 přístupné přes webový prohlížeč. Zbývající zmiňované implementace s SMC 144 nesouvisí. Byly vybrány z velkého množství aplikací, abychom si nastínili problematiku a ukázali si, jak různé implementace řeší zobrazování harmonických složek. Pro všechna řešení platí, že zobrazují aktuální data v reálném čase.

1.1 Aplikace pro SMC 144

Pro tento přístroj je k dispozici webové rozhraní, které zobrazuje aktuální naměřená data. Rozhraní je dostupné po nainstalování nejnovějšího firmwaru a je dostupné v prohlížeči pod IP adresou daného přístroje. Formou zvolenou pro zobrazení jsou převážně tabulky. Pro uživatele je poskytnut jeden graf znázorňující oscilogram napětí a proudů, který je vykreslen v canvasu. Pro zobrazení harmonických napětí je zde zvolena tabulka a vypsané hodnoty jsou pouze liché složky od 1. do 15. hodnoty (viz příloha A.1).

1.2 Vybrané desktopové implementace

Nyní bychom se podívali na programy, které nesouvisí s analyzátozem SMC 144. Jedná se o desktopové aplikace od různých firem a každá z nich je realizována jiným způsobem. Jsou zde zástupci, kteří mají zpracování přehledné, a naopak jsou tací, kteří mají řešení méně přehledné, s téměř nulovou vypovídající hodnotou. Jelikož se jedná o desktopová řešení, jsou všechny ukázky poměrně dost komplexní a obsahují velké množství funkcí a grafů.

1.2.1 Investigator od firmy Elspec

Program Investigator je komplexním řešením pro vyhodnocení kvality elektrické energie. Je založen na bázi rozličných grafů, ve kterých je zobrazeno například harmonické napětí nebo RMS. Každé ze zobrazených dat lze libovolně přibližovat pro ještě větší detaily. Graf harmonických napětí (viz příloha A.2) lze zobrazit v uživatelem daném rozmezí a výsledkem je vykreslení průměrných hodnot v daném období. [5]

1.2.2 Program APPs od firmy Janitza

Tato implementace je založena na webové bázi. Všechna data jsou dostupná skrze webovou stránku daného zařízení. Při pohledu na aplikaci je nám poskytnut přehled převážně ve formě grafů. Především zajímavé je provedení zobrazení harmonických napětí, které má realizované vykreslení limitů pro jednotlivé harmonické složky a také absolutní maximum, kterého daná složka nabývala (viz příloha A.3). [9]

1.2.3 BK-TOUCH od Elcom

Jedná se o rozhraní pro konfiguraci přístrojů, které také umožňuje zobrazovat například aktuální data. Tato aplikace je, na rozdíl od předešlých, vyvíjena tak, aby byla přívětivá pro dotykové ovládání. Graf harmonických složek je zde uživateli poskytnut v základní formě, neumožňující zobrazit liché a sudé složky zvlášť (viz příloha A.4). [3]

1.2.4 PQView

Aplikace PQView je bezpochyby nejkomplexnějším řešením. Poskytuje uživateli nespočet různých grafů s vyhodnocenými daty. Mezi typy grafů patří například histogram celkových harmonických zkreslení [16], fázorový diagram nebo harmonické složky, o které se zajímáme především. Navzdory komplexnosti celé aplikace, zobrazení harmonických složek je opět jen základní (viz příloha A.5). [4]

1.3 Vybrané mobilní aplikace

Kromě desktopových a webových variant se vytvářejí i aplikace mobilní. Z těchto řad jsem vybral jednoho zástupce mobilní aplikace navržené pro tablety, který mě jistým způsobem zaujal provedením grafu harmonických složek. Další dvě aplikace jsou především pro mobilní telefony. Hlavním rozdílem mobilních řešení oproti desktopovým je možnost interakce s grafy. Zatímco u desktopových aplikací byla možnost přibližovat data v grafu pouze u aplikace Investigator, u mobilních implementací nacházíme tuto možnost ve všech třech případech.

1.3.1 PSM-i pro analyzátor firmy PowerSight

PSM-i je aplikace navržena převážně pro tablety. Pro připojení k analyzátoru lze využít Bluetooth či Wifi technologií. V této aplikaci lze zobrazovat analýzu z naměřených dat nebo se připojit přímo na přístroj a sledovat aktuální hodnoty. Poskytuje nám dva typy zobrazení harmonických napětí, a to ve formě sloupcového grafu a ve formě fázorového diagramu. Zajímavé je provedení sloupcového grafu harmonických složek, jelikož poskytuje zobrazení harmonických napětí a proudů najednou (viz příloha A.6). První tři sloupce představují napětí a vztahují se k levé ose y, další tři pak představují proud a zobrazují se vůči ose y vpravo. [15]

1.3.2 HTAnalysis

Aplikace HTAnalysis je navržena jak pro mobilní telefony, tak i pro tablety. Provedení je zajímavé zejména proto, že lze pro libovolné napětí nebo proud zobrazit graf harmonických napětí. Všechny ze zde dostupných grafů lze libovolně přibližovat a zobrazit tak detailnější křivky. Dále je umožněno přidávat nebo odebírat veličiny, které se mají v grafu vykreslovat. Celé prostředí je velice přehledné a uživatelsky přívětivé. Graf harmonických napětí je zde v provedení sloupcového grafu s implementovaným posuvníkem, který ukazuje přesné hodnoty harmonických složek (viz příloha A.7). [7]

1.3.3 METRIX ScopeNet

Poslední aplikací je METRIX ScopeNet. Tato aplikace má jen pár funkcí, mezi které patří samozřejmě zobrazení harmonických napětí. Jak je patrné z ukázky (viz příloha A.8), lze zobrazit pouze jedno napětí až do 63. harmonické složky a není zde rozdělení na sudé a liché složky. Uživatel může ve zobrazení vybírat jednotlivé sloupce a zobrazit si tak detailní hodnoty pro danou harmonickou složku. [8]

1.4 Souhrn

Ze všech výše uvedených aplikací je na prvním místě bezpochyby program APPs od firmy Janitza, co do zpracování grafu harmonických složek. Data jsou přehledně zobrazena a obsahují již i statistiku vytvořenou z naměřených hodnot v podobě absolutního maxima. Dalším plusem této aplikace je zobrazení limitních hodnot pro každou složku harmonického napětí.

Podobným způsobem budeme v této práci zobrazovat harmonické složky v canvas elementu. Naše implementace bude obohacena o některé prvky, které se v porovnávaných aplikacích neobjevily. Hlavní částí grafu bude možnost vykreslit krabicový graf, což je jeden ze způsobů, jak rychle a srozumitelně zobrazit statistiku pro nějaký soubor seřazených dat. Krabicový graf dokáže uživateli poskytnout jasnou představu o tom, jaké bylo chování v dané době. Něco podobného v porovnávaných implementacích chybí a z mého pohledu je tato věc poměrně zásadní. Z tohoto důvodu jsem se rozhodl zaměřit se právě na tvorbu přehledného grafu harmonických složek s originálními prvky.

2 Analyzátor SMC 144

SMC 144 je přístroj pro analýzu kvality elektrické energie v síti vyvíjený firmou KMB systems, s.r.o. Je určen pro zaznamenávání spotřeby a vyhodnocení kvality elektrické energie. Instalace analyzátoru je jednoduchá, za použití DIN lišty. K ovládání přístroje není k dispozici žádný lokální ovládací prvek – nastavení je realizováno programově. [12]

2.1 Vlastnosti

Mezi veličiny, které jsou měřeny, patří například harmonická napětí a proudy, výkon, teplota, napětí nebo proud. Analyzátor je vybaven čtyřmi proudovými a napěťovými vstupy. Kontinuálně měří napěťové a proudové vstupy a zvládne vypočítat 63 složek harmonického napětí a proudu. Pro rozšíření funkcí lze nainstalovat firmwarové moduly, které umožní například archivovat události nebo měření mezi-harmonických napětí a proudů. Pro komunikaci je přístroj vybaven Ethernetovým portem a implementuje protokoly ModBus TCP a M-Bus využívané pro přenos dat a komunikaci. [10] Protokol ModBus si popíšeme níže, jelikož s jeho pomocí aplikace realizovaná v rámci této práce komunikuje a získává data.

2.2 Protokol ModBus

Protokol byl publikován v roce 1979 firmou Modicon. Jeho účelem je umožnit sériovou komunikaci elektronických zařízení po síti. Je zde využita architektura master-slave. Master bývá obvykle jeden a může mít až 127 slávů, kteří mají unikátní označení (číslo 1 až 127). Roli mastera v této komunikaci představuje PC nebo PLC a jeho úkolem je odesílat dotazy nebo zapisovat informace do slave. Na pozici slave bývají měřicí zařízení (PLC, čidla, ...), která požadavek od mastera zpracují a zpět odešlou požadovanou informaci. [17]

Organizace dat

Veškerá data jsou uložena v 16 bitových registrech. Celá paměť přístroje je členěna na úseky, které jsou adresovatelné pomocí hexadecimálního čísla. Pokud chceme ukládat 32 nebo 64 bitové hodnoty, jsou uloženy v paměti v jednotlivých 16 bitových registrech v pořadí od MSB po LSB (tzv. „velký endián“). Tomuto uložení poté odpovídá následující práce s daty. [11]

Tabulka 1 obsahuje rozčlenění úseků registrů v přístroji SMC 144. Nacházejí se zde dva typy registrů – input a holding. Input registr slouží pouze pro čtení dat. Holding registr je určen jak pro čtení, tak i pro zápis. Vzhledem k tématu této bakalářské práce je použit pouze jeden blok registrů – **Aktuální data**.

Tabulka 1: Mapa registrů [11]

Blok registrů	Adresa	Typ registru
Identifikační	0x200	Input Registr
Konfigurovatelné nastavení	0x700	Holding Registr
Nekonfigurovatelné nastavení	0x800	Input Registr
Aktuální data	0x1000	Input Registr
Elektroměr	0x2000	Input Registr
Kvalita energie	0x5000	Input Registr

3 Webové aplikace

V současné době trend využívání webových aplikací závratně stoupá. Hlavním důvodem je fakt, že aplikace jsou dostupné online a není nutné je instalovat. Moderní aplikace si dnes v převážné většině nevystačí pouze s HTML a CSS. Standardem se stalo hojně využívání skriptovacího jazyka JavaScript, který je spouštěn u uživatele v prohlížeči (client-side scripting). Dalším důležitým prvkem je využití jazyka PHP, který je spouštěn na serveru (server-side scripting).

Pro realizaci práce je využito jazyka PHP pro komunikaci s analyzátozem a získávání aktuálních dat. JavaScript obstarává pravidelné volání PHP skriptu a především pak práci s canvas elementem pro vykreslování grafu, jak je popsáno v kapitole 5.

3.1 JavaScript a AJAX

Skriptovací jazyk JavaScript byl navržen pro práci s jednotlivými HTML elementy, díky čemuž jistým způsobem zajistil dynamičnost webů (např.: proved' akci při kliknutí na element, vyvolej oznamovací okno, ...). Pojem AJAX je úzce spjat s JavaScriptem, jelikož je jednou z mnoha funkcí JavaScript frameworku zvaného jQuery. Funkce zajišťuje asynchronní vykonávání činností v rámci stránek, což znamená, že nemusíme znovu načítat celou stránku, ale je umožněno aktualizovat pouze vybranou část stránky, jako je například změna fotografie na našem profilu nebo volání PHP funkce pro získání potřebných dat. [13]

3.2 HTML5 element canvas

Canvas byl poprvé implementován ve standardu HTML5. Jak říká oficiální specifikace [18]: „Element canvas reprezentuje bitmapové plátno, které může být použito pro vytváření grafů, her nebo jiných grafických prvků.“ Avšak pro práci s canvasem potřebujeme již zmiňovaný jazyk JavaScript, který poskytuje API

obsahující sadu funkcí pro kreslení geometrických obrazců, vkládání obrázků, vytváření animací a další. Za pomoci těchto funkcí je programátor schopen vytvářet různé grafické prvky. [14]

Canvas je sám o sobě přístupný pomocí DOM (Document Object Model), což je struktura reprezentující všechny objekty umístěné na stránce, avšak vše, co programově „nakreslíme“, již jako DOM přístupné není – s každým namalovaným prvkem pak již nejde manipulovat. [6]

Element není podporován staršími prohlížeči, avšak celkové pokrytí podpory činí více než 90 %. [2] Tabulka 2 ukazuje přehled verzí nejpoužívanějších webových prohlížečů, které canvas podporují.

Tabulka 2: Podpora elementu canvas v prohlížečích [2]

Název prohlížeče	Plná podpora	Částečná podpora
Internet Explorer	v. 9 a vyšší	-
Mozilla Firefox	v. 2 a vyšší	-
Google Chrome	v. 4 a vyšší	-
Safari	v. 3.1 a vyšší	-
Opera	v. 10.1 a vyšší	-
iOS Safari	v. 3.2 a vyšší	-
Opera Mini	-	v. 8
Andriod browser	v. 3 a vyšší	v. 2.1 až 2.3
Chrome pro Andriod	v. 41	-

4 Cíl práce

Cílem této práce je návrh webového grafického rozhraní pro přístroj SMC 144 od firmy KMB systems, s.r.o., které by mělo poskytovat přehled o aktuálních datech snímaných analyzátořem v reálném čase. Vzhledem k analýze konkurenčních řešení jsme se rozhodli soustředit především na zobrazování a vyhodnocování harmonických napětí, jelikož se jedná o nejsložitější graf a na ostatní formy zobrazení, jako jsou fázorový diagram nebo heat mapa, není v této práci prostor.

Způsobů zobrazení harmonických napětí je velké množství a pokrýt je v této práci by bylo až příliš náročné. Z tohoto důvodu se soustředím na zobrazení dat do sloupcového grafu, který periodicky překresluje sloupce na aktuální data.

Požadavek na technologie dán nebyl. Zvolil jsem využití jazyka PHP pro komunikaci s analyzátořem. Graf vykresluji do HTML5 elementu canvas, se kterým se pracuje pomocí JavaScript API.

4.1 Funkční požadavky

Hlavním požadavkem je zobrazovat data v reálném čase. Data by měla být reprezentována ve formě sloupcového grafu, který by měl být přehledný a měl by umožňovat rozlišné formy zobrazení, jako je výběr jen některých napětí, zobrazení lichých či sudých harmonických složek či přizpůsobení barev jednotlivých sloupců. Dále by měl umožňovat vykreslení legendy, popisující jednotlivé prvky grafu. Pod grafem by měly být vypsány další důležité veličiny jako je celkové harmonické zkreslení nebo vypsána největší z aktuálních hodnot. Dalším důležitým požadavkem je možnost shromažďovat data od spuštění prohlížeče a následným zpracováním z nich vytvořit krabicové grafy pro jednotlivé sloupce grafu. Křabicový graf by měl být navržen tak, aby byl přehledný a lehký čitelný.

Posledním požadavkem je implementace čáry představující limit pro každou ze složek harmonického napětí. Hodnota limitů má být definovatelná programátorem, který graf bude využívat.

5 Návrh řešení

V návrhu je popsáno rozvržení aplikace do jednotlivých vrstev a je zde popsáno, jak každá z vrstev komunikuje a pracuje. Kapitola je rozdělena do dvou hlavních podkapitol – serverová část a klientská část aplikace. Toto rozložení je úzce svázáno s implementací protokolu ModBus, který, jak bylo uvedeno v teoretické části, využívá architektury master-slave.

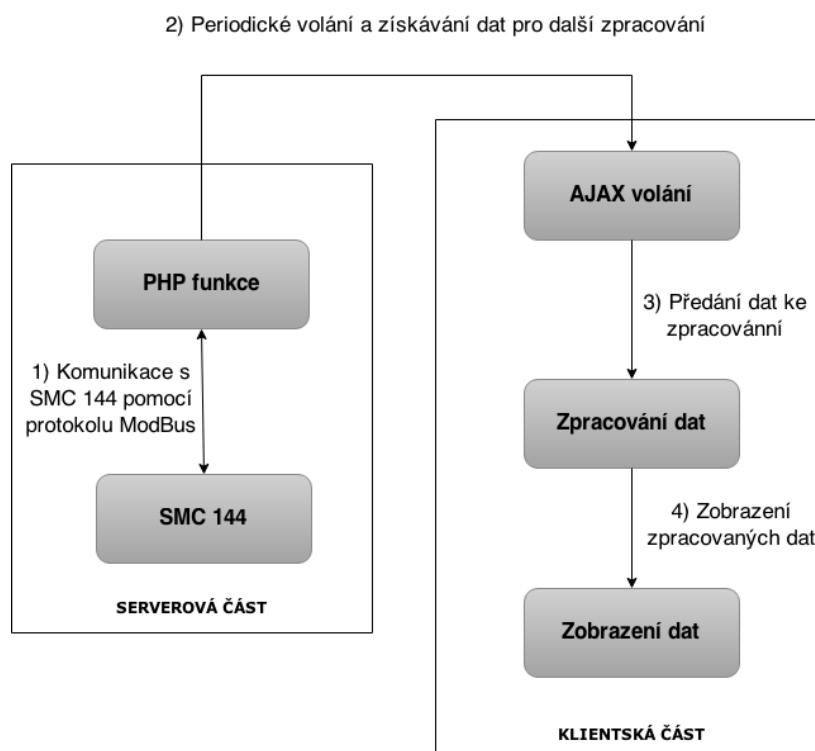
Součástí návrhu řešení je komunikační schéma celé aplikace. Ve schématu je ukázána komunikace mezi jednotlivými částmi celku. V rámci návrhu si také popíšeme detailněji serverovou část a to, jak získat data z analyzátoru a jak je dále předat ke zpracování. Klientská část bude obsahovat vlastní zpracování dat a jejich následné zobrazení.

5.1 Struktura aplikace

Aplikace je rozčleněna do serverové a klientské části. Obě tyto části jsou složeny z několika dalších bloků a navzájem spolu komunikují a předávají si požadovaná data. Pro realizaci serverové části bylo využito jazyka PHP. V klientské části se využívá jazyk JavaScript pro pravidelné načítání dat a především pro práci s elementem canvas, který slouží k vykreslení grafu harmonických napětí.

Ze schématu aplikace (viz Obrázek 1) je patrná spolupráce mezi jednotlivými bloky. Na jednom konci je analyzátor SMC 144, který nám poskytuje data, na druhém pak webový prohlížeč uživatele. Potřebná data z přístroje budeme získávat PHP skriptem, který bude zajišťovat komunikaci se zařízením. Pro zajištění periodické obnovy dat je nutné využít AJAX volání. To nám umožňuje asynchronně volat PHP funkci a získaná data zpracovávat. Je to v současné době nejefektivnější a nejrychlejší způsob takového zpracování. Druhým možným způsobem by bylo vytvoření skriptu, který by vše prováděl pomocí HTTP dotazů, ale tato varianta je neefektivní a vyžaduje mnohé znovu načítání stránky. Pokud by data od přístroje nedorazila do 5 vteřin od dotázání, aplikace zobrazí hlášku o neúspěšném načtení dat a je potřeba celou stránku znovu načíst.

Dále se data musí předat ke zpracování, aby se v dalším kroku mohla zobrazit. Zpracováním dat je myšleno získání potřebných proměnných pro inicializaci grafu, ukládání dat do předem dané struktury, či postupné zaznamenávání dat pro vykreslení krabicového grafu. Blok zobrazení dat má za úkol vykreslit předem připravená data do elementu canvas v podobě sloupcového grafu. Tento graf má několik možností zobrazení v závislosti na nastavení při vytváření objektu grafu.



Obrázek 1: Komunikační schéma aplikace

5.2 Serverová část

V rámci serverové části je nutné zajistit komunikaci se zařízením a následné získání aktuálních dat. Všechna data jsou uložena na specifických adresách, které jsou uvedené v dokumentaci pro ModBus protokol přístroje SMC 144. Konkrétně se zajímáme o data typu napětí, proud, celkové harmonické zkreslení napětí (THDU) a zejména o harmonické napětí. Jak bylo zmíněno v předešlých kapitolách, přístroj je schopen

zaznamenávat spoustu dalších veličin, ale v rámci tohoto řešení se soustředíme především na zpracování harmonických napětí. Zobrazení ostatních aktuálních dat je zde pouze vedlejší. Tabulka 3 zobrazuje přesné adresy, pod kterými jsou data přístupná.

Tabulka 3: Adresy uložení dat v paměti analyzátoru

Veličina	Adresa (HEX)	Velikost, datový typ
Napětí (U_1 až U_4)	0x1100...0x1106	32b, float
Proud (I_1 až I_4)	0x110E...0x1114	32b, float
Celkové harmonické zkreslení napětí (THDU ₁ až THDU ₄)	0x1200...0x1206	32b, float
Harmonické napětí ($U_{1H1...h50}$ až $U_{4H1...h50}$)	0x1400...0x158E	32b, float

5.2.1 Knihovna PHPModbus

Knihovna PHPModbus slouží pro komunikaci se zařízeními, která protokol ModBus implementují. Autorem je Jan Krakora, který poskytl knihovnu volně ke stažení například na www.github.com pod (L)GPL licencí. Vzhledem k tomu, že komunikace pomocí ModBus protokolu je poměrně složitá a její programování by zabralo velké množství času, zvolil jsem právě toto již hotové řešení. Společně s knihovnou je ke stažení také sada příkladů, které popisují práci s ní. Knihovna poskytuje sadu základních funkcí, které zajistí připojení k zařízení, získání dat a případně i následné zpracování dat (záleží na velikosti čtených registrů).

5.3 Klientská část

Základem klientské části je třída reprezentující sloupcový graf. Jelikož se jedná o specifické zobrazení, je nutné vše naprogramovat od samého začátku. Graf umožňuje

zobrazení jednotlivých harmonických složek napětí. Vzhledem k množství harmonických napětí je nutné zajistit vykreslení pouze dané množiny dat (sudé nebo liché složky). Samozřejmostí pro vykreslování grafů je také legenda popisující, co které prvky grafu představují.

Specifickým požadavkem na graf je možnost zobrazit limity a krabicové grafy pro každou harmonickou složku. Pro realizaci krabicového grafu se musí data určitým způsobem ukládat. Jelikož nemáme přístup k databázi, zvolil jsem ukládání dat do pole, které se začne plnit vždy od spuštění stránky, která graf obsahuje. Tento způsob je nejjednodušším možným řešením a neměly by nastat žádné potíže při jeho implementaci. Jediným omezením by mohlo být obrovské množství dat (dlouho spuštěný prohlížeč), díky čemuž by mohlo docházet ke zvyšujícím se prodlevám ve zpracování dat. V horším případě pak k vyčerpání dostupné operační paměti. Avšak jednalo by se o stovky megabajtů dat. K nastřádání takového množství dat by mohlo dojít například po ponechání zapnutého prohlížeče po dobu deseti dnů s nastavenou periodickou obnovou dat na jednu sekundu. Takto dlouhý provoz se však nepředpokládá.

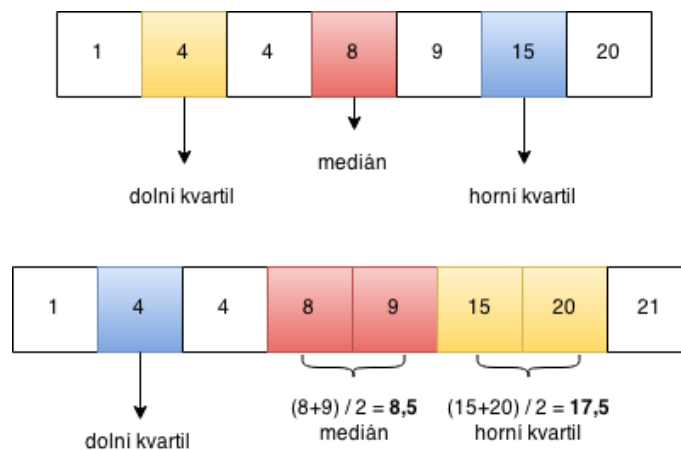
5.3.1 Způsob zobrazování harmonických napětí

Harmonické složky napětí jsou v přístroji ukládány jako procentuální. Výjimkou je pouze první harmonická složka, která je uváděna ve voltech. Z několika následujících způsobů zobrazení je implementován pouze druhý bod – „Relativně vůči nominální hodnotě (230 V). Ostatní možnosti budou doplněny v pozdějších úpravách implementace.

- 1) **Relativně vůči první harmonické** – Výchozí hodnotou pro přepočtení složek z procent na volty je měnící se první harmonická složka.
- 2) **Relativně vůči nominální hodnotě (230 V)** – Výchozí hodnota pro přepočtení složek z procent na volty je pevně dána (obvykle 230 V).
- 3) **Relativně vůči limitu harmonické** – Velikost harmonické složky se uvádí jako procento, o kolik přesáhla nebo nepřesáhla svůj limit.

5.3.2 Zobrazení krabicového grafu

Pro tento požadavek je potřeba zaznamenána data pokaždé seřadit, jelikož v nich musíme nalézt ve své podstatě tři mediány, minimum a maximum. Po nalezení prvního mediánu z celkového počtu dat je nutné nalézt zbývající dva mediány (horní a dolní kvartil) z částí, na které byla data rozdělena prvním mediánem. Výsledné hodnoty se pak používají pro zakreslení krabicového grafu. Obrázek 2 ukazuje způsob nalezení mediánů v seříděné sedmi-prvkové a osmi-prvkové množině čísel. Postup zpracování množiny s lichým počtem prvků se liší od postupu u množiny se sudým počtem prvků.



Obrázek 2: Hledání mediánů v množině dat

6 Implementace

V této kapitole se budeme věnovat samotné realizaci aplikace. Jednotlivé podkapitoly vycházejí z návrhu řešení. Popíšeme si, jak jsou jednotlivé stěžejní body aplikace realizovány, jak pomocí knihovny PHPModbus získáváme data a také práci s vytvořeným grafem harmonických napětí. V závěru kapitoly je ukázka výsledného grafu se zobrazenými krabicovými grafy.

6.1 Komunikace s analyzátozem

Jak bylo uvedeno v návrhu, pro komunikaci je potřeba vytvořit PHP funkci, která se dokáže připojit k analyzátoru a komunikovat s ním. Pro připojení a odesílání dotazů jsem využil knihovnu PHPModbus. Připojení přes protokol ModBus vyžaduje IP adresu přístroje. Dále je také nutné nastavit formu datových soketů. Je na výběr mezi TCP a UDP sokety. Pro realizaci jsem zvolil TCP, jelikož je zaručena spolehlivost doručení a doručení ve správném pořadí, na rozdíl od UDP, které nezaručuje, zda se nějaký datagram v přenosu neztratí. To by mohlo způsobit problémy, jelikož je potřeba, aby došla data byla doručena správně kvůli jejich pozdějšímu zpracování. Všechny údaje potřebné k připojení jsou uloženy v souboru *config.php* ve formě konstant, pro snadnější práci s nimi (viz Zdrojový kód 1).

```
define('MODBUS_ID', 1); //ID zařízení
define('MODBUS_IP', '147.230.75.19'); //přidělená IP adresa
define('MODBUS_PROTOCOL', 'TCP'); //nastavení soketů na TCP
```

Zdrojový kód 1: Nastavení konstant pro komunikaci

6.1.1 PHP funkce s knihovnou PHPModbus

Vzhledem k faktu, že harmonických složek je padesát a všechny tyto hodnoty se do výsledného grafu nevejdou, je zbytečné číst všech padesát hodnot. Tudíž funkce přebírá pomocí HTTP metody GET parametr *mode*, který určuje, zda chceme sudé (*even*) nebo liché (*odd*) harmonické složky. Pokud by tento parametr nebyl nastaven, je defaultně nastaveno získávání lichých složek.

Po nastavení potřebných konstant a GET parametru se lze začít dotazovat na data. Dle dokumentace k přístroji jsem si zjistil, na kterých adresách se nacházejí harmonická napětí, která chci z přístroje získat. Knihovna poskytuje devět základních funkcí pro zápis i čtení registrů. Pro čtení postačí pouze funkce *readMultipleRegisters()*, která umožňuje přečíst blok registrů a uložit je do proměnné. Jelikož se jedná o 32 bitové registry a harmonických složek je celkem 200 (čtyři vstupy napětí U1 až U4), je nutné přečíst 400 adres v paměti přístroje. Knihovna však neumožňuje získat tak velký souvislý blok dat, tudíž jsem zvolil volání funkce pro každé napětí zvlášť.

Ve výsledku tedy používám čtyři funkce *readMultipleRegisters()* a výsledné hodnoty ukládám do asociativního pole, kde klíčem je název napětí (U1 až U4) a hodnotou je pole, do kterého ukládám přečtené byty hodnot. Pokud by došlo k chybě při komunikaci, dojde k vyvolání výjimky, která je odchycena a zobrazí se chybová zpráva (viz Zdrojový kód 2).

```
try {
    //realizace čtení úseku paměti a uložení do asociativního pole
    $data["U1"] = $modbus->readMultipleRegisters(MODBUS_ID, 0x1400,
100);
    $data["U2"] = $modbus->readMultipleRegisters(MODBUS_ID, 0x1464,
100);
    $data["U3"] = $modbus->readMultipleRegisters(MODBUS_ID, 0x14C8,
100);
    $data["U4"] = $modbus->readMultipleRegisters(MODBUS_ID, 0x152C,
100);
} catch (Exception $e) { //odchycení případných chyb
    echo $e;
}
```

Zdrojový kód 2: Realizace čtení dat

6.1.2 Úprava a předání dat

Po úspěšném získání dat máme v poli uložené jednotlivé byty. Jelikož je pořadí bytů ve formátu *velký endian* a každá z hodnot má být je 32 bitové *float* číslo, musí se byty přeuspořádat do správného pořadí a následně spojit (viz kapitola 2.2).

Kvůli tomuto způsobu uložení jsem vytvořil funkci *edit_data()*, která přebírá jako parametr pole s daty a požadovanou velikost registru v bytech. V závislosti na zadané velikosti se vstupní pole dat rozdělí na několik malých polí (v našem případě čtyř prvková pole), ve kterých se postupně přeuspořádají jednotlivé páry bytů. Výsledkem pak je pole obsahující 25 čtyř prvkových polí (jednotlivé složky harmonických napětí).

Data se následně zpracují pomocí funkce *PHPType::bytes2float()* z knihovny *PHPModbus*, která slouží k převodu bytů na float hodnotu.

Posledním krokem je doplnění pole o popisky harmonických složek (h1, h3, h5, ...) s názvem klíče *labels* a následné vypsání asociativního pole funkcí *json_encode()*. Výpis pole je nutný pro propojení s klientskou částí, přesněji s AJAXovým voláním. Zdrojový kód 3 ukazuje výpis dat ve formátu *json*. Ukázka je zkrácena pouze na liché čtyři složky harmonických napětí.

```
"labels": ["h1", "h3", "h5", "h7"], "U1": [226.127, 3.61, 2.999, 2.931],  
"U2": [226.111, 3.579, 3.162, 2.827], "U3": [226.128, 3.596, 3.169, 2.  
.859], "U4": [164.994, 4.234, 0.925, 2.08]
```

Zdrojový kód 3: Výpis json formátu pomocí PHP funkce

6.2 Třída realizující graf

Hlavním prvkem práce je graf, který umožňuje získaná data zobrazit do přehledné formy. Pro snadné použití jsem pro graf vytvořil třídu, která přebírá několik parametrů (viz Tabulka 4) ovlivňující jeho výslednou podobu. Na základě nastavení se pak uvnitř třídy volají potřebné funkce, které zajišťují danou funkcionalitu. Tyto parametry jsou voleny programátorem při vytváření třídy grafu a záleží pouze na něm, v jaké formě

chce data zobrazit. Graf umožňuje, oproti zmiňovaným řešením, zobrazit statistiku přijatých dat v podobě krabicového grafu (viz Obrázek 3) a také limit pro jednotlivé složky. Dále je zde také možnost zobrazit pouze vybraná napětí namísto všech. A posledním důležitým parametrem je „přiblížení“ grafu pomocí parametru *zoom*, jelikož hodnoty, které se zobrazují, jsou malé (např.: 3 % z 230 V, 1,2 % z 230V, ...) a při velkém zobrazení by nebyly téměř vidět. Jednotlivým funkcím se věnuji v následujících podkapitolách.

6.2.1 Vykreslení os a vodících čar

Když je vytvořen objekt grafu, první věcí, která se v canvasu vykresluje, jsou osy a vodící čáry. V případě, že nechceme graf zobrazit v „přiblíženém“ módu, není potřeba mít data z přístroje, ale stačí nám pouze hodnoty dané při vytvoření objektu.

Co se týče práce s canvasem, pohyb po něm se odehrává v souřadnicích [x;y], tudíž je nutné zavést metody a konstanty pro výpočet pozic jednotlivých prvků. Osy a vodící čáry jsou nejjednodušší částí grafu pro vykreslení. Pro jejich vykreslování jsem využil čtyř metod, které jsou poskytnuty v rámci API pro práci s elementem canvas. Konkrétně se jedná o metody: *moveTo(x, y)* – přesun na prvotní pozici, ze které započne kreslení; *lineTo(x, y)* – stanovení cesty k novému bodu od pozice uvedené metodou *moveTo(x, y)*; *lineWidth = 2* – proměnná pro nastavení šířky čáry na 2 pixely; *stroke()* – vytažení trasy, která byla stanovena prvními dvěma metodami.

Vykreslení vodících čar je téměř totožné s tím rozdílem, že čar je více, a jejich pozice se mění. Tudíž je nutné pro každou vodící čáru vypočítat novou y souřadnici v cyklu *for*. Pro osy je dále nutné vytvořit textové popisky jednotlivých úseků. Na tyto účely slouží funkce *fillText(„text“, x, y)*. Opět jako v případě vodících čar se text kreslí ve smyčce *for* a souřadnice se postupně přepočítávají. Zdrojový kód 4 ukazuje způsob, kterým se v aplikaci vykreslují vodící čáry.

```

//proměnná context nám představuje referenci na element canvas
//hodnoty x a y jsou konstanty vypočtené při vytvoření objektu
//grafu
for (var n = 0; n < this.numGridLines; n++) {
    //přepočítávání y souř. a zaokrouhlení hodnoty pro hladké
    //vykreslení
    var y = Math.round((n*this.height/this.numGridLines)+this.y);
    context.beginPath();
    context.moveTo(this.x, y);
    context.lineTo(this.x + this.width, y);
    context.stroke();
}

```

Zdrojový kód 4: Vykreslení vodících čar

6.2.2 Předání dat do grafu (AJAX volání)

Samotné AJAX volání není součástí mnou vytvořené třídy, ale je nutné ji dopsat do kódu stránky, na které se vyskytuje element canvas. Tato funkce slouží k propojení PHP funkce, která poskytuje data, a třídy grafu, do které tato data potřebuji předat. Tento způsob je rychlý a jednoduchý. Abych mohl využít AJAXu, musel jsem si stáhnout knihovnu jQuery.

Volání se vytváří příkazem `$.ajax()` a stejně jako mnou vytvořený graf přebírá nastavující parametry. Mezi ty, které jsem zde využil, patří nastavení datového typu přijímaných dat na *json* formát. Dále také URL na PHP skript, ze kterého se data získávají, a proměnná, do které se mají data uložit. Nejdůležitějším parametrem je *success*, který stanovuje, co se se získanými daty bude dít. Jako hodnotu přejímá callback funkci, do které jsem zapsal kód pro předání dat do grafu (viz Zdrojový kód 5).

```

$.ajax ({
  dataType: "json",
  //adresa na PHP funkci vracející data
  url: „harmonics.php?mode=even“,
  //získaná data se uloží do proměnné „data“
  data: data,
  //funkce, která se vykoná po úspěšném získání dat
  success: function(data) {
    //předání dat do třídy grafu
    graph.setData(data);
    //předání hodnot pro vykreslení krabicového grafu
    graph.setHistory();
  }
});
//smazání obsahu canvasu kvůli vykreslení dalších hodnot
ctx.clearRect(0, 0, canv.width, canv.height);

```

Zdrojový kód 5: Realizace AJAX volání

6.2.3 Vykreslení aktuálních hodnot

Vykreslování sloupců grafu již obsahuje pokročilejší práci s canvasem a více počítání. Vždy je potřeba zobrazit danou množinu sloupců (záleží na vstupním nastavení) pro každou složku harmonického napětí. Sloupce se vykreslují z aktuálních dat uložených v poli uvnitř třídy. Vykreslování grafu obstarává metoda *drawBars()*. Využívám zde především funkce *translate(x, y)*, která zajistí nastavení nového počátku soustavy souřadnic, takže namísto počátku 0, 0 (vlevo nahoře), ze kterého se obvykle veškeré hodnoty vypočítávají, nám funkce umožní stanovit si počátek například na hodnoty 20, 50 a provádět veškeré následující výpočty v závislosti na těchto souřadnicích.

Samotné vykreslování sloupců probíhá ve dvou *for* cyklech. První zajišťuje pohyb přes všech 25 složek harmonického napětí. Druhý cyklus prochází jednotlivá napětí (U1 až U4) a tím získává data. Z přečtených hodnot se vypočítá výška sloupce dle následujícího vztahu:

$$\text{výškaSloupce} = \text{round}((\text{hodnota} / 100 * \text{maxHodnota}) * \text{velikostJednotky})$$

maxHodnota - hodnota zadaná při vytvoření třídy grafu (obvykle 230)

velikostJednotky - jednotka vypočítaná v závislosti na výšce grafu a maximální hodnotě osy y

Po vypočtení výšky se vykreslí sloupec pomocí metody *rect(x1, y1, x2, y2)*. Pokud je zvolené přibližné zobrazení a zobrazují liché složky harmonických, vepíše se k první harmonické složce jejich přibližná velikost ve voltech, jelikož osa y nabývá malých hodnot (řádově desítky voltů) a první harmonická má velikost 200 a více voltů.

6.2.4 Zakreslení limitů

Limity jsou jedním z odlišujících se prvků oproti ostatním již vzniklým řešením. Slouží především pro rozšíření představy o chování harmonických složek. Každá ze složek má svůj vlastní limit, který se určí při vytvoření objektu grafu. V grafu se limita zobrazuje jako červená lomená čára o šířce 2 pixely. Velikost limitů se udává v procentech a jejich pozice se vypočítává v závislosti na maximální hodnotě na ose y. Pokud programátor při vytváření grafu do nastavení limity nezadá, nic se v grafu nezobrazí, tudíž je tento parametr nepovinný.

Pro vykreslení opět využívám metod pro práci s canvasem, konkrétně metody *moveTo(x, y)*, *lineTo(x, y)* a *stroke()*. Stačí jen postupně ve *for* cyklu přepočítávat x a y souřadnice, kde se má čára lomit. Cyklus *for*, který zajišťuje vykreslení limitů, ukazuje Zdrojový kód 6.

```
for (var i = 0; i < this.limits.length; i++) {
    //vypočtení nové x souřadnice
    xPos = Math.round(this.width/this.limits.length*i-startFix);
    //posun ve směru osy x
    context.lineTo(xPos, yPos);
    //výpočet nové y souřadnice
    yPos = Math.round(-this.limits[i] * value);
    //přepočet proměnné, která zajišťuje přesnější vykreslení
    startFix = this.limits.length / 10;
    //posun ve směru osy y
    context.lineTo(xPos, yPos);
}
```

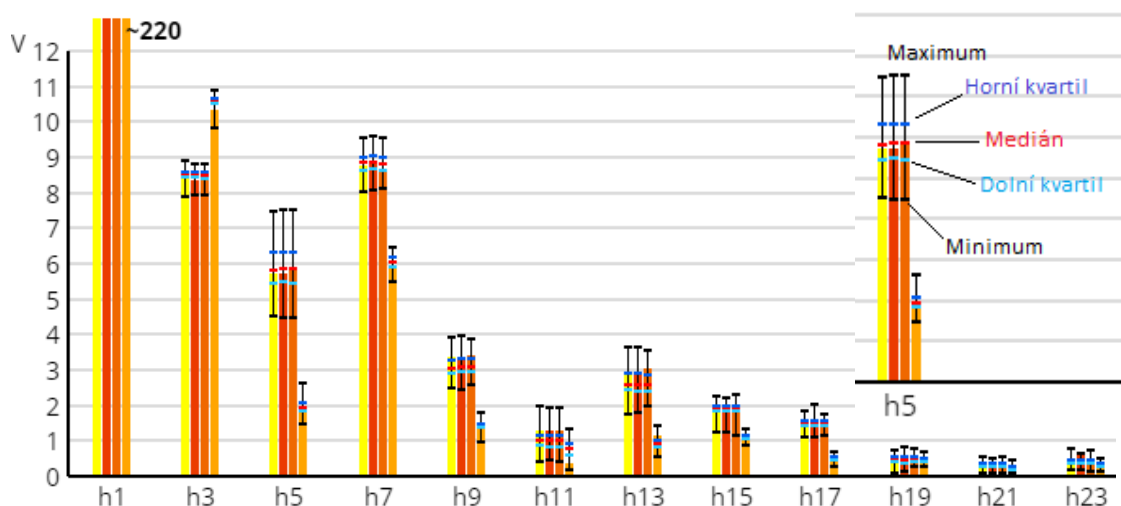
Zdrojový kód 6: For cyklus pro vykreslení limit

6.2.5 Realizace krabicového grafu

Krabicový graf je hlavním prvkem mého řešení. Byl vytvořen z důvodu upřesnění chování harmonických složek. Bylo nutné vymyslet způsob, jak do sloupcového grafu přehledně krabicový graf zobrazit pro každé napětí. Zvolil jsem způsob svislé čáry protnuté pěti vodorovnými čarami stejně dlouhými jako je šířka sloupce. Čáry jsou barevně rozlišeny, aby se zvýšila přehlednost při čtení grafů.

Pro realizaci je nutné postupně ukládat načtená data do nějaké datové struktury. Existuje více možností jak ukládání zrealizovat. Zabýval jsem se například možností ukládat hodnoty do session, ale práce s nimi byla zbytečně složitá v porovnání s pouhým uložením do pole. Dalším možným uložením by mohla být MySQL databáze nebo textový soubor. Opět jsem usoudil, že by to byl zbytečně složitý a neefektivní způsob. Nakonec, jak bylo uvedeno v návrhu, jsem zvolil využití lokálního JavaScriptového pole.

Za účelem ukládání dat používaných pro kreslení krabicového grafu jsem vytvořil v třídě grafu funkci *setHistory()*, která již uložené hodnoty v grafu ukládá do dalšího pole. Toto pole je aditivní a po každém uložení nových hodnot se všechny hodnoty setřídí od největší po nejmenší hodnotu. Setřídění je povinné kvůli následnému nalezení minima, maxima, mediánu, horního a dolního kvartilu. Výsledné pole pak například po 30 průchodech obsahuje 30 vzorků pro každou harmonickou složku, tedy celkem 750 hodnot. V ukázce je vykreslena pouze část grafu, aby byl dobře čitelný (viz Obrázek 3). Výsledný graf v prohlížeči pak obsahuje všech 25 harmonických složek.



Obrázek 3: Graf harmonických napětí s krabicovým grafem

Graf vykresluje liché harmonické složky v přibližném módu. Sloupce představují aktuální načtenou hodnotu pro napětí U1 až U4. Jak si můžete všimnout, první harmonická složka zasahuje mimo graf a krabicový graf neobsahuje. Je to z důvodu velikosti její hodnoty, která je přibližně 220 V. U vrcholů sloupců jsou zmiňované krabicové grafy. Detail krabicového grafu jedné z harmonických složek lze najít v pravém horním rohu grafu výše. Na detailním pohledu je popsáno, co které prvky krabicového grafu znamenají.

6.2.6 Zobrazení dat do tabulky

Aplikace také poskytuje zobrazení dat do tabulek. Data, která jsem zde zobrazil, však nejsou harmonické složky napětí, nýbrž ostatní veličiny jako napětí, proud či celkové harmonické zkreslení. Od přístroje dotazuji pouze čtyři veličiny, protože ostatní dostupné veličiny, které přístroj dokáže měřit (teplota, výkon, ...), nejsou v současné době připojeny, a tudíž jsou nulové. Avšak z pohledu implementace se jedná pouze o přidání dalších adres, kde se veličiny nacházejí v přístroji, do PHP funkce. V budoucích fázích bude záležet na požadavcích a všechny potřebné veličiny by se doplnily.

Z pohledu implementace se zobrazování dat do tabulek neliší od realizace grafu. Pro dotazování dat je vytvořena zvláštní funkce s trochu odlišnou strukturou, ale stejně jako funkce získávající data do grafu, vrací data ve formátu *json*. K propojení PHP funkce se stránkou zobrazující tabulku je opět využito AJAX volání. Když jsou data předána, vypisují pomocí *foreach* smyčky jednotlivé buňky obsahující data z pole. Výslednou tabulku zobrazuje Obrázek 4.

	1	2	3	4
U_{in} [V]	225.96	225.94	225.95	164.83
u_{THD} [%]	3.22	3.23	3.23	3.26
I [A]	0	0	0	0

Obrázek 4: Výsledná tabulka s daty

6.2.7 Dostupné parametry nastavení

Jak jsem již zmiňoval, graf umožňuje nastavit několik parametrů, pomocí kterých se dá přizpůsobovat výsledný vzhled. Každý parametr má vlastní datový typ, který by se měl dodržet při vytváření grafu. Tabulka 5 poskytuje seznam všech proměnných, které se dají využít. Každý z parametru je doplněn o datový typ, kterého by měl nabývat, vysvětlenou funkcionalitu a ukázkový příklad hodnot. Tučně zvýrazněné hodnoty je nutné nastavit při vytváření objektu.

Tabulka 4: Seznam parametrů pro práci s třídou grafu

Název	Datový typ	Popis	Ukázkové hodnoty
canvasId	string	Definuje ID elementu canvas, do kterého se graf vykreslí.	„harmonicsCanvas“
legend	pole hodnot	Názvy popisek pro prvky grafu (sloupce, limity). Na datovém typu hodnot v poli nezáleží.	[„U1“, „U2“, „U3“, „U4“, „limits“]

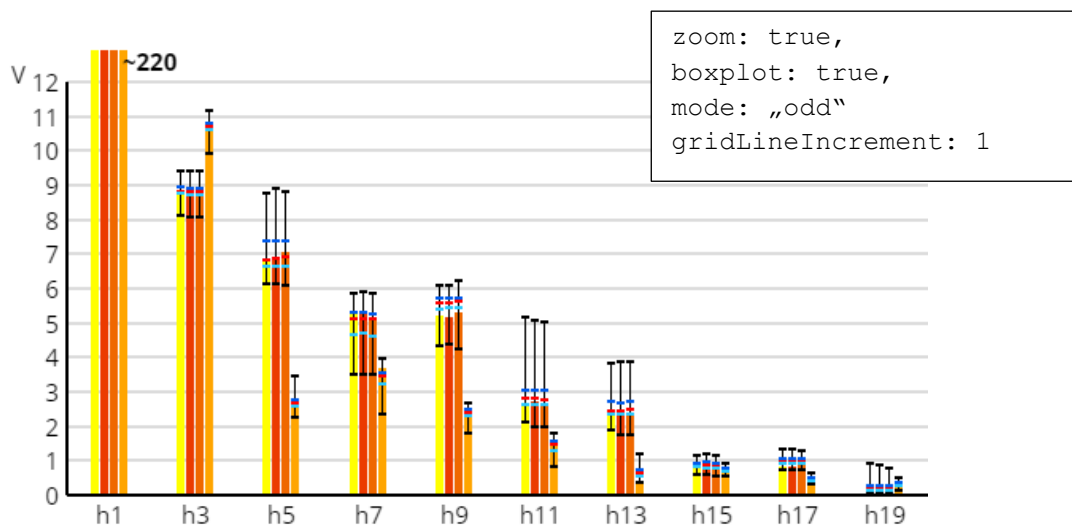
colors	pole hodnot	Nastavení barev pro každý ze zobrazovaných sloupečků.	[„red“, „blue“, „#00ff00“, „lime“]
barWidth	int	Šířka každého sloupečku v pixelech.	5
minValue	int	Minimální hodnota na ose Y.	0
maxValue	int	Maximální hodnota na ose Y.	230
gridLineIncrement	int	Hodnota určující krok, po kterém se budou zobrazovat vodící čáry osy Y.	10
displayValue	pole stringů	Omezení zobrazovaných napětí. Pokud není nastaveno, zobrazují se všechny.	[„U1“, „U3“]
boxplot	boolean	Volba zobrazení krabicového grafu.	true
zoom	boolean	Volba „přiblížení“ hodnot v grafu.	true
mode	string	Množina harmonických složek, které se budou zobrazovat (sudé / liché).	„odd“

6.2.8 Výsledné zobrazení v závislosti na nastavení

V této podkapitole bych se rád věnoval ukázkám výsledných grafů v závislosti na nastavení při vytváření objektu. Ukázky jsou pouze výřezem ze skutečných grafů, kvůli jeho skutečné velikosti. Ke každé variantě je napsána kombinace parametrů, která má za následek výsledný vzhled grafu. Nastavení ID canvasu, barev sloupců, šířku sloupce, minimální a maximální hodnotu mají všechny níže uvedené příklady stejné, a proto nejsou uvedeny.

Zobrazení lichých harmonických s krabicovými grafy

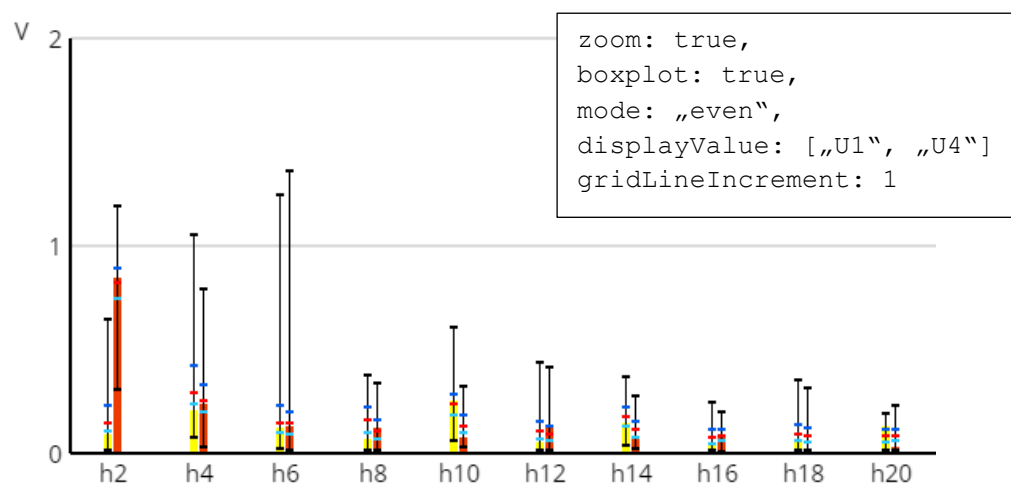
Obrázek 5 zobrazuje graf všech napětí. Ke každé složce je vykreslen krabicový graf spočítaný z 200 načtených hodnot.



Obrázek 5: Zobrazení lichých harmonických složek s krabicovým grafem

Zobrazení harmonických složek pro dvě napětí s krabicovými grafy

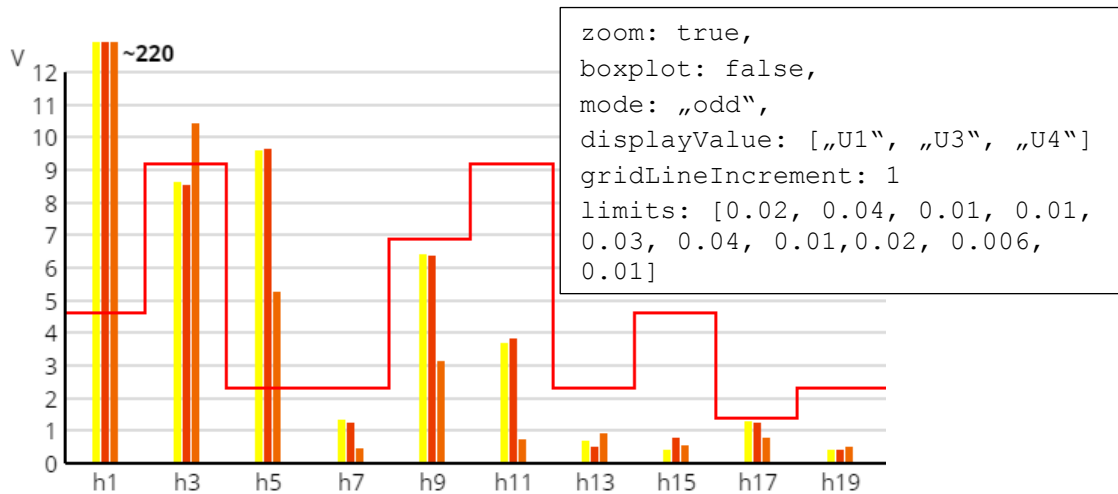
V této ukázce (viz Obrázek 6) jsem nastavil graf pro zobrazení pouze napětí U1 a U4. Jedná se opět o graf s 200 načtenými hodnotami. Pro každou harmonickou složku je znovu vykreslen krabicový graf.



Obrázek 6: Zobrazení sudých harmonických pro dvě napětí s krabicovými grafy

Zobrazení tří napětí s limitami

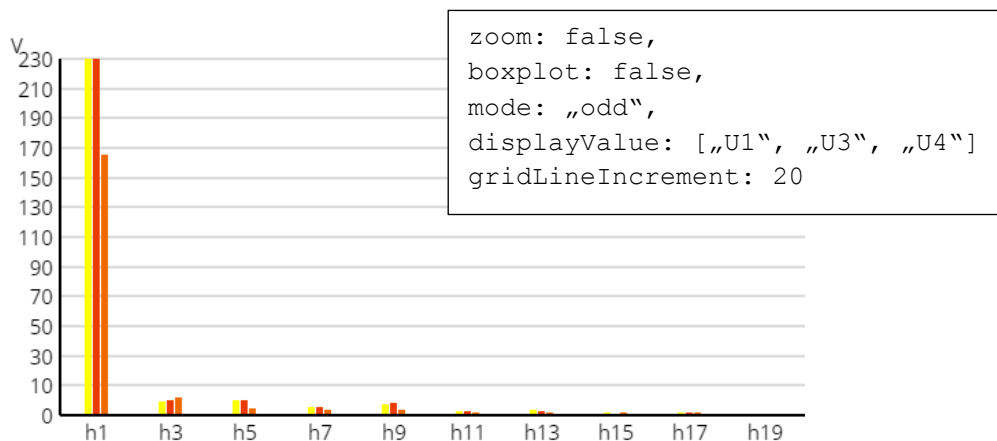
V předposlední ukázce (viz Obrázek 7) je graf tří napětí s vykreslenými limitami (červená klikatá čára). Velikost limit je zadávána jako parametr.



Obrázek 7: Zobrazení lichých harmonických napětí s vyznačenými limitami

Zobrazení tří napětí bez krabicových grafů

Poslední ukázkou zobrazení je vykreslení grafu tří napětí v nepřiblíženém režimu, bez krabicových grafů. Jak si můžete na grafu všimnout (viz Obrázek 8), 19. složka je v tomto zobrazení téměř nulová.



Obrázek 8: Zobrazení lichých složek v nepřiblíženém režimu bez krabicových grafů

6.3 Vyhodnocení rychlosti a spolehlivosti

Rychlost, a v jistém smyslu i spolehlivost, je v současné době omezena především rychlostí a odezvou internetového připojení uživatele, jelikož komunikace s analyzátozem probíhá prostřednictvím internetu. V zásadě se dá říct, že pokud se nastaví doba periodické obnovy dat na jednu a více sekund, neměly by nastat žádné problémy se získáváním dat a uživatel by neměl pozorovat výraznější prodlevu.

Navrhnuté řešení také závisí na operační paměti zařízení, ve kterém se bude zobrazovat. Toto omezení vyplývá z využití JavaScriptu, který je záležitostí klientské strany. Tento typ problému by však neměl nastat, jelikož samotné vykreslování není náročné na paměť a jedinou možnou zátěží by bylo příliš velké množství zaznamenaných dat, která se využívají pro vykreslení krabicových grafů – řádově desetitisíce zaznamenaných dat (jedna sada dat je přibližně 1 kB).

Pro otestování rychlosti komunikace jsem udělal několik testů. V testech jsem měnil periodickou hodnotu pro obnovu dat a překreslení grafu. Tyto hodnoty jsem nastavil na 0,2, 0,5 a 1 sekundu. Po zaznamenání 250 vzorků dat jsem vyhodnotil průměrnou rychlost zpracování dat (komunikace s analyzátozem, předání dat do grafu a vykreslení) v závislosti na celkovém čase. Dále jsem také otestoval dvě různá připojení – jedno pomalejší a druhé rychlé. Tabulka 5 ukazuje výsledky testů s využitím pomalejšího připojení a pod ním s využitím rychlejšího připojení.

Tabulka 5: Výsledky testování s pomalejším připojením

VÝSLEDKY POMALEJŠÍHO PŘIPOJENÍ			
Interval periodické obnovy (s)	0,2	0,5	1
Celkový čas (s)	150,8	255	387
Průměrná odezva zpracování (s)	0,6	1,02	1,55
VÝSLEDKY RYCHLEJŠÍHO PŘIPOJENÍ			
Interval periodické obnovy (s)	0,2	0,5	1
Celkový čas (s)	64,2	140,9	265,8
Průměrná odezva zpracování (s)	0,26	0,56	1,06

Odečteme-li interval periodické obnovy od průběrné odezvy zpracování, dostaneme zpoždění způsobené rychlostí internetového připojení. Jak lze vidět z výsledků pomalého připojení, čím delší je nastavena doba periodické obnovy, tím se průměrná odezva snižuje. Avšak výsledné zpoždění je markantní v porovnání s připojením rychlým. U rychlejšího připojení lze z vyhodnocených dat vidět, že zpoždění činí pouhých 0,06 s, což je téměř zanedbatelná hodnota, která by uživateli neměla překážet.

7 Závěr

Cílem práce bylo seznámení se se způsoby zobrazení veličin v oblasti měření kvality elektrické energie. Na základě získaných poznatků pak udělat návrh jednoduché webové aplikace a následně implementovat řešení pro data získaná z analyzátoru SMC 144 od firmy KMB systems, s.r.o. Veškerá data měla být zobrazována v HTML5 elementu canvas ve formě grafu v reálném čase s danou časovou prodlevou.

Za účelem komunikace s analyzátozem byla vytvořena PHP funkce, která implementuje knihovnu PHPModbus. Získaná harmonická napětí se s využitím AJAX volání předávají do vytvořené třídy grafu. Třída umožňuje zobrazit aktuální harmonická napětí načtená z analyzátoru. Součástí grafu je shromažďování přijatých hodnot, které se využívají pro vykreslení krabicového grafu pro každou naměřenou hodnotu (viz kapitola 6.2.5). Dalším prvkem je zobrazení limitních čar pro jednotlivé harmonické složky. Hodnota limitů je volena při vytváření grafu a tudíž záleží na daných podmínkách, které určují velikost těchto limitů. Druhým typem zobrazení je tabulka (viz kapitola 6.2.6). Jedná se spíše o vedlejší předmět této práce. V tabulkách jsou zobrazena základní data přečtená z přístroje (napětí, proud, celkové harmonické zkreslení).

Všechny stanovené cíle byly splněny a výsledná aplikace je plně funkční. Rychlost a spolehlivost tohoto řešení (viz kapitola 6.3) je závislá na latenci internetového připojení uživatele, jelikož aplikace komunikuje s analyzátozem prostřednictvím internetu. Může tak docházet ke zpožděním při příliš častém dotazování na přístroj (pokud je periodická obnova dat nastavena na méně než jednu sekundu), což bude mít za následek nepravidelné intervaly vykreslování dat, avšak na funkčnost řešení to vliv nemá. Problém se zpožděním by však měl v pozdějším stádiu aplikace odeznít, jelikož je v plánu toto řešení nasadit přímo do přístroje, a tím by odpadla závislost na odezvě internetového připojení. Přístroj implementuje operační systém Linux, na kterém by bylo pouze potřeba zprovoznit webový server Apache. Ten následně nakonfigurovat a nakopírovat mé řešení s potřebnými knihovnami.

7.1 Další rozvoj aplikace

Aplikace v tuto chvíli poskytuje pouze jeden typ zobrazení harmonických napětí. V pozdějších fázích vývoje bychom mohli zařadit do aplikace také graf ve formě fázorového diagramu či tzv. „heat mapy“, což jsou další způsoby, jak harmonická napětí uživateli zobrazovat.

Mezi další funkce bychom mohli zařadit propojení aplikace s již stávající databází v analyzátoru, která by mohla poskytnout naměřená data z minulosti. Díky propojení by se naskytla možnost zobrazovat statické grafy za uživatelem určené období. Stávající implementace by data z databáze byla schopna zobrazit, stačilo by je uložit do požadované struktury.

Seznam použité literatury

- [1] BRADLEY, Allen. *Power system harmonics* [online]. 2009 [vid. 2015-02-18]. Dostupné z: http://literature.rockwellautomation.com/idc/groups/literature/documents/wp/mvb-wp011_-en-p.pdf
- [2] Can I Use. *Canvas (basic support)* [online]. 2015 [vid. 2015-02-08]. Dostupné z: <http://caniuse.com/#feat=canvas>
- [3] Elcom. *BK-TOUCH* [online]. 2015 [vid. 2015-05-02]. Dostupné z: <http://www.elcom.cz/cz/produkty/merici-systemy/merici-systemy-pro-energetiku/software-pro-mereni-a-analyzu/bk-touch>
- [4] Electrotek concepts. *PQView* [online]. 2015 [vid. 2015-04-29]. Dostupné z: <http://www.pqview.com/pqview/>
- [5] Elspec. *Investigator* [online]. 2014 [vid. 2015-04-22]. Dostupné z: <http://www.elspec-ltd.com/PQInvestigator>
- [6] FULTON, Steve a Jeff FULTON. *HTML5 Canvas*. Sebastopol: O'Reilly Media, 2011. ISBN 978-1-449-39390-8.
- [7] HT-Instruments. *APP HTAnalysis* [online]. 2014 [vid. 2015-04-16]. Dostupné z: <http://www.ht-instruments.com/en/products-ht/app-htanalysis>
- [8] Chauvin Arnoux. *METRIX ScopeNet* [online]. 2014 [vid. 2015-04-16]. Dostupné z: <https://play.google.com/store/apps/details?id=scopix.scopenet>
- [9] Janitza electronics GmbH. *APPs Descriptions* [online]. ©2010–2015 [vid. 2015-04-22]. Dostupné z: <http://www.janitza.com/apps-descriptions.html>
- [10] KMB systems. *Analyzátor SMC 144* [online]. 2014 [vid. 2015-01-13]. Dostupné z: <http://www.kmb.cz/index.php/cs/digitalni-merici-pristroj-s-pameti/smc-144-pro-smart-metering>

- [11] KMB systems. *PA 144, SMC 144, SMV, SMVQ, SMP, SMPQ Multifunctional Panel Meter & Power Quality Analyzers Communication protocol* [online]. 2012 [vid. 2015-02-02]. Dostupné z: <http://www.kmb.cz/index.php/cs/component/phocadownload/category/14-dokumenty-komunikace?download=213:popis-komunikacnich-protokolu-modbus-kmblong>
- [12] KMB systems. *SMC 144* [online]. 2013 [vid. 2015-01-13]. Dostupné z: <http://www.kmb.cz/index.php/cs/component/phocadownload/category/5-merici-pristroje?download=194:smc-144-manual>
- [13] NIXON, Robin. *Learning PHP, MySQL & JavaScript, 4th Edition*. Sebastopol: O'Reilly Media, 2014. ISBN 978-1-49191-866-1.
- [14] PEHLIVANIAN, Ara a Don NGUYEN. *Jump start JavaScript*. Collingwood: SitePoint, 2013. ISBN 978-0-9873321-8-9.
- [15] PowerSight. *Data Logger, Power factor meter, Power Quality Analyzer, Three-Phase Power Monitor* [online]. 2015 [vid. 2015-04-23]. Dostupné z: <http://www.powersight.com/HTML/PRODUCTS/PSM-i%20power%20monitor%20software%20for%20iPad.html>
- [16] SANKARAN, C. *Power quality*. Boca Raton: CRC Press, 2001. ISBN 0-8493-1040-7.
- [17] Simply Modbus. *Frequently asked questions* [online]. 2006 [vid. 2015-01-27]. Dostupné z: <http://www.simplymodbus.ca/faq.htm>
- [18] W3C. *The canvas element – HTML5* [online]. 2009 [vid. 2015-02-02]. Dostupné z: <http://www.w3.org/TR/2009/WD-html5-20090825/the-canvas-element.html>

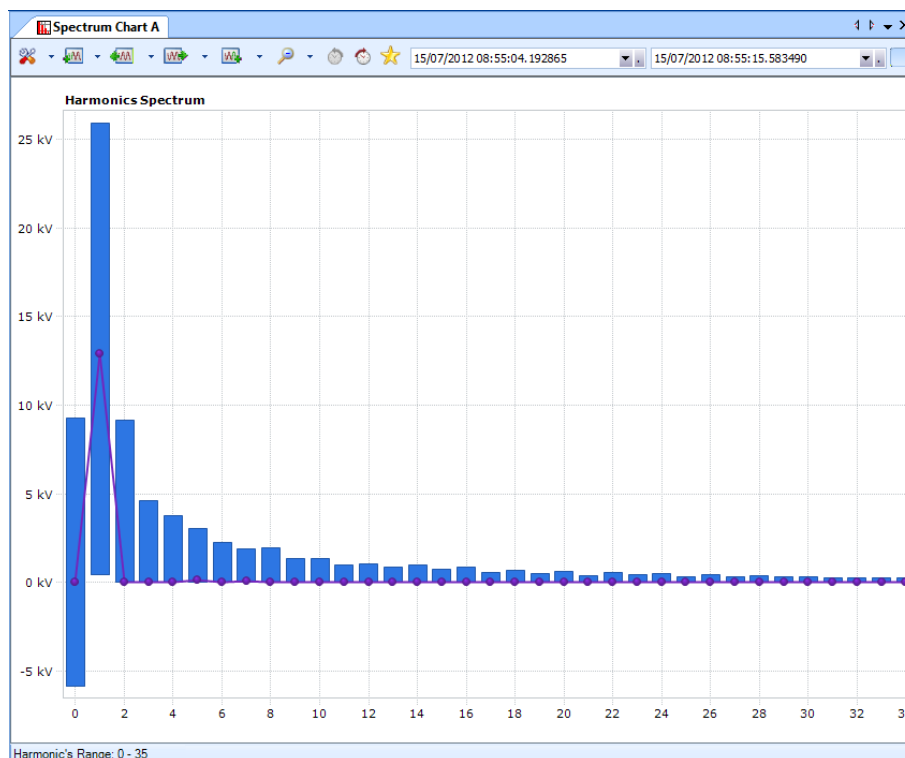
A Obrázkové přílohy

A.1 Webové rozhraní analyzátoru SMC 144

Voltage Harmonics				
Phase	L1	L2	L3	L4
U_{h1} [V]	225.08	225.1	225.12	164.25
U_{h3} [%]	1.82	1.83	1.85	2.89
U_{h5} [%]	1.3	1.31	1.31	0.74
U_{h7} [%]	0.93	0.92	0.93	0.9
U_{h9} [%]	1.41	1.4	1.41	0.95
U_{h11} [%]	0.32	0.3	0.27	0.46
U_{h13} [%]	0.38	0.37	0.37	0.06
U_{h15} [%]	0.08	0.07	0.07	0.17

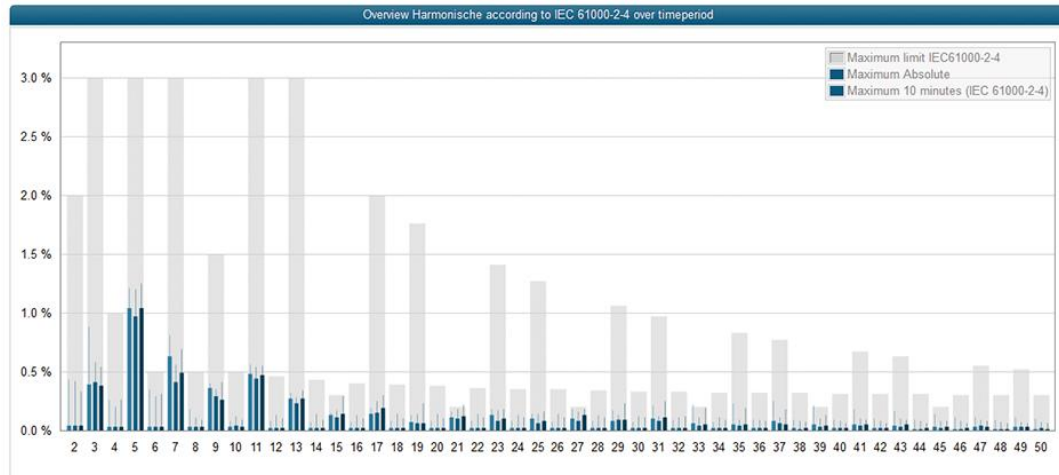
Obrázek 9: Tabulka lichých harmonických napětí analyzátoru

A.2 Program Investigator



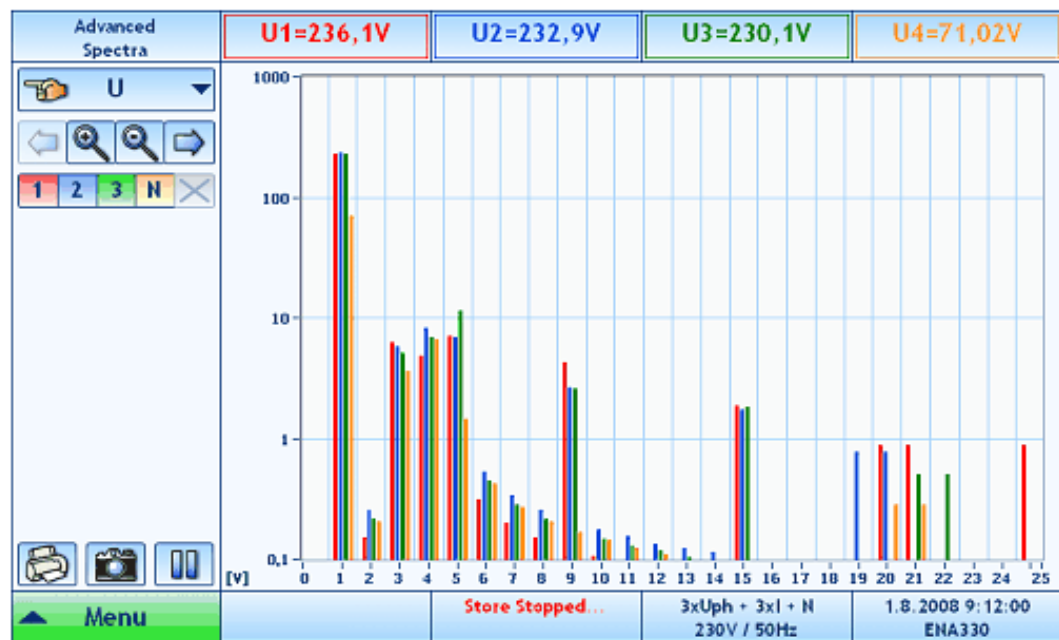
Obrázek 10: Realizace harmonických napětí v programu Investigator [5]

A.3 GUI aplikace APPs



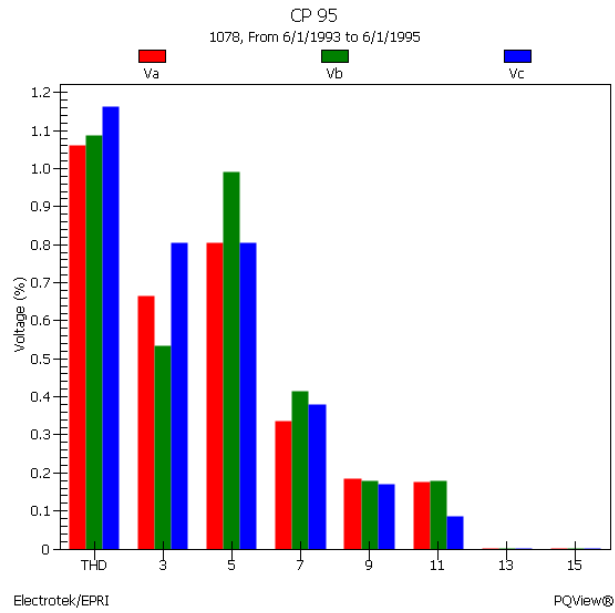
Obrázek 11: Zobrazení harmonických napětí v aplikaci APPs [9]

A.4 BK-TOUCH od Elcom



Obrázek 12: Harmonické složky v aplikaci BK-TOUCH [3]

A.5 PQView



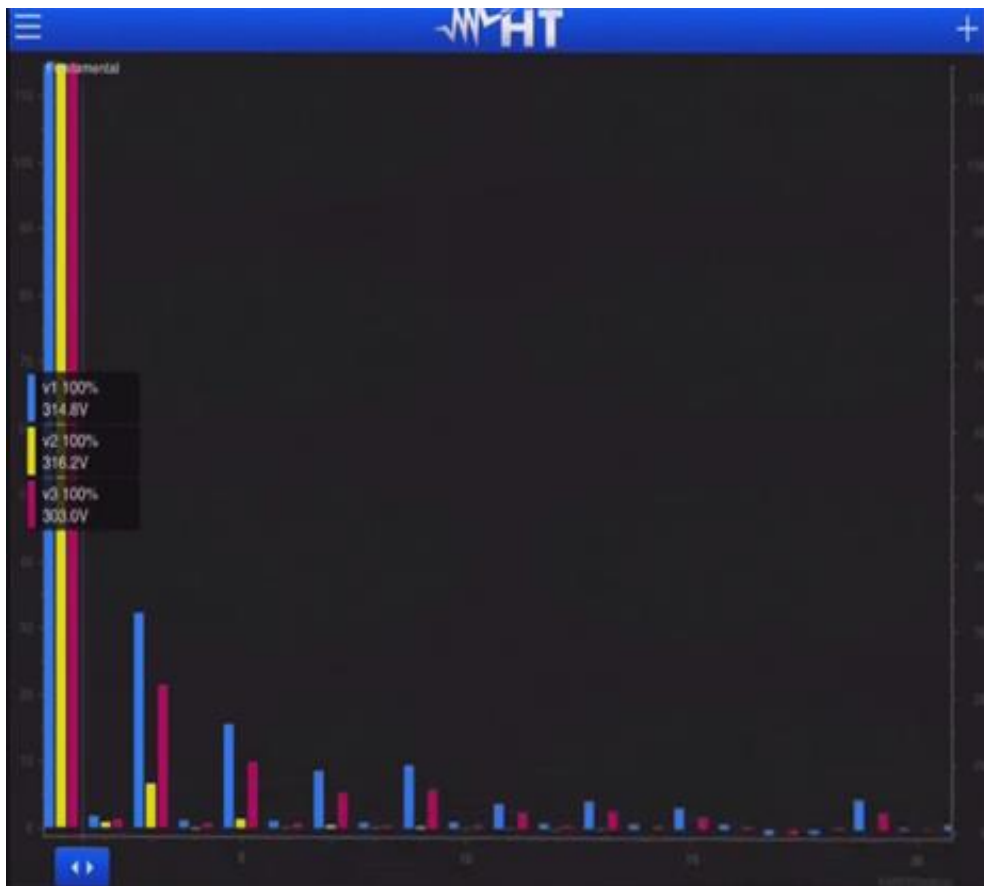
Obrázek 13: Graf harmonických napětí v programu PQView [4]

A.6 Program PSM-i



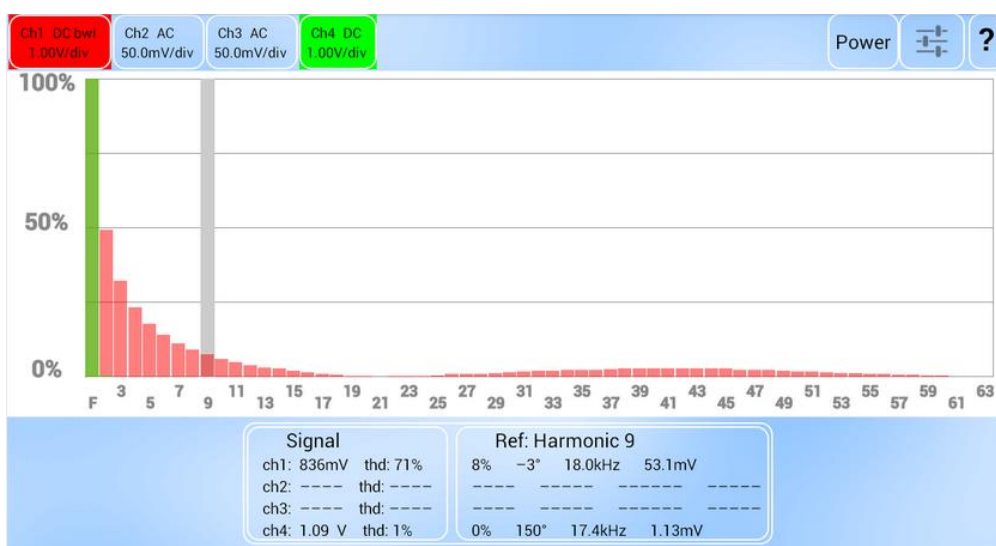
Obrázek 14: Graf harmonických napětí a proudů v aplikaci PSM-i [15]

A.7 Mobilní aplikace HTAnalysis



Obrázek 15: GUI aplikace HTAnalysis [7]

A.8 METRIX ScopeNet



Obrázek 16: Ukázka aplikace METRIX ScopeNet [8]

B Obsah přiloženého CD

- text bakalářské práce
 - bakalarska_prace_2015_Ondrej_Kubicek.pdf
 - bakalarska_prace_2015_Ondrej_Kubicek.docx
 - kopie_zadani_bakalarske_prace_2015_Ondrej_Kubicek.pdf
- zdrojový kód
 - PHPModbus (knihovna pro komunikaci s ModBusem)
 - public (css a JavaScript soubory)
- ukázky grafů (ukázka výsledného vzhledu grafu)