Obsah

Úvod 2
Obecný popis protokolu Modbus 4
Režimy komunikace 4
Datová jednotka protokolu Modbus (PDU) 4
Modbus kódy funkcí 5
Obecný adresní prostor protokolu Modbus 6
Modbus na sériové lince 6
Výchozí nastavení sériové linky 6
Struktura Modbus RTU rámce na sériové lince 6
Výpočet CRC 7
Seznam Modbus funkcí 7
Adresní prostor Modbus NLII čidel 8
Seznam cívek 8
Seznam vstupních registrů 8
Seznam uchovávacích registrů 9
Popis, syntaxe a příklad použití využívaných Modbus funkcí 11
(0x01) Funkce Čti cívky 11
(0x05) Funkce Zapiš jednu cívku 12
(0x0F) Funkce Zapiš více cívek 13
(0x04) Funkce Čti vstupní registry 14
(0x03) Funkce Čti uchovávací registry 15
(0x06) Funkce Zapiš jeden uchovávací registr 16
(0x10) Funkce Zapiš více uchovávacích registrů 17

Úvod

Tento dokument slouží k popisu Modbus protokolu použitého u čidla IL-RG-1. Verze tohoto návodu je určena **pro čidla s verzí Firmware 100 a výše**. Číslo verze FW je uvedeno na samolepícím štítku umístěném na plošném spoji uvnitř čidla

Na úvod uveďme několik užitečných informací, k řešení případných potíží:

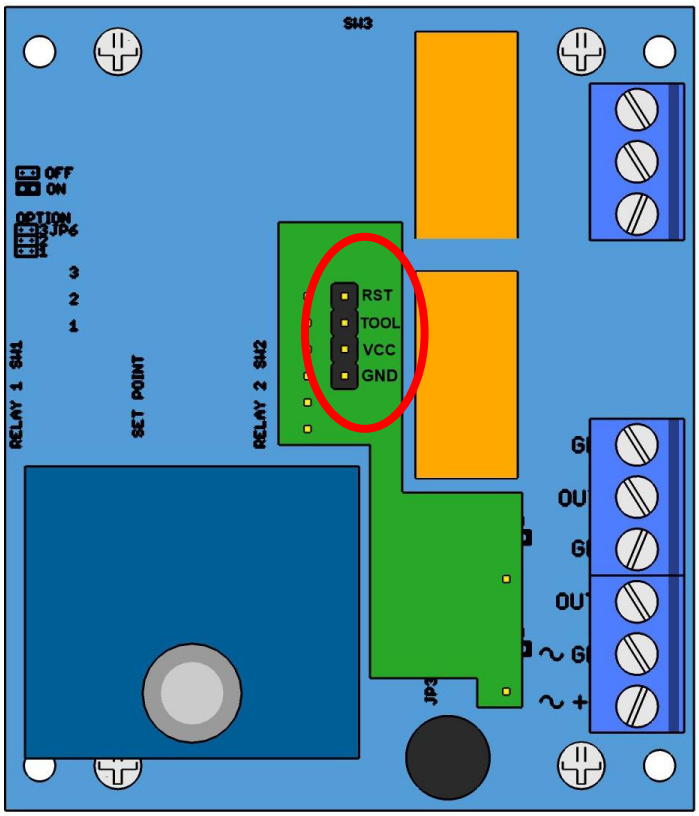
Z čidla je možné vyčítat pouze ty cívky/ registry, které jsou na čidlu dostupné. V opačném případě čidlo reaguje chybovou odpovědí s kódem chyby 0x02 – Ilegální adresa dat. Pro snadnou identifikaci čidla slouží vstupní registr **ID čidla** (viz Seznam vstupních registrů).

O tom jaké služby čidlo nabízí, pak informuje vstupní registr **Dostupné služby čidla** (viz Seznam vstupních registrů). Před jakýmkoli zápisem (cívky/ uchovávacího registru) se tedy nejdříve ujistěte, zda je tato služba čidlem podporována.

V případě potřeby trvalého uložení provedených změn v nastavení čidla je nutné využít cívky **Save configuration** (viz Seznam cívek)

Pokud je naopak nutné obnovit tovární nastavení čidla, lze využít cívky **Factory configuration** (viz Seznam cívek). **Obnovit tovární nastavení** lze v případě ztráty spojení s čidlem i **hardwarovou cestou** a to následovně:

1) Odpojíme čidlo od napájení
2) Na desce plošných spojů čidla vyhledáme konektor označený JP5 (viz obr)
3) Jeho piny TOOL a GND propojíme a připojíme napájení
4) Následně již po několika sekundách můžeme propoj odstranit, neboť je obnova dokončena



Čidlo vyžaduje ke zpracování požadavku jistý čas, proto je nutné počítat s dostatečným časem pro odpověď našeho čidla. Doba než čidlo odpoví, je různá, dle zvolené modbus funkce a počtu vyčítaných/ zapisovaných prvků. Běžně by měla být doba odezvy zanedbatelná. Nicméně v případě požadavku o uložení/obnovu nastavení čidla, zde je odezva delší (max 100ms)a je nutné s ní počítat.

V případě, že čidlo nekomunikuje, ujistěte se, že Vámi odesílané rámce jsou správné a zkontrolujte, zda na komunikační sběrnici dodržujete pauzy minimálně 4ms, pro správnou detekci konců rámce.

Čidlo pracuje v režimu tzv. **Half-duplex**. To znamená, že není schopno přijímat další požadavky do doby, než odpoví na předchozí modbus rámec.

Pro případnou **kontrolu** či ověření správnosti **výpočtů modbus crc** je možné využít on-line kalkulátoru:
https://www.lammertbies.nl/comm/info/crc-calculation.html
Kalkulátor je nutné přepnout na zadávání HEX znaků a následná výsledek CRC-16 (Modbus) má však v modbus rámci zaměněný horní a dolní byte.

Obecný popis protokolu Modbus

Modbus protokol je Master-Slave protokol. Na sběrnici je přítomen pouze 1 master a až 247 slave zařízení (v našem případě čidel). Komunikaci vždy iniciuje master zařízení. Slave pouze odpovídá na požadavky master zařízení. Modbus využívá Big-endian reprezentaci dat. To znamená že u položek větších jak 1 B je nejdříve je odeslán nejvyšší byte a nejnižší byte až na konec.

Režimy komunikace

Unicast režim:

Master osloví **jedno konkrétní slave zařízení** pomocí jeho Modbus adresy. Slave zprávu zpracuje a odpoví.

Broadcast režim:

Master osloví **všechna slave zařízení** pomocí **broadcast adresy 0**. Všechna slave zařízení zprávu zpracují, ale žádné na ni neodpoví. Broadcast požadavky master zařízení musejí být nutně **příkazy k zápisu**.

Datová jednotka protokolu Modbus (PDU)

Modbus funkce	Data
1 B	N* 1 B

Protokol Modbus definuje tři základní typy PDU:

1) **Request PDU -** Slouží k oslovení jednoho či více slave zařízení masterem.

   Pole Modbus funkce obsahuje kód dané Modbus funkce. Pole data pak dle dané Modbus funkce adresy, počet proměnných, hodnoty proměnných a jiné

2) **Response PDU -** Slouží k odeslání **kladné odpovědi** slave zařízením na přijatou Request PDU.

   Pole **Modbus funkce** obsahuje **stejnou hodnotu** jako v přijatém Request PDU. Datová část pak dle dané Modbus funkce provozní hodnoty, přečtené vstupy, cívky …

3) **Exception Response PDU -** Slouží k odeslání **záporné odpovědi** slave zařízením na přijatou Request PDU.

   Pole **Modbus funkce** obsahuje **hodnotu Modbus funkce z Request PDU + 0x80** jako signalizace neúspěchu. Datová část pak **identifikuje chybu**.

Chybové kódy v Exception Response PDU

Kód	*Druh kódu funkce*	*Význam*
0x01	Ilegální Modbus funkce	Požadovaná Modbus funkce není serverem (čidlem) podporována
0x02	Ilegální adresa dat	Zadaná adresa (cívky, registru …) je mimo serverem podporovaný rozsah
0x03	Ilegální hodnota dat	Předávaná data jsou neplatná
0x04	Selhání zařízení	Při provádění požadavku došlo k neodstranitelné chybě
0x05	Potvrzení	Kód určený k použití při programování. Server hlásí přijetí platného požadavku, ale jeho vykonání bude trvat delší dobu
0x06	Zařízení je zaneprázdněné	Kód určený k použití při programování. Server je zaneprázdněn vykonáváním dlouho trvajícího příkazu.
0x08	Chyba parity paměti	Kód určený k použití při práci se soubory. Server při pokusu přečíst soubor zjistil chybu parity
0x0A	Brána – přenosová cesta nedostupná	Kód určený k práci s bránou (gateway). Brána není schopná vyhradit interní přenosovou cestu od vstupního portu k výstupnímu. Pravděpodobně je přetížená nebo nesprávně nastavená.
0x0B	Brána – cílové zařízení neodpovídá	Kód určený k práci s bránou (gateway). Cílové zařízení neodpovídá, pravděpodobně není přítomno.

Modbus kódy funkcí

1) **Veřejné kódy funkcí** - Jsou jasně definovány a veřejně zdokumentovány. Je zaručena jejich unikátnost. Obsahují i některé nevyužité kódy pro budoucí využití.
2) **Uživatelsky definované kódy funkcí** - Umožňují uživateli implementovat funkci, která není protokolem definována. Není garantována unikátnost kódu.

Rozsahy kódů Modbus funkcí

Kód funkce	*Druh kódu funkce*
1 ... 64	Veřejné kódy funkcí
65 ... 72	Uživatelsky definované kódy funkcí
73 ... 100	Veřejné kódy funkcí
101 ... 110	Uživatelsky definované kódy funkcí
111 ... 127	Veřejné kódy funkcí

Obecný adresní prostor protokolu Modbus

Adresní prostor protokolu Modbus je založen na sadě tabulek s charakteristickými významy. Definovány jsou tyto čtyři základní tabulky:

Tabulka	*Popis*	*Přístup*	*Adresní prostor (není podmínkou)*
Diskrétní vstupy	1-bit	Pouze čtení	0x2710 až 0x4E1F
Cívky	1-bit	Čtení i zápis	0x0000 až 0x270F
Vstupní registry	16-bitů	Pouze čtení	0x7530 až 0x9C3F
Uchovávací registry	16-bitů	Čtení i zápis	0x9C40 až 0xC34F

Modbus na sériové lince

Výchozí nastavení sériové linky

Protokol Modbus RTU definuje výchozí nastavení sériové linky následovně:

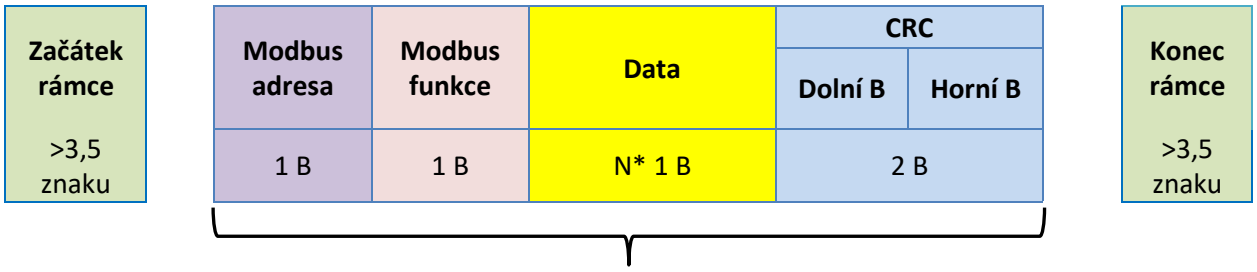
Baud rate 19200
1 start bit
8 datových bitů
1 sudý paritní bit
1 stop bit

Tomuto nastavení odpovídají výchozí hodnoty cívek "Modbus parita" a "Modbus stop bit" a uchovávacího registru "Modbus baud rate".

Dále je doporučeno implementovat Baud rate 9600 a možnost žádné parity, přičemž je tento paritní bit nahrazen druhým stop bitem. Možnost dvou stop bitů není podporována.

V režimu Modbus RTU je 1 B složen ze dvou čtyřbitových hexa znaků. Vysílání Modbus rámce je započato a ukončeno pomlkou na sběrnici delší jak 3,5 znaku. Během vysílání rámce nesmí být mezery mezi jednotlivými znaky větší jak 1,5 znaku.

Struktura Modbus RTU rámce na sériové lince



Modbus rámec na sériové lince
Maximální délka Modbus RTU rámce omezena na 32 B.

Adresování slave zařízení:

adresa	*význam*
0	Broadcast adresa
1 až 247	Individuální slave adresy
248 až 255	Rezervováno

V Modbus RTU rámcích nesoucích odpovědi určené pro master zařízení je ponechána Modbus adresa odpovídajícího slave zařízení.

Výpočet CRC

Výpočet CRC je prováděn z celého rámce včetně Modbus slave adresy, modbus funkce a datové části rámce

1. Inicializace 16-bit CRC registru na 0xFFFF.
2. Provedeme XOR prvních 8 bitů rámce se spodním bytem CRC registru a výsledek uložíme do CRC registru.
3. Provedeme posun CRC registru o 1 b doprava (směrem k LSB), MSB CRC registru vyplníme 0. Zachytíme a vyhodnotíme nejspodnější bit, který posunem vypadl.
4. Pokud byl tento bit roven 1, provedeme XOR mezi CRC registrem a hodnotou 0xA001 (generující polynom= 1+x2+x15+x16). Výsledek uložíme opět do CRC registru.
5. Opakujeme kroky 3 a 4 dokud nebude provedeno osm posunů CRC registru.
6. Provedeme XOR dalších 8 bitů rámce se spodním bytem CRC registru a opakujeme kroky 3 až 5.
7. Takto pokračujeme až do posledního byte rámce.
8. Výsledná hodnota CRC výpočtu je uložena v CRC registru.
9. Při umístění CRC hodnoty do Modbus rámce je nutné zaměnit horní a spodní byte CRC registru (viz struktura Modbus RTU rámce na sériové lince).

Seznam Modbus funkcí			
Určení	***Kód funkce***	***Příkaz***	***Veř./Uživ.***
Cívky	0x01	Čti cívky	Veřejná
	0x05	Zapiš jednu cívku	Veřejná
	0x0F	Zapiš více cívek	Veřejná
Vst. registry	0x04	Čti vstupní registry	Veřejná
Uch. registry	0x03	Čti uchovávací registry	Veřejná
	0x06	Zapiš jeden uchovávací registr	Veřejná
	0x10	Zapiš více uchovávacích registrů	Veřejná
	⋮	⋮	⋮

* Jiné Modbus funkce nejsou podporovány

Adresní prostor Modbus NLII čidel

Seznam cívek				
cívka	***Adresa cívky***	***Popis***	***Výchozí hodnota***	***R/W***
Modbus parita	0x0000	Parita 0 = sudá, 1 = žádná	0	R/W
Modbus stop bit	0x0001	Stop bit 0 = 1 stop bit, 1= žádný stop bit	0	R/W
Ulož konfiguraci	0x0002	Zápis 1 uloží aktuální konfiguraci jako výchozí První vyčtení nuluje příznak uložení konfigurace	0	R/W
Tovární konfigurace	0x0003	Zápis 1 obnoví tovární konfiguraci jako výchozí První vyčtení nuluje příznak obnovy konfigurace	0	R/W
⋮	⋮	⋮	⋮	⋮

Seznam vstupních registrů					
Vst. registr	***Adresa registru***	***Popis***	***Jednotky***	***Formát***	***R/W***
Stav senzoru	0x7530	Stav senzoru:		uint16	R
		bit 0 - signalizace bezchybného provozu senzoru	0-chyba / 1-ok		
		bit 1 - signalizace sepnutého Relé 1	0-off / 1-on		
		bit 2 - signalizace sepnutého Relé 2	0-off / 1-on		
		bit 3 - signalizace aktivního alarmu 1	0-off / 1-on		
		bit 4 - signalizace aktivního alarmu 2	0-off / 1-on		
		bit 5-15 vyhrazeno pro budoucí účely			
REFRIGERANT PPM	0x7531	Koncentrace chladiva	ppm	uint16	R
REFRIGERANT ADC	0x7532	Koncentrace chladiva	-	uint16	R
Osvětlení	0x7533	Intenzita okolního osvětlení	-	uint16	R
⋮	⋮	⋮	⋮	⋮	⋮
ID čidla	0x7538	Identifikační označení čidla: 0x0016 – IL-RG-1	-	uint16	R
⋮	⋮	⋮	⋮	⋮	⋮

Seznam uchovávacích registrů					
Uch. Registr	***Adresa registru***	***Popis***	***rozsah***	***Výchozí hodnota***	***R/W***
Modbus adresa	0x9C40	Modbus slave adresa čidla	1 až 247	1	R/W
Modbus baud rate	0x9C41	komunikační tok sběrnice	[1]	19200	R/W
Relé 1 - MK	0x9C42	Relé 1 Mód komparace 0x0000= relé vždy rozepnuto 0x0001= relé vždy sepnuto ostatní= automatický režim	0 až 0xFFFF	2	R/W
Relé 1 - KS	0x9C43	Relé 1 klidový stav (stav při neaktivním alarmu) 0x0000= v klidu rozepnuto (NO) ostatní= v klidu sepnuto (NC)	0 až 0xFFFF	0	R/W
Relé 1 - APPM	0x9C44	Alarmová hodnota 1 - PPM	0 až 1000	300	R/W
Relé 1 - AADC	0x9C45	Alarmová hodnota 1 – ADC	0 až 1000	50	R/W
Relé 1 - APER	0x9C46	Perioda detekce ADC alarmu 1	0 až 14400 min	5	R/W
Relé 1 - ATIME	0x9C47	Doba, po kterou je alarm aktivní (0- trvale aktivní, shození povelem ARST)	0 až 14400 min	0	R/W
Relé 1 - ARST	0x9C48	Reset alarmu v automatickém režimu (Zápisem 1 dojde k resetu alarmu), po prvním vyčtení se hodnota automaticky nuluje	1/0	0	R/W
Relé 2 - MK	0x9C49	Relé 2 Mód komparace	0 až 0xFFFF	2	R/W
Relé 2 - KS	0x9C4A	Relé 2 klidový stav (stav při neaktivním alarmu) 0x0000= v klidu rozepnuto (NO) ostatní= v klidu sepnuto (NC)	0 až 0xFFFF	0	R/W
Relé 2 - APPM	0x9C4B	Alarmová hodnota 2 - PPM	0 až 1000	300	R/W
Relé 2 - AADC	0x9C4C	Alarmová hodnota 2 – ADC	0 až 1000	50	R/W
Relé 2 - APER	0x9C4D	Perioda detekce ADC alarmu 2	0 až 14400 min	5	R/W
Relé 2 - ATIME	0x9C4E	Doba, po kterou je alarm aktivní (0- trvale aktivní, shození povelem ARST))	0 až 14400 min	0	R/W
Relé 2 - ARST	0x9C4F	Reset alarmu v automatickém režimu (Zápisem 1 dojde k resetu alarmu), po prvním vyčtení se hodnota automaticky nuluje	1/0	0	R/W
⋮	⋮	⋮	⋮	⋮	⋮

[1] Podporovány jsou pouze následující komunikační rychlosti 4800, 9600, 19200, 38400 a 56000

Seznam neveřejných uchovávacích registrů					
Uch. Registr	***Adresa registru***	***Popis***	***rozsah***	***Výchozí hodnota***	***R/W***
CAL1 – real ppm	0x9D00	Reálná hodnota ppm prvního kalibračního bodu	0-3000	300	R/W
CAL1 - ADC	0x9D01	Naměřená hodnota ADC v danném kalibračním bodě	0-1024	752	R/W
CAL2 – real ppm	0x9D02	Reálná hodnota ppm druhého kalibračního bodu	0-3000	3000	R/W
CAL2 - ADC	0x9D03	Naměřená hodnota ADC v danném kalibračním bodě	0-1024	470	R/W
⋮	⋮	⋮	⋮	⋮	⋮

Popis, syntaxe a příklad použití využívaných Modbus funkcí

(0x01) Funkce Čti cívky

Funkce slouží ke čtení stavu 1 až 2000 (v tomto případě omezeno na **184**) po sobě následujících cívek najednou. V požadavku je specifikována adresa první vyčítané cívky a počet cívek k vyčtení. V odpovědi je pak délka stavů cívek a stavy jednotlivých cívek přičemž jsou odesílány po 8 cívkách vždy od LSB k MSB. Je-li některý bajt neúplný, je doplněn nulami na pozicích od MSB směrem k LSB. **Log 1 = ON, Log 0 = OFF**

1) Request PDU

Modbus funkce **0x01** 1 B	**Adresa 1. cívky** **viz Seznam cívek** 2 B	**Počet cívek** **1 až max 184** 2 B

Příklad vyčtení cívek Modbus parita:

0x01	**Modbus parita** **0x00 0x00**	**Počet cívek =1** **0x00 0x01**

2) Response PDU

Modbus funkce **0x01** 1 B	**Počet bytů** **N** 1 B	**Stavy cívek** **N** B

N = Počet cívek/8, je-li zbytek po dělení nenulový pak N= (Počet cívek/8)+1

Příklad odpovědi pro parita žádná:

0x01	**Počet bytů** **0x01**	**Stavy cívek =1** **0x01**

3) Exception Response PDU

Modbus funkce |0x80 **0x81** 1 B	**Chybový kód** **1,2,3 nebo 4** 1 B

(0x05) Funkce Zapiš jednu cívku

Tato funkce slouží k nastavení jedné cívky do stavu ON nebo OFF. **0x0000 = OFF, 0xFF00 = ON**. Pozitivní odpověď je kopií požadavku.

1) Request PDU

Modbus funkce 0x05 1 B	Adresa cívky viz Seznam cívek 2 B	Stav cívky 0x0000 nebo 0xFF00 2 B

Příklad zapsání cívky Ulož konfiguraci na ON:

0x05	Ulož konfiguraci 0x00 0x02	Stav cívky = ON 0xFF 0x00

2) Response PDU

Modbus funkce 0x05 1 B	Adresa cívky viz Request PDU 2 B	Stav cívky viz Request PDU 2 B

Příklad odpovědi na zapsání cívky Ulož konfiguraci na ON:

0x05	Ulož konfiguraci 0x00 0x02	Stav cívky = ON 0xFF 0x00

3) Exception Response PDU

Modbus funkce |0x80 0x85 1 B	Chybový kód 1,2,3 nebo 4 1 B

(0x0F) Funkce Zapiš více cívek

Tato funkce louží k zapsání 1 až 1968 (v tomto případě omezeno na **184**) po sobě jdoucích cívek najednou. V požadavku je specifikována adresa první zapisované cívky, počet cívek, délka stavů cívek a stavy cívek přičemž jsou odesílány po 8 cívkách vždy od LSB k MSB. Je-li některý bajt neúplný, je dopněn nulami na pozicích od MSB směrem k LSB. **Log 1 = ON, Log 0 = OFF**. V odpovědi je pak adresa první cívky a počet zapsaných cívek.

1) Request PDU

Modbus funkce 0x0F 1 B	Adresa 1. cívky viz Seznam cívek 2 B	Počet cívek 1 až max 184 2 B	Počet bytů N 1 B	Stavy cívek N B

N = Počet cívek/8, je-li zbytek po dělení nenulový pak N= (Počet cívek/8)+1

Příklad zapsání cívek Modbus parita OFF, Modbus stop bit OFF, Ulož konfiguraci ON a Tovární konfigurace OFF:

0x0F	Modbus parita		Počet cívek = 4		Počet bytů = 1	Stavy cívek
	0x00	0x00	0x00	0x04	0x01	0x04

2) Response PDU

Modbus funkce 0x0F 1 B	Adresa 1. cívky viz Request PDU 2 B	Počet cívek viz Request PDU 2 B

Příklad odpovědi na zapsání cívek Modbus parita OFF, Modbus stop bit OFF, Ulož konfiguraci ON a Tovární konfigurace OFF:

0x0F	Modbus parita		Počet cívek = 4	
	0x00	0x00	0x00	0x04

3) Exception Response PDU

Modbus funkce |0x80 0x8F 1 B	Chybový kód 1,2,3 nebo 4 1 B

(0x04) Funkce Čti vstupní registry

Tato funkce slouží ke čtení obsahu souvislého bloku až 125 (v tomto případě omezeno na **13**) vstupních registrů. V požadavku je specifikována adresa prvního registru a počet registrů. V odpovědi odpovídá každému registru dvojice bytů.

1) Request PDU

Modbus funkce 0x04 1 B	Adresa 1. registru viz Seznam vst. reg. 2 B	Počet registrů 1 až max 13 2 B

Příklad vyčtení vstupních registrů REFRIGERANT PPM a REFRIGERANT ADC:

0x04	REFRIGERANT PPM 0x75 0x31	Počet registrů = 2 0x00 0x02

2) Response PDU

Modbus funkce 0x04 1 B	Počet bytů 2*N 1 B	Stavy registrů 2*N B

N = Počet registrů (viz Request PDU)

Příklad odpovědi na vyčtení vstupních registrů REFRIGERANT PPM a REFRIGERANT ADC:

0x04	Počet bytů 0x04	980 ppm 0x03 0xD4	335 ADC 0x01 0x4F

3) Exception Response PDU

Modbus funkce |0x80 0x84 1 B	Chybový kód 1,2,3 nebo 4 1 B

(0x03) Funkce Čti uchovávací registry

Tato funkce slouží ke čtení obsahu souvislého bloku až 125 (v tomto případě omezeno na **13**) uchovávacích registrů. V požadavku je specifikována adresa prvního registru a počet registrů. V odpovědi odpovídá každému registru dvojice bytů.

1) Request PDU

Modbus funkce **0x03** 1 B	Adresa 1. registru viz Seznam uch. reg. 2 B	Počet registrů **1 až max 13** 2 B

Příklad vyčtení uchovávacích registrů Relé 1 - MK, Relé 1 - KS

0x03	**Relé 1 - MK**		**Počet registrů = 2**	
	0x9C	**0x42**	**0x00**	**0x02**

2) Response PDU

Modbus funkce **0x03** 1 B	Počet bytů **2*N** 1 B	Stavy registrů **2*N** B

N = Počet registrů (viz Request PDU)

0x03	**Počet bytů**	**MK = AUTO**		**KS = 0**	
	0x04	**0x00**	**0x02**	**0x00**	**0x00**

3) Exception Response PDU

Modbus funkce |0x80 **0x83** 1 B	Chybový kód **1,2,3 nebo 4** 1 B

(0x06) Funkce Zapiš jeden uchovávací registr

Tato funkce slouží k zápisu jednoho uchovávacího registru. V požadavku je specifikována adresa registru, který se má zapsat a hodnota, která se má zapsat. Normální odpověď je kopií požadavku a je vrácena poté, co je registr zapsán.

1) Request PDU

Modbus funkce 0x06 1 B	Adresa registru viz Seznam uch. reg. 2 B	Stav registru 0x0000 až 0xFFFF 2 B

Příklad zapsání uchovávacího registru Modbus adresa na 10:

0x06	Modbus adresa 0x9C 0x40	Stav registru = 10 0x00 0x0A

2) Response PDU

Modbus funkce 0x06 1 B	Adresa registru viz Request PDU 2 B	Stav registru viz Request PDU 2 B

Příklad odpovědi na zapsání uchovávacího registru Modbus adresa na 10:

0x06	Modbus adresa 0x9C 0x40	Stav registru = 10 0x00 0x0A

3) Exception Response PDU

Modbus funkce |0x80 0x86 1 B	Chybový kód 1,2,3 nebo 4 1 B

(0x10) Funkce Zapiš více uchovávacích registrů

Tato funkce slouží k zápisu souvislého bloku až 120 (v tomto případě omezeno na **11**) registrů. V požadavku je specifikována adresa prvního registru, který se má zapsat, počet registrů a hodnoty, které se mají zapsat. Normální odpověď obsahuje počáteční adresu a počet zapsaných registrů.

1) Request PDU

Modbus funkce 0x10 1 B	Adresa 1. registru viz Seznam uch. reg. 2 B	Počet registrů 1 až max 11 2 B	Počet bytů 2*N 1 B	Stavy registrů 2*N B

N = Počet registrů

Příklad zapsání uchovávacích registrů Relé 1 - MK, Relé 1 - KS:

0x10	Relé 1 - MK 0x9C 0x42	Počet registrů = 2 0x00 0x02	Počet bytů = 4 0x04	MK = 1 0x00 0x01	KS = 1 0x00 0x01

2) Response PDU

Modbus funkce 0x10 1 B	Adresa 1. registru viz Request PDU 2 B	Počet registrů viz Request PDU 2 B

N = Počet registrů

Příklad odpovědi na zapsání uchovávacích registrů Relé 1 - MK, Relé 1 - KS:

0x10	Relé 1 - MK 0x9C 0x42	Počet registrů = 2 0x00 0x02

3) Exception Response PDU

Modbus funkce |0x80 0x90 1 B	Chybový kód 1,2,3 nebo 4 1 B