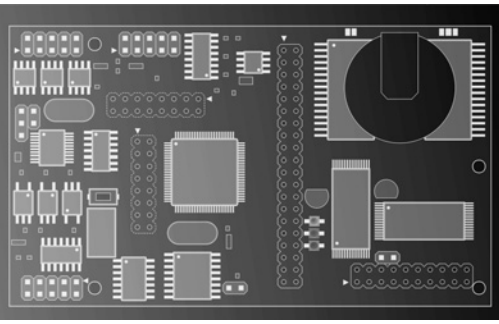
ELSACO, Jaselská 177
28000 KOLÍN, CZ
tel/fax +420-321-727753
http://www.elsaco.cz
mail: elsaco@elsaco.cz



Stavebnice PROMOS Line 2

Komunikační protokoly jednotek PL2

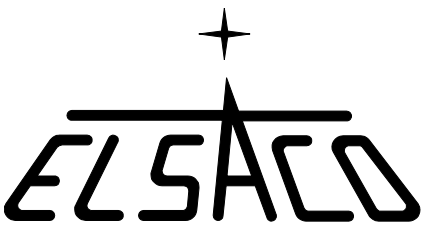
CANopen ***Epsnet***

SAM ***ModBus***

Technický manuál

revize 2
21. 07. 2008

verze 1.02



© 2005 sdružení ELSACO

Účelová publikace ELSACO

ELSACO, Jaselská 177, 280 02 Kolín 3
Tel./fax/modem: 321 727 753 / 321 727 759
Internet: **www.elsaco.cz**

Připomínky: vondruska@elsaco.cz

OBSAH

1. Architektura sběrnice CAN 5
1.1. Obecná charakteristika 5
1.1.1. Základní pojmy 5
1.1.2. Popis funkce 6
1.2. Fyzická vrstva 6
1.2.1. Kódování bitového proudu 6
1.2.2. Časování bitů a synchronizace 6
1.2.3. Přenosové médium 7
1.2.4. Topologie sběrnice 7
1.3. Linková vrstva 8
1.3.1. Rámce protokolu CAN 2.0A 8
1.3.2. Arbitráž vícenásobného přístupu 8
1.3.3. Kontrola a obsluha chyb 9
1.4. Aplikační vrstva 10
1.4.1. Základní vlastnosti implementace 10
1.4.2. Komunikační objekty 10
1.4.3. Struktura slovníku 10
1.4.4. Objekty globálního řízení 10
1.4.5. Datová komunikace 11
1.5. PDO protokol 11
1.5.1. Komunikační parametry a napojení PDO 12
1.5.2. Mapování aplikačních objektů 12
1.5.3. Propojení aplikačních objektů a proměnných . 13
1.5.4. Shrnutí 13
1.6. SDO protokol 13

2. Komunikační protokol Epsnet 15
2.1. Struktura protokolu a typy zpráv 15
2.1.1. Formáty telegramů 15
2.1.2. Struktura protokolu 15
2.2. Komunikační služby a přenos dat 16
2.2.1. Přehled služeb 16
2.2.2. Komunikační služby 16
2.3. Organizace veřejných dat PL2 18
2.3.1. Jednotky s verzí firmware 2.00 a vyšší 18
2.3.2. Jednotky bez označení verze firmware 18

3. Object Dictionary – popis 19
3.1. SDO – Service Data Objects 19
3.1.1. Komunikační objekty 19
3.1.2. Objekty definované výrobcem 19
3.1.3. Objekty podle profilu zařízení 19
3.2. Popis SDO stavebnice PL2 19
3.2.1. Standardizované komunikační objekty 19
3.2.2. Komunikační objekty dané výrobcem 20
3.2.3. Konfigurační objekty výrobce 20
3.2.4. Procesní objekty výrobce 21
3.2.5. Objekty profilu zařízení 21
3.3. PDO – Proces Data Objects 21
3.3.1. Výchozí nastavení PDO 21
3.3.2. Mapování PDO 22

4. Object dictionary – seznam objektů 27

5. Komunikační protokol ModBus 35
5.1. Struktura protokolu a typy zpráv 35
5.1.1. Formát telegramů 35
5.1.2. Typy zpráv 35
5.1.3. Komunikace pomocí Object Dictionary 36
5.1.4. Zabezpečení telegramů 37
5.2. Důležité upozornění 37

6. Komunikační protokol SAM 39
6.1. Základní charakteristika 39
6.2. Konfigurační režim 39
6.3. Obecné a konfigurační příkazy 39
6.3.1. Nastavení komunikačních parametrů 39
6.3.2. Dotaz na nastavení 40
6.3.3. Nastavení soft WatchDog Timer 40
6.3.4. Čtení EEPROM 40
6.3.5. Restart z EEPROM bloku po SWDT 40
6.3.6. Verze firmware 40
6.3.7. Jméno jednotky 40
6.3.8. Reset jednotky 40
6.3.9. Restart – inicializace z EEPROM/FLASH 40
6.3.10. Pauza 40
6.4. Obsluha binárních I/O linek 40
6.4.1. Čtení osazené konfigurace FCPU-02A 40
6.4.2. Volba režimu jednoho čítače 40
6.4.3. Volba režimu jednoho čítače u SAM-02 41
6.4.4. Volba režimu všech čítačů 41
6.4.5. Čtení okamžité hodnoty vstupů 41
6.4.6. Čtení okamžité hodnoty vstupů SAM-02 41
6.4.7. Čtení filtrované hodnoty vstupů 41
6.4.8. Čtení filtrované hodnoty vstupů SAM-02 41
6.4.9. Čtení stavu vstupů (displeje a bargrafy) 41
6.4.10. Čtení stavu čítačů 41
6.4.11. Nastavení úrovně prahu 42
6.4.12. Kombinované čtení stavu vstupů 42
6.4.13. Dotaz na periodu signálu 42
6.4.14. Nastavení stavu výstupů 42
6.4.15. Nastavení stavu výstupů u SAM-02 42
6.4.16. Vyslání hodnoty 42
6.5. Obsluha analogových I/O linek 42
6.5.1. Nastavení rozsahů analogového vstupu 42
6.5.2. Čtení analogové hodnoty 42
6.5.3. Nastavení analogových výstupů 42
6.5.4. Výběr typu domečku 42
6.5.5. Nastavení koeficientů polynomu 43
6.5.6. Aktivní odeslání hodnoty na displej 43
6.6. Obsluha číslicových zobrazovačů 43
6.6.1. Nastavení počtu číslic displeje 43
6.6.2. Zobrazení údaje na displeji 43
6.6.3. Nastavení jasu 43
6.6.4. Přečtení vstupů (XDM14A) 43
6.6.5. Režim čítače (upravený XDM25) 43
6.7. Obsluha sloupcových indikátorů 43
6.7.1. Zobrazení sloupce 43
6.7.2. Zobrazení sloupce binární mapou 43
6.7.3. Volba počtu sloupců 43
6.7.4. Přečtení vstupů 44
6.8. Komunikace dvou FCPU-02A 44

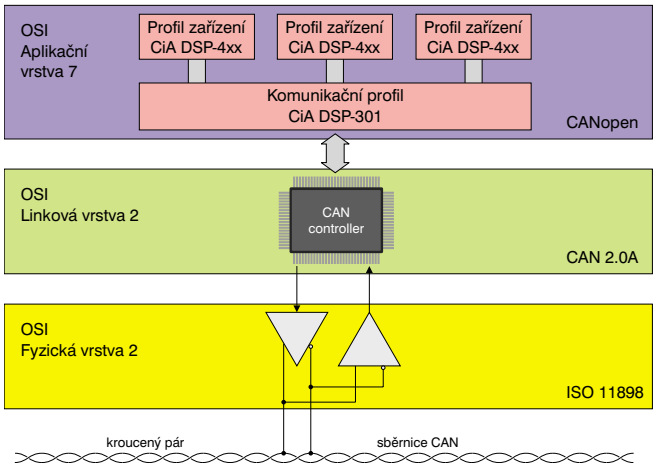
1. ARCHITEKTURA SBĚRNICE CAN

1.1. Obecná charakteristika

CAN (Controller Area Network) byl původně vyvinut pro automobilový průmysl s cílem snížit počet spojovacích vodičů mezi jednotlivými elektronickými uzly. Od začátku byl koncipován jako vysokorychlostní sériová sběrnice s multimasterovým přístupem a vysokým stupněm zabezpečení. Protože se princip sběrnice ukázal jako velmi efektivní, proniká postupně i do dalších průmyslových odvětví. K základním výhodám CAN patří:

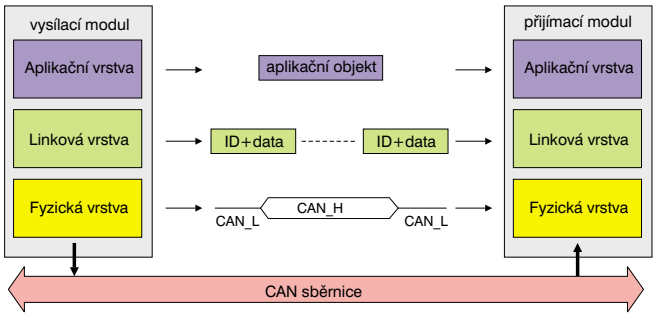
- velmi vysoká bezpečnost i v náročném prostředí
- objektově orientovaný přenos s multimasterovým přístupem
- vyloučení kolizí na sběrnici
- vysoká přenosová rychlost (10 kbit/s až 1Mbit/s)
- krátká doba odezvy pro vysoce prioritní zprávy

Vzhledem k rozšíření sběrnice podporuje CAN většina výrobců mikroprocesorů. Vestavěné řadiče zajišťují většinu základních přenosových funkcí od sestavení komunikačního objektu a přístup na sběrnici až po filtraci a kontrolu příchozích zpráv. To umožňuje velmi efektivní a spolehlivou implementaci s nízkými náklady.



Obr. 1. Struktura sběrnice CANopen dle modelu OSI

Strukturu sběrnice CAN podle modelu OSI ukazuje obr. 1. Přenosová média a fyzickou vrstvu definuje norma ISO 11898. Pro systém PROMOS line2 se jako médium používá kroucený pár. Linková vrstva podle standardu CAN 2.0A je implementována hardwarovými řadiči. Programové vybavení centrálních a periferních jednotek nad použitými hardwarovými řadiči implementuje aplikační vrstvu 7 podle standardu CANopen. Komunikační objekty (procesní, služební, synchronizační, zabezpečovací) definuje norma CiA DS301, profily zařízení jsou definovány normami CiA DS401a dalšími.



Obr. 2. Interakce mezi vrstvami

1.1.1. Základní pojmy

Podrobný popis jednotlivých vrstev je uveden v dalších kapitolách. Zde je na velmi zběžné vysvětlení základních pojmů, s jejichž pomocí je možné získat představu o základních principech CANopen komunikace.

Slovník objektů

Základním prvkem výměny dat je slovník objektů (Object Dictionary – OD). Slovník je uspořádanou skupinou objektů. Každý objekt je adresován 16-bitovým indexem, přístup k jednotlivým prvkům objektu se provádí 8-bitovým subindexem. Slovník obsahuje všechny informace o procesních veličinách a jejich komunikačních parametrech. Každý účastník CANopen komunikace (tedy modul na sběrnici CAN) má ve své paměti „svůj" slovník. V periferních modulech slovník obsahuje informace o signálech zpracovávaných tímto modulem. Centrální jednotka obsahuje slovník, který nese informace o všech procesních veličinách používaných v dané aplikaci.

Komunikační objekty

Pro předávání dat na sběrnici se používají různé typy zpráv. Protože komunikace je orientovaná objektově, nazývají se souhrnně komunikační objekty. Protokoly přenosu dat prostřednictvím sítě CANopen je možné rozdělit na administrativní, služební, procesní a speciální.

Administrativní objekty NMT (Network Management)

Slouží ke správě sítě. Správu zajišťuje hlavní stanice v síti, v systémech PROMOS je to vždy centrální jednotka. Správce sítě řídí inicializaci a konfiguraci jednotlivých uzlů a vykonává dohled nad provozem sítě. Správce také udržuje přehled o stavu jednotlivých stanic (Node guarding) a určuje jejich provozní stav.

Procesní objekty PDO

PDO (Process Data Object) jsou určeny k rychlému předávání procesních dat výhradně v provozním režimu.

Servisní objekty SDO

SDO (Service Data Object) slouží k přístupu ke všem prvkům slovníku, především k těm, které nejsou předávány procesními objekty. Používá se ke konfiguraci uzlů ve výchozím režimu a k předávání časově nekritických dat v provozním režimu.

Speciální funkční objekty

Jsou určeny pro aplikační synchronizaci (SYNC), předávání času (Time Stamp) a krizové stavy (Emergency).

Metody přístupu ke sběrnici

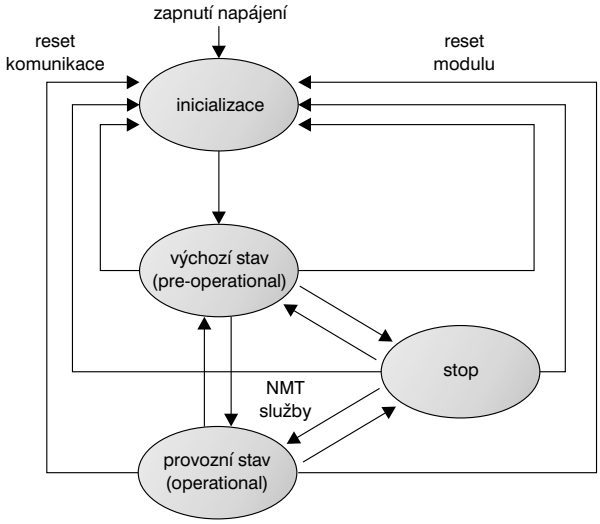
I když CANopen definuje několik modelů přístupu, v systému PROMOS se používají pouze dva – Master/Slave a Client/Server.

Metoda Master/Slave se používá v provozním režimu, kdy centrální jednotka prostřednictvím PDO zajišťuje přenos procesních dat z/do periferních jednotek.

Metoda Client/Server se používá pro přenos bloků dat delších než 8 bytů. Blok je rozdělen (segmentován) a přenos probíhá po částech. V tomto modelu centrální jednotka vystupuje jako Client a periferní jednotka jako Server. Tento model se používá v průběhu konfigurace a částečně v provozním stavu pro přenos delších datových objektů.

Stavový diagram stanice

Stavový diagram stanice uvádí obrázek 3. Po zapnutí napájení provede stanice vlastní inicializaci a automaticky přechází do výchozího stavu. V tomto stavu je dovolena konfigurace stanice protokolem SDO. PDO komunikace není dovolena. Do provozního režimu stanice přechází na povel správce sítě. V provozním režimu probíhá procesní komunikace PDO, případně i komunikace SDO. Režim stop je zvláštní stav, kdy je dovolena



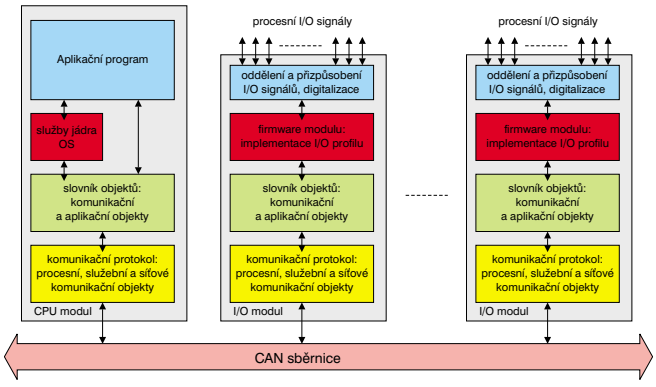
Obr. 3. Stavový diagram stanice CAN

pouze komunikace NMT – převést stanici do jiného stavu může pouze správce sítě. Při resetu modulu nebo resetu komunikace (výpadek nebo služba NMT) stanice přechází do stavu jako po zapnutí napájení a prochází znovu inicializací.

1.1.2. Popis funkce

Typické zapojení systému PROMOS line 2 uvádí obrázek 4. Procesní I/O signály zpracovávají periferní I/O jednotky. Po přizpůsobení, oddělení a digitalizaci jsou signály zpracovány firmwarem procesoru I/O jednotky. Ten zajišťuje základní zpracování signálů – pro logické signály filtraci, vyhodnocení, čítání, generování pulzů ap., pro analogové signály např. filtraci, měřítkování a převod fyzikálních jednotek na procesní veličiny a naopak. Ve skutečnosti tedy provádí převod mezi aplikačními objekty (jinak též proměnné aplikačního programu) a procesními signály.

Aplikační objekty jsou uloženy ve slovníku objektů I/O modulu. Komunikační vrstva modulu zajišťuje mapování aplikačních objektů ze slovníku do příslušných komunikačních objektů tak, aby bylo k dispozici zařízení, které se stará o distribuci dat mezi moduly (typicky CPU).



Obr. 4. Typická architektura systému PROMOS line2

Analogické schéma je i u centrálního modulu (CPU). Aplikační program po startu provede inicializaci a parametrizaci komunikační vrstvy, která se stará o distribuci dat. Komunikační vrstva prostřednictvím komunikačních objektů zajišťuje přenosy aplikačních objektů mezi slovníky I/O modulů a slovníkem CPU. Aplikační program pak pracuje s objekty ze slovníku, které reprezentují jednotlivé procesní veličiny.

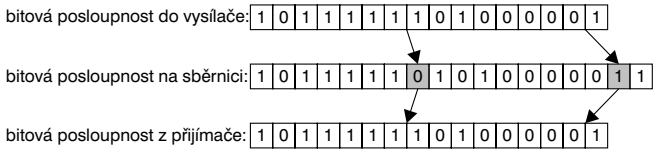
Tok dat mezi aplikačním programem v CPU a I/O moduly je tedy objektově specifikován. Součástí aplikace je komunikační profil, který určuje mapování aplikačních objektů – procesních proměnných – na konkrétní komunikační objekty. Pro aplikační program je potom lhostejné, je-li konkrétní procesní proměnná realizovaná I/O modulem 1 nebo 7, o vlastní přenosy a jejich časové rozložení se stará komunikační vrstva.

1.2. Fyzická vrstva

Tato vrstva definuje způsob a prostředky fyzického přenosu mezi jednotlivými moduly na sběrnici CAN.

1.2.1. Kódování bitového proudu

Komunikace po rozhraní CAN probíhá sériově s bitově orientovaným přenosem. Na rozdíl od běžných asynchronních sériových linek, kde jsou zprávy rozděleny do bytů je na CANu zpráva předávána jako nepřetržitý blok bitů. Přenos je synchronní, všechny stanice v síti jsou navzájem synchronizovány. Kódování bitů je bez návratu k nule (NRZ). Aby mohly být jednotlivé CAN uzly mezi sebou synchronizovány bez použití dalšího signálu, je bitový proud při vysílání modifikován – když za sebou následuje více bitů stejné úrovně, vkládá se jeden bit opačné polarity (obr. 5). Na straně přijímače jsou tyto vložené bity odstraněny. Tento proces se nazývá bit-stuffing. Délka zprávy v bitech tak sice kolísá v závislosti na obsahu, ale v signálu je za všech okolností zaručen dostatečný počet hran pro vzájemnou synchronizaci jednotlivých stanic.



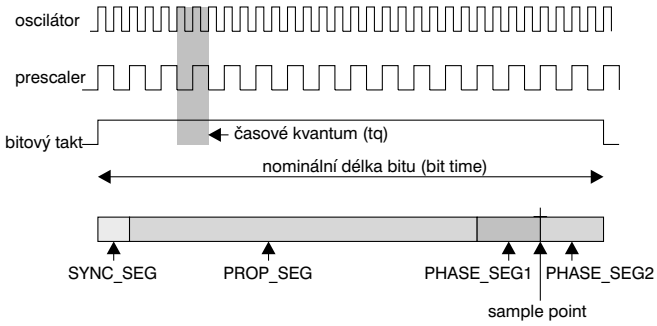
Obr. 5. Vkládání synchronizačních bitů

1.2.2. Časování bitů a synchronizace

Bitová délka je rozdělena na časová kvanta (tq = time quanta). Podle specifikace musí být 1 bit rozdělen na nejméně 8 a nejvíce 25 tq, v systému PROMOS line 2 se obvykle používá dělení na 16 tq, tedy pro 500 kbit/s 1 tq=125 ns. Všechny řadiče CAN umožňují programování struktury bitové délky – viz obr. 6. Bit je rozdělen na následující úseky:

- SYNC_SEG má délku 1 tq a slouží k vyhodnocení synchronizace jednotlivých uzlů
- PROP_SEG má délku 8 tq (obecně 1÷8) a slouží ke kompenzaci dopravního zpoždění signálu
- PHASE_SEG1 má délku 3 tq (obecně 1÷8) a slouží ke kompenzaci fázových chyb, může být protažen při resynchronizaci
- PHASE_SEG2 má délku 2 tq (obecně 2÷PHASE_ _SEG1) a slouží ke kompenzaci fázových chyb, může být zkrácen při resynchronizaci.

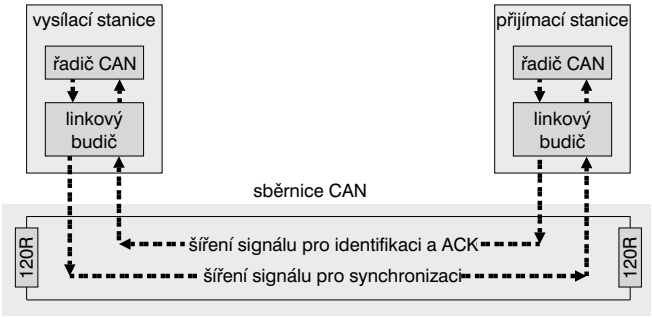
Každá stanice připojená ke CAN sběrnici má svůj řídicí krystalový oscilátor, který určuje bitovou délku. Protože jednotlivé oscilátory nemají absolutně stejný kmitočet, dochází za provozu k fázovému posunu. Pro odstranění tohoto jevu se používá mechanismus resynchronizace. Každá stanice očekává start bit zprávy ve svém segmentu SYNC_SEG. Pokud hrana start bitu ze sběrnice přijde později, stanice si podle potřeby prodlouží PHASE_SEG1 o 1 nebo více tq. Naopak přijde-li hrana



Obr. 6. Rozdělení bitové délky a vzorkovací bod

dříve, je analogicky zkrácen PHASE_SEG2. Tím jsou za provozu řadiče všech připojených stanic neustále udržovány ve vzájemném synchronismu. To je nezbytné pro zajištění arbitráže sběrnice při vícenásobném přístupu.

Do doby šíření signálu se musí započítat cesta signálu tam i zpět od nejvzdálenějších stanic (viz obr. 7). Tato doba musí být menší než je součet PROP_SEG a PHASE_SEG1. Do celkové doby musí být započteno i zpoždění linkových budičů a případného galvanického oddělovače.



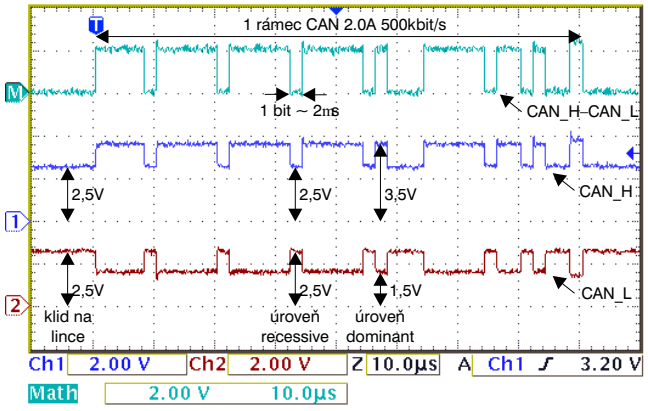
Obr. 7. Princip šíření signálu po sběrnici CAN

Umístění vzorkovacího bodu je vždy kompromisem mezi dovoleným zpožděním signálu a požadavkem na jeho kvalitu. Umístění vzorkovacího bodu dále od začátku bitu umožňuje zvýšit dovolenou dobu zpoždění signálu a tedy i delší vedení. Naopak dřívější umístění vzorkovacího bodu umožňuje zvětšit dovolenou fázovou chybu způsobenou malou strmostí hran.

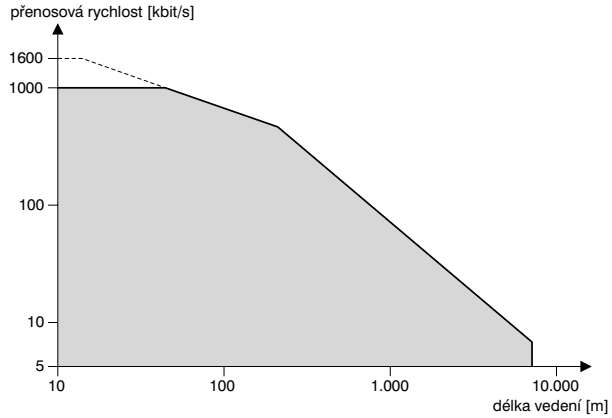
1.2.3. Přenosové médium

Pro přenos signálu se používá diferenciální vedení s charakteristickou impedancí 120 Ω. Propojení na větší vzdálenosti je nutné provést krouceným párem. Linkové vodiče se označují CAN_L a CAN_H. Na vedení se rozlišují dva stavy – neaktivní (recessive) a aktivní (dominant). Ve stavu recessive jsou oba vodiče na polovině napájecího napětí, tedy 2,5 V. Ve stavu dominant je vodič CAN_L vysílací stanicí stahován ke společnému vodiči a vodič CAN_H k napájecímu vodiči. Nominální úrovně jsou 1,5 V pro CAN_L a 3,5 V pro CAN_H. Přijímač CAN sběrnice detekuje stav dominant, pokud rozdíl napětí mezi vodiči CAN_H a CAN_L přesáhne 0,9 V. Vysílače jsou navzájem spojeny principem „otevřený kolektor“. Vyslání stavu dominant je vždy prioritní, vyšle-li kterákoliv ze stanic stav dominant, bude na sběrnici tento stav. Linka typu sběrnice je na obou koncích zakončena odpory 120 Ω. Skutečné průběhy signálů jsou zřejmé z obr. 8.

Teoreticky dosažitelnou přenosovou rychlost v závislosti na délce kabelu uvádí obr. 9. Norma ISO 11898 pro linkové budiče specifikuje maximální délku vedení 1 km, pro větší vzdálenosti je nutné použít opakovače. Dovolená délka vedení může být dále ovlivněna zpožděním reakce připojených stanic, provozní tolerancí kmitočtu oscilátorů jednotlivých stanic a



Obr. 8. Záznam průběhu signálů sběrnice CAN



Obr. 9. Závislost přenosové rychlosti na délce vedení

snížením amplitudy signálu v důsledku sériového odporu vedení. Prakticky použitelnou délku vedení uvádí tabulka:

přenosová rychlost	délka vedení	nom. délka bitu
1 Mbit/s	30 m	1 µs
800 kbit/s	50 m	1,25 µs
500 kbit/s	100 m	2 µs
250 kbit/s	250 m	4 µs
125 kbit/s	500 m	8 µs
62,5 kbit/s	1000 m	16 µs

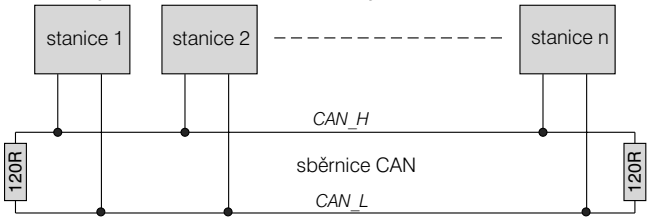
Průřez signálových vodičů by neměl být menší, než uvádí následující tabulka:

délka vedení	počet připojených stanic		
	32	64	100
100 m	0,25 mm^2	0,25 mm^2	0,25 mm^2
250 m	0,34 mm^2	0,50 mm^2	0,50 mm^2
500 m	0,75 mm^2	0,75 mm^2	1,00 mm^2

Celkový odpor vedení by neměl přesáhnout 21 Ω pro 32 stanic, 18,5 Ω pro 64 stanic a 16 Ω pro 100 stanic. Do celkového odporu je nutno započítat také přechodový odpor konektorů, svorkových spojů atd.

1.2.4. Topologie sběrnice

Z hlediska šíření signálu by sběrnice CAN měla mít lineární strukturu podle obrázku 10. V tomto případě je lhostejné, je-li centrální jednotka umístěna na kraji sběrnice nebo uvnitř.



Obr. 10. Lineární struktura sběrnice

Pokud je na sběrnici nutno utvořit odbočku, musí být splněny podmínky:

$$L_d < t_{PROPSEG} / (50 \cdot t_p) \sum_{i=1}^{n} L_{di} < t_{PROPSEG} / (10 \cdot t_p)$$

$t_{PROPSEG}$ – délka segmentu PROP_SEG bitové doby
t_p – specifické zpoždění vedení [ns/m]

Pro běžně používanou rychlost 500 kbit/s:

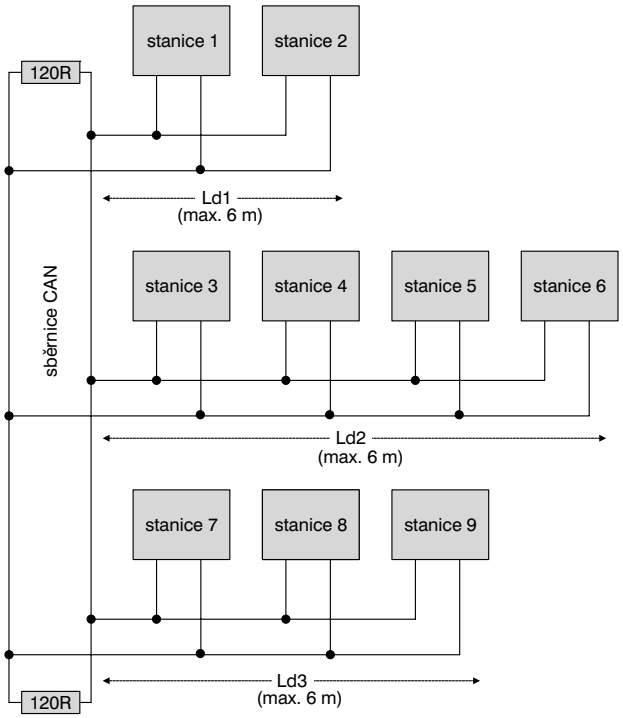
$t_{PROPSEG} = 12 \cdot tq = 12 \cdot 125\ ns = 1500\ ns$ a $t_p = 5\ ns / m$

Maximální teoretická délka jedné odbočky potom bude:

$L_d < 1500\ ns / (50 \cdot 5\ ns/m) = 6\ m$

Maximální součet teoretických délek všech odboček nesní přesáhnout:

$$\sum_{i=1}^{n} L_{di} < 1500\ ns / (10 \cdot 5\ ns/m) = 30\ m$$

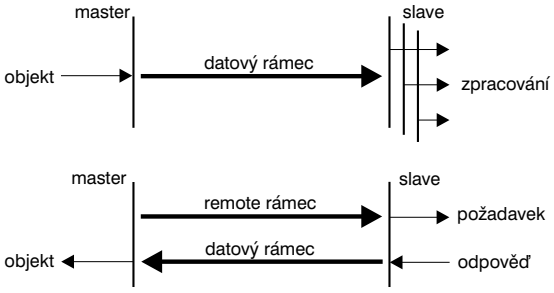


Obr. 11. Struktura sběrnice s odbočkami

Pokud je systém umístěn v jednom rozvaděči, je možné použít v podstatě libovolnou topologii sběrnice.

1.3. Linková vrstva

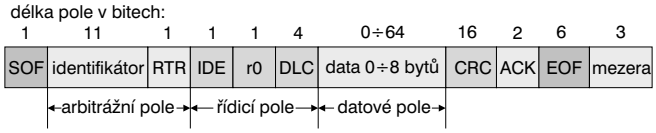
Architektura CAN komunikace nabízí několik komunikačních modelů. V systému Promos line 2 se pro přenos procesních dat používá metoda master/slave, viz obr. 12. Základní služby linkové vrstvy tedy zajišťují přenos objektu od stanice master do stanice slave a vyžádání objektu stanicí master od stanice slave. K tomuto účelu jsou na sběrnici CAN používány datové rámce (vlastní přenos až 8 bytů dat) a remote rámce (požadavek na přenos). Oba rámce mají jediný formát, který se liší pouze příznaky a fyzickou přítomností přenášených dat.



Obr. 12. Základní používané komunikační služby

1.3.1. Rámce protokolu CAN 2.0A

Pro komunikaci se požívá formát rámců podle CAN 2.0A. Strukturu rámce uvádí obrázek 13.



Obr. 13. Rámec CAN 2.0A

Rámec obsahuje následující pole:

SOF (start of frame) je startovací bit rámce, má vždy úroveň dominant. Všechny stanice používají start bit k vzájemné synchronizaci svých CAN řadičů.

Identifikátor 11-bitový identifikátor zprávy

RTR (remote request). Datový rámec má RTR bit v úrovni dominant. Rámec, který slouží k vyžádání datového rámce (remote frame) má RTR bit v úrovni recessive.

IDE (identifier extension) slouží k odlišení rozšířených rámců dle CAN 2.0B. Standardní rámce CAN 2.0A mají tento bit v úrovni dominant.

r0 rezervní bit, ve standardních rámcích CAN 2.0A má úroveň dominant

DLC (data length code) obsahuje 4 bity, určující délku datového těla (d=dominant, r=recessive):

počet datových bytů	DLC3	DLC2	DLC1	DLC0
0	d	d	d	d
1	d	d	d	r
2	d	d	r	d
3	d	d	r	r
4	d	r	d	d
5	d	r	d	r
6	d	r	r	d
7	d	r	r	r
8	r	d/r	d/r	d/r

CRC kontrolní součet rámce. Pole obsahuje 15-bitový součet následovaný 1 bitem v úrovni recessive (CRC delimiter). Kontrolní součet je vypočítáván jako zbytek po dělení polynomem $x^{15}+x^{14}+x^{10}+x^{8}+x^{7}+x^{4}+x^{3}+1$. Pokud přijímací stanice při vyhodnocení zprávy vyhodnotí neshodu přijatého a vypočítaného CRC, generuje chybový rámec pro opakování zprávy.

ACK (acknowledge) pole obshuje dva bity – ACK slot a ACK delimiter. Vysílací stanice vydává oba tyto bity ve stavu recessive. Stanice, která úspěšně přijala rámec, potvrdí příjem vysláním úrovně dominant v bitu ACK slot. Pokud vysílací stanice detekuje v bitu ACK slot stav dominant, znamená to že alespoň 1 stanice přijala rámec korektně. ACK delimiter má vždy úroveň recessive.

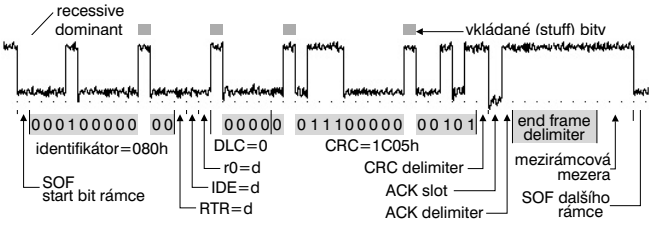
EOF (end of frame). Každý datový i remote rámec je následován sekvencí 7 bitů s úrovní recessive. Pokud nějaká stanice detekovala chybu, začne v této době vysílat chybový rámec.

Mezirámcová mezera pro zajištění synchronizace je mezi rámci mezera 3 bitových délek. Sběrnice má v této době úroveň recessive.

Záznam reálné zprávy je uveden na obrázku 14. Průběh signálu je podle zvyklostí uveden negovaný, tj. úroveň dominant je nižší než úroveň recessive. Ve skutečnosti to odpovídá kódování – úroveň dominant odpovídá nulovému bitu, recessive jedničkovému. Před vlastním rozpoznáváním je nutno vypustit vkládané (stuff) bity – po pěti po sobě následujících bitech stejné úrovně je vložen bit opačné polarity (jsou označeny šedým podtiskem). Dále je možné snadno rozpoznat jednotlivá pole. Jedná se o rámec s identifikátorem 080h, bez datového těla (DLC=0). Zpráva je potvrzena přijímací stanicí v ACK slotu. Ve skutečnosti se jedná o synchronizační rámec SYNC, proto neobsahuje ani datové tělo, ani požadavek RTR na vyslání datového rámce.

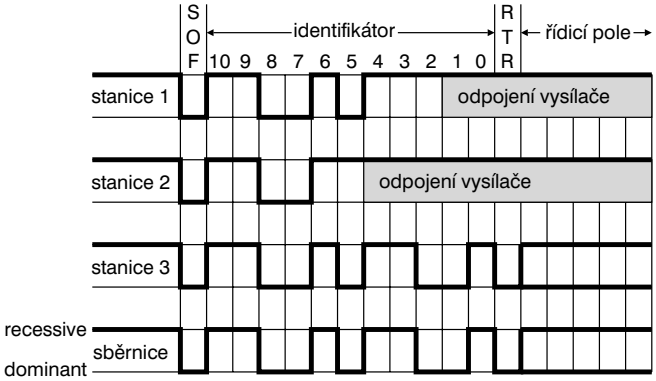
1.3.2. Arbitráž vícenásobného přístupu

Pro zamezení kolizí při vícenásobném přístupu na sběrnici se používá systém arbitáže. Každá stanice při vysílání monitoruje stav linky. Při vysílání více stanic současně má vždy přednost



Obr. 14. Záznam reálného rámce

úroveň dominant. Pokud stanice vysílá stav recessive a na lince nalezne stav dominant, přeruší vysílání a zůstane pouze na příjmu. Tento mechanismus probíhá v arbitrážním poli rámce, tedy v době vysílání identifikátoru a bitu RTR. Princip je zřejmý z obrázku 15. Stanice 1, 2 a 3 zahájily vysílání současně. Při odvysílání bitu 4 identifikátoru stanice 2 detekuje stav dominant i když vysílá recessive. Stanice 2 se tedy přeruší vysílání. Obdobně stanice 1 přeruší vysílání po odvysílání bitu 1 identifikátoru. Po odvysílání rámce stanicí 3 budou obdobně současně vysílat stanice 1 a 2.

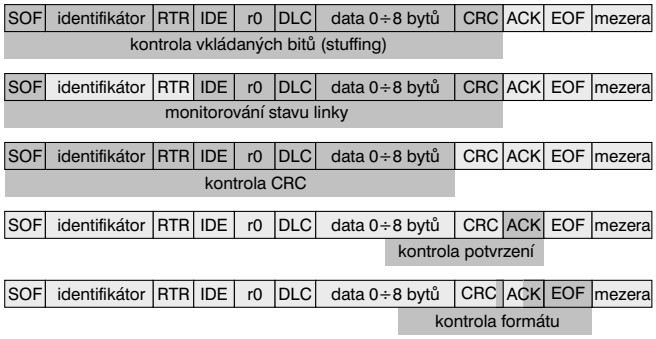


Obr. 15. Princip arbitráže sběrnice

Obecně tedy platí, že na sběrnici mají vždy přednost rámce s nižším identifikátorem, pokud je identifikátor stejný, má přednost datový rámec před remote rámcem. V protokolu CANopen mají proto vždy služební zprávy (NMT control, SYNC) nižší identifikátory a mají tak přirozenou prioritu před běžnými datovými rámci.

1.3.3. Kontrola a obsluha chyb

Protokol CAN poskytuje několik prostředků pro zabezpečení přenosu. Přehled kontrolovaných polí je uveden na obr. 16.



Obr. 16. Oblasti kontroly chyb v rámci CAN

Za provozu se kontrolují následující stavy:

Vkládání bitů (stuffing) kontroluje se počet za sebou jdoucích bitů stejné úrovně. Je-li v kontrolovaných polích objeveno 6 za sebou jdoucích bitů stejné úrovně, je v příštím bitovém intervalu generován chybový příznak. Pole ACK a EOF mají pevný formát a nejsou prokládány.

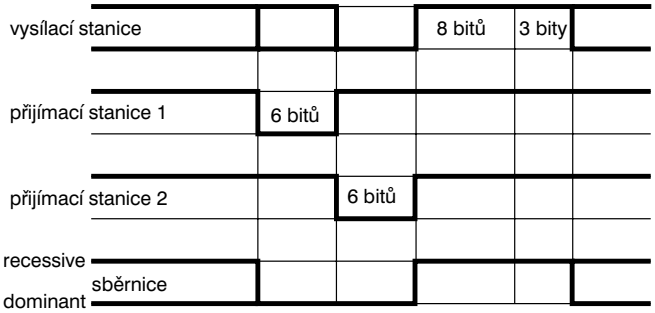
Monitorování stavu linky každá vysílací stanice při vysílání monitoruje stav linky. Při vysílání arbitrážního pole je informace použita k řízení přístupu na sběrnici. V ostatních kontrolovaných polích se při vysílání stavu recessive nesmí objevit na sběrnici stav dominant, jinak je v příštím bitovém intervalu generován chybový příznak.

CRC součástí datového i remote rámce je 15-bitový součet CRC. Pokud stanice přijímající rámec zjistí nesouhlas přijatého a vypočteného CRC, generuje chybový příznak bezprostředně po ukončení pole ACK, pokud v té době nebyl zahájen chybový příznak z jiného důvodu.

Potvrzení ACK stanice, která úspěšně přijala rámec, potvrzuje přijetí nastavením úrovně dominant v ACK slotu. Pokud vysílací stanice nenalezne dominantní potvrzovací bit, zahájí v příštím bitovém intervalu chybový příznak.

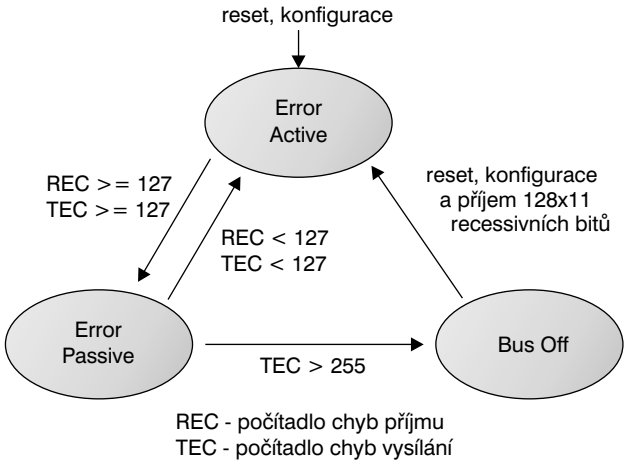
Kontrola formátu pokud vysílací stanice detekuje úroveň dominant v polích CRC delimiter, ACK delimiter a EOF, považuje přenos za neúspěšný a v příštím bitovém intervalu zahájí vysílání chybového příznaku.

Lokální detekce chyby v každé stanici se zveřejňuje na sběrnici vysláním aktivního chybového příznaku. Ten sestává ze šesti za sebou jdoucích bitových intervalů s úrovní dominant. Všechny ostatní stanice po přijetí šestého bitu detekují chybu vkládání bitů, zruší probíhající komunikaci a vysílají svůj chybový příznak. Tím se chybový stav promítne do všech připojených stanic. Situaci znázorňuje obrázek 17. Stanice 1 detekovala lokální chybu při příjmu a vysílá chybový příznak. Následně všechny ostatní stanice vysílají svůj chybový příznak. Po vypršení prodlevy 8 recessivních bitů a 3 bitů mezirámcové mezery vysílací stanice opakuje přenos.



Obr. 17. Rozšíření chyby na sběrnici

Každá stanice má vnitřní osmibitový čítač chyb samostatně pro vysílání (TEC) a příjem (REC). Stav počítadel se s výskytem chyby zvětšuje, při úspěšném přenosu rámce se snižuje. Pokud je provoz na sběrnici bez chyb, jsou obě počítadla v nulovém stavu. Vzhledem k obsluze chyb může stanice zaujímat stav aktivního zpracování chyb (Error Active) nebo pasivní stav (Error Passive). V aktivním stavu při detekci chyby vydává aktivní chybový rámec (sestává z 6 dominantních a 8 recessivních bitů). Pokud je výskyt chyb častý, zvětšuje se stav chybových čítačů. Po dosažení stavu 127 stanice přechází do pasivního stavu a po detekci chyby vysílá pouze pasivní chybový rámec (8 recessivních bitů). V pasivním stavu je stav čítače REC omezen hodnotou 127 a dále se nezvětšuje. Jestliže čítač TEC dosáhne hodnoty přes 255, dojde k přechodu do stavu odpojení sběrnice (Bus Off) a stanice ukončí veškerou aktivitu na sběrnici. V takovém případě musí dojít k softwarovému restartu a konfiguraci stanice.



Obr. 18. Stavový diagram chybových stavů

1.4. Aplikační vrstva

1.4.1. Základní vlastnosti implementace

Komunikační vrstva CANopen systému PROMOS pracuje v synchronním režimu s předem definovaným časovým rozložením rámců. Tím je zajištěno rovnoměrné vytížení sběrnice se zaručenou časovou odezvou bez nebezpečí nepředvídatelných přetížení komunikační linky. Pro použitou přenosovou rychlost 500 kbit/s je zaručený přenosový výkon čistých dat 13 kB/s.

Podporované datové typy jsou char (1 byte), int (2 byte), long/float (4 byte). Jako rozšíření lze definovat synchronizované komunikační objekty PDO, které mají garantovánu konzistenci. Je možné definovat skupinu až 5 synchronizovaných PDO a tak používat synchronizované datové objekty až 40 byte dlouhé.

Na centrální jednotce s procesorovým modulem MCPU-01 komunikační vrstva CANopen spotřebuje méně než 20% procesorového času při maximálním přenášeném objemu dat. Vrstva ke své činnosti používá časovače 3 a 7. Pro zajištění synchronizace je povoleno přerušit modul CANopen na dobu max 150 μs, průměrný výkon procesoru nesmí klesnout pod 50% maximální hodnoty (platí pro MCPU-01 s hodinovým taktem 24Mhz). Pokud je vrstva CANopen přerušena na delší dobu, může se objevit chyba komunikace. Pokud je potřeba přerušovat Canopen příliš často nebo na delší dobu, je třeba úměrně snížit objem přenášených dat snížením zatížení komunikační linky nebo snížením komunikační rychlosti.

1.4.2. Komunikační objekty

Celkový počet komunikačních objektů, využitelných pro datové přenosy je 512 pro čtení a 512 pro zápis. Identifikátory objektů se nazývají COB-ID (Communicaton object identifier) - to je vlastně 11 adresních bitů rámce CAN 2.0A . Jsou předdefinovány podle následujícího schematu:

COB-ID	10	9	8	7	6	5	4	3	2	1	0
	funkční kód				číslo modulu						

Číslo modulu (Node-ID) je identifikační číslo modulu v rámci CAN sběrnice, funční kód definuje funkci vlastního PDO objektu. Definovány jsou následující funkční kódy:

Broadcast objekty (přijímají všechny moduly)			
Objekt	**funkční kód**	**výsledný COB-ID (hex)**	**Index komunikačních parametrů (hex)**
NMT	0000	0	–
SYNC	0001	80	1005, 1006, 1007
Time Stamp	0010	100	1012, 1013

Adresované objekty			
Objekt	**funkční kód**	**výsledný COB-ID (hex)**	**Index komunikačních parametrů (hex)**
Emergency	0001	081-0FF	1014, 1015
TPDO1	0011	181-1FF	1800
RPDO1	0100	201-27F	1400
TPDO2	0101	281-2FF	1801
RPDO2	0110	301-37F	1401
TPDO3	0111	381-3FF	1802
RPDO3	1000	401-47F	1402
TPDO4	1001	481-4FF	1803
RPDO4	1010	501-57F	1403
TSDO	1011	581-5FF	1200
RSDO	1100	601-67F	1200
NMT error control	1101	701-77F	1016, 1017

1.4.3. Struktura slovníku

Slovník (Object Dictionary) je uspořádanou skupinou objektů. Přehled struktury slovníku CANopen ukazuje tabulka:

index (hex)	Objekt
0000	Rezervováno
0001-001F	Statické datové typy – standardní definice
0020-003F	Komplexní datové typy – předdefinované složené typy (20÷23 komunikační parametry)
0040-005F	Specifické datové typy
0060-009F	Datové typy profilu zařízení
00A0-00FF	Rezervováno
1000-1FFF	Komunikační profil
2000-5FFF	Profil specifikovaný výrobcem
6000-9FFF	Standardní profil
A000-FFFF	rezervováno pro aplikační objekty

Pro přístup k objektu ze slovníku se používá 16-bitový index. Pro složené datové typy (pole, struktury) jsou jednotlivé položky objektu adresovány 8-bitovým subindexem. Pokud je v objektu uložena jednoduchá proměnná, je subindex roven 0 a index adresuje přímo hodnotu proměnné. U složených typů subindex 0 určuje počet datových položek objektu (n), pod subindexy 1÷n. Jako příklad uvedeme definici komunikačního kanálu, který má jako parametry komunikační rychlost, počet datových bitů, paritu a počet stop bitů:

index	subindex	hodnota	popis
6092h	0	4	počet položek
6092h	1	9600	komunikační rychlost
6092h	2	8	počet datových bitů
6092h	3	0	parita
6092h	4	1	počet stop bitů

V systému PROMOS je použita zjednodušená struktura objektu. V jazyce C může být objekt slovníku posán:

```
struct dict {
  unsigned int index;
  char subindex;
  long data;
  };
```

Datová položka má délku vždy 4 byty, využití jednotlivých bitů v konkrétních objektech bud popsáno dále. Slovník je potom lineární pole:

```
struct dict objectdictionary[nn];
```

kde nn je velikost podle potřebného počtu objektů pro danou aplikaci.

1.4.4. Objekty globálního řízení

Vyhrazené objekty

První tři objekty slovníku centrálního modulu jsou vyhrazené pro proměnné vrstvy CANopen.

```
objectdictionary[0].data
```

udává stav komunikační vrstvy CANopen a může nabývat těchto hodnot:

hodnota	symbolicky	význam
0	initialized	normální provozní stav
1	errguardmap	fatální chyba načtení informace o připojených slave jednotkách
4	busoverload	fatální chyba - definovány příliš velké požadavky na přenos dat
5	errloadedmap	
6	errPDOmapping	fatální chyba - špatně zadané parametry guard provozní chyba - slave modul nedává informace o guard
7	errslaveguard	provozní chyba načtení informace o guard slave jednotky
10	commerror	provozní chyba na sběrnici nebo nedostatek procesorového času
0x8000 +node ID		provozní chyba guard master jednotky – chybí jednotka s číslem node ID

Fatální chyba úplně znemožní funkci CANopen a je třeba ji odstranit. Provozní chyba může vzniknout a zaniknout za provozu a je možný omezený provoz.

`objectdictionary[0].subindex`

Nabývá hodnoty 0 po ukončení komunikačního cyklu. Používají-li se synchronizované PDO, je třeba jejich obsah měnit během 2ms po detekci ukončení komunikačního cyklu, aby byla zajištěna jejich konzistence pro příští komunikační cyklus.

`objectdictionary[1].index`
`objectdictionary[1].subindex`
`objectdictionary[1].data`

Obsahují Index, subindex a data objektu probíhajícího SDO přenosu.

`objectdictionary[2].index`

COB ID vysílaného SDO

`objectdictionary[2].data`

data řízení SDO přenosu (1. byte SDO zprávy) po vyslání zprávy =0, po příjmu odpovědi vrací data typu SDO odpovědi (1. byte SDO zprávy)

Položky index jsou ve vyhrazených položkách 0,1,2. Při inicializaci CANopen vrstvy jsou nastaveny do 0.

Poslední položka v poli objectdictionary musí mít nastaven index=0xFFFF.

Konfigurace vrstvy CANopen

Centrální jednotky PROMOS line 2 s procesorovým modulem MCPU-01 mohou být konfigurovány jako aktivní nebo pasivní. Aktivní centrála řídí komunikaci na sběrnici CAN a vystupuje jako NMT master / SDO client. Taková centrála může být na sběrnici pouze jedna. Další centrální moduly na stejné sběrnici musí být konfigurovány jako pasivní, tj. NMT slave / SDO server, Potom se chová jako běžný periferní modul.

Konfigurace NMT slave

Je nutno nastavit nástedující objekty:

index	subin-dex	data	popis
1F80h	0	0	vypnutí node guarding podřízených jednotek.
100Ch	0		guard time je potřeba nastavit delší než perioda guard stanice NMT master (nejméně 3 ms)
100Dh	0	>=4	kolikátý chybný cyklus guard je považován za chybu (vzhledem k jistým omezením radiče i82527 musí být nejméně 4, tj. 12 ms)

Pro slave komunikaci centrály jsou k dispozici 4 objekty PDO pro příjem a 4 pro zápis. Pro datový PDO přenos je potřeba nastavit komunikační parametry používaných PDO v zónách 1400,1600, 1800 a 1A00h, stejně jako v případě master.

Konfigurace NMT master

NMT master jednotka je aktivní a řídí celý běh CAN sběrnice. Je potřeba nastavit následující objekty

index	subin-dex	data	popis
1400h	1	80000000h	nejvyšší bit nastavuje komunikační objekty RPDO_1-4 a TPDO_1-4 jako nepoužité (tyto objekty jsou rezervované pro slave režim)
1401h	1	80000000h	
1402h	1	80000000h	
1403h	1	80000000h	
1800h	1	80000000h	
1801h	1	80000000h	
1802h	1	80000000h	
1803h	1	80000000h	
1F80h	0	3	zapnutí hlídání (node guarding) podřízených jednotek.
1F81h	0	n	počet periferních jednotek
	1	x	x určuje parametry node guard pro jednotku s identifikátorem rovným subindexu - viz dále.
	2	x	
	...		
	n	x	

Objekt 1F81h musí mít definovány všechny subindexy, které odpovídají identifikátorům periferních modulů, jejichž stav se na sběrnici kontroluje protokolem node guarding. Parametr má následující formát:

00	pp	qq	uu
			00 – jednotka není hlídána 01 – jednotka je hlídána (guard aktivní)
		max.počet neúspěšných dotazů	
	perioda hlídání jednotky v ms		

Pokud např. objekt index 1F81 subindex 3 nastavíme na hodnotu 0x640201, bude jednotka s identifikátorem 3 testována každých 100 ms, bude tolerována maximální perioda odezvy 2x100=200 ms (pokud se jednotka jednou neozve na test přítomnosti, neznamená to ještě chybu).

Pro datovou PDO komunikaci je dále potřeba nastavit komunikační parametry.

1.4.5. Datová komunikace

Data mohou být přenášena SDO nebo PDO protokolem.

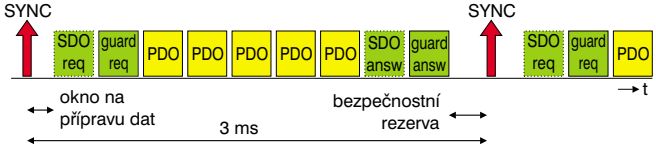
Komunikace SDO (Service Data Object) umožňuje přenos dlouhých datových bloků. Používá v procesu konfigurace jednotek nebo v provozním režimu pro přenos dat s nízkou prioritou. Komunikace SDO umožňuje přístup ke všem objektům slovníku periferní jednotky.

Komunikace PDO (Process Data Object) umožňuje přenos max. 8 bytů dat v jednom objektu. Komunikace prostřednictvím PDO je rychlá, používá jako základní mechanizmus výměny procesních dat mezi moduly výhradně v provozním režimu.

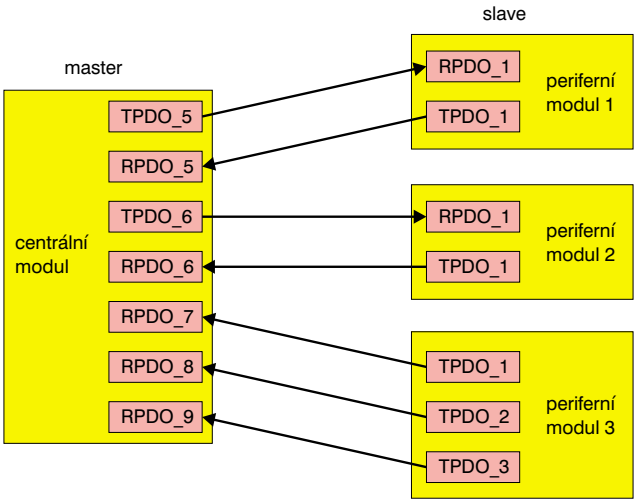
Komunikace s jednotkami PROMOS na sběrnici CAN v provozním režimu probíhá na rychlosti 500kbit/s vždy synchronně, viz obr. 19. Aktivní centrála (NMT master) s periodou 3 ms na sběrnici vysílá synchronizační zprávu (SYNC). Po příjmu SYNC si všechny periferní moduly připraví data do svých PDO. Pokud je v daném cyklu požadována SDO komunikace, zařadí se na první místo požadavek (SDO req). Následuje prostor pro 5 PDO, které se v tomto cyklu přenášejí. Pokud se na začátku cyklu vysílal požadavek SDO, je na konci cyklu SDO odpověď. Počet komunikačních objektů v jednom cyklu je volen tak, aby i v nejnepříznivějším případě (všechny rámce jsou vyžádané s odpovědí) před vysíláním dalšího SYNC zůstala bezpečnostní časová rezerva pro opakování jedné zprávy při chybě přenosu. Tak se sice nevyužije maximální přenosová kapacita sběrnice, ale je jisté, že za žádných okolností nemůže dojít k přetížení sběrnice a tím i ke zhroucení komunikace. Rozložení PDO do jednotlivých komunikačních cyklů provádí komunikační vrstva podle počtu obsluhovaných PDO a zadaných požadavků na jejich periodu. Pokud jsou požadavky na komunikaci takové, že je není možné uspokojit, bude generována chyba dříve, než dojde ke spuštění systému.

1.5. PDO protokol

Model komunikace je master/slave, přenos zahajuje centrální modul, perifení moduly jsou vždy podřízené - slave. V základním modelu komunikace mohou periferní moduly podporovat 4 PDO přijímací (RPDO) a 4 PDO vysílací (TPDO). Prostřednictvím PDO je tedy možné s jedním modulem přenášet 32 bytů tam a 32 zpět. Podle profilu mají moduly vždy předdefinováno několik PDO, které umožňují přístup k základním datovým objektům, ostatní PDO je možné namapovat na potřebné veličiny v procesu konfigurace.



Obr. 19. Organizace provozu na sběrnici CAN



Obr. 20. Napojení PDO master a slave

V centrální jednotce je počet PDO omezen definicí COB-ID objektu - je možné přenášet celkem 4x127=508 TPDO a stejný počet RPDO. První 4 RPDO a 4TPDO vyhrazeny pro slave režim, ostatní pro master režim centrálního modulu. Napojení objektů periferních modulů na objekty centrály je zcela libovolné a určuje se komunikačním profilem. Příklad napojení PDO ilustruje obr. 20. Objekty TPDO CPU jsou napojeny na RPDO periferních modulů a naopak.

Propojení jednotlivých PDO s procesními veličinami je určeno komunikačním profilem. Ten je uložen v objektech slovníku v oblasti 1000-1FFFh.

1.5.1. Komunikační parametry a napojení PDO

Pro parametry PDO přijímané komunikační vrstvou (RPDO) jsou vyhrazeny objekty 1400÷15FF, pro PDO vysílané (TPDO) objekty 1800÷19FF. Nastavení parametrů je rozdílné pro konfiguraci NMT master nebo slave.

index	označení
1400h	parametry RPDO_1 v režimu slave
1401h	parametry RPDO_2 v režimu slave
1402h	parametry RPDO_3 v režimu slave
1403h	parametry RPDO_4 v režimu slave
1404h	parametry RPDO_5 v režimu master
. . .	. . .
15FF	parametry RPDO_508 v režimu master

index	označení
1800h	parametry TPDO_1 v režimu slave
1801h	parametry TPDO_2 v režimu slave
1802h	parametry TPDO_3 v režimu slave
1803h	parametry TPDO_4 v režimu slave
1804h	parametry TPDO_5 v režimu master
. . .	. . .
19FF	parametry TPDO_508 v režimu master

Položky mají následující formát:

index	subindex	význam
14xxh 15xxh 18xxh 19xxh	1	identifikátor COB-ID
	2	perioda komunikace
	4	synchronizace objektů (nepovinná)

Povinné jsou subindexy jsou 1 a 2.

Subindex 1 určuje identifikátor komunikačního objektu. Dolních 11 bitů obsahuje COB-ID (viz 1.4.2.), nejvyšší bit (bit 31) určuje, zda je objekt aktivní (bit 31=0) nebo nepoužitý (bit 31=1)

Subindex 2 pro NMT master udává periodu komunikace v násobcích 3 ms, hodnota může být 1 až 240. Při zadání např. hodnoty 50 bude komunikace tohoto objektu zařazena každých 150 ms.

Pro NMT slave určuje typ komunikačního objektu a musí být vyplněn hodnotou 0 pro RPDO a 252h pro TPDO.

Subindex 4 umožňuje provádět vzájemně synchronizované přenosy PDO. Pokud tento subindex není definován nebo je vyplněn hodnotou 0, není tento objekt synchronizován s jinými PDO. Při nastavení hodnoty 1÷255 je tento objekt zařazen do skupiny 1÷255 a bude přenášen v jednom komunikačním okně zároveň s ostatními objekty, které mají nastavenou stejnou skupinu až do vyčerpání přenosových možností sběrnice. Maximální počet objektů v jedné skupině je 5. Pro napojení objektů podle obrázku 20musíme pro master modul nadefinovat následující objekty:

index	subind	hodnota	význam
1404h	1	181h	RPDO_5 na TPDO_1 modulu 1
1404h	2	20	perioda 60 ms
1404h	4	0	nesynchronizovaný
1405h	1	182h	RPDO_6 na TPDO_1 modulu 2
1405h	2	20	perioda 60 ms
1405h	4	0	nesynchronizovaný
1406h	1	183h	RPDO_7 na TPDO_1 modulu 3
1406h	2	20	perioda 60 ms
1406h	4	0	nesynchronizovaný
1407h	1	283h	RPDO_8 na TPDO_2 modulu 3
1407h	2	20	perioda 60 ms
1407h	4	0	nesynchronizovaný
1408h	1	383h	RPDO_9 na TPDO_3 modulu 3
1408h	2	20	perioda 60 ms
1408h	4	0	nesynchronizovaný
1804h	1	201h	TPDO_5 na RPDO_1 modulu 1
1804h	2	50	perioda 150 ms
1804h	4	1	synchronizační skupina 1
1805h	1	202h	TPDO_6 na RPDO_1 modulu 2
1805h	2	50	perioda 150 ms
1805h	4	1	synchronizační skupina 1
1806h	1	203h	TPDO_7 na RPDO_1 modulu 3
1806h	2	50	perioda 150 ms
1806h	4	1	synchronizační skupina 1

Nastavení umožní potřebnou komunikaci, objekty RPDO_1 ÷ RPDO_3 master modulu budou navzájem synchronizovány a přenášeny vždy současně v jednom komunikačním okně.

1.5.2. Mapování aplikačních objektů

Každý objekt PDO může obsahovat až 8 bytů dat. Jaká data jsou daným PDO přenášena, určují mapovací objekty. Pro PDO přijímané (RPDO – z venku do CPU) mapovací informace obsahují objekty 1600÷17FFh, pro vysílané (TPDO – z CPU ven) objekty 1A00-1BFFh.

index	označení
1600h	mapování RPDO_1 v režimu slave
1601h	mapování RPDO_2 v režimu slave
1602h	mapování RPDO_3 v režimu slave
1603h	mapování RPDO_4 v režimu slave
1604h	mapování RPDO_5 v režimu master
. . .	. . .
17FF	mapování RPDO_508 v režimu master

index	označení
1A00h	mapování TPDO_1 v režimu slave
1A01h	mapování TPDO_2 v režimu slave
1A02h	mapování TPDO_3 v režimu slave
1A03h	mapování TPDO_4 v režimu slave
1A04h	mapování TPDO_5 v režimu master
. . .	. . .
1BFF	mapování TPDO_508 v režimu master

Položky mají následující formát:

index	subindex	význam
16xxh	0	počet namapovaných dat 1 až 64
17xxh	1	mapování pole 1
1Axxh		
1Bxxh	2	mapování pole 2

Datová položka subindexu 1 a 2 určuje aplikační objekt:

index	subindex	délka
		1 byte délka pole v bitech, může být 8,16 nebo 32 bitů
	1 byte subindex objektu 0÷255	
2 byty index aplikačního objektu		

V jednom PDO může být mapováno nejvýše 8 polí. Délka každého z polí může být 8, 16 (10h) nebo 32 (20h) bitů, celková délka však musí být nejvýše 64 bitů. Pro zjednodušenou vrstvu CANopen master pro procesory Fujitsu (např. CANpig) je počet mapovaných polí omezen na dvě.

Pokud zadáme např.:

index	sub-index	hodnota	význam
1604h	0	2	dvě datová pole
1604h	1	A1010120h	4 byty objektu A101 sub 1
1604h	2	A1010220h	4 byty objektu A101 sub 2

bude to znamenat, že RPDO_5 (master -určené objektem 1404) bude mít datovou část celkem 8 bytů, z toho první 4 byty budou data aplikačního objektu A101 subindex 1 a další 4 byty data objektu A101 subindex 2:

RPDO_5:	D1	D2	D3	D4	D5	D6	D7	D8
	data objektu A101 subindex 1				data objektu A101 subindex 2			

1.5.3. Propojení aplikačních objektů a proměnných

Pro aplikační objekty jsou v režimu NMT master vyhrazeny objekty A100÷BFFFh v režimu NMT slave objekty 6000-7FFFh. Datová část aplikačního objektu obsahuje fyzickou adresu aplikační proměnné (v syntaxi C tedy např. &vstup[1]). Rozložení aplikačních objektů a jejich subindexů není nijak specifikováno, určuje je tvůrce aplikačního programu. Délka aplikační proměnné (char, int, long, ...) musí souhlasit s délkou uvedenou v mapovacích objektech. Např. pro předcházející příklad mapování objektů A101 může být propojení specifikováno například tak, že pro proměnné

```
unsigned long in[2];
```

bude aplikační objekt specifikován:

index	sub- index	hodnota	význam
A101h	0	2	dvě datová pole
A101h	1	&in[0]	adresa 1
A101h	2	&in[1]	adresa 2

1.5.4. Shrnutí

Na obrázku 21je jednoduchý příklad struktury konfigurace komunikačních vazeb centrály (NMT master) se dvěma periferními moduly. Modul 1 je např. vstupní jednotka se 16 logickými vstupy, která má nakonfigurovány dva PDO – v prvním je stav vstupů a ve druhém jsou dva šestnáctibitové čítače vstupů 0 a 1. Modul 2 je výstupní jednotka s 16 logickými výstupy.

Z obrázku je patrná struktura konfigurace:

- aplikační objekty propojují proměnné aplikačního programu, určují tedy s jakými daty se pracuje a kde jsou uložena
- mapovací objekty určují, které aplikační objekty se přenášejí v konkrétních PDO, to znamená jak se budou jednotlivá data přenášet
- parametry PDO určují propojení na PDO vnějších jednotek, tedy kam či odkud se budou data přenášet.

1.6. SDO protokol

Pro konfiguraci periferních modulů v předoperačním režimu a pro přístup k parametrům, které nejsou mapovány pro PDO přenos se používá SDO protokol. S pomocí služeb SDO je možný přístup k libovolným objektům připojených modulů.

Pro řízení SDO přenosu jsou používány vyhrazené položky slovníku 1 a 2.

```
objectdictionary[1].index
objectdictionary[1].subindex
objectdictionary[1].data
```

Obsahuje Index, subindex a data objektu probíhajícího SDO přenosu.

```
objectdictionary[2].index
```

COB-ID vysílaného SDO – pro přenos SDO jsou pro jednotky s identifikátorem 1÷7Fh vyhrazeny odpovídající hodnoty COB-ID 601÷67Fh.

```
objectdictionary[2].data
```

obsahuje kód SDO přenosu:

40h pro přenos z vnějšího modulu do centrály (čtení)
22h pro přenos z centrály do vnějšího modulu (zápis)

Po vykonání SDO přenosu je v zde uložen výsledek operace:

2200h po úspěšném průběhu čtení
6000h po úspěšném ukončení zápisu
8000h po ukončení přenosu s chybou.

Provedení SDO přenosu bude zřejmé z příkladu – v modulu s identifikátorem 1 zapíšeme objekt 6003h subindex 1 na hodnotu FFFFh. Je nutno vykonat následující přiřazení:

```
objectdictionary[1].index=0x6003;
objectdictionary[1].subindex=1;
objectdictionary[1].data=0xFFFF;
objectdictionary[2].index=0x601;
objectdictionary[2].data=0x22;
```

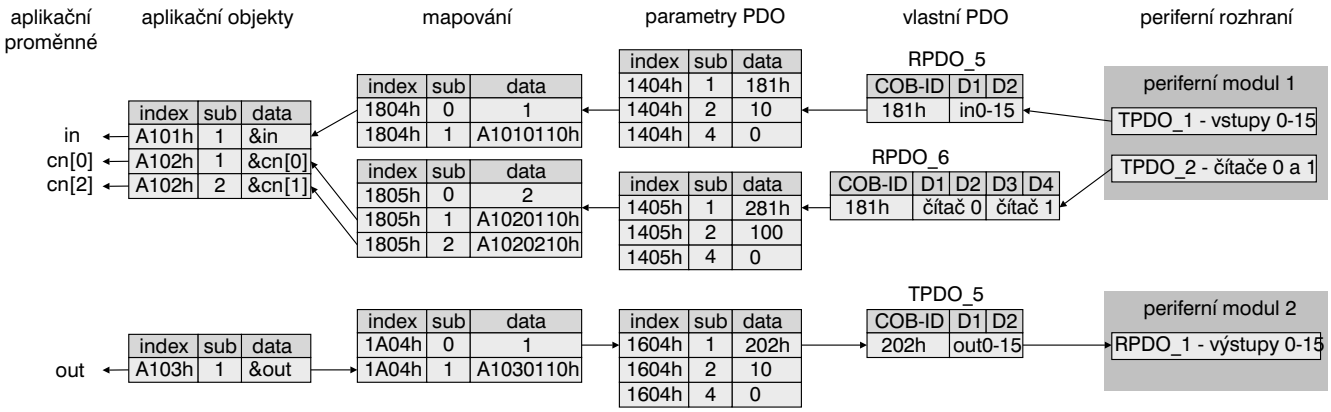
Provedením posledního přiřazení se spustí vlastní přenos.

Po úspěšném provedení (může trvat 3÷6 ms) bude v objectdictionary[2].data nastaven kód 0x6000.

Obdobně pro čtení objektu 6300h subindex 2 z modulu s identifikátorem 5 provedeme:

```
objectdictionary[1].index=0x6300;
objectdictionary[1].subindex=2;
objectdictionary[2].index=0x601;
objectdictionary[2].data=0x40;
```

Po úspěšném provedení bude objectdictionary[2].data nastaven kód 0x2200.



Obr. 21. Příklad struktury konfigurace přenosu dat v CANopen pro centrálu NMT master

2. KOMUNIKAČNÍ PROTOKOL EPSNET

Jako přenosové médium se používá běžná asynchronní sériová linka s rozhraním RS485. Pro dvoubodové spojení (1 Master + 1 Slave) je možné použít i jiné rozhraní (RS232, proudová smyčka). Způsob přenosu je poloduplexní, vysílat může vždy pouze jeden účastník komunikace. Pro přenos se používá formát s 8 datovými bity, obvykle sudou paritou (even) a 1 stop bitem. Rychlost přenosu závisí na použitém vedení a musí být stejná pro všechny stanice ve společné síti. Délka telegramu je proměnná, maximální povolená délka včetně rámce je 255 byte.

2.1. Struktura protokolu a typy zpráv

Struktura telegramů vychází z definice komunikačního protokolu PROFIBUS (DIN 19245). Popsán je přístup na sběrnici, vytváření a dekódování komunikačních paketů a přenos datových bloků.

2.1.1. Formáty telegramů

Komunikace vždy probíhá na principu výzva/odpověď. Každá stanice musí mít svoji unikátní síťovou adresu. V telegramech je vždy uvedena adresa odesílatele a příjemce, výjimkou je pouze krátké potvrzení. Výzvu zasílá vždy pouze aktivní stanice (Master). Zpracování výzvy provádí pouze stanice, jejíž síťová adresa souhlasí s adresou příjemce (DA) ve zprávě. Pokud výzva vyžaduje odpověď, odesílá ji přijímací stanice ihned po zpracování výzvy. Tím je ukončena jedna komunikační relace.

Při vysílání výzvy či odpovědi nesmí být mezi jednotlivými byty prodleva delší než trojnásobek doby vysílání jednoho znaku. Před začátkem další komunikační relace (tj. další výzvy) musí být linka v klidovém stavu (nesmí být vysílán žádný znak) po dobu alespoň trojnásobku doby vysílání jednoho znaku. Tuto dobu (tzv. SYN time) detekují všechny stanice v síti a interpretují ji jako ukončení předchozí komunikační relace. To umožňuje zasynchronizování všech přijímačů při rozpadu komunikační relace (chyba, rušení, ztráta spojení, zhroucení stanice ap.).

Aktivní stanice kontroluje také dobu reakce stanice, tj. dobu mezi ukončením přenosu výzvy a začátkem přenosu odpovědi. Maximální doba reakce (timeout) musí respektovat dopravní zpoždění použité přenosové cesty, např. při použití radiomodemu bude dopravní zpoždění podstatně větší, než při použití pevného vedení.

Zpráva bez datového pole

Používá se pro výzvy a potvrzení bez předávání dat.

Byte	Označení	Význam
0	SD1	Start Delimiter 1 úvodní znak – pevná hodnota 0x10
1	DA	Destination Address sběrnicová adresa příjemce telegramu
2	SA	Source Address sběrnicová adresa odesílatele telegramu
3	FC	Frame Control – řídící byte rámce
4	FCS	Frame Check Sum – kontrolní součet rámce
5	ED	End Delimiter ukončovací znak – pevná hodnota 0x16

Význam zprávy určuje řídící znak rámce FC. FCS obsahuje kontrolní součet znaků DA, SA a FC.

Zpráva s datovým polem

Používá se pro výzvy a potvrzení obsahující datovou část.

Byte	Označení	Význam
0	SD2	Start Delimiter 2 úvodní znak – pevná hodnota 0x68
1	LE	Length – délka datového těla +3
2	LER	Length Repeat – opakovaný byte LE
3	SD2R	Start Delimiter 2 Repeat pevná hodnota 0x68
4	DA	Destination Address sběrnicová adresa příjemce telegramu
5	SA	Source Address sběrnicová adresa odesílatele telegramu
6	FC	Frame Control – řídící byte rámce
7÷3+LE	data	Datové tělo zprávy. Délka datového těla je LE-4 bytů, nejméně však 1
4+LE	FCS	Frame Check Sum – kontrolní součet rámce
5+LE	ED	End Delimiter ukončovací znak – pevná hodnota 0x16

Význam zprávy (výzva, odpověď) určuje řídící byte rámce FC. Byte LE (LER) obsahuje počet bytů položek DA, SA, FC a datového těla. Může nabývat hodnot 4÷249. Vlastní délka datového těla je tedy LE–3, tj. 1÷246 byte. FCS obsahuje kontrolní součet znaků DA, SA, FC a bytů datového těla.

Krátké potvrzení

Používá se jako kladná odpověď na přijetí výzvy s datovým tělem, pokud v odpovědi nejsou přenášena data.

Byte	Označení	Význam
0	SACK	Short Acknowledge – pevná hodnota 0xE5

Zpráva Token

V síti s více aktivními stanicemi (Multi Master) si pomocí této zprávy předávají Token, tj. oprávnění k aktivní činnosti na sběrnici. Pasivní stanice tuto zprávu ignorují. Zpráva se nepoužívá v sítích Master-Slave s jedinou stanicí Master (Mono Master).

Byte	Označení	Význam
0	SD4	Start Delimiter 4 úvodní znak – pevná hodnota 0xDC
1	DA	Destination Address sběrnicová adresa příjemce telegramu
2	SA	Source Address sběrnicová adresa odesílatele telegramu

2.1.2. Struktura protokolu

LE – délka datového těla

V telegramech s proměnnou délkou SD2 se pro zajištění větší bezpečnosti přenosu délka datového těla uvádí dvakrát (byty LE a LER). Binární obsah LE určuje počet bytů datového těla včetně SA, DA a FC. Nejmenší hodnota LE je 4, potom telegram obsahuje právě 1 datový byte. Menší hodnota než 4 není přípustná. Maximální hodnota LE je 249, v tom případě datové tělo obsahuje 246 byte a délka celého telegramu je 255 byte.

SA, DA – byte adresy

Byty DA a SA obsahují účastnickou síťovou adresu příjemce a odesílatele telegramu.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Ext	Adresa stanice						

Ext (bit 7) vyhrazeno pro rozšíření adresy (v této verzi není implementováno)

Adresa (bity 0÷6) obsahuje binární adresu stanice v rozsahu 1÷126. Adresa 0 je vyhrazena pro budoucí použití. Adresa 127 je globální adresa pro zprávy typu Broadcast (pouze pro zprávy Send Data with No ACK). Periferní jednotky PL2 umožňují nastavení adres v rozsahu 1÷32.

FC – řídící byte rámce

Byte FC označuje typ telegramu. Typ může být výzvový nebo potvrzovací. Navíc řídící byte obsahuje přenosovou funkci a řídící informace, které zabraňují ztrátě či dublování zprávy.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Res	FR = 1	FCB	FCV	Funkce			
	FR = 0	Stn-type		b3	b2	b1	b0

Res (bit 7) Bit je rezervován (dle IEC-57, p. 5-2).

FR (bit 6) 1 = výzvový telegram, 0 = potvrzovací nebo odpovídací telegram

Pro b6=1 – výzvový telegram

FCB – Frame Count Bit (bit 5) alternující bit sledu výzvových zpráv

FCV – Frame Count bit Valid (bit 4)

0 = alternující funkce FCB neplatná
1 = alternující funkce FCB platná

Funkce pro výzvový telegram:

HEX	Funkce
0÷2	Rezervováno
3	Send Data with Acknowledge, low priority
4	Send Data with No Acknowledge, low priority
5	Send Data with Acknowledge, high priority
6	Send Data with No Acknowledge, high priority
7	Rezervováno
8	Rezervováno
9	Request FDL Status
A	Request Data with Reply, high priority
B	Request Data with Reply, low priority
C	Send and Request Data, low priority
D	Send and Request Data, high priority
E	Request Ident with Reply
F	Request LSAP-Status with Reply

Pro b6=0 – potvrzovací telegram

Stn-type

0 0 = pasívní stanice
0 1 = aktivní stanice není připravena
1 0 = aktivní stanice připravena pro Token
1 1 = aktivní stanice v Token-Ringu

Funkce pro potvrzovací telegram

HEX	Funkce
0	Acknowledge Positive (OK)
1	ACK negative, FDL/User Error (UE)
2	ACK negative, no resource available (RR)
3	ACK negative, no service activated (RS)
4	ACK positive with Request Data (RH)
5	ACK negative with Request Data (RR)
6	ACK negative with Request Data (RS)
7	Rezervováno
8	Response FDL/FMA data
9	ACK negative, no response data available (NR)
A	Response FDL/FMA data with Req. Data (RH)
B	ACK negative with Request Data (NR)
C÷F	Rezervováno

FCB, FCV – alternace sledu výzev

Pomocí bitu sledu výzev FCB (bit 5) se zabraňuje dublování zpráv u potvrzujícího účastníka a ztráta u vysílacího účastníka. Iniciátor k tomu musí pro každého příjemce vést jeden FCB. Když se výzvový telegram posílá poprvé některému příjemci musí mít nastaven FCV=0 a FCB=1. Příjemce takový telegram vyhodnocuje jako nový cyklus zpráv a ukládá si FCB spolu s adresou odesílatele (SA). V následujících telegramech stejnému příjemci musí iniciátor nastavit FCV=1 a postupně střídat nastavení FCB. Příjemce při příjmu telegramu s FCV=1 interpretuje změněný FCB jako příznak nové zprávy. Pokud FCB je stejný jako v bezprostředně předcházející zprávě stejného iniciátora, pak se jedná o opakování výzvy. V takovém případě se opakuje odeslání posledního odpovídacího nebo potvrzovacího telegramu. Příjemce musí mít pro tento případ uchován poslední odesílaný telegram spolu s adresou odesílatele až do příchodu zprávy, která se nepotvrzuje nebo příjmu telegramu Token nebo zprávy od iniciátora s jinou adresou (SA).

b5 FCB	b4 FCV	
0	0	FCB nevyhodnocovat
X	1	FCB kontrolovat
1	0	FCB uložit pro příště

FCS – kontrolní součet

U telegramů pevné délky bez datového pole (SD1) byte FCS obsahuje aritmetický součet DA, SA a FC. Při výpočtu součtu se ignoruje přenos do vyššího bytu.

U telegramů s datovým polem (SD2) byte FCS obsahuje součet DA, SA, FC a všech bytů datového těla. Při výpočtu součtu se ignoruje přenos do vyššího bytu.

2.2. Komunikační služby a přenos dat

2.2.1. Přehled služeb

V současné implementaci pasivní stanice jsou pro komunikaci k dispozici následující služby:

CONNECT navázání spojení
READN čtení bloků dat
WRITEN zápis bloků dat
WANDRN současný zápis a čtení bloků dat

Služby pro čtení a zápis datových bloků umožňují přenos max. 246 datových bytů v jedné zprávě. Vícenásobné čtení dlouhých datových oblastí zajišťuje aktivní stanice. V jedné komunikační službě je možné číst i zapisovat data různých datových bloků.

Podrobný popis jednotlivých služeb včetně struktury rámců je uveden v další kapitole.

2.2.2. Komunikační služby

CONNECT

Funkce je určena pro kontrolu spojení, např. při zahájení komunikace.

Výzva:

Byte	Označení	Význam
0	SD1	pevná hodnota 0x10
1	DA	sběrnicová adresa stanice PROMOS
2	SA	sběrnicová adresa odesílatele telegramu
3	FC	0x69
4	FCS	kontrolní součet rámce
5	ED	pevná hodnota 0x16

Odpověď:

Byte	Označení	Význam
0	SD1	pevná hodnota 0x10
1	DA	sběrnicová adresa odesílatele výzvy
2	SA	sběrnicová adresa stanice PROMOS
3	FC	0x00 – komunikace navázána
4	FCS	kontrolní součet rámce
5	ED	pevná hodnota 0x16

READN

Funkce umožňuje čtení jedné nebo více částí zveřejněných datových bloků. Každá čtená zóna je označena v datovém těle 4 bytovým deskriptorem, obsahujícím číslo bloku BLK, počáteční offset v bloku OFFS a délku LEN. Číslo bloku je 0÷255, offset v bloku je 0÷65535 a délka 1÷246. Počet čtených zón (tj. počet deskriptorů) je (LE – 4)/4.

Jednou funkcí je možné číst různé zóny jednoho bloku. Celková délka vyžádaných datových zón však nesmí překročit 246 bytů. Přenos větších datových oblastí musí zajistit aktivní stanice vícenásobným přístupem.

Stanice PROMOS provádí kontrolu čísel vyžádaných bloků a hranic zón. V případě vyžádání zóny přesahující zveřejněné bloky je vráceno negativní potvrzení s FC=0x0C.

V případě, že některá ze čtených zón je ve stanici PROMOS dočasně blokována pro přístup ze sítě (např. pro zajištění kontinuity dat v bloku), je vráceno negativní potvrzení s FC=0x09. V takovém případě je třeba požadavek opakovat.

Výzva:

Byte	Označení	Význam	
0	SD2	pevná hodnota 0x68	
1	LE	délka datového těla 4+4×počet deskriptorů	
2	LER	opakovaný byte LE	
3	SD2R	pevná hodnota 0x68	
4	DA	sběrnicová adresa stanice PROMOS	
5	SA	sběrnicová adresa odesílatele výzvy	
6	FC	0x6C	
7	0x0B	kód operace	
8	BLK	číslo bloku	
9	$OFFS_L$	offset v bloku (LSB)	první deskriptor čtené zóny
10	$OFFS_H$	offset v bloku (MSB)	
11	LEN	počet čtených bytů	
12	BLK	číslo bloku	
13	$OFFS_L$	offset v bloku (LSB)	druhý deskriptor čtené zóny
14	$OFFS_H$	offset v bloku (MSB)	
15	LEN	počet čtených bytů	
	•••		
LE	BLK	číslo bloku	
1+LE	$OFFS_L$	offset v bloku (LSB)	poslední deskriptor čtené zóny
2+LE	$OFFS_H$	offset v bloku (MSB)	
3+LE	LEN	počet čtených bytů	
4+LE	FCS	kontrolní součet rámce	
5+LE	ED	pevná hodnota 0x16	

Pozitivní odpověď:

Byte	Označení	Význam
0	SD2	pevná hodnota 0x68
1	LE	délka datového těla 3+celkový počet vyžádaných bytů
2	LER	opakovaný byte LE
3	SD2R	pevná hodnota 0x68
4	DA	sběrnicová adresa odesílatele výzvy
5	SA	sběrnicová adresa stanice PROMOS
6	FC	hodnota 0x08
7	data 1	první byte první vyžádané zóny
	•••	
3+LE	data n	poslední byte poslední vyžádané zóny
4+LE	FCS	kontrolní součet rámce
5+LE	ED	pevná hodnota 0x16

Negativní odpověď:

Byte	Označení	Význam
0	SD1	pevná hodnota 0x10
1	DA	sběrnicová adresa odesílatele výzvy
2	SA	sběrnicová adresa stanice PROMOS
3	FC	0x02 – neznámá komunikační služba 0x09 – data momentálně nedostupná 0x0C – nesprávně zadané parametry
4	FCS	kontrolní součet rámce
5	ED	pevná hodnota 0x16

WRITEN

Funkce umožňuje zápis do jedné nebo více částí zveřejněných datových bloků. Každá zapisovaná zóna je v datovém těle označena 4 bytovým deskriptorem, obsahujícím číslo bloku BLK, počáteční offset v bloku OFFS a délku LEN, následovaným vlastními přenášenými daty. Číslo bloku je 0÷255, offset v bloku je 0÷65535 a délka 1÷242.

Celkový počet zapisovaných dat včetně všech deskriptorů nesmí přesáhnout 246 bytů. Jednou funkcí je možné zapisovat I do různých zón jednoho bloku.

Stanice PROMOS provádí kontrolu čísel zapisovaných bloků a hranic zón. V případě určení zóny přesahující zveřejněné bloky je vráceno negativní potvrzení s FC=0x0C.

V případě, že některá ze zapisovaných zón je ve stanici PROMOS dočasně blokována pro přístup ze sítě (např. pro zajištění kontinuity dat v bloku), je vráceno negativní potvrzení s FC=0x09. V takovém případě je nutno požadavek opakovat.

Výzva:

Byte	Označení	Význam	
0	SD2	pevná hodnota 0x68	
1	LE	délka datového těla	
2	LER	opakovaný byte LE	
3	SD2R	pevná hodnota 0x68	
4	DA	sběrnicová adresa stanice PROMOS	
5	SA	sběrnicová adresa odesílatele výzvy	
6	FC	0x63	
7	0x0C	kód operace	
8	BLK	číslo bloku	
9	$OFFS_L$	offset v bloku (LSB)	první deskriptor zapisované zóny
10	$OFFS_H$	offset v bloku (MSB)	
11	LEN	počet čtených bytů	
12	data 0	první byte dat první zóny	data první zóny
	•••		
	data n	poslední byte dat první zóny	
	BLK	číslo bloku	poslední deskriptor zapisované zóny
	$OFFS_L$	offset v bloku (LSB)	
	$OFFS_H$	offset v bloku (MSB)	
	LEN	počet čtených bytů	
	data 0	první byte dat posl. zóny	data poslední zóny
	•••		
3+LE	data n	posl. byte dat posl. zóny	
4+LE	FCS	kontrolní součet rámce	
5+LE	ED	pevná hodnota 0x16	

Pozitivní odpověď:

Byte	Označení	Význam
0	SACK	pevná hodnota 0xE5

Negativní odpověď:

Byte	Označení	Význam
0	SD1	pevná hodnota 0x10
1	DA	sběrnicová adresa odesílatele výzvy
2	SA	sběrnicová adresa stanice PROMOS
3	FC	0x02 – neznámá komunikační služba 0x09 – zóna momentálně nedostupná 0x0C – nesprávně zadané parametry
4	FCS	kontrolní součet rámce
5	ED	pevná hodnota 0x16

WANDRN

Funkce umožňuje zápis do zóny jednoho bloku a současné čtení zóny stejného nebo jiného bloku.

Čtená zóna je označena v datovém těle 4 bytovým deskriptorem, obsahujícím číslo bloku BLK, počáteční offset v bloku OFFS a délku LEN. Zapisovaná zóna je obdobně určena deskriptorem, následovaným vlastními přenášenými daty. Číslo bloku je 0÷255, offset v bloku je 0÷65535.

Při zpracování funkce ve stanici PROMOS se nejprve provádí zápis dat a teprve potom čtení. Pokud jsou deskriptory zapisované a čtené zóny shodné, budou zpět načtena zapsaná data. Celkový počet bytů zapisovaných dat včetně obou deskriptorů nesmí přesáhnout 245 bytů. Celkový počet čtených dat nesmí přesáhnout 246 bytů.

Stanice PROMOS provádí kontrolu čísel bloků a hranic zón. V případě určení zóny přesahující zveřejněné bloky je vráceno negativní potvrzení s FC=0x0C.

V případě, že zapisovaná nebo čtená zóna je v modulu PL2 dočasně blokována pro přístup ze sítě (např. pro zajištění kontinuity dat v bloku), je vráceno negativní potvrzení s FC=0x09. V takovém případě je nutno požadavek opakovat.

Výzva:

Byte	Označení	Význam	
0	SD2	pevná hodnota 0x68	
1	LE	délka datového těla LEN_W+12	
2	LER	opakovaný byte LE	
3	SD2R	pevná hodnota 0x68	
4	DA	sběrnicová adresa stanice PROMOS	
5	SA	sběrnicová adresa odesílatele výzvy	
6	FC	0x6C	
7	0x0D	kód operace	
8	BLKR	číslo bloku	
9	$OFFS_{RL}$	offset v bloku (LSB)	deskriptor
10	$OFFS_{RH}$	offset v bloku (MSB)	čtené zóny
11	LEN_R	počet čtených bytů	
12	BLK_W	číslo bloku	
13	$OFFS_{WL}$	offset v bloku (LSB)	deskriptor
14	$OFFS_{WH}$	offset v bloku (MSB)	zapisované
15	LEN_W	počet čtených bytů	zóny
16	data 0	první byte dat zóny	
	•••		data zapisované zóny
3+LE	data n	poslední byte dat zóny	
4+LE	FCS	kontrolní součet rámce	
5+LE	ED	pevná hodnota 0x16	

Pozitivní odpověď:

Byte	Označení	Význam
0	SD2	pevná hodnota 0x68
1	LE	délka datového těla 3+celkový počet vyžádaných bytů
2	LER	opakovaný byte LE
3	SD2R	pevná hodnota 0x68
4	DA	sběrnicová adresa odesílatele výzvy
5	SA	sběrnicová adresa stanice PROMOS
6	FC	hodnota 0x08
7	data 0	první byte vyžádané zóny
	•••	
3+LE	data n	poslední byte vyžádané zóny
4+LE	FCS	kontrolní součet rámce
5+LE	ED	pevná hodnota 0x16

Negativní odpověď:

Byte	Označení	Význam
0	SD1	pevná hodnota 0x10
1	DA	sběrnicová adresa odesílatele výzvy
2	SA	sběrnicová adresa stanice PROMOS
3	FC	0x02 – neznámá komunikační služba 0x09 – zóna momentálně nedostupná 0x0C – nesprávně zadané parametry
4	FCS	kontrolní součet rámce
5	ED	pevná hodnota 0x16

2.3. Organizace veřejných dat PL2

Periferní jednotka s popisovanou komunikační vrstvou zveřejňuje svoje vybrané oblasti paměti. Tyto oblasti jsou s pomocí komunikačních služeb přístupné aktivním účastníkům komunikační sítě. Podle verze firmware se liší uspořádání bloků v paměti jednotky. Základní rozlišení modulů je možné podle chování indikační LED diody RUN.

Zveřejňovaná oblast může obsahovat až 256 bloků. Jednotlivé bloky mohou mít různou délku, max. délka jednoho bloku je 64 kB (65536 bytů). Bloky jsou číslovány 0÷255 (0÷FFh), zveřejněny jsou vždy kontinuálně za sebou, první blok má vždy číslo 0. Délka jednotlivých bloků může být různá, ale celý blok je vždy souvislý. Offset v bloku vždy začíná 0 (první byte v bloku) a může být až 65535 (FFFFH). Počet zveřejněných datových bloků a jejich délku určuje firmware periferní jednotky PL2.

2.3.1. Jednotky s verzí firmware 2.00 a vyšší

*Tyto jednotky lze rozpoznat tak, že po připojení napájecího napětí k jednotce se indikační LED dioda **RUN** rozsvítí **žlutě**. Jednotka nemusí mít připojenu komunikační linku.*

Vlastní význam jednotlivých datových bloků a jejich částí je obecně určen a vychází z definice objektů v Object Dictionary protokolu CANopen. Bloky nemohou obsahovat libovolné objekty, proměnné, struktury či pole jakéhokoliv typu. V jednom bloku nemohou být umístěny objekty různých typů. Vlastní interpretace datových objektů je určena uživatelským programem. Struktura je známa všem aktivním účastníkům společné komunikační sítě.

2.3.2. Jednotky bez označení verze firmware

*Tyto jednotky lze rozpoznat tak, že po připojení napájecího napětí k jednotce se indikační LED dioda **RUN** rozsvítí **červeně**. Jednotka nemusí mít připojenu komunikační linku.*

Jedná se o první vyrobené kusy, proto doporučujeme nechat firmware aktualizovat.

3. OBJECT DICTIONARY – POPIS

Slovník objektů (Object Dictionary) vychází z definice slovníku objektů protokolu CANopen. Jednotlivé objekty jsou specifikovány normou pro standardní profily zařízení CiA DSP401 pro I/O moduly, DSP403 pro terminály a DS301 pro komunikační parametry a to pouze normou definované objekty. Je možné je rozdělit do třech základních skupin:

- *komunikační* – standardizované
  – dané výrobcem
- *dané výrobcem* – konfigurační
  – procesní
- *definované profilem zařízení* – konfigurační
  – procesní

Podle specifikace CANopen může komunikace probíhat buď pomocí SDO (Service Data Objects) nebo PDO (Proces Data Objects). Další informace lze nalézt na internetu na adrese www.can-cia.de. Popis objektů je uveden s ohledem na objekty využívané jednotkami PROMOS Line 2.

3.1. SDO – Service Data Objects

Jak již název napovídá, jedná se o servisní objekty určené především k nastavování periferní jednotky. Nacházejí v oblasti bloků číslo 0x10 a výše. V těchto blocích jsou umístěny jednotlivé objekty Object Dictionary.

3.1.1. Komunikační objekty

Pomocí nich se nastavují rychlost komunikace po sběrnici, mapování PDO, lze vyčíst informace o jednotce, apod.

Standardizované komunikační objekty
Nacházejí se v rozsahu indexů 0x1000 až 0x1fff. Slouží především k nastavení komunikačních parametrů PDO. Obsahují též položky udávající druh zařízení a identifikaci jednotky (výrobce, typ, verzi a sériové číslo). Jejich součástí jsou i položky s kódem chyby, komunikačním timeoutem, atd.

Komunikační objekty určené výrobcem
Pomocí těchto objektů se nastavuje komunikační rychlost, prodleva odpovědi a je možno zjistit stav jednotky. Jsou umístěny na indexech 0x2000 až 0x2002.

3.1.2. Objekty definované výrobcem

Jsou specifické pro jednotlivá zařízení vyráběná určitým výrobcem. Nacházejí se v rozsahu indexů 0x2000 až 0x5fff.

Konfigurační
Do oblasti těchto objektů patří především objekty nastavující filtry na vstupech, linearizační konstanty analogových vstupů, předdefinované stavy výstupů apod.

Procesní
Obsahem těchto objektů jsou hodnoty čítačů, měřičů periody a frekvence.

3.1.3. Objekty podle profilu zařízení

Použití těchto objektů je přesně vymezeno normou CiA DSP-401 a nacházejí se na indexech od 0x6000 výše.

Konfigurační
Použity jsou především u terminálu k nastavení podsvitu displeje a atributů kurzoru.

Procesní
Tyto objekty obsahují procesní data – vstupy a výstupy (analogové, logické, textové na terminálu, stlačené klávesy).

3.2. Popis SDO stavebnice PL2

Je zde uveden podrobný popis jednotlivých objektů používaných jednotkami PROMOS Line 2. V následujícím popisu znamená:

- ***I*** index objektu
- ***SI*** subindex objektu

3.2.1. Standardizované komunikační objekty

I=0x1000 – Device Type
Obsahuje informace o typu jednotky (vstupní, výstupní, analogová, binární). Jsou použity čtyři nejnižší bity a možnosti jejich kombinace pro jednotky PL2 udává tabulka:

bit 3	bit 2	bit 1	bit 0	Hex	Typ jedn. PL2
0	0	0	1	0x01	CBI
0	0	1	0	0x02	CBO
0	0	1	1	0x03	CBIO
1	1	0	0	0x0c	CAIO

Objekt není možné mapovat do PDO.

I=0x1001 – Error Register
V registru je uložen kód posledně vzniklé chyby. V jednotkách PL2 není zatím využit, je přístupný a vrací hodnotu 0x00 (bez chyby).
Objekt není možné mapovat do PDO.

I=0x100c – Guard Time
Komunikační timeout v rozsahu 1 až 65535 ms. Výchozí hodnota je 600 ms. Spolu s Life Time Factor udává skutečný komunikační timeout.
Objekt není možné mapovat do PDO.

I=0x100d – Life Time Factor
Násobitel timeoutu objektu Guard Time, výchozí hodnota je 255. Celkový skutečný komunikační timeout je dán součinem Guard Time a Life Time Factor.
Objekt není možné mapovat do PDO.

I=0x1010 – Store parameters
Uložení nastavené konfigurace zadáním hesla SAVE pozpátku (tj. 65766173h). Nové nastavení se projeví až po restartování jednotky.
Objekt není možné mapovat do PDO.

I=0x1011 – Restore parameters
Obnovení uložené konfigurace zadáním hesla LOAD pozpátku (tj. 64616f6ch). Nové nastavení se projeví až po restartování jednotky.
Objekt není možné mapovat do PDO.

I=0x1018 – Identity Object
Obsahuje podrobné informace pro identifikaci výrobce a typu jednotky. K tomu slouží následující položky:

SI=0x01 – Vendor ID
zde je uvedeno číslo výrobce (jedinečné číslo registrovaného výrobce zařízení CANopen).

SI=0x02 – Product Code
Typové číslo jednotky. U jednotek PL2 s objednacím číslem EIxxxx.yy je uvedeno jen číslo xxxx.

SI=0x03 – Revision Number
Číslo revize jednotky. Je ve tvaru X.Y, přičemž obě čísla jsou šestnáctibitová.

SI=0x04 – Serial Number
Obsahuje sériové výrobní číslo daného typu jednotky. V jednotkách PL2 není momentálně použito.

Objekt není možné mapovat do PDO.

I=0x1400÷3 – Recieve PDO1÷4 Communication Parameter

Objekty pro nastavení způsobu komunikace PDO. Master jednotka si musí vyžádat konkrétní PDO konkrétní jednotky (např. PDO3 jednotky s adresou 7). Periferní jednotky samy nevysílají PDO bez předchozí výzvy.

SI=0x00 – Largest Subindex Suported

V jednotkách PL2 je pevně nastaven a nelze ho měnit.

SI=0x01 – COB-ID Used By PDO

V jednotkách PL2 je pevně nastaven a nelze ho měnit.

SI=0x02 – Transmission Type

Nastavení typu přenosu. Defaultní nastavení typu přenosu je 255 – komunikované hodnoty jsou přeneseny na výstupy jednotky okamžitě po přijetí PDO.

Další možná nastavení typu přenosu jsou 1÷240 – hodnoty z přijatého PDO jsou uloženy do vyrovnávacího bufferu. Na výstupy jednotky jsou přeneseny až po příjmu zprávy SYNC. Tato nastavení jsou možná od verze firmware 2.4. Příslušné položky v Object Dictionary mají nyní přístup RW. Změnu nastavení typu přenosu PDO je možné provést zápisem do příslušné položky v kterémkoli režimu provozu jednotky a tato změna je jednotkou akceptována okamžitě bez nutnosti restartu.

I=0x1600÷3 – Recieve PDO1÷4 Mapping Parameter

Tyto 4 objekty slouží k mapování přijímaných procesních objektů Object Dictionary. Podrobný popis způsobu mapování je popsán v kapitole 3.3.2.

I=0x1800÷3 – Transmit PDO1÷4 Communication Parameter

Objekty pro nastavení způsobu komunikace PDO. Master jednotka si musí vyžádat konkrétní PDO konkrétní jednotky (např. PDO3 jednotky s adresou 7). Periferní jednotky samy nevysílají PDO bez předchozí výzvy.

SI=0x00 – Largest Subindex Suported

V jednotkách PL2 je pevně nastaven a nelze ho měnit.

SI=0x01 – COB-ID Used By PDO

V jednotkách PL2 je pevně nastaven a nelze ho měnit.

SI=0x02 – Transmission Type

Nastavení typu přenosu. Defaultní nastavení typu přenosu je 253 – PDO je vysláno pouze po příjmu RTR s aktuálními daty ze vstupů jednotky.

Další možná nastavení typu přenosu jsou:

252 – PDO je vysláno pouze po příjmu RTR s daty získanými ze vstupů jednotky v okamžiku příjmu poslední zprávy SYNC.

1÷240 – PDO je vysláno po příjmu zprávy SYNC s aktuálními daty ze vstupů jednotky. Implementace je omezena tak, že PDO je vysíláno po každém příjmu zprávy SYNC bez ohledu na nastavení typu přenosu v rozsahu 1÷240.

Tato nastavení jsou možná od verze firmware 2.4. Příslušné položky v Object Dictionary mají nyní přístup RW. Změnu nastavení typu přenosu PDO je možné provést zápisem do příslušné položky v kterémkoli režimu provozu jednotky a tato změna je jednotkou akceptována okamžitě bez nutnosti restartu.

I=0x1a00÷3 – Transmit PDO1÷4 Mapping Parameter

Tyto 4 objekty slouží k mapování vysílaných procesních objektů Object Dictionary. Podrobný popis způsobu mapování je popsán v kapitole 3.3.2.

3.2.2. Komunikační objekty dané výrobcem

I=0x2000 – COM Speed

Objekt pro nastavení komunikační rychlosti jednotky připojené přes sériovou linku RS485. Lze nastavit rychlosti od 300 do 115 200 Baud.

I=0x2001 – COM Delay

Nastavení prodlevy odpovědi (u jednotek pro linku RS485). Je to minimální doba, po kterou musí jednotka po ukončení příjmu výzvy počkat, než začne vysílat odpověď (např. pro přepnutí směru u opakovače nebo přepnutí radiomodemu).

I=0x2002 – NMT State

Vrací momentální stav jednotky. Mohou nastat následující stavy:

BOOTUP (0x00)

Stav těsně po zapnutí. V tomto okamžiku jednotka nepřijímá ani nevysílá žádná data. Bezprostředně po odeznění tohoto stavu vyšle jednotka zprávu typu broadcast s NMT=0x00.

STOPPED (0x04)

Činnost jednotky zastavena. Je to stav vyvolaný uživatelsky zapsáním 0x04 do objektu. U jednotek s výstupy jsou výstupy nastaveny do stavu „zamrznuto“. Tento stav je výhodný např. při aktualizaci SW v centrále, není-li možné odpojit celou technologii.

OPERATIONAL (0x05)

Běžný stav jednotky. Následuje po stavu pre-operational. Během stavu stopped je možné do tohoto stavu přejít zapsáním 0x05 do tohoto objektu.

PRE-OPERATIONAL (0x7f)

Stav po BOOTUP a odeslání zprávy s NMT=0x00, který je vhodný pro konfiguraci jednotky pomocí SDO.

3.2.3. Konfigurační objekty výrobce

I=0x3100, 0x3102 – Binary Filter LO

Časová konstanta digitálního filtru binárních vstupů pro vstupní signál v úrovni logická 0. V oblasti 0x3100 jsou obsaženy konstanty pro 16 vstupů platných pro všechny jednotky (pro CBIO jen 8), pro 0x3102 je 48 vstupů 17 až 64 platných pouze pro FCPU-02A.

I=0x3101, 0x3103 – Binary Filter HI

Časová konstanta digitálního filtru binárních vstupů pro vstupní signál v úrovni logická 1. V oblasti 0x3101 jsou obsaženy konstanty pro 16 vstupů platných pro všechny jednotky (pro CBIO jen 8), pro 0x3103 je 48 vstupů 17 až 64 platných pouze pro FCPU-02A.

I=0x3102 – Binary Filter All

Časová konstanta digitálního filtru binárních vstupů shodná pro všechny vstupy a oba logické stavy.

I=0x3110 – Analog Filter

Časová konstanta digitálního filtru analogových vstupů.

I=0x3111 – Analog Filter All

Časová konstanta digitálního filtru analogových vstupů shodná pro všechny vstupy.

I=0x3200 – Bin Def

Uživatelsky definovatelné stavy logických výstupů při ztrátě komunikace (stav Guard Error).

I=0x3201 – Analog Def

Uživatelsky definovatelné stavy analogových výstupů při ztrátě komunikace (stav Guard Error).

I=0x3202 – User Char Def

Osm uživatelsky definovatelných znaků terminálu (např. české znaky).

I=0x3300 – Lin 0

Konstanta **d** linearizačního polynomu ax^3+bx^2+cx+d analogových vstupů.

I=0x3301 – Lin 1

Konstanta **c** linearizačního polynomu ax^3+bx^2+cx+d analogových vstupů.

I=0x3302 – Lin 2
Konstanta **b** linearizačního polynomu ax^3+bx^2+cx+d analogových vstupů.

I=0x3303 – Lin 3
Konstanta **a** linearizačního polynomu ax^3+bx^2+cx+d analogových vstupů.

3.2.4. Procesní objekty výrobce

I=0x4101 – Fast Counter
Obsah rychlého čítače impulsů. Jednotky s logickými vstupy obsahují jeden rychlý čítač do 2 MHz. HW jednotek PL2 v současné době tento čítač nepodporuje.

I=0x4121 – Fast Frequency
Obsah rychlého měřiče frekvence. Jednotky s logickými vstupy obsahují jeden rychlý měřič frekvence s rozsahem 2 MHz a přesností 100 Hz. HW jednotek PL2 v současné době tento měřič frekvence nepodporuje.

I=0x4100 – Counters
Obsah čítačů impulsů. Jednotky s logickými vstupy obsahují na každém vstupu jeden čítač.

I=0x4110 – Periods
Obsah měřičů periody. Jednotky s logickými vstupy obsahují na každém vstupu jeden měřič periody.

I=0x4120 – Frequence
Obsah měřičů frekvence. Jednotky s logickými vstupy obsahují na každém vstupu jeden měřič frekvence.

3.2.5. Objekty profilu zařízení

I=0x6000 – Bin In
Filtrované binární vstupy. Jsou poskytovány pouze vstupní signály po průchodu digitálním filtrem, nastavením časové konstanty 0 ms je filtr vyřazen z činnosti.

I=0x6000 – Read Key Code
Čtení klávesnice – status klávesnice a kód stisknuté klávesy terminálu.

I=0x6001 – Bin In
Binární vstupy terminálu.

I=0x6120 – Bin In
Binární vstupy jednotky CAIO-12.

I=0x6200 – Bin Out
Binární výstupy. U periferních jednotek SBO je každý čtvrtý bit nevyužit. Lze je použít pouze pro indikaci pomocí LED na předním panelu.

I=0x6200 – Write Lamp Code
Ovládání indikačních LED na panelu terminálu (svítí-nesvítí, bliká-nebliká, červená-zelená-obě).

I=0x6200 – Bin Out
Binární výstupy jednotky CAIO-12.

I=0x6201 – Bin Out
Binární výstupy terminálu.

I=0x6202 – Backlight
Nastavení podsvitu displeje terminálu.

I=0x6204 – Volume
Nastavení hlasitosti reproduktoru terminálu. Nastavení se provádí ve 4 krocích.

I=0x6210 – Text Monitor
Obsahuje atributy zobrazení na displeji a obraz displeje (80 znaků) terminálu.

I=0x6211 – Text Cursor
Umožňuje nastavit atributy a pozici kurzoru na displeji terminálu.

I=0x6400 – Analog In
Obsahuje hodnoty 4 analogových vstupů terminálu.

I=0x6401 – Analog In
Hodnoty analogových vstupů.

I=0x6410 – Analog Out
Nastavení analogových výstupů typu PWM.

I=0x6411 – Analog Out16
Nastavení analogových výstupů typu 16-bit D/A.

3.3. PDO – Proces Data Objects

PDO – objekty procesních dat slouží k rychlému přenosu těchto dat. Jedná se především o přenos hodnot vstupů a výstupů. Pomocí mapování PDO je možné posílat i jiná procesní data, např. hodnoty čítačů u binárních vstupů.
Každá periferní jednotka obsahuje 4 PDO pro vysílání a 4 PDO pro příjem. Každý PDO má délku 8 bytů, celkem je tedy k dispozici 32 bytů procesních dat pro příjem a 32 bytů procesních dat pro vysílání.

3.3.1. Výchozí nastavení PDO

Periferní jednotky mají z výroby podle typu namapovány objekty do přijímacích PDO podle následující tabulky:

RX		**CBI**	**CBO**	**CBIO**	**CAIO**	**CKDM**	**FCPU**
PDO1	0	–	6200 01	6200 01	6410 01	6200 00	6200 01
	1	–	6200 02	–	6410 02		6200 02
	2	–	–	–	6410 03	–	6200 03
	3	–	–	–	6410 04	–	6200 04
	4	–	–	–	6410 05	–	6200 05
	5	–	–	–	6410 06	–	6200 06
	6	–	–	–	3111 00	–	6200 07
	7	–	–	–	6200 01	–	6200 08
PDO2	0	3102 00	–	3102 00	6411 01	–	–
	1	–	–	–		–	–
	2	–	–	–	6411 02	–	–
	3	–	–	–		–	–
	4	–	–	–	6411 03	–	–
	5	–	–	–		–	–
	6	–	–	–	6411 04	–	–
	7	–	–	–		–	–
PDO3	0	–	–	–	6411 05	6201 01	–
	1	–	–	–		6201 02	–
	2	–	–	–	6411 06	–	–
	3	–	–	–		–	–
	4	–	–	–	6411 07	–	–
	5	–	–	–		–	–
	6	–	–	–	6411 08	–	–
	7	–	–	–		–	–
PDO4	0	–	–	–	6411 09	–	–
	1	–	–	–		–	–
	2	–	–	–	6411 0a	–	–
	3	–	–	–		–	–
	4	–	–	–	6411 0b	–	–
	5	–	–	–		–	–
	6	–	–	–	6411 0c	–	–
	7	–	–	–		–	–

Periferní jednotky mají z výroby podle typu namapovány objekty do vysílacích PDO podle následující tabulky:

TX		CBI	CBO	CBIO	CAIO	CKDM	FCPU
PDO1	0	6000 01	–	6000 01	6401 01	6000 01	6000 01
	1	6000 02	–	–			6000 02
	2	–	–	–	6401 02	–	6000 03
	3	–	–	–		–	6000 04
	4	–	–	–	6401 03	–	6000 05
	5	–	–	–		–	6000 06
	6	–	–	–	6401 04	–	6000 07
	7	–	–	–		–	6000 08
PDO2	0	4100 01	–	4100 01	6401 05	–	–
	1		–			–	–
	2	4100 02	–	4100 02	6401 06	–	–
	3		–			–	–
	4	4100 03	–	4100 03	6401 07	–	–
	5		–			–	–
	6	4100 04	–	4100 04	6401 08	–	–
	7		–			–	–
PDO3	0	4110 05	–	4110 05	6401 09	6001 01	–
	1		–			6001 02	–
	2	4110 06	–	4110 06	6401 0a	6400 01	–
	3		–			6400 02	–
	4	4110 07	–	4110 07	6401 0b	6400 03	–
	5		–			6400 04	–
	6	4110 08	–	4110 08	6401 0c	–	–
	7		–			–	–
PDO4	0	–	–	–	4100 01	–	–
	1	–	–	–		–	–
	2	–	–	–	4100 02	–	–
	3	–	–	–		–	–
	4	–	–	–	6120 01	–	–
	5	–	–	–		–	–
	6	–	–	–		–	–
	7	–	–	–		–	–

U čísel objektů znamená levé čtyřčíslí index (**I**) a pravé dvojčíslí subindex (**SI**). Všechna čísla jsou hexadecimální.

3.3.2. Mapování PDO

Umožňuje do PDO1÷4 namapovat další, popř. jiná procesní data než jsou namapována defaultně. Mapování se provádí samostatně pro každé vysílací a samostatně pro každé přijímací PDO. K mapování slouží objekty na indexech 0x160.. a 0x1a0... Obsahem každého objektu je podrobný seznam všech mapovaných objektů včetně délky každého objektu v bitech. Do jednoho PDO je tedy možné namapovat maximálně 64 bitů. U jednotek PL2 je podporováno pouze mapování po 8 bitech. Při mapování nesmí zůstat mezera mezi položkami jednoho mapovacího objektu. Nelze tedy provést mapování následujícím způsobem (jde o výsledný obsah mapovacích objektů):

Mapovací objekt			Mapovaný objekt		
I	SI	L	I	SI	Lb
1600	00	1	04 – počet objektů		
	01	4	6200	05	08
	02	4	6200	08	08
	03	4			
	04	4			
	05	4	4100	10	10
	06	4	4100	0a	10
	07				
	08				

Položky mapovacího objektu musí vždy následovat těsně za sebou. V opačném případě bude hlášena chyba. Struktura mapovacích objektů bude nejlépe patrná z následujícího příkladu:

U jednotky FCPU-02A s adresou 9 chceme ve vysílacích PDO mít vstupy pozic 0, 1, 4 a 7, rychlý čítač, periody vstupů 5, 9 a 16, frekvenci impulsů na vstupech 1, 2, 10 a 13 a čítače vstupů 10 a 16. Požadované pořadí bude *vstupy – čítače 16, 10 – frekvence – periody 5, 16, 9 – rychlý čítač*. Umístění mapovaných objektů ukazuje následující tabulka:

Mapovací objekt			Mapovaný objekt			Výsledný obsah		
I	SI	L	I	SI	Lb	I	SI	Lb
1600	00	1	06 – počet objektů			06 – počet objektů		
	01	4				6200	01	08
	02	4				6200	02	08
	03	4	6200	05	08	6200	05	08
	04	4	6200	08	08	6200	08	08
	05	4	4100	10	10	4100	10	10
	06	4	4100	0a	10	4100	0a	10
	07							
	08							
1601	00	1	04 – počet objektů			04 – počet objektů		
	01	4	4120	01	10	4120	01	10
	02	4	4120	02	10	4120	02	10
	03	4	4120	0a	10	4120	0a	10
	04	4	4120	0d	10	4120	0d	10
	05							
	06							
	07							
	08							
1602	00	1	04 – počet objektů			04 – počet objektů		
	01	4	4110	05	10	4110	05	10
	02	4	4110	10	10	4110	10	10
	03	4	4110	09	10	4110	09	10
	04	4	4101	00	10	4101	00	10
	05							
	06							
	07							
	08							
1603	00	1	počet objektů			počet objektů		
	01	4						
	02	4						
	03	4						
	04	4						
	05	4						
	06	4						
	07	4						
	08	4						

Dále je uveden příklad zprávy pro protokol Epsnet, která odpovídá předchozí mapovací tabulce. Číslu bloku **B** odpovídá vyšší byte indexu **I**, vyššímu bytu offsetu **HI** odpovídá subindex **SI**, nižšímu bytu offsetu **LO** odpovídá nižší byte indexu **I** (např. pro I=0x1602, SI=0x04 platí B=0x16, HI=0x04 a LO=0x02).

Protože do jednotky je potřeba pouze zapisovat, použije se zpráva `WRITEN`. Zpráva (výzva) bude vypadat následovně:

Byte	Označ.	Hodnota	Význam
0	SD2	`0x68`	start delimiter
1	LE	`0x73`	počet bytů 4 až 118
2	LER	`0x73`	
3	SD2R	`0x68`	start delimiter
4	DA	`0x09`	adresa příjemce
5	SA	`0x7E`	adresa odesílatele
6	FC	`0x63`	řídící byte rámce
7	0x0C	`0x0C`	kód operace
8	B	`0x16`	mapovací objekt PDO1
9	LO	`0x00`	
10	HI	`0x00`	
11	L	`0x01`	počet zapisovaných bytů
12	data	`0x06`	celkový počet mapovaných objektů
13	B	`0x16`	mapovací objekt PDO1
14	LO	`0x00`	
15	HI	`0x03`	
16	L	`0x04`	počet zapisovaných bytů

Byte	Označ.	Hodnota	Význam
17	data	0x08	3. položka PDO1
18		0x05	
19		0x00	
20		0x62	
21	B	0x16	mapovací objekt PDO1
22	LO	0x00	
23	HI	0x04	
24	L	0x04	počet zapisovaných bytů
25	data	0x08	4. položka PDO1
26		0x08	
27		0x00	
28		0x62	
29	B	0x16	mapovací objekt PDO1
30	LO	0x00	
31	HI	0x05	
32	L	0x04	počet zapisovaných bytů
33	data	0x10	5. položka PDO1
34		0x10	
35		0x00	
36		0x41	
37	B	0x16	mapovací objekt PDO1
38	LO	0x00	
39	HI	0x06	
40	L	0x04	počet zapisovaných bytů
41	data	0x10	6. položka PDO1
42		0x0A	
43		0x00	
44		0x41	
45	B	0x16	mapovací objekt PDO2
46	LO	0x01	
47	HI	0x00	
48	L	0x01	počet zapisovaných bytů
49	data	0x04	celkový počet mapovaných objektů
50	B	0x16	mapovací objekt PDO2
51	LO	0x01	
52	HI	0x01	
53	L	0x04	počet zapisovaných bytů
54	data	0x10	1. položka PDO2
55		0x01	
56		0x20	
57		0x41	
58	B	0x16	mapovací objekt PDO2
59	LO	0x01	
60	HI	0x02	
61	L	0x04	počet zapisovaných bytů
62	data	0x10	2. položka PDO2
63		0x02	
64		0x20	
65		0x41	
66	B	0x16	mapovací objekt PDO2
67	LO	0x01	
68	HI	0x03	
69	L	0x04	počet zapisovaných bytů
70	data	0x10	3. položka PDO2
71		0x0A	
72		0x20	
73		0x41	
74	B	0x16	mapovací objekt PDO2
75	LO	0x01	
76	HI	0x04	
77	L	0x04	počet zapisovaných bytů
78	data	0x10	4. položka PDO2
79		0x0D	
80		0x20	
81		0x41	
82	B	0x16	mapovací objekt PDO3
83	LO	0x02	
84	HI	0x00	
85	L	0x01	počet zapisovaných bytů
86	data	0x01	celkový počet mapovaných objektů
87	B	0x16	mapovací objekt PDO3
88	LO	0x02	
89	HI	0x01	
90	L	0x04	počet zapisovaných bytů
91	data	0x10	1. položka PDO3
92		0x05	
93		0x10	
94		0x41	
95	B	0x16	mapovací objekt PDO3
96	LO	0x02	
97	HI	0x02	
98	L	0x04	počet zapisovaných bytů
99	data	0x10	2. položka PDO3
100		0x10	
101		0x10	
102		0x41	
103	B	0x16	mapovací objekt PDO3
104	LO	0x02	
105	HI	0x03	
106	L	0x04	počet zapisovaných bytů
107	data	0x10	3. položka PDO3
108		0x09	
109		0x10	
110		0x41	
111	B	0x16	mapovací objekt PDO3
112	LO	0x02	
113	HI	0x04	
114	L	0x04	počet zapisovaných bytů
115	data	0x10	4. položka PDO3
116		0x00	
117		0x01	
118		0x41	
119	FCS	0xC1	kontrolní součet bytů 4 až 118
120	ED	0x16	end delimiter

Po této výzvě vrátí jednotka odpověď **0xE5**.

4. OBJECT DICTIONARY – SEZNAM OBJEKTŮ

V následující tabulce je uveden kompletní seznam všech objektů (včetně jednotlivých položek objektu) implementovaných v periferních jednotkách PROMOS Line 2 (se sběrnicí CAN i RS485) a verzí SW 2.00 a vyšší.

CANopen			Epsnet				Modbus				
I	SI	L	B	Lo	Hi	L	I	L	Typ	A	Popis
Standardizované komunikační položky											
1000	00	4	10	00	00	4			VAR	ro	Device type – u Modbusu není implementován
1001	00	1	10	01	00	1	0040	0001	VAR	ro	Error Register
100c	00	2	10	0c	00	2	0300	0001	VAR	rw	Guard Time [ms]
100d	00	1	10	0d	00	1	0340	0001	VAR	rw	Life Time Factor
1010	00	1	10	10	00	1	0400	0001	RECORD	ro	Store parameters – largest subindex supported
1010	01	4	10	10	01	4	0401	0002	RECORD	rw	Store parameters – save all param. (heslo „save“ pozpátku 65766173h)
1011	00	1	10	11	00	1	0440	0001	RECORD	ro	Restore default parameters – largest subindex suported
1011	01	4	10	11	01	4	0441	0002	RECORD	rw	Restore default parameters – restore all default parameters (heslo „load“ pozpátku, tj. 64616f6ch)
1018	00	1	10	18	00	1	0600	0001	RECORD	ro	Identity Object – Number of entries (počet položek)
1018	01	4	10	18	01	4	0601	0002	RECORD	ro	Identity Object – Vendor ID – (číslo výrobce)
1018	02	4	10	18	02	4	0602	0002	RECORD	ro	Identity Object – Product code
1018	03	4	10	18	03	4	0603	0002	RECORD	ro	Identity Object – Revision number
1018	04	4	10	18	04	4	0604	0002	RECORD	ro	Identity Object – Serial number
1400	00	1	14	00	00	1	1000	0001	RECORD	ro	Receive PDO1 Communication Parameter – largest subindex suported
1400	01	4	14	00	01	4	1001	0002	RECORD	ro	Receive PDO1 Communication Parameter – COB-ID used by PDO
1400	02	1	14	00	02	1	1002	0001	RECORD	rw	Receive PDO1 Communication Parameter – transmission type
1401	00	1	14	01	00	1	1040	0001	RECORD	ro	Receive PDO2 Communication Parameter – largest subindex suported
1401	01	4	14	01	01	4	1041	0002	RECORD	ro	Receive PDO2 Communication Parameter – COB-ID used by PDO
1401	02	1	14	01	02	1	1042	0001	RECORD	rw	Receive PDO2 Communication Parameter – transmission type
1402	00	1	14	02	00	1	1080	0001	RECORD	ro	Receive PDO3 Communication Parameter – largest subindex suported
1402	01	4	14	02	01	4	1081	0002	RECORD	ro	Receive PDO3 Communication Parameter – COB-ID used by PDO
1402	02	1	14	02	02	1	1082	0001	RECORD	rw	Receive PDO3 Communication Parameter – transmission type
1403	00	1	14	03	00	1	10c0	0001	RECORD	ro	Receive PDO4 Communication Parameter – largest subindex suported
1403	01	4	14	03	01	4	10c1	0002	RECORD	ro	Receive PDO4 Communication Parameter – COB-ID used by PDO
1403	02	1	14	03	02	1	10c2	0001	RECORD	rw	Receive PDO4 Communication Parameter – transmission type
1600	00	1	16	00	00	1	1800	0001	RECORD	rw	Receive PDO1 Mapping Parameter
1601	00	1	16	01	00	1	1840	0001	RECORD	rw	Receive PDO2 Mapping Parameter
1602	00	1	16	02	00	1	1880	0001	RECORD	rw	Receive PDO3 Mapping Parameter
1603	00	1	16	03	00	1	18c0	0001	RECORD	rw	Receive PDO4 Mapping Parameter
1800	00	1	18	00	00	1	1400	0001	RECORD	ro	Transmit PDO1 Communication Parameter – largest subindex suported
1800	01	4	18	00	01	4	1401	0002	RECORD	ro	Transmit PDO1 Communication Parameter – COB-ID used by PDO
1800	02	1	18	00	02	1	1402	0001	RECORD	rw	Transmit PDO1 Communication Parameter – transmission type
1801	00	1	18	01	00	1	1440	0001	RECORD	ro	Transmit PDO2 Communication Parameter – largest subindex suported
1801	01	4	18	01	01	4	1441	0002	RECORD	ro	Transmit PDO2 Communication Parameter – COB-ID used by PDO
1801	02	1	18	01	02	1	1442	0001	RECORD	rw	Transmit PDO2 Communication Parameter – transmission type
1802	00	1	18	02	00	1	1480	0001	RECORD	ro	Transmit PDO3 Communication Parameter – largest subindex suported
1802	01	4	18	02	01	4	1481	0002	RECORD	ro	Transmit PDO3 Communication Parameter – COB-ID used by PDO
1802	02	1	18	02	02	1	1482	0001	RECORD	rw	Transmit PDO3 Communication Parameter – transmission type
1803	00	1	18	03	00	1	14c0	0001	RECORD	ro	Transmit PDO4 Communication Parameter – largest subindex suported
1803	01	4	18	03	01	4	14c1	0002	RECORD	ro	Transmit PDO4 Communication Parameter – COB-ID used by PDO
1803	02	1	18	03	02	1	14c2	0001	RECORD	rw	Transmit PDO4 Communication Parameter – transmission type
1a00	00	1	1a	00	00	1	1c00	0001	RECORD	rw	Transmit PDO1 Mapping Parameter
1a01	00	1	1a	01	00	1	1c40	0001	RECORD	rw	Transmit PDO2 Mapping Parameter
1a02	00	1	1a	02	00	1	1c80	0001	RECORD	rw	Transmit PDO3 Mapping Parameter
1a03	00	1	1a	03	00	1	1cc0	0001	RECORD	rw	Transmit PDO4 Mapping Parameter
Komunikační a stavové položky											
2000	00	2	20	00	00	2	0800	0001	VAR	rw	COM speed: 3 = 300 Bd; 6 = 600 Bd; 1 = 1 200 Bd; 2 = 2 400 Bd; 4 = 4 800 Bd; 9 = 9 600 Bd; 19 = 19 200 Bd; 38 = 38 400 Bd; 57 = 57 600 Bd; 115 = 115 200 Bd
2001	00	4	20	01	00	4	0840	0002	VAR	rw	COM delay
2002	00	1	20	02	00	1	0880	0001	VAR	rw	NMT state: 0 = BOOTUP; 4 = STOPPED; 5 = OPERATIONAL; 127 = PRE–OPERATIONAL
Časová konstanta filtrů binárních vstupů v ms – logická nula na vstupu											
3100	00	1	31	00	00	1	5000	0001	ARRAY	ro	binary filter lo – počet vstupů
3100	01	1	31	00	01	1	5001	0001	ARRAY	wo	binary filter lo – vstup 1
3100	02	1	31	00	02	1	5002	0001	ARRAY	wo	binary filter lo – vstup 2
3100	03	1	31	00	03	1	5003	0001	ARRAY	wo	binary filter lo – vstup 3
3100	04	1	31	00	04	1	5004	0001	ARRAY	wo	binary filter lo – vstup 4

CANopen			Epsnet				Modbus				
I	SI	L	B	Lo	Hi	L	I	L	Typ	A	Popis
3100	05	1	31	00	05	1	5005	0001	ARRAY	wo	binary filter lo – vstup 5
3100	06	1	31	00	06	1	5006	0001	ARRAY	wo	binary filter lo – vstup 6
3100	07	1	31	00	07	1	5007	0001	ARRAY	wo	binary filter lo – vstup 7
3100	08	1	31	00	08	1	5008	0001	ARRAY	wo	binary filter lo – vstup 8
3100	09	1	31	00	09	1	5009	0001	ARRAY	wo	binary filter lo – vstup 9
3100	0a	1	31	00	0a	1	500A	0001	ARRAY	wo	binary filter lo – vstup 10
3100	0b	1	31	00	0b	1	500B	0001	ARRAY	wo	binary filter lo – vstup 11
3100	0c	1	31	00	0c	1	500C	0001	ARRAY	wo	binary filter lo – vstup 12
3100	0d	1	31	00	0d	1	500D	0001	ARRAY	wo	binary filter lo – vstup 13
3100	0e	1	31	00	0e	1	500E	0001	ARRAY	wo	binary filter lo – vstup 14
3100	0f	1	31	00	0f	1	500F	0001	ARRAY	wo	binary filter lo – vstup 15
3100	10	1	31	00	10	1	5010	0001	ARRAY	wo	binary filter lo – vstup 16

Časová konstanta filtrů binárních vstupů v ms – logická jednička na vstupu

3101	00	1	31	01	00	1	5040	0001	ARRAY	ro	binary filter hi – počet vstupů
3101	01	1	31	01	01	1	5041	0001	ARRAY	wo	binary filter hi – vstup 1
3101	02	1	31	01	02	1	5042	0001	ARRAY	wo	binary filter hi – vstup 2
3101	03	1	31	01	03	1	5043	0001	ARRAY	wo	binary filter hi – vstup 3
3101	04	1	31	01	04	1	5044	0001	ARRAY	wo	binary filter hi – vstup 4
3101	05	1	31	01	05	1	5045	0001	ARRAY	wo	binary filter hi – vstup 5
3101	06	1	31	01	06	1	5046	0001	ARRAY	wo	binary filter hi – vstup 6
3101	07	1	31	01	07	1	5047	0001	ARRAY	wo	binary filter hi – vstup 7
3101	08	1	31	01	08	1	5048	0001	ARRAY	wo	binary filter hi – vstup 8
3101	09	1	31	01	09	1	5049	0001	ARRAY	wo	binary filter hi – vstup 9
3101	0a	1	31	01	0a	1	504A	0001	ARRAY	wo	binary filter hi – vstup 10
3101	0b	1	31	01	0b	1	504B	0001	ARRAY	wo	binary filter hi – vstup 11
3101	0c	1	31	01	0c	1	504C	0001	ARRAY	wo	binary filter hi – vstup 12
3101	0d	1	31	01	0d	1	504D	0001	ARRAY	wo	binary filter hi – vstup 13
3101	0e	1	31	01	0e	1	504E	0001	ARRAY	wo	binary filter hi – vstup 14
3101	0f	1	31	01	0f	1	504F	0001	ARRAY	wo	binary filter hi – vstup 15
3101	10	1	31	01	10	1	5050	0001	ARRAY	wo	binary filter hi – vstup 16

Jednotná časová konstanta filtrů binárních vstupů – všechny vstupy a obě logické úrovně (mimo FCPU)

3102	00	1	31	02	00	1	5080	0001	VAR	wo	global binary filter

Časová konstanta filtrů binárních vstupů v ms – logická nula na vstupu (jen FCPU–02A)

3102	00	1	31	02	00	1	5080	0001	ARRAY	ro	binary filter lo – počet vstupů
3102	01	1	31	02	01	1	5081	0001	ARRAY	wo	binary filter lo – vstup 17
3102	02	1	31	02	02	1	5082	0001	ARRAY	wo	binary filter lo – vstup 18
3102	03	1	31	02	03	1	5083	0001	ARRAY	wo	binary filter lo – vstup 19
3102	04	1	31	02	04	1	5084	0001	ARRAY	wo	binary filter lo – vstup 20
3102	05	1	31	02	05	1	5085	0001	ARRAY	wo	binary filter lo – vstup 21
3102	06	1	31	02	06	1	5086	0001	ARRAY	wo	binary filter lo – vstup 22
3102	07	1	31	02	07	1	5087	0001	ARRAY	wo	binary filter lo – vstup 23
3102	08	1	31	02	08	1	5088	0001	ARRAY	wo	binary filter lo – vstup 24
3102	09	1	31	02	09	1	5089	0001	ARRAY	wo	binary filter lo – vstup 25
3102	0a	1	31	02	0a	1	508A	0001	ARRAY	wo	binary filter lo – vstup 26
3102	0b	1	31	02	0b	1	508B	0001	ARRAY	wo	binary filter lo – vstup 27
3102	0c	1	31	02	0c	1	508C	0001	ARRAY	wo	binary filter lo – vstup 28
3102	0d	1	31	02	0d	1	508D	0001	ARRAY	wo	binary filter lo – vstup 29
3102	0e	1	31	02	0e	1	508E	0001	ARRAY	wo	binary filter lo – vstup 30
3102	0f	1	31	02	0f	1	508F	0001	ARRAY	wo	binary filter lo – vstup 31
3102	10	1	31	02	10	1	5090	0001	ARRAY	wo	binary filter lo – vstup 32
3102	11	1	31	02	11	1	5091	0001	ARRAY	wo	binary filter lo – vstup 33
3102	12	1	31	02	12	1	5092	0001	ARRAY	wo	binary filter lo – vstup 34
3102	13	1	31	02	13	1	5093	0001	ARRAY	wo	binary filter lo – vstup 35
3102	14	1	31	02	14	1	5094	0001	ARRAY	wo	binary filter lo – vstup 36
3102	15	1	31	02	15	1	5095	0001	ARRAY	wo	binary filter lo – vstup 37
3102	16	1	31	02	16	1	5096	0001	ARRAY	wo	binary filter lo – vstup 38
3102	17	1	31	02	17	1	5097	0001	ARRAY	wo	binary filter lo – vstup 39
3102	18	1	31	02	18	1	5098	0001	ARRAY	wo	binary filter lo – vstup 40
3102	19	1	31	02	19	1	5099	0001	ARRAY	wo	binary filter lo – vstup 41
3102	1a	1	31	02	1a	1	509A	0001	ARRAY	wo	binary filter lo – vstup 42
3102	1b	1	31	02	1b	1	509B	0001	ARRAY	wo	binary filter lo – vstup 43
3102	1c	1	31	02	1c	1	509C	0001	ARRAY	wo	binary filter lo – vstup 44
3102	1d	1	31	02	1d	1	509D	0001	ARRAY	wo	binary filter lo – vstup 45
3102	1e	1	31	02	1e	1	509E	0001	ARRAY	wo	binary filter lo – vstup 46
3102	1f	1	31	02	1f	1	509F	0001	ARRAY	wo	binary filter lo – vstup 47
3102	20	1	31	02	20	1	50A0	0001	ARRAY	wo	binary filter lo – vstup 48
3102	21	1	31	02	21	1	50A1	0001	ARRAY	wo	binary filter lo – vstup 49
3102	22	1	31	02	22	1	50A2	0001	ARRAY	wo	binary filter lo – vstup 50
3102	23	1	31	02	23	1	50A3	0001	ARRAY	wo	binary filter lo – vstup 51
3102	24	1	31	02	24	1	50A4	0001	ARRAY	wo	binary filter lo – vstup 52
3102	25	1	31	02	25	1	50A5	0001	ARRAY	wo	binary filter lo – vstup 53
3102	26	1	31	02	26	1	50A6	0001	ARRAY	wo	binary filter lo – vstup 54
3102	27	1	31	02	27	1	50A7	0001	ARRAY	wo	binary filter lo – vstup 55

CANopen			Epsnet				Modbus				
I	SI	L	B	Lo	Hi	L	I	L	Typ	A	Popis
3102	28	1	31	02	28	1	50A8	0001	ARRAY	wo	binary filter lo – vstup 56
3102	29	1	31	02	29	1	50A9	0001	ARRAY	wo	binary filter lo – vstup 57
3102	2a	1	31	02	2a	1	50AA	0001	ARRAY	wo	binary filter lo – vstup 58
3102	2b	1	31	02	2b	1	50AB	0001	ARRAY	wo	binary filter lo – vstup 59
3102	2c	1	31	02	2c	1	50AC	0001	ARRAY	wo	binary filter lo – vstup 60
3102	2d	1	31	02	2d	1	50AD	0001	ARRAY	wo	binary filter lo – vstup 61
3102	2e	1	31	02	2e	1	50AE	0001	ARRAY	wo	binary filter lo – vstup 62
3102	2f	1	31	02	2f	1	50AF	0001	ARRAY	wo	binary filter lo – vstup 63
3102	30	1	31	02	30	1	50B0	0001	ARRAY	wo	binary filter lo – vstup 64

Časová konstanta filtrů binárních vstupů v ms – logická jednička na vstupu (jen FCPU–02A)

3103	00	1	31	03	00	1	50C0	0001	ARRAY	ro	binary filter hi – počet vstupů
3103	01	1	31	03	01	1	50C1	0001	ARRAY	wo	binary filter hi – vstup 17
3103	02	1	31	03	02	1	50C2	0001	ARRAY	wo	binary filter hi – vstup 18
3103	03	1	31	03	03	1	50C3	0001	ARRAY	wo	binary filter hi – vstup 19
3103	04	1	31	03	04	1	50C4	0001	ARRAY	wo	binary filter hi – vstup 20
3103	05	1	31	03	05	1	50C5	0001	ARRAY	wo	binary filter hi – vstup 21
3103	06	1	31	03	06	1	50C6	0001	ARRAY	wo	binary filter hi – vstup 22
3103	07	1	31	03	07	1	50C7	0001	ARRAY	wo	binary filter hi – vstup 23
3103	08	1	31	03	08	1	50C8	0001	ARRAY	wo	binary filter hi – vstup 24
3103	09	1	31	03	09	1	50C9	0001	ARRAY	wo	binary filter hi – vstup 25
3103	0a	1	31	03	0a	1	50CA	0001	ARRAY	wo	binary filter hi – vstup 26
3103	0b	1	31	03	0b	1	50CB	0001	ARRAY	wo	binary filter hi – vstup 27
3103	0c	1	31	03	0c	1	50CC	0001	ARRAY	wo	binary filter hi – vstup 28
3103	0d	1	31	03	0d	1	50CD	0001	ARRAY	wo	binary filter hi – vstup 29
3103	0e	1	31	03	0e	1	50CE	0001	ARRAY	wo	binary filter hi – vstup 30
3103	0f	1	31	03	0f	1	50CF	0001	ARRAY	wo	binary filter hi – vstup 31
3103	10	1	31	03	10	1	50D0	0001	ARRAY	wo	binary filter hi – vstup 32
3103	11	1	31	03	11	1	50D1	0001	ARRAY	wo	binary filter hi – vstup 33
3103	12	1	31	03	12	1	50D2	0001	ARRAY	wo	binary filter hi – vstup 34
3103	13	1	31	03	13	1	50D3	0001	ARRAY	wo	binary filter hi – vstup 35
3103	14	1	31	03	14	1	50D4	0001	ARRAY	wo	binary filter hi – vstup 36
3103	15	1	31	03	15	1	50D5	0001	ARRAY	wo	binary filter hi – vstup 37
3103	16	1	31	03	16	1	50D6	0001	ARRAY	wo	binary filter hi – vstup 38
3103	17	1	31	03	17	1	50D7	0001	ARRAY	wo	binary filter hi – vstup 39
3103	18	1	31	03	18	1	50D8	0001	ARRAY	wo	binary filter hi – vstup 40
3103	19	1	31	03	19	1	50D9	0001	ARRAY	wo	binary filter hi – vstup 41
3103	1a	1	31	03	1a	1	50DA	0001	ARRAY	wo	binary filter hi – vstup 42
3103	1b	1	31	03	1b	1	50DB	0001	ARRAY	wo	binary filter hi – vstup 43
3103	1c	1	31	03	1c	1	50DC	0001	ARRAY	wo	binary filter hi – vstup 44
3103	1d	1	31	03	1d	1	50DD	0001	ARRAY	wo	binary filter hi – vstup 45
3103	1e	1	31	03	1e	1	50DE	0001	ARRAY	wo	binary filter hi – vstup 46
3103	1f	1	31	03	1f	1	50DF	0001	ARRAY	wo	binary filter hi – vstup 47
3103	20	1	31	03	20	1	50E0	0001	ARRAY	wo	binary filter hi – vstup 48
3103	21	1	31	03	21	1	50E1	0001	ARRAY	wo	binary filter hi – vstup 49
3103	22	1	31	03	22	1	50E2	0001	ARRAY	wo	binary filter hi – vstup 50
3103	23	1	31	03	23	1	50E3	0001	ARRAY	wo	binary filter hi – vstup 51
3103	24	1	31	03	24	1	50E4	0001	ARRAY	wo	binary filter hi – vstup 52
3103	25	1	31	03	25	1	50E5	0001	ARRAY	wo	binary filter hi – vstup 53
3103	26	1	31	03	26	1	50E6	0001	ARRAY	wo	binary filter hi – vstup 54
3103	27	1	31	03	27	1	50E7	0001	ARRAY	wo	binary filter hi – vstup 55
3103	28	1	31	03	28	1	50E8	0001	ARRAY	wo	binary filter hi – vstup 56
3103	29	1	31	03	29	1	50E9	0001	ARRAY	wo	binary filter hi – vstup 57
3103	2a	1	31	03	2a	1	50EA	0001	ARRAY	wo	binary filter hi – vstup 58
3103	2b	1	31	03	2b	1	50EB	0001	ARRAY	wo	binary filter hi – vstup 59
3103	2c	1	31	03	2c	1	50EC	0001	ARRAY	wo	binary filter hi – vstup 60
3103	2d	1	31	03	2d	1	50ED	0001	ARRAY	wo	binary filter hi – vstup 61
3103	2e	1	31	03	2e	1	50EE	0001	ARRAY	wo	binary filter hi – vstup 62
3103	2f	1	31	03	2f	1	50EF	0001	ARRAY	wo	binary filter hi – vstup 63
3103	30	1	31	03	30	1	50F0	0001	ARRAY	wo	binary filter hi – vstup 64

Časová konstanta filtrů analogových vstupů v ms (dolní propust 1. řádu)

3110	00	1	31	10	00	1	5200	0001	ARRAY	ro	analog filter – počet vstupů
3110	01	2	31	10	01	2	5201	0001	ARRAY	wo	analog filter – vstup 1
3110	02	2	31	10	02	2	5202	0001	ARRAY	wo	analog filter – vstup 2
3110	03	2	31	10	03	2	5203	0001	ARRAY	wo	analog filter – vstup 3
3110	04	2	31	10	04	2	5204	0001	ARRAY	wo	analog filter – vstup 4
3110	05	2	31	10	05	2	5205	0001	ARRAY	wo	analog filter – vstup 5
3110	06	2	31	10	06	2	5206	0001	ARRAY	wo	analog filter – vstup 6
3110	07	2	31	10	07	2	5207	0001	ARRAY	wo	analog filter – vstup 7
3110	08	2	31	10	08	2	5208	0001	ARRAY	wo	analog filter – vstup 8
3110	09	2	31	10	09	2	5209	0001	ARRAY	wo	analog filter – vstup 9
3110	0a	2	31	10	0a	2	520A	0001	ARRAY	wo	analog filter – vstup 10
3110	0b	2	31	10	0b	2	520B	0001	ARRAY	wo	analog filter – vstup 11
3110	0c	2	31	10	0c	2	520C	0001	ARRAY	wo	analog filter – vstup 12

CANopen			Epsnet				Modbus				
I	SI	L	B	Lo	Hi	L	I	L	Typ	A	Popis
Jednotná časová konstanta filtrů analogových vstupů – všechny vstupy											
3111	00	2	31	11	00	1	5240	0001	VAR	wo	global analog filter
Uživatelsky definované stavy binárních výstupů v režimu Guard Error											
3200	00	1	32	00	00	1	6000	0001	ARRAY	ro	bin def – počet osmic výstupních dat
3200	01	1	32	00	01	1	6001	0001	ARRAY	wo	bin def – 1. osmice výstupních dat
3200	02	1	32	00	02	1	6002	0001	ARRAY	wo	bin def – 2. osmice výstupních dat
3200	03	1	32	00	03	1	6003	0001	ARRAY	wo	bin def – 3. osmice výstupních dat
3200	04	1	32	00	04	1	6004	0001	ARRAY	wo	bin def – 4. osmice výstupních dat
3200	05	1	32	00	05	1	6005	0001	ARRAY	wo	bin def – 5. osmice výstupních dat
3200	06	1	32	00	06	1	6006	0001	ARRAY	wo	bin def – 6. osmice výstupních dat
3200	07	1	32	00	07	1	6007	0001	ARRAY	wo	bin def – 7. osmice výstupních dat
3200	08	1	32	00	08	1	6008	0001	ARRAY	wo	bin def – 8. osmice výstupních dat
Uživatelsky definované hodnoty analogových výstupů v režimu Guard Error											
3201	00	1	32	01	00	1	6040	0001	ARRAY	ro	analog def – počet analogových výstupů
3201	01	1	32	01	01	1	6041	0001	ARRAY	wo	analog def – analogový výstup 1
3201	02	1	32	01	02	1	6042	0001	ARRAY	wo	analog def – analogový výstup 2
3201	03	1	32	01	03	1	6043	0001	ARRAY	wo	analog def – analogový výstup 3
3201	04	1	32	01	04	1	6044	0001	ARRAY	wo	analog def – analogový výstup 4
3201	05	1	32	01	05	1	6045	0001	ARRAY	wo	analog def – analogový výstup 5
3201	06	1	32	01	06	1	6046	0001	ARRAY	wo	analog def – analogový výstup 6
Uživatelsky definované znaky terminálu CKDM-11/12											
3202	00	40	32	02	00	40	6080	0040	VAR	rw	řetězec 64 znaků definujících tvar 8 uživatelských znaků displeje
Konstanty linearizačního polynomu $x=ax^3+bx^2+cx+d$ analogových vstupů (typ float)											
3300	00	1	33	00	00	1	7000	0001	ARRAY	ro	lin 0 – konstanta „d“ linearizačního polynomu – počet vstupů
3300	01	4	33	00	01	4	7001	0002	ARRAY	wo	lin 0 – vstup 1
3300	02	4	33	00	02	4	7002	0002	ARRAY	wo	lin 0 – vstup 2
3300	03	4	33	00	03	4	7003	0002	ARRAY	wo	lin 0 – vstup 3
3300	04	4	33	00	04	4	7004	0002	ARRAY	wo	lin 0 – vstup 4
3300	05	4	33	00	05	4	7005	0002	ARRAY	wo	lin 0 – vstup 5
3300	06	4	33	00	06	4	7006	0002	ARRAY	wo	lin 0 – vstup 6
3300	07	4	33	00	07	4	7007	0002	ARRAY	wo	lin 0 – vstup 7
3300	08	4	33	00	08	4	7008	0002	ARRAY	wo	lin 0 – vstup 8
3300	09	4	33	00	09	4	7009	0002	ARRAY	wo	lin 0 – vstup 9
3300	0a	4	33	00	0a	4	700A	0002	ARRAY	wo	lin 0 – vstup 10
3300	0b	4	33	00	0b	4	700B	0002	ARRAY	wo	lin 0 – vstup 11
3300	0c	4	33	00	0c	4	700C	0002	ARRAY	wo	lin 0 – vstup 12
3301	00	1	33	01	00	1	7040	0001	ARRAY	ro	lin 1 – konstanta „c“ linearizačního polynomu – počet vstupů
3301	01	4	33	01	01	4	7041	0002	ARRAY	wo	lin 1 – vstup 1
3301	02	4	33	01	02	4	7042	0002	ARRAY	wo	lin 1 – vstup 2
3301	03	4	33	01	03	4	7043	0002	ARRAY	wo	lin 1 – vstup 3
3301	04	4	33	01	04	4	7044	0002	ARRAY	wo	lin 1 – vstup 4
3301	05	4	33	01	05	4	7045	0002	ARRAY	wo	lin 1 – vstup 5
3301	06	4	33	01	06	4	7046	0002	ARRAY	wo	lin 1 – vstup 6
3301	07	4	33	01	07	4	7047	0002	ARRAY	wo	lin 1 – vstup 7
3301	08	4	33	01	08	4	7048	0002	ARRAY	wo	lin 1 – vstup 8
3301	09	4	33	01	09	4	7049	0002	ARRAY	wo	lin 1 – vstup 9
3301	0a	4	33	01	0a	4	704A	0002	ARRAY	wo	lin 1 – vstup 10
3301	0b	4	33	01	0b	4	704B	0002	ARRAY	wo	lin 1 – vstup 11
3301	0c	4	33	01	0c	4	704C	0002	ARRAY	wo	lin 1 – vstup 12
3302	00	1	33	02	00	1	7080	0001	ARRAY	ro	lin 2 – konstanta „b“ linearizačního polynomu – počet vstupů
3302	01	4	33	02	01	4	7081	0002	ARRAY	wo	lin 2 – vstup 1
3302	02	4	33	02	02	4	7082	0002	ARRAY	wo	lin 2 – vstup 2
3302	03	4	33	02	03	4	7083	0002	ARRAY	wo	lin 2 – vstup 3
3302	04	4	33	02	04	4	7084	0002	ARRAY	wo	lin 2 – vstup 4
3302	05	4	33	02	05	4	7085	0002	ARRAY	wo	lin 2 – vstup 5
3302	06	4	33	02	06	4	7086	0002	ARRAY	wo	lin 2 – vstup 6
3302	07	4	33	02	07	4	7087	0002	ARRAY	wo	lin 2 – vstup 7
3302	08	4	33	02	08	4	7088	0002	ARRAY	wo	lin 2 – vstup 8
3302	09	4	33	02	09	4	7089	0002	ARRAY	wo	lin 2 – vstup 9
3302	0a	4	33	02	0a	4	708A	0002	ARRAY	wo	lin 2 – vstup 10
3302	0b	4	33	02	0b	4	708B	0002	ARRAY	wo	lin 2 – vstup 11
3302	0c	4	33	02	0c	4	708C	0002	ARRAY	wo	lin 2 – vstup 12
3303	00	1	33	03	00	1	70C0	0001	ARRAY	ro	lin 3 – konstanta „a“ linearizačního polynomu – počet vstupů
3303	01	4	33	03	01	4	70C1	0002	ARRAY	wo	lin 3 – vstup 1
3303	02	4	33	03	02	4	70C2	0002	ARRAY	wo	lin 3 – vstup 2
3303	03	4	33	03	03	4	70C3	0002	ARRAY	wo	lin 3 – vstup 3
3303	04	4	33	03	04	4	70C4	0002	ARRAY	wo	lin 3 – vstup 4
3303	05	4	33	03	05	4	70C5	0002	ARRAY	wo	lin 3 – vstup 5
3303	06	4	33	03	06	4	70C6	0002	ARRAY	wo	lin 3 – vstup 6
3303	07	4	33	03	07	4	70C7	0002	ARRAY	wo	lin 3 – vstup 7
3303	08	4	33	03	08	4	70C8	0002	ARRAY	wo	lin 3 – vstup 8
3303	09	4	33	03	09	4	70C9	0002	ARRAY	wo	lin 3 – vstup 9

CANopen			Epsnet				Modbus				
I	SI	L	B	Lo	Hi	L	I	L	Typ	A	Popis
3303	0a	4	33	03	0a	4	70CA	0002	ARRAY	wo	lin 3 – vstup 10
3303	0b	4	33	03	0b	4	70CB	0002	ARRAY	wo	lin 3 – vstup 11
3303	0c	4	33	03	0c	4	70CC	0002	ARRAY	wo	lin 3 – vstup 12

Rychlý čítač a měřič frekvence – do 2 MHz

4101	00	2	41	01	00	2	3040	0001	VAR	ro	fcounter
4121	00	2	41	21	00	2	3440	0001	VAR	ro	ffreguence

Čítače impulsů na binárních vstupech

4100	00	1	41	00	00	1	3000	0001	ARRAY	ro	counters – počet vstupů
4100	01	2	41	00	01	2	3001	0001	ARRAY	ro	counters – vstup 1
4100	02	2	41	00	02	2	3002	0001	ARRAY	ro	counters – vstup 2
4100	03	2	41	00	03	2	3003	0001	ARRAY	ro	counters – vstup 3
4100	04	2	41	00	04	2	3004	0001	ARRAY	ro	counters – vstup 4
4100	05	2	41	00	05	2	3005	0001	ARRAY	ro	counters – vstup 5
4100	06	2	41	00	06	2	3006	0001	ARRAY	ro	counters – vstup 6
4100	07	2	41	00	07	2	3007	0001	ARRAY	ro	counters – vstup 7
4100	08	2	41	00	08	2	3008	0001	ARRAY	ro	counters – vstup 8
4100	09	2	41	00	09	2	3009	0001	ARRAY	ro	counters – vstup 9
4100	0a	2	41	00	0a	2	300a	0001	ARRAY	ro	counters – vstup 10
4100	0b	2	41	00	0b	2	300b	0001	ARRAY	ro	counters – vstup 11
4100	0c	2	41	00	0c	2	300c	0001	ARRAY	ro	counters – vstup 12
4100	0d	2	41	00	0d	2	300d	0001	ARRAY	ro	counters – vstup 13
4100	0e	2	41	00	0e	2	300e	0001	ARRAY	ro	counters – vstup 14
4100	0f	2	41	00	0f	2	300f	0001	ARRAY	ro	counters – vstup 15
4100	10	2	41	00	10	2	3010	0001	ARRAY	ro	counters – vstup 16

Čítače impulsů na binárních vstupech jednotky CAIO-12

4100	00	1	41	00	00	1	3000	0001	ARRAY	ro	counters – počet vstupů
4100	01	2	41	00	01	2	3001	0001	ARRAY	ro	counters – vstup 1
4100	02	2	41	00	02	2	3002	0001	ARRAY	ro	counters – vstup 2
4100	03	2	41	00	03	2	3003	0001	ARRAY	ro	counters – vstup 3
4100	04	2	41	00	04	2	3004	0001	ARRAY	ro	counters – vstup 4
4100	05	2	41	00	05	2	3005	0001	ARRAY	ro	counters – vstup 5
4100	06	2	41	00	06	2	3006	0001	ARRAY	ro	counters – vstup 6
4100	07	2	41	00	07	2	3007	0001	ARRAY	ro	counters – vstup 7
4100	08	2	41	00	08	2	3008	0001	ARRAY	ro	counters – vstup 8
4100	09	2	41	00	09	2	3009	0001	ARRAY	ro	counters – vstup 9
4100	0a	2	41	00	0a	2	300a	0001	ARRAY	ro	counters – vstup 10
4100	0b	2	41	00	0b	2	300b	0001	ARRAY	ro	counters – vstup 11
4100	0c	2	41	00	0c	2	300c	0001	ARRAY	ro	counters – vstup 12

Měřiče periody impulsů na binárních vstupech

4110	00	1	41	10	00	1	3200	0001	ARRAY	ro	periods – počet vstupů
4110	01	2	41	10	01	2	3201	0001	ARRAY	ro	periods – vstup 1
4110	02	2	41	10	02	2	3202	0001	ARRAY	ro	periods – vstup 2
4110	03	2	41	10	03	2	3203	0001	ARRAY	ro	periods – vstup 3
4110	04	2	41	10	04	2	3204	0001	ARRAY	ro	periods – vstup 4
4110	05	2	41	10	05	2	3205	0001	ARRAY	ro	periods – vstup 5
4110	06	2	41	10	06	2	3206	0001	ARRAY	ro	periods – vstup 6
4110	07	2	41	10	07	2	3207	0001	ARRAY	ro	periods – vstup 7
4110	08	2	41	10	08	2	3208	0001	ARRAY	ro	periods – vstup 8
4110	09	2	41	10	09	2	3209	0001	ARRAY	ro	periods – vstup 9
4110	0a	2	41	10	0a	2	320a	0001	ARRAY	ro	periods – vstup 10
4110	0b	2	41	10	0b	2	320b	0001	ARRAY	ro	periods – vstup 11
4110	0c	2	41	10	0c	2	320c	0001	ARRAY	ro	periods – vstup 12
4110	0d	2	41	10	0d	2	320d	0001	ARRAY	ro	periods – vstup 13
4110	0e	2	41	10	0e	2	320e	0001	ARRAY	ro	periods – vstup 14
4110	0f	2	41	10	0f	2	320f	0001	ARRAY	ro	periods – vstup 15
4110	10	2	41	10	10	2	3210	0001	ARRAY	ro	periods – vstup 16

Měřiče frekvence signálů na binárních vstupech

4120	00	1	41	20	00	1	3400	0001	ARRAY	ro	frequence – počet vstupů
4120	01	2	41	20	01	2	3401	0001	ARRAY	ro	frequence – vstup 1
4120	02	2	41	20	02	2	3402	0001	ARRAY	ro	frequence – vstup 2
4120	03	2	41	20	03	2	3403	0001	ARRAY	ro	frequence – vstup 3
4120	04	2	41	20	04	2	3404	0001	ARRAY	ro	frequence – vstup 4
4120	05	2	41	20	05	2	3405	0001	ARRAY	ro	frequence – vstup 5
4120	06	2	41	20	06	2	3406	0001	ARRAY	ro	frequence – vstup 6
4120	07	2	41	20	07	2	3407	0001	ARRAY	ro	frequence – vstup 7
4120	08	2	41	20	08	2	3408	0001	ARRAY	ro	frequence – vstup 8
4120	09	2	41	20	09	2	3409	0001	ARRAY	ro	frequence – vstup 9
4120	0a	2	41	20	0a	2	340a	0001	ARRAY	ro	frequence – vstup 10
4120	0b	2	41	20	0b	2	340b	0001	ARRAY	ro	frequence – vstup 11
4120	0c	2	41	20	0c	2	340c	0001	ARRAY	ro	frequence – vstup 12
4120	0d	2	41	20	0d	2	340d	0001	ARRAY	ro	frequence – vstup 13
4120	0e	2	41	20	0e	2	340e	0001	ARRAY	ro	frequence – vstup 14

CANopen			Epsnet				Modbus				
I	SI	L	B	Lo	Hi	L	I	L	Typ	A	Popis
4120	0f	2	41	20	0f	2	340f	0001	ARRAY	ro	frequence – vstup 15
4120	10	2	41	20	10	2	3410	0001	ARRAY	ro	frequence – vstup 16

Binární vstupy

6000	00	1	60	00	00	1	8000	0001	ARRAY	ro	bin in – počet osmic vstupních dat
6000	01	1	60	00	01	1	8001	0001	ARRAY	ro	bin in – 1. osmice vstupních dat
6000	02	1	60	00	02	1	8002	0001	ARRAY	ro	bin in – 2. osmice vstupních dat
6000	03	1	60	00	03	1	8003	0001	ARRAY	ro	bin in – 3. osmice vstupních dat
6000	04	1	60	00	04	1	8004	0001	ARRAY	ro	bin in – 4. osmice vstupních dat
6000	05	1	60	00	05	1	8005	0001	ARRAY	ro	bin in – 5. osmice vstupních dat
6000	06	1	60	00	06	1	8006	0001	ARRAY	ro	bin in – 6. osmice vstupních dat
6000	07	1	60	00	07	1	8007	0001	ARRAY	ro	bin in – 7. osmice vstupních dat
6000	08	1	60	00	08	1	8008	0001	ARRAY	ro	bin in – 8. osmice vstupních dat

Čtení klávesy z terminálu

6000	00	2	60	00	00	2	8000	0001	VAR	ro	Read Key Code

Binární vstupy terminálu CKDM-12

6001	00	1	60	01	00	1	8040	0001	ARRAY	ro	bin in – počet osmic vstupních dat u terminálu CKDM-12
6001	01	1	60	01	01	1	8041	0001	ARRAY	ro	bin in – 1. osmice vstupních dat
6001	02	1	60	01	02	1	8042	0001	ARRAY	ro	bin in – 2. osmice vstupních dat

Binární vstupy jednotky CAIO-12

6120	00	1	61	20	00	1	9400	0001	ARRAY	ro	bin in – počet položek vstupních dat (délky 32 bitů) jednotky CAIO-12
6120	01	4	61	20	01	4	9401	0002	ARRAY	ro	bin in – vstupní data jednotky CAIO-12

Binární výstupy

6200	00	1	62	00	00	1	a000	0001	ARRAY	ro	bin out – počet osmic výstupních dat
6200	01	1	62	00	01	1	a001	0001	ARRAY	wo	bin out – 1. osmice výstupních dat
6200	02	1	62	00	02	1	a002	0001	ARRAY	wo	bin out – 2. osmice výstupních dat
6200	03	1	62	00	03	1	a003	0001	ARRAY	wo	bin out – 3. osmice výstupních dat
6200	04	1	62	00	04	1	a004	0001	ARRAY	wo	bin out – 4. osmice výstupních dat
6200	05	1	62	00	05	1	a005	0001	ARRAY	wo	bin out – 5. osmice výstupních dat
6200	06	1	62	00	06	1	a006	0001	ARRAY	wo	bin out – 6. osmice výstupních dat
6200	07	1	62	00	07	1	a007	0001	ARRAY	wo	bin out – 7. osmice výstupních dat
6200	08	1	62	00	08	1	a008	0001	ARRAY	wo	bin out – 8. osmice výstupních dat

Binární výstupy jednotky CAIO-12

6200	00	1	62	00	00	1	a000	0001	VAR	ro	bin out – počet osmic výstupních dat
6200	01	1	62	00	00	1	a001	0001	VAR	ro	bin out – 1. osmice výstupních dat
6200	02	1	62	00	00	1	a002	0001	VAR	ro	bin out – 2. osmice výstupních dat

Ovládání indikačních LED terminálu CKDM-11/12

6200	00	2	62	00	00	2	a000	0001	VAR	ro	Write Lamp Code

Binární výstupy terminálu CKDM-12

6201	00	1	62	01	00	1	a040	0001	ARRAY	ro	bin out – počet osmic výstupních dat
6201	01	1	62	01	01	1	a041	0001	ARRAY	wo	bin out – 1. osmice výstupních dat
6201	02	1	62	01	02	1	a042	0001	ARRAY	wo	bin out – 2. osmice výstupních dat

Nastavení podsvětlení displeje terminálu CKDM-11/12

6202	00	1	62	02	00	1	a080	0001	RECORD	ro	backlight – počet položek
6202	01	1	62	02	01	1	a081	0001	RECORD	wo	zapnutí/vypnutí podsvitu
6202	02	1	62	02	02	1	a082	0001	RECORD	wo	intenzita podsvitu

Nastavení hlasitosti u terminálu CKDM-11/12

6204	00	1	62	04	00	1	a100	0001	VAR	wo	nastavení hlasitosti (ve 4 krocích)

Textový monitor terminálu CKDM-11/12

6210	00	1	62	10	00	1	a200	0001	RECORD	ro	text monitor – počet položek
6210	01	1	62	10	01	1	a201	0001	RECORD	ro	atributy zobrazení
6210	02	50	62	10	02	50	a202	0050	RECORD	rw	řetězec 80 znaků displeje (obraz displeje)

Nastavení kurzor displeje terminálu CKDM-11/12

6211	00	1	62	11	00	1	a240	0001	RECORD	ro	text cursor – počet položek
6211	01	1	62	11	01	1	a241	0001	RECORD	rw	atributy kurzoru
6211	02	1	62	11	02	1	a242	0001	RECORD	rw	pozice kurzoru

Analogové vstupy terminálu CKDM-12

6400	00	1	64	00	00	1	c000	0001	ARRAY	ro	analog in – počet vstupů
6400	01	2	64	00	01	2	c001	0001	ARRAY	ro	analog in – vstup 1
6400	02	2	64	00	02	2	c002	0001	ARRAY	ro	analog in – vstup 2
6400	03	2	64	00	03	2	c003	0001	ARRAY	ro	analog in – vstup 3
6400	04	2	64	00	04	2	c004	0001	ARRAY	ro	analog in – vstup 4

Analogové vstupy

6401	00	1	64	01	00	1	c040	0001	ARRAY	ro	analog in – počet vstupů
6401	01	2	64	01	01	2	c041	0001	ARRAY	ro	analog in – vstup 1
6401	02	2	64	01	02	2	c042	0001	ARRAY	ro	analog in – vstup 2
6401	03	2	64	01	03	2	c043	0001	ARRAY	ro	analog in – vstup 3
6401	04	2	64	01	04	2	c044	0001	ARRAY	ro	analog in – vstup 4
6401	05	2	64	01	05	2	c045	0001	ARRAY	ro	analog in – vstup 5

CANopen			Epsnet				Modbus				
I	SI	L	B	Lo	Hi	L	I	L	Typ	A	Popis
6401	06	2	64	01	06	2	c046	0001	ARRAY	ro	analog in – vstup 6
6401	07	2	64	01	07	2	c047	0001	ARRAY	ro	analog in – vstup 7
6401	08	2	64	01	08	2	c048	0001	ARRAY	ro	analog in – vstup 8
6401	09	2	64	01	09	2	c049	0001	ARRAY	ro	analog in – vstup 9
6401	0a	2	64	01	0a	2	c04a	0001	ARRAY	ro	analog in – vstup 10
6401	0b	2	64	01	0b	2	c04b	0001	ARRAY	ro	analog in – vstup 11
6401	0c	2	64	01	0c	2	c04c	0001	ARRAY	ro	analog in – vstup 12
Analogové výstupy PWM											
6410	00	1	64	10	00	1	c200	0001	ARRAY	ro	analog out – počet výstupů
6410	01	1	64	10	01	1	c201	0001	ARRAY	wo	analog out – výstup 1
6410	02	1	64	10	02	1	c202	0001	ARRAY	wo	analog out – výstup 2
6410	03	1	64	10	03	1	c203	0001	ARRAY	wo	analog out – výstup 3
6410	04	1	64	10	04	1	c204	0001	ARRAY	wo	analog out – výstup 4
6410	05	1	64	10	05	1	c205	0001	ARRAY	wo	analog out – výstup 5
6410	06	1	64	10	06	1	c206	0001	ARRAY	wo	analog out – výstup 6
Analogové výstupy DA 16-bit (jen CAIO-12)											
6411	00	1	64	11	00	1	c240	0001	ARRAY	ro	analog out – počet výstupů
6411	01	2	64	11	01	2	c241	0001	ARRAY	wo	analog out – výstup 1
6411	02	2	64	11	02	2	c242	0001	ARRAY	wo	analog out – výstup 2
6411	03	2	64	11	03	2	c243	0001	ARRAY	wo	analog out – výstup 3
6411	04	2	64	11	04	2	c244	0001	ARRAY	wo	analog out – výstup 4
6411	05	2	64	11	05	2	c245	0001	ARRAY	wo	analog out – výstup 5
6411	06	2	64	11	06	2	c246	0001	ARRAY	wo	analog out – výstup 6
6411	07	2	64	11	07	2	c247	0001	ARRAY	wo	analog out – výstup 7
6411	08	2	64	11	08	2	c248	0001	ARRAY	wo	analog out – výstup 8
6411	09	2	64	11	09	2	c249	0001	ARRAY	wo	analog out – výstup 9
6411	0a	2	64	11	0a	2	c24a	0001	ARRAY	wo	analog out – výstup 10
6411	0b	2	64	11	0b	2	c24b	0001	ARRAY	wo	analog out – výstup 11
6411	0c	2	64	11	0c	2	c24c	0001	ARRAY	wo	analog out – výstup 12

5. KOMUNIKAČNÍ PROTOKOL MODBUS

Jako přenosové médium se používá běžná asynchronní sériová linka s rozhraním RS485. Pro dvoubodové spojení (1 master + 1 slave) je možné použít i jiné rozhraní (RS232, proudová smyčka). Způsob přenosu je poloduplexní, vysílat může vždy pouze jeden účastník komunikace. Pro přenos se používá formát s 8 datovými bity, obvykle sudou paritou (even) a 1 stop bitem. Rychlost přenosu závisí na použitém vedení a musí být stejná pro všechny stanice ve společné síti. Délka telegramu je proměnná, maximální povolená délka včetně rámce je 255 byte.

5.1. Struktura protokolu a typy zpráv

ModBus je komunikační protokol řídicích systémů firmy Modicon, které mají prostor komunikačních dat rozdělen do několika datových oblastí. Každá oblast obsahuje šestnáctibitové registry. Periferní jednotky PL2 mají zveřejněny 16 registrů. V nich jsou data odpovídající datovým objektům PDO (4× 8 bytů procesních dat protokolu CANopen – viz. odstavec 3.3.1. na straně 21).

Telegramy mohou být přenášeny v jednom ze dvou módů – ASCII nebo RTU. Jednotky PL2 používají mód RTU, popsán bude jen tento mód.

5.1.1. Formát telegramů

Komunikace vždy probíhá na principu výzva/odpověď. Každá stanice musí mít svoji unikátní adresu. V telegramech typu „výzva“ je vždy uvedena adresa příjemce, v telegramech typu „odpověď“ je vždy uvedena adresa odesílatele. Výzvu zasílá pouze stanice master. Zpracování výzvy provádí pouze stanice, jejíž adresa souhlasí s adresou příjemce uvedené ve zprávě. Pokud výzva vyžaduje odpověď, odesílá ji přijímací stanice ihned po zpracování výzvy. Tím je ukončena jedna komunikační relace.

Při vysílání výzvy či odpovědi nesmí být mezi jednotlivými byty prodleva delší než 1,5-násobek doby vysílání jednoho znaku. Před začátkem další komunikační relace (tj. další výzvy) musí být linka v klidovém stavu (nesmí být vysílán žádný znak) po dobu alespoň 3,5-násobku doby vysílání jednoho znaku. Tuto dobu detekují všechny stanice a interpretují ji jako ukončení předchozí komunikační relace. To umožňuje zasynchronizování všech přijímačů při rozpadu komunikační relace (chyba, rušení, ztráta spojení, zhroucení stanice ap.).

Stanice master kontroluje také dobu reakce stanice slave, tj. dobu mezi ukončením přenosu výzvy a začátkem přenosu odpovědi. Maximální doba reakce (timeout) musí respektovat dopravní zpoždění použité přenosové cesty, např. při použití radiomodemu bude dopravní zpoždění podstatně větší než při použití pevného vedení.

Složení telegramu typu **výzva** vypadá následovně:

Start	T1–T2–T3–T4	klid na lince min. 3,5 znaku
Address	1 byte	adresa **přijímače (slave)**
Function	1 byte	kód operace
data	n byte	přenášená data
CRC Check	2 byte	zabezpečení CRC
End	T1–T2–T3–T4	klid na lince min. 3,5 znaku

Složení telegramu **odpověď** je velmi podobné (jednotka slave zopakuje položky "adresa" a "kód operace"):

Start	T1–T2–T3–T4	klid na lince min. 3,5 znaku
Address	1 byte	adresa **vysílače (slave)**
Function	1 byte	kód operace
data	n byte	přenášená data
CRC Check	2 byte	zabezpečení CRC
End	T1–T2–T3–T4	klid na lince min. 3,5 znaku

5.1.2. Typy zpráv

Při komunikaciJednotky PL2 zpracovávají následující čtyři typy zpráv:

Kód	Název	Charakteristika
03	Read Holding Registers	čtení obsahu záchytných registrů
04	Read Input Registers	čtení vstupů
16	Preset Multiple Registers	nastavení hodnot v posloupnosti registrů
17	Report Slave ID	vrací informace o typu a stavu jednotky na dané adrese

Periferní jednotky PL2 na kód zprávy 03 a 04 vrací stejné hodnoty (stejnou oblast registrů).

Čtení obsahu záchytných registrů

Funkce vrací obsah 16 registrů od offsetu 0. V registrech je obsah všech čtyř PDO s defaultně napamovanými objekty (viz. odstavec 3.3.1. na straně 21). Příklad zprávy (výzva) pro jednotku s adresou 1 je v následující tabulce:

Byte	Hodnota	Význam
00	0x01	adresa příjemce (slave)
01	0x03	kód operace
02	0x00	offset HI
03	0x00	offset LO
04	0x00	počet položek HI
05	0x10	počet položek LO
06	0x44	CRC LO
07	0x06	CRC HI

Bude-li na této adrese např. jednotka SBIO-xx, bude odpověď vypadat následovně (budou-li aktivní vstupy X1 a X3):

Byte	Hodnota	Význam			
00	0x01	adresa odesílatele (slave)			
01	0x03	kód operace			
02	0x00	počet odesílaných bytů			
03	0x00	Registr 0	PDO1	byte 1	*vstupy X8 až X15*
04	0x0A			byte 0	*vstupy X0 až X7*
05	0x00	Registr 1		byte 3	
06	0x00			byte 2	
07	0x00	Registr 2		byte 5	
08	0x00			byte 4	
09	0x00	Registr 3		byte 7	
10	0x00			byte 6	
11	0x00	Registr 4	PDO2	byte 1	
12	0x00			byte 0	
13	0x00	Registr 5		byte 3	
14	0x00			byte 2	
15	0x00	Registr 6		byte 5	
16	0x00			byte 4	
17	0x00	Registr 7		byte 7	
18	0x00			byte 6	
19	0x00	Registr 8	PDO3	byte 1	
20	0x00			byte 0	
21	0x00	Registr 9		byte 3	
22	0x00			byte 2	
23	0x00	Registr 10		byte 5	
24	0x00			byte 4	
25	0x00	Registr 11		byte 7	
26	0x00			byte 6	
27	0x00	Registr 12	PDO4	byte 1	
28	0x00			byte 0	
29	0x00	Registr 13		byte 3	
30	0x00			byte 2	
31	0x00	Registr 14		byte 5	
32	0x00			byte 4	
33	0x00	Registr 15		byte 7	
34	0x00			byte 6	

Byte	Hodnota	Význam
35	0x32	CRC LO
36	0x7F	CRC HI

Čtení obsahu vstupních registrů

Funkce vrací obsah 16 registrů od offsetu 0. V registrech je obsah všech čtyř PDO s defaultně napamovanými objekty (viz. odstavec 3.3.1. na straně 21). Příklad zprávy (výzva) je v následující tabulce:

Byte	Hodnota	Význam
00	0x01	adresa příjemce (slave)
01	0x04	kód operace
02	0x00	offset HI
03	0x00	offset LO
04	0x00	počet položek HI
05	0x10	počet položek LO
06	0xF1	CRC LO
07	0xC6	CRC HI

Bude-li na této adrese např. jednotka SBIO-xx, bude odpověď vypadat stejně jako v předchozím odstavci – lišit se bude jen kód operace (byte 01) a zabezpečení CRC (byty 35 a 36).

Nastavení posloupnosti registrů

Funkce zapisuje do 16 registrů od offsetu 0. V prvních 16 registrech je obsah všech čtyř PDO s defaultně napamovanými objekty (viz. odstavec 3.3.1. na straně 21). Příklad zprávy (výzva) je v následující tabulce (pro SAIO-xx):

Byte	Hodnota	Význam			
00	0x01	adresa příjemce (slave)			
01	0x10	kód operace			
02	0x00	offset HI			
03	0x00	offset LO			
04	0x00	počet položek HI			
05	0x10	počet položek LO			
06	0x20	počet datových bajtů			
07	0x22	Registr 0	PDO1	byte 1	*výstup DA1*
08	0x33			byte 0	*výstup DA0*
09	0x56	Registr 1		byte 3	*výstup DA3*
10	0x11			byte 2	*výstup DA2*
11	0x12	Registr 2		byte 5	*výstup DA5*
12	0x34			byte 4	*výstup DA4*
13	0x00	Registr 3		byte 7	
14	0x00			byte 6	
15	0x00	Registr 4	PDO2	byte 1	
16	0x00			byte 0	
17	0x00	Registr 5		byte 3	
18	0x00			byte 2	
19	0x00	Registr 6		byte 5	
20	0x00			byte 4	
21	0x00	Registr 7		byte 7	
22	0x00			byte 6	
23	0x00	Registr 8	PDO3	byte 1	
24	0x00			byte 0	
25	0x00	Registr 9		byte 3	
26	0x00			byte 2	
27	0x00	Registr 10		byte 5	
28	0x00			byte 4	
29	0x00	Registr 11		byte 7	
30	0x00			byte 6	
31	0x00	Registr 12	PDO4	byte 1	
32	0x00			byte 0	
33	0x00	Registr 13		byte 3	
34	0x00			byte 2	
35	0x00	Registr 14		byte 5	
36	0x00			byte 4	
37	0x00	Registr 15		byte 7	
38	0x00			byte 6	
39	0x18	CRC LO			
40	0xF4	CRC HI			

Po této výzvě jednotka nastaví své výstupy na hodnoty 0x33, 0x22, 0x11, 0x56, 0x34 a 0x12. Odpověď bude následující:

Byte	Hodnota	Význam
00	0x01	adresa odesílatele (slave)
01	0x10	kód operace
02	0x00	offset HI
03	0x00	offset LO
04	0x00	počet položek HI
05	0x10	počet položek LO
06	0xC1	CRC LO
07	0xC5	CRC HI

Informace o jednotce

Funkce vrací stav (operational nebo jiný) a typ jednotky na dané adrese. Po odeslání odpovědi se jednotka přepne do stavu operational (pokud v něm již nebyla). Zpráva (výzva) vypadá následovně:

Byte	Hodnota	Význam
00	0x01	adresa příjemce (slave)
01	0x11	kód operace
02	0xC0	CRC LO
03	0x2C	CRC HI

Pokud je předchozí výzva poslána do jednotky SBO-1x, je z jednotky vrácena následující odpověď:

Byte	Hodnota	Význam	
00	0x01	adresa odesílatele (slave)	
01	0x11	kód operace	
02	0x04	počet bytů	
03	0xFF	ID kód jednotky	*0xFF pro moduly PL2*
04	0x00	stav jednotky	*0xFF ... RUN, 0x00 ... ostatní*
05	0xBD	typ jednotky	*hexadecimální podoba typu jednotky za EI*
06	0x16		
07	0x39	CRC LO	
08	0xCB	CRC HI	

V odpovědi je vracen typ jednotky v hexadecimální podobě podle následující tabulky:

Jednotka	Typ v hexa	Typ dekadicky
SAIO-11	0x16D1	5841
SAIO-12	0x16D2	5842
SBI-11/12	0x16C7	5831
SBO-11/12	0x16BD	5821
SBIO-11/12	0x16DB	5851
SKDM-11/12	0x16F9	5881
FCPU-02A	0x151A	5402

5.1.3. Komunikace pomocí Object Dictionary

Každá jednotka má implementován slovník objektů (Object Dictionary). Příslušná adresa a délka objektu se zjistí v kapitole 4. (Objekt Dictionary – seznam objektů) takto:

V tabulce se vyhledá požadovaná položka. Např. čítač na prvním binárním vstupu jednotky SBI/SBIO je v odstavci „Čítače impulsů na binárních vstupech“, řádek 2. Ve sloupcích nadepsaných „**MODBUS**“ se zjistí potřebná adresa a počet registrů. Adresa je ve sloupci nadepsaném „**I**“, v tomto případě 0x3001, a počet registrů ve sloupci nadepsaném „**L**“, v tomto případě 1. Komunikační relace pro vyčtení čítače na binárním vstupu 0 bude pak vypadat takto:

Výzva:

Byte	Hodnota	Význam
00	0x01	adresa příjemce (slave)
01	0x03	kód operace
02	0x30	offset HI
03	0x01	offset LO
04	0x00	počet položek HI
05	0x01	počet položek LO
06	0xDA	CRC LO
07	0xCA	CRC HI

Odpověď:

Byte	Hodnota	Význam
00	0x01	adresa příjemce (slave)
01	0x03	kód operace
02	0x02	počet bytů
03	0x00	hodnota položky
04	0x5F	
05	0xF8	CRC LO
06	0x7C	CRC HI

Vysledné číslo je pak v odpovědi, v tomto případě 0x005F, čili 95.

5.1.4. Zabezpečení telegramů

Každý telegram je zabezpečen pomocí CRC (Cyclic Redundancy Check). Jeho hodnota je dvoubytová a je umístěn v posledních dvou bytech telegramu – nejprve nižší byte, potom vyšší byte. Je vypočítán vysílacím zařízením a připojen ke zprávě. Přijímací zařízení přepočítává CRC během příjmu a potom porovná vypočtenou hodnotu s hodnotou CRC v přijaté zprávě. Pokud se nerovnají, znamená to chybu.

Postup generování CRC

1. naplnění 16-bitového registru hodnotou **0xFFFF**
2. bitová operace XOR prvního bytu zprávy s nižším bytem registru a uložení výsledku do registru.
3. posun obsahu registru o jeden bit vpravo, nejvyšší bit se vynuluje, vypadlý nejnižší bit se zpracovává.
4. zpracování vypadlého bitu – je-li 0, opakuje se krok 3, je-li 1, provede se bitová operace XOR registru s hodnotou **0xA001**
5. opakování kroků 3 a 4, dokud není provedeno 8 posunů registru
6. opakování kroků 2 až 5, dokud nejsou zpracovány všechny byty zprávy
7. výsledný obsah registru je hodnota CRC. Ve zprávě je umístěn nejdříve nižší byte a pak vyšší byte CRC (viz příklady telegramů v předchozích odstavcích).

5.2. Důležité upozornění

Při komunikaci je třeba dbát jedné zásady: při sestavování zpráv podle odstavce 5.1.2. **musí být offset vždy** 0x0000. Při nenulovém offsetu jednotka provádí přepočet na položku v Object Dictionary.

Při komunikaci podle odstavce 5.1.2. nemusí být délka vždy 16 registrů (všechna čtyři PDO), ale může být i nižší. Příklad zprávy (výzva) pro přečtení hodnot vstupů pro jednotku SBI/SBIO může vypadat následovně:

Byte	Hodnota	Význam
00	0x01	adresa příjemce (slave)
01	0x03	kód operace
02	0x00	offset HI
03	0x00	offset LO
04	0x00	počet položek HI
05	0x01	počet položek LO
06	0x84	CRC LO
07	0x0A	CRC HI

Odpověď vypadá následovně (pro aktivní vstupy X1 a X3):

Byte	Hodnota	Význam			
00	0x01	adresa odesílatele (slave)			
01	0x03	kód operace			
02	0x00	počet odesílaných bytů			
03	0x00	Registr 0	**PDO1**	byte 1	*vstupy X8 až X15*
04	0x0A			byte 0	*vstupy X0 až X7*
05	0x32	CRC LO			
06	0x7F	CRC HI			

6. KOMUNIKAČNÍ PROTOKOL SAM

Tento protokol je „výběhový" a v tomto manuále je uveden pouze pro úplnost a z důvodu zpětné kompatibility s moduly SAM-xx. U nových typů výrobků již není podporován.

6.1. Základní charakteristika

Tento popis definuje strukturu komunikačních příkazů pro jednotky PROMOS Line 2 podporující protokol SAM. Jako přenosové médium se používá asynchronní komunikační rozhraní RS485. Pro komunikaci se používá rychlost od 2400 do 38400 Bd, formát znaku 8 bitů, parita volitelná. Zprávy jsou sestaveny z ASCII znaků, každá zpráva je zakončena znakem **<CR>** (hexadecimálně 0D). Volitelně může být zařazen kontrolní součet, který se umísťuje před ukončovací znak **<CR>** a posílá se jako dva ASCII znaky hexadecimálního čísla aritmetrického součtu modulo 256 všech znaků ve zprávě kromě ukončovacího **<CR>**. Pokud v popisu jednotlivých příkazů není stanoveno jinak, jsou všechny číselné parametry zadávány pomocí hexadecimálních ASCII číslic.

Jednotky se vždy chovají jako „slave" – pouze odpovídají na výzvu. Doba, za kterou jednotka vyšle odpověď (reakční doba), je nastavitelná konfiguračním příkazem. Na jedné sběrnici RS485 může být umístěno více jednotek. V tom případě musí mít každá nastavenu svoji adresu, která je odlišná od všech ostatních na této sběrnici.

Před prvním použitím musí být jednotka nakonfigurována. Adresu, komunikační rychlost a ostatní provozní parametry si jednotky uchovávají ve své vnitřní paměti EEPROM. Parametry zůstávají uchovány po i vypnutí napájení.

Pro číslování I/O linek 64-bitových jednotek FCPU-02A platí konvence podle následující tabulky:

číslo linky	pozice / bit	číslo linky	pozice / bit	číslo linky	pozice / bit	číslo linky	pozice / bit
0	P0/0	16	P2/0	32	P4/0	48	P6/0
1	P0/1	17	P2/1	33	P4/1	49	P6/1
2	P0/2	18	P2/2	34	P4/2	50	P6/2
3	P0/3	19	P2/3	35	P4/3	51	P6/3
4	P0/4	20	P2/4	36	P4/4	52	P6/4
5	P0/5	21	P2/5	37	P4/5	53	P6/5
6	P0/6	22	P2/6	38	P4/6	54	P6/6
7	P0/7	23	P2/7	39	P4/7	55	P6/7
8	P1/0	24	P3/0	40	P5/0	56	P7/0
9	P1/1	25	P3/1	41	P5/1	57	P7/1
10	P1/2	26	P3/2	42	P5/2	58	P7/2
11	P1/3	27	P3/3	43	P5/3	59	P7/3
12	P1/4	28	P3/4	44	P5/4	60	P7/4
13	P1/5	29	P3/5	45	P5/5	61	P7/5
14	P1/6	30	P3/6	46	P5/6	62	P7/6
15	P1/7	31	P3/7	47	P5/7	63	P7/7

6.2. Konfigurační režim

Po zapnutí napájení (nebo resetu) jednotka na 1,5 sec přechází do stavu očekávání příkazu přepnutí do konfiguračního režimu. Komunikační rychlost je nastavena na 2400 Bd, 8 bitů bez parity, adresa je 00. Pro vstup do konfiguračního režimu je třeba v této době poslat tři znaky <ESC>. Pokud jednotka přejde do konfiguračního režimu, ohlásí to zasláním znaku **:** (dvojtečka). Každý další přijatý znak <ESC> bude kvitován další dvojtečkou, která se nezapisuje do konfigurace.

Po vstupu do konfiguračního režimu je možné zjistit jméno jednotky a přečíst uložené konfigurační příkazy. Jméno se zjistí zadáním **?/**, jednotka vrátí **/<jméno>*<verze>** (např. /SAM-01*19990120). Po zadání **??** jednotka vyšle **?<obsah EEPROM>**, tj. všechny příkazy uložené v EEPROM (za každý znak **<CR>** navíc přidá **<LF>**, aby bylo zobrazení čitelné na terminálu), a nastaví ukazatel zápisu zpět na začátek (jako bezprostředně po vstupu do konfiguračního režimu). V konfiguračním režimu bude jednotka všechny zaslané příkazy ukládat do vnitřní konfigurační paměti. Ukončení konfiguračního režimu se provede znakem **!** (vykřičník). Pokud jednotka po startu nepřijme tři znaky **<ESC>**, přejde po vypršení doby 1,5 s do normálního provozního režimu. Pokud je jednotka v konfiguračním režimu a po dobu 5 minut nepřijme žádný znak, přechází automaticky do normálního provozního režimu.

Jako konfigurační příkaz může být použit kterýkoliv platný příkaz. Příjem příkazů jednotka nijak nepotvrzuje. Přijaté znaky se ihned zapisují do pracovní zóny, která se po ukončení konfiguračního režimu přepíše do vnitřní FLASH. Výjimkou je pouze znak **<ESC>**, který se kvituje dvojtečkou. Interpretace uložených příkazů se provede po výstupu z konfiguračního režimu a vždy při přechodu do provozního režimu po dalším zapnutí napájení jednotky. Pokud nebudou zadány žádné konfigurační příkazy, („**!**" ihned po vstupu do konfiguračního režimu) bude provedeno pouze tovární nastavení. Pokud nebyly zadány žádné konfigurační příkazy, je možné režim opustit zadáním ***** nebo vypnutím napájení a původní konfigurace zůstane beze změny. Pokud bude v průběhu zadání konfiguračních příkazů odpojeno napájení jednotky, může být původní konfigurace porušena a jednotku je nutno nakonfigurovat znovu.

6.3. Obecné a konfigurační příkazy

6.3.1. Nastavení komunikačních parametrů

%aannttccff<CR>

aa stávající adresa jednotky. Při prvním zapnutí nebo po návratu k továrnímu nastavení je adresa 00.

nn nová adresa (01÷FF). Pokud bude přijetí tohoto příkazu úspěšné, bude pro další komunikaci jednotka dostupná na adrese **nn**, adresa **aa** bude uvolněna.

tt prodleva odpovědi v ms. Dovolený rozsah je 00÷FF, což odpovídá 0÷255 ms. Tento parametr umožňuje zpozdit vyslání odpovědi v případě, že okamžitá odpověď je na závadu. To může být důležité např. pro prodlevu na přepnutí směru přenosu opakovače na lince RS485, přepnutí příjem/vysílání radiomodemu ap. Tato prodleva není závislá na nastavení časového kroku parametrem **ff**.

cc kód nastavení rychlosti:

01	300 Bd
02	600 Bd
03	1200 Bd
04	2400 Bd
05	4800 Bd
06	9600 Bd
07	19200 Bd
08	38400 Bd (alternativně 28800 Bd)
09	57600 Bd

ff je 8 konfiguračních bitů:

bit 7 – integrační filtr měřené hodnoty / časový krok
0 ~ 50Hz, 1 ~ 60Hz pro analogové desky, 0 ~ 1 ms, 1 ~ 2 ms pro desky binárních I/O nastavení časového kroku – 0 odpovídá 1 ms, 1 odpovídá 2 ms. Nastavení časového kroku ovlivňuje veškeré doby a časové prodlevy. V popisu příkazů jsou uvedeny doby pro nastavení 0 (1 ms). Při nastavení 1 (2 ms) je nutno všechny časy a prodlevy násobit 2

bit 6 – kontrolní součet – 0 vypnuto, 1 zapnuto. Pokud je nastavením 1 zapnuto, bude v příchozích zprávách i odpovědích zařazován kontrolní součet před ukončovacím znakem **<CR>**.

bit 5 – parita – 0 vypnuta, 1 zapnuta

bit 4 – typ parity – 0 lichá (odd), 1 sudá (even). Je-li bit 5 nulový (parita vypnuta), je tento bit bezvýznamný.

Odpověď:

!nn<CR> – O.K.

?aa<CR> – chyba, příkaz nebyl proveden.

Při správném přijetí je příkaz ihned proveden a teprve potom je vrácena odpověď. Při změně parity či kontrolního součtu bude odpověď vyslána s nově nastavenými parametry. Komunikační rychlost bude změněna pouze, je-li příkaz vykonán z konfigurační EEPROM po startu.

6.3.2. Dotaz na nastavení

$aa2<CR>

aa adresa jednotky

Odpověď:

!aattccff<CR> – O.K

kde **tt**, **cc** a **ff** jsou stejnojmenné parametry, zapsané příkazem nastavení komunikace **%aa...**

?aa<CR> – chyba

6.3.3. Nastavení soft WatchDog Timer

%aaWnnnn<CR>

aa adresa jednotky

nnnn perioda WatchDog v msec, dovolený rozsah je 0001÷FFFF, což odpovídá 1÷65535 ms. Zadáním 0000 je softwarový WatchDog Timer zakázán. Po nastavení nenulové hodnoty bude jednotka vyžadovat periodické „občerstvování“ hodnoty releových výstupů (příkaz **@aaX**) nejpozději do nastavené doby. Nebude-li „občerstvení“ výstupů provedeno včas, jednotka uvede všechny výstupy do neaktivního stavu (relé rozepnou) a rozsvítí se červená indikační LED „Blk“.

Odpověď:

!nn<CR> – O.K.

?aa<CR> – chyba, příkaz nebyl proveden

6.3.4. Čtení EEPROM

$aaE<CR>

aa adresa jednotky

Odpověď:

!:xx..xx<CR>

xx..xx udává obsah EEPROM/FLASH, uložený v režimu konfigurace. Počet znaků může být až 240, obsah zahrnuje i znaky **<CR>** a ukončovací **!**, pokud byl konfigurační režim korektně ukončen.

Pokud je tento příkaz použit v EEPROM, slouží jako označení začátku bloku příkazů vykonaných po vypršení softwarového WatchDog Timeru

6.3.5. Restart z EEPROM bloku po SWDT

Po vypršení softwarového WatchDog Timeru vykoná příkazy uložené v EEPROM za příkazem **$aaE<CR>**.

$aaS

aa adresa modulu

Odpověď:

bez odpovědi

6.3.6. Verze firmware

$aaF<CR>

aa adresa jednotky

Odpověď:

!aarrrrmmdd<CR>

kde **rrrrmmdd** udává rok/měsíc/den revize firmware.

6.3.7. Jméno jednotky

$aaM<CR>

aa adresa jednotky

Odpověď:

!aaSAM-02<CR>

6.3.8. Reset jednotky

reset~aa<CR>

aa adresa jednotky, pro **aa**=00 reset všech jednotek

Odpověď:

nic (při inicializaci mlčí)

6.3.9. Restart – inicializace z EEPROM/FLASH

Příkaz provede novou inicializaci z EEPROM/FLASH.

$aaX<CR>

aa adresa jednotky

Odpověď:

nic (při inicializaci mlčí)

nebo **!aa<CR>** – O.K.

6.3.10. Pauza

Příkaz umožňuje „pozastavit“ činnost jednotky na definovanou dobu, ve které jednotka neprovádí interpretaci přijatých příkazů.

$aaWtt<CR>

aa adresa jednotky

tt doba v 0,01 sec

Odpověď:

!nn<CR> – O.K.

?aa<CR> – chyba

6.4. Obsluha binárních I/O linek

6.4.1. Čtení osazené konfigurace FCPU-02A

Příkaz umožňuje zjistit, jaké I/O piggy jsou osazeny na jednotlivých pozicích jednotky FCPU-02A.

$aaCnn<CR>

aa adresa jednotky

nn číslo linky 0÷63 (viz tabulka v kap 6.1.)

Odpověď:

!nnc<CR>

nn – adresa jednotky

c – typ linky:

I – vstupní (např. PBI-11)

O – výstupní (např. PBO-11)

N – neosazen I/O piggy modul

U – osazen neznámý I/O piggy

6.4.2. Volba režimu jednoho čítače

Příkaz umožňuje nastavit režim čítače na zvoleném vstupu.

@aaMcchhll<CR>

aa adresa jednotky

cc nejvyšší bit **cc** volí čítanou hranu (1 – vzestupná, 0 – sestupná), dolních 6 bitů (pro FCPU-02A, resp. dolní 4 bity pro SBI-11/12 a SAM-04, popř. dolní 3 bity pro SBIO-11/12) hodnoty **cc** určuje číslo linky

hh minimální požadovaná šířka pulsu v úrovni H v msec, 01÷FF odpovídá 1÷255 ms, 0 odpovídá 256 ms

ll minimální požadovaná šířka pulsu v úrovni L v msec, 01÷FF odpovídá 1÷255 ms, 0 odpovídá 256 ms

Odpověď:

!aa<CR> – O.K.

?aa<CR> – chyba

Linky, které nebudou mít nastaven filtr v konfiguraci, budou používat standardní nastavení 5 ms pro každou úroveň.

6.4.3. Volba režimu jednoho čítače u SAM-02

@aaMbphhll<CR>

aa adresa modulu
b bity odpovídající vstupům (bit 0 ~ X0, .. bit 3 ~ X3), jejichž čítače se mají nastavit
p polarita čítací hrany (0..vzestupná, 1..sestupná), pro každý čítač vybraný parametrem b se nastavuje jeden bit, tedy bit0 pro X1 atd.
hh minimální požadovaná šířka pulsu v úrovni H v msec, 01÷FF odpovídá 1÷255ms, 0 odpovídá 256 ms
ll minimální požadovaná šířka pulsu v úrovni L v msec, 01÷FF odpovídá 1÷255ms, 0 odpovídá 256 ms

Po zpracování příkazu jsou všechny zvolené čítače vynulovány.

Odpověď:

!aa<CR> – O.K.

6.4.4. Volba režimu všech čítačů

Příkaz umožňuje nastavit režim čítačů na všech vstupech najednou.

@aaNcchhll<CR>

aa adresa jednotky
cc nejvyšší bit **cc** volí čítanou hranu (1 – vzestupná, 0 – sestupná)
hh minimální požadovaná šířka pulsu v úrovni H v msec, 01÷FF odpovídá 1÷255 ms, 0 odpovídá 256 ms
ll minimální požadovaná šířka pulsu v úrovni L v msec, 01÷FF odpovídá 1÷255 ms, 0 odpovídá 256 ms

Odpověď:

!aa<CR> – O.K.
?aa<CR> – chyba

Linky, které nebudou mít nastaven filtr v konfiguraci, budou používat standardní nastavení 5 ms pro každou úroveň.

6.4.5. Čtení okamžité hodnoty vstupů

@aaI<CR>

aa adresa jednotky

Odpověď:

!aacc<CR> pro SBIO-11/12
!aaccdd<CR> pro SBI-11/12 a SAM-04
!aaccddeeffgghhiijj<CR> pro FCPU-02A
aa adresa jednotky
ccddeeffgghhiijj hodnota odpovídající stavu vstupních linek:
cc – linky 0÷7 (všechny jednotky)
dd – linky 8÷15 (mimo SBIO-11/12)
ee – linky 16÷23 (jen FCPU-02A)
ff – linky 24÷31 (jen FCPU-02A)
gg – linky 32÷39 (jen FCPU-02A)
hh – linky 40÷47 (jen FCPU-02A)
ii – linky 48÷55 (jen FCPU-02A)
jj – linky 56÷63 (jen FCPU-02A)

Pro linky FCPU-02A/03 (viz tabulka v kap 6.1.), které jsou konfigurovány jako výstupní, odpovídá vrácená hodnota nastavenému stavu výstupu.

6.4.6. Čtení okamžité hodnoty vstupů SAM-02

@aaI<CR>

aa adresa modulu

Odpověď:

!aaoi<CR> – O.K.
o – aktuální stav výstupů nastavený příkazem **@aan**, bit 0 ~ Y0, ... bit 3 ~ Y3.
i – aktuální stav vstupů v době zpracování příkazu, bit0 ~ X0, ... bit 3 ~ X3.

6.4.7. Čtení filtrované hodnoty vstupů

@aaY<CR>

aa adresa jednotky

Odpověď:

!aaccddeeffgghhiijj<CR>
aa – adresa jednotky
ccddeeffgghhiijj – hodnota odpovídající stavu vstupních linek:
cc – linky 0÷7 (všechny jednotky)
dd – linky 8÷15 (mimo SBIO-11/12)
ee – linky 16÷23 (jen FCPU-02A)
ff – linky 24÷31 (jen FCPU-02A)
gg – linky 32÷39 (jen FCPU-02A)
hh – linky 40÷47 (jen FCPU-02A)
ii – linky 48÷55 (jen FCPU-02A)
jj – linky 56÷63 (jen FCPU-02A)

Pro linky, které jsou konfigurovány jako výstupní, odpovídá vrácená hodnota nastavenému stavu výstupu. Čtený stav udává stav vstupní linky po průchodu filtrem pro čítače – vrací se hodnota 1, jestliže vstup je ve stavu 1 alespoň po nastavenou dobu **hh** a hodnota 0, jestliže vstup je ve stavu 0 alespoň po nastavenou dobu **ll** (viz příkaz **@aaM...** pro volbu režimu čítačů).

6.4.8. Čtení filtrované hodnoty vstupů SAM-02

@aaY<CR>

aa adresa modulu

Odpověď:

!aaoi<CR>
o aktuální stav výstupů nastavený příkazem **@aan**, bit 0 ~ Y0, ... bit 3 ~ Y3.
i filtrovaný stav vstupů – číslo odpovídající po bitech hodnotám na vstupech BI0÷BI3 po průchodu filtrem pro čítače – vrací se hodnota 1, jestliže vstup je ve stavu 1 trvale alespoň nastavenou dobu **hh** a hodnota 0, jestliže vstup je trvale alespoň po dobu **ll** (viz příkaz **@aaM...** pro volbu režimu čítačů).

6.4.9. Čtení stavu vstupů (displeje a bargrafy)

$aaI<CR>

aa adresa modulu

Odpověď:

?aa<CR> – chyba
!aabnnmm<CR> – O.K., kde:
b bit 2=0 -> proběhl RESET, bity 0 a 1 udávají aktuální stav vstupů S1 a S2
nn, **mm** – indikují počet hran modulo 256 detekovaných na vstupech, **nn** odpovídá vstupu S2 a **mm** vstupu S1. To umožňuje detekovat krátké stlačení tlačítka i mimo komunikační relaci.

6.4.10. Čtení stavu čítačů

@aaPnn<CR>

aa adresa jednotky
nn číslo linky, jejíž čítač se má vracet v odpovědi:
0 ÷ 7 pro SBIO-11/12 a SAM-04
0 ÷ 15 pro SBI-11/12
0 ÷ 63 pro FCPU-02A (viz tabulka v kap. 6.1.)

Odpověď:

>xxxx<CR> – stav čítače linky **nn**
<xxxx<CR> – pokud se jedná o první čtení po startu (reset nebo zapnutí napájení)

6.4.11. Nastavení úrovně prahu

@aaUnnmmmm<CR>

aa adresa jednotky
nn číslo vstupní linky
mmmm úroveň prahu

Odpověď:
!aa<CR> – O.K.
?aa<CR> – chyba

Nastavení úrovně prahu pro testování přítomnosti pulzů na vstupní lince. Pokud je měřená perioda impulsů vstupního signálu menší než nastavený práh, je indikována přítomnost pulzů na vstupní lince (logická 1). Po zapnutí jsou nastaveny všechny prahy na 0. Pokud je v **nn** nastaven nejvyšší bit, jsou prahy všech linek nastaveny na stejnou hodnotu.

6.4.12. Kombinované čtení stavu vstupů

@aaV<CR>

aa adresa jednotky

Odpověď:
stejná jako u příkazu **@aaY<CR>**.

Vstupní linky, u kterých je nastaven práh testování přítomnosti pulzů na hodnotu 0, je vracen stejný údaj jako u příkazu **@aaY**. Vstupní linky s nastaveným prahem větším než 0 vracejí 1, pokud je detekována přítomnost impulzů (jinak vracejí 0).

6.4.13. Dotaz na periodu signálu

@aaTmm<CR>

aa adresa jednotky
mm číslo vstupní linky

Odpověď:

!aaTnnnn<CR>

Dotaz na periodu signálu na lince **mm**, kde **nnnn** je perioda měřená od předposlední do poslední náběžné hrany na lince **mm**. Jsou brány hrany filtrovaného vstupu. Pokud poslední měřená perioda byla např. 1234, a od poslední hrany uběhlo více času než 1234, pak vrácená hodnota **nnnn** je čas od poslední hrany do okamžiku dotazu. Časová jednotka měření periody odpovídá 10ms. Je-li perioda větší než 10×65535 ms, vrácená hodnota **nnnn** je FFFF.

6.4.14. Nastavení stavu výstupů

@aaOcc<CR>	pro SBIO-11/12
@aaOccdd<CR>	pro SBI-11/12
@aaOccddeeffgghhiijj<CR>	pro FCPU-02A

aa adresa jednotky
ccddeeffgghhiijj – hodnota s požadovaným stavem výstupních linek:
cc – linky 0÷7 (všechny jednotky)
dd – linky 8÷15 (mimo SBIO-11/12)
ee – linky 16÷63 (jen FCPU-02A)
ff – linky 24÷31 (jen FCPU-02A)
gg – linky 32÷39 (jen FCPU-02A)
hh – linky 40÷47 (jen FCPU-02A)
ii – linky 48÷55 (jen FCPU-02A)
jj – linky 56÷63 (jen FCPU-02A)

U FCPU-02A hodnoty zadané pro linky (viz tabulka v kap 6.1.) nastavené jako vstupní jsou ignorovány.

Odpověď:
!aa<CR> – O.K.
?aa<CR> – chyba

6.4.15. Nastavení stavu výstupů u SAM-02

@aan<CR>

aa adresa modulu
n číslo odpovídající po bitech výstupům Y, bit 0 ~ Y0, ... bit 3 ~ Y3.

Odpověď:
!aa<CR> – O.K.

6.4.16. Vyslání hodnoty

@aaTccxoa

aa adresa modulu
cc adresa cílového modulu
x maska XOR
o maska OR
a maska AND

Pozn.: pracuje s filtrovanými hodnotami vstupů

6.5. Obsluha analogových I/O linek

6.5.1. Nastavení rozsahů analogového vstupu

%aaRn±ddd.d±hhh.h<CR>
nebo
#aaRn±ddd.d±hhh.h<CR>

aa adresa jednotky
n číslo vstupu (0÷B pro SAIO-11, 0÷3 pro SAM-01)
±ddd.d±hhh.h dolní a horní hranice rozsahu. Čísla musí být vždy zadána 4 ciframi se znaménkem (znaménko + je zde povinné), obě čísla musí mít desetinnou tečku na stejné pozici. Např.:

```
+.0000+.9999
−0.500+0.500
+00.00+10.00
−100.0+100.0
+0000.+1000.
```

Nastavená dolní hranice bude odpovídat měřené analogové hodnotě při čísle 0 z převodníku a horní hranice nejvyššímu číslu z převodníku. Nastavené hranice rozsahů musí být v souladu s použitou konfigurační odporovou sítí příslušného vstupu.

Odpověď:
!aa<CR> – O.K.
?aa<CR> – chyba, příkaz nebyl proveden

6.5.2. Čtení analogové hodnoty

#aan<CR>

aa adresa jednotky
n číslo vstupu (0÷B pro SAIO-11, 0÷3 pro SAM-01)

Odpověď:
>+123.4<CR> – analogová hodnota je vracena vždy v takovém formátu, jak byly zadány dolní a horní hranice rozsahu.

6.5.3. Nastavení analogových výstupů

#aaOndd.d<CR>

aa adresa jednotky
n číslo analogového výstupu (0÷5 pro SAIO-11)
dd.d hodnota od 00.0 do 99.9 nastavuje výstup na požadovanou hodnotu.

Odpověď:
!aa<CR> – O.K.
?aa<CR> – chyba

6.5.4. Výběr typu domečku

Příkaz je určen jen pro moduly SAM-01 od verze firmware 20041005.

#aaSnAIPx-ttm

aa adresa modulu
n číslo kanálu
x druh domečku (U,V,I,R,B)
tt dvojmístné typové číslo (uživatelské typy 9xx nejsou podporovány)

`m` modifikace U – unipolarní
B – bipolarní
5 – Ni5000
6 – Ni6180

Odpověď:
`!aa` – O.K.

6.5.5. Nastavení koeficientů polynomu

Příkaz je určen jen pro moduly SAM-01 a to od verze firmware 20041005.

`#aaLn(k1:k2:k3)`
`aa` adresa modulu
`n` číslo kanálu
`k1,k2,k3` float hodnoty linearizačního polynomu

Odpověď:
`!aa` – O.K.

6.5.6. Aktivní odeslání hodnoty na displej

Příkaz je určen jen pro moduly SAM-01 od verze firmware 20041005. Příkaz se používá společně s příkazy $aaE, $aaS. Frekvence odesílání se nastavuje softwarovým WatchDogem.

`#aaTddf`
`aa` adresa modulu
`dd` adresa cílového displeje
`f` bity 0 a 1 – číslo kanálu
3 – se znaménkem
2 – 4 nebo 3 cifry

Příklad konfigurace SAM-01 (posílá hodnotu vstupu 0 na displej s adresou 2 každých cca 0,25 s):

```
%01010A0700
%01W00F0
%00R0+0000.+4095.
%00R1+0000.+4095.
%00R2+0000.+4095.
%00R3+0000.+4095.
$01E
#01T024
$01S
```

6.6. Obsluha číslicových zobrazovačů

6.6.1. Nastavení počtu číslic displeje

`"aaWn<CR>`
`aa` adresa modulu
`n` – počet obsluhovaných číslic displeje, povolené hodnoty jsou 0÷9, A÷F, 1÷F označuje počet číslic 1 až 15, 0 označuje 16 číslic.

Odpověď:
`!aa<CR>` – O.K.
`?aa<CR>` – chyba

6.6.2. Zobrazení údaje na displeji

`"aaT<číslo><CR>`
`aa` adresa modulu
`<číslo>` = zobrazovaný údaj, např. 123.4 Obecně lze zadat jakýkoliv ASCII znak, který může být interpretován na sedmisegmentovém displeji.

Vždy je nutno zadat všechny znaky, pro které je displej nastaven. Výjimku tvoří znak „`.`“, který je jakoby „navíc“ a interpretuje se jako tečka za číslem. Pokud se mají rozsvítit pouze 4 tečky, je nutno zadat jako znak mezeru a za ní tečku. Je nutné se vyvarovat textových řetězců, které mohou být interpretovány jako příkazy. Také je možné zadávat přímo binární obraz sedmisegmentového indikátoru uvedením ASCII hexadecimálního bytu uvozeného zpětným lomítkem. Přiřazení jednotlivých segmentů je následující:

bit	7	6	5	4	3	2	1	0
segment	a	b	c	d	e	f	g	t

Např. příkaz `"01\92\92\92\92<CR>` rozsvítí na všech 4 místech displeje s adresou `01` tři vodorovné čárky (svítí segmenty a,d,g).

Odpověď
`!aa<CR>` – O.K.
`?aa<CR>` – chyba, příkaz nebyl proveden.

6.6.3. Nastavení jasu

`"aaJn<CR>`
`aa` adresa modulu
`n` udává požadovaný jas, povolené hodnoty jsou 0÷9, A÷F, 0 - nejnižší jas, F - nejvyšší jas

Odpověď:
`!aa<CR>` – O.K.
`?aa<CR>` – chyba

6.6.4. Přečtení vstupů (XDM14A)

`"aaI`

Odpověď:

`!aabnnmm`
`b` bit 2 =0 ... proběhl RESET
bit 0 a 1 ... aktuální stav vstupů
`nn`, `mm` počet hran detekovaných na vstupech (modulo 256)

6.6.5. Režim čítače (upravený XDM25)

`"aaCnnjjtt`
`nn` minimální délka čítaného pulsu v nule
`jj` minimální délka čítaného pulsu v jedničce
`tt` minimální délka nulovacího pulsu v nule

Odpověď:
`!aa`

6.7. Obsluha sloupcových indikátorů

6.7.1. Zobrazení sloupce

Příkaz zobrazí souvislý svítící sloupec.

`^aaSsnn<CR>`
`s` číslo sloupce 0 nebo 1 (0 levý sloupec)
`nn` výška svítícího sloupce (0 až 23)

Odpověď:
`!aa<CR>` – O.K.
`?aa<CR>` – chyba

6.7.2. Zobrazení sloupce binární mapou

Příkaz umožňuje ovládat nezávisle každou svítivou diodu.

`^aaVsbbbbbb<CR>`
`s` číslo sloupce 0 nebo 1 (0 levý sloupec)
`bbbbbb` šest hexadecimálních číslic, definujících svítící mapu. Pravý bit pole odpovídá spodní LED.

Odpověď:
`!aa<CR>` – O.K.
`?aa<CR>` – chyba

6.7.3. Volba počtu sloupců

`^aaWv`
`v` bity 0 az 2 ... převrácený sloupec (implicitně 1)
bit 3 ... výška sloupce 3 nebo 4 bajty

Odpověď:

!aa O.K.
?aa chyba

6.7.4. Přečtení vstupů

^aaI

Odpověď:

!aabnnmm

b bit 2 =0 ... proběhl RESET
bit 0 a 1 ... aktuální stav vstupů
nn, **mm** ... počet hran detekovaných na vstupech (modulo 256)

6.8. Komunikace dvou FCPU-02A

Tyto změny byly implementovány, aby bylo možno propojit dvě jednotky FCPU-02A mezi sebou a vhodnou konfigurací dosáhnout, aby si jednotky samy navzájem předávaly stavy svých vstupů na výstupy jednotky protější.

Příklad konfigurace jednotky 1:

```
%01010A0700
%01W0400
```

Příklad odpovídající konfigurace jednotky 2:

```
%02020A0700
%02W00F0
$02E
@02W01
>@01W02
```