

MODBUS Client

Protokol MODBUS Client

[Podporované typy a verzie zariadení](#)
[Konfigurácia komunikačnej linky](#)
[Parametre protokolu linky](#)
[Konfigurácia komunikačnej stanice](#)
[Konfigurácia meraných bodov](#)
[Poznámka k zariadeniu FloBoss 103](#)
[Poznámka k regulátorom Honeywell](#)
[Literatúra](#)
[Zmeny a úpravy](#)
[Revízie dokumentu](#)

Podporované typy a verzie zariadení

Protokol realizuje klientsku (master) komunikáciu s ubovonými zariadeniami podporujúcimi štandard **MODBUS RTU** a **MODBUS ASCII** vo verziách sériovej komunikácie a štandard **MODBUS TCP** alebo variantu **MODBUS over TCP** v prípade TCP/IP komunikácie. Navyše podporuje rozšírenia:

- **Bytový mód** umožňujúci pracovať so zariadeniami, ktoré vracajú hodnoty registrov ako 1 bajt (na rozdiel od MODBUS štandardu, v ktorom je hodnota registra 2-bajtová).
- **Variabilný mód** umožňujúci pracovať so zariadeniami, ktoré vracajú hodnoty registrov s inou veľkosťou ako sú štandardné 2 bajty. Implementovaný bol kvôli podpore prietokomera FloBoss 103 od firmy Fisher Controls International (v súčasnosti súčasť Emerson Process Management): 1-bajtové premenné, 4-bajtové unsigned/signed integrity, textové reakcie dĺžky 10,12,20,40 znakov, 6-bajtová asová znakov a iné.

Konfigurácia komunikačnej linky

- Kategória linky [Serial](#) (sériová komunikácia)
- Kategória linky [SerialOverUDP Device Redundant](#) (sériová komunikácia).
- Kategória linky [RFC2217 Client](#) (sériová komunikácia).
- Kategória linky [TCP/IP-TCP](#) a [TCP/IP-TCP Redundant](#) (MODBUS TCP, MODBUS over TCP). Bežne sa používa rezervovaný TCP port číslo 502, ale je možné použiť akýkoľvek iný podľa nastavenia komunikujúceho zariadenia. číslo linky je nepoužívané, nastavte napr. hodnotu 1.
Pozn: V prípade redundantných systémov je možné zadávať aj viacero mien/adries oddelených iarkami.
Pozn: V prípade PLC typu WAGO 750-8100 a komunikácie cez MODBUS TCP bolo nutné nastaviť v asových parametroch stanice malú periódu dotazovania (napr. 1 sekunda). V prípade vašej periódy (5 sekúnd) dochádzalo pomerne často k zatváraní spojenia zo strany PLC.

Parametre protokolu linky

Dialóg [konfigurácia linky](#) - záložka **Parametre protokolu**.

Ovplyvňujú niektoré voliteľné parametre protokolu. Môžu byť zadane nasledovné parametre protokolu linky:

Tab. . 1

Parameter	Popis	Jednotka / rozmer	Náhradná hodnota
Immediate Disconnect	Parameter je implementovaný iba pre kategórie linky TCP/IP-TCP a TCP/IP-TCP Redundant . Parameter aktivuje rozpájanie sa TCP spojenia po vykonaní každého itacieho cyklu, prípadne po zápise hodnoty. Parameter bol implementovaný kvôli problémom so stabilitou spojenia na mobilných GPRS sieach.	YES/NO	NO
Tcp No Delay	Nastavenie "Tcp No Delay"=True parametra spôsobí nastavenie nízkoúrovňového parametra socketov TCP_NODELAY, čím sa vypne prednastavené spájanie paketov. Parameter je implementovaný iba pre kategórie linky TCP/IP-TCP a TCP/IP-TCP Redundant .	YES/NO	NO

Konfigurácia stanice

- Komunikovaný protokol "**Modbus Client**".
- Adresa stanice je dekadické číslo zvyčajne v rozsahu 1 až 247. Adresa 0 je rezervovaná ako broadcast.

Parametre protokolu stanice

Dialóg [konfigurácia stanice](#) - záložka **Parametre protokolu**.

Ovplyvňujú niektoré voliteľné parametre protokolu. Môžu byť zadane nasledovné parametre protokolu stanice:

Tab. . 1

Parameter	Popis	Jednotka	Náhradná hodnota
Retry Count	Maximálny počet opakovaní výzvy. Ak po poslaní tohto potu výziev nedôjde odpoveď, stanica prechádza do stavu komunikanej chyby.	s	2
Retry Timeout	Timeout pred opakovaním výzvy, pokiaľ neprišla odpoveď.	s	0.1
Wait First Timeout	Timeout medzi poslaním výzvy a prvým ítaním odpovede.	s	0.1
Wait Timeout	Timeout medzi ítaniami odpovede.	s	0.1
Max. Wait Retry	Maximálny počet opakovaní ítania odpovede.	-	20
Start Silent Interval	"Start silent interval" pred štartom vysielania v RTU móde.	ms	50
Stop Silent Interval	"Stop silent interval" po ukončení vysielania v RTU móde.	ms	50
Little Endian Mode	Poradie bajtov v Little-endian móde pre 4-bajtové premenné. Jednotlivé možnosti udávajú, do ktorých bajtov (1-najnižší, 4-najvyšší) pôjdu jednotlivé bajty z komunikácie: • 2143 - najskôr je prijaté nižšie slovo, potom vyššie slovo (vyšší bajt v rámci slova vždy ako prvý) • 3412 - najskôr je prijaté vyššie slovo, potom nižšie slovo (nižší bajt v rámci slova vždy ako prvý) • 1234 - bajty sú prijaté od najnižšieho po najvyšší (opak Big-endianu)	-	2143
Byte mode	Špeciálny bajtový mód prenosu, v ktorom majú hodnoty registrov dĺžku 1 bajt a nie 2 byty ako je definované v špecifikácii protokolu MODBUS.	YES/NO	NO
Variable mode	Špeciálny variabilný mód prenosu, v ktorom majú hodnoty registrov variabilnú dĺžku. Podľa toho, či je <i>Variable mode</i> nastavený na hodnotu <i>little-endian</i> alebo <i>big-endian</i> sa aplikuje príslušné dátové kódovanie t.j. tzv. endianness (<i>little-endian</i> =prvý je odosielaný menej významný byte, <i>big endian</i> =prvý je odosielaný významnejší byte). Hodnota <i>OFF</i> vypína variabilný mód. Poznámka 1: Variabilný a bajtový mód sú nezlúčené a je dovolené zapnúť iba jedného z nich. Poznámka 2: Zariadenie Emerson FloBoss 103: textové reakcie a 6-bajtová asová znaka sa posielajú vždy od najnižšieho bajtu. Poznámka 3: Variabilný mód je implementovaný iba pre Protocol Mode=RTU. Poznámka 4: Standardne podľa implicitných hodnôt parametrov Byte mod=NO a Variable mode=OFF, teda podľa špecifikácie protokolu MODBUS, sa automaticky uplatňuje kódovanie dát <i>big-endian</i> .	OFF little-endian big-endian	OFF
Full debug	Výpis dodatkových ladiacich informácií o komunikácii na linke.	YES/NO	NO
Protocol mode	Mód protokolu: RTU alebo ASCII . Nastavenie sa aplikuje iba v prípade sériovej komunikácie.	"RTU " "ASCII"	"RTU"
Addressing model	Nastavenie adresného modelu protokolu MODBUS: "MODBUS PDU" dáta sú adresované od 0 do 65535. "MODBUS data Model" dáta sú adresované od 1 do 65536. Poznámka: Implicitná hodnota je <i>MODBUS PDU</i> , v prípade nastavenia <i>MODBUS data Model</i> je objekt s adresou X adresovaný v <i>MODBUS PDU</i> ako X-1. Po zmene nastavenia tohto parametra reštartujte príslušný komunikovaný proces.	"MODBUS PDU" "MODBUS data Model"	"MODBUS PDU"
TCP/IP protocol variant	Výber varianty protokolu v prípade TCP/IP komunikácie: "MODBUS TCP" je variant komunikácie bez zabezpečenia kontrolnou sumou. Zabezpečenie prenosu vykonávajú spodné vrstvy TCP protokolu. "MODBUS over TCP" je variant, v ktorej je ako payload prenosu použitý mód protokolu MODBUS RTU aj s kontrolnou sumou.	"MODBUS TCP" "MODBUS over TCP"	"MODBUS TCP"
Max. Registers	Maximálny počet registrov žiadaný v jednej výzve.	-	100
Max. Bytes	Maximálny počet bytov žiadaný v jednej výzve (iba v "Byte mode").	-	100
Skip Unconfigured	Nikdy nežiadať hodnoty z adries, ktoré nie sú nakonfigurované. Popis a príklad: Standardne sa posielajú výzvy na dáta, ktoré sa obmedzujú parametrom protokolu "Max. Registers" alebo "Max. Bytes". Ak sú nakonfigurované napr. merané body s adresami "Holding Registers" 1, 2 a 5, odosiela sa jedna výzva požadujúca 5 registrov od adresy 1 aj, keď merané body s adresami 3 a 4 nie sú nakonfigurované - pretože je efektívnejšie a rýchlejšie získa požadované údaje jednou výzvou ako dvomi aj za cenu ítania nepotrebných dát. V prípade nastavenia parametra "Skip Unconfigured" na YES sa odošlú dve výzvy, prvá žiadajúca dva registre od adresy 1 a druhá žiadajúca jeden register na adrese 5.	YES/NO	NO

Check Receive Length	Ak je parameter nastavený na YES, tak pri prijatí odpovede na ítanie dát sa kontroluje, i džka dát v odpovedi zodpovedá množstvu registrov v požiadavke na ítanie: - ak je zapnutý bajtový mód (Byte mode=YES), tak džka prijatých dát sa musí rovna množstvu registrov - ak nie je zapnutý bajtový ani variabilný mód, tak džka prijatých dát sa musí rovna dvojnásobku množstva registrov - ak je zapnutý variabilný mód (Variable mode=little-endian alebo big-endian), kontrola zatia nie je implementovaná Kontrola má zmysel na linkách s vysokými (a variabilnými) latenciami - napr. GPRS siete - na detekciu a vyhnutie sa situácii, ke dôjde k opakovaniu výzvy (#1) v dôsledku timeoutu a následne k príjmu odpovede na opakovanú výzvu, priom táto je už považovaná za odpoveď na alšiu výzvu (#2). To spôsobí, že merané body adresované výzvou #2 získajú chybné hodnoty.	YES/NO	NO
----------------------	---	--------	----

Konfigurácia meraných bodov

Možné typy hodnôt bodov pre nevariabilný mód: **Ai, Ao, Di, Do, Ci, Co, Txtl**.

Možné typy hodnôt bodov pre variabilný mód: **Ai,Ao,Di,Do,Ci,Cout,Txtl,TxtO,TiA**.

Adresa meraného bodu:

V protokole MODBUS je základný adresný priestor rozdelený na objekty typov:

- Coils (ítanie/zápis) - binárne stavy.
- Discrete Inputs (ítanie) - binárne vstupy.
- Holding Registers (ítanie/zápis) - stavové registre.
- Input Registers (ítanie) - vstupné registre.

V každom adresnom priestore daného typu registra je nezávislá adresácia s vekosou adresy 2 bajty, to znamená fyzicky adresy od 0 do 65535 (tzv. *MODB US PDU* addressing model). Niektoré zariadenia pracujú s adresáciou od 1 (tzv. *MODBUS data Model*), v takom prípade treba pri konfigurácii meraných bodov v systéme D2000 odíta v adrese -1 alebo zmeni nastavenie parametra protokolu [Addressing model](#) na *MODBUS data Model*.

Adresa meraného bodu môže mať **základný** alebo **rozšírený** tvar (pre variabilný mód).

Základný tvar adresy meraného bodu:

Tvar adresy je $[I|U|Uu|U|f|F|L|L|S|S|B|X|sn].[an].[An].[d|D][b][s]RdFn[-WrFn[d]].Address[.BitNr]$ kde:

- Prvé písmeno (písmená) udáva typ meraného bodu:
 - I** - Integer16 (implicitná hodnota, ke nie je uvedené inak) - jeden register znamienkovo.
 - U** - Unsigned16 - jeden register neznamienkovo.
 - Uu** - Unsigned16 - jeden register neznamienkovo, priom sa z neho berie do úvahy iba vyšší bajt (1. v poradí)
 - Ui** - Unsigned16 - jeden register neznamienkovo, priom sa z neho berie do úvahy iba nižší bajt (2. v poradí)
 - f** - Float (4 byty = 2 registre) - ítajú sa 2 registre s adresou *Address* a *Address+1*, prenášané v big-endian poradí (vi [Poznámku](#)).
 - F** - Float (4 byty = 2 registre) - ítajú sa 2 registre s adresou *Address* a *Address+1*, prenášané v little-endian poradí (tzv. Modicon formát) (vi [Poznámku](#)).
 - L** - Unsigned long (4 byty = 2 registre) - ítajú sa 2 registre s adresou *Address* a *Address+1*, neznamienkovo, prenášané v big-endian poradí (vi [Poznámku](#)).
 - LI** - Unsigned long (4 byty = 2 registre) - ítajú sa 2 registre s adresou *Address* a *Address+1*, neznamienkovo, prenášané v little-endian poradí (vi [Poznámku](#)).
 - S** - Signed long (4 byty = 2 registre) - ítajú sa 2 registre s adresou *Address* a *Address+1*, znamienkovo, prenášané v big-endian poradí (vi [Poznámku](#)).
 - SI** - Signed long (4 byty = 2 registre) - ítajú sa 2 registre s adresou *Address* a *Address+1*, znamienkovo, prenášané v little-endian poradí (vi [Poznámku](#)).
 - B** - Byte neznamienkovo, iba horných 8 bitov hodnoty registra.
 - X** - Byte neznamienkovo, iba dolných 8 bitov hodnoty registra.
 - sn** - Textový string s dĺžkou *n* znakov, jeden register je jeden znak, íta sa *n* registrov s adresou *Address* až *Address+n-1*.
 - an** - Textový string s dĺžkou *2*n* znakov, jeden register sú dva ASCII znaky, znaky sú prenášané v poradí v akom sú v reakcii, íta sa *n* registrov s adresou *Address* až *Address+n-1*.
 - An** - Textový string s dĺžkou *2*n* znakov, jeden register sú dva ASCII znaky, znaky sú prenášané v big-endian poradí (tj. "1234" sa prenáša ako "2143"), íta sa *n* registrov s adresou *Address* až *Address+n-1*.
- Modifikátor **d** hovorí, že íslo je 8-bajtové íslo (4 za sebou idúce registre). Je použitý pri typoch *L, LI, S, SI, F, f* a pomocou neho je možné zdefinovať 8-bajtový integer bez znamienka/so znamienkom ako aj 8-bajtový float (varianty big endian <B8>...<B1> a little endian <B1>...<B8>). Modifikátor **D** hovorí, že íslo je 8-bajtové íslo (4 za sebou idúce registre). Je použitý pri typoch *LI, SI, F* a pomocou neho je možné zdefinovať 8-bajtový integer bez znamienka/so znamienkom ako aj 8-bajtový float (vo formáte little endian <B2><B1><B4><B3><B6><B5><B8><B7>).
Poznámka: pri použití modifikátorov **d** a **D**, musí byť meraný bod Analógového typu (Ai), pretože celoíselný typ (Ci) v D2000 je implementovaný ako 4-bajtová premenná a mohlo by dôjsť k pretečeniu. Zápis celoíselnej hodnoty (Co) ako 8-bajtového typu je podporený.
- Modifikátor **b** hovorí, že íslo je BCD kódované. Je použitý pri meraných bodoch typu *I, U, B, L, LI*.
- Modifikátor **s** hovorí, že sa íta register so stavom premennej (Unsigned16) na adrese *Address* a za ním Float hodnota v big endian formáte na adrese *Address+1 .. Address+2*. Je použitý pri meraných bodoch typu *f* a implementovaný kvôli kalorimetru Endress+Hauser RMS621. Tabuľka uvádza hodnoty Status a prevod na D2000 príznaky.

Status register	D2000 príznaky
0 : Invalid value	Weak

1 : Measured value valid	Valid
2 : Overflow warning 3 : Overflow error 4 : Underflow warning 5 : Underflow error 6 : Saturated steam alarm 7 : Error in differential pressure calculation 8 : Wrong medium for DP calculation 9 : Wrong value range - DP calculation inaccurate 10 : Differential pressure - general error 11 : Range overshoot (Tsat > 350 etc.) on 12 : Change in state of aggregation 26 : Differential pressure --> general error 99 : No measured value is assigned to the register in the setup of the ModBus	Weak

- Parameter **RdFn** je funkcia Modbus protokolu pre ítanie údajov. Implementované sú nasledovné funkcie:
 - 1 - Read Coils: ítanie binárnych stavov.
 - 2 - Read Discrete Inputs: ítanie binárnych vstupov.
 - 3 - Read Holding Registers: ítanie stavových registrov (Integer16/Unsigned16 a Float32 - íta 2 za sebou nasledujúce registre).
 - 4 - Read Input Registers: ítanie vstupných registrov (Integer16/Unsigned16 a Float32 - íta 2 za sebou nasledujúce registre).
 - 0 - V tomto prípade nebude vykonávané ítanie hodnoty ale iba zápis, je nutné aby bola nastavená funkcia pre zápis **WrFn**.
- Parameter **WrFn** je funkcia Modbus protokolu pre zápis údajov. Implementované sú nasledovné funkcie:
 - 5 - Write Single Coil: zápis binárnych stavov (implicitne pre *Read Coils*).
 - 6 - Write Single Register: zápis stavových registrov (implicitne pre *Read Holding Registers*).
 - 16 - Write Multiple Registers: zápis viacerých stavových registrov, táto funkcia musí by použitá pri zápise dvojregistrových typov (ako *Float*, *Unsigned long* at.).
Pozn: funkciu je možné použiť aj na zápis viac ako dvoch registrov naraz, pokiaľ sa použije textový string. Príklad: ak do textového meraného bodu s adresou a3.0-16.#8A00 (t.j. textový reazec pokrývajúci 3 registre, t.j. majúci dĺžku 6 znakov) zapíšeme reazec '123456', tak sa zapíšu do registrov 0x8A00 až 0x8A02 hexadecimalne hodnoty 0x3132, 0x3334 a 0x3536 (ASCII kód '1' je 0x31, '2' je 0x32 at.).
 - 22 - Mask Write Register: zápis ovplyvňujúci iba hodnotu konkrétneho bitu *BitNr* stavového registra. Možno použiť iba pre hodnoty typu *Do* s nastaveným parametrom adresy *BitNr*.
- Parameter **d** aktivuje funkciu "delayed write". Odoslanie hodnoty sa oneskorí až do príchodu požiadavky na zápis hodnoty objektu bez tohto parametra *d*. Všetky naakumulované požiadavky na zápis sa následne odošlú a ak je nastavená funkcia na zápis **WrFn** na "Write Multiple Registers" tak sa snaží odoslať hodnoty v jednom pakete.
- Parameter **Address** je dvojbyťová adresa registra (0-65536), vi tiež parameter protokolu [Addressing model](#).
Pozn: adresu je možné zadávať aj v hexadecimalnom tvare za použitia mriežky (#), napr. #50CE
- Parameter **BitNr** je číslo bitu v slove. Pre binárne stavy a vstupy sú prípustné hodnoty 0-7, pre vyítanie bitu zo 16-bitových stavových alebo vstupných registrov sú prípustné hodnoty 0-15.

Poznámka ku poradiu prenášaných bytov a registrov

1. Protokol MODBUS používa štandardne tzv. big-endian, iže významnejší byte registra (MSB) je prenášaný ako prvý. Príklady:

Prijaté byty registra MSB-LSB	Typ meraného bodu	Hodnota
0x00 0x01	I, U	1
0xFF 0xFE	I	-2
0xFF 0xFE	U	65534
0x01 0x02	B	1
0x01 0x02	X	2

2. V prípade ítania hodnôt z dvoch registrov v poradí big-endian sú prijaté byty analyzované takto:

Most significant register (Adresa ADR)		Least significant register (Adresa ADR+1)	
MSB	LSB	MSB	LSB

Príklady pre dvojregistrové hodnoty v big-endian poradí:

Prijaté byty registra (MSB-LSB)	Prijaté byty registra + 1 (MSB-LSB)	Typ meraného bodu	Hodnota
0x00 0x00	0x00 0x01	L, S	1
0xFF 0xFF	0xFF 0xFE	S	-2
0x00 0x01	0x00 0x02	L, S	65538
0x3F 0x80	0x00 0x00	f	1.0
0xC0 0x00	0x00 0x00	f	-2.0

3. V prípade íťania hodnôt z dvoch registrov v poradí little-endian sú prijaté byty analyzované takto:

Least significant register (Adresa ADR)		Most significant register (Adresa ADR+1)	
MSB	LSB	MSB	LSB

Príklady pre dvojregistrové hodnoty v little-endian poradí, ak **Little Endian Mode=2143**

:

Prijaté byty registra (MSB-LSB)	Prijaté byty registra + 1 (MSB-LSB)	Typ meraného bodu	Hodnota
0x00 0x01	0x00 0x00	LI, SI	1
0xFF 0xFE	0xFF 0xFF	SI	-2
0x00 0x02	0x00 0x01	LI, SI	65538
0x00 0x00	0x3F 0x80	F	1.0
0x00 0x00	0xC0 0x00	F	-2.0

Príklady konfigurácie:

- **1.10** - íťanie hodnoty binárneho stavu s adresou 10 funkciou *Read Coils*.
- **3.1** - 16-bitové číslo znamienkové vyíťávané funkciou *Read Holding Registers* z adresy 1 (tiež ako *I3.1*).
- **U3.1** - 16-bitové číslo bez znamienka vyíťávané funkciou *Read Holding Registers* z adresy 1.
- **I3-6.1000** - 16-bitové číslo so znamienkom vyíťávané funkciou *Read Holding Registers* z adresy 1000 a zapisované funkciou *Write Single Register* (keže táto funkcia je implicitná, adresa mohla by aj *I3.1000* alebo *3.1000*).
- **S3.321** - 32-bitové znamienkové číslo íťané funkciou *Read Holding Registers* z registrov 321 a 322.
- **B1.20.0** - bit vyíťávaný funkciou *Read Coils* z adresy 20 ako nultý bit v bajte.
- **s10.3.123** - textový string o dĺžke 10 znakov (2 bajty na znak) od adresy 123 íťaný funkciou *Read Holding Registers*.
- **a5.3.123** - textový string o dĺžke 10 znakov (1 bajt na znak) od adresy 123 íťaný funkciou *Read Holding Registers*.
- **U0-6.456** - iba zápis 16-bitového neznamienkovej hodnoty do registra 456, zápis realizovaný funkciou *Write Single Register*, íťanie hodnoty registra sa nerealizuje.

Rozšírený tvar adresy meraného bodu:

Tvar adresy je $[xN][I|U|F|B|C|T][b]RdFn[-WrFn].Address[.BitNr]$ kde:

- xN hovorí o pote bajtov, ktoré sa íťajú, resp. zapisujú. Platné hodnoty N sú 1,2,4 (v kombinácii s typmi *I,U,F*), 6 pre typ *T* a ubovoné číslo pre typ *C*.
- Písmeno udáva typ meraného bodu. Oproti štandardným *I,U,F,B* pribudli ďalšie dva:
 - **C** - textový reazec pevnej dĺžky (napr. *x10.C3.1001* je 10-znakový reazec na adrese 1001)
 - **T** - asová znaka s dĺžkou 6 bajtov (ss:mi:hh dd:mm:yy)
- Význam ostatných parametrov je zhodný s nevariabilným módom.

Príklady konfigurácie sú uvedené alej v [Poznámke k zariadeniu FloBoss 103](#).

Poznámka k zariadeniu FloBoss 103

- konfiguraný softvér ROCLINK800
- default login LOI, heslo 1000
- nadviazanie spojenia s FloBoss 103: poklika na DirectConnect (máme pripojené cez COM1, na strane FloBoss 103 pripojený k LOI-local interface)
- menu *Configure->Modbus->Configuration*

poda nastavenia "Byte Order" nastavi parameter "Variable Mode" na stanici v D2000:

 - ak "Least Significant Byte first", tak "Little endian"
 - ak "Most Significant Byte first", tak "Big endian"
- merané body sú na strane FloBoss 103 konfigurované cez menu *Configure -> Modbus -> Registers*
- podporované sú nasledovné typy (v alšom texte *n* oznauje 16-bitovú adresu):
 - Binárny vstup:
 - adresa v D2000: *1.n*, napr. *1.1001*, premenná typu Di/Dout
 - adresa v FloBoss 103: premenná typu *BIN*

Funkcia: 1
Starting/ending register: *n*
 - Binárny výstup:
 - adresa v D2000: *1.n*, napr. *1.1001*, premenná typu Dout
 - adresa v FloBoss 103: premenná typu *BIN r/w*

Funkcia: 1 (kvôli íťaniu)
Starting/ending register: *n*

- Funkcia: 5 (kvôli zápisu)
Starting/ending register: n
- Unsigned Int 8 bitov vstup:
 - adresa v D2000: $x1.B3.n$, napr. $x1.B3.1003$, premenná typu Ci/Co
 - adresa v FloBoss 103: premenná typu $U\text{INT}8$
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Unsigned Int 8 bitov výstup:
 - adresa v D2000: $x1.B3.n$, napr. $x1.B3.1003$, premenná typu Co
 - adresa v FloBoss 103: premenná typu $U\text{INT}8$ r/w
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Funkcia: 6
 - Starting/ending register: n
 - Unsigned Int 16 bitov vstup:
 - adresa v D2000: $x2.U3.n$, napr. $x2.U3.1004$, premenná typu Ci/Co
 - adresa v FloBoss 103: premenná typu $U\text{INT}16$
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Unsigned Int 16 bitov výstup:
 - adresa v D2000: $x2.U3.n$, napr. $x2.U3.1004$, premenná typu Co
 - adresa v FloBoss 103: premenná typu $U\text{INT}16$ r/w
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Funkcia: 6
 - Starting/ending register: n
 - Signed Int 16 bitov vstup:
 - adresa v D2000: $x2.I3.n$, napr. $x2.I3.1005$, premenná typu Ci/Co
 - adresa v FloBoss 103: premenná typu $I\text{NT}16$
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Signed Int 16 bitov výstup:
 - adresa v D2000: $x2.I3.n$, napr. $x2.I3.1005$, premenná typu Co
 - adresa v FloBoss 103: premenná typu $I\text{NT}16$ r/w
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Funkcia: 6
 - Starting/ending register: n
 - Unsigned Int 32 bitov vstup:
 - adresa v D2000: $x4.U3.n$, napr. $x4.U3.1006$, premenná typu Ci/Co
 - adresa v FloBoss 103: premenná typu $U\text{INT}32$
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Unsigned Int 32 bitov výstup:
 - adresa v D2000: $x4.U3.n$, napr. $x4.U3.1006$, premenná typu Co
 - adresa v FloBoss 103: premenná typu $U\text{INT}32$ r/w
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Funkcia: 6
 - Starting/ending register: n
 - Float 32 bitov vstup:
 - adresa v D2000: $x4.F3.n$, napr. $x4.F3.1008$, premenná typu Ai/Ao
 - adresa v FloBoss 103: premenná typu FL
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Float 32 bitov výstup:
 - adresa v D2000: $x4.F3.n$, napr. $x4.F3.1008$, premenná typu Co
 - adresa v FloBoss 103: premenná typu FL r/w
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Funkcia: 6
 - Starting/ending register: n
 - String (N bajtov) vstup:
 - adresa v D2000: $x1N.C3.n$, napr. $x10.C3.1010$, premenná typu TxtI/TxtO
 - adresa v FloBoss 103: premenná typu $ACm(AC10,AC12,AC20,AC30,AC40)$
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - String (N bajtov) výstup:
 - adresa v D2000: $xN.C3.n$, napr. $x10.C3.1010$, premenná typu Co
 - adresa v FloBoss 103: premenná typu ACN r/w ($AC10,AC12,AC20,AC30,AC40$)
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n
 - Funkcia: 6
 - Starting/ending register: n
 - as a dátum 6 bajtov vstup:
 - adresa v D2000: $x6.T3.n$, napr. $x6.T3.1010$, premenná typu TiA/TxtI
 - adresa v FloBoss 103: premenná typu $DT6$
 - Funkcia: 3A alebo 3B
 - Starting/ending register: n

- **Poznámka 1:** FloBoss 103 podporuje lokálny aj monotónny as - preto musí konfigurácia stanice v D20000 zodpoveda konfigurácii FloBoss-u
- **Poznámka 2:** Nastavenie asu a dátumu je možné, ale treba nakonfigurovať zvlášť merané body pre sekundu, minútu, hodinu, deň, mesiac a rok ako *Unsigned Int 8 bitov* a následne zapisovať do nich.

Poznámka k regulátorom Honeywell

Základné parametre a aktuálne dáta týchto regulátorov nie sú štandardne ítané prostredníctvom funkcií 0x01 až 0x04. Na ich ítanie a zápis je potrebné použiť funkcie 0x14/0x15 Read / write configuration reference data. Tieto zariadenia štandardne používajú "big endian" poradie bytov. Preto pre správnu funkciu nie je potrebné meniť parametre, ktoré menia bytový režim a endianitu.

Príklady konfigurácie meraného bodu:

20.039 - 16-bitové číslo z adresy 39(0x27)

f20.040 - 32-bitové reálne číslo z adresy 40(0x28)

Literatúra

- MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b, December 28, 2006. <http://www.modbus.org>



Blog

O protokole Modbus si môžete prečítať aj blogy

- [Komunikácia – Modbus protokol](#)
- [Komunikácia – Modbus protokol, časť 2](#)

Zmeny a úpravy

-

Revízie dokumentu

- Ver. 1.0 - 27. november 2006 - Vytvorenie dokumentu.
- Ver. 1.1 - 21. november 2007 - Aktualizácia dokumentu.
- Ver. 1.2 - 24. apríl 2009 - Aktualizácia dokumentu.
- Ver. 1.3 - 19. november 2010 - Aktualizácia dokumentu.
- Ver. 1.4 - 6. december 2010 - Aktualizácia dokumentu.



Súvisiace stránky:

[Komunikované protokoly](#)