

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ

KATEDRA ŘÍDICÍ TECHNIKY



DIPLOMOVÁ PRÁCE

**Záznamník dat pro vlakovou sběrnici WTB**

Praha 2003

Radek Křivský

## **Abstrakt**

Cílem této diplomové práce je navrhnout a realizovat software Front-End jednotky záznamníku dat pro vlakovou sběrnici WTB (*Wire Train Bus*). Tato práce vychází z projektu WTB komunikační jednotky realizovaného firmou *UniControls a.s.*

Záznamník dat se skládá z PC a Front-End jednotky, která je tvořena WTB jednotkou a procesorovým modulem. WTB jednotka snímá všechna data, která se na WTB sběrnici vyskytnou a zapisuje je do sdílené paměti. Odtud jsou tyto záznamy načítány procesorovým modulem a posílány po *Ethernet* sběrnici do PC, kde jsou ukládány do souboru s možností jejich dalšího zpracování. Spuštění resp. pozastavení komunikace mezi *Front-End* jednotkou a PC je řízeno uživatelem. Pro ověření funkčnosti celého zařízení je realizován software, který interpretuje přenesená data.

Tato diplomová práce také obsahuje stručný popis standardu IEC 61375-1 (*Train Communication Network*, TCN), jehož součástí je specifikace komunikačního protokolu *Wire Train Bus*. Pro popis algoritmů je použit jazyk SDL (*Specification and Description Language*). Základní popis SDL je v této práci též uveden.

### **Abstract**

The objective of this diploma thesis is to implement software for the Data Logger Front-End unit for *Wire Train Bus* (WTB). This thesis is based on the project WTB Communication unit built by *UniControls, Inc.*

The Data Logger comprises a PC and the Front-End unit, which comprises a WTB unit and a processor module. The WTB unit receives all data appearing on the *Wire Train Bus* and saves it in its shared memory. From there, these records are read by the processor module and sent via *Ethernet* to the PC, where they are saved to a file for possible off-line processing. Starting and holding of communication between the Front-End unit and the PC is controlled by the user. To prove the functionality of the whole device software interpreting the transferred data was implemented.

This diploma thesis also contains a brief description of the standard IEC 61375-1 (*Train Communication Network*, TCN), which includes the specification of *Wire Train Bus* communication protocol. Algorithms are described in *Specification and Description Language* (SDL). Base information about SDL is also presented.

## **Prohlášení**

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém 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 23. května 2003

.....

podpis

*Na tomto místě bych rád poděkoval vedoucímu a oponentovi této diplomové práce Dr. Ing. Zdeněku Hanzálkovi a Ing. Dobromilu Nenutilovi za odborné vedení a vytvoření dobrých pracovních podmínek při řešení této diplomové práce. Zvláště bych chtěl poděkovat Ing. Pavlu Milenovskému a Ing. Aleši Hajnému za četné konzultace, vstřícný přístup a vždy ochotnou pomoc. Dále děkuji Ing. Hynku Haškovi za pomoc s tvorbou grafického rozhraní k softwaru popsaného v kapitole 5.2.*

## Obsah

<b>1</b>	<b>STANDARDSY PRO ŘÍDICÍ SYSTÉMY VLAKOVÝCH SOUPRAV .....</b>	<b>8</b>
<b>2</b>	<b>TRAIN COMMUNICATION NETWORK.....</b>	<b>9</b>
2.1	VLASTNOSTI TCN SÍTĚ.....	9
2.1.1	Struktura TCN sítě.....	9
2.1.2	Způsob komunikace na sběrnících TCN sítě.....	10
2.1.2.1	Třídy přenášených dat .....	10
2.1.2.2	Provoz na sběrnících.....	11
2.1.2.3	Přístup na sběrnice.....	11
2.1.2.4	Forma datového přenosu .....	11
2.1.3	Spolehlivost TCN sítě.....	12
2.1.4	Vrstvy TCN protokolu.....	13
2.1.5	Řízení TCN sítě - Network Management.....	13
2.2	VOZOVÁ SBĚRNICE MVB.....	14
2.3	VLAKOVÁ SBĚRNICE WTB.....	15
2.3.2	Telegramy a rámce přenášené na WTB sběrnici .....	17
2.3.2.1	Typy WTB Telegramů.....	18
2.3.2.2	Integrita WTB rámců.....	19
2.3.3	Inaugurace WTB sběrnice .....	19
2.3.4	Redundance uzlů na WTB sběrnici .....	21
2.3.4.1	Redundance Master uzlu - správy sběrnice .....	21
<b>3</b>	<b>JAZYK SDL.....</b>	<b>22</b>
3.1	PŘÍKLAD POUŽITÍ SDL .....	23
3.2	POUŽITÍ SDL V TÉTO PRÁCI.....	24
<b>4</b>	<b>ROZBOR DANÉHO PROBLÉMU .....</b>	<b>25</b>
4.1	SPECIFIKACE PROBLÉMU .....	25
4.1.1	Požadovaná funkce záznamníku dat.....	25
4.1.2	Požadovaná struktura záznamníku dat .....	26
4.2	POPIS WTB KOMUNIKAČNÍ JEDNOTKY - WTB 2000 .....	27
4.2.1	Záznam dat v jednotce WTB 2000.....	27
4.2.2	Vestavěné testy v jednotce WTB 2000.....	28
4.2.3	Doba příjmu a vyslání WTB_ rámce jednotkou WTB 2000 .....	28
4.3	POROVNÁNÍ FRONT-END JEDNOTKY A JEDNOTKY WTB 2000 .....	29
<b>5</b>	<b>NÁVRH SOFTWARE ZÁZNAMNÍKU DAT.....</b>	<b>30</b>
5.1	SOFTWARE WTB JEDNOTKY .....	30
5.1.1	Algoritmus činnosti softwaru WTB jednotky .....	30
5.1.2	Popis jednotlivých částí softwaru WTB jednotky .....	31
5.1.2.1	Chování WTB jednotky vzhledem k WTB sběrnici.....	31
5.1.2.2	Příjem dat - zpracování přerušení.....	31
5.1.2.3	Struktura oblasti záznamů dat.....	32
5.1.2.4	Přepínání stránek záznamů .....	33
5.1.3	Testování softwaru WTB jednotky .....	33
5.1.3.1	Testování správnosti ukládání dat do paměti.....	33
5.1.3.2	Měření času nutného k příjmu WTB_ rámce .....	36
5.2	SOFTWARE PROCESOROVÉHO MODULU A SOFTWARE PRO PC.....	37
5.2.1	Algoritmus činnosti softwaru procesorového modulu a softwaru pro PC.....	37
5.2.2	Uživatelské rozhraní softwaru pro PC.....	39
5.2.3	Časové nároky na komunikaci mezi procesorovým modulem a PC.....	39
5.2.4	Testování softwaru procesorového modulu a softwaru pro PC.....	41
5.2.5	Operační systém OS-9.....	42
5.3	SOFTWARE PRO INTERPRETACI PŘENESENÝCH DAT .....	43

5.3.1	Popis činnosti softwaru pro interpretaci přenesených dat .....	43
5.3.2	Testování softwaru pro interpretaci přenesených dat .....	45
<b>6</b>	<b>ZÁVĚREČNÉ ZHODNOCENÍ VÝSLEDKŮ .....</b>	<b>46</b>
<b>7</b>	<b>SEZNAM POUŽITÉ LITERATURY A DALŠÍCH ZDROJŮ .....</b>	<b>47</b>
<b>PŘÍLOHY I.....</b>	<b>.....</b>	<b>48</b>
A.1	TERMÍNY A ZKRATKY .....	48
A.2	SHRNUTÍ VLASTNOSTÍ TRAIN COMMUNICATION NETWORK .....	49
A.3	FOTOGRAFIE.....	50
<b>B</b>	<b>PŘÍLOHY II - CD DISK .....</b>	<b>51</b>

## **1 Standardy pro řídicí systémy vlakových souprav**

Standardizace je důležitým předpokladem pro rozvoj v mnoha průmyslových odvětvích. Jak je uvedeno v [1] platí to i pro oblast železniční dopravy, jenž má v každém ohledu nadnárodní působnost. Proto se již po řadu let připravují a uvádějí do života standardy, které mimo jiné zajišťují, že vlakové soupravy mohou být sestaveny z vozů různých výrobců a jsou schopny komunikovat se zabezpečovacími a dalšími stacionárními zařízeními v různých státech jednotným způsobem.

Základními standardy pro řešení řídicích systémů vlakových souprav jsou IEC 61375-1 (*Train\_Communication\_Network*, TCN) a UIC 556 (*Information Transmission in the Train*). Zatímco IEC standard specifikuje komunikační protokoly, které se používají k přenosu dat, UIC standard definuje obsah přenášených dat a řízení komunikační sítě dle aktuální kompozice soupravy. Teprve implementace obou výše uvedených standardů zajišťuje interoperabilitu u vozů různých výrobců.

Vlaková souprava je systémem s proměnlivou kompozicí. Tím se liší od řídicích systémů pro jiné aplikační oblasti. Ke změně kompozice dochází v důsledku řady událostí, např. rozpojení soupravy, spojení souprav, změně stanoviště strojvedoucího, výpadku systému některého vozu apod. Nutnost vyrovnat se s celou řadou možných událostí činí řídicí systém vlakových souprav výrazně složitějším ve srovnání s běžnými systémy procesního řízení.



## 2 Train Communication Network

Tato kapitola poskytuje základní charakteristiku standardu IEC 61375-1 [2].

*Train Communication Network* (vlaková komunikační síť, TCN) definuje komunikační systém mezi elektronickými zařízeními umístěnými v témže nebo v různých vozech vlakové soupravy za účelem:

- řízení vozidla (řízení pohonů, ovládání brzd, dveří, světel, vytápění apod.),
- dálkové diagnostiky a údržby,
- zajištění informačních služeb pro cestující.

Základní cíle TCN jsou:

- Interoperabilita na úrovni vlaku - umožnit propojení vozů různých výrobců,
- Interoperabilita na úrovni vozu - umožnit snadnou výměnu jednotlivých zařízení ve voze,
- Podpora při uvádění do provozu, údržbě a redukce ceny životního cyklu výrobku,
- Otevřenost k externím systémům, propojení na pozemní systémy.

### 2.1 Vlastnosti TCN sítě

Přehledné shrnutí základních vlastností TCN je uvedeno v příloze A.2.

#### 2.1.1 Struktura TCN sítě

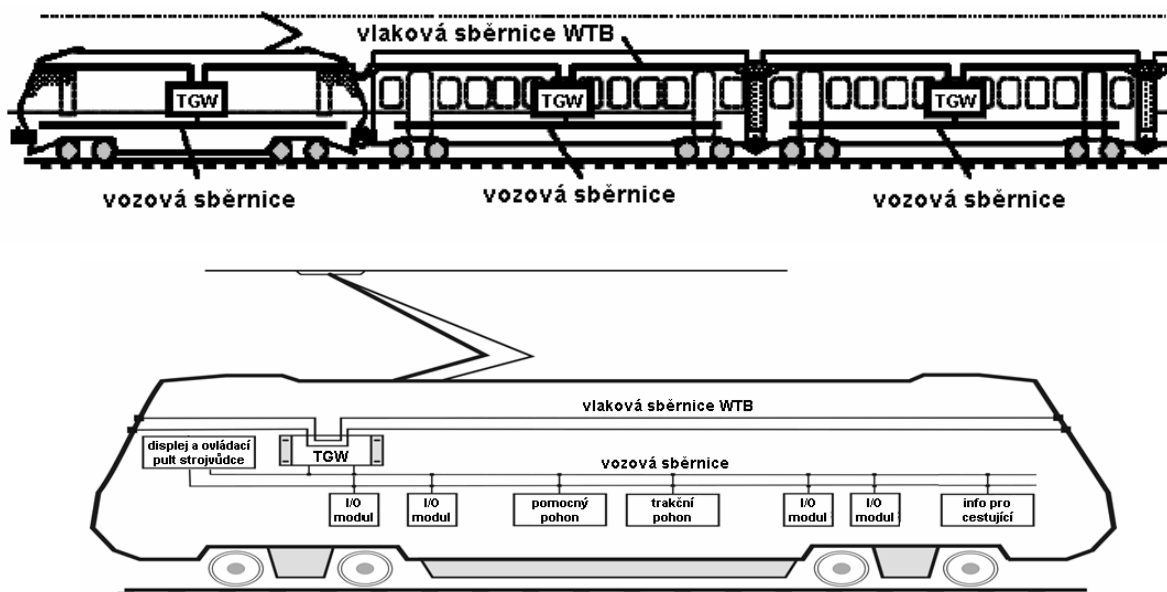
TCN má dvouúrovňovou hierarchickou strukturu (obr. 2.1):

- Vozová sběrnice (*Multifunctional\_Vehicle\_Bus*, MVB) propojuje zařízení umístěné uvnitř jednoho vozu nebo nedělitelné skupiny vozů.
- Vlaková sběrnice (*Wire\_Train\_Bus*, WTB) propojuje jednotlivé vozy celé soupravy. Je schopná vlastní konfigurace v případě zprovoznění vlakové soupravy nebo po změně její kompozice. (Tento proces se nazývá „inaugurace“.)

Vozová a vlaková sběrnice jsou propojeny přes komunikační uzel (*Train\_Bus\_Gateway*, TGW). TCN tak umožňuje přímou komunikaci mezi zařízeními připojenými k vozovým

sběrnicím různých vozů. Do některých TGW nemusí být připojena žádná vozová sběrnice, naopak, do některých jich může být zapojeno více.

Poznámka: Pokud se má na mysli pouze WTB sběrnice, nepoužívá se termín *Train\_Bus\_Gateway*, ale termín *Node* resp. *uzel*.



Obr. 2.1 Struktura TCN sítě [2]

## 2.1.2 Způsob komunikace na sběrnicích TCN sítě

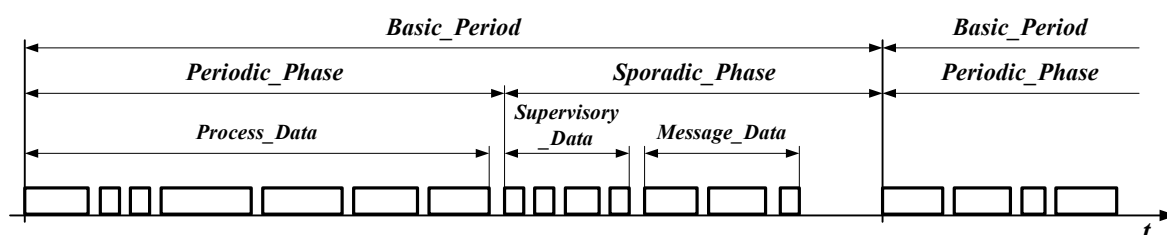
### 2.1.2.1 Třídy přenášených dat

Obě sběrnice - MVB i WTB - přenášejí tři třídy dat:

- *Process\_Data* (procesní data) - časově kritická, periodicky přenášená data, která jsou přijímána více zařízeními. Jsou používána hlavně pro přenos omezeného počtu krátkých procesních proměnných - *Variable* (např. pro řízení pohonů, ovládání brzd apod.).
- *Message\_Data* - méně urgentní data, na vyžádání zaslána konkrétnímu zařízení. Zpravidla se jedná o delší zprávy - *Message* (např. pro diagnostiku).
- *Supervisory\_Data* - interní data pro řízení linkové vrstvy - přenášena za účelem dohledu nad jednotlivými zařízeními v síti, předávání správy sběrnice MVB a při inauguraci WTB sběrnice.

### 2.1.2.2 Provoz na sběrnících

Provoz na sběrnících probíhá v časových kvantech - základních periodách (*Basic\_Period*, BP). Délka jedné BP je 1 ms na MVB a 25 ms na WTB. *Basic\_Period* se dělí na *Periodic\_Phase* a *Sporadic\_Phase*. Během *Periodic\_Phase* se uskutečňuje přenos periodicky přenášených dat (*Process\_Data*), zatímco *Sporadic\_Phase* je určena k přenosu nepravidelně přenášených dat (*Message\_Data*, *Supervisory\_Data*).



Obr. 2.2 Provoz na sběrnících TCN sítě

TCN kombinuje přenos dat dvou tříd s odlišnými časovými nároky. Maximální zpoždění přenosu procesních proměnných v celé TCN síti je deterministické. Časově kritické procesní proměnné jsou přenášeny mezi aplikacemi v různých vozech za méně než 100 ms, mezi aplikacemi v témže voze za méně než 50 ms. Požadavky na časování MVB jsou větší než na WTB. Časová odezva na MVB je přibližně desetkrát rychlejší než na WTB.

### 2.1.2.3 Přístup na sběrnice

Jak na MVB, tak na WTB sběrnici je provoz řízen *Master* zařízením (na WTB myšleno *Master uzlem*). Z tohoto důvodu je na sběrnici vždy jedno zařízení *Master* a všechna ostatní zařízení jsou *Slave*. *Slave* zařízení nemohou přistupovat na sběrnici bez vyzvání správce sběrnice (*Master* zařízení). Obecně na sběrnících může být více zařízení, které se mohou stát *Master* zařízením.

### 2.1.2.4 Forma datového přenosu

Komunikace na MVB i WTB probíhá formou *Telegramů*, které se skládají z *Master\_Frame* rámce a následného *Slave\_Frame* rámce.

Master zařízení (správce sběrnice) vyšle *Master\_Frame* rámec, který dekodují všechna zařízení (na WTB pod pojmem „zařízení“ je myšleno „uzel“), čímž žádá určité zařízení o dané informace. Poté adresované zařízení odpovídá - vysílá *Slave\_Frame* rámec, který může být přijímán více zařízeními.

Interval mezi *Master\_Frame* a *Slave\_Frame* musí být kratší než  $43 \mu\text{s}$  (MVB) /  $300 \mu\text{s}$  (WTB). Aby bylo zařízení schopné odpovědět v tak krátkém čase, jsou jednotlivé *Slave\_Frame* rámce připraveny předem. K tomuto účelu každé zařízení umísťuje svá *Process\_Data* do registrů (*Port*) a *Message\_Data* do front (*Queue*) ve vyhrazené paměti *Traffic\_Memory*.

Master zařízení periodicky vyzývá jednotlivá zařízení k přenosu obsahu jejich *Port* registrů v předem definovaných sekvencích stanovených před započítím provozu / při inauguraci. Tedy pro každý *Port* registr je definována *Individual\_Poll\_Period* / *Individual\_Period* (individuální perioda dotazování), která je násobkem *Basic\_Period*.

### 2.1.3 Spolehlivost TCN sítě

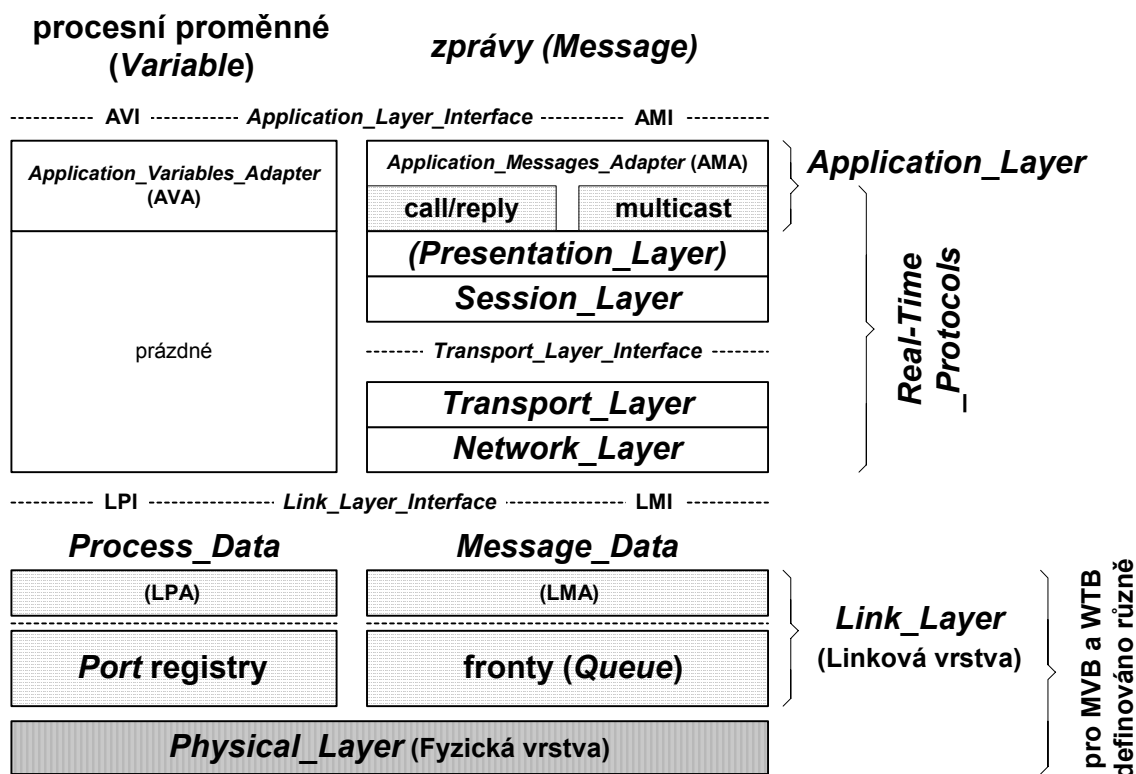
TCN je možné nakonfigurovat tak, že chyba či nefunkčnost kterékoliv komponenty nezpůsobí nefunkčnost sítě. Úroveň redundance závisí na požadavcích kladených na konkrétní aplikaci. TCN je schopný se vyrovnat s následujícími jevy:

- občasné porušení přenášených dat nebo jejich ztráta,
- přerušení přenosového média nebo zkrat na něm,
- trvalé vysílání poškozeného vysílače na linku,
- zamítnutí přístupu na médium,
- konfigurační chyba,
- krátkodobá ztráta *uzlů* a jejich opětná integrace do soupravy.

Obnovení standardní komunikace na WTB sběrnici v případě změny kompozice soupravy nebo krátkodobého rozpojení soupravy v důsledku chyby nějakého zařízení (ztráta *uzlů*) netrvá déle než 1s při 22 vozech.

### 2.1.4 Vrstvy TCN protokolu

TCN protokol (obrázek 2.3) je rozvrstven dle ISO/OSI modelu. Fyzická a linková vrstva jsou definovány pro MVB a WTB sběrnice různě. Vyšší vrstvy TCN protokolu jsou společné (definovány tzv. *Real-Time Protocols*, RTP).



Obr. 2.3 Rozvrstvení TCN protokolu

Poznámka: Některé termíny a zkratky na obrázku 2.3 jsou uvedeny pouze pro úplnost a dále se nepoužívají. Proto jejich význam není v textu blíže vysvětlen.

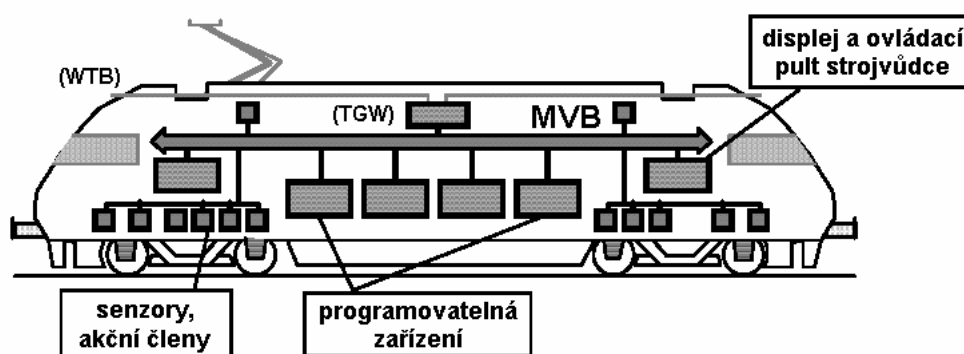
### 2.1.5 Řízení TCN sítě - Network Management

*Network Management* (TNM) poskytuje soubor služeb, které podporují jednu nebo více ze čtyř fází:

- testování,
- uvádění do provozu,
- běžný provoz,
- údržba.

## 2.2 Vozová sběrnice MVB

Vozová sběrnice MVB (*Multifunctional\_Vehicle\_Bus*) je určena k propojení zařízení umístěných uvnitř jednoho vozu nebo uvnitř nedělitelné skupiny vozů vlakové soupravy. Poskytuje propojení jak všech programovatelných zařízení mezi sebou, tak i propojení těchto zařízení se senzory a akčními členy. MVB je schopna adresovat až 4095 zařízení, z nichž 256 jsou schopna účasti komunikace na úrovni přenosu zpráv. Doba odezvy nepřekročí 16 ms. Struktura MVB a její základní komponenty jsou znázorněny na následujícím obrázku.



Obr. 2.4 Vozová sběrnice MVB v lokomotivě [2],

V jednom voze může být několik vozových sběrnic s vlastním *Master* zařízením, které mohou být připojeny na vlakovou sběrnici pomocí *Train\_Bus\_Gateway*.

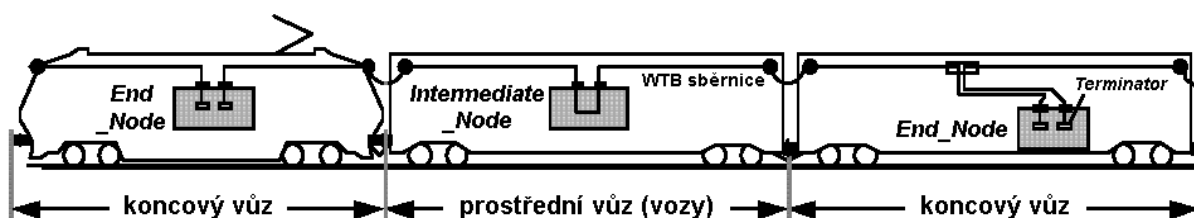
Na MVB se lze setkat se třemi typy přenosového média:

- kroucená dvojlinka (do vzdálenosti až 20 m),
- stíněná kroucená dvojlinka a transformátory pro galvanické oddělení (do vzdálenosti až 200 m),
- optické vlákno (do vzdálenosti až 2000 m).

Společnou charakteristikou všech typů fyzického rozhraní je přenosová rychlost 1,5 Mbit/s. Elektrická média mívají topologii typu sběrnice, zatímco optické médium má většinou topologii typu hvězda. MVB používá tzv. *Manchester* kódování.

## 2.3 Vlaková sběrnice WTB

WTB (*Wire Train Bus*) je sériová sběrnice navržena, pro propojení vozidel, které mohou být při běžném provozu spojovány a rozpojovány. Na WTB může být připojeno až 32 uzlů (*Node*), které umožňují její propojení s vozovou sběrnici, případně dalšími zařízeními. WTB je schopna komunikačně propojit až 22 vozů vlakové soupravy, což odpovídá maximální délce kabelu 860 m. Strukturu WTB a její základní komponenty znázorňuje obrázek 2.5.



Obr. 2.5 Vlaková sběrnice WTB [2]

Provoz sběrnice je řízen jedním z uzlů - *Master uzlem*, ostatní se označují *Slave uzly*. Uzly se dále dělí na prostřední (*Intermediate\_Node*) a koncové (*End\_Node*).

### 2.3.1 Fyzické rozhraní WTB sběrnice

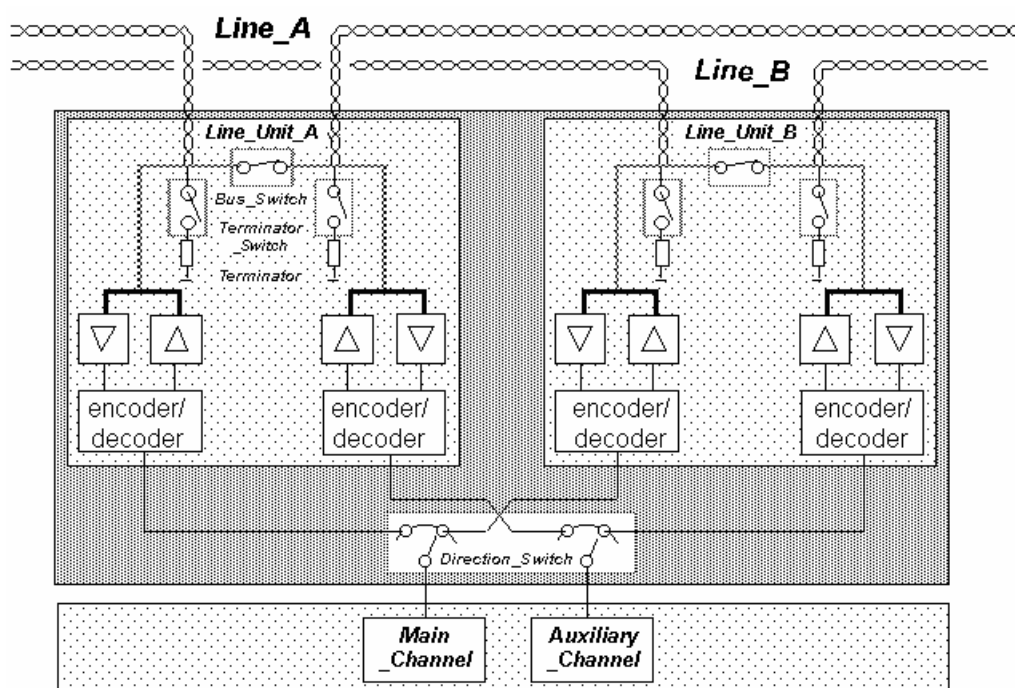
Jako přenosové médium je použita stíněná kroucená dvojlinka. Přenosová rychlost je 1 Mbit/s. Signály přenášené na WTB se vyznačují vysokou imunitou proti rušení. To je dáno použitým kódováním - tzv. inverzní *Manchester* kódování, fázovým závěsem založeným na DFT a statistickým vyhodnocením signálu 0/1 (32 vzorků).

WTB (ale také MVB) podporuje redundanci ve smyslu vedení signálu ve dvou kabelech (kanály *Line\_A*, *Line\_B*). Každý uzel komunikuje s WTB pomocí *Medium Attachment Unit* (připojovací jednotka, MAU), která obsahuje dvě *Line\_Unit* - pro každý kabel jednu. Tato struktura je znázorněna na obrázku 2.6 níže. Uzel vždy vysílá do obou kanálů, ale přijímá pouze z jednoho, zatímco sleduje druhý a kontroluje, zda je stále funkční. Maximální časový rozdíl mezi signály v jednotlivých kanálech (*skew*) na straně přijímače nesmí překročit 32  $\mu$ s.

Každá *Line\_Unit* obsahuje dva vysílače/přijímače (pro každý směr jeden), které jsou od sběrnice galvanicky odděleny transformátorovou vazbou. Dále obsahuje *Bus\_Switch*

a pro každý směr jeden *Terminator* a *Terminator\_Switch*. Spínač *Bus\_Switch*, spojuje oba úseky sběrnice pokud je daný uzel *Intermediate\_Node*. Impedance *Terminator* a spínač *Terminator\_Switch* slouží k impedančnímu přizpůsobení v případě, že se jedná o *End\_Node*.

Přepínač *Direction\_Switch* spíná *Main\_Channel* (hlavní kanál) a *Auxiliary\_Channel* (pomocný kanál) k odpovídajícím úsekům sběrnice. Pokud je daný uzel *Intermediate\_Node* komunikace se účastní pouze *Main\_Channel*. Pokud se jedná o *End\_Node*, *Main\_Channel* je připojen k úseku sběrnice směrem k *Master uzlu* a provozuje běžnou komunikaci s ostatními uzly, zatímco *Auxiliary\_Channel* testuje konec sběrnice, zda na ní nebyl připojen další uzel.



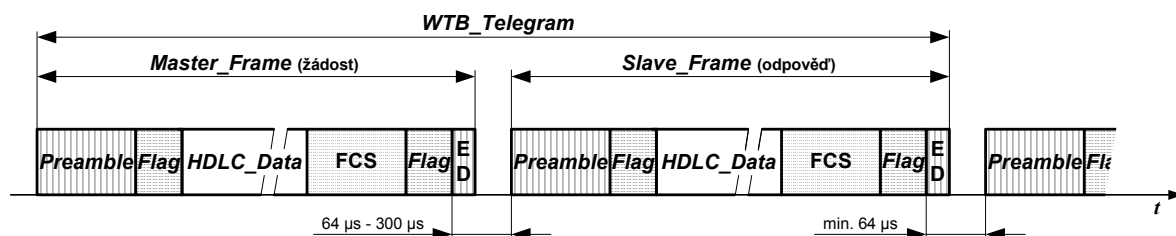
Obr. 2.6 *Medium\_Attachment\_Unit* [3]

*Medium\_Attachment\_Unit* může dále obsahovat tzv. *Fritting\_Circuit*. Tento obvod zajišťuje elektrické čištění zoxidovaných kontaktů konektorů, spojujících WTB sběrnici mezi jednotlivými vozy, stejnosměrným proudem podloženým pod užitečný signál.



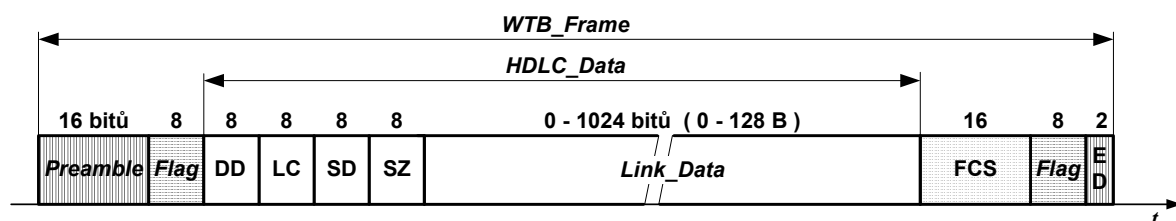
### 2.3.2 Telegramy a rámce přenášené na WTB sběrnici

Komunikace na WTB sběrnici je uskutečňována formou *WTB\_Telegramů* (obr. 2.7), které se skládají z *Master\_Frame* a *Slave\_Frame* rámců.



Obr. 2.7 WTB\_Telegram

Všechny rámce na WTB mají stejné kódování. Jedná se o bitově orientovaný formát dle standardu HDLC. *WTB\_Frame* (obr. 2.8) uvozuje *Preamble*, což je opakovaná sekvence „01“ sloužící pro synchronizaci. Dále obsahuje dva *Flag* znaky („01111110“), jenž ohraničují vlastní *HDLC\_Data* spolu s *Frame\_Check\_Sequence* (kontrolní sekvence rámce, FCS). Aby se zabránilo výskytu sekvence odpovídající *Flag* znaku uvnitř *HDLC\_Data*, vysílač po každých pěti po sobě jdoucích „1“ vkládá „0“. Tedy v nejhorším případě se objem dat zvětší téměř o 20 %. (Maximální délka *WTB\_Frame* rámce je 1323 bitů.) Příjímač vložené bity opět odstraňuje. *WTB\_Frame* zakončuje *End\_Delimiter* (koncový oddělovač, ED).



Obr. 2.8 WTB\_Frame [3]

*HDLC\_Data* se skládají ze čtyř úvodních bytů (DD, LC, SD, SZ) a *Link\_Data*. Význam úvodních bytů je následující:

DD - *Destination\_Device* (adresa příjemce),

LC - *Link\_Control* (typ dat),

SD - *Source\_Device* (adresa odesílatele),

SZ - *Link\_Data\_size* (velikost *Link\_Data* v bytech).

Poznámka: V této práci je pro *WTB\_Frame* často také používán termín *WTB\_rámeček*.

### 2.3.2.1 Typy WTB Telegramů

Na WTB jsou definovány tři typy *Telegramů*. *Slave uzly* vždy odpovídá rámcem stejného typu, jakým byl *Master uzlem* dotázán.

#### *Process\_Data\_Telegram*

*Master\_Frame* neobsahuje kromě úvodních čtyř bytů žádná další data (SZ = 0), SD = 01. *Slave\_Frame* je přijímán všemi ostatními zařízeními (DD = FF). K zvýšení integrity během změny kompozice soupravy jsou první dva byty *Process\_Data* vyhrazeny pro obsah daného rámce.

*Uzly* s časově kritickými *Process\_Data* (například motorové vozy) mohou být dotazovány každou *Basic\_Period* (25 ms), jiné s méně urgentními daty jsou dotazovány v násobcích BP - v tzv. *Individual\_Period*. Toto je určeno během inaugurace.

V každé *Basic\_Period* je dotazován jeden *End\_Node*. Tím je kontrolována integrita soupravy.

#### *Message\_Data\_Telegram*

*Master\_Frame* neobsahuje kromě úvodních čtyř bytů žádná další data (SD = 01).

*Slave\_Frame* může být adresován jak pouze jednomu zařízení, tak také všem. *Message\_Data* začínají tzv. *Network\_Header*, která se skládá z *Final\_Network\_Address* (2 B), *Origin\_Network\_Address* (2 B) a *Message\_Transport\_Control* (1 B).

#### *Supervisory\_Data\_Telegram*

Jak *Master\_Frame*, tak *Slave\_Frame* mohou obsahovat *Supervisory\_Data*.

Některé *Supervisory\_Data\_Telegramy* mohou obsahovat pouze *Master\_Frame*.

### 2.3.2.2 Integrita WTB rámců

Integrita *WTB\_rámců* je zajištěna následujícími kontrolními mechanismy:

- *Frame\_Check\_Sequence* je CRC kód, který poskytuje Hamingovu vzdálenost čtyři. (Generační polynom CRC kódu je standardní CCITT polynom.)
- Aby byl daný bit shledán opačného významu, muselo by dojít, při použití inverzním *Manchester* kódování, k chybě v obou polovinách příslušné bitové buňky.
- Skutečná velikost rámce musí odpovídat velikosti uváděné v bytu SZ (*Link\_Data\_size*).

### 2.3.3 Inaugurace WTB sběrnice

Inaugurace (*Inauguration*) je proces konfigurace vlakové sběrnice. Probíhá v případě zprovoznění vlakové soupravy, po změně její kompozice (např. rozpojení soupravy, spojení souprav, změna stanoviště strojvedoucího) nebo v důsledku nedostupnosti koncového uzlu či zhroucení *Master uzlu* sběrnice.

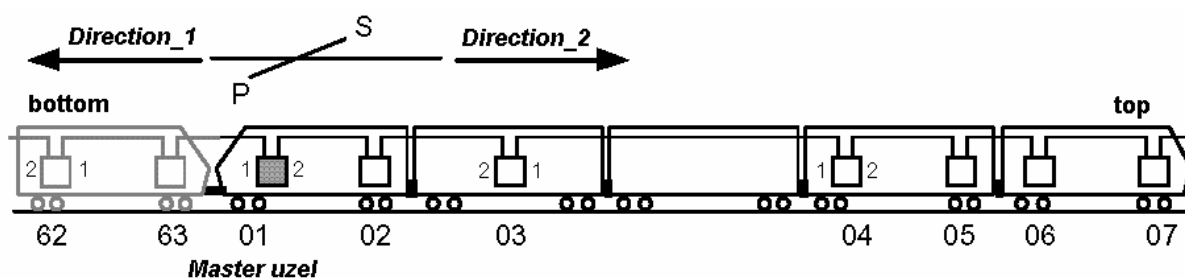
Inaugurace se skládá z následujících kroků:

- Určení *Master uzlu* sběrnice.
- *Master* postupně detekuje *uzly* v obou směrech. Jednotlivé úseky sběrnice, mezi již nalezenými *uzly*, se propojují v jeden celek na obou koncích zakončený *Terminator* impedancemi (obr 2.5 výše).
- Každému *uzlu* je přidělena adresa respektující jeho pozici vzhledem k *Master uzlu*.
- Každý *uzel* oznamuje *Master uzlu*, jak často chce být dotazován na svá *Process\_Data* (tzv. *Individual\_Period*) a svou charakteristiku (*Node\_Descriptor*).
- *Master* určí dotazovací strategii a vytvoří tzv. *Topography* (informaci o topografii sítě).
- Každý *uzel* obdrží informaci o topografii sítě a kdy bude dotazován na svá *Process\_Data*.

Poté začne běžný provoz sběrnice - *Master* začne dotazovat jednotlivé *uzly*. Celý proces inaugurace netrvá déle než 1s.

Při inauguraci jsou dána následující pravidla:

- *Master* obdrží adresu 01.
- *Master* určí „bottom“ soupravy - *Direction\_1* a „top“ soupravy - *Direction\_2*. (Dáno orientací *Line\_A* a *Line\_B* (*Side\_A*, *Side\_B*) - nezávisí na směru jízdy.)
- *Master* pojmenuje uzly ve směru *Direction\_1* sestupně začínající adresou 63 a uzly ve směru *Direction\_2* vzestupně začínající adresou 02.
- Strana vlaku odpovídající straně *Side\_A* vozu, ve kterém je *Master uzel*, je nazvána „P“, druhá „S“.



Obr. 2.9 Schéma adresace uzlů [3]

Na WTB se lze setkat s různými typy konfigurace. Nejčastější jsou tyto:

- Jeden *Strong\_Node* (silný) uzel a *Slave* uzly.  
*Strong\_Node* je určen aplikací stát se *Strong\_Master* uzlem.
- Všechny uzly jsou *Weak\_Node* (slabé) uzly.  
Jeden z nich se stane *Weak\_Master* uzlem do doby, než se objeví nějaký silnější *Master uzel*.

Při inauguraci mohou nastat dva typy kolizí:

- *Master\_Conflict* (např. po spojení souprav),
- *Weak\_Master\_Conflict* (během běžné inaugurace).

Pro řešení těchto problémů jsou definována jasná pravidla.

### **2.3.4 Redundance uzlů na WTB sběrnici**

*Uzly* na WTB sběrnici zpravidla duplikovány nejsou. Ovšem některé aplikace mohou být závislé přímo na funkcích poskytovaných pouze daným *uzlem* (např. *uzly* odpovídající řídicím, koncovým nebo motorovým vozům). V takovém případě je nutné, pro zvýšení spolehlivosti, daný *uzel* fyzicky zdvojit.

#### **2.3.4.1 Redundance Master uzlu - správy sběrnice**

Pokud dojde k výpadku *Master uzlu* na sběrnici, jejíž provoz je řízen *Weak\_Master uzlem*, (který není fyzicky zdvojen,) správu sběrnice převezme některý další *Weak\_Node uzul*.

Pokud se jedná o sběrnici řízenou *Strong\_Master uzlem*, je daný *uzel* zpravidla fyzicky zdvojen. Poté mluvíme o tzv. *On\_Line\_Master uzlu*, který za normálních okolností řídí sběrnici a *Stand\_By\_Master uzlu*, jenž je připraven převzít správu sběrnice v okamžiku, kdy *On\_Line\_Master* selže.

### 3 Jazyk SDL

Tato kapitola uvádí základní vlastnosti jazyka SDL (*Specification and Description Language*). Uvedená tvrzení vychází z [4], [5] a [6].

SDL je moderní, objektově orientovaný, programovací jazyk vyšší úrovně s hierarchickou strukturou. Původně byl vyvinut jako standard pro telekomunikační průmysl. (SDL je standard Z.100 mezinárodní telekomunikační unie ITU-T). Ovšem brzy se začal používat také v mnoha dalších oblastech. SDL je určen pro popis složitých, komunikačních systémů a událostmi řízených systémů. Dále je také vhodný pro popis paralelních algoritmů a systémů reálného času.

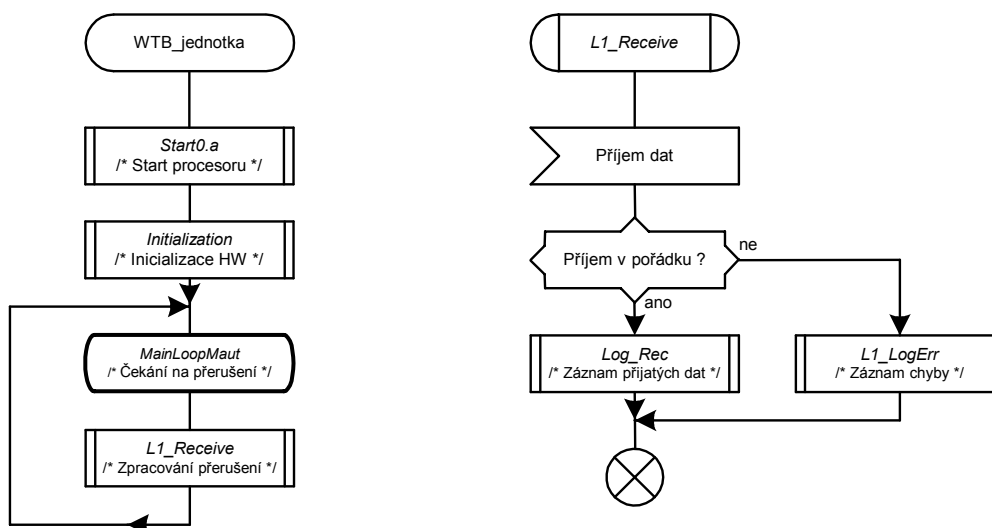
SDL má dvě rovnocenné formy zápisu, textovou (SDL/PR) a grafickou (SDL/GR). Častěji je používaná forma grafická. SDL má bohatou gramatiku, což zajišťuje jednoznačnost vyjádření. Díky tomu je možné vytvořit nástroje pro simulaci systémů popsanych v SDL. Přesnost a formálnost SDL poskytuje také možnost kompilace kódu do některého z jazyků nižší úrovně, jako např. jazyk C/C++.

Další předností grafické verze je fakt, že specifikaci problému v SDL (program) lze zároveň použít pro dokumentaci. Grafické vyjádření v SDL je jednoduše srozumitelné, což napomáhá lepší komunikaci mezi zákazníkem a výrobcem.

SDL je dnes používáno řadou předních výrobců v různých oblastech průmyslu. Nejčastěji je použito v automatizační technice, telekomunikačním, automobilovém a leteckém průmyslu, ale lze se s ním setkat také v dalších bezpečnostně kritických aplikacích.

### 3.1 Příklad použití SDL

Programování v grafické verzi SDL spočívá v propojování jednotlivých funkčních bloků. Popis grafického značení bloků pro různé typy operací je popsán v kapitole 3.2. Příklad použití SDL/GR je zobrazen na následujícím obrázku.

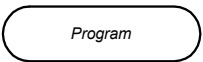

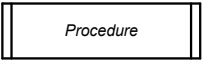
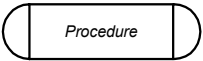



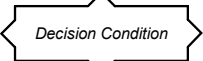
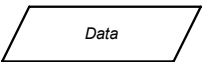
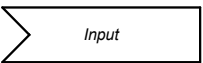
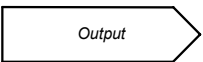


Obr. 3.1 Příklad použití SDL

Diagram na obrázku 3.1 znázorňuje algoritmus programu *WTB\_jednotka* a jím volané procedury *L1\_Receive*. Po spuštění programu *WTB\_jednotka* se volají procedury *Start0.a* a *Initialization*. Poté se ve smyčce *MainLoopMaut* čeká na přerušení. V případě jeho vyvolání se volá obsluhovací rutina *L1\_Receive*. Po obslužení přerušení se běh programu vrací do smyčky *MainLoopMaut*. V proceduře *L1\_Receive* dojde k příjmu dat a jejich zaznamenání do paměti. Pokud příjem proběhl v pořádku, záznam vykoná procedura *Log\_Rec*, v opačném případě procedura *L1\_LogErr*.

### 3.2 Použití SDL v této práci

V této práci je jazyk SDL použit pouze pro grafické vyjádření algoritmů - vývojových diagramů jednotlivých částí softwaru. Zmíněné diagramy obsahují bloky resp. symboly popsané v následující tabulce.

Název symbolu	Značka symbolu	Popis symbolu
<i>Start_symbol</i>		Začátek programu <i>Program</i>
<i>Stop_symbol</i>		Konec programu
<i>Procedure_call_symbol</i>		Volání procedury <i>Procedure</i>
<i>Procedure_start_symbol</i>		Začátek procedury <i>Procedure</i>
<i>Procedure_return_symbol</i>		Konec procedury
<i>Block_symbol</i>		Vykonání operace <i>Block</i>
<i>State_symbol</i>		Program je ve stavu <i>State</i> (zpravidla čekání na určitou událost)
<i>Decision_symbol</i>		Větvení programu na základě podmínky <i>Decision Condition</i>
<i>Save_symbol</i>		Uložení dat <i>Data</i>
<i>Input_symbol</i>		Vstup dat <i>Input</i>
<i>Output_symbol</i>		Výstup dat <i>Output</i>

Tabulka 3.1 Symboly jazyka SDL/GR použité v této práci

Poznámka: Pro lepší přehlednost daných algoritmů není v této práci záměrně dodržována hierarchická struktura SDL. Algoritmy procedur (funkcí) jsou často rozkreslovány přímo v algoritmu hlavní smyčky.



## 4 Rozbor daného problému

Výsledkem této diplomové práce má být software *Front-End\_jednotky\_záznamníku\_dat* pro vlakovou sběrnici WTB. Tato kapitola specifikuje požadované vlastnosti *záznamníku\_dat* a popisuje projekt WTB komunikační jednotky (*WTB 2000*) [9], [10], [11] realizovaný firmou *UniControls a.s.*, z kterého tato práce vychází.

### 4.1 Specifikace problému

#### 4.1.1 Požadovaná funkce záznamníku dat

*Front-End\_jednotka* má snímat všechny typy rámců, které jsou pro komunikaci na vlakové sběrnici WTB definovány standardem IEC 61375-1 [2] a přenášet je vhodným protokolem, v některém ze standardizovaných formátů, do PC.

Zaznamenávány mají být tyto informace:

- čas přijetí rámce (strojový čas [ $\mu$ s] a UTC čas [s]),
- stavová informace (z kterého kanálu a směru je daný rámec přijat),
- celá *HDLC\_Data* (DD, LC, SD, SZ a *Link\_Data*).

Dále je požadováno zaznamenávat informace o vnitřních událostech (např. sepnutí spínačů *Terminator\_Switch* a *Bus\_Switch*) a o detekovaných chybách. U těchto záznamů se má zaznamenávat:

- čas vzniku události / detekce chyby (strojový čas [ $\mu$ s] a UTC čas [s]),
- o jakou událost/chybu se jedná,
- případně další informace specifické pro danou událost/chybu.

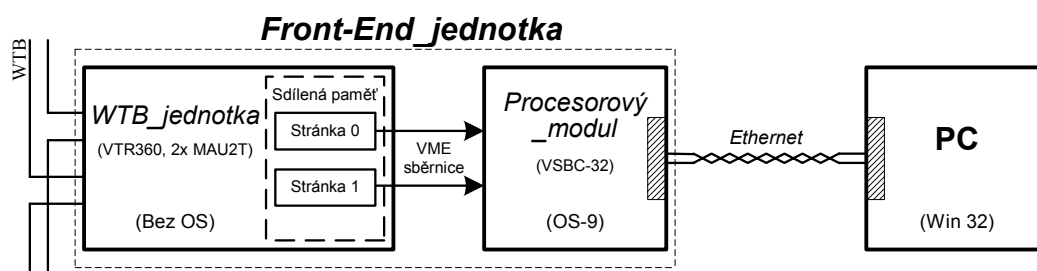
Strojový čas [ $\mu$ s] má mít význam hlavně pro určení časové vzdálenosti mezi dvěma po sobě jdoucími záznamy, UTC čas [s] udává absolutní čas (počet vteřin od 1.1.1970 dle greenwichského času).

#### 4.1.2 Požadovaná struktura záznamníku dat

Záznamník\_dat se skládá ze tří částí (obrázek 4.1):

- *WTB\_jednotka* - karta *VTR360* (procesor *MOTOROLA MC68360*), dvě karty *MAU2T* (mimo jiné obsahují budiče WTB sběrnice),
- *procesorového\_modulu* - karta *VSBC-32* (procesor *MOTOROLA MC68360*),
- PC.

*WTB\_jednotka* spolu s *procesorovým\_modulem* tvoří *Front-End\_jednotku*.



Obr. 4.1 Struktura záznamníku\_dat

*WTB\_jednotka* provádí vlastní odposlech WTB sběrnice. Tedy přijímá všechny typy *WTB\_rámců* a spolu se stavovými informacemi a časem přijetí daného rámce je ukládá do sdílené paměti. Podobně také zaznamenává informace o vnitřních událostech a chybách. *Procesorový\_modul* pouze čte data ze sdílené paměti, kam přistupuje po VME sběrnici, a posílá je po *Ethernet* sběrnici do PC, kde se ukládají do souboru s možností jejich dalšího zpracování. Spuštění resp. ukončení komunikace mezi *Front-End\_jednotkou* a PC je řízeno uživatelem z PC. Pro ověření funkčnosti celého zařízení je požadováno realizovat software, který interpretuje a zobrazí přenesená data.

Jak již bylo naznačeno, rozhraní mezi *WTB\_jednotkou* a *procesorovým\_modulem* je řešeno pomocí sdílené paměti. Ta se skládá ze dvou stránek, přičemž je střídavě vždy jedna určena k zápisu a druhá ke čtení.

Procesor *WTB\_jednotky* obsahuje multifunkční komunikační procesor, který je schopen mimo jiné vysílat a přijímat HDLC rámeček.

## 4.2 Popis WTB komunikační jednotky - WTB 2000

Jednotka *WTB 2000* [9], [10], [11] implementuje fyzickou a linkovou vrstvu komunikace vlakové sběrnice WTB dle standardu IEC 61375-1 [2].

Hardware jednotky *WTB 2000* se skládá z karty *VTR360* a dvou karet *MAU2T*.

Software jednotky *WTB 2000* obsahuje tyto komponenty:

- Inicializace,
- Základní ladicí prostředek (monitor),
- Zavaděč a programátor,
- Zabudované testy (*MAU2000-TEST*),
- WTB - fyzická vrstva,
- WTB - linková vrstva,
- WTB - obsluha rozhraní master procesoru,
- WTB - servisní diagnostika.

Software jednotky *WTB 2000* pracuje bez operačního systému.

### 4.2.1 Záznam dat v jednotce WTB 2000

Komponenta „WTB - fyzická vrstva“ realizuje mimo jiné také funkci ukládání všech událostí, které proběhly na daném *uzlu* za účelem zpětného rozboru. Toto je zajištěno pomocí tří záznamníků dat popsaných níže. Záznam dat v jednotce *WTB 2000* slouží pouze k servisním účelům. Zobrazení záznamů je možné přes servisní terminál nebo z *procesorového modulu*.

„Záznamník inicializační sekvence“ (*InitLog*) slouží k záznamu inicializační sekvence jednotky. (Jedná se o vnitřní událost daného *uzlu*.) Tvoří ho jedna stránka o 128 položkách.

„Hlavní záznamník dat“ (*Log*) zaznamenává vše kromě inicializační sekvence (přijaté a vyslané rámce, vnitřní události a detekované chyby). Za běžného provozu se aktuální záznamová stránka po jejím zaplnění opět přepisuje daty novými. (Jedna stránka obsahuje standardně 1024 položek.) Nastane-li mimořádná událost, zpravidla fatální chyba, je aktuální stránka „zmrazena“ a je založena stránka nová, do které pokračuje ukládání záznamů. Počet

stránek může být až 64. Dále tento záznamník dat může pracovat ve dvou režimech. V prvním režimu poslední stránku již „zmrazit“ nelze. Ve druhém, v případě „zmražení“ poslední stránky, se pro zápis nových dat opět uvolňuje stránka první atd.

„Záznamník chyb“ (*ErrorLog*) slouží k záznamu detekovaných chyb. Jedná se o jednu stránku o 1024 položkách, která se po jejím naplnění opět přepisuje.

Struktura jednotlivých záznamů (*LOG ITEM*) je následující:

- čas vyslání/přijetí rámce (strojový čas [ $\mu$ s],  
pokud se jedná o záznam chyby, tak také reálný čas [měsíc, den, hodina, minuta]),
- stavová informace (*Ctrl* byte, *Code* byte - „záhlaví“ záznamu),
- prvních 10 B *HDLC\_Data* / informace specifické pro danou událost/chybu.

#### **4.2.2 Vestavěné testy v jednotce WTB 2000**

Po připojení servisního terminálu k jednotce *WTB 2000* lze spustit komponentu *MAU 2000 - TEST*, jenž poskytuje provádění zabudovaných testů a konfiguraci parametrů jednotky. Mimo jiné umožňuje také čtení záznamníků dat a konfiguraci jejich parametrů.

#### **4.2.3 Doba příjmu a vyslání WTB\_rámce jednotkou WTB 2000**

Jednotka *WTB\_2000* je schopna vysílat *WTB\_rámce* s minimální periodou 198  $\mu$ s. Doba odezvy *Slave\_Frame* na *Master\_Frame* je mimálně 250  $\mu$ s.

### 4.3 Porovnání Front-End jednotky a jednotky WTB 2000

*Front-End\_jednotka* využívá některé hardwarové a softwarové komponenty vytvořené v projektu *WTB 2000*. Z údajů uvedených výše, mezi požadavky na *Front-End\_jednotku* a vlastnostmi jednotky *WTB 2000*, vyplývají níže uvedené podstatné rozdíly.

Zásadní rozdíl je v požadované funkci obou zařízení. Záznam dat v jednotce *WTB 2000* slouží pouze k servisním účelům. Trvale se uchovávají pouze záznamy předcházející mimořádným událostem. Od *Front-End\_jednotky* je požadováno zaznamenávat všechna data a přenášet je do PC k jejich dalšímu zpracování. Tomuto požadavku odpovídá také hardware *Front-End\_jednotky* obsahující kromě *WTB\_jednotky* také *procesorový\_modul*.

Jednotka *WTB 2000* přijímá pouze rámce určené danému uzlu a účastní se komunikace na sběrnici. Naproti tomu *Front-End\_jednotka* má zaznamenávat všechny typy rámců vyskytující se na WTB sběrnici a vlastní komunikace se účastnit nemá.

Dalším rozdílem je struktura jednotlivých záznamů ukládaných dat. Oproti jednotce *WTB 2000* je požadováno zaznamenávat celá *HDLC\_Data* a ke každému záznamu ukládat také odpovídající UTC čas.

Na rozdíl od jednotky *WTB 2000* není požadováno zaznamenávat inicializační sekvenci jednotky a zvláště vést záznamník chyb.

Rozdílný je také systém stránkování paměti ukládaných dat.

## 5 Návrh softwaru záznamníku dat

Tato kapitola popisuje návrh softwaru *Front-End\_jednotky*, jenž se skládá ze softwaru *WTB\_jednotky* a softwaru *procesorového\_modulu*. Software pro PC zajišťující komunikaci s *Front-End\_jednotkou* a ukládání přenesených dat do souboru je řešen společně se softwarem *procesorového\_modulu*. Software pro PC sloužící k interpretaci a zobrazení přenesených dat je řešen samostatně.

### 5.1 Software WTB jednotky

Software *WTB\_jednotky* vychází ze softwaru jednotky *WTB 2000* [9], hlavně z jeho komponenty „WTB - fyzická vrstva“. K zajištění požadované funkce je nutné daný software částečně modifikovat, částečně doplnit novými funkcemi a částečně odstranit.

Zdrojový kód je psán tak, že části, ve kterých je řešena vazba na konkrétní hardware nebo řešeno konkrétní rozhraní, jsou realizovány v samostatných modulech.

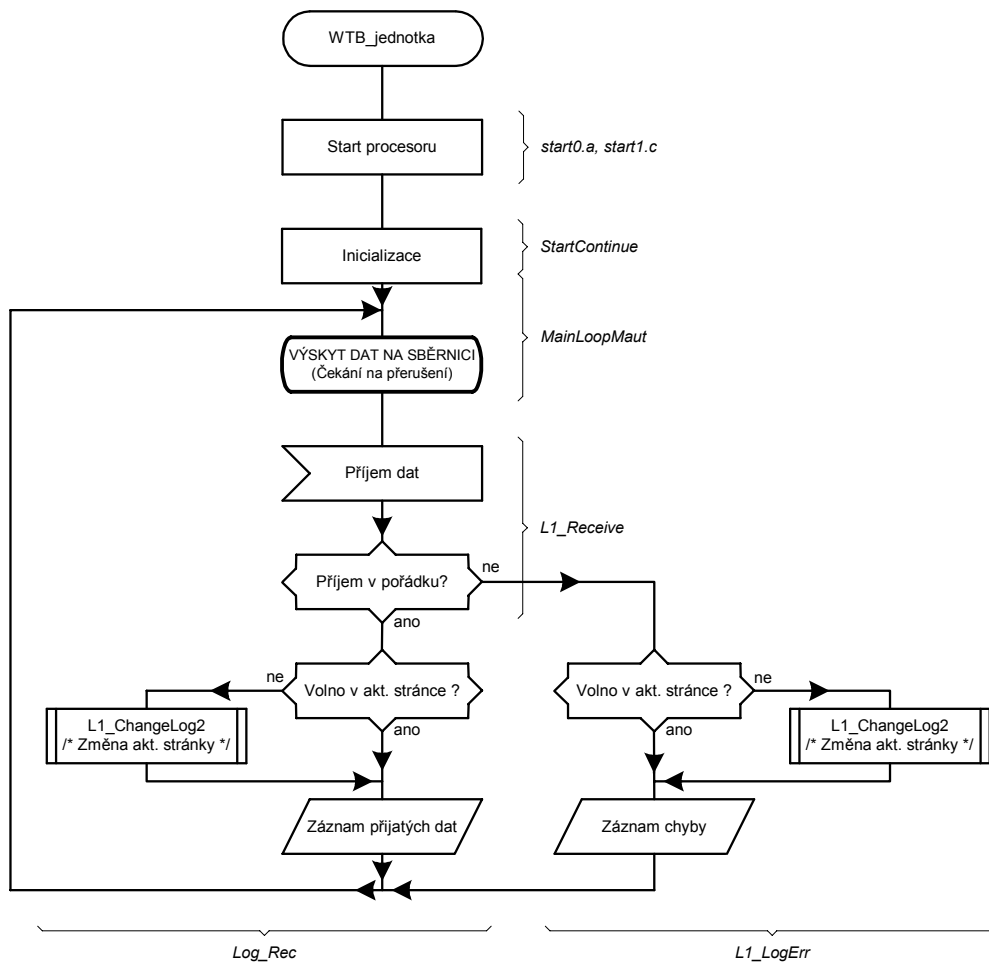
Pro vývoj softwaru *WTB\_jednotky* bylo použito integrované vývojové prostředí *RadiSys Hawk v. 2.3* a překladač *Ultra C v. 2.5*.

Software *WTB\_jednotky* pracuje bez operačního systému.

#### 5.1.1 Algoritmus činnosti softwaru WTB jednotky

Algoritmus činnosti *WTB\_jednotky* znázorňuje vývojový diagram na obr 5.1 níže.

Po nastartování procesoru proběhne inicializace (mimo jiné spuštění UTC časovače, inicializace paměti pro záznam dat, inicializace a test hardware, nastavení parametrů příjmu). Poté následuje čekání na výskyt dat na WTB sběrnici. V případě jejich výskytu, je vyvoláno přerušení, jenž je obslouženo přerušovací rutinou, pomocí které se daná data přijmou a uloží do paměti. Pokud se příjem nezdaří, do paměti se uloží záznam o vzniklé chybě. Před zápisem do paměti se kontroluje, zda je v aktuální stránce záznamů ještě volno, případně dojde k přepnutí mezi aktivní a neaktivní stránkou. Analogicky jsou též zaznamenávány informace o vnitřních událostech (to diagram na obr. 5.1 nezobrazuje).



Obr. 5.1 Algoritmus chování WTB jednotky

## 5.1.2 Popis jednotlivých částí softwaru WTB jednotky

### 5.1.2.1 Chování WTB jednotky vzhledem k WTB sběrnici

WTB\_jednotka přijímá všechna data, která se na WTB sběrnici vyskytují (nastaveno ve funkci *MainLoopMaut*). Nahrazení funkcí *L1\_Send* a *pL2\_Receive* funkcemi prázdnými zajišťuje, že WTB\_jednotka nevysílá žádná data.

### 5.1.2.2 Přijem dat - zpracování přerušení

Výskyt dat na WTB sběrnici vyvolá přerušení od odpovídajícího sériového kanálu, jenž je zpracováno funkcí *L1\_Receive*. Odtud se poté volá odpovídající funkce či makro provádějící vlastní zápis do sdílené paměti (*Log\_Rec*, *L1\_Log* nebo *L1\_LogErr*). Funkce *L1\_Receive* je řešena tak, že jsou zaznamenávána data z obou kanálů.

### 5.1.2.3 Struktura oblasti záznamů dat

Paměť, do které se ukládají záznamy, je tvořena dvěma stránkami. Každá z nich obsahuje 1024 záznamů (zvoleno, ověřeno níže). Oblast záznamů dat má následující strukturu:

- *BASE\_TABLE* (základní struktura) obsahuje mimo jiné ukazatele na počátek záhlaví jednotlivých stránek záznamů (*LOG\_PG\_HD*) a informaci o tom, jaká stránka je v daný čas aktivní.
- *LOG\_PG\_HD* obsahuje mimo jiné ukazatele na počátek a konec dané stránky, ukazatel na následující záznam a příznak, zda je daná stránka v daný čas uvolněna ke čtení.
- *LOG\_ITEM* (jednotlivé záznamy)

Jednotlivé záznamy jsou pevné délky (144 B). Struktura jednoho záznamu (*LOG\_ITEM*) je uvedena v následující tabulce.

<i>LOG_ITEM</i>		
<b>Symbol</b>	<b>Velikost</b>	<b>Význam</b>
<i>Usec</i>	4 B	čas přijetí rámce - strojový čas [ $\mu$ s]
<i>Utc</i>	4 B	čas přijetí rámce - UTC čas [s]
<i>Ctrl</i>	1 B	stavová informace
<i>Code</i>	1 B	stavová informace
<i>Dat</i>	132 B	<i>HDLC_Data</i> / informace specifické pro danou událost/chybu
<i>Spare1</i>	1 B	Nevyužito (využito pouze pro pomocnou informaci)
<i>Spare2</i>	1 B	Nevyužito

Tabulka. 5.1 Struktura jednotlivých záznamů (*LOG\_ITEM*)

Dvojice *Ctrl* a *Code* bytů určuje o jaký typ záznamu se jedná a další jeho vlastnosti (případně také k jakému kanálu se daný záznam vztahuje).

Položky *Spare1* a *Spare2* jsou prázdné. Jejich účelem je pouze upravení velikosti struktury *LOG\_ITEM* tak, aby byla dělitelná čtyřmi. (Případně je lze využít v budoucnu pro přenos dalších informací.)



Jednotlivé stránky záznamů, jejich záhlaví (*LOG\_PG\_HD*) a *BASE\_TABLE* jsou umístěny ve sdílené paměti na konstantních adresách.

Z výše uvedeného vyplývá, že velikost jedné stránky, při zanedbání jejího záhlaví, je 147 456 B. Tedy obě stránky zaujmou v paměti méně než 300 kB. To je vyhovující, protože zaručená velikost volné paměti je 512 kB. Nicméně, pro jistotu je při inicializaci stránek testováno, zda je velikost volné paměti dostatečná.

#### 5.1.2.4 Přepínání stránek záznamů

Přepínání stránek záznamů je implementováno ve funkci *L1\_ChangeLog2*. Přístup procesorového modulu do sdílené paměti je řízen příznakem *ReadReady\_fl* umístěným v záhlaví odpovídající stránky záznamů (*LOG\_PG\_HD*). Příznak je nastavován *WTB\_jednotkou* na „1“ poté, co je daná stránka zaplněna a nastavena jako neaktivní. Příznak je nastavován na „0“, bezprostředně předtím, než je daná stránka znovu aktivována pro zápis.

Navržený software neobsahuje zpětnou kontrolu, zda byla daná stránka přečtena předtím, než se do ní začne opět zapisovat. Předpokládá se, že procesorový modul načte stránku z paměti a pošle ji do PC rychleji, než *WTB\_jednotka* stihne druhou stránku zaplnit (5.2.3). Jiné řešení by bylo problematické, protože v případě čekání, až procesorový modul přečte všechna uložená data, by nebylo možné ukládat data nová, což by mělo za následek ztrátu případných nových dat.

#### 5.1.3 Testování softwaru WTB jednotky

Testování softwaru *WTB\_jednotky* bylo provedeno pomocí zabudovaných komunikačních testů v *MAU 2000 - TEST*.

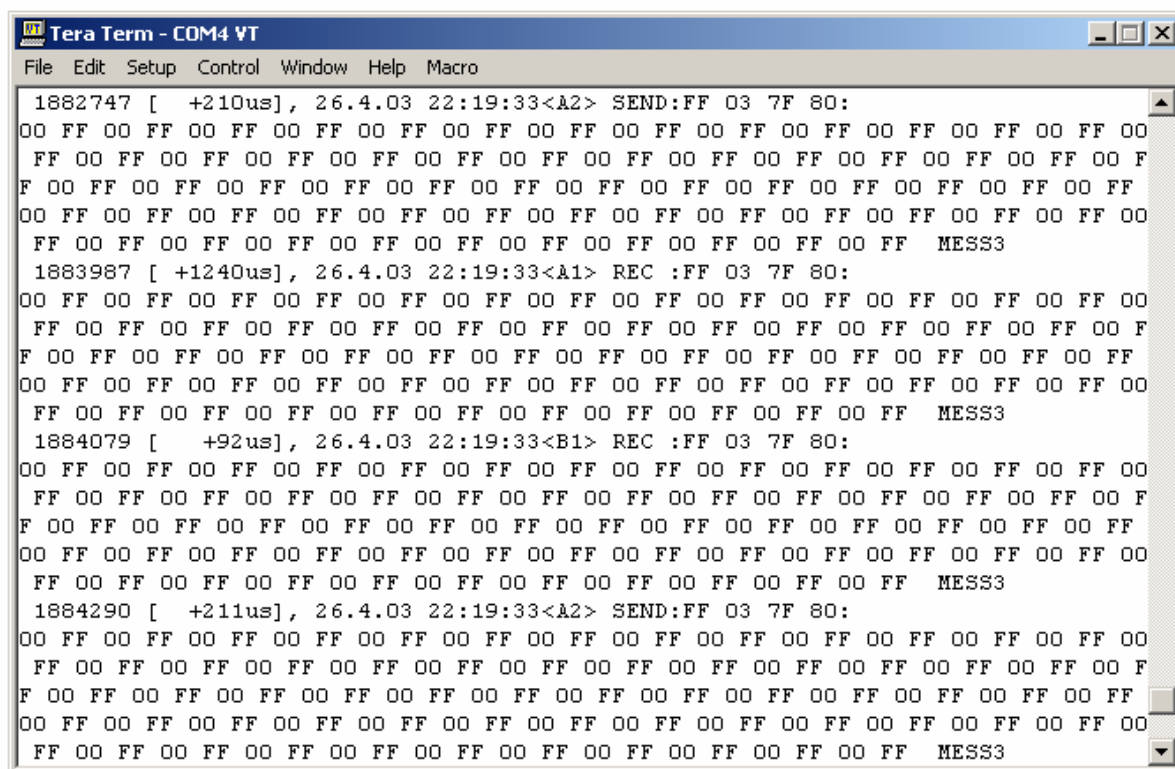
Spojení mezi testovacím PC a *WTB\_jednotkou* bylo provedeno přes sériové rozhraní a komunikace navázána pomocí terminálového programu *TeraTerm Pro v. 2.3*.

##### 5.1.3.1 Testování správnosti ukládání dat do paměti

Vlastní testy spočívaly ve spuštění určitého testu vysílajícího *WTB\_rámce* a následné analýzy zda jsou data ukládána do paměti správným způsobem. To bylo prováděno pomocí

zobrazování stránek záznamů pomocí *MAU 2000 - TEST* a nahlédnutím přímo do sdílené paměti pomocí procesu *debug* spuštěném v *procesorovém\_modulu*. Příklad výpisů stránek záznamů poskytují následující obrázky 5.2 - 5.4.

Poznámka: Možnost vysílání *WTB\_rámce* a funkčnost komponenty *MAU 2000 - TEST* ve *WTB\_jednotce* jsou ponechány pouze pro ladící a testovací účely, tedy jejich trvání je dočasné.

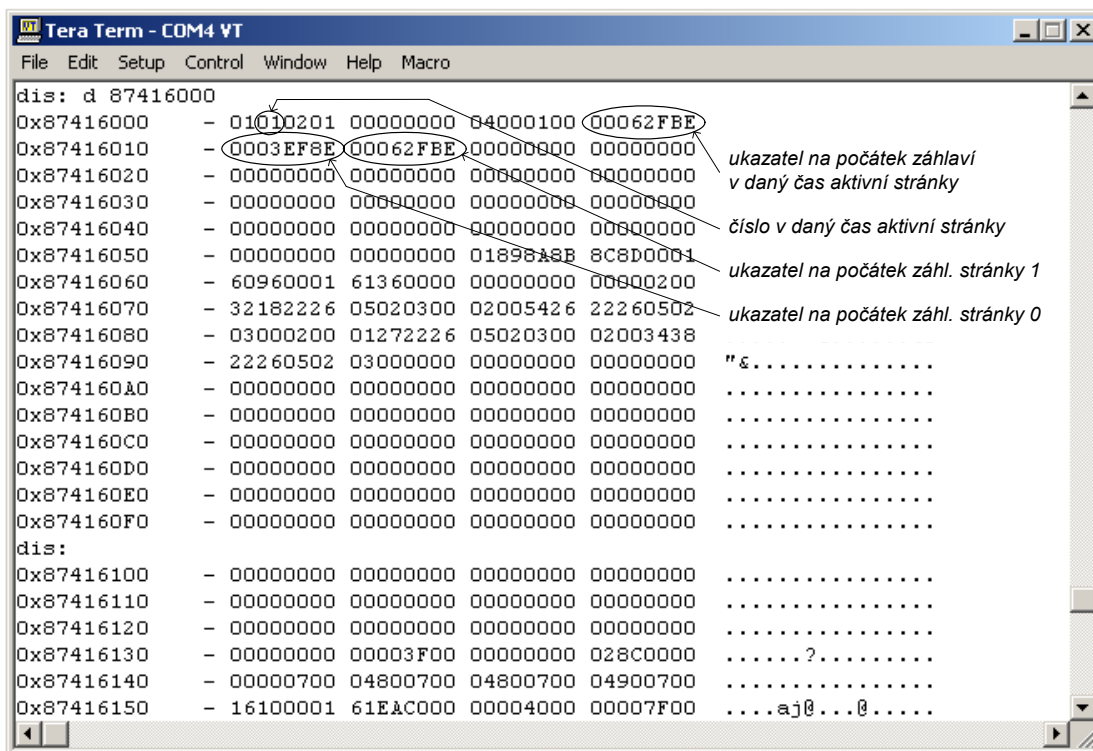


```
Tera Term - COM4 VT
File Edit Setup Control Window Help Macro

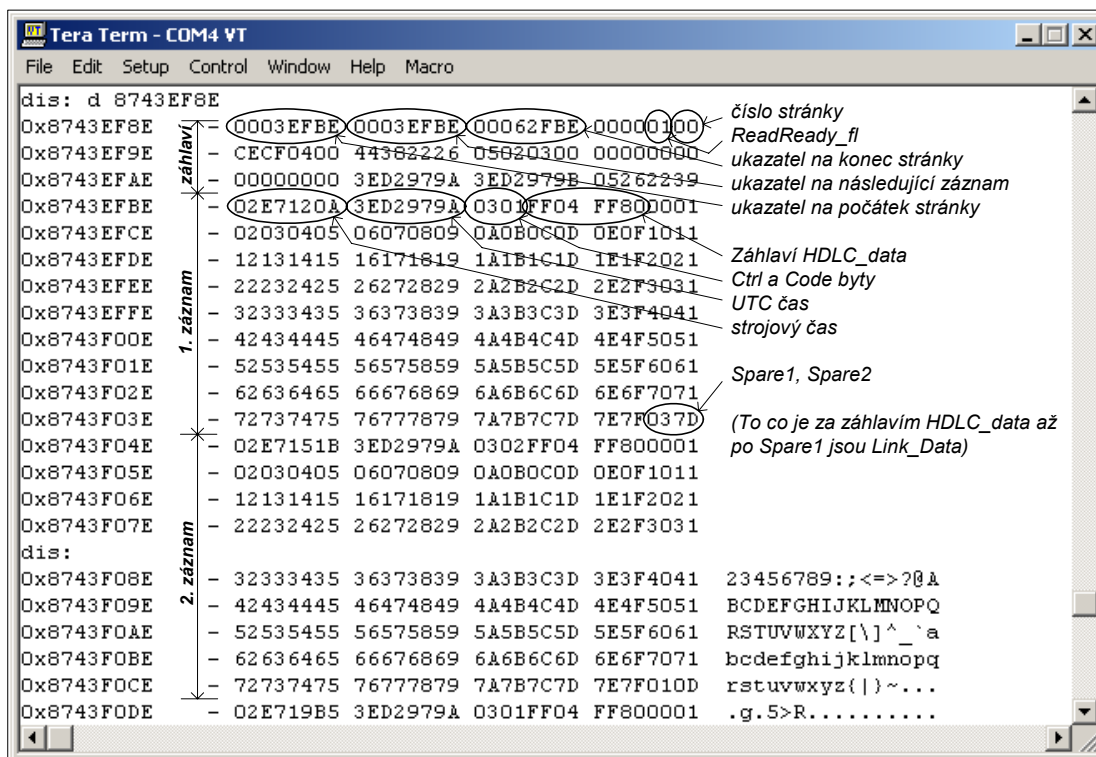
1882747 [ +210us], 26.4.03 22:19:33<A2> SEND:FF 03 7F 80:
00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00
FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 F
F 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00
FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
MESS3
1883987 [ +1240us], 26.4.03 22:19:33<A1> REC :FF 03 7F 80:
00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00
FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 F
F 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00
FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
MESS3
1884079 [ +92us], 26.4.03 22:19:33<B1> REC :FF 03 7F 80:
00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00
FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 F
F 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00
FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
MESS3
1884290 [ +211us], 26.4.03 22:19:33<A2> SEND:FF 03 7F 80:
00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00
FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 F
F 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00
FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
MESS3
```

Obr. 5.2 Příklad výpisu části záznamů pomocí *MAU 2000 - TEST*

Obrázek 5.2 zobrazuje příklad výpisu čtyř záznamů. Postupně se jedná o záznamy vyslání *WTB\_rámce*, jeho příjem z obou kanálů a vyslání dalšího *WTB\_rámce*.



Obr. 5.3 Příklad výpisu paměti - BASE\_TABLE



Obr. 5.4 Příklad výpisu paměti - záhlaví a počátek pole záznamů stránky 0

### 5.1.3.2 Měření času nutného k příjmu WTB rámce

Měření času nutného k příjmu daného *WTB\_rámce* z obou kanálů a provedení odpovídacích záznamů do paměti (funkce *LI\_Receive*) bylo provedeno s následujícími výsledky:

Velikost <i>Link_Data</i> [B]	Doba trvání příjmu [ $\mu$ s]
4	208
32	217
64	230
128	245

Tabulka 5.2 Čas nutný k příjmu *WTB\_rámce* a jeho záznamu do paměti

*WTB\_rámec* má délku 82 - 1323 bitů (trvá 82 - 1323  $\mu$ s). Prodleva mezi jednotlivými *WTB\_rámci* je 64 - 300  $\mu$ s. Tedy minimální teoretická perioda *WTB\_rámců* je 146  $\mu$ s. *WTB\_jednotka* má dvě vstupní vyrovnávací paměti.

Z uvedeného vyplývá, že při použití *záznamníku\_dat* na WTB sběrnici, kde více než dva *WTB\_rámce* jsou posílány s prodlevou menší než 245  $\mu$ s není možné. Ovšem pro WTB sběrnici, kde jsou použity WTB komunikační jednotky *WTB 2000* jsou naměřené parametry vyhovující (viz 4.2.3).

Zvýšení rychlosti příjmu *WTB\_rámce* by bylo možné použitím rychlejšího procesoru případně přepsáním zdrojového kódu funkce *LI\_Receive* v jazyce Asembler.

## 5.2 Software procesorového modulu a software pro PC

V této kapitole je popsán software *procesorového\_modulu* a software pro PC zajišťující komunikaci s *procesorovým\_modulem* resp. s *Front-End\_jednotkou* a ukládání přenesených dat do souboru.

Pro vývoj softwaru *procesorového\_modulu* bylo použito integrované vývojové prostředí *RadiSys Hawk v. 2.3* a překladač *Ultra C v. 2.5*. Software pro PC byl vyvinut v prostředí *Microsoft Visual C++ v. 6.0*.

Software pro PC pracuje pod operačním systémem Windows (jakoukoli 32 bitovou verzí). Software *procesorového\_modulu* pracuje pod operačním systémem OS-9, který je v této kapitole také stručně charakterizován (5.2.5).

Komunikace mezi *procesorovým\_modulem* a PC probíhá po *Ethernet* sběrnici, na které je použit protokol TCP/IP [12]. Vytvořené programy komunikují po této sběrnici pomocí *Socket* rozhraní [13].

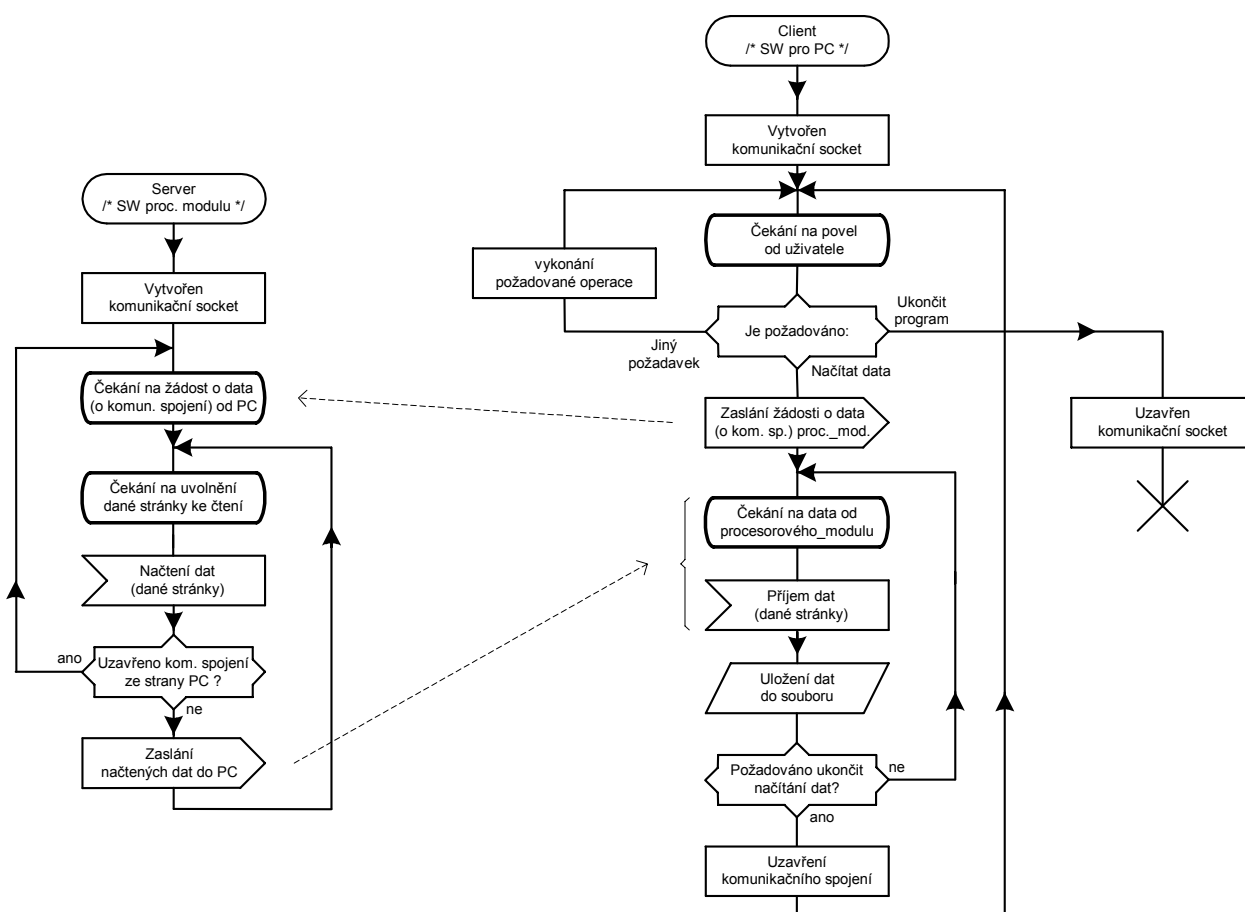
### 5.2.1 Algoritmus činnosti softwaru procesorového modulu a softwaru pro PC

Algoritmus činnosti softwaru *procesorového\_modulu* a s ním souvisejícího softwaru pro PC znázorňuje vývojový diagram na obr. 5.5 níže. Komunikace mezi *procesorovým\_modulem* a PC je typu *Client* (PC) - *Server* (*procesorový\_modul*).

Po spuštění programu *procesorového\_modulu* je vytvořen komunikační *Socket*. Dále se čeká, dokud PC nezažádá o navázání komunikačního spojení a tím o data. V případě obdržení žádosti je vytvořeno komunikační spojení a spuštěn komunikační cyklus popsáný níže. Po spuštění programu *Client* (v PC) je vytvořen komunikační *Socket* a čeká se na povel uživatele. Lze zadat cestu, kam ukládat přijatá data, je možné spustit načítání dat, případně může být program ukončen. Pokud je požadováno načítat data, je vyslána žádost o navázání komunikačního spojení - o data *procesorovému\_modulu*. Následuje čekání na požadovaná data.

Vlastní přenos dat probíhá v cyklech odpovídajících jednotlivým stránkám záznamů. *Procesorový\_modul* čeká na zaplnění dané stránky a její uvolnění ke čtení. Poté jsou záznamy načteny ze sdílené paměti a poslány do PC. Analogicky na straně PC, se daný cyklus skládá z čekání na data od *procesorového\_modulu*, jejich příjmu a uložení do souboru.

Komunikační cyklus (přenos dat) může být ukončen na základě žádosti uživatele. Ta se testuje na konci každého cyklu. Pokud je požadováno přenos dat ukončit, dojde k uzavření komunikačního spojení ze strany PC. To vyvolá chybu při posílání následujících dat a tím i uzavření komunikačního spojení na straně *procesorového\_modulu*. Poté se čeká na další povel od uživatele. Tím se výše popsaný algoritmus znovu opakuje.

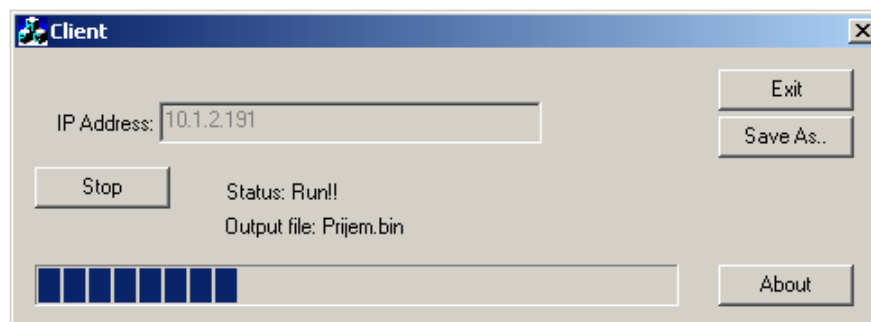


Obr. 5.5 Algoritmus komunikace mezi procesorovým modulem a PC

Popsaný software přenáší záznamy pevné délky (144 B) bez ohledu na délku užitečných dat.

### 5.2.2 Uživatelské rozhraní softwaru pro PC

Grafické rozhraní programu *Client* znázorňuje následující obrázek.



Obr. 5.6 Grafické rozhraní programu *Client*

Funkce jednotlivých tlačítek jsou zřejmé. Stisk tlačítka „Start“ resp. „Stop“ je chápán jako povel k započetí resp. pozastavení přenosu dat. Po stisknutí tlačítka „Save As“ je možné zadat cestu a název souboru kam se mají data ukládat. Stisk tlačítka „Exit“ je interpretován jako žádost o ukončení programu. Před první žádostí o data je nutné zadat IP adresu, kterou již dále není možné měnit. (Nepředpokládá se, že bude požadováno krátce za sebou načítat data z různých *Front-End* jednotek.) Grafické rozhraní dále poskytuje informaci o aktuálním stavu činnosti programu, jménu výstupního souboru a obsahuje ukazatel množství přijatých dat.

### 5.2.3 Časové nároky na komunikaci mezi procesorovým modulem a PC

Velikost jedné stránky záznamů je 144 kB (1,125 Mbitů). Teoretická přenosová rychlost použité *Ethernet* sběrnice (ISO 8803-3) je 10 Mb/s. Odhad reálné rychlosti přenosu uživatelských dat je 8,5 Mb/s. Doba zaplnění jedné stránky záznamů závisí na aktuálním zatížení sběrnice, délce jednotlivých *WTB\_rámců* (82 - 1323  $\mu$ s) a struktuře ukládaných informací (poměru záznamů přijatých rámců a chybových záznamů).

Kalkulace doby zaplnění jedné stránky záznamů a odpovídajících nároku na rychlost jejího vyčtení a přenosu dat do PC, za teoreticky nejnepříznivějších okolností, je uvedena v následující tabulce 5.3.

Velikost <i>Link_Data</i> :	0 B
-----------------------------	-----

Velikost <i>WTB_rámce</i> :		82 bitů
Prodleva mezi <i>WTB_rámci</i> :		64 $\mu$ s
Perioda <i>WTB_rámců</i> :		146 $\mu$ s
jedna chyba na x <i>WTB_rámců</i>	Doba zaplnění jedné str. [ms]	min. přenosová rychlost [Mb/s]
1	49,83	22,57
4	66,45	16,93
1024	74,72	15,06

Tabulka 5.3 Teoretické nejnepříznivější časové poměry

Tato kalkulace ukazuje, že za těchto předpokladů je požadováno přenášet data rychlostí až 22,6 Mb/s, což je nerealizovatelné. Ovšem výše uvažované předpoklady mohou nastat pouze teoreticky. Komunikace na WTB sběrnici se skládá ze sekvence rámců *Master\_Frame* (zpravidla žádná *Link\_Data*) a *Slave\_Frame* (až 128 B *Link\_Data*). Tedy je nereálné, aby délka *Link\_Data* všech *WTB\_rámců* byla nulová. Reálný předpoklad, minimální průměrné délky *Link\_Data* *WTB\_rámců* je 32 B. Kalkulaci časových poměrů za těchto okolností a odpovídajících nároků na rychlost komunikace uvádí tabulka 5.4 níže.

Velikost <i>Link_Data</i> :		32 B
Velikost <i>WTB_rámce</i> :		338 bitů
Prodleva mezi <i>WTB_rámci</i> :		64 $\mu$ s
Perioda <i>WTB_rámců</i> :		402 $\mu$ s
jedna chyba na x <i>WTB_rámců</i>	Doba zaplnění jedné str. [ms]	min. přenosová rychlost [Mb/s]
1	137,2	8,2
4	183,0	6,1
1024	205,7	5,5

Tabulka 5.4 Reálné nejnepříznivější časové poměry

Za těchto předpokladů je požadovaná přenosová rychlost (tabulka 5.4) dosažitelná. Problém by mohl nastat pouze při vysoké průměrné četnosti záznamů detekovaných chyb (více než jedna na jeden *WTB\_rámec*), což je nepravděpodobné.



#### **5.2.4 Testování softwaru procesorového modulu a softwaru pro PC**

Testování softwaru bylo provedeno pomocí vestavěných testů v *MAU 2000 -TEST*. Časově nejnáročnější test, který lze pomocí této komponenty realizovat má za následek zaplnění jedné stránky záznamů za 131 ms. Tomu odpovídá požadavek přenosu dat mezi *procesorovým\_modulem* a PC rychlostí 8,4 Mbit/s.

Jednotlivé testy byly prováděny následujícím způsobem. Pomocí *MAU 2000 -TEST* bylo spuštěno vysílání *WTB\_rámců* v počtu odpovídajícím zaplnění pěti stránek záznamů. Vysílané rámce byly snímány *záznamníkem\_dat*. Poté byla přijatá data v PC analyzována.

Při provádění popsaných testů pro různé nároky na rychlost přenosu dat byla data na přijímací straně vždy v pořádku.

### 5.2.5 Operační systém OS-9

Operační systém OS-9 [14], [15] byl původně vyvinut pro procesory *Motorola 6809*. Dnes je jeho použití širší, běží na platformách *Motorola 68xxx*, *Motorola PPC*, *X86* a dalších. Jedná se o *embedded* operační systém reálného času určený pro aplikace v náročných průmyslových podmínkách.

Operační systém OS-9 má modulární architekturu tvořenou hierarchií systémových procesů a utilit, jádra operačního systému, správce vstupně výstupních zařízení (*I/O Manager*), správce souborů (*File\_Manager*) a ovladačů vstupně výstupních zařízení (*Device\_Driver*). Objekty operačního systému (jádro, ovladače zařízení, procesy a sdílené datové oblasti) mají formu samostatných modulů obsahujících identifikační hlavičku a jejichž integrita je zajištěna CRC kódem. OS-9 dokáže tyto moduly dynamicky spravovat, to znamená moduly odebírat, přidávat nové nebo je měnit. Operační systém OS-9 je založen na procesech, pro které poskytuje *preemptivní multitasking* založený na prioritách procesů. OS-9 umožňuje použití ochrany paměti mezi procesy. Ladění procesů umožňuje křížový *debugger*, který je součástí vývojového prostředí *Hawk*. Na použitém hardwaru (karta VSBC-32 s procesorem *MOTOROLA MC68360*) zabírá OS-9 v minimální konfiguraci 1 MB paměti ROM a využívá 1 MB paměti RAM.

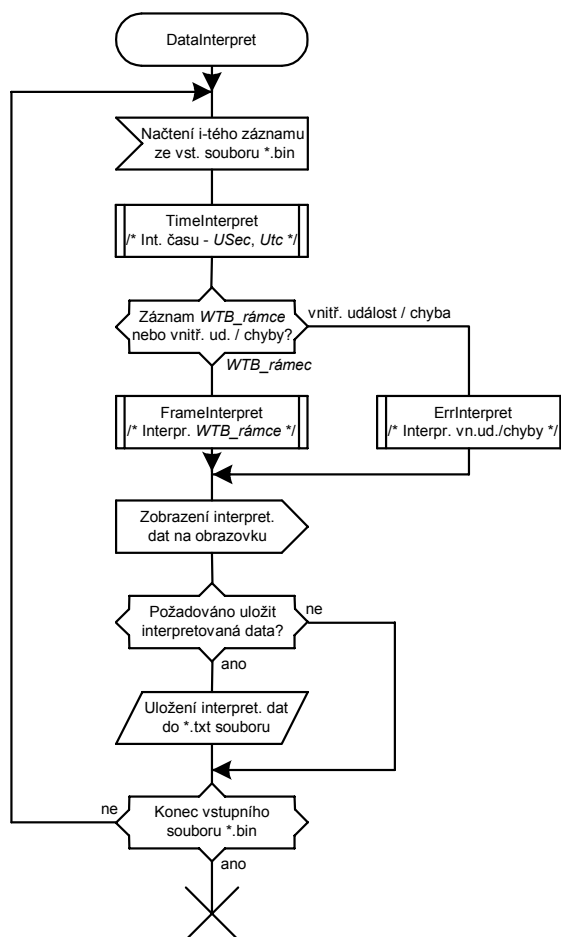
### 5.3 Software pro interpretaci přenesených dat

Tato kapitola popisuje software, který interpretuje a zobrazuje data přenesená do PC. Tento software dále umožňuje uložení interpretované formy dat do textového souboru.

Popisovaný software byl vyvinut v prostředí *Microsoft Visual C++ v. 6.0.* a pracuje pod operačním systémem Windows (jakoukoli 32 bitovou verzí).

#### 5.3.1 Popis činnosti softwaru pro interpretaci přenesených dat

Algoritmus činnosti softwaru pro interpretaci přenesených dat (program *DataInterpret*) znázorňuje následující obrázek.



Obr. 5.7 Algoritmus činnosti programu *DataInterpret*

Činnost programu *DataInterpret* probíhá v jednotlivých cyklech, ve kterých je vždy zpracován jeden záznam (LOG\_ITEM). Postupně dochází k načtení daného záznamu (144 B)

ze vstupního souboru, interpretaci jednotlivých položek záznamu a vypsání interpretované formy dat na obrazovku, případně také jejich uložení do výstupního souboru. Vlastní interpretaci dat zajišťují funkce *TimeInterpret* a *FrameInterpret* resp. *ErrInterpret*.

Interpretovaná forma dat obsahuje strojový čas [ $\mu\text{s}$ ], zpoždění oproti minulému záznamu, reálný čas [s] odvozený z UTC času, informaci o jaký typ záznamu se jedná, k jakému kanálu se daná informace vztahuje a vlastní *HDLC\_Data*, případně informaci specifikující danou vnitřní událost nebo chybu. Jedná se tedy o interpretaci fyzické vrstvy dat. Příklad výpisu zobrazuje obr. 5.8.

Program je řešen jako konzolová aplikace. Cesty k vstupnímu a výstupnímu souboru se zadávají jako parametry při spouštění programu. Pokud je zadána pouze cesta k vstupnímu souboru, je rozuměno, že uživatel si přeje interpretovaná data pouze prohlédnout a nikoli je ukládat do souboru. Příklad zobrazení dat znázorňuje následující obrázek.

```

D:\DataInterpret.exe
1083643153 [ +286us ], 21.05.03 22:44:00 <A2> SEND: FF 05 7F 80:08 08 08 08 08 0
1083644329 [ +1176us ], 21.05.03 22:44:00 <A1> RECU: FF 05 7F 80:08 08 08 08 08 0
1083644420 [ +91us ], 21.05.03 22:44:00 <B1> RECU: FF 05 7F 80:08 08 08 08 08 0
1083644706 [ +286us ], 21.05.03 22:44:00 <A2> SEND: FF 05 7F 80:09 09 09 09 09 0
1083645881 [ +1175us ], 21.05.03 22:44:00 <A1> RECU: FF 05 7F 80:09 09 09 09 09 0
1083645973 [ +92us ], 21.05.03 22:44:00 <B1> RECU: FF 05 7F 80:09 09 09 09 09 0
1083646260 [ +287us ], 21.05.03 22:44:00 <A2> SEND: FF 05 7F 80:0A 0A 0A 0A 0A 0
1083647436 [ +1176us ], 21.05.03 22:44:00 <A1> RECU: FF 05 7F 80:0A 0A 0A 0A 0A 0
1083647527 [ +91us ], 21.05.03 22:44:00 <B1> RECU: FF 05 7F 80:0A 0A 0A 0A 0A 0
1083647813 [ +286us ], 21.05.03 22:44:00 <A2> SEND: FF 05 7F 80:0B 0B 0B 0B 0B 0
1083648988 [ +1175us ], 21.05.03 22:44:00 <A1> RECU: FF 05 7F 80:0B 0B 0B 0B 0B 0
1083649081 [ +93us ], 21.05.03 22:44:00 <B1> RECU: FF 05 7F 80:0B 0B 0B 0B 0B 0
1083649368 [ +287us ], 21.05.03 22:44:00 <A2> SEND: FF 05 7F 80:0C 0C 0C 0C 0C 0
1083650543 [ +1175us ], 21.05.03 22:44:00 <A1> RECU: FF 05 7F 80:0C 0C 0C 0C 0C 0
1083650635 [ +92us ], 21.05.03 22:44:00 <B1> RECU: FF 05 7F 80:0C 0C 0C 0C 0C 0
1083650920 [ +285us ], 21.05.03 22:44:00 <A2> SEND: FF 05 7F 80:0D 0D 0D 0D 0D 0
1083652095 [ +1175us ], 21.05.03 22:44:00 <A1> RECU: FF 05 7F 80:0D 0D 0D 0D 0D 0
1083652187 [ +92us ], 21.05.03 22:44:00 <B1> RECU: FF 05 7F 80:0D 0D 0D 0D 0D 0
1083652475 [ +288us ], 21.05.03 22:44:00 <A2> SEND: FF 05 7F 80:0E 0E 0E 0E 0E 0
1083653651 [ +1176us ], 21.05.03 22:44:00 <A1> RECU: FF 05 7F 80:0E 0E 0E 0E 0E 0
1083653742 [ +91us ], 21.05.03 22:44:00 <B1> RECU: FF 05 7F 80:0E 0E 0E 0E 0E 0
1083654028 [ +286us ], 21.05.03 22:44:00 <A2> SEND: FF 05 7F 80:0F 0F 0F 0F 0F 0
1083655205 [ +1177us ], 21.05.03 22:44:00 <A1> RECU: FF 05 7F 80:0F 0F 0F 0F 0F 0
1083655298 [ +93us ], 21.05.03 22:44:00 <B1> RECU: FF 05 7F 80:0F 0F 0F 0F 0F 0
1083655582 [ +284us ], 21.05.03 22:44:00 <A2> SEND: FF 05 7F 80:10 10 10 10 10 1
1083656757 [ +1175us ], 21.05.03 22:44:00 <A1> RECU: FF 05 7F 80:10 10 10 10 10 1
1083656849 [ +92us ], 21.05.03 22:44:00 <B1> RECU: FF 05 7F 80:10 10 10 10 10 1

```

Obr. 5.8 Uživatelské rozhraní programu *DataInterpet*

### **5.3.2 Testování softwaru pro interpretaci přenesených dat**

Software pro interpretaci přenesených dat byl testován na datech získaných při provádění zabudovaných testů v *MAU 2000 -TEST*. Výstupy programu *DataInterpet* odpovídaly charakteru prováděných testů a shodovaly se s výpisy v *MAU 2000 -TEST*.

## 6 Závěrečné zhodnocení výsledků

V rámci této diplomové práce byl realizován software *Front-End\_jednotky\_záznamníku\_dat* a software pro PC zajišťující komunikaci s *Front-End\_jednotkou* a ukládání přenesených dat do souboru. Dále byl realizován software pro interpretaci fyzické vrstvy přenesených dat.

Provedené testy prokázaly požadovanou funkci zařízení. Ovšem použití realizovaného *záznamníku\_dat* je omezené kvůli dlouhé době potřebné k příjmu daného *WTB\_rámce* a jeho uložení do paměti. Tato hodnota byla naměřena 208 - 245  $\mu$ s. Tato vlastnost je způsobena nízkou rychlostí použitého procesoru *WTB\_jednotky*. (Teoreticky je minimální perioda *WTB\_rámce* 146  $\mu$ s.) Pro WTB sběrnici, kde jsou použity WTB komunikační jednotky *WTB 2000* jsou parametry realizovaného *záznamníku\_dat* vyhovující. Nicméně, před jeho uvedením do provozu by bylo vhodné provést některé úpravy navržené níže.

Snížení nároků na rychlost přenosu dat mezi *procesorovým\_modulem* a PC lze docílit úpravou softwaru tak, aby nebylo přenášeno vždy celých 144 B záznamu, ale pouze užitečná data. To by si vyžadovalo přidat do struktury jednoho záznamu další položku definující délku užitečných dat daného záznamu. *Procesorový\_modul* by posílal do PC pouze užitečná data a při interpretaci dat by se jednotlivé rámce četli na základě informace v položce určující užitečnou délku dat. Je zřejmé, že tímto způsobem by se také výrazně snížil objem ukládaných dat (myšleno binárních souborů).

Přestože při provedených testech nedocházelo ke ztrátě dat, obecně ji nelze vyloučit. Tedy bylo by vhodné software doplnit o kontrolu integrity přenášených dat. To lze zajistit číslováním jednotlivých stránek záznamů, přenášet tuto informaci spolu s daty dané stránky a při interpretaci přenesených dat kontrolovat posloupnost čísel stránek, případně upozornit na výpadek části dat.

Software pro interpretaci přenesených dat lze vylepšit o funkci, která v datech rozpozná záhlaví linkové, síťové či transportní vrstvy a prezentuje je srozumitelnějším způsobem.

Realizaci výše navržených úprav bych se chtěl dále věnovat.

## 7 Seznam použité literatury a dalších zdrojů

- [1] Nenutil Dobromil: **Moderní řešení řídicích systémů vlakových souprav**, Sborník přednášek z doprovodné konference Pragoregula 2003, Praha 2003, str. 25-28
- [2] International Electrotechnical Commission, **Train Communication Network**, International Standard IEC 61375-1, 1999
- [3] Internetové stránky projektu **ROSIN - část popisující TCN**, <http://www.labs.it/rosin/tncorso/tutindex.htm>, 15.10.2003
- [4] International Telecommunication Union, **Specification and description language (SDL)**, International Standard Z.100, 1999
- [5] **SDL-RT**, standard V1.2, 2002
- [6] Internetové stránky společnosti **Telelogic - popis SDL**, <http://www.telelogic.com/products/tau/languages/sdl.cfm>, 30.4.2003
- [7] Internetové stránky organizace **SDL Forum Society**, <http://www.sdl-forum.org>, 30.4.2003
- [8] Internetové stránky **SDL-RT**, <http://www.sdl-rt.org>, 30.4.2003
- [9] **Software projektu WTB komunikační jednotky WTB 2000**, SW 0228A, UniControls a.s., 1999
- [10] Nenutil Dobromil: **Wire Train Bus** dle dok. 9/413/CDV - **zadávací specifikace**, ZS053, Praha 1999, UniControls a.s.
- [11] Pavel Milenovský, **WTB 2000 - Fyzická vrstva L1 - Popis návrhu softwaru**, SD0228B, Praha 1999, UniControls a.s.
- [12] Libor Dostálek, Alena Kabelová: **Velký průvodce protokoly TCP/IP a systémem DNS**, Praha 2000, Computer Press, str. 46-48, 103-112, 207-230
- [13] Microware Documentation - documentation set revision 3.2.4, RadiSys Corp., kapitola Networking and Communications - **Using LAN Communications Pak**, str. 226-252
- [14] Microware Documentation - documentation set revision 3.2.4, RadiSys Corp., kapitola OS-9 Operating System - **OS-9 Technical Manual**
- [15] Microware, **OS-9**, Propagační materiál OS-9/1044.0027

## Přílohy I

### A.1 Termíny a zkratky

Následující tabulka poskytuje vysvětlení vybraných termínů a zkratk použitých v této práci.

(Anglický) termín	Zkratka	Český překlad nebo vysvětlení termínu
<i>Train Communication Network</i>	TCN	Vlaková komunikační síť (standard IEC 61375)
Interoperabilita	-	Schopnost zařízení komunikovat a spolupracovat s jiným zařízením
<i>Multifunctional Vehicle Bus</i>	MVB	Multifunkční vozová sběrnice (komunikační protokol definovaný TCN)
<i>Wire Train Bus</i>	WTB	Vlaková sběrnice (komunikační protokol definovaný TCN)
<i>Train Bus Gateway (Node, uzel)</i>	TGW (-, -)	Komunikační uzel, zpravidla propojuje vozovou a vlakovou sběrnici. (Z hlediska WTB se jedná o <i>Node</i> či <i>uzel</i> .)
<i>Real-Time Protocols</i>	RTP	Protokoly reálného času - definují vyšší vrstvy TCN sítě.
<i>Network Management</i>	TNM	Řízení TCN sítě
<i>Process Data</i>	-	Procesní data
<i>Message Data</i>	-	Zprávy
<i>Supervisory Data</i>	-	Řídicí a dohlížecí data
<i>Master</i>	-	Správce sběrnice
<i>Bus Administrator</i>	-	Několik zařízení na MVB, které jsou schopny stát se správcem sběrnice.
<i>Slave</i>	-	Zařízení/ <i>uzel</i> na sběrnici, který se řídí pokyny správce sběrnice.
<i>Point-to-Point</i>	-	Způsob distribuce dat - „z bodu do bodu“ - adresována pouze jednomu zařízení
<i>Broadcast</i>	-	Způsob distribuce dat - adresována všem ostatním zařízením (zdrojově orientována)
<i>Datasets</i>	-	Soubor více procesních proměnných
-	DFT	Diskrétní Fourierova transformace
<i>Cyclic Redundancy Check</i>	CRC	Zabezpečovací cyklický kód
<i>High Data Link Control</i>	HDLC	Protokol, jehož definici formátu rámce TCN síť využívá.
<i>záznamník dat</i>	-	Zařízení pro záznam dat z WTB sběrnice
<i>Front-End jednotka</i>	-	Předsazená jednotka (myšleno vůči PC), část <i>záznamníku dat</i> navrhovaná v této práci
<i>WTB jednotka</i>	-	Část <i>Front-End jednotky</i> navrhované v této práci
<i>WTB 2000</i>	-	Projekt WTB komunikační jednotky realizovaný firmou <i>UniControls a.s.</i>
<i>Universal Coordinated Time</i>	UTC	Počet vteřin od 1.1.1970 dle greenwichského času
<i>embedded</i> operační systém	-	Operační systém s min. nároky na periferní zařízení (např. nepotřebuje zařízení se souborovým systémem)
<i>debugger</i>	-	ladicí program

Tabulka A.1 Termíny a zkratky



## A.2 Shrnutí vlastností Train Communication Network

Následující tabulka shrnuje vlastnosti TCN sítě. (Jedná se o překlad části tabulky A.1 v [2].)

Charakteristika	Multifunction Vehicles Bus	Wire Train Bus	
Struktura	Pevná struktura i pevné adresy jednotlivých zařízení	Proměnlivá, schopná vlastní konfigurace, jestliže se změní kompozice	
Médium	Kroucená dvojlinka (až 20 m, 32 zařízení), Stíněná kroucená dvojlinka (až 200 m, 32 zařízení), Optické vlákno (až 2000 m, 2 zařízení)	Stíněná kroucená dvojlinka (až 860 m, 32 uzlů, 22 vozů)	
Fyzická redundance	Zdvojení média	Zdvojení média	
Kódování	<i>Manchester</i>	Invertované <i>Manchester s preamble</i>	
Přenosová rychlost	1,5 Mbit/s	1 Mbit/s	
Adresace	12-bitová pro: - <i>Process Data</i> (logická adresa), - <i>Message Data</i> (fyzická adresa)	8-bitová adresa pro: - <i>Process Data</i> (jeden pro jeden uzel), - <i>Message Data</i>	
Fyzická adresace	<i>Point-to-Point, Broadcast</i>	<i>Point-to-Point, Broadcast</i>	
Velikost užitečných dat	16, 32, 64, 128 nebo 256 bitů	4 - 132 B	
Integrita dat	Přenášené rámce obsahují <i>Check Sequence</i> (kontrolní sekvence) a <i>Frame Size Check</i> (kontrola velikosti rámce), použité kódování má Hammingovu vzdálenost 8.	FCS-16 ( <i>Frame Check Sequence</i> - kontrolní sekvence rámce) pro každý rámec, kontrola velikosti rámce, použité kódování má Hammingovu vzdálenost 4.	
Přístup na sběrnici	Řízen <i>Master</i> zařízením	Řízen <i>Master uzlem</i>	
Předávání řízení sběrnice	Řízení sběrnice je cyklicky předáváno mezi několika <i>Bus Administrator</i> zařízeními.	Každý uzel se může stát <i>Master uzlem</i> (záleží na typu konfigurace).	
Redundance <i>Master uzlu</i>	Automatické předávání správy sběrnice	Řízení sběrnice se předává dalšímu uzlu po inauguraci.	
Služby linkové vrstvy	<i>Process Data</i>	Distribuce periodická	<i>Broadcast</i> šířen <i>Dataset</i>
	<i>Message Data</i>	Distribuce na vyžádání	<i>Point-to-Point</i> nebo <i>Broadcast</i> šířena data
	<i>Supervisory Data</i>	Distribuce periodická / na vyžádání	Data pro <i>Network Management</i>

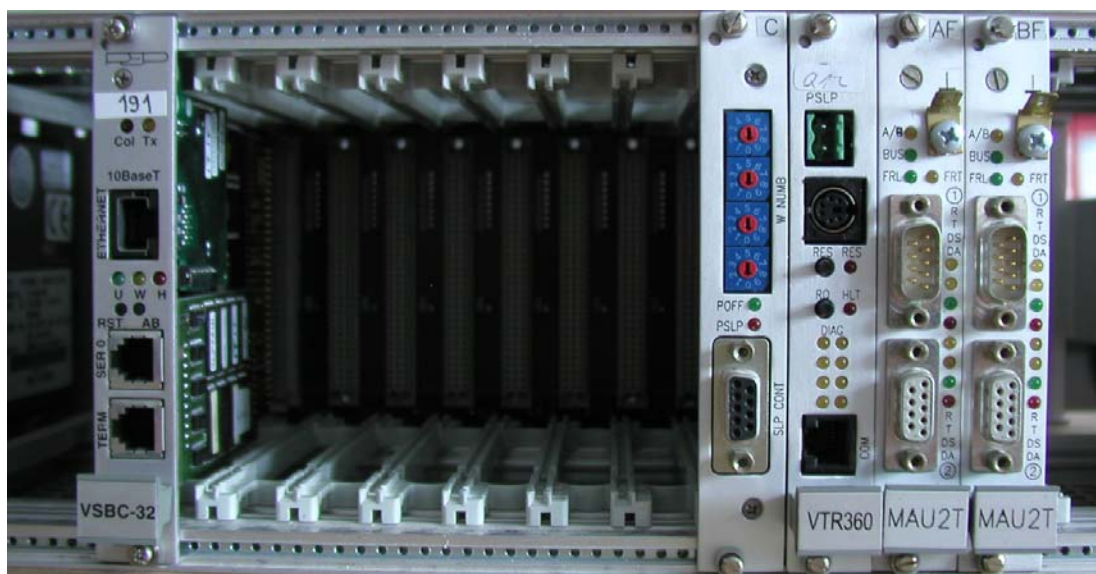
Tabulka A.2 Shrnutí vlastností Train Communication Network

Poznámka: Při *Manchester* kódování je logická „1“ reprezentována změnou úrovně signálu uprostřed časového úseku odpovídajícího jednomu bitu z horní na dolní a logická „0“ opačně, z dolní na horní úroveň. Při invertovaném *Manchester* kódování jsou logická „1“ a „0“ reprezentovány opačně.

### A.3 Fotografie



Obr. A.1 Pracoviště



Obr. A.2 Hardware Front-End jednotky

## **B Přílohy II - CD disk**

Na přiloženém CD disku se nachází tato data:

- zdrojové kódy a překlady programů popsaných v kapitolách 5.2 a 5.3 (software *procesorového modulu* a software pro PC),
- elektronická verze textu této práce.

Pro uvedení kompletního softwaru *WTB\_jednotky* na přiloženém CD disku nebyl získán souhlas od firmy *UniControls a.s.*