

RS485/MODBUS-RTU ver. 3.0

Komunikace s převodníkem probíhá na principu MASTER - SLAVE.

Protokol MODBUS má tuto strukturu:

<t_{off}> <slave adresa> <funkce> <data> <CRC> <t_{off}>

Význam jednotlivých částí protokolu

část příkazu	význam	počet bitů
<t _{off} >	časová prodleva delší než 3,5 znaku	
<slave adresa>	adresa z rozsahu <1 ... 247>	8 bitů
<funkce>	číselné označení funkce	8 bitů
<data>	význam je dán popisem jednotlivých funkcí N	* 8 bitů
<CRC>	kontrolní součet	16 bitů

Příkaz je představován N-ticí osmibitových dat. Pokud dojde během vysílání dat k časové prodlevě větší než představuje čas vyslání čtyř osmibitových dat při dané Bd rychlosti, je přijímání přerušeno a příkaz se dekóduje. První prováděnou kontrolou je výpočet a kontrola CRC (kontrolního součtu). Pokud převodník dostane nesrozumitelná data, neodpovídá. Pokud dostane data se svou adresou, při dekódování však přístroj narazí na syntaktickou chybu, odpovídá chybovým hlášením ([viz tabulka](#)).

Pokud přístroj řádně přijal příkaz, odpovídá po době delší než t_{off} stejnou strukturou jako byl dotaz.

Obsah

RS485/MODBUS-RTU ver. 3.0.....	1
Obsah.....	1
Popis funkcí.....	1
Funkce 03 H a 04 H.....	2
Funkce 06H.....	3
Funkce 08H.....	4
Funkce 11H.....	5
Tabulka chybových hlášení.....	6
Obsazení paměti EEPROM.....	7
Tabulka 1 - Význam bitů v konfiguračním slově.....	8
Tabulka 2 - Význam bitů v nastavení komunikace.....	9

Popis funkcí

číslo funkce	význam	data příkaz	data odpověď
03 _H	čtení 16-bitových dat	16b - adresa čteného registru	8b - počet bytů v odpovědi
04 _H		16b - počet N čtených registrů	2*N bytů - data z registrů
06 _H	zápis 16b registru do paměti převodníku	16b - adresa zápisu 16b - hodnota zapisovaných dat	16b - adresa zápisu 16b - hodnota zapsaných dat
08 _H	reset převodníku	0001 _H FF00 _H	0001 _H FF00 _H
11 _H	Report Slave ID	nejsou	8b - počet bytů v odpovědi (počet je vždy 02 _H) MSB - typ převodníku LSB - verze SW

RS485/MODBUS-RTU ver. 3.0

Funkce 03_H a 04_H

Funkce 03_H a 04_H jsou totožné a slouží pro čtení hodnoty z určeného 16-ti bitového registru převodníku.

Struktura příkazu je následující:

<adresa převodníku> <03 nebo 04> <adresa registru> <počet čtených registrů> <CRC>

Význam parametrů příkazu	
adresa převodníku	8b hodnota adresy z rozsahu <1 .. 247>
adresa registru	16b adresa prvního čteného paměťového místa
počet čtených registrů	16b hodnota počtu po sobě čtených registrů přípustné jsou jen hodnoty 0001 _H , 0002 _H a 0004 _H
CRC	kontrolní součet

Obsazení paměti převodníku a význam jednotlivých registrů je dáno [tabulkou](#).

Odpověď po řádně zadaném příkazu je:

<adresa převodníku> <03 nebo 04> <2*N> <N*16b hodnot po sobě jdoucích registrů> <CRC>

V případě chyby v příkazu přichází chybové hlášení ([viz. tabulka](#)) nebo převodník neodpoví vůbec.

Příklady příkazů		
popis	příklad příkazu	příklad odpovědi
čtení vstupní hodnoty 32b	01 04 0003 0002 81CB _H	01 04 04 FFFF FFCD 7BC5 _H (-0,51 °C)
čtení 2*vstupní hodnoty 2*32b	01 04 0001 0004 A009 _H	01 04 08 0000 0280 FFFF FFCD A470 _H (+6,40 a -0,51 °C)

[zpět](#)

RS485/MODBUS-RTU ver. 3.0

Funkce 06_H

Funkce 06_H slouží pro zápis hodnoty do určeného 16-ti bitového registru převodníku.

Struktura příkazu je následující:

<adresa převodníku> <06> <adresa registru> <16-ti bitová hodnota> <CRC>

Význam parametrů příkazu	
adresa převodníku	8b hodnota adresy z rozsahu <1 .. 247>
adresa registru	16b adresa paměťového místa, na které se provádí zápis
zapisovaná data	16b hodnota zapisovaná na danou adresu
CRC	kontrolní součet

Obsazení paměti převodníku a význam jednotlivých registrů je dáno [tabulkou](#).

Odpověď po řádně zadaném příkazu je"

<adresa převodníku> <06> <adresa registru> <16-ti bitová hodnota> <CRC>

V případě chyby v příkazu přichází chybové hlášení ([viz. tabulka](#)) nebo převodník neodpoví vůbec.

Příklady příkazů		
popis	příklad příkazu	příklad odpovědi
zápis 16b hodnoty	01 06 1032 0C02 A804 _H	01 06 1032 0C02 A804 _H

Poznámka: Hodnoty zapsané do paměťových míst určující konfiguraci přístroje se projeví až po resetu ([funkce 08h](#)).

[zpět](#)

RS485/MODBUS-RTU ver. 3.0

Funkce 08_H

Funkce 08_H slouží pro provedení SW resetu převodníku.

Struktura příkazu je následující:

<adresa převodníku> <08> <0001FF00_H> <CRC>

Význam parametrů příkazu	
adresa převodníku	8b hodnota adresy z rozsahu <1 .. 247>
0001FF00 _H	pevně daná konstanta
CRC	kontrolní součet

Odpověď po řádně zadaném příkazu je shodná se zadaným příkazem.

V případě chyby v příkazu přichází chybové hlášení ([viz. tabulka](#)) nebo převodník neodpoví vůbec.

Poznámka: Příkaz Reset je nutné provést vždy po změně konfigurace převodníku i po změně v konfiguraci komunikace.

[zpět](#)

RS485/MODBUS-RTU ver. 3.0

Funkce 11_H

Funkce 11_H slouží pro identifikaci převodníku a obsaženého SW.

Struktura příkazu je následující:

<adresa převodníku> <11> <CRC>

Význam parametrů příkazu	
adresa převodníku	8b hodnota adresy z rozsahu <1 .. 247>
CRC	kontrolní součet

Odpověď po řádně zadaném příkazu je:

<adresa převodníku> <11> <počet> <typ> <SW> <CRC>

Význam parametrů v odpovědi	
parametr	význam
adresa převodníku	hodnota adresy uvedená v příkazu
počet	počet bytů v odpovědi (zde vždy 02 _H)
typ	70 _H - PPL112 6E _H - PPL110 64 _H - PPL100 D2 _H - PXL210 D4 _H - PXL212
SW	číslo SW obsaženého v převodníku

V případě chyby v příkazu přichází chybové hlášení ([viz. tabulka](#)) nebo převodník neodpoví vůbec.

Příklad příkazu		
popis	příklad příkazu	příklad odpovědi
identifikace převodníku PXL212 verze SW 3	01 11 C0 2C _H	01 11 02 D403 A23D _H

[zpět](#)

RS485/MODBUS-RTU ver. 3.0

Tabulka chybových hlášení

Pokud dojde k chybě po volání funkce, je v odpovědi číslo funkce zvýšeno o 80_H. V datech pak následuje číslo chyby.

Chybové hlášení má tento tvar:

<adr> <funkce +80_H> <číslo chyby> <CRC>

Význam chybových čísel	
01 _H	neznámá funkce
02 _H	chybný počet registrů
03 _H	chyba v příkazu (chybná data)
04 _H	vstup mimo rozsah (rozpojen, zkratován)

Příklad chybového hlášení převodníku s adresou 02_H se vstupem mimo rozsah při volání funkce 03_H bude vypadat takto:

(02 83 04 B0 F3)_H

[zpět](#)

RS485/MODBUS-RTU ver. 3.0

Obsazení paměti EEPROM

Pro zápis hodnoty i adresy příslušné paměti je použita 16 bitová hodnota zapsaná v hexadecimálním tvaru.

Popis obsazení paměti EEPROM převodníku		
Paměťové místo	Význam obsahu	Dostupnost
0001 _H a 0002 _H	32b vstup 1 (long integer)	jen čtení
0003 _H a 0004 _H	32b vstup 2 (long integer)	
0011 _H	16b vstup 1	
0012 _H	16b vstup 2	
1000 _H až 1029 _H	linearizační data	čtení i zápis
102A _H	konfigurační slovo (viz tab.1)	
102B _H	korekce vstupu 1 *)	
102C _H	korekce vstupu 2 *)	
102D _H	MSB číslo měsíce kalibrace LSB číslo roku kalibrace	
1032 _H	MSB nastavení komunikace (viz tab.2) LSB adresa převodníku (z výroby 01 _H)	
1034 _H a 1035 _H	32b výrobní číslo převodníku	jen čtení

*) Hodnota korekce vstupu představuje 16 bitové hexadecimální číslo v doplňkovém tvaru. Můžeme tak vyjádřit jak kladný tak i záporný posuv o daný počet digitů.

Posuv o +1 digit tak vyjádříme číslem 0x0001,

posuv o -1 digit pak číslem 0xFFFF

[zpět](#)

RS485/MODBUS-RTU ver. 3.0

Tabulka 1 - Význam bitů v konfiguračním slově

Bitu	Význam	Popis hodnot
16 (MSB)	nemá význam	0
15		
14		
13		
12		
11		
10		
9		
8		
7	reakce na přetečení rozsahu	0 - při přetečení chyba 1 - při přetečení asi 6% nad nebo pod rozsah
6	nemá význam	0
5	filtr	0 - filtr OFF 1 - filtr ON
4	nemá význam	0
3	prohození pořadí hodnot při 32b vstupu	0 - 16b MSB pak 16b LSB 1 - 16b LSB pak 16b MSB
2	kompence	0 - 3W nebo komp. stud. konce 1 - 2W nebo bez komp. stud. konce
1 (LSB)	rozlišení vstupního zesilovače (souvisí s rychlostí převodu)	0 - 15 bitů (pomalejší převod) 1 - 14 bitů (rychlejší převod)

[zpět](#)

RS485/MODBUS-RTU ver. 3.0

Tabulka 2 - Význam bitů v nastavení komunikace

Bitu	Význam	Popis hodnot
16 (MSB)	nemá význam	0
15		
14		
13	rychlost komunikace [Bd]	00 - 19200Bd
12		01 - 9600Bd 10 - 4800Bd 11 - 2400Bd
11	parametry komunikace (počet datových bitů parita počet stop bitů)	1xx - 8N1 **)
10		000 - 8E1
9		001 - 8O1 01x - 8N2
8	adresa přístroje (vyjádřena binárně)	čísla z rozsahu <1 .. 247>
7		
6		
5		
4		
3		
2		
1		

**) Pro x nezáleží na hodnotě bitu.

[zpět](#)