



EPNP – UŽIVATELSKÁ PŘÍRUČKA

EPNP (Encapsulated PesNet Protocol) je textový protokol umožňující komunikaci se sítí automatů a periférií firmy MICROPEL přes komunikační převodníky CA21,CA3 a CA4.

**dokumentace protokolu pro verzi firmware převodníků CA
02.2016**

EPNP – uživatelská příručka

© Tomáš Navrátil, Ing.
Viktor Soukal, Ing.
MICROPEL s.r.o.

všechna práva vyhrazena
kopírování publikace dovoleno pouze bez změny textu a obsahu
<http://www.micropel.cz>

Obsah

	Změny dokumentace:	4
1	Co je EPNP?	5
1.1	Komunikace protokolem EPNP	5
1.2	Časové odezvy komunikátoru při komunikaci EPNP	5
1.3	Algoritmus komunikace EPNP	6
2	Struktura rámců	6
3	Seznam příkazů protokolu EPNP	11
3.1	GetServerInfo op.kód: 0x01	11
3.2	LogIn op.kód: 0x03	13
3.3	LogOut op.kód: 0x04	13
3.4	EraseConfig op.kód: 0x09	14
3.5	FlashConfig op.kód: 0x0A	14
3.6	UpdateConfig op.kód: 0x0B	15
3.7	EraseFW op.kód: 0x0C	15
3.8	FlashFW op.kód: 0x0D	16
3.9	UpdateFW op.kód: 0x0E	16
3.10	ServerReset op.kód: 0x0F	17
3.11	ServerRestart op.kód: 0x10	17
3.12	GetPLCList op.kód: 0x05	18
3.13	GetPLCHardList op.kód: 0x06	18
3.14	GetPLCInfo op.kód: 0x21	19
3.15	PrgCluster op.kód: 0x22	20
3.16	DnlFinish op.kód: 0x23	20
3.17	PLCRun op.kód: 0x24	21
3.18	PLCStop op.kód: 0x25	21
3.19	PLCReboot op.kód: 0x26	22
3.20	PLCSetTime op.kód: 0x27	22
3.21	PLCSetDate op.kód: 0x28	23
3.22	PLCRebootFreeze op.kód: 0x29	23
3.23	PLCSetSpeed op.kód: 0x12	24
3.24	GetRxWords op.kód: 0x13	24
3.25	GetRxBits op.kód: 0x14	25
3.26	GetRxLongs op.kód: 0x15	25
3.27	ReadNetWords op.kód: 0x16	26
3.28	WriteNetWords op.kód: 0x17	26
3.29	ReadNetBits op.kód: 0x18	27
3.30	WriteNetBits op.kód: 0x19	27
3.31	ReadNetLongs op.kód: 0x1A	28

3.32	WriteNetLongs	op.kód: 0x1B	28
3.33	ReadSTPWord	op.kód: 0x4E	29
3.34	WriteSTPWord	op.kód: 0x4F	29
3.35	ReadSTPBit	op.kód: 0x50	30
3.36	WriteSTPBit	op.kód: 0x51	30
3.37	ReadRAMBit	op.kód: 0x2A	31
3.38	WriteRAMBit	op.kód: 0x2B	31
3.39	ReadRAMByte	op.kód: 0x40	32
3.40	WriteRAMByte	op.kód: 0x41	32
3.41	ReadRAMWord	op.kód: 0x44	33
3.42	WriteRAMWord	op.kód: 0x45	33
3.43	ReadRAMLong	op.kód: 0x48	34
3.44	WriteRAMLong	op.kód: 0x49	34
3.45	ReadRAMByteBlock	op.kód: 0x42	35
3.46	WriteRAMByteBlock	op.kód: 0x43	35
3.47	ReadRAMWordBlock	op.kód: 0x46	36
3.48	WriteRAMWordBlock	op.kód: 0x47	36
3.49	ReadRAMLongBlock	op.kód: 0x4A	37
3.50	WriteRAMLongBlock	op.kód: 0x4B	37
3.51	ReadRAM64B	op.kód: 0x2C (pouze CA4)	38
3.52	WriteRAM64B	op.kód: 0x2D (pouze CA4)	38
3.53	ReadRAM64W	op.kód: 0x2E (pouze CA4)	39
3.54	WriteRAM64W	op.kód: 0x2F (pouze CA4)	39
3.55	ReadRAM64L	op.kód: 0x30 (pouze CA4)	40
3.56	WriteRAM64L	op.kód: 0x31 (pouze CA4)	40
3.57	SetClrRAMBitMask	op.kód: 0x33 (pouze CA4)	41
3.58	ReadEEByte	op.kód: 0x4C	41
3.59	WriteEEByte	op.kód: 0x4D	42
3.60	SetRAMBitMask	op.kód: 0x52	42
3.61	ClrRAMBitMask	op.kód: 0x53	42
3.62	ServerBusy	op.kód: 0x6E	44
4	Příloha A – Seznam příkazů		45
5	Příloha B – seznam chybových hlášení		48
6	Příloha C – předávané struktury		49

Změny dokumentace:

05.2009:

zaveden nový rámec ReadRAMbit a WriteRAMbit

dokument doplněn o přílohu C – předávané datové struktury u složitějších příkazů EPNP

10.2009:

popsány časové odezvy protokolu a doporučený algoritmus komunikace EPNP (1.2, 1.3)

popsán asynchronní, nevyžádaný rámec ServerBusy (3.55)

03.2015

pouze pro CA4 - zavedeny nové blokové rámce pro čtení a zápis do paměti automatů pomocí 32bitové adresy (týká se automatů MPC400, MT424, MT7)

viz: od 3.51 ReadRAM64B, ReadRAM64W, ReadRAM64L

a WriteRAM64B, WriteRAM64W, WriteRAM64L

02.2016

pouze pro CA4 - zavedena nová operace SetClrRAMBitMask (kód 0x33) nastavení / smazání bitů v rozšířené paměti pomocí 32-bitové adresy a 8-bitové masky (týká se automatů MPC400, MT424, MT7)

1 Co je EPNP?

EPNP (Encapsulated PesNet Protocol) je textový komunikační protokol umožňující komunikaci se sítí PesNet přes komunikační převodníky firmy MICROPEL. V současné době tento protokol používají převodníky CA21 (USB), CA3 (RS232 nebo GSM) a CA4 (Ethernet / Internet).

Základní vlastnosti:

- α **Přístup do sítě PLC MICROPEL** – sada příkazů protokolu EPNP poskytuje kompletní množinu služeb, nástrojů a funkcí pro pohodlnou správu a monitorování stanic v síti.
- α **Proměnná délka rámců** – délka rámce protokolu EPNP je omezena pouze kapacitou paměti použitého převodníku (typicky je to 1024B) a umožňuje tak vysílání, resp. příjem velkého počtu požadavků, resp. odpovědí v jednom rámci.

1.1 Komunikace protokolem EPNP

Komunikátory CA21, CA3 a CA4 se liší ve způsobu, kterým se navazuje spojení s protější stranou, což je dáno různým typem rozhraní (viz uživatelské manuály jednotlivých komunikátorů). Po ustanovení tohoto spojení je již možné komunikovat protokolem EPNP nezávisle na typu komunikátoru. Každý z převodníků je schopen přijmout libovolné množství rámců protokolu EPNP až do celkové velikosti 8KB. Jeden rámec přitom může mít délku maximálně 1024B včetně řídicích znaků a kontrolní sumy a obsahuje N požadavků (čtení, zápis apod). Poté co komunikátor dekóduje jednotlivé požadavky v rámcích, začne je vykonávat a odesílat odpovědi na ně opět v rámcích a to ve stejném pořadí v jakém tyto požadavky přišly do komunikátoru. Komunikátor se vždy snaží vytvářet rámce s odpověďmi o co největší délce. Rámce ukončuje kontrolní sumou a odesílá pokud již nemá žádné požadavky ke zpracování, pokud rámec dosáhl své maximální délky nebo pokud se protahuje zpracování dalšího požadavku.

1.2 Časové odezvy komunikátoru při komunikaci EPNP

Požadavky protokolu EPNP se rozdělují na dvě základní skupiny. První skupina nevyžaduje přístup komunikátoru do sítě automatů a proto je na ně odezva okamžitá (například požadavek GetServerInfo, LogIn apod.). Druhá skupina požadavků již přistupuje na zvolený automat nebo periférii. Tedy odezva na požadavky z této druhé a početnější skupiny záleží již není okamžitá, ale záleží na několika faktorech. Především na složitosti daného požadavku (například vyčtení informace o automatu je komplexnější a časově náročnější požadavek než třeba čtení bytu z paměti automatu). Dále pak na komunikační rychlosti sítě a také na kvalitě sítě. Horší kvalita sítě znamená vyšší chybovost komunikace mezi automaty a tedy nutnost častějšího opakování dotazu (příkazu) než je celkově požadavek EPNP vyřízen.

V případě dlouhého zpracování požadavku na straně komunikátoru (více jak 1000ms), informuje komunikátor o své činnosti klienta (tedy toho kdo právě zpracováváný požadavek zaslal) posláním tzv. asynchronního - nevyžádaného příkazu (viz příkaz ServerBusy). Tento příkaz je zasílán klientovi každých 1000ms do doby, než je původní požadavek klienta zpracován a odeslán klientovi (ať už úspěšně, nebo s chybou). Počet zaslaných asynchronních rámců ServerBusy není omezen a záleží na časové náročnosti zpracování požadavku EPNP a na počtu opakování v komunikátoru v případě poruchy komunikace na síti automatů. Platí, že do 20s musí přijít odpověď na původní požadavek EPNP.

1.3 Algoritmus komunikace EPNP

Vzhledem k výše uvedeným pravidlům komunikace a časovým odezvám je doporučený následující algoritmus komunikace.

1. zaslat požadavek EPNP do komunikátoru (například čtení wordu)
2. nastavit v klientovi timeout pro příjem odpovědi na 1500ms
- 3 čekat na odpověď EPNP, na asynchronní příkaz ServerBusy nebo na timeout
 - i) vyprší-li timeout, tak vyhlásit chybu převodníku (odpojený kabel ap.)
 - ii) pokud je přijata odpověď na požadavek EPNP (v našem příkladě čtení wordu), je možné poslat další požadavek EPNP
 - iii) pokud je přijat rámeček ServerBusy tak se presunout do bodu 2.

Pozn.: pro zabezpečení proti případné chybě firmwaru komunikátoru je možné nastavit ještě celkový ochranný timeout pro vyřízení každého požadavku EPNP na 20s (20.000ms)

2 Struktura rámců

Rámce protokolu EPNP jsou tvořeny požadavky na stanice nebo komunikační převodník, resp. odpověďmi na tyto požadavky. Počet těchto požadavků nebo odpovědí na ně v jednom rámci je omezen maximální délkou rámce, která je ve všech typech převodníků nastavena na 1024 B. Struktura rámce je následující (n je celkový počet požadavků v rámci):

Operátor	Data	$(n - 1) * (\text{Operátor} + \text{Data})$	Suma	Terminátor
----------	------	---	------	------------

Význam jednotlivých polí:

- Operátor – uvozuje požadavek, může být těchto typů:
 - Operátor Příkaz – v rámci požadavků uvozuje data požadavku, jež se má vykonat. V rámci odpovědi na požadavky uvozuje data těchto odpovědí. V rámci s požadavky je

reprezentován znakem '*'. V rámci s odpověďmi nabývá dvou hodnot. Znak '*' značí úspěšně vykonaný požadavek a uvozuje data odpovědi, kdežto znak '!' znamená, že požadavek nemohl být z nějakého důvodu vykonán, nebo nastala chyba a uvozuje chybové hlášení.

- α Operátor Změna adresy – určuje adresu stanice v síti PesNet, pro niž jsou určeny všechny následující příkazy až do další změny adresy nebo do konce rámce. Tento operátor je reprezentován znakem '@' a následující datové pole sestává z bajtu s požadovanou adresou.
- α Data – obsahuje data operátoru, jejich délka je závislá na typu operátoru. První bajt tohoto pole je identifikátor příkazu. Jednotlivé bajty se reprezentují hexadecimálně pomocí ASCII znaků '0' - '9', 'A' - 'F' nebo 'a' - 'f'. Každý bajt je reprezentován dvojicí těchto znaků, např. číslo 171 decimálně (0xAB hexadecimálně) je reprezentováno dvojicí znaků 'A' a 'B'. Pořadí bajtů v širších datových typech je typu Big-Endian, tzn. MSB bajt je přenášen jako první, LSB jako poslední, např. hodnotě 512 (dekadicky) odpovídá 0200.
- α Suma – obsahuje kontrolní součet celého rámce (modulo 256) přes všechny ostatní znaky rámce. Toto pole je reprezentováno pomocí 3 znaků (znak '#' a dva znaky z rozsahu '0' - '9', 'A' - 'F' nebo 'a' - 'f' reprezentující jednobajtovou hodnotu součtu)
- α Terminátor – ukončuje rámec. Reprezentuje se pomocí ASCII znaku CR (0x0D).

Příklad1

Pro vyčtení bloku dvou síťových wordů D32 a D33 je třeba poslat rámec:

***160220#55<CR>**

význam:

- α * - operátor příkaz
- α **16** – číslo požadavku (vyčtení bloku síťových wordů)
- α **0220** – data požadavku, **02** je počet žádaných wordů a **20** je index prvního wordu
- α **#55** – kontrolní součet MODULO256 všech ostatních znaků v rámci v rámci, tzn.:
 $0x2A (*) + 0x31 (1) + 0x36 (6) + 0x30 (0) + 0x32 (2) + 0x32 (2) + 0x30 (0) = 0x155$
 $MODULO256 (0x155) = 0x55$
- α **<CR>** - terminátor rámce, bajt 0x0D

Binární podoba rámce:

0x2A	0x31	0x36	0x30	0x32	0x32	0x30	0x23
0x35	0x35	0x0D					

Komunikátor vrátí následující rámec v případě že operace proběhla bez chyb:

***16022012345678#F9<CR>**

význam:

- α * - operátor příkaz
- α **16** – číslo požadavku (vyčtení bloku síťových wordů)
- α **022012345678** – data požadavku, **02** je počet žádaných wordů, **20** je index prvního wordu ($0x20 = 32$), **1234** hodnota prvního wordu (dekadicky 4660), **5678** je hodnota druhého wordu (dekadicky 22136).
- α **#F9** – kontrolní součet MODULO256 všech ostatních znaků v rámci v rámci, tzn.:
 $0x2A (*) + 0x31 (1) + 0x36 (6) + 0x30 (0) + 0x32 (2) + 0x32 (2) + 0x30 (0) + 0x31 (1)$
 $+ 0x32 (2) + 0x33 (3) + 0x34 (4) + 0x35 (5) + 0x36 (6) + 0x37 (7) + 0x38 (8)$
 $= 0x2F9, \text{MODULO}256 (0x2F9) = 0xF9$
- α **<CR>** - terminátor rámce, bajt 0x0D

Binární podoba rámce:

0x2A	0x31	0x36	0x30	0x32	0x32	0x30	0x31
0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x23
0x46	0x39	0x0D					

Komunik

átor vrátí následující rámec v případě že operace proběhla s chybou:

!16022035#84<CR>

význam:

- α **!** - odpověď na požadavek, při jehož vykonávání došlo k chybě
- α **16** – číslo požadavku (vyčtení bloku síťových wordů)
- α **0220** – data odpovědi na požadavek, **02** je počet žádaných wordů, **20** je index prvního wordu
- α **35** – číslo chyby, v tomto případě značí, že k převodníku není připojena síť automatu
- α **#84** – kontrolní součet MODULO256 všech ostatních znaků v rámci v rámci, tzn.:
 $0x21 (*) + 0x31 (1) + 0x36 (6) + 0x30 (0) + 0x32 (2) + 0x32 (2) + 0x30 (0) + 0x33 (3)$
 $+ 0x35 (5) = 0x184, \text{MODULO}256 (0x184) = 0x84$
- α **<CR>** - terminátor rámce, bajt 0x0D

Binární podoba rámce:

0x21	0x31	0x36	0x30	0x32	0x32	0x30	0x33
0x35	0x23	0x38	0x34	0x0D			

Příklad2 (čtení ze zásobníku automatu)

Zásobník automatu je umístěn v paměti od adresy 0x1800(hex). Pro vyčtení dvou položek ze zásobníku, například Stack[1] a Stack[2] použijeme příkaz na čtení wordů (položka v zásobníku je dlouhá 2 bajty tk. Jeden word) a čtené adresy pak mají hodnotu 0x1802 a 0x1804, protože parametr adresy má bajtový význam. Pozn.: můžeme napsat vzorec pro výpočet adresy zásobníku takto: adresa = 0x1800 + 2 x index.

Výsledný rámec pro automat s adresou 3: @03*441802*441806#61<CR>

význam:

- α @ - operátor určení adresy stanice
- α 03 – adresa stanice (automat v síti PesNet s adresou 3)
- α * - operátor příkaz
- α 44 – číslo prvního příkazu (vyčtení wordu z paměti automatu)
- α 1802 – data příkazu, 1802 je bajtová adresa prvního požadovaného wordu
- α 44 - číslo druhého příkazu (vyčtení wordu z paměti automatu)
- α 1806 – data příkazu, 1806 je bajtová adresa druhého požadovaného wordu
- α #61 – kontrolní součet MODULO256 všech ostatních znaků v rámci v rámci, tzn.:
 $0x40 (@) + 0x30 (0) + 0x33 (3) + 0x2A (*) + 0x34 (4) + 0x34 (4) + 0x31 (1) + 0x38 (8) + 0x30 (0) + 0x32 (2) + 0x2A (*) + 0x34 (4) + 0x34 (4) + 0x31 (1) + 0x38 (8) + 0x30 (0) + 0x36 (6) = 0x361, \text{MODULO}256 (0x361) = 0x61$
- α <CR> - terminátor rámce, bajt 0x0D

Binární podoba rámce:

0x40	0x30	0x33	0x2A	0x34	0x34	0x31	0x38
0x30	0x32	0x2A	0x34	0x34	0x31	0x38	0x30
0x36	0x23	0x36	0x31	0x0D			

Komunikátor vrátí

následující rámec v případě že operace proběhla bez chyb:

@03*4418021234*4418065678#05<CR>

význam:

- α @ - operátor určení adresy stanice
- α 03 – adresa stanice (automat v síti PesNet s adresou 3)
- α * - operátor příkaz
- α 44 – číslo prvního příkazu (vyčtení wordu z paměti automatu)
- α 18021234 – data odpovědi, 1802 je bajtová adresa prvního požadovaného wordu, 1234 je jeho hodnota (4660 dekadicky)
- α 44 - číslo druhého příkazu (vyčtení wordu z paměti automatu)
- α 18065678 – data požadavku, 1806 je bajtová adresa druhého požadovaného wordu, 5678 je jeho hodnota (22136)
- α #05 – kontrolní součet MODULO256 všech ostatních znaků v rámci v rámci, tzn.:
 $0x40 (@) + 0x30 (0) + 0x33 (3) + 0x2A (*) + 0x34 (4) + 0x34 (4) + 0x31 (1) + 0x38 (8) + 0x30 (0) + 0x32 (2) + 0x31 (1) + 0x32 (2) + 0x33 (3) + 0x34 (4) + 0x2A (*) + 0x34 (4) + 0x34 (4) + 0x31 (1) + 0x38 (8) + 0x30 (0) + 0x36 (6) + 0x35 (5) + 0x36 (6) + 0x37 (7) + 0x38 (8) = 0x505, \text{MODULO}256 (0x505) = 0x05$

α <CR> - terminátor rámce, bajt 0x0D

Příklad 3: vestavěným administrátorským právem je „admindat“ – příkaz pro povolení čtení a zápisu všech paměťových oblastí je následující:

***0361646D696E64617400#5E**

odpověď :

***03#8D**

3 Seznam příkazů protokolu EPNP

Příkazy protokolu EPNP rozšiřují možnosti standardního PesNet protokolu definováním tzv. makropříkazů. Tyto makropříkazy umožňují např. výčet bloků paměti jediným požadavkem, což efektivně snižuje počet nutných operací a lépe využívá přenosového média (omezením počtu přenášených znaků).

V následujícím textu budou popsány všechny příkazy protokolu EPNP. Kromě slovního popisu funkce bude u každého příkazu uveden jeho operační kód (číslo příkazu), struktura, význam datového pole a jeho délka. Stejným způsobem bude vždy popsána odpověď na požadavek. V případě, že vykonávání požadavku skončilo chybou, obsahuje odpověď chybový kód o délce 1B. Význam tohoto kódu lze nalézt v příloze B.

Příkazy jsou rozděleny do skupin podle jejich zaměření a funkce. Souhrnný seznam příkazů se zkráceným popisem funkce a významu dat lze najít v tabulce v příloze A.

Pozn.: veškerá data jsou zde uvedena již po překladu z ASCII reprezentace (viz. kapitola 2), tzn. původní dvojice ASCII znaků je zde uvažována jako jeden bajt. Výjimku tvoří znaky operátorů, které jsou pro přehlednost ponechány v ASCII podobě (znaky '*', '@', '!' a '#').

3.1 GetServerInfo

op.kód: 0x01

Slouží k identifikaci komunikačního převodníku. Odpovědí je datová struktura ServerInfo obsahující informace o typu zařízení, verzi jeho firmware a jeho konfigurace.

Požadavek:

Operátor	Číslo příkazu
*	0x01

Odpověď:

Operátor	Číslo příkazu	Data
*	0x01	struktura ServerInfo

Chybová odpověď:
není definována

Obsah datového pole je tvořen datovou strukturou ServerInfo. Délka a složení této struktury se liší podle použitého komunikačního převodníku a verze jeho firmware.

Pro komunikátory CA21 a CA3, od verze firmware 1.060 má datová struktura délku 44B a její význam je následující:

Název položky	Délka	Typ dat	Popis	Příklad
InfoSize	2B	integer	délka struktury ServerInfo v bajtech	0x002C
FW version	8B	řetězec	číslo firmware komunikátoru	1.060
SerialNumber	8B	řetězec	sériové číslo komunikátoru	0B-2048
ServerName	8B	řetězec	jméno komunikátoru	CA_kotel
ConfigName	8B	řetězec	jméno zataženého souboru konfigurace	test.dnl
PesNetLoad	1B	byte	maximální zatížení sítě PesNet	0x40
PesNetAddress	1B	byte	adresa v síti PesNet	0x1E
PesType	7B	řetězec	typ zařízení podle PIO type	CA21
CACfg	1B	byte	vybrané bity z konfigurace komunikátoru	0x0C

Pro komunikátor CA4, verze firmware 1.060 má datová struktura délku 64B a význam je následující:

Název položky	Délka	Typ dat	Popis	Příklad
InfoSize	2B	integer	délka struktury ServerInfo v bajtech	0x0040
FW version	8B	řetězec	číslo firmware komunikátoru	1.060
SerialNumber	8B	řetězec	sériové číslo komunikátoru	0B-2048
ServerName	8B	řetězec	jméno komunikátoru	CA_kotel
ConfigName	8B	řetězec	jméno zataženého souboru konfigurace	test.dnl
PesNetLoad	1B	byte	maximální zatížení sítě PesNet	0x40
PesNetAddress	1B	byte	adresa v síti PesNet	0x1E
Pratype	7B	řetězec	typ zařízení podle PIO type	CA4
CACfg	1B	byte	vybrané bity z konfigurace	0x0C
MAC	6B	pole bajtů	MAC adresa komunikátoru	0x00,0x20,0x4A,0x93,0x45,0x00
IP	4B	pole bajtů	IP adresa komunikátoru	0xC0,0xA8,0x00,0x64
TCPport	2B	integer	TCP port komunikátoru	0x2711
GateWay	4B	pole bajtů	Výchozí brána komunikátoru	0x00,0x00,0x00,0x00
Mask	4B	pole bajtů	Maska podsítě komunikátoru	0xFF,0xFF,0xFF,0x00

3.2 LogIn

op.kód: 0x03

Slouží k autorizaci uživatele. Pokud komunikátor nalezne zadané jméno ve svém seznamu uživatelů, vrátí kladnou odpověď. V opačném případě vrátí chybové hlášení. Pokud komunikátor autorizaci nevyžaduje (závisí na jeho nastavení), vrací vždy kladnou odpověď. Vstupním parametrem je řetězec se jménem o délce 9B.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x03	jméno uživatele [9B]

Odpověď:

Operátor	Číslo příkazu
*	0x03

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x03	ErrorCode [1B]

Příklad: vestavěným administrátorským právem je „adminat“ – příkaz pro povolení čtení a zápisu všech paměťových oblastí je následující:

*0361646D696E64617400#5E

odpověď :

*03#8D

3.3 LogOut

op.kód: 0x04

Odhlásí aktuálně přihlášeného uživatele.

Požadavek:

Operátor	Číslo příkazu
*	0x04

Odpověď:

Operátor	Číslo příkazu
*	0x04

Chybová odpověď:

není definována

3.4 EraseConfig

op.kód: 0x09

Vymaže dočasnou konfigurační banku paměti typu FLASH určenou k uložení nové konfigurace serveru. Stávající konfigurace není tímto příkazem ovlivněna.

Požadavek:

Operátor	Číslo příkazu
*	0x09

Odpověď:

Operátor	Číslo příkazu
*	0x09

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x09	ErrorCode [1B]

3.5 FlashConfig

op.kód: 0x0A

Nahraje blok dat s novou konfigurací serveru do dočasné konfigurační banky. Vstupní data tvoří blok o velikosti 256B a číslo stránky v rozsahu 0 - 31, do které má být uložen.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x0A	číslo stránky [1B] + konfigurační pole [256B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x0A	číslo stránky [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x0A	číslo stránky [1B]	ErrorCode [1B]

3.6 UpdateConfig

op.kód: 0x0B

Aktivuje nově nahranou konfiguraci serveru. Tento příkaz musí být proveden po nahrání všech bloků konfigurace (pomocí příkazu FlashConfig). Vstupním parametrem je 16ti bitový kontrolní součet celé configurační banky, kterým se ověří již uložené bloky nové konfigurace a poté dojde k výměně stávající configurační banky za novou. Po této operaci server provede restart a inicializaci systému na základě nové konfigurace.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x0B	kontrolní součet [2B]

Odpověď:

Operátor	Číslo příkazu
*	0x0B

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x0B	ErrorCode [1B]

3.7 EraseFW

op.kód: 0x0C

Vymaže dočasnou banku paměti typu FLASH pro nový firmware. Stávající firmware není tímto příkazem ovlivněn.

Požadavek:

Operátor	Číslo příkazu
*	0x0C

Odpověď:

Operátor	Číslo příkazu
*	0x0C

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x0C	ErrorCode [1B]

3.8 FlashFW

op.kód: 0x0D

Nahráje blok dat nového firmwaru do dočasné banky. Vstupní data tvoří blok dat o velikosti 256B a číslo stránky v rozsahu 0 – 255, do které má být tento blok uložen.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x0D	číslo stránky [1B] + konfigurační pole [256B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x0D	číslo stránky [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x0D	číslo stránky [1B]	ErrorCode [1B]

3.9 UpdateFW

op.kód: 0x0E

Slouží k aktivaci nově nahraného firmwaru komunikátoru. Tento příkaz musí být proveden po nahrání všech bloků (pomocí příkazů FlashFW). Vstupním parametrem je 16ti bitový kontrolní součet, kterým se ověří již uložené bloky nového firmwaru. Poté dojde k výměně stávajícího firmwaru za nový a restartu celého zařízení.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x0E	kontrolní součet [2B]

Odpověď:

Operátor	Číslo příkazu
*	0x0E

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x0E	ErrorCode [1B]

3.10 ServerReset

op.kód: 0x0F

Resetuje komunikační převodník, přičemž dojde k přerušení spojení. K resetu dojde až po odeslání odpovědi.

Požadavek:

Operátor	Číslo příkazu
*	0x0F

Odpověď:

Operátor	Číslo příkazu
*	0x0F

Chybová odpověď:

není definována

3.11 ServerRestart

op.kód: 0x10

Provede restart serveru a jeho inicializaci. Subsystem spravující chod v síti PesNet není tímto příkazem ovlivněn a příjem dat ze sítě PLC zůstane zachován.

Požadavek:

Operátor	Číslo příkazu
*	0x10

Odpověď:

Operátor	Číslo příkazu
*	0x10

Chybová odpověď:

není definována

3.12 GetPLCList

op.kód: 0x05

Vrátí seznam aktivních stanic v síti. Nekomunikující, nebo nefunkční stanice v seznamu nejsou obsaženy.

Požadavek:

Operátor	Číslo příkazu
*	0x05

Odpověď:

Operátor	Číslo příkazu	Data
*	0x05	pole bajtů [31B]

Chybová odpověď:

není definována

Data odpovědi tvoří pole bajtů reprezentující 31 možných adres stanic v síti. Nenulová hodnota značí přítomnost stanice jejíž adresa je rovná indexu tohoto bajtu v poli. Význam bajtů je následující:

0x00 – na této adrese (adresa = index tohoto bajtu v seznamu) není aktivní stanice

0x01 – značí stanici komunikující rychlostí 57600Bd

0x02 - značí stanici komunikující rychlostí 19200Bd

0x04 - značí stanici komunikující rychlostí 9600Bd

0x08 - značí stanici komunikující rychlostí 2400Bd

3.13 GetPLCHardList

op.kód: 0x06

Spustí nástroj prohledání stanic v síti PesNet, který umožňuje nalézt i nekomunikující stanice (například z důvodu nesprávně nastavené komunikační rychlosti). Nástroj ruší po dobu detekce stanic běžný provoz v síti.

Požadavek:

Operátor	Číslo příkazu
*	0x06

Odpověď:

Operátor	Číslo příkazu	Data
*	0x06	pole bajtů [31B]

Chybová odpověď:
není definována

Data odpovědi tvoří pole bajtů reprezentující 31 možných adres stanic v síti. Nenulová hodnota značí přítomnost stanice nebo stanic jejichž adresa je rovná indexu tohoto bajtu v poli. Pokud je v bajtu nastaveno více bitů, je na dané adrese více stanic a tudíž kolize. Pořadí nastaveného bitu v bajtu značí rychlost stanice:

všechny bity nulové – na dané adrese není žádná stanice

nastavený bit 0 - značí stanici komunikující rychlostí 57600Bd

nastavený bit 1 - značí stanici komunikující rychlostí 19200Bd

nastavený bit 2 - značí stanici komunikující rychlostí 9600Bd

nastavený bit 3 - značí stanici komunikující rychlostí 2400Bd

3.14 GetPLCInfo

op.kód: 0x21

Vrátí strukturu *PLC_Info* s informacemi o konkrétní stanici. Adresa stanice, ke které se příkaz vztahuje je určena operátorem změna adresy.

Struktura PLC_info viz seznam předávaných struktur – Příloha C.

Požadavek:

Operátor	Číslo příkazu
*	0x21

Odpověď:

Operátor	Číslo příkazu	Data
*	0x21	struktura PLC_Info

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x21	ErrorCode [1B]

3.15 PrgCluster

op.kód: 0x22

Provede zatažení bloku dat s uživatelským programem do stanice. Vstupní data tvoří blok dat dlouhý 256B a adresa sektoru ve stanici, na kterou má být blok umístěn.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x22	sektor [2B] + data [256B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x22	sektor [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x22	sektor [2B]	ErrorCode [1B]

3.16 DnlFinish

op.kód: 0x23

Provede aktivaci nově zataženého uživatelského programu ve stanici. Vstupní data tvoří struktura *Dnl_Desc* s informacemi o programu, kterou se ověří správnost právě zataženého programu a poté případně dojde k aktivaci nového programu.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x23	Dnl_Desc [28B]

Odpověď:

Operátor	Číslo příkazu
*	0x23

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x23	ErrorCode [1B]

3.17 PLCRun

op.kód: 0x24

Spustí uživatelský program v stanici.

Požadavek:

Operátor	Číslo příkazu
*	0x24

Odpověď:

Operátor	Číslo příkazu
*	0x24

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x24	ErrorCode [1B]

3.18 PLCStop

op.kód: 0x25

Zastaví uživatelský program v stanici.

Požadavek:

Operátor	Číslo příkazu
*	0x25

Odpověď:

Operátor	Číslo příkazu
*	0x25

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x25	ErrorCode [1B]

3.19 PLCReboot

op.kód: 0x26

Provede reboot stanice.

Požadavek:

Operátor	Číslo příkazu
*	0x26

Odpověď:

Operátor	Číslo příkazu
*	0x26

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x26	ErrorCode [1B]

3.20 PLCSetTime

op.kód: 0x27

Nastaví reálný čas v stanici. Vstupní data tvoří datová struktura *PLC_Time* s požadovaným časem.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x27	PLC_Time[3B]

Odpověď:

Operátor	Číslo příkazu
*	0x27

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x27	ErrorCode [1B]

Struktura PLC_Time:

Název položky	Délka	Typ dat
Hour	1B	bajt
Min	1B	bajt
Sec	1B	bajt

3.21 PLCSetDate

op.kód: 0x28

Nastaví datum v stanici. Vstupní data tvoří datová struktura *PLC_Date*.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x28	PLC_Date[4B]

Odpověď:

Operátor	Číslo příkazu
*	0x28

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x28	ErrorCode [1B]

Struktura PLC_Date:

Název položky	Délka	Typ dat
Year	1B	bajt
Month	1B	bajt
Day	1B	bajt
Week	1B	bajt

3.22 PLCRebootFreeze

op.kód: 0x29

Provede reboot automatu s tím, že se automat spustí do speciálního módu BOOT (používá se při programování automatu firmvarem).

Požadavek:

Operátor	Číslo příkazu
*	0x29

Odpověď:

Operátor	Číslo příkazu
*	0x29

Chybová odpověď:

Operátor	Číslo příkazu	Chybové hlášení
!	0x29	ErrorCode [1B]

3.23 PLCSetSpeed op.kód: 0x12

Nastaví všem stanicím v síti stejnou komunikace rychlost. Vstupním parametrem je požadovaná rychlost. Tento příkaz po dobu zpracování ruší provoz na síti.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x12	rychlost [1B]

Odpověď:

Operátor	Číslo příkazu
*	0x12

Chybová odpověď:

není definována

Význam bajtu rychlost: **0** – 57600Bd, **1** – 19200Bd, **2** – 9600Bd, **3** – 2400Bd

3.24 GetRxWords op.kód: 0x13

Vrátí hodnoty síťových proměnných typu word (D32 – D63), které byly serverem od jeho posledního resetu přijaty. Vstupními parametry je index prvního požadovaného wordu a počet.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x13	počet [1B] + počáteční index[1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x13	N [1B] + SI [1B] + N * (index [1B] + hodnota [2B])

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x13	počet [1B] + počáteční index[1B]	ErrorCode [1B]

kde N je počet wordů, které byly komunikátorem přijaty ze sítě,

SI je počáteční index původního dotazu

položky počáteční index a index mohou nabývat hodnot 0x20 – 0x3F (32 – 63)

3.25 GetRxBits

op.kód: 0x14

Vrátí hodnoty síťových proměnných typu bit (M64 –M127), které byly serverem od jeho posledního resetu přijaty. Vstupními parametry je index prvního požadovaného bitu a počet.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x14	počet [1B] + počáteční index[1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x14	N [1B] + SI [1B] + N * (index [1B] + hodnota [1B])

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x14	počet [1B] + počáteční index[1B]	ErrorCode [1B]

kde N je počet bitů, které byly komunikátorem přijaty ze sítě,

SI je počáteční index původního dotazu

položky počáteční index a index mohou nabývat hodnot 0x40 – 0x7F (64 – 127)

3.26 GetRxLongs

op.kód: 0x15

Vrátí hodnoty síťových proměnných typu long (LW0 – LW255), které byly serverem od jeho posledního resetu přijaty. Vstupními parametry je index prvního požadovaného bitu a počet.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x15	počet [1B] + počáteční index[1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x15	N [1B] + SI [1B] + N * (index [1B] + hodnota [4B])

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x15	počet [1B] + počáteční index[1B]	ErrorCode [1B]

kde N je počet bitů, které byly komunikátorem přijaty ze sítě,

SI je počáteční index původního dotazu

položky počáteční index a index mohou nabývat hodnot 0x00 – 0xFF (0 – 255)

3.27 ReadNetWords

op.kód: 0x16

Vrátí blok síťových proměnných typu word (D32 – D63). Vstupní parametrem je počáteční index a počet položek.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x16	počet [1B] + index[1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x16	počet [1B] + index [1B] + počet * hodnota [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x16	počet [1B] + index[1B]	ErrorCode [1B]

položka index může nabývat hodnot 0x20 – 0x3F (32 – 63)

3.28 WriteNetWords

op.kód: 0x17

Zapíše blok síťových proměnných typu word (D32 – D63). Vstupní data tvoří seznam nových hodnot, jež mají být zapsány a index prvního zapisovaného wordu.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x17	počet [1B] + index [1B] + počet * hodnota [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x17	počet [1B] + index [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x17	počet [1B] + index[1B]	ErrorCode [1B]

položka index může nabývat hodnot 0x20 – 0x3F (32 – 63)

3.29 ReadNetBits

op.kód: 0x18

Vrátí blok síťových proměnných typu bit (M64 – M127). Vstupní parametrem je počáteční index a počet položek.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x18	počet [1B] + index[1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x18	počet [1B] + index [1B] + počet * hodnota [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x18	počet [1B] + index[1B]	ErrorCode [1B]

položka index může nabývat hodnot 0x40 – 0x7F (64 – 127)

3.30 WriteNetBits

op.kód:0x19

Zapíše blok síťových proměnných typu bit (M64 – M127). Vstupní data tvoří seznam nových hodnot, jež mají být zapsány a index prvního zapisovaného bit.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x19	počet [1B] + index [1B] + počet * hodnota [1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x19	počet [1B] + index [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x19	počet [1B] + index[1B]	ErrorCode [1B]

položka index může nabývat hodnot 0x40 – 0x7F (64 – 127)

3.31 ReadNetLongs

op.kód: 0x1A

Vrátí blok síťových proměnných typu long (LW0 – LW255). Vstupní parametrem je počáteční index a počet položek.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x1A	počet [1B] + index[1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x1A	počet [1B] + index [1B] + počet * hodnota [4B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x1A	počet [1B] + index[1B]	ErrorCode [1B]

položka index může nabývat hodnot 0x00 – 0xFF (0 – 255)

3.32 WriteNetLongs

op.kód: 0x1B

Zapíše blok síťových proměnných typu long (LW0 – LW255). Vstupní data tvoří seznam nových hodnot, jež mají být zapsány a index prvního zapisovaného long.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x1B	počet [1B] + index [1B] + počet * hodnota [4B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x1B	počet [1B] + index [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x1B	počet [1B] + index[1B]	ErrorCode [1B]

položka index může nabývat hodnot 0x00 – 0xFF (0 – 255)

3.33 ReadSTPWord

op.kód: 0x4E

Přečte systémový word stanice. Vstupem je jednobajtový index.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x4E	index [1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x4E	index [1B] + data [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x4E	index[1B]	ErrorCode [1B]

položka index může nabývat hodnot 0x00 – 0xFF (0 – 255)

3.34 WriteSTPWord

op.kód: 0x4F

Zapíše systémový word stanice. Vstupem je jednobajtový index a zapisovaná hodnota.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x4F	index [1B] + data [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x4F	index [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x4F	index[1B]	ErrorCode [1B]

položka index může nabývat hodnot 0x00 – 0xFF (0 – 255)

3.35 ReadSTPBit

op.kód: 0x50

Přečte systémový bit stanice. Vstupem je dvoubajtový index.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x50	index [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x50	index [2B] + data [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x50	index[2B]	ErrorCode [1B]

položka index může nabývat hodnot 0x00 – 0x13F (0 – 319)

3.36 WriteSTPBit

op.kód: 0x51

Zapíše systémový word stanice. Vstupem je dvoubajtový index a zapisovaná hodnota.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x51	index [2B] + data [1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x51	index [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x51	index[2B]	ErrorCode [1B]

položka index může nabývat hodnot 0x00 – 0x13F (0 – 319)

3.37 ReadRAMBit

op.kód: 0x2A

Přečte bit z paměti PLC. Vstupním parametrem je 16ti bitová adresa bytu (kde se nachází požadovaný bit) a osmibitová maska (1-128) určující pozici bitu v daném bytu.

Požadavek:

Operátor	Číslo příkazu	Data	
*	0x2A	adresa [2B]	maska[1B]

Odpověď:

Operátor	Číslo příkazu	Odpověď
*	0x2A	adresa [2B] +maska[1B] + data [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Odpověď	Chybové hlášení
!	0x2A	adresa [2B] + maska[1B]	ErrorCode [1B]

3.38 WriteRAMBit

op.kód: 0x2B

Příkaz nastaví/vynuluje bit v paměti automatu. Tento bit je určen 16-ti bitovou adresou a osmibitovou maskou. Bit se nastaví na jedničku pokud jsou předávaná data nenulová. Pokud jsou předávaná data nulová, bit se nastaví na nulu.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x2B	adresa [2B] + maska[1B] + data [1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x2B	adresa [2B] + maska[1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x2B	adresa [2B] + maska [1B]	ErrorCode [1B]

3.39 ReadRAMByte

op.kód: 0x40

Přečte bajt z paměti PLC. Vstupním parametrem je 16ti bitová adresa, výstupním pak hodnota.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x40	adresa [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x40	adresa [2B] + data [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x40	adresa [2B]	ErrorCode [1B]

3.40 WriteRAMByte

op.kód: 0x41

Zapíše bajt do paměti PLC. Vstupními parametry je 16ti bitová adresa a hodnota.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x41	adresa [2B] + data [1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x41	adresa [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x41	adresa [2B]	ErrorCode [1B]

3.41 ReadRAMWord

op.kód: 0x44

Přečte word z paměti PLC. Vstupním parametrem je 16ti bitová adresa, výstupním pak hodnota.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x44	adresa [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x44	adresa [2B] + data [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x44	adresa [2B]	ErrorCode [1B]

3.42 WriteRAMWord

op.kód: 0x45

Zapíše word do paměti PLC. Vstupními parametry je 16ti bitová adresa a hodnota.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x45	adresa [2B] + data [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x45	adresa [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x45	adresa [2B]	ErrorCode [1B]

3.43 ReadRAMLong

op.kód: 0x48

Přečte long z paměti PLC. Vstupním parametrem je 16ti bitová adresa, výstupním pak hodnota.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x48	adresa [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x48	adresa [2B] + data [4B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x48	adresa [2B]	ErrorCode [1B]

3.44 WriteRAMLong

op.kód: 0x49

Zapíše long do paměti PLC. Vstupními parametry je 16ti bitová adresa a hodnota.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x49	adresa [2B] + data [4B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x49	adresa [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x49	adresa [2B]	ErrorCode [1B]

3.45 ReadRAMByteBlock

op.kód: 0x42

Přečte blok položek typu bajt z paměti PLC. Vstupním parametrem je počet položek a počáteční 16ti bitová adresa, výstupním pak hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x42	počet [1B] + adresa [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x42	počet [1B] + adresa [2B] + počet * data [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x42	počet [1B] + adresa [2B]	ErrorCode [1B]

3.46 WriteRAMByteBlock

op.kód: 0x43

Zapíše blok položek typu bajt do paměti PLC. Vstupním parametrem je počet položek, počáteční 16ti bitová adresa a hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x43	počet [1B] + adresa [2B] + počet * data [1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x43	počet [1B] + adresa [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x43	počet [1B] + adresa [2B]	ErrorCode [1B]

3.47 ReadRAMWordBlock

op.kód: 0x46

Přečte blok položek typu word z paměti PLC. Vstupním parametrem je počet položek a počáteční 16ti bitová adresa, výstupním pak hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x46	počet [1B] + adresa [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x46	počet [1B] + adresa [2B] + počet * data [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x46	počet [1B] + adresa [2B]	ErrorCode [1B]

3.48 WriteRAMWordBlock

op.kód: 0x47

Zapíše blok položek typu word do paměti PLC. Vstupním parametrem je počet položek, počáteční 16ti bitová adresa a hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x47	počet [1B] + adresa [2B] + počet * data [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x47	počet [1B] + adresa [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x47	počet [1B] + adresa [2B]	ErrorCode [1B]

3.49 ReadRAMLongBlock op.kód: 0x4A

Přečte blok položek typu long z paměti PLC. Vstupním parametrem je počet položek a počáteční 16ti bitová adresa, výstupním pak hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x4A	počet [1B] + adresa [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x4A	počet [1B] + adresa [2B] + počet * data [4B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x4A	počet [1B] + adresa [2B]	ErrorCode [1B]

3.50 WriteRAMLongBlock op.kód: 0x4B

Zapíše blok položek typu long do paměti PLC. Vstupním parametrem je počet položek, počáteční 16ti bitová adresa a hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x4B	počet [1B] + adresa [2B] + počet * data [4B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x4B	počet [1B] + adresa [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x4B	počet [1B] + adresa [2B]	ErrorCode [1B]

3.51 ReadRAM64B op.kód: 0x2C (pouze CA4)

Přečte blok až 64 položek typu byte z paměti PLC pomocí 32bitové adresy. Vstupním parametrem je počet položek a počáteční 32 bitová adresa, výstupním pak hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x2C	počet [1B] + adresa [4B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x2C	počet [1B] + adresa [4B] + počet * data [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x2C	počet [1B] + adresa [4B]	ErrorCode [1B]

3.52 WriteRAM64B op.kód: 0x2D (pouze CA4)

Zapíše blok až 64 položek typu byte do paměti PLC. Vstupním parametrem je počet položek, počáteční 32 bitová adresa a hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x2D	počet [1B] + adresa [4B] + počet * data [1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x2D	počet [1B] + adresa [4B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x2D	počet [1B] + adresa [4B]	ErrorCode [1B]

3.53 ReadRAM64W

op.kód: 0x2E (pouze CA4)

Přečte blok až 64 položek typu word z paměti PLC. Vstupním parametrem je počet položek a počáteční 32 bitová adresa, výstupním pak hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x2E	počet [1B] + adresa [4B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x2E	počet [1B] + adresa [4B] + počet * data [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x2E	počet [1B] + adresa [4B]	ErrorCode [1B]

3.54 WriteRAM64W

op.kód: 0x2F (pouze CA4)

Zapíše blok až 64 položek typu word do paměti PLC. Vstupním parametrem je počet položek, počáteční 32 bitová adresa a hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x2F	počet [1B] + adresa [4B] + počet * data [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x2F	počet [1B] + adresa [4B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x2F	počet [1B] + adresa [4B]	ErrorCode [1B]

3.55 ReadRAM64L op.kód: 0x30 (pouze CA4)

Přečte blok až 64 položek typu long z paměti PLC. Vstupním parametrem je počet položek a počáteční 32ti bitová adresa, výstupním pak hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x30	počet [1B] + adresa [4B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x30	počet [1B] + adresa [4B] + počet * data [4B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x30	počet [1B] + adresa [4B]	ErrorCode [1B]

3.56 WriteRAM64L op.kód: 0x31 (pouze CA4)

Zapíše blok až 64 položek typu long do paměti PLC. Vstupním parametrem je počet položek, počáteční 32ti bitová adresa a hodnoty.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x31	počet [1B] + adresa [4B] + počet * data [4B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x31	počet [1B] + adresa [4B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x31	počet [1B] + adresa [4B]	ErrorCode [1B]

3.57 SetClrRAMBitMask op.kód: 0x33 (pouze CA4)

Nastaví/smaže bity v určeném bajtu v paměti automatu. Vstupem je 32ti bitová adresa bajtu, 8-bitová maska bitů a hodnota přepínače setclr, jehož nenulová hodnota určuje, že se bity nastavují. Pokud je hodnota přepínače setclr nulová, tak se maskou vybrané bity mažou.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x33	adresa [4B] + maska [1B] + setclr[1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x33	adresa [4B] + maska [1B] + setclr[1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x33	počet [1B] + adresa [4B]	ErrorCode [1B]

3.58 ReadEEByte

op.kód: 0x4C

Přečte bajt z EEPROM paměti automatu. Vstupem je 16ti bitová adresa, výstupem jednobajtová hodnota.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x4C	adresa [2B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x4C	adresa [2B] + data [1B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x4C	adresa [2B]	ErrorCode [1B]

3.59 WriteEByte

op.kód: 0x4D

Zapíše bajt do EEPROM paměti automatu. Vstupem je 16ti bitová adresa.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x4D	adresa [2B] + data [1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x4D	adresa [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x4D	adresa [2B]	ErrorCode [1B]

3.60 SetRAMBitMask

op.kód: 0x52

Nastaví bity v rámci bajtu v paměti automatu. Vstupem je 16ti bitová adresa a maska bitů, jež se mají nastavit.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x52	adresa [2B] + maska [1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x52	adresa [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x52	adresa [2B]	ErrorCode [1B]

3.61 ClrRAMBitMask

op.kód: 0x53

Vynuluje bity v rámci bajtu v paměti automatu. Vstupem je 16ti bitová adresa a maska bitů, jež se mají vynulovat.

Požadavek:

Operátor	Číslo příkazu	Data
*	0x53	adresa [2B] + maska [1B]

Odpověď:

Operátor	Číslo příkazu	Data
*	0x53	adresa [2B]

Chybová odpověď:

Operátor	Číslo příkazu	Data	Chybové hlášení
!	0x53	adresa [2B]	ErrorCode [1B]

3.62 ServerBusy

op.kód: 0x6E

Komunikátor sám zasílá klientovy tento požadavek jako nevyžádanou odpověď v případě, že mu zpracování předchozího požadavku EPNP trvá déle než 1s a tento požadavek opakuje dokud není předchozí příkaz EPNP zpracován (úspěšně nebo s chybou). Součástí příkazu, tedy spíše odpovědi jsou dva bajty, které mají nemají uživatelský význam.

Požadavek:

Tento příkaz je zasílán jako nevyžádaný – nemá část „požadavek“

Odpověď:

Operátor	Číslo příkazu	Data
*	0x6E	data [2B]

Chybová odpověď:

Tento příkaz je zasílán jako nevyžádaný – nemá chybovou odpověď“

4 Příloha A – Seznam příkazů

Následující seznam shrnuje příkazy protokolu EPNP. Ve sloupcích vstup a výstup jsou uvedeny vstupní a výstupní parametry příkazu. V hranatých závorkách je vždy délka parametru v bajtech. Sloupec A značí, zda-li příkaz vyžaduje adresu PLC stanice nastavenou pomocí operátoru změna adresy (a zároveň rozděluje příkazy podle okamžité odezvy a delší odezvy).

ID hex	ID dec	Název požadavku	Popis	Vstup	Výstup	A
0x01	1	GetServerInfo	vrátí strukturu ServerInfo	-	ServerInfo	
0x03	3	LogIn	autorizuje uživatele	uživ. jméno [8B]	-	
0x04	4	LogOut	odhlásí uživatele			
0x05	5	GetPLCList	vrátí seznam stanic	-	PLC list [31B]	
0x06	6	GetPLCHardList	vrátí seznam stanic (i nekomunikujících)	-	PLC list [31B]	
0x09	9	EraseConfig	vymaže dočasnou banku pro konfiguraci	-	-	
0x0A	10	FlashConfig	nahráže blok konfigurace do dočasné banky	stránka [1B] + data [256B]	stránka [1B]	
0x0B	11	UpdateConfig	zkontroluje platnost nové konfigurace a případně ji nastaví	crc [2B]	crc [2B]	
0x0C	12	EraseFW	vymaže dočasnou banku pro firmware	-	-	
0x0D	13	FlashFW	nahráže blok firmware do dočasné banky	stránka [1B]+ data [256B]	stránka [1B]	
0x0E	14	UpdateFW	zkontroluje platnost nového firmware a případně ji nastaví	crc [2B]	crc [2B]	
0x0F	15	ServerReset	provede reset komunikátoru	-	-	
0x10	16	ServerRestart	provede restart komunikátoru	-	-	
0x12	18	PLCSetSpeed	nastaví všem PLC v síti rychlost komunikace	rychlost [1B]	rychlost [1B]	
0x13	19	GetRxWords	vrátí seznam a hodnoty síťových proměnných typu word, které byly serverem od jeho posledního resetu přijaty	počet [1B], index [1B]	reálný počet [1B], index [1B], reálný počet * data [2B]	
0x14	20	GetRxBits	vrátí seznam a hodnoty síťových proměnných typu bit, které byly serverem od jeho posledního resetu přijaty	počet [1B], index [1B]	reálný počet [1B], index [1B], reálný počet * data [1B]	

ID hex	ID dec	Název požadavku	Popis	Vstup	Výstup	A
0x15	21	GetRxLongs	vrátí seznam a hodnoty síťových proměnných typu long, které byly serverem od jeho posledního resetu přijaty	počet [1B], index [1B]	reálný počet [1B], index [1B], reálný počet * data [4B]	
0x16	22	ReadNetWords	vrátí blok síťových proměnných typu word	počet [1B], index [1B]	počet [1B], index [1B], počet * data [2B]	
0x17	23	WriteNetWords	zapiše blok síťových proměnných typu word	počet [1B], index [1B], počet * data [2B]	počet [1B], index [1B]	
0x18	24	ReadNetBits	vrátí blok síťových proměnných typu bit	počet [1B], index [1B]	počet [1B], index [1B], počet * data [1B]	
0x19	25	WriteNetBits	zapiše blok síťových proměnných typu bit	počet [1B], index [1B], počet * data [1B]	počet [1B], index [1B]	
0x1A	26	ReadNetLongs	vrátí blok síťových proměnných typu long	počet [1B], index [1B]	počet [1B], index [1B], počet * data [4B]	
0x1B	27	WriteNetLongs	zapiše blok síťových proměnných typu long	počet [1B], index [1B], počet * data [4B]	počet [1B], index [1B]	
0x21	33	GetPLCInfo	vrátí strukturu PLC_Info	-	PLC_Info[31]	
0x22	34	PrgCluster	provede zatažení bloku dat s uživatelským programem do stanice	stránka [2B], data [256B]	stránka [1B]	
0x23	35	DnlFinish	provede aktivaci nově zataženého uživatelského programu ve stanici	DnlDesc [28B]	-	
0x24	36	PLCRun	Spustí uživatelský program v stanici	-	-	
0x25	37	PLCStop	zastaví uživatelský program v stanici	-	-	
0x26	38	PLCReboot	provede reboot stanice	-	-	
0x27	39	PLCSetTime	nastaví čas stanice	PLC_Time [3B]	-	
0x28	40	PLCSetDate	nastaví datum stanice	PLC_Date [4B]	-	
0x29	41	PLCRebootFreeze	uvede automat do stavu zamrznutí	-		
0x2A	42	ReadRAMBit	přečte bit z paměti automatu	adresa[2B],maska[1]	adresa[2B],maska[1],data[1]	

ID hex	ID dec	Název požadavku	Popis	Vstup	Výstup	A
0x2B	43	WriteRAMBit	změní hodnotu bitu v paměti automatu podle předaných dat	adresa[2B],maska[1],data[1]	adresa[2B],maska[1]	
0x40	64	ReadRAMByte	přečte bajt z RAM PLC	adresa [2]	adresa [2] , data [1]	
0x41	65	WriteRAMByte	zapiše bajt do RAM PLC	adresa [2], data [1]	adresa [2]	
0x42	66	ReadRAMByteBlock	přečte blok bajtů z RAM PLC	počet [1] ,adresa [2]	počet [1],adresa [2], počet * data [1]	
0x43	67	WriteRAMByteBlock	zapiše blok bajtů do RAM PLC	počet [1] , adresa [2], počet * data [1]	počet [1] , adresa [2]	
0x44	68	ReadRAMWord	přečte word z RAM PLC	adresa [2]	adresa [2] , data [2]	
0x45	69	WriteRAMWord	zapiše word do RAM PLC	adresa [2], data [2]	adresa [2]	
0x46	70	ReadRAMWordBlock	Přečte blok wordů z RAM PLC	počet [1] ,adresa [2]	počet [1],adresa [2], počet * data [2]	
0x47	71	WriteRAMWordBlock	zapiše blok wordů do RAM PLC	počet [1] , adresa [2], počet * data [2]	počet [1] , adresa [2]	
0x48	72	ReadRAMLong	Přečte long z RAM PLC	adresa [2]	adresa [2] , data [4]	
0x49	73	WriteRAMLong	zapiše long do RAM PLC	adresa [2], data [4]	adresa [2]	
0x4A	74	ReadRAMLongBlock	Přečte blok longů z RAM PLC	počet [1] ,adresa [2]	počet [1],adresa [2], počet * data [4]	
0x4B	75	WriteRAMLongBlock	zapiše blok longů do RAM PLC	počet [1] , adresa [2], počet * data [4]	počet [1] , adresa [2]	
0x2C		ReadRAM64B	Přečte až 64 bytů pomocí dlouhé adresy	počet [1] , adresa [4]	počet [1] , adresa [4], počet * data [1]	
0x2D		WtiteRAM64B	Zapiše až 64 bytů pomocí dlouhé adresy	počet [1] , adresa [4]; počet * data [1]	počet [1] , adresa [4]	
0x2E		ReadRAM64W	Přečte až 64 wordů pomocí dlouhé adresy	počet [1] , adresa [4]	počet [1] , adresa [4], počet * data [2]	
0x2F		WtiteRAM64W	Zapiše až 64 wordů pomocí dlouhé adresy	počet [1] , adresa [4]; počet * data [2]	počet [1] , adresa [4]	

0x30		ReadRAM64L	Přečte až 64 longwordů pomocí dlouhé adresy	počet [1] , adresa [4]	počet [1] , adresa [4], počet * data [4]	
0x31		WriteRAM64L	Zapiše až 64 longwordů pomocí dlouhé adresy	počet [1] , adresa [4]; počet * data [4]	počet [1] , adresa [4]	
0x4C	76	ReadEEByte	Přečte bajt z EEPROM PLC	adresa [2]	adresa [2], data [1]	
0x4D	77	WriteEEByte	zapiše bajt do EEPROM PLC	adresa [2], data [1]	adresa [2]	
0x4E	78	ReadSTPWord	Přečte systémový word z PLC	adresa [1]	adresa [1], data [2]	
0x4F	79	WriteSTPWord	Zapiše systémový word z PLC	adresa [1], data [2]	adresa [1]	
0x50	80	ReadSTPBit	Přečte systémový bit z PLC	adresa [2]	adresa [2], data [1]	
0x51	81	WriteSTPBit	Zapiše systémový bit z PLC	adresa[2],data[1]	adresa [2]	
ID hex	ID dec	Název požadavku	Popis	Vstup	Výstup	A
0x52	82	SetRAMBitMask	nastaví bity podle masky v rámci bajtu v RAM automatu	Adresa [2], maska [1]	adresa [2]	
0x53	83	ClrRAMBitMask	vynuluje bity podle masky v rámci bajtu v RAM automatu	adresa [2], maska [1]	adresa [2]	
0x6E	110	ServerBusy	Komunikátor asynchronně zasilá, pokud je zaneprázdněn zpracováváním požadavku	není	Data[2]	x

5 Příloha B – seznam chybových hlášení

Většina příkazů protokolu EPNP vrací při chybě jednobajtový kód. Jeho význam je následující.

č. HEX	č. DEC	význam	důvod
0x01	1	chyba při vysílání do sítě PLC	
0x02	2	chyba při zastavování sítě PLC	komunikátor není schopen zastavit komunikaci v síti PLC
0x03	3	chyba při příjmu ze sítě PLC	chyba PLC nebo vedení
0x04	4	timeout při příjmu ze sítě PLC	PLC v požadovaném čase neodpovídá
0x05	5	nesprávná délka přijatého rámce z PLC	chyba PLC nebo vedení

0x06	6	nesprávná suma přijatého rámce z PLC	chyba PLC nebo vedení
0x07	7	PLC neodpovídá	PLC není na síti, nebo není schopný komunikace
0x10	16	neznámé číslo firmware PLC	chyba PLC, nebo neznámý typ
0x11	17	neznámý typ CPU stanice	chyba PLC, nebo neznámý typ
0x13	19	adresa nebo index leží mimo rozsah	
0x15	21	PLC je v provozu	při zatahování uživatelského programu musí být PLC zastaveno
0x16	22	nesprávný součet K1	kontrolní součet zataženého programu nesouhlasí s aktivačním součtem
0x17	23	chyba při zatahování uživatelského programu do PLC	chyba PLC
0x19	25	neplatná konfigurace	v komunikátoru je nahraná neplatná konfigurace, je nutné ji přehrát
0x1A	26	nelze smazat stránku konfigurace	závažná chyba v komunikátoru, kontaktujte výrobce
0x1B	27	nelze nahrát stránku konfigurace	stránka nebyla smazána
0x27	39	nelze zapsat data	
0x28	40	autorizace neproběhla	uživatelské jméno nebylo nalezeno
0x29	41	neautorizovaná operace	operace vyžaduje autorizaci
0x30	48	nejsou připojeny žádné automaty nebo nebyla navázána komunikace	

6 Příloha C – předávané struktury

Některé příkazy EPNP používají složitější datovou strukturu.

Jméno struktury	Délka	Použití v příkazu
PLC_Info	76	GetPLCInfo, op 33

Položka (zápis v jazyce C)	Délka v bytech	Pozice ve struktuře	význam
unsigned int FWVers	2	0	Verze FW automatu
unsigned char Status	1	2	Stavový bajt automatu

char PesType[7]	7	3	Typ automatu (K1, 302...)
unsigned int SerialNo	2	10	Sériové číslo automatu
unsigned int ProgSize	2	12	Velikost uloženého programu
unsigned char ProgType	1	14	Typ uloženého programu
char PesName[8]	8	15	Jméno automatu
unsigned char MemCfg	1	23	Konfigurace paměti
unsigned char SysDsc[5]	5	24	SYSDYS – systémové informace
unsigned char DnlDesc[32]	32	29	Descriptor DNL
unsigned char DnlTable[14]	14	61	Tabulka DNL