

## Obsah

Úvod.....	2
Obecný popis protokolu Modbus .....	3
Režimy komunikace .....	3
Datová jednotka protokolu Modbus (PDU) .....	3
Modbus kódy funkcí.....	4
Obecný adresní prostor protokolu Modbus .....	5
Modbus na sériové lince .....	5
Výchozí nastavení sériové linky .....	5
Struktura Modbus RTU rámce na sériové lince .....	5
Výpočet CRC.....	6
Seznam Modbus funkcí.....	6
Adresní prostor Modbus NLB čidel .....	7
Seznam cívek.....	7
Seznam vstupních registrů.....	7
Seznam uchovávacích registrů.....	8
Popis, syntaxe a příklad použití využívaných Modbus funkcí .....	8
(0x01) Funkce Čti cívky.....	8
(0x05) Funkce Zapiš jednu cívku .....	9
(0x0F) Funkce Zapiš více cívek .....	10
(0x04) Funkce Čti vstupní registry.....	11
(0x03) Funkce Čti uchovávací registry .....	12
(0x06) Funkce Zapiš jeden uchovávací registr .....	13
(0x10) Funkce Zapiš více uchovávacích registrů .....	14
(0x42) Funkce Čti vstupní registry v náhodném pořadí .....	15
(0x43) Funkce Čti vstupní registry v náhodném pořadí 2 .....	16



## Úvod

Tento dokument slouží k popisu Modbus protokolu použitého u čidel NLB s komunikačními rozhraními IQRF. Verze tohoto návodu je určena **pro čidla s verzí Firmware 406 a vyšší**. Číslo verze FW je uvedeno na samolepícím štítku umístěném na plošném spoji uvnitř čidla

**Na úvod uvedme 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](#)).

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](#)).

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.

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

Pro případnou **kontrolu** či ověření správnosti **výpočtů modbus crc** je možné využít on-line kalkulatoru:

<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 trvajících příkazů.
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

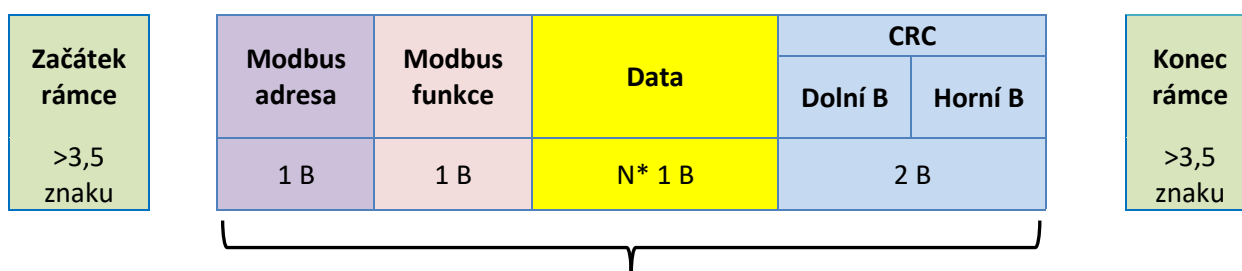
## Modbus na sériové lince

### Výchozí nastavení sériové linky

**Baud rate 19200**  
**1 start bit**  
**8 datových bitů**  
**žádný paritní bit**  
**1 stop bit**

Tomuto nastavení odpovídají komunikační parametry nastavené v čidle.

### 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= $1+x^2+x^{15}+x^{16}$ ). 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
	0x42	Čti vstupní registry v náhodném pořadí 1 *	<b>Uživatelská</b>
	0x43	Čti vstupní registry v náhodném pořadí 2	<b>Uživatelská</b>
Uch. registry	0x03	Čti uchovávací registry	Veřejná
	0x06	Zapiš jeden uchovávací registr	Veřejná
	0x10	Zapiš více uchovávacích registrů	Veřejná
	⋮	⋮	⋮

\* Pro nové implementace doporučujeme nepoužívat.

Jiné Modbus funkce nejsou podporovány.



## Adresní prostor Modbus NLB čidel

### Seznam cívek

<i>cívka</i>	<i>Adresa cívky</i>	<i>Popis</i>	<i>Výchozí hodnota</i>	<i>R/W</i>
Ulož konfiguraci	0x0000	Zápis 1 uloží aktuální konfiguraci jako výchozí	0	R/W
		První vyčtení nuluje příznak uložení konfigurace		
Tovární konfigurace	0x0001	Zápis 1 obnoví tovární konfiguraci jako výchozí	0	R/W
		První vyčtení nuluje příznak obnovy konfigurace		
⋮	⋮	⋮	⋮	⋮

### Seznam vstupních registrů

<i>Vst. registr</i>	<i>Adresa registru</i>	<i>Popis</i>	<i>Jednotky</i>	<i>format</i>	<i>R/W</i>
Stav senzoru	0x7530	Status of the sensor	-	uint16	R
		bit 0 – platná data CO2	0-ne / 1-ano		
		bit 1 – platná data RH	0-ne / 1-ano		
		bit 2 – Platná data T	0-ne / 1-ano		
		bit 3-15 vyhrazeno pro budoucí účely			
CO2	0x7531	Koncentrace oxidu uhličitého	ppm	uint16	R
RH	0x7532	Relativní vlhkost vzduchu	%	uint16	R
T	0x7533	Teplota okolního vzduchu	0,1 °C	uint16	R
IVOC	0x7534	Koncentrace VOC	ppm	uint16	R
Baterie	0x7535	Voltáž baterie senzoru	mV	uint16	R
ID čidla	0x7536	0x0081 – NLB-CO2 0x0082 – NLB-CO2-5 (CO2 do 5000 ppm) 0x0083 – NLB-CO2+RH 0x0084 – NLB-CO2+RH-5 (CO2 do 5000 ppm) 0x0085 – NLB-CO2+RH+T 0x0086 – NLB-CO2+RH+T-5 (CO2 do 5000 ppm) 0x0087 – NLB-IVOC 0x0088 – NLB-IVOC+RH 0x0089 – NLB-IVOC+RH+T 0x008A – NLB-RH 0x008B – NLB-RH+T ... 0x0098 – NLB-TVOC 0x0099 – NLB-TVOC+RH 0x009A – NLB-TVOC+RH+T		uint16	R
⋮	⋮	⋮	⋮	⋮	⋮
TVOC	0x7543	Koncentrace TVOC	µg/m <sup>3</sup>	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 <sup>1)</sup>
Perioda měření	0x9C41	Perioda měření	60 až 65535s	600s	R/W
Povolení LED indikace	0x9C42	bit 0 – povolení signalizace periody měření bit 1 – povolení signalizace vysílání bit 2 – povolení signalizace baterie pod 20% bit 3 – povolení signalizace baterie pod 5%	0 až 16	16	R/W
⋮	⋮	⋮	⋮	⋮	⋮

<sup>1)</sup> IQRF pouze R

## 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 MSB k LSB. 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	Adresa 1. cívky viz Seznam cívek	Počet cívek 1 až max 184
0x01 1 B	2 B	2 B

#### Příklad vyčtení cívky Ulož konfiguraci:

0x01	Ulož konfiguraci 0x00 0x00	Počet cívek =1 0x00 0x01

#### 2) Response PDU

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

$N = \text{Počet cívek}/8$ , je-li zbytek po dělení nenulový pak  $N = (\text{Počet cívek}/8)+1$

#### Příklad odpovědi pro neuloženou konfiguraci:

0x01	Počet bytů 0x01	Stavy cívek =0 0x00





### 3) Exception Response PDU

<b>Modbus funkce   0x80</b>	<b>Chybový kód</b>
<b>0x81</b>	<b>1,2,3 nebo 4</b>
1 B	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 kopíí požadavku.

### 1) Request PDU

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

**Příklad zapsání cívky Ulož konfiguraci na ON:**

<b>0x05</b>	<b>Ulož konfiguraci</b>	<b>Stav cívky = ON</b>
	<b>0x00 0x00</b>	<b>0xFF 0x00</b>

### 2) Response PDU

<b>Modbus funkce</b>	<b>Adresa cívky</b>	<b>Stav cívky</b>
<b>0x05</b>	<b>viz Request PDU</b>	<b>viz Request PDU</b>
1 B	2 B	2 B

**Příklad odpovědi na zapsání cívky Ulož konfiguraci na ON:**

<b>0x05</b>	<b>Ulož konfiguraci</b>	<b>Stav cívky = ON</b>
	<b>0x00 0x00</b>	<b>0xFF 0x00</b>

### 3) Exception Response PDU

<b>Modbus funkce   0x80</b>	<b>Chybový kód</b>
<b>0x85</b>	<b>1,2,3 nebo 4</b>
1 B	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 MSB k LSB. Je-li některý bajt neúplný, je doplněn 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

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

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

**Příklad zapsání cívek Ulož konfiguraci OFF, Tovární konfigurace ON**

<b>0x0F</b>	<b>Ulož konfiguraci</b> <b>0x00 0x00</b>	<b>Počet cívek = 2</b> <b>0x00 0x02</b>	<b>Počet bytů = 1</b> <b>0x01</b>	<b>Stavy cívek</b> <b>0x02</b>
-------------	---	--	--------------------------------------	-----------------------------------

### 2) Response PDU

<b>Modbus funkce</b> <b>0x0F</b> 1 B	<b>Adresa 1. cívky</b> <b>viz Request PDU</b> 2 B	<b>Počet cívek</b> <b>viz Request PDU</b> 2 B
--	---	---

**Příklad odpovědi na zapsání cívek Ulož konfiguraci OFF, Tovární konfigurace ON**

<b>0x0F</b>	<b>Ulož konfiguraci</b> <b>0x00 0x00</b>	<b>Počet cívek = 2</b> <b>0x00 0x02</b>
-------------	---	--

### 3) Exception Response PDU

<b>Modbus funkce   0x80</b> <b>0x8F</b> 1 B	<b>Chybový kód</b> <b>1,2,3 nebo 4</b> 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

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

#### Příklad vyčtení vstupních registrů CO2 a RH:

<b>0x04</b>	<b>CO2</b>	<b>Počet registrů = 2</b>
	<b>0x75    0x31</b>	<b>0x00    0x02</b>

### 2) Response PDU

<b>Modbus funkce</b>	<b>Počet bytů</b>	<b>Stavy registrů</b>
<b>0x04</b>	<b>2*N</b>	
1 B	1 B	<b>2*N B</b>

**N = Počet registrů (viz Request PDU)**

#### Příklad odpovědi na vyčtení vstupních registrů CO2 a RH:

<b>0x04</b>	<b>Počet bytů</b>	<b>CO2 = 980 ppm</b>	<b>RH = 335 ‰</b>
	<b>0x04</b>	<b>0x03    0xD4</b>	<b>0x01    0x4F</b>

### 3) Exception Response PDU

<b>Modbus funkce   0x80</b>	<b>Chybový kód</b>
<b>0x84</b>	<b>1,2,3 nebo 4</b>
1 B	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

<b>Modbus funkce</b>	<b>Adresa 1. registru viz Seznam uch. reg.</b>	<b>Počet registrů</b>
<b>0x03</b>		<b>1 až max 13</b>
1 B	2 B	2 B

### Příklad vyčtení uchovávacích registrů Modbus adresa, Perioda měření

<b>0x03</b>	<b>Modbus adresa</b>	<b>Počet registrů = 2</b>
	<b>0x9C 0x40</b>	<b>0x00 0x02</b>

### 2) Response PDU

<b>Modbus funkce</b>	<b>Počet bytů</b>	<b>Stavy registrů</b>
<b>0x03</b>	<b>2*N</b>	
1 B	1 B	<b>2*N B</b>

**N = Počet registrů (viz Request PDU)**

### Příklad odpovědi na vyčtení uchovávacích registrů Modbus adresa, Perioda měření

<b>0x03</b>	<b>Počet bytů</b>	<b>MB Adresa=1</b>	<b>Per. měření=600s</b>
	<b>0x04</b>	<b>0x00 0x01</b>	<b>0x02 0x58</b>

### 3) Exception Response PDU

<b>Modbus funkce   0x80</b>	<b>Chybový kód</b>
<b>0x83</b>	<b>1,2,3 nebo 4</b>
1 B	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

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

Příklad zapsání uchovávacího registru Perioda měření na 100s:

<b>0x06</b>	<b>Perioda měření</b>	<b>Stav registru = 100</b>
	<b>0x9C 0x41</b>	<b>0x00 0x64</b>

### 2) Response PDU

<b>Modbus funkce</b>	<b>Adresa registru viz Request PDU</b>	<b>Stav registru viz Request PDU</b>
<b>0x06</b>	<b>2 B</b>	<b>2 B</b>
1 B		

Příklad odpovědi na zapsání uchovávacího registru Per. měření na 100:

<b>0x06</b>	<b>Modbus adresa</b>	<b>Stav registru = 10</b>
	<b>0x9C 0x41</b>	<b>0x00 0x64</b>

### 3) Exception Response PDU

<b>Modbus funkce   0x80</b>	<b>Chybový kód</b>
<b>0x86</b>	<b>1,2,3 nebo 4</b>
1 B	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	Adresa 1. registru viz Seznam uch. reg.	Počet registrů	Počet bytů	Stavy registrů
<b>0x10</b>		<b>1 až max 11</b>	<b>2*N</b>	
1 B	2 B	2 B	1 B	<b>2*N B</b>

**N = Počet registrů**

**Příklad zapsání uchovávacích registrů Modbus adresa, Perioda měření:**

0x10	Modbus adresa		Počet registrů = 2		Počet bytů = 4	MB adresa=2		Per měření = 60s	
	0x9C	0x40	0x00	0x02	0x04	0x00	0x02	0x00	0x3C

### 2) Response PDU

Modbus funkce	Adresa 1. registru viz Request PDU	Počet registrů viz Request PDU
<b>0x10</b>		
1 B	2 B	2 B

**N = Počet registrů**

**Příklad odpovědi na zapsání uchovávacích registrů Modbus adresa, Perioda měření:**

0x10	Modbus adresa		Počet registrů = 2	
	0x9C	0x40	0x00	0x02

### 3) Exception Response PDU

Modbus funkce   0x80	Chybový kód
<b>0x90</b>	<b>1,2,3 nebo 4</b>
1 B	1 B



## (0x42) Funkce Čti vstupní registry v náhodném pořadí 1

Tato funkce slouží ke čtení až 6 vstupních registrů. V požadavku je specifikován počet registrů a adresy jednotlivých registrů. V odpovědi pak počet registrů, adresa registru a příslušící hodnota registru.

**Pro nové implementace doporučujeme využít modbus funkci 0x43!**

### 1) Request PDU

<b>Modbus funkce</b>	<b>Počet registrů</b>	<b>Adresa 1. registru viz Seznam vst. reg.</b>	...	<b>Adresa n. registru viz Seznam vst. reg.</b>
<b>0x42</b>	<b>1 až max 6</b>			
1 B	2 B	2 B		2 B

**Příklad vyčtení vstupních registrů CO2 a T:**

<b>0x42</b>	<b>Počet registrů = 2</b>	<b>CO2</b>		<b>T</b>	
	<b>0x00    0x02</b>	<b>0x75    0x31</b>	<b>0x75    0x33</b>		

### 2) Response PDU

<b>Modbus funkce</b>	<b>Počet registrů viz Request PDU</b>	<b>Adresa 1. registru viz Request PDU</b>	<b>Stav 1. registru</b>	...	<b>Adresa n. registru viz Request PDU</b>	<b>Stav n. registru</b>
<b>0x42</b>						
1 B	2 B	2 B	2 B		2 B	2 B

### 3) Exception Response PDU

<b>Modbus funkce   0x80</b>	<b>Chybový kód</b>
<b>0xC2</b>	<b>1,2,3 nebo 4</b>
1 B	1 B



## (0x43) Funkce Čti vstupní registry v náhodném pořadí 2

Tato funkce vznikla optimalizací předcházející funkce 0x42. Funkce slouží ke čtení až 13 vstupních registrů. V požadavku je specifikován počet registrů a adresy jednotlivých registrů (s offsetem 0x7530). V odpovědi je pak počet registrů, odpovídající hodnoty registrů.

### 1) Request PDU

<b>Modbus funkce</b>	<b>Počet registrů</b>	<b>Adresa 1. registru</b> viz Seznam vt. reg.	...	<b>Adresa n. registru</b> viz Seznam vt. reg.
<b>0x43</b>	<b>1 až max 13</b>			
1 B	1 B	1 B		1 B

### Příklad vyčtení vstupních registrů CO2 a T:

<b>0x43</b>	<b>Počet registrů = 2</b> <b>0x02</b>	<b>CO2 <sup>1)</sup></b> <b>0x01</b>	<b>T</b> <b>0x03</b>
-------------	--	---	-------------------------

<sup>1)</sup> Adresa CO2 0x7531 = offset 0x7531 + 0x01

### 2) Response PDU

<b>Modbus funkce</b>	<b>Počet registrů</b> viz Request PDU	<b>Stav</b> <b>1. registru</b>	...	<b>Stav n. registru</b>
<b>0x43</b>				
1 B	1 B	2 B		2 B

### Příklad odpovědi na vyčtení vstupních registrů CO2 a T:

<b>0x43</b>	<b>Počet registrů = 2</b> <b>0x02</b>	<b>CO2 =980ppm</b> <b>0x03 0xD4</b>	<b>T=21,5°C</b> <b>0x00 0xD7</b>
-------------	--	--	-------------------------------------

### 3) Exception Response PDU

<b>Modbus funkce   0x80</b>	<b>Chybový kód</b>
<b>0xC2</b>	<b>1,2,3 nebo 4</b>
1 B	1 B

