



# DME 230/400

## Rozhraní Profinet

pro řadu servopohonů

- BN6773

- BN6783

Typ:

DME 230x4-PN

DME 400x8-PN

Číslo dílu:

81703,00103

81703,00112

Publikační číslo: 160616



Dunkermotoren GmbH | Allmendstraße 11 | D-79848 Bonndorf/ Schwarzwald Telefon +49  
(0) 7703 930-0 | Fax +49 (0) 7703 930-210/ 212 | [info@dunkermotoren.com](mailto:info@dunkermotoren.com)

**Servopohony nové generace**  
**Digitální servopohony**  
**pro přímé připojení k síti**  
**Rozhraní Profinet**  
**Návod k obsluze 6770.237, V 1.0**

Tento návod k obsluze platí pro

- Servopohony nové generace, kompaktní design, s integrovaným bezpečnostním systémem
  - BN 6771 až BN 6774 s vestavěným napájecím zdrojem pro jednofázové připojení střídavého proudu
  - BN 6781 až BN 6787 s vestavěným napájecím zdrojem pro třífázové připojení napájení
- Provoz přes osobní počítač se softwarem SPP Windows
- Přístup k funkcím zařízení přes komunikační rozhraní

Tento návod k obsluze platí společně s

- Návod k obsluze 6710.201 (Funkce a parametry)\*
- Návod k obsluze 6770.202 (Připojení a uvedení do provozu)
- Návod k obsluze 6710.207 (SPP Windows Command and Starting Software)\*

\* Dostupné ve funkci nápovědy SPP Windows a ke stažení

ESR Pollmeier GmbH  
Lindenstr. 20  
64372 Ober-Ramstadt  
Spolková republika Německo  
Telefon +49 6167 9306-0  
Fax + 49 6167 9306-77

E-mail [info@esr-pollmeier.de](mailto:info@esr-pollmeier.de)  
[www.esr-pollmeier.de](http://www.esr-pollmeier.de)

**Verze dokumentu**

2015-09-14 V 1.0, KS

nové na základě německé verze

Copyright ESR Pollmeier GmbH, 64372 Ober-Ramstadt, Německo ESR je registrovaná ochranná známka společnosti ESR Pollmeier GmbH.

Všechna práva, včetně práv na překlad, vyhrazena. Žádná část tohoto návodu k obsluze nesmí být bez předchozího písemného souhlasu společnosti ESR Pollmeier GmbH kopírována, reprodukována, ukládána nebo zpracovávána v informačním systému nebo přenášena v jakékoli jiné formě.

Tento návod k obsluze byl pečlivě připraven. Společnost ESR Pollmeier GmbH však nepřebírá žádnou odpovědnost za chyby v tomto návodu k obsluze ani za možné následky. Rovněž nelze přijmout žádnou odpovědnost za přímé nebo nepřímé škody způsobené zneužitím zařízení.

Při používání zařízení je třeba dodržovat příslušné předpisy týkající se bezpečnostní techniky a elektromagnetické kompatibility.

Změny vyhrazeny.

# Obsah

Podívejte se také na rejstřík na konci tohoto dokumentu.

<b>1</b>	<b>Předběžné poznámky</b>	<b>5</b>
1.1	Ab...út.thje..D. e..S.Gozrhac...tio.n.....	5
1,2	Fu...nc..tion.n...B lo.C..k..S.....	5
<b>2</b>	<b>Bezpečnostní pokyny</b>	<b>7</b>
2.1	Ty..p..e...o..f..In . s..tru.c..tio.n..s.....	7
<b>3</b>	<b>Technické specifikace</b>	<b>8</b>
<b>4</b>	<b>Profinet Úvod</b>	<b>9</b>
<b>5</b>	<b>Termíny a zkratky</b>	<b>11</b>
<b>6</b>	<b>Připojení a uvedení do provozu</b>	<b>14</b>
6.1	B...uS.Co..n..n.e.c.tio..n.....	14
6,2	C...o..d..v.G.S.w svědění.....	14
6,3	I...ED. s.....	14
<b>7</b>	<b>Aktivace a deaktivace modulů</b>	<b>16</b>
7,1	P...A.B8.Mó.du.le.....	16
7,2	P...Z.D 16.Mo.du.le.....	16
7,3	mil.o..d..u..le.P.A ramE te r..s.....	16
7,4	C...o..n..n.e.c.tio.n..M na ito.rin.g.....	17
<b>8</b>	<b>Procesní datová komunikace</b>	<b>18</b>
8,1	O...u..tp.u..t..P. raC. Es..s..DA ta.....	18
8,2	palce..pu..t..Pr..o..ce ss..D . a..ta.....	19
<b>9</b>	<b>Komunikace parametrů</b>	<b>20</b>
9,1	S...tandAr..d...a..n..d...Ex..te.n..d..e..d..S.E rv.ic.e..s.....	20
9,2	S...tandAr..d...S..e..rv.ic.e..s.....	20
9.2.1	Pa.r..a.m .e..te.r..C.hAnnEl..S. tru.c..tu.re.....	21
9.2.2	Se.r..vic.e...B.y.te..S. tru.ctu re.....	21
9.2.3	Tra...ns.mje.s.io.n..Fo.r.m . na.....	24
9.2.4	Př...mple.s...PA ra mE ter...Co..m.mu.nic.a.tio.n.w.th..th.e..S.Erv.o..D.rive.....	24
9.2.4.1	W...ritin.g...P..a..ra.m .e..te.rs...(E x.a.m.p.le).....	25
9.2.4.2	R.e..a..d..in.g...P . a..ra mE te rs. (E x.a.m. p..le).....	26
9.2.4.3	R.e..s..p..o..n..se...w.to..Err.o..r..S.ta.tu s...(E x.a.m.p le.).....	27

9,3	Extended Service.....	28
9.3.1	ParameterChannelStructure.....	28
9.3.2	ServiceByteStructure.....	29
9.3.3	ControlByteStructure.....	30
9.3.4	SequentialDataTransfermission.....	30
9.3.4.1	InitiateRead R. například ..... (SequentialRead).....	30
9.3.4.2	ReadSEGment.....	31
9.3.4.3	InitiateWriteRead Request...(Initiate SnapshotWrite).....	32
9.3.4.4	WriteSEGment.....	33
9.3.4.5	Abort of Data Transfermission by DriveSystem.....	34
9.3.4.6	Abort of Data Transfermission by Master.....	35
9,4	Error Codes.....	36
<b>10</b>	<b>Diagnostika alarmu</b>	<b>38</b>
<b>11</b>	<b>Typy protokolů</b>	<b>39</b>
11,1	ESRP protocol.....	39

# 1 Předběžné poznámky

## 1.1 O tomto popisu

Tento návod k obsluze 6770.237 vysvětluje rozhraní Profinet digitálních servopohonů nové generace.

Jsou použitelné společně s

- Návod k obsluze „Připojení a uvedení do provozu“ servopohonu (je součástí dodávky servopohonu)
  - Návod k obsluze 6770.202 (servopohony nové generace)
- Návod k obsluze „Funkce a parametry“ servopohonu (dostupný ve funkci nápovědy SPP Windows a ke stažení)
  - Návod k obsluze 6710.201

také v závislosti na vybavení,

- Návod k obsluze „SPP Windows Command and Commissioning Software“ (dostupný ve funkci nápovědy SPP Windows a ke stažení)
  - Návod k obsluze 6710.207
- Návod k obsluze „Part Program“ (spouštění pohybových sekvencí nezávisle na ovladači vyšší úrovně; k dispozici ve funkci nápovědy SPP Windows a ke stažení)
  - Návod k obsluze 6710.231



Pro práci s rozhraním Profinet jsou navíc vyžadovány následující dokumenty:

- návod k obsluze modulů Profinet použitých pro kontrolér a odpovídající software (IO kontrolér), jakož i
- konfigurační nástroj kompatibilní s IO kontrolérem.



Před přístupem k servopohonu přes rozhraní Profinet podle tohoto návodu k obsluze uveďte servopohon (servopohon a servomotor) do provozu. K tomu je zapotřebí PC s řídicím a zprovozňovacím softwarem SPP Windows. Ujistěte se prosím, že jsou tyto požadavky splněny.

## 1.2 Funkční bloky

Pro snadnou integraci systémů servopohonů do automatizačních systémů jsou k dispozici funkční bloky.

Ty jsou k dispozici pro Siemens Simatic S7 a různé regulátory podle IEC 61131-3. Pro další informace kontaktujte ESR.

Komunikace probíhá přes Profinet IO.

Podporované funkce:

- parametrizace systémů servopohonu regulátorem (např. po zapnutí)

- spouštění pohybů (relativní/absolutní polohování, návrat do výchozí polohy, nastavení rychlosti, ...)
- ovlivnění řízení polohování (dílčí program) integrované v systému pohonu
- vstup a výstup binárních signálů (softwarové vstupy/výstupy)
- ukázkové programy pro použití knihovny funkcí jako základu pro vývoj vlastních programů

Funkční bloky jsou založeny na specifikaci PLCopen „Function Blocks for Motion Control“, která je založena na IEC 61131-3.

Další informace viz katalogový list 6710.260.

Funkční bloky zjednodušují používání mnoha funkcí popsaných v tomto návodu k obsluze. Parametrizační kroky, které musíte provést sami, abyste parametrizovali rozhraní Profinet, jsou popsány v návodu k obsluze funkčních bloků.

## 2 Bezpečnostní pokyny

V každém případě dodržujte bezpečnostní pokyny v návodu k obsluze „Připojení a uvedení do provozu“ (6770.202), jakož i varování a rady na okrajích všech návodů k obsluze.



Přístup k servopohonu přes rozhraní Profinet může vyvolat pohyby systému pohonu. Pokud není pohon a/nebo stroj správně nastaven a zajištěn, může dojít k ohrožení zdraví a života osob.



Proto je přístup přes rozhraní Profinet zakázán, dokud nebudou splněny požadavky směrnice o strojních zařízeních.



U sběrnice systémů může být účastník sběrnice zvenčí neviditelně ovlivněn. To může vést k neočekávanému (nekontrolovatelnému) chování systému. Neuvádějte sběrnici do provozu, pokud jste se ujistili, že jsou všichni účastníci správně připojeni a nakonfigurováni.

### 2.1 Typ pokynů

V každém případě je třeba dodržovat varování a rady na okraji:



- **Nebezpečí** zdraví a života v důsledku úrazu elektrickým proudem nebo pohybu měniče.



- **Pozor:** Nedodržení porušuje bezpečnostní předpisy nebo zákonná ustanovení a může vést ke zranění osob nebo věcným škodám.



- **Kontrola:** Před uvedením do provozu a v případě poruch nebo problémů nejprve zkontrolujte tyto položky.



- **Tip,** užitečný tip



### 3 Technické specifikace

Rozhraní Profinet je instalováno v servopohonech New Generation jako modul (volba F9). Přiřazení pinů a úroveň signálu odpovídají standardu Profinet IEC 61918 nebo IEC 61784-5. Sběrníkové připojení je galvanicky odděleno od servopohonu.

Další vlastnosti rozhraní Profinet servopohonů:

Software	Stack software v binárním nebo zdrojovém formátu
Požadavky	100 Mbit/s plně duplexní, přepínaný Ethernet
Jazyk pro soubor GSD	GSDML
Topologie	hvězda, čára, strom a prsten
Funkčnost	<ul style="list-style-type: none"> <li>• Zařízení PROFINET podle specifikace V2.3, třída shody B</li> <li>• Klient redundance médií</li> <li>• Poskytovatel a předplatitel multicastu</li> </ul>
Počet řadičů PROFINET pro současnou komunikaci (sdílená zařízení)	2
Počet připojení na řadič PROFINET	2
Protokoly	Ethernet, IP, ARP, DCP, DHCP, RPC, RT, IRT, ESR
Max. velikost I/O dat na komunikační vztah	1 024 bajtů
Výměna dat	<ul style="list-style-type: none"> <li>• I/O data cyklicky</li> <li>• Alarmy, přístup pro čtení a zápis do záznamů také acyklický</li> </ul>
Max. počet účastníků	neomezený
Max. konfigurační data v zařízení	8 kbyte
Max. data parametrů v zařízení	8 kbyte
Max. I/O data na komunikační vztah	1 440 bajtů
Podpora profilů	Ano

## 4 Úvod do sítě Profinet

PROFINET (Process Field Network) je otevřený standard průmyslového Ethernetu od Profibus & Profinet International (PI) pro automatizaci. Profinet využívá TCP/IP a IT standardy, je kompatibilní s Ethernetem v reálném čase a umožňuje integraci systémů provozních sběrnic.

Profinet má modulární strukturu, takže si funkčnost může vybrat uživatel. Liší se především způsobem výměny dat, aby byly splněny požadavky na rychlost.

Pro Profinet existují varianty Profinet IO a Profinet CBA:

- Profinet IO (Input - Output) byl vytvořen pro připojení lokální periferie k řadiči. Pro různé oblasti použití jsou dostupné funkce a vlastnosti v reálném čase rozděleny do tří tříd shody CC-A, CC-B a CC-C.
- Profinet CBA (Component Based Automation) je určen pro komunikaci založenou na komponentách přes TCP/IP a komunikaci v reálném čase pro požadavky v reálném čase v modulárním strojírenství. Oba typy komunikace lze používat paralelně.

Profinet IO a Profinet CBA mohou komunikovat současně ve stejném sběrniciovém systému. Lze je provozovat samostatně i kombinovat, takže část zařízení s Profinet IO se z pohledu zařízení jeví jako zařízení Profinet CBA.

Pro servopohony nové generace se používá Profinet IO.

Systém Profinet IO se skládá z následujících zařízení:

- IO ovladač
- IO zařízení (polní zařízení). Skládá se z několika modulů a podmodulů. Submoduly obsahují jednotlivé vstupní a výstupní signály pro proces.
- IO supervizor je vývojový nástroj, typicky založený na PC za účelem parametrizace a diagnostiky jednotlivých IO zařízení.

Minimální systém Profinet IO se skládá z alespoň jednoho IO řadiče, který řídí jedno nebo několik IO zařízení. V případě potřeby lze navíc dočasně přidat jeden nebo několik IO supervizorů.

Pokud existují dva IO systémy ve stejné IP síti, IO kontrolér může také sdílet vstupní signál jako sdílený vstup přístupem ke stejnému submodulu v IO zařízení v režimu čtení.

Každé automatizační zařízení s rozhraním Ethernet může plnit funkci IO kontroléru i IO zařízení současně.

Pro zvýšení dostupnosti lze Profinet implementovat také s redundancí systému. V tomto případě jsou nakonfigurovány dva IO řadiče ovládající stejná IO zařízení.

Mezi IO řadičem a IO zařízeními je vytvořen aplikační vztah (AR).

Prostřednictvím této AR jsou definovány komunikační vztahy (CR) s různými charakteristikami:

- Record Data CR pro acyklický přenos parametrů
- IO Data CR pro cyklickou výměnu procesních dat
- Alarm CR pro signalizaci alarmů v reálném čase

## 5 Termíny a zkratky

Tato část poskytuje stručný přehled hlavních termínů a zkratek. Nenahrazuje originální dokumenty ani řádné školení.

### **Acyklická data parametrů (Record Data CR)**

Acyklický datový provoz se používá pro události, které se neustále neopakují. Příkladem acyklického datového provozu je zasílání parametrizačních a konfiguračních dat při spouštění periferního zařízení do zařízení nebo zasílání diagnostické zprávy z periferního zařízení do centrální jednotky v běžícím provozu.

Acyklická data využívají protokol UDP/IP.

### **Alarmy (acyklická data alarmu)**

Alarmy jsou speciální acyklické zprávy, které v případě potřeby přenášejí periferní zařízení do řídicí jednotky. Ty jsou časově kritické, a proto jsou přenášeny přímo přes Ethernet jako cyklická data. Na rozdíl od cyklických dat je však musí příjemce potvrdit.

### **Název autobusu**

Jedinečný název zařízení účastníka komunikace v síti Profinet.

### **Cyklická data (IO Data CR)**

Obsahem cyklického datového provozu jsou data, která centrální jednotka odesílá na periferii, a také data, která periferní zařízení načítá na svých vstupech a odesílá do centrální jednotky ke zpracování. Obvykle je v každém cyklu takovýto cyklický datový balíček odeslán z centrální jednotky do periferního zařízení a naopak.

### **Rozšířené služby**

Způsob přenosu v parametrické komunikaci; umožňuje přenos parametrů > 4 bajty.

Viz „Kanál parametrů“.

### **soubor GSDML**

Elektronický datový list podle EN 50170; textový soubor ve formátu ASCII.

Pomocí souborů GSDML lze systém Profinet konfigurovat otevřeně (nezávisle na výrobci). Jeden nebo více disků s rozhraním Profinet je zavedeno do hlavního serveru Profinet pomocí souboru GSDML jako vstupu pro konfigurační nástroj.

### **Vstupní data**

Zpracovat data, která slave používá k odpovědi masteru.

### **IO Data ČR**

Viz cyklická data.

### IP adresa

IP adresa servopohonu v síti Profinet je obvykle definována v konfiguračním nástroji Profinet a přenesena do servopohonu při doběhu sběrnice. Alternativně může být uložen v servopohonu.

### Master/slave aplikace

Master nebo slave aplikace se často nazývají aplikace. Z pohledu Profinet je aplikace software nainstalovaný nad softwarem Profinet.

- Hlavní aplikace:       např. firmware
- Slave aplikace:       programu PLC

### Moduly, popisy modulů

Rozhraní Profinet lze také provozovat jako moduly. Mezi klíčová slova je uložen jeden nebo více popisů modulu *ModulaEndModule* v souboru GSDML (definované typy modulů). Z těchto typů modulů lze během konfigurace aktivovat až *Max\_Modules*.

V souboru GSDML servopohonů je definováno několik modulů. Každý modul lze aktivovat pouze jednou, ale nemusí být aktivován.

### Výstupní data

Procesní data, která master posílá do slave.

### Parametry

(Obecně) třída dat přenášených individuálním přístupem. Komunikace parametrů vyžaduje, aby byly parametry adresovány pomocí indexu a subindexu.

Parametry servopohonu a jejich adresy (indexy) jsou uvedeny v návodu k obsluze 6710.201 „Funkce a parametry“.

### Kanál parametrů

Část cyklické komunikace, která zajišťuje přístup k parametrům (objektům) servopohonu pomocí Profinetu.

Když je kanál parametrů aktivní, master aplikace získá přístup ke všem parametrům servopohonů (kromě nebo místo řízení procesních dat).

Pro přenos dat jsou k dispozici dva způsoby:

- Se standardními službami lze přenášet parametry dlouhé až 4 bajty v páru požadavek/odpověď.
- S rozšířenými službami lze přenášet parametry více než 4 bytes.

Segmenty dat jsou přenášeny v posloupnosti párů požadavek/odpověď (segmentovaný přenos).

### **Zpracovat data**

(Obecně) třída dat požadovaná pro řízení a monitorování procesu.

Podle Profinet se procesní data vyměňují jako uživatelská data mezi masterem a jedním nebo více slave v opakujících se cyklech.

Informace o směru se vždy vztahují na velitele. Master posílá svá cyklická výstupní data do slave a slave reaguje odesláním svých cyklických vstupních dat.

„Cyklický“ je ve skutečnosti s ohledem na Profinet (a tedy vstupní a výstupní data) nadbytečný, protože procesní data vždy odkazují na data vyměňovaná v cyklu.

### **Zpracovat datový kanál**

Část cyklické komunikace, která zajišťuje kontinuální cyklický přístup k procesním datům servopohonu přes Profinet.

### **Záznam dat ČR**

Viz acyklická data parametrů.

### **Segmentovaný převod**

Metoda rozšířených služeb v komunikaci parametrů; dlouhé parametry jsou rozděleny do segmentů přenášných v sekvenci párů požadavek/odpověď.

Viz Kanál parametrů.

### **Standardní služby**

Způsob přenosu v parametrické komunikaci; umožňuje přenos parametrů o délce až 4 bajtů.

Viz Kanál parametrů.

## 6 Připojení a uvedení do provozu

Pro připojení a zobrazení stavu jsou servopohony vybaveny následujícími prvky:

- autobusové spojení
- LED diody
- kódovací spínač

Tyto prvky jsou umístěny na předním panelu servopohonů s rozhraním Profinet.

### 6.1 Sběrníkové připojení

X4.1/F7 Profinet-In: konektor RJ45

X4.2/F7 Profinet-Out: konektor RJ45

Rozhraní Profinet se skládá z přepínače Profinet se 2 konektory. Přiřazení konektorů a úroveň signálu odpovídají standardu Ethernet IEEE 802.3 a také standardu IEC 61918 nebo 61784-5.

Konektory X 4.1 a X 4.2, které lze oba použít pro toto připojení, jsou umístěny na předním panelu servopohonů s rozhraním Profinet. Pouze jeden z těchto portů je nutný pro připojení k síti Profinet, druhý port je určen pro volitelné připojení dalších zařízení.

Vzhledem k kabeláži se standardním Ethernetem není nutná koncovka prvního nebo posledního účastníka sběrnice.

### 6.2 Kódovací přepínač

Kódovací přepínač má 16 různých poloh (0-F). Prvních 15 pozic (0-E) nastavuje předdefinované názvy zařízení, na pozici F je nutné název přiřadit pomocí konfiguračního nástroje nebo se použije název tam přiřazený a uložený.

Poloha přepínače	Název zařízení
0	ac-servo-ng
1	ac-servo-ng-1
2	ac-servo-ng-2
...	
D	ac-servo-ng-13
E	ac-servo-ng-14
F	konfigurační nástroj

### 6.3 LED

Sběrníkové LED diody servopohonů slouží k zobrazení vnitřních stavů zásobníku Profinet. Některé vnitřní stavy jsou zobrazovány prostřednictvím různých flash kódů.

**Chyba LED (červená)**

Postavení	Flash sekvence	Popis
NE-CHYBA	Vypnuto	Žádná chyba
DOUBLE_ADDRESS_ERROR	Bliká 2×, poté 1 s zhasne	Byl zjištěn konflikt adres. Sítová komunikace může být narušena.
HARDWARE_ERROR	Bliká 3×, poté 1 s zhasne	Možné příčiny jsou nesprávný přístup k MAC nebo přerušovaný tok dat.
APPL_WATCHDOG_EXPIRED	Bliká 1×krátce, pak 1 s vypnuto	Platnost hlídacího psa aplikace vypršela.
PROTOCOL_ERROR	Na	Došlo k chybě specifické pro protokol.

**LED Run (zelená)**

Postavení	Flash sekvence	Popis
BEFORE_INIT	Vypnuto	Zásobník není inicializován
INIT	1× blikne, poté 1 s zhasne	Stack se inicializuje
ONLINE	Blikání	Stack je online (k zařízení lze přistupovat přes síťové rozhraní)
PŘIPOJENO	Na	Stack je online (k zařízení lze přistupovat přes síťové rozhraní)

**LED Aux1 (žlutá)**

Postavení	Flash sekvence	Popis
NO_PAR	Vypnuto	Kanál parametrů (PZD8) není nakonfigurován nebo nakonfigurován, ale dosud není používán.
PAR_ACCESS	Blikání	Kanál parametrů (PZD8) aktivní, probíhá komunikace
PAR_USED	Na	Kanál parametrů (PZD8) aktivní, ale momentálně žádná komunikace.

**LED Aux2 (žlutá)**

LED Aux2 je vyhrazena pro funkci blesku, a proto je aktivní pouze po spuštění příslušné funkce konfiguračním nástrojem Profinet.



## 7 Aktivace a deaktivace modulů

Stejně jako u Profibusu i u Profinetu jsou moduly pro komunikaci procesních dat předdefinovány. Moduly se zasouvají do tzv. modulových slotů. V současné době je servopohon vybaven 64 modulovými sloty, z nichž pouze prvních 6 může být použito, protože v GSDML je definováno pouze 6 různých modulů a ty jsou přiřazeny k definovaným modulovým slotům. V současné době jsou v souboru GSDML definovány následující moduly:

Modul	Slot	Funkce
PAR8 IN/OUT	1	Kanál parametrů
PZD16 IN/OUT	2	Zpracovat datový kanál
PAR8 IN	3	Alternativní kanál parametrů IN
PAR8 OUT	4	Alternativní kanál parametrů OUT
PZD16 IN	5	Testovací kanál IN
PZD16 OUT	6	Testovací kanál OUT

PZD = Procesní datový kanál; PAR = Kanál parametrů

Standardně jsou přiřazeny pouze první dva moduly. Obsazení a funkce modulů PAR8 IN/OUT odpovídá modulu Profibus PAR (kanál parametrů), PZD16 IN/OUT odpovídá modulu Profibus PZD (kanál procesních dat). Kombinovaný 24bajtový modul PAR+PZD neexistuje. Není to nutné, protože moduly Profinet lze používat nezávisle na sobě. Prostřednictvím modulů PAR8 IN a PAR8 OUT lze ovládat druhý kanál parametrů, moduly PZD16 IN a PZD16 OUT slouží pouze pro účely interního testování a neměly by se používat.

### 7.1 Modul PAR8

Funkce a přiřazení modulu PAR8 přesně odpovídá modulu Profibus PAR, podporovány jsou všechny služby zde definované. Modul PAR je definován jako oktetový řetězec 8, tzn. e jako 8bajtové pole bez další funkční identifikace jednotlivých bajtů.

### 7.2 Modul PZD16

Funkce a přiřazení modulu PZD16-Modul odpovídá modulu Profibus PZD. V souboru GSDML je kanál PZD16 definován jako struktura se svým aktuálním přiřazením (řídící slovo, cílová poloha, stavové slovo, aktuální poloha...) . V závislosti na konfiguračním nástroji se tato informace nepoužívá, stejně jako u Profibusu je PZD16 považováno za 16bajtové pole. Jiné nástroje tyto informace správně interpretují a poskytují jednotlivé proměnné kanálu procesních dat.

### 7.3 Parametry modulu

Ostatní parametry dostupné přes konfigurační nástroj jsou přiřazeny standardním modulům PAR IN/OUT a PZD IN/OUT. Jedná se o:

Parametr	Výchozí	Funkce
Aktivujte diagnostický alarm	Vypnuto	Aktivuje hlášení poplachů prostřednictvím tohoto modulu.
Aktivujte sledování připojení	Vypnuto	Aktivuje monitorování připojení prostřednictvím tohoto modulu.

## 7.4 Sledování připojení

Pokud je aktivováno sledování spojení alespoň jednoho modulu, dojde při ztrátě spojení po rozběhu sběrnice k vyvolání poruchy v servopohonu. Tato funkce tedy zhruba odpovídá master watchdog monitoringu Profibus.

## 8 Procesní datová komunikace

Komunikace procesních dat rozhraní Profinet servopohonů je realizována prostřednictvím tzv. procesního datového kanálu.

Kanál procesních dat obsahuje 16 bytů na směr (výstupní a vstupní procesní data). Na kanálu procesních dat jsou mapovány proměnné digitálních servopohonů; mapování proměnných na jednotlivých bytech kanálu procesních dat je popsáno v následujících tabulkách, vždy po jedné pro výstupní a vstupní procesní data.

Pokud je kanál procesních dat aktivní, délka cyklických dat se zvýší o 16 bajtů. Kanál procesních dat zabírá posledních 16 bytů v cyklickém kanálu.

Jiné délky kanálů procesních dat a mapování jiných proměnných jsou možné na vyžádání, kontaktujte prosím ESR.

Funkce proměnných je popsána v návodu k obsluze 6710.201 „Funkce a parametry“. Věnujte prosím zvláštní pozornost sekci Popisy proměnných.



Pro komunikaci procesních dat v cyklickém kanálu musí být jednotlivým vstupním a výstupním datům cyklické komunikace přiřazeny správné uživatelské adresy v masteru.

### 8.1 Výstupní procesní data

Výstupní procesní data jsou přenášena z masteru (např. PLC, PC) do servopohonu.

Profinet výstupní procesní data		
Zpracovat data		Variabilní
Byte	Bit	
1		Řídicí slovo osy (bit 0 .. 7)
2		Řídicí slovo osy (bit 8 .. 15)
3		Cílová rychlost (bit 31 .. 24)
4		Cílová rychlost (bit 23 .. 16)
5		Cílová rychlost (bit 15 .. 8)
6		Cílová rychlost (bit 7 .. 0)
7		Cílová pozice (bit 31 .. 24)
8		Cílová pozice (bit 23 .. 16)
9		Cílová pozice (bit 15 .. 8)
10		Cílová pozice (bit 7 .. 0)
11	7 .. 4	Digitální výstupy O 2.3 .. 2.0 (softwarové výstupy)
	3 .. 0	Digitální výstupy O 1.3 .. 1.0 (přepínání výstupů na X7)
12		Digitální vstupy I 9.7 .. 9.0 (softwarové vstupy)
13		Nastavená hodnota točivého momentu (bit 15 .. 8)
14		Nastavená hodnota točivého momentu (bit 7 .. 0)

Profinet výstupní procesní data		
Zpracovat data		Variabilní
Byte	Bit	
15		Provozní režim osy (bit 15 .. 8)
16		Provozní režim osy (bit 7 .. 0)

## 8.2 Vstupní procesní data

Vstupní procesní data jsou přenášena ze servopohonu do masteru (např. PLC, PC).

Vstupní procesní data Profinet		
Zpracovat data		Variabilní
Byte	Bit	
1		Stavové slovo osy (bit 0 .. 7)
2		Stavové slovo osy (bit 8 .. 15)
3		Skutečná rychlost (bit 31 .. 24)
4		Skutečná rychlost (bit 23 .. 16)
5		Skutečná rychlost (bit 15 .. 8)
6		Skutečná rychlost (bit 7 .. 0)
7		Skutečná poloha (bit 31 .. 24)
8		Skutečná poloha (bit 23 .. 16)
9		Skutečná poloha (bit 15 .. 8)
10		Skutečná poloha (bit 7 .. 0)
11		Digitální vstupy I 1.7 .. 1.0 (přepínání vstupů na X7)
12		Digitální vstupy I 3.7 .. 3.0
13		Aktuální proud (bit 15 .. 8)
14		Aktuální proud (bit 7 .. 0)
15		Digitální výstupy O 8.7 .. 8.0 (softwarové výstupy)
16		Digitální výstupy O 9.7 .. 9.0 (softwarové výstupy)

## 9 Komunikace parametrů

S rozhraním Profinet je komunikace parametrů realizována prostřednictvím kanálu parametrů. Pomocí kanálu parametrů má Profinet master přístup ke všem parametrům servopohonu (proměnným). Proměnné lze číst a zapisovat přes kanál parametrů a jsou adresovány pomocí indexu a subindexu. Návod k obsluze 6710.201 „Funkce a parametry“ obsahuje konkrétnější informace o proměnných.

Parametry se přenášejí v cyklickém kanálu navíc k obvyklým procesním datům. Kanál parametrů může být aktivován nebo deaktivován masterem.

Když je kanál parametrů aktivní, délka cyklických dat se zvětší o 8 bajtů. Kanál parametrů zabírá prvních 8 bajtů v cyklickém kanálu.

### 9.1 Standardní a rozšířené služby



Všechny parametry (proměnné) servopohonu až do délky 4 bajtů jsou přístupné se standardními službami. Parametry delší než 4 bajty lze přenášet i s rozšířenými službami. To platí pouze pro několik parametrů, které ve většině aplikací nemusí mít master:

Parametry > 4 bajty			
Index	Jméno	Typ dat	Komentář
5ee6	Popis motoru	VisStr 16	
5f43	Vyrovňovací paměť trasování 1	Pole 16 z OctStr 128	Vyrovňovací paměť pro záznam dat; používané funkcemi osciloskopu příkazového a spouštěcího softwaru SPP Windows
5f45	Vyrovňovací paměť trasování 2		
5f47	Vyrovňovací paměť trasování 3		
5f5f	Dílčí program	Pole 71 z OctStr 128	Vytvořeno pomocí editoru partprogramu SPP Windows
5f81	Detail závady	Pole 4 z VisStr 16	Další informace pro několik chybových kódů os

Způsob přenosu standardních služeb nebo rozšířených služeb se vybírá pomocí servisního bajtu.

### 9.2 Standardní služby

Téměř všechny parametry servopohonu (proměnné) jsou přístupné se standardními službami.



Pokud hlavní aplikace nevyžaduje přístup k několika parametrům delším než 4 bajty, je třeba implementovat pouze standardní služby. To značně snižuje implementační úsilí požadované na hlavní straně.

### 9.2.1 Struktura kanálu parametrů

Struktura kanálu parametrů je identická pro oba směry přenosu, jako požadavek ve výstupních datech z masteru na slave a jako odpověď ve vstupních datech z slave na master.

Master nesmí odeslat druhý požadavek, dokud neobdrží odpověď na první požadavek.

Když je kanál parametrů aktivní, ve standardních službách se v cyklickém kanálu před skutečnými daty procesu přenáší následujících 8 bajtů:

Kanál parametrů, standardní služby	
Byte	Obsah
1	Servisní bajt
2	Subindex
3	Index (bit 15 .. 8)
4	Index (bit 7 .. 0)
5	Data/Chyba 1 (bajt s nejvyšší hodnotou)
6	Data/Chyba 2
7	Data/Chyba 3
8	Data/Chyba 4 (bajt s nejnižší hodnotou)

Řízení požadavku a odpovědi kanálu parametrů se provádí přes servisní bajt (byte 1). Servisní byte také vybírá standardní služby jako způsob přenosu. Následující část podrobně popisuje strukturu bajtů služby.

Bajty 2 až 4 adresují parametr. Požadavek obsahuje indexy a podindexy pro adresování parametru. V odpovědi jsou vráceny index a podindex skutečně použité k adresování parametru.

Bajty 5 až 8 obsahují data nebo chybový kód.

### 9.2.2 Struktura servisního bajtu

Servisní bajt (byte 1 v kanálu parametrů) je prvním přeneseným bajtem. Má nejnižší relativní adresu v cyklickém kanálu (adresa 0).

Jednotlivé bity mají následující význam:

Service Byte (standardní služby)	
Bit	Obsah
0..2	Kódování služby: <ul style="list-style-type: none"> <li>• 000zásobník= 0: bez požadavku, standardní služby</li> <li>• 001zásobník= 1: požadavek na čtení, standardní služby (čtení dat ze servopohonu)</li> <li>• 010zásobník= 2: požadavek na zápis, standardní služby (zápis dat na servopohon)</li> <li>• 100zásobník= 4: rozšířené služby (čtení nebo zápis)</li> </ul>
3	Rezervováno (redundantní, ale vždy by mělo být nastaveno na 0)
4 .. 5	Standardní služby: délka dat (počet platných bytes) v polích Data/Chyba <ul style="list-style-type: none"> <li>• 00zásobník= 0:1 bajt</li> <li>• 01zásobník= 1:2 byte</li> <li>• 10zásobník= 2:3 bajty</li> <li>• 11zásobník= 3:4 bajty</li> </ul> Rozšířené služby: kód pro žádost o čtení nebo zápis
6	Handshake (kód, který iniciuje nový požadavek): Master změní tento bit pro každý nový požadavek. Servopohon zkopíruje tento bit do zprávy s odpovědí.
7	Stav (informace o chybě ze servopohonu): S tímto bitem odezva servopohonu informuje master o tom, zda byl požadavek úspěšně dokončen nebo ne. <ul style="list-style-type: none"> <li>• 0 = požadavek úspěšně dokončen</li> <li>• 1 = žádost není dokončena; došlo k závadě. Data v poli Data/Chyba jsou chybová zpráva.</li> </ul>

Požadovaná služba je kódována v bitech 0 až 2. Hodnoty 001zásobník(žádost o přečtení) a 010zásobník(požadavek na zápis) jsou vyhrazeny pro master, protože slave nesmí odesílat požadavky. Servopohony ESR reagují 000zásobník(žádná žádost).

Když servopohony ESR obdrží neplatné servisní kódování, odpoví zprávou o chybovém stavu (kód chyby 06 05 00 10<sub>hex</sub>, Neplatný požadavek Parametr).

Ve standardních službách určují bity 4 a 5 délku dat (počet platných bajtů) v polích Data/Chyba 1 až 4:

- Pro požadavky na zápis z masteru a odezvy na čtení z slave musí být specifikován počet bajtů přenesených v polích Data/Error 1 až 4 (00zásobníkza 1 bajt, 01zásobníkpro 2 bajty atd.). Příjímač tyto informace zkontroluje.
- Pro požadavky na čtení z masteru a pro zápis odpovědi z slave jsou tyto informace nadbytečné a nejsou nutné. Příjímač nemusí kontrolovat informace. Servopohony ESR vždy pošlou 11zásobníkV odpovědi na psaní - se stejně jako v odpovědi na chybový stav, protože délka chybových kódů je vždy 4 bajty.

Kanál parametrů je řízen bitem 6 (handshake). Zprávy s požadavkem a odpovědí jsou přenášeny v cyklickém kanálu, dokud je uživatel znovu nepřepíše. V autobuse jsou dva různé stavy:

- Bity handshake požadavku a odpovědi se neshodují:

- Master odeslal nový požadavek a čeká na odpověď. Požadavek velitele je platný.

Master kontroluje bity handshake v odpovědích od slave a neprovádí žádnou další akci. Nesmí posílat žádné další požadavky, dokud neobdrží od podřízeného zařízení odpověď na aktuální požadavek (se stejným bitem handshake).

- Slave detekuje nový požadavek a zpracuje ho. Stará odpověď podřízeného zařízení je neplatná.

Jakmile slave obdrží nový požadavek, již nekontroluje bity handshake v požadavcích od mastera. Slave zkopíruje bit handshake v odpovědi na nový požadavek (bity handshake v požadavku a odpovědi se nyní shodují).

- Požadavek a odpověď bity handshake se shodují:

- Slave odeslal odpověď a čeká na nový požadavek. Odpověď podřízeného zařízení je platná.

Slave kontroluje bity handshake v požadavcích od mastera a neprovádí žádnou další akci. Nebude posílat žádné další odpovědi, dokud neobdrží nový požadavek od mastera (s jiným bitem handshake).

- Master detekuje reakci podřízeného zařízení a zpracovává ji. Starý požadavek mastera je neplatný.

Jakmile master obdrží odpověď, již nekontroluje bity handshake v odpovědích od slave. Když vytvoří nový požadavek, změní bit handshake (bity handshake v požadavku a odpovědi se neshodují).

Pro hlavní stranu se doporučuje následující postup:

- Bit handshake je po inicializaci okamžitě nastaven na 0.
- Master přečte bit handshake poslední zprávy s požadavkem (stále na sběrnici), invertuje ji a odešle ve své nové zprávě s požadavkem.
- Servisní byte (byte 1) se změněným bitem handshake je posledním přenášeným po sběrnici. Nová žádost je pak platná.

Bit 7 (stav) specifikuje obsah polí Data/Error v odpovědi od slave zařízení:

- Stav = 0: data

Do polí Data/Chyba se přenese hodnota parametru, která zabírá 1 až 4 bajty v závislosti na formátu dat.

- Stav = 1: chyba

Pole Data/Chyba obsahují informace popisující chybu. Tento bit musí být v požadavku masteru nastaven na 0.



### 9.2.3 Formát přenosu

V polích Data/Error lze přenášet hodnoty parametrů až do délky 4 bajtů. Pokud je parametr delší než 1 bajt, je přenášen ve formátu Motorola (nejvyšší bajt/slovo jako první). Parametry kratší než 4 bajty se přenášejí v prvních datových polích:

Parametr Channel		Přenášený prvek			
Byte	Data	Byte	Slovo	3 bajty *	Dvojitě slovo
5	Údaje 1	Bit 7...0	Bit 15 .. 8	Bajt 1	Bit 31 .. 24
6	Údaje 2	–	Bit 7...0	Bajt 2	Bit 23 .. 16
7	Údaje 3	–	–	Bajt 3	Bit 15 .. 8
8	Údaje 4	–	–	–	Bit 7...0

\* Délka pouze 3 znakových řetězců

### 9.2.4 Příklady: Komunikace parametrů se servopohonem



Pro parametrickou komunikaci v cyklickém kanálu musí být vstupní a výstupní data cyklické komunikace přiřazena správným adresám uživatelských dat v Profinet masteru.



Požadavek by měl být vytvořen „zpětně“, aby neúplné požadavky nemohly být odeslány servopohonu a zpracovány jím. Jako poslední se nastavuje první (servisní) byte, který spolu s bitem handshake signalizuje nový požadavek servopohonu.

Master by měl čekat na odpověď slave pouze po omezenou dobu. Užitečná je implementace funkce sledování času.

Požadavek na čtení nebo zápis hlavního serveru se vytvoří následovně (v tomto pořadí):

- Data/chybové pole:
  - požadavek na čtení: zadejte 0.
  - požadavek na zápis: zadejte hodnotu parametru.
- Do polí Index a Subindex zadejte adresu požadovaného parametru.
- Servisní bajt:
  - Nastavte bity servisního kódu podle požadovaného typu požadavku (čtení nebo zápis).
  - Do bitu délky zadejte délku parametru.
  - Nastavte stavový bit na 0 (žádná chyba).
  - Změňte bit pro podání ruky.
- Spustte funkci sledování času.

Chcete-li vyhodnotit odezvu servopohonu, postupujte takto:

- Zkontrolujte, zda je bit handshake ve vstupních datech shodný s bitem ve výstupních datech.

Pokud ano, byla přijata odpověď a funkci sledování času lze zastavit. Pokud čas vyprší a nepřijde žádná nová odpověď, mas-

ter by měl odeslat odpovídající chybovou zprávu.

- Zkontrolujte, zda je nebo není nastaven stavový bit:
  - Pokud bit není nastaven, požadavek byl úspěšně proveden. Pole Data/Error obsahuje požadovanou hodnotu parametru (odpověď na čtení) nebo je prázdné (odpověď na zápis).
  - Pokud je bit nastaven, požadavek nebyl proveden. Došlo k chybě. Pole Data/Chyba obsahuje informace o chybě.

#### 9.2.4.1 Zápis parametrů (příklad)

Maximální aktuální částka (index 6073<sub>hex</sub>, datový typ unsigned16 = word) má být nastaven na 200 %.

- Zpráva od mastera do servopohonu:  
 5200 6073 07D0 0000<sub>hex</sub>(pokud byl bit handshake dříve 0)  
 1200 6073 07D0 0000<sub>hex</sub>(pokud byl bit handshake dříve 1)

Byte	Hodnota	Obsah
1	52 <sub>hex</sub> nebo 12 <sub>hex</sub>	Bit 0 .. 2: xxxx x010 = požadavek na zápis
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx01 xxxx = 2bajtová data v poli Data/chyba
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	00 <sub>hex</sub>	Podindex = 00 <sub>hex</sub>
3	60 <sub>hex</sub>	Index = 6073 <sub>hex</sub>
4	73 <sub>hex</sub>	
5	07 <sub>hex</sub>	Data = 07D0 <sub>hex</sub> (= 2000)
6	D0 <sub>hex</sub>	
7	00 <sub>hex</sub>	
8	00 <sub>hex</sub>	

- Odezva pohonu, pokud je provedena bez chyb:  
 5000 6073 0000 0000<sub>hex</sub>(pokud byl bit handshake 0)  
 1000 6073 0000 0000<sub>hex</sub>(pokud bit handshake byl 1)

Byte	Hodnota	Obsah
1	50 <sub>hex</sub> nebo 10 <sub>hex</sub>	Bit 0 .. 2: xxxx x000 = žádný požadavek
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx11 xx01 = 2bajtová data v poli Data/Chyba
		Bit 6:        x1xx xxxx = odešle zpět handshake 1 x0xx xxxx = odešle zpět handshake 0
		Bit 7:        0xxx xxxx = požadavek proveden
2	00 <sub>hex</sub>	Podindex = 00 <sub>hex</sub>
3	60 <sub>hex</sub>	Index = 6073 <sub>hex</sub>
4	73 <sub>hex</sub>	
5 .. 8	00 <sub>hex</sub>	Data = 0000 0000 <sub>hex</sub> (žádný význam)

#### 9.2.4.2 Parametry čtení (příklad)

Napětí stejnosměrné sběrnice (index 6079<sub>hex</sub>, datový typ unsigned16 = word) má být načten.

- Zpráva od mastera do servopohonu:

5100 6079 0000 0000<sub>hex</sub>(pokud byl bit handshake dříve 0)

1100 6079 0000 0000<sub>hex</sub>(pokud byl bit handshake dříve 1)

Byte	Hodnota	Obsah
1	51 <sub>hex</sub> nebo 11 <sub>hex</sub>	Bit 0 .. 2: xxxx x001 = požadavek na čtení
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx01 xxxx = 2bajtová data v poli Data/chyba
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	00 <sub>hex</sub>	Podindex = 00 <sub>hex</sub>
3	60 <sub>hex</sub>	Index = 6079 <sub>hex</sub>
4	79 <sub>hex</sub>	
5 .. 8	00 <sub>hex</sub>	Data = 0000 0000 <sub>hex</sub> (žádný význam)

- Odezva pohonu, pokud je provedena bez chyb:

5000 6079 0140 0000<sub>hex</sub>(pokud byl bit handshake 1)

1000 6079 0140 0000<sub>hex</sub>(pokud bit handshake byl 0)

Byte	Hodnota	Obsah
1	50 <sub>hex</sub> nebo 10 <sub>hex</sub>	Bit 0 .. 2: xxxx x000 = žádný požadavek
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx01 xxxx = 2bajtová data v poli Data/chyba
		Bit 6:        x1xx xxxx = odešle zpět handshake 1 x0xx xxxx = odešle zpět handshake 0
		Bit 7:        0xxx xxxx = požadavek proveden
2	00 <sub>hex</sub>	Podindex = 00 <sub>hex</sub>
3	60 <sub>hex</sub>	Index = 6079 <sub>hex</sub>
4	79 <sub>hex</sub>	
5	01 <sub>hex</sub>	Data = 0140 <sub>hex</sub> (= 320 V)
6	40 <sub>hex</sub>	
7	00 <sub>hex</sub>	Data = 0000 <sub>hex</sub> (žádný význam)
8	00 <sub>hex</sub>	

### 9.2.4.3 Odpověď se stavem chyby (příklad)

Popis motoru (*Název motoru*, index 5EE6<sub>hex</sub>, datový typ viditelný řetězec16 = 16bitový řetězec znaků) se pokouší přečíst pomocí standardního požadavku na čtení (až 4 bajty na délku).

- Zpráva od mastera do servopohonu:

7100 5EE6 0000 0000<sub>hex</sub>(pokud byl bit handshake dříve 0)

3100 5EE6 0000 0000<sub>hex</sub>(pokud byl bit handshake dříve 1)

Byte	Hodnota	Obsah
1	71 <sub>hex</sub> nebo 31 <sub>hex</sub>	Bit 0 .. 2: xxxx x001 = požadavek na čtení
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx11 xxxx = 4bajtová data v poli Data/Chyba
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	00 <sub>hex</sub>	Podindex = 00 <sub>hex</sub>
3	5E <sub>hex</sub>	Index = 5EE6 <sub>hex</sub>
4	E6 <sub>hex</sub>	
5 .. 8	00 <sub>hex</sub>	Data = 0000 0000 <sub>hex</sub> (žádný význam)

- Odezva disku se stavem chyby:

F000 5EE6 0605 0012<sub>hex</sub>(pokud byl bit handshake 1)

B000 5EE6 0605 0012<sub>hex</sub>(pokud bit handshake byl 0)

Byte	Hodnota	Obsah
1	F0 <sub>hex</sub> nebo B0 <sub>hex</sub>	Bit 0 .. 2: xxxx x000 = žádný požadavek
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx11 xxxx = 4bajtová data v poli Data/Chyba
		Bit 6:        x1xx xxxx    = odešle zpět handshake 1 x0xx xxxx    = odešle zpět handshake 0
		Bit 7:        1xxx xxxx = požadavek nebyl proveden, chyba
2	00 <sub>hex</sub>	Podindex = 00 <sub>hex</sub>
3	5E <sub>hex</sub>	Index = 5EE6 <sub>hex</sub>
4	E6 <sub>hex</sub>	
5	06 <sub>hex</sub>	Kód chyby = 06 05 00 12 <sub>hex</sub> (data jsou příliš dlouhá nebo příliš krátká)
6	05 <sub>hex</sub>	
7	00 <sub>hex</sub>	
8	12 <sub>hex</sub>	

## 9.3 Rozšířené služby

Rozšířené služby se používají k přenosu parametrů delších než 4 bajty. Je povoleno přenášet parametry kratší nebo rovné 4 bajtům pomocí rozšířených služeb (k tomu jsou však zapotřebí 2 páry požadavek/odpověď).



Pokud master aplikace nevyžaduje přístup k několika parametrům delším než 4 bajty, musí být implementovány pouze standardní služby. V tomto případě lze značně snížit náklady na implementaci rozšířených služeb požadovaných na hlavní straně.

### 9.3.1 Struktura kanálu parametrů

Struktura kanálu parametrů je identická pro oba směry přenosu, jako požadavek ve výstupních datech z masteru na slave a jako odpověď ve vstupních datech z slave na master.

Master nesmí odeslat druhý požadavek, dokud neobdrží odpověď na první požadavek.

Když je kanál parametrů aktivní, rozšířené služby přenášejí následujících 8 bajtů v cyklickém kanálu před skutečnými daty procesu:

Kanál parametrů, rozšířené služby		
Byte	Obsah první zprávy (iniciovat) nebo zprávy o chybovém stavu	Obsah následujících zpráv (čtení segmentu nebo zápis segmentu)
1	Servisní bajt	Servisní bajt
2	Subindex	Řídicí bajt
3	Index (bit 15 .. 8)	Údaje 1
4	Index (bit 7 .. 0)	Údaje 2
5	Data/Chyba 1 (bajt s nejvyšší hodnotou)	Údaje 3
6	Data/Chyba 2	Údaje 4
7	Data/Chyba 3	Údaje 5
8	Data/Chyba 4 (bajt s nejnižší hodnotou)	Údaje 6

Struktura první zprávy a chybové zprávy je shodná se strukturou ve standardních službách. Další bajty se používají k přenosu dat v následujících zprávách. Řídicí bajt řídí sekvenční přenos.

Řízení požadavků a odpovědí v kanálu parametrů se provádí přes servisní bajt (byte 1). Používá se také pro výběr rozšířených služeb jako způsobu přenosu.

Řídicí bajt udává počet platných datových bajtů v polích Data 1 až Data 6. Obsahuje stavový bit, který indikuje, zda následují další segmenty nebo zda je ukončen přenos parametru.

### 9.3.2 Struktura servisního bajtu

Servisní bajt (byte 1 v kanálu parametrů) je prvním přeneseným bajtem. Má nejnižší relativní adresu v cyklickém kanálu (adresa 0).

Jednotlivé bity mají následující význam:

Service Byte (rozšířené služby)	
Bit	Obsah
0..2	Kódování služby: <ul style="list-style-type: none"> <li>• 000zásobník= 0: přenos dat přerušen měničem nebo odezva na přerušení masterem (bit 7 = 1), rozšířené služby</li> <li>• 100zásobník= 4: rozšířené služby (čtení nebo zápis, viz bity 4 a 5)</li> <li>• 001zásobník= 1 a 010zásobník= 2: požadavek na čtení nebo zápis, standardní služby</li> </ul>
3	rezervováno (redundantní, ale vždy by mělo být nastaveno na 0)
4 .. 5	Rozšířené služby: kódy pro žádost o čtení nebo zápis <ul style="list-style-type: none"> <li>• 00zásobník= 0: zahájí požadavek na čtení (zahájí čtení segmentu) nebo žádný požadavek (u podřízených jednotek odpověď)</li> <li>• 01zásobník= 1: čtení datového segmentu ze servopohonu (čtení segmentu)</li> <li>• 10zásobník= 2: zahájí požadavek na zápis (zahájí zápis segmentu)</li> <li>• 11zásobník= 3: zápis datového segmentu na servopohon (zápis segmentu) Standardní služby: délka dat</li> </ul>
6	Handshake (kód, který iniciuje nový požadavek): Master změní tento bit pro každý nový požadavek. Servopohon zkopíruje bit ve své zprávě s odpovědí.

Service Byte (rozšířené služby)	
Bit	Obsah
7	Stav (informace o chybě ze servopohonu): Servopohon používá tento bit ve své odpovědi, aby informoval master, zda byl požadavek úspěšně proveden nebo ne. V rozšířených službách hodnota 1 způsobí přerušení sekvenčního přenosu dat. Master může také iniciovat toto přerušení. <ul style="list-style-type: none"> <li>• 0 = požadavek byl úspěšně proveden</li> <li>• 1 = požadavek nevyřízen; došlo k chybě. Přenos sekvenčních dat se zastaví. Data v poli Data/Error jsou chybová zpráva.</li> </ul>

Bity 0 až 2 jsou kódovány 100<sub>zásobník</sub> v rozšířených službách. Zatímco ve standardních službách, slave vždy odpoví servisním kódem 000<sub>zásobník</sub> (= žádný požadavek), v rozšířených službách odpoví servisním kódem 100<sub>zásobník</sub> (= rozšířené služby).

Lze tedy jasně rozlišovat mezi odpověďmi v rozšířených službách a odpověďmi ve standardních službách.

Po obdržení neplatného servisního kódu odpoví servopohonu ESR zprávou o chybovém stavu (kód chyby 06 05 00 10<sub>hex</sub>, neplatný parametr požadavku).

Bity 4 a 5 určují typ požadavku v rozšířených službách (čtení nebo zápis).

Stejně jako u standardních služeb je kanál parametrů řízen bitem 6 (handshake).

Bit 7 (stav) určuje obsah polí Data/Error v odpovědi podřízeného zařízení. Když stav = 1, sekvenční přenos je zrušen. Master může také iniciovat toto přerušení.

### 9.3.3 Struktura řídicího bytu

Řídicí bajt je přenášen jako bajt 2 v kanálu parametrů při sekvenčním přenosu dat od druhé zprávy dále.

Jednotlivé bity mají následující význam:

Control Byte	
Bit	Obsah
0	Stav sekvenčního přenosu: <ul style="list-style-type: none"> <li>• 0: existují další segmenty</li> <li>• 1: neexistují žádné další segmenty</li> </ul>
1 .. 3	Počet platných datových bytů v Data 1 .. Data 6, platné hodnoty: 001 <sub>zásobník</sub> (1) až 110 <sub>zásobník</sub> (6)
4 .. 7	rezervováno (redundantní, ale vždy by mělo být nastaveno na 0)

### 9.3.4 Sekvenční přenos dat

#### 9.3.4.1 Zahájit požadavek na čtení (zahájit čtení segmentu)

Master specifikuje způsob přenosu (bity 0 až 2 v servisním byte = 100<sub>zásobník</sub>, rozšířené služby) a typ požadavku (bity 4 a 5 = 00<sub>zásobník</sub>, Iniciovat Segment Read) se servisním bytem v úvodní zprávě požadavku na čtení do měniče. Adresa parametru, který se má číst, je uvedena v polích subindex a index.

Zpráva pro Drive System: Spustíte čtení segmentu v rozšířených službách		
Byte	Hodnota	Obsah
1	44 <sub>hex</sub> nebo 04 <sub>hex</sub>	Bit 0 .. 2: xxxx x100 = rozšířené služby
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5:    xx00 xxxx = spustí nový požadavek na čtení (Zahájit čtení segmentu)
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	...	Subindex
3 .. 4	...	Index
5 .. 8	00 <sub>hex</sub>	Data = 0000 0000 <sub>hex</sub> (žádný význam)

Pokud byl požadavek úspěšně proveden, měnič ve své odpovědi vrátí délku parametru jako 32bitovou hodnotu bez prefixu (ve formátu Motorola):

Odpověď od Drive System(zahájit čtení segmentu)		
Byte	Hodnota	Obsah
1	44 <sub>hex</sub> nebo 04 <sub>hex</sub>	Bit 0 .. 2: xxxx x100 = rozšířené služby
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx00 xxxx = žádný požadavek
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	...	Subindex
3 .. 4	...	Index
5	...	Délka (bit 31 .. 24)
6	...	Délka (bit 23 .. 16)
7	...	Délka (bit 15 .. 8)
8	...	Délka (bit 7 .. 0)

Aktuální maximální délka parametru je 128 bajtů.

#### 9.3.4.2      Přečtete si Segment

Jakmile je inicializován požadavek na čtení, jeden nebo více požadavků na čtení segmentů (bity 4 a 5 v servisním bajtu = 01<sub>zásobník</sub>, čísl segment) postupujte podle údajů délka:



Zpráva pro systém pohonu: Přechíst segment		
Byte	Hodnota	Obsah
1	54 <sub>hex</sub> nebo 14 <sub>hex</sub>	Bit 0 .. 2: xxxx x100 = rozšířené služby
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx01 xxxx = čtení segmentu
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	00 <sub>hex</sub>	Řídicí bajt = 00 <sub>hex</sub> (žádný význam)
3 .. 8	00 <sub>hex</sub>	Data = 0000 0000 0000 <sub>hex</sub> (žádný význam)

Slave používá řídicí bajt k určení počtu platných datových bajtů v Data 1 .. Data 6 (nnn = 1 až 6, počínaje Data 1) a indikuje, zda budou následovat další segmenty:

Odpověď od Drive System(číst segment)		
Byte	Hodnota	Obsah
1	44 <sub>hex</sub> nebo 04 <sub>hex</sub>	Bit 0 .. 2: xxxx x100 = rozšířené služby
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx00 xxxx = žádný požadavek
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	0X <sub>hex</sub>	Bit 0:        xxxx xxx0 – pokud následují další segmenty xxxx xxx1 – pokud nenásledují žádné další segmenty
		Bit 1 .. 3:    xxxx nnnx = počet platných datových bajtů v datech 1 až 6
		Bit 4 .. 7: 0000 xxxx (rezervováno)
3	...	Údaje 1
4	...	Údaje 2
5	...	Údaje 3
6	...	Údaje 4
7	...	Údaje 5
8	...	Údaje 6

Pokud je přenášeno méně než 6 datových bajtů, jsou platná data vždy na začátku (například nnn = 4: Data 1 až Data 4 platná).

#### 9.3.4.3 Zahájit požadavek na zápis (zahájit zápis do segmentu)

Master používá tento servisní bajt při inicializaci zprávy s požadavkem na zápis do měniče k určení metody přenosu (bity 0 až 2 v servisním bajtu = 100<sub>zásobník</sub>, rozšířené služby) a typ požadavku (bity 4 a 5 = 10<sub>zásobník</sub>, spustit požadavek na zápis, zahájit zápis segmentu). Je zadána adresa parametru, který má být zapsán

v polích subindex a index. Délka parametru je uvedena jako 32bitová hodnota bez předpony (ve formátu Motorola) v polích Data 1 až Data 4.

<b>Zpráva pro Drive System: Spustíte zápis segmentu v rozšířených službách</b>		
Byte	Hodnota	Obsah
1	64 <sub>hex</sub> nebo 24 <sub>hex</sub>	Bit 0 .. 2: xxxx x100 = rozšířené služby
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5:    xx10 xxxx = spustit požadavek na zápis (zahájit zápis segmentu)
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	...	Subindex
3 .. 4	...	Index
5	...	Délka (bit 31 .. 24)
6	...	Délka (bit 23 .. 16)
7	...	Délka (bit 15 .. 8)
8	...	Délka (bit 7 .. 0)

Aktuální maximální délka parametru je 128 bajtů.

Pokud byl požadavek úspěšně proveden, pohonný systém požadavek potvrdí ve své odpovědi:

<b>Odpověď od Drive System(k zahájení zápisu segmentu)</b>		
Byte	Hodnota	Obsah
1	44 <sub>hex</sub> nebo 04 <sub>hex</sub>	Bit 0 .. 2: xxxx x100 = rozšířené služby
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx00 xxxx = žádný požadavek
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	...	Subindex
3 .. 4	...	Index
5 .. 8	00 <sub>hex</sub>	Data = 0000 0000 <sub>hex</sub> (žádný význam)

#### 9.3.4.4 Napište segment

Po iniciaci požadavku na zápis se vyšle jeden nebo více požadavků na segment dat o zápisu (bity 4 a 5 v servisním bajtu = 11 zásobník, napsat segment) následovat v závislosti na délka dat. Master používá řídicí bajt k určení počtu platných datových bajtů v Data 1 .. Data 6 (nnn = 1 až 6, počínaje Data 1) a udává, zda budou následovat další segmenty:

Zpráva pro systém pohonu: Napište segment		
Byte	Hodnota	Obsah
1	54 <sub>hex</sub> nebo 14 <sub>hex</sub>	Bit 0 .. 2: xxxx x100 = rozšířené služby
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx11 xxxx = zapíše datový segment
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	0X <sub>hex</sub>	Bit 0:        xxxx xxx0 – pokud následují další segmenty xxxx xxx1 – pokud nenásledují žádné další segmenty
		Bit 1 .. 3:    xxxx nnnx = počet platných datových bajtů v datech 1 až 6
		Bit 4 .. 7: 0000 xxxx (rezervováno)
3	...	Údaje 1
4	...	Údaje 2
5	...	Údaje 3
6	...	Údaje 4
7	...	Údaje 5
8	...	Údaje 6

Pokud je přenášeno méně než 6 datových bajtů, jsou platná data vždy umístěna na začátku (například nnn = 4: Data 1 až Data 4 platná).

Pokud byl požadavek úspěšně proveden, pohon ve své odpovědi požadavek potvrdí:

Odpověď od Drive System(napsat segment)		
Byte	Hodnota	Obsah
1	44 <sub>hex</sub> nebo 04 <sub>hex</sub>	Bit 0 .. 2: xxxx x100 = rozšířené služby
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx00 xxxx = žádný požadavek
		Bit 6:        x1xx xxxx – pokud bylo handshake dříve 0 x0xx xxxx – pokud bylo handshake dříve 1
		Bit 7:        0xxx xxxx = žádná chyba
2	00 <sub>hex</sub>	Řídicí bajt = 00 <sub>hex</sub> (žádný význam)
3 .. 8	00 <sub>hex</sub>	Data = 0000 0000 <sub>hex</sub> (žádný význam)

#### 9.3.4.5 Přerušování přenosu dat systémem pohonu

Pohon může přerušit sekvenční přenos dat odesláním zprávy o chybovém stavu (bity 0 až 2 v servisním byte = 000zásobníkbit 7 = 1).

Bajty 5 až 8 zprávy o chybovém stavu obsahují kód chyby:

Přerušení sekvenčního přenosu dat systémem pohonu		
Byte	Hodnota	Obsah
1	F0 <sub>hex</sub> nebo B0 <sub>hex</sub>	Bit 0 .. 2: xxxx x000 = žádný požadavek
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx11 xxxx = 4bajtová data v poli Data/Chyba
		Bit 6: <u>x1xx xxxx</u> = odešle zpět handshake 1 x0xx xxxx = odešle zpět handshake 0
		Bit 7:        1xxx xxxx = požadavek nebyl proveden, chyba
2	...	Subindex
3 .. 4	...	Index
5 .. 8	...	Kód chyby

Slave přeruší přenos dat, protože došlo k chybě. Dříve přenesená data již nejsou platná:

- V požadavku na zápis jednotka zahodí všechna dříve přenesená data.
- V požadavku na čtení master zahodí všechna dříve přenesená data.

#### 9.3.4.6 Přerušení přenosu dat Master

Master může také kdykoli přerušit sekvenční přenos dat nastavením bitu 7 (stav) v servisním bajtu na 1 (pouze rozšířené služby):

Přerušení sekvenčního přenosu dat Master		
Byte	Hodnota	Obsah
1	B4 <sub>hex</sub> nebo 84 <sub>hex</sub>	Bit 0 .. 2: xxxx x100 = rozšířené služby
		Bit 3:        xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx00 xxxx (bez významu, protože bit 7 = 1)
		Bit 6: <u>x1xx xxxx</u> = odešle zpět handshake 1 x0xx xxxx = odešle zpět handshake 0
		Bit 7:        1xxx xxxx = chyba
2	00 <sub>hex</sub>	Řídicí bajt = 00 <sub>hex</sub> (žádný význam)
3 .. 8	00 <sub>hex</sub>	Data = 0000 0000 <sub>hex</sub> (žádný význam)

Pohon odpoví chybovým stavovým hlášením. Kód chyby je 0, protože chyba není na straně jednotky:

Odezva od systému pohonu(k přerušení přenosu mistrem)		
Byte	Hodnota	Obsah
1	F0 <sub>hex</sub> nebo B0 <sub>hex</sub>	Bit 0 .. 2: xxxx x000 = žádný požadavek
		Bit 3: xxxx 0xxx (rezervováno)
		Bit 4 .. 5: xx11 xxxx 4bajtová data v poli Data/Chyba
		Bit 6: x1xx xxxx = odešle zpět handshake 1 x0xx xxxx = odešle zpět handshake 0
		Bit 7: 1xxx xxxx = požadavek nebyl proveden
2	00 <sub>hex</sub>	Podindex = 00 <sub>hex</sub> (žádný význam)
3 .. 4	00 <sub>hex</sub>	Index = 0000 <sub>hex</sub> (žádný význam)
5 .. 8	00 <sub>hex</sub>	Kód chyby = 0000 0000 <sub>hex</sub> (žádná chyba v systému pohonu)

Když je přenos dat přerušen, všechna dříve přenesená data již nejsou platná:

- V požadavku na zápis jednotka zahodí všechna dříve přenesená data.
- V požadavku na čtení master zahodí všechna dříve přenesená data.

## 9.4 Chybové kódy

Pokud dojde k chybě (stavový bit = 1), servopohon ESR odešle informaci popisující chybu v polích Data/Error. Jednotlivá pole mají následující význam:

Parametr Channel		Obsah
Byte	Chyba	
5	Chyba 1	Třída chyb
6	Chyba 2	Kód chyby
7	Chyba 3	Dodatečný kód (bajt s nejvyšší hodnotou)
8	Chyba 4	Dodatečný kód (bajt s nejnižší hodnotou)

Jsou definovány následující chybové kódy:

Parametr Channel				Význam
Chyba 1	Chyba 2	Chyba 3	Chyba 4	
Třída	Kód	Dodatečný kód		
00 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	Žádná chyba
06 <sub>hex</sub>	03 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	Bez oprávnění k přístupu
06 <sub>hex</sub>	05 <sub>hex</sub>	00 <sub>hex</sub>	10 <sub>hex</sub>	Neplatný parametr požadavku
06 <sub>hex</sub>	05 <sub>hex</sub>	00 <sub>hex</sub>	11 <sub>hex</sub>	Neplatný podindex
06 <sub>hex</sub>	05 <sub>hex</sub>	00 <sub>hex</sub>	12 <sub>hex</sub>	Data jsou příliš dlouhá nebo příliš krátká
06 <sub>hex</sub>	07 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	Objekt neexistuje
06 <sub>hex</sub>	08 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	Datové typy se neshodují
08 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	20 <sub>hex</sub>	Žádost nelze splnit

Parametr Channel				Význam
Chyba 1	Chyba 2	Chyba 3	Chyba 4	
Třída	Kód	Dodatečný kód		
08 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	21 <sub>hex</sub>	Nelze provést kvůli místnímu ovladači
08 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	22 <sub>hex</sub>	Nelze provést z důvodu provozního stavu zařízení
08 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	30 <sub>hex</sub>	Mimo rozsah hodnot
08 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	40 <sub>hex</sub>	Kolize s jinými hodnotami

## 10 Diagnostika alarmu

Diagnostické alarmy odesílá servopohon v případě příchozích a odchozích událostí. Funkčně tedy odpovídají diagnostice Profibus, jejich obsah však není kompatibilní s diagnostickými zprávami Profibus. Aby se alarmy objevily v parametru regulátoru *aktivovat diagnostický alarm* musí být aktivován alespoň pro jeden modul. Poté jsou alarmy hlášeny do ovladače přiřazeného k tomuto modulu.

Diagnostické alarmy jsou odesílány jako ExtChannelDiag protokolu PROTOCOL\_SPECIFIC (hodnota 80000000<sub>hex</sub>) typ. Protože chybové kódy servopohonu jsou vždy 16bitové, pro Profinet však pouze hodnoty od 0000<sub>hex</sub> do 7FFF<sub>hex</sub> jsou povoleny v poli ChannelDiag diagnostické zprávy je kód chyby rozdělen na pole ChannelDiag (vysoký bajt) a ExtChannelDiag (nízký bajt) diagnostické zprávy. Vzhledem k tomu, že hodnota 0000<sub>hex</sub> není povoleno v poli ExtChannelDiag, nízký bajt v tomto poli je navíc spojen OR s 0100<sub>hex</sub>. Příklad: Kód chyby A011<sub>hex</sub> se přenáší jako A0<sub>hex</sub> v poli ChannelDiag a 0111<sub>hex</sub> (0011<sub>hex</sub> nebo 0100<sub>hex</sub>) v poli ExtChannelDiag.

Pro všechny známé chybové kódy jsou záznamy (kombinace ChannelDiag a ExtChannelDiag) uloženy v souboru GSDML s odpovídajícím textem, takže konfigurační nástroj může vypsát jak chybový kód servopohonu, tak i textovou zprávu o chybě. pokud to nástroj podporuje (funguje např. v TIA V13).

## 11 Typy protokolů

Protokoly Ethernet, IP, ARP, DCP, DHCP, RPC, RT a IRT jsou standardem Profinet IO. Výrobci zařízení mohou navíc implementovat další protokoly.

Servopohony New Generation s rozhraním Profinet IO podporují uvedené protokoly a navíc protokol ESR specifický pro výrobce, který je podrobně popsán níže.

### 11.1 Protokol ESR

Protokol ESR používaný SPP Windows přes TCP ovladač Comm-TCP standardně běží na portu 6668. SPP Windows tedy může komunikovat přímo se servopohonem přes CommTCP a Ethernet.

Protokol ESR v podstatě mapuje rozhraní DLL SPP Windows na jednoduchý protokol ASCII. Klient (PC) odešle ASCII příkaz na server (servopohon), který na něj odpoví textem ASCII.

Příkazy vždy začínají znakem, za kterým následuje žádný nebo několik parametrů ve formě hexadecimálního řetězce ASCII. Příkaz je ukončen řetězcem znaků CR/LF. Řetězec odpovědi se vždy skládá z chybového kódu ve formě hexadecimálního řetězce ASCII, po kterém následuje ne až několik dat odpovědi ve formě hexadecimálního řetězce ASCII. Řetězec odpovědi je ukončen znakem LF.

Jsou definovány následující příkazy:

Příkaz	Formát	Data výsledků	Příklad / Poznámka
Číst	Riiiiissll  iiii = index ss = podindex ll = délka	dd... data objektu (pouze pokud kód chyby = 0)	R60410002 (přečíst stavové slovo) Počet dat odpovídá délce objektu, je nutné dodržet pořadí bajtů.
Napsat	Wiiiiisslldd...  iiii = index ss = podindex ll = délka	žádný, pouze kód chyby	W604000020F80 (zapište hodnotu 0x800f <sub>hex</sub> do kontrolní slovo) Délka dat musí odpovídat uvedené délce objektu, je nutné dodržet pořadí bajtů.
Identifikovat	já	3×16 datových bajtů, každý 16 bajtů <i>pro prodejce, model a revize</i> struny.	První bajt každého řetězce obsahuje délku řetězce, bajty 2 .. 16 (max.) řetězce.
Určete endian režimu	M	1 datový bajt: 00 = big endian (Motorola) 01 = little endian (Intel)	Určuje formát, ve kterém jsou datové bajty přenášeny s příkazy a výsledky.



Příkaz	Formát	Data výsledků	Příklad / Poznámka
Určete chybu zpráva	Eeeeeeee  eeeeeeee = 32bitový kód chyby, pro který je třeba určit chybovou zprávu	n datových bajtů řetězce chybové zprávy	E000000AC Kód chyby neobsahuje 0, ale délku výsledného řetězce. Pořadí bajtů parametru eeeeeeee vždy ve formátu Motorola.
Určete rozhraní	F	4 datové bajty	Identifikace rozhraní jako 32bitové slovo, dodržujte pořadí bajtů.
Určete ID uzlu	N	4 datové bajty	ID uzlu jako 32bitové slovo, dodržujte pořadí bajtů.

Řetězec odpovědi má jednu z následujících forem:

Varianta	Formát	Poznámka
16bitové chybové kódy	eeeedd...  eeee = 16bitový kód chyby  dd ... = nepovinná výsledná data (viz tabulka příkazů)	S výjimkou příkazu E nejsou data dodána, pokud kód chyby není roven 0000.
32bitové chybové kódy	Xeeeeeeedd ... (předpona X)  eeeeeeee = 32bitový kód chyby  dd ... = nepovinná výsledná data (viz tabulka příkazů)	S výjimkou příkazu E nejsou data dodána, pokud kód chyby není roven 0000.

Poznámky k protokolu:

- Parametry příkazu (index, kód chyby) jsou vždy přenášeny ve formátu big endian (Motorola). Na rozdíl od toho jsou datové bajty příkazů pro čtení a zápis přenášeny buď ve formátu big nebo little endian. Proto by měl být formát přenosu určen pomocí příkazu M přímo po navázání spojení. Datové bajty je třeba odpovídajícím způsobem změnit.

Příklad: Hodnota 0x800F<sub>hex</sub> je napsáno na kontrolní slovo pomocí W60400002800F pro režim 0 (big endian) a W604000020F80 pro režim 1 (malý endian).

- Příkazy musí být ukončeny pomocí CR/LF, protože interní vyhodnocení začíná CR. Proto není povolena identifikace konce řádku ve formátu UNIX (pouze LF). Na rozdíl od toho obsahuje řetězec odpovědi pouze LF na konci řádku:

- Pro jednoduché testy lze použít telnet (zadejte port 6668).

Zvláštnosti přímé komunikace s TCP serverem v servopohonu:

- Režim endian servopohonu je vždy little endian (příkaz režimu vždy dodává 01).

- Nezávisle na konfiguraci zařízení a případně nastavených názvech sběrnic poskytují příkazy F (určení rozhraní) a N vždy stejné konzistentní hodnoty.
- Příkaz E poskytuje chybové zprávy v angličtině.

# Index

## - A -

Acyklická data parametrů 11  
 Budíky 11  
 Aux1 (LED) 14  
 Aux2 (LED) 14

## - B -

Autobusové spojení 14  
 Název autobusu 11

## - C -

Nelze provést kvůli provoznímu stavu zařízení (chyba) 36  
 Nelze provést kvůli místnímu ovladači (chyba) 36

Pozor (bezpečnostní pokyn) 7  
 Zkontrolujte (bezpečnostní pokyn) 7  
 Kódovací spínač 14  
 Kolize s jinými hodnotami (chyba) Připojení 14 36  
 Sledování připojení 17  
 Řídicí bajt 30  
 Ovladač 9  
 Cyklická data 11

## - D -

Nebezpečí (bezpečnostní pokyn) Data jsou příliš dlouhá nebo příliš krátká (chyba) Typy dat se neshodují (chyba) 36 36  
 Zařízení 9  
 Název zařízení 14  
 Diagnostika alarmu 28

## - E -

Chyba (LED) 14  
 Chybové kódy 36  
 protokol ESR 39  
 Rozšířené služby 28

## - F -

Funkční bloky 5

## - G -

soubor GSD 11  
 soubor GSDML 11, 16

## - já -

Vstupní data 11  
 Vstupní procesní data 19  
 Neplatný parametr požadavku (chyba) 36  
 Neplatný podindex (chyba) 36  
 IO data 11  
 IP adresa 11

## - L -

LED diody 14  
 LED diody (Error, Run, Aux1, Aux2) 14

## - M -

Master watchdog monitoring Master/slave 17 11  
 aplikace Popis modulů 1  
 Parametry modulu 16  
 Sloty pro moduly 16  
 Moduly 11  
 Motorola formát 24

## - N -

Bez oprávnění k přístupu (chyba) Počet účastníků 8 36

## - O -

Objekt neexistuje (chyba) 36  
 Mimo rozsah hodnot (chyba) 36  
 Výstupní data 11  
 Výstupní procesní data 18

## - P -

PAR 16  
 modul PAR8 16  
 Kanál parametrů 11, 16, 20  
 Komunikace parametrů 20  
 Parametry 11  
 PLCopen 5  
 Zpracovat data 11  
 Zpracovat datový kanál 11, 16, 18  
 Procesní datová komunikace 18

Profinet 9

Typy protokolů 39

Protokoly (Ethernet, IP, ARP, DCP, DHCP, RPC,  
RT, IRT, ESR) 8

PZD 16

modul PZD16 16

## **- R -**

Záznam dat 11

Požadavek nelze provést (chyba) 36

Spustit (LED) 14

## **- S -**

Bezpečnostní pokyny

Pozor 7

Označení CE 7

Kontrola 7

Nebezpečí 7

EMC 7

Tip 7

Segmentovaný převod 11

Servisní bajt 21, 29

Standardní služby 11, 20

Vedoucí 9

## **- T -**

Tip (bezpečnostní 7  
pokyny) Formát přenosu 24

## **- W -**

Sledování hlídacího psa 17