

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce
Automatizované odečty fakturačních elektroměrů

Martin Cvek

Vedoucí práce: Ing. Ondřej Nývlt

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Výpočetní technika

květen 2012

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Ing. Ondřeji Nývtovi za ochotu a rady. Dále bych chtěl poděkovat své rodině a všem, kteří mi pomáhali, za podporu při studiu.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 25.5.2012

Abstract

The subject of this bachelor thesis is technical aspects of automated remote reading of electricity meters. The main task is to present a design and an implementation of PC software for remote reading of these meters. The thesis includes also the comparison of two communication protocols (ČSN EN 62056-21 and MODBUS RTU), which are suitable for this task.

Abstrakt

Tato bakalářská práce se zabývá technickou stránkou automatizovaného odečtu fakturačních elektroměrů. Hlavním předmětem práce je návrh a implementace programového vybavení pro PC sloužícího pro odečet elektroměru. Dále práce obsahuje porovnání komunikačního protokolu dle normy ČSN EN 62056-21 s protokolem MODBUS RTU.

Obsah

1	Úvod.....	1
1.1	Obsah práce.....	1
2	Protokol MODBUS.....	3
2.1	Úvod.....	3
2.2	Popis protokolu.....	3
2.3	Datový a adresový model.....	5
2.4	MODBUS transakce.....	5
2.5	Kódy funkcí.....	7
2.6	Implementace MODBUS na sériové lince.....	8
2.6.1	Režim přenosu RTU.....	8
3	Protokol dle normy ČSN EN 62056-21.....	11
3.1	Protokol přenosu dat.....	11
3.2	Fyzikální vlastnosti.....	11
3.3	Znakový přenos.....	11
3.4	Definice zpráv.....	12
3.4.1	Speciální znaky vyskytující se ve zprávách.....	12
3.4.2	Zabezpečení přenosu zpráv.....	12
3.4.3	Zpráva s požadavkem.....	12
3.4.4	Zpráva s identifikací.....	13
3.4.5	Zpráva s volbou potvrzení/možnosti.....	13
3.4.6	Zpráva s daty (s výjimkou programovacího režimu).....	14
3.4.7	Zpráva s potvrzením.....	14
3.4.8	Zpráva s požadavkem na opakování.....	14
3.4.9	Zpráva s programovacím povel.....	14
3.4.10	Zpráva s programovacím povel..... používající nepovinné dílčí bloky.....	15
3.4.11	Zpráva s daty (programovací režim).....	15
3.4.12	Zpráva s daty (programovací režim)..... používající nepovinné dílčí bloky.....	15
3.4.13	Zpráva o chybách.....	16
3.4.14	Zpráva o přerušení.....	16
3.5	Struktura datového bloku a datového souboru.....	16
3.6	Režim protokolu C.....	18
4	Porovnání protokolu MODBUS RTU (RS485) a protokolu dle ČSN EN 62056-21.....	19
4.1.1	Základní vlastnosti protokolů.....	19
4.1.2	Struktura protokolů.....	19
4.1.3	Zabezpečení přenášených dat.....	20
4.1.4	Efektivita protokolů.....	20
5	Aplikace pro automatizované odečty.....	21
5.1	Požadavky na aplikaci.....	21
5.2	Návrh aplikace.....	21
5.3	Implementace aplikace.....	22
5.3.1	Konzolová aplikace.....	22
5.3.2	Aplikace s grafickým uživatelským rozhraním.....	24
5.4	Testy aplikace.....	26
6	Závěr.....	27
6.1	Možná rozšíření.....	27
7	Použitá literatura.....	29
A	Obsah příloženého CD.....	31

B	Nastavení odečtu elektroměru ZPA ED 310.....	33
C	Datový blok odečtu elektroměru.....	35
D	Uživatelská příručka.....	37
	D.1 Konzolová aplikace.....	37
	D.2 GUI aplikace.....	38

Seznam obrázků

Obrázek 2.1: MODBUS - příklady implementace.....	3
Obrázek 2.2: Struktura MODBUS rámce.....	4
Obrázek 2.3: MODBUS transakce.....	4
Obrázek 2.4: Stavový diagram MODBUS transakce.....	6
Obrázek 2.5: Pořadí bitů v režimu RTU.....	9
Obrázek 2.6: Struktura rámce zprávy RTU.....	9
Obrázek 3.1: Struktura rámce znaku dle ČSN EN 62056-21.....	12
Obrázek 3.2: ČSN EN 62056-21: Zpráva s požadavkem.....	12
Obrázek 3.3: ČSN EN 62056-21: Zpráva s identifikací.....	13
Obrázek 3.4: ČSN EN 62056-21: Zpráva s volbou potvrzení/možnosti.....	13
Obrázek 3.5: ČSN EN 62056-21: Zpráva s daty.....	14
Obrázek 3.6: ČSN EN 62056-21: Programovací povel.....	14
Obrázek 3.7: ČSN EN 62056-21: Programovací povel s dílčími bloky.....	15
Obrázek 3.8: ČSN EN 62056-21: Zpráva s daty - programovací režim.....	15
Obrázek 3.9: ČSN EN 62056-21: Zpráva s daty - programovací režim - dílčí bloky.....	15
Obrázek 3.10: ČSN EN 62056-21: Zpráva o chybách.....	16
Obrázek 3.11: ČSN EN 62056-21: Zpráva o přerušení.....	16
Obrázek 3.12: ČSN EN 62056-21: Datový blok – režim odečtu.....	16
Obrázek 3.13: ČSN EN 62056-21: Datový blok – programovací režim.....	17
Obrázek 3.14: ČSN EN 62056-21: Datový řádek.....	17
Obrázek 3.15: ČSN EN 62056-21: Datový soubor.....	17
Obrázek 3.16: ČSN EN 62056-21: Schéma režimu protokolu C.....	18
Obrázek D.1: GUI aplikace: přehled odečtu.....	38
Obrázek D.2: GUI aplikace: nastavení.....	38
Obrázek D.3: GUI aplikace: odečet v tabulce.....	39

Seznam tabulek

Tabulka 2.1: Datový model MODBUS.....	5
Tabulka 2.2: Chybové kódy MODBUS.....	6
Tabulka 2.3: Veřejné kódy funkcí MODBUS.....	7
Tabulka 3.1: ČSN EN 62056-21: Přenosové rychlosti.....	13
Tabulka 5.1: Třída TSerialPort.....	22
Tabulka 5.2: Třída TElectrometer.....	23
Tabulka 5.3: Návrátové hodnoty konzolové aplikace.....	24
Tabulka 5.4: Třída TReadingThread.....	25
Tabulka 5.5: Třída TPrefix.....	25

1 Úvod

Fakturační elektroměry jsou nedílnou součástí elektrické rozvodné sítě. Tak jako v jiných oblastech i zde je snaha o co největší automatizaci a digitalizaci. Proto dochází k přechodu od analogových elektroměrů k elektronickým, řízeným mikropočítačem. S digitalizací přichází i snadná možnost výměny dat mezi tarifním zařízením a počítačem. Výměnou dat se rozumí odečet, programování a řízení elektroměru. Tato bakalářská práce se zabývá místní výměnou dat, dálkovou výměnou dat tento dokument nerozebírá.

Komunikace s elektroměrem může probíhat přes komunikační rozhraní elektrické nebo optické, za použití různých protokolů. Tento dokument se zabývá komunikací přes elektrické rozhraní, konkrétně RS485, za použití protokolu podle normy ČSN 62056-21.

Cílem této bakalářské práce je vytvoření programové vybavení pro odečet elektroměru vybavené grafickým uživatelským rozhraním a porovnání protokolu MODBUS RTU s protokolem ČSN 62056-21.

Zanětí práce vzniklo na základě požadavku pražské firmy PRONIX s.r.o., která je předním integrátorem systémů napájení elektrickou energií.

1.1 Obsah práce

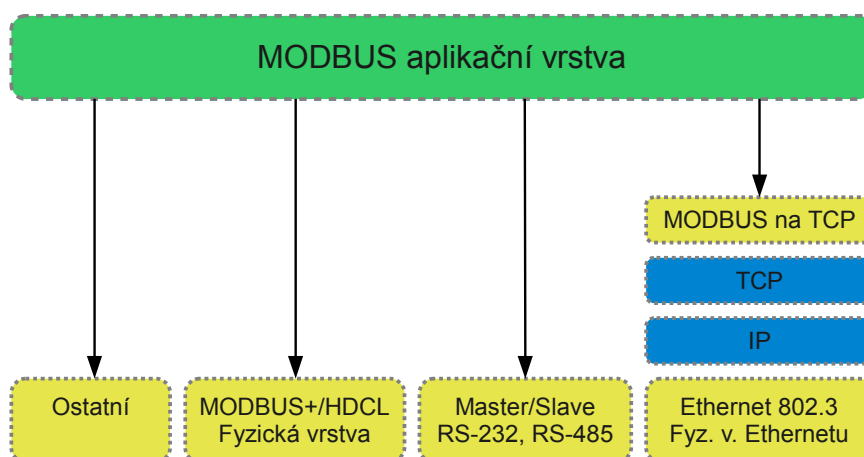
Nejprve je v kapitole 2 popsán komunikační protokol MODBUS a jeho implementace na sériové lince MODBUS RTU. V kapitole 3 jsou uvedeny hlavní rysy protokolu dle normy ČSN 62056-21 s ohledem na jeho použití v implementační části bakalářské práce. Kapitola 4 porovnává protokoly MODBUS RTU a protokolu dle ČSN 62056-21. Kapitola 5 obsahuje návrh a popis implementace programového vybavení odečty elektroměrů s využitím sériového rozhraní RS485.

2 Protokol MODBUS

Tato kapitola obsahuje popis hlavních rysů protokolu MODBUS [1] a implementace MODBUS RTU na sériové lince [2]. Kapitola obsahuje ilustrace inspirované obrázky v [1] a [2].

2.1 Úvod

MODBUS je protokol aplikační vrstvy, úrovně 7 v ISO/OSI modelu. Byl vytvořeno roku 1979 firmou Modicon, má jednoduchou strukturu a jeho využití stále roste. Poskytuje komunikaci typu klient–server mezi zařízeními připojenými na sběrnicích a sítích různého typu. V současné době je podporována řada implementací počínaje sériovou linkou RS-232, RS-422 a RS-485, přes optické a rádiové sítě, po sítě využívající protokol TCP/IP. Pro propojení různých druhů implementací se používá *gateway* (brána).

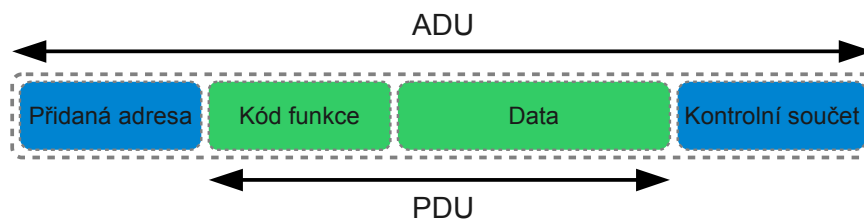


Obrázek 2.1: MODBUS - příklady implementace

2.2 Popis protokolu

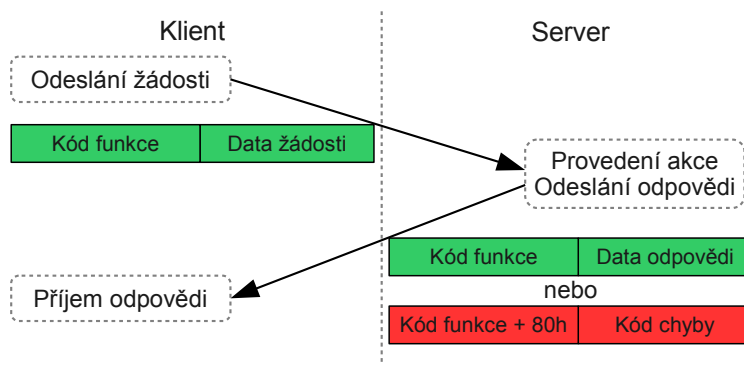
MODBUS protokol definuje strukturu zprávy na úrovni protokolu a na aplikační úrovni. Na úrovni protokolu, nezávisle na typu implementace, se struktura nazývá PDU – Protocol Data Unit. Na aplikační úrovni je struktura PDU rozšířena a nazývá se ADU – Application Data Unit.

MODBUS ADU sestavuje klient, který začíná MODBUS transakci. Server na každou zprávu klienta odpoví. Přijetím odpovědi klientem, končí MODBUS transakce. Kód funkce určuje operaci, jakou má server provést. Obsah datové části je závislý na požadované funkci. Datová část může být vypuštěna, záleží na dané funkci.



Obrázek 2.2: Struktura MODBUS rámce

Pro kód funkce je vyhrazen jeden bajt a využívá se rozsah hodnot 1 až 255. Hodnoty 128 až 255 jsou vyhrazeny pro chybové kódy odpovědi. Kód funkce 0 není platným kódem. Pokud operace proběhne správně, tak je kód funkce v odpovědi serveru stejný jako ve zprávě klienta. V případě neúspěchu má kód funkce odpovědi nastavený nejvyšší bit na logickou jedničku, jinak je jeho hodnota stejná jako u požadavku. V datové části zprávy odpovědi od serveru je v případě neúspěchu kód chyby (*exception code*), pro určení příčiny selhání operace. Pro případ, že server neodpoví na žádost je vhodné stanovit časový limit na doručení odpovědi. Poslední část ADU je kontrolní součet, který nemusí být v rámci obsažen, pokud je bezchybnost přenosu rámců zajištěna jiným protokolem, například TCP.



Obrázek 2.3: MODBUS transakce

Protokol MODBUS definuje tři základní typy zpráv:

- Požadavek (Request PDU)
1 bajt kód funkce + n bajtů dat požadavku
- Odpověď (Response PDU)
1 bajt kód funkce (stejný jako v požadavku) + n bajtů dat odpovědi
- Chybová odpověď (Exception Response PDU)
1 bajt kód funkce (kód funkce požadavku + 80h) + 1 bajt kód chyby

Velikost MODBUS PDU je omezena. Toto omezení vzniklo v první implementaci přes sériovou linku a platí i pro ostatní implementace. Maximální velikost ADU na HW vrstvě RS-485 je 256 bajtů. ADU pro komunikaci po sériové lince se skládá z adresy (1 bajt), PDU a CRC (2 bajty). Tím je určena velikost PDU na 253 bajtů. Velikost ADU pro jiné implementace se liší a je obvykle větší. Například ADU na TCP/IP má velikost 260 bajtů.

Při posílání vícebajtové položky se používá reprezentace Big-Endian. To znamená, že nejprve se odesílá nejvyšší bajt a nejnižší bajt se posílá jako poslední.

2.3 Datový a adresový model

Datový model MODBUS je založen na sadě tabulek s charakteristickým významem. Jsou definovány čtyři základní tabulky.

Tabulka	Typ objektu	Režim přístupu	Popis
Diskrétní vstup (Discrete Inputs)	1-bitový	Pouze čtení	Data poskytovaná I/O systémem
Cívky (Coils)	1-bitový	Čtení / Zápis	Data modifikovatelná aplikačním programem
Vstupní registry (Input Registers)	16-bitový	Pouze čtení	Data poskytovaná I/O systémem
Výstupní registry (Holding Registers)	16-bitový	Čtení / Zápis	Data modifikovatelná aplikačním programem

Tabulka 2.1: Datový model MODBUS

Každá tabulka může mít vlastní adresový prostor v aplikační paměti zařízení nebo se můžou adresové prostory překrývat (částečně nebo úplně). Pro každou ze základních tabulek protokol umožňuje výběr z 65536 položek. Operace čtení a zápisu jsou navrženy tak, aby mohli zahrnovat více po sobě jdoucích datových položek do maximální velikosti dat závislé na kódu funkce transakce.

V MODBUS PDU jsou datové položky adresovány v rozsahu od 0 do 65535.

V MODBUS datovém modelu jsou položky v datových blocích číslovány od 1 do n .

Mapování mezi datovým modelem MODBUS a aplikací zařízení je na výrobci.

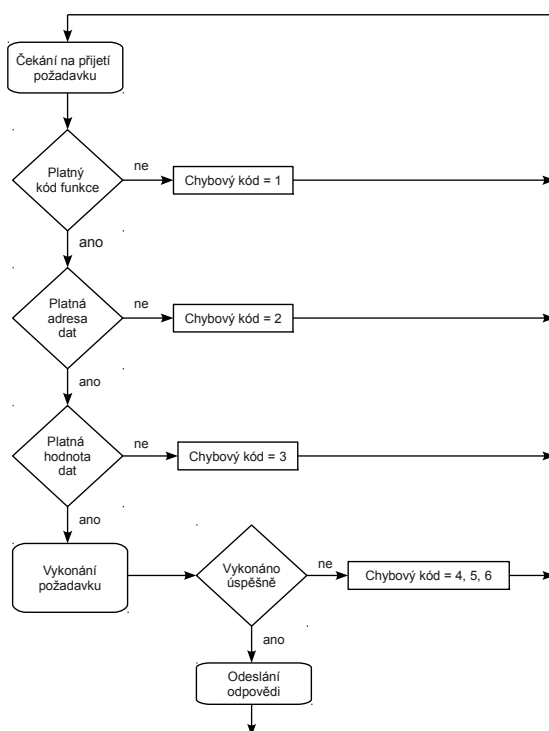
2.4 MODBUS transakce

Obecný popis transakce na straně serveru zachycuje stavový diagram na obrázku 2.4. Jakmile server přijme zprávu, zpracuje ji a odešle odpověď. Odpověď může být dvojího typu. Pozitivní odpověď v případě úspěšného zpracování zprávy a vykonání dané operace. Nebo odpověď negativní pokud obsah zprávy neodpovídá specifikaci, případně pokud se vyskytne chyba při vykonávání zprávy. Pokud server přijme neplatnou (chybné CRC, parita, ...) nebo neúplnou žádost vlivem komunikační chyby, neodesílá žádnou odpověď. Klient v tomto případě, po vypršení časového limitu na příjem odpovědi, indikuje chybu.

U negativních odpovědí (*exception responses*) je v poli kódu funkce hodnota z požadavku s nastaveným nejvyšším bitem. V datovém poli je kód chyby (*exception code*), který určuje její příčinu. Kódy chyb jsou definovány v následující tabulce 2.2.

Kód	Název	Popis
01	ILLEGAL INSTRUCTION	Instrukce není serverem podporována.
02	ILLEGAL DATA ADDRESS	Zadaná adresa je mimo podporovaný rozsah.
03	ILLEGAL DATA VALUE	Neplatná hodnota dat.
04	SLAVE DEVICE FAILURE	Neodstranitelná chyba při provádění požadavku serverem.
05	ACKNOWLEDGE	Indikace přijetí požadavku, jehož provádění bude trvat delší dobu. Klient pak může v dalších krocích zjišťovat průběh zpracování operace.
06	SLAVE DEVICE BUSY	Zařízení je zaneprázdněné. Vykonává se předchozí požadavek.
08	MEMORY PARITY ERROR	Při čtení ze souboru došlo k chybě parity.
0A	GATEWAY PATH UNAVAILABLE	Nelze vyhradit přenosovou cestu od vstupního k výstupnímu portu brány. Brána je chybně nastavená nebo přetížená.
0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	Cílové zařízení, umístěné za bránou, neodpovídá. Zařízení se nenachází v síti.

Tabulka 2.2: Chybové kódy MODBUS



Obrázek 2.4: Stavový diagram MODBUS transakce

2.5 Kódy funkcí

Kódy funkcí MODBUS mají tři kategorie:

- Veřejné kódy funkcí
 - jasně definované, veřejně zdokumentované
 - je garantována jejich unikátnost
 - schvalované společnostmi MODBUS-IDA.org
 - jsou k nim dostupné testy shody
 - obsahují jak přiřazené, tak nepřiřazené kódy rezervované pro budoucí použití
- Uživatelsky definované kódy funkcí
 - určené pro implementaci funkcí, které nejsou součástí specifikace
 - dva vyhrazené rozsahy – 65 až 72 a 100 až 110
 - lze požádat přesun funkcionality do veřejných kódů
 - negarantuje se unikátnost kódu funkce
- Rezervované kódy funkcí
 - Kódy funkcí používané některými společnostmi, které nejsou určeny pro veřejné použití

				Funkce	Pod-funkce
Datové přístupy	1-bitový přístup	Fyzické diskrétní vstupy	Čtení diskrétních vstupů	2	
		Cívky (<i>coils</i>) nebo interní bity	Čtení cívky	1	
			Zápis cívky	5	
			Zápis více cívek	15	
	16-bitový přístup	Fyzické vstupní registry	Čtení vstupního registru	4	
		Fyzické výstupní registry nebo interní registry	Čtení výstupního registru	3	
			Zápis jednoho registru	6	
			Zápis více registrů	16	
			Čtení / zápis více registrů	23	
			Zápis registru s maskováním	22	
			Čtení FIFO fronty	24	
		Záznamy v souborech		Čtení záznamu ze souboru	20
			Zápis záznamu do souboru	21	
Diagnostika			Čtení stavu	7	
			Diagnostika	8	0-18, 20
			Čtení čítače komunikačních událostí	11	
			Čtení záznamu komunikačních událostí	12	
			Sdělení identifikace klienta	17	
			Zjištění identifikace serveru	43	14
Ostatní			Zapouzdřený přenos	43	13, 14
			CANopen obecný odkaz	43	13

Tabulka 2.3: Veřejné kódy funkcí MODBUS

2.6 Implementace MODBUS na sériové lince

MODBUS na sériové lince je protokol typu Master-Slave, který se nachází ve 2. úrovni ISO/OSI modelu. Aplikační protokol MODBUS se nachází v 7. úrovni OSI modelu. Úrovně 3 až 6 ISO/OSI modelu jsou prázdné. Na sériové sběrnici je v roli klienta Master a v roli serveru Slave.

Na fyzické úrovni lze použít různá rozhraní (RS485, RS232). Běžně se používá dvou vodičové rozhraní RS485 (TIA/EIA-485). Dále se používají čtyřvodičové RS485 nebo také RS232 pro spoje na krátkou vzdálenost.

V systému typu Master-Slave je jeden uzel (Master), který vysílá příkazy a jeden z uzlů typu Slave je zpracuje a odešle odpověď. Na sběrnici může být až 247 Slave jednotek. Master posílá požadavky ve dvou režimech:

- unicast – zpráva je Master jednotkou adresována konkrétní Slave jednotce
- broadcast – Master jednotka posílá zprávu všem jednotkám na sběrnici, žádná ze Slave jednotek neodpovídá

Adresní prostor je rozdělen do tří částí:

- broadcast – adresa 0
- adresy jednotek – adresy 1 až 247
- rezervované adresy – adresy 248 až 255

Adresa každé jednotky musí být na dané sběrnici jedinečná. Adresa Master jednotky není pevně určena.

Sériový přenos může probíhat ve dvou režimech: RTU a ASCII. Všechna zařízení na sběrnici musí mít nastaven stejný režim přenosu a parametry sériové linky. RTU režim musí implementovat všechna zařízení, ASCII režim je volitelný. Popis režimu ASCII není předmětem tohoto dokumentu, proto zde bude popsán jen režim RTU.

2.6.1 Režim přenosu RTU

V režimu přenosu RTU (*Remote Terminal Unit*) je každý bajt přenesen v rámci o velikosti 11 bitů. Tento rámec má následující složení:

- 1 start bit
- 8 datových bitů – nejméně významný bit se odesílá jako první
- 1 paritní bit
- 1 stop bit

Standardně se používá se sudá (*even*) parita. Lze použít i jinou paritu, ale každé zařízení musí podporovat sudou paritu. Pro zajištění maximální kompatibility se staršími zařízeními je doporučeno podporovat i režim bez parity. V takovém případě jsou použity dva stop bity, aby byla zachována celková velikost rámce.



Obrázek 2.5: Pořadí bitů v režimu RTU

Rámec zprávy RTU je zabezpečen proti chybám přenosu CRC (*Cyclical Redundancy Checking*). Je použito 16-bitové CRC s generujícím polynomem $x^{16} + x^{15} + x^2 + 1$. Výpočet CRC začíná nastavením všech bitů 16-bitového registru na log. 1. Při odesílání se prohodí pořadí bajtů CRC, nejprve se odesílá nižší bajt. CRC se počítá z celé zprávy, tj. od adresy po poslední datový bajt.



Obrázek 2.6: Struktura rámce zprávy RTU

V režimu RTU jsou zprávy oddělené klidovým intervalem na sběrnici. Tento interval je dlouhý minimálně 3,5 krát doba trvání odeslání jednoho znaku. Zpráva se musí odesílat jako souvislý proud dat. Maximální přípustná mezera mezi jednotlivými znaky je 1,5 krát doba odesílání znaku. Pokud je doba mezi znaky delší, je příjem dat považován za nekompletní a přijatá data se zahodí.

Při vyšších přenosových rychlostech může docházet k vysokému zatížení CPU kvůli kontrole časů mezi přijímanými znaky a kontrole klidových intervalů. Proto se u přenosových rychlostí vyšších než 19200 baudů používají pevně dané časové limity. Maximální mezera mezi jednotlivými znaky je 750 μ s a minimální klidový interval je 1,750 ms.

Každé zařízení by mělo standardně komunikovat na rychlosti 19200 baudů a musí také podporovat rychlost 9600 baudů. Navíc může implementovat i jiné přenosové rychlosti. Odchylka od přenosové rychlosti musí být na straně vysílače maximálně 1%, přijímač musí pracovat bezchybně pokud odchylka přenosové rychlosti nepřesáhne 2%.

Na připojení zařízení ke sběrnici lze využít konektory RJ45, mini-DIN nebo D-Shell. Jejich zapojení je specifikováno v [2] kapitola 3.5 *Mechanical Interfaces*.

Pro vizuální diagnostiku by zařízení měla být vybavena LED indikací. Indikovat se může přenášení zprávy, chybový stav nebo zda je zařízení zapnuté.

3 Protokol dle normy ČSN EN 62056-21

Popis protokolu je součástí normy ČSN EN 62056: Měření elektrické energie – Výměna dat pro odečet elektroměru, řízení tarifu a regulaci zátěže – části 21: Přímá místní výměna dat [3]. Obrázky v této kapitole jsou převzaté ze zdrojové literatury.

Tato norma specifikuje protokol a technické vybavení pro místní výměnu dat elektroměrů. Připojení tarifního zařízení a příručního terminálu (*HHU – hand-held unit*) nebo jednotky, která má podobné funkce, může být trvalé nebo odpojitelné pomocí optické nebo elektrické vazby. Protokol je konstruován s ohledem na jeho použití v oblasti eklektického měření, hlavně z hlediska elektrického oddělení a zabezpečení dat. Norma je zkvalitněním již existujících systémů, které se používají v praxi. Přidává optický přenos, možnost změnu přenosové rychlosti, různé režimy protokolu.

3.1 Protokol přenosu dat

Protokol umožňuje několik režimů protokolu. Tyto režimy jsou označeny písmeny A, B, C, D a E. V režimech A, B, C a E je výměna dat obousměrná a iniciuje ji HHU vysláním požadavku. HHU funguje v režimech A, B a C jako zařízení typu Master a tarifní zařízení je typu Slave. V režimu E je HHU v roli klienta a tarifní zařízení jako server. Tyto režimy umožňují jak odečet, tak programování tarifního zařízení.

Režim protokolu D umožňuje pouze odečet, výměna dat je jednosměrná. Přenos je iniciován například stiskem tlačítka na elektroměru.

Tento dokument se bude podrobněji zabývat režimem protokolu C, který je použit v aplikaci pro odečet elektroměru.

3.2 Fyzikální vlastnosti

V normě jsou uvedeny tyto rozhraní pro přenos dat:

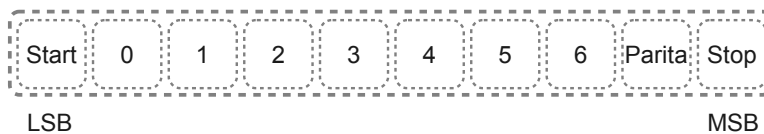
- Proudová smyčka – může být zapojena jedna nebo více podřízených stanic
- Rozhraní V.24 / V.28 – může být zapojena jedna nebo více podřízených stanic
- Optické rozhraní – komunikace pouze s jednou podřízenou stanicí

3.3 Znakový přenos

Typ použitého přenosu je sériový asynchronní, poloduplexní. Výchozí přenosová rychlost je 300 baudů. Další normalizované rychlosti jsou 600, 1200, 2400, 4800, 9600 a 19200 baudů. Zařízení mohou podporovat i jiné přenosové rychlosti. Formát přenášených znaků:

- 1 start bit
- 7 datových bitů
- 1 paritní bit
- 1 stop bit

Zabezpečení znaků je sudou paritou.



Obrázek 3.1: Struktura rámce znaku dle ČSN EN 62056-21

V režimu protokolu E lze používat transparentnost slabik. V tom případě se přenáší 1 start bit, 8 datových bitů, 1 stop bit.

3.4 Definice zpráv

3.4.1 Speciální znaky vyskytující se ve zprávách

Ve zprávách uvedených v následujících odstavcích se vyskytují tyto znaky:

- | | |
|-----------|-----------|
| • ACK 06h | • STX 02h |
| • NAK 15h | • ETX 03h |
| • CR 0Dh | • SOH 01h |
| • LF 0Ah | • EOT 04h |

3.4.2 Zabezpečení přenosu zpráv

Přenos dat může být zabezpečen kontrolním znakem bloku (BCC). Tento kontrolní znak musí splňovat ISO/IEEC 1155:1978. Kontrolní znak je vypočten z přenášených dat (znak STX, popř. SOH se do něj nezapočítává) a znaku ETX. Určí se jako exkluzivní logický součet (XOR) uvedených bajtů. Kontrolní znak se přenáší za znakem ETX.

3.4.3 Zpráva s požadavkem

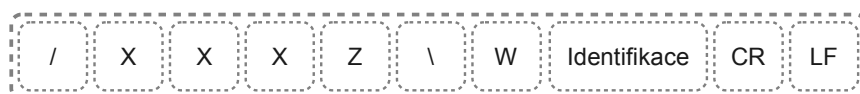
Počáteční zpráva od HHU, kterou začíná výměna dat mezi HHU a elektroměrem. Adresa zařízení je nepovinná a může obsahovat maximálně 32 znaků. Znaky mohou být číslice, velká a malá písmena (A až Z) a mezera. Nevýznamné nuly se nesmí vyhodnocovat, tj. jsou ignorovány. Pokud adresa není odeslána, musí tarifní zařízení odpovědět.



Obrázek 3.2: ČSN EN 62056-21: Zpráva s požadavkem

3.4.4 Zpráva s identifikací

Odpověď tarifního zařízení na zprávu s požadavkem.



Obrázek 3.3: ČSN EN 62056-21: Zpráva s identifikací

Kde:

- XXX – označení výrobce, tři velká písmena vyjma třetího, které může být malé. Pokud je malé, znamená to, že minimální doba odezvy je 20ms místo 200ms.
- Z – určení přenosové rychlosti – podle tabulky 3.1
- \ a W – nepovinné součásti pole Identifikace
- Identifikace – maximálně 16 tiskových znaků s výjimkou /, ! a \.

Přenosová rychlost závisí na rychlosti režimu protokolu. Výchozí rychlost je 300 baudů s výjimkou režimu protokolu D, kde činí 2400 baudů.

Rychlost [baud]	300	600	1200	2400	4800	9600	19200	Rezervováno pro budoucí použití
Režim B		A	B	C	D	E	F	G, H, I
Režim C	0	1	2	3	4	5	6	7, 8, 9

Tabulka 3.1: ČSN EN 62056-21: Přenosové rychlosti

3.4.5 Zpráva s volbou potvrzení/možnosti

Zprávu odesílá HHU a určuje v ní požadovanou přenosovou rychlost pro následující komunikaci a další charakteristiky přenosu.



Obrázek 3.4: ČSN EN 62056-21: Zpráva s volbou potvrzení/možnosti

Kde:

- V – řídicí znak protokolu
 - 0 – procedura normálního protokolu
 - 1 – procedura sekundárního protokolu
 - 2 – procedura HDLC protokolu
 - 3-9 – rezervováno pro budoucí použití

- Z – volba přenosové rychlosti, viz. tabulka 3.1
- Y – řídicí znak protokolu
 - 0 – odečet dat
 - 1 – programovací režim
 - 2 – dvojkový režim (HDLC)
 - 3-5 a A-Z – rezervováno pro budoucí použití
 - 6-9 – použití specifikované výrobcem

3.4.6 Zpráva s daty (s výjimkou programovacího režimu)

Normální odpověď tarifního zařízení na zprávu s volbou potvrzení/možnosti. Datový blok obsahuje například kompletní datový soubor.



Obrázek 3.5: ČSN EN 62056-21: Zpráva s daty

V datovém bloku se mohou vyskytovat všechny tiskové znaky, znaky CR a LF. Nemohou zde být znaky / a !.

3.4.7 Zpráva s potvrzením

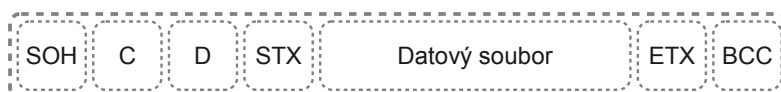
Jeden znak s hodnotou ACK.

3.4.8 Zpráva s požadavkem na opakování

Jeden znak s hodnotou NAK.

3.4.9 Zpráva s programovacím povel

Programování a blokově orientovaný přenos dat. Tuto zprávy může odesílat jak HHU, tak tarifní zařízení.



Obrázek 3.6: ČSN EN 62056-21: Programovací povel

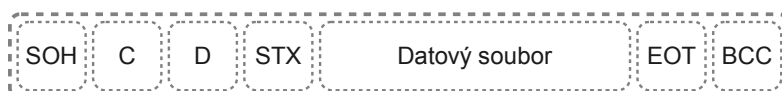
Kde:

- C – identifikátor zprávy s povel
 - P – povel s heslem
 - W – povel zápisu
 - R – povel odečtu

- E – povel provedení
- B – povel výstupu
- Další znaky jsou rezervované pro budoucí použití
- D – identifikátor typu povelu (označuje variantu povelu)
 - Tyto identifikátory specifikuje norma v kapitole 6.3.14 bod 19) [3]

3.4.10 Zpráva s programovacím povel používající nepovinné dílčí bloky

Používá se u režimu protokolu C a dlouhých zpráv. Tuto zprávy může odesílat jak HHU, tak tarifní zařízení.

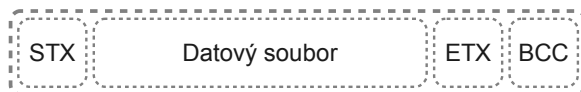


Obrázek 3.7: ČSN EN 62056-21: Programovací povel s dílčími bloky

Význam polí C a D je stejný jako u předchozí zprávy.

3.4.11 Zpráva s daty (programovací režim)

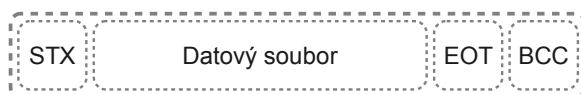
Používá se u blokově orientovaného přenosu dat v programovacím režimu. Používá se pro zápis a čtení dat z tarifního zařízení.



Obrázek 3.8: ČSN EN 62056-21: Zpráva s daty - programovací režim

3.4.12 Zpráva s daty (programovací režim) používající nepovinné dílčí bloky

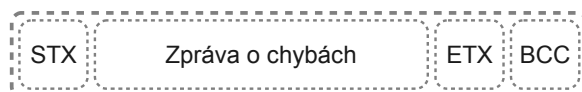
Používá se pro dlouhé zprávy blokově orientovaného přenosu dat v programovacím režimu. Používá se pro zápis a čtení dat z tarifního zařízení.



Obrázek 3.9: ČSN EN 62056-21: Zpráva s daty - programovací režim - dílčí bloky

3.4.13 Zpráva o chybách

Používá se u blokově orientovaného přenosu dat. Odesílá ji tarifní zařízení pokud nastane chyba při zápisu nebo čtení dat.

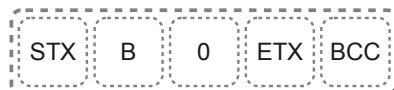


Obrázek 3.10: ČSN EN 62056-21: Zpráva o chybách

Pole zpráva o chybách může obsahovat maximálně 32 tiskových znaků vyjma (,), *, / a !. Je ohraničena jak u struktury datového souboru. Konkrétní obsah je záležitostí výrobce.

3.4.14 Zpráva o přerušení

Používá se u blokově orientovaného přenosu dat. Odesílá ji HHU a ukončuje tím přenos dat.



Obrázek 3.11: ČSN EN 62056-21: Zpráva o přerušení

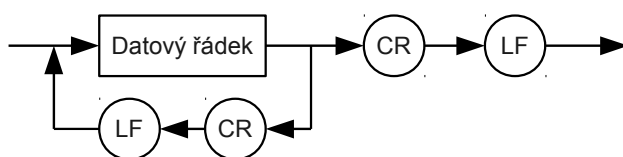
Kde:

- B – je hodnota znaku 'B' (podle popisu pole C u zprávy s programovacím povel 3.4.9)
- 0 – je hodnota znaku '0' (podle popisu pole D u zprávy s programovacím povel 3.4.9)

3.5 Struktura datového bloku a datového souboru

Tato kapitola obsahuje syntaktická schémata polí datový blok a datový soubor z definic zpráv.

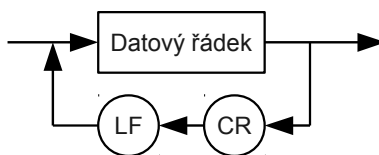
Syntaktické schéma datového bloku v režimu odečtu:



Obrázek 3.12: ČSN EN 62056-21: Datový blok – režim odečtu

V režimu odečtu může mít datový řádek maximálně 78 znaků.

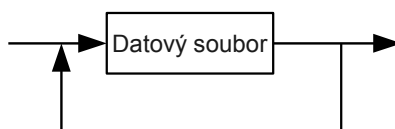
Syntaktické schéma datového bloku v programovacím režimu:



Obrázek 3.13: ČSN EN 62056-21: Datový blok – programovací režim

Datový řádek v programovacím režimu není zakončen znaky CR, LF.

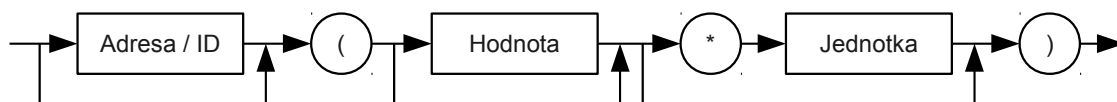
Syntaktické schéma datového řádku:



Obrázek 3.14: ČSN EN 62056-21: Datový řádek

Posloupnost datových řádků či datových souborů není pevně stanovena.

Syntaktické schéma datového souboru:



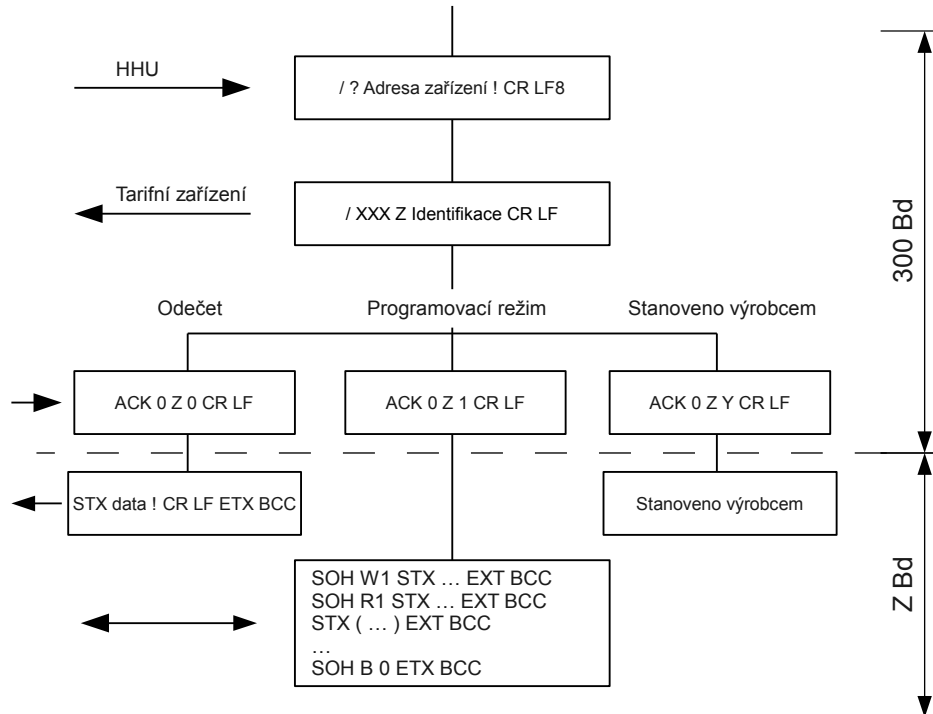
Obrázek 3.15: ČSN EN 62056-21: Datový soubor

Kde:

- Adresa / ID – maximálně 16 tiskových znaků s výjimkou (,), / a !.
- Hodnota – maximálně 32 tiskových znaků s výjimkou (,), *, / a !. U desetinných hodnot se použijí tečky (ne čárky) a jsou považovány za znaky.
- Jednotka – maximálně 16 tiskových znaků s výjimkou (,), / a !.

3.6 Režim protokolu C

Tento režim protokolu umožňuje odečet dat, programování a další činnosti stanovené výrobcem. Režim protokolu C je použit v aplikaci, která vznikla jako součást této bakalářské práce. Probíhá obousměrná komunikace s možností přepínání přenosové rychlosti.



Obrázek 3.16: ČSN EN 62056-21:Schéma režimu protokolu C

Komunikace začíná na přenosové rychlosti 300 baudů tím, že HHU odešle zprávu s požadavkem. Pokud není zadána adresa zařízení odpoví tarifní zařízení zprávou s identifikací. Pokud se komunikuje přes sběrnici a je na ní připojeno více tarifních zařízení, je potřeba zadat adresu tarifního zařízení, kterému se adresuje požadavek. Identifikace zařízení obsahuje informaci o maximální přenosové rychlosti a době odezvy podporované tarifním zařízením. Po uplynutí doby odezvy HHU odešle paket s potvrzením obsahující volbu režimu (odečet, programovací režim, jiný režim) a volbou přenosové rychlosti. Následná komunikace může probíhat buď na základní přenosové rychlosti 300 baudů nebo na rychlosti určené ve zprávě identifikace (pole Z). Komunikace je ukončena po odeslání všech dat tarifním zařízením. HHU nepotvrzuje úspěšnost přenosu. Pokud přenos selže, může HHU opakovat odeslání požadavku a tím se opakuje celý přenos.

Doba t_r mezi příjmem zprávy a odesláním odpovědi je minimálně 200 ms (resp. 20 ms) a maximálně 1500 ms. Pokud odpověď nebyla přijata vysílací zařízení čeká před dalším přenosem 1500 ms až 2200 ms. Maximální doba mezi dvěma odesílanými znaky ve zprávě je 1500 ms.

4 Porovnání protokolu MODBUS RTU (RS485) a protokolu dle ČSN EN 62056-21

Srovnání v této kapitole se týká pouze RS485 implementace obou protokolů. Porovnání ostatních implementací protokolů, jako je například implementace MODBUS na TCP/IP, nejsou předmětem tohoto dokumentu.

Protokoly lze srovnávat z různých hledisek. V této kapitole je porovnání protokolů podle jejich možností, struktury, efektivity a zabezpečení přenášených dat.

4.1.1 Základní vlastnosti protokolů

Oba protokoly jsou vytvořené pro nasazení v průmyslovém prostředí. Protokol MODBUS je obecný a využívá ho velké množství zařízení. Je to protokol typu Master-Slave, kde každému požadavku přísluší odpověď (s výjimkou broadcastu). Umožňuje propojení do větších sítí než je jedna větev RS485.

Protokol dle ČSN EN 62056-21 je navržen speciálně pro komunikaci s tarifními zařízeními jako jsou elektroměry. To znamená například snadný odečet a zabezpečení programovacího režimu. Umožňuje i jednosměrnou komunikaci iniciovanou událostí mimo specifikaci protokolu (například stisk tlačítka).

4.1.2 Struktura protokolů

Protokol MODBUS RTU má pevně daný rámec zprávy. Zpráva začíná adresou a končí CRC. Maximální velikost zprávy je stanovena na 256 bajtů. Protokol má řadu veřejných (předdefinovaných) funkcí a jednotnou správu chyb. K těmto funkcím může výrobce přidávat další dle potřeby. Pro potřeby tarifních zařízení se nabízí použití funkcí čtení/zápisu do souboru, kde tyto soubory by mohli být ekvivalentem datového souboru u protokolu dle ČSN EN 62056-21.

V protokolu dle ČSN EN 62056-21 není pevně stanovený rámec zprávy. Velikost zprávy není omezena zdola ani shora. Minimální velikost zprávy je jeden bajt (zprávy ACK, NAK). Díky neomezené velikosti zprávy je možné odeslat celý blok dat na jednou. Při odečtu elektroměru ZPA ED 310 se jedná o blok dat velikosti přibližně 1260 bajtů. Neomezení velikosti zprávy může mít negativní dopad na chování aplikací používající tento protokol. Pokud by zařízení vysílalo velmi dlouhou zprávu, nemusí se vejít do paměti vyhrazené v přijímači. Přijímač nezná předem velikost přijímané zprávy a nemá jak rozhodnout, jestli je zpráva platná nebo došlo k chybě. Proto je vhodné stanovit maximální velikost zprávy a větší zprávy vyhodnotit jako chybu.

Velikost rámce pro přenos znaku u MODBUS protokolu je 11 bitů a přenáší se 8 datových bitů. U protokolu dle ČSN EN 62056-21 je velikost rámce pro přenos znaku 10 bitů a přenáší se 7 datových bitů. Toto neplatí pro transparentní režim protokolu E, kde se přenáší 8 datových bitů.

Protokol MODBUS přenáší binární data, ne jen tisknutelné znaky. Zatímco protokol dle ČSN EN 62056-21 přenáší jen tisknutelné znaky s tím, že některé znaky vyhrazené a nemohou se vyskytovat v datech zprávy. Tyto znaky se používají k ohraničení datového souboru.

U protokolu MODBUS je začátek zprávy dán vypršením klidové doby na sběrnici. Oba protokoly

vyžadují klidovou dobu na sběrnici mezi zprávami a definují maximální prodlevu mezi vysílanými znaky. U MODBUS protokolu jsou oba tyto časy hodně malé (maximální prodleva mezi znaky je 1,5 násobek času odeslání znaku – cca 1,7 ms u rychlosti 9600 baudů) a je obtížné je kontrolovat pouze softwarovými prostředky. Proto se využívají časovače mikropočítačů, které hlídají tyto časové limity. U protokolu dle ČSN EN 62056-21, kde oba časy jsou větší než jedna sekunda, již problémy s implementací kontroly časových limitů nejsou ani v čistě softwarovém provedení.

4.1.3 Zabezpečení přenášených dat

Protokol MODBUS i protokol dle ČSN EN 62056-21 zabezpečují přenášená data proti chybám při přenosu dat na dvou úrovních: na úrovni každého přenášeného znaku a na úrovni zprávy.

Zabezpečení na úrovni znaků zajišťuje parita. Standardně se používá sudá parita. Protokol MODBUS povoluje i komunikaci s jiným typem parity, případně bez parity. V takovém případě je místo paritního bitu stop bit navíc aby byla zachována velikost rámce znaku.

Zpráva je u protokolu MODBUS zakončena 16-bitovým CRC. Ve spojení s paritou a maximální velikostí zprávy 256 bitů je možnost neodhalení chyb nízká.

Protokol dle ČSN EN 62056-21 nezabezpečuje všechny přenášené zprávy. Zabezpečeny jsou jen zprávy s daty. Zpráva s požadavkem, identifikací a ostatní krátké zprávy zabezpečeny proti chybám přenosu nejsou. K zabezpečení zpráv se používá 8-bitový kontrolní součet, určený jako exkluzivní součet přenášených bajtů. Vzhledem k tomu, že zprávy mohou být velké (v řádu kilobajtů), tak je toto zabezpečení samo o sobě slabé. Ale vzhledem k tomu, že jsou jednotlivé znaky chráněny paritou a používají se jen tisknutelné znaky (z výjimkou řídicích), tak je riziko neodhalení chyby při přenosu nízké.

4.1.4 Efektivita protokolů

Efektivitu protokolů srovnám z hlediska poměru velikosti zprávy ku velikosti přenášených dat.

U protokolu MODBUS RTU se přenáší minimálně 4 bajty, plus datová část. Tato datová část může být teoreticky vyplněna jedním souborem záznamů. To znamená, že při použití kódu funkce 20 – čtení záznamu ze souboru, zbývá $256 - 4 - 8 = 244$ bajtů pro samotná data. Tyto datové bajty mohou obsahovat libovolnou hodnotu. Pro větší počet dat je potřeba odeslat více zpráv. Protože načtená data jsou až v odpovědi musí této zprávě předcházet požadavek. Tento požadavek má velikost 12 bajtů. V nejlepším případě je poměr velikosti zprávy ku přenášeným datům 268:244 bajtů.

Protokol dle ČSN EN 62056-21 přenáší data v textové podobě a využívá jen 7 bitů na jeden znak. Maximální velikost hodnoty datového souboru je 32 tiskových znaků. K ní je potřeba připočítat identifikaci datového souboru – x znaků. Na datovém řádku je místo pro dva datové soubory s velikostí hodnoty 32 znaků, celková velikost datového řádku nesmí přesáhnout 78 znaků. V datovém bloku jsou datové řádky oddělené dvěma znaky (CR, LF). Obálka datového bloku ve zprávě má velikost 6 znaků. Zpráva se na rozdíl od MODBUS RTU nepotvrzuje. Pro odeslání přibližně stejného množství dat, jako u předchozího příkladu protokolu MODBUS RTU, je potřeba 8 datových souborů, tj. $8 \cdot 32 = 256$ znaků. Na rozlišení těchto souborů stačí jeden znak (identifikace datového souboru – x). Celková velikost zprávy je tedy $8 \cdot (32 + 1 + 2) + 4 \cdot 2 + 6 = 294$ znaků. Poměr velikosti zprávy ku přenášeným datům činí v nejlepším případě 294:256 znaků.

Rozdíl poměrů přenesených dat ku velikosti zprávy mezi těmito dvěma protokoly v tomto srovnání je menší než 5%.

5 Aplikace pro automatizované odečty

5.1 Požadavky na aplikaci

Aplikace bude provozována na průmyslových počítačích architektury Intel x86 s operačním systémem Windows XP Embedded. Tyto počítače mají dotykové LCD s nízkým rozlišením (800x600, 1024x768) oproti běžným monitorům, při tvorbě grafické části aplikace je na to potřeba brát ohled. Aplikace provádí odečet elektroměru ZPA ED 310, který je připojen k počítači přes sériovou sběrnici RS485. Elektroměr komunikuje pomocí protokolu dle normy ČSN EN 62056-21 s použitím režimu protokolu C. Komunikace tímto protokolem je popsána v kapitole 3 tohoto dokumentu. Specifikace položek (datového souboru) v datovém bloku odečtu je uvedena v zákaznickém listu elektroměru, který je jako příloha B tohoto dokumentu.

5.2 Návrh aplikace

Pro větší flexibilitu se aplikace rozdělila na dvě podle typu uživatelského rozhraní: grafickou a konzolovou. Konzolová aplikace se snadno zakomponuje do jiného systému.

Konzolová aplikace provede odečet elektroměru podle zadaných parametrů příkazové řádky. Načtený odečet pak zobrazí na standardní výstup nebo uloží do souboru. Konzolová aplikace používá jen nastavení z příkazové řádky. Jedná se o samostatný spustitelný soubor, neinstaluje se.

Aplikace s grafickým uživatelským rozhraním umožňuje odečet elektroměru, uložení odečtu do souboru, zobrazení hlavních údajů z odečtu, zobrazení všech údajů z odečtu a načtení uloženého souboru odečtu. Veškerá konfigurace aplikace se provádí přes grafické rozhraní. Aplikace si pamatuje posledně použité hodnoty nastavení. Aplikace se skládá ze spustitelného souboru, dynamických knihoven a konfiguračního souboru. Instalátor provede instalaci aplikace na cílový počítač a případně vytvoří odkazy na ploše a ve „start menu“. Díky instalátoru je umožněna snadná odinstalace aplikace standardní cestou.

Formát datového souboru používaného aplikacemi, je shodný s formátem datového bloku odečtu.

Protože se při odečtu elektroměru vždy přenáší celý datový blok a není možné specifikovat vybrané data pro odečet, není realizována část zadání – vyčtení pouze zvolených položek.

Aplikace byly napsány v jazyku C++. Tento jazyk byl zvolen proto, že ve vyšších jazycích (jako je například jazyk Java) nejsou standardně knihovny pro práci se sériovým portem. Pro jazyk Java existují knihovny třetích stran pro práci se sériovým portem, ale tyto projekty nejsou dostatečně stabilní a není žádná záruka jejich dalšího vývoje a údržby. Případně tyto knihovny neumožňují plnou kontrolu sériového portu, tak jako je to možné přes Win32 API. Grafická aplikace je vytvořena za použití knihoven Qt [4].

Pro tvorbu aplikací bylo použito tyto vývojové prostředí:

- MS Visual Studio C++ 2010 Express Edition – pro tvorbu konzolové aplikace
- Qt Creator 2.4 a knihovny Qt verze 4.8 (s využitím kompilátoru obsaženém v MS Visual Studio C++ 2010) – pro tvorbu grafické aplikace

5.3 Implementace aplikace

5.3.1 Konzolová aplikace

Aplikace je rozdělena do několika základních modulů:

- *SerialPort* – modul pro práci se sériovým portem
- *Electrometer* – modul pro komunikaci s elektroměrem
- hlavní modul aplikace

Moduly pro práci se sériovým portem a hlavní modul aplikace jsou závislé na platformě. Vyskytují se v nich volání Win32 API [5] pro komunikaci se sériovým portem a nastavení kódování znaků vypisovaných na konzoli. Kromě volání těchto funkcí aplikace používá standardní knihovnu C++. Pokud by aplikace měla běžet pod jiným operačním systémem, je nutné nahradit modul *SerialPort* a nahradit nastavení kódování vypisovaných znaků.

Aplikace má české uživatelské rozhraní. Všechny chyby, které mohou vzniknout při chodu aplikace, mají přiřazený svůj číselný kód. Jednotlivé moduly mají svoje rozsahy kódů chyb. Texty vypisované na konzoli se vyskytují je v hlavním modulu aplikace. Takto lze snadno provést případný překlad všech textů v aplikaci.

Moduly nižší úrovně aplikace (práce se sériovým portem a komunikace s elektroměrem) jsou součástí i aplikace s grafickým uživatelským rozhraním.

Následuje popis jednotlivých modulů konzolové aplikace.

Modul pro práci se sériovým portem

Název modulu je *SerialPort* a jeho funkčnost je realizována ve třídě *TSerialPort* a využívá třídu *ESerialPort* pro zpracování výjimek vzniklých v tomto modulu.

Metoda	Popis
<i>Konstruktor</i>	Otevře sériový port a nastaví jeho parametry jako jsou přenosová rychlost, počet datových bitů, parita, počet stop bitů.
<i>SetTimeouts</i>	Nastaví časové limity pro čtení a zápis dat z/na sériový port.
<i>SetBaudRate</i>	Nastaví přenosovou rychlost.
<i>Send</i>	Odešle data na sériový port.
<i>RecvN</i>	Přijme zadaný počet bajtů ze sériového portu.
<i>ReceivedBytesCount</i>	Zjistí počet přijatých bajtů ve vstupní vyrovnávací paměti portu.
<i>EnumPorts</i>	Statická metoda. Zjistí názvy sériových portů aktuálně přítomných v počítači.

Tabulka 5.1: Třída *TSerialPort*

Konstruktor třídy generuje v případě chyby (například pokud nelze otevřít port, protože už je otevřen) výjimky třídy *ESerialPort*. Tato třída obsahuje atribut *error* s kódem chyby, která výjimku způsobila.

Metody třídy výjimky negenerují. Místo toho vracejí hodnotu *false*, případně jinou návratovou hodnotu signalizující chybu pokud nemají návratový typ *boolean*.

Uzavření sériového portu se provádí v destruktoru třídy.

Modulu *SerialPort* je přiřazen rozsah kódu chyb od 200 do 299.

Modul pro komunikaci s elektroměrem

Název modulu je *Electrometer* a jeho funkčnost je realizována ve třídě *TElectrometer*. V tomto modulu je realizován odečet elektroměru pomocí protokolu dle ČSN EN 62056-21, režim protokolu C. Umožňuje volby přenosové rychlosti a případné zadání adresy tarifního zařízení, se kterým se má komunikovat. Modul komunikace s elektroměrem používá modul *SerialPort*. Tento modul negeneruje výjimky, případné chyby jsou indikovány návratovou hodnotou volané metody a specifikace chyby je v atributu *lastError*. Modulu *Electrometer* je přiřazen rozsah kódu chyb od 100 do 199. V atributu *lastError* třídy *TElectrometer* se mohou vyskytovat i chybové kódy z modulu *SerialPort*, který se zde používá.

Metoda	Popis
<i>Konstruktor</i>	Inicializuje objekt.
<i>ReadAllData</i>	Provede odečet elektroměru, podle zadaných parametrů metody.
<i>ReceiveIdentification</i>	Přijme identifikaci elektroměru.
<i>RecvDataBlock</i>	Přijme datový blok od elektroměru.

Tabulka 5.2: Třída *TElectrometer*

Protože je celý odečet elektroměru vykonán v rámci jednoho volání metody *ReadAllData* je třída *TElectrometer* vybavena mechanismem pro předčasné ukončení odečtu a indikací počtu přečtených bajtů z elektroměru. Tento mechanismus se v konzolové aplikaci nevyužívá, aplikace končí vždy až po dokončení odečtu nebo výskytu chyby. Při uzavření okna s konzolí nebo stisku kláves Ctrl+C se o ukončení aplikace postará systém Windows.

Pokud operace odeslání požadavku a přijetí identifikace proběhne v pořádku, metoda *ReceiveIdentification* uloží přijatou identifikaci do veřejného atributu *identification* třídy.

Metoda *RecvDataBlock* v případě úspěšného přijetí datového bloku uloží tento blok do veřejného atributu *dataBlock*.

Hlavní modul konzolové aplikace

V tomto modulu se provádí rozpoznání parametrů zadaných na příkazové řádce, provedení odečtu podle zadaných parametrů za využití modulu *Electrometer* a následný výpis odečtu na konzoli nebo jeho uložení do souboru.

Program má jeden povinný parametr – název sériového portu. Ostatní parametry jsou volitelné: nastavení přenosové rychlosti, adresy zařízení, nastavení uložení odečtu do souboru. Název souboru do kterého se odečet uloží může být buď zadán jako parametr programu nebo se odvodí od komunikační adresy elektroměru. Název takového souboru má potom tento formát: *em_XXXXX.txt*, kde *XXXXX* je komunikační adresa elektroměru. Soubor se uloží do aktuálního adresáře.

Pokud dojde k chybě odečtu program končí výpisem chyby na konzoli. Pokud parametry příkazové řádky nejsou platné, odečet se neprovádí a program končí výpisem nápovědy na konzoli. Aplikace nastavuje návratovou hodnotu podle výsledku operace.

Program interně používá pro práci s textovými řetězci datový typ `wchar_t` resp. `std::wstring` ze standardní knihovny C++. Toto je nativní typ pro podporu znaků Unicode v systému Windows. Pro bezproblémové zobrazování diakritiky v okně konzole je potřeba mít nastavený TrueType font. U rastrových fontů se nebude zobrazovat diakritika, pokud není zvolena v systému Windows jako výchozí znaková sada Win1250.

Hodnota	Popis
0	Odečet proběhl v pořádku.
-1	Chybný parametr na příkazové řádce.
-2	Nepodařilo se zapsat odečet do souboru.
100	Chyba při zápisu na sériový port.
101	Chyba při čtení identifikace elektroměru. Byla přijata nějaká data, ale nebyla to identifikace.
102	Chyba při příjmu dat. Vypršel časový limit na příjem dat.
103	Chybný začátek zprávy s datovým blokem.
104	Chybný kontrolní součet.
105	Elektroměr neodpovídá. Na požadavek nepřišla žádná odpověď.
200	Nelze otevřít sériový port.
201	Chyba při nastavování portu. Nelze načíst nastavení časových limitů.
202	Chyba při nastavování portu. Nelze nastavit časové limity operací.
203	Chyba při nastavování portu. Nelze nastavit parametry portu.
204	Chyba při nastavování portu. Nelze nastavit sledování událostí na portu.

Tabulka 5.3: Návrátové hodnoty konzolové aplikace

5.3.2 Aplikace s grafickým uživatelským rozhraním

Veškerá funkčnost aplikace je dostupná z hlavního okna aplikace. Rozdělení této podkapitoly není po modulech jako u konzolové části, ale podle prováděných funkcí.

Odečet elektroměru

Vlastní odečet je realizován pomocí modulů konzolové aplikace 5.3.1 *SerialPort* a *Electrometer*. Protože odečet může trvat relativně dlouhou dobu (přibližně 50 sekund při základní rychlosti 300 baudů), není metoda *ReadAllData* volána z hlavního vlákna aplikace. Práci s tímto vláknem zajišťuje třída *TReadingThread* a aplikace může v hlavním vlákně zpracovávat požadavky uživatele.

Metoda	Popis
<i>run</i>	Provádí odečet podle nastavených atributů třídy.
<i>Terminate</i>	Signalizuje objektu třídy <i>Electrometer</i> , aby předčasně ukončil odečet elektroměru.
<i>BytesRead</i>	Vrátí aktuální počet přijatých bajtů od elektroměru.

Tabulka 5.4: Třída *TReadingThread*

Pro ukončení odečtu a zjištění počtu přijatých bajtů se používají atributy *terminate* a *bytesRead* třídy *TElectrometer*, které jsou periodicky kontrolovány a aktualizovány při odečtu.

V průběhu odečtu se zobrazuje počet přijatých bajtů ve stavovém řádku. Pro tuto činnost je využit časovač, který poskytují knihovny Qt, a periodicky aktualizuje stavový řádek. Po ukončení odečtu se ve stavovém řádku zobrazí informace o délce trvání odečtu.

Zobrazení dat odečtu

Odečet elektroměru je zobrazen ve třech formátech:

- hlavní data odečtu – jako jsou výrobní číslo elektroměru, hodnoty dodávky a odběru pro jednotlivé tarify a fáze, aktuální tarif elektroměru
- všechna data odečtu – zobrazená v tabulce s textovým popisem u každé položky
- data odečtu – zdrojová data, tak jak byla přijata z elektroměru

Data odečtu mohou být načtena ze souboru nebo odečtena z elektroměru.

Pro snazší práci s daty z odečtu se využívá třídy *TPrefix*. Každý objekt této třídy obsahuje prefix a k němu textový popis. Tyto hodnoty jsou obsažené v zákaznickém listu elektroměru. Objekty třídy *TPrefix* se vytvářejí v metodě *FillPrefixes* a ukládají do seznamu prefixů, který je atributem třídy.

Atribut	Popis
<i>prefix</i>	Textová hodnota prefixu.
<i>name</i>	Textový popis datového souboru.
<i>vauleCntrl</i>	Ukazatel na vizuální prvek zobrazující hodnotu v přehledu v hlavním okně (QLineEdit). NULL pokud není prefixu přiřazen.
<i>unitsLbl</i>	Ukazatel na vizuální prvek v hlavním přehledu zobrazující jednotky (QLabel). NULL pokud není prefixu přiřazen.

Tabulka 5.5: Třída *TPrefix*

U hodnoty, která se zobrazuje v přehledu hlavních dat odečtu, je nastavený atribut *valueCntrl*. Případně je nastavený i *unitsLbl*, pokud se od má od dané hodnoty použít pro zobrazení i jednotka. Předpokládá se, že jednotky jsou u hodnot stejného typu stejné. Například jednotky pro dodávku v různých tarifech. Tyto atributy jsou nastavené na adresy vizuálních prvků je zobrazují.

Při načítání dat, ze souboru nebo po odečtu, se postupně procházejí jednotlivé datové soubory. Podle ID datového souboru, které odpovídá prefixu se určí textový popis a zda odpovídá hodnotě vizuální prvek v hlavním přehledu. Hodnoty s popisem a jednotkami se potom přidávají do tabulky všech hodnot a případně se zobrazí v hlavním přehledu.

V současné implementaci je nastavení prefixů zadáno napevno v metodě *FillPrefixes*. Pro snazší

použití aplikace s elektroměry s jiným souborem prefixů by metoda *FillPrefixes* mohla načítat definici prefixů ze souboru.

Nastavení aplikace

Pro nastavení čísla portu, přenosové rychlosti a komunikační adresy elektroměru slouží speciální dialogové okno. Ve volbě sériového portu se zobrazují jen aktuálně instalované porty v systému Windows. Ke zjištění těchto portů se používá statické metody *EnumPorts* třídy *TSerialPort*. Nastavení zobrazené v tomto dialogovém okně se ukládají při ukončení aplikace do konfiguračního souboru. Při spuštění aplikace se odtud parametry načtou. Pokud tento soubor neexistuje nebo nějaký parametr v něm chybí, použije se jeho výchozí hodnota. Konfigurační soubor se nachází je stejném adresáři jako spustitelný, má příponu *ini* a jméno jako spustitelný soubor.

Poznámky k implementaci

Tato aplikace využívá knihovny Qt. Díky jejím možnostem, lze snadno přeložit všechny texty vyskytující se v aplikaci, pokud by to bylo potřeba. Pak by bylo možné zvolit jazyk uživatelského rozhraní aplikace.

Pro ikony aplikace byly použity volně použitelné ikony ze serveru <http://icongal.com>.

Instalační soubor grafické aplikace byl vytvořen pomocí Inno Setup <http://www.jrsoftware.org>.

Při implementaci byla zjištěna potenciální neshoda mezi komunikačním protokolem dle ČSN EN 62056-21 a reálným odečtem elektroměru ZPA DE 310. V datovém souboru podle protokolu má být mezi hodnotou a jednotkou znak oddělovací hvězdičky *, viz Obrázek 3.15: ČSN EN 62056-21: Datový soubor. Ale v datovém souboru odečtu je hodnota od jednotky v některých případech oddělena znakem mřížky #. Je to u hodnot, které vznikly jako součet (například celková dodávka – součet dodávky v jednotlivých fázích). Podle specifikace protokolu by se obsah takového datového souboru měl brát jen jako jedna hodnota. Aplikace ji ale vyhodnocuje jako spojení hodnoty a jednotky.

Drobnou odchylkou od normy je i to, že elektroměr nepodporuje jen dvě přenosové rychlosti. Podle normy by měl elektroměr podporovat jen základní rychlost 300 baudů a rychlost přijatou ve zprávě s identifikací (Zpráva s identifikací pole Z). Elektroměr ale podporuje i přenosové rychlosti mezi základní a rychlostí specifikovanou ve zprávě s identifikací. U zapůjčeného elektroměru jsou to standardní přenosové rychlosti v intervalu 300 až 4800 baudů.

5.4 Testy aplikace

Konzolová aplikace byla testována na funkčnost s využitím všech možností vstupních parametrů. Dále byla testována kontrola detekce chyb, jako jsou různé komunikační chyby (dočasné rozpojení linky RS485), nemožnost otevření sériového portu nebo chyb při zápisu dat do souboru.

U aplikace s grafickým rozhraním byly provedeny stejné testy jako u konzolové aplikace. Navíc bylo ověřeno uchování nastavení a načítání dat odečtu ze souboru.

Testy probíhaly na systému Windows XP Home Edition, který je nejbližší cílovému operačnímu systému.

Obě aplikace těmito testy prošly úspěšně.

6 Závěr

Hlavním cílem této bakalářské práce bylo vytvoření aplikace pro odečet elektroměrů. Druhým cílem bylo porovnání komunikačních protokolů používaných u elektroměrů. Aplikace byla zhotovena ve dvou provedeních: konzolovém (pro snazší integraci do existujících systémů) a s grafickým uživatelským rozhraním. Na základě provedených testů lze soudit, že aplikace jsou plně funkční. Avšak testy zatím nebyly provedeny v cílovém prostředí.

Práce dále ukazuje dva komunikační protokoly ČSN EN 62056-21 a MODBUS RTU, které se používají ke komunikaci s elektroměry. Obsahuje jejich srovnání z hlediska zabezpečení přenášených dat a efektivity protokolu.

Při analýze protokolu použitého ke komunikaci s elektroměrem (protokol dle ČSN EN 62056-21) bylo zjištěno, že díky jeho vlastnostem nelze plně realizovat zadání práce. Není možné vybrat položky, které mají být vyčteny, ale vždy se vyčtou všechny položky odečtu elektroměru.

6.1 Možná rozšíření

Aplikace s grafickým uživatelským rozhraním aktuálně podporuje jen jednu sadu prefixů datového souboru. Tímto je vázána na konkrétní zákaznické provedení firmware elektroměru. Pro podporu jiných provedení elektroměru by bylo vhodné rozšířit aplikaci o možnost načtení konfigurace elektroměru ze souboru.

Cílovou platformou aplikací je Microsoft Windows. Protože je aplikace psaná v jazyku C++ s použitím knihovny Qt nebude problém ji upravit pro běh na jiné platformě. Největší část úprav se by se týkala práce se sériovým portem. Nicméně toto nebylo předmětem zadání práce.

7 Použitá literatura

- [1] Modbus protocol specification
http://modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf, stav z: 20.5.2012
- [2] Modbus Serial Line Protocol and Implementation Guide V1.02
http://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf, stav z: 20.5.2012
- [3] norma ČSN EN 62056-21 - Měření elektrické energie - Výměna dat pro odečet elektroměru řízení tarifu a regulace zátěže - Část 21: Přímá místní výměna dat, datum účinnosti: 1.4.2004
- [4] Qt – A cross-platform application and UI framework, <http://qt.nokia.com>, stav z: 23.5.2012
- [6] zákaznický list elektroměru ZPA ED 310 DR 30, vytvořen: 29.9.2009
- [5] Win32 API reference, <http://msdn.microsoft.com/en-us/library>, stav z: 10.5.2012
- [7] Modbus organization, <http://www.modbus.org>, stav z: 20.5.2012
- [8] norma ČSN EN 61036 - Střídavé statické činné elektroměry (třída 1 a 2), datum účinnosti: 1.1.1998
- [9] norma ČSN EN 62052-11 - Vybavení pro měření elektrické energie (AC) - Všeobecné požadavky, zkoušky a zkušební podmínky - Část 11: Elektroměry, datum účinnosti: 1.4.2004

Příloha A: Obsah přiloženého CD

- readme.txt - popis obsahu CD
- text
 - CvekMartin-BP.pdf - text bakalářské práce ve formátu pdf
 - CvekMartin-BP.odt - text bakalářské práce ve formátu Openoffice
- exe
 - AutoOdecetC.exe - přeložená konzolová aplikace
 - AutoOdecet-setup.exe - instalátor grafické aplikace
- src
 - AutoOdecetC - adresář se zdrojovými kódy konzolové aplikace
 - AutoOdecet - adresář se zdrojovými kódy grafické aplikace

Příloha B: Nastavení odečtu elektroměru ZPA ED 310

Nastavení je převzato ze zákaznického listu [6].

Jméno položky	Des.míst	Prefix standardní hodnota	Celkem	L1	L2	L3
Chyba elektroměru		F.F	X			
Zákaznické pořadové číslo		0.0				
Výrobní číslo		C.1	X			
Komunikační adresa		C.90	X			
Odběr + dodávka v T1	0 až 4	1.8.1	X			
Odběr + dodávka v T2		1.8.2	X			
Odběr + dodávka v T3		1.8.3				
Odběr + dodávka v T4		1.8.4				
Odběr + dodávka v ΣTx		1.8.0	X			
Dodávka v T1		2.8.1				
Dodávka v T2		2.8.2				
Dodávka v T3		2.8.3				
Dodávka v T4		2.8.4				
Dodávka v ΣLx v ΣTx		2.8.0				
Odběr v Lx v ΣTx		21.8.0 41.8.0 61.8.0		X	X	X
Dodávka v Lx v ΣTx		22.8.0 42.8.0 62.8.0		X	X	X
Datum posledního odečtu		C.9.3	X			
Datum a čas 1		0.9.1				
Datum a čas 2		0.9.2				
Počet výp.ku napětí fáze Lx		C.7.1 C.7.2 C.7.3		X	X	X
Počet výpadků napájení		C.7.0	X			
Konstanta IR [imp/kWh]		0.3.0				
Konstanta S0 [imp/kWh]		0.3.3				
Datum kalibrace						
Identifikace konfigurace		0.2.0				
Zákaznické číslo provedení		0.2.1	X			
Provedení elektroměru						
Datum konfigurace		C.2.1				
Doba čít. odb. + dod. v T1		C.81.1				
Doba čít. odb. + dod. v T2		C.81.2				
Doba čít. odb. + dod. v T3		C.81.3				
Doba čít. odb. + dod. v T4		C.81.4				
Doba čít. odb. + dod. v ΣTx		C.81.0				
Doba čít. dod. v T1		C.82.1				
Doba čít. dod. v T2		C.82.2				
Doba čít. dod. v T3		C.82.3				
Doba čít. dod. v T4		C.82.4				
Doba čít. dod. v T1-T4		C.82.0				
Provozní čas		C.50	X			
Efektivní napětí	1	32.7 52.7 72.7				
Efektivní proud	3	31.7 51.7 71.7				
Ief L1+L2+L3 se znaménky	3	91.7				
Výkon	4	1.6.1 1.6.2 1.6.3				
Okm. účinník - cos φ	3	33.7 53.7 73.7				
Maximální proud	3	31.6.0 51.6.0 71.6.0		X	X	X
Max. výkon ve fázi	3	21.6.0 41.6.0 61.6.0		X	X	X
Čas od nulování spotřeby		0.9.7				
Čas od nulování maxim		0.9.3				
Nulovatelný čítač času 1		C.9.1				
Nulovatelný čítač času 2		C.9.2				
Aktuální tarif		0.6.0	X			

Příloha C: Datový blok odečtu elektroměru

```
/ZPA4ZE310.iv30_007
F.F(000000)
0.0.1(02477552)
C.1(02477552)
C.90(477552)
1.8.1(0000051.3#kWh)
1.8.2(0000000.5*kWh)
1.8.0(0000052.9*kWh)
2.8.1(0000000.0#kWh)
2.8.2(0000000.0*kWh)
2.8.0(0000000.0*kWh)
21.8.0(0000050.9*kWh)
41.8.0(0000000.9*kWh)
61.8.0(0000000.9*kWh)
22.8.0(0000000.0*kWh)
42.8.0(0000000.0*kWh)
62.8.0(0000000.0*kWh)
C.9.3(28-01-10 13:22)
0.9.1()
0.9.2()
C.7.1(0004)
C.7.2(0024)
C.7.3(0018)
C.7.0(0031)
0.3.0(10000.000*i\kWh)
0.3.3(00250.000*i\kWh)
0.C.1(08-07-31 11:59)
0.2.0(YYYY)
0.2.1(ED310_I_DR_30)
0.4.2(xxxx)
C.2.1(08-07-31 11:59)
C.81.1(00009666:21#h:min)
C.81.2(00000000:09*h:min)
C.81.0(00009666:46*h:min)
C.82.1(00000000:01#h:min)
C.82.2(00000000:00*h:min)
C.82.0(00000000:01*h:min)
C.50(00009667:36*h:min)
32.7(240.9*v)
52.7(000.5*v)
72.7(000.4*v)
31.7(000.017*A)
51.7(000.000*A)
71.7(000.000*A)
91.7(000.015*A)
1.6.1(00.0018*kW)
1.6.2(00.0000*kW)
1.6.3(00.0000*kW)
33.7(0.437*cos)
53.7(0.000*cos)
73.7(0.000*cos)
31.6.0(000.382*A)
51.6.0(000.234*A)
71.6.0(000.234*A)
91.6.0(000.703*A)
21.6.0(000.050*kW)
41.6.0(000.000*kW)
61.6.0(000.050*kW)
81.6.0(000.101*kW)
0.9.7(00009667:34*h:min)
0.9.3(00009665:07*h:min)
C.9.1(00009667:34*h:min)
C.9.2(00009667:34*h:min)
0.6.0(1)
!
```


Příloha D: Uživatelská příručka

D.1 Konzolová aplikace

Konzolová aplikace je samostatný spustitelný soubor – *AutoOdecetC.exe*.

Formát příkazové řádky:

```
AutoOdecetC.exe [-s | -v xxx] [{-n | -N} aaa] COMx [vystup]
```

Parametry příkazové řádky:

- **-s** komunikace s elektroměrem bude probíhat na maximální rychlosti
- **-b xxx** komunikace s elektroměrem bude probíhat na maximální rychlosti
xxx může být 300, 600, 1200, 2400, 4800, 9600, 19200
- **-n aaa** bude se komunikovat se zařízením s adresou **aaa**, výstup dat na konzoli
- **-N aaa** bude se komunikovat se zařízením s adresou **aaa**, výstup dat do souboru
formát názvu souboru: *em_aaa.txt*
- **COMx** název sériového portu
- **vystup** název souboru, kam se mají uložit data odečtu

Po spuštění bez zadání parametrů, vypíše program nápovědu. Zprávy o chybách se vždy vypisují na konzoli. Po ukončení programu je nastaven návratový kód, dle: Tabulka 5.3: Návratové hodnoty konzolové aplikace.

D.2 GUI aplikace

Po spuštění se zobrazí hlavní okno aplikace s aktivní záložkou přehled odečtu. Poté je možno načíst datový soubor z disku nebo provést odečet elektroměru.

The screenshot shows the 'Odečet elektroměrů' application window. The 'Přehled' tab is active, showing the following data:

Výrobní číslo		Čas odečtu	
02477552			
Identifikace		Chyba	
ZPA4ZE310.iv30_007		000000	
Komunikační adresa		Aktuální tarif	
477552		1	

	T1	T2	ΣTx
Dodávka + odběr	0000051.3	0000000.5	0000052.9 kWh
Dodávka	0000000.0	0000000.0	0000000.0 kWh

	L1	L2	L3
Odběr	0000050.9	0000000.9	0000000.9 kWh
Dodávka	0000000.0	0000000.0	0000000.0 kWh
Ef. napětí	242.2	000.3	000.2 V
Okm. účinník	0.425	0.000	0.000 cos

Obrázek D.1: GUI aplikace: přehled odečtu

Průběh odečtu se zobrazuje ve stavovém řádku. Po ukončení odečtu se zde zobrazí čas, jak dlouho trval odečet. Data odečtu je možno uložit do souboru. Přednastavený název souboru je jako u konzolové aplikace. To znamená *em_aaa.txt*, kde *aaa* je komunikační adresa.

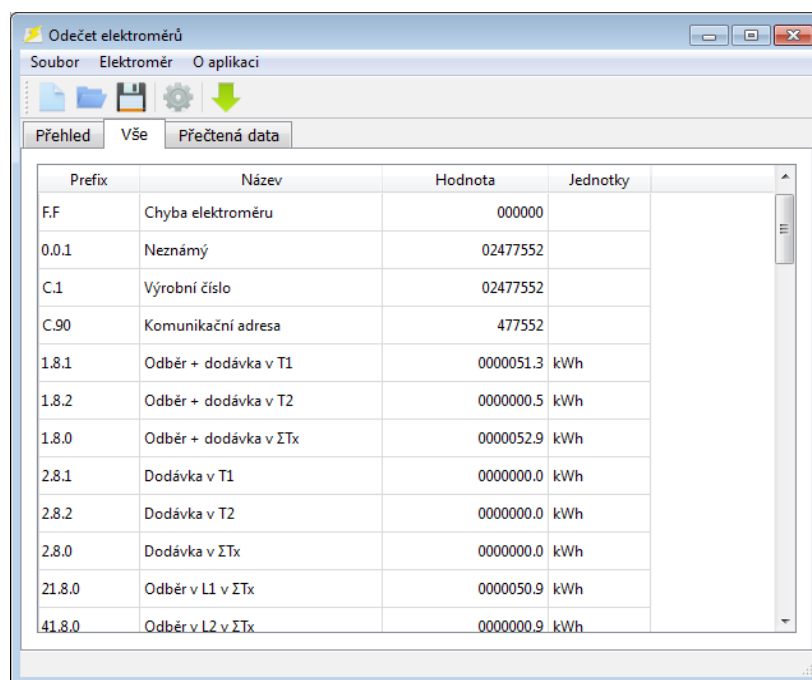
The screenshot shows the 'Nastavení' (Settings) dialog box. It contains the following settings:

- Nastavení portu: Číslo portu (COM3), Přenosová rychlost (4800), Adresa zařízení (empty field).
- Buttons: OK, Zrušit.

Obrázek D.2: GUI aplikace: nastavení

Před prvním použitím je nutné provést nastavení aplikace. Volba přenosové rychlosti obsahuje konkrétní hodnoty rychlosti a také položku *maximální*, kdy se automaticky vybere nejvyšší, elektroměrem podporovaná, rychlost.

Na další záložce jsou všechny hodnoty z odečtu s textovým popisem. U položek, kde není popis znám, je ve sloupci *Název* uvedeno *Neznámý*. Hodnoty jsou zobrazovány včetně nevýznamných nul.



The screenshot shows a window titled "Odečet elektroměrů" with a menu bar (Soubor, Elektroměr, O aplikaci) and a toolbar. Below the toolbar are three tabs: "Přehled", "Vše", and "Přečtená data". The "Přečtená data" tab is active, displaying a table with the following data:

Prefix	Název	Hodnota	Jednotky
F.F	Chyba elektroměru	000000	
0.0.1	Neznámý	02477552	
C.1	Výrobní číslo	02477552	
C.90	Komunikační adresa	477552	
1.8.1	Odběr + dodávka v T1	0000051.3	kWh
1.8.2	Odběr + dodávka v T2	0000000.5	kWh
1.8.0	Odběr + dodávka v ΣTx	0000052.9	kWh
2.8.1	Dodávka v T1	0000000.0	kWh
2.8.2	Dodávka v T2	0000000.0	kWh
2.8.0	Dodávka v ΣTx	0000000.0	kWh
21.8.0	Odběr v L1 v ΣTx	0000050.9	kWh
41.8.0	Odběr v L2 v ΣTx	0000000.9	kWh

Obrázek D.3: GUI aplikace: odečet v tabulce

Poslední záložka s názvem přečtená data obsahuje kompletní data z odečtu. Tak jak jsou uvedeny v příloze C.