MODBUS 协议用户手册

V1.4

目录

1 数据传输模式

MODBUS 采用 RTU 和 ASCII 两种方式进行数据传输。RTU 模式下，采用 8bit 二进制字符，ASCII 模式下采用 7bit ASC 字符。将 RTU 模式下的一个字节的高 4 位和低 4 位分开，变成 2 个字节，这样就是 ASCII 模式下传输的字节。比如 RTU 模式下的数据 0x1A，那么 ASCII 模式下就是 0x31 0x41 两个字节，所以 ASCII 模式下的帧长度为 RTU 模式下的 1 倍。

注：根据两种模式的传输特点，ASCII 模式抗干扰能力较强，故推荐使用 ASCII 模式；RTU 模式下波特率推荐使用大于 9600bps 的；另，ASCII 模式在无校验是数据位数必须是 8。

RTU 传输模式的数据帧采用 CRC 校验，ASCII 模式采用 LRC 校验。

下表总结了两种传输模式的区别：

传输模式	ASCII (7 bit)	RTU (8 bit)
编码格式	ASCII 码 (‘0’-‘9’ ‘A’-‘F’)	8bit 二进制 (0x00 – 0xff)
起始位	1	1
数据位	7，8	8
校验位	无、奇、偶	无、奇、偶
停止位	1、2	1、2
帧校验	LRC	CRC16

2 寄存器和数据类型

下表列举了几种寄存器和数据类型

寄存器类型	数据长度	寄存器数量	描述
COIL	1 bit	-	布尔变量(ON OFF)
FLOAT	32 bit	2	32 位浮点数(IEEE754 格式)
INT	16 bit	1	无符号整型(0x0 - 0xFFFF)
DOUBLE	64 bit	4	64 位浮点数(IEEE754 格式)

2.1 COIL

布尔变量 0xFF00 -> ON 0x0000 -> OFF

2.2 FLOAT

使用 2 个寄存器存储单精度 IEEE754 格式的浮点数。

每个浮点数包含 4 个字节，具体定义如下：

SEEEEEEE EMMMMMMM MMMMMMMM MMMMMMMM

S:符号位 0->正 1->负 (1 位)

E:阶码 (8 位)

M:尾数的小数部分 (23 位)

例如：0xC1480000 = -12.5

2.3 DOUBLE

使用 4 个寄存器存储单精度 IEEE754 格式的浮点数。

每个浮点数包含 8 个字节，具体定义如下：

S:符号位 0->正 1->负 (1 位)

E:阶码 (11 位)

M:尾数的小数部分 (52 位)

2.4 INT

使用 1 个寄存器存储无符号整型数。

例如：0x0025 = 37 0x1234 = 4660

3 数据帧格式定义

3.1 CMD=0x03(读 1 个或多个寄存器)

本例数据为读取小信号切除（地址：0x0030）的数据帧，仪表地址=1。

请求帧：上位机->仪表

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	03	30 33
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	30	33 30
寄存器数量高字节	00	30 30
寄存器数量低字节	02	30 32
帧校验	C4 04	43 41
包尾	NONE	0D 0A

应答帧：仪表->上位机

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	03	30 33
数据长度	04	30 34
寄存器 0x0030 的数据高字节	00	30 30
寄存器 0x0030 的数据低字节	00	30 30
寄存器 0x0031 的数据高字节	3F	33 46
寄存器 0x0031 的数据低字节	00	30 30
帧校验	EB C3	42 39
包尾	NONE	0D 0A

本应答帧返回的小信号切除数据为 0.5。

3.2 CMD=0x05(写 COIL 变量)

本例数据为清除总量的数据帧，仪表地址=1。

请求帧：上位机->仪表

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	05	30 35
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	90	39 30
COIL 变量高字节	FF	46 46
COIL 变量低字节	00	30 30
帧校验	8C 17	36 42
包尾	NONE	0D 0A

应答帧：仪表->上位机

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	05	30 35
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	90	39 30
COIL 变量高字节	FF	46 46
COIL 变量低字节	00	30 30
帧校验	8C 17	36 42
包尾	NONE	0D 0A

3.3 CMD=0x06(写单个寄存器)

本例数据为写流量单位=m3/h 的数据帧，仪表地址=1。

请求帧：上位机->仪表

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	06	30 36
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	00	30 30
变量高字节	00	30 30
变量低字节	00	30 30
帧校验	89 CA	46 39
包尾	NONE	0D 0A

应答帧：仪表->上位机

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	06	30 36
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	00	30 30
变量高字节	00	30 30
变量低字节	00	30 30
帧校验	89 CA	46 39
包尾	NONE	0D 0A

3.4 CMD=0x10(写多个寄存器)

本例数据为写阻尼时间=0.1s 的数据帧，仪表地址=1。

请求帧：上位机->仪表

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	10	31 30
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	20	32 30
寄存器数量高字节	00	30 30
寄存器数量低字节	02	30 32
数据长度	04	30 34
写入寄存器 0x0020 的高字节	CC	43 43
写入寄存器 0x0020 的低字节	CD	43 44
写入寄存器 0x0021 的高字节	3D	34 44
写入寄存器 0x0021 的低字节	CC	43 43
帧校验	4F DD	32 37
包尾	NONE	0D 0A

应答帧：仪表->上位机

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	10	31 30
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	20	32 30
寄存器数量高字节	00	30 30
寄存器数量低字节	02	30 32
帧校验	40 02	43 44
包尾	NONE	0D 0A

3.5 故障返回帧

对于不能正确完成操作的请求帧，仪表将返回一个报告故障代码的返回帧，以报告不能完成操作的具体原因。

比如将瞬时流量单位设置成 m3，由于仪表不能支持 m3 这个流量单位，所以将返回如下的故障应答帧。

故障应答帧：仪表->上位机

数据场名称	**RTU 示例数据(HEX)**	**ASC 示例数据(HEX)**
包头	NONE	3A
仪表地址	01	30 31
功能码	86	38 36
故障码	43	34 33
帧校验	03 91	39 31
包尾	NONE	0D 0A

注：1．故障返回帧中的功能码=请求帧的功能码+0x80

2．具体的故障代码请参考 附录 1：故障码

4 数据帧校验算法

4.1 LRC 校验

// LRC 校验范围：从“仪表地址”到 LRC 帧校验码的前一个字节

```
void LRC(unsigned char *buf, unsigned int len)
{
    unsigned int i;
    LRC = 0;
    for (i=0; i<len; i++)
    {
        LRC += buf[i];
    }
    LRC = 0xff - LRC;
    LRC++;
}
```

4.2 CRC16 校验

```
const unsigned char  TAB_CRC_H[] = {
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40
};

const unsigned char  TAB_CRC_L[] = {
    0x00,0xC0,0xC1,0x01,0xC3,0x03,0x02,0xC2,0xC6,0x06,0x07,0xC7,0x05,0xC5,0xC4,0x04,
    0xCC,0x0C,0x0D,0xCD,0x0F,0xCF,0xCE,0x0E,0x0A,0xCA,0xCB,0x0B,0xC9,0x09,0x08,0xC8,
    0xD8,0x18,0x19,0xD9,0x1B,0xDB,0xDA,0x1A,0x1E,0xDE,0xDF,0x1F,0xDD,0x1D,0x1C,0xDC,
    0x14,0xD4,0xD5,0x15,0xD7,0x17,0x16,0xD6,0xD2,0x12,0x13,0xD3,0x11,0xD1,0xD0,0x10,
    0xF0,0x30,0x31,0xF1,0x33,0xF3,0xF2,0x32,0x36,0xF6,0xF7,0x37,0xF5,0x35,0x34,0xF4,
    0x3C,0xFC,0xFD,0x3D,0xFF,0x3F,0x3E,0xFE,0xFA,0x3A,0x3B,0xFB,0x39,0xF9,0xF8,0x38,
    0x28,0xE8,0xE9,0x29,0xEB,0x2B,0x2A,0xEA,0xEE,0x2E,0x2F,0xEF,0x2D,0xED,0xEC,0x2C,
    0xE4,0x24,0x25,0xE5,0x27,0xE7,0xE6,0x26,0x22,0xE2,0xE3,0x23,0xE1,0x21,0x20,0xE0,
```

```
    0xA0,0x60,0x61,0xA1,0x63,0xA3,0xA2,0x62,0x66,0xA6,0xA7,0x67,0xA5,0x65,0x64,0xA4,
    0x6C,0xAC,0xAD,0x6D,0xAF,0x6F,0x6E,0xAE,0xAA,0x6A,0x6B,0xAB,0x69,0xA9,0xA8,0x68,
    0x78,0xB8,0xB9,0x79,0xBB,0x7B,0x7A,0xBA,0xBE,0x7E,0x7F,0xBF,0x7D,0xBD,0xBC,0x7C,
    0xB4,0x74,0x75,0xB5,0x77,0xB7,0xB6,0x76,0x72,0xB2,0xB3,0x73,0xB1,0x71,0x70,0xB0,
    0x50,0x90,0x91,0x51,0x93,0x53,0x52,0x92,0x96,0x56,0x57,0x97,0x55,0x95,0x94,0x54,
    0x9C,0x5C,0x5D,0x9D,0x5F,0x9F,0x9E,0x5E,0x5A,0x9A,0x9B,0x5B,0x99,0x59,0x58,0x98,
    0x88,0x48,0x49,0x89,0x4B,0x8B,0x8A,0x4A,0x4E,0x8E,0x8F,0x4F,0x8D,0x4D,0x4C,0x8C,
    0x44,0x84,0x85,0x45,0x87,0x47,0x46,0x86,0x82,0x42,0x43,0x83,0x41,0x81,0x80,0x40
};
```

// CRC 校验范围：从“仪表地址”到 CRC 帧校验码的前一个字节

```
void CRC(unsigned char *buf, unsigned int len)
{
    unsigned int i;
    unsigned char CRC_H , CRC_L , index, ch;
    CRC_H = 0xff;
    CRC_L = 0xff;
    for (i=0; i<len; i++)
    {
        ch = buf[i];
        index = CRC_H ^ ch;
        CRC_H = CRC_L ^ TAB_CRC_H[index];
        CRC_L = TAB_CRC_L[index];
    }
}
```

5 仪表变量地址定义

以下为仪表支持数据变量信息列表，数据均为 HEX 类型，括号内为十进制的组态王和 PLC 地址。数据向上位机发送方式均为地地址内容先发送。例如：

浮点数 10.123 对应的单精度 float 的十六进制为

0x4121F7CF

仪表向上位机发送时为：F7 CF 41 21

双精度 double 的十六进制为

0x40243EF9DB22D0E5

仪表向发送时为：D0 E5 DB 22 3E F9 40 24

对于某些 PLC（比如 S7-200)来说，可能由于存储方式不同（特别是浮点数），可以通过移位和逻辑或来完成数据的解析。

变量名	寄存器地址	寄存器长度	读指令	写指令
COIL 类型				
清总量	0091	***	***	05
INT 类型				
瞬时流量单位（参见附录 2）	0000	0001	03	06

瞬时流量小数点位数（参见附录3）	0008	0001	03	06
总量单位（参见附录2）	0010	0001	03	06
总量小数点位数（参见附录3）	0018	0001	03	06
脉冲电平	0050	0001	03	06
设备地址	0078	0001	03	06
通讯协议（参见附录3）	0058	0001	03	***
波特率（参见附录3）	0060	0001	03	***
数据位（参见附录3）	0068	0001	03	***
校验方式（参见附录3）	0070	0001	03	***
LONG 类型				
总量扩展	0500	0002	03	***
FLOAT 类型				
阻尼时间	0020	0002	03	10
小信号切除	0030	0002	03	10
频率上限	0038	0002	03	10
脉冲当量	0040	0002	03	10
脉冲宽度	0048	0002	03	10
低报警点（百分比）	00C0	0002	03	10
高报警点（百分比）	00D0	0002	03	10
流体密度	0098	0002	03	10
瞬时流量	0708	0002	03	***
总量	0504	0002	03	***
DOUBLE 类型				
量程（对应20MA）	0028	0004	03	10
传感器系数	0088	0004	03	10

总量	0090	0004	03	***
瞬时流量	0700	0004	03	***

6 附录 1：故障码

0x01	无效指令码
0x02	无效的寄存器地址
0x30	参数超上限
0x31	参数超下限
0x32	参数选择项错误
0x40	无效的寄存器长度
0x41	寄存器不支持当前的指令码
0x42	寄存器未指定功能
0x43	瞬时流量单位不存在
0x44	总量单位不存在
0x45	最高频率输出超上限
0x46	最低频率输出超下限
0x47	最高流速超上限
0x48	占空比超上限
0xFE	数据帧混乱
0xFF	数据帧校验错误

7 附录 2：常用单位定义

瞬时流量	m3/h	0

	m3/m	1
	m3/s	2
	L/h	3
	L/m	4
	L/s	5
	USG/h	6
	USG/m	7
	USG/s	8
	kg/h	9
	kg/m	10
	kg/s	11
	t/h	12
	t/m	13
	t/s	14
累积量	L	0
	m3	1
	USG	2
	kg	3
	t	4

8 附录 3：符号代号定义

modbus 通讯模式	RTU	0
	ASCII	1

校验方式	奇校验	0
	偶校验	1
	无校验	2
波特率	1200bps	0
	2400bps	1
	4800bps	2
	9600bps	3
	19200bps	4
	38400bps	5
数据位数	7位	0
	8位	1
小数点位数	小数点后0位	0
	小数点后1位	1
	小数点后2位	2
	小数点后3位	3

9 附录 4：口径代号定义

口径	代号	口径	代号
DN1	0	DN600	27

DN1.5	1	DN700	28
DN2	2	DN750	29
DN3	3	DN800	30
DN4	4	DN900	31
DN5	5	DN1000	32
DN6	6	DN1100	33
DN8	7	DN1200	34
DN10	8	DN1300	35
DN15	9	DN1350	36
DN20	10	DN1400	37
DN25	11	DN1500	38
DN32	12	DN1600	39
DN40	13	DN1700	40
DN50	14	DN1800	41
DN65	15	DN2000	42
DN80	16	DN2100	43
DN100	17	DN2200	44
DN125	18	DN2300	45
DN150	19	DN2400	46
DN200	20	DN2500	47
DN250	21	DN2600	48
DN300	22	DN2700	49
DN350	23	DN2800	50
DN400	24	DN2900	51
DN450	25	DN3000	52
DN500	26		