ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

fakulta elektrotechnická - katedra řídicí techniky

Diplomová práce

Modul pro průmyslovou komunikaci

Přemysl Šůcha

leden 2003

ČVUT Praha

Anotace

Diplomová práce popisuje návrh univerzálního komunikačního rozhraní mezi Profibus DP/PA a RS232. To je založeno na zákaznických obvodech DPC31 a SIM1, které jsou speciálně určeny pro zařízení Profibus slave. Použitelnost modulu je ukázána na reálné aplikaci rozhraní mezi sběrnicí Profibus DP/PA a kontrolérem stejnosměrných bezkartáčových motorů.

Teoretická část začíná úvodem do problematiky prostředí s nebezpečím výbuchu. Důraz je kladen na Profibus PA a jiskrovou bezpečnost. Dále jsou vysvětleny základní pojmy z oblasti Profibusu. Vlastní návrh popisuje vývoj jak hardwarové tak softwarové části řešeného problému.

Pro lepší ilustraci je práce doplněna přílohami se schématy, detailním popisem implementovaného profilu, výpisem konfiguračního GSD souboru a ukázkami provedených měření osciloskopem.

Annotation

The diploma thesis describes a design of universal communication module between Profibus DP/PA and RS232. Design is based on ASIC components DPC31 and SIM1 that are specially designed for Profibus slave devices. Utility of this module is shown on real application - interface between Profibus DP/PA and controller of brushless DC motors.

Theoretic part begins by introduction into hazardous areas. Problem of Profibus PA and intrinsically safe systems is emphasised. Further the basis of Profibus is explained. Proper design explains development of hardware and software.

Nine appendixes with schematics, detail description of implemented profile, list of configuration GSD file and some samples of measurements with oscilloscope extend description.

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

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

V Praze dne ……. …….

podpis

Poděkování

Chci vyslovit svou úctu a uznání všem, kdo mi přímo či nepřímo pomáhali při vzniku této práce. Zejména děkuji Ing. Zdeňku Hanzálkovi, Dr., vedoucímu mé diplomové práce a Ing. Pavlu Burgetovi. Dále pak Ing. Petru Smolíkovi, který se mnou vždy ochotně konzultoval mou práci a pomáhal mi při vývoji praktické části.

Velký dík patří rodičům, kteří mi v průběhu studia poskytli zázemí. Speciální poděkování patří Ing. Miloslavu Žižkovi, trpělivému a laskavému učiteli, který je mým velkým vzorem.

Obsah

1. Úvod ... 8
2. Elektrická zařízení pro výbušnou plynnou atmosféru ... 9
2.1 Úvod ... 9
2.2 Termíny a definice ... 9
2.3 Členění prostorů a zařízení z hlediska nebezpečí výbuchu ... 10
2.4 Typy ochran ... 12
2.5 Jiskrová bezpečnost (Intrinsic safety) ... 13
2.6 Výběr zařízení podle skupiny výbušnosti plynů ... 14
2.7 Značení ... 14
3. Profibus ... 15
3.1 Obecný popis ... 15
3.2 Fyzická vrstva ... 17
3.2.1 Profibus DP ... 17
3.2.2 Profibus PA ... 18
3.2.2.1 Úvod ... 18
3.2.2.2 Profibus PA a jiskrová bezpečnost ... 19
3.2.2.3 Napájecí zdroje ... 20
3.2.2.4 Zařízení Profibus PA ... 22
3.3 Spojová vrstva ... 23
3.3.1 Obecný formát telegramů ... 23
3.3.2 Chování zařízení DP/PA slave ... 25
3.3.3 Některé telegramy podrobněji ... 26
3.3.3.1 Změna adresy stanice slave ... 26
3.3.3.2 Odpověď na Slave_Diag ... 27
3.3.3.3 Parametrizační telegram ... 28
3.3.3.4 Konfigurační telegram ... 29
3.3.3.5 Globální řízení ... 30
3.3.4 Konfigurace systému Profibus DP/PA ... 31
4. Profily ... 32
4.1 Úvod ... 32
4.1.1 Model a základní myšlenka zařízení Profibus-PA ... 32
4.1.2 Device manager ... 33
4.1.3 Propojení mezi jednotlivými bloky ... 34
4.1.4 Mapování profilu na Profibus-DP ... 35
4.2 Parametry objektů ... 36
4.2.1 Rozdělení parametrů ... 36
4.2.2 Mód bloku ... 37
4.3 Aplikace profilu ... 37
4.3.1 Funkční blok ... 38
4.3.1 Převodní blok ... 41
5. Návrh hardwaru ... 42
5.1 Profibus Development Board ... 42
5.1.1 Formulace požadavků ... 42
5.1.2 Procesorová část ... 42

5.1.3 Rozhraní ... 45
5.2 ProfiProg ... 47
5.3 Realizace a Oživení ... 48
5.3.1 Návrh plošného spoje a součástková základna ... 48
5.3.1 Oživení ... 48
5.3.1.1 ProfiProg ... 48
5.3.1.2 PDB ... 49
5.4 Závěr ... 52
6. Vývoj softwaru ... 53
6.1 ProfiProg ... 53
6.1.1 ProfiProg – programátor ... 53
6.1.1 ProfiProg – Rozhraní na PC (Comm) ... 55
6.2 PDB ... 55
6.2.1 V1SL ... 55
6.2.2 Ověření funkce V1SL a PDB ... 58
6.2.2 Implementace profilu kontroléru ... 59
6.2.2.1 Display a sériový kanál ... 59
6.2.2.2 Implementace profilu do vzorové aplikace ... 59
6.2.2.3 Změna konfigurace vzorové aplikace ... 61
6.2.3 Rozhraní PDB a TG_v3.01 ... 66
6.2.3.1 Komunikace na straně PDB ... 66
6.2.3.2 Komunikace na straně TG_v3.01 ... 68
6.3 Závěr ... 69
7. Závěr ... 70
8. Použitá literatura ... 71

Přílohy

I. Schéma Profibus Development Board
II. Schéma programátoru ProfiProg
III. Profil – Řízení polohy
IV. Protokol komunikace s TG_v3.01
V. GSD soubor
VI. Inicializace zařízení PDB
VII. Měření
VIII. Program STEP7
IX. Fotografie

1. Úvod

Cílem mé diplomové práce bylo navrhnout rozhraní mezi Profibus DP/PA a RS232. Myšlenka tohoto zařízení vznikla na základě konkrétních požadavků výrobců průmyslových zařízení. Běžně vyvíjená zařízení neumožňují připojení na Profibus DP či PA a úprava stávajícího hardwaru je buď obtížná či finančně náročná. Naproti tomu jsou velice často vybaveny rozhraními jako je RS232, RS485 či jiné. Asi nejčastější je rozhraní RS232. To je způsobeno tím, že drtivá většina počítačů je vybavena tímto rozhraním, které se za dobu své existence stalo velice oblíbeným.

Práce je tématicky rozčleněna do dvou částí. První se zabývá teoretickým základem, který je nezbytný k seznámení se s daným tématem. Začíná úvodem do problematiky elektrických zařízení v prostředí s nebezpečím výbuchu, kde je používán Profibus PA. Zbytek pak tvoří popis Profibusu jak je definován normami [1], [2] a [3].

V druhé části je popsán návrh zařízení (univerzálního modulu), které bylo pracovně nazváno PDB (Profibus Development Board). Je rozdělena na dvě kapitoly. První se zabývá návrhem hardwarové stránky a druhá softwarové. Je zde také uvedena konkrétní aplikace PDB jako rozhraní mezi sběrnicí Profibus DP/PA a kontrolérem bezkartáčových motorů TG_v3.01.

Práce je doplněna devíti přílohami, které dokumentují vývoj zařízení. Věřím, že má práce bude přínosem a umožní další vývoj zařízení pro průmyslovou sběrnici Profibus.

2. Elektrická zařízení pro výbušnou plynnou atmosféru

2.1 Úvod

Je-li zařízení instalováno v prostoru, kde může být v atmosféře přítomna nebezpečná koncentrace a nebezpečné množství hořlavých plynů, par, mlhy hořlavých vláken nebo prachů, musí být podle evropské normy EN 60079 použita ochranná opatření pro snížení pravděpodobnosti výbuchu v důsledku iniciace oblouky, jiskrami nebo horkými povrchy, vznikající buď v normálním provozu nebo za stanovených poruchových podmínek. Tato norma je evropskou verzí české normy ČSN EN 60079 z března 1999 a ruší původní normu ČSN 33 2320 Elektrotechnické předpisy – Předpisy pro elektrická zařízení v místech s nebezpečím výbuchu hořlavých plynů a par z února 1996. Zde došlo k úplnému přepracování normy, změně postupů pro zařazování prostorů s nebezpečím výbuchu a nově byli doplněny požadavky pro instalaci jednotlivých typů ochrany proti výbuchu.

2.2 Termíny a definice

prostor s nebezpečím výbuchu (hazardous area) – prostor ve kterém je nebo může být přítomna výbušná plynná atmosféra v takovém množství, že jsou nutná speciální opatření pro konstrukci, instalaci a používání elektrických zařízení.

prostor bez nebezpečí výbuchu (non-hazardous area) – prostor, ve kterém se neočekává výskyt výbušné plynné atmosféry v takovém množství, aby byla nutná speciální opatření pro konstrukci, instalaci a použití elektrických zařízení v těchto prostorech.

maximální povrchová teplota (maximum surface temperature) – nejvyšší teplota, která vznikne za provozu v nejnepříznivějších podmínkách (avšak v rámci tolerancí) na kterékoliv části nebo povrchu elektrického zařízení, která by mohla způsobit vznícení okolní výbušné atmosféry.

skupina nevýbušného elektrického zařízení (group of an electrical apparatus for explosive atmosphere) – klasifikace zařízení ve vztahu k prostředí, ve kterém bude zařízení používáno.

typy ochrany (type of protection) – zvláštní opatření (prostředky), použité pro konstrukci elektrického zařízení za účelem vyloučení iniciace okolní výbušné atmosféry tímto zařízením.

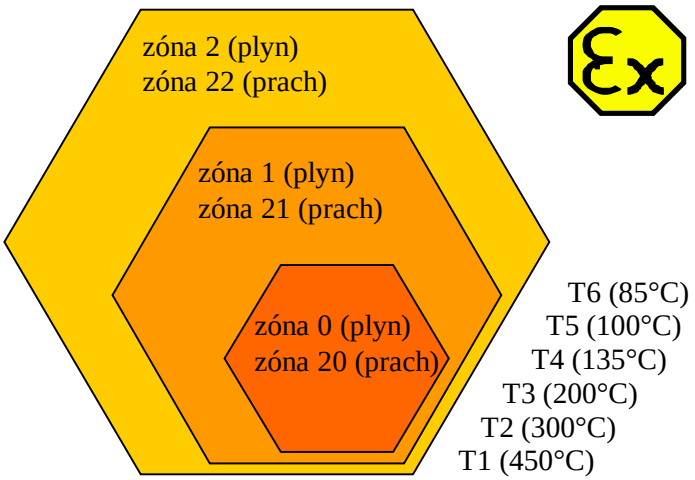
Jiskrově bezpečné zařízení (intrinsically safe apparatus) – elektrické zařízení, ve kterém jsou všechny obvody jiskrově bezpečné.

CENELEC (Europen Committe for Electotechnical Standardisation) – Evropská komise pro normalizaci v elektrotechnice.

IEC (International Elektrotechnical Commision) – Mezinárodní normalizační komise.

2.3 Členění prostorů a zařízení z hlediska nebezpečí výbuchu

Podle pravděpodobnosti výskytu nebezpečné koncentrace v uzavřeném prostoru a podle doby jejího trvání rozeznáváme tři zóny a prostory (viz obr. 2-1)



Obr. 2-1 Členění prostorů z hlediska nebezpečí výbuchu a teplotní třídy (viz tabulka 2-4)

Zóna 0 – prostor, ve kterém je výbušná plynná atmosféra přítomna trvale nebo po dlouhé období. (> 1000h / rok)

Zóna 1 – prostor, ve kterém může vzniknout výbušná plynná atmosféra za normálního provozu. (> 10h ≤ 1000h / rok)

Zóna 2 – prostor, ve kterém výbušná plynná atmosféra může za normálního provozu vzniknout nebo může vzniknout pouze na krátké období. (> 0h ≤ 10h / rok)

Zóna 20 – prostor, ve kterém je výbušná prachová atmosféra přítomna trvale nebo po dlouhé období. (> 1000h / rok)

Zóna 21 – prostor, ve kterém může vzniknout výbušná prachová atmosféra za normálního provozu. (> 10h ≤ 1000h / rok)

Zóna 22 – prostor, ve kterém výbušná prachová atmosféra může za normálního provozu vzniknout nebo může vzniknout pouze na krátké období. (> 0h ≤ 10h / rok)

Nevýbušná elektrická zařízení jsou rozdělena do dvou skupin:

Skupina I – elektrická zařízeni pro doly s výskytem metanu

Skupina II – elektrická zařízení pro prostory s výbušnou atmosférou, jiné než doly s výskytem metanu. Tato skupina se pro výbušné plynné atmosféry dále dělí a tři podskupiny IIA, IIB a IIC. V úvahu je brána podskupina plynu.

Skupina výbušnosti	I	IIA	IIB	IIC
Zástupce	metan	propan	ethylen	vodík
MESG [mm]	1,14	> 0,9	0,5 ÷ 0,9	< 0.5

Tab. 2-1 Rozdělení plynů a par do skupin modle MESG (IEC)

Podle nebezpečnosti se plyny dělí do skupin podle tabulky 2-1. Rozhodujícím ukazatelem je tzv. maximální experimentální bezpečná spára MESG. Ta se zjišťuje experimentálně na zkušebních zařízeních podle IEC 79-1A. Výjimku tvoří důlní plyn (metan), který je zařazen do samostatné skupiny I. Z hlediska nebezpečí výbuchu je nejnebezpečnější skupina IIC reprezentovaná např. vodíkem.

Na základě členění prostorů do zón a dělení zařízení do skupin jsou specifikovány kategorie

Kategorie	**Skupina**	**Prostor**
M1	I	doly s výskytem metanu
M2	I	doly s výskytem metanu
1G	II	zóna 0
2G	II	zóna 1
3G	II	zóna 2
1D	II	zóna 20
2D	II	zóna 21
3D	II	zóna 22

Tab. 2-2 Kategorie zařízení

Dalším významným faktorem uplatňujícím se v problematice výbuchů je teplota. Elektrické zařízení musí být voleno tak, aby maximální povrchová teplota nedosáhla teploty vznícení, kteréhokoliv plynu nebo pár, které mohou být přítomny. Symboly pro teplotní třídy, které mohou být vyznačeny na elektrickém zařízení mají význam podle následující tabulky.

Teplotní třída elektrického zařízení	**Maximální teplota elektrického zařízení**	**Teplota vznícení plynu nebo páry**
T1	450 °C	> 450 °C
T2	300 °C	> 300 °C
T3	200 °C	> 200 °C
T4	135 °C	> 135 °C
T5	100 °C	> 100 °C
T6	85 °C	> 85 °C

Tab. 2-3 Maximální povrchová teplota zařízení

2.4 Typy ochran

Elektrické zařízení by mělo být, pokud je to prakticky možné, umístěno v prostoru bez nebezpečí výbuchu. Kde to není možné splnit, tam by mělo být zařízení umístěno v co nejméně nebezpečném prostoru. V takových případech je nutné přikročit k různým typům ochran. V následující tabulce je uveden přehled používaných typů ochran v prostředích s nebezpečím výbuchu.

Označení	**CENELEC standard**	**IEC standard**	**Název**	**Použití (zóna)**
o	EN 50015	IEC 79-6	Olejový závěr (Oil encapsulation)	1,2
p	EN 50016	IEC 79-2	Závěr s vnitřním přetlakem (Overpressure encapsulation)	1,2
q	EN 50017	IEC 79-5	Pískový závěr (Sand encapsulation)	1,2
d	EN 50018	IEC 79-1	Pevný závěr (Pressure resistant encapsulation)	1,2
e	EN 50019	IEC 79-7	Zajištěné provedení (Increased safety)	1,2
m	EN 50028	IEC 79-18	Zalití zalévací hmotou (Cast encapsulation)	1,2
i	EN 50020 EN 50039	IEC 79-11	Jiskrová bezpečnost (Intrinsic safety)	0,1,2
n	EN 50021	IEC 79-15	Elektrická zařízení pro zónu 2 (Electrical components for zone 2)	2

Tab. 2-4 Typy ochran před výbuchem

Nejčastější používané typy ochran jsou:

Jiskrová bezpečnost (*Intrinsic safety*) – Princip jiskrové bezpečnosti spočívá v konstrukci a zabezpečení obvodů a zařízení tak, aby v normálním provozu a při jedné nebo dvou poruchách v obvodu a na součástkách, u kterých je možnost vzniku zkratu nebo přerušení, byla energie omezena tak, aby nebyla schopna jiskrou nebo vytvořeným teplem (horkými povrchy) způsobit iniciaci výbušné směsi.

Pevný závěr (*Pressure resistant encapsulation*) – Nejrozšířenějším typem ochrany před výbuchem je pevný závěr. Protože princip pevného závěru vychází z předpokladu, že uvnitř závěru může vzniknout výbuch, musí být závěr dostatečně pevný. Proto je konstrukce robustní a velice těžká, což je hlavní nevýhodou tohoto typu ochrany. Pro zabránění přenosu výbuchu zevnitř závěru do vnější atmosféry musí být všechny spáry mezi vnějškem a vnitřkem závěru přesně definovány, tzn. že musí mít určitou minimální délku a velmi malou maximální šířku spáry (vzduchové mezery mezi spojenými částmi).

Zajištěné provedení (*Increased safety*) – Dalším velice často používaným typem ochrany před výbuchem je zajištěné provedení. Tento typ ochrany je možné použít pouze pro nejiskřící zařízení, která se nepřiměřeně neohřívají. Princip ochrany spočívá v mechanickém provedení a zabezpečení zařízení ochrannými systémy tak, aby byl vznik jisker a nedovolených horkých míst za normálního provozu a při poruchových stavech nepravděpodobný.

Závěr s vnitřním přetlakem (*Overpressure encapsulation*) – Tento typ ochrany před výbuchem využívá techniky, která brání vzniku výbušné atmosféry uvnitř závěru tím, že přivádí do závěru ochranný plyn (čistý vzduch nebo inertní plyn), jež je udržován na tlaku vyšším, než je okolní tlak.

2.5 Jiskrová bezpečnost (Intrinsic safety)

Jiskrově bezpečná zařízení určená pro zóny 1 a 2 jsou označována „ib“. Tato kategorie musí splňovat požadavky kladené normou IEC 79-11. Podle této normy jiskrově bezpečná zařízení pro zónu 0 musí splňovat přísnější požadavky dané kategorií „ia“.

V některých státech Evropy a rovněž v ČR se dosud vyžaduje pro napájení jiskrově bezpečných obvodů v zóně 0 galvanické oddělení jiskrově bezpečných obvodů od obvodů, které nejsou jiskrově bezpečné (oddělovače). Dále se v této zóně nepovoluje použití Zenerovy bariéry, protože její spolehlivost závisí na jejím správném uzemnění. Při instalaci jiskrově bezpečných obvodů v zóně 0 je nutno věnovat pozornost především:

- Zajištění pečlivého oddělení jiskrově bezpečných obvodů od ostatních obvodů , čímž se zabrání zavlečení energie z jiných elektrických zdrojů do jiskrově bezpečných obvodů.
- Vlivu elektrických parametrů kabelu (indukčnosti a kapacity) na jiskrovou bezpečnost zařízení.
- Vyloučení vzniku indukovaného napětí při vedení jiskrově bezpečných obvodů v blízkosti silových vedení.
- Označení jiskrově bezpečných obvodů pro zabránění záměny v zapojení a bezpečné údržbě a kontrole jiskrově bezpečných zařízení.

Uzemnění jiskrově bezpečných obvodů smí být prováděno, pouze je-li to nutné z funkčních nebo bezpečnostních důvodů. Spojení s uzemněním musí být provedeno mimo zónu 0, avšak co nejblíže zařízení v zóně 0.

Kabely, které obsahují jiskrově bezpečné obvody musí být označeny. Je-li plášť nebo povrch značen barvou, musí být použito světle modré barvy. Kabely takto označené, nesmí být používány pro jiné účely. Jsou-li jiskrově bezpečné kabely nebo všechny kabely které jsou jiskrově bezpečné pancéřované, s kovovým pláštěm nebo stíněním, pak se označení jiskrově bezpečných kabelů nevyžaduje.

V zóně 0 lze kromě jiskrově bezpečných zařízení a systémů kategorie „ia“ používat i jiná nevýbušná zařízení, která mají zvýšenou úroveň ochrany před výbuchem zajištěnou většinou kombinací dvou nebo více typů ochrany (např. pevný závěr, který zároveň splňuje požadavky na závěr s vnitřním tlakem). Tato zařízení musí být pro použití v zóně 0 odzkoušena a schválena na zkušebně a v certifikátu musí mít výslovně uvedeno, že jsou vhodná pro použití v zóně 0 (dříve v prostoru s nebezpečím výbuchu SNV 3).

Taková zařízení jsou většinou označována jako Exs, kde „s“ znamená speciální závěr. Požadavky na tento druh ochrany nejsou pevně stanoveny a ověření bezpečnosti těchto zařízení je ponecháno na zkušebnách.

2.6 Výběr zařízení podle skupiny výbušnosti plynů

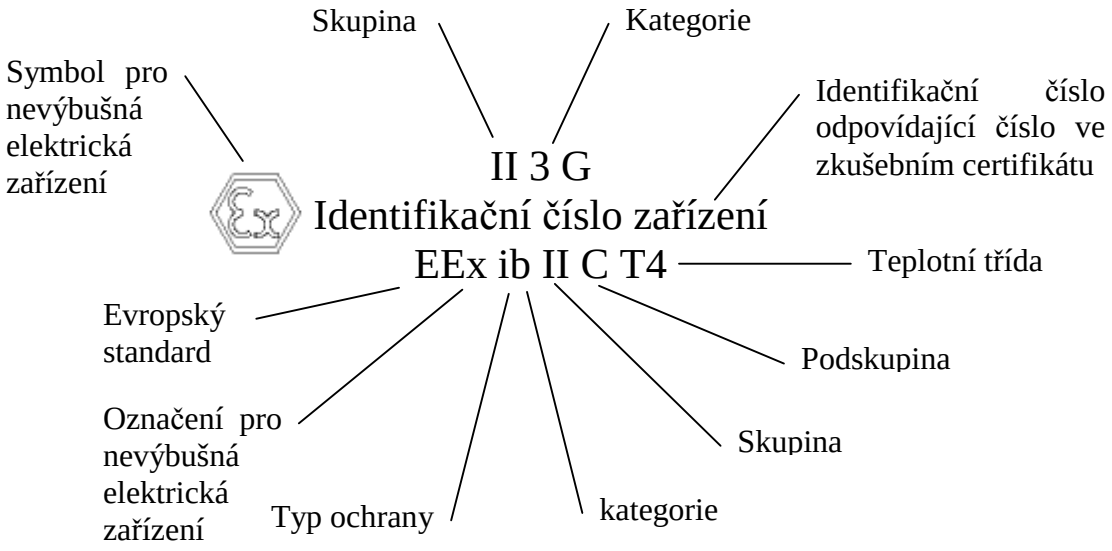
Elektrická zařízení s typem ochrany „e“, „m“, „o“, „p“ a „q“ jsou normálně zařízení skupiny II a jejich použití je možné pro všechny plyny. Elektrická zařízení s typem ochrany „d“ a „i“ jsou rozdělena do podskupiny IIA, IIB nebo IIC a jejich použití je omezeno pro plyny odpovídající skupiny výbušnosti plynů podle následující tabulky 2-1. Elektrické zařízení s typem ochrany „n“ je obvykle zařízení skupiny II, avšak pokud obsahuje uzavřené (zapouzdřené) spínací zařízení, nezápalné součásti nebo zařízení a obvody s omezenou energií, pak je zařízení zařazeno do podskupin IIA, IIB nebo IIC.

Podskupina výbušnosti	**Skupina zařízení**
IIA	IIA, IIB nebo IIC
IIB	IIB nebo IIC
IIC	IIC

Tab. 2-5 Vztah mezi rozdělením plynů/par a skupinami zařízení

2.7 Značení

Podle CENELE a IEC jsou zařízení určená pro prostředí s nebezpečí výbuchu značena podle následujícího schématu:



Další podrobnosti o této problematice lze nalézt přímo v příslušných normách nebo v literatuře [8] a [9].

3. Profibus

3.1 Obecný popis

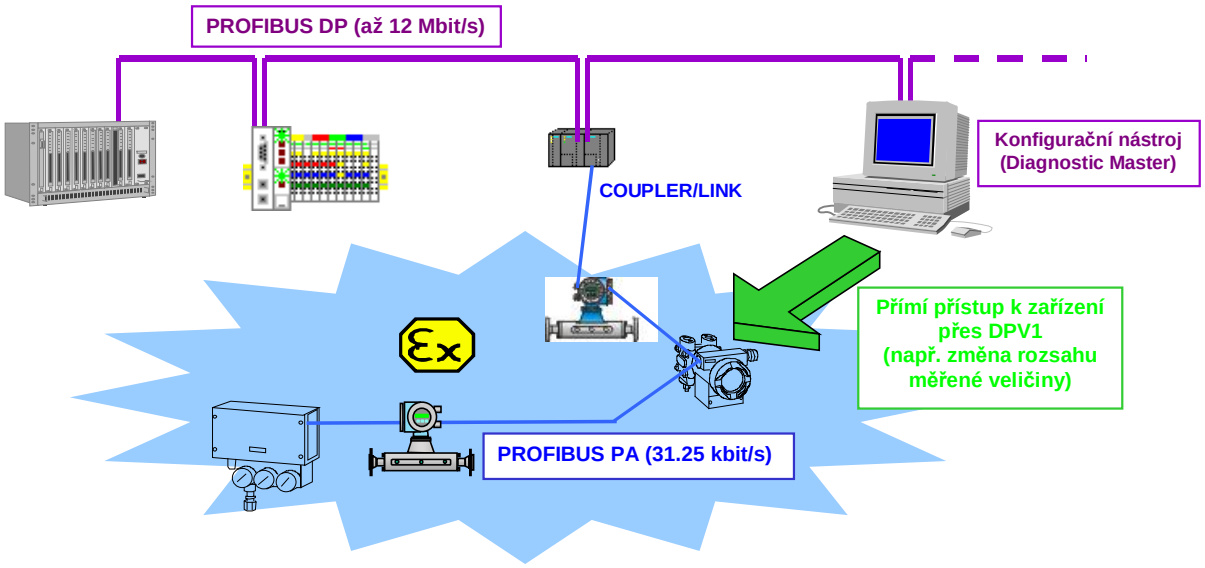
Profibus je průmyslovou sběrnicí, která nachází v automatizační technice poměrně široké uplatnění. Jedná se o standard firmy Siemens a její název vznikl z PROcess Field BUS. Asi největší předností je otevřenost pro všechny výrobce. Tím není zákazník omezen pouze na jednoho výrobce ale může libovolně kombinovat zařízení různých výrobců.

Profibus je specifikován normou EN 50170 [1]. V roce 1998 došlo k rozšíření této normy takzvanou DPV1 (Profibus DP/V1) [2]. V současné době definuje norma tři komunikační standardy:

Profibus FMS (*Fieldbus Message Specification*) – Sběrnice určená pro náročné komunikační úlohy. Dnes se již prakticky nepoužívá.

Profibus DP (*Decentralized Pheriphery*) – Sběrnice určená pro průmyslové použití. Vyznačuje se vysokou přenosovou rychlostí (až 12Mbit/s).

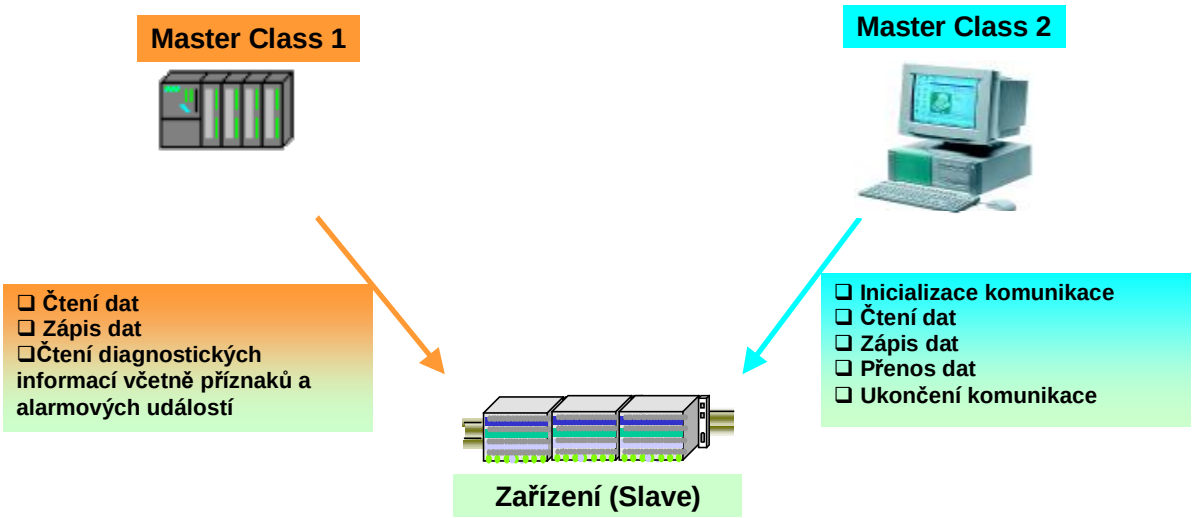
Profibus PA (*Process Automation*) – Sběrnice určená pro prostředí s nebezpečím výbuchu a tudíž se používá převážně v petrochemickém průmyslu. Oproti Profibusu DP má sníženu přenosovou rychlost (31,25kbit/s).



Obr. 3-1 Příklad struktury sběrnice Profibus

Základní norma specifikuje základní funkce pro diagnostiku, konfiguraci a **cyklickou komunikaci** mezi zařízeními typu Master Class 1 a slave. Cyklická komunikace je výměna nakonfigurovaných dat mezi těmito zařízeními. Rozšíření dané normou DPV1 rozšiřuje diagnostiku a konfiguraci. Dále zavádí takzvanou **acyklickou komunikaci** mezi zařízeními Master Class 1, 2 a slave. Ta složí převážně ke konfiguraci zařízení typu slave. **Master Class 1** tedy vykonává hlavně cyklickou komunikaci se svými podřízenými slave stanicemi. Naproti tomu **Master Class 2** slouží výhradně ke konfiguraci a diagnostice sběrnice.

Profibus je sběrnicí multi-master, což znamená, že na sběrnici může být i více zařízení typu Master Class 1 či 2. Avšak v danou chvíli má právo přístupu na sběrnici pouze jedno zařízení master. Stanice master spolu komunikují po logickém kruhu (token ring) s oběhem pověření (token passing). Pokud master obdrží pověření, provede výměnu dat se svými stanicemi slave a předá pověření další stanici master v logickém kruhu (nebo sám sobě, pokud je jediný). Pokud dojde ke ztrátě pověření či odpojení zařízení master, je logický kruh rekonfigurován a master s nejnižší adresou je zodpovědný za vygenerování nového pověření. K rekonfiguraci kruhu dochází i v případě připojení nové stanice master. Každý master je zodpovědný za svá zařízení slave (interval adres daný konfigurací), které je možné také za provozu odpojovat a zase připojovat. K dispozici je 127 adres (0 až 126) pro adresaci stanic master a slave bez rozdílu. Protože je vyžadována nejméně jedna stanice master k dispozici je až 126 adres pro stanice. Adresa 127 je vyhrazena jako globální adresa tj. zprávy typu broadcast a multicast (zprávy všem stanicím nebo jen skupině stanic). Globální adresa je rozšířena v datové části zprávy (telegramu).



Obr. 3-2 Master Class 1 a 2 (rozšíření DPV1)

3.2 Fyzická vrstva

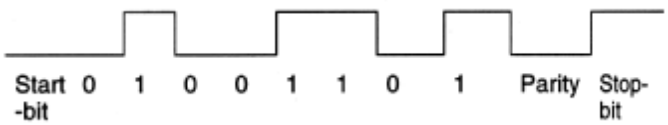
Jedná se o nejníže položenou vrstvu v ISO/OSI modelu. Definuje přenosové médium, připojovací konektory a elektrické úrovně včetně kódování a topologie sítě.

3.2.1 Profibus DP

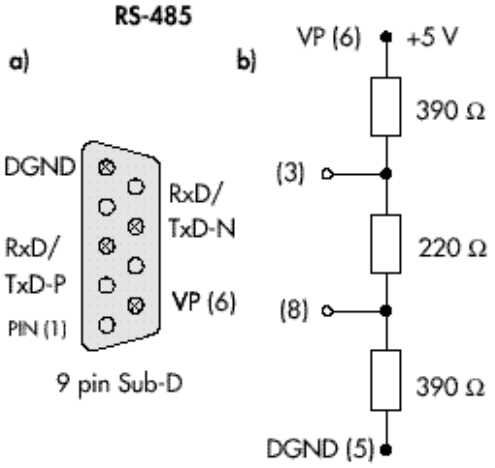
Jak již bylo uvedeno, jeden z rozdílů mezi Profibusem DP a PA je ve fyzické vrstvě. Profibus DP je definován evropskou normou EN 50 170. Tato vrstva tedy definuje:

Přenosové médium:	stíněný kroucený dvoudrát zakončený terminátory (RS485) nebo optické vlákno
Konektor:	viz obrázek 3-4.
Topologie sítě:	sběrnice, hvězda, kruh a jejich kombinace.
Kódování:	označováno jako NRZ (Non Return to Zero). Viz. Obr 3-3.
Přenosová rychlost:	9,6;19,2; 45,45; 93,75; 187,5; 500; 1500; 3000; 6000; 12000 [kbit/s]
Formát přenosu:	11 bitů na byte (1 start bit, 8 datových bitů (LSB první), 1 paritní bit (sudá parita) a 1 stop bit)
Zabezpečení přenosu:	Jeden paritní bit na každý znak a jeden byte kontrolního součtu FSC (Frame Check Sequence)
Počet stanic na segment:	Pro krouceny dvoudrát je maximální počet 32. U optického vlákna je dán maximální adresací tj. 127 stanic.

Jak je patrné, hlavní výhodou Profibusu DP je především vysoká přenosová rychlost (12Mbit/s), proměnná topologie sítě a poměrně jednoduchá instalace.



Obr. 3-3 Kódování NRZ (Non Return to Zero)



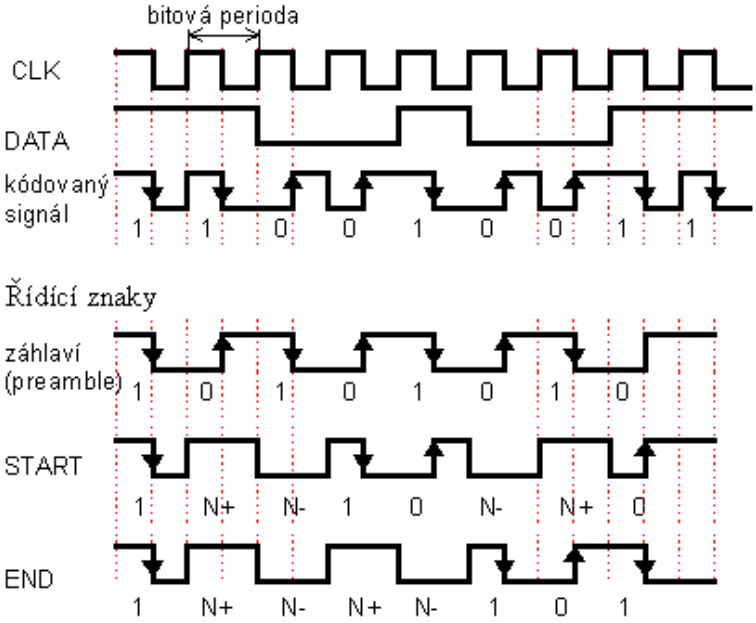
Obr. 3-4 Zapojení konektoru pro připojení ke sběrnici Profibus DP

3.2.2 Profibus PA

3.2.2.1 Úvod

Profibus PA je přizpůsoben pro práci ve výbušném prostředí, označovaném jako Ex (Explosive protection). Tato odlišnost od Profibusu DP je definována normou IEC 1158-2.

Přenosové médium:	kroucený dvoudrát zakončený terminátory (stíněný/nestíněný)
Topologie sítě:	sběrnice, hvězda a jejich kombinace.
Kódování:	Manchester II. Viz Obr 3-5.
Přenosová rychlost:	31,25 [kbit/s]
Formát přenosu:	8 bitů na byte (8 datových bitů (MSB první))
Zabezpečení přenosu:	16-bit CRC (Cyclic Redundancy Check)
*Počet stanic na segmen*t:	maximálně 32 stanic na segment (pro výbušná prostředí je navíc maximální počet omezen odběrem ze sběrnice). Maximálně 4 opakovače.



Obr. 3-5 Kódování Manchester II

Z předchozího je patrné, že u Profibusu PA byla podstatně snížena komunikační rychlost a proto je označován jako Low Speed – H1. Zde fyzická vrstva vedle kódování a dekódování zprávy doplňuje zprávu o znaky záhlaví (preamble), začátek (start) a konec (end) (viz obr. 3-5). Při příjmu samozřejmě dochází k jejich odstranění. Znak záhlaví slouží k synchronizaci generátoru hodinového kmitočtu v přijímači zprávy. Začátek a konec zprávy vymezuje vlastní zprávu.

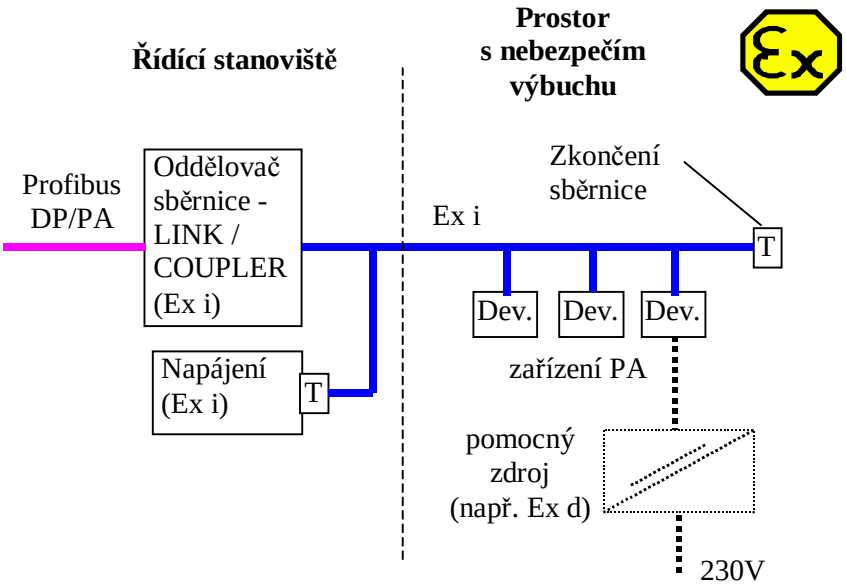
Pro přenos po vedení se využívá, jak už bylo zmíněno v úvodu, kódovaní Manchester II (obr. 3-5). Jedná se o synchronní způsob kódovaní, kde datový signál je sloučen s hodinovým signálem. Výsledný kódovaný signál interpretuje stav log. 1 jako záporný (z log. 1 do log. 0) přechod bitové periody, stav log. 0 jako kladný (z log. 0 do log. 1) přechod. Kódování znaků

záhlaví, začátku a konce zprávy je řešeno pomocí speciálních bitových sekvencí, jejichž součástí jsou stavy N+ a N- (bez přechodu uprostřed bitové periody).

Vysílací zařízení kóduje signál pomocí vysílání proudu ±9mA. Na stejnosměrném napájení tak vznikne modulační napětí U_{p-p} = 0,75÷1V (špička-špička). Napájecí napětí může být v rozsahu 9÷32V a je závislé na různých faktorech jako: použitém převodníku, délce přívodních vodičů nebo typu prostředí, ve kterém zařízení pracuje.

3.2.2.2 Profibus PA a jiskrová bezpečnost

Profibus PA je převážně používán v chemickém a petrochemickém průmyslu, protože je konstruován jako takzvaně jiskrově bezpečný (viz kapitola 2.5). Proberme nyní parametry a vlastnosti Profibusu PA v těchto prostředích. Na následujícím obrázku je znázorněna typická architektura sběrnice v prostředí s nebezpečím výbuchu.



Obr. 3-6: Typická architektura sběrnice v prostředí s nebezpečím výbuchu

Sběrnice je dvouvodičová a přenáší se po ní jak data, tak i napájecí napětí. Povolený počet zařízení závisí na maximálním povoleném proudu, který lze ze sběrnice odebírat. Zařízení s vyšším odběrem je možno napájet z pomocného zdroje, je však nutné dodržet podmínky kladené na zařízení tohoto typu v odpovídajícím prostředí.

Požadavky na tento typ sběrnice je specifikován modelem **FISCO** (Fieldbus Intrinsically Safe Concept) [4]. Ten byl navržen zkušebním ústavem Physikalisch Technische Bundesanstalt (PTB) v Německu ve spolupráci s významnými výrobci v tomto odvětví. Model specifikuje následující obecná omezení:

- Zařízení musí splňovat požadavky jiskrově bezpečného zařízení
- Pokud zařízení vysílá, neodebírá ze sběrnice žádný proud
- Všechny pasivní spotřebiče (stanice slave) odebírají konstantní proud
- Slave pracuje jako pasivní spotřebič proudu

- Vnitřní kapacita a indukčnost zařízení je omezena
- Hlavní sběrnice musí být zakončena na obou koncích

Následující tabulka ukazuje model FISCO pro zařízení v kategorii EEx ib IIC/IIB a EEx ia IIC.

	EEx ib IIC / IIB	**EEx ia IIC**
napájecí zdroj	$U_S = 14 \div 24V$ I_K – zkratový proud Např: $I_K = 128mA$ při $U_S = 15V$ (IIC) $I_K = 280mA$ při $U_S = 15V$ (IIB)	$U_S = 14 \div 20V$ $U_0 \geq 2 \cdot U_S$ Např: $I_K = 215mA$ při $U_S = 15V$ (IIC)
kabel (na km)	R' = 15 ÷ 150Ω (odpor smyčky) L' = 0,4 ÷ 1mH C' = 80 ÷ 200nF l_{MAX} = 5000m (maximální délka)	R' = 15 ÷ 150Ω (odpor smyčky) L' = 0,4 ÷ 1mH C' = 80 ÷ 200nF l_{MAX} = 1000m (maximální délka)
zakončení	R = 90 ÷ 100Ω C = 0 ÷ 2,2μF	R = 90 ÷ 100Ω C = 0 ÷ 2,2μF

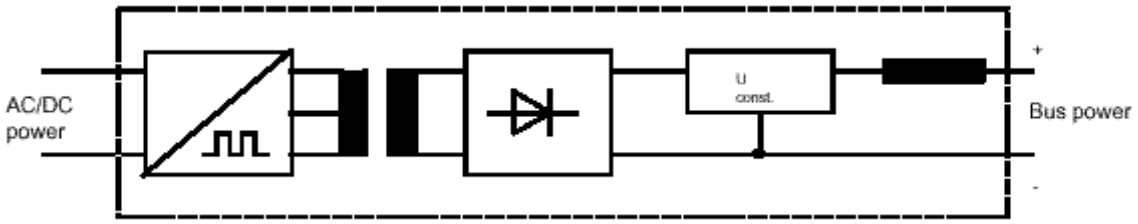
Tab. 3-1 Omezení stanovená modelem FISCO

3.2.2.3 Napájecí zdroje

Jiskrově bezpečné napájecí zdroje se stejně jako zařízení Profibus PA v modelu FISCO rozdělují do dvou kategorií na ia a ib. Zařízení, která nejsou takzvaně jiskrově bezpečná (Not Intrinsically Safe) je možné provozovat společně s omezovačem. Napájecí zdroje se většinou umisťují mimo prostory s nebezpečím výbuchu (viz obr. 2-4). Pokud však zařízení musí být umístěno v této zóně je potřeba zajistit jeho ochranu před výbuchem například pevným závěrem „d“. Napájecí zdroje často bývají součástí oddělovačů sběrnice. Tabulka 3-2 ukazuje elektrické charakteristiky napájecího zdroje.

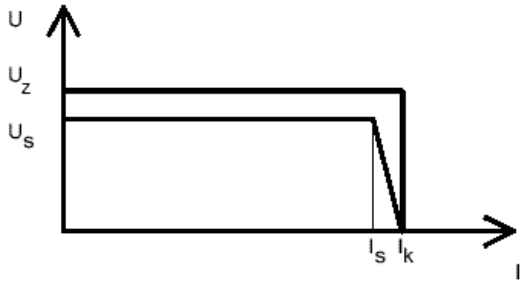
	Bez jiskrové bezpečnosti	Jiskrově bezpečné ib	Jiskrově bezpečné ia
výkon		< 1,8W	< 1,2W
výstupní napětí	≤ 32V	≤ 17,5V	≤ 24V
výstupní zvlnění	≤ 16mV	≤ 16mV	≤ 16mV
výstupní impedance	≥ 3kΩ	≥ 400Ω	≥ 400Ω
útlum	≥ 50dB	≥ 50dB	≥ 50dB

Tab. 3-2 Charakteristiky napájecích zdrojů

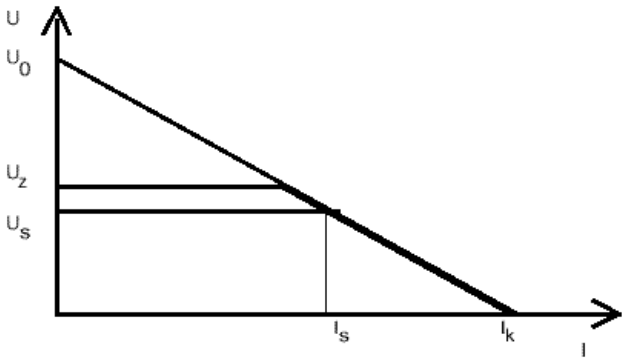


Obr. 3-7 Blokové schéma napájecího zdroje bez jiskrové bezpečnosti

Napájecí zdroje kategorie „ib“ mají obdélníkovou nebo lichoběžníkovou VA charakteristiku podle obrázku 3-8a. U_S je výstupní napětí, U_Z maximální napájecí napětí, I_S je výstupní proud a I_K je maximální zkratový proud zdroje. Minimální výstupní napětí bylo experimentálně stanoveno na 13,5V.

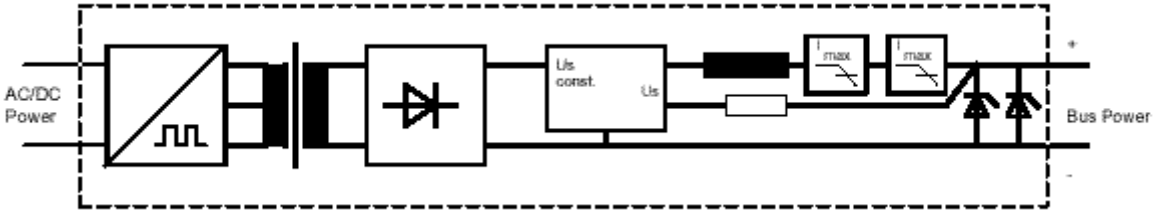


Obr. 3-8a VA charakteristika napájecího zdroje v kategorii „ib“

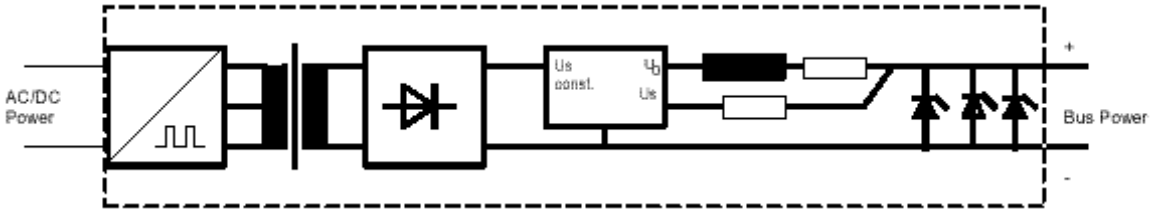


Obr. 3-8b VA charakteristika napájecího zdroje v kategorii „ib“

Napájecí zdroj v kategorii „ia“ se liší hlavně průběhem VA charakteristiky. Jeho charakteristika je znázorněna na obrázku 3-8b. Její průběh je lineární a konstanta U_0 je napětí naprázdno. Vnitřním provedením se tento zdroj liší tím, že omezovač proudu z kategorie „ib“ je nahrazen fixním rezistorem, který omezuje maximální zkratový proud. Maximální výstupní napětí je omezeno Zenerovými diodami. Jeho blokové schéma je na obrázku 3-10.



Obr. 3-9 Blokové schéma napájecího zdroje v kategorii „ib“



Obr. 3-10 Blokové schéma napájecího zdroje v kategorii „ia“

3.2.2.4 Zařízení Profibus PA

Zařízení Profibus PA, která mají být používána v prostředí s nebezpečím výbuchu musejí mít certifikát, že jsou jiskrově bezpečné. Zde musí být uvedeno že zařízení je v souladu s modelem FISCO a v jaké kategorii. Dále zde musí být uveden maximální pracovní proud, který zařízení odebírá ze sběrnice, maximální proud, který zařízení může do sběrnice dodat, maximální odebíraný proud v případě poruchy, maximální odebíraný výkon atd. Pokud je zařízení napájeno z pomocného zdroje, musí certifikát obsahovat poznámku o tom, že zařízení je galvanicky odděleno od sběrnice.

Důležitým parametrem u těchto zařízení je teplotní třída. Protože maximální dodávaný výkon je omezen je omezena i maximální teplota, která může na zařízení vzniknout. Například pokud je napájení omezeno na 1,3W je teplotní třída takového zařízení T4 (viz. kapitola 2.3).

3.3 Spojová vrstva

3.3.1 Obecný formát telegramů

Spojová vrstva definuje formát telegramů, bezpečnostní mechanismy a dostupné přenosové služby. Pro Profibus DP/PA jsou důležité dvě následující přenosové služby.

SRD: **Send and request data with acknowledge**. Data jsou odeslána a přijata v jednom komunikačním cyklu. Tato služba se používá k výměně dat mezi stanicí master a podřízenou stanicí. Master pošle data a podřízená stanice odpoví potvrzujícím telegramem, který rovněž může obsahovat data. Používá se převážně pro cyklickou komunikaci.

SDN: **Send data with no acknowledge**. Služba umožňuje vysílat telegramy bez potvrzování. Využívá se například k zaslání dat jedné skupině (*multicast*) nebo všem (*broadcast*) zařízením slave současně.

Podle počátečního znaku (Start Delimiter) v telegramu Profibus DP/PA rozeznává následující typy telegramů:

SD1: **Request FDL Status**. Telegram, který je vysílán aktivní stanicí (master) určený k nalezení nových aktivních stanic Profibus slave. Zařízení master tento telegram zasílá periodicky v intervalech daných tzv. **GAP faktorem**. Ten je udán násobkem cyklů na sběrnici. Tímto telegramem je vždy ověřena jedna konkrétné adresa a příštím je ověřena adresa následující.

SD1 (0x10)	DA	SA	FC	FCS	ED (0x16)

SD2: **Telegram**. Datový telegram s proměnnou délkou (datových bloků).

SD2 (0x68)	LE	LEr	SD (0x68)	DA	SA	FC	DSAP (0x3C)	SSAP (0x3E)	DU	FCS	ED (0x16)

SD3: **Telegram**. Datový telegram s neměnnou délkou (délka datových bloků je 8 byteů).

SD3 (0xA2)	DA	SA	FC	DU	FCS	ED (0x16)

SD4: **Token telegram**. Telegram pro předání pověření mezi dvěmi zařízeními.

SD4 (0xDC)	DA	SA

Struktura telegramů Profibus DP a PA se liší pouze nepatrně. Tento rozdíl je způsoben odlišnostmi ve fyzické vrstvě. Ukažme tento rozdíl (znázorněno modře) např. na telegramu SD2.

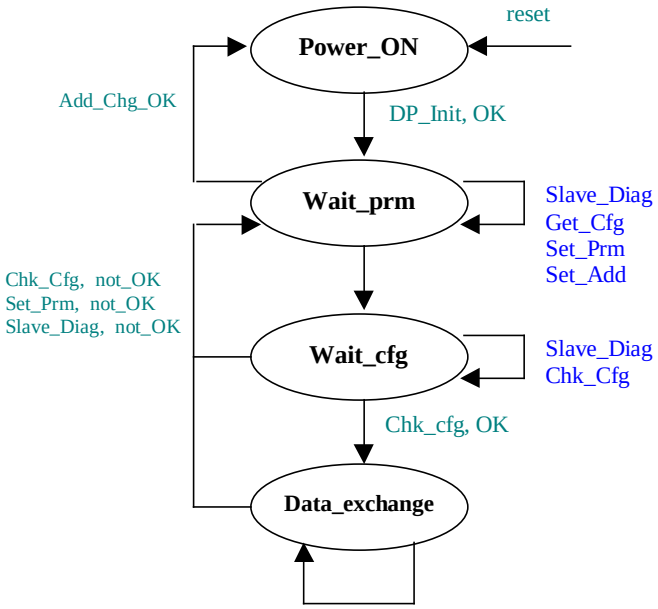
P	**SOF**	SD2	LE	LEr	SD2	DA	SA	FC	DSAP (0x3C)	SSAP (0x3E)	DU	**CRC**	**EOF**

SD – Start Delimiter. Slouží k rozlišení typů telegramů.
FC – Function Code. Řídící oktet, který indikuje, zda se jedná o dotazovací telegram nebo telegram s odpovědí. Dále obsahuje informace tykající se vlastní spojové vrstvy v ISO/OSI modelu.
LE – Length. Délka datových údajů v telegramu (DU) včetně DA, SA, FC, DSAP, SSAP.
LEr – Length Repeated
DA – Destination adress. Cílová adresa (0 ÷127).
SA – Source adress. Zdrojová adresa (0 ÷127).
DSAP – Destination Service Acces Point. Pokud je nastaven nejvyšší bit v DA a SA určuje službu, která má být provedena (jedná se vlastně o místo obsluhy daného telegramu ve firmwaru periferního zařízení – pomáhá adresátu rozlišit telegramy).
SSAP – Source Service Acces Point. Jako DSAP ale ze strany vysílající stanice.
DU – Data Unit. Přenášená data (0 ÷ 244 bytů).
FCS – Frame Checking Sequence. Informace zajištující zabezpečení přenášených dat.
ED – End Delimiter. Zakončuje telegram.
P – 16-bit záhlaví (Preamble)
SOF – START (Start of Frame)
EOF – END (End of Frame)
CRC – 16-bit CRC (Cyclic Redundancy Check)

Přístupové body (Service Acces Points) lze chápat jako komunikační kanály, které specifikují o jakou službu se při komunikaci jedná. Nejdůležitější z nich jsou uvedeny v následující tabulce 3-3:

DP Master	**DP Slave**	**Funkce**
62	55	Změna adresy
62	56	Čtení vstupů
62	57	Čtení výstupů
62	58	Globální řízení
62	59	Čtení konfigurace (**Get_Cfg**)
62	60	Čtení diagnostických informací (**Slave_Diag**)
62	61	Zaslání parametrů (**Set_Prm**)
62	62	Kontrola konfigurace (**Chk_Cfg**)

Tab. 3-3 SAP - Přístupové body definované základní normou [1]



Obr. 3-11 Stavový diagram DP/PA slave

3.3.2 Chování zařízení DP/PA slave

Chování zařízení DP/PA lze popsat stavovým diagramem znázorněným na obrázku 3-11. Provedením inicializačních procedur se zařízení dostane z počátečního stavu Power_On do stavu Wait_prm. V tomto stavu přijímá slave pouze následující typy telegramů. Jsou to Set_Add, Slave_Diag, Get_Cfg a Set_Prm.

Set_Add – Pomocí této služby může master změnit adresu periferního zařízení.
Slave_Diag – Master používá tento telegram k získání diagnostických informací od zařízení.
Get_Cfg – Označuje požadavek stanice master na přijetí aktuální konfigurace zařízení typu slave.
Set_Prm – Touto službou master zasílá nové parametry zařízení slave.

Slave odpoví na parametrizační telegram pouze krátkým potvrzením. Pokud došlo k chybě při parametrizaci, oznámí ji slave až při další odpovědi na telegram Slave_Diag. Pokud je zařízení úspěšně nakonfigurováno přejde do stavu Wait_Prm. Nyní přijímá telegramy Chk_Cfg a Slave_Diag.

Chk_Cfg – Význam je stejný jako u telegramu Get_Cfg pouze s tím rozdílem, že data jsou přenášena směrem od zařízení master a slave porovnává přijatou konfiguraci s konfigurací skutečnou.

Slave na telegram Chk_Cfg odpoví krátkým potvrzením a v případě chybné konfigurace dá o této skutečnosti stanici master na vědomí v další odpovědi na telegram Slave_Diag. Kladným potvrzením v telegramu Slave_Diag slave přejde do stavu výměny dat – Data_Exchange. V tomto stavu akceptuje telegramy Data_Exchange, Slave_Diag, Chk_Cfg,

Get_Cfg, Rd_Inp, Rd_Out a příkazy globálního řízení. Příkazy globálního řízení jsou příkazy určené skupině zařízení typu slave z důvodu synchronizace jejich dat. Jedná se o příkazy:

Clear_Data – Uvede výstupy zařízení slave do definovaného stavu. Například pro stav nouze.
Freeze – Periferní zařízení zmrazí své vstupy. Tím je možné zjistit hodnotu všech vstupů v daném okamžiku. K opětnému uvolnění dojde telegramem Unfreez.
Sync – Výstupní data, která přijdou v následujícím telegramu, budou uložena do vyrovnávací paměti, ale nebudou fyzicky zapsána na výstupy. K tomu dojde až příchodem telegramu Unsync. Slouží k synchronizaci výstupů dané skupiny zařízení.

3.3.3 Některé telegramy podrobněji

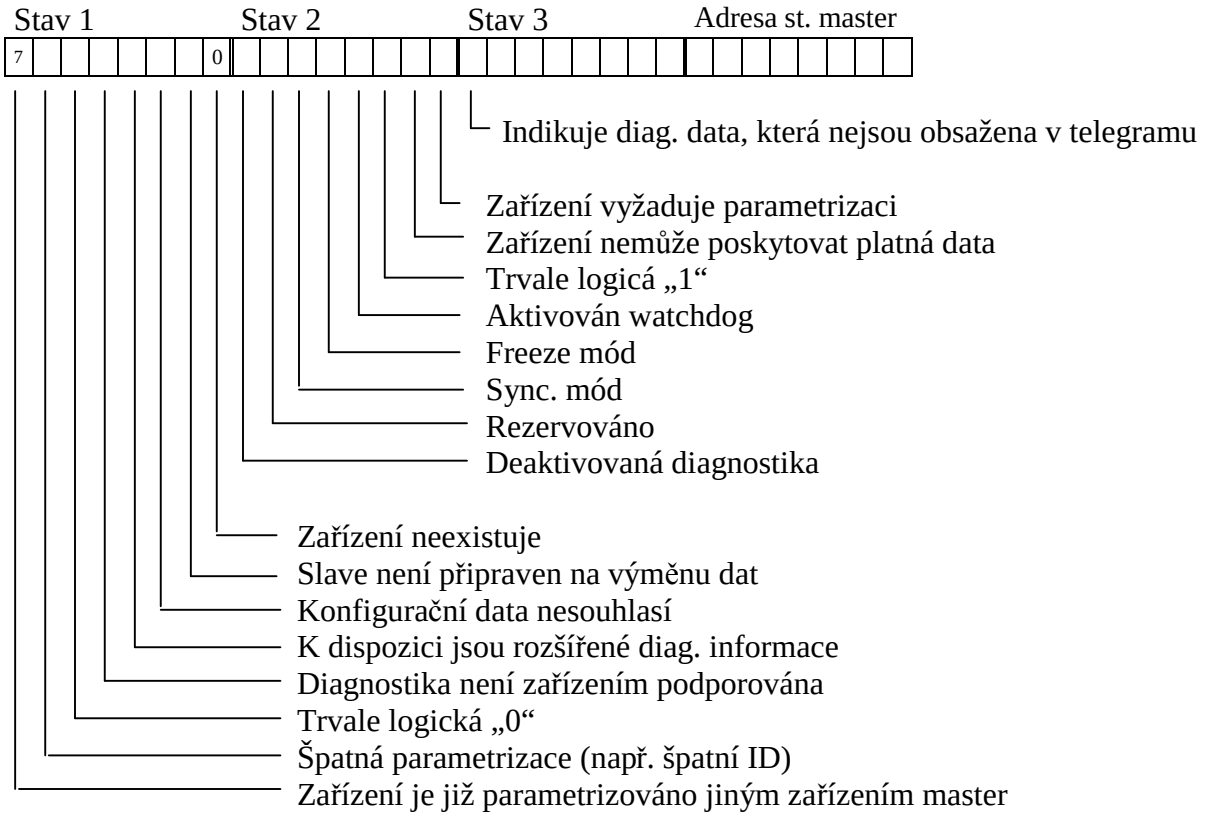
3.3.3.1 Změna adresy stanice slave

Ve stavu Power_ON může Master Class 2 použít SAP=55 ke změně adresy slave zařízení. Délka telegramu je 9 (LE=9). V prvním bytu v datové oblasti telegramu se přenáší nová adresa. V druhém a třetím se přenáší identifikační číslo zařízení. Toto číslo je přiřazeno PNO (Profibus Trade Organization) a je jedinečné. Slouží k jednoznačnému přiřazení konfiguračního GSD souboru k daného zařízení. Posledním, čtvrtým datovým bytem se povoluje či zakazuje další změna adresy. Pokud je roven nule, jsou další změny adresy povoleny. Pokud je roven jedné, je další změna adresy zakázána.

Nová adresa								ID (high)								ID (low)								Změny							
7							0																								

3.3.3.2 Odpověď na Slave_Diag

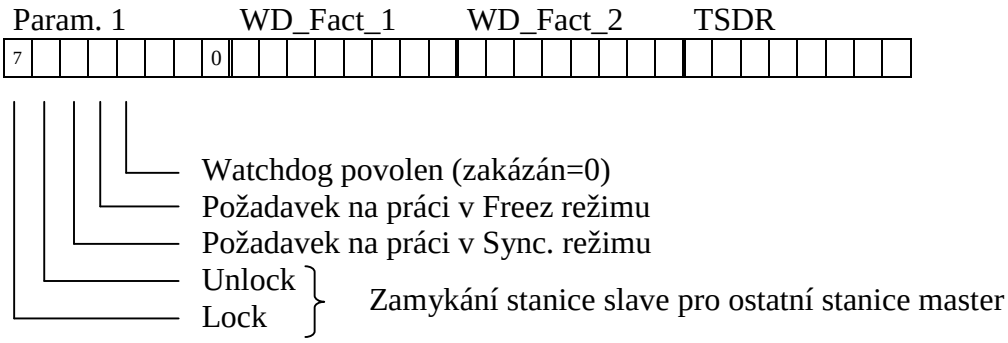
Pomocí telegramu Slave_Diag (SAP=0x60) se zařízení master dotazuje podřízeného zařízeni na diagnostické informace. V odpovědi na tento telegram slave odesílá telegram, jehož datová oblast je veliká minimálně šest bytů, jimiž stanici master informuje o svém stavu.



Význam bitů informujících o stavu zařízení v prvních třech bytech je uveden nad tímto odstavcem. Spodních sedm bitů třetího bytu je rezervováno. Čtvrtý byte obsahuje adresu stanice master po parametrizaci (0xFF – bez parametrizace). V pátém a šestém se přenáší identifikační číslo zařízení. V dalších již nepovinných datových bytech se přenáší rozšířené diagnostické informace. Bližší informace je možné najít například v [5].

3.3.3.3 Parametrizační telegram

Parametrizační telegram používá master ke své identifikaci a zároveň jím specifikuje, v jakém režimu má zařízení slave pracovat. Slave na tento telegram odpoví pouze takzvaným krátkým potvrzením 0xE5 (short acknowledge). A to i v případě že došlo k chybě. Tuto skutečnost oznámí až v odpovědi na Slave_Diag.



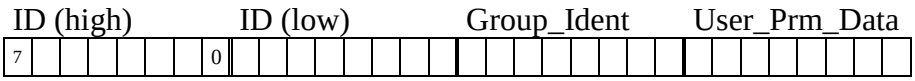
Prvním bytem master říká podřízenému zařízení v jakém režimu má pracovat. Spodní číst tohoto bytu je rezervována. Druhé dva byty určují hodnotu watchdogu.

$$WD = 10ms \cdot WD_Fact_1 \cdot WD_Fact_2 \ [s]$$

Funkce watchdog umožňuje zařízení slave monitorovat, zda je master stále aktivní. Čtvrtý byte obsahuje hodnotu **TSDR**. Tato hodnota reprezentuje minimální čas, který musí slave čekat než smí odpovědět. Udává se v násobcích T_{BIT}.

$$T_{BIT} = \frac{1}{\text{baud rate}} [s]$$

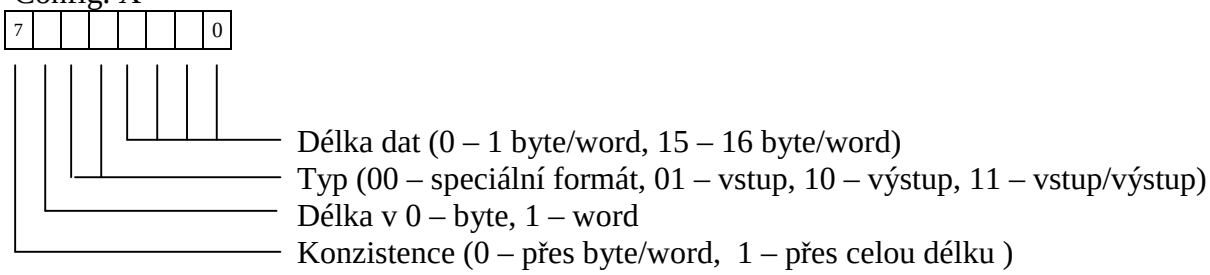
Hodnota TSDR může být minimálně $11 \cdot T_{BIT}$. V pátém a šestém bytu se opět přenáší identifikační číslo zařízení. V sedmém master zasílá přidělené identifikační číslo skupiny (adresa pro zasílání příkazu jedné skupině). Posléze následují uživatelská parametrizační data.



3.3.3.4 Konfigurační telegram

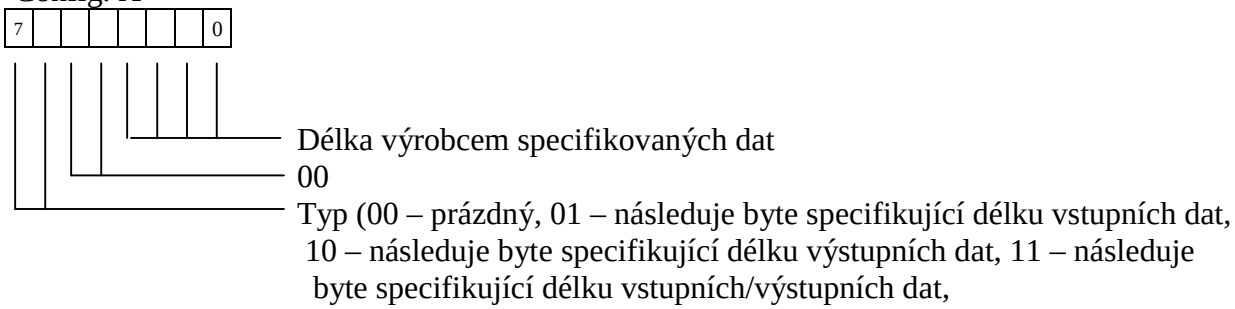
Po parametrizaci musí master zaslat podřízeném zařízení konfigurační telegram. Tím specifikuje komunikaci během cyklické výměny dat. Datová část tohoto telegramu může být dlouhá maximálně 16 bytů.

Config. X

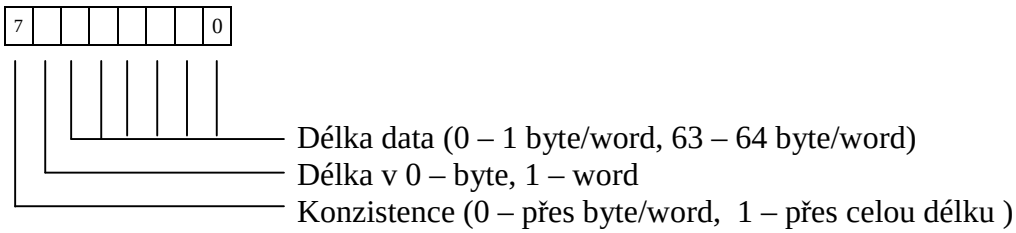


Pokud jsou data přenášena ve formátu word, je nejprve přenášen vyšší byte. Pokud je typem specifikován speciální formát (Typ = 00), má tento konfigurační byte poněkud odlišný význam.

Config. X



Délka



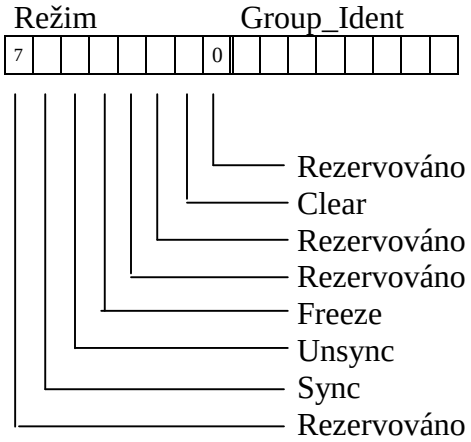
Stejně jako u parametrizačního telegramu slave potvrdí přijetí pouze odpovědí 0xE5 – short acknovledge. Úspěšné provedení konfigurace zjistí master následným zasláním telegramu Slave_Diag.

Ještě se však vraťme k parametru, který určuje konzistenci dat. Pokud pracujeme s daty jako je byte či word např. na 16-ti bitových procesorech, dochází k těmto operacím v rámci jednoho strojového cyklu. Větší datové úseky však není možné zpracovávat jedinou instrukcí. V tomto okamžiku nastává problém. Pokud nastane přerušení způsobené požadavkem na komunikaci a dojde k přístupu k těmto rozpracovaným datům, pak tato situace způsobí, že přečteme část předešlých a část nových dat. Tím dojde k přenesení nekorektních údajů, které

mohou způsobit vážnou chybu v systému. Proto je zde příznak konzistence, který indikuje, že data musejí být přenášena najednou a systém musí zajistit jejich konzistenci.

3.3.3.5 Globální řízení

Zařízení master může zasílat telegramy jednak konkrétnímu zařízení, které je určeno svou adresou danou při inicializaci tak i více zařízením najednou. To znamená všem zařízením na sběrnici nebo skupině zařízení určené adresou Group_Ident zaslanou při parametrizaci. Mezi příkazy globálního řízení patří sync/unsync, freeze/unfreeze, clear. Na tyto telegramy zařízení slave neodpovídají. Požadované režimy podřízených zařízení zasílá mater v prvním datovém bytu jak je naznačeno na obrázku níže. V druhém se nachází identifikační číslo skupiny.



Režimy:

Clear – Při zaslání tohoto telegramu, přepne slave své výstupy do definovaného stavu. Používá se například pro havarijní události

Freeze – Při zaslání tohoto telegramu si zařízení slave přečte aktuální hodnotu svých vstupů a tuto hodnotu drží až do příchodu příkazu **Unfreeze**.

Sync – Jedná se o jistou obdobu příkazu Freeze ale vztaženou k výstupům zařízení. Při příchodu tohoto příkazu podřízené zařízení počká na příchod dat od stanice master. Tyto data však nezapíše na výstup okamžitě ale až v době příchodu příkazu **Unsync**. Tato funkce je určena k synchronizaci výstupů daných zařízení.

3.3.4 Konfigurace systému Profibus DP/PA

V předchozí kapitole jsme popsali základní principy komunikace na sběrnici Profibus. Nyní naznačme konfiguraci celého systému z pohledu zařízení Master Class 2. Každé zařízení na Profibusu je charakterizováno svým takzvaným GSD souborem (Master Class 1 a slave). Jedná se o soubor s příponou *.gs?, kde otazník znamená jazyk, kterým GSD soubor napsán.

? = d – default (jazykově nezávislý)
? = e – anglický jazyk
? = g – německý jazyk
? = f – francouzský jazyk
…

Tento soubor obsahuje informace nutné pro konfiguraci konkrétního zařízení na Profibusu. Každému zařízení je přiřazen GSD soubor pomocí takzvaného PNO identifikačního čísla. Toto číslo je ověřováno během parametrizace nebo změny adresy, jak bylo vysvětleno v předchozí kapitole. Tento soubor dále obsahuje název zařízení, verzi jak hardwaru tak i software, informace o podporovaných komunikačních rychlostech a maximální hodnotu TSDR. Dále obsahuje uživatelská parametrizační data, konfigurační data a další volitelné parametry.

4. Profily

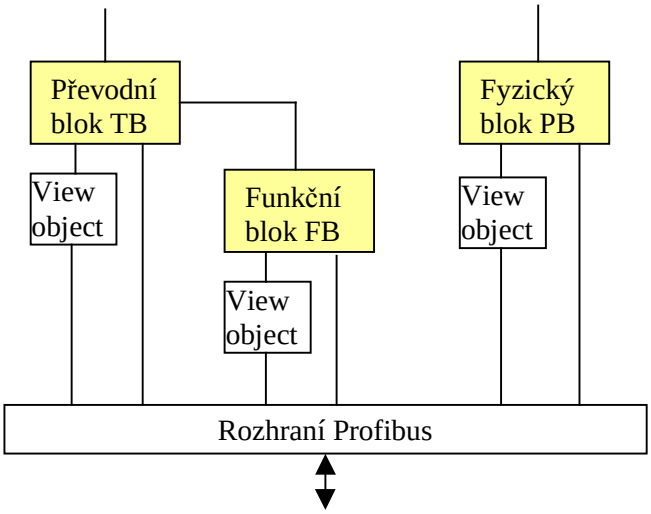
4.1 Úvod

Asi nejzajímavější vlastností Profibusu je, že umožňuje zaměnit libovolné zařízení stejným typem od jiného výrobce bez nutnosti úpravy softwaru. Toho je docíleno tím, že je zaveden standard (**profil**) pro typická zařízení používaná v průmyslu. Ten specifikuje vrstvu, která je v ISO/OSI modelu nadřazena sedmé, nejvyšší vrstvě. Jedná se o konstanty a proměnné (parametry) zařízení typu slave které musí povinně zpřístupnit nadřazenému zařízení. Parametry jsou sdruženy do bloků a jejich velikost a umístění je specifikováno normou [3]. Profil byl vydáván organizací PNO (Profibus Nutzer Organisation) v Německu, nyní Profibus International. V současné době je k dispozici verze 3.

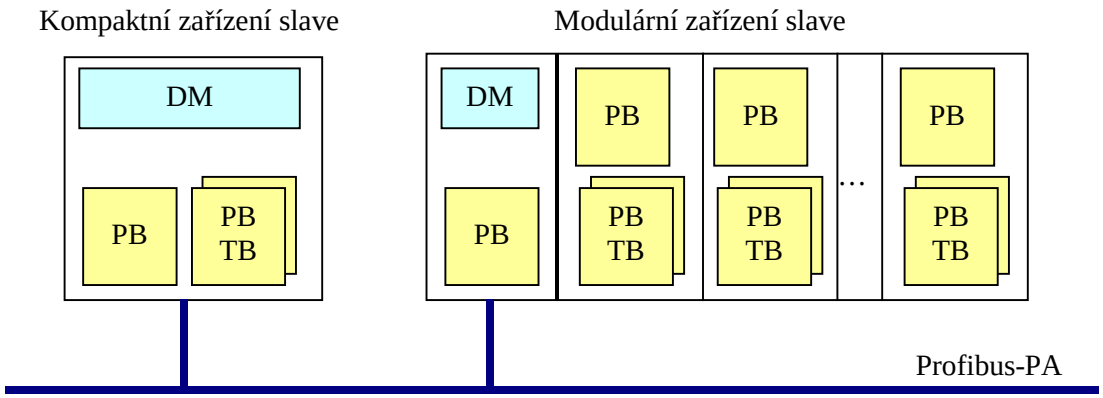
4.1.1 Model a základní myšlenka zařízení Profibus-PA

Profil definuje dvě třídy A a B. Třída A definuje společné parametry, třída B je pouze jejím rozšířením jako jsou parametry pro údržbu a diagnostiku. Samotný profil má adresářovou strukturu. Na vrcholu této struktury je tzv. Device Manager DM. Ten obsahuje odkazy na jednotlivé části profilu – bloky (blocks). Rozeznáváme tři základní typy bloků:

- **Fyzický blok** (*Physical Block*) **PB** – Obsahuje především údaje hardwarového rázu. Jsou zde informace nezbytné pro vlastní identifikaci zařízení, informace o aktuálním stavu a základní nastavení.
- **Převodní blok** (*Transducer block*) **TB** – Je blok, který přistupuje přímo k vlastnímu I/O v zařízení slave. Jeho údaje jsou často přímo ve strojových jednotkách. Jeho úkolem je zprostředkovat vzájemnou komunikaci mezi I/O a funkčním blokem (viz obr. 4-1). Obsahuje také většinu informací přímo se týkajících I/O zařízení
- **Funkční blok** (*Function Block*) **FB** – Jeho hlavním úkolem je vzájemný převod mezi převodním blokem a daty posílanými na sběrnici. Provádí změny měřítek a korekci dat.



Obr. 4-1 Profil - Vnitřní struktura



Obr. 4-2 Kompaktní a modulární zařízení

Takzvané kompaktní zařízení obsahuje pouze jeden fyzický blok FB a jeden či několik funkčních a převodních bloků FB a TB (viz obr. 4-2). Zařízení členíme podle uspořádání profilů na kompaktní nebo modulární. Kompaktní zařízení jsou chápána jako jednoduchá zařízení, u které se nepředpokládá zpracování veličiny různými způsoby. Jsou to hlavně zařízení s jedním senzorem, ale ani tento fakt nemusí být rozhodující, protože můžeme požadovat zpracovávat jednu hodnotu několika předem nastavenými způsoby.

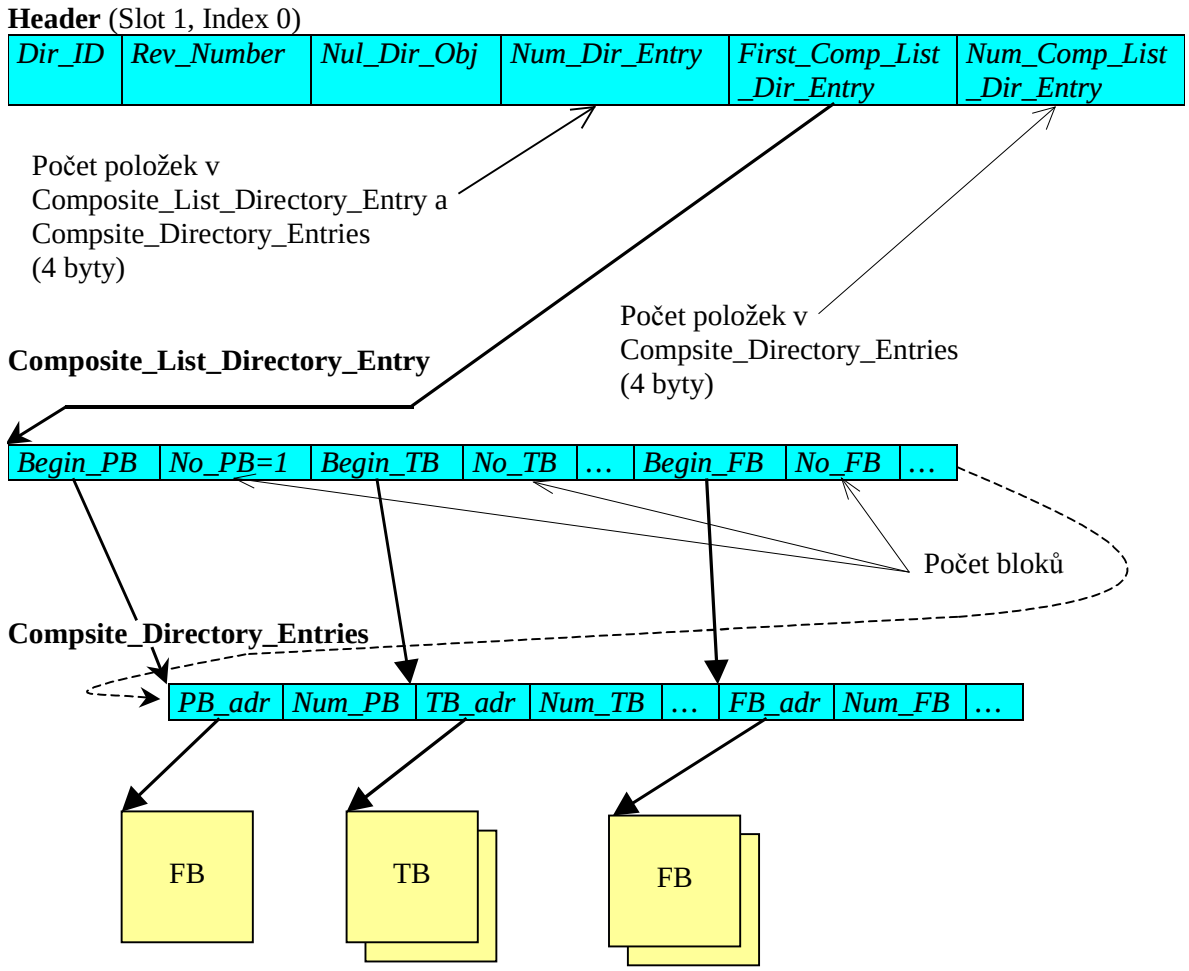
Typickým zástupcem modulárních zařízení jsou logické I/O. U těchto zařízení se předpokládá možnost rozdělení dat na podmnožiny a následné rozdílné zpracování. Který modul bude aktuálně využíván se určuje zasláním konfiguračního telegramu během inicializace na sběrnici (viz. kapitola 3.3.3.4).

4.1.2 Device manager

Jak již bylo uvedeno v předchozí kapitole, strukturu bloku zastřešuje tzv. Device Manager DM. V něm jsou uloženy informace o umístění jednotlivých bloků. Skládá se z:

- Header – Hlavička.
- Composite_List_Directory_Entries – Souhrnné informace o blocích.
- Composite_Directory_Entries – Ukazatele na umístění bloků a informace o počtu parametrů.

Vlastní adresace je reprezentována dvojicí čísel tzv. index, slot. viz kapitola 4.1.4. Její strukturu nejlépe znázorňuje obrázek 4-3. Toto téma je dále rozvedeno v příloze III, kde je uvedena implementace profilu na konkrétním zařízení včetně vysvětlení jednotlivých konstant a proměnných.



Obr. 4-3 Device Manager

4.1.3 Propojení mezi jednotlivými bloky

Na obrázku 4-3 je znázorněna nejednodušší struktura vazeb mezi bloky. Během přenosu dat mezi IO a sběrnicí je propojen vždy jeden převodní TB a jeden funkční blok FB. Jejich vazba je určena parametrem *chanel* ve funkčním bloku FB.

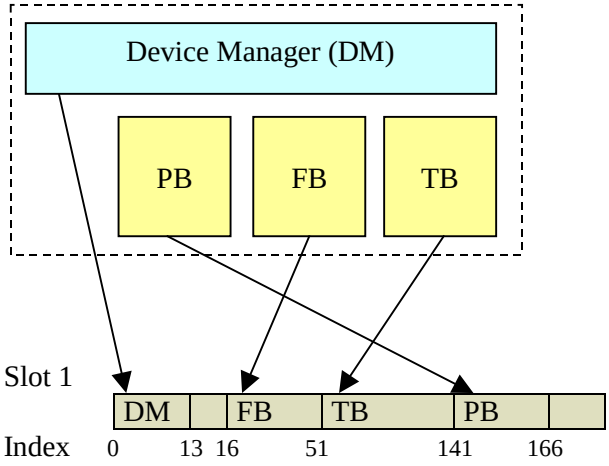
Vazba mezi funkčními bloky FB, například FB typu vstup a FB typu vstup je zajištěna přes takzvaný Link Object LO. Každé vazbě přísluší právě jeden LO. Jejich umístění je dáno v Device Manager stejně jako pro bloky PB, FB a TB.

Ale vraťme se ještě v krátké poznámce k obrázku 4-1. Zde jsou nakresleny přímá datová propojení mezi rozhraním Profibus a všemi bloky přes tzv. View Object. Jedná se o část bloků, které obsahují data, která jsou často čtena. Pomocí View Objectu můžeme přečíst všechny konstanty a proměnné které jsou v něm obsaženy jediným požadavkem na čtení. To zjednodušuje a zrychluje komunikaci.

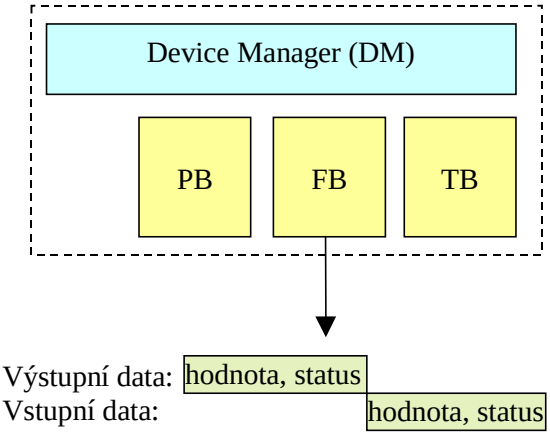
4.1.4 Mapování profilu na Profibus-DP

Komunikaci na Profibusu můžeme rozdělit na takzvanou cyklickou a acyklickou. Pokud chce master přistupovat ke všem konstantám a proměnným v zařízení slave, využívá k tomu takzvanou acyklickou komunikaci. K adresaci se zde využívá dvojice slot, index. Obě adresy v rozsahu 0 až 254 a tudíž celkem 65025 možných parametrů. Tato adresa jednoznačně definuje, ke které konstantě či proměnné chceme přistupovat. Je však ještě potřeba znát její délku. To již je záležitostí jednotlivý zařízení na sběrnici aby věděli, jak dlouhá data mají přijmout či poslat. Acyklická komunikace se využívá především ke konfiguraci stanice slave a tudíž je převážně záležitostí zařízení Master Class 2.

Podmnožinu všech parametrů tvoří parametry určené pro cyklickou výměnu. Jedná se o vstupní či výstupní parametry funkčního bloku na straně Profibusu. Zasláním konfiguračního telegramu zařízení master nakonfiguruje své podřízené zařízení, čímž mu sdělí, které parametry z množiny parametrů určených pro cyklickou výměnu si budou vzájemně posílat. Pokud konfigurace dopadne úspěšně, zahájí vzájemnou cyklickou komunikaci. Toto je hlavně záležitostí zařízení Master Class 1.



Obr. 4-4 Mapování při acyklické komunikaci



Obr. 4-5 Mapování při acyklické komunikaci

Specifikace profilu definuje následující požadavky na mapování profilu na modul Profibus-DP.

- Každý FB leží ve vlastním slotu.
- Pokud zařízení obsahuje více než 1 FB, jsou mapovány sekvenčně za sebou. Výjimku tvoří poslední pravidlo.
- Všechny FB začínají na indexu 16 v daném slotu.
- Header začíná vždy na slotu 1, indexu 0.
- PB může být umístěn ve slotu 1, pro kompaktní(jednoduché) zařízení.
- PB může být umístěn ve slotu 0, pro modulární zařízení kvůli oddělení alarmů mezi FB a PB.

- Pokud zařízení obsahuje více než 1 FB, jsou data při cyklickém telegramu čtená ve stejném pořadí tak, jak jsou uspořádány ve struktuře Composite Directory Entry.
- Umístění TB ve slotu je na volbě výrobce. Pokud zařízení obsahuje jeden FB a TB, je doporučeno umístit tyto bloky do jednoho slotu.
- Umístění PB u zařízení s více jak jedním PB záleží na výrobci.
- Nepoužívané FB jsou reprezentovány jako identifikátor prázdného slotu v konfiguračním řetězci.
- FB, které alokují více než jeden slot, jsou reprezentovány jako identifikátor prázdného slotu pro další sloty.

4.2 Parametry objektů

4.2.1 Rozdělení parametrů

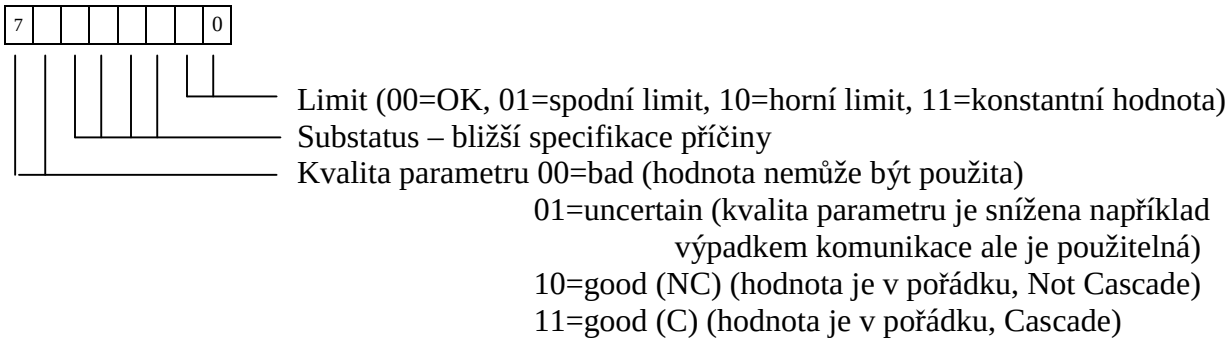
Každý blok se skládá z parametrů. Jak již bylo uvedeno podle implementovaných parametrů dělíme zařízení na třída A a třída B. Třída A popisuje společné parametry pro jednoduchá zařízení. Možnosti těchto zařízení jsou omezena na základní funkce. Třída B rozšiřuje možnosti třídy A a přidává další skupiny parametrů např. pro identifikaci, údržbu a diagnostiku. Každý blok obsahuje minimálně sedm (funkční blok osm) standardních parametrů. V závislosti na třídě pak jsou parametry povinné nebo nepovinné. Blok dále může obsahovat parametry navržené výrobcem, které nejsou součástí profilu. Ty se nacházejí na konci bloku.

Každý parametr bloku má svůj specifický význam a na základě toho ještě rozlišujeme parametry vstupní (input), výstupní (output) a obsažený (contained).

- Vstupní parametr - Je parametr který je zapisován zvenčí bloku. Jeho hodnota je vyžívána vnitřním algoritmem bloku.
- Výstupní parametr - Výstupem vnitřního algoritmu bloku je výstupní parametr. Ten může být zaveden na nějaký vstupní parametr jiného bloku
- Obsažený parametr - Jedná se o parametr jehož hodnota je konfigurována operátorem či nadřazeným zařízením. Tento parametr by neměl být zaveden na vstup či výstup jiného bloku.

O kvalitě vstupních a výstupních parametrů vypovídá jejich status, poslední byte, kde je uložena tato informace. Tento byte je definován pro všechny parametry stejně.

status



4.2.2 Mód bloku

Jak jsme se již zmínili v předchozí kapitole o rozdělení parametrů, každý blok obsahuje sedm (funkční blok osm) standardních parametrů. Nejpodstatnější jsou však dva. Prvním je takzvaný Block_Object. Jedná se o úplně první parametr, který obsahuje potřebné informace k identifikaci bloku jako je typ bloku (PB, FB nebo TB), verze profilu, adresu View_Objectu atd.

Druhým parametrem, přesněji dvojicí parametrů jsou Target_Mode a Mode_Blk. Tyto dva parametry určují v jakém režimu (modu) bude daný blok pracovat. V revizi 3.0 rozeznáváme tyto následující režimy bloku:

- Out of service (O/S) – Blok není vyhodnocován. Výstupní hodnota je udržována na poslední hodnotě či hodnotě nastavené operátorem pro případ selhání.
- Local Override (LO) - Blok není vyhodnocován. Podobný režim jako Man.
- Manual (Man) – Blok není vyhodnocován a výstupní parametr je zapisován operátorem.
- Automatic (Auto) – Blok je vyhodnocován na základě vstupních parametrů.
- Remote Cascade (Rcas) - Blok je vyhodnocován na základě parametrů RCas.

Parametrem Target_Mode operátor oznamuje, v jakém režimu má daný blok pracovat. Vedle toho parametr Mode_Blk obsahuje informace o tom v kterém režimu blok skutečně pracuje, které režimy jsou přípustné a který režim je požadován pro normální práci bloku.

4.3 Aplikace profilu

V předchozí kapitole jsme se pokoušel objasnit základní principy a myšlenky profilů. Probral jsme strukturu, obecné požadavky a návaznost na služby sběrnice Profibus. Nyní se zaměřme na konkrétní aplikaci. Norma se snaží ujednotit parametry využívané podobnými typy zařízení a z toho pohledu je rozdělujeme na:

- Digitální I/O
- Snímače (tlaku, teploty, průtoku, …)
- Akční členy (elektrické, elektro-pneumatické, elektro-hydraulické)
- Analyzátory (analogové, digitální)
- Konfigurovatelné (kombinace předešlých typů)

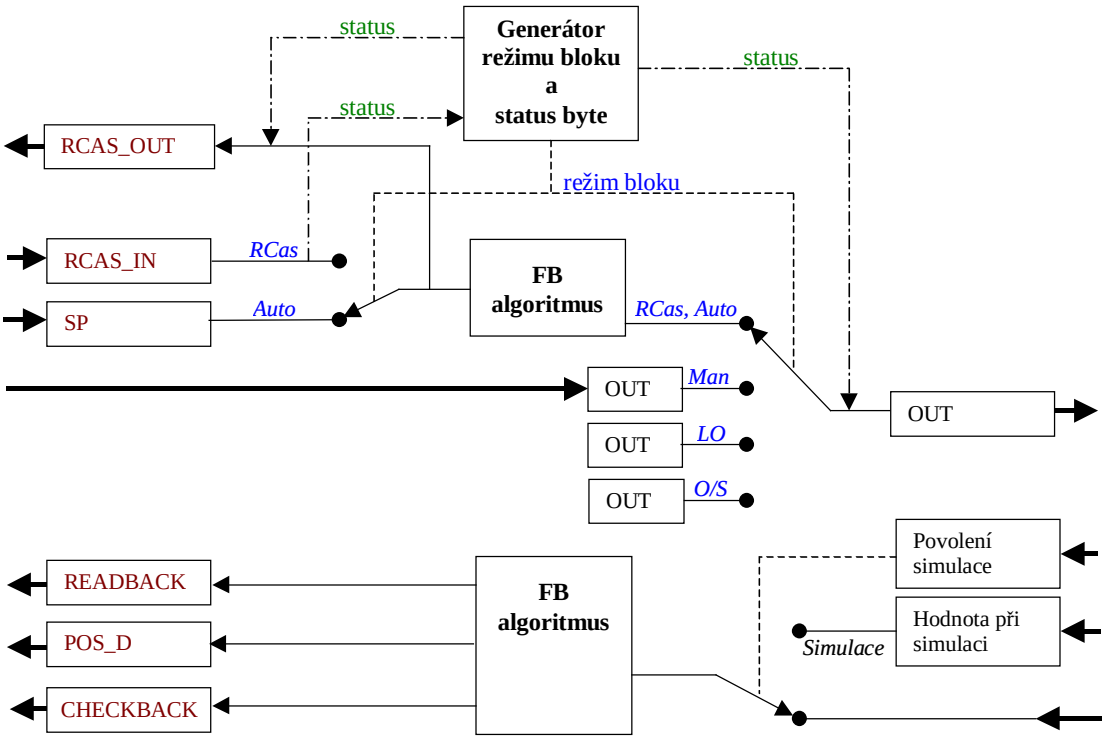
Zaměřme se nyní na akční členy. Norma v revizi 3.0 specifikuje profil pro akční členy s konečným řízením jako jsou polohová serva a ventily. Nezahrnuje zařízení typu ventilátor, čerpadlo nebo rychlostní servo. Proberme nyní rozšiřující parametry funkčního a převodního bloku, jak je definováno pro jednotky pohonu (actuators).

4.3.1 Funkční blok

Funkční blok je stejný pro všechny typy pohonů. Obsahuje šest parametrů které jsou mapovány na cyklickou komunikaci (SP, READ_BACK, RCAS_IN, RCAS_OUT, POS_D, CHECK_BACK). Z toho jsou dva nepovinné (RCAS_IN, RCAS_OUT).

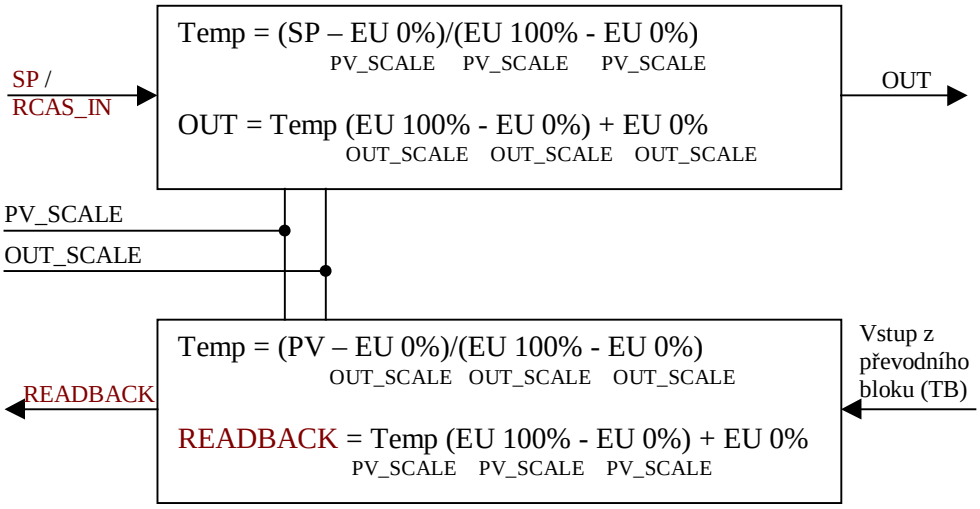
SP – Setpoint. Poloha požadovaná uživatelem v jednotkách daných parametrem PV_SCALE.
READ_BACK – Aktuální poloha také v jednotkách PV_SCALE.
RCAS_IN - Poloha požadovaná uživatelem v jednotkách daných parametrem PV_SCALE a režimu Rcas. Parametr je nepovinný.
RCAS_OUT – Požadovaná poloha v jednotkách PV_SCALE . Parametr je nepovinný.
POS_D – Aktuální pozice - diskrétně hodnota.
CHECK_BACK – Informace o stavu zařízení (bližší informace [3] část 6 str. 13)

Práci funkčního bloku nejlépe popisuje následující model na obrázku 4-5.



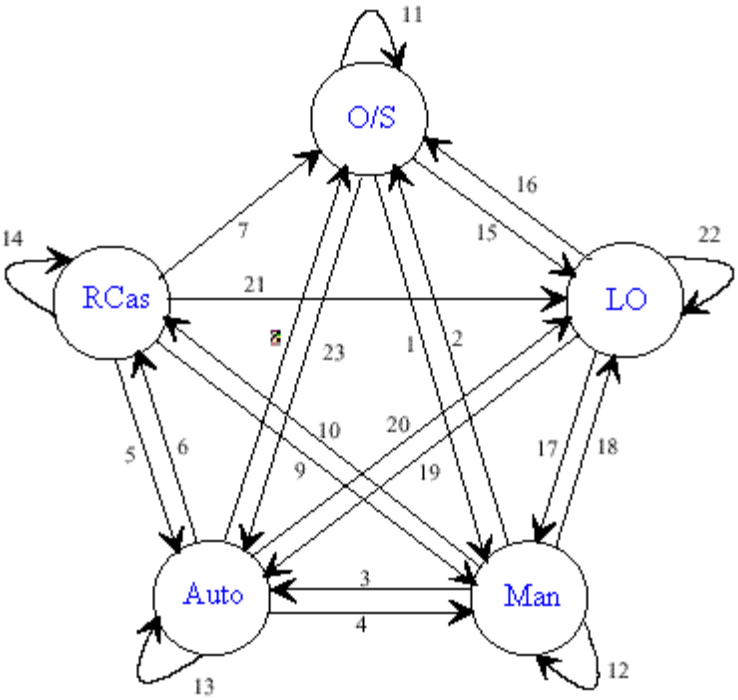
Obr. 4-5 Model funkčního bloku akčního členu

FB - Algoritmus



Obr. 4-6 Algoritmus funkčního bloku

Převod vstupních parametrů na výstupní je závislý na stavu ve kterém se funkční blok aktuálně nachází. Tento stav je ovlivněn jednak požadavkem od operátora (parametr FB - Target_Mode) nebo nadřazeného zařízení ale také na statusu vstupních parametrů. Přechod mezi stavy lze popsat stavovým diagramem na obrázku 4-7 a tabulkou popisující jednotlivé přechody.



Obr. 4-7 Stavový diagram funkčního bloku akčního členu

Přechod	Režim požadovaný operátorem (Target_Mode)	Aktuální režim	Status vstupních parametrů	Status RCAS_IN	Status RCAS_IN != GOOD(C)	Actual_ Mode
2,7,8,11	-	-	!= ok	-	-	O/S
2,7,8,11, 16	O/S	-	ok	-	-	O/S
15,18,20, 21,22	LO	-	ok	-	-	LO
1,4,9,12, 17	Man	-	ok	-	-	Man
3,5,13,19 ,23	Auto	-	ok	-	-	Auto
19	RCas	LO	ok	-	-	Auto
23	RCas	O/S	ok		-	Auto
6	RCas	Auto	ok	good(C)-IA	-	RCas
13	RCas	Auto	ok	!= good(C)-IA	-	Auto
10	RCas	Man	ok	good(C)-IA	-	RCas
12	RCas	Man	ok	!= good(C)-IA	-	Man
14	RCas	RCas	ok	!= good(C)	ne	RCas
5	RCas	RCas	ok	!= good(C)	ano	Auto
14	RCas	RCas	ok	good(C)	-	RCas
5	RCas	RCas	ok	good(C)-IFS	-	Auto
Poznámka: IA a IFS jsou vedlejší příznaky pro kvality Good(Cascade). IA je initialisation acknowledge a IFA je initiate fail safe.						

Tab. 4-1 Popis stavového modelu funkčního bloku z obrázku 4-7

Funkční blok ale také generuje status ke svým výstupním parametrům. O to se stará, jak je znázorněno na obrázku 4-5, blok „Generátor režimu bloku a status byte“. Ten je pak předán Převodnímu bloku (TB).

Aktuální režim	Status SP	Status RCAS_IN	Status OUT
O/S	-	-	bad
LO	-	-	good(NC)
Man	-	-	poslední hodnota
Auto	!=bad & != good(NC)-IFS	-	good(NC)
Auto	bad (fail safe time aktivován)	-	good(NC)
Auto	bad (fail safe time aktivován) | good(NC)-IFS	-	viz příloha 1 (FAIL_SAFE_TYPE)
RCas	-	good (C)	good(NC)
Poznámka: IFS je substatusy pro kvality Good(Cascade). IFA je initiate fail safe.			

Tab. 4-2 Chování generátoru status byte z obr 4-5. v závislosti na SP a aktuálním režimu bloku.

4.3.1 Převodní blok

Verze 3.0 rozlišuje dva typy převodních bloků (TB). Pro elektrický akční člen a pro elektro-pneumatický akční člen. Základní parametry jsou však společné. Výstupní parametr OUT z funkčního bloku (FB) je přenášen do parametru POSITIONING_VALUE v převoním bloku (TB). Hodnota je předávána v jednotkách daných parametrem OUT_SCALE. Naopak údaje o poloze akčního členu jsou přenášeny do funkčního bloku (FB) z parametru FEEDBACK_VALUE v převodním bloku (TB). Opět v jednotkách daných OUT_SCALE. Bližší popis všech parametrů pro zařízení třídy B je uvedeno v příloze III. Bližší informace je možné nalézt ve [3].

5. Návrh hardwaru

5.1 Profibus Development Board

5.1.1 Formulace požadavků

Mým úkolem bylo realizovat modul pro průmyslovou sběrnici Profibus DP/PA umožňující připojení jednoduchých IO zařízení, která neumožňují komunikaci po této sběrnici. Za hlavní cíl mé práce jsem si vytyčil jednoduchost ale na druhé straně univerzálnost. Zvýšený důraz byl přitom kladen na část Profibus PA. Stanovené požadavky lze formulovat následujícími body.

Profibus PA
a) Co nejmenší odběr.
b) Galvanické oddělení od sběrnice.
c) Rozhraní RS232 pro komunikaci s IO zařízeními.
d) Dostatečně velký paměťový prostor.
e) Možnost jednoduché a rychlé aktualizace softwaru.
f) Napájecí napětí 3,3V
g) Provedení SMD.

Profibus DP
a) Snadná rekonfigurace DP/PA.
b) Co nejvyšší možná komunikační rychlost.

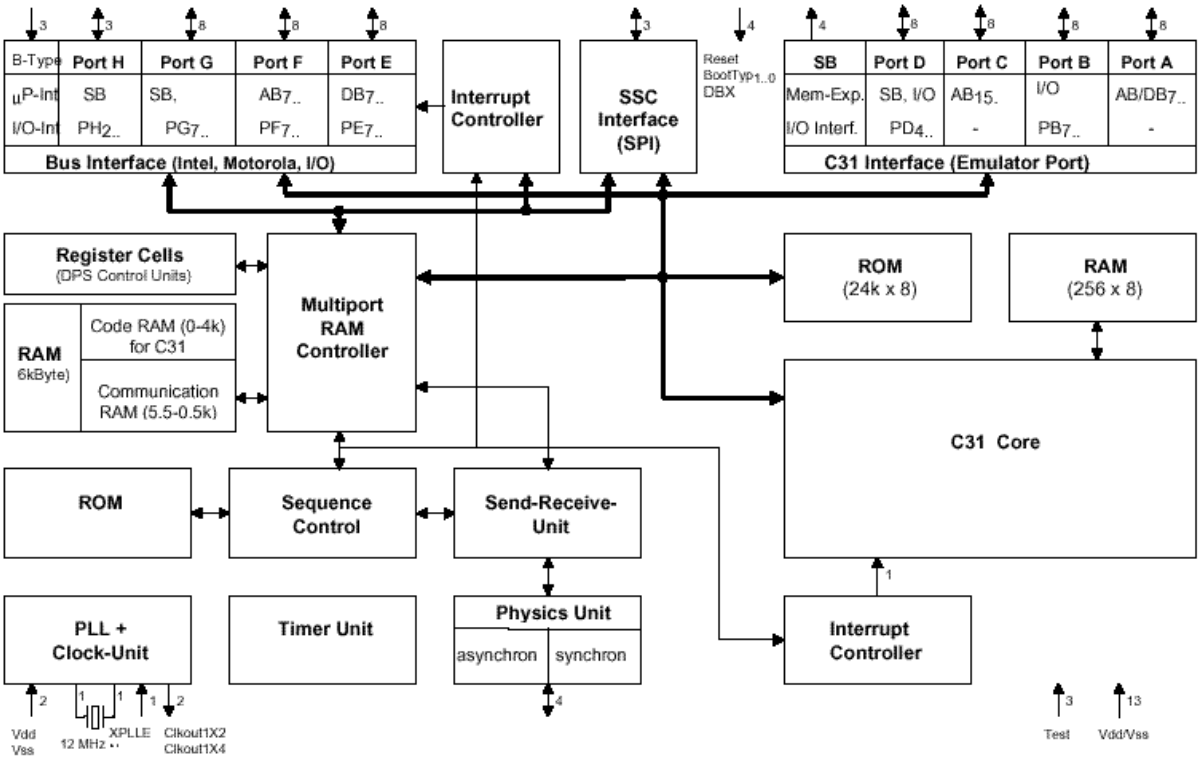
5.1.2 Procesorová část

Modul jsem pracovně pojmenoval PDB - Profibus Development Board. Po zvážení požadavků jsem zvolil za základ tohoto zařízení odvod DPC31 [10]. Jedná se o zákaznický obvod firmy Siemens, který zároveň obsahuje jádro procesoru C31. Navíc obsahuje třetí časovač a 8kB interní datové paměti. Ta je využívána hlavně pro konfiguraci rozhraní a komunikační buffery. Napájecí napětí tohoto obvodu je V_{DD} = 3,3V (±10%) ale napětí na IO signálech může být až V_{DD} + 3,0V. To umožňuje provozovat tento procesor v zapojeních jak s 3,3V tak i 5V logikou. Aby spotřeba obvodu DPC31 byla co nejmenší není tento obvod vybaven vnitřními pull-up rezistory. Tuto skutečnost je nutné patřičně zohlednit při návrhu zařízení.

Tento obvod je uzpůsoben tak, aby mohl pracovat jednak samostatně. To znamená, že je zapnuto vnitřní játro C31 (viz obr. 5-1) na kterém běží uživatelský program. A nebo je toto jádro vypnuto a pomocí vyvedené vnitřní sběrnice (8-bit data, 13-bit adresa) na bránách PE, PF, PG, a PH je možno připojit externí procesor jako je 80C31/32, Motorola HC11 a také procesory 80C166, Intel X86 a Motorola HC16/HC916. Obvod je dále vybaven rozhraním SPI.

Při komunikaci na Profibusu obvod DPC31 pracuje ve dvou modech. **Asynchronní**, kdy oscilátor běží na 12MHz a je zapnuta jednotka fázového závěsu (PLL). Potom obvod interně pracuje na kmitočtu f_{SYS}=48MHz. V tomto módu DPC31 realizuje fyzickou vrstvu (ISO/OSI)

jako RS485 na rychlostech 9,6kbit/s až 12Mbit/s (Profibus DP). Nebo v **synchronním** módu, kdy je jednotka fázového závěsu vypnuta. Oscilátor může běžet na f_{SYS}=16, 8, 4 a 2MHz (2MHz nejsou doporučovány). Vnitřní frekvence je přitom poloviční. Takto DPC31 realizuje fyzickou vrstvu v kódování Manchester II na rychlosti 31,25kbit/s a tudíž Profibus PA. Kmitočet f_{SYS} vydělený dvěma a čtyřmi je k dispozici na signálech CLOCKOUT1X2 a CLOCKOUT1X4. Pro ilustraci jsou průběhy těchto signálů jsou uvedeny v příloze VII (Měření) na obr. VII-1.



Obr. 5-1 Vnitřní blokové schéma obvodu DPC31

V této části návrhu byl kladen důraz na minimalizaci ztrátového výkonu desky. V úvahu připadaly dvě varianty. Buď použít DPC31 pouze jako rozhraní k procesoru řady X51 firmy Atmel s interní pamětí FLASH. Alternativní možností bylo využít interního procesoru v DPC31. Kompilací vzorové aplikace ke knihovně V1SL [6] jsem zjisti, že minimální velikost programové paměti musí být 30kB. To je více nežli velikost interní paměti FLASH v dostupných procesorech X51. Proto v úvahu připadla pouze varianta s interním procesor C31.

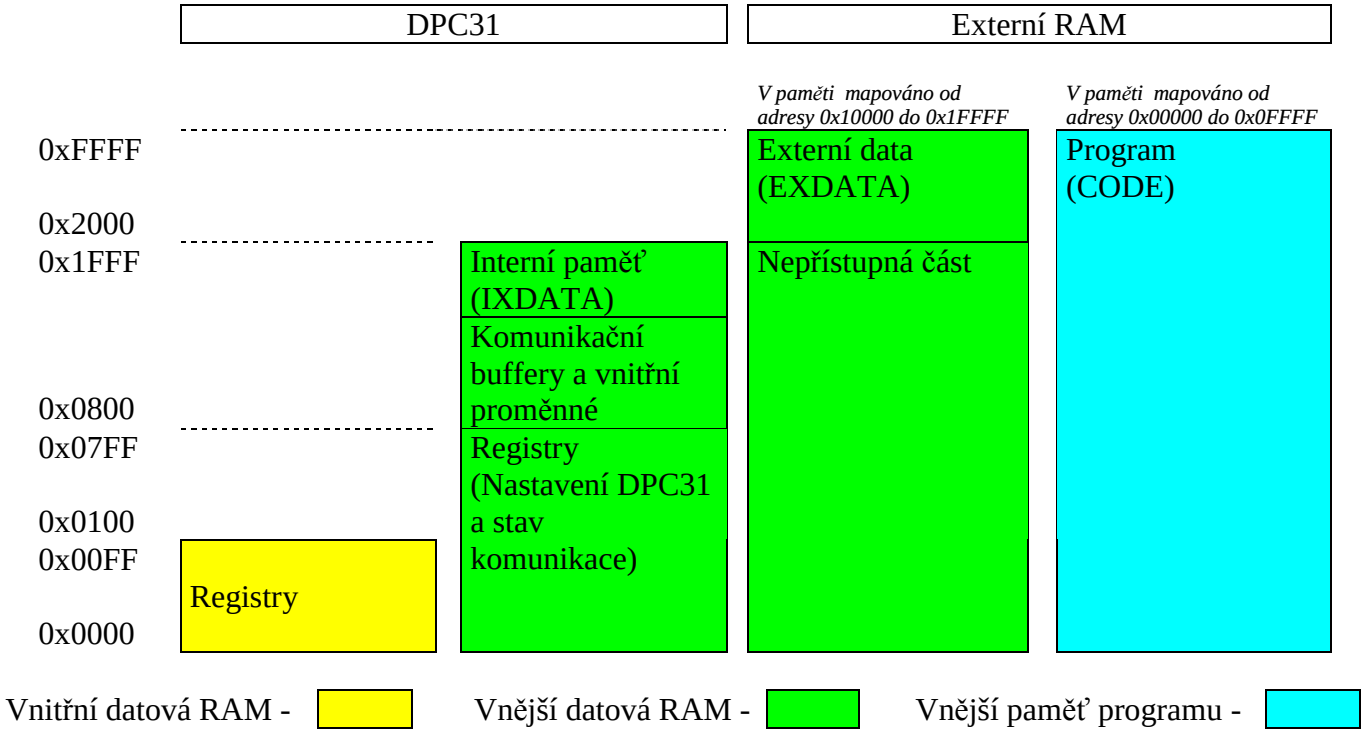
Vzhledem k stanoveným požadavkům, jsem se rozhodl vybavit PDB 64kB paměti pro program a 64kB pro data. Paměťová část je tvořena jedinou pamětí RAM o velikosti 128kB. K nahrání programu je zapotřebí použít externí programátor (viz kapitola 5.2). Spodních 64kB je mapována jako oblast programu a horních 64kB je mapováno jako paměť dat. Konkrétně se jedná o paměť CMOS SRAM BS62LV1024 s velice nízkou spotřebou. Paměť je zálohována lithiovou baterií B-CR2025 (3,0V) pomocí dohlížecího obvodu DS1836C. V případě přerušení napájecího napětí přejde signál RST tohoto obvodu do úrovně logická nula. Tento signál je využit k zákazu přístupu do paměti (signál CE2 paměti) a k přepnutí procesoru do stavu reset. Mnou použitý dohlížecí obvod také dovoluje monitorovat napájecí napětí a včas varovat před jeho výpadkem. Měřené napětí je přes odporový dělič zavedeno na analogový vstup IN. Zde je monitorován

pokles napětí pod úroveň 1,25V. V případě poklesu měřeného napětí pod tuto hodnotu přejde výstup NMI do úrovně log. 0, čímž generuje signál PWR_FAIL. Ten je možno pomocí zkratovací propojky (SW8) přivést na přerušení číslo 1 (XINT1) obvodu DPC31. Napájecí napětí procesorové části PDB je stabilizováno obvodem LM1085IS-3,3. Jedná se o stabilizátor s malým úbytkem napětí (min. 1,5V). Zapojení s dohlížecím obvodem monitoruje pokles vstupního napětí pod úroveň 4,75V. Potom pro vstupní dělič R_1 a R_2 musí platit:

$$4{,}75 = \frac{R_1 + R_2}{R_1} \cdot 1{,}25$$

Při volbě odporu R_1 = 100kΩ je odpor R_2 na základě tohoto vztahu R_2 = 280kΩ.

Vraťme se k paměťovému prostoru procesorové části. Na následujícím obrázku je znázorněn paměťový prostor PDB. Mapování externí paměti RAM je řešeno signály XCSCODE a XCSDATA z obvodu DPC31. Pokud procesor přistupuje do paměti programu, je signál CSCODE v úrovni logická 0. Obdobně pokud se jedná o přístup do paměti nad 0x2000 (nad 8kB interní paměti dat), signál XCSDATA je v úrovni log. 0. Signál XCSCODE je zaveden na nejvyšší bit paměti RAM (A16), čímž je vyřešeno rozdělení paměti na datovou a programovou část.



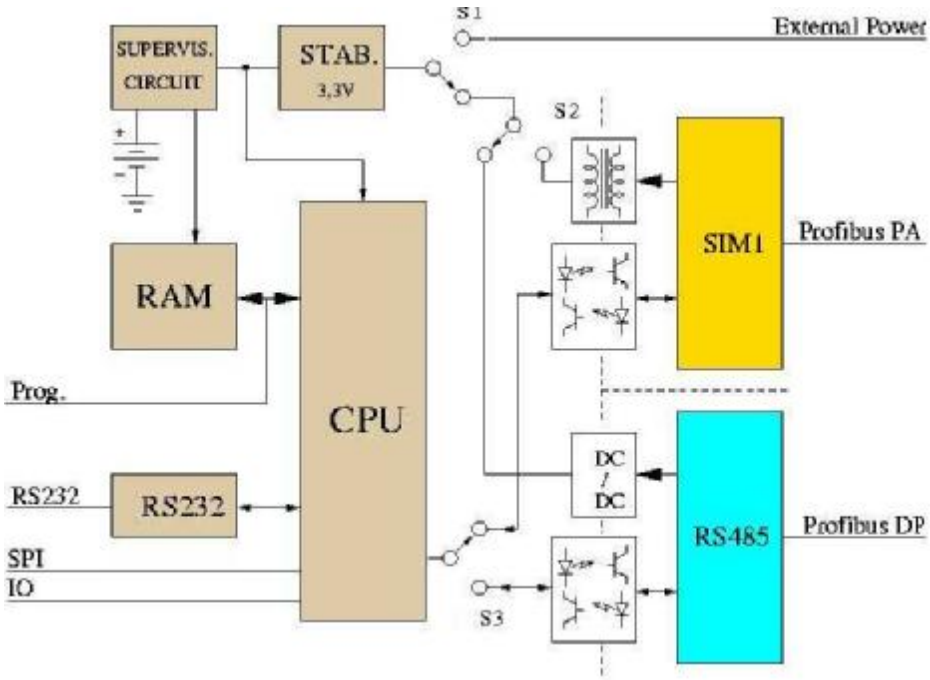
Obr. 5-2 Mapování paměťových prostor v PDB

Na tomto místě je nutné zdůraznit, že oba signály XCSCODE a XCSCODE musí být zapojeny aby nedocházelo ke konfliktům s pamětí uvnitř obvodu DPC31. Konkrétně v mém případě jsem tento konflikt na sběrnici vyřešil zavedením logického součinu signálů XCSCODE a XCSCODE do vstupu $\overline{CE1}$ paměti RAM. Skutečnost, že v systému je fyzicky pouze jedna

paměť jak pro data tak pro program je vyřešena logickým součinem signálů RD a PSEN. Pro lepší pochopení odkazuji opět na přílohu VII (Měření) na obrázky VII-2 – VII-5, kde jsou uvedeny průběhy signálů XCSCODE a XCSDATA.

5.1.3 Rozhraní

V předchozí kapitole byl popsán návrh procesorové části PDB. Pro lepší názornost uvádím na následujícím obrázku blokové schéma, které je výsledkem předchozího návrhu a požadavků formulovaných v kapitole 5.1.1.

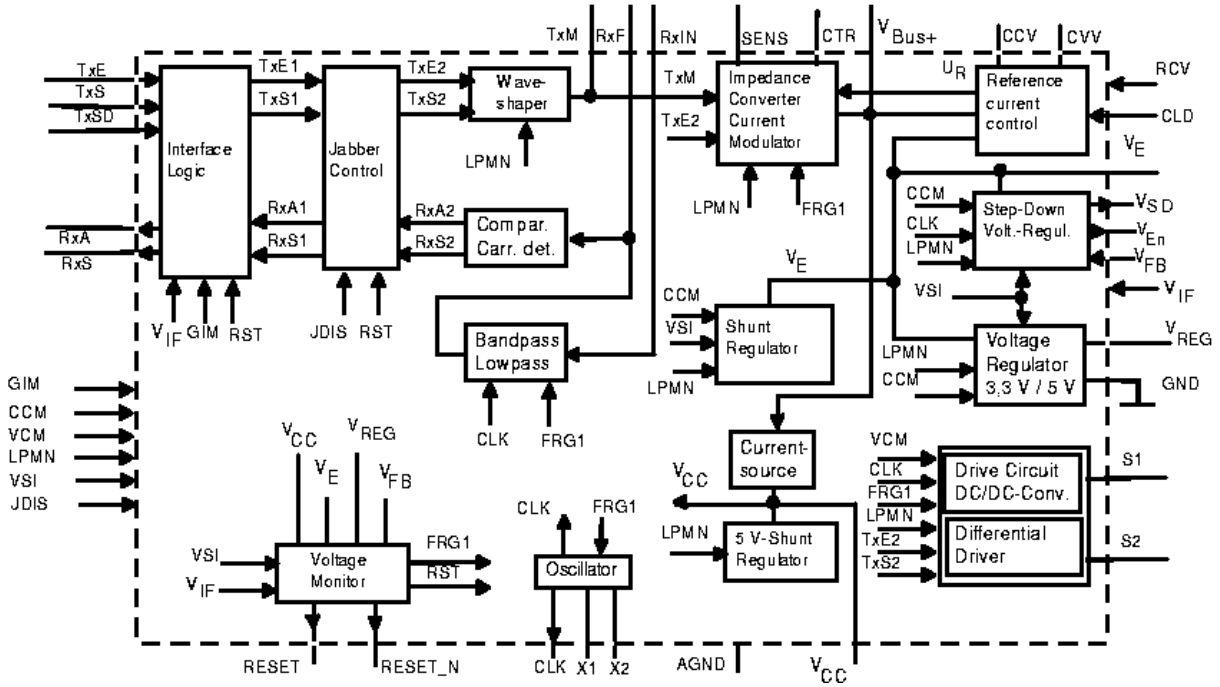


Obr. 5-3 Blokové schéma

V této části bych se rád věnoval popisu rozhraní použitých v mém návrhu. Rozhraní RS232 tvoří obvod MAX3232. Jedná se o variantu klasického budiče sběrnice RS232, která pracuje již při napájecím napětí 3,0V (max. 5,5V). Další výhodou je to, že tento obvod nevyžaduje tak velké externí kondenzátory jako běžné obvody pro RS232 (pouze 0,1μF).

Rozhraní Profibus PA je tvořeno obvodem SIM1 [11] firmy Siemens (blokové schéma viz obr. 5-4). Ten umožňuje příjem a vysílání na sběrnici rychlostí 31,25kbit/s. Dále generuje napájecí napětí ze sběrnice (až 40mA) a obsahuje obvod pro řízení DC/DC převodu tohoto napětí. Obvod může pracovat ve třech základních režimech:

1) Konstantní proud – sběrnice je zatěžována konstantním proudem (jednoduché zapojení).
2) Konstantní výkon – sběrnice je zatěžována konstantním výkonem (minimální ztráty)
3) Lokální napájení – obvod není napájen ze sběrnice.



Obr. 5-4 Vnitřní blokové schéma obvodu SIM1

Na desce PDB je použito základní zapojení v režimu odběru konstantního proudu s galvanickým oddělením pomocí čtyř optočlenů. Tzn. signály TxE, TxSD, RxA a RxS jsou k obvodu DPC31 připojeny přes oddělovací optočleny. Odběr ze sběrnice je nastaven odporem R_{15}, který je připojen mezi signál CCV a zem. Pro proud odebíraný ze sběrnice přibližně platí:

$$I_{BUS} = \frac{R_{15}}{10000} \; [mA]$$

Velkou výhodou použitého zapojení je, že vstupní signál ze sběrnice je přiveden přes diodový můstek a proto nezáleží na polaritě vstupního signálu. Tím je také obvod SIM1 chráněn proti zničení záporným napětím.

Za základ rozhraní Profibus DP mi sloužilo výrobcem doporučené zapojení obvodu 75176 (transiever TTL/RS485) a DPC31. Toto zapojení bylo navíc zjednodušeno podle katalogového zapojení optočlenů HCPL-0721 firmy Agilent [19]. O galvanické oddělení napájecího napětí se stará monolitický DC/DC měnič CDDDW1-0505S.

Konfiguraci PDB je vyřešena zkratovacími propojkami. SW10-SW13 přepínají vstupní signál do DPC31. Buď je připojen obvod SIM1 a jedná se o Profibus PA, nebo je připojen budič 75176 a jedná se o Profibus DP. Přepínače SW6 a SW7 určují, co bude bráno jako zdroj napájecího napětí. SW1 až SW5 konfigurují obvod DPC31 jako zařízení slave Profibus DP nebo PA. Viz následující tabulka.

Přepínač	Popis	DP	PA
SW1	Povolení / zákaz fázového závěsu PLL	1 2 3	1 2 3
SW2	V_{DD} fázového závěsu PLL	1 2	1 2
SW3	GND fázového závěsu PLL	1 2	1 2
SW4	Krystal 4MHz	1 2	1 2
SW5	Krystal 12MHz	1 2	1 2
SW6	Napájení z DP / externí napájení	1 2 3 nebo 1 2 3	1 2 3
SW7	Napájení PA / z SW6	1 2 3	1 2 3 nebo 1 2 3
SW8	Včasné varování před výpadkem napájení		
SW9	Reset procesoru		
SW10	RTS – Povolení vysílání (asynch.) / TxE - Povolení vysílání (synch.)	1 2 3	1 2 3
SW11	TxD – Vysílaná data (asynch.) / TxS - Vysílaná data (synch.)	1 2 3	1 2 3
SW12	XCTS – Vysílání povoleno (asynch.) / RxA – Pomocný signál při příjmu (asynch.)	1 2 3	1 2 3
SW13	RxD – Přijímaná data (asynch.) / RxS – Přijímaná data (synch.)	1 2 3	1 2 3

Tab. 5-1 Hardwarová konfigurace PDB

Výsledkem mého návrhu je schéma, které je uvedeno v příloze I.

5.2 ProfiProg

Protože je v PDB použita pouze statická RAM zálohovaná baterií, bylo nutné vyřešit nahrávání programu do této paměti. V rámci své práce jsem navrhl programátor, pomocí kterého je možné nahrát program. Protože PDB pracuje s napájecím napětím 3,3V, bylo nutné, aby tento programátor pracoval také na tomto napájecím napětí. Jeho jádrem je jednočipový mikroprocesor firmy Atmel AVR 90S8515 [20] (dále řídící procesor).

Myšlenka programátoru je taková, že připojením programátoru k PDB (přes konektor J1) se připojí napájecí napětí 3,3V na desku programátoru. V tom okamžiku dojde k přepnutí

interního procesoru v DPC31 na PDB do stavu reset. A zároveň je také obvod DPC31 přepnut do režimu debug, aby se směry signálů na sběrnici (jako je PC, PA, ALE, PSEN, XRD, XWR) přepnuly na vstupní. Tím dojde k uvolnění sběrnice na PDB a je možno přistupovat do paměti RAM. Pak stačí pouze ovládat příslušné signály pro čtení a zápis do paměti a je možno nahrát požadovaný program. Ten je z počítače předán přes sériovou linku. Informace o průběhu programování je indikována LED diodou D7 (viz schéma v příloze II) ovládaná signálem MODE.

Jelikož mi nebyl k dispozici programátor obvodů Atmel AVR, rozhodl jsem se vybavit toto zařízení také jednoduchým obvodem, který umožní naprogramovat řídící procesor. Zde mi byla inspirací konstrukce univerzálního programátoru PonyProg viz [25]. Přepínač SW1 určuje, zda bude procesor při programování napájen přímo ze sériového portu (1 a 2 spojeny), nebo bude napájen z externího zdroje (2 a 3 spojeny). Schéma programátoru ProfiProg je uvedeno v příloze II.

5.3 Realizace a Oživení

5.3.1 Návrh plošného spoje a součástková základna

Návrh plošného spoje je proveden v systému OrCAD 7.1. Motivy plošných spojů jsou uvedeny v příloze a jejich elektronická podoba se nachází na CD, které je přiloženo k této práci. Součástky jsem se snažil volit tak, aby byly co možná nejdostupnější. Bohužel požadavek na napájecí napětí 3,3V a provedení SMD značně zužilo a znesnadnilo výběr. Většina součástek pochází ze sortimentu prodejny GM Electonics [30]. Obvody DPC31 a SIM1 byli dodány mým vedoucím diplomové práce. Nevětší problém byl se získáním paměti RAM, která byla nakonec zakoupena ve firmě Ryston Electronics [32]. Obvod MAX3232 byl společně s miniaturním SMD transformátorem zakoupen ve Spezial Electornics [31]. Dohlížecí obvod DS1836C jsem získal jako vzorek od Spoerle Elecronics [33]. Bylo by však vhodné použít jiný, lépe dostupný obvod. Plošný spoj pro programátor ProfiProg byl vyroben firmou Vladimír Kohout [34] a PDB v Plošné spoje Korunní [35].

5.3.1 Oživení

5.3.1.1 ProfiProg

Jako první jsem osadil programátor ProfiProg. Procesor a MAX3232 je osazen do jednořadých precizních patic, aby obvody mohly být případně vyměňovány. Poté jsem připojil na konektor JP1 napájecí napětí a zkontroloval odběr desky a funkci oscilátoru procesoru. Následně jsem začal zkoušet programování řídícího procesoru. Přepínačem SW1 jsem desku přepnul do režimu externího napájení a připojil ji konektorem P1 k počítači. Zde se objevila první chyba. Při návrhu plošného spoje jsem přehlédl zrcadlové očíslování signálů na konektorů CANNON. U konektoru P1 byla tato závada vyřešena úpravou kabelu a u konektoru P2 úpravou přímo na plošném spoji. Funkčnost prověřil jednoduchý program, který prováděl komunikaci po sériové lince přes konektor P2. Ještě bych rád upozornil, že jako řídící procesor byl použit AT90S8515-8,

což není verze, která pracuje na 3,3V. Protože však byl zvolen nižší kmitočet (3,6864MHz), je možné tento procesor také využít.

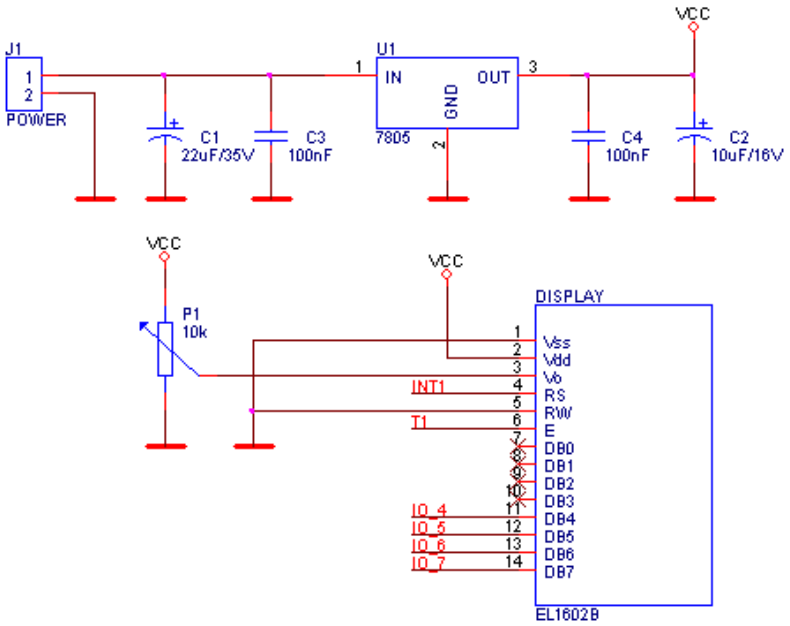
Pro ilustraci uvádím v příloze IX fotografii hotového zařízení.

5.3.1.2 PDB

V tomto okamžiku bylo vše připraveno na oživení hlavní desky – PDB. Osadil jsem desku a zkratovacími propojkami ji nastavil do režimu Profibus PA s externím napájením (viz tabulka 5-1). S tím rozdílem, že přepínače SW10 až SW13 byli ponechány úplně rozpojené. Tím byly vyřazeny z funkce všechny optočleny. Na konektor J2 jsem přivedl napájecí napětí o velikosti 5V a změřil odběr desky, který činil 13mA. Dále bylo potřeba ověřit činnost oscilátoru a průběhy signálů ALE a PSEN na sběrnici. Poté jsem začal snižovat napětí napájecího zdroje, abych ověřil činnost dohlížecího obvodu DS1836C.

Následně mohl být nahrán první program. Tím se objevil další nedostatek návrhu. Programátoru se velice často nedařilo dostat na sběrnici a nemohl provést naprogramování. Z tohoto důvodu bylo potřeba opakovaně připojovat programátor a čekat na úspěšný pokus. Také odběr desky byl během programování nepřiměřeně vysoký (více jak 100mA). Chyba spočívala v tom, že jsem se zpočátku mylně domníval, že aktivováním vstupu RESET obvodu DPC31, dojde k uvolnění sběrnice. Společně s tímto signálem je potřeba přivést na vstup DBX úroveň logická 1, čímž DPC31 přejde do režimu debug. Tím dojde k uvolnění sběrnice a pak teprve programátor může provést přenos dat do paměti RAM.

Další problém byl, že po nahrání prvního programu náhle stoupla spotřeba desky (35mA). Problém byl ve vyvedené vnitřní sběrnici na portech PE, PF, PG a PH. Pokud je využíván vnitřní procesor, je možné tyto signály použít jako IO. Procesor k nim pak přistupuje přes vnější pomět dat umístěné v DPC31. V mé aplikaci jsou tyto signály nevyužity a protože po resetu jsou nastaveny jako vstupní, je možné je přizemnit. Bohužel programátor pro zápis programu do paměti RAM používá signál WR, který je také připojen na XWR obvodu DPC31. Aktivací tohoto signálu dojde k zápisu jak do paměti RAM tak do DPC31. Tím dojde k náhodnému přepnutí

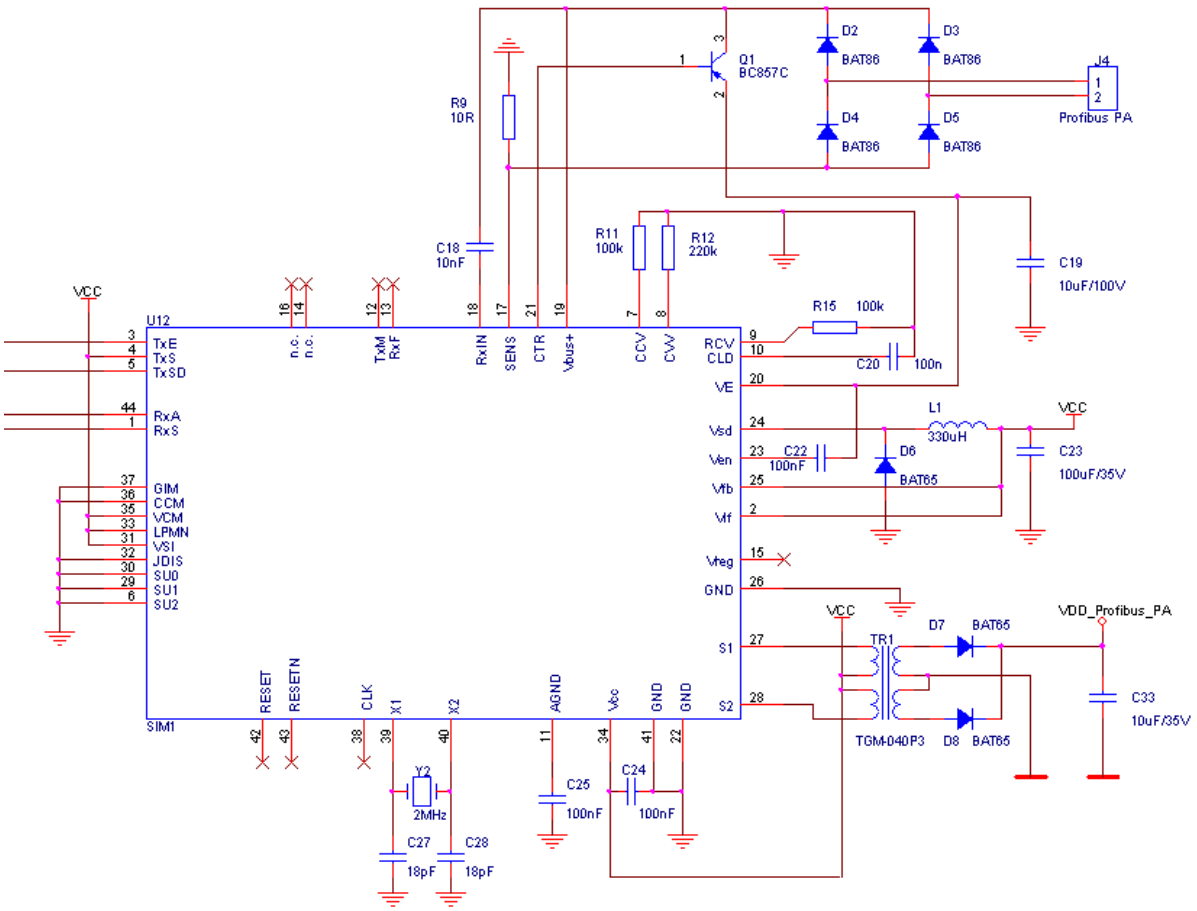


Obr. 5-5 Externí display

některých signálů portu PE až PH na výstupní a k jejich částečnému poškození. Tuto chybu nebylo možné na již vyrobené desce úplně opravit ale funkčnost zařízení to neomezilo.

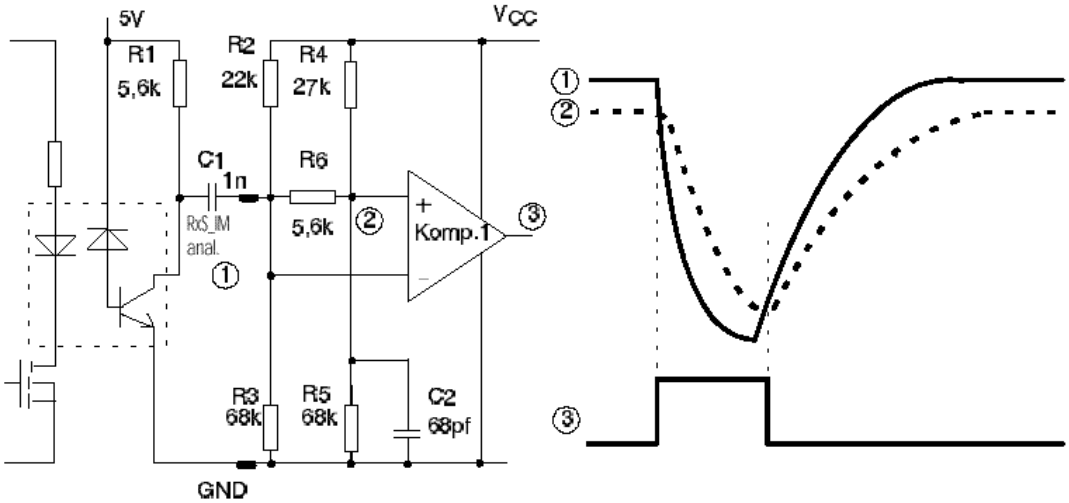
V závěru oživování procesorové části jsem ještě vyzkoušel sériový kanál a komunikaci s externím displejem. Displej je určen pro účely vývoje a je připojen přes konektor J3 pomocí čtyř datových a dvou řídících vodičů.

Následně přišlo na řadu rozhraní Profibus PA. Na konektor J4 jsem připojil stejnosměrné napětí 10V. Následovala kontrola velikosti odběru a funkce oscilátoru. V původním návrhu bylo použito zapojení obvodu SIM1 v režimu, kdy odebírá ze sběrnice konstantní výkon a napájecí napětí pro procesorovou část je odděleno transformátorem. SIM1 obsahuje obvod, který umožňuje řídit transformátorový DC/DC měnič signály S1 a S2 (f=250kHz). Použitý transformátor měl příliš malou indukčnost a nadměrně zvyšoval odběr. To se projevovalo tím, že pokud byl odebíraný proud příliš omezen (odporem R_{11}) napětí na vstupu VE nedosáhlo požadovaného minima 6,6V a monitor napětí (voltage monitor – viz obr 5-4) nezahájí funkci obvodu. Původně navrhované schéma je na následujícím obrázku 5-6.



Obr. 5-6 Původní návrh zapojení obvodu SIM1 v režimu s konstantním odběrem výkonu

Od záměru použít toto zapojení mě odradil vysoký odběr optočlenů U11, U15 a U17. Protože se mi nepovedlo získat optočleny HCPL-070A, bylo nutno místo nich použít HCPL-0700, které mají horší vlastnosti. Tím stoupla spotřeba a na základě již zmíněného problému s oddělovacím transformátorem bylo nezbytné rekonfigurovat zapojení obvodu SIM1 do režimu s odběrem konstantního proudu. Tím tato verze PDB přišla o možnost napájení přímo ze sběrnice. Řešením by bylo nastavit obvody SIM1 a DPC31 do režimů, které dovolují snížit energetickou náročnost jejich vzájemné komunikace přes optočleny. Tato úprava spočívá v použití pouze dvou optočlenů a zapnutí speciálního modulátoru. Na straně obvodu SIM1 signálem GIM do úrovně log. jedna a u DPC31 nastavením příslušného bitu v konfiguračním registru. Tato varianta navíc vyžaduje na straně obvodu DPC31 zapojení s externím komparátorem (viz obr. 5-7). To byl hlavní důvod, proč tato variantu nebyla použita již v prvním návrhu. Bližší informace je možné najít v dokumentaci k obvodům SIM1 [11] a DPC31 [10].



Obr.5-7 Externí komparátor na straně DPC31 určený pro spojení se SIM1 pouze dvěma optočleny

Korekce těchto chyb byla provedena přímo na desce plošného spoje. Funkce rozhraní Profibus PA byla odzkoušena nahráním zkušebního programu a připojením na Profibus (viz obr. 5-8). Při experimentech jsem zjistil, že signály TxD_TxS není zapotřebí a je možné je vypustit. Průběhy signálů RxD_RxS, XCTS_RxA, TxD_TxS a RTS_TxE změřené na osciloskopu jsou uvedeny v příloze VII.

V části Profibus DP nebylo možné sehnat původně navrhované optočleny HCPL-0721. Proto bylo potřeba provést modifikaci na zapojení s HCPL-0601. Ty jsou sice v provedení SMD ale jejich mezní přenosová rychlost jen 10Mbit/s. Před ověřováním bylo nutné provést rekonfiguraci PDB na zapojení pro Profibus DP (podle tabulky 5-1) a nahrát upravenou verzi zkušebního programu pro Profibus DP. Byly odzkoušeny všechny přenosové rychlostí 9,6; 93,75; 187,5 500; 1500; 3000; 6000 a 12000 kBit/s. Rozhraní pracovalo až do přenosové rychlosti 6000 kBit/s, což je dáno maximální přenosovou rychlostí použitých optočlenů.

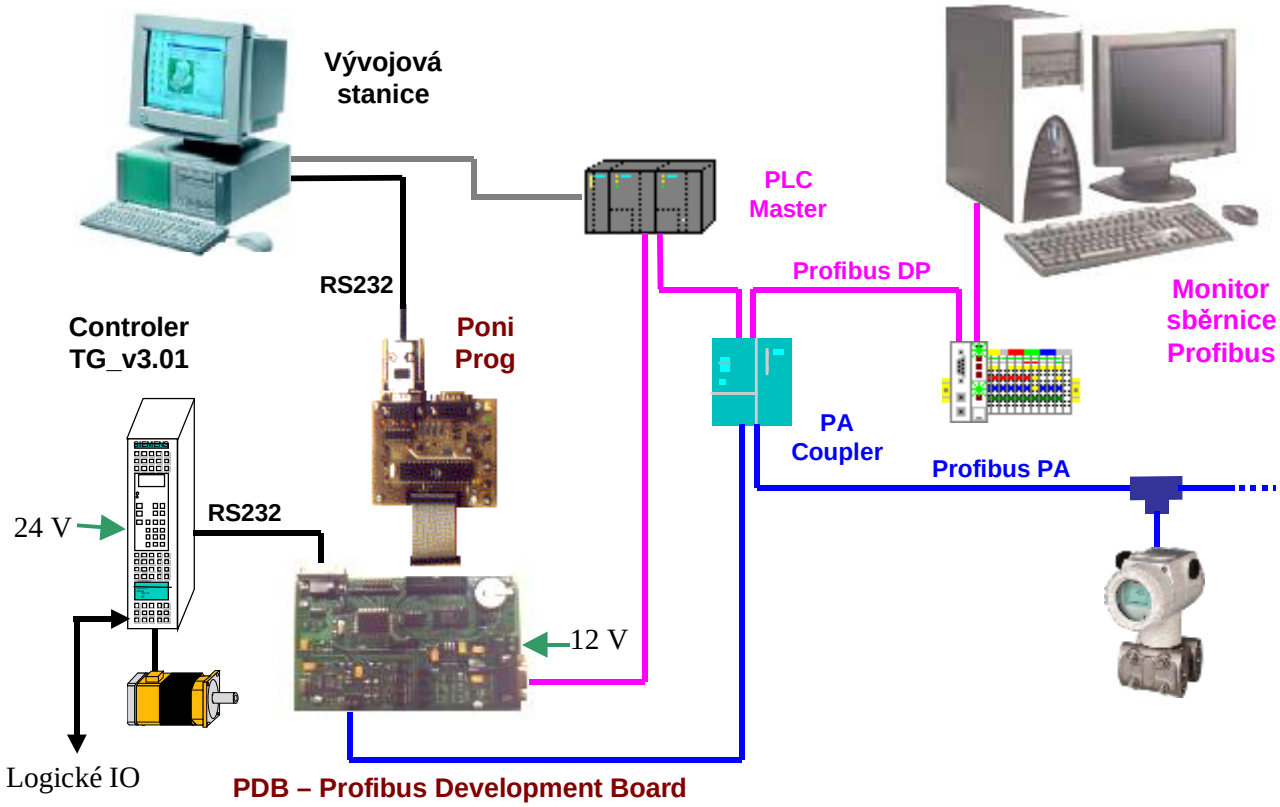
5.4 Závěr

Po hardwarové stránce zařízení plně funguje. Avšak je zapotřebí realizovat úpravy, které byly navrženy v předchozích odstavcích. U rozhraní PA je potřeba se vrátit k zapojení obvodu SIM1 v režimu s konstantním odběrem výkonu ze sběrnice. To obnáší použití komparátoru z obrázku 5-7. Pouze s tím rozdílem, že odpor R1 musí odpovídat napájecímu napětí 3,3V a použitým optočlenům. Dále je nutné provést návrh oddělovacího transformátoru DC/DC měniče a tím umožnit napájení procesorové části ze sběrnice.

Naproti tomu v části DP je potřeba použít rychlejší optočleny, které by umožnily komunikační rychlost až 12MBit/s. Jedna z možností je slevit z požadavku na SMD provedení optočlenů a použít například HCPL-7721 či jeho varianty, které jsou lépe dostupné.

U procesorové části by se měl upravit návrh logiky řídící sběrnici a odstranit tím chyby vzniklé neuvažováním signálu DBX. Ve stávajícím zapojení je při nahrávání programu stále zapisováno i do paměti v obvodu DPC31. Problém čtení paměti RAM při procesu programování (verifikace) je vyřešen tím, že jako strobovací signál je použit signál PSEN místo RD. Ale při zápisu je používán signál WR, který zároveň zapíše i do obvodu DPC31.

Dále je nezbytné připojit k nepoužitým portům PE až PH pull-up rezistory a tím zabránit jejich zničení. Bylo by také možno vyvést je na konektor, čímž by se zlepšila univerzálnost zařízení.



Obr. 5-8 Připojení kontroléru TG_v3.01 přes PDB ke sběrnici Profibus DP/PA

6. Vývoj softwaru

Nedílnou součástí této práce byl vývoj a testování softwaru pro zařízení Profibus Development Board (PDB). Vývoj softwarové i hardwarové části se neodmyslitelně prolínají, avšak pro větší přehlednost jsem rozdělit tyto části do samostatných oddílů.

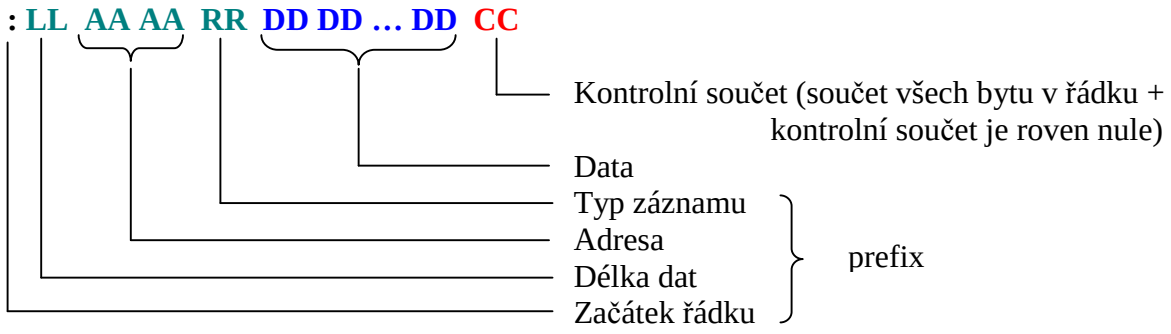
6.1 ProfiProg

6.1.1 ProfiProg – programátor

Práci na softwarové části bylo potřeba zahájit vývojem programu pro řídící procesor programátoru ProfiProg. K tomu byl použit vývojový nástroj firmy Atmel AVRStudio verze 4, který je volně dostupné na Internetu [20]. Výhodou tohoto prostředí je vysoká rychlost překladu a vestavěný simulátor.

Požadavky na programátor:
a) Spolehlivost nahrání programu
b) Vysoká rychlost programování
c) Jednoduchá komunikace
d) Bezpečnost připojení na sběrnici na PDB

Prvním úkolem byl vývoj programu pro řídící procesor. Pro sériový kanál jsem zvolil pevnou komunikační rychlost 115200 bit/s (1 start bit, 8 datových bitů a 1 stop bit). Požadavkem na programátor bylo, aby data vygenerovaná překladačem ve formátu Intel-HEX nahrál do paměti RAM na PDB. Aby programátor nutně nevyžadoval obslužný software na PC, je navržen tak, aby stačilo pouze poslat soubor ve formátu Intel-HEX na příslušný sériový port v počítači. To znamená komunikace ve formátu ASCII. Návrh vychází z formátu Intel-HEX, jehož řádek má strukturu:



Vždy dva znaky tvoří jeden byte jak hexadecimální číslo. Řídící procesor akceptuje dva typy záznamu: 00 – standardní datový záznam a 01 – konec souboru. Záznamy typu 02 až 05 nevyužívá. Vedle toho jsou použity některé speciální znaky pro řízení a diagnostiku programátoru. Přehled řídících znaků je uveden v následující tabulce 6-1.

Znak	Význam
‘!’	*Restart komunikace*. (diagnostika) Pokud programátor detekuje chybu (viz tabulka 6-2) zablokuje svou funkci, dokud neobdrží tento řídící znak.
‘#’	*Test spojení s ProfiProg*. (diagnostika) Tento příkaz slouží k ověření připojení programátoru ProfiProg. Za pozitivní odpověď je považován příchod znaku ‘*’
‘*’	*Test připojení PDB*. (diagnostika) Tímto příkazem je ověřeno propojení programátoru a PDB. Pozitivní odpovědí je příchod znaku ‘*’. ProfiProg testuje, zda došlo k uvolnění sběrnice. Konkrétně tak, že vyzkouší porty PA a PC. Každý bit je zkoušen zvlášť a to tak, že je nastaven do úrovně log 0. a ostatní jsou v log. 1. Následně je provedeno zpětné čtení skutečného stavu na daném portu. Pokud stav nesouhlasí, je uživateli oznámena chyba. Při chybě na portu PA je zaslán znak ‘A’ a v případě chyby na portu PC znak ‘C’.
‘:’	*Zápis do paměti programu v PDB*. (Počátek řádku souboru dat ve formátu Intel-HEX)
‘/’	*Verifikace paměti programu v PDB*. (Jako ‘:’ ale probíhá pouze verifikace)

Tab. 6-1 Příkazy pro komunikaci PC-ProfiProg

Zápis programu probíhá tak, že program vyhodnotí příchozí znak (zdrojový soubor *prog.asm*). Pokud se jedná pouze o příkaz k diagnostice, provede jej. Pokud se jedná o příkazy zápisu do paměti programu či jeho verifikaci, pokračuje vyhodnocením zbytku prefixu. Po jeho vyhodnocení zapisuje příchozí data do paměti řídícími signály přes konektor J1 (zdrojový soubor *ram.asm*) a provádí zpětné čtení. Při verifikaci je prováděno pouze čtení z požadované adresy. Na konci dojde k ověření kontrolního součtu a pokud je vše v pořádku, potvrdí provedení operace zasláním znaku ‘\0’. Pokud dojde k chybě, vrátí její číslo (viz tabulka 6-2) a indikuje příslušnou chybu diodou LED1 (zdrojový soubor *common.asm*). V tomto stavu programátor nepřijímá žádné příkazy kromě příkazu pro restart komunikace.

Návratová hodnota	Indikace	Význam
0	Zhasnuto	Operace úspěšně dokončena
1	Dvě bliknutí rychle po sobě a delší pauza	Chybná syntax
2	Pomalé blikání	Chyba kontrolního součtu
3	Velmi rychlé blikání	Chyba při verifikaci

Tab. 6-2 Návratové hodnoty programátoru ProfiProg při zápisu nebo verifikaci programu do PDB.

Pro ladící účely umožňuje programátor číst jak z programové tak datové paměti. To se realizuje tak, že je zaslán standardní prefix Intel-HEX. Avšak jako typ záznamu se uvede ‘10’ pro čtení paměti programu a ‘20’ paměti dat. Následně programátor vrátí požadovaná data z požadované adresy ale pouze v binární podobě. Díky problémům s přístupem na sběrnici, není čtení datové paměti plně funkční (viz kapitola 5.3.1.2).

Pro úplnost uvádím, že obslužné funkce pro práci se sériovým kanálem se nalézají ve zdrojovém souboru *rs232.asm* a definice konstant a maker v *def.asm*. Tyto zdrojové soubory se nalézají na přiloženém CD v adresáři `Software\ProfiProg`.

6.1.1 ProfiProg – Rozhraní na PC (Comm)

Jak je uvedeno ve formulaci požadavků v úvodu tohoto oddílu, od programátoru byla vyžadována vysoká spolehlivost zápisu programu do PDB. Proto jsem napsal jednoduchý obslužný program na PC, usnadňující uživateli nahrání a verifikaci programu. Pracovně byl nazván Comm. Avšak tento program není k naprogramování PDB nezbytně nutný. Protože komunikační protokol je postaven na struktuře souborů ve formátu Intel-HEX, není potřeba tyto data jakkoliv při přenosu upravovat. Programování lze provést tak, že přeložený soubor v tomto formátu odešleme (např. obyčejným příkazem *copy*) na příslušný sériový port v PC.

Komunikační program (Comm) jsem řešil v prostředí Microsoft Visual C++ jako konzolovou aplikaci. Informace o používaném komunikačním portu (číslo COM portu) a jméno souboru se vstupními daty (přeložený program ve formátu Intel-HEX) je programu předáván jako parametry z příkazového řádku.

Na začátku program vyhodnotí vstupní parametry, otevře vstupní soubor a komunikační kanál. Pokud dojde k chybě, dá tuto skutečnost uživateli na vědomí společně s nápovědou. Po úspěšné inicializaci se provede test vstupního souboru. To znamená ověření správného formátu a kontrolních součtů. V případě úspěšnosti této operace program přejde do fáze ověřování komunikace. Nejprve zašle příkaz pro otestování spojení s programátorem ProfiProg.. Pokud odpověď není pozitivní je zaslán příkaz pro restart komunikace (znak ‘!’) a test spojení je opakován. Po úspěšném ověření spojení je programátor požádán o ověření spojení s deskou PDB. Pokud dojde k chybě je tato skutečnost dána uživateli na vědomí.

Nyní je možno zaslat požadovaná data a provést vlastní naprogramování. Data jsou zasílána po řádcích a na konci každého je očekáváno potvrzení o souhlasícím CRC (příchod znaku ‘\0’). Pokud dojde k chybě, je pokus třikrát opakován. Po dokončení programování je provedena verifikace. Ta je prováděna stejným způsobem jako samotné programování. Pouze znaky indikující začátek řádku ‘:’ jsou nahrazeny znakem ‘/’. Pokud vše proběhne úspěšně, je proces programování ukončen.

Program pro komunikaci používá služeb Windows API a jeho funkčnost byla odzkoušena pod Windows 98, NT a 2000. Zdrojové soubory se nacházejí v adresáři `Software\Comm` na přiloženém CD. Hlavní program se nachází ve stejnojmenném souboru *Comm.cpp*. Při tvorbě knihovny (*comport.cpp* a *comport.h*) pro sériovou komunikaci mi byla inspirací již existující knihovna získaná na Internetu [28]. Dále mi významným pomocníkem byly internetové stránky společnosti Microsoft [27].

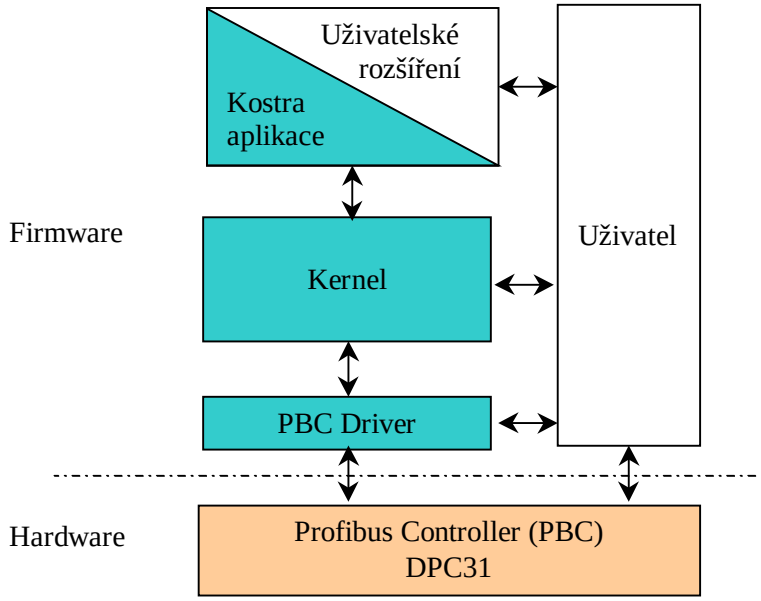
6.2 PDB

6.2.1 V1SL

K obvodu DPC31 vyvinula firma Siemens uživatelské rozhraní (firmware) označené jako V1SL [6]. Ten je určen pro snadnější implementaci zařízení typu Profibus slave DP/PA. Tento programový balík vyhovuje jak základní normě Profibus DP [1], tak i jejímu rozšíření Profibus DPV1 [2].

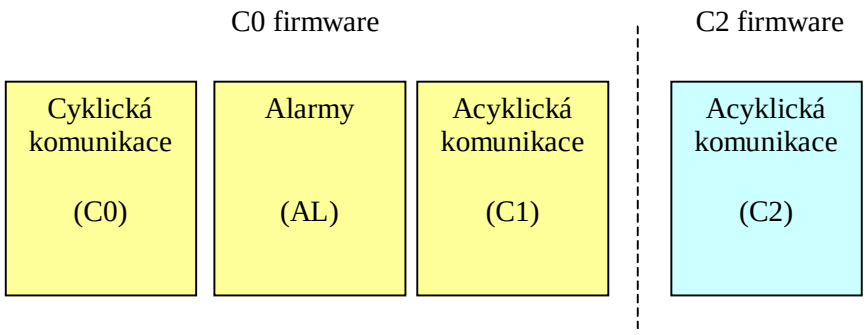
Základní rysy V1SL jsou:

- Implementace zařízení slave podle normy Profibus DP a Profibus DPV1
- Možnost využít pouze potřebnou část firmwaru.
- Přenosová rychlost 9,6kbit/s až 12Mbit/s.
- Multi-instanční operace (více komunikačních kanálů C0 a (nebo) C2)
- Multi-device operace (spolupráce s více zařízeními)
- Maximální délka vstupních dat 244 bytů a výstupních také 244 bytů
- Konzistence dat přes maximální uživatelskou délku dat (tzn. 244 bytů)
- Nezávislost na operačním systému
- Použitelnost na procesorech 8031, 80C16x, i80x86, Pentium a kompatibilní.



Obr. 6-1 Struktura V1SL

Na obrázku 6-1 je znázorněna struktura V1SL (zvýrazněna modrou barvou). Nejblíže komunikačnímu rozhraní PBC je PBC Driver. Tato komponenta se nazývá PBC. Její výměnou nebo modifikací je možné použít jiné rozhraní než je DPC31. Kernel je část balíku V1SL, který implementuje normu Profibus DP a DPV1. Obsahuje komponenty, které je možno selektivně při překladu přidat či odebrat. Jedná se o komunikaci se zařízením typu Master Class 1 (Cyklická výměna dat, alarmy a acyklická výměna dat) označené jako C0 firmware a komunikaci se zařízením typu Master Class 2 (acyklická výměna dat). Tato komponenta se nazývá C2 firmware.



Obr.6-2 Komponenty jádra (Kernelu)

Pro větší přehlednost součásti jednotlivých komponent jako jsou proměnné, makra a funkce jsou označovány prefixy. Ty jsou uvedeny v tabulce 6-3.

Prefix	**Komponenta**
v1sl_…/ V1SL_…	Všechny identifikátory firmwaru, které mají obecný význam
v1sl_c0…/ V1SL_C0…	Vstupní funkce/výstupní makra které se vztahují k cyklické výměně se zařízením Profibus Master Class 1.
v1sl_al…/ V1SL_AL…	Vstupní funkce/výstupní makra které se vztahují k alarmům.
v1sl_c1…/ V1SL_C1…	Vstupní funkce/výstupní makra které se vztahují k acyklické výměně se zařízením Profibus Master Class 1.
v1sl_c2…/ V1SL_C2…	Vstupní funkce/výstupní makra které se vztahují k acyklické výměně se zařízením Profibus Master Class 2.
pbc_…/ PBC_…	Všechny identifikátory komponenty PBC, které se nevztahují ke konkrétnímu PBC.
pbc_dpc31_…	Identifikátor pro vstupní funkce, které se vztahují speciálně k obvodu DPC31.

Tab. 6-3 Značení proměnných, funkcí a maker v knihovně V1SL

Nyní bych se rád vrátil k obrázku 6-1, kde je znázorněna struktura V1SL. Uživatel jí předává potřebné informace voláním tzv. vstupních funkcí. Opačně pokud potřebuje DPV1 slave (potažmo V1SL) volat uživatele, děje se to pomocí tzv. výstupních funkcí. Ty jsou realizovány buď jako makra, která jsou nahrazena při překladu odpovídajícím kódem daným konfigurací. Druhý způsob je pomocí call-back funkcí, které uživatel zpřístupní V1SL.

Protože softwarový balík V1SL poměrně rozsáhlý, je organizován v adresářové struktuře.

Adresář	**Obsah**
. \BAT_DIR	Dávkové programy pro kompilaci + make-file
. \SRC_DIR	Zdrojové kódy
. \LST_DIR	Listing
. \OBJ_DIR	Přeložené soubory
. \TOOL_DIR	Překladač, linker, standardní knihovny …

Tab. 6-4 Adresářová struktura softwarového balíku V1SL

6.2.2 Ověření funkce V1SL a PDB

Součástí balíku V1SL je kostra aplikace [7] (viz obr. 6-1). Konkrétně se jedná o teploměr na Profibus DP, kdy je použit obvod DPC31 s externím procesorem řady 8051. Identifikační číslo zařízení PNO je 0x00, 0x00 a adresa na sběrnici je 3. Délka uživatelských parametrizačních dat (DPV1 parametrizace) jsou 3 byty (0x00, 0x00, 0x00). Jde o kompaktní zařízení, jehož konfigurační data jsou 0x13, 0x23. To znamená že při cyklické komunikaci se zařízením Master Class 1 přijímá 4 byty (0x13) o vysílá také 4 byty (0x23) v konzistenci pouze přes jednotlivé byty (viz kapitola 3.3.3.4). Přijatá data jsou vysílaná zpět zařízení master. Dále je obsažen příklad použití acyklické komunikace se zařízeními Master Class 1 a 2. Zde je přístupný pouze index 2 a 3 v slotu 0 (u komunikace s stanicí Master Class 1 je číslo slotu ignorováno). Přes index 2 je možno číst data vyměňovaná cyklickou komunikací. Zápis není povolen a vrací chybu „přístup odmítnut“ (V1SL_EC1_ACC_CODE_ACCESS_DENIED). Přes index 3 je přístupná proměnná typu pole (*user_data*), ze které lze číst i do ní zapisovat.

Pro ověření PDB bylo nutné provést překlad tohoto příkladu a provést ověření funkčnosti zařízení jako celku. Nakonfiguroval jsem PDB k připojení na Profibus PA s externím napájením a provedl modifikaci V1SL. Informace o nastavení firmwaru jsem získal převážně z práce Petra Smolíka [13]. Nastavení je potřeba provést v souboru *sys_main.c* (`.\SRC_DIR\SYSTEM\`). Protože na PDB je použit interní procesor je nutné vyřadit reset PBC (DPC31). To se provede zakomentováním těla funkce *sys_init_hardware(void);*. Dále je potřeba ve funkci *sys_init_path_info(void);* nastavit:

```
sys_v1sl_pbc_ldb.baud_control                        = PBC_BAUDRATE_SYN_31_25;
sys_v1sl_pbc_ldb.asic.dpc31.com_syn_physic           = TRUE;
sys_v1sl_pbc_ldb.asic.dpc31.com_user_ram_segments    = DPC31_USER_RAM_SEGMENTS;
```

a přidat řádky:

```
sys_v1sl_pbc_ldb.asic.dpc31.com_mode_syn_low  = PBC_DPC31_MODE_SYN_L_ADD_SIGNAL |
                                                PBC_DPC31_MODE_SYN_L_QUICK_SYNC;
sys_v1sl_pbc_ldb.asic.dpc31.com_mode_syn_high = PBC_DPC31_MODE_SYN_H_CLOCK_IN_4_MHZ |
                                                PBC_DPC31_MODE_SYN_H_PREAMBLE_8_BYTE;
sys_v1sl_pbc_ldb.asic.dpc31.c0_dx_tact_beat_out  = 0;
```

Význam tohoto nastavení bude vysvětlen dále. Pro překlad byl použit překladač Keil 6.10 [26] a přeložený program (*v1slsamp.hex*) nahrán do PDB. Následně jsem zařízení připojil na sběrnici a programem SCOPE PROFIBUS ověřil, že se zařízení nachází v tzv. live-listu (seznam zařízení připojených na sběrnici). Na základě parametrů zkušební aplikace jsem vytvořil testovací GSD soubor a provedl konfiguraci sběrnice v prostředí Simatic Manager a odzkoušel cyklickou komunikaci. Acyklická komunikace byla ověřena v prostředí PROTest.

6.2.2 Implementace profilu kontroléru

6.2.2.1 Display a sériový kanál

Před implementací konkrétního zařízení, jsem připravil obsluhu periferií jako je display a sériový kanál. V tabulce 6-5 jsou uvedeny příslušné metody. Řešení obsluhy sériového kanálu bude vysvětleno níže. Displej je inicializován funkcí *void init_display(void)* z funkce main v souboru *sys_main.c* (`.\SRC_DIR\SYSTEM\`). Jeho přítomnost není nezbytná pro funkci PDB ale velice mi usnadnila odstraňování chyb během vývoje. Implementací funkce *char putchar(char c)* je předefinován standardní výstup knihovny *stdio.h*. Tím je možné používat funkce jako je například *printf* pro přístup na display.

Metody	**Význam**
Sériový kanál (.\SRC_DIR\USR\.) usr.c	
void ini_serial (void)	Inicializace.
void dev_send_command(char dev_data)	Čeká na odvysílání předchozího znaku a pošle znak. Tato funkce může být použita pouze tehdy, když je zakázaná obsluha přerušení od sériového kanálu.
void serial (void) interrupt 4 using 2	Obsluha příjmu i vysílání sériového kanálu.
Display (.\SRC_DIR\SYSTEM\) lcd_drv.c a lcd_drv.h	
void init_display(void)	Inicializace.
char putchar(char c)	Zápis znaku na aktuální pozici kurzoru.
void gotoxy(unsigned char x, unsigned char y)	Změna aktuální pozice kurzoru.

Tab. 6-5 Funkce pro obsluhu sériového kanálu a displeje

6.2.2.2 Implementace profilu do vzorové aplikace

Hlavním úkolem mé práce bylo odzkoušet funkčnost PDB jako rozhraní mezi Profibus a kontrolérem bezkartáčových motorů TG_v3.01. Vývoj bylo potřeba zahájit sestavením profilu dle normy [3]. Výsledkem jsou tabulky (viz příloha III) popisující strukturu implementovaného profilu. Byl použit model jednoduchého zařízení s jedním fyzickým blokem (PB), jedním funkčním blokem (FB) a jedním převodním (TB). Fyzický blok je typu pohon (actuator). Funkční typu PID regulátor (PID contoroller) a převodní elektrický pohon (electic actuator). Profil je implementován ve třídě B a vedle povinných a některých nepovinných parametrů je implementováno také několik uživatelských. Fyzický blok obsahuje navíc tři parametry, pomocí kterých je možné nastavit sériový kanál. Převodní blok je rozšířen o čtyři parametry. První dva určující požadovanou rychlost pojezdu a zrychlení při rozjezdu. Další dva jsou pak vztaženy k logickým vstupům a výstupům.

Vlastní implementace profilu spočívala v úpravě vzorové aplikace. Po resetu procesoru je vykonána inicializace na úrovni asembleru v souboru *sys_51.asm* (`.\SRC_DIR\SYSTEM\`). Tento soubor byl ponechán beze změny. Odtud je volána hlavní funkce *void main(void).* Zde

jsem odstranil reset DPC31 a přidal inicializace displeje. Potom je provedena inicializace obvodu DPC31 voláním funkce *void sys_init_path_info(void)* a V1SL funkcí *void sys_init_slave(void)*. Na konci funkce main je zavolána uživatelská aplikace – funkce *void usr_main(void)*.

Výkonné jádro uživatelské aplikace se nachází v souboru *usr.c* (`.\SRC_DIR\USR\`). Jako první je provedena inicializace uživatelských proměnných. Zde bylo potřeba doplnil inicializaci datové struktur profilu voláním funkcí *init_data_static()* a *init_data()*. První inicializuje celou strukturu profilu na implicitní hodnoty. Druhá nastaví profil do počátečního stavu. Tzn. inicializace datových struktur jako jsou ST_REV, MODE_BLK, ALARM_SUM … . Volání funkce *init_data_static()* je zde pouze z vývojových důvodů. Normálně je tato funkce volána při pokynu zařízení master k restartu.

Následně jsou otevřeny komunikační kanály C0 a C2 firmwaru funkcemi *v1sl_c0_open_channel(0)* a *v1sl_c2_open_channel(0)*. Na závěr inicializace jsou povolena přerušení INT0 od DPC31 a přerušení od sériového kanálu. Obsluhu událostí od Profibusu (přerušení od PBC) není nutné odsluhovat pod přerušením. Obslužnou funkci *pbc_dpc31_int_handler()* lze v běžných aplikacích volat cyklicky v jistých minimálních časových intervalech. Ale obecně je lepší umístit tuto obsluhu pod přerušení. Obě přerušení jsou nechána na stejné (nižší) prioritě dané inicializací procesoru.

Celý program je vykonáván v nekonečné smyčce *while (1)*. Nejprve jsou obslouženy buffery pro příjem a vysílání dat při cyklické komunikaci. Jako pomocný buffer zde slouží proměnná *usr_data_buffer*.

Adresa bufferu pro vysílaná data se získá voláním funkce *v1sl_c0_get_input_ptr()*, která vrátí jeho adresu. Po jeho naplnění je volána funkce *v1sl_c0_input_update()*, která potvrdí aktualizaci dat.

Obdobně pro příjem je adresa odpovídajícího bufferu získána voláním funkce *v1sl_c0_get_output_info(&usr_data.c0.output_info)*. Pokud proběhne úspěšně vrátí *V1SL_OK* a naplní strukturu *usr_data.c0.output_info*. Ta obsahuje pointer na příchozí data *usr_data.c0.output_info.ptr* a jejich status *usr_data.c0.output_info.state*. Na základě statusu jsou pak vyhodnocena. Nová data indikuji v programu nastavením proměnné *profibus_update=1*.

Při příchodu nových dat je volána funkce *block_comp_in()*, která provede aktualizaci funkčního a následně převodního bloku, jak bylo vysvětleno v kapitole 4.3. Při každém průchodu tělem smyčky *while (1)* je volána funkce *block_comp_out()*, který provede také aktualizaci datových profilů ale v opačném směru než funkce *block_comp_in()*. Ta přečte z převodního bloku výstupní údaje a provede příslušné aktualizace obou bloků. Následně je provedena obsluha displeje a ošetření událostí ve fyzickém bloku a celí děj se opakuje.

Acyklická komunikace se zařízeními Master Class 1 a 2 je obsloužena společně metodami *void usr_read_ds(V1SL_LL_DS_READ_PTR ds_read_ptr)* pro požadavek na čtení a *void usr_write_ds(V1SL_LL_DS_WRITE_PTR ds_write_ptr)* při požadavku na zápis do profilu. Chybové události (viz tabulka 6-6) jsou oznámeny nadřazenému zařízení.

Chyba	**Význam**
V1SL_EC1_ACC_CODE_SLOT_INVALID	Chyba adresace (chybný slot)
V1SL_EC1_ACC_CODE_INDEX_INVALID	Chyba adresace (chybný index)
V1SL_EC1_ACC_CODE_ACCESS_DENIED	Záznam je nepřístupný pro zápis nebo profil je zamknut funkcí WRITE_LOCKING (viz III),
V1SL_EC1_ACC_CODE_WRITE_LEN	Pokus o zápis špatné délky dat

Tab. 6-6 Chybové kódy podporované aktuální verzí převodníku PDB – TG_v3.01

6.2.2.3 Změna konfigurace vzorové aplikace

Vedle implementace profilu bylo potřeba upravit konfiguraci PBC a V1SL. Nastavení jsou soustředěna v souboru *sys_cfg.h* v adresáři `.\SRC_DIR\COMM_DEV\` . Podrobnosti lze najít v [6 str. 148].

Symbol	**Hodn.**	**Význam**
Povolení překladu modulů		
SYS_C0_SUPPORT	1	Povolení (1) / zákaz (0) modulu C0
SYS_C1_SUPPORT	1	Povolení (1) / zákaz (0) modulu C1
SYS_C2_SUPPORT	1	Povolení (1) / zákaz (0) modulu C2
SYS_AL_SUPPORT	1	Povolení (1) / zákaz (0) modulu AL
Hardwarové nastavení		
SYS_PBC_HARDWARE_ADDRESS	0x0000	Adresa DPC31 (syst. sběrnice)
SYS_PBC_COM_MODE		Hardwarové nastavení DPC31
SYS_PBC_RESET_PIN		Definice signálu, kterým resetuje DPC31
Nastavení zařízení slave		
SYS_PBC_MIN_TSDR	11	Min T_{SDR} viz 3.3.3.3 (11÷255)
SYS_PBC_USER_WD_VALUE	500	Uživatelský watchdog (0-vypnut, max. 65535)
SYS_PBC_STATION_ADDRESS	30	Adresa Profibus slave
SYS_PNO_IDENT_NUM_HIGH	0x98	PNO identifikační číslo
SYS_PNO_IDENT_NUM_LOW	0x08	
SYS_C0_LEN_MAX_SSA	0	Velikosti uživatelských bufferů pro komunikaci
SYS_C0_LEN_MAX_PRM	16	
SYS_C0_LEN_MAX_CFG	5	
SYS_C0_LEN_MAX_USER_DIAG	32	
SYS_C0_LEN_MAX_INPUT	128	
SYS_C0_LEN_MAX_OUTPUT	128	
SYS_C0_NO_ADDRESS_CHANGE	TRUE	TRUE – zakáže změnu Profibus slave adresy
SYS_C0_SYNC_SUPPORTED	TRUE	TRUE – povoluje podporu sync mód
SYS_C0_FREEZE_SUPPORTED	TRUE	TRUE – povoluje podru freeze mód
SYS_C1_USER_DATA_LEN	240	Velikost bufferu (-4) pro C1 a AL
SYS_C2_USER_DATA_LEN	240	Velikost bufferu (-4) pro C2
SYS_C2_CONNECT_COUNT	2	Maximální počet spojení C2 (1÷7)
SYS_C2_INITIATE_PDU_SIZE	64	Max délka dat při operaci Data Set.
SYS_C2_POLL_TIMEOUT	100	Čas, během kterého je slave schopen odpovědět.
SYS_C2_FINAL_COMM_POINT	TRUE	TRUE – konečné zařízení, FALSE – link
SYS_C2_AUTO_INITIATE_RESP	TRUE	TRUE–C2 generuje odpověď na požadavek na inicializaci.
SYS_C2_PROFILE_FEATURES_SUPPORTED_1	0	Pokud SYS_C2_AUTO_INITIATE_RESP je TRUE, používají se tyto údaje pro odpověď na požadavek na inicializaci.
SYS_C2_PROFILE_FEATURES_SUPPORTED_2	0	
SYS_C2_PROFILE_IDENT_NUM_LOW	0x02	
SYS_C2_PROFILE_IDENT_NUM_HIGH	0x64	
SYS_C2_SUBNET	1	Aktivace některých rozšíření daných DPV1

Tab. 6-7 Nastavení V1SL v souboru sys_cfg.h

Nastavení těchto konstant je pak využíváno k inicializaci ve funkcích *void sys_init_path_info(void)* v souboru *sys_main.c* a *void usr_main(void)* ze souboru *usr.c*. Protože vzorový příklad k V1SL je koncipován pro zapojení s externím procesorem (pro Profibus DP), je potřeba ještě provést některé úpravy právě ve funkci *void sys_init_path_info(void)*. A právě tímto se liší nastavení pro zařízení Profibus slave DP a PA. Úprava na Profibus DP je jednodušší a proto začneme s ní.

Pro zařízení Profibus DP je potřeba přidat následující řádky:

```
sys_v1sl_pbc_ldb.asic.dpc31.com_user_ram_segments          = DPC31_USER_RAM_SEGMENTS;
sys_v1sl_pbc_ldb.asic.dpc31.c0_dx_tact_beat_out            = 0;
```

První řádek říká, kolik externí datové pamětí v DPC31 může ovladač PBC celkově použít. Hodnota je zadávána v počtu 32 bytových segmentů. V případě, že je přiřazena hodnota *DPC31_USER_RAM_SEGMENTS*, ovladač PBC si alokuje veškerou dostupnou paměť DPC31 (do 0x2000). K alokaci paměti dochází až za běhu programu a pokud je paměť využívána uživatelskou aplikací skončí program běhovou chybou „Fatal error“. Je však možné přiřadit nižší hodnotu. Abych se přesvědčil, že je možné pro jednoduchou aplikaci jako je tato vystačit pouze s pamětí, kterou obsahuje DPC31, zkusil jsem omezit celkovou paměť dat na velikost 8kB (0x2000). Využití této paměti je možné vyčíst ze struktury *V1SL_STRUC_PBC_DPC31* definované v souboru *v1sl_com.h* v `.\SRC_DIR\COMM_DEV\`. V1SL využívá staticky adresový prostor až do 0x0980. Má aplikace vyžaduje datovou oblast (xdata) velkou 0x0A02 bytů. Omezením počtu segmentů pro ovladač PBC na 45 se povedlo umístit uživatelská data nad tuto oblast (ve *v1sl_51.mak*).

Proměnná *sys_v1sl_pbc_ldb.asic.dpc31.c0_dx_tact_beat_out* určuje, zda signály PB_3 a PB_2 obvodu DPC31 budou ovládány jako IO signály vnitřního procesoru nebo budou řízeny událostmi na Profibusu (příchod nových dat a příjem telegramu globálního řízení). Přiřazením hodnoty 0, jsou tyto signály využívány jako IO vnitřního procesoru.

Pro zařízení Profibus PA je navíc potřeba modifikovat řádky:

```
sys_v1sl_pbc_ldb.baud_control                              = PBC_BAUDRATE_SYN_31_25k;
sys_v1sl_pbc_ldb.asic.dpc31.com_syn_physic                 = TRUE;
```

Tím se umožní přepnutí procesoru do synchronního režimu s přenosovou rychlostí 31,25kbit/s. Dále je potřeba provést inicializaci synchronního režimu přidáním řádků:

```
sys_v1sl_pbc_ldb.asic.dpc31.com_mode_syn_low    = PBC_DPC31_MODE_SYN_L_ADD_SIGNAL |
                                                  PBC_DPC31_MODE_SYN_L_QUICK_SYNC;
sys_v1sl_pbc_ldb.asic.dpc31.com_mode_syn_high   = PBC_DPC31_MODE_SYN_H_CLOCK_IN_4_MHZ |
                                                  PBC_DPC31_MODE_SYN_H_PREAMBLE_8_BYTE;
```

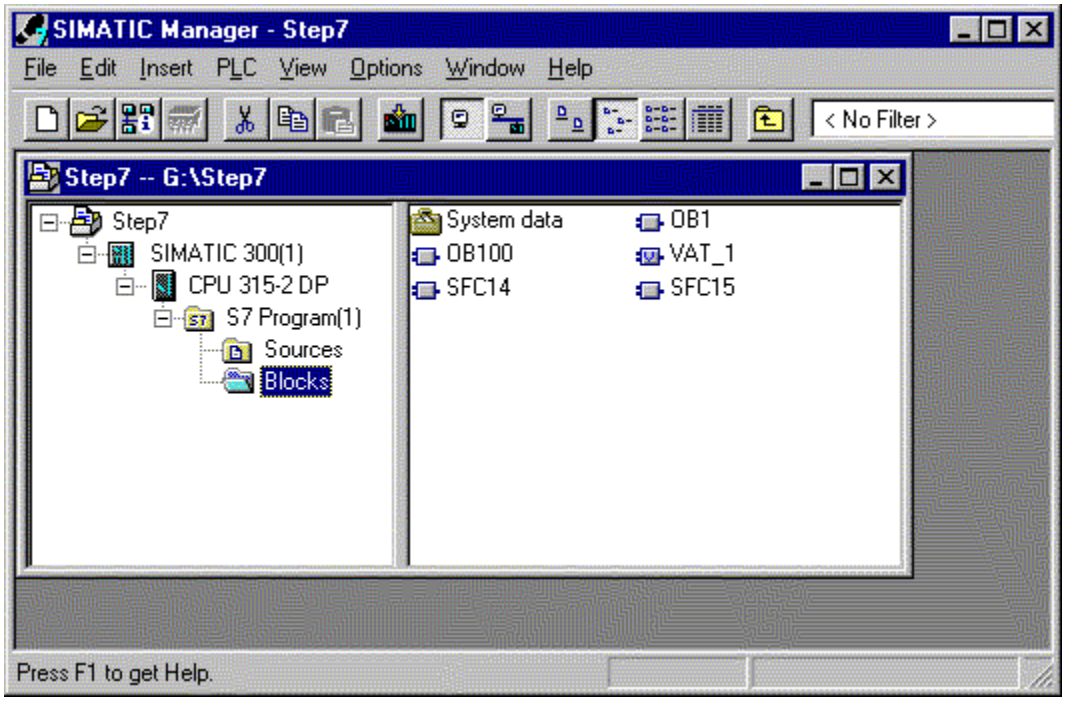
Význam těchto nastavení je vysvětlen v [6].

Na závěr bylo potřeba změnit konfigurační data a jejich kontrolu při konfiguraci zařízením master před zahájením cyklické výměny dat. Zařízení Profibus musí umožňovat cyklickou výměnu všech povinných parametrů profilu, které jsou určeny pro cyklickou komunikaci. V příloze III jsou tyto položky označeny cyc ve sloupečku „Trans.“. Na strance www.profibus.com jsem si opatřil GSD soubory podobných zařízení. Po jejich prostudování jsem se rozhodl implementovat rozhraní pro kontrolér jako modulární slave. Konkrétně byly vytvořeny tři moduly s konzistentními daty. Díky modulárnosti, jsou přístupné všechny parametry dané normou ale zároveň je na uživateli, která data skutečně potřebuje.

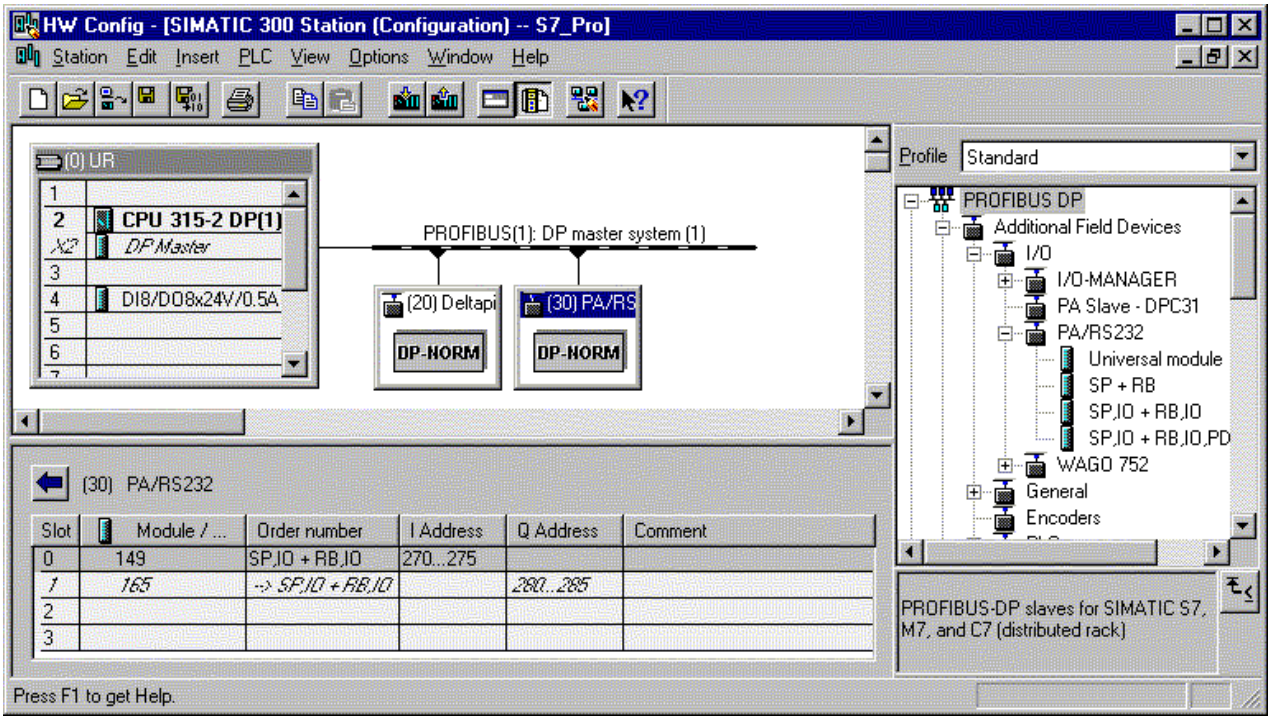
číslo modulu	Konfigurační data	Přijímané proměnné	Odesílané proměnné
MOD_1	0x94,0xA4	SP	READ_BACK
MOD_2	0x95,0xA5	SP, IO_OUTPUT	READ_BACK, IO_INPUT
MOD_3	0x95,0xA5,0xA1,0xA2	SP, IO_OUTPUT	READ_BACK, IO_INPUT, POS_D, CHECK_BACK

Tab. 6-8 Moduly podporované rozhraním PDB – TG_v3.01

Tyto úpravy je nutné provést v souboru *usr.c*. Zaprvé je potřeba změnit implicitní konfigurační data v proměnné *usr_data.c0.real_cfg*_ve funkci *void usr_main(void)* a v závislosti na délce těchto dat upravit jejich přepis do komunikačního bufferu ve funkci *void usr_c0_real_cfg_buffer_changed(void).* Volba modulu se provádí ve funkci *void usr_c0_new_cfg(V1SL_…).* Přijatá konfigurační data jsou porovnána se třemi definovanými moduly. Pokud tyto údaje souhlasí do proměnné *actual_module* je přiřazeno číslo aktuálního modulu. Na základě této proměnné pak hlavní funkce *void usr_main(void)* vyplňuje komunikační buffery cyklické komunikace. Délku dat, která je odesílána z těchto bufferů, nastavuje V1SL automaticky.



Obr. 6-3 Projekt v prostředí Simatic Manager



Obr. 6-4 Hardwarová konfigurace v prostředí Simatic Manager

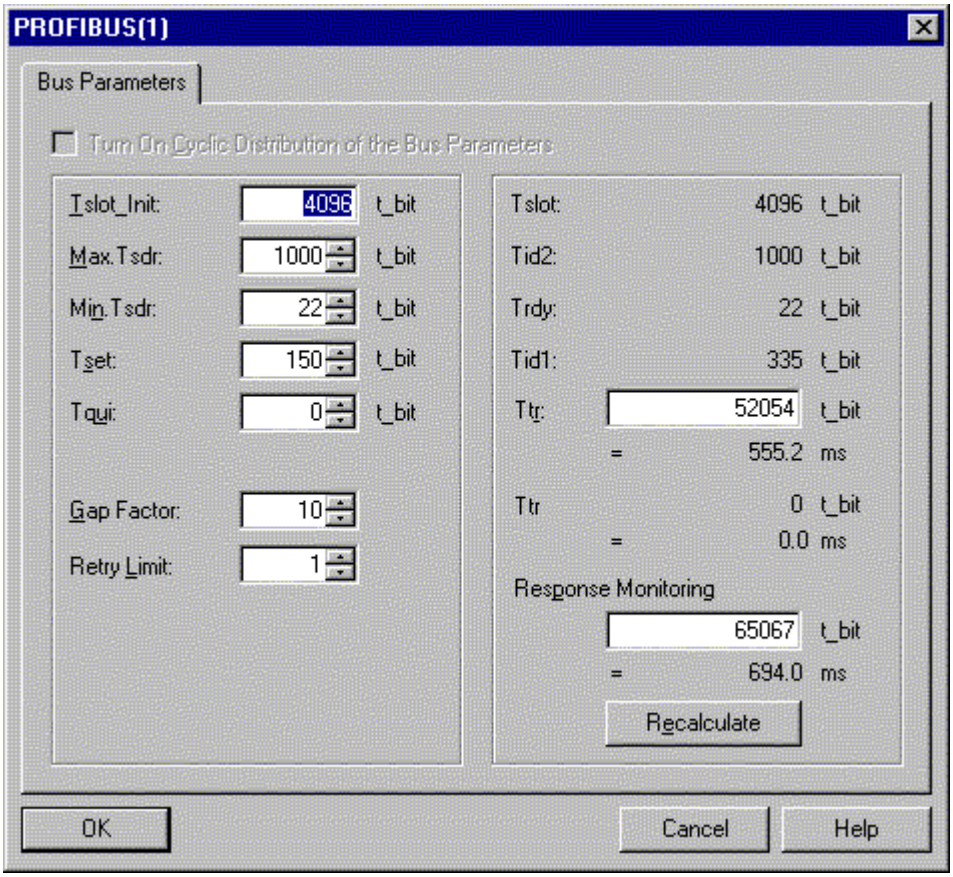
6.2.2.2 Konfigurace na sběrnici

Na základě konfigurace V1SL jsem vytvořil GSD soubor pro popis implementovaného zařízení. Rozhraní jsem připojil ke sběrnici, konkrétně na demonstrátor Profibus DP/PA [24].

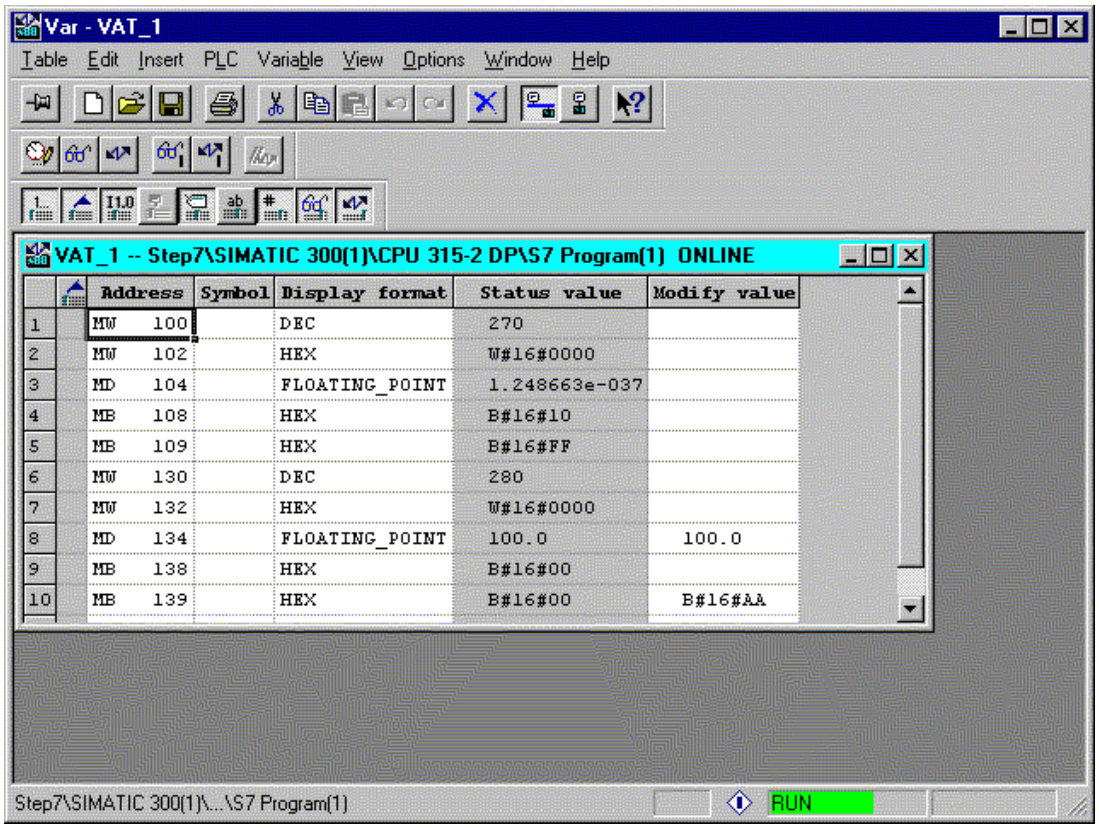
Zařízením master bylo PLC SIEMENS řady S7-300 CPU 315 2 DP (adr. 2). Na sběrnici Profibus DP byly vedle PLC jako zařízení master Class1 připojeny ještě zařízení slave (WAGO 752-323 – logické IO (adr. 10)) a převodník Profibus DP/PA od firmy PEPPERL+FUCHS. Na tento převodník byly připojeny další zařízení Profibus PA (měřič tlaku Deltapilot S (adr. 20) a CERABAR (adr. 22), dále pak snímač výšky hladiny Prosonic T (adr. 21) a teploměr TF 211 (adr. 23)).

V prostředí Simatic Manager jsem vytvořil projekt, který byl pracovně nazván Step7. V hardwarové konfiguraci bylo nejprve potřeba importoval vytvořený GSD soubor a následně provést konfigurací sběrnice. Z katalogu jsem vložil lištu Simatic 300, Rack, Rail a do ní na pozici 2 umístil PLC (CPU 315-2 DP). Adresu stanice master jsem nastavil 2 a přenosovou rychlost 93,75 kbit/s. Parametry sběrnice byly nastaveny, jak je znázorněno na obrázku 6-5. Na pozici 4 v liště jsem vložil příslušnou jednotku logických vstupů a výstupů DI8/DO8x24V/0,5A. Pak bylo možné přistoupit ke konfiguraci zařízeni slave. Pro kontrolu jsem do konfigurace přidal zařízení Deltalilot na adrese 20 a jeho vstupní adresu nastavil 260 (směr do PLC). Pak už jsem z katalogu přidal své zařízení a přiřadil mu adresu 30. Zvolil jsem modul č.2 (SP,IO+RB,IO) a přiřadil mu vstupní adresu 270 a výstupní 280. Konfigurace byla ověřena jejím nahráním do PLC.

Pak jsem vytvořil v prostředí Simatic Step7 jednoduchý program určený k testování. Protože posílaná i přijímaná data jsou konzistentní jako celek, je potřeba k nim přistupovat pomocí funkcí SFC14 a SFC15. Program obsahuje dva časovače, které periodicky měnily výstup



Obr. 6-5 Parametry sběrnice Profibus nastavené v hardwarové konfiguraci



	Address	Symbol	Display format	Status value	Modify value
1	MW 100		DEC	270	
2	MW 102		HEX	W#16#0000	
3	MD 104		FLOATING_POINT	1.248663e-037	
4	MB 108		HEX	B#16#10	
5	MB 109		HEX	B#16#FF	
6	MW 130		DEC	280	
7	MW 132		HEX	W#16#0000	
8	MD 134		FLOATING_POINT	100.0	100.0
9	MB 138		HEX	B#16#00	
10	MB 139		HEX	B#16#00	B#16#AA

Obr. 6-6 VAT tabulka v Simatic Manageru

na kontroléru. Výpis programu je v příloze VIII. K tomuto programu je v projektu Simatic Manageru vytvořena tzv. VAT tabulka, ve které jsou vytvořeny všechny důležité proměnné. Pomocí nich je pak možné sledovat datovou výměnu s kontrolérem a ovládat jej. Vytvořený projekt se nachází na přiloženém CD v adresáři `Software\Step7`.

6.2.3 Rozhraní PDB a TG_v3.01

V tomto okamžiku byla oživena a plně funkční část rozhraní mezi Profibus a PDB. Teď už zbývala pouze práce na rozhraní PDB a TG_v3.01 přes sériový kanál. Rozhraní RS232 není samo osobě příliš rychlé a navíc v Režimu PA je díky nízkému kmitočtu procesoru problém s generování přenosové rychlosti. Stávající protokol implementovaný v TG_v3.01 byl nevyhovující a proto bylo nezbytné navrhnout nový. Požadavkem na něj byla co největší efektivita s spolehlivost. Navržený protokol je založen na 9-bitové komunikaci. Sériový kanál procesorů řady 8051 umožňuje v módech 2 a 3 vysílat devátý (paritní) bit. Tento bit se ovládá bitem TB8 pro vysílání a v RB8 je hodnota devátého bitu při příjmu. Je plně v režii uživatele, jaká hodnota devátého bytu bude vyslána.

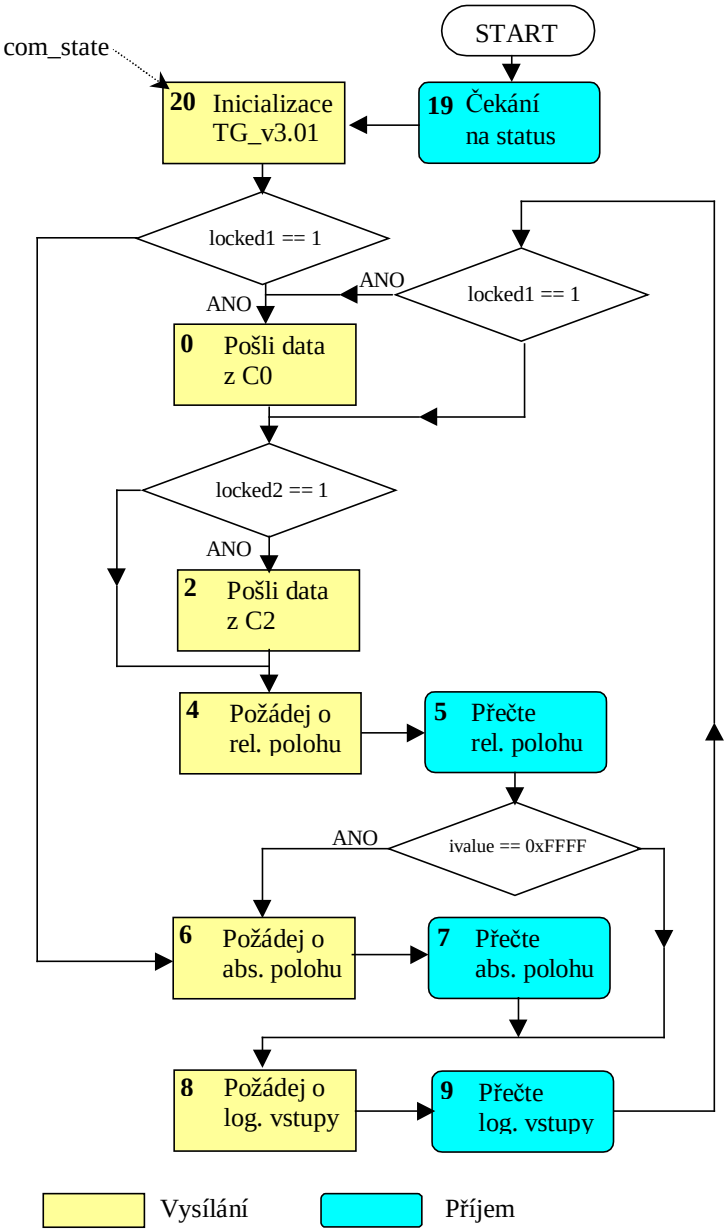
Při komunikaci je devátý byt využit jako příznak určující, zda se jedná o příkaz (TB8=0) či o data (TB8=1). Tento způsob je myšlen tak, aby jakýkoliv příchod příkazu, kdy devátý byt je nulový přerušil předchozí komunikaci ať je v jakékoliv fázi rozpracování. Seznam příkazů a formát zasílaných dat určený pro komunikaci s TG_v3.01 je uveden v příloze IV.

6.2.3.1 Komunikace na straně PDB

U těchto typů rozhraní, je potřeba dosáhnout co nejvyšší komunikační rychlosti a proto je obsluha RS232 umístěna pod přerušením. Rozhraní komunikace tvoří struktura rs232_buf. Ta obsahuje komunikační buffery pro zasílání příkazů a dat (*out[2][10]* - data z C0, *out2[2][10]* – data z C2 a *out_ini[2][25]* – inicializační sekvence příkazů). Tyto buffery jsou koncipovány obecně a to tak, že *buffer[0] […]* obsahuje odesílaná data a *buffer[1] […]* říká, zda se jedná o příkaz či o data. Na základě *buffer[1] […]* se pak obsluha komunikace nastavuje bit TB8. Všechny komunikační buffery ve struktuře rs232_buf jsou navíc charakterizovány dvěmi proměnnými. První je délka zprávy aktuálně uložené v bufferu (*out_len*, *out_len2* a *out_len_ini*). Druhá se týká pouze bufferů *out* a *out2* a je příznakem určující jejich dostupnost (*locked*, *locked2*). Pokud tento příznak má hodnotu 0, je buffer přístupný a je možno do něj vložit zprávu k odeslání. V opačném případě obsahuje hodnotu 1 a to znamená že buffer je používán k vysíláni a nesmí se do něj zapisovat.

Struktura *rs232_buf* dále obsahuje pomocné proměnné pro výměnu dat a watchdog (*com_watchdog*) pro monitorování stavu přerušení a tím potažmo komunikace. Dále pak příznak, zda je komunikace aktivní (*com_live*).

Obsluha přerušení je koncipována jako stavový automat, kde aktuální stav je definován proměnnou *com_state*. Přechod mezi stavy je vysvětlen vývojovým diagramem na obrázku 6-7.



Obr. 6-7 Algoritmus komunikace se zařízením TG_v3.01

Inicializace sériového rozhraní je provedena ve funkci *void usr_main(void)*. Nejprve je inicializována přenosová rychlost voláním funkce *void ini_serial(void)*. Tato funkce zároveň provede inicializaci struktury *rs232_buf*. Konkrétně nastaví počáteční stav obsluhy přerušení (*com_state* = 19) a odemkne oba buffery pro C0 a C2. Voláním funkce *void dev_startup(void)* je do bufferu *rs232_buf.out_ini* vložena inicializační zpráva pro TG_v3.01. Poslední inicializační metodou před povolením přerušení je *void dev_diag(void)*. Tato funkce jako jediná (kromě obsluhy přerušení od sériového kanálu) zapisuje přímo na sériový port. Proto je na jejím začátku zakázáno přerušení od sériové linky. Následně je zaslán TG_v3.01 příkaz na zjištění jeho stavu. Potom je inicializována struktura *rs232_buf* a je opět povoleno přerušení.

Proměnná *rs232_buf.watchdog* je čítač, který je inkrementován, kdykoliv je přijat či odvysílán nějaký znak. V hlavní programové smyčce je jeho hodnota monitorována a pokud hodnota stagnuje, je voláním funkce *void dev_diag(void)* vyvolána diagnostika TG_v3.01 a restart obsluhy komunikace. O úspěšně probíhající datové výměně mezi PDB a TG_v3.01 vypovídá proměnná *rs232_buf.com_live*. Pokud je její hodnota rovna jedné je vše v pořádku. Pokud je rovna nule, spojení neexistuje.

Vlastní propojení komunikačního rozhraní RS232 a profilu spočívá v tom, že je konec funkce *void block_comp_in(void)* je rozšířen o plnění bufferu *rs232_buf.out* potřebnými proměnnými z převodního bloku (*transducer_block.positioning_value.value.fvalue* (poloha) a *transducer_block.io_output* (log. výstupy na TG_v3.01)). Přenos v opačném směru, tzn. dat z Tgv_3.01 do PDB, je ošetřen na začátku funkce *void block_comp_out(void)*. Zde je naplněna proměnná:

```
transducer_block.feedback_value.value.fvalue = rs232_buf.feedback;
```

Protože proměnná logických vstupů (*transducer_block.io_input*) je velikosti byte, její hodnota je do převodního bloku zapisována přímo z obsluhy přerušení. Tento přístup totiž nemůže způsobit nekonzistenci dat.

Buffer *out2* určený pro komunikaci přes C2 (ale i C1) je využíván ve funkci *void usr_write_ds(V1SL_LL_DS_WRITE_PTR ds_write_ptr*). Princip je stejný jako u C0. Veškeré operace se strukturou rs232_buf, které mohou způsobit nekonzistenci dat, musejí být prováděny při zakázané obsluze přerušení.

Před vlastním připojením kontroléru TG_v3.01 byla komunikace ověřena jednoduchým programem na PC, který emuloval TG_v3.01. Testovací program je koncipován jako jednoduchá konzolová aplikace v prostředí Visual C++. Program nepodporuje 9-bitovou komunikaci ale pro účely odzkoušení je postačující. Pro tyto účely byl sériový kanál v PDB přepnut v režimu 1, tzn. komunikace bez devátého bitu. Tato aplikace se opět nachází na přiloženém CD v adresáři `Software\Motor`.

6.2.3.2 Komunikace na straně TG_v3.01

Úprava komunikace v kontroléru TG_v3.01 nebylo přímo součástí mé práce ale abych dosáhl stavu plně funkčního zařízení, bylo potřeba provést úpravy i v něm. Úkol spočíval v modifikaci stávající obsluhy komunikace, konkrétně rozšíření komunikace o devátý bit a změna komunikace z ASCII na binární způsob výměny dat. Jádrem TG_v3.01 je procesor DSP56803. Stávající software je psán v jazyce C s využitím knihovny pro procesory Motorola DSP568XX Embedded SDK 2.4. Vývojovým prostředím mi byl překladač a debuger CodeWarrior 5.0 firmy Metrowerks.

Změna softwaru spočívala v modifikaci těla obsluhy přerušení od sériového kanálu a úpravy jeho inicializace. Zde bylo potřeba povolit devítibitovou komunikaci a nastavit příslušnou komunikační rychlost. V těle přerušení je upraveno vyhodnocení příchozích příkazů podle specifikace v příloze IV. Protože většina vývojových nástrojů pro sériový kanál neumožňuje ovládat devátý byt, opět jsem si napsal jednoduchý ladící nástroj, pro zasílání příkazů TG_v3.01. Jedná se o grafickou aplikaci vyvinutou v Prostředí Visual C++. Ta umožňuje posílat kontroléru základní příkazy (na přiloženém CD v adresáři `Software\TGDrive`).

6.3 Závěr

V této části mé práce bylo hlavním úkolem odzkoušet knihovnu V1SL na PDB. Tím byla ověřena funkčnost hardwaru a to jak na Profibusu PA tak i DP. Vzorový příklad aplikace k V1SL jsem pak upravil a rozšířil na konkrétní zařízení, čímž bylo vytvořeno plně funkční zařízení. Již při vlastním návrhu bylo jasné, že především v aplikaci na Profibus PA bude problém s komunikační rychlostí sériového kanálu. Problém je v tom, že musí být použít krystal 2, 4, 8 nebo 16 MHz. Dělením tohoto kmitočtu není snadné dosáhnout standardní komunikační rychlosti. Samozřejmě je možné použít i nestandardní. Pomocí uživatelských parametrů v profilu lze nastavit kteroukoliv rychlost, kterou procesor při použitém krystalu dokáže vygenerovat ale to již snižuje schopnost převodníku připojit ho k obecnému zařízení. Konkrétně v mém případě byla použita přenosová rychlost 1200 bit/s při komunikaci na Profibusu PA a 14400 bit/s na Profibusu DP.

Po odzkoušení jsem se zabýval měřeními, kterými bylo potřeba ověřit myšlenku převodník Profibus/RS232. Program z přílohy VIII vytvořený v prostředí Simatic Step 7 skrze Profibus a PDB přiváděl na logický výstup kontroléru periodický signál (0,5s log. 1, 3s log 1). Software v PDB jsem upravil tak, aby při příchodu nových dat, došlo ke změně úrovně (komplement) signálu T0. Na osciloskopu jsem pak měřil rychlost odezvy kontroléru. Tento údaj je směrodatný, protože PDB zasílá tyto data do TG_v3.01 jako poslední. Zpoždění, která jsem naměřil byla typicky 10ms (maximálně 16ms). To vše při PDB v režimu Profibus DP (f_{SYS}=48MHz) a přenosové rychlost sériového kanálu 14400 bit/s (rychlost sběrnice Profibus 93,75Kbit/s).

Je patrné že toto řešení bezpečně vyhoví pro aplikace Profibus PA ale u náročnějších aplikací Profibusu DP, obzvláště při vyšších komunikačních rychlostech bude potřeba použít pružnější rozhraní. Jako řešení se vybízí rozhraní SPI, u kterého by nebyl problém s komunikační rychlostí jako u sériového kanálu. Navíc obě zařízení jak PBC, tak i TG_v3.01 tímto rozhraním disponují.

7. Závěr

Výsledkem této diplomové práce je plně funkční zařízení PDB, které umožňuje vyvíjet jednoduché i složitější aplikace na průmyslovou sběrnici Profibus DP/PA. Velkou předností je velice nízký odběr, který jej dovoluje využít i v prostorech s nebezpečím výbuchu. Provedení SMD a snížené napájecí napětí na 3,3V řadí PDB mezi zařízení s moderní koncepcí. Výrobní náklady nepřesahují 2 000,-Kč, což je zlomek ve srovnání s podobnými výrobky zahraničních firem. Jejich ceny se pohybují okolo 100 000,-Kč.

Vedle toho se povedlo odzkoušet PDB jako rozhraní ke kontroléru TG_v3.01. Je ho však možné použít i v mnoha jiných případech. Jednak jako vývojový prostředek, pro vývoj vlastní aplikace, nebo jako funkční rozhraní k již existujícím zařízením.

Uvítal bych, kdyby tato práce byla dále rozvíjena a prohlubována. Při vývoji zařízení se vyskytlo několik nedostatků, které je třeba další verzí odstranit. Navrhuji tyto úpravy:

1) Jinak navrhnout řídící logiku sběrnice PDB s ohledem na signál DBX.
2) Nepoužité vývody vnitřní sběrnice obvodu DPC31 zapojit přes pull-up rezistory a případně je vyvést na konektor.
3) Obvod SIM1 opět zapojit v režimu, kdy odebírá ze sběrnice konstantní výkon a komunikaci s DPC31 realizovat pomocí energeticky méně náročného zapojení (se dvěma optočleny a externím komparátorem).
4) V nové verzi plošného spoje opravit chybu se špatně očíslovanými signály na konektorech CANNON.
5) Bylo by dobré vybavit rozhraní PDB indikačními LED diodami (zelená a červená), které signalizují stav komunikace. Také by byl užitečný přepínač (zkratovací propojka), pomocí kterého by bylo možno zamykat zařízení proti zápisu (WRITE_LOCKING).
6) Využít pro komunikaci s TG_v3.01 (či jiným zařízením) rozhraní SPI místo RS232, aby byla urychlena reakce periférie.
7) Komunikaci s externím rozhraním galvanicky oddělit, aby byly splněny požadavky na jiskrovou bezpečnost pro libovolné periferní zařízení. Vedle toho by se tím zamezilo vniku rušivých signálů na desku, které mohou poškodit informace uložené v paměti RAM.

8. Použitá literatura

[1] PROFIBUS SPECIFICATION, EN 50 170, 1996

[2] PROFIBUS-DP Extension to EN50170(DPV1), Version 2.0, PNO, duben 1998

[3] PROBUS-PA, Profile for Process Control Devices, Version 3.0, PNO , říjen 1999

[4] PROFIBUS-PA User and Installation Guideline, Version 1.1, PNO, 09/1996

[5] Manfred Popp, The Rapid Way to Profibus-DP, PROFIBUS Nutzeroroganisation, Germany 1997

[6] V1SL user description, Firmware for Siemens ASIC DPC31 DPV1, Version 1.0, SIMATIC NET, 1999

[7] V1SL getting started, DPC31 Slave Firmaware, Version 1.0, SIMATIC NET, 1999

[8] Jan Pohludka, Jaromír Hrubý, Elektrická zařízení v prostorách s nebezpečím výbuchu hořlavých plynů, par a prachů, IN-EL, Praha 1999

[9] Vladimír Šíma, Jan Pohludka, Elektrické rozvody a vnější vlivy, II Díl – Nebezpečí výbuchu hořlavých plynů a par, IN-EL, Praha 1997

[10] DPC31 Siemens PROFIBUS Controller, Hardware description, Version 0.2, SIMATIC NET, 1999.

[11] SIM1 user description, Version 1.1, SIMATIC NET, 1997

[12] SIM1 extension to the user description, Version 1.1, SIMATIC NET, 2000

[13] Petr Smolík, Průmyslová sběrnice Profibus DP Extended, Diplomová práce, ČVUT v Praze, 2001

[14] Miroslav Ličko, Průmyslová sběrnice Profibus, Diplomová práce, ČVUT v Praze, 2000

[15] DALLAS Semiconductor, katalogové listy DS1836/A/B/C/D, www.maxim-ic.com

[16] National Semiconductor, katalogové listy LM1085

[17] Halo Electoronic, katalogové listy TGM-040P3, www.halolectronics.com

[18] BSI, katalogové listy BS62LV1024

[19] Agilent Technologies, katalogové listy HCPL-0721, HCPL-0600, HCPL-0700, HCPL-070A, www.agilent.com

[20] Atmel, katalogové listy AT90S8515, www.atmel.com

[21] Motorola, DSP56F801/803/805/807 User's Manual, www.motorola-dsp.com

[22] Elatec, katalogové listy displejů ELXXXX, www.ElatecEurope.cz

[23] Profibus, internet, www.profibus.com

[24] Profibus CZ, internet, www.profibus.cz

[25] PonyProg seriál device programmer, internet, www.lancos.com

[26] Keil, internet, www.keil.com

[27] MSDN, internet, www.microsoft.com

[28] CSerialPort, a freeware, internet, http://indigo.ie/~pjn

[29] Milan Babák, Ladislav Chládek, Architektura a technické vlastnosti jednočipových mikrořadičů 8051, TESLA ELTOS, Praha 1991

Dodavatelé součástek a výrobci plošných spojů

[30] GM Electoronic, www.gme.cz

[31] Spezial Electoronic, www.spezial.cz

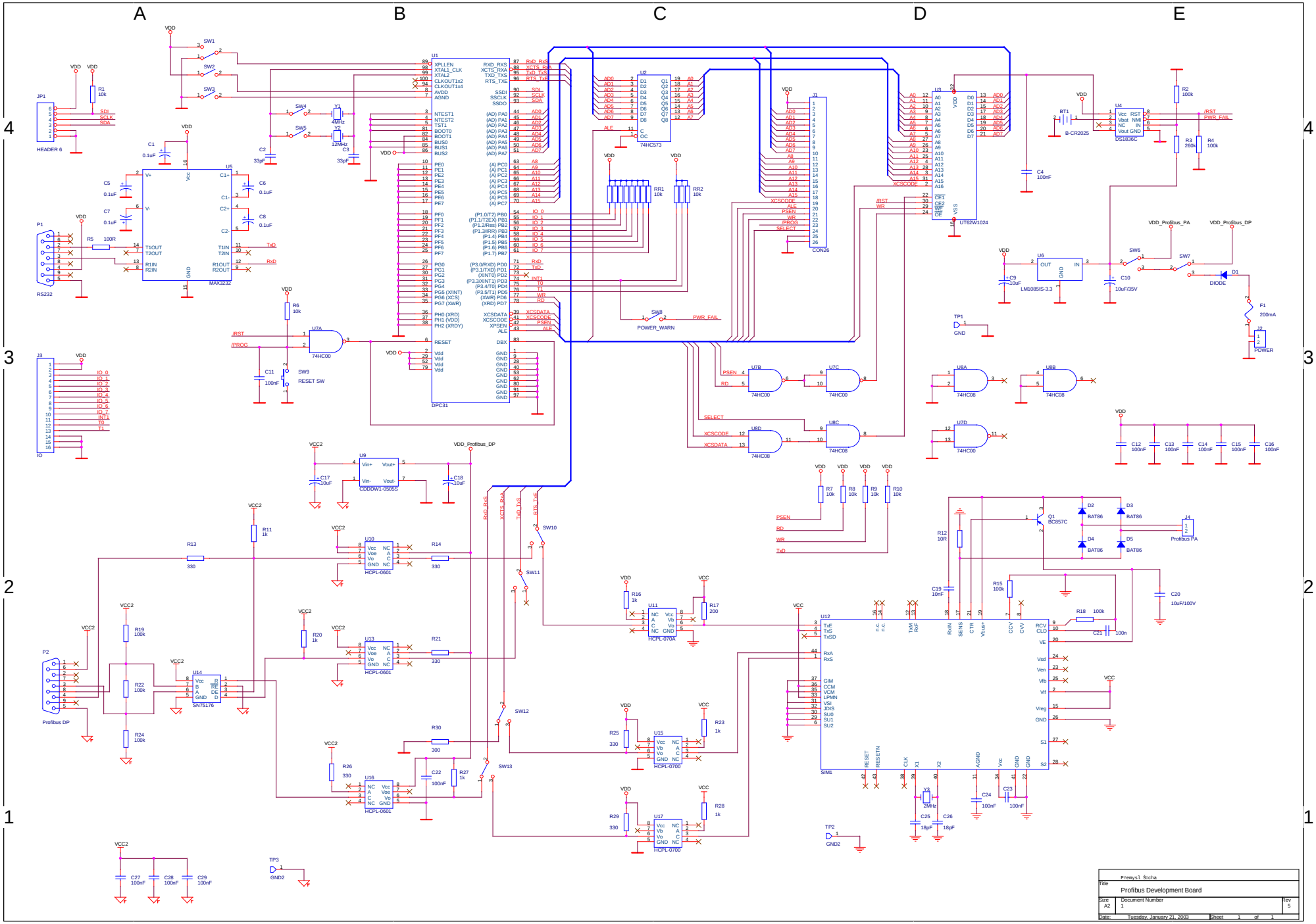
[32] RYSTON Electoronic, www.ryston.cz

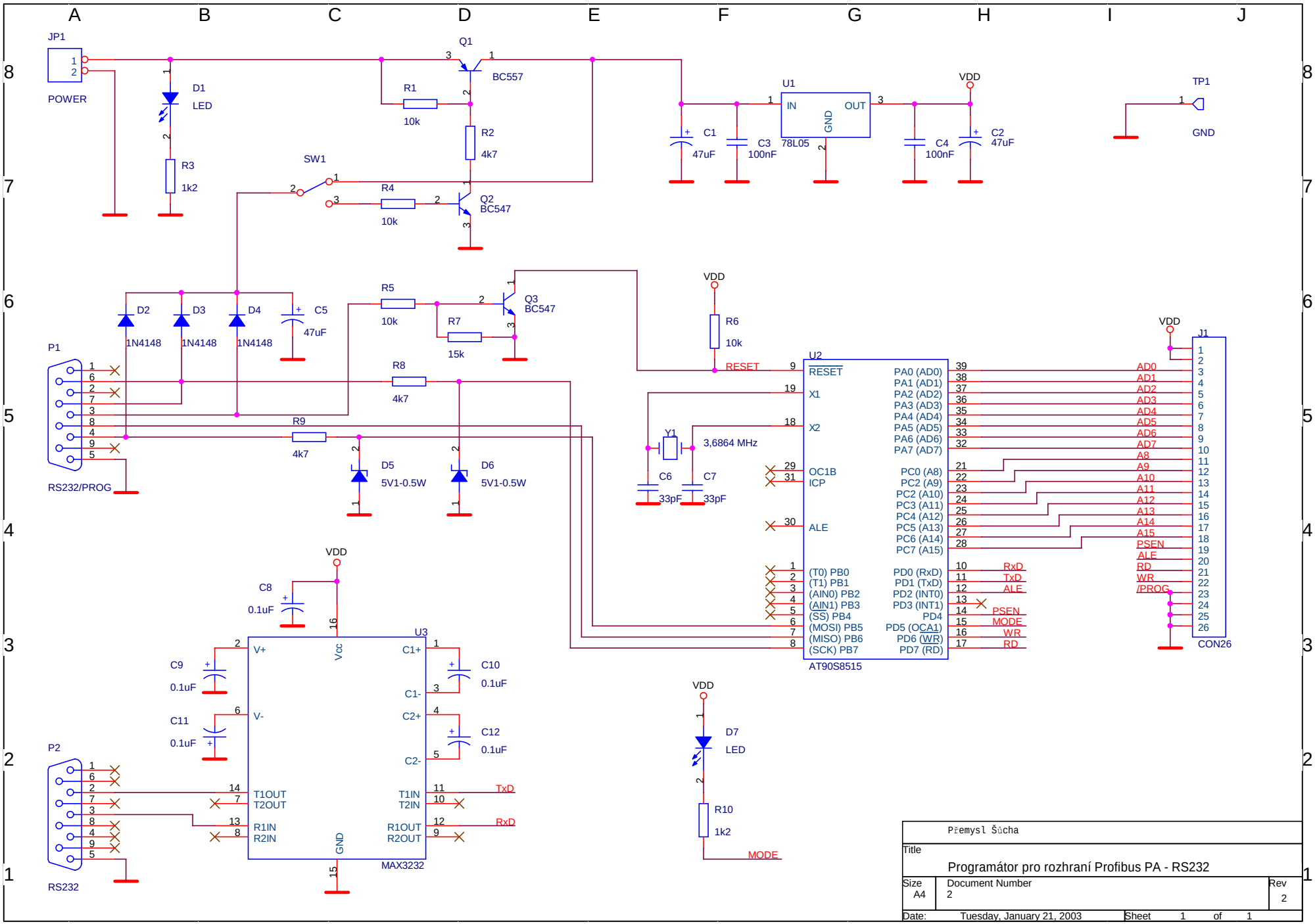
[33] Spoerle Electoronic, www.spoerle.cz

[34] Vladimír Kohout , www.volny.cz/plspoj

[35] Plošné spoje Korunní, www.volny.cz/psk

Přílohy


U1
DPC31
U2
74HC573
U3
UT62W1024
CON26
U4
DS1836C
U5
MAX3232
U6
LM1085IS-3.3
U7A
74HC00
U7B
74HC00
U7C
74HC00
U7D
74HC00
U8A
74HC08
U8B
74HC08
U8C
74HC08
U8D
74HC08
U9
CDDDW1-0505S
U10
HCPL-0601
U11
HCPL-070A
U12
SIM1
U13
HCPL-0601
U14
SN75176
U15
HCPL-0700
U16
HCPL-0601
U17
HCPL-0700
JP1
HEADER 6
P1
RS232
J3
IO
P2
Profibus DP
J4
Profibus PA
J2
POWER
D1
DIODE
F1
200mA
Q1
BC857C
D2
D3
D4
D5
BAT86
BT1
B-CR2025
Y1
4MHz
Y2
12MHz
Y3
2MHz
SW1
SW2
SW3
SW4
SW5
SW6
SW7
SW8
SW9
SW10
SW11
SW12
SW13
POWER_WARN
RESET SW
TP1
GND
TP2
GND2
TP3
GND2
VDD
VDD_Profibus_PA
VDD_Profibus_DP
VCC
VCC2
Przemysl Sicha
Profibus Development Board
Size A2
Document Number 1
Rev 5
Date: Tuesday, January 21, 2003
Sheet 1 of 1


JP1
POWER
D1
LED
R3
1k2
Q1
BC557
R1
10k
R2
4k7
SW1
R4
10k
Q2
BC547
U1
IN
OUT
GND
78L05
C1
47uF
C3
100nF
C4
100nF
C2
47uF
VDD
TP1
GND
D2
1N4148
D3
1N4148
D4
1N4148
C5
47uF
R5
10k
Q3
BC547
R7
15k
R6
10k
P1
RS232/PROG
R8
4k7
R9
4k7
D5
5V1-0.5W
D6
5V1-0.5W
U2
RESET
X1
X2
OC1B
ICP
ALE
Y1
3,6864 MHz
C6
33pF
C7
33pF
PA0 (AD0)
PA1 (AD1)
PA2 (AD2)
PA3 (AD3)
PA4 (AD4)
PA5 (AD5)
PA6 (AD6)
PA7 (AD7)
PC0 (A8)
PC2 (A9)
PC2 (A10)
PC3 (A11)
PC4 (A12)
PC5 (A13)
PC6 (A14)
PC7 (A15)
(T0) PB0
(T1) PB1
(AIN0) PB2
(AIN1) PB3
(SS) PB4
(MOSI) PB5
(MISO) PB6
(SCK) PB7
PD0 (RxD)
PD1 (TxD)
PD2 (INT0)
PD3 (INT1)
PD4
PD5 (OCA1)
PD6 (WR)
PD7 (RD)
AT90S8515
J1
CON26
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
A8
A9
A10
A11
A12
A13
A14
A15
PSEN
ALE
RD
WR
/PROG
RxD
TxD
ALE
PSEN
MODE
WR
RD
C8
0.1uF
U3
V+
Vcc
C1+
C1-
C2+
C2-
V-
T1OUT
T2OUT
R1IN
R2IN
T1IN
T2IN
R1OUT
R2OUT
GND
MAX3232
C9
0.1uF
C10
0.1uF
C11
0.1uF
C12
0.1uF
P2
RS232
D7
LED
R10
1k2
MODE
Přemysl Šůcha
Title
Programátor pro rozhraní Profibus PA - RS232
Size
A4
Document Number
2
Rev
2
Date:
Tuesday, January 21, 2003
Sheet
1
of
1

III. Profil – Řízení polohy (Actuator)

Device manager – DM (Slot=1, Index=0)

Rel. Index	Parametr	Typ.	Datový typ	Stor.	Vel.	Acc.	Trans.	Implicitní hodnota	M/O	Význam
0	Directory Header *-Dir_ID* *-Rev_Number* *-Nul_Dir_Obj* *-Num_Dir_Entry* *-First_Comp_List_Dir_Entry* *-Num_Comp_List_Dir_Entry*	Arr.	U16	C	6	r	a	 00,00, 00,01, 00,01, 00,06, 00,01, 00,03	M	 Nepoužito Číslo revize adresáře. Počet objektů v adresáři. Počet položek v následující struktuře Ukazatel na Composite_List_Directory_Entry Počet položek v Composite_List_Directory_Entry
1	Composite_List_Directory_Entry/Compsite_Directory_Entries *-Begin_PB (Index,Offset)* *-Num_PB* *-Begin_TB (Index,Offset)* *-Num_TB* *-Begin_FB (Index,Offset)* *-Num_FB* *-PB (Index,Slot)* *-Num_elem* *-TB (Index,Slot)* *-Num_elem* *-FB (Index,Slot)* *-Num_elem*	Arr.	U16	C	12	r	a	 01,04, 00,01, 01,05, 00,01, 01,06, 00,01, 01,160, 00,33, 01,70, 00,81, 01,16, 00,51	M	Umístění bloků v profilu
2-8	Directory_Continuous									
9-13	Reserved									

Fyzický blok – PB (Slot=1, Index=160)

Rel. Index	**Parametr**	**Typ.**	**Datový typ**	**Stor.**	**Vel.**	**Acc.**	**Trans.**	**Implicitní hodnota**	**M/O**	**Význam**
0	BLOCK_OBJECT	Rec.	DS-32	C	20	r	C/a		M	Charakteristika bloku
	-Reserved		(U8)					250		Vyhrazeno
	-Block Object		(U8)					01		Typ bloku
	-Parent Class		(U8)					02		Typ zařízení
	-Class		(U8)					250		Bližší specifikace typu zařízení
	-DD Reference		(U32)					0		Pro využití v příští verzi
	-DD Revision		(U16)					0		Pro využití v příští verzi
	-Profile		(Ostr.2)					0x40,0x82		Číslo profilu a typ zařízení
	-Profile Revision		(U.16)					3		Revize profilu
	-Execution Time		(U8)					0		Pro využití v příští verzi
	-Number_of _Parameters		(U16)					33		Počet parametrů
	-Address of VIEW_1		(U16)					01,193		Adresa VIEW objektu
	-Number of Views		(U8)					1		Počet VIEW objektů v bloku
1*	ST_REV	Smp.	U16	N	2	r	C/a	0	M	Čítač změn statických prom. (S) v bloku
2	TAG_DESC	Smp.	OctetStr	S	32	r,w	C/a	“ “	M	Jednoznačný popis
3	STRATEGY	Smp.	U16	S	2	r,w	C/a	0	M	Určeno pro spojování bloků
4	ALERT_KEY	Smp.	U8	S	1	r,w	C/a	0	M	Identifikační číslo
5	TARGET_MODE	Smp.	U8	S	1	r,w	C/a	AUTO	M	Mód bloku požadovaný řídící aplikací
6*	MODE_BLK	Rec.	DS-37	D	3	r	C/a		M	Mód bloku
	-Actual		(U8)					AUTO		Aktuální mód bloku
	-Permitted		(U8)					AUTO		Podporované módy
	-Norman		(U8)					AUTO		Mód během normální operace.
7*	ALARM_SUM	Rec.	DS-42	D	8	r	C/a		M	Stav alarmů bloku
	-Current		(Ostr.2)					0		1, pokud dojde k alarmu (po 10s je nulován)
	-Unacknowledged		(Ostr.2)					0		Pro využití v příští verzi
	-Unreported		(Ostr.2)					0		Pro využití v příští verzi
	-Disabled		(Ostr.2)					0		Pro využití v příští verzi
8	SOFTWARE_REVISION	Smp.	VStr.	Cst	16	r	C/a	“1.0”	M	Číslo revize softwaru
9	HARDWARE_REVISION	Smp.	VStr.	Cst	16	r	C/a	“1.0”	M	Číslo revize hardwaru
10	DEVICE_MAN_ID	Smp.	VStr.	Cst	16	r	C/a	251	M	Identifikační číslo výrobce.
11	DEVICE_ID	Smp.	VStr.	Cst	16	r	C/a	-	M	Identifikační číslo zařízení dané výrobcem.
12	DEVICE_SER_Num	Smp.	VStr.	Cst	16	r	C/a	0001	M	Sériové číslo zařízení
13*	DIAGNOSIS	Smp.	Ostr.	D	4	r	C/a	0	M	Diagnostické informace o zařízení
14	DIAGNOSIS_EXTENSION	Smp.	Ostr.	D	6	r	C/a	0	O	Rozšířené diag. informace (spec. výrobcem)

15	DIAGNOSIS_MASK	Smp.	Ostr.	Cst.	4	r	C/a	0	M	Podporované diagnostické bity (1=podporována, 0=nepodporováno)
16	DIAGNOSIS_MASK_EXT.	Smp.	Ostr.	Cst.	6	r	C/a	0	O	
17	DEVICE_CERTIFICATION	Smp.	Vstr.	Cst.	32	r	C/a	-	O	Certifikát zařízení (např. EX certifikát)
18	WRITE_LOCKING	Smp.	U16	N	2	r,w	C/a	2457	O	0-zápis do profilu je zakázán (kom této prom.)
19	FACTORY_RESET	Smp.	U16	S	2	r,w	C/a	0	O	1-reset, 2506-warmstart, 2712-res. adresy
20	DESCRIPTOR	Smp.	Ostr.	S	32	r,w	C/a	-	O	Uživatelský text
21	DEVICE_MESSAGE	Smp.	Ostr.	S	32	r,w	C/a	-	O	Uživatelský text
22	DEVICE_INSTAL_TIME	Smp.	Ostr.	S	16	r,w	C/a	-	O	Datum instalace zařízení
23	LOCAL_OP_ENA	Smp.	U8	N	1	r,w	C/a	-	O	Režim komunikace s lokálním terminálem.
24	IDENT_NUMBER_SEL.	Smp.	U8	S	1	r,w	C/a	1	M(B)	Volba identifikačního čísla
25	HW_WRITE_PROT.	Smp.	U8	D	1	r	C/a	-	O	Stav hw. přepínače, který zamyká profil. 0-odemčeno, 1-zamčeno
26-32	Reserved									
33	View object (*)						C/a			
34	RS232_PARAMETERS - Clock - Baud	Rec.	 (U8) (U32)	N	5	r	C/a	 4 1200	U	Parametry sériového kanálu Frekvence krystalu [MHz] Aktuální vypočítaná přenosová rychlost
35	RS232_SETTINGS - Mode - Smod - TH1	Rec.	 (U8) (U8) (U8)	S	3	r,w	C/a	 3 1 247	U	Nastavení sériového kanálu Mód sériového portu (2 nebo 3) Dvojnásobná přenosová rychlost Rychlost přenosu daná rychlostí přetečení časovače T1 (mód 2)
36	RS232_REC_WATCHDOG	Smp.	(U16)	S	2	r,w	C/a	20	U	Watchdog pro sériovou komunikaci

Funkční blok – FB (Slot=1, Index=16)

Rel. Index	Parametr	Typ.	Datový typ	Stor.	Vel.	Acc.	Trans.	Implicitní hodnota	M/O	Význam
0	BLOCK_OBJECT	Rec.	DS-32	C	20	r	C/a		M	Charakteristika bloku
	-Reserved		(U8)					250		Vyhrazeno
	-Block Object		(U8)					02		Typ bloku
	-Parent Class		(U8)					03		Typ zařízení
	-Class		(U8)					01		Bližší specifikace typu zařízení
	-DD Reference		(U32)					0		Pro využití v příští verzi
	-DD Revision		(U16)					0		Pro využití v příští verzi
	-Profile		(Ostr.2)					0x40,0x82		Číslo profilu a typ zařízení
	-Profile Revision		(U.16)					3		Revize profilu
	-Execution Time		(U8)					0		Pro využití v příští verzi
	-Number_of _Parameters		(U16)					51		Počet parametrů
	-Address of VIEW_1		(U16)					01,67		Adresa VIEW objektu
	-Number of Views		(U8)					1		Počet VIEW objektů v bloku
1*	ST_REV	Smp.	U16	N	2	r	C/a	0	M	Čítač změn statických prom. (S) v bloku
2	TAG_DESC	Smp.	Ostr.	S	32	r,w	C/a	“ “	M	Jednoznačný popis
3	STRATEGY	Smp.	U16	S	2	r,w	C/a	0	M	Určeno pro spojování bloků
4	ALERT_KEY	Smp.	U8	S	1	r,w	C/a	0	M	Identifikační číslo
5	TARGET_MODE	Smp.	U8	S	1	r,w	C/a	AUTO	M	Mód bloku požadovaný řídící aplikací
6*	MODE_BLK	Rec.	DS-37	D	3	r	C/a		M	Mód bloku
	-Actual		(U8)					AUTO		Aktuální mód bloku
	-Permitted		(U8)					O/S||MAN|AUTO		Podporované módy
	-Norman		(U8)					AUTO		Mód během normální operace.
7*	ALARM_SUM	Rec.	DS-42	D	8	r	C/a		M	Stav alarmů bloku
	-Current		(Ostr.2)					0		1, pokud dojde k alarmu (po 10s je nulován)
	-Unacknowledged		(Ostr.2)					0		Pro využití v příští verzi
	-Unreported		(Ostr.2)					0		Pro využití v příští verzi
	-Disabled		(Ostr.2)					0		Pro využití v příští verzi
8	BATCH	Rec.	DS-67	S	10	r,w	C/a		M	Pro využití v tzv. „Batch“ aplikacích
	-BATCH_ID		(U32)					0		
	-RUP		(U16)					0		
	-OPERATION		(U16)					0		
	-PHASE		(U16)					0		
9	SP	Rec.	DS-33	D	5	r,w	I/cyc	-	M(A,B)	Požadovaná poloha v režimu AUTO. Zadává se
	-Value		(Float)							v jednotkách daných PV_SCALE.

	-Status		(U8)							
11	PV_SCALE *-EU at 100%* *-EU at 0%* *-Units Index* *-Decimal point*	Rec.	DS-36 Float Float U16 I8	S	11	r,w	C/a	100,0%	M(A,B)	Parametry pro přepočet SP (stroj. jedn - %) -horní rozsah -spodní rozsah -jednotky (např. 1013 jsou milimetry) -Počet platných digitů od desetinné čárky
12*	READ_BACK *-Value* *-Status*	Rec.	DS-33 (Float) (U8)	D	5	r	O/cyc	-	M(B)	Aktuální pozice v jednotkách PV_SCALE
14	RCAS_IN *-Value* *-Status*	Rec.	DS-33 (Float) (U8)	D	5	r,w	I/cyc	-	O(B)	Požadovaná poloha v jednotkách PV_SCALE v režimu RCas.
21	IN_CHANEL	Smp.	U16	S	2	r,w	C/a	-	M(B)	Odkaz na používaný převodní blok
22	OUT_CHANEL	Smp.	U16	S	2	r,w	C/a	-	M(B)	Odkaz na používaný převodní blok
23	FSAFE_TIME	Smp.	Float	S	4	r,w	C/a	0	M(B)	Čas v sekundách pro detekci selhání
24	FSAFE_TYPE	Smp.	U8	S	1	r,w	C/a	1	M(B)	Reakce zařízení na selhání požad. polohy i po uplynutí času FSAFE_TIME 0=pro SP je použita hodnota FSAFE_VALUE 1=používá poslední hodnotu SP 2=pozice je dána ACTUATOR_ACTION (TB)
25	FSAFE_VALUE	Smp.	Float	S	4	r,w	C/a	0	M(B)	SP s případě selhání (FSAFE_TYPE=1)
27	RCAS_OUT *-Value* *-Status*	Rec.	DS-33 (Float) (U8)	D	5	r	O/cyc		O(B)	Požadovaná poloha ve FB v jednotkách PV_SCALE
31*	POS_D *-Value* *-Status*	Rec.	DS-34 (U8) (U8)	D	2	r	O/cyc		M(B)	Diskrétní hodnota aktuální polohy 0=neinicializováno., 1=zavřeno,2=otevřeno,3=mezi.
32	SETP_DEVITATION	Smp.	Float	D	4	r	C/a	-	O(B)	Odchylka skutečné polohy READ_BACK a požadované polohy SP v % z rozsahu
33*	CHECK_BACK	Smp.	Ostr.	D	3	r	O/cyc	-	M(B)	Diagnostické informace o zařízení
34	CHECK_BACK_MASK	Smp.	Ostr.	Cst.	3	r	C/a	-	M(B)	Diag. informace podporované zařízením.
35	SIMULATE *-Simulate_Status* *-Simulate_Value* *-Simulate_Enabled*	Rec.	DS-50 U8 float U8	S	6	r,w	C/a	 disabled	M(B)	Parametry pro simulaci aktuální polohy READBACK
36	INCREASE_CLOSE	Smp.	U8	S	1	r,w	C/a	0	M(B)	směr pohybu v režimech Rcas and AUTO. 0=pozitivní směr, 1=negativní směr
37	OUT	Rec.	DS-33	D	5	r,w	C/a	-	M(B)	Požadovaná poloha ve strojových jednotkách..

	-Value *-Status*		(Float) (U8)							Výstupní proměnná bloku
38	OUT_SCALE *-EU at 100%* *-EU at 0%* *-Units Index* *-Decimal point*	Rec.	DS-36 Float Float U16 I8	S	11	r,w	C/a	-	M(A,B)	Parametry pro přepočet OUT (% - stroj. jedn.) -horní rozsah -spodní rozsah -jednotky (např. 1013 jsou milimetry) -Počet platných digitů od desetinné čárky
39-48	Reserved								M(A,B)	
49	First man. spec. par.								O(A,B)	
50	View object (*)						C/a			

Převodní blok – TB (Slot=1, Index=70)

Rel. Index	Parametr	Typ.	Datový typ	Stor.	Vel.	Acc.	Trans.	Implicitní hodnota	M/O	Význam
0	BLOCK_OBJECT *-Reserved* *-Block Object* *-Parent Class* *-Class* *-DD Reference* *-DD Revision* *-Profile* *-Profile Revision* *-Execution Time* *-Number_of _Parameters* *-Address of VIEW_1* *-Number of Views*	Rec.	DS-32 (U8) (U8) (U8) (U8) (U32) (U16) (Ostr.2) (U.16) (U8) (U16) (U16) (U8)	C	20	r	C/a	 250 03 05 01 0 0 0x40,0x82 3 0 81 01,151 1	M	Charakteristika bloku Vyhrazeno Typ bloku Typ zařízení Bližší specifikace typu zařízení Pro využití v příští verzi Pro využití v příští verzi Číslo profilu a typ zařízení Revize profilu Pro využití v příští verzi Počet parametrů Adresa VIEW objektu Počet VIEW objektů v bloku
1*	ST_REV	Smp.	U16	N	2	r	C/a	0	M	Čítač změn statických prom. (S) v bloku
2	TAG_DESC	Smp.	Ostr	S	32	r,w	C/a	“ “	M	Jednoznačný popis
3	STRATEGY	Smp.	U16	S	2	r,w	C/a	0	M	Určeno pro spojování bloků
4	ALERT_KEY	Smp.	U8	S	1	r,w	C/a	0	M	Identifikační číslo
5	TARGET_MODE	Smp.	U8	S	1	r,w	C/a	AUTO	M	Mód bloku požadovaný řídící aplikací
6*	MODE_BLK *-Actual* *-Permitted* *-Norman*	Rec.	DS-37 (U8) (U8) (U8)	D	3	r	C/a	 AUTO AUTO AUTO	M	Mód bloku Aktuální mód bloku Podporované módy Mód během normální operace.
7*	ALARM_SUM -Current -Unacknowledged -Unreported -Disabled	Rec.	DS-42 (Ostr.2) (Ostr.2) (Ostr.2) (Ostr.2)	D	8	r	C/a	 0 0 0 0	M	Stav alarmů bloku 1, pokud dojde k alarmu (po 10s je nulován) Pro využití v příští verzi Pro využití v příští verzi Pro využití v příští verzi
8	ACT_ROT_DIR	Smp.	U8	S	1	r,w	C/a	0	O(B)	Směr pohybu 0=ve směru hod. ručiček (otevírání), 1=proti směru hodinových ručiček (zavírání)
9	ACT_STROKE_TIME_DEC	Smp.	Float	S	4	r	C/a	-	O(B)	Minimální čas potřebný k přesunu z polohy otevřeno do polohy zavřeno [s]
10	ACT_STROKE_TIME_INC	Smp.	Float	S	4	r	C/a	-	O(B)	Minimální čas potřebný k přesunu z polohy zavřeno do polohy otevřeno [s]

11	ACT_TRAV_TIME	Smp.	Float	S	4	r	C/a	-	O(B)	Detekce překročení počtu hodin v provozu
12	ANTI_PUMP_CL	Smp.	Float	S	4	r,w	C/a	-	O(B)	Min. vzdálenost od polohy otevřeno.U ventilů tento parametr zabraňuje „čerpání“ (pump effect).
13	ANTI_PUMP_OP	Smp.	Float	S	4	r,w	C/a	-	O(B)	Min. vzdálenost od polohy zavřeno.U ventilů tento parametr zabraňuje „čerpání“ (pump effect).
14	BREAK_STRENGTH	Smp.	Float	S	4	r,w	C/a	-	O(B)	Síla brzdy
15	BYPASS_SETP_CL	Smp.	Float	S	4	r,w	C/a	-	O(B)	Parametr pro rychlý přesun z polohy zavřeno do polohy otevřeno
16	BYPASS_SETP_OP	Smp.	Float	S	4	r,w	C/a	-	O(B)	Parametr pro rychlý přesun z polohy otevřeno do polohy zavřeno
17	TAB_ENTRY							-	O(B)	Proměnné tabulky určené pro linearizaci
18	TAB_X_Y_VALUE							-	O(B)	
19	TAB_MIN_NUMBER							-	O(B)	
20	TAB_MAX_NUBER							-	O(B)	
21	TAB_ACTUAL_NUMBER							-	O(B)	
22	DEAD_BAND	Smp.	Float	S	4	r,w	C/a	-	O(B)	Velikost mrtvého pásma v procentech z celkového pracovního rozsahu.
23	DEVICE_CALIB_DATE	Smp.	Ostr.	S	16	r,w	C/a	-	O(B)	Datum poslední kalibrace.
24	DEVICE_CONFIG_DATE	Smp.	Ostr.	S	16	r,w	C/a	-	O(B)	Datum poslední konfigurace
25	LIN_TYPE							0	M(B)	Typ linearizace (0-bez linearizace,1-tabulkou, …)
26	MAX_TORQUE	Smp.	Float	S	4	r,w	C/a	-	O(B)	Maximální povolený točivý moment
27	MOTOR_ON_TIME	Smp.	Float	D	4	r	C/a	-	O(B)	Celková doba v provozu (motoru) [hod]
28	NUM_LIMIT_CUT_OFF	Smp.	Float	D	4	r	C/a	-	O(B)	
29	NUM_MOT_ON_CYCL	Smp.	Float	D	4	r	C/a	-	O(B)	Celkový počet start-stop cyklů motoru
30	NUM_MOT_ON_HOUR	Smp.	U8	D	1	r	C/a	-	O(B)	Celkový počet start-stop cyklů motoru během poslední hodiny
31	NUM_TORQ_CUT_OFF	Smp.	Float	D	4	r	C/a	-	O(B)	
32	RATED_TRAVEL	Smp.	Float	S	4	r,w	C/a	-	M(B)	
33	SELF_CALIB_CMD	Smp.	U8	N	1	r,w	C/a	0	M(B)	Způsob inicializace zařízení
34	SELF_CALIB_STATUS	Smp.	U8	N	1	r	C/a	0	M(B)	Výsledek kalibrace
35	SERVO_GAIN_1	Smp.	Float	S	4	r,w	C/a	-	O(B)	Proporcionální konstanta serva
36	SERVO_RATE_1	Smp.	Float	S	4	r,w	C/a	-	O(B)	Diferenční konstanta serva
37	SERVO_RESET_1	Smp.	Float	S	4	r,w	C/a	-	O(B)	Sumační konstanta serva
38	SETP_CUTOFF_DEC	Smp.	Float	S	4	r,w	C/a	-	M(B)	
39	SETP_CUTOFF_INC	Smp.	Float	S	4	r,w	C/a	-	M(B)	

40	SETP_CUTOFF_MODE	Smp.	U8	S	1	r,w	C/a	3	O(B)	
41	TORQUE_ACTUAL	Rec.	DS-33	D	5	r	C/a	-	O(B)	Aktuální koutící moment
42	TORQUE_LIM_CL	Smp.	Float	S	4	r,w	C/a	-	O(B)	Maximální povolený kroutící moment pro směr zavírání
43	TORQUE_LIM_OP	Smp.	Float	S	4	r,w	C/a	-	O(B)	Maximální povolený kroutící moment pro směr otevírání
44	TORQUE_UNIT	Smp.	U16	S	2	r,w	C/a	-	O(B)	Jednotky kroutícího momentu nebo síly
45	TOTAL_VALVE_TRAVEL	Smp.	Float	D(S)	4	r,w	C/a	-	O(B)	Parametry pro ventily ventilu
46	TOT_VALVE_TRAV_LIM	Smp.	Float	S	4	r,w	C/a	-	O(B)	
47	TRAVLE_LIMIT_LOW	Smp.	Float	S	4	r,w	C/a	0	M(B)	
48	TRAVEL_LIMIT_UP	Smp.	Float	S	4	r,w	C/a	100	M(B)	
49	TRAVEL_RATE_DEC	Smp.	Float	S	4	r,w	C/a	-	M(B)	
50	TRAVEL_RATE_INC	Smp.	Float	S	4	r,w	C/a	-	M(B)	
51	VALVE_MAINT_DATE	Smp.	Ostr.	S	16	r,w	C/a	-	O(B)	Datum poslední údržby ventilu
55	TAB_OP_CODE							-	O(B)	Proměnné tabulky určené pro linearizaci
56	TAB_STATUS							-	O(B)	
57	POSITIONING_VALUE *-Value* *-Status*	Rec.	DS-33 (Float) (U8)	D	5	r	C/a	-	M(B)	Požadovaná poloha v jednotkách OUT_SCALE.
58	FEEDBACK_VALUE *-Value* *-Status*	Rec.	DS-33 (Float) (U8)	D	5	r	C/a	-	M(B)	Aktuální pozice v jednotkách OUT_SCALE.
59	VALVE_MAN	Smp.	Ostr.	S	16	r,w	C/a	-	M(B)	Jméno výrobce ventilu
60	ACTUATOR_MAN	Smp.	Ostr.	S	16	r,w	C/a	-	O(B)	Jméno výrobce pohonu
61	VALVE_TYPE	Smp.	U8	S	1	r,w	C/a	-	M(B)	Typ ventilu (1,2,3)
62	ACTUATOR_TYPE	Smp.	U8	Cst.	1	r	C/a	1	M(B)	Typ pohonu 0 = elektro-pneumatický 1 = elektrický 2 = elektro-hydraulický 3 = jiný
63	ACTUATOR_ACTION	Smp.	U8	S	1	r,w	C/a	-	M(B)	Bezpečná pozice pro případ výpadku napájení 0 = neinicializováno 1 = otevřený 2 = zavřený 3 = aktuální
64	VALVE_SER_NUM	Smp.	Ostr.	S	16	r,w	C/a	-	O(B)	Sériové číslo ventilu
65	ACTUATOR_SER_NUM	Smp.	Ostr.	S	16	r,w	C/a	-	O(B)	Sériové číslo pohonu
66	ADD_GEAR_SER_NUM	Smp.	Ostr.	S	16	r,w	C/a	-	O(B)	Sériové číslo převodovky
67	ADD_GEAR_MAN	Smp.	Ostr.	S	16	r,w	C/a	-	O(B)	Jméno výrobce převodovky

68	ADD_GEAR_ID	Smp.	Ostr.	S	16	r,w	C/a	-	O(B)	Identifikační číslo převodovky dané výrobcem
69	ADD_GEAR_INST_DATE	Smp.	Ostr.	S	16	r,w	C/a	-	O(B)	Datum instalace převodovky
70-79	Reserved								M(B)	
80	First man. spec.par.								O(B)	
81	View object (*)									
82	SPEED	Smp.	U32	S	4	r,w	C/a		U	Rychlost pojezdu
83	ACCEL	Smp.	U32	S	4	r,w	C/a		U	Zrychlení při rozjezdu a zastavování
84	IO_INPUT	Smp.	U8	D	1	r	C/a		U	Hodnota logických vstupů
85	IO_OUTPUT	Smp.	U8	S	1	r,w	C/a		U	Hodnota logických výstupů

Legenda

Rel. Index: Relativní index v rámci bloku. **Tučně** jsou označeny implementované parametry a *kurzívou* pak neimplementované. Značka * vedle čísla znamená, že příslušný parametr je zahrnut do struktury View object konkrétního bloku.

Typ objektu: **Smp**. – Jednoduchá proměnná, **Rec**. – Struktura

Datové typy: **U8**, **U16**, **U32**, **I8**, **Float**, **Vstr**. , **Ostr**. , **DS-n** – Datová struktura (viz [3])

Umístění: **N** – (Non-volatile) proměnná, která musí být pamatována i při výpadku napájení. **S** – Statická proměnná. Jako N ale navíc inkrementuje čítač statických revizí ST_REV. **D** – Dynamická proměnná. Hodnota je vypočítávána či čtena z jiného bloku. **Cst** –Konstanta. Parametr se v zařízení nemění.

Acc: **r** – parametr pro čtení, **w** – parametr pro zápis

Užití parametru: **C** – Obsažený, **I** – Vstupní, **O** – Výstupní

Mandatory/Optional: **M** – Povinný parametr. **O** – Volitelný parametr. **U** – Uživatelský parametr specifikovaný výrobcem.

IV. Protokol komunikace s TG_v3.01

C	Příkaz	Data	Odpověď	Popis
0	***Power ON***			Zapni napájení
1	***Power OFF***			Vypni napájení
2	***Movement to left***			Pohyb doleva
3	***Movement to right***			Pohyb doprava
4	***Stop***			Stop
5	***Go to target position***			Jdi na zadanou pozici
6				
7				
8	***Initialise current position***	signed long		Inicializace pozice
9	***Target position (absolute)***	signed long		Cílová pozice
10	***Target position (relative)***	signed int		Cílová pozice zadaná relativně vzhledem k poslední cílové pozici
11	***Tell position (absolute)***		signed long	Aktuální pozice
12	***Tell position (relative)***		signed int (0x0FFFF–out of range)	Aktuální pozice učená relativně vzhledem k poslednímu čtení
13	***Speed***	unsigned long		Rychlost pohybu
14	***Acceleration***	unsigned long		Zrychlení při rozjezdu a zastavování
15	***Emergency Deceleration***	unsigned long		Zastavování při příkazu ***Stop***
16	***Proportional constant***	unsigned int		Parametr servosmyčky – proporcionální konstanta
17	***Differential constant***	unsigned int		Parametr servosmyčky – diferenční konstanta
18	***Sum constant***	unsigned int		Parametr servosmyčky – sumační konstanta
19				
20	***Logical Outputs***	unsigned char		Přečte hodnotu logických výstupů
21	***Set logical bit***	unsigned char (1-bit0, 2-bit1, …)		Nastaví požadovaný výstupní bit do hodnoty log. 1
22	***Clear logical bit***	unsigned char (1-bit0, 2-bit1, …)		Nastaví požadovaný výstupní bit do hodnoty log. 0
23	***Logical Inputs***		unsigned char	Přečte hodnotu logických vstupů
24				
25				
26	***Cart identification***		char [4]	Přečte identifikační řetězec zařízení
27	***Status***		unsigned char	0x00=vypnutý, 0x01zapnutý, 0x03=pohyb, 0x10=chyba
28	***Error flag***		unsigned char	Chyba

***Tučně** označená čísla příkazů jsou podporovány aktuální verzí*

V. GSD soubor

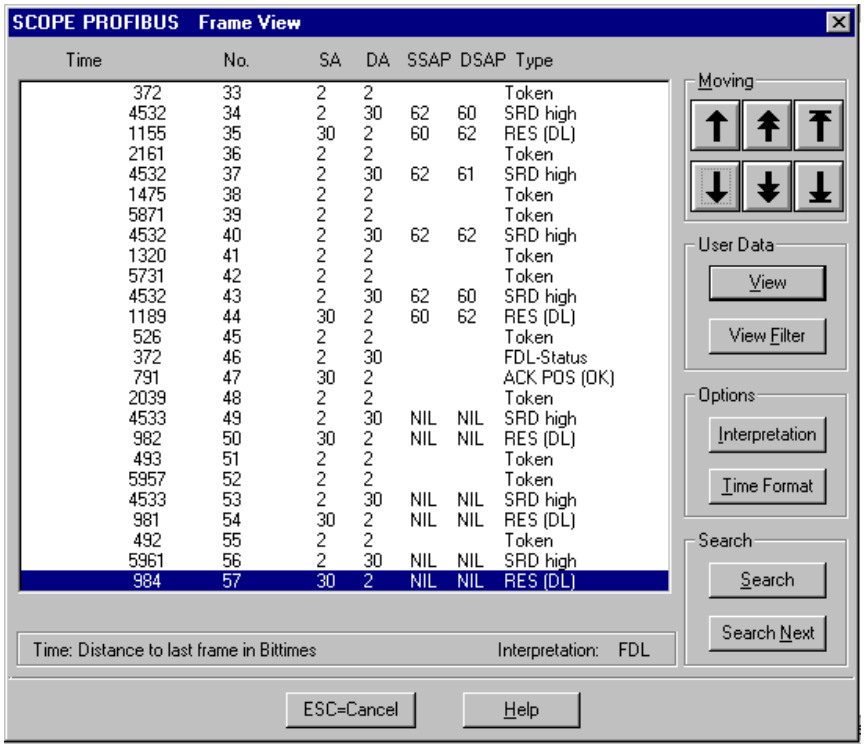
```
;==========================================================================
; PA-Slave      : PDB
; Zařízeni      : Rozhraní Profibus DP/PA - TG_v3.01
; Datum         : 10.10.2002
;==========================================================================
;
#Profibus_DP
GSD_Revision        = 2                                 ; DP/V1
;
;- Nastavení ---------------------------------------------------------------
;
Vendor_Name         = "Premysl Sucha"
Revision            = "3.0"
Model_Name          = "PA/RS232"
Ident_Number        = 0x9808
Protocol_Ident      = 0                                 ; PROFIBUS-DP
Station_Type        = 0                                 ; DP-Slave
Implementation_Type = "DPC31"
Hardware_Release    = "HW 1.00"
Software_Release    = "SW 1.00"
Slave_Family        = 3@TdF@K335                        ; PA-Profile
;
;  bitmap: *.DIB formát 70x40 pixel, 16 barev
;
;Bitmap_Device       = ""
;Bitmap_Diag         = ""
;Bitmap_SF           = ""
;
;
;- Přenosová rychlost -------------------------------------------------------
;
9.6_supp            = 1
19.2_supp           = 1
31.25_supp          = 1
45.45_supp          = 1
93.75_supp          = 1
187.5_supp          = 1
500_supp            = 1
1.5M_supp           = 1
3M_supp             = 1
6M_supp             = 1
12M_supp            = 0
;
; Max. časové odezvy
;
MaxTsdr_9.6         = 60
MaxTsdr_19.2        = 60
MaxTsdr_31.25       = 100
MaxTsdr_45.45       = 200
MaxTsdr_93.75       = 1000
MaxTsdr_187.5       = 1000
MaxTsdr_500         = 1000
MaxTsdr_1.5M        = 1000
MaxTsdr_3M          = 10000
MaxTsdr_6M          = 10000
```

```
;
;*************************************************************************
;**        Uživatelská parametrizační data                              **
;*************************************************************************
User_Prm_Data_Len = 3
User_Prm_Data     = 0, 0, 0

;- DP-specifikace -------------------------------------------------------
;
Freeze_Mode_supp    = 0
Sync_Mode_supp      = 0
Set_Slave_Add_supp  = 0
Fail_Safe           = 1
Min_Slave_Intervall = 200
Max_Diag_Data_len   = 6
;
;- Hardwarová specifikace ------------------------------------------------
;
Redundancy          = 0
Repeater_Ctrl_Sig   = 0
24V_Pins            = 0
;Physical_Layer      = 1                              ; IEC 1158-2
;
;
;- Moduly ----------------------------------------------------------------
;
Modular_Station     = 1                              ; Modulární Slave
Max_Module          = 1
Max_Input_Len       = 6
Max_Output_Len      = 11
Max_Data_Len        = 17
;
;
; Module SP(vstup) + READ_BACK(výstup)
Module  = "SP + RB" 0x94,0xA4
Endmodule
;
; Module SP,IO_OUT(vstup) + READ_BACK,IO_IN(výstup)
Module  = "SP,IO + RB,IO" 0x95,0xA5
Endmodule
;
; Module SP,IO_OUT(vstup) + READ_BACK,IO_IN,POS_D,CHECK_BACK(výstup)
Module  = "SP,IO + RB,IO,PD,CB" 0x95,0xA5,0xA1,0xA2
Endmodule

;
;- EOF -------------------------------------------------------------------
;
```

VI. Inicializace zařízení PDB

č.	SA	DA	SSAP	DSAP	Typ	FC	SD	Data
33	2	2	-	-	Token	-	0xDC	
34	2	30	62	60	SRD high	0x6D	0x68	0x02,0x05,0x00,0xFF,0x98,0x08
35	30	2	60	62	RES (DL)	0x08	0x68	
36	2	2	-	-	Token	-	0xDC	0x88,0x08,0x09,0x16,0x98,0x08,0x0 0,0x00,0x00,0x00
37	2	30	62	61	SRD high	0x5D	0x68	
38	2	2	-	-	Token	-	0xDC	
39	2	2	-	-	Token	-	0xDC	0x14,0x24
40	2	30	62	62	SRD high	0x7D	0x68	
41	2	2	-	-	Token	-	0xDC	
42	2	2	-	-	Token	-	0xDC	
43	2	30	62	60	SRD high	0x5D	0x68	0x00,0x0C,0x00,0x02,0x98,0x08
44	30	2	60	62	RES (DL)	0x08	0x68	
45	2	2	-	-	Token	-	0xDC	
46	2	30	-	-	FDL-Status	0x49	0x10	
47	30	2	-	-	ACK POS(OK)	0x00	0x10	
48	2	2	-	-	Token	-	0xDC	
49	2	30	NIL	NIL	SRD high	0x7D	0x10	
50	30	2	NIL	NIL	RES (DL)	0x08	0x68	0x00,0x00,0x00,0x01
51	2	2	-	-	Token	-	0xDC	
52	2	2	-	-	Token	-	0xDC	
53	2	30	NIL	NIL	SRD high	0x7D	0x10	
54	30	2	NIL	NIL	RES (DL)	0x08	0x68	0x00,0x00,0x00,0x01

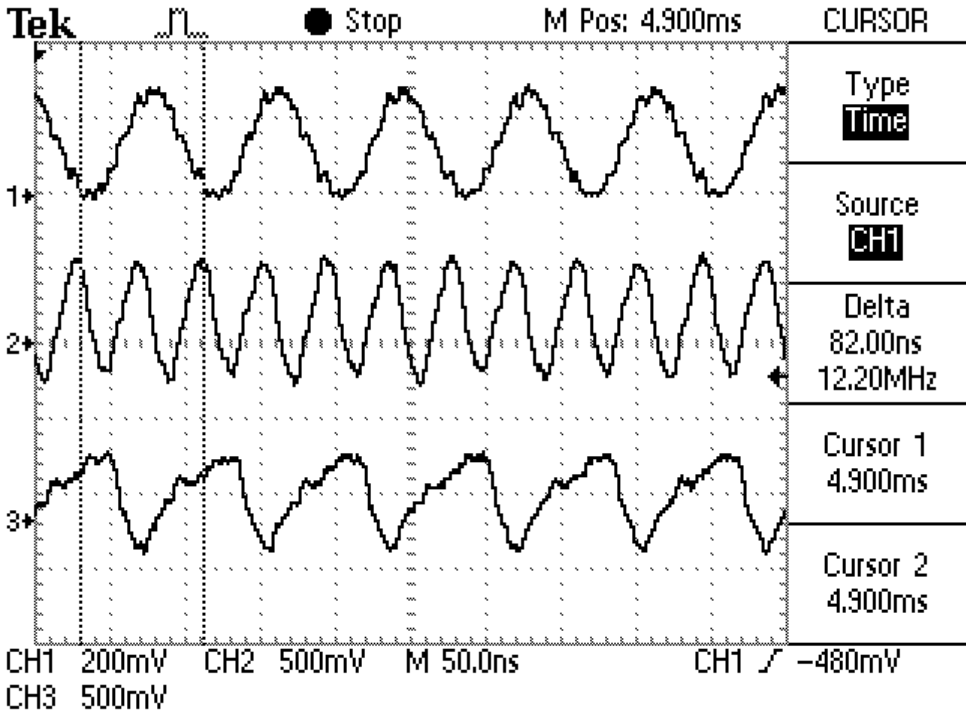
Při měření byl nastaven filtr tak, aby byla registrována pouze zařízení Master (adr. 2) a PDB (adr. 30)



Obr. VI-1 Program SCOPE PROFIBUS

VII. Měření

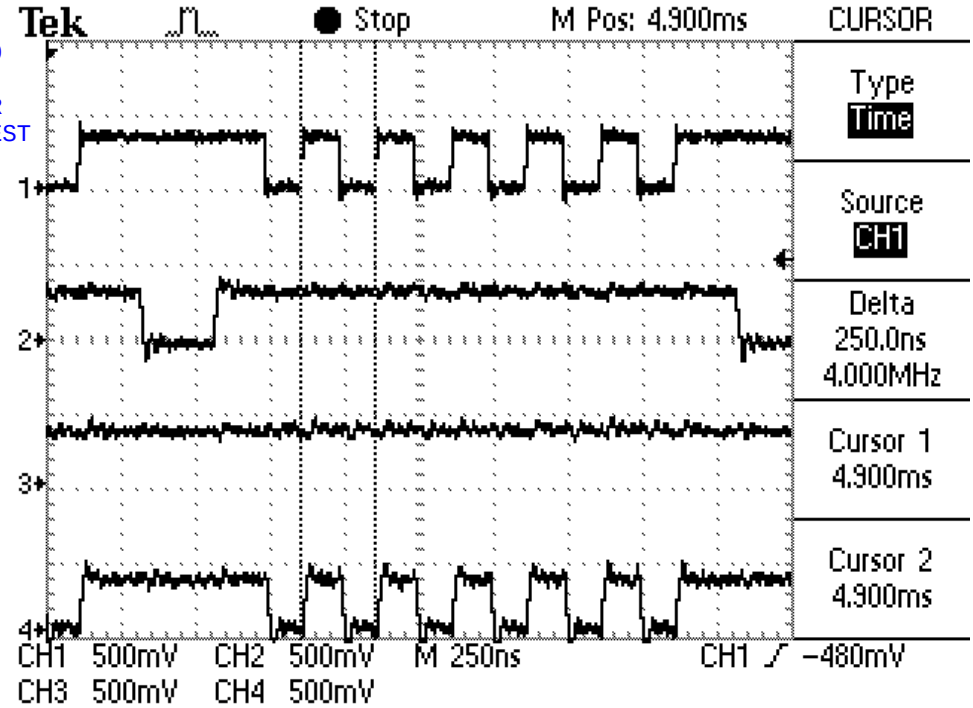
Oscilátor DPC31



Obr. VII-1 Měření generátoru hodinového signálu obvodu DPC31. PDB v režimu Profibus DP (krystal 12MHz, zapnuta jednotka fázového závěsu PLL). 1-XTAL_CLK, 2-CLKOUT1x2, 3- CLKOUT1x4

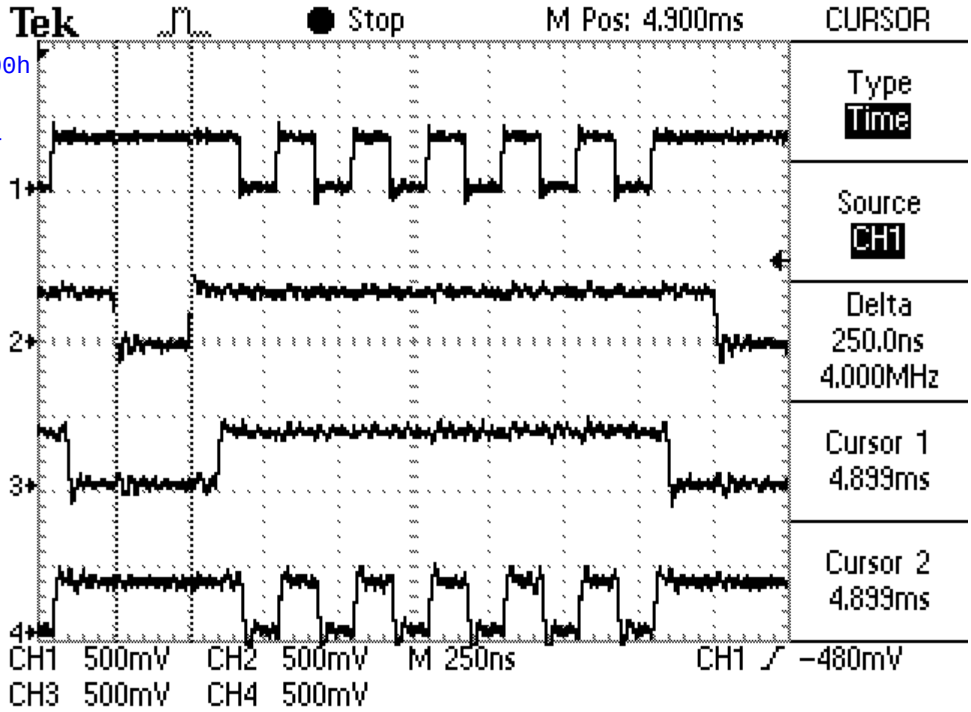
Sběrnice PDB

```
…
        mov     DPTR,#0
LOOP_TEST:
        movx    a,@DPTR
        jmp     LOOP_TEST
…
```



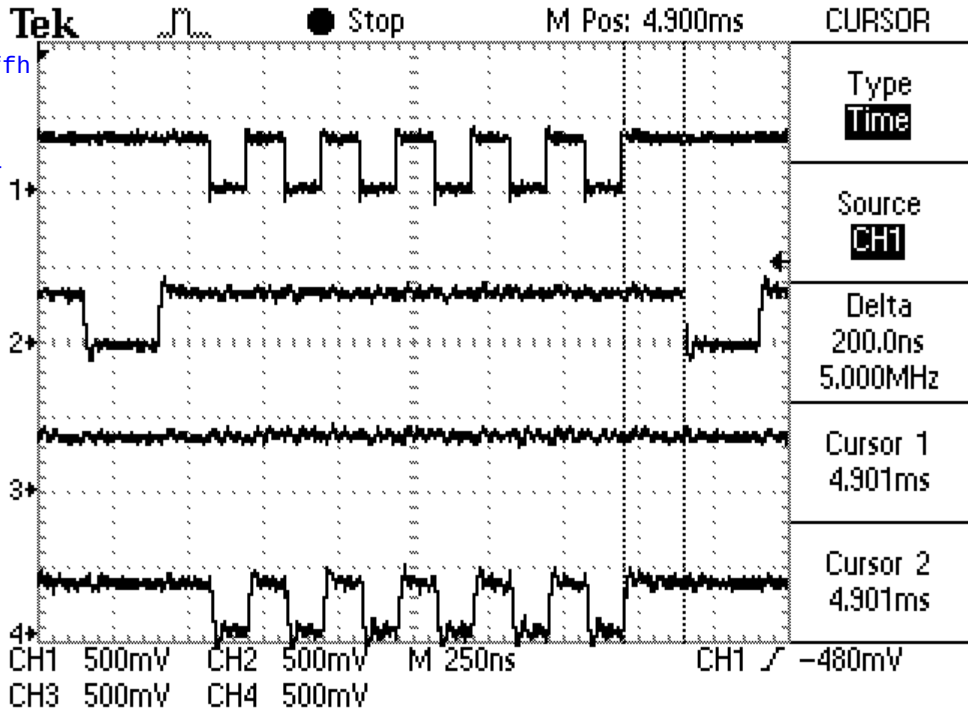
Obr. VII-2 Měření signálů na sběrnici. Čtení z externí paměti dat v DPC31. PDB v režimu Profibus DP (krystal 12MHz, zapnuta jednotka fázového závěsu PLL). 1-PSEN, 2-RD, 3-XCSDATA, 4-XCSCODE

```
…
        mov     DPTR,#2000h
LOOP_TEST:
        movx    a,@DPTR
        jmp     LOOP_TEST
…
```



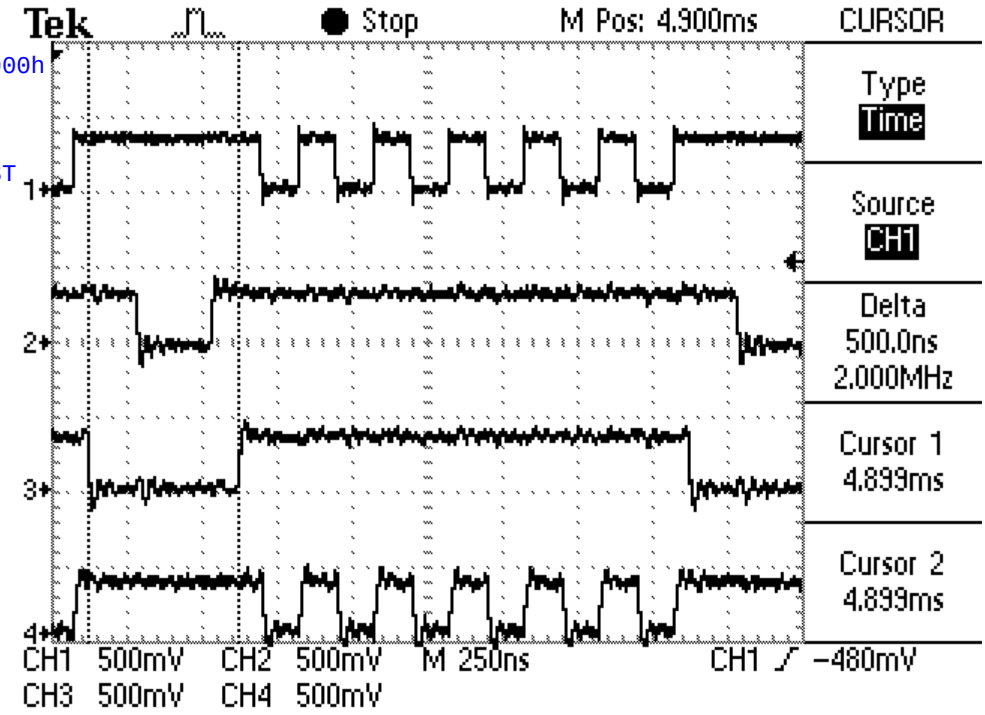
Obr. VII-3 Měření signálů na sběrnici. Čtení z externí paměti dat v RAM. PDB v režimu Profibus DP (krystal 12MHz, zapnuta jednotka fázového závěsu PLL). 1-PSEN, 2-RD, 3-XCSDATA, 4-XCSCODE

```
…
        mov     DPTR,#1fffh
        mov     a,#0ffh
LOOP_TEST:
        movx    @DPTR,a
        jmp     LOOP_TEST
…
```



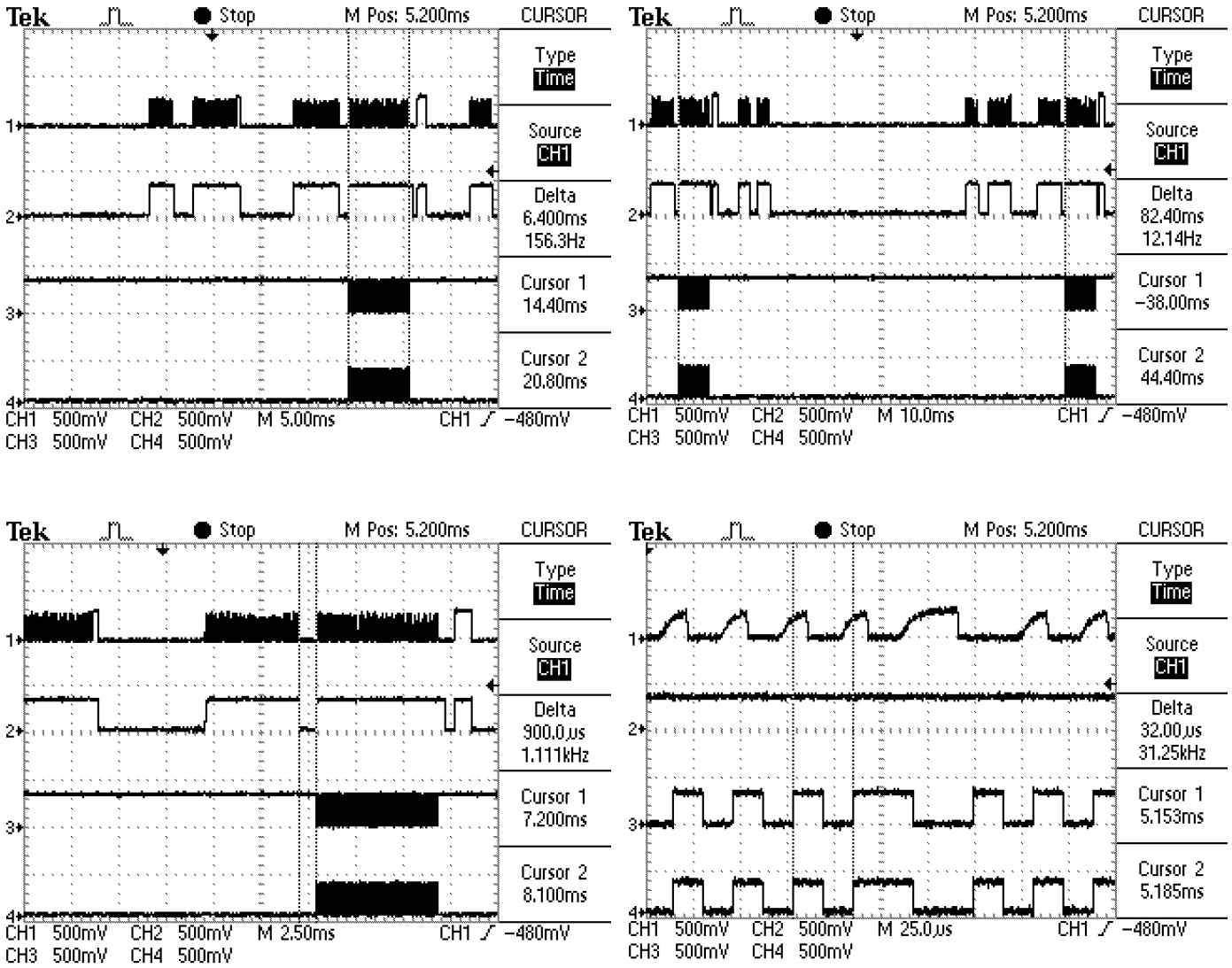
Obr. VII-4 Měření signálů na sběrnici. Zápis do externí paměti dat v DPC31. PDB v režimu Profibus DP (krystal 12MHz, zapnuta jednotka fázového závěsu PLL). 1-PSEN, 2-WR, 3-XCSDATA, 4-XCSCODE

```
…
        mov     DPTR,#2000h
        mov     a,#0ffh
LOOP_TEST:
        movx    @DPTR,a
        jmp     LOOP_TEST
…
```



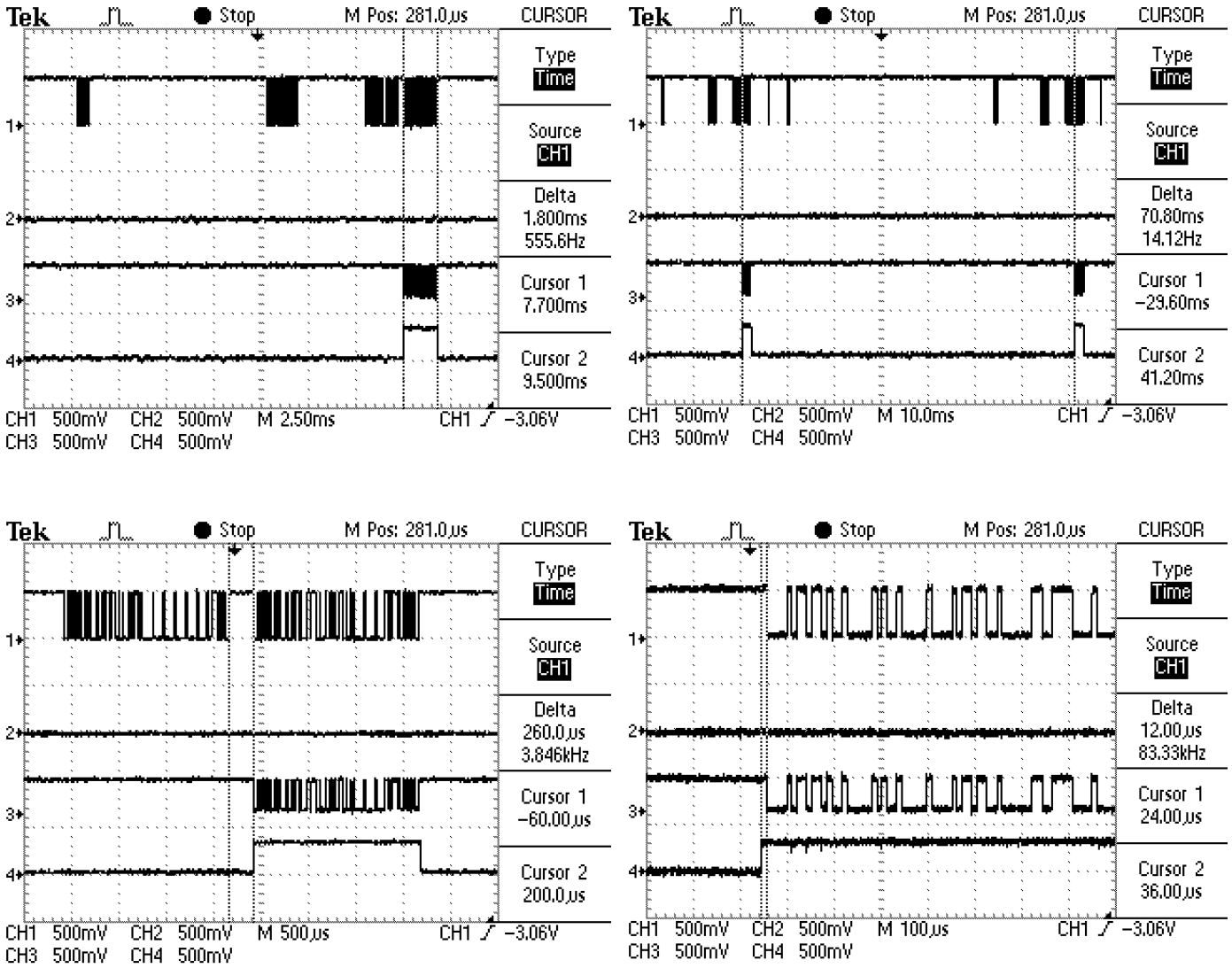
Obr. VII-5 Měření signálů na sběrnici. Zápis do externí paměti dat v RAM. PDB v režimu Profibus DP (krystal 12MHz, zapnuta jednotka fázového závěsu PLL). 1-PSEN, 2-WR, 3-XCSDATA, 4-XCSCODE

Profibus PA



Obr. VII-6 Měření signálů na sběrnici Profibus PA (konfigurace podle obr. 7-4). PDB v režimu Profibus PA (krystal 4MHz, vypnuta jednotka fázového závěsu PLL). 1-RxD_RxS, 2-XCTS_RxA, 3-TxD_TxS, 4-RTS_TxE

Profibus DP



Obr. VII-7 Měření signálů na sběrnici Profibus DP (93,75kbit/s, konfigurace podle obr. 7-4). PDB v režimu Profibus DP (krystal 12MHz, zapnuta jednotka fázového závěsu PLL). 1-RxD_RxS, 2-XCTS_RxA, 3-TxD_TxS, 4-RTS_TxE

Měření byla provedena osciloskopem Tektronix TDS224 4 kanály (100MHz, 1GS/s)

OB1 - <offline>

"Cycle Execution"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 20.12.2002 08:59:12dop.
Interface: 15.02.1996 16:51:12odp.
Lengths (block/logic/data): 00368 00240 00032

Address	Declaration	Name	Type	Initial value	Comment
0.0	temp	OB1_EV_CLASS	BYTE		Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
1.0	temp	OB1_SCAN_1	BYTE		1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
2.0	temp	OB1_PRIORITY	BYTE		1 (Priority of 1 is lowest)
3.0	temp	OB1_OB_NUMBR	BYTE		1 (Organization block 1, OB1)
4.0	temp	OB1_RESERVED_1	BYTE		Reserved for system
5.0	temp	OB1_RESERVED_2	BYTE		Reserved for system
6.0	temp	OB1_PREV_CYCLE	INT		Cycle time of previous OB1 scan (milliseconds)
8.0	temp	OB1_MIN_CYCLE	INT		Minimum cycle time of OB1 (milliseconds)
10.0	temp	OB1_MAX_CYCLE	INT		Maximum cycle time of OB1 (milliseconds)
12.0	temp	OB1_DATE_TIME	DATE_AND_TIME		Date and time OB1 started

Block: OB1 "Main Program Sweep (Cycle)"

Komunikace s Profibus Development Board

Network: 1

Cteni konzistentnich dat z rozhrani Profibusu

```
L     270
T     MW   100
CALL  "DPRD_DAT"            SFC14              -- Read Consistent Data of a Standard DP Slave
 LADDR  :=MW100
 RET_VAL:=MW102
 RECORD :=P#M 104.0 BYTE 6
```

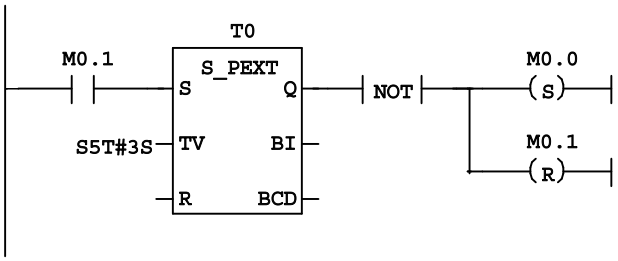
Network: 2

Zapis konzistentnich dat do rozhrani Profibusu

```
L     280
T     MW   130
CALL  "DPWR_DAT"            SFC15              -- Write Consistent Data to a Standard DP Slave
 LADDR  :=MW130
 RECORD :=P#M 134.0 BYTE 6
 RET_VAL:=MW132
```

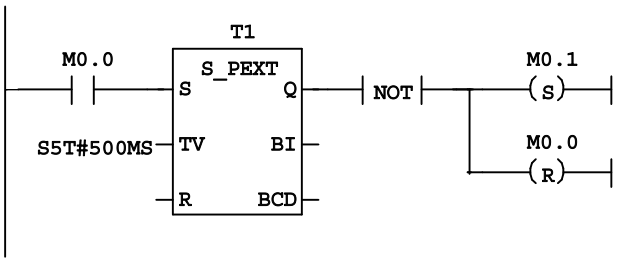
Network: 3

Timer 0



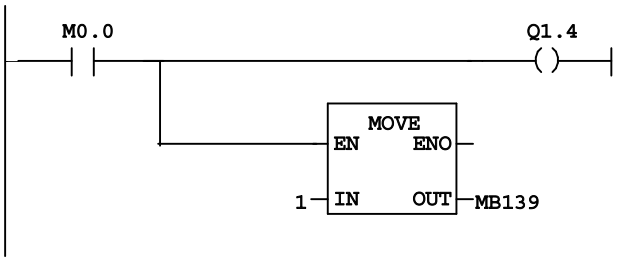
Network: 4

Timer 1



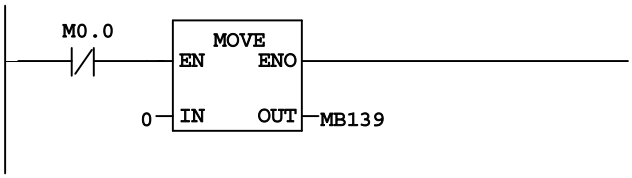
Network: 5

Ligicky vystup na PLC (Q1.4) do log. 1 a logicky vystup na servopohonu take do log. 1



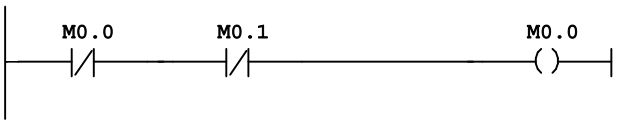
Network: 6

Logicky vystup na servopohonu do log. 0

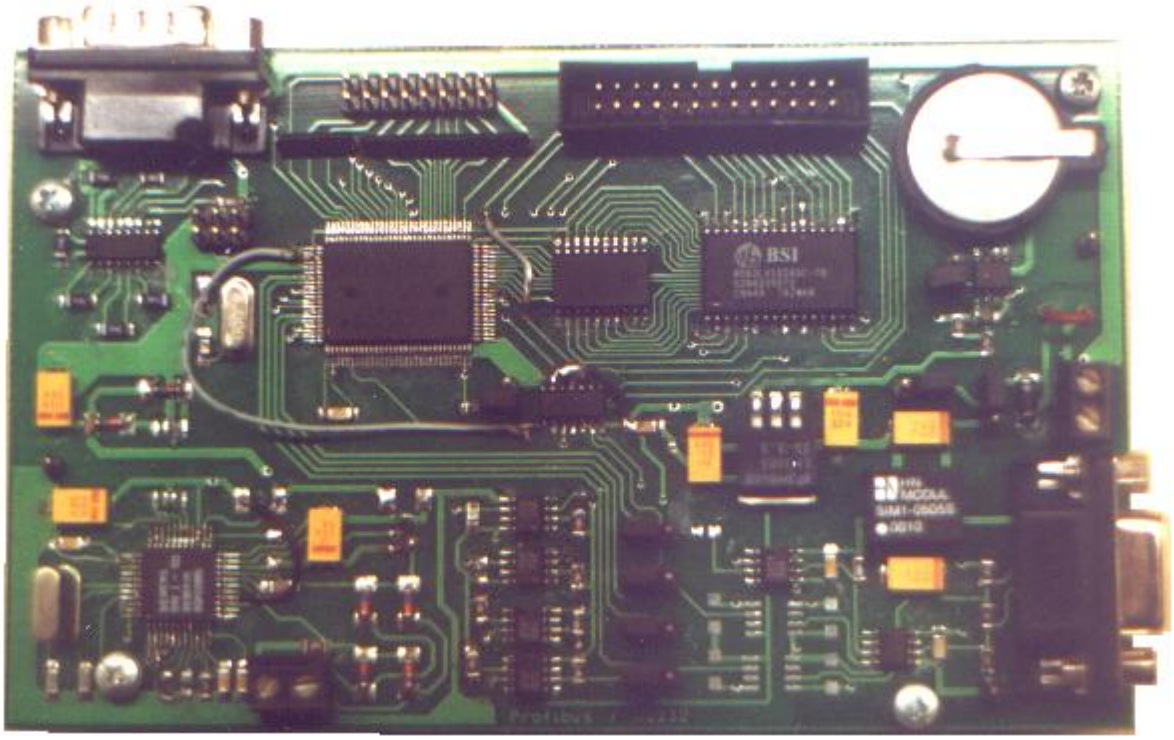


Network: 7

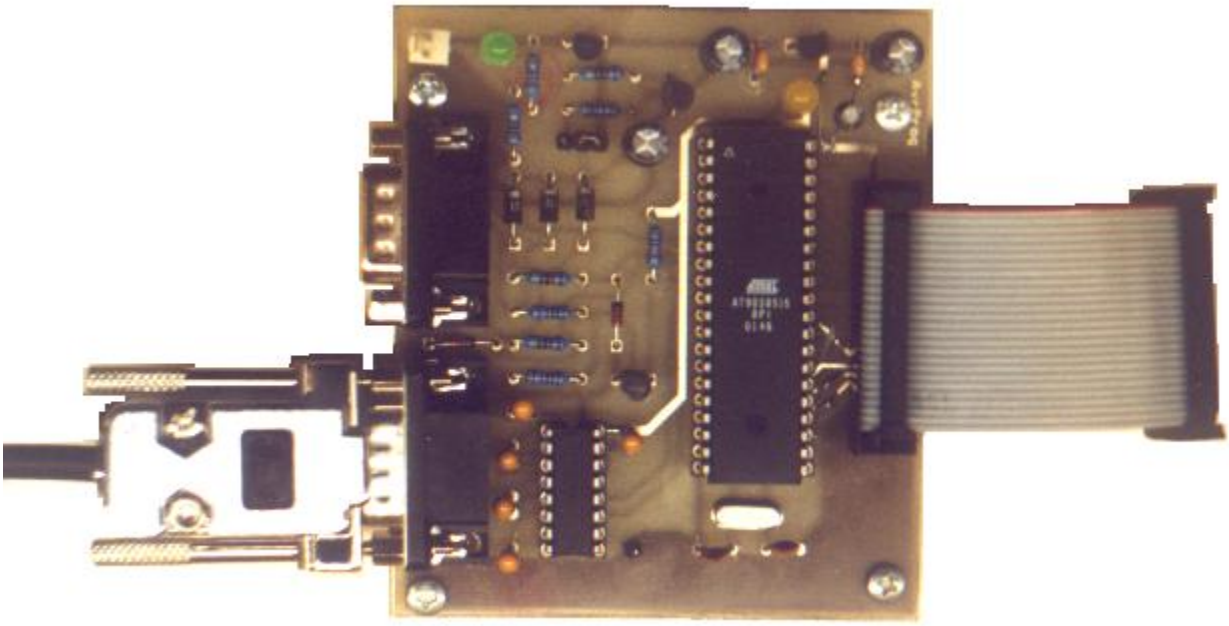
Inicializace



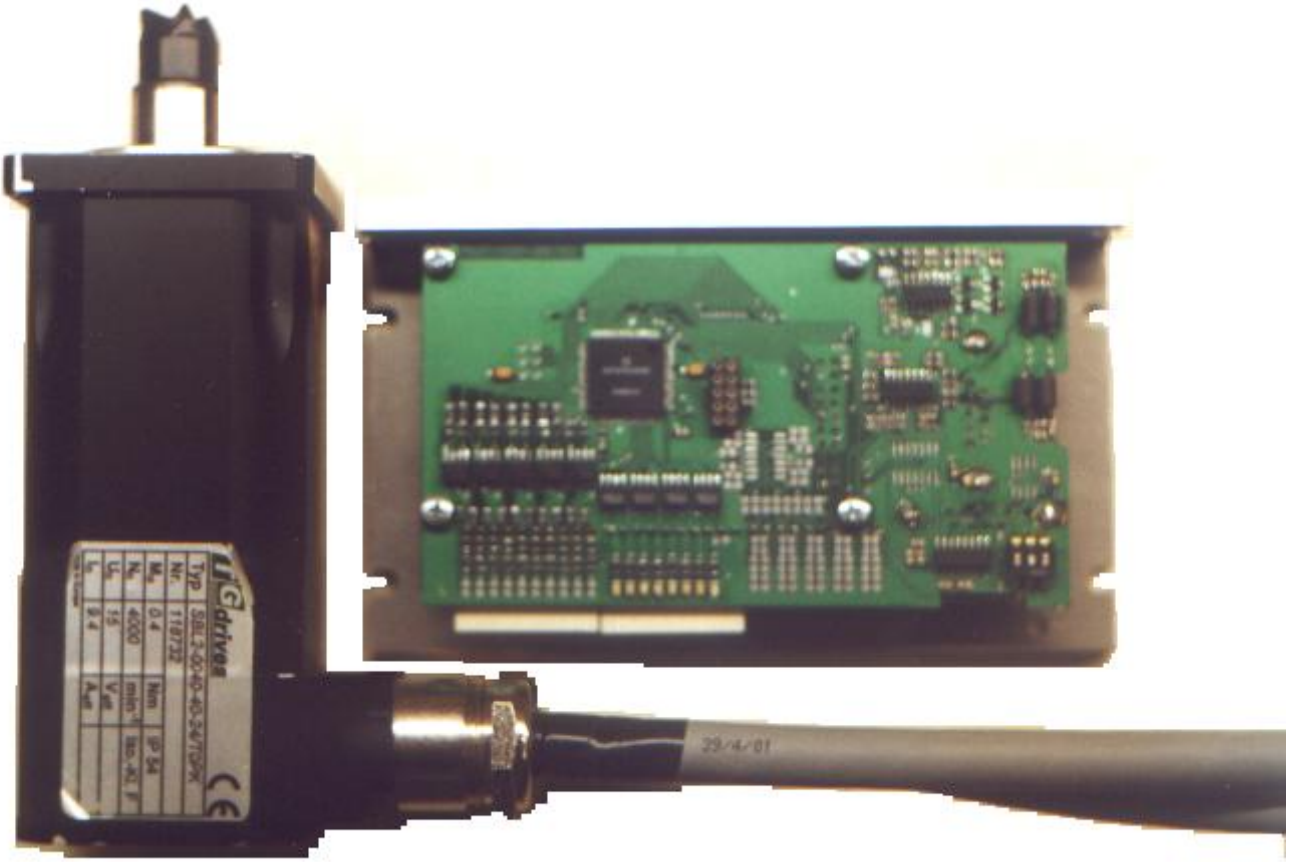
IX. Fotografie



Obr. IX-1 PDB (Profibus Development Board)



Obr. IX-2 ProfiProg (programátor desky PDB)



Obr. IX-3 Kontrolér TG_v3.01se stejnosměrným bezkartáčovým motorem (zapůjčeno firmou Ing. Miloslav Žižka)