ModBus – Protokol

Petr Novák / novakpe@fel.cvut.cz / 2021-05-06

Obsah

1 Úvod

Modbus je celkem jednoduchý protokol pro čtení a zápis do (registrů) vzdáleného zařízení. Je založen na principu jeden „master“ a několik „slave“ zařízení. Komunikaci vždy řídí master a slave pouze odpovídá na dotaz (nikdy sám bez vyzvání nevysílá).
Přenášený paket neobsahuje žádný úvodní ani koncový znak. Konec paketu se detekuje podle odmlky (nevysílání) na dobu 1.5 až 2 vysílané bytes.
Existuje (stručně) několik typů Modbus:

- **Modbus RTU** – Binární běžný sériový přenos (RS232/RS485/…).
- **Modbus ASCII** – Obdoba Modbus RTU, ale přenos pomocí AsciiHex formátu.
- **Modbus TCP** – Klasický Ethernet TCP/IP.
- … (některé další modifikace)

2 Protokol

Přenášená data / pakety se skládají z těchto částí:

Adresa zařízení	Povel / Kód funkce	Data	CRC

Adresa zařízení může mít hodnotu:

- **0 až 247** – Běžná zařízení
- **248 až 255** – Vyhrazené adresy

Data:

- Pokud jsou data více bytová, tak je vždy přenášen nejvyšší byte jako první a nejnižší byte jako poslední.

Komunikace s koncovým (slave) zařízením probíhá pomocí čtení a zápisu hodnot / registrů. Tyto hodnoty / registry jsou rozděleny do čtyř tabulek (v každé je 9999 hodnot / registrů / položek):

Číslo registru	Adresa (Hex)	Přístup	Název	Typ
1 - 9999	0000 – 270E	čtení / zápis	diskrétní výstup(y)	DO
10001 - 19999	0000 – 270E	čtení	diskrétní vstup(y)	DI

30001 - 39999	0000 – 270E	čtení	analogové výstup(y)	AI
40001 - 49999	0000 – 270E	čtení / zápis	analogové vstup(y)	AO

V komunikaci se přenáší povel, který současně určuje typ registru v tabulce. Proto v datech se již přenáší adresa registru pouze v rozsahu 0000 – 9999. Význam zapisovaných / četných hodnot:

- DO / DI – Digitální bitové hodnoty. Při přenos jsou komprimovány do byte zprava.
- AO / AI – Analogové 16bits / 2B hodnoty.

Základní nejpoužívanější příkazy / kódy protokolu (Data Access):

Povel / Kód	**Činnost**	**Typ hodnoty**	**Typ přenosu**
01 (0x01)	Čtení DO (nastavený digitální výstup/y)	bit	čtení
02 (0x02)	Čtení DI (aktuální digitální vstup/y)	bit	čtení
03 (0x03)	Čtení AO (nastavený analogový výstup/y)	16bits / 2B	čtení
04 (0x04)	Čtení AI (aktuální analogový vstup/y)	16bits / 2B	čtení
05 (0x05)	Zápis jednoho DO (nastavení digitálního výstupu)	bit	zápis
06 (0x06)	Zápis jednoho AO (nastavení analogového výstupu)	16bits / 2B	zápis
15 (0x0F)	Zápis více DO (nastavení digitálních výstupů)	bit(s)	zápis
16 (0x10)	Zápis více AO (nastavení analogových výstupů)	16bits / 2B	zápis
23 (0x17)	Read/Write Multiple Registers		
22 (0x16)	Mask Write Register		
24 (0x18)	Read FIFO Queue		

Některé další příkazy / kódy protokolu (jen stručně):
File records
20 (0x14) Read File Record
21 (0x15) Write File Record
Diagnostics
7 (0x07) Read Exception Status (serial only)
8 (0x08) Diagnostic (serial only)
11 (0x0B) Get Com Event Counter (serial only)
12 (0x0C) Get Com Event Log (serial only)
17 (0x11) Report Slave ID (serial only)
43 (28)? Read Device Identification
Other
43 (28)? Encapsulated Interface Transport

Z tabulky je zřejmě jak probíhá přenos dat do zařízení / slave. Význam některých označení:

- **DO** – Digitální výstup typu „1bit“. Zápisem je nastavena hodnota digitálního výstupu, čtením se získá aktuálně nastavená hodnota výstupu / výstupního registru (nikoli nějakého vstupu).
- **DI** – Digitální vstup typu „1bit“. Nelze zapisovat. Čtením se získá aktuální stav (externího) skutečného digitálního vstupu.
- **AO** – Analogový výstup typu „16bits/2B“. Zápisem je nastavena hodnota analogového výstupu (DA převodníku), čtením se získá aktuálně nastavená hodnota (DA převodníku) výstupu / výstupního registru (nikoli nějakého vstupu).

- **AI** – Analogový vstup typu „16bits/2B“. Nelze zapisovat. Čtením se získá aktuální stav (externího) skutečného analogového vstupu (hodnota z AD převodníku).

Registry DO / DI / AO / AI lze v podstatě využít zcela podle vlastní potřeby, jedná se v podstatě pouze o registry s obecným významem.

V paketu mohou být přenášena tato čísla / hodnoty:

Typ hodnoty	Rozsah	Příklad
Bool / 1bit	0 / 1	0001 (bits v byte)
16bits celé bez znaménka	0 - 65535	12345 (0x 3930 H/L)
16bits celé se znaménkem	(zhruba -/+ 32767	…
32bits celé bez znaménka	0 – 4 294 967 265	0x11223344 (0x 44332211 H/L)
32bits celé se znaménkem	(zhruba) -/+ 2 147 483 632	…
32bits desetinné IEEE	…	…
Dva ASCII znaky	2x 0-255	PN (0x4E50 H/L)
Čtyři ASCII znaky	4x 0-255	Petr (0x72746550 H/L)

Příklady paketů pro komunikaci:

0x01 – Čtení jednoho / více digitálních výstupů DO (read OUT registr)
0x02 – Čtení jednoho / více digitálních vstupů DI (read IN registr)
Master –> Slave

Adresa	0x01/0x02	RegH	RegL	CountH	CountL	CrcH	CrcL

Slave –> Master

Adresa	0x01/0x02	Length	DATA	CrcH	CrcL

Data – bytes obsahující čtené / přenášené (aktuálně nastavené) digitální hodnoty
První byte obsahuje D7,D6,D5,D4,D3,D2,D1,D0
Druhý byte obsahuje D15,D14,D13,D12,D11,D10,D9,D8
Dále podle počtu přenášených bitů (nepoužité bity v posledním byte jsou „0“)

0x03 – Čtení jednoho / více analogového výstupů AO (read OUT reg)
0x04 – Čtení jednoho / více analogového vstupů AI (read IN reg)
Master –> Slave

Adresa	0x03/0x04	RegH	RegL	CountH	CountL	CrcH	CrcL

Slave –> Master

Adresa	0x03/0x04	Length	DATA	CrcH	CrcL

Data – bytes obsahující čtené / přenášené (aktuálně nastavené) analogové hodnoty
Dvojice bytes (16bits/2B), podle počtu přenášených hodnot

Analog1H	Analog1L	Analog2H	Analog2L	…	…

0x05 – Zápis jednoho digitálního výstupu DO (write OUT reg)
Master –> Slave

Adresa	0x05	RegH	RegL	ValueH	ValueL	CrcH	CrcL

Value – FF(H)00(L) zapnuto / 00(H)00(L) vypnuto
Slave –> Master

Adresa	0x05	RegH	RegL	ValueH	ValueL	CrcH	CrcL

(v odpovědi je v podstatě celý přijatý paket jako jeho potvrzení)

0x06 –> Zápis jednoho analogového výstupu AO (write OUT reg)

Master – Slave

Adresa	0x06	RegH	RegL	ValueH	ValueL	CrcH	CrcL

Value – zapisovaná 16bits/2B hodnota H/L

Slave –> Master

Adresa	0x06	RegH	RegL	ValueH	ValueL	CrcH	CrcL

(v odpovědi je v podstatě celý přijatý paket jako jeho potvrzení)

0x0F – Zápis několika digitálních výstupů DO (write OUT reg)

Master –> Slave

Adresa	0x0F	RegH	RegL	CountH	CountL	Length	DATA	CrcH	CrcL

Data – bytes obsahující zapisované / přenášené (nastavované) digitální výstupy

První byte obsahuje D7,D6,D5,D4,D3,D2,D1,D0

Druhý byte obsahuje D15,D14,D13,D12,D11,D10,D9,D8

Dále podle počtu přenášených bitů (nepoužité bity v posledním byte jsou „0“)

Slave –> Master

Adresa	0x0F	RegH	RegL	CountH	CountL	CrcH	CrcL

(vrací umístění a počet zapsaných hodnot)

0x10 – Zápis několika analogových výstupů AO (read OUT reg)

Master –> Slave

Adresa	0x10	RegH	RegL	CountH	CountL	Length	DATA	CrcH	CrcL

Data – bytes obsahující zapisované / přenášené analogové hodnoty

Dvojice bytes (16bits/2B), podle počtu přenášených hodnot

Analog1H	Analog1L	Analog2H	Analog2L	…	…

Slave –> Master

Adresa	0x10	RegH	RegL	CountH	CountL	CrcH	CrcL

(vrací umístění a počet zapsaných hodnot)

Odpovědi:

- Pokud je požadavek zpracován úspěšně tak je v odpovědi vrácen stejný povel / kód jako byl přijat.
- Pokud nebyl požadavek zpracován, tak je u vráceného povelu / kódu nastaven bit7=1 jako indikace vzniklé chyby (například 0x01, při chybě 0x81).
- V případě chyby je paket následující:

Adresa	Povel+0x80	Kód Chyby	CrcH	CrcL

- Jsou definovány následující kódy pro chyby

Kód	Popis
01	Přijatý kód funkce je neplatný / nelze jej vykonat
02	Adresa registru není dostupná / mimo rozsah
03	Datová hodnota je neplatná
04	Vyskytla se neznámá chyba při vykonávání
05	Vykonávání trvá dlouho (zamezuje vypršení časového limitu v master)
06	Zařízení je zaneprázdněno, požadavek je potřeba zaslat znova později
07	Zpracování selhalo (pro povely 13 a 14)
08	Chyba v zařízení (signalizace potřeby opravy)

- Pokud je paket přijat nesprávně, například špatný kontrolní součet, tak se na něho nijak nereaguje (zahodí se).

3 Kontrolní součet

Každý přenášený paket je ukončen kontrolním součtem zvaným CRC. Jde o 16bits / 2B hodnotu, první je přenášen HIGH a druhý LOW byte. Zde je (možná ne zcela ideální) ukázka jak stanovit CRC pro vložení do paketu.

```
// Compute the MODBUS RTU CRC
UInt16 ModRTU_CRC(byte[] buf, int len)
{
 UInt16 crc = 0xFFFF;

 for (int pos = 0; pos < len; pos++)
 {
  crc ^= (UInt16)buf[pos];        // XOR byte into least sig. byte of crc

  for (int i = 8; i != 0; i--)    // Loop over each bit
  {
   if ((crc & 0x0001) != 0) {     // If the LSB is set
    crc >>= 1;                   // Shift right and XOR 0xA001
    crc ^= 0xA001;
   }
   else                          // Else LSB is not set
    crc >>= 1;                   // Just shift right
  }
 }
 // Note, this number has low and high bytes swapped, so use it accordingly (or swap bytes)
 return crc;
}
```

Pro ověření činnosti jsou uvedeny některé pakety a jejich CRC:
…

4 Implementace

Na první pohled se zdají přenášené pakety a tedy celý protokol velmi jednoduchý. Vše je pravda až na jednu velmi podstatnou věc a tou je detekce konce paketu. Paket neobsahuje žádné úvodní ani koncové značky, podle kterých by jej bylo možno snadno detekovat. Lze pouze detekovat konec paketu pomocí pauzy v přenosu o délce 1.5 až 2 vysílané bytes. Tato skutečná časová délka samozřejmě závisí na aktuálně nastavené přenosové rychlosti.
Následující tabulka shrnuje nějaké parametry pro přenos. Jednotlivé sloupce obsahují tyto hodnoty:

- Nastavená sériové přenosová rychlost.
- Počet přenášených bytes (B) za vteřinu.
- Čas trvání přenosu jednoho byte (B).

- Čas detekce klidu jako značky konce paketu.

První řádek (v každém z řádků tabulky) uvádí velmi letmo stanového hodnoty (avšak v podstatě zcela postačující) a druhý řádek uvádí přesné hodnoty stanovené výpočtem / měřením (bude doplněn).

Baud Rate	Přenos	Jeden byte	Čas klidu (1.5-2B)
9600	1 kB/s	1 ms	1.5 – 2 ms
19200	2 kB/s	0.52 ms	0.78 – 1.04 ms
38400	3.8 kB/s	0.26 ms	0.39 – 0.52 ms
57600	5.7 kB/s	0.17 ms	0.26 – 0.34 ms
115200	11 kB/s	0.087 ms	0.13 – 0.15 ms

Poznámky:

- Přenos jednoho byte (B) se skládá zhruba z 10 (až 11) bitů (start bit, 8 data bits, 1-2 stop bits)

5 Poznámky

…