ZJEDNODUŠENÝ PROTOKOL EPNP

Zjednodušený protokol EPNP podporovaný komunikátory i automaty
řady 400

edice 06.2019
verze 1.2

Zjednodušený protokol EPNP

http:/www.micropel.cz

Obsah

Zjednodušený protokol EPNP

EPNP je textový protokol navržený pro možnost přistupovat vzdáleně k zařízením v síti MICROPEL. Přenášené hodnoty v datových rámcích jsou přítomny v textovém tvaru ve formě 2, 4 nebo 8 hexadecimálních znaků v závislosti na datovém typu hodnoty. Více-bajtové hodnoty se v komunikačních rámcích vyskytují ve formátu Big-endian, tj. nejvyšší bajt hodnoty je přítomen jako první.

Protokol podporují výrobky řady 400

Nová, zjednodušená verze EPNP je implementována ve firmware všech novějších komunikátorů a automatů (řada 400 / CA5) s rozhraním USB, Ethernet nebo GPRS. Nová verze EPNP redukuje počet příkazů a většinu příkazů z dřívější verze implementované ve starších převodnících již nezná. Protokol též zavádí možnost číslovat rámce požadavků a zjednodušuje vyhodnocení chyb. To, zda EPNP komunikátor komunikuje zjednodušenou verzí EPNP lze zjistit z obsahu odpovědi na *příkaz GetServerInfo*.

Obecná struktura EPNP rámce

Do protokolu byla doplněna možnost číslovat rámce na straně zadavatele příkazu. Odpovídající strana poté stejné číslo (SID) vyplní do odpovědi. Díky tomu lze snadno ověřit příslušnost odpovědí k zadanému příkazu.

Struktura **nečíslovaného** EPNP požadavku:

@	ADR	*	MD	N x Data	#	SUM	CR

Struktura odpovědi na požadavek:

@	ADR	*	CMD	M x Data	#	SUM	CR

Struktura chybové odpovědi na požadavek:

@	ADR	!	CMD	ERR	#	SUM	CR

Struktura **číslovaného** EPNP požadavku:

@	ADR	+	CMD	SID	N x Data	#	SUM	CR

Struktura odpovědi na požadavek:

@	ADR	-	CMD	SID	M x Data	#	SUM	CR

Struktura chybové odpovědi na požadavek:

@	ADR	?	CMD	SID	ERR	#	SUM	CR

Bajtovou hodnotu v EPNP rámci reprezentují 2 hexadecimální znaky! Např. dekadická hodnota 8 je reprezentována znaky „08“, hodnota 253 pak znaky „FD“.

- ADR … 1 bajt, adresa cílového automatu v připojené síti PESNET, platný rozsah je 00 až 1F (0 až 31 dekadicky), hodnota 1F (31 dekadicky) určuje, že cílem je automat, s nímž je provedeno přímé spojení (komunikátor)
- CMD … 1 bajt, kód příkazu (typ EPNP požadavku) zadaný vysílací stranou
- ERR … 1 bajt, kód chyby zpracování požadavku
- N x Data … N bajtů, datový obsah požadavku, závislý na typu požadavku (může platit i N=0)

- M x Data … M bajtů, datový obsah odpovědi, závislý na typu požadavku (může platit i M=0)
- SID … 1 bajt, libovolný číselný identifikátor (00 až FF) přidělený EPNP požadavku vysílací stranou
- SUM … 1 bajt, suma číselných hodnot všech znaků rámce umístěných před znakem #
- CR … ukončující znak s hodnotou 13 (Carriage Return)

Heslo EPNP komunikace

Heslo EPNP komunikace je 20bitová uživatelem nastavitelná hodnota, která, pokud není nastavena nulová, kóduje EPNP komunikaci na straně vysílače resp. dekóduje na straně přijímače. Aby se vysílač a přijímač domluvily, musí používat stejné heslo. **Přenášená data po zakódování neodpovídají popisu EPNP**.

Softwarové nástroje MICROPEL kódování komunikace podporují. Kdo má zájem vytvořit vlastní nástroj komunikující s automaty sítě MICROPEL protokolem EPNP, má možnost komunikovat pouze nekódovaně (s heslem 0).

Popis EPNP příkazů

Zde následuje popis příkazů určených k využití v uživatelských aplikacích, které budou přistupovat do sítě automatů MICROPEL. V případě příkazů čtení a zápisu hodnot je třeba používat adresy na datové proměnné dle →*Adresy paměťových prostorů automatů*.

příkaz ReadRam

Příkaz vyčte 1 až 64 položek zadaného typu z paměti RAM automatu. Počet a typ položek řídí bajt CTRL. Kódování bajtu CTRL uvádí níže uvedená tabulka. Kompatibilitu se staršími automaty zajišťuje stránka 0 tj. rozsah EPTR od 0 do 0xFFFF. V tomto rozsahu použije FW komunikátoru k vyčtení paměti příkazy PESNET kompatibilní s automaty MPC300 atp. Pro automaty podporující nové mapování pamětí lze používat i ostatní adresy paměťových prostorů. V případě datového typu bit vrací příkaz hodnotu 0 nebo 1.

Požadavek:

Operátor	Číslo příkazu	Adresa	CTRL			
*	0x2E	EPTR [4B]	CTRL [1B]	#	Suma	CR

Odpověď:

Operátor	Číslo příkazu	Adresa	CTRL	Data			
*	0x2E	EPTR [4B]	[1B]	[N*SZ*1B]	#	Suma	CR

N … počet položek, SZ … bajtová velikost položky

Chybová odpověď:

Operátor	Číslo příkazu	Chybový kód			
!	0x2E	ErrorCode [1B]	#	Suma	CR

Příklad: Požadavek a odpověď čtení jednoho longwordu z paměťové adresy 0x00000604 automatu na adrese 2 v síti PESNET. Cílovým registrem je tedy síťový longword LW[1]. Rámec požadavku je číslovaný, vysílač mu přidělil identifikační hodnotu 0x5A:

```
@02+2E5A00000604C1#B8
@02-2E5A00000604C1000003E8#5A
```

Oba rámce jsou ukončeny nezobrazitelným znakem CR. Vyčtena byla dekadická hodnota 1000 (000003E8 hexadecimálně).

příkaz WriteRam

Příkaz zapíše 1 až 64 položek zadaného typu do paměti RAM automatu. Počet a typ položek řídí bajt CTRL. Kódování bajtu CTRL uvádí níže uvedená tabulka. Kompatibilitu se staršími automaty zajišťuje stránka 0 tj. rozsah EPTR od 0 do 0xFFFF. V tomto rozsahu použije FW komunikátoru k vyčtení paměti příkazy PESNET kompatibilní s automaty MPC300 atp. Pro automaty podporující nové mapování pamětí lze používat i ostatní adresy paměťových prostorů. V případě datového typu bit jsou hodnota i index bitu uvnitř bajtu na příslušné adrese obsaženy přímo v hodnotě CTRL - část rámce se zapisovanými hodnotami má pak nulovou délku.

Požadavek:

Operátor	Číslo příkazu	Adresa	CTRL	Data			
*	0x2F	EPTR [4B]	[1B]	[N*SZ*1B]	#	Suma	CR

N … počet položek, SZ … bajtová velikost položky

Odpověď:

Operátor	Číslo příkazu	Adresa	CTRL			
*	0x2F	EPTR [4B]	CTRL [1B]	#	Suma	CR

Chybová odpověď:

Operátor	Číslo příkazu	Chybový kód			
!	0x2F	ErrorCode [1B]	#	Suma	CR

Význam bitů hodnoty CTRL:

B7	B6	B5	B4	B3	B2	B1	B0
0	0	x	x	hodnota	index bitu		
0	1	počet přenášených Byte[1B] (hodnota 0 znamená počet=64)					
1	0	počet přenášených Word[2B] (hodnota 0 znamená počet=64)					
1	1	počet přenášených LongWord[4B] (hodnota 0 znamená počet=64)					

Příklad: Požadavek a odpověď nastavení 2. bitu (bit[1]) bajtu na paměťové adrese 0x00000208 automatu s adresou 2 v síti PESNET. Cílovým bitem je tak bit M[1]:

```
@02*2F0000020809#37
@02*2F0000020809#37
```

Oba rámce jsou ukončeny nezobrazitelným znakem CR. Požadavek i odpověď jsou v případě zápisu hodnoty bitu nečíslovaným rámcem stejné.

příkaz GetServerInfo

Příkaz vrátí strukturu s informacemi o typu připojeného EPNP komunikátoru. Z obsahu je možno zjistit, zda komunikátor podporuje zjednodušený EPNP, nebo zda jde o starší převodník.

Požadavek:

Operátor	Číslo příkazu			
*	0x01	#	Suma	CR

Odpověď:

Operátor	Číslo příkazu	Data odpovědi			
*	0x01	ServerInfo	#	Suma	CR

Chybová odpověď:

Operátor	Číslo příkazu	Chybový kód			
!	0x01	ErrorCode [1B]	#	Suma	CR

Datová struktura ServerInfo bude délky 44 bajtů, pouze starší převodník CA4 vysílá i více bajtů, úvodních 44 bajtů tvoří ale vždy následující položky:

```
WORD InfoSize[2]; // počet následujících datových bajtů (v Big-endian)
BYTE FwVersion[8]; // textově číslo verze firmware
BYTE SerialNumber[8]; // textově výrobní číslo, u starších komunikátorů!
BYTE ServerName[8]; // textově uživatelské jméno komunikátoru
BYTE ConfigName[8]; // textově název, u starších komunikátorů!
BYTE PesNetLoad; // zatížení sítě, u starších komunikátorů!
BYTE PesNetAddress; // nastavená adresa pro síť PESNET
BYTE PesType[7]; // textově typ zařízení
BYTE CAcfg; // má význam jen u starších komunikátorů!
```

Nejsnažší rozpoznání zda komunikátor, jenž odpověděl, podporuje zjednodušený EPNP provedeme dle obsahu FwVersion, kde nalezneme textově zapsané desetinné číslo verze firmware (dle fwn), např. text „5.056“. Pokud ((fwn*1000) <= 1099 || (fwn*1000)-3320 <= 4999-3320), jde o starší typ převodníku, který komunikuje historickým EPNP. V opačném případě by měl komunikátor podporovat novější EPNP. Z pole PesType lze též zjistit konkrétní typ zařízení, např. „CA5E“.

příkaz GetPlcList

Příkaz vrátí seznam automatů na lince PESNET. Indexy pole odpovídají adresám a hodnoty bajtů představují kódy komunikačních rychlostí - komunikace mezi automaty může běžet vždy jen na jedné z podporovaných komunikačních rychlostí.

- 0x01 – 460800 Baud
- 0x02 – 115200 Baud
- 0xFF – 57600 Baud
- 0xFD – 19200 Baud
- 0xFA – 9600 Baud
- 0xE8 – 2400 Baud
- 0x00 - automat není přítomen

Požadavek:

Operátor	Číslo příkazu			
*	0x05	#	Suma	CR

Odpověď:

Operátor	Číslo příkazu	Data odpovědi			
*	0x05	Kódy kom rychlostí [31B]	#	Suma	CR

Chybová odpověď:

Operátor	Číslo příkazu	Chybový kód			
!	0x05	ErrorCode [1B]	#	Suma	CR

informační rámec ServerBusy

Rámec vysílá komunikátor automaticky po určité době během zpracování jiného EPNP příkazu, jehož zpracování trvá delší dobu. Příkaz nemá definovánu chybu, protože k jeho chybnému zpracování nemůže dojít. Pokud tento příkaz vyšleme komunikátoru, je interpretován jako neznámý. Komunikátor příkaz vysílá ve tvaru:

Operátor	Číslo příkazu	Data			
*	0x6E	0x0000	#	0x65	CR

Adresy paměťových prostorů automatů

Následující tabulka adres ukazuje uživatelsky důležité EPNP adresní prostory paměťových míst datové paměti všech automatů.

Vzhledem k tomu, že se po rozšíření možností nedaly všechny přístupné datové prostory mapovat do stránky 0 (adresy 0x0000 až 0xFFFF) a zároveň byla snaha původní adresy zachovat, lze nyní do některých paměťových míst v novější řadě automatů a periferií (MPC400 aj.) přistupovat pomocí dvou různých adres. Na stránce 0 lze např. nalézt pouze počátek dané paměti, pomocí dlouhých adres je vždy zpřístupněna celá část.

Pro starší automaty a periferie (MPC300 aj.) je kompatibilita zachována, k automatům lze prostřednictvím zařízení s novější komunikační vrstvou EPNP bez problémů přistupovat pomocí adresy PESNET a příslušných adres paměti ve stránce 0.

Pozor! Do prostoru síťových bitů M je nutno zapisovat jednotlivé bity, jestliže požadujeme, aby byla zapsaná hodnota automaticky odeslána všem stanicím v síti PESNET.

Paměťový prostor, registr	Typ	Počáteční krátká adresa (stránka 0)	Velikost prostoru [B] (stránka 0)	Počáteční dlouhá adresa	Velikost prostoru [B]
I (vstupy MPC300, an.)	word[32]	0x00000000	64	-	-
O (výstupy MPC300, an.)	word[32]	0x00000040	64	-	-
D	word[32]	0x00000080	64	-	-
síťové D (D32 až D63)	word[32]	0x000000C0	64	-	-
W (spec. funkční registry)	word[128]	0x00000100	256	0x24000100	256
SECOND (RTC vteřiny)	word	0x00000110	2	0x24000110	2
MINUTE (RTC minuty)	word	0x00000112	2	0x24000112	2
HOUR (RTC hodiny)	word	0x00000114	2	0x24000114	2
DAY (RTC dny)	word	0x00000116	2	0x24000116	2
MONTH (RTC měsíce)	word	0x00000118	2	0x24000118	2
YEAR (RTC roky)	word	0x0000012A	2	0x2400012A	2
WEEK (RTC den týdne)	word	0x0000012C	2	0x2400012C	2
X (vstupy MPC300, dig.)	bit[32]	0x00000200	4	-	-
Y (výstupy MPC300, dig.)	bit[32]	0x00000204	4	-	-
M	bit[64]	0x00000208	8	-	-
síťové M (M64 až M127)	bit[64]	0x00000210	8	-	-
IONodeList, seznam přihlášených IO uzlů	bit[1024]	0x00000228	128	0x21030000	128
IOVirtNodeList	bit[1024]	0x000002A8	128	0x21030080	128

síťové LW **síťové LI** **síťové F**	longword[256] longint[256] float[256]	0x00000600	1024	-	-
SmallPNET síťové M	bit[64]	0x000013B8	8	-	-
SmallPNET síťové D	word[32]	0x000013C0	64	-	-
SmallPNET síťové LW **SmallNET síťové LI** **SmallPNET síťové F**	longword[256] longint[256] float[256]	0x00001400	1024	-	-
StackB **StackW** **StackI** **StackLW** **StackLI** **StackF**	byte[23552] word[11776] int[11776] longword[5888] longint[5888] float[5888]	0x00001800	23552	0x23000000	23552
IONode, datové struktury IO uzlů	byte[N][40] (od indexu 16 jde o uzly EXBUS)	0x00007400	N*40, N<=76 dle typu zařízení	0x21000000	N*40, dle typu zařízení
UserRAM	paměť proměnných jazyka SIMPLE4	0x00008010	<= 32752, dle typu zařízení	0x22000000	dle typu zařízení

Seznam hodnot chybových kódů

Následující seznam obsahuje definované hodnoty chybových kódu, které lze obdržet v chybovém rámci, a popisuje jejich význam:

00 (0x00) … Bez chyby - chybový rámec s tímto kódem by neměl být obdržen.
16 (0x10) … Automat neodpovídá.
17 (0x11) … Překročen časový limit vykonání příkazu.
18 (0x12) … Překročen časový limit vykonání příkazu kvůli nedostupnosti sítě.
19 (0x13) … Chybná odezva sítě.
20 (0x14) … Kolize na lince.
48 (0x30) … Chyba v kontrolním součtu.
49 (0x31) … Chyba v kontrolním součtu.
50 (0x32) … Chyba v obsahu komunikačního rámce.
51 (0x33) … Kontrolní součet chybí nebo je neplatný.
64 (0x40) … Chybná délka komunikačního rámce.
65 (0x41) … Překročena délka pro IOBuffer.
66 (0x42) … IOBuffer je plný, poslední příkaz nebyl vyzvednut.
67 (0x43) … IOBuffer je prázdný, neobsahuje platná data.
68 (0x44) … Chyba zpracování / vyhodnocení komunikačního rámce.
80 (0x50) … Neplatný / neznámý příkaz při zpracovávání externího příkazu.
81 (0x51) … Neplatný / neznámý příkaz při spuštění externího příkazu.
82 (0x52) … Neplatný / neznámý příkaz při dekódování příkazu PESNET nebo EXBUS.
83 (0x53) … Neplatná hodnota pole CMDF komunikačního rámce.
84 (0x54) … Chyba zpracování příkazu dle hodnoty CMDF.
85 (0x55) … Neplatný / neznámý příkaz EXBUS.
86 (0x56) … Vnitřní chyba zpracovávání příkazu.
87 (0x57) … Neplatné parametry příkazu.
88 (0x58) … Neznámá paměťová adresa příkazu.
89 (0x59) … Neplatná paměťová adresa příkazu.

96 (0x60) … Strojek zpracování příkazů je zaneprázdněn.
97 (0x61) … Přístup odepřen / poslední IO příkaz zatím nebyl ukončen.
98 (0x62) … Příkaz nezpracován z důvodu probíhajícího vyhledávání stanic sítě PESNET.
99 (0x63) … Strojek zpracování blokového zápisu se nachází v chybě.
100 (0x64) … Strojek zpracování blokového čtení se nachází v chybě.
112 (0x70) … Nalezen neplatný blok programového kódu.
113 (0x71) … Zaznamenána systémová chyba vyvolaná vykonáváním uživatelského kódu.