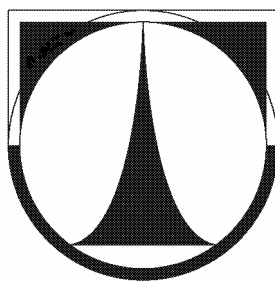


TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA TEXTILNÍ

FAKULTA MECHATRONIKY A
MEZIOBOROVÝCH INŽENÝRSKÝCH STUDIÍ



DIPLOMOVÁ PRÁCE

LIBEREC 2006

LUKÁŠ KUBÍK

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta textilní

Fakulta mechatroniky a mezioborových inženýrských
studií

OBOR: 3106T005

ODĚVNÍ TECHNOLOGIE

Zaměření: **Řízení technologických procesů**

Katedra oděvnictví

Katedra elektrotechniky a elektrických systémů

Vizualizace chodu řízeného pohonu s měničem Motion Control firmy Siemens

Visualisation of the function of the kontrol drive based on the system
SIEMENS-MOTION CONTROL

Vedoucí: Doc. ing. Pavel Rydlo Ph.D.

ROZSAH: Počet stránek: 52
Počet obrázků: 16
Počet tabulek: 7
Počet příloh: 4

P r o h l á š e n í

Prohlašuji, že předložená *diplomová* práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb. O právu autorském a o právech souvisejících s právem autorským).

Souhlasím s umístěním *diplomové* práce v Univerzitní knihovně TUL.

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

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Beru na vědomí, že si svou diplomovou práci mohu vyzvednout v Univerzitní knihovně TUL po uplynutí pěti let po obhajobě.

V Liberci, dne

.....

Lukáš Kubík

P o d ě k o v á n í

Na tomto místě bych chtěl poděkovat vedoucímu diplomové práce *Doc. ing. Pavlu Rydlovi* PhD. za ochotu, pomoc, metodické vedení a cenné rady bez nichž by tato práce nevznikla. Dále bych rád poděkoval konzultantovi *ing. Matrinu Diblíkovi* za ochotu a vstřícnost k mým dotazům.

Anotace

V této práci je stručně shrnuta problematika frekvenčních měničů, komunikace přes sériový port pomocí USS protokolu firmy Siemens. Dále obsahuje popis jednoduchého nástroje, vytvořeného v rámci této DP, pro práci s měniči Motion Control. Umožňujícího získání parametrické sady měniče a jednoduchou práci s tím to měničem.

Annotation

This work briefly summarizes problems of frequency converter, communication by serial port by the help of USS protocol of Siemens Company. It also includes description of simple tool for operating with converters Motion Control created in terms of this diploma work. This tool allows obtaining of parametrical set of converter and simple operation with it.

Obsah

ANOTACE	5
OBSAH.....	6
SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK	9
ÚVOD	10
ZADÁNÍ DIPLOMOVÉ PRÁCE	10
1 ANALÝZA PROBLEMATIKY	11
1.1 FREKVENČNÍ MĚNIČ.....	11
<i>1.1.1 Princip funkce.....</i>	<i>11</i>
<i>1.1.2 Rozdělení</i>	<i>12</i>
Přímé měniče	12
Nepřímé měniče	13
<i>1.1.3 Parametry.....</i>	<i>14</i>
Parametr pro čtení a zápis	14
Parametr pro čtení	15
1.2 KOMUNIKAČNÍ ROZHRAŇÍ	15
<i>1.2.1 Profibus</i>	<i>16</i>
<i>1.2.2 Sériové rozhraní RS 485 / RS 232</i>	<i>17</i>
Elektrické parametry rozhraní	17
<i>1.2.3 USS protokol.....</i>	<i>18</i>
Základní struktura.....	18
Popis základní struktury	19
<i>1.2.4 Netto data</i>	<i>20</i>
PKW oblast.....	20
PZD oblast.....	23
2 ŘEŠENÍ ZADANÉHO PROBLÉMU	25
2.1 STRUKTURA PBE ODPOVĚDI.....	25
Number of array elements	26
Normalization	26
Parameter attribut	26
Access rights.....	26

Text.....	27
Lower limit.....	27
Upper limit	27
ID a ID extension	27
2.2 POUŽITÉ KOMPONENTY.....	32
2.2.1 <i>Komponenty integrované v delphi</i>	32
Formulář (Form).....	32
Tlačítko (Button).....	32
Text (Label).....	32
Tabulka (Table).....	32
Spojení s daty (DataSource).....	32
Mřížka (DBGrid).....	33
2.2.2 <i>Komponenta implementované do delphi</i>	33
Komunikace z portem (AdpComPort)	33
2.3 POPIS PROGRAMU TEST	33
2.3.1 <i>Funkce dostupné v programu</i>	33
2.4 PROGRAM PARAMETR.....	33
2.4.1 <i>Charakteristika programu</i>	33
2.4.2 <i>Funkce dostupné pro uživatele</i>	34
Standardní rozhraní	34
Úplné rozhraní.....	35
2.4.3 <i>Struktura programu</i>	36
Programová část	37
Komunikační knihovny	37
Databáze	37
2.4.4 <i>Filosofie práce s programem</i>	39
3 ZHODNOCENÍ A ZÁVĚR	41
LITERATURA	42
PŘÍLOHY	43
PŘÍLOHA A	43
PŘÍLOHA B	46
Unity.....	46
Vizuální rozhraní.....	47

PŘÍLOHA C	51
<i>PBE</i>	51
ID.....	51
ID extension	51
Normalization.....	52
<i>Request or change text</i>	52
<i>Binektory a konektory</i>	52

Seznam obrázků

OBRÁZEK 1 SCHÉMA TROJFÁZOVÉHO STRÍDAČE	12
OBRÁZEK 2 NEPŘÍMÝ MĚNIČ	13
OBRÁZEK 3 ILUSTRACE PARAMETRŮ	15
OBRÁZEK 4 A) PTP POMOCI USS PROT. B) PROFIBUS	16
OBRÁZEK 5 BLOKOVÉ SCHÉMA SBĚRNICE RS485	18
OBRÁZEK 6 STRUKTURA USS TELEGRAMU	18
OBRÁZEK 7 STRUKTURA ADRESOVÉHO BYTU	19
OBRÁZEK 8 STRUKTURA VLASTNÍHO OBSAHU TELEGRAMU	20
OBRÁZEK 9 STRUKTURA PKW OBLASTI	20
OBRÁZEK 10 STRUKTURA PARAMETRU PKE	21
OBRÁZEK 11 STRUKTURA IND OBLASTI	22
OBRÁZEK 12 STRUKTURA PBE ODPOVĚDI	25
OBRÁZEK 13 SEZNAM BINEKTORŮ Z DRIVE MONITORU	31
OBRÁZEK 14 STRUKTURA FUNGOVÁNÍ PROGRAMU	36
OBRÁZEK 15 PROGRAM PARAMETR PO DOKONČENÍ SKENOVÁNÍ	40
OBRÁZEK 16 STRUKTURA ADRESÁŘE PROJEKTU	40

Seznam tabulek

TABULKA 1 TYP PŘÍKAZU PRO SMĚR MASTER -> SLAVE	21
TABULKA 2 TYP ODPOVĚDI PRO SMĚR SLAVE -> MASTER	21
TABULKA 3 OBSAH PZD1 A PZD2	23
TABULKA 4 STRUKTURA ŘÍDÍCÍHO SLOVA STW (SMĚR MASTER -> SLAVE)	23
TABULKA 5 STRUKTURA STAVOVÉHO SLOVA ZSW (SMĚR SLAVE -> MASTER)	24
TABULKA 6 ID ELEMENT PBE ODPOVĚDI	28
TABULKA 7 ROZDĚLENÍ TYPŮ PARAMETRU PODLE PRVNÍCH 8 BITŮ	29

Úvod

V této práci jsou použity formulace, věty, větná spojení a obrázky z níže uvedených zdrojů.

Zadání diplomové práce

Zadáním diplomové práce je vytvořit program na platformě Windows sloužící k řízení frekvenčních měničů firmy Siemens, komunikující pomocí USS protokolu.

Tento program by měl umět:

1. číst a zapisovat parametry frekvenčního měniče prostřednictvím komponenty ApdComPort od firmy Async Professional
2. získat parametrovou sadu neznámého měniče a umožnit jejich editaci a vizualizaci
3. vytvářet samostatné uživatelské struktury parametrů

1 Analýza problematiky

V této části jsou obecně shrnuty pojmy a principy problematiky frekvenčních měničů. K čemu slouží a jaké je jejich využití. Jejich rozvoj umožnilo:

1. vývoj výkonových prvků
2. vývoj procesorové techniky

V dnešní době se pro pohony, kde je požadavek na řízení otáček, ve stále větší míře používají střídavé elektromotory. Jejich výhodou je nízká cena a malá náročnost na údržbu. Díky jejich rozmanitosti mohou obsáhnout velkou škálu aplikací.

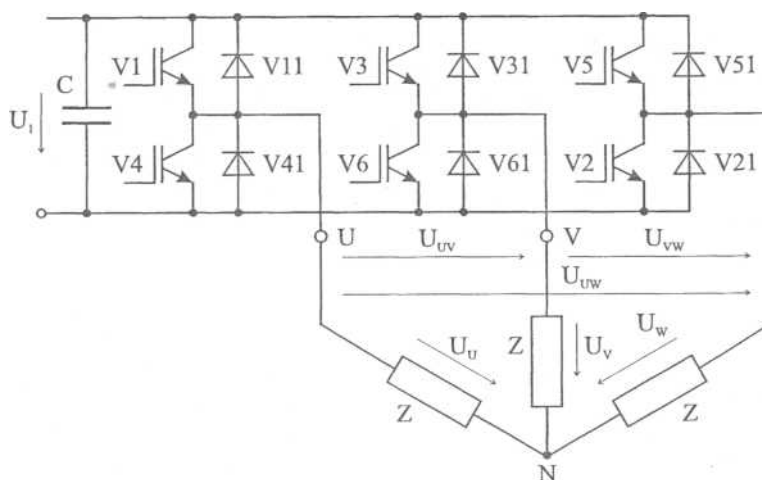
V případě požadavku přesného řízení se v minulosti používaly stejnosměrné motory. Ty mají řadu nevýhod a tou je jejich cena, velká hmotnost a opotřebení kartáčků. Z tohoto důvodu došlo k rozvoji metod a poté i řízení pomocí frekvenčních měničů. Pro aplikace s vysokými požadavky na dynamiku se používají synchronní servomotory.

1.1 Frekvenční měnič

Frekvenční měnič umožňuje měnit a řídit otáčky synchronních a asynchronních motorů pomocí statorové frekvence. Díky tomu je lze použít k řízení jednoduchých i náročných úkolů jako např. balící automaty či stroje v textilním průmyslu.

1.1.1 Princip funkce

Frekvenční měniče mají svůj základ v jednoduchých střídačích, nejjednoduššími zástupci jsou jednofázové střídače. S těmito se nejčastěji setkáme v záložních zdrojích pro nepřetržité napájení (UPS). Pro napájení průmyslových elektrických strojů se používají střídače třífázové.



Obrázek 1 Schéma trojfázového střídače

Přidáním jedné větve s tranzistory a diodami ke schématu jednofázového střídače dostaneme třífázový střídač. Na výstupu střídače je formátována třífázová soustava napětí s proměnným kmitočtem a efektivní hodnotou základního harmonického napětí.

Toto je jejich nejjednodušší základ. K tomuto základu současné měniče přidružili velké množství doprovodných obvodů, které umožňují např. připojit k měniči čidla, vytvářet virtuální spojky a mnoho dalších.

Frekvenčních měničů je na trhu velké množství. Liší se jak svou vnitřní strukturou, rozsahem parametrové sady a mnohými dalšími vlastnostmi. Jen firma Siemens vyrábí několik typových řad měničů, které se dále výkonově liší. Firma Siemens má na trhu v této oblasti pět typových řad měničů MASTERDRIVES, MICROMASTER, SINAMICS, SIMOREG DC MASTER a nyní nově Simatic ET200S FC. Každá řada v sobě zahrnuje několik podskupin, např. u řady MASTERDRIVES to jsou Motion Control a Vector Control.

1.1.2 Rozdělení

Měniče můžeme dělit podle mnoha aspektů a náhledů. Jedním z mnoha přístupů k jejich rozdělení je podle způsobu měnění vstupního napětí.

Přímé měniče

Přímé měniče lze rozdělit do dvou skupin - se sít'ovou a s vlastní komutací.

Síťová komutace

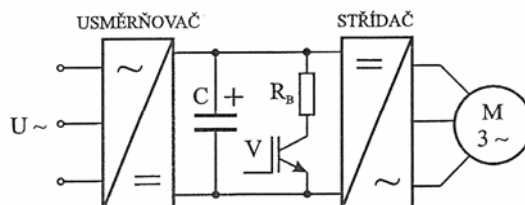
Přímé měniče se síťovou komutací, nazývané cyklokonvertory, jsou osazeny klasickými tyristory a jejich výstupní napětí může dosahovat pouze přibližně třetinového kmitočtu napájecího napětí. V současnosti se používají především pro speciální pohony velkých výkonů.

Vlastní komutace

Měniče s vlastní komutací se nazývají maticové měniče. Maticové měniče jsou osazeny nejčastěji součástkami IGBT. Výstupní napětí je formováno pulzně šířkovou modulací tak, že úseky vstupních napětí jsou připojovány na výstupní fáze a z nich je skládán průběh výstupního napětí s proměnným kmitočtem. Řízení maticových měničů vyžaduje složité řídicí algoritmy, mohou se ale považovat za perspektivní řešení.

Nepřímé měniče

Jedná se o nejrozšířenější zapojení, tvořené vstupním usměrňovačem, stejnosměrným meziobvodem a střídačem. V tomto případě je v meziobvodu zapojen velký vyhlazovací kondenzátor.



Obrázek 2 Nepřímý měnič

Usměrňovač na vstupu měniče je nejčastěji diodový, dříve se používaly i tyristorové usměrňovače, které umožňovaly měnit napětí meziobvodu. U dnešních měničů se změna efektivní hodnoty výstupního napětí realizuje pulzně šířkovou modulací. Moderním řešením je použití vstupního kompatibilního usměrňovače.

V případě, že je na vstupu měniče použit diodový usměrňovač, nebo při napájení ze stejnosměrné sítě, která není schopna přijmout zpět elektrickou energii, je třeba řešit problém generátorického brzdění. Činná energie se předává z asynchronního motoru přes střídač do stejnosměrného meziobvodu. Přes diodový usměrňovač není možno tuto energii dále transportovat. Nabíjení kondenzátoru v meziobvodu touto energií by vedlo k rychlému a

značnému nárůstu jeho napětí. Proto se paralelně ke kondenzátoru připojuje brzdný odpor R_B , který je dalším výkonovým polovodičovým spínacím prvkem připojen v případě, že napětí kondenzátoru překročí určitou mez.

Činná energie generovaná brzdícím asynchronním strojem se potom maří v brzděném odporu R_B viz. obrázek 3.

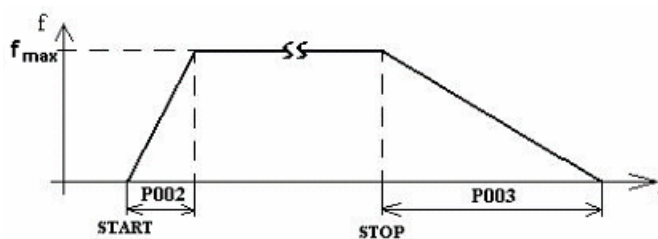
Kromě toho mají měniče také nepříznivý vliv na napájecí síť. Tyto nežádoucí vlivy jsou největší při použití vstupních diodových usměrňovačů a jsou způsobeny nepříznivým průběhem proudu. Proud má v tomto případě tvar úzkých pulzů, kterými je přes usměrňovač dobíjen kondenzátor v meziobvodu. Aby se tyto vlivy měniče minimalizovaly, připojuje se na jeho vstup LC filtr. Jeho nevýhodou však je zvýšení odebíraného jalového výkonu.

1.1.3 Parametry

Slovo parametr je v oblasti měničů primární proměnou. Veškeré vlastnosti a chování měniče se nastavuje pomocí parametrů. Každá firma používá jiné označování parametru. Firma Siemens u měničů MASTERDRIVES Motion Control deklaruje základní dva typy parametrů. Typy určené pro čtení a zápis (P,H,U,L) a jen ke čtení (r,d,n,c). V parametrech mohou být uloženy různé druhy hodnot: hexadecimální, celé či reálné číslo a bitové hodnoty. Zvláštním typem parametru pro čtení a zápis jsou propojovací parametry binektory a konektory.

Parametr pro čtení a zápis

Parametry slouží k modifikaci funkcí, nastavení PI regulátorů a přepínačů. Pomocí tohoto typu parametru je možné číst a zapisovat hodnoty. Např. hodnotou parametru P002 na Obrázku 3 můžeme měnit t a tím určovat strmost nárůstu na maximální výstupní frekvenci. Strmost poklesu určíme parametrem P003. Tento příklad je čistě ilustrativní. U měničů MASTERDRIVES Motion Control se ovšem dostáváme do oblasti velmi rozsáhlých parametrových sad. U této sady se nevyskytují jen jednoduché parametry (P003), ale i parametry složené (P054.1 ... 15). Kompletní popis a rozsah parametrů je vždy uveden v příručce ke konkrétnímu frekvenčnímu měniči.



Obrázek 3 Ilustrace parametrů

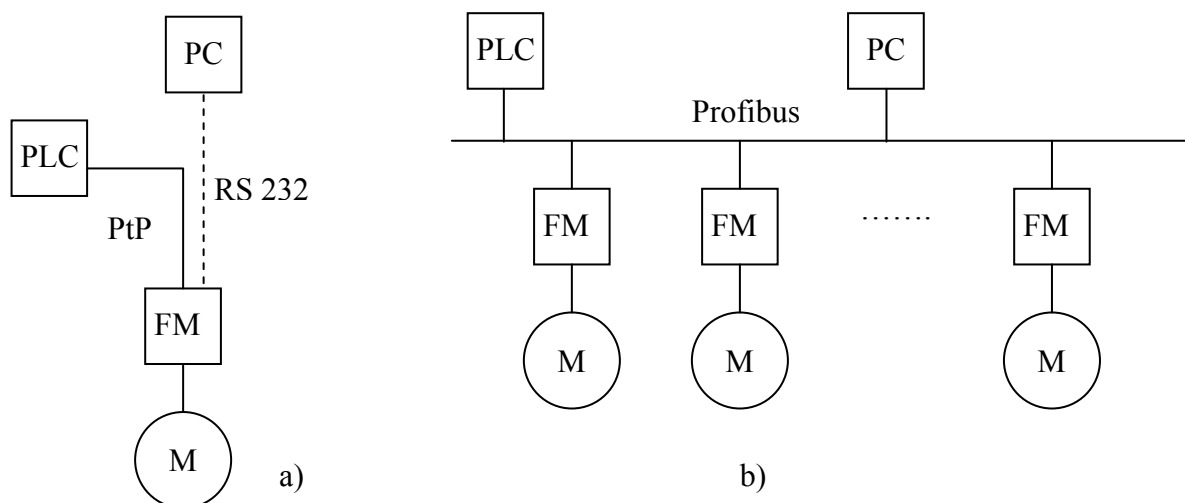
Parametr pro čtení

Tento typ parametrů nelze měnit. Slouží převážně k sledování chování měniče a k němu připojených pohonů. Kompletní popis a rozsah parametrů je uveden v příručce ke konkrétnímu frekvenčnímu měniči.

1.2 Komunikační rozhraní

V průmyslové praxi se k řízení většinou používají PLC (Programmable Logic Controller) pro jejich efektivitu a relativně nízkou cenu. Pomocí PLC může řídit jeden frekvenční měnič, pomocí PtP (Point to Point) za použití USS protokolu, ale také až 32 měničů, pokud ke komunikaci použijeme Profibus. V praxi se také samozřejmě vyskytují i jiné možnosti komunikačních rozhraní např.: průmyslový Ethernet, Profinet, CAN a další možnosti.

Frekvenční měniče firmy Siemens nečastěji komunikují přes hardwarové rozhraní Profibus, RS 485 a RS 232. Pro softwarovou komunikaci se používá USS protokol navržený firmou Siemens.



Obrázek 4 a) PtP pomocí USS prot. b) Profibus

1.2.1 Profibus

Průmyslová sběrnice Profibus je určena pro automatizaci výrobních linek (výroba automobilů, plnicí linky, skladové systémy), pro domovní automatizaci (klimatizace, vytápění), pro procesní automatizaci (chemický a petrochemický průmysl, papírenský, čističky odpadních vod) a pro řízení výroby a distribuce energie. Profibus podporuje čtyři topologie – sběrnice, strom, hvězda a kruh. Preferovanou topologií je sběrnice. Maximální délka sítě Profibus pro RS 485 je závislá na přenosové rychlosti.

Profibus může využívat tři přenosové technologie. Jejich volba je závislá na prostředí, v němž je sběrnice provozována (rušení, nebezpečí výbuchu):

1. RS-485 (high speed – H2) – Profibus DP/FMS

- asynchronní kódování NRZ, přenosová rychlost od 9,6 kb/s do 12Mb/s,
- stíněná kroucená dvojlinka, 32 stanic v segmentu, celkem maximálně 127 stanic,
- pomocí opakováčů lze síť prodloužit do 10 km, připojení 9-pinovým D-Sub konektorem,

2. optické vlákno - Profibus DP/FMS

- maximální délka sběrnice závisí na typu optického vlákna (do 80 km),

- topologie segmentu – kruh nebo hvězda,
- možnost použít převodník mezi RS-485 a optickým vláknem.

3. IEC 1158-2 (Low Speed - HI) - Profibus PA

- synchronní kódování Manchester II s rychlostí 31,25 kb/s,
- volitelná jiskrová bezpečnost a volitelné napájení po sběrnici,
- stíněná nebo nestíněná kroucená dvojlinka
- délka segmentu maximálně 1 900 m (ne pro EEx), síť lze prodloužit pomocí čtyř opakovačů, podporuje topologie sběrnice, strom a nebo jejich kombinace,
- 10 až 32 stanic v segmentu (závisí na třídě EEx a proudové spotřebě), maximálně 127 stanic.

1.2.2 Sériové rozhraní RS 485 / RS 232

Ke každému sériovému komunikačnímu portu PC může být připojeno médium RS 232. Většina dnešních počítačů PC obsahuje standardně dva sériové komunikační porty: COM1 a COM2. Oba jsou vybaveny rozhraním dle normy RS 232 C. Pro připojení média RS 485 je nutné použít vhodný komunikační převodník.

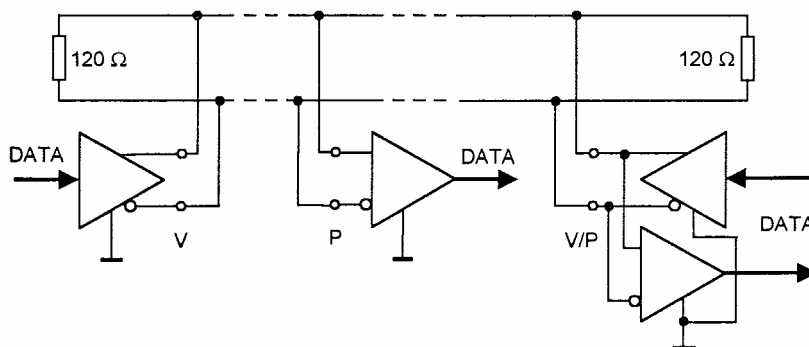
Elektrické parametry rozhraní

Standard RS 232 C

Při sériové komunikaci jsou data vysílána jako posloupnost jednotlivých bitů, přičemž v jednom časovém okamžiku je přenášen vždy jediný bit. Přenášené bity nabývají logických hodnot 0 nebo 1. V případě RS 232 C odpovídá log. 1 napěťové úrovni -3 až -15 V, log. 0 úrovni $+3$ až $+15$ V. Obvody rozhraní jsou nesymetrické, proto se uvedené vztahují k potenciálu nulového signálového vodiče. [5]

Standard RS 485

Sběrnice RS 485 se používá zejména v oblasti průmyslových distribuovaných systémů. Ke sběrnici může být připojen neomezený počet vysílačů s třístavovým výstupem (samozřejmě pouze jeden může být aktivní), počet přijímačů je omezen na 32. Sběrnice je diferenciallyně upořádaná, což jednak umožňuje dosažení přenosové rychlosti až 10 Mb/s.

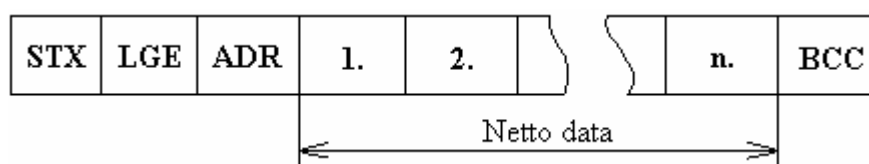


Obrázek 5 Blokové schéma sběrnice RS485

1.2.3 USS protokol

USS protokol (Universelles Serielles Schnittstellen protokoll) je znakový protokol firmy Siemens, který definuje metody sériové komunikace na principu Master-Slave. Každý znakový rámec má start bit, 8 datových bitů, sudou paritu a jeden stop bit. Komunikace mezi zařízeními probíhá prostřednictvím tzv. telegramů o délce 14 Bytů.

Pomocí tohoto protokolu je master (program) schopen komunikovat až z 31 slave (měniče).



Obrázek 6 Struktura USS telegramu

Základní struktura

- STX (Start of Text): 1 Byte, označuje začátek telegramu
- LGE (Telegramlänge): 1 Byte, označuje délku telegramu
- ADR (Adressbyte): 1 Byte, označuje adresu Slave zařízení a typ telegramu

Netto data: obsah zprávy (tuto oblast rozeberu v odstavci 1.2.4)

BCC: 1 Byte, kontrola správnosti komunikace

Popis základní struktury

STX

Byte označující začátek telegramu. Jeho hodnota je vždy 02 Hex.

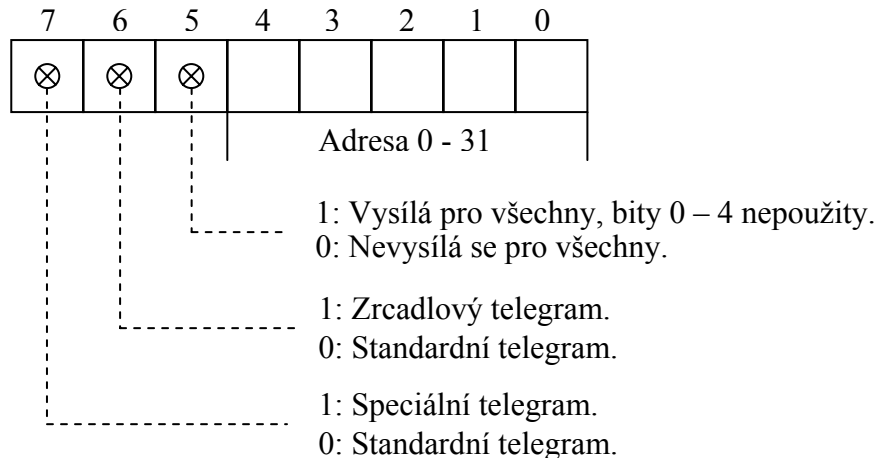
LGE

Byte udávající délku telegramu. Její hodnota je dána jako počet Bytů vlastní Netto zprávy plus adresový Byte plus BCC Byte.

$$LGE = \text{Netto Byte} + \text{ADR} + \text{BCC} \quad (1)$$

ADR

Adresa zařízení, kterému je určen telegram.



Obrázek 7 Struktura adresového bytu

Zrcadlový telegram

Přijatý telegram měnič odešle v neměněné podobě zpět.

Speciální telegram

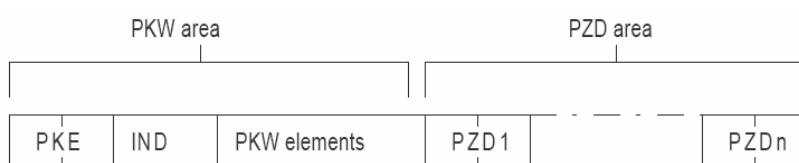
Je určen na specifické úkoly. Podrobně v [4] část A odstavec 5.4

BCC

Kontrolní součet telegramu umožňuje zajistit integritu přijatých potažmo odeslaných dat. Hodnota BCC se vypočítá ze všech předchozích Bytů telegramu pomocí logické funkce XOR.

1.2.4 Netto data

V této části telegramu jsou přenášena samotná uživatelská data. Má dvě hlavní části PKW a PZD.



Obrázek 8 Struktura vlastního obsahu telegramu

PKW oblast

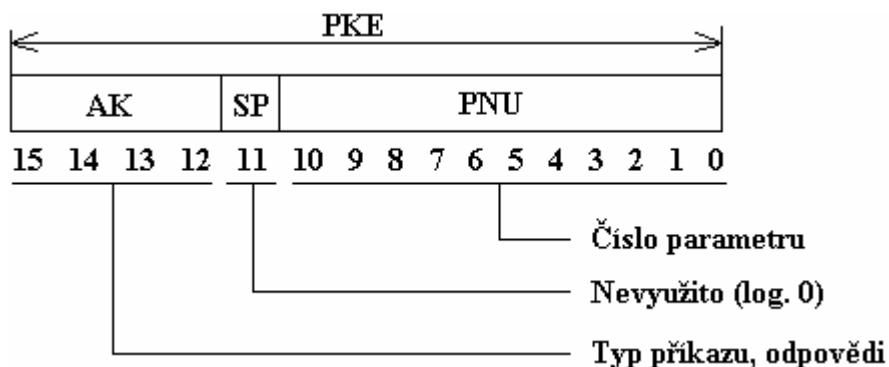
Jak je vidět z obrázku 9 PKW oblast má tři části PKE, IND a PWE1 až PWE_m. Každá z těchto částí má přesně specifikovanou úlohu.



Obrázek 9 Struktura PKW oblasti

PKE

Obsahem této části je přenášet číslo parametru a řídicí slovo, příkaz co má měnič vykonat, respektive odpověď měniče.



Obrázek 10 Struktura parametru PKE

AK – Bity 12 až 15 obsahují typ příkazu nebo odpovědi.

Tabulka 1 Typ příkazu pro směr Master -> Slave

Bit 15 14 13 12	Význam
0 0 0 0	Žádný příkaz
0 0 0 1	Dotaz na hodnotu parametru (čtení)
0 1 0 0	Dotaz na PBE prvek
0 0 1 0	Dotaz na hodnotu parametru (čtení) typu word
1 1 1 0	Dotaz na hodnotu parametru (čtení) typu word do EPROM
1 1 1 1	Dotaz na textový popis parametru (čtení)

Tabulka 2 Typ odpovědi pro směr Slave -> Master

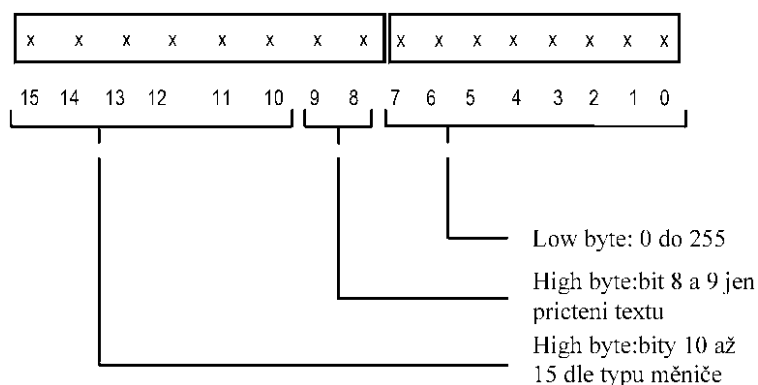
Bit 15 14 13 12	Význam
0 0 0 0	Žádná odpověď
0 0 0 1	Vysílání hodnoty parametru
0 0 1 1	Vysílání PBE prvku
0 0 1 0	Vysílání hodnoty parametru typu word
1 1 1 0	PKW bez hodnoty
1 1 1 1	Vysílání textového popisu

Kompletní tabulky naleznete v [4] část C odstavec 4.2.1.1.

PNU – Bity 0 až 10 obsahují číslo parametru, do kterého zapisujeme novou hodnotu nebo který čteme.

IND

Hodnoty této oblasti jsou závislé na nastavení AK v PKE části.



Obrázek 11 Struktura IND oblasti

Pokud je AK v PKE oblasti nastaveno řídicí slovo jiné než 15 (1 1 1 1) pak Low Byt v intervalu od 1 do 254 udává, který znak nás zajímá, pokud je nastavena hodnota 255, odešlou se z měniče všechny znaky. Pokud je řídicí slovo 15, má hodnota Low bytu podobný význam, ale vstupují zde ještě bity 8 a 9, jejich kombinace dále specifikuje řídicí slovo. Kompletní znění nalezne v [4] část C odstavec 4.2.2., kde jsou též uvedeny tabulky z popisem jednotlivých řídicích slov a jejich interpretací.

PWE

PWE část v oblasti PKW transportuje samotná data. Počet PWE částí je závislý na tom, zda je používána PZD oblast, pokud ano, maximální počet je 108 Wordu, v opačném případě jich může být až 124.

Pokud zapisujeme novou hodnotu příslušného parametru do měniče, je zde zapisovaná hodnota přenášena. Při čtení parametru z měniče je zde v odpovědi hodnota čtená.

Hodnota PWE je přenášena desítkově celočíselně. Celočíselné parametry jsou rovny přímo parametru PWE. Hodnota parametrů s jedním desetinným místem je rovna $PWE * 0,1$. Se dvěma desetinnými místy pak obdobně $PWE * 0,01$.

PZD oblast

Tato oblast je určena k neustálému řízení měniče. PZD oblast telegramu obsahuje minimálně dva PZD prvky.

Tabulka 3 Obsah PZD1 a PZD2

Směr komunikace	PZD1	PZD 2
Master → Slave	Řídicí slovo STW	Hodnota hlavního parametru (f) HSW
Slave → Master	Stavové slovo ZSW	Aktuální hodnota hlavního parametru (f, I) HIW

Tabulka 4 Struktura řídicího slova STW (směr Master -> Slave)

Bit	Hodnota	Význam	Poznámky
0	1 0	Zapnuto Vypnuto 1	Zabezpečení měnič ve stavu „Připraven k běhu“.
1	1 0	Běh Vypnuto 2	Okamžité blokování impulsů, vypnutí pohonu.
2	1 0	Běh Vypnuto 3	Rychlý stop motoru.
3	1 0	Běh povolen Běh blokován	
4	1 0	Rozběh povolen Rozběh blokován	
5	1 0	Rozběh povolen Rozběh pozastaven	
6	1 0	Požadovaná hodnota povolena Požadovaná hodnota blokována	
7	1 0	Potvrzení Bez významu	
8	1 0	Tipování vpravo Bez tipování vpravo	
9	1 0	Tipování vlevo Bez tipování vlevo	
10	1 0	PZD platné PZD neplatné	Data od Master zařízení jsou platná. Data od Master zařízení jsou neplatná.
11	1 0	Běh vpravo Vypnuto 1	Pokud současně bit 0 = 1, vzniká příkaz pro běh.
12	1 0	Běh vlevo Vypnuto 1	Pokud současně bit 0 = 1, vzniká příkaz pro běh.
13			Nevyužito.
14			Nevyužito.

15	1 0	PZD2 = proud PZD2 = frekvence	Ve zpátečním telegramu bude obsah HIW hodnota proudu. Obsah HIW bude hodnota frekvence.
----	--------	----------------------------------	--

Tabulka 5 Struktura stavového slova ZSW (směr Slave -> Master)

Bit	Hodnota	Význam	Poznámky
0	1 0	Připravenost ke startu Vypnuto	
1	1 0	Připravenost k chodu Vypnuto	
2	1 0	Běh povolen Běh blokován	
3	1 0	Porucha Bez poruchy	
4	1 0	Vypnuto 2	
5	1 0	Vypnuto 3	
6	1 0	Rozběh blokován Rozběh neblokován	
7	1 0	Varování Bez varování	
8	1	Nepoužito	
9	1 0	Požadováno řízení Řízení přístrojem	Samostatný chod
10	1 0	f v pořádku f nižší než nastavená	Nižší hodnota frekvence, než je nastavená hodnota.
11	1 0	Běh vpravo Běh vpravo vypnut	
12	1 0	Běh vlevo Běh vlevo vypnut	
13			Nevyužito.
14			Nevyužito.
15			Nevyužito.

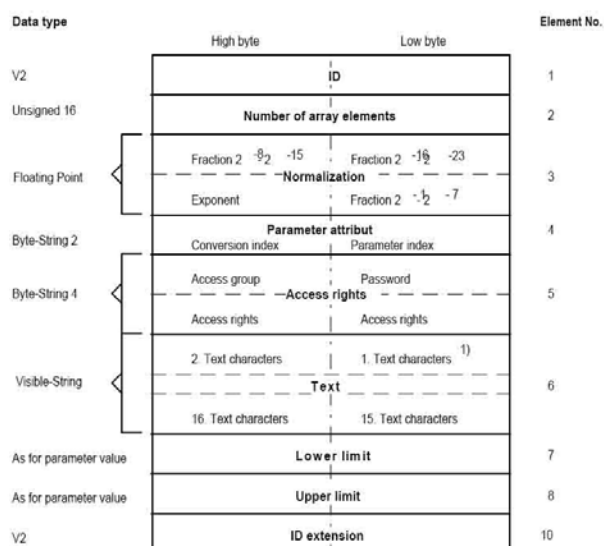
2 Řešení zadaného problému

Mým úkolem bylo vytvořit jednoduchý nástroj pro komunikaci s frekvenčním měničem včetně vizualizace a editace parametrů frekvenčního měniče. Základním stavebním kamenem tohoto programu je komunikace přes sériový port (RS 232 C) pomocí USS protokolu. Od této komunikace se odvíjí následující vizuální zobrazení získaných dat.

V rámci této práce vznikly dva programy. První se nazývá TEST. Jak samotný název napovídá, sloužil k testování jednotlivých algoritmů a komunikace. Je možné do něj nahlédnout na přiloženém CD. Druhý program se jmenuje PARAMETR, tento už slouží jako finální verze. Samozřejmě je možné na něm dále pracovat a rozvíjet jej.

Pro vytvoření a správnou funkci programů TEST a PARAMETR bylo nutné spraveně interpretovat odpovědi z měniče a dekodovat jednotlivé jeho části. Jak se uvedeno v odstavci 1.2.4 Netto data. Na každou otázku má měnič přesně deklarovanou odpověď, jak je naznačeno v tabulkách 1 a 2. Jako nejdůležitější se projevila odpověď na doraz PBE, která v sobě soustřeďuje komplexní informace o jednotlivých parametrech.

2.1 Struktura PBE odpovědi



Obrázek 12 Struktura PBE odpovědi

V této odpovědi jsou uloženy komplexní informace o jednotlivých parametrech. Bez dekódování této odpovědi není možné správně interpretovat získaná data. Je třeba říct, že parametry nabývají různých typů hodnot, reálná čísla, hodnoty v 16 či 2 soustavě a různého druhu Binoktody, Konektory.

Jak je vidět z obrázku č.12, odpověď je strukturovaná do jednotlivých elementů. Na tyto elementy se můžeme ptát jednotlivě podle jejich čísla, a nebo najednou na celek. To nabízí skvělou možnost pro skenování měniče.

Jednotlivé elementy mají různý význam, a to jak obsahově, tak ve svém významu pro práci s měničem.

Number of array elements

V tomto elementu je přenášeno, kolik pod parametrů má parametr (P026.1 ... 40), v případě parametru 26 bude vrácena hodnota 40.

Normalization

Tento element obsahuje hodnotu, která slouží jako indikátor počtu desetinných míst, kterých může hodnota parametru nabývat. V dokumentaci k USS protokolu je uvedeno následující pořadí bytu, ve kterých má hodnota normalizátoru přijít, a to exponent a mantisa. Ve skutečnosti přichází v opačném pořadí, než jak je uvedeno na obrázku č. 12.

Parameter attribut

Tento element se dělí na dvě části *Parameter index* a *Conversion index*, v nich je uložena jednotka, kterou má parametr. Příklad: Parametr r009 je ve °C a v *Parameter index* přijde hodnota 14 a *Conversion index* hodnota + 100. Kompletní výpis je uveden v příloze A. Když má přijít záporná hodnota conversion index, je zaslána jako bytový doplněk. Správnou hodnotu získáme tak, že od přichozích dat odečteme 256, takto nám vyjde záporná hodnota. Příklad: Máli přijít -3, měnič nám pošle hodnotu 253, po odečtení získáme požadovanou -3.

Access rights

Je sice pravda, že tato oblast se dělí na čtyři podoblasti, a to *Access group*, *Password*, *Access rights* a *Access rights*, jenže se v tomto elementu vrací pouze dvě hodnoty. U

parametrů typu P (H, U, L) to je 52 416 v binární soustavě 1100 1100 1100 0000 a u parametrů typu r (d, n, c) to je 34 944 v binární soustavě 1000 1000 1000 0000.

Text

V tomto elementu je přenášen krátký popis parametru. Jazyk, ve kterém měnič vrátí popis, je závislý na nastavení specifického parametru.

Lower limit

Data pro získání hodnoty dolní meze parametru po vynásobení *Normalization* získáme přesnou hodnotu dolní meze. Pokud má být dolní mez záporná, přichází z měniče opět jako doplněk. Zde však vyvstává problém, jak zjistit, že jde o doplněk a nikoliv o skutečnou hodnotu. Dolní meze totiž může nabývat hodnoty word (od 0 do 65 535) u parametru s hodnotou double word (0 až 4 294 967 295). Nelze to bohužel odvodit ze známých hodnot jako u *Conversion index*, kde jsou hodnoty předem známy. Proto záporné dolní meze v programu jsou získány prostým porovnáním příchozí horní a dolní meze. V případě, že hodnota dolní meze je větší než horní, odečte se adekvátní hodnota podle typu hodnoty u wordu 65 536 a u double wordu 4 294 967 296.

Upper limit

Data pro získání hodnoty horní meze, opět je musíme vynásobit *Normalization*.

ID a ID extension

Tyto dvě oblasti PBE odpovědi spolu velmi úzce souvisí, je v nich zakódováno velké množství informací. Bez jejich rozluštění není možné vytvořit program, který by správně interpretoval přijatá data.

V ID elementu měnič vrací konstantní počet hodnot. U těchto hodnot jsem hledal souvislosti a vztahy pro interpretaci dat. Rozložil jsem si příchozí hodnoty na jednotlivé bity a hledal jsem podobné rysy.

Tabulka 6 ID element PBE odpovědi

číslo	ID															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
36	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
515	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1
516	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
518	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0
519	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1
522	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
546	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0
547	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1
548	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0
556	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0
1030	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0
1542	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0
1571	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	1
4100	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
4102	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
4132	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0
5126	0	0	0	1	0	0	1	0	0	0	0	0	0	1	1	0
5155	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	1
12294	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0
16387	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
16388	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
16390	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0
16391	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1
16420	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0
16899	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1
16900	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0
16902	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0
16903	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1
16932	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0
17411	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1
17412	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0
17414	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0
17415	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	1
17444	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0
17923	0	1	0	0	0	1	1	0	0	0	0	0	0	0	1	1
17924	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0
17926	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0
17927	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	1
17955	0	1	0	0	0	1	1	0	0	0	1	0	0	0	1	1
17956	0	1	0	0	0	1	1	0	0	0	1	0	0	1	0	0
20486	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0

20516	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	
21507	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1
21508	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
21510	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0
21540	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0

Prvním úkolem bylo nalézt označení typu L2, V2, I2, O2, I4 a O4. Toto označení jsem přebral z kompendia k měniči od firmy Siemens.

Typy hodnoty parametru:

- L2 Nastavitelný parametr (binektor, konektor)
- V2 Binární parametr
- I2, I4 16, 32 – bitový parametr s možností záporné hodnoty
- O2, O4 16, 32 – bitový parametr s kladnou hodnotou

V první úvaze jsem hledal toto označení v nejnižších 4 bitech, které již na první pohled nabývaly podobných hodnot. Tato první úvaha se však po hlubším prozkoumání stala základem určení typu parametru. Označení totiž je uloženo v nejnižších 8 bitech. Při tomto hledání jsem zjistil, že parametry se stejným ID jsou stejného typu př.: parametry s ID 515 jsou všechny r.

Tabulka 7 Rozdělení typů parametru podle prvních 8 bitů

Hodnota 7 - 0 bitu ID						
L2	V2	I2	I4	O2	O4	???
36	547	3	4	6	7	522
548	1571	515	516	518	519	546
4132	5155	16387	4100	1030	16391	556
16420	17955	16899	16388	1542	16903	
16932		17411	16900	4102	17415	
17444		17923	17412	5126	17927	
17956		21507	17924	16390		
20516			21508	16902		
21540				17414		
				17926		
				12294		
				20486		
				21510		

V tabulce 7 je sloupec označený ???, u těchto parametrů není zcela jasné, o jaké parametry jde, v kompendiu se vůbec nevyskytují. Z krátkých textových popisů, které má

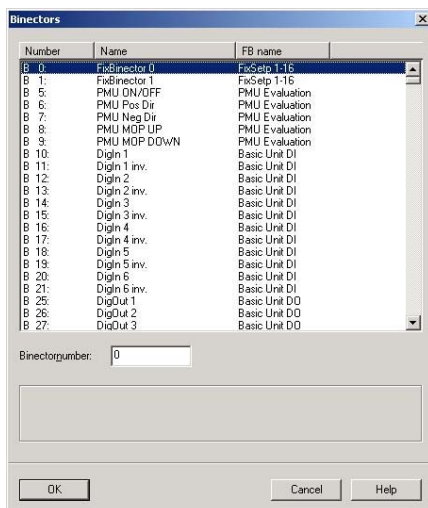
měníč v sobě uložen, se nedalo odvodit, k čemu jsou určeny. Předpoklad je tovární nastavení.

Dalším krokem bylo nalezení identifikace, že jde o pole parametru (P 026.001 ... 040). Jako shodný bit se ukázal bit číslo 14, který byl společný všem polím. Pokud byl nastaven na hodnotu 1, jednalo se o pole.

Následně jsem se zaměřil na popis parametru a hodnot parametru. Podíváme-li se do Drive Monitoru zjistíme, že některé parametry nenabývají jen numerických hodnot, ale jsou interpretovány různými textovými doprovody. Ty jsou uloženy v odpovědi a dotaz číslo 15 (*Request or change text*). Samozřejmě to, že má parametr textový popis, je zakódováno v ID. Hledal sem tedy společný bit pro tyto parametry. Tím se ukázal bit číslo 10. Specifikuje, zda má parametr uložen v měniči text. Způsob dotazování na textový zápis popíšu v další podkapitole.

Parametry, které mají v 10. bitu ID hodnotu 1, mají textový popis. Tento bit nemá ovšem vždy stejný význam u parametrů, které jsou typu pole (P221.001 ... 004) texty popisující jednotlivé podparametry, oproti tomu u nerozšířených parametrů (r001) popisuje jednotlivé hodnoty parametru.

Jedny z textově popsaných parametrů, v Drive Monitoru, jsou konektory a binektory viz. obrázek 13. Tyto popisy ovšem nejsou uloženy ve frekvenčním měniči. Označení zda jde o konektor nebo binektor je uloženo v ID extension. Jde-li o binektor, je ve 3 bitu 1, u konektoru je ve 4 bitu 1. U těchto parametrů jsem narazil na zvláštní případ. Hodnoty, které lze zapsat a přečíst do a z měniče, si vzájemně neodpovídají s označením binektorů a konektorů v Drive Monitoru. Jedinou spojitost jsem našel v jejich pořadí. Příklad: u binektoru P 030.001 lze zapsat hodnoty 0, 1, 5, 6, 7, 8, 9, 16, 17, 18, 19, 20, 21, 22, 24, 25, 32, 33, 37, 38, 39, ... atd., jde však o to, že hodnota 21 odpovídá binektoru 15 a hodnota 37 binektoru 25. Jak jsem již uvedl, jediná spojitost mezi označením a hodnotami je pořadí. Když se podíváme na výše uvedenou řadu hodnot, nevyskytuje se v ní ani opakující se cyklus hodnot, ze kterých by se dal odvodit nějaký závěr.



Obrázek 13 Seznam binektorů z Drive monitoru

Další informace uložené v ID extension jsou v:

- 1 bitu parametr typu pole má textově popsány i hodnoty
- 8 bitu parametr je označen jako BDS (BICO data set)
- 9 bitu parametr je označen jako FDS (function data set)
- 10 bitu označuje, zda je o duální konektor, do kterého jdou připojit 2 binektory

Toto jsou všechny informace, které se mi podařilo získat z PBE odpovědi.

Odpovědi na Request or change text

V této odpovědi vrací měnič textový popis parametru, podparametru, hodnoty nebo obě tyto informace. U parametrů, které mají mít textový doplněk hodnoty parametru, se musíme dotazovat na jednotlivé hodnoty podle rozsahu parametru, textové popisy jsou uloženy za sebou počínaje číslem 1, nikoliv nejnižší hodnotou parametru. Parametry typu pole, které mají textový popis jednotlivých podparametrů, mají tyto texty uloženy stejným způsobem jako u popisu hodnot. V případě kombinace obojího popisu je potřeba při získávání popisu hodnot uložit do 9 bitu IND oblasti telegramu 1. Což znamená provést dvojité dotazování. První na popis podparametrů druhé na popis hodnot.

2.2 Použité komponenty

Pro vytvoření programů TEST a PARAMETR byly použity komponenty implementované v delphi. Dále jsem doinstaloval free komponentu na obsluhu seriového portu. Tuto komponentu vytvořila firma Async Professional. Pro kompilaci, rekompilaci, odeslání a přijetí telegramu jsem použil dynamickou knihovnu, kterou zkonstruoval a naprogramoval ing. Martin Diblík, člen katedry elektrotechniky a elektrotechnických systémů na TU Liberec.

2.2.1 Komponenty integrované v delphi

Stručný popis některých použitých komponent.

Formulář (Form)

Základní komponentou (objektem) je samotný formulář (okno), ten má sám osobě mnoho vlastností a událostí. Z nich nejdůležitější pro program byla událost OnCreate kde byli a provedeny operace potřebné před zahájením samotné práce s programem.

Tlačítko (Button)

Tlačítko je nejběžnější komponenta ve všech aplikacích. Její nejpodstatnější událostí je OnClick. Během této události se ošetřují hlavní funkce tlačítka.

Text (Label)

Tato komponenta slouží k vypsání textové informace uživateli.

Tabulka (Table)

Komponenta Table slouží k zapouzdření databáze pro práci v delphi, do které budou ukládány data. O jednotlivých měničích.

Spojení s daty (DataSource)

Data Source propojuje zapouzdřenou databázi s delphi a tím umožňují s touto databází pracovat.

Mřížka (DBGrid)

DBGrid je specializovaný typ mřížky pro zobrazování dat uložených v databázi. Takto komponenta nevlastní sama žádná data jako je tomu u string gridu. Dala by se přirovnat k oknu do databáze.

2.2.2 Komponenta implementované do delphi

Komunikace z portem (AdpComPort)

Tato komponenta je specializovaná na obsluhování sériového portu, podle standardu RS 252 C, ale taky RS 485. U této komponenty bylo pro účel programů nutné správně ošetřit proceduru TriggerAvail, která se vyvolá při příchodu dat na port.

2.3 Popis programu TEST

Program Test sloužil jako testovací nástroj pro komunikaci přes sériový port a testování základní struktury programu Parametr. Prováděly se na něm i testy jednotlivých algoritmů před zahájením práce na programu Parametr.

2.3.1 Funkce dostupné v programu

Program umožňuje komunikaci přes sériový port. Skenování měniče a ukládání získaných dat do pracovní databáze. Čtení a zápis hodnot parametrů jen do textového pole. Ruční vytváření telegramu.

Program Test je stručně popsán v příloze B. Do jeho zdrojových kódů je možno nahlédnout na přiloženém CD ve složce *Program_Test*.

2.4 Program PARAMETR

2.4.1 Charakteristika programu

Program Parametr je určen pro práci s měniči firmy Siemens komunikujících pomocí USS protokolu přes komunikační rozhraní RS 232 nebo RS 485. Pokud je měnič vybaven rozhraním RS 485, je potřeba použít externí hardwarový převodník s RS 232 na RS 485.

Program slouží k jednoduchému řízení frekvenčních měničů. Umožňuje získat parametrou sadu měniče, číst a zapisovat hodnoty do parametru. Číst a zapisovat může do RAM i do EEPROM paměti. Pokud je program nastaven v off-line režimu, pracuje pouze s daty uloženými v databázi. Po připojení k měniči se data v databázi automaticky přepíše podle aktuálního stavu v měniči. Z toho vyplývá, že pokud chce uživatel provádět změny, musí být připojen k měniči, a to buď k RAM nebo EEPROM paměti.

RAM a EEPROM paměť měniče

Paměti RAM a EEPROM jsou pro měnič zdrojem hodnot parametrů. Paměť EEPROM je primárním zdrojem hodnot parametru pro měnič. Zde jsou trvale uloženy hodnoty parametru i pro vypnutí měniče. Do paměti RAM se po startu měniče nahraje kompletní parametrická sada z paměti EEPROM. S takto nahranou sadou měnič pak pracuje. Z toho vyplývá, že dokud uživatel nezmění hodnotu parametru v EEPROM paměti, nedochází k trvalé změně hodnoty parametru.

2.4.2 Funkce dostupné pro uživatele

Program je koncipován na bázi projektu. Uživatel si vytváří pro jednotlivé měniče samostatné projekty. Dále má k dispozici dvě uživatelská rozhraní *Standardní* a *Úplné*, pod kterými jsou seskupeny jednotlivé funkce. Jejich komplexní popis je uveden v uživatelském manuálu.

Standardní rozhraní

Standardní rozhraní umožňuje běžnou práci s měničem. K tomu slouží následující funkce.

Vytvoření, otevření, editace a uzavření projektu

Program umožňuje práci s projektem jakožto zdrojem základních informací pro celou práci s příslušným měničem.

Skenování měniče

Program získá parametrickou sadu měniče.

Změna hodnoty parametru

Umožňuje měnit hodnotu parametru podle režimu komunikace.

Režim komunikace

Program má tři režimy komunikace Off-line, On-line RAM nebo EEPROM. Ke změně hodnoty parametru v měniči dochází jen v posledních dvou případech.

Vytvoření výběru parametrů

Uživatel může vytvářet vlastní seskupení parametrů.

Protokol projektu

Umožňuje vytvořit txt dokument o projektu (měniči) podle aktuálně zvoleného výběru parametrů.

Cyklické čtení hodnot

Program cyklicky čte hodnoty parametrů které jsou aktuálně zobrazeny.

Úplné rozhraní

Úplné rozhraní ponechává všechny výše zmíněné funkce v platnosti a přidává k nim další. Ty jsou však určeny spíše ke komplexnímu přístupu k problematice měničů firmy Siemens a vyžadují již znalost USS protokolu.

Otevření a Uzavření komunikačního portu

Možnost pracovat s portem.

Vytváření telegramu

Uživatel může tvořit jednotlivé části telegramu ručně.

Odeslání a Přijetí telegramu

Možnost postupného zpracování telegramu.

Ruční skenování

Umožňuje provést ve třech krocích skenování měniče. Jde o alternativní postup výše zmíněné funkce.

Výpis činností programu

Uživatel může sledovat v textovém poli, co program vykonává. Jaké informace se budou zobrazovat, uživatel nastaví v *Nastavení Zobrazování* viz. [6,odstavec 4.4.2.].

Mazání textového pole

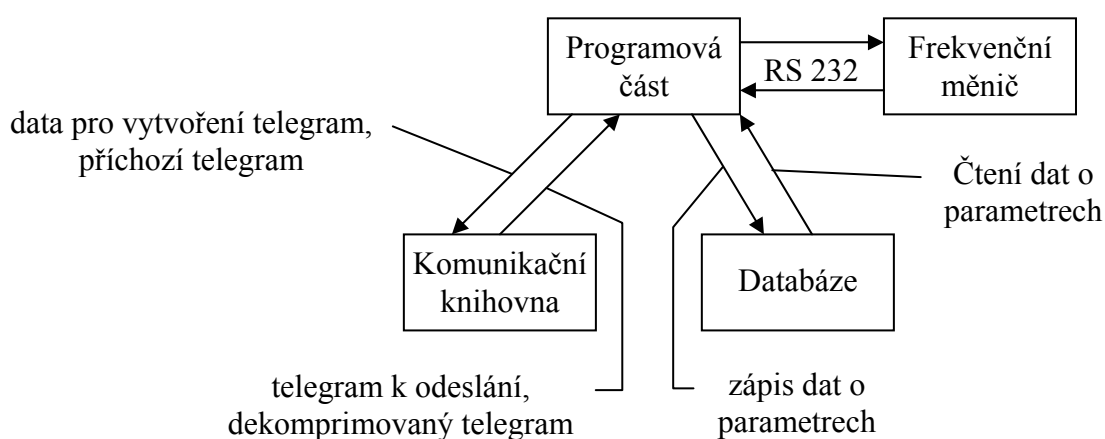
Uživatel může smazat textové pole.

Mazání databáze

Uživatel může smazat celou databázi.

2.4.3 Struktura programu

Program se skládá ze tří základních částí. Programová část, komunikační knihovny a databáze. Z těchto tří částí programová část zastřešuje veškerý chod programu a komunikaci s měničem. Komunikační knihovna vytváří ze vstupních dat telegram, který vrátí programu a ten jej odešle do měniče. Databáze pak slouží jako úložiště dat o parametrové sadě měniče.



Obrázek 14 Struktura fungování programu

Programová část

Programová část se skládá ze čtrnácti unit. Jednotlivé unity představují buď formuláře, nebo jsou v nich soustředěny logické celky programu. Např. unit Pom_pro obsahuje všechny procedury, které přímo neošetřují události komponent uložených ve formuláři.

Komunikační knihovny

Komunikační knihovny vytváří ze vstupních dat telegram k odeslání a naopak přijatý telegram rozloží na samostatná data. Dále umožňuje práci s komunikačním portem pomocí komponenty AdpComPort integrované do programové části.

Databáze

Databáze slouží jako úložiště dat a zdroj informací pro program o jednotlivých parametrech. Databáze má celkem 18 sloupců, a to řádek, číslo, parametr, popis, index, text, hodnota, rozměr, horní mez, dolní mez, typ, datový typ, zápor, ID, ID extension, maxindex, volný 1.

Řádek

Řádek slouží jako primární identifikační klíč při orientaci v databázi. Je to integerovská hodnota odvozená od řádku databáze a počtu podparametrů jednotlivých parametrů parametrické sady měniče.

Číslo

Číslo je reálné označení parametru, se kterým pracuje měnič. Např. parametr U001 má číslo 2001. V databázi je ukládána jako reálné číslo.

Parametr

Je pravým opakem sloupce číslo. Je v něm uloženo právě uživatelské označení parametru (U001). Z toho vyplývá, že tento sloupec je typu string, a to v max. délce 4 znaků.

Popis

Zde je uložen krátký textový popis, který je získán z PBE odpovědí viz. odstavec 2.1.

Jde o string v délce 50 znaků, tato hodnota byla empiricky odvozena.

Index

V tomto sloupci je uložena aktuální hodnota podparametrů. Jde o integrovskou hodnotu.

Text

Pokud má parametr nějaký další textový popis, je uložen v tomto sloupci, jako stringová hodnota o délce 100 znaků, tato hodnota byla empiricky odvozena. Tento text je získaný z odpovědi na dotaz 15 viz. tabulka 1 a 2.

Hodnota

V tomto sloupci je uložena zobrazovací hodnota parametru. Tento sloupec je typu string a to v délce 20 znaků, tato hodnota byla opět empiricky odvozena. Takto je ukládán z důvodu toho, že mají být některé parametry zobrazeny jako binární hodnota.

Rozměr

Obsahuje rozměr parametru. Jde o stringovou hodnotu o délce 10 znaku, opět empiricky odvozená hodnota. Všechny rozměry, kterých mohou parametry nabývat, jsou v příloze A.

Horní a Dolní mez

Tyto dva sloupce mají jasný význam, jsou v nich uloženy horní a dolní meze parametru. Pro jednoduchost práce jsou tyto dva sloupce uloženy jako string v délce 20 znaků.

Normalizátor

Udává kolik desetinných míst může mít hodnota parametru. Je to reálná hodnota.

Typ

Zde je uložena hodnota Access rights viz. odstavec 2.1. Je ukládána jako integer.

Datový Typ

Zde je uloženo označení, o jaký datový typ parametru jde, L2, O4 atd. ukládá se

hodnota spodních 8 bitů ID. Jde tedy o integer.

Zápor

Jde o identifikátor, zda může být parametr i záporný. Ukládá se jako integer, ale nabývá vždy hodnot 0 nebo 1.

ID a ID extension

Zde se ukládá celá hodnota ID a ID extension elementu z PBE odpovědi, viz. odstavec 3.1. Jde o hodnoty typu integer.

MaxIndex

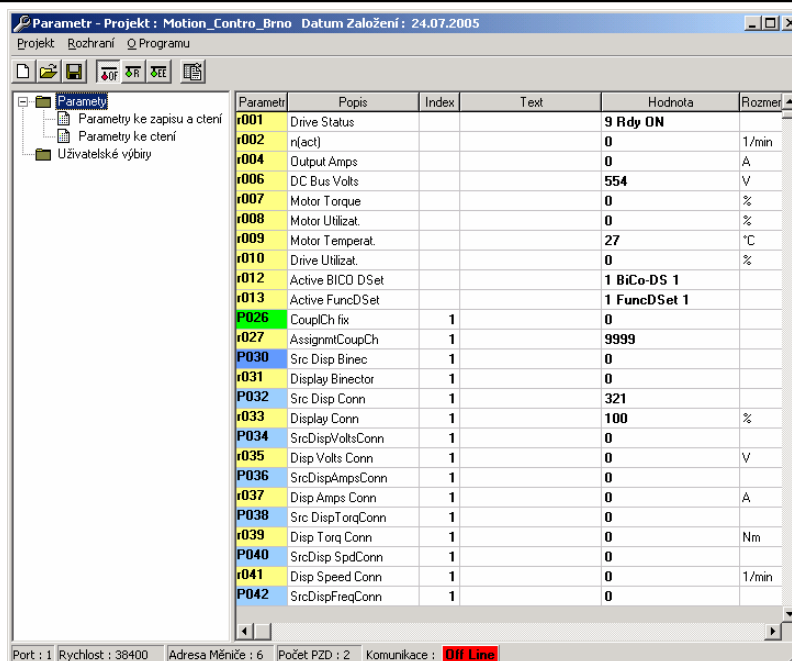
Jde o pomocnou hodnotu pro práci s podparametry. Jak název napovídá, je zde uložen maximální počet podparametru. Tedy integerovská hodnota.

Volný 1

Jde o sloupec, se kterým se nepracuje, je integerovského typu a má sloužit pro případný další vývoj programu.

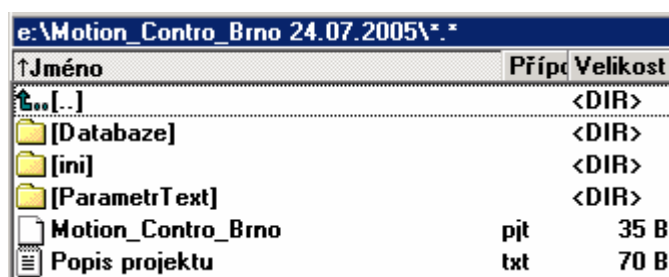
2.4.4 Filosofie práce s programem

Program je koncipován na bázi projektů. Před zahájením práce s měničem je tedy nutné vytvořit projekt. Každý projekt v sobě zahrnuje samostatnou databázi, popis parametrů a projektu. Dalším krokem je získání parametrické sady měniče, kterou se naplní nově vzniklá databáze projektu. Po dokončení skenování můžeme začít pracovat s měničem. Jednotlivé parametry můžeme měnit jednoduchým double-klíkem na hodnotu parametru, tím se vyvolá dialogové okno k příslušnému parametru viz. [6] odstavec 5.3.



obrázek 15 Program PARAMETR po dokončení skenování

V rámci projektu vznikne samostatná složka projektu. V ní se shromažďují komplexní informace o projektu. Tato složka je přenositelná do příští či jiné instalace programu. Na obrázku 16 je vidět vnitřní struktura adresáře.



Obrázek 16 Struktura adresáře projektu

Jak je vidět z obrázku 16, jsou v adresáři projektu tři složky a dva soubory. V složce *Databáze* jsou uloženy soubory databáze. Složka *ini* skrývá ini soubory o projektu, ve kterých je uloženo nastavení programu a projektu, a *Parametr Text* skrývá soubory o textových popisech jednotlivých parametru měniče. Soubor *Popis projektu.txt* se vytvoří při založení projektu a je v něm uložen stručný popis projektu, který je možné v průběhu práce měnit. Zbývá poslední soubor *Motion_Control_Brno.pjt*, tento soubor slouží jako inicializační pro otvírání celého projektu.

Před zahájením práce s programem je doporučuji prostudovat si *uživatelský manuál*, který je na přílohovém CD a po nainstalování programu ho naleznete ve složce programu Parametr.

3 Zhodnocení a závěr

Cílem této diplomové práce bylo proniknutí do jednotlivých struktur komunikace frekvenčních měničů firmy Siemens a jejího USS protokolu. Jedním z hlavních cílů bylo zanalyzovat PBE odpověď. Dalším úkolem bylo vytvořit jednoduchý nástroj pro práci s měniči firmy Siemens, komunikujících pomocí USS protokolu.

Pro vytvoření uživatelského software se ukázalo zanalyzování PBE odpovědi jako absolutně nezbytné. To se bohužel zcela nepodařilo, protože se nepodařilo od firmy Siemens získat podrobnější informace týkající se parametrů přenášených USS protokolem. Bylo tedy nutné provést empirické zkoumání jednotlivých elementů PBE odpovědi. Většina elementů, až na ID a ID extension, byla snadno analyzovatelná, neboť jejich názvy mnohokrát napovídaly o jejich obsahu. U ID a ID extension však nebylo možné dovodit co je jejich obsahem. Po získání soupisu hodnot ID a ID extension bylo možné porovnáním získaných a známých informací z kompendia, získat alespoň částečný obraz o informacích uložených v ID a ID extension. Tento částečný obraz o informacích uložených v ID a ID extension byl dostačující pro vytvoření jednoduchého softwaru.

Přínosem práce je vytvoření programů TEST a PARAMETR. Pomocí programu TEST bylo prováděno zkoumání možností komunikace a informací uložených v PBE odpovědi. Stručný přehled získaných informací je uveden v příloze C. Pomocí programu PARAMETR je možné pracovat s měniči firmy Siemens, komunikujícími pomocí USS protokolu. Primárně je program určen pro měniče Motion Control.

V příloze Diplomové práce je instalační CD s programem PARAMETR a Uživatelským manuálem k tomuto programu. Dále na CD naleznete Zdrojové kódy obou programů a elektronickou verzi diplomové práce a uživatelský manuál ve formátu pdf.

Literatura

- [1] Pavelka, Jiří; Čerovský, Zdeněk, Javůrek, Jiří, *Elektrické pohony*. 2. vyd. Praha : ČVUT, 2001, ISBN 80-01-02314-1
- [2] Binzinger, Thomas *Naučte se programovat v Delphi : Podrobný průvodce začínajícího uživatele.-1. vyd.-*Praha : Grada Publishing, 1998.-342 s. ISBN 80-7169-685-4
- [3] Jarůrch, Jiří *Regulace moderních elektrických pohonů*.GRADA 2003 ISBN 80-247-0507-9
- [4] SIEMENS AG: Universal Serial Interface Protocol - USS Protocol,
- [5] Haasz, Vladimír; Roztočil, Jaroslav; Novák Jiří.;*Číslicové měřicí systémy*.Praha: ČVUT, 2000. 315s.ISBN 80-01-02219-6
- [6] Kubík, Lukáš ;*Uživatelský manuál k programu PARAMETR*. Příloha diplomové práce

Přílohy

Přílohou bez označení je uživatelský manuál k programu PARAMETR, který je vložen v zadní části DP ve formátu A5. Dále jej naleznete na příloženém CD. Do zdrojových kódů programů můžete nahlédnout na příloženém CD.

Příloha A

@=""	"4-6"="μs"
"0+0"=""	"5+0"="N"
"1+0"="m"	"5+3"="kN"
"1-3"="mm"	"5+6"="MN"
"1-6"="μm"	"6+0"="Pa"
"1+3"="km"	"6+3"="kPa"
"2+0"="m"	"6+2"="mbar"
"2-6"="mm"	"6+5"="bar"
"2+6"="km"	"7+0"="kg"
"3+0"="ml"	"7-3"="g"
"3-3"="l"	"7-6"="mg"
"4+0"="s"	"7+3"="t"
"4+70"="min"	"8+0"="J"
"4+74"="h"	"8+3"="kJ"
"4+77"="d"	"8+6"="MJ"
"4-3"="ms"	

Vizualizace chodu řízeného pohonu s měničem Motion Control firmy Siemens

"8+74"="Wh"	"13+66"="mm/min"
"8+75"="kWh"	"13+67"="m/min"
"8+76"="MWh"	"13+68"="km/min"
"9+0"="W"	"13+71"="mm/h"
"9+3"="kW"	"13+72"="m/h"
"9+6"="MW"	"13+73"="km/h"
"9-3"="mW"	"14+0"="ml/s"
"10+0"="VA"	"14+67"="ml/min"
"10+3"="kVA"	"14+72"="ml/h"
"10+6"="MVA"	"14-3"="l/s"
"10-3"="mVA"	"14+66"="l/min"
"11+0"="1/s"	"14+71"="l/h"
"11+67"="1/min"	"15+0"="kg/s"
"11+72"="1/h"	"15-3"="g/s"
"12+0"="rad"	"15+3"="t/s"
"12+79"="''"	"15+66"="g/min"
"12+78"="''"	"15+67"="kg/min"
"12+80"="°"	"15+68"="t/min"
"12+81"=""	"15+71"="g/h"
"13+0"="m/s"	"15+72"="kg/h"
"13-3"="mm/s"	"15+73"="t/h"

"16+0"="Nm"	"23+0"="Ohm"
"16+3"="kNm"	"23-3"="mOhm"
"16+6"="Mnm"	"23+3"="kOhm"
"17+0"="K"	"23+6"="MOhm"
"17+100"="°C"	"24+0"="%"
"17+101"="°F"	"25+0"="%"
"18+0"="K"	"26-3"="g/kg"
"19+0"="J/K*kg"	"27+0"="%"
"19+3"="kJ/K*kg"	"28+0"="Hz"
"19+6"="MJ/K*kg"	"28+3"="kHz"
"20+0"="J/kg"	"28+6"="MHz"
"20+3"="kJ/kg"	"28+9"="Ghz"
"20+6"="MJ/kg"	"29+0"="Nm/A"
"21+0"="V"	"30+0"="h"
"21+3"="kV"	
"21-3"="mV"	
"21-6"="μV"	
"22+0"="A"	
"22-3"="mA"	
"22+3"="kA"	
"22-6"="μA"	

Příloha B

Úvodem musím zdůraznit, že se jedná o pracovní program a nikoliv o software pro koncového uživatele.

Program komunikuje z frekvenčním měničem firmy Siemens, umí získat kompletní parametrou sadu měniče a editovat jeho hodnoty. Do pracovní databáze ukládá kompletní PBE element a navíc ještě pomocné proměnné k jednotlivým parametrům. Je určen k odzkoušení a odladění algoritmu pro práci z měničem.

Unity

Program je strukturován do tří unity, Pom_Fun_Pro, uss_header a Test_1. Tyto unity mají zpřehlednit kód programu.

Pom_Fun_Pro

V Pom_Fun_Pro jsou uloženy veškeré pomocné funkce a procedury, které mají zajišťovat specifické úkoly.

uss_header

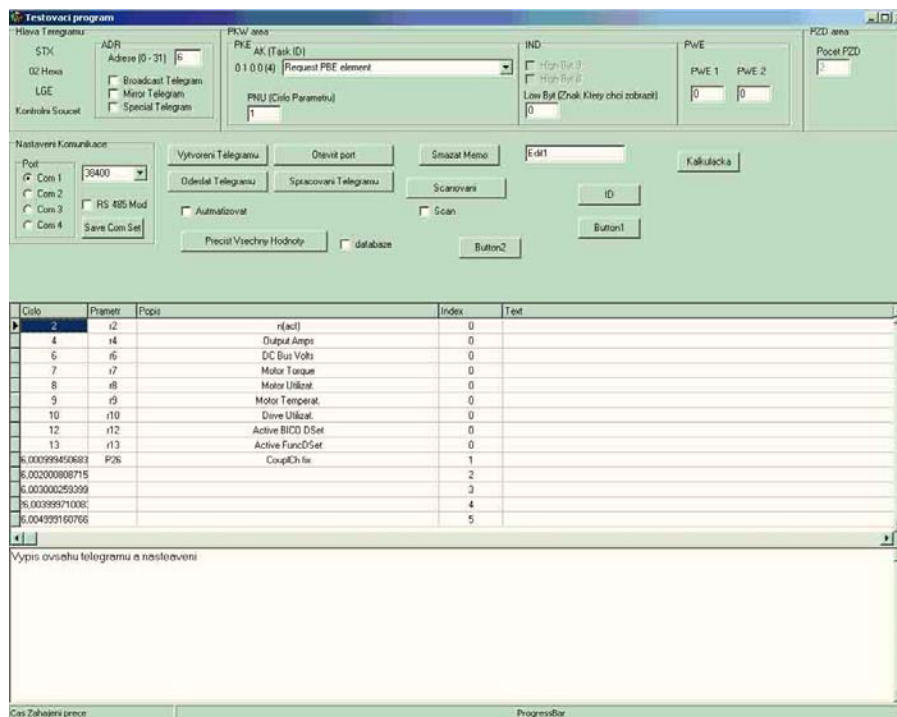
Zde jsou kumulovány deklarace tříd a konstant, umožňuje to jejich snadnou modifikaci.

Test_1

Test_1 je unit samotného formuláře a tudíž obsahu je procedury jednotlivých komponent uložených na tomto formuláři.

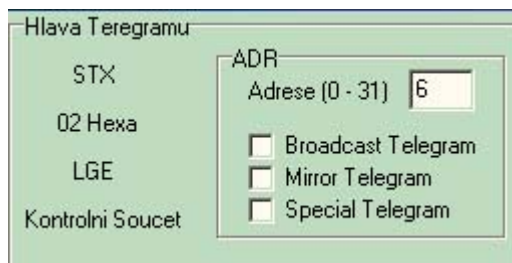
Vizuální rozhraní

Toto rozhraní je koncipováno jako komplexní pohled na telegram a příchozí data, z toho tedy vychází jeho vzhled. Při spuštění programu se uživateli (vývojáři) zobrazí kompletní pohled na databázi a zadávací oblast.



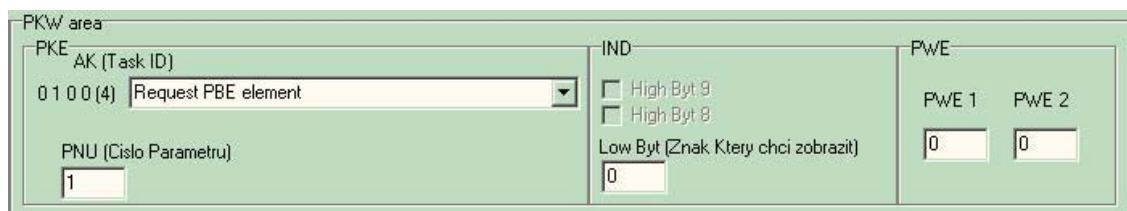
Obrázek 1 Zobrazení programu

Jak je vidět na Obrázku č.1, v horní části okna je vytvořen strukturální pohled na telegram.



Obrázek 2 Hlavička telegramu

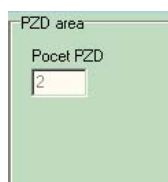
Na Obrázku č.2 je zobrazena hlavička vytvářeného telegramu, můžeme zde zvolit jednak typ telegramu a adresu měniče.



Obrázek 3 PKW oblast

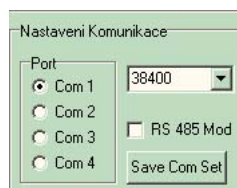
V této části volíme jednotlivé typy dotazů, číslo parametru, hodnotu Low bytu a hodnotu parametru. Pokud chceme zjistit, či editovat hodnotu parametru pole (P026.004), zadáváme do Low bytu číslo podparametru. U dotazu na PBE do něj zadáme číslo elementu, který nás zajímá, nebo můžeme zadat hodnotu 255, v tom případě měnič vrátí celý PBE element. U dotazů na text platí totéž jako u PBE, avšak neplatí-li, že v po zadání 255 vrátí celý textový element. Zde musíme zadávat jednotlivé hodnoty postupně.

Dále je zde prostor pro zadání hodnoty pro zapsání do parametru PWE 1 a 2, místo na zadávání nové hodnoty parametru je závislé na typu parametru, pokud je to word, zadáváme novou hodnotu do PWE 1, v případě doulewordu zadáme hodnotu do PWE 2, pokud je ovšem nová hodnota větší než 65 535, musíme jí ručně rozdělit do obou PWE oblastí. Příklad: Chci zapsat hodnoty 70 000 do PWE 1, zapíšeme 1 a do PWE2 zapíšeme 4 464.



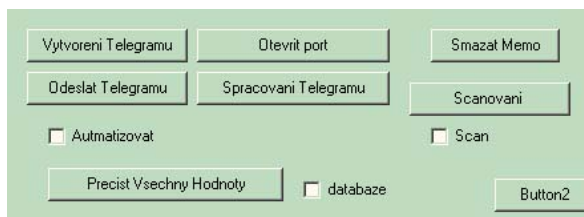
Obrázek 4 Počet PZD

Jak je jasné už na první pohled, zde zadáváme délku PZD oblasti. Tento údaj je velmi důležitý, pokud není nastavena stejná délka, jak v programu, tak v měniči nebude probíhat komunikace.



Obrázek5 Nastavení komunikace

Zde nastavíme samotný základ komunikace-port přes který budeme komunikovat, a rychlost komunikace v baudech. Také můžeme nastavit komunikaci pomocí standardu RS 485.



Obrázek 6 Funkční tlačítka

Na obrázku č. 6 vidíme několik tlačítek, nejdůležitější z nich je *otevřít port*, to zajistí otevření portu a tím umožní samotnou komunikaci. Dalším tlačítkem je *vytvoření telegramu*, jak je zřejmé, vytvoří telegram a jeho obsah vypíše do textové oblasti v dolní části okna. Na něj logicky navazuje *odeslání telegramu*, to odešle telegram na sériový port. Poté, co nám měnič vrátí telegram, tato informace se vypíše do textového pole, můžeme zmačknout tlačítko *zpracování telegramu*, po kliku na toto tlačítko provede program kontrolu správnosti telegramu, pokud přišla správná odpověď, telegram zpracuje a vytiskne ho do textového pole. Pro zautomatizování těchto úkonů slouží CheckBox se jménem *Automatizovat*, tím se spojí provedení všech tří úkonů.

Další společně provázané prvky jsou CheckBox *Scan* a tlačítko *Scanování*, zajišťuje scanování měniče, před zahájením scannování je třeba vymazat databázi, k tomu slouží tlačítko *Smazat memo*, po jeho stisknutí se program dotáže, zda opravdu chcete smazat databázi.

Nyní zbývá objasnit už jen tlačítko *přečíst všechny hodnoty* a CheckBox *databáze*, opět spolu souvisí. Funkci, kterou zavolá tlačítko po kliknutí přečtení všech hodnot zapsaných v databázi a zapíše je zpět.

Císlo	Parametr	Popis	Index	Text
2	r2	r(act)	0	
4	r4	Output Amps	0	
6	r6	DC Bus Volts	0	
7	r7	Motor Torque	0	
8	r8	Motor Utilizat.	0	
9	r9	Motor Temperat.	0	
10	r10	Drive Utilizat.	0	
12	r12	Active BICD DSet	0	
13	r13	Active FuncDSet	0	
6.000999450683	P26	CouplCh fix	1	
6.002000808715			2	
6.003000259399			3	
6.00399971008			4	
6.004999160766			5	

Vypis obsahu telegramu a nastavení

Obrázek 7 Grid a Memo

Do těchto dvou vizualizačních prostředků ukládám přijatá data, v databázi, kterou zobrazuje grid jsou data uchována trvale, kdežto v memo pouze po dobu běhu programu, je možné si je uložit pomocí Ctrl + C a potom je vložit do jakéhokoliv textového dokumentu.

Příloha C

Zde jsou shrnuty informace získané při práci na DP.

PBE

V odpovědi PBE jsou ID a ID extension částečně dekodované informace. Již známe jsou.

ID

Prvních 8 bitů (0 – 7) udává typ parametru:

- O4 = 7; DW bez znaménka
- I4 = 4; DW se znaménkem
- O2 = 6; W bez znaménka
- I2 = 3; W se znaménkem
- L2 = 36; Konektory a binektory
- V2 = 35; Binární

neznáme parametry (nevizuální v Drive monitoru)

- NeViz546 = 34;
- NeViz522 = 10;
- NeViz556 = 44;

10 bit pokud má hodnotu 1 znamená, že parametr má textový popis.

14 bit pokud má hodnotu 1 znamená, že parametr je typu pole.

ID extension

Pokud je v příslušném bitu uložena hodnota 1 pak má daný následující význam:

- 1 bit označuje, že parametr typu pole má textově popsány i hodnoty
- 3 bit označuje, že parametr je binektor
- 4 bit označuje, že parametr je konektor

- 8 bit označuje, že parametr je označen jako BDS (BICO data set)
- 9 bit označuje, že parametr je označen jako FDS (function data set)
- 10 bit označuje, že označuje zda je o duální konektor do kterého jdou připojit 2 binektory

Normalization

Přichází v opačném pořadí, než je uvedeno v dokumentaci USS protokolu.

Request or change text

V odpovědi posílá měnič popis hodnoty nebo podparametru. Podle typu textového popisu je třeba modifikovat dotaz. Popisy nelze získat najednou vložím 255 do Low Bytu. U popisu hodnot se musí vkládat do Low Bytu hodnoty od 1 do max. hodnoty parametru zvětšeného o 1. U parametru typu pole se udává index pole. V případě parametru typu pole s popisem podparametru i hodnot získání popisu podparametru probíhá stejně jako u pole. Pro získání popisu hodnot je třeba vložit do IND Higt byt 9 jednička a pak stejně jako u běžného parametru.

Binektory a konektory

U binektorů a konektorů si neodpovídají skutečné hodnoty s označením v kompendiu. Je mezi nimi spojitost jen v posloupnosti. Př.: Hodnota 21 odpovídá binektoru 15 a hodnota 37 binektoru 25.