

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

Studijní program: N2612 Elektrotechnika a informatika  
Studijní obor: 3902T005 Automatické řízení a inženýrská informatika

**Grafické uživatelské rozhraní pro měřicí  
přístroje v prostředí operačního systému  
GNU/Linux**

**Graphical user interface for measure devices  
in GNU/Linux operating system  
environment**

**Diplomová práce**

Autor: **Bc. Tomáš Bedrník**  
Vedoucí práce: Ing. Jan Kraus, Ph.D.  
Konzultant: Ing. Viktor Bubla

V Liberci 3. ledna 2013

## Zadání

1. Zanalyzujte stávající řešení grafických uživatelských rozhraní vybraných, běžně užívaných, komerčních analyzátorů kvality.
2. Seznamte se s formáty pro komunikaci a ukládání archivních dat poskytnutých analyzátorů kvality a doplňte jejich podporu (čtení, zápis) v prostředí OS Linux.
3. Rozšiřte vlastní vyvíjené grafické rozhraní o komfortní zobrazení dostupných aktuálních i archivních hodnot přístroje.
4. Navržená řešení řádně zdokumentujte, otestujte jejich správnou funkci a diskutujte další možnosti jejich praktického využití.

## Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou 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 diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou 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.

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum        3. ledna 2013

Podpis

## Poděkování

Chtěl bych poděkovat především vedoucímu diplomové práce ing. Janu Krausovi Ph.D. za pomoc a trpělivost při řešení problémů. Také bych chtěl poděkovat ing. Tomáši Tobiškovi a ing. Pavlu Štěpánovi za pomoc při vývoji knihovny libceaarchive a komunikačním protokolem Modbus.

Samozřejmě také rodině, za podporu ve studiu na vysoké škole.

## Abstrakt

Hlavním cílem práce bylo vytvořit grafického klienta na základě návrhu, který vzniknul během mé bakalářské práce. Klient je určen pro zobrazování aktuálních i archivních dat z přístrojů na měření kvality elektrické energie od firmy KMB systems. Při jeho vývoji byl kladen důraz na snadnost ovládání na dotykových obrazovkách s omezeným rozlišením ve ztížených podmínkách průmyslového použití.

Klient je primárně určen pro běh na operačním systému GNU/Linux bez X serveru na platformě ARM, ale jeho vývoj byl směřován k multiplatformnímu použití. Je možné ho provozovat i na počítačích s běžnými linuxovými distribucemi a částečně i na Windows. Případná portace na další platformy by díky výběru frameworku Qt a minimálnímu použití dalších knihoven neměla být příliš složitá.

V teoretické části se práce věnuje porovnání rozhraní dostupných analyzátorů kvality elektrické energie a navrhuje konkrétní doporučení pro zlepšení uživatelského rozhraní klienta. Dále se věnuje popisu komunikačního protokolu Modbus a archivních souborů „.cea“, které používají přístroje firmy KMB systems. Struktura archivních souborů je složitá a uložená binární data neobsahují informace o jejich struktuře, proto byl vývoj knihovny pro jejich čtení složitý.

V praktické části je popsána struktura aplikace a na konkrétních příkladech jsou ukázány jednotlivé typy zobrazení dat. Dále je popsán a ukázán způsob systems ovládání a nastavení aplikace a jejich jednotlivých částí, především grafů. Jsou zde zmíněna i úskalí kompilace pro různé platformy. Na konci jsou představeny vyvinuté knihovny, jejich struktura a aplikační rozhraní.

Výsledkem práce je grafický klient, který pomocí jedné knihovny dokáže načítat aktuální data z protokolu Modbus. Pomocí druhé knihovny dokáže načítat data z archivů, které vytvářejí některé přístroje z měřených dat. K oběma knihovnám byla vypracována podrobná dokumentace.

*Klíčová slova: klient, komunikace, grafika, GNU/Linux, embedded*

## Abstract

Main goal of thesis was to design a graphic client based on draft designed in my bachelor's thesis. Client is designed to display actual and archive data from power monitoring devices developed by KMB systems company. During the development it was important to design its user interface to be easy-to-use on touch screens with limited resolution and under hard industry conditions.

Client is primarily designed to run on GNU/Linux operating system without X server on ARM platform, but it's development was directed to multiplatform usage. It can be used on computers with common linux distributions and partially also on the Windows OS. Potential porting to another platforms shouldn't be too difficult because of chosen Qt framework and minimal usage of other libraries.

In theoretical part the thesis compares user interfaces of various available power quality analysers and proposes particular recommendations for improving the client's graphical interface. Further it is concerned on description of communication protocol Modbus and archives „.cea“, which are used by devices, developed by KMB systems. The structure of the archives is complex and binary data stored do not contain information on the structure itself, therefore the development of the library for accessing the files was problematic.

In practical part the structure of the application is described and on particular examples different types of data presentations are shown. Further, means of application settings and usage are shown, mainly of the graphs. Some problems encountered during compilation for different platforms are described here. Finally, the developed libraries - their structure and application interface - are presented.

Outcome of the work is a graphical client capable of accessing current real-time data with the use of a designated library and the Modbus protocol. With the use of another library it can read data from archives, created by some of the devices. Both of the libraries are provided with a detailed documentation.

*Key words: client, communication, graphics, GNU/Linux, embedded*

# Obsah

<b>Prohlášení</b>	<b>3</b>
<b>Poděkování</b>	<b>4</b>
<b>Abstrakt</b>	<b>5</b>
<b>1 Úvod</b>	<b>12</b>
<b>2 Analýza stávajících řešení</b>	<b>13</b>
2.1 ENVIS.Daq . . . . .	13
2.2 PQube . . . . .	13
2.3 BK-Touch . . . . .	14
2.4 Výsledky analýzy . . . . .	14
2.5 Archivní data . . . . .	15
<b>3 Zdroje dat</b>	<b>16</b>
3.1 Modbus . . . . .	16
3.2 KMB Long . . . . .	17
3.3 Archivy .cea . . . . .	18
3.3.1 Soubory arch . . . . .	18
<b>4 Framework Qt</b>	<b>20</b>
4.1 Signály a sloty . . . . .	21
4.2 Grafická knihovna QWT . . . . .	21
<b>5 Realizace</b>	<b>23</b>
5.1 Knihovna libacdata . . . . .	23
5.2 Knihovna libceaarchive . . . . .	24
5.3 Struktura aplikace . . . . .	24
5.4 Aktuální data . . . . .	24
5.4.1 Společná část . . . . .	25
5.4.2 Proudý a napětí . . . . .	25
5.4.3 Výkony . . . . .	26
5.4.4 Fázory . . . . .	26
5.4.5 Uživatelsky definované hodnoty . . . . .	26

5.4.6	Grafy . . . . .	27
5.4.7	Elektroměr . . . . .	29
5.5	Archivy . . . . .	29
5.5.1	Grafy . . . . .	30
5.5.2	Nastavení . . . . .	31
5.5.3	Uživatelské grafy . . . . .	32
5.6	Uživatelské rozvržení . . . . .	33
5.6.1	Nastavení . . . . .	34
5.7	Nastavení aplikace . . . . .	35
5.8	Virtuální klávesnice . . . . .	35
5.9	Překlad a podporované platformy . . . . .	36
5.9.1	Embedded Qt . . . . .	36
5.9.2	ARM procesory . . . . .	37
5.9.3	Windows . . . . .	38
5.9.4	Android . . . . .	40
5.10	Překlad jazyka aplikace . . . . .	40
5.11	Konfigurace aplikace . . . . .	42
<b>6</b>	<b>Knihovny</b>	<b>42</b>
6.1	Libactdata . . . . .	42
6.2	Libceaarchive . . . . .	43
<b>7</b>	<b>Závěr</b>	<b>46</b>
	<b>Seznam použité literatury</b>	<b>47</b>
	<b>Příloha A - Porovnání grafického rozhraní konkurenčních přístrojů</b>	<b>49</b>
	<b>Příloha B - Ukázky zdrojových kódů</b>	<b>64</b>
	<b>Příloha C - Obsah CD</b>	<b>72</b>



## Seznam obrázků

1	Základní Modbus rámec jak je popsán v [1]	16
2	Modbus komunikace, jak je popsána v [1]	17
3	Schéma aplikace	23
4	Zobrazení aktuálních dat	25
5	Tabulka proudů a napětí	26
6	Uživatelsky definovaná data	27
7	Graf s aktivním nástrojem „ukazovátko“	27
8	Archivní data, graf ULL	30
9	Archivní data, kombinovaný graf ULN a IL	31
10	Nastavení zobrazení archivních dat	31
11	Archivní data, uživatelský graf	33
12	Ukázka záložky s uživatelským rozvržením	33
13	Nastavení uživatelského rozvržení	34
14	Virtuální klávesnice	36
15	Apliace v emulované systému na Advanced Risc Machine (ARM) platformě	38
16	Port aplikace na Windows	40
17	UML diagram knihovny Libactdata	43
18	UML diagram knihovny Libcearchive	44

## Seznam zdrojových kódů

1	Ukázka použití virtuální klávesnice	35
2	Příkaz pro spuštění emulovaného systému	37
3	Ukázka přidání textového popisku připraveného pro překlad	41
4	Ukázka výběru jazyka aplikace	41
1	Ukázka použití knihovny libcearchive	64
2	Ukázka konfiguračního souboru projektu libcearchive	65
3	Ukázka použití knihovny libactdata	66
4	Ukázka konfiguračního souboru	67

## **Seznam použitých zkratk**

**ADU** Application Data Unit

**API** Application Programming Interface

**ARM** Advanced Risc Machine

**BSD** Berkeley Software Distribution

**OSI** Open Systems Interconnection

**GNU** GNU's Not Unix!

**GPL** General Public Licence

**GUI** Graphic User Interface

**HMI** Human Machine Interface

**HTML** HyperText Markup Language

**initrd** initial RAM disk

**IP** Internet Protocol

**LCD** Liquid Crystal Display

**LGPL** Lesser General Public Licence

**MOC** Meta Object Compiler

**OS** Operační systém

**PDU** Protocol Data Unit

**QEMU** Quick EMUlator

**QWT** Qt Widgets for Technical Applications

**RAM** Random-access memory

**RTU** Remote Terminal Unit

**TCP** Transmission Control Protocol

**UML** Unified Modeling Language

**VPN** Virtual Private Network

**XML** Extensible Markup Language

# 1 Úvod

Zadání diplomové práce navazuje na semestrální projekty a moji bakalářskou práci [7]. Cílem práce bylo vyvinout grafického klienta pro zobrazování dat měřicích přístrojů firmy KMB systems. Návrh klienta vznikl již během mé bakalářské práce, ale rozhraní vycházelo striktně z firemní aplikace ENVIS.Daq, která je určená pro klasický desktop. Pro zlepšení ovládání aplikace na dotykovém displeji ve ztížených podmínkách bylo nutné zanalyzovat stávající řešení a vybrat vlastnosti vhodné pro nasazení v zamýšleném prostředí.

Od bakalářské práce se rozšířilo množství dat, které přístroje poskytují pomocí protokolu Modbus, bylo tedy nutné rozšířit i knihovnu pro čtení aktuálních dat. Navíc bylo nutné kompletně přepracovat rozhraní knihovny, které již nevyhovovalo novým požadavkům aplikace. Původní návrh klienta uměl zobrazovat pouze data v tabulkách a fázory. Základní podpora grafů byla doplněna v semestrálním projektu. V rámci diplomové práce bylo nutné především rozšířit možnosti grafů a implementovat podporu pro přizpůsobování velikosti písma rozměrům okna aplikace.

Dalším velkým cílem práce bylo vytvořit knihovnu pro čtení a zápis archivních dat. Aplikace měla dokázat přečíst archivy, které měřicí přístroje vytvářejí z nasbíraných dat a zobrazit je co nejčitelněji v omezeném prostředí aplikace. Formát archivů byl vyvinut ve firmě KMB systems a data jsou ukládána v binární podobě bez informací o jejich struktuře, proto byl vývoj knihovny náročný a zdlouhavý.

## 2 Analýza stávajících řešení

Jedním z cílů práce bylo porovnat stávající řešení zobrazování aktuálních dat. Výrobci zařízení pro měření kvality elektrické energie existuje pouze několik, pro účely analýzy byly vybrány řešení PQube od firmy Power Standards Lab, BK-Touch od firmy ELCOM, a.s. a jako referenční typ zobrazení byl použit software ENVIS.Daq od firmy KMB systems.

Vyvíjený klient musí umožňovat zobrazovat aktuální i archivní data v grafické podobě. Jeho primární určení je pro dotykové obrazovky, kde poběží v celoobrazovkovém režimu jako primární aplikace. Tomuto musí odpovídat i ovládání aplikace. Není určeno konkrétní zařízení, na kterém má běžet, proto je nutné počítat s různými rozlišeními a přizpůsobit tomu klienta. Ovládání musí být přizpůsobeno zvyklostem pro dotykové obrazovky a nedostupnosti klávesnice. Všechny výše uvedené vlastnosti nespĺňuje žádná dostupná aplikace.

Cílem analýzy je vybrat z porovnávaných aplikací ty vlastnosti, které vyhovují předpokládanému použití vyvíjeného klienta.

### 2.1 ENVIS.Daq

Grafický klient, který je výstupem této práce je rozšířením základního návrhu klienta, který vznikl během mé bakalářské práce. Tento klient vychází z desktopové aplikace ENVIS.Daq firmy KMB systems. Aby byla zachována alespoň částečná kontinuita ovládání napříč aplikacemi firmy KMB systems, bylo by vhodné, aby výsledný grafický klient v rámci možností zachoval rozvržení aplikace ENVIS.Daq.

ENVIS.Daq je aplikace určená pro desktopové použití na Operační systém (OS) Windows, tomu odpovídá i ovládání. Pro využití v požadovaných podmínkách jsou ovládací prvky špatně dostupné, barvy málo výrazné a zobrazovaná data malá. [15]

### 2.2 PQube

Tato aplikace se zobrazuje v běžném internetovém prohlížeči jako HyperText Markup Language (HTML) stránka. Obnova dat se provádí ručně novým načtením celé stránky, což je možné nastavit automaticky po zvoleném čase v internetovém prohlížeči, pokud to podporuje. Není možné nastavit jaké veličiny budou zobrazovány – zobrazí se všechny, které typ, zapojení a nastavení přístroje umožňuje. Data jsou zobrazena ve dvousloupcových tabulkách nebo jako grafy ve statických obrázcích. [16]

Oproti ostatním porovnávaným aplikacím vyniká v zobrazení grafů, které jsou sice sta-

tické, ale velmi dobře provedené. Obsahují histogram a pokud jsou dostupná data pro minimum, maximum a průměr dané veličiny, minimum a maximum je vykresleno světlejší barvou jako plocha, ve které je sytější křivka označující průměrnou hodnotu.

### 2.3 BK-Touch

Jako jediná z porovnávaných aplikací je pro dotykové ovládání určena aplikace BK-Touch. Z požadavků nespĺňuje pouze nutnost přizpůsobení různým rozlišením. Barvy jsou výrazné, zobrazovaná data i grafy dobře čitelné, přepínání mezi různými daty jednoduché a použitelné pro dotykové ovládání.

Obsahuje tabulky, do kterých může uživatel zvolit vlastní veličiny. Umožňuje navolení několika různých tabulek a mezi nimi potom přepínat. Obsahuje též grafy, do kterých mohou uživatelé volit vlastní veličiny. Oproti aplikaci ENVIS.Daq v ní chybí především nastavení intervalu načítání dat.[13]

Na této aplikaci je vidět, jak má vypadat aplikace určená pro dotykové ovládání a ztížené podmínky v průmyslovém prostředí, ale její ovládání je dost odlišné od aplikace ENVIS.Daq. Z těchto důvodů je vhodné se zde inspirovat v mnoha věcech, ale zachovat pokud možno rozhraní aplikace ENVIS.Daq.

### 2.4 Výsledky analýzy

Kompletní analýza je v příloze A. Byla vypracována před začátkem práce, v průběhu byly doporučení, které z analýzy vyplynuly, přizpůsobovány aktuálním potřebám. Celá analýza by se dala shrnout v několika bodech, konkrétní připomínky jsou v příloze.

- V co možná největší míře zachovat rozhraní aplikace ENVIS.Daq,
- používat výrazné barvy,
- velké ovládací prvky použitelné pro dotykové ovládání,
- vyhnout se použití pravého tlačítka myši,
- vytvořit virtuální klávesnici pro psaní textu,
- na všech stránkách vždy viditelná tabulka s uživatelsky definovanými veličinami,
- tabulka nebo widget s uživatelsky definovanými veličinami s možností vybrat velikost i barvy,

- grafy s uživatelsky volitelnými veličinami s podrobným nastavením vlastností grafu,
- minimum a maximum zobrazovat v grafu světlejší barvou jako plochu, ve které se pohybuje křivka průměrné hodnoty.

## 2.5 Archivní data

Analýza se nezabývala zobrazováním archivních dat, protože existující aplikace pro zobrazování archivů jsou určena pro desktop a umožňují komplexní analýzu dat. Budoucí aplikace se nemá pokoušet o komplexní analýzu dat, má pouze zobrazovat základní archivní data. Výsledky analýzy zobrazování aktuálních dat jsou dobře použitelné i pro archivní data.

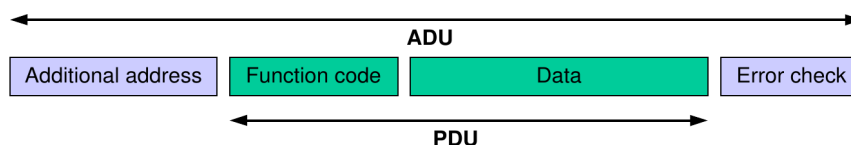
### 3 Zdroje dat

Aplikace rozlišuje dva typy dat – aktuální a archivní. Aktuální data jsou získávána z měřicího přístroje za běhu – postupně tak, jak je měří a vypočítává. Přístroje KMB podporují dva komunikační protokoly – Modbus a KMB Long. Archivní data jsou ukládána některými měřicími přístroji a stahují se najednou do archivů cea. Tyto archivy jsou určeny primárně pro analyzační software ENVIS.

#### 3.1 Modbus

Modbus je komunikační protokol na aplikační vrstvě Open Systems Interconnection (OSI) modelu. Poskytuje komunikaci typu klient/server mezi zařízeními připojenými na různých typech sběrnic nebo sítí. Od roku 1979 je de facto standardem a je používán na milionech zařízeních.

Protokol definuje jednoduché datové jednotky – Protocol Data Unit (PDU), které jsou nezávislé na komunikačních vrstvách, které leží pod ní. Při použití protokolu na konkrétní sběrnici nebo síti mohou být použita další pole v aplikační datové jednotce – Application Data Unit (ADU). Maximální délka PDU je limitována první implementací po sériové lince RS485 na 253 bytů. Komunikace probíhá systémem požadavek – odpověď na základě funkčních kódů. [1]



Obrázek 1: Základní Modbus rámeček jak je popsán v [1]

Funkčních kódů je 255, 1-127 pro požadavek a 128-255 pro odpovídající chybu. Kódy 1-65, 72-100 a 110-127 jsou veřejné, definované ve specifikaci, 66-71 a 101-126 je volný pro uživatelské použití. [1]

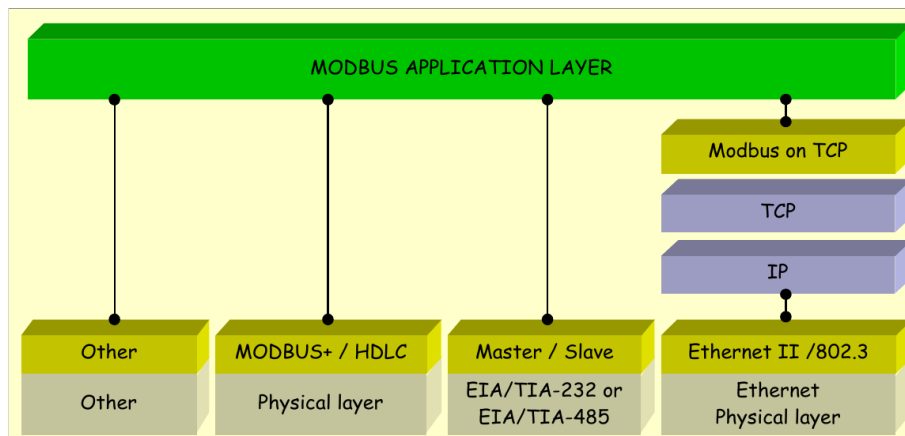
Protokol používá big-endian reprezentaci dat a adres. Data jsou historicky rozdělena do čtyř základních tabulek - diskrétní vstupy, cívky, vstupní registry a zadržovací registry. Každá tabulka má vlastní funkční kódy pro čtení a zápis. Toto rozdělení se dnes již příliš nedodrжуje, často se slučují vstupní a zadržovací registry – používají stejnou paměť a je možné zaměňovat i funkční kódy. Tak je tomu i v případě měřicích přístrojů firmy KMB



systems.

V současné době existují tři varianty modbusu. Jedna, označovaná jako Modbus TCP, využívá Transmission Control Protocol (TCP)/Internet Protocol (IP) po Ethernetu, Modbus Remote Terminal Unit (RTU) využívá sériovou linku RS485 nebo RS232 a poslední komunikuje po Modbus+, což je vysokorychlostní síť s předáváním pověření. [1]

Vyvíjená aplikace podporuje Modbus RTU a TCP.



Obrázek 2: Modbus komunikace, jak je popsána v [1]

Pomocí protokolu TCP/IP je možné komunikovat jak po lokální síti, tak přes internet, kde je ovšem nutné, aby server (měřicí přístroj) měl veřejnou IP adresu. Tento požadavek lze obejít použitím Virtual Private Network (VPN).

### 3.2 KMB Long

Firma KMB systems vyvinula vlastní protokol pro komunikaci s měřicími přístroji, protože Modbus nevyhovoval všem požadavkům. Některá aktuální data jsou dostupná pouze přes tento protokol. Díky tomu, že je protokol určen k jednomu konkrétnímu účelu, může být velice jednoduchý.

Komunikace je typu master/slave a požadavek/odpověď. Komunikační kanál je nastaven na 8 bitů, žádná parita a jeden stop bit. [2]

Každá zpráva – požadavek a odpověď mají jednoduchý formát:

- Adresa přístroje (1 byte)
- Délka zprávy v bytech (2 byty)
- Typ zprávy (1 byte)

- Tělo zprávy (různá délka)
- Kontrolní součet modulo 256 (1 byte)

Tento protokol není ve výsledné aplikaci implementován, bylo by nutné naprogramovat příslušnou knihovnu, což by překračovalo rozsah diplomové práce.

### 3.3 Archivy .cea

Pro ukládání archivních dat z přístrojů je využíván proprietární formát .cea. K tomu formátu neexistuje veřejná dokumentace, pouze několik interních dokumentů, které se struktuře archivů věnují jen povrchně. Vývoj tedy probíhal na základě konzultací ve firmě KMB systems a útržcích zdrojových kódů v C# z aplikace ENVIS.

Základní soubor s příponou cea je standardní zip archiv, který má pouze změněnou příponu. Archiv obsahuje složku Info a jeden nebo více souborů s příponou arch. Složka info dále obsahuje soubory TreeList.xml a VersionHistory.txt.

Soubor TreeList.xml je standardní Extensible Markup Language (XML) soubor, který obsahuje informace o přístroji, který archiv zaznamenal. Je zde uloženo jméno přístroje, verze bootloaderu, adresa, číslo, typ, verze hardwaru, softwaru a další údaje.

Soubor VersionHistory.txt obsahuje informace o softwaru, který archiv vytvořil.

#### 3.3.1 Soubory arch

Vlastní archivní data jsou uložena v souborech s příponou arch. Těchto souborů může být, a většinou i je, víc. Každý potom obsahuje jiná data. Jména souborů jsou určena podle přístroje, ze kterého pocházejí, typu archivu a data stažení z přístroje.

Typů archivů je několik, například Hlavní archiv, Elektroměr, HDO, LOG, M Profil, Pmax archiv, PQ Hlavní archiv, PQ oscilogram, PQ průběh událostí, PQ události nebo S Profil. Data v jednotlivých arch souborech jsou binární a dělí se na tři základní části.

#### Hlavička

První částí je hlavička. Data zde jsou kódována jako little endian. Je zde uložena verze archivu, typ, počet archivů uvnitř jednoho souboru, časy začátku a konce archivu a počet setů nastavení.

## **Konfigurace**

Po hlavičce následuje část s nastavením. Data jsou zde kódována jako big endian. Tato část může obsahovat několik konfigurací, ale většinou je pouze jedna. Každá konfigurace se skládá z několika struktur, které obsahují nastavení přístroje. Nejdůležitější pro další čtení souboru je informace o koeficientu, kterým se násobí napěťové a proudové veličiny měřené přístrojem, které jsou často 1, ale ne ve všech případech.

## **Data**

Poslední a největší část obsahuje vlastní data, která jsou kódována jako big endian. Data jsou uložena po záznamech – jeden záznam odpovídá jednomu měřicímu cyklu přístroje. Na začátku datové části je uveden počet cyklů, jejich délka a konfigurace, ke které patří. Datová část může obsahovat data pro více různých konfigurací.

## 4 Framework Qt

Pro vývoj klienta bylo nutné vybrat grafickou knihovnu. V prostředí GNU's Not Unix! (GNU)/Linux existuje knihoven, které by byly použitelné pro vývoj klienta, velké množství. Framework Qt byl jako základ pro vývoj zvolen již během mé bakalářské práce. Qt byl vybrán především pro svoji multiplatformnost, snadnost použití a možnost běhu na linuxových embedded zařízeních bez X serveru. Většina ostatních porovnávaných možností byly pouze knihovny. Na rozdíl od nich framework mění i způsob programování tím, že rozšiřuje možnosti standardního C++ o vlastní syntaxi. Jeho možnosti jsou také mnohem širší, než pouhé vykreslování grafické části aplikace.[7]

Qt je multiplatformní framework pro vývoj nejen grafických aplikací, který je rozšířen především v prostředí GNU/Linux, ale je dostupný pro velké množství platform. V současné době podporuje Windows, Windows CE a Windows Mobile, Linux/X11 (včetně ostatních unix OS, např. Berkeley Software Distribution (BSD)), Embedded Linux, Windows Embedded, Mac OS X, Symbian, Nokia N9 (MeeGo), INTEGRITY, QNX, VxWorks (Real-Time operační systémy) a Android (v současné době v experimentálním stádiu).

Díky podpoře komunity existují porty i pro další platformy, které nejsou oficiálně podporovány a část z nich je pouze experimentální. Mezi tyto platformy patří Haiku, OS/2, OpenSolaris, iPhone, webOS, Amazon Kindle DX, Blackberry a další. První verze frameworku vyšla v roce 1992, v současné době vývoj vede komunitní Qt Project a firma Digia, která poskytuje komerční variantu frameworku a klade si za cíl portaci na platformu Android a Windows 8. [3]

Framework umožňuje vývoj grafických i textových programů a obsahuje velké množství modulárních komponent. Nad základním jádrem jsou komponenty pro Graphic User Interface (GUI), 2D grafiku, 3D grafiku (OpenGL), multithreading, jádro webového prohlížeče webkit, XML, D-Bus, multimédia, síť a mnoho dalších. Jako jazyk je použito C++, které je doplněno o další vlastnosti, které zajišťuje předkompilátor – Meta Object Compiler (MOC). Oproti čistému C++ je syntaxe rozšířena především o signály a sloty.

Jsou dostupné tři licence – General Public Licence (GPL) verze 3.0, Lesser General Public Licence (LGPL) verze 2.1 a komerční. LGPL licence zakazuje především statické linkování, požaduje poskytnout upravené verze frameworku pod stejnou licenci a není dostupná pro všechny výše jmenované platformy. GPL verze knihovny také není dostupná pro všechny platformy a je vhodná pokud i výsledný program je vydán pod licenci GPL. Komerční verze

nemá tyto omezení a navíc obsahuje placenou podporu, více modulů (například modul pro diagramy), lepší vývojářské nástroje a další výhody. Vývoj aplikace směřuje k jejímu komerčnímu užití, připadá v úvahu použití LGPL nebo komerční licence. V současné době postačuje použití LGPL, protože její omezení ovlivňují použití knihovny pouze minimálně. Nejvýznamnější vlastnost, kterou by bylo vhodné využít z komerční verze knihovny, je statické linkování, ale tato výhoda nevyváží cenu za licenci.

## 4.1 Signály a sloty

Pro komunikaci mezi objekty používá Qt systém signálů a slotů. Ostatní toolkity pro tento typ komunikace využívaly callback.

Callback je ukazatel na funkci. Když chce probíhající funkce oznámit nějakou událost, je jí předán ukazatel na jinou funkci (callback). Probíhající funkce zavolá callback když je potřeba. To má dvě zásadní chyby. Není to typově bezpečné – nikdy není jisté, že probíhající funkce zavolá callback se správnými argumenty. Druhou nevýhodou je silná provázanost callbacku a probíhající funkce, protože funkce musí vědět, jaký callback zavolat.

Signál v Qt je vyslán při nějaké události, například stisku tlačítka. Qt má mnoho předdefinovaných signálů, ale je možné vytvářet vlastní signály. Slot je funkce, která se zavolá jako odpověď na vydaný signál. Opět existuje mnoho předdefinovaných slotů, ale je možné vytvářet vlastní.

Tato metoda komunikace je typově bezpečná – signál musí odpovídat slotu, jinak se nespojí a umožňuje malou provázanost signálu a slotu. Třída, která vysílá signál neví, kde je zpracován a slot neví, odkud signál přišel. Aby byl při přijetí signálu zavolán patřičný slot, je nutné spojit odpovídající signály a sloty metodou *connect*. Je možné spojit víc signálů k jednomu slotu a jeden signál může být napojen na více slotů.[4]

Signál je zvláštní metoda, která nemá tělo a nevrací hodnotu, má pouze prototyp. Slot je běžná metoda, kterou můžeme spojit se signálem nebo jí volat i přímo.

## 4.2 Grafická knihovna QWT

Pro zobrazování grafů je v aplikaci použita knihovna Qt Widgets for Technical Applications (QWT), která je nejpoužívanějším řešením pro zobrazování grafů v Qt. Za vývojem současné verze stojí pouze jeden vývojář – Uwe Rathmann, přesto knihovna nabízí rozsáhlé možnosti pro zobrazení grafů, měřítek, posuvníků, ciferníků, kompasů, teploměrů a dalších součástí GUI použitelných v technických aplikacích.

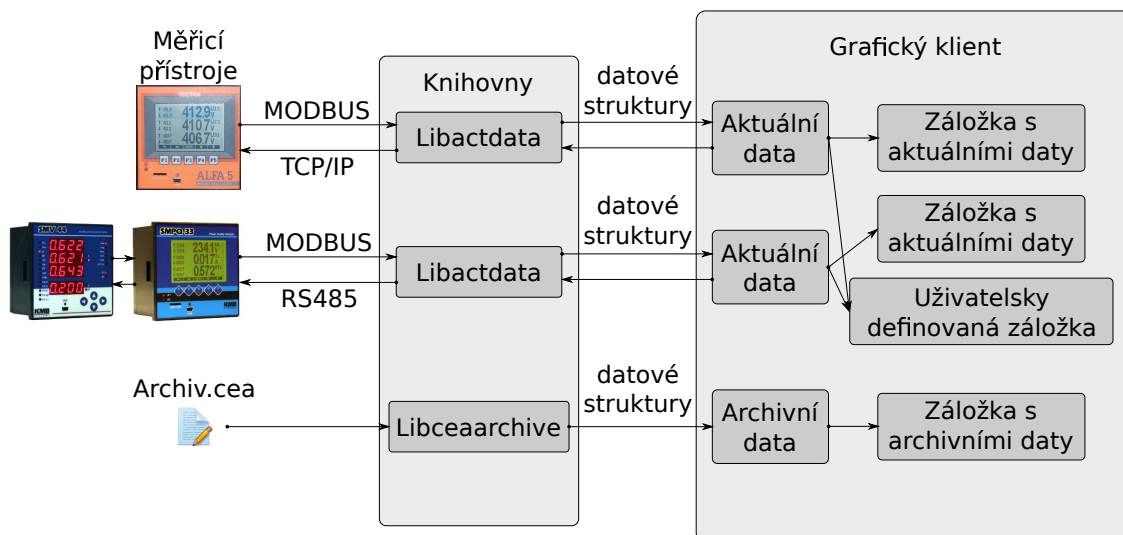
Pro zobrazení grafů nabízí mnoho užitečných vlastností. Umožňuje zobrazení dvou nezávislých os  $X$  a  $Y$ , které mohou být i logaritmické. Obsahuje i nástroje pro Zoom, posun grafu a kurzory. Další její výhodou je její jednoduchá rozšiřitelnost. Je velice modulární a není problém pro požadovanou funkci vytvořit odvozenou třídu, která bude naše požadavky implementovat. Mnoho metod již obsahuje metody, které přenechají část svých funkcí externím třídám.

Knihovna je distribuována pod vlastní licencí, která je odvozená od LGPL, na rozdíl od ní umožňuje statické linkování a je méně restriktivní. [5]

## 5 Realizace

Aby bylo možné dosáhnout co největší modularity, byla oddělena samostatná grafická část a dvě knihovny pro získávání dat.

Samotná grafická aplikace je napsaná v C++ s frameworkem Qt a knihovnou QWT pro zobrazování grafů. Aplikace používá pro načítání dat již zmíněné knihovny.



Obrázek 3: Schéma aplikace

### 5.1 Knihovna libacdata

Pro čtení aktuálních dat byla vyvinuta knihovna libacdata, která vznikla při mé bakalářské práci. V rámci diplomové práce byla rozšířena o podporu novější verze měřicích přístrojů a bylo změněno Application Programming Interface (API), které nevyhovovalo nové struktuře grafické aplikace.

Knihovna podporuje načítání dat pomocí protokolu Modbus a z historických důvodů i XML souborů. Pro Modbus využívá LGPL knihovnu Libmodbus, která je nejpoužívanější knihovnou pro tento protokol v prostředí GNU/Linux a distribuce ji začínají zahrnovat jako standardní součást. Pro čtení XML souborů je využita knihovna libxmlrw, která vznikla v rámci semestrálního bakalářského projektu a využívá standardní linuxovou knihovnu libxml2.

Knihovna je naprogramovaná v čistém C++, aby bylo možné ji využít v případných dalších projektech v rámci KMB systems. Je modulární a přidání dalšího podporovaného protokolu není problém, v budoucnu se počítá s podporou protokolu KMB Long.

## 5.2 Knihovna libceaarchive

Pro čtení dat z archivů .cea je použita knihovna libarchive, která byla nově naprogramována, protože žádná hotová neexistuje. Jak již bylo zmíněno výše, největší problém při její tvorbě byla absence dokumentace a více verzí archivů. Knihovna je opět naprogramovaná v čistém C++. V současné době umí číst hlavní archiv a archiv elektroměru. Přidávání dalších archivů není vzhledem k nedostupnosti dokumentace příliš jednoduché a není to prioritou vývoje. Pro rozbalování .cea souborů je použita knihovna ZZiplib, která je šířena pod LGPL licenci.[6]

## 5.3 Struktura aplikace

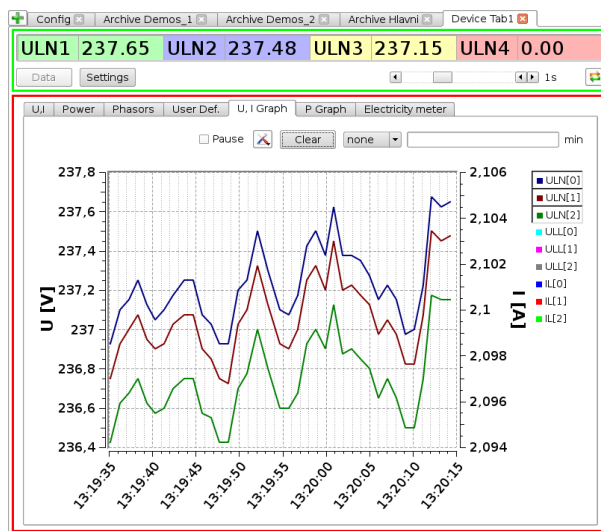
Uživatelské rozhraní aplikace se skládá ze jednotlivých záložek, které zobrazují data nebo konfigurace. Existují čtyři druhy záložek. Jeden druh je spojen s konkrétním měřicím přístrojem, zobrazuje aktuální měřená data a umožňuje jeho konfiguraci. Druhý typ zobrazuje data z archivů, třetí umožňuje vlastní uspořádání aktuálních dat z více přístrojů a čtvrtý slouží pro nastavování aplikace.

## 5.4 Aktuální data

Záložka s aktuálními daty je vázána na konkrétní přístroj. Skládá se ze dvou částí – zobrazování aktuálních dat a konfigurace. Do konfigurační části se během diplomové práce pouze přidala jedna záložka a je popsána v mé bakalářské práci.[7]

Část, zobrazující aktuální data, byla téměř celá přepracována a výrazně rozšířena. Jak je vidět na obrázku č. 4, horní část obsahuje data a nastavení, které je vidět z celé této části aplikace. Pro názornost je orámována zeleně. Spodní část obsahuje záložky, které zobrazují jednotlivá data nebo konfigurace, v obrázku je orámována červeně.





Obrázek 4: Zobrazení aktuálních dat

#### 5.4.1 Společná část

U horního okraje je tabulka se čtyřmi veličinami, které může uživatel vybrat v nastavení – toto je jediná část nastavení, která se oproti bakalářské práci změnila. Na výběr má proudy, napětí a harmonické zkreslení pro tři fáze a nulu. Činný, jalový, činný první harmonické, jalový první harmonické, deformační a zdánlivý výkon pro tři fáze a nulu, pro první čtyři jmenované i celkový výkon. Dále má na výběr přetečené, podtečené veličiny a nevyváženost proudu, napětí a úhlu.

Vlevo pod tabulkou jsou tlačítka pro přepínání mezi aktuálními daty a konfigurací. Vedle nich je posuvník, který nastavuje rychlost aktualizace dat a tlačítko pro okamžitou aktualizaci dat. Rychlost aktualizace dat je možné volit průběžnou, po 200, 400, 600, 800 ms, 1, 5, 10, 15, 20, 30, 45, 90, 120 sekund nebo pouze ruční pomocí tlačítka.

#### 5.4.2 Proudý a napětí

První záložka v datové části zobrazuje proudy, napětí a další související veličiny. Vše je zobrazeno v tabulce, která svým rozvržením kopíruje rozvržení aplikace ENVIS.Daq.[8]

Tato tabulka, stejně jako všechny v celé aplikaci, mění svoji velikost a velikost písma podle velikosti aplikace, takže se dokáže přizpůsobit různým obrazovkám a rozlišením.

Nulová hodnota se v tabulce vyskytuje ve dvou případech. Buďto je daná veličina skutečně nulová, nebo častěji přístroj, ke kterému zobrazená data přísluší, neumí danou veličinu měřit, neumí ji předat vybraným protokolem nebo nejsou zapojené potřebné vstupy.

	ULN[V]	ULL[V]	I[A]	uTHD[%]	iTHD[%]
1.	236.45	0.00	2.10	2.01	0.00
2.	236.27	0.00	2.10	2.01	0.00
3.	235.95	0.00	2.10	2.01	0.00
4.	0.00		0.00	0.00	0.00
Freq[Hz]	50.00			UnbU[%]	100.00
Analog	0.00			Unbl[%]	0.00
underfl.	U4 I4 I3 I2 I1	fiUnbl[°]			0.00
overfl.	OK				
IO	On: Led1				
Err	16				

Obrázek 5: Tabulka proudů a napětí

### 5.4.3 Výkony

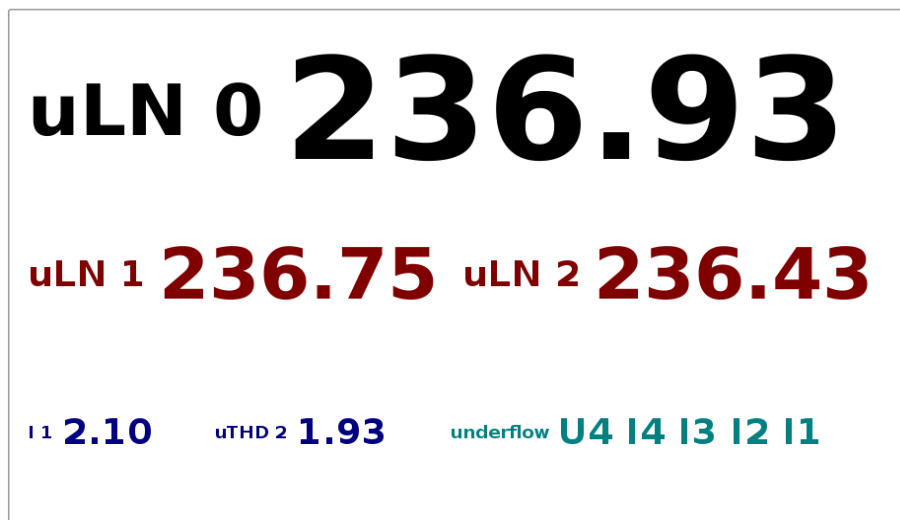
Výkonové veličiny jsou zobrazeny na další záložce. Opět jsou zobrazeny ve dvou tabulkách jako proudy a napětí. Aplikace dokáže zobrazit šest výkonů - činný, jalový, zdánlivý, deformační, činný první harmonické a jalový první harmonické. Pro všechny zobrazuje 3 fáze, výkon procházející nulou a výkon ve všech třech fázích.

### 5.4.4 Fázory

Další záložka zobrazuje fázory. Zde byly od bakalářské práce pouze zvýrazněny barvy. Změny proběhly v knihovně libactdata. Změnil se firmware měřicích přístrojů, které nyní umožňují načítání fázorových dat i pomocí protokolu Modbus, což bylo doplněno i do knihovny libactdata.

### 5.4.5 Uživatelsky definované hodnoty

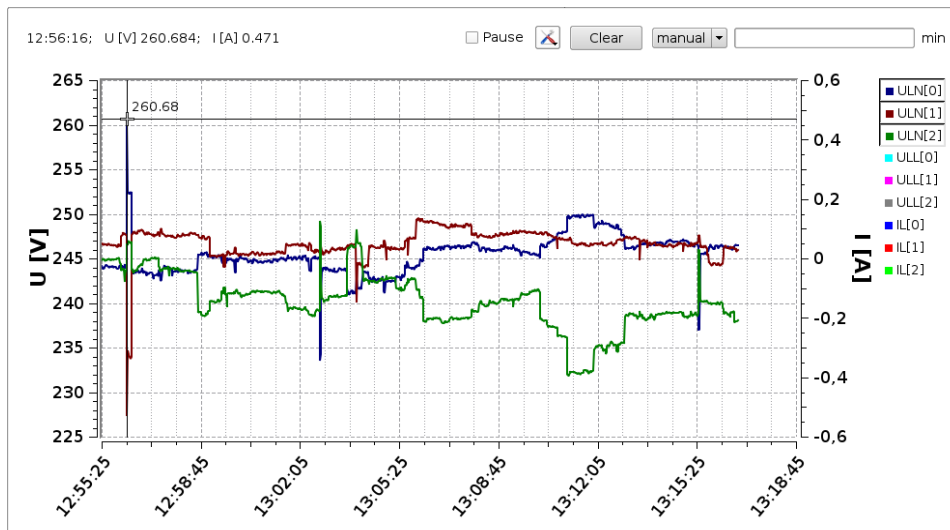
Na této záložce se zobrazují uživatelsky definované hodnoty. V konfiguračním souboru si uživatel nastaví popis veličiny, veličinu, barvu písma a velikost písma popisku a veličiny. Na výběr jsou stejné veličiny jako v tabulce čtyř veličin na horním okraji obrazovky.



Obrázek 6: Uživatelsky definovaná data

#### 5.4.6 Grafy

Další dvě záložky obsahují grafy napětí a proudu, respektive výkonu. Zobrazení grafu se dělí na tři části. Nad grafem je pruh, který obsahuje ovládací prvky, pod nimi vlevo je samotný graf a vpravo interaktivní legenda.



Obrázek 7: Graf s aktivním nástrojem „ukazovátko“

#### Ovládací prvky

Na pravé straně je textové pole, do kterého se dá vepsat časový interval v minutách, který má graf zobrazovat. Pokud není vyplněno nebo v něm je něco jiného než desetinné číslo,

zobrazují se data od posledního resetu nebo zapnutí aplikace. Vedle je výběrové pole, které obsahuje několik hodnot, které se po výběru vepíší do textového pole a funguje to stejně, jako by je tam vepsal uživatel. Dále napravo je tlačítko „Clear“, které vymaže veškeré dosud nasbírané hodnoty.

Dalším ovládacím prvkem je výběrové pole se symbolem nástrojů. To skrývá skupinu „Radio button“, kterou si uživatel volí nástroj. Na začátku je vybrané ukazovátko. To po najetí do grafu zobrazí vpravo od kurzoru aktuální hodnoty os Y. Po kliknutí se nalevo od ovládacích prvků zobrazí hodnoty všech os i s popisky veličin. Názorně je to vidět na obrázku č. 7.

Druhým nástrojem je lupa. Pokud je zastaveno zobrazování grafu, klasickým výběrem obdélníku označíme oblast, která se má zvětšit. Pokud není zastaveno, bude se takto chovat pouze do načtení další hodnoty – potom se zachová zvětšení v ose Y, ale osa X se automaticky nastaví podle aktuálních dat nebo zvoleného intervalu. Zrušení zoomu probíhá pomocí pravého tlačítka myši. Protože návrh počítá s dotykovým ovládním, kde se stisk pravého tlačítka špatně simuluje, je dalším nástrojem zrušení zoomu. Po jeho výběru a stisku pravého tlačítka v oblasti grafu dojde ke zrušení lupy.

Posledním nástroje je posun grafu. Po najetí do grafu a držení stisknutého levého tlačítka dochází k pohybu zobrazované oblasti. Opět funguje pouze při zastaveném zobrazování grafu nebo do načtení další hodnoty.

Posledním ovládacím prvkem je zaškrtačací políčko, které zastaví zobrazování grafu. Hodnoty se dále načítají ve zvoleném intervalu, je pouze pozastavené zobrazování. To je vhodné zejména pokud chce uživatel používat zoom pro zobrazení detailů v grafu.

## **Interaktivní legenda**

Legenda zobrazuje standardně barvu a popis dané křivky. Po kliknutí na položku legendy se tato křivka skryje nebo zobrazí v grafu a položka se zvýrazní. Při prvním zobrazení grafu se automaticky skryjí položky, které jsou nulové – tzn. ty, které měřicí přístroj pravděpodobně neměří.

## **Vlastní graf**

Graf podporuje zobrazení dvou os Y – levé a pravé. Osa X je pouze jedna – časová, ale v případě potřeby je možné zobrazit další osu na horním okraji. Osy se automaticky přizpůsobují zobrazovaným datům tak, aby byly co nejpodrobnější, ale zobrazovaly všechna data.

## **Graf napětí a proudů**

První graf v sekci aktuálních dat zobrazuje proudy a napětí, je vidět na obrázcích číslo 4 a 7.

## **Graf výkonů**

Druhý graf zobrazuje všechny měřené výkony – činný, jalový, zdánlivý, deformační, činný a jalový první harmonické.

### **5.4.7 Elektroměr**

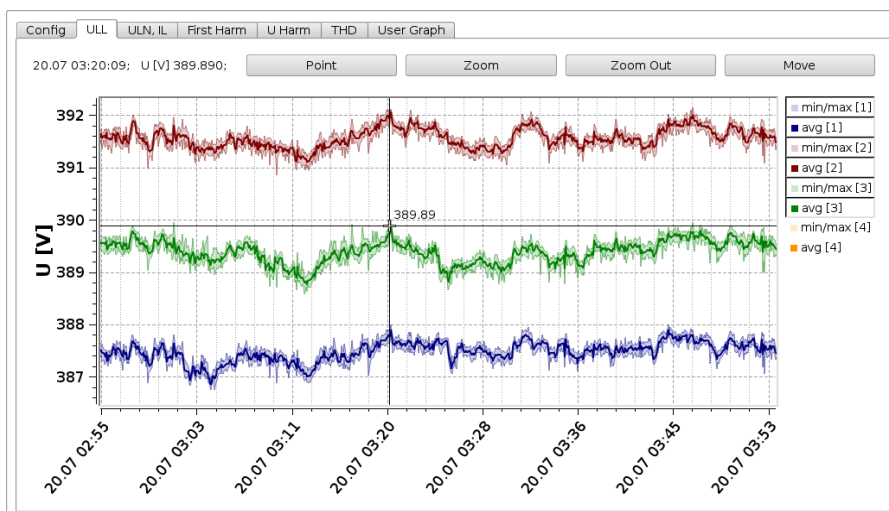
Poslední záložka obsahuje elektroměr, který obsahuje tři záložky – celkový součet, minulý a současný měsíc. V každé záložce jsou tři tabulky. Jedna pro výpis podle tarifů, druhá podle fází a třetí s maximy. První dvě obsahují spotřebovanou, vyrobenou, induktivní a kapacitativní energii.

## **5.5 Archivy**

Druhým typem zobrazení jsou archivní data. Tento typ zobrazení otevře vybraný archiv, rozbálí všechny archivní soubory do dočasného umístění a zobrazí vybraná archivní data. V současné době umožňuje zobrazit data pouze z hlavního archivu, ale knihovna libceaarchive, která byla vyvinuta pro čtení archivních dat, umí číst i archiv elektroměru.

Je připraveno 6 typů grafů a uživatel má možnost nastavit vlastní grafy s vybranými veličinami. Jak je vidět na obrázku číslo 8, připravené grafy jsou ULL, kombinovaný ULN s IL, graf prvních harmonických, proudových a napěťových harmonických a harmonického zkruslení.

Možnosti zobrazení dat jsou omezena zejména plánovanou velikostí a rozlišením obrazovky a dotykovým ovládáním. Cílem aplikace není suplovat software ENVIS od firmy KMB systems, který je určený pro analýzu archivních dat. Z těchto důvodů jsou implementovány pouze základní možnosti zobrazení.



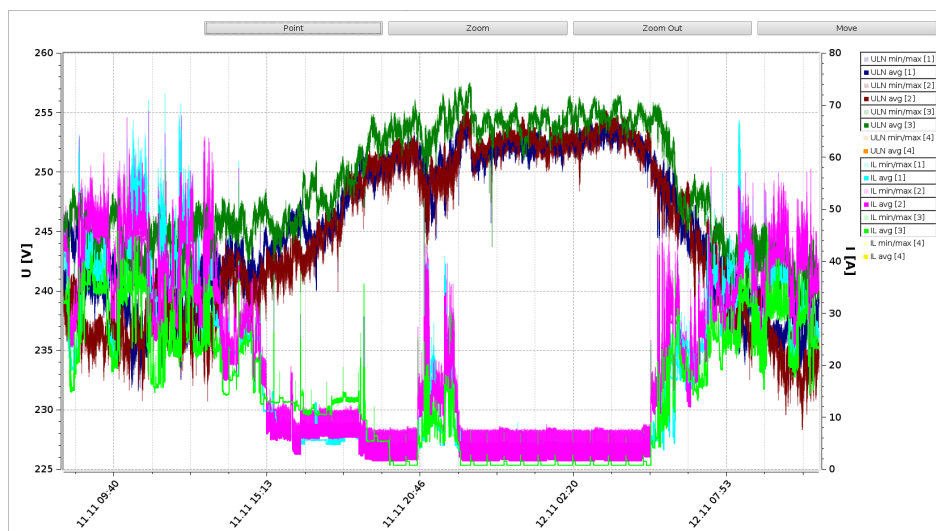
Obrázek 8: Archivní data, graf ULL.

V archivních datech se mohou ukládat tři druhy veličin - průměrné, maximální a minimální. Na obrázku číslo 8 je vidět použití minimálních a maximálních hodnot. V grafu se zobrazí jako plocha, vykreslená stejnou barvou jako průměrné hodnoty, pouze světlejší.

### 5.5.1 Grafy

Grafy vypadají velmi podobně jako ty použité pro zobrazování aktuálních dat. Na rozdíl od nich nemají ovládací prvky pro ovládání načítání dat a nástroje jsou zobrazeny jako tlačítka, ale mají totožnou funkci. Změnila se osa X, která nyní kromě času obsahuje i datum.

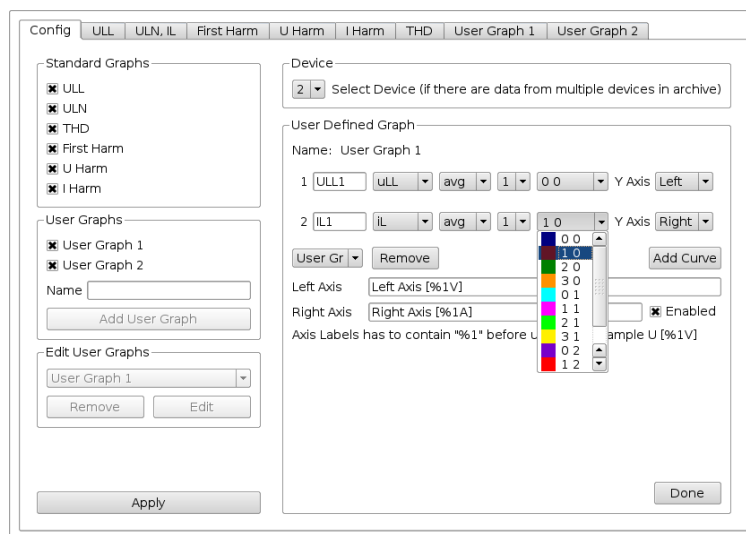
Jak je vidět na obrázku číslo 9, zobrazení dat z celého archivu najednou bývá nepřehledné, proto je nutné oproti aktuálním datům ve větší míře používat nástroj zoom, jako je to ukázáno na obrázku číslo 8.



Obrázek 9: Archivní data, kombinovaný graf ULN a IL.

### 5.5.2 Nastavení

Výběr zobrazených grafů, konkrétního zařízení z archivu a nastavení uživatelských grafů se provádí v záložce „Config“, jak je ukázáno na obrázku 10.



Obrázek 10: Nastavení zobrazení archivních dat.

Nastavení je rozděleno na tři části. Vlevo jsou zaškrťovací políčka pro výběr zobrazovaných grafů, možnost přidání uživatelských grafů a jejich smazání či editace. Tyto ovládací prvky jsou na obrázku zakázány, protože právě probíhá editace uživatelského grafu v pravé části.

Vpravo nahoře je sekce „Device“, kde se vybírá konkrétní zařízení, pokud jich archiv

obsahuje více.

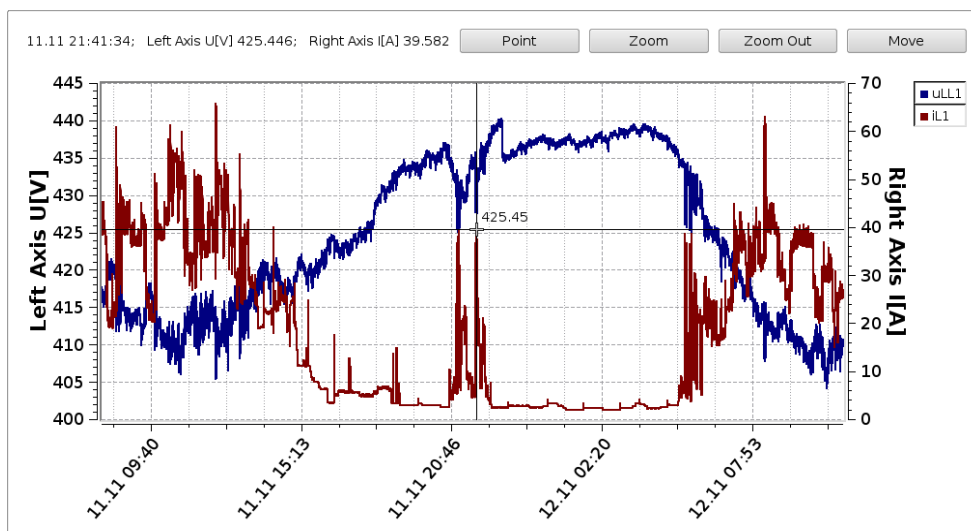
Většinu pravé části zabírá nastavení uživatelských grafů, které je dostupné po vybrání konkrétního grafu. Nahoře jsou zobrazeny jednotlivé křivky grafu a jejich nastavení. Křivky je možné vytvořit a smazat tlačítky níže. Dole je nastavení os grafu. Graf může obsahovat dvě osy Y – levou a pravou. Zde můžeme nastavit popisek osy a povolit pravou osu, která je na začátku vypnutá. Do popisků os je možné napsat libovolný text, ale je vhodné zařadit symbol „%1“. Při zobrazení grafu aplikace kontroluje, jestli nejsou zobrazované hodnoty příliš malé nebo velké, pokud zjistí, že je vhodné změnit jednotky za kilo, mili nebo podobné, vynásobí všechny zobrazované hodnoty příslušným koeficientem a symbol „%1“ nahradí příslušným multiplikátorem - m pro mili, k pro kilo atd.

K nastavení konkrétní křivky slouží vždy jeden řádek. V textovém poli se nastavuje popisek křivky. V prvním výběrovém poli se nastavuje veličina. K dispozici jsou napětí uLN a uLL, proud iL nebo harmonické zkreslení uTHD a iTHD. Ve druhém poli se vybírá typ dat – maximální, minimální a průměrné. V dalším poli se vybírá fáze, v dalším barva křivky a v posledním osa, ke které bude křivka přiřazena.

### 5.5.3 Uživatelské grafy

Uživatelské grafy jsou určeny k vybrání konkrétních sledovaných veličin a sledování vztahů mezi nimi. Na omezené velikosti displeje a při dotykovém ovládní může být zobrazení velkého množství dat problém, proto bylo umožněno uživatelům pro lepší přehlednost vybrat pouze ty veličiny, které je v konkrétní situaci zajímají. Jak je vidět na obrázku číslo 11, je toto rozbrazení přehlednější, než to na obrázku číslo 9. Samozřejmě je možné i v klasických grafech skrýt křivky, které nejsou důležité a tím zvýšit jejich přehlednost.

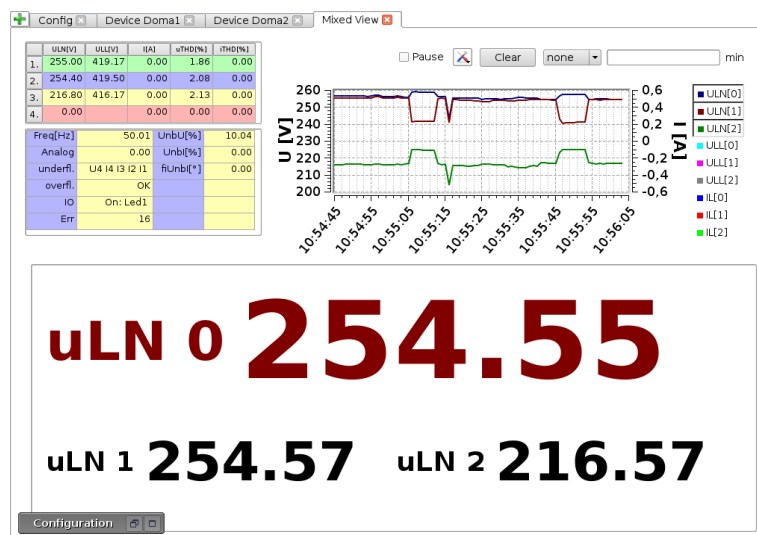




Obrázek 11: Archivní data, uživatelský graf.

## 5.6 Uživatelské rozvržení

V dalším typu záložky si uživatel může na obrazovku poskládat prvky zobrazující aktuální data z více přístrojů. Může nastavit jejich velikost, rychlost načítání dat, typ a přístroj, ze kterého načítají data. Tento typ záložky je připraven pro použití jako hlavní obrazovka aplikace, pokud bude použita například jako Human Machine Interface (HMI) na dotykovém Liquid Crystal Display (LCD) na rozvodné skříní.



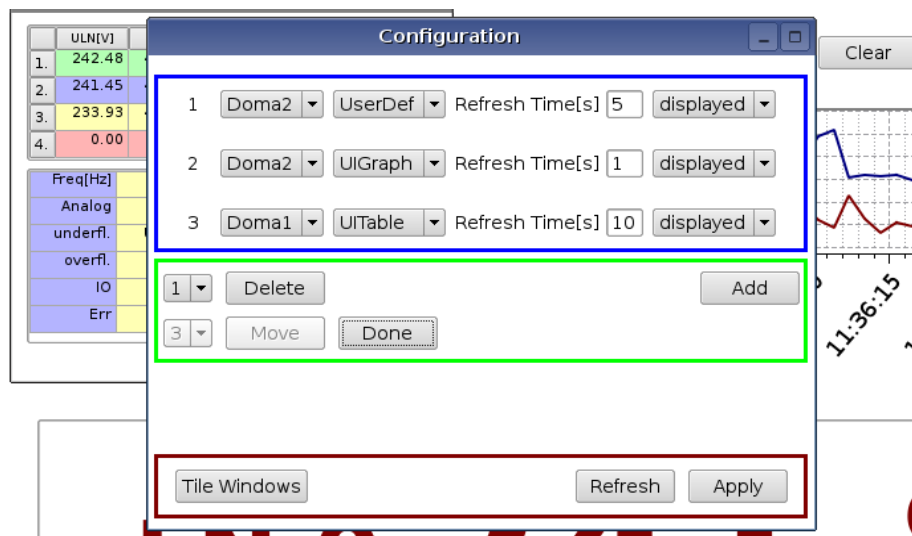
Obrázek 12: Ukázka záložky s uživatelským rozvržením.

Tento typ umí zobrazit všechny záložky dostupné v části aktuální data. Na rozdíl od nich se interval načítání dat nastavuje pro každý prvek zvlášť v nastavení. Jednotlivé prvky

tedy mohou mít různý interval aktualizace. Nevýhodou tohoto řešení je nutnost pro každý prvek otevřít nové spojení s přístrojem. Pokud tedy bude použito velké množství prvků s krátkým časem aktualizace, nebo dokonce průběžnou aktualizací, může dojít k zahlcení měřicího přístroje. K tomuto jevu dojde tím pravděpodobněji, čím je komunikace mezi aplikací a přístrojem pomalejší. Naopak pokud přístroje komunikují pouze na krátkou vzdálenost, tyto problémy se téměř nevyskytují. Je tedy nutné zvolit rychlost komunikace adekvátně ke konkrétnímu použití měřicích přístrojů a aplikace.

### 5.6.1 Nastavení

Vzhledem k povaze zobrazení je jeho konfigurace složitější. Provádí se v samostatném okně, které je vždy přítomno, ale běžně je minimalizované a nejde zavřít.



Obrázek 13: Nastavení uživatelského rozvržení.

Jak je vidět na obrázku číslo 13, konfigurace se skládá ze tří částí. Horní, orámovaná modře, obsahuje seznam prvků a jejich konkrétní nastavení – přístroj, se kterým komunikují, typ prvku, interval aktualizace dat a nastavení viditelnosti.

Druhá část, orámovaná zeleně, umožňuje smazat nebo posunout vybraný prvek. Při normálním zobrazení mají prvky skryté orámování a záhlaví, za které je možné je posunovat a měnit jejich velikost. K jejich zobrazení dojde až po zmáčknutí tlačítka „Move“, jak je to vidět na obrázku číslo 13. V pozadí je vidět i okno číslo 3 (s tabulkou proudů a napětí), které je právě upravováno. K opětovnému skrytí okrajů a záhlaví dojde po stisku tlačítka „Done“. Tlačítkem „Add“ se přidávají další prvky do horní části.

V poslední části, orámované červeně, je možné všechny prvky poskládat do dlaždic, aktualizovat nastavení nebo aplikovat změny a zapsat konfiguraci.

## 5.7 Nastavení aplikace

Poslední záložkou je „Config“, kde se nastavuje jazyk aplikace a jsou zde vypsány použité knihovny a jejich licence. Tato záložka může být pouze jedna. V budoucnosti se na tomto místě plánuje globální nastavení aplikace, které v současném stádiu vývoje není nutné nebo se nastavuje jednotlivě u konkrétních záložek.

## 5.8 Virtuální klávesnice

Protože primárně je aplikace určena pro dotykové ovládání, bylo nutné vytvořit dotykovou klávesnici. Framework Qt sice nenabízí přímo integrovanou klávesnici, ale její vytvoření a včlenění do aplikace je velice jednoduché a názorně předvedené v jedné z ukázkových aplikací distribuovaných společně s frameworkem.

Pro použití klávesnice není nutné dělat žádné změny v textových polích, klávesnice se pouze zaregistruje v aplikaci, k čemuž slouží kód v ukázce číslo 1. Klávesnice je potom implementovaná ve třídě `MyInputPanelContext` a `MyInputPanel`. První se stará o logickou část – zasílání signálů a odchyťávání událostí, druhá obsahuje samotnou klávesnici.

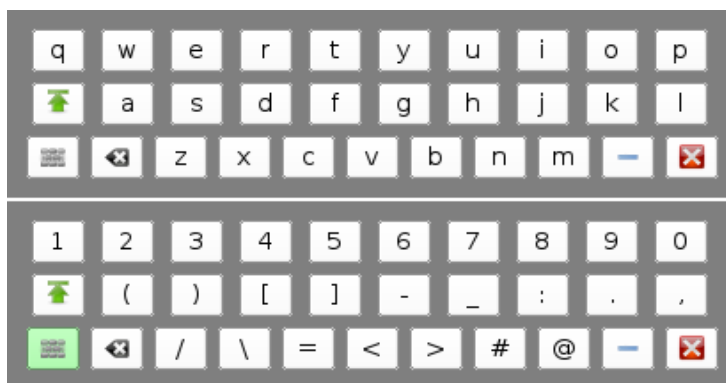
Zdrojový kód 1: Ukázka použití virtuální klávesnice

```
QApplication a(argc , argv , QApplication :: GuiServer );  
...  
MyInputPanelContext *ic = new MyInputPanelContext ;  
a . setInputContext ( ic );
```

Vzhledem k omezené velikosti displeje bylo nutné vyvinout klávesnici co nejmenší. Nutnost psát i speciální znaky a znaky s českou diakritikou byla vyřešena tlačítkem, které přepíná tři různé módy klávesnice a klávesou capslock. Na obrázku číslo 14 jsou vidět první dva módy. V prvním se píše písmena bez diakritiky, capslock přepíná mezi velkými a malými písmeny. Ve druhém módu se píše čísla a speciální znaky, capslock přepíná dvě sady speciálních znaků. Ve třetím módu se píše písmena s diakritikou, capslock přepíná velká a malá písmena.

Klávesnice obsahuje kromě proměnných znaků ještě pět speciálních kláves, které se nemění. Již zmíněný capslock a tlačítko pro přepínání módů. U nich je změnou barvy klávesy indikována aktivace capslocku nebo příslušného módu. Další speciální tlačítko s obrázkem šipky s křížkem je backspace a tlačítko s modrou vodorovnou čarou je mezerník.

Klávesnice se zobrazí po kliknutí do aktivního textového pole a skryje se tlačítkem s obrázkem křížku.



Obrázek 14: Virtuální klávesnice.

## 5.9 Překlad a podporované platformy

Vývoj všech komponent probíhá na linuxu x86/64, konkrétně distribuce ArchLinux. Průběžně je vše testováno na tom samém systému s embedded Qt a ve virtuálním stroji s debian x86 a embedded Qt.

### 5.9.1 Embedded Qt

Framework Qt umožňuje kompilaci pro takzvaný embedded Linux. Tímto termínem se označuje systém se standardním systémem GNU/Linux, ale bez X serveru, který se normálně stará o vykreslování grafiky.

Při kompilaci Qt ze zdrojových kódů je možné použitím konfigurační volby „-embedded“ zajistit, aby byla knihovna zkompileovaná pro tuto platformu. Aplikace zkompileované s embedded Qt nejdou spustit v prostředí X serveru. Místo toho se spouští přímo z konzole a o vykreslování grafiky se stará samotné Qt a grafika se vykresluje přímo do framebufferu.

Neexistují předkompilované balíčky s embedded Qt, pro každou platformu je tedy nutné kompilovat embedded Qt znovu, což je, zvláště na pomalejších strojích, velmi zdlouhavé a trvá několik hodin. Pro jiné platformy než x86 je možné použít křížovou kompilaci.

## 5.9.2 ARM procesory

Původně bylo v plánu otestovat aplikaci na reálném zařízení s ARM procesorem. Toto zařízení se bohužel nepodařilo zajistit, proto probíhalo testování pouze v softwaru Quick EMUlator (QEMU), který umí emulovat i ARM procesory.

### QEMU

Tento software může fungovat jako emulátor procesoru nebo klasický virtualizační software. Pro testování je využito jeho schopnosti emulovat ARM procesory a je použit ke spuštění celého operačního systému. [11]

Pro platformu ARM není dostupný zavaděč, proto je nutné jádro a initial RAM disk (initrd) instalátoru nastavit pomocí parametrů qemu. Po instalaci a zkopírování jádra a initrd hotového systému se již emulovaný systém spouští příkazem z ukázky číslo 2, kde parametr „-M“ udává typ emulovaného procesoru, „-m“ dostupnou Random-access memory (RAM), „-kernel“ jádro, „-initrd“ initrd, „-hda“ obraz disku a „-append“ parametry jádra.

Zdrojový kód 2: Příkaz pro spuštění emulovaného systému

```
qemu-system-arm -M versatilepb -m 512 -kernel \
  vmlinuz-2.6.32-5-versatile -initrd initrd.img-2.6.32-5-versatile
-hda debian.img -append "root=/dev/sda1"
```

Pro testování byla použita linuxová distribuce Debian Squeeze, což je stabilní verze, která je dostupná pro mnoho platforem včetně EABI ARM. [12]

### Křížová kompilace

Pro procesor ARM je nutné ručně zkompilevat framework Qt, knihovnu QWT a další systémové knihovny využité v aplikaci a vytvořených knihovnách. Kompilovat vše v emulovaném systému by bylo jednodušší, ale časově velmi náročné, protože systém je pomalý. Navíc při reálném nasazení na ARM procesoru je kompilace na koncovém zařízení často nemožná. Z toho důvodu je nutné využít křížovou kompilaci. Při tomto způsobu kompilace se na hostitelském systému generuje spustitelný kód pro cílovou platformu. V tomto případě byla hostitelská platforma GNU/Linux x86\_64 a cílová GNU/Linux ARM. Jako kompilátor byl

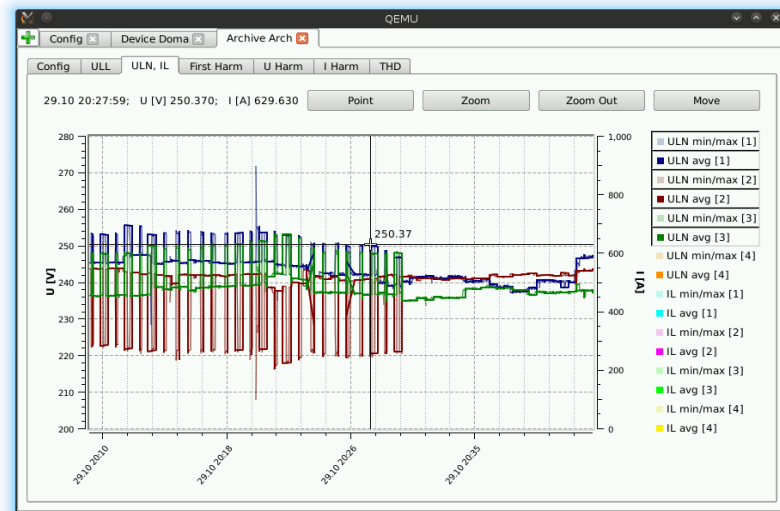
zvolen GNU C a C++ obsažený ve vývojovém prostředí Sourcery CodeBench Lite Edition od firmy Mentor Graphics. [10]

Vývojové prostředí obsahuje pouze základní systémové knihovny, proto je nutné ručně zkompileovat Qt, QWT, zlib, zziplib, openssl, libmodbus, libxml2 a výsledné hlavičkové soubory a knihovny umístit do adresářů vývojového prostředí.

Kompilace aplikace a vytvořených knihoven probíhá v prostředí Qt Creatoru ve kterém nastavíme cestu ke křížovému kompilátoru a knihovně Qt zkompileované pro požadovanou platformu. V souborech s nastavením projektů již většinou není třeba nic měnit, u aplikace je nutné změnit pouze cesty ke knihovnám. Příklad takového souboru je v ukázce číslo 2.

Před spuštěním v emulovaném operačním systému je nutné zkompileované knihovny a aplikaci zkopírovat do virtuálního disku, nastavit systémovou proměnnou, určující cestu ke knihovnám, a povolit čtení ze souborů označujících zařízení myši a klávesnice – „/dev/input/mice“ a „/dev/tty0“.

Spuštěná aplikace vypadá téměř totožně jako na testované platformě, jak je vidět na obrázku číslo 15.



Obrázek 15: Apliace v emulované systému na ARM platformě

### 5.9.3 Windows

Největším problémem při portaci aplikace na Windows činí knihovny. Jelikož je primární platformou Embedded linux, byl tomu přizpůsoben i výběr knihoven, až později bylo rozhodnuto vytvořit i port pro Windows, což přineslo určité problémy. Většina knihoven není standardně dostupná pro tuto platformu a musí se ručně kompilovat, některé knihovny musely být na-

hrazeny jejich multiplatformní alternativou.

## MinGW

Protože většina použitých knihoven potřebuje při kompilaci program make a configure script, byl použit MinGW [9], což je minimalistické vývojové prostředí, které usnadňuje portaci aplikací z prostředí GNU/Linuxu do Windows. Obsahuje Bourne Shell interpreter příkazového řádku, kompilátor GCC, make a další potřebné utility. Qt pro Windows existuje ve třech základních verzích - pro Visual Studio 2008, 2010 a MinGW.

Instalační balík pro MinGW obsahuje i příkazový řádek s nastavenými systémovými proměnnými, který je vhodný pro kompilaci druhé největší použité knihovny – QWT. Další potřebné knihovny je již vhodnější kompilovat v příkazovém řádku z balíku MinGW, který používá bash.

Knihovny libacdata, libarchive a libxmlrw byly původně vyvíjeny v prostředí NetBeans a používaly makefile generovaný tímto prostředím. Pro portaci se ukázala jako vhodnější volba použít QtCreator i pro tyto knihovny i když nepoužívají žádnou část Qt frameworku. Pro kompilaci na Windows není nutné nic nastavovat, pouze pomocí podmínek v konfiguračním souboru projektu upravit cesty pro hlavičkové soubory a knihovny. Konfigurační soubor pro knihovnu Libceaarchive je v příloze B. Obdobné změny je třeba udělat i v konfiguračním souboru samotné aplikace.

V současné době tedy aplikace jde spustit v prostředí Windows, ale není příliš stabilní, proto není vhodná k produkčnímu nasazení. Odstranění chyb způsobujících nestabilitu a odladění aplikace pro platformu Windows by zabralo hodně času. Jelikož tento port není prioritou, tak v současném stádiu vývoje aplikace dostačuje možnost běhu na této platformě. Ve Window funguje všechno stejně jako v linuxové verzi a díky použitému frameworku Qt jsou odchylky ve vzhledu aplikace pouze minimální, jak je vidět na obrázku číslo 16.



Obrázek 16: Port aplikace na Windows

#### 5.9.4 Android

Firma Digia, která v současné době stojí za vývojem frameworku Qt, si jako hlavní cíl pro příští vývoj klade portaci a rozšíření frameworku na mobilních platformách, především Android a iOS. Pro aplikaci to přináší do budoucna příslib relativně jednoduché portace na tyto platformy.

Největším problémem opět nebude samotná aplikace a vyvinuté knihovny, ale použité systémové knihovny. Při vývoji bylo na tento problém myšleno, proto je použití dalších systémových knihoven omezeno na minimum a pouze pro nejnútnejší, většinou ne klíčové funkce. Případné použití více knihoven, poskytujících podobnou funkcionalitu na různých platformách, nebude znamenat velké změny ve vytvořené aplikaci a knihovnách.

#### 5.10 Překlad jazyka aplikace

Aplikace je psána primárně v angličtině a do češtiny je následně překládána. Knihovna Qt má pro překlad připravené kvalitní nástroje. S jejich pomocí je možné řešit i problém množných čísel v řetězcích, obsahující proměnný integer a text, který na něj reaguje.

Všechny textové řetězce, které se mají přeložit, jsou ve zdrojovém kódu obaleny funkcí „tr()“ nebo nějakou její alternativou. Přidání popisku do okna potom vypadá následovně.



### Zdrojový kód 3: Ukázka přidání textového popisku připraveného pro překlad

```
connectionLabel->setText ( tr ( " Connection _ type : " ) );
```

V nastavení Qt projektu je přidán řádek s informací o existenci překladu – „TRANSLATIONS = viewer\_cs\_CZ.ts“. Pro vlastní překlad je nejprve nutné spustit program „lupdate“, který je součástí frameworku Qt. Tím se vygenerují soubory s řetězci pro překlad, specifikované v konfiguračním souboru – v tomto případě „viewer\_cs\_CZ.ts“.

Samotný překlad se provádí v grafické aplikaci Qt Linguist. V té se po otevření souboru s textovými řetězci pro překlad objeví jejich seznam a informace o stavu jejich přeložení. Pokud je program překládán do více jazyků, je možné vidět všechny překlady najednou. Po dokončení a uložení úprav se programem „lrelease“ vygeneruje výsledný soubor s překladem, v tomto případě „viewer\_cs\_CZ.qm“, který je nutné přidat do konfiguračního souboru projektu.

Výběr jazyka aplikace se provádí ve funkci „main“ grafického programu. V ukázce zdrojového kódu č. 4 je vidět část funkce main. Výběr jazyka aplikace je zde proveden na základě nastavení, které je uloženo ve skupině „Global“ ve formátu „cs\_CZ“ pro češtinu a „en“ pro angličtinu. Metoda „translator.load“ se pokusí načíst soubor s překladem, pokud se mu to nepodaří, použije se základní jazyk aplikace, v tomto případě angličtina. V aplikaci jsou místo přímého načítání použity takzvané „resources“ soubory, které jsou flexibilnější.

### Zdrojový kód 4: Ukázka výběru jazyka aplikace

```
QApplication a(argc , argv , QApplication :: GuiServer );  
...  
QSettings s ;  
QTranslator translator ;  
s . beginGroup ( " Global " ) ;  
translator . load ( " viewer _ " + s . value ( " language " ) . toString ( ) ) ;  
s . endGroup ( ) ;  
a . installTranslator ( & translator ) ;
```

## 5.11 Konfigurace aplikace

Veškeré nastavení aplikace se provádí v jednom konfiguračním souboru, protože je při vývoji výhodnější mít všechna nastavení pohromadě. Do budoucna se počítá s rozdělením konfiguračních souborů podle záložek, ke kterým náleží. Ukázka konfiguračního souboru je v příloze B. Soubor používá standardní syntaxi pro ukládání nastavení frameworku Qt. Kvůli multiplatformnosti nezáleží na velikosti písmen, podporuje skupiny a pole.

Soubor je rozdělen na několik skupin a vždy obsahuje alespoň dvě – „Default“ a „Global“. V globální skupině jsou uložena nastavení pro celou aplikaci – jazyk aplikace, jména a typy záložek, ve skupině „Default“ jsou uloženy základní hodnoty, které se ukáží při vytváření nového připojení k zařízení nebo nového archivu.

Soubor může dále obsahovat několik skupin, které jsou pojmenované podle záložky, ke které náleží. Jejich obsah se liší podle toho, jestli jde o záložku s aktuálními, archivními nebo smíšenými daty. Nastavení aktuálních dat obsahují typ a informace potřebné ke spojení se zařízením nebo cesty k XML souborům a nastavení uživatelsky definovaného zobrazení. Konfigurace archivních dat obsahuje grafy, které se mají zobrazit, pokud jsou nastaveny uživatelské grafy, je zde uloženo nastavení i pro ně. Nastavení pro smíšená data obsahuje pro každý zobrazený widget jeho typ, interval načítání, zařízení ze kterého se mají číst data, jeho viditelnost, pozici a velikost.

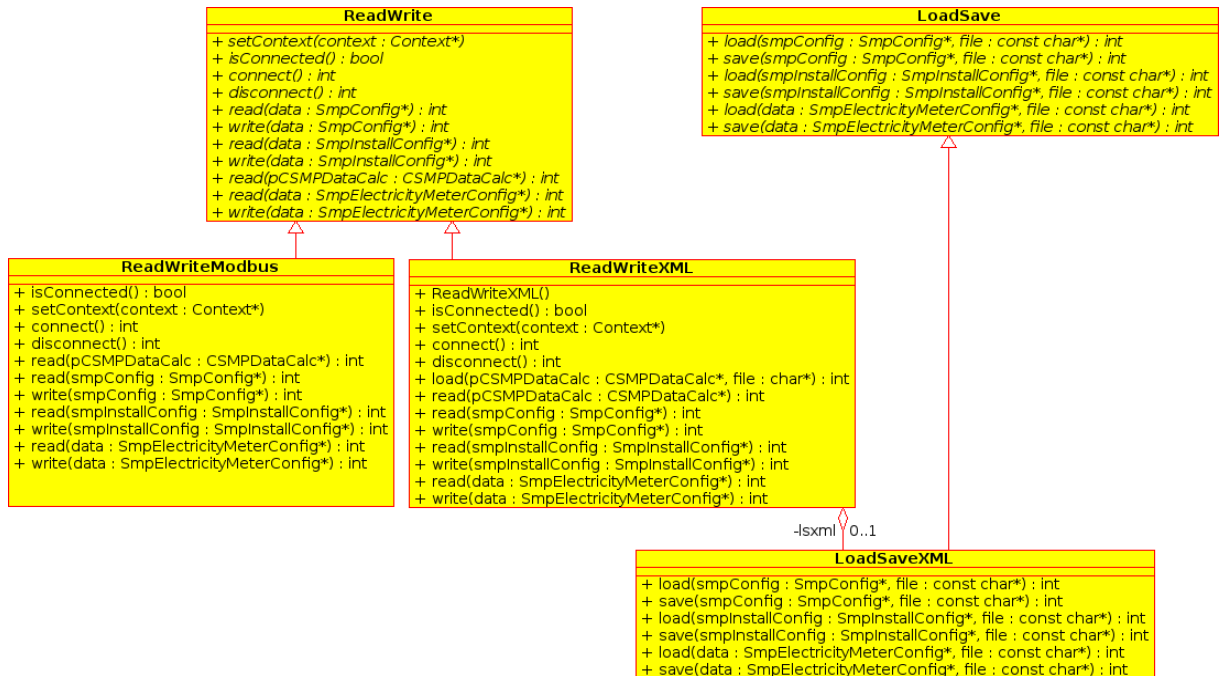
## 6 Knihovny

### 6.1 Libactdata

Knihovna Libactdata, vznikla v rámci mé bakalářské práce a v diplomové práci byla rozšířena o podporu čtení fázorů pomocí protokolu MODBUS a bylo změněno API, aby bylo lépe použitelné ve změněné grafické aplikaci. Nutnost změny si vyžádalo především přidání smíšeného zobrazení, kdy je na jedné obrazovce několik widgetů, které načítají data z několika přístrojů najednou. Druhou výraznou změnou v aplikaci bylo přesunutí načítání aktuálních dat o úroveň výše tak, aby všechny tabulky a grafy používaly jednoho spojení a ne každá své, jako tomu bylo v původním návrhu.

Knihovna je určená pro načítání aktuálních dat z měřicího přístroje. Umí komunikovat pomocí protokolu MODBUS po TCP/IP a RS485. Z historických důvodů umí načítat a ukládat i data z XML souborů.

Knihovna je podrobně popsána v mé bakalářské práci a v příloze na CD je podrobný manuál, proto zde uvedu pouze základní Unified Modeling Language (UML) diagram, příklad použití s novým API je v příloze B. [7]



Obrázek 17: UML diagram knihovny Libactdata.

Jak je vidět ve zdrojovém kódu v příloze B, pro získání dat pomocí této knihovny je nutné nejdřív naplnit třídu „Context“ daty nutnými pro připojení. To se provádí metodami „setConnType“, „setModbusIP“ a podobně. Potom se vytvoří instance třídy „ReadWriteFactory“ a metodě „chooseRWType“ předáme naplněný „context“ jako parametr, návratovou hodnotou bude instance některé konkrétní implementace virtuální metody „ReadWrite“. Jaká konkrétní implementace bude vybrána záleží na obsahu třídy „Context“.

Vlastní komunikační API tvoří metody virtuální třídy „ReadWrite“. Spojení je nutné zahájit metodou „connect“ a ukončit metodou „disconnect“. Vlastní data se načítají přetíženou metodou „read“, která jako parametr přijímá datové struktury a podle typu struktury ji naplní patřičnými daty. Obdobně pracuje přetížená metoda „write“.

## 6.2 Libceaarchive

Druhou knihovnou, která vznikla celá v rámci diplomové práce, byla libceaarchive. Tato knihovna slouží pro čtení a částečně i zápis archivních souborů, které byly popsány v kapitole 2.3. Z uvedených typů archivů je implementováno čtení archivu elektroměru a hlavního,

částečně potom zápis archivu elektroměru. Čtení je použito v grafické aplikaci, zápis je pouze v knihovně, ale bylo rozhodnuto, že grafický klient v současné fázi nebude podporovat zápis archivů.

Knihovna je napsaná v čistém C++, bez frameworku Qt, protože je možné, že bude využita i v jiných aplikacích v rámci firmy KMB systems. Knihovna rozbalí archiv do dočasného úložiště, vybere požadovaný archivní soubor, načte požadovaná data do paměti a smaže dočasné soubory. Pro rozbalování používá knihovnu zziplib, která vnitřně používá standardní knihovnu zlib. Bylo by možné použít zlib přímo, což by nabízelo více možností, ale implementace by byla výrazně složitější. [6]



Obrázek 18: UML diagram knihovny Libcearchive.

UML diagram knihovny je vidět na obrázku číslo 18, příklad jejího začlenění do programu je v příloze B.

Použití je jednoduché. Rozhraní knihovny tvoří pouze veřejné metody třídy ReadWriteArchive. V konstruktoru třídy se předá cesta k souboru s archivem. V metodě „setDebug“ je možné nastavit množství vypisovaných ladících informací, které jsou defaultně vypnuty. Dále se metodou „openArchive“ otevře archiv a rozbalí do dočasného úložiště. Metodou „readMainArchive“ se z rozbalených souborů vybere hlavní archiv, pokud je jich zde více, vybere se ten s číslem předaným v parametru nRec. Dalšími parametry jsou struktury, kam se mají načíst data a konfigurace z archivu. Pokud metoda vrátí 0, vše proběhlo v pořádku a je možné zobrazit načtená data. Pokud je návratová hodnota jiná, indikuje to chybu při načítání dat.

Struktury pro data a konfiguraci se alokují ve funkci „readMainArchive“. Datová struktura se alokuje až podle toho, co je načteno v konfiguraci pomocí neveřejné funkce „allocateMainData“. Po skončení práce s daty je nutné smazat dočasně rozbalené soubory metodou „closeArchive“ a pomocí veřejné metody „freeMainData“ uvolnit paměť po datové struktuře. Struktura pro konfiguraci se uvolňuje standardě pomocí funkce „free“.

Kompletní manuál vygenerovaný pomocí programu Doxygen je na přiloženém CD.

## 7 Závěr

Výsledkem práce je grafický klient schopný zobrazovat aktuální i archivní data, který je schopný běžet na několika platformách. Primárně je určen pro použití na embedded zařízeních s ARM procesorem a operačním systémem GNU/Linux bez X serveru, ale je schopný běžet i na počítačích s běžnými linuxovými distribucemi. Je možné ho spustit i na počítačích s Windows, kde ale jeho vývoj nebyl prioritou, proto se zde chová nestabilně.

Framework Qt, na kterém je aplikace postavena, je dostupný pro velké množství platform. Případnou portaci na tyto platformy tak bude brzdit pouze nedostupnost dalších použitých knihoven. Z důvodů minimalizace takovýchto problémů byl kladen důraz na co nejmenší používání dalších knihoven, proto by nemělo být rozšíření podporovaných platform příliš složité.

Rozhraní klienta bylo optimalizováno pro dotykové ovládání ve ztížených podmínkách, například v rozvodných skříních. Při návrhu se také kladl důraz na kontinuitu ovládání napříč aplikacemi firmy KMB systems, proto se vzhled klienta podobá aplikaci ENVIS.Daq, která slouží k podobným účelům ve Windows.

Dále byla vyvinuta knihovna libceaarchive pro čtení a zápis archivů cea, které používají měřicí přístroje firmy KMB systems a analyzační software ENVIS. Knihovna libactdata, která slouží ke komunikaci s měřicími přístroji, byla rozšířena a bylo přepracováno její rozhraní. Pro obě knihovny byly vytvořeny manuály popisující jejich strukturu a použití.

Pokud bude zájem aplikaci dovést do stádia, kdy by se dala komerčně využít, bude pravděpodobně nutné doplnit čtení aktuálních dat pomocí protokolu KMB Long nebo rozšířením firmwaru měřicích přístrojů umožnit čtení více veličin pomocí protokolu Modbus. Konkrétně se jedná o harmonické složky, které aplikace umí zobrazovat pouze z archivů. Pokud se bude po aplikaci požadovat i zápis archivních dat, bude nutné rozšířit podporu zápisu v knihovně libceaarchive.

## Seznam použité literatury

- [1] Modbus Specification. *Modbus Application Protocol Specification*. V1.1b. Modbus-IDA, 2006. Dostupné z: [http://www.modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b.pdf](http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf)
- [2] Communication Protocol Manual. *SMV, SMP, SMPQ Multifunctional Panel Meters & Power Quality Analyzers Communication Protocol Manual*. 2. vyd. Liberec: KMB Systems, 2009. Dostupné z: <http://kmb.cz/index.php/cs/ke-stazeni/category/5-merici-pristroje?download=57:popis-komunikace-pro-smv-smp-a-smpq-eng>
- [3] DIGIA. *Qt* [online]. Helsinki Finland, ©2012 [cit. 2012-11-18]. Dostupné z: <http://qt.digia.com/>
- [4] NOKIA. *Online Reference Documentation* [online]. ©2011 [cit. 2012-11-18]. Dostupné z: <http://doc.qt.digia.com/>
- [5] *Qwt User's Guide* [online]. 6.0.1. [2011] [cit. 2012-11-19]. Dostupné z: <http://qwt.sourceforge.net/>
- [6] DRAHEIM, Guido. *ZZiplib: ZIP File Access* [online]. 2009-08-23 [cit. 2012-12-06]. Dostupné z: <http://zziplib.sourceforge.net/zzip-file.html>
- [7] BEDRNÍK, Tomáš. *Klient pro zobrazení dat virtuálního měřicího přístroje v prostředí GNU/Linux*. Liberec, 2010-06-23. Bakalářská práce. Technická univerzita v Liberci, Fakulta mechatroniky, Ústav mechatroniky a technické informatiky. Vedoucí práce Jan Kraus.
- [8] KMB SYSTEMS, s.r.o. *Software ENVIS: Uživatelská příručka pro podporované měřicí přístroje Verze 1.1*. Liberec, 2012. Dostupné z: <http://kmb.cz/index.php/cs/ke-stazeni/category/2-software?download=197:envis-v11-uzivatelska-prirucka>
- [9] *MinGW: Minimalistic GNU for Windows* [online]. May 25th, 2012 [cit. 2012-12-09]. Dostupné z: <http://www.mingw.org/>
- [10] Sourcery CodeBench Lite Edition. MENTOR GRAPHICS. *The EDA Technology Leader* [online]. [2011] [cit. 2012-12-11]. Dostupné z: <http://www.mentor.com/embedded-software/sourcery-tools/sourcery-codebench/editions/lite-edition/>

- [11] *QEMU: open source processor emulator* [online]. 3 December 2012 [cit. 2012-12-11]. Dostupné z: <http://wiki.qemu.org/>
- [12] Debian: Debian "squeeze"Release Information. *Debian* [online]. Oct 21 2012 [cit. 2012-12-11]. Dostupné z: <http://www.debian.org/releases/stable/index.en.html>
- [13] BK-Touch Zobrazení on-line měřených dat (2. část). *ELCOM, a.s.* [online]. 2010 [cit. 2012-09-29]. Dostupné z: [http://www.elcom.cz/touch/bk\\_touch\\_02.html](http://www.elcom.cz/touch/bk_touch_02.html)
- [14] ELCOM, a.s. - Divize Virtuální instrumentace. *BK-ELCOM Komplexní systém monitoringu a analýzy kvality elektřiny: Uživatelský manuál BK-Touch Aplikace pro ovládání analyzátoru Verze 2.8.0.* Ostrava - Pustkovec, 2009. Dostupné z: <http://www.elcom.cz/files.php?id=/4/download/bk-elcom/BK-TOUCH.pdf>
- [15] KMB SYSTEMS, s.r.o. *Software ENVIS: Uživatelská příručka pro podporované měřicí přístroje Verze 1.1.* Liberec, 2012. Dostupné z: <http://kmb.cz/index.php/cs/ke-stazeni/category/2-software?download=197:envis-v11-uzivatelska-prirucka>
- [16] POWER STANDARDS LAB. *PQube(R) Installation & Users's Manual: Firmware Rev 2.1.* Alameda USA, 2012, 113 s. Dostupné z: <http://www.powerstandards.com/Download/PQube%20Manual%202.1.pdf>



# Příloha A - Porovnání grafického rozhraní konkurenčních přístrojů

## Obsah

A.1	Úvod . . . . .	50
A.2	Aktuální data . . . . .	51
A.2.1	ENVIS.Daq . . . . .	52
A.2.2	PQube . . . . .	52
A.2.3	BK-Touch . . . . .	52
A.2.4	Návrh realizace . . . . .	53
A.3	Elektroměr . . . . .	54
A.3.1	KMB DAQ . . . . .	54
A.3.2	BK-Touch . . . . .	54
A.3.3	PQube . . . . .	54
A.3.4	Návrh realizace . . . . .	54
A.4	Fázorový diagram . . . . .	55
A.4.1	Porovnání . . . . .	55
A.4.2	Návrh realizace . . . . .	55
A.5	Zobrazení harmonických . . . . .	57
A.5.1	Porovnání . . . . .	57
A.5.2	Návrh realizace . . . . .	57
A.6	Historická data . . . . .	58
A.6.1	ENVIS.DAQ . . . . .	59
A.6.2	BK-Touch . . . . .	59
A.6.3	PQube . . . . .	59
A.6.4	ION . . . . .	60
A.6.5	Návrh realizace . . . . .	60
A.6.6	Histogram . . . . .	60
A.7	Celkový vzhled aplikace . . . . .	61
A.8	Závěrečná doporučení . . . . .	62
	<b>Seznam použité literatury</b>	<b>62</b>

## A.1 Úvod

Dokument má za cíl porovnat několik konkurenčních uživatelských rozhraní s řešením firmy KMB systems – programem ENVIS.Daq, ze kterého vycházela i původní verze grafického klienta, která vznikla v rámci mé bakalářské práce. Cílem je navrhnout strukturu zobrazování dat v klientovi tak, aby umožňoval zobrazovat všechny potřebné údaje, přitom byl přehledný a jednoduše ovladatelný na malých dotykových obrazovkách i větších monitorech.

## A.2 Aktuální data

Meter	Value	
L1-N	240.8V	
L2-N	242.1V	
L3-N	241.0V	
N-E	0.6V	
L1-L2	418.6V	
L2-L3	418.3V	
L3-L1	417.2V	
L1 Amp	16.04A	
L2 Amp	0.74A	
L3 Amp	1.69A	
Frequency	49.935Hz	
Voltage THD	Max	1.60%
	L1	1.60%
	L2	1.50%
	L3	1.60%
Current TDD	Max	2.30%
	L1	2.30%
	L2	0.50%
	L3	0.60%
L1 Flicker	$P_{inst}$	0.00
	$P_{ST}$	1.35
	$P_{LT}$	0.04
L2 Flicker	$P_{inst}$	0.00
	$P_{ST}$	2.01
	$P_{LT}$	0.05
L3 Flicker	$P_{inst}$	0.00
	$P_{ST}$	0.26
	$P_{LT}$	0.04
IEC Zero Sequence V	0.10%	
IEC Negative Sequence V	0.20%	
IEC Zero Sequence I	89.90%	
IEC Negative Sequence I	82.50%	
L1-N Voltage Fundamental	240.8V	0deg
L2-N Voltage Fundamental	242.2V	-120deg
L3-N Voltage Fundamental	241.0V	120deg
L1 Current Fundamental	15.84A	-2deg
L2 Current Fundamental	0.49A	-41deg
L3 Current Fundamental	1.61A	109deg
User Counter (since 2012/05/17)		0

Obrázek 1: Aktuální data – PQube

	uLL[V]	uLN[V]	I[ma]	uTHD[%]	iTHD[%]
1	0	239,0	20,80	2,440	145,8
2	0	238,8	20,60	2,440	149,5
3	0	238,5	20,40	2,440	149,7
4		0	0	0	0
f[Hz]	50,00		UnbU[%]	0	
Temp	52,40		UnbI[%]	0	
podtečení	U4,I4		fiUnbI[°]	0	
přetečení	OK				
IO	ON: LED1				
Chyba	(0)				

Obrázek 2: Aktuální data – DAQ

	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
U RMS [V]	23,05k	23,21k	22,79k	6,715k	23,02k
I RMS [A]	242,6	254,8	253,6	12,89	250,4
P [W]	5,499M	5,553M	5,710M	85,27k	16,76M
Q [VAR]	1,011M	-2,037M	-897,2k	-14,90k	-4,224M
S [VA]	5,591M	5,915M	5,780M	86,56k	17,29M
Cos [-]	0,9845	0,9402	0,9963	0,9851	0,9733
f [Hz]	50,01				

Obrázek 3: Základní aktuální data – BK-Touch

	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
Avg [V]	22,22k	22,42k	22,55k	6,879k	22,40k
U RMS	22,22k	22,42k	22,55k	6,879k	22,40k
Avg [%]	5,994	4,975	5,112	3,328	5,379
THD U	5,994	4,975	5,112	3,328	5,379
Avg [V]	22,18k	22,39k	22,52k	6,875k	---
U h1	22,18k	22,39k	22,52k	6,875k	---
Avg [Rad]	Fáze L1	Fáze L2	Fáze L3	Fáze N	---
Phi U h1	-1,571	2,615	0,5174	-1,057	---
Min [V]	Fáze L1	Fáze L2	Fáze L3	Fáze N	---
U RMS pp	22,13k	22,37k	22,47k	6,879k	---
Max [V]	Fáze L1	Fáze L2	Fáze L3	Fáze N	---
U RMS pp	22,22k	22,42k	22,55k	6,882k	---
Avg [-]	Fáze L1	Fáze L2	Fáze L3		---
PST	0,8205	0,8519	0,8107	---	---
Avg [-]	Fáze L1	Fáze L2	Fáze L3		---
PLT	5,616	3,489	7,356	---	---
Avg [%]					---
Nesym. U	---	---	---	---	0,5045
Avg [Hz]					Celkově
f	---	---	---	---	50,00

Obrázek 4: Volitelná Aktuální data – BK-Touch

### A.2.1 ENVIS.Daq

Jako hlavní aplikace pro porovnávání slouží ENVIS.DAQ od firmy KMB.

Měřené veličiny jsou uspořádány do čtyř tabulek ve dvou záložkách. Obnova dat je skokově nastavitelná od 200 ms do 90 s nebo ručně. Není možné vybírat, které veličiny budou zobrazovány. Pokud přístroj některou z nich neměří, zobrazí 0. Pomocí přepínačů je možné skrývat a zobrazovat jednotlivé fáze.

První tabulka zobrazuje proudy, napětí a několik dalších veličin:  $u_{LL}$ ,  $u_{LN}$ ,  $I$ ,  $u_{THD}$  (%). Ve druhé tabulce jsou  $f$ , teplota, nesymetrie  $UnbU$ ,  $UnbI$  a  $fiUnbI$  (měřené metodou – ANSI/IEEE), signalizace podtečení, přetečení, výstupů a chyb.

Na další záložce jsou ve dvou tabulkách výkony:  $P$ ,  $Q$ ,  $S$ ,  $D$  účinníky  $PF$  a  $\cos \varphi$ , výkony prvních harmonických  $P_{fh}$  a  $Q_{fh}$  a Flicker –  $P_{ST}$ ,  $P_{LT}$ . [3]

### A.2.2 PQube

Oproti ostatním umožňuje zobrazovat  $u_{THD}$  a  $i_{THD}$  ve  $V/A$  nebo % a počítá i maximum ze všech fází. Flicker kromě standardních  $P_{ST}$ ,  $P_{LT}$  udává navíc i  $P_{inst}$ , nesymetrii  $U$  a  $I$  je možno měřit a zobrazovat třemi metodami – ANSI/IEEE, IEC, GB. Je zde také  $U$  a  $I$  1. harmonické (velikost a úhel).

V tabulce výkonů zobrazuje navíc špičkový  $P$ ,  $S$  a  $I_{RMS}$ . Další tabulka obsahuje  $U$  a  $I$  (velikost a úhel) vybrané harmonické. [6] [5]

### A.2.3 BK-Touch

Toto prostředí obsahuje dva způsoby zobrazení aktuálních dat. V první, zjednodušené, tabulce jsou pevně definované veličiny, druhá umožňuje vybrat z několika přednastavených skupin nebo definovat vlastní skupiny. Obnova dat je nastavená pevně, nedá se změnit.

Zjednodušená tabulka obsahuje jen několik základních veličin a její předností je přehlednost. Rozšířená tabulka obsahuje 4 přednastavené skupiny a 3 volitelné. V základní skupině je o několik veličin víc, než ve zjednodušené tabulce. Ve skupině napětí je oproti ENVIS.DAQ navíc  $U$  1. harmonické ( $V, ^\circ$ ), min. a max.  $U_{RMSpp}$ . Ve skupině proudů je navíc  $I$  1. harmonické ( $A, ^\circ$ ) a min. a max.  $I_{RMSpp}$ .

Do dalších tří definovatelných skupin je možné volit z velkého množství všech veličin, které přístroj měří a vypočítává. Oproti ENVIS.DAQ umožňuje navíc zobrazit púlperiodové  $U$ ,  $I$ , několik impedancí, symetrické komponenty  $U$ ,  $I$ ,  $P$  a  $Q$ , více druhů nesymetrie a pro

velkou část veličin průměrnou, max. a min. hodnotu. [2] [1]

#### A.2.4 Návrh realizace

- Zobrazení ve dvou základních tabulkách jako ENVIS.DAQ – vertikálně fáze, horizontálně veličiny,
- jedna nebo více uživatelsky definovatelných tabulek, kam půjdou vybrat všechny měřené a vypočítávané veličiny, půjde měnit velikost písma jednotlivých veličin nezávisle na sobě, do tabulky bude možné vkládat i grafy,
- možnost manuálního obnovení dat nebo nastavení intervalu,
- rozhodnout, zda podle velikosti obrazovky měnit počet zobrazovaných veličin nebo jejich velikost.

## A.3 Elektroměr

Celkově 13.9.2012 - 29.9.2012					Současný měsíc 13.9.2012 - 29.9.2012					Minulý měsíc 13.9.2012 - 13.9.2012				
3p Fázové														
	Činná energie				Jalová energie									
	Import [kWh]	Export [kWh]	Induktivní [kvarh]	Kapacitní [kvarh]										
Tarif 1	2,602	0	0	1,336										
Tarif 2	0	0	0	0										
Tarif 3	0	0	0	0										
Celkově	2,602	0	0	1,336										

	Pmax		Čas
	Hodnota[kW]		
Tarif 1	0,008		13.9.2012 19:20:00
Tarif 2			
Tarif 3			
Celkově	0,008		13.9.2012 19:20:00

Čas nulování	Čas záznamu	Čas nulování Pmax
13.9.2012 18:51:23	29.9.2012 14:45:00	13.9.2012 18:51:23

Resetuj elektroměr      Resetuj Pmax

Obrázek 5: Elektroměr – ENVIS.DAQ

Avg [Wh]	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
AP	18,01M	17,92M	19,16M	296,1k	55,09M
Avg [VArh]	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
AQ	3,342M	-6,587M	-2,696M	-52,22k	-13,67M
Avg [VAh]	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
AS	18,32M	19,09M	19,35M	300,7k	56,76M
Avg [Wh]	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
AP1	17,98M	17,90M	19,17M	295,4k	55,05M
Avg [VArh]	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
AQ1	3,200M	-6,484M	-1,647M	-51,62k	-12,90M
Avg [Wh]	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
AP imp	18,01M	17,92M	19,16M	296,1k	55,09M
Avg [Wh]	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
AP exp	0,000	0,000	0,000	0,000	0,000
Avg [VArh]	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
AQL	3,342M	0,000	0,000	0,000	0,000
Avg [VAh]	Fáze L1	Fáze L2	Fáze L3	Fáze N	Celkově
AQC	0,000	-6,587M	-2,696M	-52,22k	-13,67M

Obrázek 6: Elektroměr – BK-Touch

Energy <small>(since 2012/09/27)</small>	5,266MWh
Apparent Energy <small>(since 2012/09/27)</small>	6,406MVAh
Reactive Energy <small>(since 2012/09/27)</small>	3,219MVARh
Carbon <small>(since 2012/09/27)</small>	1,707Mg
Carbon Rate	0,29kg/jh

Obrázek 7: Elektroměr – PQube

### A.3.1 KMB DAQ

Klasický elektroměr obsahuje pouze ENVIS.DAQ, ostatní umožňují pouze zobrazit některé veličiny energie za dobu od posledního vynulování.

### A.3.2 BK-Touch

Umožňuje zobrazit veličiny energie do uživatelsky definované tabulky dat. Oproti ENVIS.DAQ má navíc činnou a jalovou energii 1. harmonické a zdánlivou energii. [2] [1]

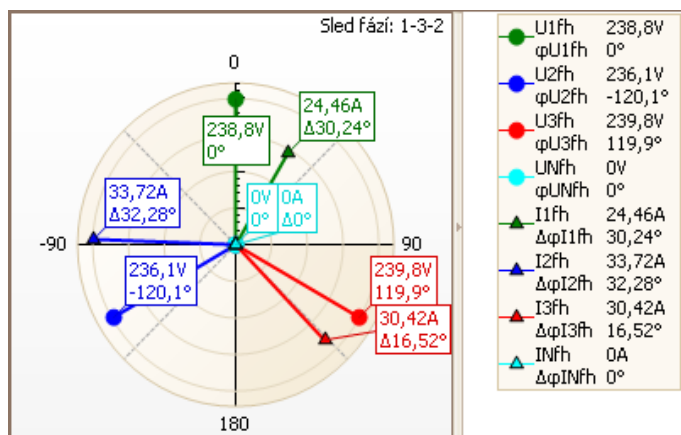
### A.3.3 PQube

Veličiny energie také zobrazuje mezi aktuálními daty, navíc má zdánlivou energii a vypočtené množství CO<sub>2</sub> vytvořené spotřebovanou energií. [6] [5]

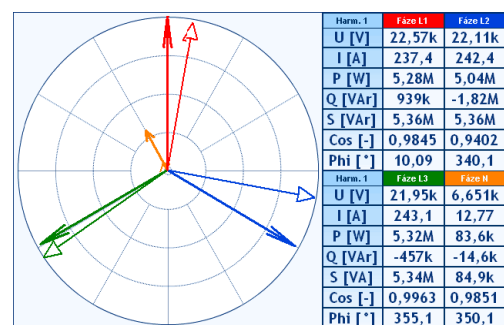
### A.3.4 Návrh realizace

- Ponechat stávající způsob zobrazení z ENVIS.DAQ,
- případně přidat některé veličiny.

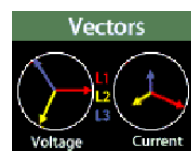
## A.4 Fázorový diagram



Obrázek 8: Fázorový diagram – ENVIS.DAQ



Obrázek 9: Fázorový diagram – BK-Touch



Obrázek 10: Fázorový diagram – PQube

### A.4.1 Porovnání

Fázorový diagram v programu ENVIS.DAQ je vždy zobrazen pro napětí i proud a nejde to změnit, lze pouze skrýt jednotlivé fáze. U každé čáry zobrazuje i její velikost a úhel. BK-Touch umožňuje zobrazovat jen U, jen I nebo obě veličiny zároveň a umožňuje vybrat i harmonickou. V tabulce zobrazuje navíc i velikosti P, Q, S, Cos φ.[1] PQube umožňuje zobrazovat fázorový diagram pouze na displej přístroje, ne ve webové aplikaci.[6]

Tabulka, která je použitá v BK-Touch, je přehlednější než textový zápis ENVIS.DAQ, který ale má u každé veličiny přímo symboly, což je názornější.

### A.4.2 Návrh realizace

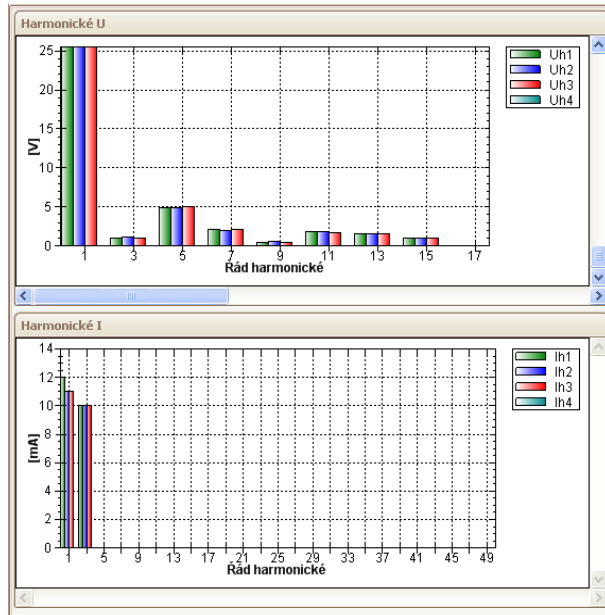
U fázorového diagramu je nutné zobrazovat pouze velikost U, I a jejich úhly, další veličiny se zobrazují na jiných obrazovkách.

- Ponechat rozvržení zobrazení z ENVIS.DAQ,
- veličiny zobrazit v tabulce s ponecháním zobrazení příslušných symbolů,
- v nastavení komponenty umožnit zobrazovat US/EU styl – 0° nahore nebo v pravo,

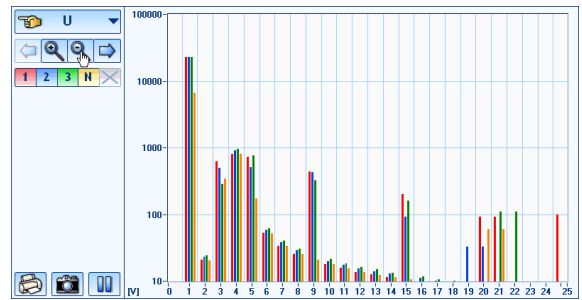
- v nastavení přímo na obrazovce umožnit vypínání zobrazování hodnot v grafu – občas to bývá nepřehledné,
- případně přidat možnost zobrazovat i jiné harmonické než 1., pokud je to možné.



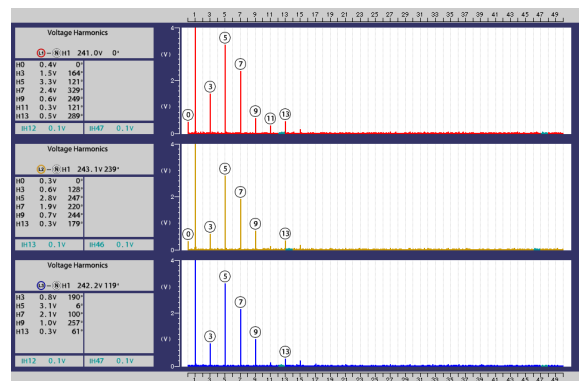
## A.5 Zobrazení harmonických



Obrázek 11: Harmonické – ENVIS.DAQ



Obrázek 12: Harmonické – BK-Touch



Obrázek 13: Harmonické – PQube

### A.5.1 Porovnání

ENVIS.DAQ zobrazuje U a I ve 2 grafech. Přepínačem je možné skrývat jednotlivé fáze. Zoom se provádí myší a je možný horizontálně i vertikálně. Zobrazují se jen liché harmonické. Vedle zobrazení spektra umožňuje i vypsát data i do tabulky.

BK-Touch umožňuje v jednom grafu přepínat mezi harmonickými pro U, I, P, Q a S. Umožňuje vypínat jednotlivé fáze. Zoom je možný pouze horizontálně a to pomocí tlačítek. Zobrazuje sudé, liché i 0. harmonickou. [1]

PQube zobrazuje každou fázi v jiném grafu, neumožňuje zoom, zobrazuje sudé, liché i 0. harmonickou. [5]

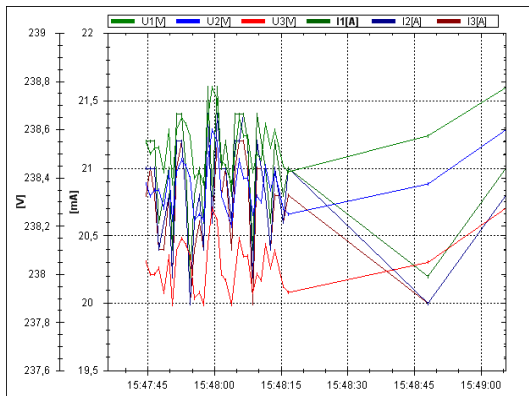
### A.5.2 Návrh realizace

Nejpodstatnějším problémem zobrazení harmonických je možnost přiblížení. Pro dotykový displej se nabízí řešení pomocí tlačítek jako má BK-Touch, to je sice jednoduché, ale nedosahuje takového komfortu jako zoom výběrem požadované oblasti. Jako vhodná se jeví varianta kom-

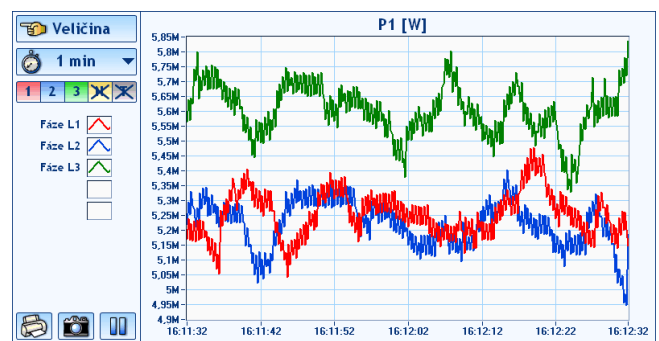
binace obou metod - pro vertikální zoom použít tlačítka, ale zároveň umožnit i zoom pomocí výběru oblasti myši nebo tažením prstu, případně stylusu. Otázkou je, jestli to nebude příliš komplikované na realizaci v rámci použitých komponent.

- Jeden graf s přepínáním zobrazované veličiny,
- všechny fáze v jednom grafu s možností jejich vypnutí,
- vertikální zoom pomocí tlačítek,
- prozkoumat možnost zoomu výběrem oblasti na dotykových obrazovkách,
- možnost povolit zobrazování sudých harmonických a 0,
- nastavitelná možnost zobrazovat v absolutních hodnotách a %.

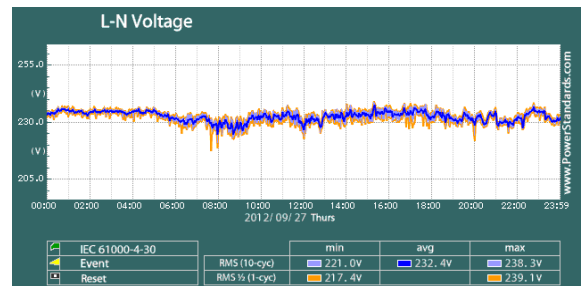
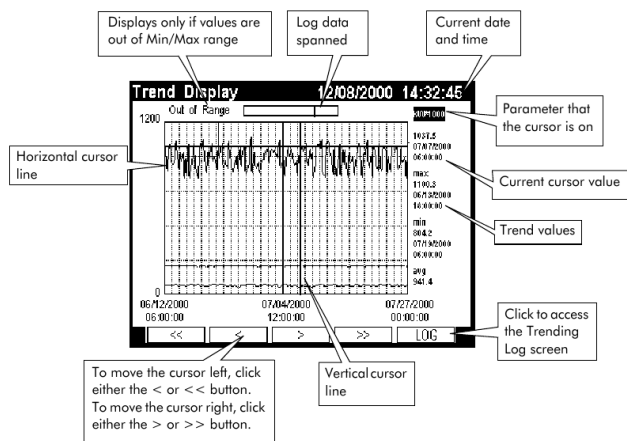
## A.6 Historická data



Obrázek 14: Graf – DAQ



Obrázek 15: Graf – BK-Touch



Obrázek 17: Graf – PQube

Obrázek 16: Graf – ION

### A.6.1 ENVIS.DAQ

Možnosti jsou velmi omezené. Zobrazuje U a I, P a Q nebo  $\cos \varphi$  a PF pro všechny fáze, nelze nic vypnout. Od spuštění programu po nastaveném intervalu zaznamená aktuální hodnotu a posune graf, to má za následek nepříjemné skákání grafu a změnu jeho měřítka. Ruční změna měřítka funguje pouze do načtení další hodnoty. Pokud se změní interval měření, graf vypadá jako na obrázku 14. Jedinou možností jak ovlivnit množství měřených dat je ruční vymazání všech dat. [3]

Přímo na měřicích přístrojích SMP a SMPQ je možné na vestavěném displeji zobrazit denní a týdenní graf.

### A.6.2 BK-Touch

Možno vybírat mezi zobrazením velkého množství veličin, pro každou lze případně skrývat jednotlivé fáze. Lze volit zobrazovaný interval – 1 min, 10 min a 1 h, ale ne frekvenci měření. [1]

### A.6.3 PQube

Umožňuje zobrazit pouze statické grafy za den nebo týden, které obsahují vždy jednu veličinu. V každém grafu je výrazně zobrazena průměrná hodnota, ale světlými barvami jsou vidět i křivky min. a max. pro 1 a 10 cyklů. [5]

#### A.6.4 ION

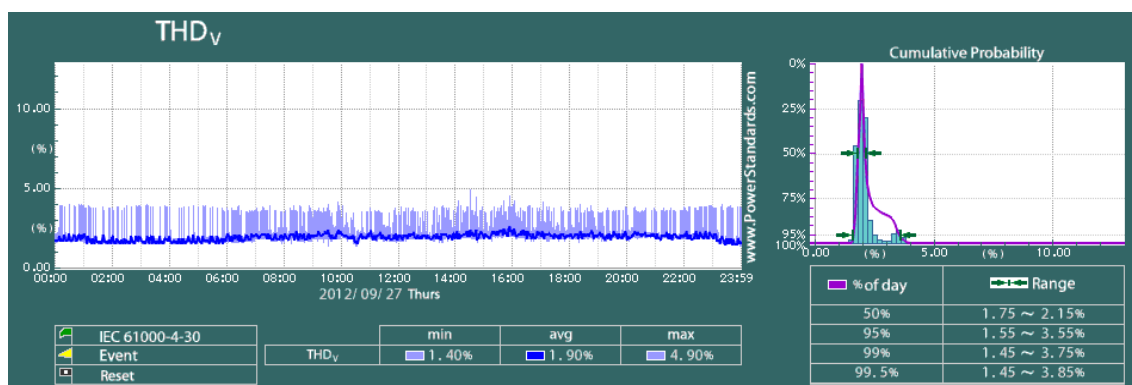
Měřicí přístroj ION umožňuje v jednom grafu zobrazit čtyři veličiny. Jde nastavit interval zaznamenávání a délka zobrazeného úseku. V grafu umožňuje nastavit kurzory, pro které vedle grafu zobrazuje jejich hodnoty. [7]

#### A.6.5 Návrh realizace

Současný způsob zobrazení v ENVIS.DAQ je nevhodný. Jako nejlepší řešení se jeví zobrazovat data volitelně od posledního smazání nebo za poslední 1 min, 10 min, 1 h, 10 h, den, týden – na to je nutné číst archiv, což ENVIS.DAQ neumí.

- Výběr způsobu zobrazení dat – od posledního resetu nebo za poslední 1 min, 10 min, 1 h, 10 h, den, týden a uživatelsky definovaný interval,
- v jednom grafu umožnit zobrazovat víc veličin - víc os y,
- osy y volitelně na levé straně nebo jedna nalevo a jedna napravo,
- skrývatelné kurzory,
- zvážit možnost zobrazovat světle min. a max. hodnoty pro data z archivu.

#### A.6.6 Histogram



Obrázek 18: Histogram – PQube

Histogram z porovnávaných aplikací umí zobrazit pouze PQube. Měl by ho umět i ENVIS s pluginem, který momentálně nejde stáhnout. [5]

Protože histogram zabírá místo vedle grafu, kterého na menších displejích není mnoho, připadají v úvahu dvě možnosti zobrazení. V samostatném tabu, kde ale ztrácí kontakt s

daty zobrazenými v grafu, nebo do tabu s grafy, kde se bude zobrazovat buďto vedle nebo v další vrstvě přes graf a půjde skrýt.

## A.7 Celkový vzhled aplikace



Obrázek 19: Ovládání – ENVIS.DAQ



Obrázek 20: Ovládání – BK-Touch

Aplikace ENVIS.DAQ má z celkového ovládání oproti konkurenci přínosný posuvník ovládající rychlost načítání dat a signalizaci stavu zařízení, naopak tlačítka pro skrývání fází by se mohla přesunout jinam. V horní liště zabírají místo důležitějších věcí a pro některé tabulky a grafy jsou zbytečné nebo nefungují.

Aplikace BK-Touch je přizpůsobena jednomu rozlišení a velikosti displeje, proto vypadá velice kompaktně a její ovládání je na dotykové obrazovce výrazně lepší než ENVIS.DAQ. Umožňuje také zobrazit více veličin a jednotlivé obrazovky a grafy více přizpůsobit. Jako uživatelsky velmi přínosné se jeví zobrazování 4 veličin v horním pruhu ve všech obrazovkách, kde je možné zobrazovat U, I, P, Q nebo S. [1]

- V horním pruhu zobrazovat posuvník pro ovládání rychlosti načítání dat a tlačítka pro manuální obnovení,
- uživatelsky definované veličiny místo tlačítek pro skrývání fází,
- možnost vybrat různé veličiny a fáze – např. U, I, P, Q pro 1. fázi,
- počet zobrazovaných veličin měnit podle velikosti displeje,
- skrývání fází přesunout buď do tabů, které to vyžadují, nebo do nastavení,
- používat výraznější barvy, které jsou dobře čitelné a rozlišitelné.

## A.8 Závěrečná doporučení

V současné době grafický klient kopíruje uživatelské rozhraní aplikace ENVIS.DAQ. Toto rozhraní je navrženo pro desktop, proto jsou některé jeho části nevhodné pro embedded zařízení s dotykovou obrazovkou. Přesto navrhuji, aby grafický klient i nadále v co největší míře z toho rozhraní vycházel, aby uživatelé, kteří budou přicházet do styku s grafickým klientem i aplikací ENVIS.DAQ nebyli zmateni tím, že se stejná data zobrazují odlišně.

Řadu prvků uživatelského rozhraní a způsobů zobrazení dat je nutné přizpůsobit omezenému rozlišení a dotykovému ovládání. Velmi dobře je v tomto směru optimalizovaná aplikace BK-Touch. Je nutné vyřešit přizpůsobení uživatelského rozhraní displejům s různou velikostí a rozlišením, což je nutné otestovat na více displejích.

## Seznam použité literatury

- [1] BK-Touch Zobrazení on-line měřených dat (2. část). *ELCOM, a.s.* [online]. 2010 [cit. 2012-09-29]. Dostupné z: [http://www.elcom.cz/touch/bk\\_touch\\_02.html](http://www.elcom.cz/touch/bk_touch_02.html)
- [2] ELCOM, a.s. - Divize Virtuální instrumentace. *BK-ELCOM Komplexní systém monitoringu a analýzy kvality elektřiny: Uživatelský manuál BK-Touch Aplikace pro ovládání analyzátoru Verze 2.8.0.* Ostrava - Pustkovec, 2009. Dostupné z: <http://www.elcom.cz/files.php?id=/4/download/bk-elcom/BK-TOUCH.pdf>
- [3] KMB SYSTEMS, s.r.o. *SMV, SMP, SMPQ Multifunkční panelové přístroje a analyzátoři kvality energie: Návod k obsluze Firmware v. 1.0.* Liberec, 2011. Dostupné z: <http://kmb.cz/index.php/cs/ke-stazeni/category/5-merici-pristroje?download=58:manual-smv-smp-a-smpq>
- [4] KMB SYSTEMS, s.r.o. *Software ENVIS: Uživatelská příručka pro podporované měřicí přístroje Verze 1.1.* Liberec, 2012. Dostupné z: <http://kmb.cz/index.php/cs/ke-stazeni/category/2-software?download=197:envis-v11-uzivatelska-prirucka>
- [5] PQube: Live World Map of Power Quality. POWER STANDARDS LAB. *Power Standards Lab* [online]. 2012 [cit. 2012-09-27]. Dostupné z: <http://maps.pqube.com>
- [6] POWER STANDARDS LAB. *PQube(R) Installation & Users's Manual: Firmware Rev 2.1.* Alameda USA, 2012, 113 s. Dostupné z: <http://www.powerstandards.com/Download/PQube%20Manual%202.1.pdf>

- [7] SCHNEIDER ELECTRIC. *PowerLogic ION7550 / ION7650 User Guide*. Saanichton, Canada, 2010. Dostupné z: [http://download.schneider-electric.com/files?L=en&p=&p\\_docId=&p\\_docId=&p\\_Reference=70002-0248-07&p\\_EnDocType=User%20guide&p\\_File\\_Id=3408992&p\\_File\\_Name=70002-0248-07.pdf](http://download.schneider-electric.com/files?L=en&p=&p_docId=&p_docId=&p_Reference=70002-0248-07&p_EnDocType=User%20guide&p_File_Id=3408992&p_File_Name=70002-0248-07.pdf)

## Příloha B - Ukázky zdrojových kódů

Zde jsou ukázky zdrojových kódů, které byly moc dlouhé do samotného textu práce.

Zdrojový kód 1: Ukázka použití knihovny libceaarchive

```
mainData *mD;
config *cfg;

ReadWriteArchive* rwa = new ReadWriteArchive \
    ("/path/Device.cea");
rwa->setDebug(OFF);
rwa->openArchive();

int result = rwa->readMainArchive(&mD,&cfg,&nRec);
if(result == 0)
{
    //zde se zobrazí data, která jsou uložena ve strukture mD
}

if (result != -1 && result != -2)
{
    rwa->freeMainData(nRec,&mD);
    free(cfg);
}
rwa->closeArchive();
delete rwa;
```



## Zdrojový kód 2: Ukázka konfiguračního souboru projektu libceaarchive

```
TEMPLATE = lib
CONFIG += console
CONFIG -= app_bundle
CONFIG -= qt

INCLUDEPATH += ../lib

SOURCES += \
    ReadWriteArchive.cpp \
    ReadHeader.cpp \
    ReadData.cpp \
    ReadConfig.cpp \
    swap.cpp

HEADERS += \
    ReadData.h \
    ReadHeader.h \
    ReadWriteArchive.h \
    ReadConfig.h \
    swap.h

win32 {
    INCLUDEPATH += C:/MinGW/include

    LIBS += -LC:\MinGW\bin \
        -lzlib1 \
        -lzip
}
```

### Zdrojový kód 3: Ukázka použití knihovny libactdata

```
Context *context = new Context ();
context->setConnType (0);
context->setModbusIP (" 192.168.0.10 ");
context->setModbusPort (s.value (502));

ReadWriteFactory rwf = new ReadWriteFactory ();
ReadWrite *rw = rwf->chooseRWType (context );

if (rw->connect () != 0) {
    printf ("\n_ERROR");
    return -1;
}

data = new CSMPDataCalc;
cntd = false;
rw->read (data);
connected = rw->isConnected ();

if (connected) {
    //zde zobrazim nactena data
}
rw->disconnect ();
delete data;
delete Context;
delete ReadWriteFactory;
```

#### Zdrojový kód 4: Ukázka konfiguračního souboru

```
[ Default ]
modbusBaud=9600
modbusDataBit=5
modbusDevice=
modbusParity=even
modbusStopBit=1
connection=1
modbusip=192.168.1.99
modbusiptext=192.168.1.99
modbusport=502

[ Global ]
TabNames\1\File=
TabNames\1\Name=Config
TabNames\1\Type=Config
TabNames\size=5
TabNames\2\Name=Demo
TabNames\2\Type=Archive
TabNames\2\File=/path/Demos_SMP_Devices.cea
TabNames\3\Name=TUL1
TabNames\3\Type=Device
TabNames\4\Name=TUL2
TabNames\4\Type=Device
TabNames\5\Name=Mixed
TabNames\5\Type=Mixed
language=en

[TUL1]
connection=1
modbusiptext=s9.mti.tul.cz
modbusip=147.230.184.193
modbusport=502
```

```

filenameSmpConfig=
filenameSmpInstallConfig=
filenameSmpElectricityMeterConfig=
filenameSmpActData=
UserDefinedWidget \1\name=uLN 0
UserDefinedWidget \1\value=uLN[0]
UserDefinedWidget \1\row=1
UserDefinedWidget \1\fontSizeName=40
UserDefinedWidget \1\fontSizeValue=80
UserDefinedWidget \2\name=uLN 1
UserDefinedWidget \2\value=uLN[1]
UserDefinedWidget \2\row=2
UserDefinedWidget \2\fontSizeName=20
UserDefinedWidget \2\fontSizeValue=40
UserDefinedWidget \3\name=uLN 2
UserDefinedWidget \3\value=uLN[2]
UserDefinedWidget \3\row=2
UserDefinedWidget \3\fontSizeName=20
UserDefinedWidget \3\fontSizeValue=40
UserDefinedWidget \size=3
ActDataRefreshTime=1

[Demo]
UserGraphs\1\Name=User Graph
UserGraphs\1\Enabled=true
UserGraphs\size=1
UserGraphs\1\LAxis=Left Axis U[%1V]
UserGraphs\1\RAxis=Right Axis I[%1A]
UserGraphs\1\RAxisEnabled=true
UserGraphs\1\Graph\1\Name=uLL1
UserGraphs\1\Graph\1\Value=uLL
UserGraphs\1\Graph\1\Attribute=avg
UserGraphs\1\Graph\1\Phase=1

```

```
UserGraphs\1\Graph\1\YAxis=Left
UserGraphs\1\Graph\1\Color=0 0
UserGraphs\1\Graph\2\Name=iL1
UserGraphs\1\Graph\2\Value=iL
UserGraphs\1\Graph\2\Attribute=avg
UserGraphs\1\Graph\2\Phase=1
UserGraphs\1\Graph\2\YAxis=Right
UserGraphs\1\Graph\2\Color=1 0
UserGraphs\1\Graph\size=2
ULL=true
ULN=true
THD=true
FirstHarm=true
UHarm=true
IHarm=false
Device=2

[TUL2]
connection=1
modbusiptext=s8.mti.tul.cz
modbusip=147.230.184.192
modbusport=502
filenameSmpConfig=
filenameSmpInstallConfig=
filenameSmpElectricityMeterConfig=
filenameSmpActData=
UserDefinedWidget\1\name=uLN 0
UserDefinedWidget\1\value=uLN[0]
UserDefinedWidget\1\row=1
UserDefinedWidget\1\fontSizeName=40
UserDefinedWidget\1\fontSizeValue=80
UserDefinedWidget\2\name=uLN 1
UserDefinedWidget\2\value=uLN[1]
```

UserDefinedWidget \2\row=2  
UserDefinedWidget \2\fontSizeName=20  
UserDefinedWidget \2\fontSizeValue=40  
UserDefinedWidget \3\name=uLN 2  
UserDefinedWidget \3\value=uLN[2]  
UserDefinedWidget \3\row=2  
UserDefinedWidget \3\fontSizeName=20  
UserDefinedWidget \3\fontSizeValue=40  
UserDefinedWidget \size=3  
ActDataRefreshTime=1

[Mixed]

Widgets \1\ Id=59  
Widgets \1\ Connection=TUL1  
Widgets \1\ Refresh=10  
Widgets \1\ Type=UITable  
Widgets \1\ Hidden=false  
Widgets \size=4  
Widgets \2\ Id=234  
Widgets \2\ Connection=TUL1  
Widgets \2\ Refresh=10  
Widgets \2\ Type=UITable  
Widgets \2\ Hidden=false  
Widgets \3\ Id=833  
Widgets \3\ Connection=TUL1  
Widgets \3\ Refresh=1  
Widgets \3\ Type=UIGraph  
Widgets \3\ Hidden=false  
Widgets \4\ Id=352  
Widgets \4\ Connection=TUL2  
Widgets \4\ Refresh=1  
Widgets \4\ Type=UIGraph  
Widgets \4\ Hidden=false

```
Widgets\1\Size=@Size(591 486)
Widgets\1\Position=@Point(593 0)
Widgets\2\Size=@Size(591 484)
Widgets\2\Position=@Point(593 486)
Widgets\3\Size=@Size(593 486)
Widgets\3\Position=@Point(0 0)
Widgets\4\Size=@Size(593 484)
Widgets\4\Position=@Point(0 486)
```

## Příloha C - Obsah CD

Příložené CD obsahuje zdrojové soubory této práce ve formátu  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  včetně původních obrázků a schémat. V samostatné složce jsou umístěny dva anglicky psané manuály k vytvořeným knihovnám libmodbus a libceaarchive. Ve druhé samostatné složce jsou licence LGPL v2.1 a BSD, pod kterými jsou licencované některé použité knihovny.

V další samostatné složce jsou spustitelné soubory klienta a knihoven včetně potřebných hlavičkových souborů ve verzích pro dvě platformy – x86-64, na které probíhal vývoj, a embedded ARM, pro kterou je aplikace určena.