

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

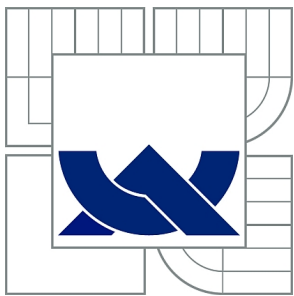
DATA KONCENTRÁTOR PRO CHYTRÉ SÍŤE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

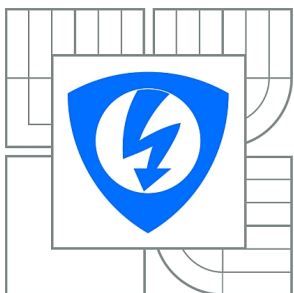
Bc. LEŠEK FRANEK

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

DATA KONCENTRÁTOR PRO CHYTRÉ SÍTĚ

DATA CONCENTRATOR FOR SMART GRIDS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. LEŠEK FRANEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL KUČERA, Ph.D.

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Lešek Franek

ID: 106433

Ročník: 2

Akademický rok: 2011/2012

NÁZEV TÉMATU:

Data koncentrátor pro chytré sítě

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte problematiku inteligentních sítí (Smart Grid) a proveďte rešerši dostupných systémů pro tuto technologii. Navrhněte data koncentrátor pro sběr dat z inteligentních sítí. Data koncentrátor by měl splňovat následující požadavky:

- vyčítání dat z elektroměrů,
- identifikace nových elektroměrů,
- routing v PLC sítích,
- vypořádání se s mřížovými sítěmi a přeslechy,
- komunikace s nadřazenými systémy,
- diagnostika.

Pro vybranou funkcionalitu koncentrátor implementujte a otestujte.

DOPORUČENÁ LITERATURA:

[1] Yaghmour, K., Masters, J., Ben-Yossef, G., Gerum, P.: Building Embedded Linux Systems. O'Reilly Media; Second Edition edition, August 22, 2008. ISBN-13: 978-0596529680.

[2] Sioshansi, F. P.: Smart Grid: Integrating Renewable, Distributed & Efficient Energy. Academic Press; 1 edition, November 10, 2011. ISBN-13: 978-0123864529.

Termín zadání: 6.2.2012

Termín odevzdání: 21.5.2012

Vedoucí práce: Ing. Pavel Kučera, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

ABSTRAKT

Cílem práce je návrh Data koncentrátoru pro inteligentní sítě (Smart Grids). Data koncentrátor stanoví rozhraní mezi serverovými systémy distribučních společností a koncovými zařízeními, které představují elektroměry, vodoměry, plynoměry a další zařízení. V práci byl vytvořen návrh hardwarového i softwarového řešení navíc byla popsána tvorba vlastní distribuce OS Linux.

KLÍČOVÁ SLOVA

Data koncentrátor, Smart Grids, Inteligentní sítě, komunikace po elektrickém vedení, PLC, embedded zařízení, sběr dat, Linux, UML, C++, elektroměr, SQLite

ABSTRACT

The goal is to design data concentrator for Smart Grids. Data Concentrator provides the interface between the server systems of distribution companies and end devices, which are electricity meters, water meters, gas meters and other equipment. There are hardware and software design solutions and there is also discussed creating its own distribution of Linux.

KEYWORDS

Data concentrator, Smart Grids, power line communication, PLC, embedded device, data collection, Linux, UML, C++, electricity meter, SQLite

FRANEK, Lešek *Data koncentrátor pro chytré sítě*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2012. 114 s. Vedoucí práce byl Ing. Pavel Kučera, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Data koncentrátor pro chytré sítě“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Pavlu Kučerovi Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

Děkuji firmě Modemtec s.r.o. za umožnění vývoje tohoto zařízení a odbornou pomoc při jeho návrhu.

Brno

.....

(podpis autora)

OBSAH

Titulní list	1
Zadání	2
Obsah	6
Seznam obrázků	11
Seznam tabulek	12
Úvod	13
1 Inteligentní sítě (Smart Grids)	14
1.1 Současný stav inteligentních sítí v Evropě	14
1.1.1 Česko	15
1.1.2 Itálie	15
1.1.3 Španělsko	15
1.1.4 Německo	16
1.1.5 Francie	16
1.2 Popis jednotlivých komponent v inteligentních sítích	16
1.2.1 Vodoměry, plynoměry, kalorimetry a další	16
1.2.2 LCD panely, mobilní telefony, televize	17
1.2.3 Elektroměry	17
1.2.4 Data koncentrátory	17
1.2.5 Servery	18
2 Analýza požadavků na data koncentrátor	19
2.1 Základní požadavky na hardware	19
2.2 Základní požadavky na funkčnost	19
2.3 Podrobnější specifikace požadavků na funkčnost	20
2.3.1 Příkazy	20
2.3.2 Možnosti komunikace s měřiči	20
2.3.3 Možnosti konfigurace a diagnostiky	20
2.3.4 Sběr dat	21
2.4 Poznatky z praxe	21
2.4.1 Škálovatelnost	21
2.4.2 Univerzálnost	21
2.4.3 Hardware	22

2.4.4	Životnost	22
3	Návrh hardware	23
3.1	Blokové schéma	23
3.2	Rozložení desek plošných spojů	25
3.3	Základní deska	25
3.3.1	Napájení	25
3.3.2	Akumulátor	25
3.3.3	Power management	26
3.3.4	Watchdog	26
3.4	Procesorová deska	26
3.4.1	Procesor	26
3.4.2	Operační paměť	26
3.4.3	Nevolatilní paměť	27
3.4.4	Sběrnice	28
3.4.5	Servisní port	28
3.4.6	Ethernet	29
3.4.7	Realizace	29
3.4.8	BIOS	34
3.5	Deska s hradlovým polem	36
3.5.1	FPGA čip	36
3.5.2	PLC modem	36
3.5.3	IEC62056-21	36
3.5.4	Optické rozhraní	37
3.5.5	Průmyslová rozhraní	37
3.6	Deska pro provoz v bateriovém režimu s bezpečnostními funkcemi	37
3.6.1	Mikroprocesor	37
3.6.2	Displej	38
3.6.3	Tlačítka	38
3.6.4	Digitální vstupy	38
3.6.5	Reléové výstupy	38
3.6.6	1 wire, I2C, analogové vstupy	39
3.7	Deska s budiči, galvanickým oddělením a svorkovnicí pro jednotlivé periférie	39
3.8	Deska s funkcí sumačního elektroměru	39
3.9	Desky s různými moduly jako je GPRS, WIFI a další	39
3.9.1	CF	39
3.9.2	HDD	40
3.9.3	GPRS	40

3.9.4	WIFI	41
3.9.5	Přídavné karty	41
3.10	Krabička a výsledný vzhled	41
4	Operační systém	43
4.1	Volba operačního systému	43
4.1.1	Windows CE	43
4.1.2	Linux	43
4.2	Vlastní distribuce Linuxu	44
4.2.1	Bitbake	44
4.2.2	Open Embedded	44
4.2.3	Zápis do paměti FLASH	45
4.2.4	Součástí distribuce	46
4.3	Jádro	47
4.4	Knihovny	47
4.5	Programy	48
4.5.1	BusyBox	48
4.5.2	AuFS	48
4.5.3	Init	48
4.5.4	Udev	49
4.5.5	DropBear	49
4.5.6	Nano	49
4.5.7	E2fsck	49
4.5.8	Syslog-ng	49
4.5.9	PicoCom	50
4.5.10	PPP	50
4.6	Konfigurace a skripty	50
4.6.1	Init	50
4.6.2	Fstab	51
4.6.3	Hostname	52
4.6.4	Informace o verzi OS	53
4.6.5	Přihlašovací obrazovka	53
4.6.6	DropBear	53
4.6.7	Ruční překlad adres	53
4.6.8	Přemístění konfiguračních souborů	53
4.7	Zavaděč	54
4.8	Automatizovaný skript pro tvorbu CF	55

5	Návrh software	56
5.1	Případy užití	56
5.1.1	Celkový systém	56
5.1.2	Jádro (Core)	58
5.1.3	Vyčítání dat (Read data)	59
5.1.4	Správa topologie (Manage topology)	60
5.1.5	Konfigurace a diagnostika (Configuration, diagnostic)	61
5.1.6	Události, logy, alarmy (Events, logs, alarms)	62
5.2	Rozdělení funkčnosti	64
5.2.1	Init	65
5.2.2	Servers	65
5.2.3	Schedulers	65
5.2.4	Diagnostic	66
5.2.5	Queue	66
5.2.6	Logger	66
5.2.7	DC cron	66
5.2.8	Validation	67
5.2.9	Model základního rozložení funkčnosti DC	67
5.2.10	Databáze a synchronizace	67
5.3	Databáze	68
5.3.1	Volba typu databáze	68
5.3.2	Srovnání výkonu	68
5.3.3	Srovnání nabízených funkcí	70
5.3.4	Závěr	70
5.3.5	Datový model	70
5.3.6	Ladění a diagnostika	72
5.4	Program Init	73
5.4.1	Start	73
5.4.2	Init	73
5.4.3	Upgrade	75
5.4.4	Watchdog	75
5.4.5	Diagnostic	75
5.5	Programy pro komunikaci s nadřazenými systémy	76
5.6	Program pro komunikaci s měřiči na protokolu MT29	77
5.6.1	Příkazy ze serveru	77
5.6.2	Mřížové sítě	78
5.6.3	Důležité úkoly spojené s routingem	79
5.6.4	Vyčítání dat	79
5.6.5	Diagnostika	80

5.6.6	Routing s nízkou prioritou	80
5.6.7	Priorita diagnostiky a routingu s nízkou prioritou	81
5.6.8	Algoritmus pro sestavování topologie	81
5.6.9	Mřížové sítě	83
5.6.10	Odesílání paketu	85
5.7	Program pro logování	87
5.8	Program pro práci s frontou příkazů	88
6	Srovnání s konkurencí	89
	Závěr	91
	Literatura	92
	Seznam symbolů, veličin a zkratk	95
	Seznam příloh	96
A	Datový model	97
B	Seznam parametrů	103
C	Objektový návrh	110

SEZNAM OBRÁZKŮ

1.1	Pyramidová struktura technologie	18
3.1	Blokové schéma HW	24
3.2	Výsledná procesorová deska [29]	29
3.3	Servisní konektor [23]	32
3.4	RJ45 konektor do panelu [33]	32
3.5	IEC62056-21 [41]	36
3.6	Použitý displej	38
3.7	Adaptér pro CF [35]	39
3.8	Možné uchycení HDD v DC [34]	40
3.9	GPRS modem [27]	40
3.10	Grafický návrh krabičky zařízení	42
5.1	Diagram užití celkového systému	56
5.2	Diagram užití částí jádra	58
5.3	Diagram užití částí vyčítání dat	59
5.4	Diagram užití částí správa topologie	60
5.5	Diagram užití částí konfigurace a diagnostika	61
5.6	Diagram užití částí události, logy, alarmy	62
5.7	Seznam programů	64
5.8	Grafické znázornění rozdělení funkčnosti DC	67
5.9	Srovnání výkonu databází	69
5.10	Formulář pro definici parametrů v databázi	71
5.11	Schéma běhu programu Init	74
5.12	Počet metrik v závislosti na počtu elektroměrů (max 1000)	82
5.13	Počet metrik v závislosti na počtu elektroměrů (max 120)	82
5.14	Schéma mřížové sítě	84
5.15	Diagram zobrazující průběh odesílání paketu	85

SEZNAM TABULEK

1.1	Výhody inteligentních sítí [1]	14
3.1	Srovnání uvažovaných procesorů [36] [37] [38] [31] [39] [40]	30
4.1	Popis připojení oddílů ze souboru fstab	52
4.2	Přemístění konfiguračních souborů	54
5.1	Srovnání výkonu databází	69
5.2	Srovnání funkcí databází	70
6.1	Srovnání s konkurencí [17] [18] [16] [15]	90

ÚVOD

Cílem práce je navrhnout embedded zařízení, které vyčítá data z měřičů a následně je poskytuje nadřazeným systémům. Hlavní uplatnění by mělo nalézt v oblasti inteligentních sítí (Smart Grids) a inteligentního měření spotřeby energií (Smart Metering). Zařízení tak bude schopno vyčíst data z elektroměrů, plynoměrů, vodoměrů případně dalších měřičů přímo po elektrické síti (PLC - Power line communication), případně jiných sběrnicích. Tato data poskytne energetickým společnostem, které je mohou využít k přesnějšímu a průběžnému vyúčtování. Zákazníkům mohou nabídnout možnost sledování spotřeby a hledání cest k úsporám. Další funkcí zařízení je brána, která dokáže směřovat příkazy z nadřazených systémů k měřičům. Díky tomu mohou energetické společnosti za běhu upravovat tarifní kalendář nebo hodnotu pseudo jističe a tím nabídnout větší spektrum cenových tarifů. Velmi podrobné a přesné informace o spotřebě a stavu sítě v jednotlivých bodech umožní energetickým společnostem lépe udržovat stabilitu sítě a usnadní hledání černých odběrů. Pro neplatící zákazníky bude možné aktivovat sociální program s omezením spotřeby jen na nejnütnější spotřebiče. Další možností pro problémové zákazníky jsou platby pomocí předplacenek. Omezení spotřeby může být také užitečné v krizových situacích kdy by jinak hrozil celkový black out.

Vzhledem k rozdílným požadavkům jednotlivých společností je nutné aby zařízení bylo maximálně škálovatelné. Pro zajištění komunikace s měřiči je nutné implementovat funkčnosti jako vyhledávání nových měřičů, hledání a optimalizace cest přes opakovače. Důležitou vlastností zařízení musí být také rozsáhlá možnost konfigurace a diagnostiky.

1 INTELIGENTNÍ SÍTĚ (SMART GRIDS)

Inteligentní sítě (Smart grids) jsou sítě, které dokážou monitorovat odběr i dodávku jednotlivých energií a umožňují tak regulaci výroby těchto energií a jejich optimální přenos k zákazníkům.

Základem celého systému jsou data o stavu sítě, které je nutné dostat do řídicích center výrobců a distributorů v co nejkratším čase. Tyto data jsou poskytována zpět spotřebitelům, díky tomu mají zákazníci možnost detailně sledovat svojí spotřebu, ceny jednotlivých energií a usnadňuje se tak cesta k hledání úspor. Díky tomu, že data o spotřebě jsou dostupná prakticky ihned, odpadá nutnost placení měsíčních záloh a fakturačního období o délce jeden rok. Tarify spotřeby jednotlivých energií můžou díky tomu nabídnout větší pestrost na jakou jsme zvyklí například od mobilních operátorů.

Systém umožňuje také tok dat směrem od distributorů těchto energií k zákazníkům. Díky tomu je možná okamžitá změna tarifů, limiteru (obdobu hodnoty jističe) a jiných parametrů, například v případě, že zákazník odjíždí na delší dobu mimo domov. Řízení odběru na koncových místech lze také s výhodou využít v době krizových situací, kdy inteligentní sítě můžou zabránit kolapsu elektrické sítě. Přehled výhod, které přináší tento systém pro spotřebitele a pro dodavatele energií:

Spotřebitelé	Dodavatelé
Vzdálený odečet dat	Vzdálené ovládání měřidel
Využití nových tarifů	Přehled o odběrných místech
Podpora pro dodávku od spotřebitelů (solární panely)	Sběr dat pro následné analýzy
Možnost okamžitě sledovat spotřebu	Zkvalitnění dispečerského řízení
Snížení neoprávněných odběrů	

Tab. 1.1: Výhody inteligentních sítí [1]

1.1 Současný stav inteligentních sítí v Evropě

Základním impulsem pro budování inteligentních sítí byl SET Plan (Strategic Energy Technology Plan) Evropské unie, který předpokládá, že do roku 2020 se sníží emise skleníkových plynů o 20 %, podíl obnovitelných zdrojů na výrobě elektřiny bude 20% a účinnost energetických technologií se zvýší o 20 %. Vše vzhledem ke stavu v roce 1990.

V rámci SET Planu začala v roce 2010 činnost Evropské průmyslové iniciativy pro chytré sítě (EEGI). Tato iniciativa je tvořena distributory a technologickými společnostmi, kteří dávají důraz na rozvoj konceptu Smart Grids. EEGI se zaměřuje na demonstrační projekty po celé Evropě, jejichž cílem je vyzkoušet jednotlivé funkční celky Smart Grids.[1]

V současné době je vývoj nebo nasazení inteligentních sítí nejvíce patrný v těchto zemích:

1.1.1 Česko

Společnost ČEZ nainstalovala do českých domácností v rámci svého pilotního projektu přes 30 tisíc inteligentních elektroměrů. Tyto moderní přístroje, které podrobně měří spotřebu elektrické energie, používají domácnosti ve třech vybraných oblastech, a to v Pardubicích, ve Vrchlabí a na Jeřmanicku.[1]

Vlastní pilotní projekt rozjela také společnost PRE v Praze.

1.1.2 Itálie

Vládní nařízení na povinnou instalaci Smart Meters platí již od roku 2006. V roce 2011 by měly chytrá měřidla tvořit 95 % všech instalovaných. Doposud byly vyměněny ve 32 milionech italských domácností (cca 85 % odběratelů).[1]

Energetická společnost Enel připravuje pilotní demonstrační projekt Smart Grids na jihu Itálie s cílem vyzkoušet aktivní řízení decentralizovaných zdrojů a spotřeby na vn úrovni distribuční sítě. Do projektu bude zapojeno cca 8 000 odběratelů a decentralizované zdroje, především FVE a VTE.[1]

1.1.3 Španělsko

Energetická firma Iberdrola spouští v roce 2010 pilotní projekt v regionu Valencie (region Castellón). Na 100 tisíc domácností je v tomto regionu již vybaveno Smart Meters. Projekt pokračuje s cílem vyzkoušet řízení nn a vn distribučních sítí pomocí víceúrovňového řešení implementace Smart Meteringu.[1]

Energetická společnost Endesa v roce 2009 spustila 4-letý pilotní projekt SmartCity v lokalitě Málaga.[1]

1.1.4 Německo

Konsorcium firem a univerzity v Karlsruhe spustilo v roce 2009 pilotní projekt výstavby chytré sítě v průmyslovém regionu Karlsruhe-Stuttgart na jihu Německa pod názvem MeRegio. Do projektu se zapojí na 1 000 odběratelů z řad domácností, průmyslových podniků, výrobních a skladovacích jednotek. V německém Mannheimu se také realizuje pilotní projekt Smart Grids pod názvem Model City Mannheim (MoMa). Další projekty spouští i velcí provozovatelé distribučních sítí (např. E.ON, RWE). Projekty konceptu Smart Grids jsou obecně v Německu výrazně podporovány a to i na vládní úrovni.[1]

1.1.5 Francie

V březnu 2010 distribuční společnost ERDF spustila ve dvou regionech pilotní projekt Smart Grids, zahrnující na 300 tisíc domácností. Na základě poznatků z tohoto testování pak v letech 2012 až 2017 proběhne instalace Smart Meters v 35 milionech francouzských domácností. ERDF také spouští rozsáhlý projekt s novou architekturou na úrovni nn i vn distribučních sítí na jihu Francie, v příměstské části Nice. Projekt bude zahrnovat integraci lokálních výrobních zdrojů, testování konceptu active demand response, jednotek akumulace el. energie, testování infrastruktury dobíjecích stanic i konceptu chytrých budov, tzv. smart homes.[1]

1.2 Popis jednotlivých komponent v inteligentních sítích

1.2.1 Vodoměry, plynoměry, kalorimetry a další

Jedná se o zařízení měřící jiné druhy energie než elektrickou, případně jiná zařízení, která získávají data jako jsou průmyslové teploměry, hladinoměry a další. Tato zařízení komunikují s elektroměrem pomocí ZigBee, Wireless M-BUSu, metalického M-BUSu nebo v případě impulsních měřidel pomocí impulsů. Vždy dochází k přenosu dat směrem z těchto měřidel do elektroměru. V závislosti na inteligenci těchto měřidel může docházet k jejich povelování nebo konfiguraci směrem z elektroměru.

1.2.2 LCD panely, mobilní telefony, televize

Pomocí těchto zařízení je zákazník informován o aktuálním stavu spotřeby, aktuálním tarifu a jiných užitečných informacích, které mu umožňují měnit spotřebu podle aktuální situace a tím i úsporu nákladů. Přispívá to k záměru energetických společností regulovat odběr energií, zejména té elektrické.

1.2.3 Elektroměry

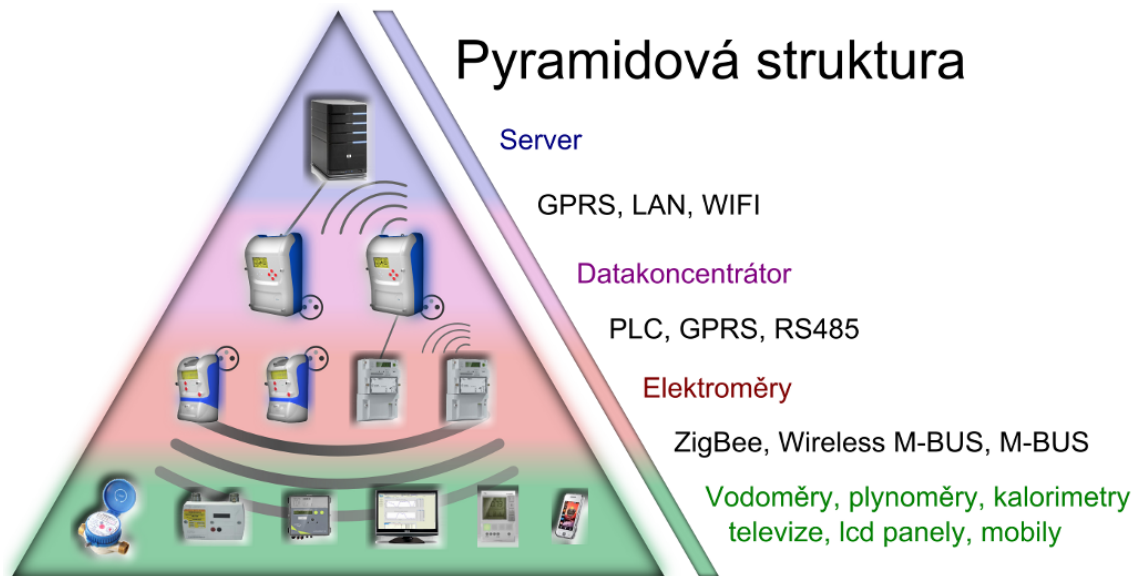
Pomocí elektroměrů je měřena spotřeba elektrické energie, napětí v příslušném místě, maxima výkonu, jsou zde zaznamenávány různé typy událostí a další užitečná data. Elektroměr může vykonávat úkony jako změna tarifu, spínání relé v závislosti na tarifu, odpojení zákazníka, limitování odběru atd. V elektroměru se může nacházet modem pro PLC síť, po které jsou data přenášena na data koncentrátor. Modem může být zhotoven modulárně nebo jako součást elektroměru. Mezi další způsoby komunikace patří přenos GPRS, který je využíván v místech kde selhává PLC, v tomto případě jsou data přenášena přímo na server. Další možností je rádiový přenos opět s využitím data koncentrátoru. Součtový elektroměr v rozvaděči nebo elektroměry v průmyslu mohou být připojeny k data koncentrátoru metalickým vedením, jako je M-BUS, RS-485, Ethernet a další.

1.2.4 Data koncentrátoary

Data koncentrátor představuje rozhraní mezi přenosem dat po elektrické nebo rádiové síti a jiným přenosem, nejčastěji typu TCP/IP. Vzhledem k této definici by se možná lépe hodilo označení router, nicméně data koncentrátor plní mnoho dalších funkcí, jak bude uvedeno dále v této práci a proto je označení data koncentrátor mnohem vhodnější. Data koncentrátor se nachází v trafostanici (DTS), jelikož PLC přenos přes transformátor neprojde a pro energetickou společnost je velmi výhodné umístit jej do stávající instalace. Koncentrátor obsluhuje v průměru přibližně 100 elektroměrů, ale v případě sídlišť a jiných hustých zástaveb může počet elektroměrů překročit 1000. Data na server/y jsou zaslány prostřednictvím LAN, WIFI, v případě nedostupnosti těchto spojení, pomocí GPRS. Funkčnost data koncentrátoru bude podrobně popsána v další části této práce.

1.2.5 Servery

Na konci řetězce se nachází server/y, které data zpracují vhodným způsobem. Všechny nebo jen část dat je poskytnuta klientským stanicím. Zaměstnanci operátorského centra můžou zasílat příkazy a měnit tak stav jednotlivých zařízení. Díky tomu můžou předejít kolapsu sítě, nebo nastavit levnější tarif pokud je přebytek elektrické energie v síti.



Obr. 1.1: Pyramidová struktura technologie

2 ANALÝZA POŽADAVKŮ NA DATA KONCENTRÁTOR

Požadavky na data koncentrátor (DC) vycházejí ze tří hlavních zdrojů. Hlavním zdrojem požadavků je dokument [12] vydaný Asociací distributorů energií v Holandsku. Jedná se o velmi podrobně zpracovaný dokument obsahující základní popis jednotlivých zařízení z oblasti Smart Meteringu. Druhým zdrojem jsou požadavky jednotlivých firem, které doplňují obecný dokument [12] o více konkrétní funkcionality a specifické požadavky. Zde byl uvažován především dokument společnosti ČEZ Měření [13], v potaz byl brán také dokument polské společnosti Energa [14]. Třetím velmi důležitým pilířem jsou zkušenosti z praxe získané na pilotních projektech společností PRE a ČEZ Měření.

2.1 Základní požadavky na hardware

- Fast Ethernet RJ45 [14]
- PLC modem [14]
- Záložní baterie pro 3 dny provozu nebo možnost připojení UPS [14]
- Provozní teplota od -30°C do 60°C [14]
- Pouze pasivní chlazení [14]
- Vnější ochrana minimálně IP44 [14]

2.2 Základní požadavky na funkčnost

- Software pro diagnostiku a konfiguraci zařízení a poskytnutí co nejvíce informací včetně lokality kde se nachází. [12] možnost zálohovat veškerou konfiguraci zařízení, provést upgrade. Vše vzdáleně nebo pomocí přenosného zařízení. [14]
- Natavení času včetně časového pásma a letního času. Pravidelná synchronizace s nadřazeným systémem, možnost konfigurace přes servisní rozhraní. Synchronizace času v elektroměrech [12], čas se nesmí lišit o více jak 10 s. Vlastní RTC s přesností lepší než jedna vteřina na den.[14]
- Automatické nalezení měřičů a vytvoření topologie pro komunikaci. [12] [13]. Nalezení nových měřičů do 12 hodin.[14]
- Možnost zabezpečené komunikace jak s nadřazeným systémem, tak s měřiči. [12]
- Podpora alespoň 1024 měřičů na jedno zařízení. [12]
- Automatická kontrola správné činnosti zařízení (Self-check). [12]

- Pokročilé funkce logování veškeré činnosti a nastalých situací. [12] [13]
- Periodické a okamžité vyčítání dat z měřičů. [12] [13] [14]
- Provádění příkazů z nadřazeného systému. [12] [13] [14]
- Upgrade firmware v DC a měřičích. [12] [14]
- Přeposlání do nadřazeného systému alarmů z měřičů do 1 minuty. [14]
- Možnost lokálně i vzdáleně restartovat zařízení včetně uvedení do výrobního stavu. [12] [14]

2.3 Podrobnější specifikace požadavků na funkčnost

Společnost ČEZ Měření se po zkušenostech z pilotního projektu rozhodla upřesnit a lépe specifikovat některé funkcionality v dokumentu [13].

2.3.1 Příkazy

- Řízení priorit jednotlivých příkazů.
- Nadřazený systém může číst stav jednotlivé fronty úloh a jejich zaplnění, vymaz všech nebo vybraných úloh ve frontě.
- Nadřazený systém může řídit parametry jednotlivých úloh a atomických operací (time out, seskupení, atp.).

2.3.2 Možnosti komunikace s měřiči

- Možnost multicastingu a broadcastingu v sítí elektroměrů.
- Atomicita příkazů na PLC sítí na úrovni příkazů dotaz – odpověď.
- Podpora mřížových sítí na PLC.

2.3.3 Možnosti konfigurace a diagnostiky

- Nadřazený systém může číst a nastavovat hodnoty v DC (routovací tabulky, čas fronty úloh, atp.).
- Nadřazený systém může zjišťovat stav PLC/RF rozhraní, resetovat rozhraní, de/aktivovat rozhraní.
- Nadřazený systém může číst statistiky provozu PLC/RF rozhraní, úspěšnost komunikace, latence.
- Nadřazený systém může zasahovat do routovací tabulky: zapisovat (modifikovat), mazat záznamy v routovací tabulce.
- Nadřazený systém může číst logy - uchování událostí pro sledování práce DC, analýzu provozu.

- Diagnostika elektroměrů prostřednictvím DC.

2.3.4 Sběr dat

- Autonomní běh úloh sběru dat podle definic zadaných z nadřazeného systému.
- Možnost zjistit stavy zpracování dané úlohy - ne/běží zpracování, absolutní (případně i procentní) úroveň vyřízení a úspěšnosti všech kroků úlohy.
- Komprimace uložených dat - úspora přenosových kapacit
- Možnost šifrování přenosu dat - jak po PLC, tak po TCP-IP, včetně správy klíčů.
- Možnost uchování dat na zákazníkem určenou dobu.
- Čtení statusu elektroměrů a sumačního registru energie s volitelnou strategií:
 - Všechny – pouze při malém počtu elektroměrů.
 - Náhodný výběr – dle potřeby četnosti (délky intervalu) takto sebraných dat definované uživatelem - např. důležité odběry.

Poznámka: Čtení statusů – bude sloužit zejména k poskytování dat (nejčastěji na lokální úrovni DTS) pro řízení distribuční soustavy a dalším zařízením pro regulaci v rámci technologií Smart Grids.

2.4 Poznatky z praxe

2.4.1 Škálovatelnost

Ukazuje se, že různí zákazníci mají zcela odlišné požadavky na toto zařízení. Právě kvůli tomu by mělo být možné zařízení co nejvíce přizpůsobit požadavkům zákazníka při co nejmenším úsilí.

2.4.2 Univerzálnost

Systém by měl být navrhnout tak aby bylo jednoduché přizpůsobit jej pro komunikace s různými servery a tím i různými protokoly. Měla by být v budoucnu možnost jednoduše přidat elektroměry komunikující na jiném standardu jak co se týče komunikačního média tak protokolu. Cílem je dosáhnout skutečnosti, že díky data koncentrátoru budou moci být v síti elektroměry různých výrobců a i když tyto elektroměry nebudou komunikovat na stejném standardu bude vše fungovat.

2.4.3 Hardware

Vzhledem k tomu, že zařízení je nejčastěji umísťováno do DTS kde není přivedena síť Ethernet je vhodné vybavit zařízení ještě GPRS modemem.

Velkou výhodou zařízení může získat pokud v sobě zahrne činnost sumačního elektroměru, který se stejně jako DC nachází v DTS.

Zařízení by mělo být schopno detekovat nestandardní stavy jako je otevření dvířek DTS, sundání krytu DC, překročení teploty atd. Detekce těchto stavů by měla fungovat i v době výpadku elektřiny.

Dobře by měla být přístupná rozhraní RJ45 pro spojení se serverem, RS232 pro servis a diagnostiku a USB pro připojení obvyklých periférií. Dále na interní svorkovnici by mělo být co nejvíce rozhraní, které se vyskytují u snímačů jak pro domácí, tak průmyslové využití.

Z důvodu rychlé diagnostiky problému musí být zařízení vybaveno prvky umožňujícími rychlou identifikaci místa problému. K tomu nejlépe poslouží LED diody, displej případně tlačítka pro ovládání displeje. Další možností je optorozhraní pro připojení přenosného terminálu s optohlavou, který je standardním vybavením techniků. Může být použita libovolná kombinace těchto prvků.

2.4.4 Životnost

Zařízení by mělo být dimenzováno na životnost 15 let jako celek, s možností servisních zásahů a výměny komponent s nižší životností jako je například baterie. Pokud jde o SW je nutné zohlednit zejména počet přepisů FLASH paměti.

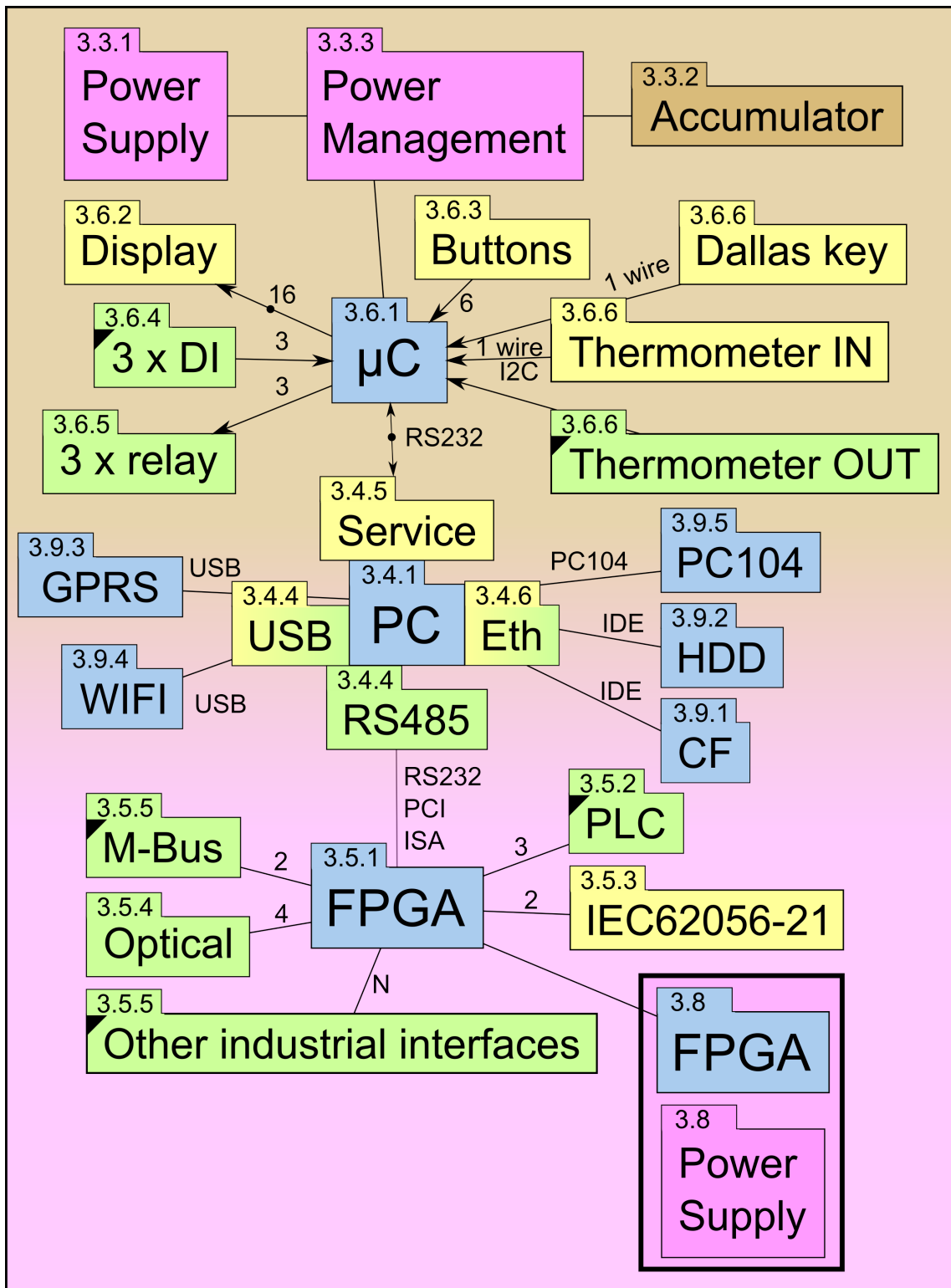
3 NÁVRH HARDWARE

3.1 Blokové schéma

Na blokovém schématu na obrázku 3.1 jsou znázorněny jednotlivé části zařízení. Jejich podrobný popis bude uveden v dalším textu.

Použité konvence:

- Na hnědém pozadí jsou umístěny prvky, které jsou funkční i v bateriovém provozu. Na fialovém pozadí prvky funkční pouze při napájení ze sítě. Hladký barevný přechod je způsoben tím, že procesorová deska je sice funkční i v bateriovém provozu, ale v omezené míře.
- Čísla jednotlivých bločků odkazují na čísla kapitol této práce.
- Prvky zajišťující napájení ze sítě mají fialovou barvu, prvky zajišťující provoz na baterii hnědou barvu.
- Prvky vyvedené přímo na kryt zařízení jsou obarveny žlutě, prvky vyvedené na svorkovnici uvnitř zařízení zeleně a samostatné desky uvnitř zařízení modře.
- Vyplněným rohem jsou označeny výstupy, které jsou galvanický oddělené.
- Vzájemné propojení prvků je znázorněno pomocí čar, kde vedle čáry je počet spojů nebo typ sběrnice, šipkou může být naznačen směr toku dat.



Obr. 3.1: Blokové schéma HW

3.2 Rozložení desek plošných spojů

Celé zařízení můžeme rozdělit na samostatné součásti:

- Základní deska obsahující napájení, případně další prvky, které nejsou na dalších deskách.
- Procesorová deska se standardními perifériemi
- Deska s hradlovým polem tvořící rozhraní pro průmyslové a jiné nestandardní periférie
- Deska pro provoz v bateriovém režimu s bezpečnostními funkcemi
- Deska s budiči, galvanickým oddělením a svorkovnicí pro jednotlivé periférie
- Deska s funkcí sumačního elektroměru
- Desky s různými moduly jako je GPRS, WIFI a další

Tyto desky mohou být rozděleny ještě na více desek, zejména z důvodu nutnosti realizace částí pomocí vícevrstvé technologie.

3.3 Základní deska

3.3.1 Napájení

Síťové napájení je realizováno pomocí síťového transformátoru, který je možné napájet ze 3 fází. Zařízení bude schopné fungovat při výpadku 2 fází. Transformátorový zdroj je použit vzhledem k menšímu rušení, než by způsobil zdroj spínaný.

3.3.2 Akumulátor

Pokud dojde k výpadku napájení na všech připojených fázích, dojde k odpojení napájení u všech součástí zařízení kromě části pro bateriový provoz. Akumulátor by měl poskytnout energii po dobu 7 dnů, přičemž v předem určených intervalech bude docházet k probuzení hlavní procesorové části a komunikačního rozhraní pro komunikaci s nadřazeným serverem. Dále je nutné zohlednit požadovanou životnost zařízení vzhledem k životnosti akumulátoru, případně navrhnout interval pro výměnu akumulátoru. .

3.3.3 Power management

Obvod, který zajistí nabíjení akumulátoru, bude poskytovat informace o stavu akumulátoru a zajistí přepnutí na bateriový provoz v případě výpadku napájení.

3.3.4 Watchdog

Standardně by měl funkci Watchdogu plnit obvod FPGA, ale v případě, že budou použita standardní rozhraní, nacházející se na procesorové desce popsané v kapitole 3.4 a bude požadavek na použití externího watchdogu, bude vhodné jej umístit na základní desku.

3.4 Procesorová deska

3.4.1 Procesor

Pro realizaci požadovaných funkcí by měl plně postačovat procesor s frekvencí v rozmezí 300 MHz až 1.5 GHz, v závislosti na probíhajících analýzách bilancí, diagnostických nástrojích, případně integrací prezentační vrstvy. Vzhledem k předpokládanému použití OS Linux, by měl mít procesor širokou podporu v jádře tohoto OS a ovladačích.

Těmto požadavkům vyhovují procesory standardu X86 a procesory architektury ARM. V případě architektury ARM nelze použít čisté Linuxové jádro, ale je nutné aplikovat záplaty pro daný typ procesoru, nicméně výrobci dodávají již upravené jádro pro příslušný procesor nebo lze využít databáze projektu Open Embedded, kde jsou shromážděny informace, které záplaty je nutné použít pro konkrétní procesory.

Nesmíme zapomínat, že procesor musí být určen pro použití v průmyslovém prostředí, což znamená rozšířený rozsah teplot a chipset s dostatkem periférií jako UART, CAN, SPI a další. Jelikož bude PLC modem implementován v FPGA, je nutné zohlednit i možnosti připojení FPGA.

Srovnání nejzajímavějších možností poskytuje tabulka 3.1

3.4.2 Operační paměť

V paměti RAM budou kromě nezbytných dat pro činnost OS a ostatních programů uchovávány data, které se často mění, jako metriky spojení s jednotlivými snímači, ukazatele do kruhových bufferů snímačů atd. Tato data budou v pravidelných intervalech zálohována do paměti FLASH. Kromě toho budou v paměti RAM dočasně uschovány data ze snímačů, než budou odeslána na server.

Vzhledem k těmto faktům je požadována velikost operační paměti mezi 128 MB až 1 GB. Typ a rychlost paměti RAM je určena typem procesoru.

3.4.3 Nevolatilní paměť

Nevolatilní paměť můžeme rozdělit na 2 kategorie se zásadně odlišnými parametry:

FLASH

Jedná se o nevolatilní paměť s omezeným množstvím přepisů, které může dosáhnout řádu miliónů. Vhodná pro použití jako základní typ paměti pro uchování OS, konfiguračních dat, případně i části dat ze senzorů. Paměť sice umožňuje omezený počet přepisů nicméně díky vysokému počtu přepisů není nutné používat paměť v režimu jen pro čtení. Pokud budeme předpokládat životnost 15 let a FLASH paměť umožňující 2 milióny přepisů, můžeme celý obsah paměti přepsat každé 4 minuty. Je ale nutné si uvědomit, že použitelnost FLASH paměti nekončí skokově, ale dochází k postupnému zpomalování zápisu.

Paměť FLASH může být v podobě čipu, paměťové karty, SSD disku, USB FLASH disku. Pro toto zařízení se nejlépe hodí paměť na čipu, která umožňuje přelakování a tím lepší ochranu proti vnějším vlivům. Dále je velmi výhodné použít paměťovou kartu, která umožňuje snadnou manipulaci s daty a škálovatelnost paměti na zařízení, které je již v provozu. Na paměťových kartách lze jednoduše distribuovat nové verze SW. V průmyslu se nejvíce používají compact flash (CF) paměti, které se vyrábějí v průmyslovém provedení s rozšířeným teplotním rozsahem. V poslední době se začaly objevovat také SD karty v průmyslovém provedení.

Základní požadovaná kapacita se bude pohybovat od 256 MB do 4 GB.

HDD

HDD reprezentuje paměťové zařízení s velkou kapacitou pro data až TB. Nicméně vzhledem k pohyblivým součástem se nehodí do průmyslu. Jelikož se ale občas objevují požadavky na uchování všech vyčtených dat přímo na DC po dobu měsíců až let, případně umístění náročného vizualizačního SW přímo do DC, může být jedinou možností použití právě HDD i za cenu menší odolnosti a rizika ztráty dat. Alternativou může být použití SSD disků, nicméně za cenu, která může v případě vyšších kapacit přesáhnout cenu zbývajících HW.

3.4.4 Sběrnice

Sběrnice lze rozdělit na 3 základní kategorie:

Paralelní sběrnice

Jako zástupce lze uvést sběrnice: PCI, PCIe, ISA, AMBA. V DC je bude možné využít pro komunikaci s FPGA, kde budou realizována průmyslová komunikační rozhraní a bude vyžadována vyšší rychlost komunikace. Dalším využitím je možnost připojení přídavných karet, zejména průmyslových měřících a komunikačních karet, pro realizaci specifických požadavků, bez nutnosti vyvíjet vlastní HW. Typ použité sběrnice se odvíjí od podpory v procesoru, zejména u platformy ARM není podpora samozřejmostí, ale vyšší řady bývají vybaveny sběrnici AMBA, případně nějakou podobnou nebo dokonce PCI nebo PCIe sběrnici.

Sériové sběrnice

V DC budou použity k propojení většiny logických obvodů. Jako zástupce lze jmenovat RS232, RS485, SPI, I2C, 1-Wire a další. Typ použitých sběrnic bude záviset hlavně na podpoře v jednotlivých logických obvodech.

Průmyslové sběrnice

K realizaci průmyslových sběrnic bude využita samostatná deska s obvodem FPGA. Nicméně pokud na procesoru či chipsetu budou umístěny průmyslové periférie, není důvod je nevyužít a v případě, že zákazníkovi budou tyto periférie postačovat, není nutné do zařízení umístit desku s FPGA obvodem. Zejména lze zmínit sběrnice RS232 nebo RS485. Pochopitelně je nutné zajistit buzení s galvanickým oddělením.

3.4.5 Servisní port

Port, na kterém bude možné spustit konzoli pro přístup do OS Linux, provést nezbytné zásahy přímo na místě, zejména v případě kdy zařízení nekomunikuje s nadřazeným systémem. Port bude možné využít také pro základní konfiguraci při výrobě.

Těmto požadavkům vyhovují sériový port a ethernetový port. Ethernetový port má výhodu v tom, že lze konzoli spustit i na dálku a provést diagnostiku, či konfiguraci vzdáleně. Pomocí sériového portu lze navíc provést konfiguraci BIOSu a je zde menší riziko, že v důsledku chyby v konfiguraci nebude fungovat.

3.4.6 Ethernet

Zařízení by mělo disponovat nejméně jedním Ethernet portem pro připojení k sítí, na které se nachází server, případně k připojení k externímu GPRS modemu. Stejný port bude možné využít pro potřeby diagnostiky vzhledem k možnosti přiřazení více IP adres jednomu rozhraní. Postačovat by měla rychlost 10 Mbit/s, nicméně v dnešní době se již standardně používá rychlost 100 Mbit/s nebo i vyšší.

3.4.7 Realizace

Platforma

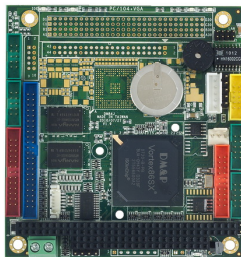
Zvažovány byly 3 varianty:

- Vlastní řešení na platformě ARM
- Vlastní řešení na platformě x86
- Hotová průmyslová deska

Důležité parametry, které je nutné znát před rozhodnutím jsou:

- Předpokládaná roční výroba v počáteční fázi v řádech stovek kusů
- Požadavek na co největší škálovatelnost výkonu a tím i výsledné ceny, vzhledem k rozdílným požadavkům zákazníků
- Vzhledem k předpokládanému množství tlak na co nejnižší cenu

Srovnání všech uvažovaných procesorů lze nalézt v tabulce 3.1.



Obr. 3.2: Výsledná procesorová deska [29]

U platformy ARM lze za plus považovat jednodušší návrh vzhledem k tomu, že většina součástí je přímo v samotném čipu. Zvažovány byly zejména modely iMX534 od společnosti Freescale a od společnosti Texas Instruments AM3892 a AM3505. Jako zajímavá možnost se jevil připravovaný chip od společnosti ST SPEAr1310, který bude využívat ARM Cortex A9 a obsahuje širokou škálu průmyslových periférií. Hlavní nevýhodou této platformy je prakticky nulová škálovatelnost, alespoň v případě nejvýkonnějších procesorů, které

jsou zde zmíněny a vysoká cena výkonných procesorů Cortex A8 a stále odsouvané termíny uvedení generace Cortex A9.

Jako zástupce platformy x86 byl uvažován zejména procesor Intel Atom řady Tunnel Creek, který je jako jeden z mála určen pro průmyslové použití. Hlavní výhodou tohoto procesoru je škálovatelnost, kdy je možné použít frekvence: 600, 1000, 1300, 1600 MHz. Byť neoficiálně lze získat informace, že procesory na frekvenci

	iMX534 Freescall	AM3892 TI	AM3505 TI	Vortex86	AM1808 TI	ATOM 6X5CT
CPU [MHz]	800	1200	600	300, 800	375	600 - 1300
Spojení s FPGA	Amba	GPMC	GPMC	PCI/ISA	PRU	PCIE (na čipu)
UART	3	3	3	5	3	-
Ethernet	1	2	1	1	0	-
SATA	1	2	0	1	0	-
PCI/PCIe	0	ano	0	ano	0	ano
CAN	2	0	1	0	0	-
IrDA	ano	ne	ano	ne	ne	-
SPI	ano	ano	ano	ne	ano	-
cena kitu [Kč]	2 527	33 920	dostupný (3 392)	dostupný (4 000)	15 382	13 500
cena procesoru [Kč]	458	1 132	339	?	237	1102 (373)

Tab. 3.1: Srovnání uvažovaných procesorů [36] [37] [38] [31] [39] [40]

1600 MHz mají značné problémy ve funkčnosti. Díky škálovatelnosti bylo toto řešení velmi zajímavé a cenově srovnatelné se stejně výkonnými procesory ARM. Další zajímavou možností bylo použití řady Stellarion, což jsou procesory řady Tunnel Creek, ale v pouzdře obsahují FPGA čip, který by bylo možné využít k tvorbě průmyslových sběrnic. Zde se jako zásadní nevýhoda projevil fakt, že je použit FPGA Altera Aria, který je příliš velký pro naše řešení a celé řešení značně prodražuje. Ztrácí se tak úspora získána umístěním do jednoho pouzdra. Hlavní nevýhodou řešení, založeného na platformě x86, je složitý návrh desky, který se vyplácí při výrobě tisíců kusů ročně.

Zvolena byla možnost použití již hotové průmyslové desky. Zvažována byla řešení firem Advantech, Portwell, SECO, I-COP. I-COP je tchajvanská dceřiná společnost firmy DM&P jež vyrábí procesory Vortex, které používá v některých svých deskách více firem, mezi jinými například zmiňovaný Advantech. Aby byl uspokojen požadavek na maximální škálovatelnost, je vhodné zvolit standardizovaný formát desky. Nejobvyklejší formáty jsou: mini-ITX, nano-ITX, PC104, Qseven a mnoho dalších. S ohledem na co nejmenší rozměry připadají v úvahu desky PC104 a moduly Qseven.

Deska PC104+ obsahuje sběrnice PCI a ISA, zatímco Qseven obsahuje modernější PCIe. Nicméně v případě Qseven se jedná pouze o modul, ke kterému je nutné připojit nosnou desku s perifériemi, toto řešení vychází po finanční stránce méně výhodně. Navíc je nutná podpora PCIe sběrnice v procesoru.

Po zvážení všech aspektů byly zvoleny desky společnost I-COP VDX-6350DE [29] a VSX-6150E-V2 [30], které se liší pouze rychlostí procesoru a velikostí operační paměti. Fotografie desky je na obrázku 3.2. Jejich popis je umístěn níže a podrobné informace lze nalézt v datasheetech na CD.

Škálovatelnost

V případě potřeby je možné nahradit tyto desky jinou PC104 deskou například s procesorem Intel Atom.

Procesor

Na desce se nachází jeden z procesorů Vortex firmy DM&P [32] [31]. Tyto procesory jsou unikátní tím, že jsou postaveny na platformě 486 a jedná se o SoC - System-On-Chip jež integruje procesor, BIOS, severní a jižní můstek do jednoho čipu. Jelikož se jedná o procesor navržený pro průmyslové použití obsahuje potřebné periférie a funguje v rozšířeném rozsahu teplot. Vývojový cyklus je stanoven na 10 let a k další změně dojde v roce 2016. Za hlavní nedostatek těchto procesorů lze považovat fakt, že nepodporují funkce pro snížení spotřeby energie (ACPI), takže není možné přejít do režimu spánku v bateriovém provozu. Spotřeba těchto procesorů je velmi nízká.

Škálovatelnost

Zajímavé jsou modely Vortex86SX [31] a Vortex86DX [32] jež jsou pinově kompatibilní a liší se zejména frekvencí 300 a 800 MHz.

Operační paměť

Použita je paměť DDR2 v podobě čipu. V případě desky SX je možné objednat velikost 128 MB a v případě desky DX velikosti 256 nebo 512 MB.

Nevolatilní paměť

Deska obsahuje 44 pinový Enhanced IDE port, který kromě standardních IDE pinů obsahuje navíc napájení. Pomocí tohoto rozhraní je možné bez problémů připojit jak CF kartu, tak HDD.

Sběrnice

Obě desky standardně obsahují ISA sběrnici, je možné přiojednat osazení PCI sběrnice.

Ze sériových sběrnic deska obsahuje 3xRS232 konektor a 1 nastavitelný RS232 nebo RS485 konektor, dále 2 USB porty. Další rozhraní jako I2C nebo 1-Wire by bylo možné teoreticky realizovat pomocí GPIO pinů, jichž je na desce 16.

Dále deska obsahuje paralelní port.

Servisní port



Jako servisní port bude využit jeden RS232 port s hardwarovým řízením, který bude mít speciální konektor do panelu [23], který je možné vidět na obrázku 3.3. Jako servisní kabel bude použit usb->RS232 [26] kabel s konektorem v podobě protikusů ke konektoru na panelu.

Obr. 3.3: Servisní konektor [23]

Škálovatelnost

Jako levnější variantu konektoru je možné použít klasický Canon konektor jehož cena nepřesahuje 10 Kč.

Ethernet



(a) RJ45 konektor do panelu samice



(b) RJ45 konektor do panelu samec



(c) Krytka konektoru do panelu

Obr. 3.4: RJ45 konektor do panelu [33]

Obě desky obsahují ethernetový port s rychlostí 10/100 Mb/s. K desce je nutné připojit konektor. Tento konektor bude umístěn na krytu zařízení, tedy je nutné použít RJ45 konektor do panelu [33]. Konektor se skládá ze 3 částí: samice obrázek 3.4a, samec obrázek 3.4b a krytka 3.4c.

Škálovatelnost

V případě nutnosti omezit náklady na minimum, je možné místo konektoru do panelu použít klasický konektor umístěný dovnitř zařízení a síťový kabel k němu přivést přes průchodku. Pochopitelně za cenu sníženého komfortu při připojování kabelu.

V případě, že nebude ethernetový konektor vůbec třeba, například bude využit pouze interní GPRS modem, je možné použít levnější desku bez ethernetového konektoru.

Další porty

Další zajímavou součástí desky jsou 2 piny, jejichž spojením dojde k resetu celé desky. Toho lze využít u externího watchdogu, kdy není nutné odpojovat napájení.

Na desce jsou ještě konektory pro připojení klávesnice a myši přes PS2 port, nicméně zde si neumím v tomto konkrétním případě představit praktické využití.

Grafická karta

Desky neobsahují grafickou kartu, tu nahrazuje konzole na servisních portech. V případě potřeby použít vizualizaci na monitoru je možné přikoupit externí grafickou kartu kompatibilní s PC104 platformou, nebo přímo desku vybavenou grafickou kartou.

Provozní teplota

Standardně se desky vyrábí pro rozsah teplot -20 až +70°C, ale za příplatek je výrobce schopen dodat desky pro rozsah -40 až +85°C.

Záruka a MTBF

Výrobce poskytuje záruku 15 měsíců. Pro desku Vortex86SX-6150E-V2 byla naměřena hodnota MTBF rovná 48 547 hodin [28], což je přibližně 5,5 roku. Hodnota MTBF byla určena na základě těchto rovnic [28]:

$$\lambda = \frac{A}{B \cdot C \cdot D} \quad (3.1)$$

kde A je počet zařízení, která selhala po testu

B je počet hodin, které test trval

C počet zařízení použitých v testu

D akcelerační teplotní koeficient

$$MTBF = \lambda^{-1} \quad (3.2)$$

kde MTBF je střední doba mezi poruchami a λ je intenzita poruch

Po dosazení do těchto rovnic hodnot naměřených v [28] získáváme:

$$\lambda = \frac{64}{1000 \cdot 13 \cdot 239} = 2,06 \cdot 10^{-5} \quad (3.3)$$

$$MTBF = (2,06 \cdot 10^{-5})^{-1} = 48547 \text{ hodin} \quad (3.4)$$

Jelikož desky procházejí výstupní kontrolou, můžeme pro průběh intenzity poruch uvažovat pouze střední část vanové křivky, kde je hodnota λ konstantní. Na základě tohoto poznatku lze vypočítat pravděpodobnost bezporuchového provozu v čase t dle vztahu: [11]

$$f(x) = \lambda \cdot e^{-\lambda t} \quad (3.5)$$

Po dosazení získáváme vztah:

$$f(x) = 2,06 \cdot 10^{-5} \cdot e^{-2,06 \cdot 10^{-5} t} \quad (3.6)$$

3.4.8 BIOS

Přímo v čipu se nachází standardní AMI BIOS od společnosti American Megatrends [24]. Do BIOSu lze vstoupit po stisknutí tlačítka DEL.

Hlavní obrazovka

Na hlavní obrazovce jsou zobrazeny základní informace o HW a je zde možné nastavit datum a čas. To lze nastavit i z OS, proto není nutné jej nastavovat zde. Dále je zde zobrazen stav čítače MTBF, který počítá počet hodin, které byla deska v provozu a čítač, který čítá počet selhání způsobených chybami na nejnižší úrovni.

Pokročilé nastavení

Na záložce CPU je možné snížit frekvenci procesoru, toto je možné provést i programově, nicméně změna frekvence má naprosto minimální vliv na spotřebu.

Zde je možné konfigurovat IDE sběrnici, kromě klasických nastavení je zde možnost nastavit ochranu proti zápisu na médium, bohužel tuto možnost lze nastavit pouze pro všechna paměťová zařízení najednou. Jelikož DC uchovává data i v nevolatilní paměti je zde ponechána možnost, která umožňuje zápis na disk.

Na záložce pro konfiguraci vzdáleného přístupu, je možné nakonfigurovat sériovou konzoli, kterou je možné použít místo klávesnice a monitoru. Volí se sériový port, v tomto případě byl zvolen port COM1, rychlost byla zvolena 115200 baud/s, je použito HW řízení a jako typ terminálu je zvolen typ ANSI. Není možné použít paritu.

Záložky USB a LAN poskytují možnost deaktivovat tyto periférie.

PCIPnP

Zde jsou podrobná nastavení pro PCI sběrnici, jsou ponechány výchozí hodnoty.

Boot

Zde jako zajímavost lze uvést schopnost bootování ze sítě, tato možnost, ale není využita, lze ji ale využít pro diagnostické účely a ladění. Dále je možné zvolit rozsáhlost P.O.S.T..

Bezpečnost

Jelikož je BIOS přístupný po restartu zařízení přes servisní port je vhodné nastavit heslo do BIOSu.

Chipset

V případě severního můstku je možné nastavit časování paměti DRAM, nicméně není k tomu důvod a je ponecháno automatické nastavení.

Na jižním můstku je možné konfigurovat ISA sběrnici, sériové porty a paralelní port. Dále jsou zde nastavení pro 2 watchdogy, u kterých je možné nastavit typ signálu po vypršení (Reset, IRQ) a čas po jehož vypršení je signál odeslán. Všechny tyto parametry je možné nastavovat programově. V BIOSu je nastaven watchdog na čas 512 s, což je dostatečný čas pro zavedení OS a spuštění programu pro občerstvování watchdogu. Díky tomu dojde k restartu zařízení pokud by došlo k uváznutí během zavádění OS. Samotný program pro občerstvování watchdogu v OS si může následně zkrátit časový interval pro kontrolu. V případě 2 GPIO portů je možné nastavit, které bity jsou vstupní a které výstupní. Toto je možné měnit programově, takže není nutné to nastavovat v BIOSu. Dále je zde možnost konfigurace spojení s pamětí FLASH a nastavení chybového signálu v případě, že jsou použity 2 desky jako redundantní, ani jedna možnost není využita.

Změny oproti výchozímu nastavení

- Advanced -> Remote Access Configuration
 - Remote Access = Enabled
 - Serial port number = COM1
 - Serial Port Mode = 115200 8,n,1
 - Flow Control = Hardware
- Security -> Change Supervisor Password = heslo
- Chipset -> SouthBridge Configuration -> WatchDog Configuration
 - WatchDog 0 Function = Enabled

– WatchDog 0 Timer = 512 Sec

3.5 Deska s hradlovým polem

Na desce s hradlovým polem budou realizovány zejména komunikační porty, které nejsou umístěné na procesorové desce. V případě, že nebude potřeba realizovat dodatečné periferie, tedy budou postačovat periferie na procesorové desce: RS232, RS485, Ethernet, není nutné tuto desku vůbec použít.

Spojení s procesorovou deskou je možné realizovat pomocí RS232, ISA nebo PCI sběrnice.

3.5.1 FPGA čip

Použity jsou FPGA firmy Altera, vzhledem dlouhodobým pozitivním zkušenostem. Jedná se o typ Cyclone 4, do budoucna bude možné použít Cyclone 5 nebo v případě velmi náročných požadavků řadu Arria.

3.5.2 PLC modem

Modem vyvinutý firmou ModemTec s.r.o., který používá proprietární způsob modulace. Do budoucna bude možné realizovat také určité standardizované způsoby komunikace jako Prime, G3, S-FSK, BPSK.

3.5.3 IEC62056-21



Rozhraní označované jako “optohlava” nebo optická sonda, je považována za standardní rozhraní pro provádění diagnostiky zaměstnanci energetických společností. Přesto, že jeho použití je běžné zejména v případě elektroměrů, je nanejvýš vhodné nabídnout jej i v případě DC.

Obr. 3.5: IEC62056-21 [41]

3.5.4 Optické rozhraní

Z důvodu relativně častého výskytu 2-traf, kdy jsou hned vedle sebe umístěná 2 trafo a jejich vedení je souběžné, což způsobuje problém s přeslechy a nutnost synchronizace komunikace mezi těmito sítěmi, je jako vhodnější varianta použití pouze jednoho koncentrátoru s distribuovanými výstupy pro připojení v jednotlivých trafo stanicích. Aby bylo možné dodržet veškeré předpisy, je nejvhodnější připojit distribuované jednotky pomocí optických kabelů.

3.5.5 Průmyslová rozhraní

Další rozhraní pro komunikaci s různými druhy měřičů, jako příklad lze uvést: M-Bus, Profibus, Can, HART, AS-Interface, Realtime Ethernet. Budiče a galvanické oddělení je realizováno na desce s výstupy.

3.6 Deska pro provoz v bateriovém režimu s bezpečnostními funkcemi

Jedná se o desku, která bude v provozu i v případě výpadku napájení. V případě, že zákazník nebude požadovat toto chování, je možné potřebné funkcionality přemístit na desku s hradlovým polem, případně procesorovou desku a tuto desku zcela vynechat.

3.6.1 Mikroprocesor

Hlavní logiku bude zajišťuje energeticky nenáročný procesor MSP od Texas Instruments. Vzhledem k možnosti složitější vizualizace na grafickém displeji je nutné zvolit typ s dostatečnou FLASH pamětí nebo připojit externí paměť typu FLASH.

Základní funkci procesoru je vytváření záznamů o nebezpečných situacích, jako je překročení kritické teploty, otevření krytu zařízení, otevření dvířek rozvaděče, přítomnost magnetického pole (napadení magnetem). Zejména v případě sundání krytu nebo otevření dvířek rozvaděče je nutné aby tyto události byly zaznamenány i v případě výpadku napájení.

Další funkci je řízení DC v případě bateriového provozu. V pravidelných intervalech dojde k probuzení procesorové desky a GPRS modulu a je umožněno odeslat aktuální informace serveru. Perioda těchto probuzení se s časem prodlužuje.

Datové spojení mezi hlavním procesorem na procesorové desce je realizováno pomocí RS232.

3.6.2 Displej



Obr. 3.6: Použitý displej

Díky umístění displeje na desku s bateriovým provozem, je možné poskytnout potřebné informace pro montáž do rozvaděče, ještě v době montáže kdy není DC připojen k napětí. Dále je možné provést základní diagnostiku pomocí displeje i v době výpadku napájení. Zařízení obsahuje grafický podsvětlený displej s rozlišením 128x64 pixelů. Displej má zabudovaný řadič [25] kompatibilní s nejpoužívanějšími řadiči KS0107/KS0108. Umístěn je

pod ochranným plexisklem.

Škálovatelnost

V případě potřeby je možné použít jiný displej se srovnatelnými rozměry, případně použít externí panel, který by již ale nebyl připojený k procesoru pro bateriový provoz, ale ke grafické kartě připojené k PC104 desce. Pochopitelně je také možné displej zcela vynechat a použít pouze štítek se základními informacemi, případně indikační LED diody.

3.6.3 Tlačítka

Na krytu zařízení budou umístěna buď 4 tlačítka, z čehož dva budou zajišťovat posun ve vizualizaci a dále zde bude tlačítko zpět a OK. V případě použití 6 tlačítek bude možný posun všemi směry - 4 tlačítka uspořádána do kříže + dva potvrzovací.

3.6.4 Digitální vstupy

1 digitální vstup signalizující sundání krytu zařízení. 2 galvanicky oddělené vstupy, 1 signalizující otevření dvířek rozvaděče a druhý pro signalizaci jiné mimořádné události.

3.6.5 Reléové výstupy

Reléové výstupy umožňují spuštění nějakého zařízení nebo signalizaci, v případě nastání určité události. Příkladem může být zapnutí topení nebo naopak větrání, v případě překročení určité teploty.

3.6.6 1 wire, I2C, analogové vstupy

Rozhraní pro připojení interních čidel teploty, Dallas klíče, případně dalších periférií. V případě snímání teploty v rozvaděči, je nutné vstup z čidla galvanicky oddělit. Dallas klíč připojený pomocí rozhraní 1 wire je možné využít k řízení přístupu pro servisní techniky.

3.7 Deska s budiči, galvanickým oddělením a svorkovnicí pro jednotlivé periférie

Vzhledem k vyšší ceně obvodů galvanického oddělení, jsou umístěny tyto součásti na speciální desce, kterou je možné upravit pro jednotlivé zákazníky přesně podle požadovaných rozhraní.

3.8 Deska s funkcí sumačního elektroměru

Do zařízení bude možné umístit samostatný modul, ve kterém bude realizovaný elektroměr. Základem tohoto modulu bude hradlové pole s funkcí elektroměru. Díky tomu DC poskytne funkci sumačního elektroměru.

3.9 Desky s různými moduly jako je GPRS, WIFI a další

3.9.1 CF



Paměť CF je možné díky kompatibilnímu režimu připojit přímo k rozhraní IDE. Rozměry CF jsou: 43x36x5 mm, kapacita se pohybuje v rozmezí 2 MB až 128 GB. K uchycení je nejvýhodnější použít adaptér který je možné vidět na obrázku 3.7.

Obr. 3.7: Adaptér pro CF [35]



Obr. 3.8: Možné uchycení HDD v DC [34]

Rozměry

43x36x5 mm + adaptér

3.9.2 HDD

Vzhledem ke kompaktním rozměrům je výhodnější použít HDD velikosti 2.5" s rozměry: 100x70x9,5 mm. K uchycení je možné použít L profil, který je možné vidět na obrázku 3.8. K připojení je možné využít rozhraní SATA nebo IDE, vzhledem k sjednocení s CF se jako vhodnější jeví rozhraní IDE.

Rozměry

100x70x9,5 mm + L profil pro uchycení

3.9.3 GPRS



Obr. 3.9: GPRS modem [27]

Jelikož se DTS vyskytují zejména v terénu kde není možné připojit Ethernet nebo jiné kabelové síťové rozhraní je nejvhodnější využít služeb mobilních operátorů a použít spojení GPRS. GPRS modemy se v drtivé většině připojují přes sériové rozhraní UART. Je nutné počítat s vytažením konektoru pro připojení externí antény. Modul, který zajišťuje síťovou komunikaci, je vidět na obrázku 3.9. GPRS bude použito jako přídatný modul. Vzhledem k nedostatku sériových rozhraní lze považovat za výhodu pokud je možné GPRS modul připojit přes USB rozhraní. Další výhodou je možnost využít technologie EDGE případně 3G.

Použité moduly

Použity jsou moduly firmy Cinterion [27], vzhledem k možnosti vestavěné podpory USB, podpory technologií GSM, GPRS, EDGE a 3G. Tyto moduly je možné připojit pomocí pájecích plošek nebo speciálního konektoru. Přičemž na výběr je několik řad, které jsou v rámci sebe kompatibilní, bohužel nejsou kompatibilní co se týče typu konektoru nebo rozmístění pájecích plošek jednotlivé řady mezi sebou. Modul pro svou funkčnost vyžaduje připojení napájení, držáku SIM karty, externí antény a pomocí USB nebo sériové linky spojení s PC.

Rozměry

33,9x35x3,3 mm

3.9.4 WIFI

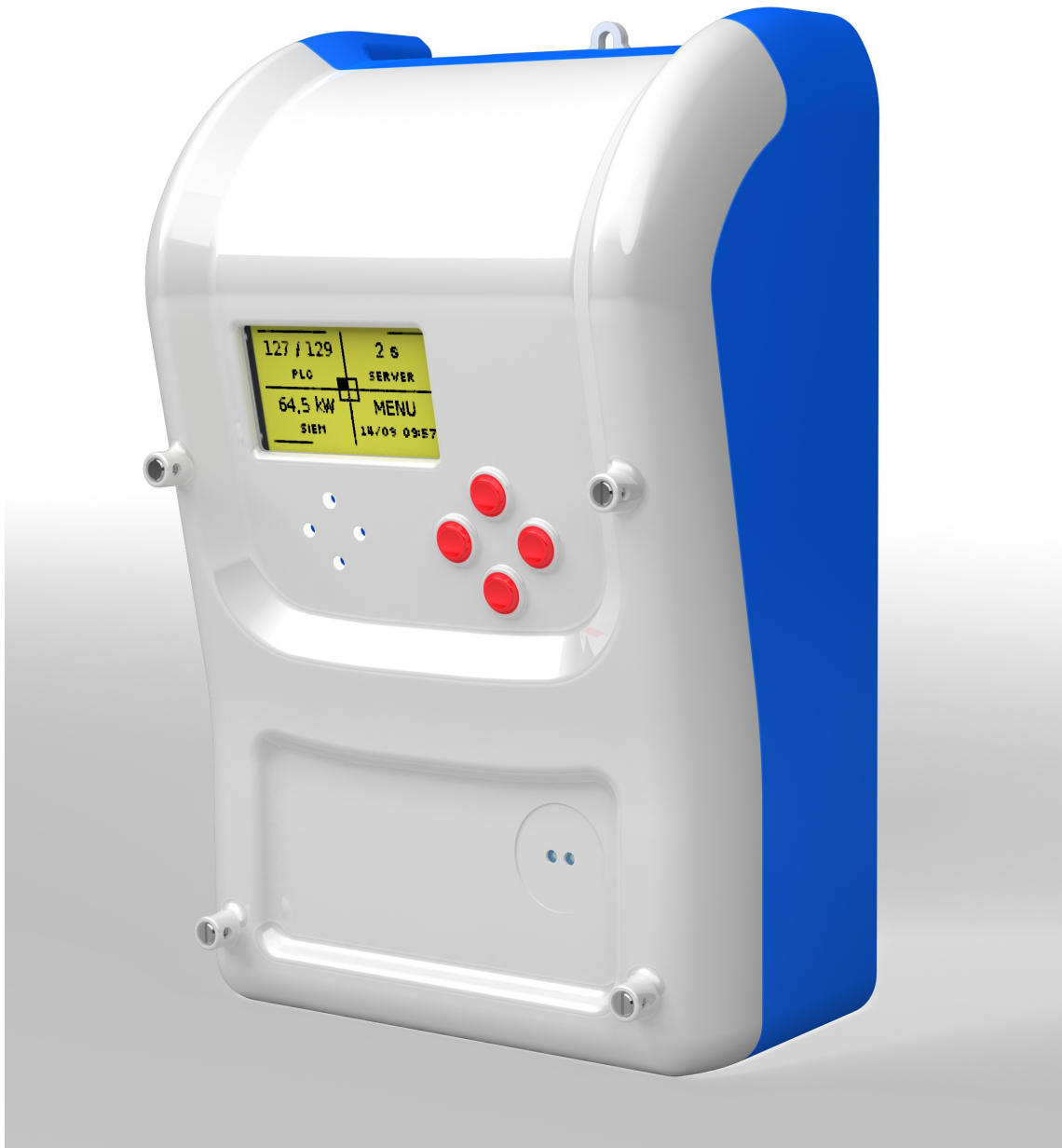
V případě, že v dané lokalitě není kabelové spojení, ale je dostupná síť wifi, je vhodnější použít wifi spojení než GPRS spojení. Pro wifi spojení se ideálně hodí použít USB modul.

3.9.5 Přídavné karty

Vzhledem k tomu, že procesorová deska respektuje standard PC104+, je možné velmi pohodlně použít přídavné desky respektující tento standard. Jako příklad lze uvést měřicí karty, grafickou kartu, komunikační rozhraní jako například Profibus.

3.10 Krabička a výsledný vzhled

Krabička obsahuje potřebné otvory pro displej, tlačítka a optické rozhraní. Celý kryt je možné sundat a dostat se tak ke svorkovnici, či jednotlivým částem zařízení. Šrouby chránící kryt mají možnost použití plomby. Sundání krytu je indikováno pomocí mikropsínače. Ve spodní části krytu je možné vyvrtat otvory pro konektory a průchodky. Pasivní chladič je umístěn v zadní části. Grafický návrh krabičky vyhotovený designérským studiem se nachází na obrázku 3.10.



Obr. 3.10: Grafický návrh krabičky zařízení

4 OPERAČNÍ SYSTÉM

Vzhledem k složitosti celého zařízení je velmi vhodné využít výhod, které poskytují operační systémy, což umožní pohodlnější programování vlastních aplikací a možnost využít aplikace třetích stran. Velkou výhodou jsou hotové ovladače pro velké množství hardware.

4.1 Volba operačního systému

Vzhledem k tomu, že není požadováno aby zařízení reagovalo v reálném čase, není nutné používat reálnový operační systém. Z běžných pokročilých operačních systému pro embedded zařízení lze jmenovat Linux a Windows CE. Oba systémy lze velmi dobře přizpůsobit pro konkrétní HW a tím omezit na minimum využití zdrojů.

4.1.1 Windows CE

Windows CE je komerční produkt firmy Microsoft, kdy je zpoplatněna každá licence poplatkem 100 až 385 Kč v závislosti od množství použitých součástí v systému, například podpora ethernet rozhraní. Dále je nutné zakoupit nástroj, který umožní sestavení vlastní distribuce Windows CE podle konkrétních požadavků. Programování probíhá nejčastěji ve Visual Studiu.

4.1.2 Linux

Linux je komunitní projekt jehož součástí je jádro operačního systému, knihovny, aplikace užitečné pro správu a chod systému. Jádro Linuxu je šířeno pod GPL licencí a jeho použití není nijak zpoplatněno. K Linuxu neexistuje oficiální podpora, ale různé firmy poskytují možnost technické podpory. Využít lze taky služeb široké komunity. Přístup k systémovým voláním se řídí standardem POSIX a jako vývojové prostředí lze s výhodou použít například Eclipse.

Vzhledem k nižším nákladům a otevřenosti zdrojových kódů byl zvolen operační systém Linux [4].

4.2 Vlastní distribuce Linuxu

Lze použít jednu z již připravených distribucí nebo vytvořit vlastní distribuci OS Linux [2] [3]. Jelikož se jedná o velmi specifické zařízení, nalezení vhodné hotové distribuce a její úprava by bylo složitější než vytvoření vlastní distribuce na míru. Proto byla zvolena tato varianta.

4.2.1 Bitbake

Pro automatizaci a zjednodušení procesu kompilace všech součástí lze použít program Bitbake. Tento program pracuje s jakýmsi předpisy (recipes) podle kterých provádí jednotlivé akce. Předpis nejčastěji obsahuje tyto akce:

- stažení zdrojových kódů přes http, git, svn, ...
- aplikování patchů
- konfigurace
- kompilace
- tvorba adresářové struktury
- výběr výsledných souborů a umístění do adresářové struktury včetně nastavení práv
- vytvoření distribučního balíčku

Po spuštění program Bitbake stáhne kompilační řetězec pro požadovanou platformu a začne zpracovávat příslušné předpisy. Ve skutečnosti jsou předpisy dost složité a je možné v nich specifikovat závislosti mezi nimi, provádět větvení podle výsledné architektury nebo jiných proměnných.

4.2.2 Open Embedded

Naštěstí není nutné psát všechny předpisy ručně, ale lze využít databázi projektu Open Embedded [22], který je zaměřený zejména na tvorbu Linuxových distribucí pro mobilní telefony na platformě ARM. Proto obsahuje již připravené konfigurace a nastavení pro tyto procesory i přímo pro vývojové desky. Například byl proveden pokus nasadit Linux na vývojovou desku Crane Board a nebyl zaznamenán jediný problém. Stejně tak se dá použít projekt i pro platformu x86. Projekt obsahuje předpisy na sestavení 2160 programů, u kterých si ještě můžeme zvolit preferovanou verzi. Jedinou nevýhodou je fakt, že se jedná o komplexní projekt zaměřený na mobilní telefony, takže není vždy zcela jednoduché dohledat ve kterém předpisu jsou definována příslušná nastavení. Navíc předpisy jsou tvořeny samotnými uživateli, takže jejich kvalita nemusí být vždy stoprocentní.

Velkou výhodou je možnost použití překryvných vrstev (overlay), kdy si nastavíme

prioritu jednotlivých adresářů s předpisy a upravené předpisy umístíme do adresářů s vyšší prioritou, díky čemuž nemusíme editovat přímo původní předpisy. Navíc tak můžeme jednoduše provést aktualizaci těchto předpisů z webu, aniž bychom ztratili námi provedené změny.

4.2.3 Zápis do paměti FLASH

Největším problémem embedded systémů je, že jelikož obsahují paměť typu FLASH, je nutné minimalizovat počet prepisů v této paměti. Navíc musíme počítat s možností výpadku napájení, které může způsobit chybu při zápisu a poškodit celý diskový oddíl. Jako řešení by se mohlo jevit nastavit celý disk jen pro čtení. S tímto nastavením by ovšem nefungoval ani Linux, který potřebuje zapisovat data při své činnosti a značně by to omezovalo i funkčnost samotného zařízení. Proto je možné situaci řešit pomocí 3 způsobů:

Symbolické odkazy

V tomto případě vedou ze systémového oddílu symbolické odkazy na soubory, do kterých chceme zapisovat na oddíl s možností zápisu. Oddíl s možností zápisu může být umístěn i v paměti RAM. Díky tomu jsme schopni eliminovat jak opakované zápisy do paměti FLASH, tak i poškození systémového oddílu díky tomu, že je systémový oddíl jen pro čtení. Značnou nevýhodou tohoto řešení je fakt, že musíme nadefinovat symbolický odkaz pro každý takovýto soubor a pokud na nějaký zapomeneme systém nebude pracovat korektně. Pokaždé když přidáme něco nového do systému, musíme zkontrolovat zda nepotřebuje zapisovat do nějakého souboru a umístit symbolický odkaz na tento soubor na příslušné místo.

Kopie systému v paměti RAM

Po zavedení systému dojde k překopírování systémového oddílu do paměti RAM a následně je zaměněn systémový oddíl za oddíl v paměti RAM. Díky tomu se opět eliminují opakované zápisy do paměti FLASH i možnost poškození systémového oddílu. Hlavní nevýhody jsou delší doba spouštění systému, způsobena nutností kopírování celého oddílu a ztráta značné části paměti RAM, jelikož se v ní nachází celý systém. Tyto nevýhody je možné do značné části eliminovat symbolickými odkazy, které vedou z části RAM do paměti FLASH u souborů do kterých se určitě nebude zapisovat. Tento postup je výhodnější oproti předchozímu v tom, že pokud na nějaký soubor zapomeneme, tak se nic nestane, pouze se bude kopírovat do paměti RAM i když by nemusel. V souvislosti s tímto řešením se lze někdy dočíst nebo doslechnout, že díky umístění systému v paměti RAM je celý systém rychlejší,

v případě embedded systémů, kdy běží neustále stejné programy je tento vliv zcela marginální a naopak se prodlužuje start systému, kdy dochází ke kopírování celého oddílu. Nicméně jedná se o nevhodnější způsob pokud chceme realtime systém, protože po startu je již přístup do paměti RAM deterministický.

Speciální souborové systémy

Tyto souborové systémy dokážou namapovat více oddílů do jednoho adresáře. Díky tomu můžeme do jednoho adresáře namapovat systémový oddíl s příznakem jen pro čtení a oddíl v paměti RAM s možností zápisu. Speciální souborový systém zajistí funkci CoW - Copy on write, kdy kopie souboru v paměti RAM je vytvořena až při požadavku do souboru zapisovat. Díky tomu je opět zajištěna bezpečnost paměti FLASH i systémového oddílu a navíc nemusíme vytvářet symbolické odkazy ani kopírovat celý systémový oddíl do paměti RAM. Pravděpodobně jedinou nevýhodou těchto souborových systémů je, že nejsou standardní součástí jádra Linuxu a nacházejí se v experimentální větvi. Proto je nutné zdrojové soubory jádra upravit před kompilací. Fakt, že je toto řešení zařazeno do experimentální větve může vyvolávat nedůvěru, ale na druhou stranu právě toto řešení je používáno v téměř všech LiveCD distribucích Linuxu a je považováno za spolehlivé.

Zvolené řešení

Zvolena byla možnost speciálního souborového systému vzhledem k výhodám popisovaným výše. Nejznámějším zástupcem v těchto souborových systémech je UnionFS, nicméně i na samotných webových stránkách projektu se dočteme, že v dnešní době je již výhodnější použít AuFS (Another union file system). Proto byla zvolena tato možnost. Je nutné upravit zdrojové soubory jádra, nastavit příslušné volby v konfiguraci jádra a zkompilovat a přidat do systému programy pro mapování a odmapování tohoto souborového systému.

4.2.4 Součástí distribuce

Distribuce sestává s těchto součástí:

- Jádro
- Knihovny
- Programy
- Konfigurace programů
- Zavaděč

4.3 Jádno

Jádno je základní a nedílnou součástí operačního systému.

Jelikož bylo rozhodnuto použít speciální souborový systém, který není přímo podporován v oficiálním jádře systému Linux, máme 2 možnosti, stáhnout oficiální jádro a upravit jej pomocí patchů nebo stáhnout již upravené jádro přímo ze zdrojů projektu AuFS konkrétně přes protokol git `git://git.c3sl.ufpr.br/pub/scm/aufs/aufs2-2.6.git`.

Verze jádra byla zvolena dle doporučení a podpory výrobce základní desky, tedy 2.6.29. Dá se předpokládat, že přechod na novější verzi jádra by nepředstavoval žádný problém, nicméně není důvod proč se nedržet doporučení výrobce desky.

Dále je nutné provést konfiguraci jádra. Požadavkem zde je, aby jádro obsahovalo jen to co alespoň za nějakého předpokladu bude využito. Díky tomu se sníží jeho nároky na paměť FLASH i RAM. V tomto případě lze s výhodou použít konfigurační soubor dodávaný výrobcem desky, který je optimalizovaný přímo na konkrétní HW řešení. Nicméně jedná se o minimální konfiguraci pouze těch součástí, které jsou umístěny přímo na desce. Pokud připojíme určité zařízení například do USB portu je nutné pro něj doplnit podporu. Jednotlivé součásti můžeme do jádra doplnit přímo nebo jako moduly, které zavedeme do systému až v případě, že je skutečně potřebujeme. V našem případě byla konfigurace výrobce desky doplněna o tyto změny:

- Byl povolen souborový systém AuFS a nastaveny jeho parametry
- Byla přidána podpora pro komunikaci pomocí GPRS modemu
- Byla přidána podpora pro zařízení USB_ACM - jedná se o rozhraní podobné FTDI používané zejména v případě modemů

Zkompilování a vytvoření výsledného balíčku bylo provedeno pomocí programu Bitbake popsaného výše.

4.4 Knihovny

První část knihoven tvoří moduly jádra, které se zavádějí dynamicky při běhu systému. Tyto knihovny se zkompilují během kompilace jádra, jen opět pomocí programu Bitbake vytvoříme distribuční balík těchto knihoven.

Druhou část knihoven tvoří standardní knihovna a jiné pomocné knihovny. Jako standardní knihovna byla zvolena EGLIBC, tedy Embedded GLIBC, tato knihovna obsahuje všechny funkce jako klasická knihovna GLIBC, jen má redukovanou velikost výsledného binárního souboru a větší podporu pro cross-kompilaci pro jiné platformy. Mezi další knihovny můžeme zařadit ncurses pro pseudo grafické rozhraní

v konzolových aplikacích. Knihovny pro šifrování, knihovny pro komunikaci pomocí sítě Ethernet a další.

Další knihovny jsou součástí programů a jsou kompilovány a přidávány do systému během přípravy programu.

4.5 Programy

Základní programy používané pro provoz operačního systému, případně pro jeho správu. Programy nutné pro provoz DC budou diskutovány v další části tohoto dokumentu.

4.5.1 BusyBox

Jedná se o program, který byl vytvořen právě pro embedded systémy, integruje základní Linuxové nástroje do jednoho programu, díky čemuž šetří místo. Spojením jádra systému a programu BusyBox jsme schopni získat základ fungujícího systému. V konfiguraci programu před kompilací si můžeme zvolit, které nástroje má obsahovat, díky tomu můžeme některé vynechat a použít plnohodnotné programy z projektu GNU Core Utilities. Použita byla verze programu 1.18.5 o velikosti 396KB.

4.5.2 AuFS

Jedná se o nástroje pro připojování a odpojování speciálního souborového systému popsaného výše. Verze nástrojů 3, celková velikost 564KB.

4.5.3 Init

Program Init je spuštěn jako první program v systému a podle pokynů v souboru `/etc/inittab` spouští jednotlivé skripty a programy umístěné v systému. Program Init může obsahovat i BusyBox, ale ten je zjednodušený a nerespektuje POSIX standard pro formát souboru `inittab`, proto byl zvolen originální program Init jehož velikost je zcela zanedbatelná.

4.5.4 Udev

Program Udev je démon, který běží v systému a hlídá zda nebyl připojen nějaký HW. Pokud tomu tak je, podívá se do svých pravidel a v případě potřeby zavede potřebný modul do jádra a vytvoří přípojný bod ve složce /dev. Stejně tak provede opačné akce v případě odpojení HW. V případě použití stálé konfigurace HW by tento program nebyl nutný, ale jelikož k DC bude možné připojit například GPRS modem přes USB, vyhneme se tak nutnosti mít rozdílnou konfiguraci systému pro oba případy nebo nutnosti mít stále v jádře zavedeny moduly potřebné pro GPRS modem. Verze programu 171, velikost 196KB.

4.5.5 DropBear

Jedná se o SSH server, který umožňuje vzdálené připojení k terminálu a přenos souborů pomocí SCP. DropBear je opět optimalizován s ohledem pro použití v embedded systémech. Verze 0.52, velikost 104KB.

4.5.6 Nano

Nano je textový editor pro úpravu textových souborů z příkazového řádku. Díky tomuto programu můžeme editovat konfigurační soubory přes sériovou konzoli nebo vzdáleně přes SSH konzoli. Alternativou by byl program Vim, ale uživatelské rozhraní programu Nano bylo zvoleno jako více přívětivé. Verze 2.2.5, velikost 92KB.

4.5.7 E2fsck

Program pro kontrolu souborového systému EXT2. Verze 1.41.14, velikost 84KB.

4.5.8 Syslog-ng

Démon, který se stará o logování, v systému Linux je použit jednotný systém logování, který většina standardních programů využívá. Zprávy z těchto programů končí právě v tomto démonu a jsou nim zpracovány podle předpisů v souboru syslog-ng.conf. Tento program je součástí programu BusyBox, ale opět pouze v okleštěné podobě, jelikož bude k uchovávání logů využita databáze, pro snadnější práci s logy, je nutné použít plnohodnotnou verzi programu. Verze 2.0.5, velikost 52KB.

4.5.9 PicoCom

Jedná se o ještě více osekanou variantu programu MiniCom, což je program pro komunikaci po sériové lince. Využit bude k diagnostickým účelům, kdy bude možné přímo z konzole odeslat libovolnou zprávu na sériovou linku nebo do GPRS modemu. Verze 1.4, velikost 12KB.

4.5.10 PPP

PPP démon je program, který se stará o vytvoření internetového spojení pomocí GPRS modemu. Verze 2.4.5, velikost 144KB.

4.6 Konfigurace a skripty

Nejsložitější částí procesu vytváření vlastní distribuce je konfigurace jednotlivých programů a vytvoření potřebných skriptů pro běh systému.

4.6.1 Init

Chování programu Init se specifikuje v souboru `/etc/inittab`, kde ale je specifikováno pouze, že se mají spustit skripty v příslušných složkách a na konci jsou zde spuštěny terminály.

Skutečnou práci odvádějí skripty ve složkách `/etc/rcS` a `/etc/rcX` kde X je číslo udávající režim, ve kterém se systém nachází. Standardně se jedná o režim číslo 5. Tyto skripty mají za úkol připravit systém pro bezproblémový provoz a spustit demony, které běží na pozadí systému. Po startu systému se spouští tyto skripty v pořadí jak jdou za sebou:

1. Banner - oznámení že systém startuje
2. Copy-rootFS - tento skript provádí více operací:
 - přemountování systémového oddílu v paměti FLASH do složky `/rootflash` aby byl tento oddíl přístupný i po překrytí souborovým systémem AuFS. Díky tomu bude možné provést případné změny v systémovém oddílu v paměti FLASH.
 - vytvoření oddílu v paměti RAM o maximální velikosti 20MB a připojení do složky `/rootram`, tento oddíl bude použit k překrytí systémového oddílu v paměti FLASH.
 - vytvoření adresářové struktury na oddílu v paměti RAM.
3. MountAll - připojení všech disků do příslušných adresářů. Podle konfigurace v souboru `/etc/fstab`, která bude diskutována později

4. Modutils - zavedení potřebných modulů do jádra
5. Udev - spuštění Udev démona
6. Syslog - spuštění Syslog démona
7. Devpts - vytvoření speciálního adresáře pro správu zařízení
8. Hostname - nastavení jména zařízení
9. Configure - akce prováděné pouze při prvním spuštění systému
10. Networking - nastavení síťových rozhraní dle konfigurací v adresáři `/etc/network/`
11. Bootmisc - různé systémové akce jako nastavení přístupových práv k terminálům, synchronizace času atd.
12. Urandom - inicializace generátoru náhodných čísel
13. Finish - vytvoření souboru `.configured` indikujícího, že je systém nakonfigurován
14. Rmnologin - povolí přihlašování do systému
15. DropBear - spuštění SSH serveru

4.6.2 Fstab

Soubor `/etc/fstab` je zpracován skriptem `mountall.sh` a podle předpisů v něm uvedených jsou připojeny jednotlivé oddíly disků, či virtuální oddíly například v paměti RAM. V tabulce 4.1 lze nalézt popis pro připojení všech použitých oddílů. Jako první je připojen systémový oddíl jehož umístění je zadáno jako parametr zavaděče. Následuje připojení dalších 10 oddílů z paměti FLASH. Dále jsou zde oddíly `proc`, `devpts`, `usbfs` což jsou speciální oddíly pro chod OS Linux. Následují dva oddíly vytvořené v paměti RAM, jeden je použit jako zařízení pro sdílenou paměť a druhý pro potřeby DC. Jako poslední jsou zavedeny 3 speciální souborové systému typu AuFS, popis tohoto systému je v podkapitole 4.2.3. V tabulce je jako atribut hvězdička, ale ve skutečnosti je tam uvedeno na příklad pro složku `/etc:` `dirs=/rootram/etc=rw:/rootflash/etc=ro,noatime,rw` což znamená že do složky `/etc` je zavedena složka `/rootflash/etc` pouze pro čtení a je překryta složkou `/rootram/etc`, do které je možné zapisovat. Jako atributy jsou použity parametry:

- `ro` - jen pro čtení, oddíl je chráněn proti zápisu
- `rw` - možnost čtení i zápisu
- `noatime` - neaktualizuje se datum posledního přístupu, díky čemuž se minimalizuje počet zápisu na paměť FLASH
- `sync` - nepoužívá se optimalizační buffer, ale data jsou do paměti zapisována okamžitě
- `size` - maximální velikost oddílu v paměti RAM

Předposlední sloupeček záloha udává, které oddíly mají být zálohovány v případě, že používáme program pro zálohování, což v našem případě neplatí. Poslední sloupeček

fsck udává v jakém pořadí mají být jednotlivé souborové systémy kontrolovány v případě, že je spuštěn program pro kontrolu, tato možnost je využita.

zdroj	cíl	typ	atributy	záloha	fsck
rootfs	/	auto	ro,noatime	1	1
/dev/hdc1	/dc	auto	rw,noatime,sync	1	2
/dev/hdc5	/linux	auto	rw,noatime,sync	1	1
/dev/hdc6	/configs	auto	ro,noatime	1	2
/dev/hdc7	/modemtec	auto	ro,noatime	1	2
/dev/hdc8	/keys	auto	ro,noatime	1	2
/dev/hdc9	/upgrades	auto	rw,noatime,sync	1	2
/dev/hdc10	/backups	auto	rw,noatime,sync	1	2
/dev/hdc11	/data	auto	rw,noatime,sync	1	2
proc	/proc	proc	defaults	0	0
devpts	/dev/pts	devpts	mode=0620,gid=5	0	0
usbfs	/proc/bus/usb	usbfs	defaults	0	0
tmpfs	/dev/shm	tmpfs	mode=0777	0	0
tmpfs	/ramdisk	tmpfs	size=60m	0	0
none	/etc	aufs	*	0	0
none	/lib	aufs	*	0	0
none	/var	aufs	*	0	0

Tab. 4.1: Popis připojení oddílů ze souboru fstab

4.6.3 Hostname

v souboru `/etc/hostname` je vhodné nastavit název zařízení.

4.6.4 Informace o verzi OS

Informace o verzi OS jsou v souborech `/etc/os-release` a `/etc/version`.

4.6.5 Přihlašovací obrazovka

V souboru `/etc/issue` lze nastavit vzhled přihlašovací obrazovky do terminálu. Zde je uveden název společnosti, kontaktní osoby a upozornění proti neoprávněnému přístupu.

4.6.6 DropBear

V souboru `/etc/default/dropbear` je možné nastavit chování tohoto SSH serveru. Nastavena byla uvítací obrazovka, změněno bylo úložiště pro klíče a certifikáty, tak aby byly na zapisovatelné části FLASH paměti a nebylo je nutné generovat při každém zapnutí do paměti RAM. Je použita asymetrická šifra RSA. Dále je možné změnit port, na kterém server běží, výchozí nastavení je 22.

4.6.7 Ruční překlad adres

V souboru `/etc/hostname` je možné nadefinovat příslušným doménovým názvům IP adresy. Takto je možné používat v celém systému doménová jména i když není dostupný DNS server.

4.6.8 Přemístění konfiguračních souborů

Jelikož lze předpokládat, že některé konfigurační soubory se můžou během provozu měnit je vhodnější je umístit mimo systémový oddíl aby nebylo nutné při změně odemykat celý tento oddíl pro zápis. Proto bylo využito mechanismu symbolických odkazů. Navíc byly konfigurační soubory roztrženy do kategorií:

- `internal` - interní použití, konfigurace nejčastěji pouze před zasláním zákazníkovi
- `network` - nastavení sítě, může provést zákazník sám

Které soubory byly kde přemístěny je zaznamenáno v tabulce 4.2:

zdroj	cíl
/etc/issue	/configs/internal/issue
/etc/issue.net	/configs/internal/issue.net
/etc/default/dropbear	/configs/internal/dropbear
/etc/network/interfaces	/configs/network/interfaces
/etc/network/options	/configs/network/options
/etc/hostname	/configs/network/hostname
/etc/hosts	/configs/network/hosts
/etc/syslog-ng.conf	/configs/internal/syslog-ng.conf

Tab. 4.2: Přemístění konfiguračních souborů

4.7 Zavaděč

Hlavním úkolem zavaděče je aktivace jádra OS. Zavaděč je umístěn v tabulce MBR a není jej možné zkopírovat jako soubor, ale je nutné použít nějaký pokročilejší nástroj. Pro zavedení Linuxu jsou nejpoužívanější tyto zavaděče:

- Grub - univerzální, široké spektrum možností
- Lilo - starší, při změně konfigurace vyžaduje aktualizaci MBR, stále používaný
- Syslinux - jednoduchý vhodný pro embedded systémy
- X-loader, U-boot - dvojice používaná pro inicializaci HW a aktivaci jádra na platformě ARM

Vzhledem k použití platformy x86 a faktu, že se jedná o embedded zařízení byl použit zavaděč Syslinux.

V konfiguraci musí být na prvním řádku uvedena konfigurace sériové konzoly. V tomto případě:

serial 0 9600 0x112, kde 0x112 znamená hardwarové řízení, 1 stop bit, 8 datových bitů, zavaděč nepodporuje paritu

Dále je nastaven výchozí profil:

DEFAULT linux

A je uvedena konfigurace profilu:

LABEL linux # název profilu

SAY Now booting the kernel for DC # zobrazení informace na konzoli

KERNEL bzImage # umístění jádra OS

APPEND root=/dev/hdc3 console=tty0 console=ttyS0,9600n8r # parametry pro Linux, umístění systémového oddílu a nastavení sériové konzoly.

4.8 Automatizovaný skript pro tvorbu CF

Z důvodu potřeby vytváření většího množství CF karet s tímto systémem, byla tato činnost automatizována pomocí skriptu `makeCF`. Na začátku jsou definována umístění jednotlivých částí systému. Dále jsou ověřeny parametry skriptu zejména parametr `-d`, který udává název zařízení, které CF reprezentuje. V dalším kroku se ověří zda je skript spuštěný s oprávněním správce aby mohl provádět následující operace. Je ověřeno zda bylo zadáno dobré zařízení jako parametr. Následně je zařízení rozděleno na 10 oddílů:

- DC FAT32 - oddíl přístupný z OS Windows, umožňuje provést upgrade, změnu základních nastavení nebo přístup k logům běžným uživatelům. Bohužel OS Windows nedokáže zpřístupnit více oddílů na vyměnitelném médiu a uvedený oddíl musí být jako první, což je poměrně limitující.
- Kernel EXT2 - umístění jádra a zavaděče, jen pro čtení
- Root EXT2 - systémový oddíl, jen pro čtení
- Linux EXT2 - konfigurace a klíče systému, lze zapisovat
- Configs EXT2 - konfigurace zařízení, jen pro čtení
- Modemtec EXT2 - programu pro funkci DC, jen pro čtení
- Keys EXT2 - klíče a certifikáty, zapisovatelné
- Upgrades EXT2 - soubory pro aktualizaci SW, zapisovatelné
- Backups EXT2 - zálohy, zapisovatelné
- Data EXT2 - získána data, zapisovatelné

Jelikož není možné mít více jak 4 primární oddíly je ještě jako 4 oddíl zařazen logický oddíl. Oddíly jen pro čtení, mohou být v případě potřeby nějaké změny, odemčeny pro zápis. To by mělo nastat pouze ve výjimečných případech jako je změna verze SW, změna konfigurace, nebo přidání určité funkčnosti.

V dalším kroku je nainstalován zavaděč. Dále jsou zkopírovány všechny potřebné soubory a je zavolán příkaz `sync`, který provede zápis na disk.

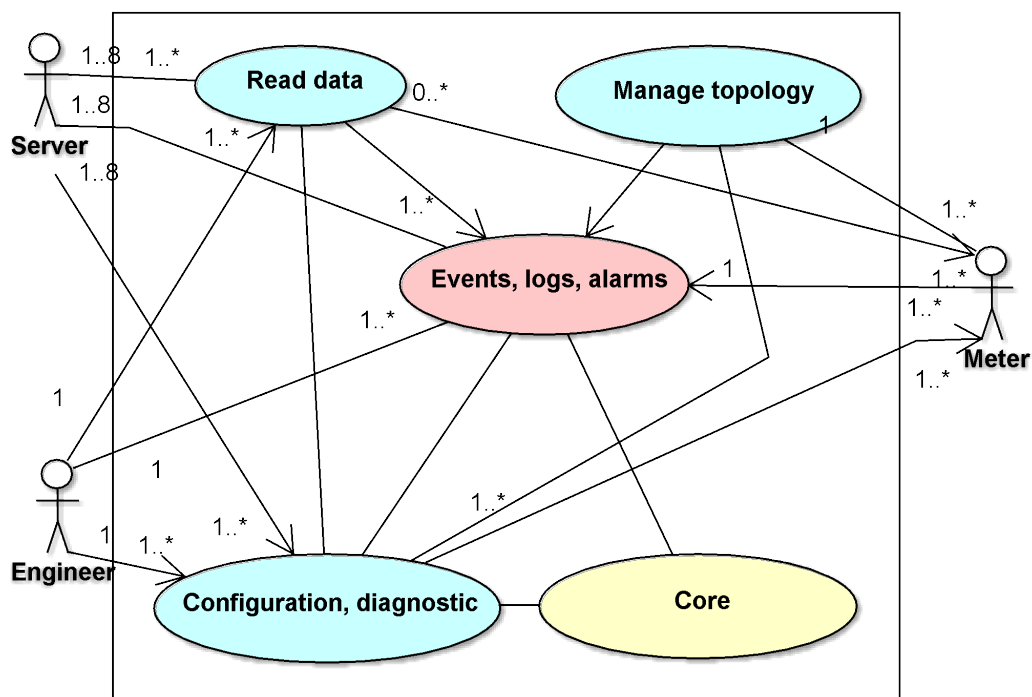
5 NÁVRH SOFTWARE

5.1 Případy užití

V této kapitole budou znázorněny základní služby, které DC poskytuje a interakce s okolními systémy. Ke znázornění byl využit diagram užití (use case diagram) standardu UML [10] vytvořený v programu ArgoUML [21]. Pro lepší znázornění prezentované problematiky jsem se rozhodl nedržet zcela standardu a upravil jsem význam některých prvků:

- čára zakončená šipkou - objekt, ke kterému míří šipka je řízen objektem z něhož šipka vychází
- přerušovaná čára - stejně jako plná čára, ale jedná se o podmíněnou situaci, ke které může a nemusí docházet

5.1.1 Celkový systém



Obr. 5.1: Diagram užití celkového systému

Aktéři

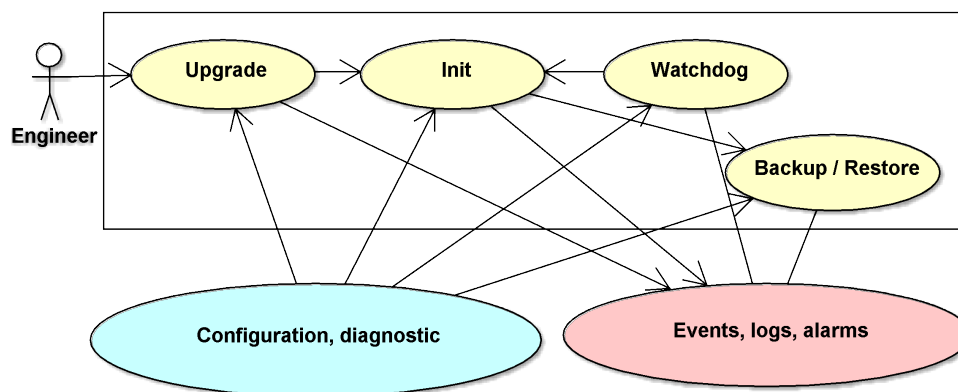
- Server - Server distribuční společnosti, který přijímá data z DC a odesílá příkazy směrem k DC. Počet serverů byl stanoven na 1 až 8, jelikož DC může přijímat data z elektroměrů různých distributorů a také různých distribuovaných médií, např. elektřina, voda, plyn. Tyto společnosti jsou často konkurenty a lze si jen těžko představit, že by data putovala na společný server. Proto budou tato data rozdělena již na úrovni DC a budou putovat směrem k příslušným serverům. Komunikaci může řídit server, kdy záleží čistě na něm kdy si vyžádá data z koncentrátoru nebo naopak koncentrátor, který budto odešle data v předem dohodnutou dobu nebo po nashromáždění dostatečně velkého balíku dat. Stejně tak může DC některá nebo všechna data odesílat ihned.
- Engineer - Technik provádějící diagnostiku na místě nebo vzdáleně přes diagnostický server. Přístup může být realizován z PC, PDA, tabletu pomocí sériové linky, rozhraní IEC62056-21 nebo přes diagnostický server pomocí sítě Ethernet. Některá data jako alarmy budou odesílána okamžitě, část dat bude na vyžádání.
- Meter - Koncové zařízení z něhož jsou vyčítána data a které vykonává příkazy. Nejčastěji elektroměr. Komunikace je typu master-slave, kde masterem je koncentrátor.

Služby

- Core (Jádro) - Část zodpovídající za chod všech součástí spojených s DC, upgrade systému a monitorování stavu systému.
- Read data (Vyčítání dat) - Část zodpovědná za vyčítání dat z měřičů a přesun těchto dat na server.
- Manage topology (Správa topologie) - Část zodpovědná za tvorbu komunikačních kanálů mezi DC a měřiči. Vyčítání je možné z mnoha měřičů na různých rozhraních.
- Configuration, diagnostic (Konfigurace a diagnostika) - Část umožňující konfiguraci a diagnostiku jak samotného DC, tak měřičů.
- Events, logs, alarms (Události, logy, alarmy) - Část zodpovědná za zachycení asynchronních události z měřičů a DC a jejich zpracování.

5.1.2 Jádro (Core)

Část zodpovídající za chod všech součástí spojených s DC, upgrade systému a monitorování stavu systému.



Obr. 5.2: Diagram užití částí jádra

Služby

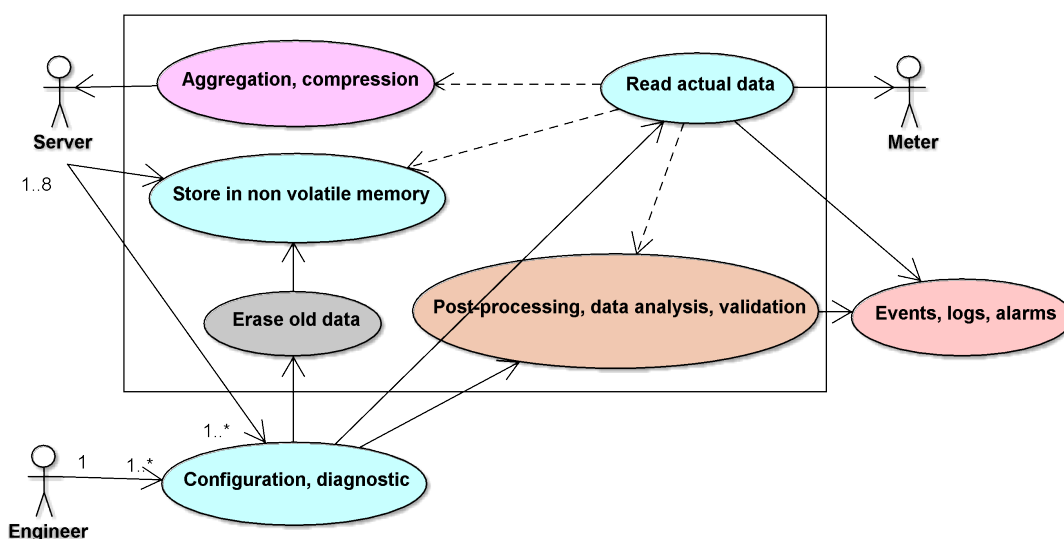
- Upgrade - Část zodpovídající za upgrade jednotlivých částí DC. Ke spuštění této služby dojde v případě, že dorazí požadavek z modulu diagnostiky a konfigurace nebo při startu budou nalezeny speciální soubory nebo balíčky. Nejdříve bude ověřeno zda soubory nejsou poškozeny, následně dojde ke kontrole zda jsou kompatibilní s ostatními součástmi systému, následně dojde k zálohování současné verze systému a bude proveden samotný upgrade. Následovat bude kontrola zda upgrade proběhl úspěšně a vymazání dočasných souborů, které již nejsou potřeba.
- Init - Služba, která zodpovídá za korektní spuštění všech součástí systému ve specifikovaném pořadí a neustálou kontrolu zda v nějaké součásti nedošlo k chybě a k jejímu zhroucení, pokud tato událost nastane je nutno tuto součást restartovat. Zastavení, restartování, spuštění jednotlivých součástí bude možné zadáním příkazu přes diagnostický a konfigurační modul. Služba také zodpovídá za korektní ukončení všech součástí při vypínání nebo restartu celého systému. Při startu je provedena obnova zálohovaných dat a kontrola verzí jednotlivých součástí.
- Backup / Restore (Záloha / Obnova) - Při důležitých změnách dat, v pravidelných intervalech nebo při vypínání systému může být zavolána tato služba, která provede zálohu požadované součásti v případě, že by došlo k pádu celého

systemu, například v důsledku výpadku napájení. Při startu systému bude zavolána tato služba, aby provedla obnovu těchto dat ze zálohy.

- Watchdog - Služba neustále kontrolující využití systémových prostředků, důležitých asynchronních událostí, která v případě problému tuto skutečnost zaznamená a provede restart problematické součástí nebo celého zařízení.

5.1.3 Vyčítání dat (Read data)

Část zodpovědná za vyčítání dat z měřičů a přesun těchto dat na server.



Obr. 5.3: Diagram užití částí vyčítání dat

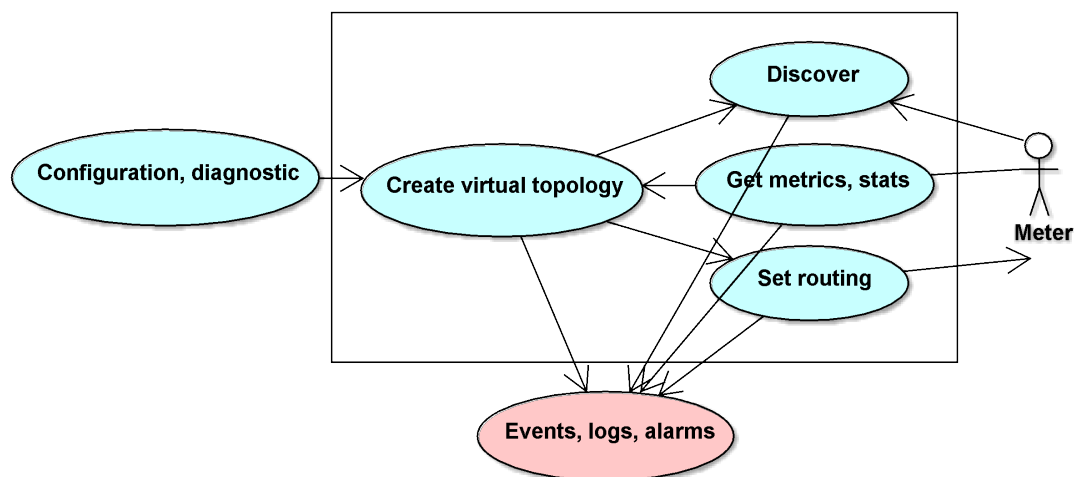
Služby

- Read actual data (Vyčítání aktuálních dat) - Vyčítání dat o spotřebě, stavu sítě a jiných užitečných dat z měřičů. Vyčítání probíhá buďto v pravidelných intervalech nebo pokud jsou v měřiči nevyčtená data. Vyčítání je možné jak od nejstarších dat k nejnovějším, tak i od nejnovějších k nejstarším. Služba se autonomně rozhoduje jaký typ dat z kterého měřiče vyčte nejdříve, toto chování je možné parametrizovat.
- Aggregation, compression (Agregace, komprese) - Po vyčtení dat z měřičů nemusí být data okamžitě odeslána na server, ale je možné je shromáždit na DC a odesílat na server ve stanovených časech nebo periodách, případně na vyžádání serveru. Díky tomu je možné větší objemy dat účinně komprimovat.

- Store in non volatile memory (Uložení v nevolatilní paměti) - Určitá data mohou být uložena pouze v nevolatilní paměti zařízení a dostupná serveru pouze na vyžádání.
- Erase old data (Smazání starých dat) - Jelikož má zařízení omezenou paměť, je nutné stará data průběžně mazat podle nakonfigurované strategie. Například postupné řídnutí, kdy se s postupem do minulosti zvětšuje vzorkovací perioda.
- Post-processing, data analysis, validation (Zpracování dat) - DC může data sám vyhodnocovat, kontrolovat a analyzovat a pouze v případě, že dojde k nestandardnímu stavu informuje server.

5.1.4 Správa topologie (Manage topology)

Část zodpovědná za tvorbu komunikačních kanálů mezi DC a měřiči. Vyčítání je možné z mnoha měřičů na různých rozhraních.



Obr. 5.4: Diagram užití částí správa topologie

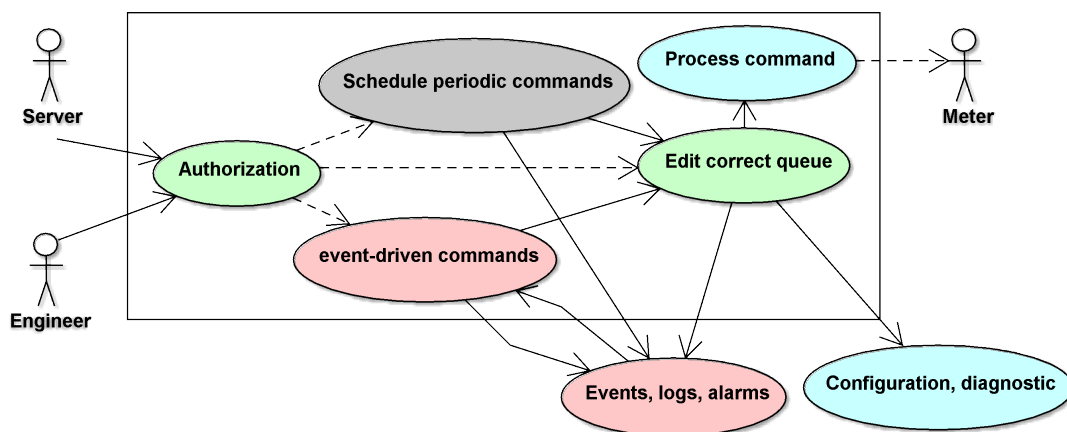
Služby

- Discover (Objevování) - Objevování nových elektroměrů, pokud je aktivní při hledání DC, jsou využity údaje z odhadu virtuální topologie, ke zvolení vhodného segmentu sítě, ve kterém bude provedeno hledání.
- Get metrics, stats (Získávání metrik, statistických dat) - získání údajů užitečných pro odhad topologie. Mezi tyto údaje lze zařadit kvalitu signálu mezi jednotlivými měřiči, jejich GPS polohu, případně další údaje a jejich časový průběh.

- Create virtual topology (Odhad virtuální topologie) - Na základně získaných metrik a případně jiných údajů lze provést odhad skutečné topologie a na základě tohoto odhadu najít nejlepší cesty k jednotlivým měřičům. Tvorbu odhadu topologie je možné ovlivnit také z konfiguračního modulu, kde je možné zadat informace, které máme o topologii z jiných zdrojů, jako jsou topologické mapy energetických společností. V tom případě můžeme zadat část nebo celou topologii měřičů ručně.
- Set routig (Nastavení směrování a opakování) - Služba, která zajišťuje nastavení směrovačů, opakováčů v síti. Kromě nastavení automatických záznamů dokáže nastavovat také manuální záznamy.

5.1.5 Konfigurace a diagnostika (Configuration, diagnostic)

Část umožňující konfiguraci a diagnostiku jak samotného DC tak měřičů.



Obr. 5.5: Diagram užití částí konfigurace a diagnostika

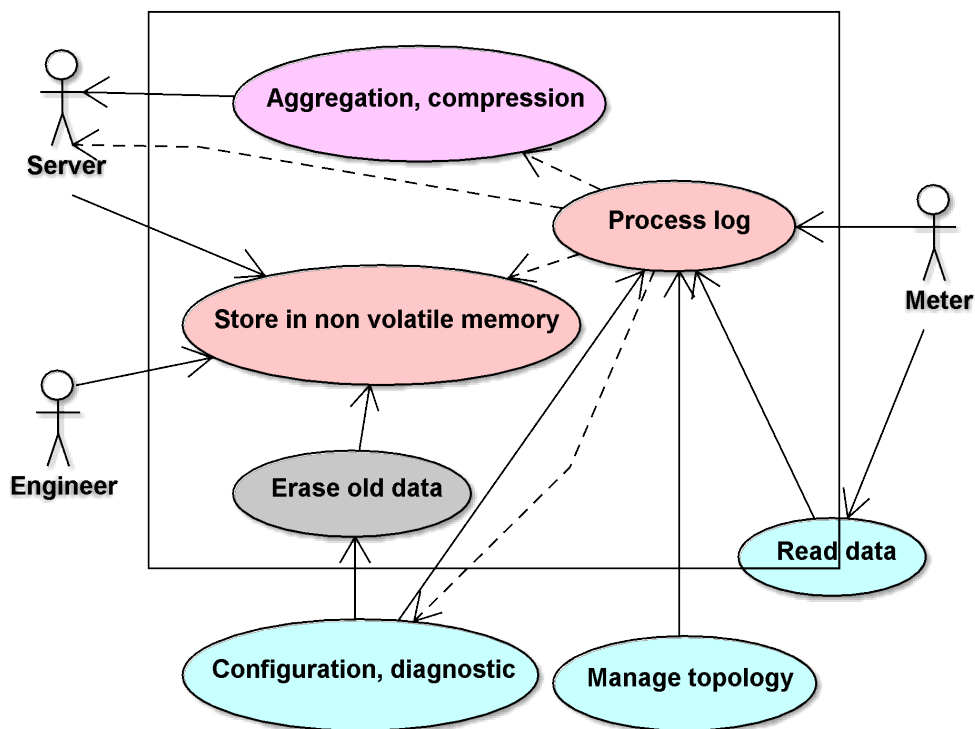
Služby

- Authorization (Autorizace) - Vzhledem k tomu, že k DC může být připojeno více serverů, je nutné zajistit aby například server vodárenské společnosti nemohl přepnout tarif na elektroměru. Z tohoto důvodu je nejdříve nutné ověřit zda má příslušný server možnost provést tento příkaz.
- Schedule periodic commands (Naplánování periodických příkazů) - Zadání příkazů, které se budou vykonávat v předem stanovenou dobu nebo s pravidelnou periodou. Kromě orientace podle data by měl systém zvládat orientaci podle dnů v týdnu a svátků.

- Event-driven commands (Nastavení příkazu řízeného událostmi) - Zadání příkazů, které se vykonají v případě, že nastane určitá asynchronní událost.
- Edit correct queue (Změna konkrétní fronty) - Přidání příkazu do příslušné fronty může provést přímo server nebo jedna z předchozích dvou služeb. Server může navíc měnit prioritu příkazu, měnit jejich pozici nebo je z fronty odstranit.
- Process command (Vykonání příkazu) - Vykonat příkaz může přímo koncentrátor nebo jej odešle na příslušné rozhraní a vykoná ho příslušný měřič.

5.1.6 Události, logy, alarmy (Events, logs, alarms)

Část zodpovědná za zachycení asynchronních události z měřičů a DC a jejich zpracování.



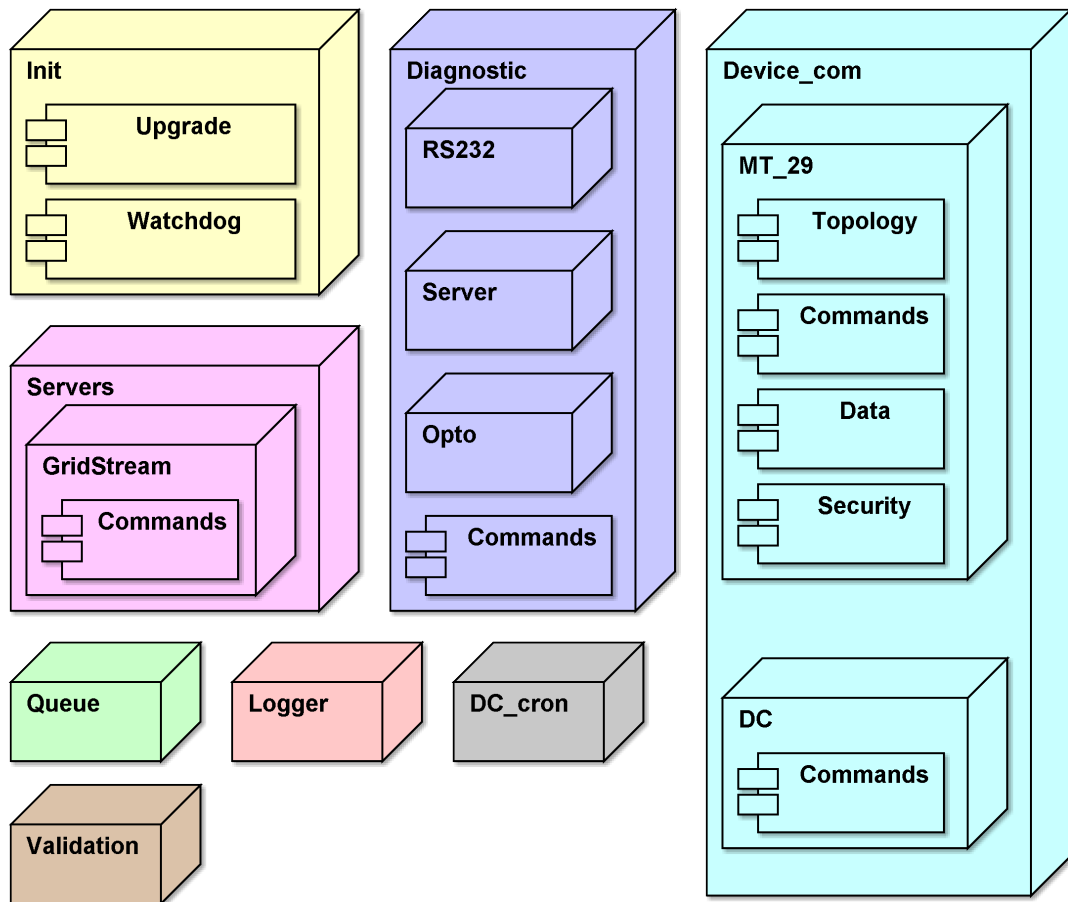
Obr. 5.6: Diagram užití částí události, logy, alarmy

Služby

- Process log (Zpracování logu) - Služba přijímá logy z DC nebo měřičů a rozhodne jakým způsobem je zpracovat na základě konfigurace a diagnostiky. Log může být odeslán okamžitě na server, uschován na DC a dostupný na vyžádání, odeslán ve větším balíku s daty nebo odeslán technikovi případně speciálnímu serveru.
- Aggregation, compression (Agregace, komprese) - Po vyčtení dat z měřičů nemusí být data okamžitě odeslána na server, ale je možné je shromáždit na DC a odesílat na server ve stanovených časech nebo periodách, případně na vyžádání serveru. Díky tomu je možné větší objemy dat účinně komprimovat.
- Store in non volatile memory (Uložení v nevolatilní paměti) - Určitá data mohou být uložena pouze v nevolatilní paměti zařízení a dostupná serveru pouze na vyžádání.
- Erase old data (Smazání starých dat) - Jelikož má zařízení omezenou paměť, je nutné stará data průběžně mazat podle nakonfigurované strategie. Například postupné řídnutí, kdy se s postupem do minulosti zvětšuje vzorkovací perioda.

5.2 Rozdělení funkčnosti

Cílem této kapitoly je rozdělit jednotlivé služby z kapitoly 5.1 Případy užití, mezi jednotlivé programy. K lepší orientaci bylo zvoleno barevné rozlišení jednotlivých programů, které koresponduje s barevným označením ve schématech z kapitoly 5.1 Případy užití. Jako krychle jsou označeny programy, jako obdélníky sdílené knihovny vlastní výroby.



Obr. 5.7: Seznam programů

5.2.1 Init

Program Init představuje jádro celého systému. Mezi jeho hlavní úlohy patří ověření integrity systému, start, ukončení a hlídání programů. Dále provádí zálohy databází a jejich případnou obnovu. V neposlední řadě se stará o upgrade celého zařízení. Základem je program Init, který má dynamicky přilinkovanou knihovnu s třídami pro upgrade systému a kontrolu běhu celého systému a jednotlivých programů. Program Init může dynamicky zavést jednotlivé knihovny, které jsou schopny zpracovat požadované příkazy, příkladem může být požadavek na restart celého systému nebo ukončení nějakého programu.

5.2.2 Servers

Skupina programů zajišťujících komunikaci s jednotlivými nadřazenými systémy. Pro každý nadřazený systém je zde jeden program. Minimální počet programů jsou 2. Jeden pro nadřazený systém příslušné společnosti a jeden pro diagnostický a konfigurační systém. Programy mohou plnit úlohu jak klienta tak serveru pro odesílání dat a příjem příkazů. Komunikace s jednotlivými nadřazenými systémy může probíhat ve stejných nebo odlišných časových okamžicích. Program dynamicky za běhu zavádí knihovny s příslušnými příkazy, které zpracovává.

5.2.3 Schedulers

Jedná se o soubor n procesů pro n rozhraní směřujících ke zdrojům dat. Jako příklad můžeme uvést PLC rozhraní, GPRS rozhraní, RS-485, které budou mít každé svůj plánovač. Plánovač pro aktuálně prováděné úkoly (tasks) vytvoří samostatná vlákna, která budou přistupovat k příslušnému rozhraní. V případě rozhraní, které neumožňují zaslání více zpráv najednou např. PLC, bude přístup k rozhraní chráněn mutexem a současně nepoběží víc jak 2 až 3 vlákna. V případě kdy by běželo vždy pouze jedno vlákno, by docházelo ke ztrátám kdy by vlákno zjišťovalo co má odeslat na linku a po přijetí dat by tato data vlákno zpracovávalo. V těchto okamžicích by bylo rozhraní nevyužité. Proto i v případě rozhraní s 1 přístupem je použito více vláken. V případě rozhraní s více souběžnými přístupy např. GPRS, je možné vytvořit libovolný počet vláken a není nutné je synchronizovat pomocí mutexu.

Navíc zde může nastat situace, že bude nutné synchronizovat více rozhraní mezi sebou. Tato situace nastane v případě, že k jednomu koncentrátoru budou připojeny dvě nezávislé PLC sítě, u kterých ovšem bude docházet k vzájemnému rušení. Z tohoto důvodu bude nutné synchronizovat i tyto plánovače mezi sebou. K synchronizaci bude buď použito prostředků operačního systému nebo bude problém řešen nadřazeným plánovačem, který bude plánovat činnost podřízených plánovačů.

Dále je zde například zpracovávání příkazů týkajících se pouze data koncentrátoru. Součástí jednotlivých programů jsou závislé od jeho typu. Na obrázku 5.7 je uveden příklad pro komunikaci s elektroměry s komunikací na protokolu MT29, která dynamicky zavádí knihovny pro tvorbu a údržbu topologie, vyčítání dat a šifrování komunikace. Dále jsou za běhu dynamicky zaváděny příkazy pro provedení jednotlivých příkazů týkajících se metrů. Program, který zpracovává příkazy týkající se DC, zavádí za běhu dynamické knihovny s příkazy, které zpracovává.

5.2.4 Diagnostic

Sada programů pro konfiguraci a diagnostiku DC. Součástí jsou programy pro jednotlivé komunikační sběrnice, pomocí kterých může konfigurace probíhat. Na obrázku 5.7 jsou uvedeny jako příklady RS232, Opto rozhraní, TCP-IP spojení. Dále zde mohou být další podpůrné programy. Jednotlivé programy mohou zavádět další dynamické knihovny například knihovny pro vykonání určitých příkazů.

5.2.5 Queue

Program pro práci s frontou příkazů. Program zajistí autorizaci příkazu a jeho zařazení do fronty. Program dokáže pracovat s příkazy ve frontě. V případě příkazu na metr, mění parametry metrů v cache paměti na DC.

5.2.6 Logger

Program přijme zprávu o nastalé události a pomocí pravidel uložených v databázi se rozhodne zda událost vyhodnotit jako alarm, který se má okamžitě odeslat nadřazenému systému nebo jej zařadit do logů uchovaných v paměti DC.

5.2.7 DC cron

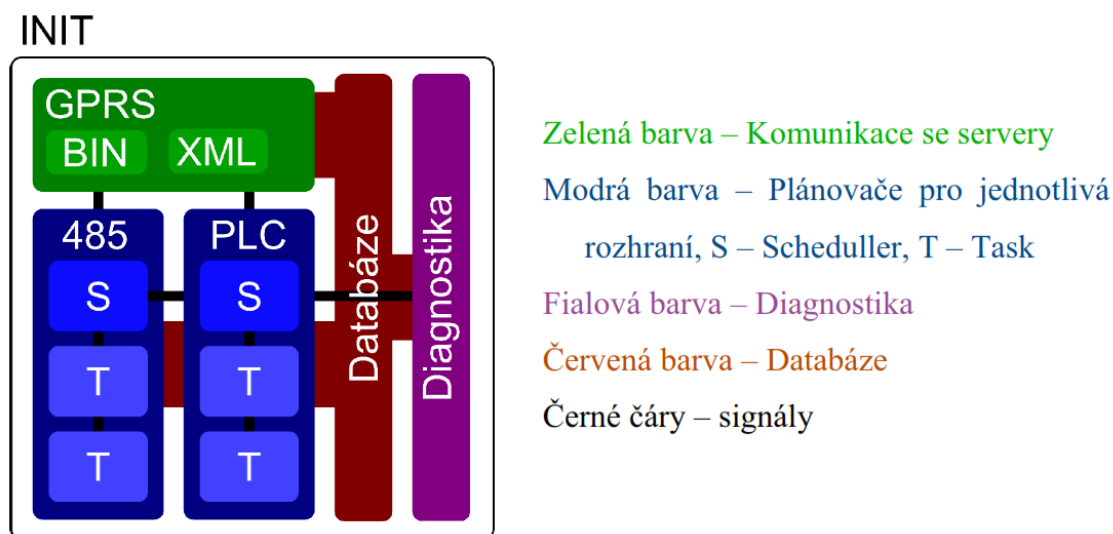
Jedná se o obdobu programu Cron ze systému Unix, který spouští určité činnosti ve specifikovaných okamžicích. Je ale upravený pro potřeby DC. Základem je použití databáze místo konfiguračního souboru a místo skriptů jsou zařazovány do fronty příkazy nebo skupiny příkazů. Kromě základního definování času podle specifikací Cronu je doplněna podpora pro pracovní dny a víkendy a svátky.

5.2.8 Validation

Program nebo sada programů, které vyhodnocují nashromážděná data z měřičů a pokud dojde k určité nesrovnalosti, jako je možnost černého odběru nebo poklesu napětí, je okamžitě informován nadřazený systém.

5.2.9 Model základního rozložení funkčnosti DC

Do modelu jsou zahrnuty jen nejdůležitější programy, nejsou zde podpůrné programy.



Obr. 5.8: Grafické znázornění rozdělení funkčnosti DC

5.2.10 Databáze a synchronizace

Veškerá podstatná data všech procesů a vláken budou uložena v databázi. K této databázi budou mít přístup všechny aplikace. Značně se tak usnadňuje správa sdílené paměti, kdy je již zajištěná dokonalá synchronizace přístupu k těmto datům. Čtení lze provádět paralelně, zatímco pro zápis je zajištěn unikátní přístup. Dále je možné využít transakce, vyhledávání a třídění dat podle standardu SQL. Jediným synchronizačním prostředkem mezi procesy, o který se musí starat programátor budou zprávy systému MPI, kdy si procesy dají vědět, že v databázi došlo ke změně dat.

Ve skutečnosti se nebude jednat o jednu databázi, ale o soubor více databází. Konfigurační data budou uložena v paměti FLASH a průběžná data budou uložena v databázi umístěné v paměti RAM. Ta bude v pravidelných intervalech a při výpadku napájení zálohována do paměti FLASH.

5.3 Databáze

Jako úložiště dat by bylo možné použít struktury v paměti RAM. Vzhledem k tomu, že zařízení bude zpracovávat velké množství dat, nad kterými budou probíhat operace třídění, filtrování, sdružování atd. a dále budou tyto data sdílena mezi více procesy, bylo rozhodnuto o požití databáze.

Databáze nebo přesněji sada databází slouží k uchování všech parametrů zařízení a různých informací o systému. Díky tomu se podstatně zjednoduší sdílení těchto informací, nejen mezi samotnými programy, ale především mezi diagnostickou a konfigurační nadstavbou. Data, která informují o stavu zařízení a často se mění jsou uchována v paměti RAM.

5.3.1 Volba typu databáze

Vzhledem k poměrně malým požadavkům na výkon a funkce databáze a naopak snaze o minimalizaci nákladů bylo vybíráno z open source variant. Uvažovány byly Firebird, SQLite, Postgree, MySQL. MySQL ačkoliv poskytuje vysoký výkon a velmi dobrou funkční výbavu, byla vyřazena z důvodu nevyhovující licenční politiky v oblasti embedded zařízení. Postgree databáze je databáze určená pro větší aplikace na serverech, a příliš se nehodí pro použití v embedded zařízeních. Uvažovány byly databáze SQLite [9] což je poměrně zjednodušená databáze bez serveru, uložená v jednom souboru a databáze Firebird [8], která je open source pokračováním databáze InterBase od společnosti Borland. Databáze Firebird byla uvažována zejména vzhledem k předchozím zkušenostem a vysokému výkonu.

5.3.2 Srovnání výkonu

Aby bylo možné se rozhodnout mezi databázemi SQLite a Firebird, byly provedeny testy výkonu těchto databází. Testování probíhalo na 2 platformách 386 a 486 z koprocosem, jednalo se o desky popsané v kapitole 3.4.7 týkající se realizace procesorových desek.

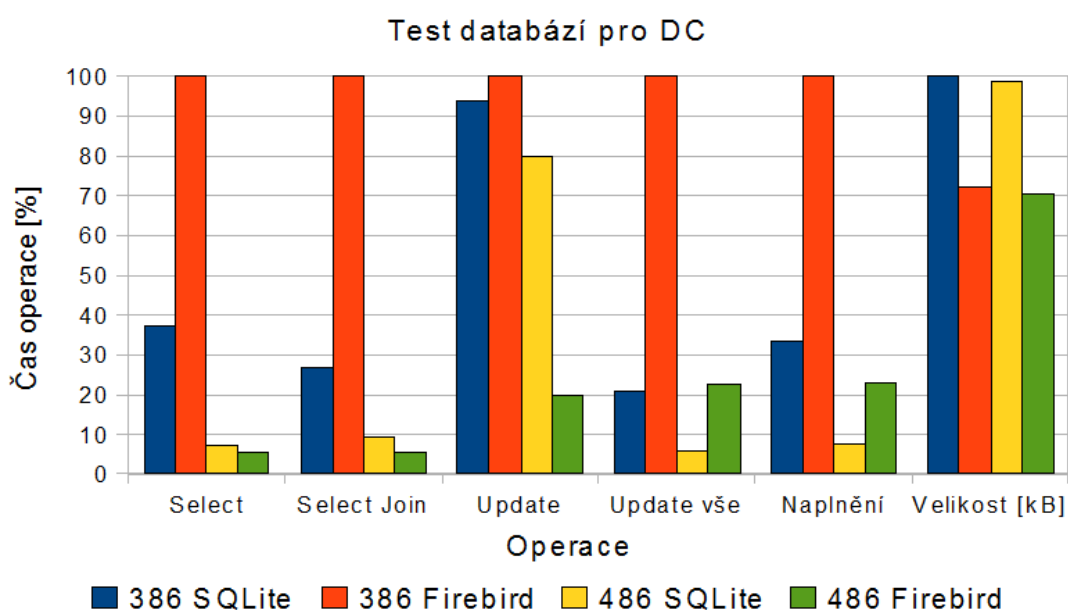
Vysvětlení jednotlivých operací jež byly testovány:

- Select je výběr jednoho řádku z 200 seřazených podle neindexované hodnoty,
- Select Join je výběr jednoho řádku ze dvou spojených tabulek 200 a 19900 hodnot.
- Update je úprava řádku vybraného podle klíče
- Update vše je úprava 19900 řádků
- Naplnění je doba naplnění databáze jednotlivými inserty
- Velikost je velikost výsledné databáze

Jednotlivé operace nemají takový význam jako jejich srovnání

	[s]	Select	Select Join	Update	Update vše	Naplnění
386	SQLite	0,07	3,3	0,05	4,1	160
	Firebird	0,2	12,4	0,05	20	480
486	SQLite	0,01	1,17	0,04	1,14	36,6
	Firebird	0,01	0,67	0,01	4,49	109

Tab. 5.1: Srovnání výkonu databází



Obr. 5.9: Srovnání výkonu databází

Z grafu 5.9 je patrné, že SQLite dosahuje více vyrovnané výsledky na obou platformách. Firebird dosahoval podstatně horší výkon než SQLite na platformě 386, v případě platformy 486 byly výsledky srovnatelné. V případě platformy 486 byl Firebird lepší co se týče všech druhů čtení, úpravy jednoho řádku a jeho databáze zabírala přibližně 70% místa co SQLite.

5.3.3 Srovnání nabízených funkcí

Dále je nutné srovnat nabízené funkcionality těchto databází.

SQLite [9]	Firebird [8]
+ malé nároky na systém	+ velmi rychlé transakce
+ umožňuje propojení databází	+ TCP-IP server
+ možnost DB v RAM	+ propracované zálohy
- velmi pomalé transakce	- nepodporuje unsigned datové typy

Tab. 5.2: Srovnání funkcí databází

U SQLite lze ocenit zejména malé nároky na celkový systém a možnost propojení databází. Díky tomu je možné umístit jednu databázi do paměti FLASH a druhou do paměti RAM a následně je propojit tak, že se s nimi pracuje jako s jednou, což je velmi pohodlné. SQLite umožňuje umístit databázi přímo do zásobníku, nevýhodou tohoto řešení je, že nelze použít sdílenou paměť a přistupovat k této databázi z více procesů. Další nevýhodou je absence TCP-IP serveru, který ale není potřeba a lze použít server třetích stran, například pro účely diagnostiky databáze. V případě Firebirdu lze ocenit především rychlé transakce díky multigenerační architektuře a možnost provádění záloh za běhu. Velkou nevýhodou v případě Firebirdu je absence bez znaménkových datových typů, které se velmi často používají v měřících a bylo by nutné vždy provádět konverzi. Další výhodou Firebirdu je větší počet datových typů, díky čemuž je schopen lépe optimalizovat velikost samotné databáze, což se projevilo v předchozím srovnání výkonů.

5.3.4 Závěr

Vzhledem k malým požadavkům na procesor, možnosti propojení databází a podpoře bez znaménkových datových typů byla zvolena databáze SQLite.

5.3.5 Datový model

Součástí návrhu je i datový model databáze. V současné době obsahuje tabulky pro uložení parametrů a frontu příkazů. Podrobně zdokumentovaný datový model včetně diagramu vazeb lze nalézt v příloze A, obdélníky s názvem FLASH RAM znamenají, že příslušná část databáze se nachází jak v paměti FLASH tak v paměti RAM. Neoznačené části se nacházejí v paměti FLASH.

Parametry

Parametry jsou označovány číselně a umožňují nekonečné zanoření oddělené tečkou, např 1.25.36.178, díky tomu získáváme kompatibilitu s SNMP protokolem a navíc je můžeme uspořádat do logických celků. V databázi je tato hodnota uložena jako cesta k rodiči typu string 1.25.36 a číslo parametru v příslušné kategorii zde 178, díky tomu se zjednoduší operace při vyhledávání a procházení parametrů. Dále je zde informace o datovém typu parametru, o jeho umístění v paměti FLASH nebo RAM, zda lze do parametru zapisovat. Dále parametr obsahuje název, který lze použít v kódu. Pokud není parametr definován použije se výchozí hodnota, což je také parametr. Označení výchozího parametru je součástí definice parametru. Každý parametr může obsahovat hodnotu typu celé číslo, číslo s plovoucí řádovou čárkou a řetězec znaků, více standardních datových typů SQLite stejně nepodporuje. Dále je ještě uchováno datum poslední změny a předchozí hodnota parametru. V případě, že se parametr týká metru, což je uvedeno v jeho definici, může obsahovat více hodnot, jednu pro každý metr nebo skupinu metrů. Dále je zde popis parametrů v jednotlivých jazycích, který lze využít pro diagnostické a konfigurační nástroje nebo pro generování dokumentace.

Pro pohodlné vkládání definic parametrů do databáze byl vytvořen nástroj s formulářem v programu OpenOffice Base, který je vidět na obrázku 5.10

id	parentTree	tree	dataType	defaultValue	meter	writable	destination	name
25	1.4.0.9	1	0		<input type="checkbox"/>	<input type="checkbox"/>		program
26	1.4.0.9.1	1	1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	type
27	1.4.0.9.1	2	3		<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	hash
28	1.4.0.9.1	3	1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	autorun
29	1.4.0.9.1	4	1 default		<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	cpuLimit
30	1.4.0.9.1	5	1 default		<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	memoryLimit
31	1.4.0.9.1	6	1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	respawn
32	1.4.0.9.1	7	1		<input type="checkbox"/>	<input type="checkbox"/>	2	pid
33	1.4.0.9.1	8	1		<input type="checkbox"/>	<input type="checkbox"/>	2	mpi

Záznam 28 z 44 * (1)

Params_id	Languages_id	name	description
28	1	autorun	Pořadí, ve kterém se spustí příslušný program po startu DC. Pokud se program nemá spouštět po startu nastaví se pořadí na 0. Pokud má víc programu stejné pořadí rozhoduje jejich ID.

Obr. 5.10: Formulář pro definici parametrů v databázi

Seznam všech prozatím definovaných parametrů DC lze nalézt v příloze B.

Fronta příkazů

Ve frontě příkazů je příkaz reprezentován číslem příkazu a prioritou. Dále je zde informace v jakém stavu se příkaz nachází, kdy byl do fronty zařazen, kdy byl zpracován, pro jaký proces je příkaz určen, zda má čekat na nějaký předchozí příkaz až bude dokončený než bude vykonán. Dalším parametrem je nastavené chování pokud příkaz nelze provést, může nabývat jednu z těchto možností:

- Provést příkaz jakmile to bude možné
- Nastavení timeoutu na pevné datum a čas (1.2.2011 12:00)
- Nastavení timeoutu relativně (po 10 minutách)
- Pokusit se příkaz provést n krát po sobě, a pokud selže skončit chybou

Příkazy mohou být řazeny do skupin. Sdružené příkazy nebo také skupiny příkazu se zásadně neliší od těch jednotlivých, jediný rozdíl je v tom, že směrem k serveru se soubor těchto příkazů jeví jako jedna úloha, kterou server může pozastavit, smazat, naplánovat a zjistit přesný stav, ve kterém se úloha nachází, které její části byly splněny a které ne. Stejně tak může server pracovat s jednotlivými příkazy sdružené úlohy jednotlivě.

Speciálním případem sdružených příkazů jsou transakce, v transakcích mohou být použity pouze příkazy které transakce podporují, lze je tedy vrátit zpět nebo se vykonávají na měřících, které podporují transakce.

Každý příkaz může mít libovolný počet argumentů, jednoho ze tří typů: celé číslo, číslo s plovoucí řádovou čárkou a řetězec.

Fronty jsou ve skutečnosti dvě, jedna v paměti FLASH a druhá v paměti RAM. Podle typu příkazu nebo požadavku ze serveru se příkaz zařadí do jedné z front. Ve výsledku se tyto fronty spojí do jedné. Rozdíl je ten, že fronta ve FLASH paměti je chráněna proti výpadku napájení, zatímco časté a nepříliš důležité příkazy mohou být v paměti RAM, díky čemuž se nepoškozuje paměť typu FLASH a v případě výpadku napájení je server dotázán aby zaslal nevykonané příkazy znovu.

5.3.6 Ladění a diagnostika

Pro ladění a diagnostiku databáze lze využít program SQLite Maestro, který je dodáván s adaptérem pro přístup do databáze přes TCP-IP spojení. Jedná se PHP skript, který je nutné umístit tak aby byl dostupný přes webový server zařízení. Další možností je propojit souborové systémy, například protokolem SAMBA, který podporuje i MS Windows a k databázi přistupovat pomocí jednoho z volně dostupných nástrojů pro správu SQLite databáze.

5.4 Program Init

Základní popis programu Init lze nalézt v podkapitole 5.2.1 o rozdělení funkčnosti.

Vzhledem k tomu, že podrobný objektový návrh čítá více jak 70 stran, není přímou součástí této práce a lze jej najít na přiloženém DVD. V této dokumentaci lze nalézt seznam použitých objektů včetně podrobné dokumentace všech argumentů i funkcí. Diagramy tříd lze nalézt také v příloze C.

Implementace předpokládá jazyk C++ [6] [5] s použitím knihoven STL a Boost [19].

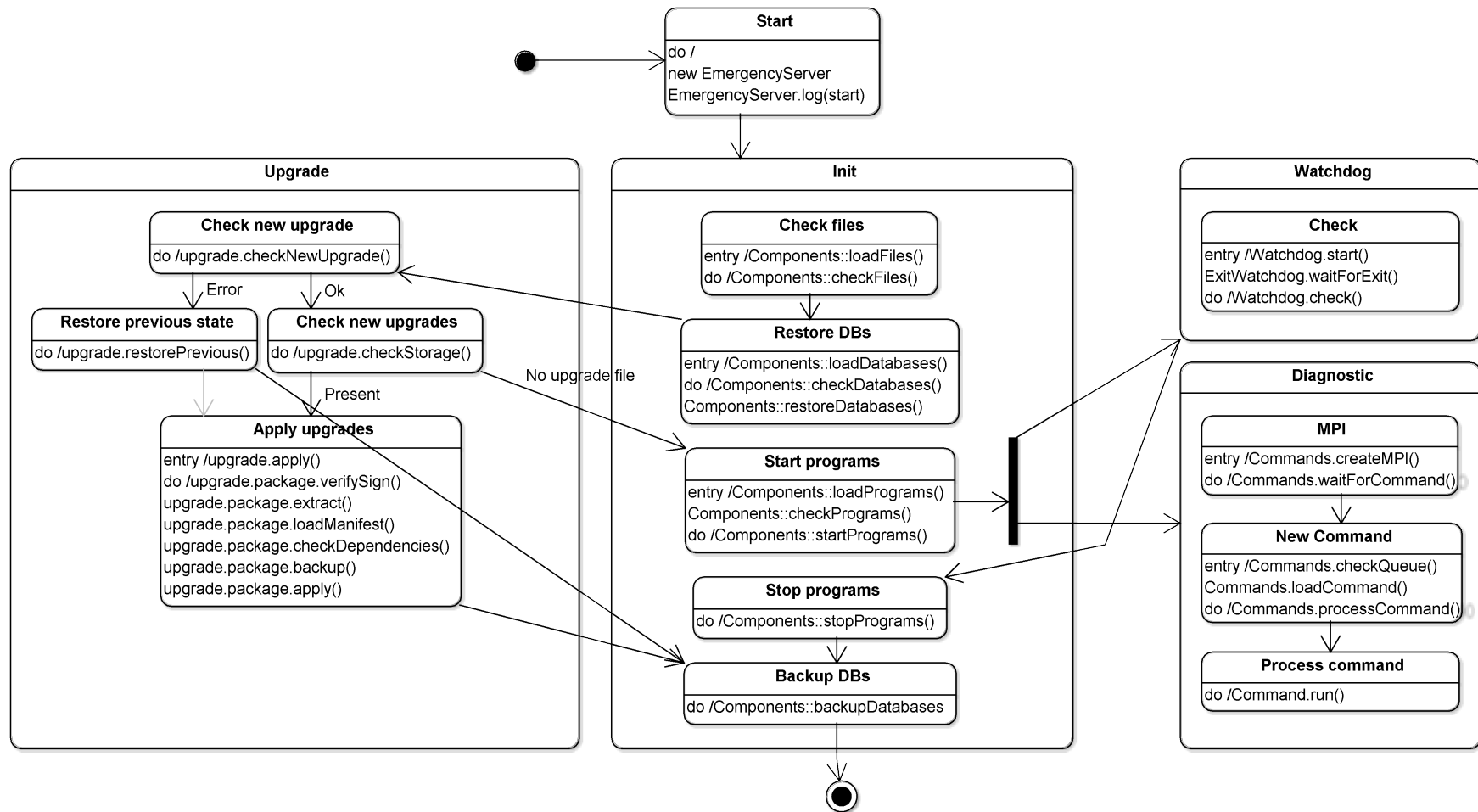
Schéma běhu programu na obrázku 5.11 je rozděleno do několika logických částí. Každá z těchto částí bude popsána v následujících podkapitolách.

5.4.1 Start

Při startu programu se vytvoří spojení s nouzovým serverem, který pouze přijímá zprávy o základním běhu DC. Jeho výhoda je, že se jedná o zcela primitivní komunikaci bez šifrování, tunelování a podobných prvků, které by mohly selhat. V tomto případě se odešle log o tom, že došlo ke startu celého zařízení.

5.4.2 Init

Je jádro celého programu, které zajišťuje běh všech programů a databázi. Nejdříve je zkontrolována integrita celého zařízení, závislosti mezi programy, následuje obnova databází. Pokud vše dopadne v pořádku, je spuštěn modul pro upgrade zařízení, popsáný v podkapitole 5.4.3. Pokud nedošlo k upgradu a zařízení není nutné restartovat, jsou spuštěny všechny potřebné programy. Dále je spuštěn modul s watchdogy popsáný v podkapitole 5.4.4 a paralelně s ním modul pro příjem příkazů popsáný v podkapitole 5.4.5. Pokud je požadavek na restart zařízení, uskuteční se pokus o regulérní ukončení všech programů, pokud selže jsou programy ukončeny násilným způsobem. Dále dojde k zálohování databází a restartu programu Init nebo celého zařízení.



Obr. 5.11: Schéma běhu programu Init

5.4.3 Upgrade

Nejdříve se provede kontrola zda v poslední době došlo k upgradu a zda nedochází k opakovaným chybám v důsledku onoho upgradu. Pokud by k chybám docházelo, dojde k vrácení programů do stavu před upgradem. Pokud chyba nenastala, provede se kontrola zda je přítomen balíček s novým upgradem. Pokud je balíček přítomen, provede se ověření podpisu veřejným klíčem. Pokud je podpis správný, provede se rozbalení balíčku a podle příložného manifestu se zkontrolují veškeré závislosti. Pokud jsou závislosti v pořádku, provede se záloha současného stavu systému a provede se upgrade, po kterém následuje záloha databází a restart systému. V případě že není přítomen soubor s balíčkem upgradu, pokračuje se ve spouštění programu dle popisu v podkapitole 5.4.2.

5.4.4 Watchdog

Program obsahuje sérií watchdogů, které hlídají jak samotné programy a databáze, tak i samotný systém. V nastavitelných periodických intervalech se kontroluje využití procesoru, paměti RAM, paměti FLASH a také se provádí záloha databází. Dále se kontroluje zda nedošlo ukončení programu v důsledku chyby nebo zda se program nezasekl v určitém místě. V případě že nastane problém dojde podle nastavení k restartu příslušné součásti nebo k restartu celého zařízení.

5.4.5 Diagnostic

Nejdříve je vytvořen mechanismus MPI pro komunikaci s ostatními programy. Následně je zahájeno čekání na zprávu o novém příkazu. Po přijetí zprávy je provedena kontrola fronty příkazů a pokud je vše v pořádku, provede se zavedení dynamické knihovny s konkrétním příkazem, pokud již v minulosti nebyla zavedena a provede se příslušný příkaz.

5.5 Programy pro komunikaci s nadřazenými systémy

Základní popis skupiny programů týkajících se TCP-IP komunikace lze nalézt v kapitole Rozdělení funkčnosti v podkapitole 5.2.2.

Pro komunikaci s jednotlivými nadřazenými systémy je v DC vždy samostatná aplikace. Ostatní součástí DC zašlou zprávu pomocí MPI, vždy když se objeví nové data, log, nebo alarm. Aplikace komunikující s nadřazeným serverem se sama rozhodne jak s touto zprávou naloží. V případě, že tyto data zpracovala, indikuje tuto skutečnost v příznaku u těchto dat. Pokud jsou všechny potřebné příznaky aktivní, jsou tato data zpracována a můžou se smazat. Pro různé typy dat lze využít různé strategie odesílání dat. Například data z měřičů jsou odesílány hromadně o půlnoci. Alarmy jsou odesílány okamžitě, část logů je odesílána s daty a část je uchována po určitou dobu na DC a jsou dostupné pouze na vyžádání. Možné scénáře pro odesílání dat do nadřazeného systému:

- Data jsou odeslána na server okamžitě jak jsou vyčtená z měřičů.
- Data jsou ponechána v paměti RAM dokud si je nevyžádá server.
- Data jsou ponechána v paměti RAM a v určitý čas jsou hromadně odeslána na server. Díky tomu je možné odesílat data v době kdy je GPRS síť minimálně vytížená a dosáhnout tak výhodnějších tarifů od operátorů.
- Data jsou ponechána trvale v paměti FLASH nebo připojeném SSD nebo HDD a poskytována serveru na vyžádání nebo poskytována pomocí zabudované prezentační vrstvy. Toto řešení je vhodné pro velmi malé zákazníky s malým počtem měřičů.

Primárně je použit proprietární binární protokol optimalizovaný pro minimální tok dat a vysokou latenci GPRS spojení, přičemž součástí nadřazeného systému může být adaptér, který převede tento protokol na nějakou více standardizovanou verzi např. DSMR P3.2 [20]. Nedá se vyloučit i nativní použití tohoto standardu pro aplikace s kvalitním TCP-IP spojením, jako například drátové spojení Ethernet. Každý nadřazený systém může používat jinou modifikaci, či verzi příslušného protokolu nebo i zcela odlišný protokol.

Při navazování komunikace s nadřazeným serverem dochází ke kontrole autenticity, pomocí kryptografického algoritmu AES. Na DC jsou nahrány veřejné klíče serverů, se kterými může DC komunikovat. Nové klíče můžou přidat pouze již ověřené servery.

5.6 Program pro komunikaci s měřiči na protokolu MT29

Jedná se o jeden z plánovačů popsaných v podkapitole 5.2.3, určený pro plánování činnosti na PLC síti s protokolem MT29. Jednotlivé části plánovače je možné deaktivovat. Seznam jednotlivých úkolů seřazených podle priority:

- Synchronizace v rámci mřížové sítě
 - Routing1
 - * Vyhodnocení příčiny nedostupnosti modemu
 - * Ověření dostupnosti nedostupných routerů
 - * Aktualizace cest v routerech
 - Vyčítání dat
 - * Vyčítání události
 - * Aktualizace indexů
 - * Vyčítání odečtových záznamů
 - * Vyčítání profilů
 - * Vyčítání napětí
 - * ...
 - Diagnostika
 - * Cachování parametrů z elektroměrů
 - * Upgrade Firmware
 - Routing2
 - * Identifikace
 - * Zjišťování metrik

5.6.1 Příkazy ze serveru

Mezi každý bod může být umístěn příkaz z horní vrstvy v závislosti podle jeho priority v rámci plánovače. V rámci jedné úrovně plánovače jsou příkazy odlišeny vedlejší prioritou příkazu. Před vykonáním příkazu se bude kontrolovat zda je modem dostupný, pouze v případě kdy je použita nejvyšší priorita bude DC posílat příkaz bez ohledu na dostupnost elektroměru. Toto je, ale poměrně nebezpečné, jelikož v případě nastavení, že příkaz musí být proveden a modem je dlouhodobě nedostupný bude DC odesílat neustále pouze tento příkaz a nebude vykonávat nic jiného. Proto by příkaz s největší prioritou měl být použit ve zcela výjimečných situacích.

Vzhledem k rušení na síti nemusí být možné provést příkaz okamžitě. Proto každý příkaz bude mít jeden z těchto režimů, na základě kterých DC rozhodne co dělat v případě, že je elektroměr nedostupný:

- Provést příkaz jakmile to bude možné
- Nastavení timeoutu na pevné datum a čas (1.2.2011 12:00)
- Nastavení timeoutu relativně (po 10 minutách)
- Pokusit se příkaz provést n krát po sobě, a pokud selže skončit chybou

Sdružené příkazy

Sdružené příkazy se zásadně neliší od těch jednotlivých. Jediný rozdíl je v tom, že směrem k serveru se soubor těchto příkazů jeví jako jedna úloha, kterou server může pozastavit, smazat, naplánovat a zjistit přesný stav, ve kterém se úloha nachází, které její části byly splněny a které ne. Stejně tak může server pracovat s jednotlivými příkazy sdružené úlohy jednotlivě.

Broadcastové a multicastové příkazy

Zde jsou dvě varianty těchto příkazů:

- Skutečný broadcast nebo multicast jak jsme zvyklí ze sítě LAN. V tomto případě není možné vyčítat z elektroměru jakákoliv data. Elektroměry v tomto režimu neodpovídají a tudíž není možné zjistit, na které z nich byl příkaz doručen.
- Falešný broadcast nebo multicast vytvoří sdružený příkaz, kdy bude stejný příkaz odeslán jednotlivě na všechny elektroměry. Takto je možné odeslat libovolný příkaz a lze sledovat informace o průběhu provádění příkazu podle popisu v odstavci Sdružené příkazy.

5.6.2 Mřížové sítě

Zcela nadřazená je synchronizace v rámci mřížových sítí, kdy DC má přidělená časová okna ve kterých může vysílat. Pokud toto okno aktuálně přidělené nemá, musí mlčet. Nicméně příkaz z nejvyšší prioritou může být odeslán bez ohledu na přidělené okno. Stejně tak může DC dostat příkaz aby přesto, že má přidělené okno po určitou dobu nekomunikovalo.

5.6.3 Důležité úkoly spojené s routingem

Dále zde jsou klíčové úkoly pro routing. Jejich vysoká priorita je dána tím, že zajišťují funkčnost jednotlivých cest, bez kterých by nebylo možné provést další úkoly.

Vyhodnocení příčiny nedostupnosti modemu

Pokud bude mít elektroměr N krát za sebou problém s komunikací, kde N je nastavitelný parametr, spustí se hledání příčiny jeho nedostupnosti, jelikož jeho nedostupnost může být způsobena nedostupností routeru v cestě. Pokud je opravdu nedostupný router v cestě, jsou označeny jako nedostupné všechny elektroměry za tímto routerem a je jím nastaven čas poslední nedostupnosti na aktuální čas a čas od kdy jsou nedostupné taky na aktuální čas. V případě, že je nedostupný pouze konkrétní elektroměr, je označen jako nedostupný a je mu nastaven čas poslední nedostupnosti a čas od kdy nedostupnost trvá.

DC se pokouší komunikovat s nedostupným elektroměrem v případě, že uplyne nastavitelný čas od poslední nedostupnosti. Tento čas se navíc zvyšuje s narůstajícím dobou od začátku nedostupnosti.

Ověření dostupnosti nedostupných routerů

Jelikož nedostupnost routeru je vážný problém, je nutné je kontrolovat zda již není dostupný nad rámec kontrol běžných elektroměrů, k tomuto slouží právě tato část plánovače.

Aktualizace cest v routerech

V případě, že dojde k přepočítání cest v síti PLC. Je tato část plánovače zodpovědná za vymazání starých routovacích tabulek z routerů a nahrání nových routovacích tabulek do nových routerů. Nastavování a mazání neprobíhá na routerech, které jsou označeny jako nedostupné.

5.6.4 Vyčítání dat

Následuje vyčítání užitečných dat z elektroměrů. Jsou zde různá data, která se vyčítají v různých časových intervalech a události jež vznikají asynchronně a je nutné v pravidelných časových intervalech kontrolovat zda nedošlo ke vzniku události. V tomto případě nejenom, že je možné pozastavit vyčítání libovolného typu dat, ale je možné nastavovat prioritu těmto úlohám a tak měnit uspořádání v rámci plánovače.

Aktualizace indexů

Nejdříve dojde ke kontrole posledních indexů ve všech kruhových bufferech a na základě této informace jsme schopni zjistit kolik záznamů nám, z kterého elektroměru chybí. Zda se provede aktualizace indexů a u kterého elektroměru rozhoduje kritériální funkce, jejíž parametry je čas poslední aktualizace indexů a ručně nastavitelný parametr. Tato kritériální funkce musí být větší než určité minimum a v případě, že tuto podmínku splňuje více elektroměrů je vybrán elektroměr s největším výsledkem funkce. Pochopitelně elektroměr musí být dostupný.

Vyčítání dat

Pro každý druh dat je zde speciální úloha plánovače. Priorita těchto úloh je znamenána v seznamu výše. V rámci jedné úlohy se rozhodnutí zda něco vyčítat a z kterého elektroměru opěr řídí kritériální funkcí s parametry: počet chybějících záznamů a ručně nastavitelný parametr pro zvýšení priority. Opět vyčítání probíhá pouze z dostupných elektroměrů.

5.6.5 Diagnostika

Dále jsou zde úkony spojené s diagnostikou elektroměrů, jako je upgrade firmware, cachování jejich nastavení nebo provádění nastavení s nízkou prioritou.

5.6.6 Routing s nízkou prioritou

Poslední je routing s nízkou prioritou, jehož úkolem je snaha o zdokonalování jednotlivých cest a hledání nových elektroměrů. V počáteční fázi kdy je data koncentrátor nainstalován, je prioritní najít pokud možno všechny elektroměry, proto budou vyčítání dat a diagnostika na začátku deaktivovány a až po ručním nebo automatickém potvrzení, že vyhledávání dopadlo úspěšně začne systém vyčítat data. Nicméně stále se bude snažit nalézt nové elektroměry, které se zde mohou objevit v rámci údržby nebo změny podmínek na síti. Toto hledání již ovšem bude probíhat s nízkou prioritou.

5.6.7 Priorita diagnostiky a routingu s nízkou prioritou

Poslední dvě části plánovače mají stejnou prioritu a plánovač bude přidělovat tyto úkoly v určitém poměru, tak aby mohly být vykonávány průběžně všechny. Je to z toho důvodu, že i v případě, že probíhá upgrade firmware nebo hledání nových elektroměrů, stále by zde měla být snaha o zkvalitňování routovacích cest a opačně.

5.6.8 Algoritmus pro sestavování topologie

Identifikace nových měřičů

Nově měřiče se vyhledávají takovým způsobem, že DC zašle dotaz s číslem udávajícím s jakou pravděpodobností dosud nenalezený měřič odpoví. Tyto výzvy může vysílat přímo DC nebo je může odeslat prostřednictvím nějakého již nalezeného měřiče.

Algoritmus zkouší s různými úrovněmi pravděpodobnosti zasílat výzvy do všech segmentů sítě. Upřednostňuje pravděpodobnosti, při kterých dochází k nalezení nových měřičů.

Algoritmus by bylo možné značně zefektivnit, pokud by byly k dispozici GPS souřadnice měřičů, které se nacházejí v blízkosti DC. Na základě těchto dat by algoritmus byl schopen upřednostnit určité segmenty sítě a nastavit co nejvhodnější pravděpodobnost pro výzvu k registraci. Tato úvaha není zcela zcestná jelikož energetické společnosti často GPS souřadnice při montáži zaznamenávají a mají je uložené v nadřazeném systému. Je tedy pouze nutné, dostat tyto data do DC.

Další možností by byla detekce kolizí na síti, kdy by bylo možné detekovat kdy nastala kolize a podle toho odhadnout pravděpodobnost pro registrační výzvu. Tuto možnost, ale modem v současné době nepodporuje.

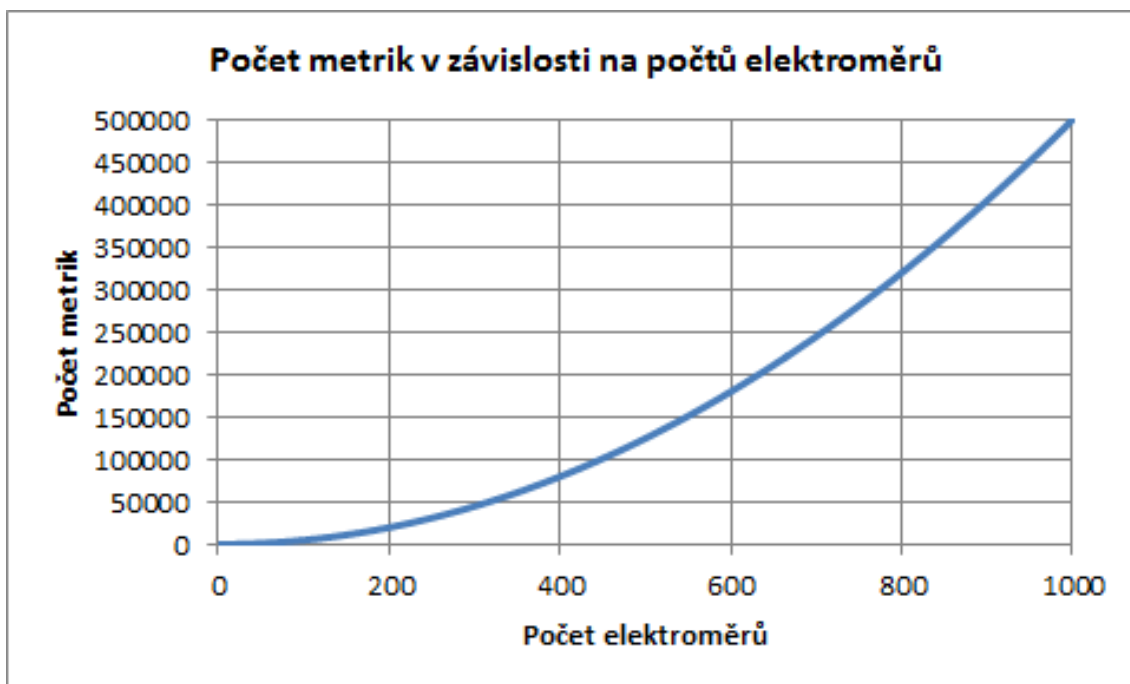
Zjišťování metrik

Metriky jsou nutnou součástí pro pozdější návrh cest. Bohužel PLC síť v tomto ohledu připomíná spíše síť rádiovou, kdy je potencionálně možné aby se slyšely jakékoliv dvojice měřičů nebo měřič a DC. Počet metrik, které je tak nutno pravidelně zjišťovat se rovná kombinaci druhé třídy z n měřičů.

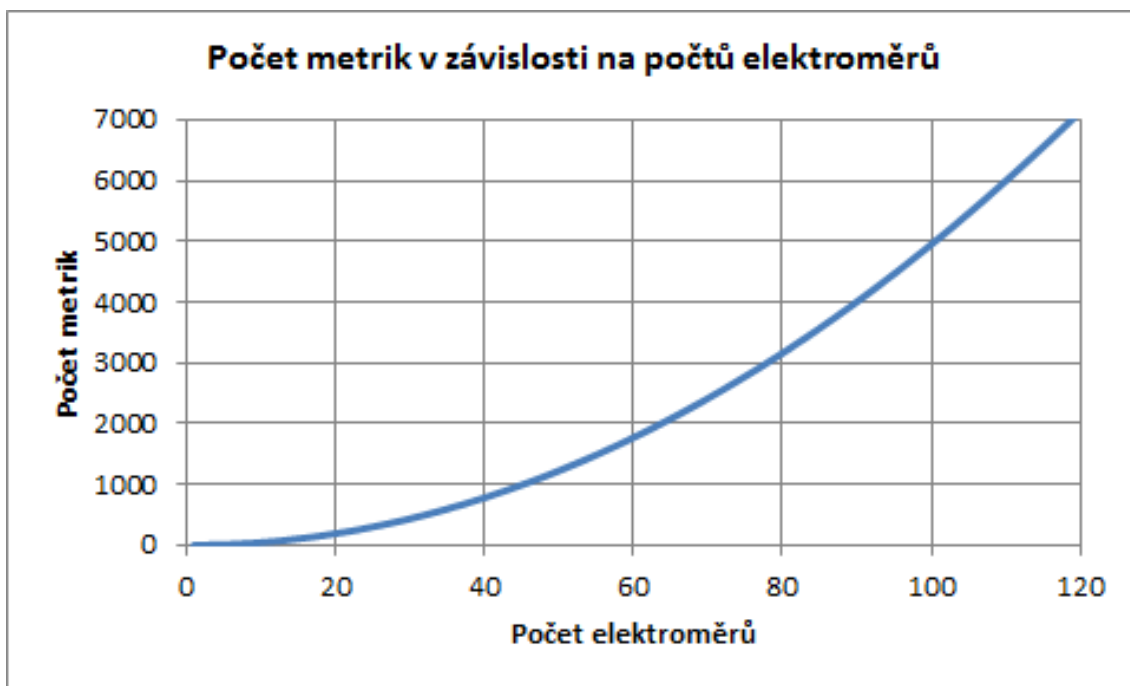
$$C_2(n) = \binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{(n_m + 1)}{2!(n_m + 1 - 2)!} = \frac{n_m^2 + n_m}{2} \quad (5.1)$$

kde n_m je počet měřičů v síti. V rovnici bylo nutné uvažovat počet uzlů v síti n roven $n_m + 1$ jelikož kromě měřičů je nutno za uzel uvažovat také koncentrátor.

Pokud dosadíme do rovnice 5.6.8 dostaneme grafy z obrázků 5.12 a 5.13.



Obr. 5.12: Počet metrik v závislosti na počtů elektroměrů (max 1000)



Obr. 5.13: Počet metrik v závislosti na počtů elektroměrů (max 120)

Jelikož stav sítě závisí především od běžících spotřebičů, které generují rušení a tedy od chování uživatelů. Je vhodné měřit metriky pro každou hodinu zvlášť a navíc ještě rozdělit dny na pracovní a nepracovní. Získáváme tak 2 krát 24 kategorií, tedy

48 kategorií pro metriky. Kategorie můžeme chápat jako fuzzy množiny a pokud je naměřená hodnota o půl páté, metrika se započítá s poloviční váhou jak do čtyř hodin tak do pěti hodin.

K vyhodnocení metrik bude použit filtr klouzavý vážený průměr s exponenciálním zapomínáním. Díky tomu se podaří eliminovat zákmity způsobené chvilkovým zapnutím určitého spotřebiče, ale zároveň budou mít aktuální hodnoty větší váhu, tedy pokud dojde k přerušení signálu v určitém místě, poměrně rychle se to projeví v metrice.

Z grafů 5.12 je patrné, že pro 1000 měřičů bude nutné pravidelně kontrolovat zhruba 500 000 metrik. V případě, že bychom byli schopni zkontrolovat jednu metriku za vteřinu a jako reprezentativní bychom považovali 100 měření pro každou kategorii metrik, trvalo by 278 dní než bychom tohoto stavu dosáhli. Pro obvyklejší variantu 120 měřičů s 7000 metrikami, by se tento čas pohyboval kolem 4 dní. Přičemž za optimální by šlo považovat méně jak jeden den. Proto je nutné počet metrik, které se budou často měřit zmenšit. Proto je nutné vytvořit model topologie, na základě kterého budou odhadnuty významné metriky a ty budou měřeny s větší frekvencí než ostatní. Model je možné sestavit na základě GPS souřadnic měřičů, již naměřených metrik nebo kombinací obou údajů.

Algoritmus pro hledání cest

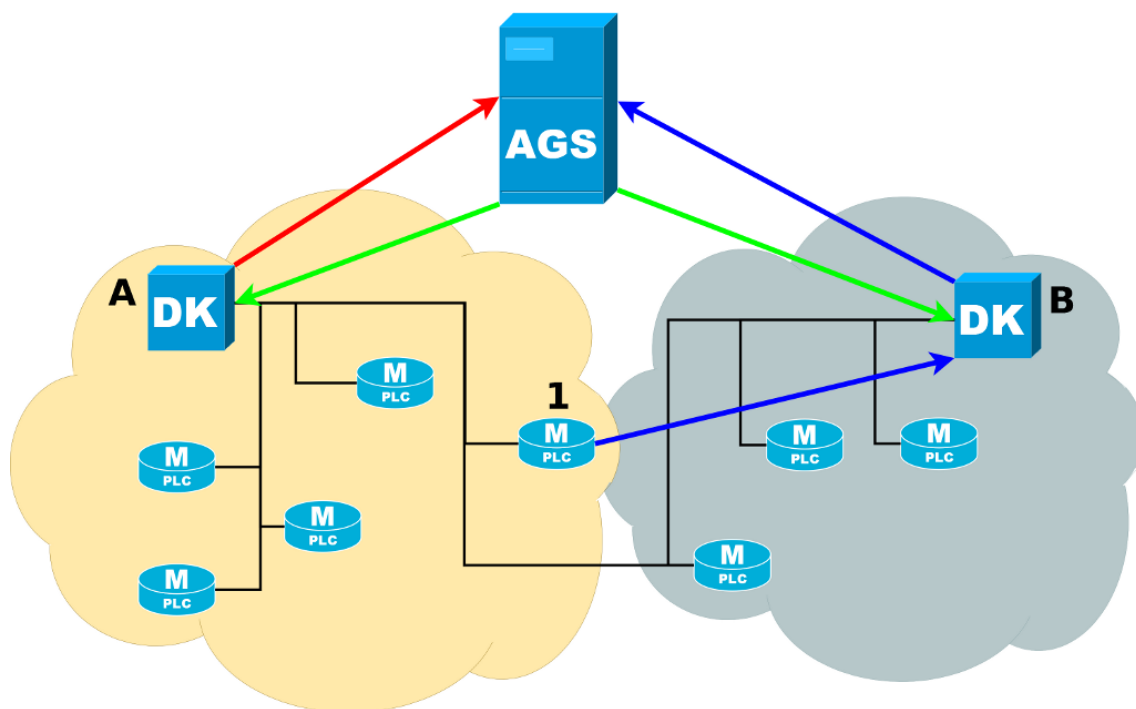
Jakmile jsou zjištěny metriky pro jednotlivé uzly sítě, je výpočet optimálních cest jednoduchý. Pro výpočet cest byl zvolen Dijkstrův algoritmus [7], což je nejrychlejší algoritmus pro hledání cest v případě použití nezáporných metrik a bez možnosti použití heuristik. Stejný algoritmus je použit i v případě protokolu OSPF sítě Ethernet.

Větším problémem je nastavení setrvačnosti vypočtených cest, jelikož změna topologie sebou nese značnou režii. Je tedy nutné vhodně nastavit jak silně musí klesnout metriky a na jak dlouhou dobu aby došlo ke změně.

5.6.9 Mřížové sítě

Popis situace z obrázku 5.14 Existují dvě nezávislé sítě patřící DC A a DC B. Modem 1 patří do sítě A. DC B odesílá identifikační paket, který obsahuje jeho adresu.

Modré šipky – Modem 1 tento paket přijímá, porovnává ho s adresou DC k němuž je registrován, zjišťuje, že se jedná o jiný DC a tedy (s nastavenou pravděpodobností) odpoví. Paket, který přijme DC B obsahuje adresu modemu 1 a adresu DC do jehož sítě patří tedy A. Informuje o nastalém konfliktu AGS (agregační server - nadřazený systém schopný zpracovat tento konflikt), zasílá mu adresu modemu u něž konflikt



Obr. 5.14: Schéma mřížové sítě

nastal (1), adresu DC kterému modem patří A, součet metrik se kterými se může k tomuto modemu dostat.

Červená šipka – AGS server si vyžádá součet metrik cesty k modemu 1 od DC A .

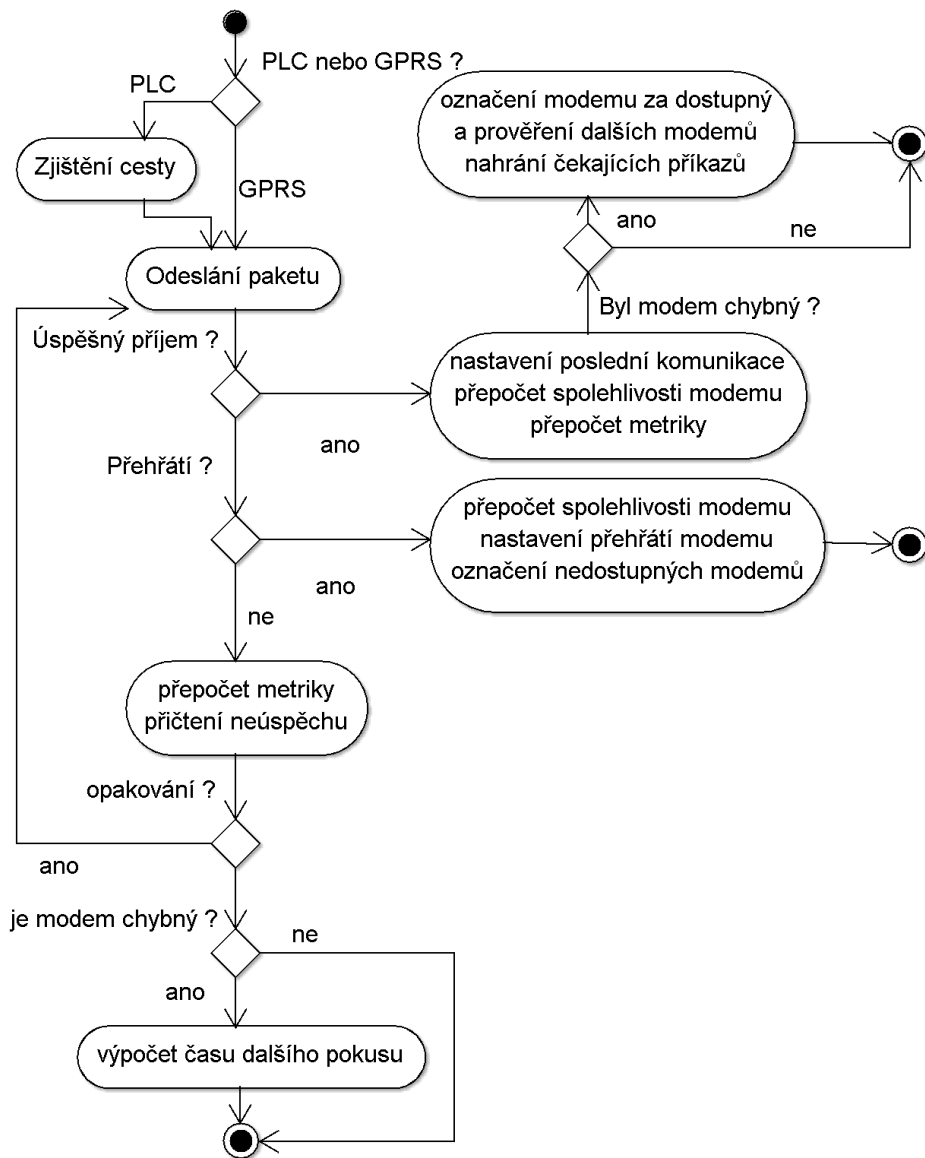
Zelené šipky – AGS server tyto metriky porovná a pokud metrika DC B vychází lépe odešle DC A zprávu, že modem 1 bude patřit jinému DC. DC B odešle zprávu, že si má modem 1 zaregistrovat. Pokud se jedná o první konflikt mezi těmito DC, přidělí AGS server každému z nich časové pásma, ve kterých můžou komunikovat a rozdělí mezi ně cesty.

I když nyní patří modem 1 DC B nezabraňuje to DC A aby nadále zjišťoval metriky k tomuto modemu a v případě, nalezení cesty s lepší metrikou bude informovat AGS, který opět rozhodne komu bude modem patřit.

V případě možného peer to peer spojení mezi DC lze vynechat AGS server a rozhodování provádět na úrovni DC.

5.6.10 Odesílání paketu

Diagram popisující odesílání paketu je na obrázku 5.15.



Obr. 5.15: Diagram zobrazující průběh odesílání paketu

Jakmile jsou data pro odeslání připravená, zavolá se funkce pro odeslání paketu. V případě že se jedná o měřič na PLC síti, což v případě protokolu MT29 platí, je

zjištěna cesta k měřiči a přidána do paketu. Následuje odeslání paketu a čekání na odpověď po nastavenou dobu.

Pokud odpověď dorazí v pořádku, nastaví se čas poslední komunikace s měřičem a aktualizuje se spolehlivost měřiče a metrika příslušného spojení. Pokud byl měřič označen jako nedostupný, je označen jako dostupný a pokud byl opakovačem, prověří se i dostupnost měřičů v segmentu za ním.

Pokud přijde odpověď se zprávou o přehřátí měřiče, je aktualizována spolehlivost měřiče, nastaven příznak přehřátí a měřič je označen jako nedostupný.

Pokud nedorazí žádná odpověď, je aktualizována metrika spojení i spolehlivost měřiče. V případě potřeby se může zopakovat odeslání paketu. Pokud všechny opakování skončí neúspěšně a modem je označený jako nedostupný, nastaví se čas příštího pokusu o komunikaci.

5.7 Program pro logování

Jedná se o program podobný programu Syslog s prostředí Linux. Tento program přijímá zprávy pomocí MPI. Tyto zprávy obsahují číslo logu, datum a čas a sadu parametrů. Program Logger načte sadu podmínek pro jednotlivé parametry zapsané v mini skriptovacím jazyce a na základě nich se rozhodne co s logem provede dál. Má následující možnosti:

- Nastaví prioritu logu
- Změní log na alarm
- Log odešle na server
- Log uschová na určitou dobu v paměti
- Log zahodí

V pravidelných intervalech bude spuštěna úloha pro údržbu logů, která provede smazání starých logů. Některé logy, například ty, obsahující kvalitu signálu na určité cestě, nemusí být mazány podle strategie, že jsou smazány nejstarší záznamy, ale může být s časem snižována vzorkovací frekvence nebo může být použit určitý ztrátový kompresní algoritmus, například interpolace nebo extrapolace, případně statistický odhad.

5.8 Program pro práci s frontou příkazů

Datový model fronty příkazů a související popis lze nalézt v podkapitole 5.3.5 týkající se databáze. Program pomocí MPI přijme příkaz z určitého zdroje (nadřazený server, konfigurační rozhraní, ...), ověří zda je zdroj oprávněn zadat tento příkaz a zda může použít příslušnou prioritu. Následuje zjištění podrobností o příkazu a zařazení do fronty. Po vyřízení příkazu program informuje zdroj, že byl příkaz vyřízen. V případě transakcí program kontroluje zda proběhlo vše v pořádku, pokud ne zařadí příkazy pro návrat do předchozího stavu.

Dále program umožňuje odstraňovat, modifikovat příkazy zdrojům, kterým patří nebo mají k tomu oprávnění. Dále poskytuje informace o příkazech a skupinách příkazů.

Plánování příkazů

Příkaz může být odeslán v jednom z těchto režimů:

- Provést okamžitě jak to bude možné
- Provést v přesném časovém okamžiku
- Provádět pravidelně, nastaven je počáteční okamžik a interval vykonávání s informací zda tento interval změnit v případě, že dochází ke změně času zimní <-> letní
- Kalendář příkazů – týdenní kalendář s možností nastavení svátku, stejná funkčnost jako kalendář v elektroměru

I příkazy vytvořené automaticky na základě času mají nastavitelnou prioritu a je možné provádět na nich operace ze serveru. Všechny typy příkazu, kromě těch prováděných okamžitě jsou plánovány pomocí programu DC_cron jehož popis lze nalézt v podkapitole 5.2.7 o rozdělení funkčnosti.

6 SROVNÁNÍ S KONKURENCÍ

Situace na trhu se Smart Meteringem je nejvíce ovlivněna neexistencí dominujícího standardu pro PLC komunikaci. V současné době existuje celá řada standardů a ještě více proprietárních řešení. Ani výrobky různých firem komunikující na určitém standardu nejsou vzájemně kompatibilní, vzhledem k nejednoznačné implementaci standardu a nutnosti dalších funkcionalit nad rámec standardu. Pro aplikační vrstvu přenosu dat z měřičů se ujal standard DLMS, který přežil z doby komunikace po sériové lince. Jelikož se, ale jedná o textový protokol, vytvořený tak aby byl přímo dobře čitelný člověkem, není pro pomalou PLC komunikaci příliš vhodný a spousta firem jej nevyužívá. V důsledku toho se dá říci, že každá firma s produkty pro Smart Metering má i vlastní data koncentrátor.

Hlavní odlišnosti tohoto koncentrátoru je podpora protokolu MT29 a dalších protokolů firmy ModemTec pro komunikaci po PLC síti. Kromě toho je významnou vlastností implementace požadavků z [12], [13] a poznatků z praxe. Díky tomu by měl navržený koncentrátor splňovat nejaktuálnější požadavky současného trhu.

Vzhledem k tomu, že se požadavky zákazníku velmi odlišují je důležitou součástí škálovatelnost celého řešení a dostatečná rezerva výkonu pro složitější funkcionality. Díky tomu má zákazník jistotu, že v případě nových požadavků v budoucnu bude možné zařízení rozšířit o dodatečný HW nebo SW. Vzhledem k velmi vysoké dynamice vývoje v oblasti Smart Meteringu a Smart Gridů se jedná o velmi významnou vlastnost.

Zřízení na trhu můžeme rozdělit do tří kategorií. Velmi jednoduchá zařízení bez OS, zařízení s OS, ale bez možnosti zásadního škálování, zařízení s OS a s možností škálování jak SW tak HW. První skupina bez OS se v poslední době ukazuje jako velmi problematická a vzhledem k nemožnosti uspokojit většinu požadavků se stává nevyhovující, příkladem může být získání IP adresy z DHCP serveru nebo jakékoli jiné funkcionality považované za samozřejmost. Pevně nakonfigurovaná zařízení s OS mají jednodušší vývoj, ale existuje riziko, že neuspokojí veškeré zákazníky nebo naopak nabídnou spoustu funkcionalit a naddimenzovaný HW, který zákazník nepotřebuje. Představiteli této kategorie mohou být koncentrátoři firem ADD, Echelon nebo Iskra. V případě Iskry mě zaujal parametr maximálního měsíčního toku z jednoho měřiče stanovený na 10kB, přičemž již v současné době představuje tok čistých dat z měřiče podle požadavků ČEZu nebo PRE zhruba 70kB a dá se předpokládat, že s rozšířením možností elektroměrů toto číslo ještě naroste. Poslední skupinu škálovatelných zařízení představují výrobky českých výrobců ZPA a zde popisovaného výrobku firmy ModemTec. Díky tomu jsou tyto firmy schopny se přizpůsobit velmi náročným požadavkům ze strany zákazníků a dosahují nejlepších výsledků v pilotním projektu firmy ČEZ.

Tabulka 6.1 ukazuje srovnání HW jak bylo uvedeno v jednotlivých datasheetech. Je vidět, že konfigurace jednotlivých výrobků jsou dosti podobné a o jejich kvalitách rozhoduje množství implementovaných funkcionalit v oblasti SW a dále možnosti konfigurace a diagnostiky chování celého zařízení.

	ModemTec	ZPA	ADD	Echelon	Iskra
OS	Linux	Linux	?	?	Windows CE
CPU	386/486	ARM 9	?	386	?
Spojení s nadřazeným systémem	Ethernet, GPRS, WIFI, RS232/485	Ethernet, GPRS, WIFI, RS232	Ethernet, GPRS	Ethernet	Ethernet, GPRS, RS485
Spojení s měřiči	MT29, RS485, Wireless- MBUS	PLC, RS485, M-BUS, ZigBee	PLC	PLC	PLC

Tab. 6.1: Srovnání s konkurencí [17] [18] [16] [15]

ZÁVĚR

Byl vytvořen návrh zařízení, který respektuje nejenom zadání, ale i požadavky specifikované Nizozemskou asociací distributorů energií [12], společností ČEZ [13], společnosti Energa [14] a zkušenostmi z praxe.

Zařízení bylo rozděleno na šest samostatných desek popsaných v kapitole 3. Základní desku, procesorovou desku, desku s hradlovým polem, desku pro bateriový provoz, desku se vstupy a výstupy a desku se sumačním elektroměrem. Díky tomuto rozdělení je zaručena velmi dobrá škálovatelnost celého zařízení, což je velmi výhodné vzhledem k velmi rozdílným požadavkům zákazníků. Dále je možné připojit k zařízení další moduly jako například GPRS modem.

Jako operační systém byl zvolen OS Linux vzhledem k nízkým nákladům a otevřenosti zdrojových kódů. Pomocí projektu Open Embedded byla vytvořena vlastní distribuce, která obsahuje minimalisticky nakonfigurované jádro pro příslušný hardware, potřebné knihovny a základní aplikace. Systém byl správně nakonfigurován včetně podpory pro sériovou konzoli, web server a SSH konzoli. Paměť je rozdělena na 10 samostatných oddílů, aby případné poškození jednoho z nich nezpůsobilo příliš velké škody. Pro systémový oddíl je použit přístup copy on write, kdy vše je uloženo v paměti FLASH, ale při požadavku na zápis nebo změnu, dojde ke zkopírování souboru do paměti RAM. Vše je podrobně popsáno v kapitole 4.

Návrh software byl realizován pomocí standardizovaného jazyka UML a pro implementaci byl zvolen jazyk C++. Nejdříve byly vytvořeny případy užití týkající se tohoto zařízení, které následoval datový a objektový model. Vytvořen byl seznam parametrů, které bude možné konfigurovat. Programy komunikují mezi sebou pomocí zpráv MPI a databáze SQLite. Pro základní část zařízení byl vytvořen objektový návrh, ze kterého byla vygenerována kostra podrobně okomentovaného kódu, ze kterého byla vygenerována dokumentace pro programátora, která je přiložena na DVD. Dále byl navržen plánovač pro komunikaci s elektroměry na protokolu MT29 včetně algoritmů pro objevování nových elektroměrů, automatickou optimalizaci cest a nastavení opakovačů. Byl navrhnout způsob jak se vypořádat s přeslechly a mřížovými sítěmi. Stejně jako hardware i software je maximálně modulární a je možné přidat další protokoly pro komunikaci s měřiči jiných firem. Stejně tak lze pomocí dynamicky linkovaných knihoven přidat za běhu další příkazy, které zařízení provede. Vše je detailně popsáno v kapitole 5.

Aby bylo možné zařízení uvést na trh, je nutné dokončit návrh desek plošných spojů, podle přiložené dokumentace naprogramovat jádro celého zařízení a přidat moduly pro komunikaci s již konkrétními servery a měřiči.

LITERATURA

- [1] ČEZ, a. s. [online]. 2010 [cit. 2012-04-02]. ČEZ Smart Grids. Dostupné z URL: <<http://www.futuremotion.cz/smartgrids/cs/index.html>>.
- [2] YAGHMOUR, Karim, et al. *Building Embedded Linux Systems*. 2nd edition. Sebastopol USA : O
- [3] JELÍNEK, Lukáš. *Vytváříme vlastní distribuci Linuxu : Od návrhu po fungující systém*. 17.03.2010. Praha : Computer Press, 2010. 304 s. ISBN 8025124339.
- [4] MASTERS, Jon; BLUM, Richard. *Linux profesionálně : programování aplikací*. Vyd. 1. Brno : Zoner Press, 2008. 539 s. ISBN 978-80-86815-71-8.
- [5] C. MARTIN, Robert. *Čistý kód : návrhové vzory, refaktování, testování a další techniky agilního programování*. Brno : Computer Press, 2009. 424 s. ISBN 978-80-251-2285-3.
- [6] LIBERTY, Jesse. *Naučte se C++ za 21 dní*. Praha : Computer Press, 2002. 766 s. ISBN 80-7226-774-4.
- [7] DIJKSTRA, E. W. A note on two problems in connexion with graphs. In: *Numerische Mathematik*. 1. vyd., 1959. DOI: 10.1007/BF01386390.
- [8] CÍSAŘ, Pavel. *InterBase/FireBird*. 10.06.2003. Praha : Computer Press, 2003. 464 s. ISBN 8072269461.
- [9] OWENS, Michael. *The definitive guide to SQLite*. New ed. Berkeley, CA: Apress, 2006. ISBN 978-159-0596-739.
- [10] HANA, Kanisová. *UML srozumitelně*. Vyd. 1. Brno: Computer Press, 2004, 157 s. ISBN 80-251-0231-9.
- [11] POLSTEROVÁ, Helena a Zdenka ROZSÍVALOVÁ. *Řízení jakosti*. Skripta. Vysoké učení technické v Brně.
- [12] KEMA CONSULTING. *Dutch Smart Meter Requirements: Main Document*. 3.0. 2010. Functional and technical specifications Smart Meters, B101.
- [13] ČEZ MĚŘENÍ, s.r.o. *Funkcionalita koncentrátoru*. 1.2. 2011.
- [14] ENERGA –OPERATOR S.A. *Opis wymagań funkcjonalnych - Koncentrator*. 2010.
- [15] ISKRAEMECO (UK) LTD. *P2LPC: Data concentrator for AMM-systems*. Cambridge. Dostupné z: <http://www.iskraemeco.co.uk>

- [16] ECHELON. *The NES Data Concentrator*. 2011. Dostupné z: <http://www.echelon.com/products/controllers/meter-data-concentrator/default.htm>
- [17] ZPA SMART ENERGY A.S. *DATA KONCENTRÁTOR CAM 3500*. Trutnov, 2009. Dostupné z: http://www.zpa.cz/index.php/cz/produkty_a_reseni__1/speciality/cam_3500
- [18] Routers/Data Concentrators. ADD GRUP. *LinkedIn* [online]. [cit. 2012-04-12]. Dostupné z: <http://www.linkedin.com/company/add-grup/routers-data-concentrators-650290/product>
- [19] *Boost 1.49.0 Library Documentation* [online]. 2012 [cit. 2012-04-14]. Dostupné z: http://www.boost.org/doc/libs/1_49_0/
- [20] KEMA CONSULTING. *Dutch Smart Meter Requirements: P3.2 Companion Standard*. 3.0. 2010. Functional and technical specifications Smart Meters.
- [21] RAMIREZ, Alejandro. *ArgoUML User Manual: A tutorial and reference description*. 0.32.2. 2010.
- [22] OPENEMBEDDED TEAM. *OpenEmbedded User Manual*. 2009.
- [23] BINDER. *Bajonett: Subminiatursteckverbinder Serie 710*.
- [24] DMP ELECTRONICS INC. *Vortex86SX-A9100 SoC: AMI BIOS Reference Manual*. 1. vyd. Taiwan, 2007.
- [25] CRYSTALFONTZ AMERICA, Inc. *SPECIFICATION: CFAG12864I-STI-TN*. Spokane Valley.
- [26] FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD. *USB to RS232 Serial Converter Range of Cables: Datasheet. 1.3*. Glasgow, 2009.
- [27] CINTERION. *MC75i, TC65i, and TC63i Wireless Modules*. Munich, 2008.
- [28] DMP ELECTRONICS INC. *MTBF Report: Vortex86SX-6150E-V2*. Taipei, 2009.
- [29] ICOP TECHNOLOGY INC. *User's Manual: VDX-6350E / VDX-6350E-PLUS*. 1.0A. 2009.
- [30] ICOP TECHNOLOGY INC. *User's Manual: VSX-6150E-V2 / VSX-6150E-V2-PLUS*. 1.1A. 2010.

- [31] DMP ELECTRONICS INC. *Vortex86SX: 32-BIT x86 Embedded SoC*. 1.001.
- [32] DMP ELECTRONICS INC. *Fact Sheet: Vortex86DX*. 0.9A. 2008.
- [33] MULTICOMP. *Waterproof RJ45 CAT.5E STP*. 2011.
- [34] HDD mount. ECVV. *ECVV* [online]. [cit. 2011-09-14]. Dostupné z: <http://www.ecvv.com/product/2010965.html>
- [35] CF slot. ECVV. *ECVV* [online]. [cit. 2011-09-14]. Dostupné z: <http://www.ecvv.com/product/2011135.html>
- [36] FREESCALE SEMICONDUCTOR. *IMX53xA: Automotive and Infotainment Applications Processors*. 2011. Dostupné z: http://cache.freescale.com/files/32bit/doc/data_sheet/IMX53AEC.pdf?pspll=1
- [37] TEXAS INSTRUMENTS INC. *AM389x Sitara™: ARM® Processors*. 2011. Dostupné z: <http://www.ti.com/lit/ds/symlink/am3892.pdf>
- [38] TEXAS INSTRUMENTS INC. *AM3517/05 Sitara™: ARM Microprocessors*. 2011. Dostupné z: <http://www.ti.com/lit/ds/symlink/am3505.pdf>
- [39] TEXAS INSTRUMENTS INC. *AM1808: ARM Microprocessor*. 2011. Dostupné z: <http://www.ti.com/lit/ds/symlink/am1808.pdf>
- [40] INTEL. *Intel® Atom™ Processor E6xx Series*. 004US. 2011. Dostupné z: <http://download.intel.com/embedded/processor/datasheet/324208.pdf#iid=3790>
- [41] OPTICKÉ SONDY IM9000. I-COM, s.r.o. *I-com* [online]. [cit. 2012-04-17]. Dostupné z: <http://www.i-com.cz/opticke-sondy-im9000>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

DC	data koncentrátor – data concentrator
DTS	distribuční transformační stanice
PLC	komunikace po elektrické síti - Power line communication
RF	rádio frekvenční
IEM	inteligentní elektroměr
SW	software
HW	hardware
OS	operační systém
FPGA	programovatelné hradlové pole - Field-programmable gate array
RAM	paměť s přímým přístupem - random-access memory (zde spíše ve významu volatilní paměť)
FRAM	ferroelectric RAM
CF	compact flash
HDD	pevný disk - hard disk drive
SD	Secure Digital - typ paměťové karty
P.O.S.T.	Power-on self-test
MTBF	střední doba mezi poruchami - mean time between failures
MPI	systém pro zasílání zpráv - Message Passing Interface
AGS	agregační server, nadřazený systém se specifickou funkcionalitou
ACPI	Advanced Configuration and Power Interface