众山DTU Modbus协议手册

众联万物 智慧未来

我们用心创造

目录

前　　言

感谢您使用成都众山科技有限公司提供的DTU产品。

本手册主要介绍众山 DTU Modbus 协议。

适用型号：ZSDR3411、ZSNR311、ZSR2184、ZSLR311。

版权声明

版本信息

文档名称：众山DTU Modbus协议手册

版本：1.01

修改日期： 2018 年 07 月 09 日

相关文档

1、《ZSDxxxx DTU Easy 控件接口说明》

2、《众山 DTU 脚本编程手册》

3、《网络模式选择》

4、《串口及远程控制协议》

一、Modbus协议简介

众山 DTU 在具有模拟量采集、数字量采集和开关量控制的型号中，均使用 Modbus 协议进行这些信号的采集与控制，本文描述的内容适用于众山所有带有这些功能的产品。使用此协议不仅可以从串口端控制或采集 DTU 的 DI/DO/AI，还可以从中心服务器端远程控制或采集 DTU 的 DI/DO/AI，甚至用户还可以把此协议中的数据包设置在 DTU 的脚本参数的@C 命令中，让 DTU 使用此协议根据脚本定义的规则自动进行控制或采集。DTU 脚本参数的@C 命令相当于代替用户的中心下发指令且能周期执行。

Modbus 协议设备都具有唯一的 Modbus 地址，众山 DTU 默认 Modbus 地址为 100，用户可以修改，只有当协议中的地址与 DTU 相同或为广播地址 00，且协议符合 Modbus 的 CRC 校验规则的协议包 DTU 才会进行处理，否则 DTU 作为透传数据。所以当用户在 DTU 的串口端接有同样是 Modbus 协议的设备时，其地址必须与 DTU 地址不一致，DTU 才能透传用户设备的 Modbus 上下行数据。

二、寄存器地址

不同的型号支持的 DI/DO/AI 数可能不一样，具体信息请参考相应型号的数据手册或致电众山科技进行咨询，当对一个该型号设备不支持的寄存器操作时，Modbus 协议将返回错误。

寄存器地址	名称	字节数	说明	备注
模拟量输入				
0000	AI1_H	2	模拟量通道 1 高	每个模拟量通道占 2 个 Modbus 寄存器，4 个字节，格式为浮点数，浮点数格式符合 IEEE 754 标准
0001	AI1_L	2	模拟量通道 1 低	
0002	AI2_H	2	模拟量通道 2 高	
0003	AI2_L	2	模拟量通道 2 低	
0004	AI3_H	2	模拟量通道 3 高	
0005	AI3_L	2	模拟量通道 3 低	
0006	AI4_H	2	模拟量通道 4 高	
0007	AI4_L	2	模拟量通道 4 低	
0008	AI5_H	2	模拟量通道 5 高	
0009	AI5_L	2	模拟量通道 5 低	
000A	AI6_H	2	模拟量通道 6 高	
000B	AI6_L	2	模拟量通道 6 低	
000C	AI7_H	2	模拟量通道 7 高	
000D	AI7_L	2	模拟量通道 7 低	
000E	AI8_H	2	模拟量通道 8 高	
000F	AI8_L	2	模拟量通道 8 低	
数字量输入				

0010	DI1	2	数字量输入 1	0000 表示低 0001 表示高
0011	DI2	2	数字量输入 2	
0012	DI3	2	数字量输入 3	
0013	DI4	2	数字量输入 4	
开关量输出				
0014	DO1	2	开关量输出 1	0000 表示断开 0001 表示闭合
0015	DO2	2	开关量输出 2	
0016	DO3	2	开关量输出 3	
0017	DO4	2	开关量输出 4	

三、协议详解

地址域	功能码	数据	差错检验

Modbus 使用“big-Endian”表示地址和数据项，这就意味着当发射多个字节时，首先发送最高字节。

例如：寄存器地址为 0x0014，首先发送的是 0x00，然后才是 0x14。

一个正常的 Modbus 响应：响应功能码=请求功能码。

一个 Modbus 的异常响应：响应功能码=请求功能码+0x80，提供一个异常码来指示差错原因。

3.1 功能码描述

3.1.1 01 读线圈

可以使用此功能码读取继电器 DO1~DO4 的状态。

请求 PDU 详细说明了起始地址，即指定第一个线圈的地址和线圈数量，从零开始寻址线圈，因此寻址线圈 1-4 为 0-3。

响应 PDU 中 N 个字节的线圈状态的每一个 bit 位代表一个线圈的状态，状态 1=ON, 0=OFF。第一个字节的最低位 LSB 代表第 0 号线圈的状态（即起始地址指定的线圈号为 0 号线圈），其他线圈依次类推，一直到这个字节的最高位 MSB 为止，并且后续字节中都是由低到高代表连续的各线圈状态。

如果线圈数量不是 8 的倍数，将用零填充剩余最后数据字节中的剩余比特，字节数量域说明了数据的完整字节数。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x01
起始地址	2 个字节	0x0014 至 0x0017
线圈数量	2 个字节	n(1 至 4)
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x01
字节数	1 个字节	N
线圈状态	N 个字节	N=n/8, 或 N=n/8+1
CRC 校验	2 个字节	

注: 线圈状态的字节数 N=线圈数量 n/8, 如果余数不等于 0，则 N=n/8+1

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x81 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读离散量 DO1-DO4 的实例

请求		响应	
地址	64	地址	64
功能码	01	功能码	01
起始地址高 H	00	字节数	01
起始地址低 L	14	DO1-DO4 状态	0D
线圈数量高 H	00	CRC 校验高 H	8E
线圈数量低 L	04	CRC 校验低 L	81
CRC 校验高 H	74		
CRC 校验低 L	38		

发送：6401001400047438 DTU 响应：6401010D8E81

DO1-DO4 的状态字节为 0D，二进制 00001101，DO1 是这个字节的 LSB(第 0 位)为 1 表示闭合，DO2 是第 1 位为 0 表示断开，DO3 是第 2 位为 1 表示闭合，DO4 是第 3 位为 1 表示闭合，用 0 填充剩余的 4 位。

3.1.2 02 读离散量输入

可以使用此功能码读取数字量输入 DI1~DI4 的状态。

请求 PDU 详细说明了起始地址，即指定第一个输入量的地址和数量，从零开始寻址，因此寻址输入量 1-4 为 0-3。

响应 PDU 中 N 个字节的输入量状态的每一个 bit 位代表一个输入量的状态，状态 1=HIGH, 0=LOW。第一个字节的最低位 LSB 代表第 0 号输入量的状态（即起始地址指定的输入量为 0 号输入

量），其他输入量依次类推，一直到这个字节的最高位 MSB 为止，并且后续字节中都是由低到高代表连续的各输入量状态。

如果输入量数量不是 8 的倍数，将用零填充剩余最后数据字节中的剩余比特，字节数量域说明了数据的完整字节数。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x02
起始地址	2 个字节	0x0010 至 0x0013
输入数量	2 个字节	n(1 至 4)
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x02
字节数	1 个字节	N
输入状态	N 个字节	N=n/8， 或 N=n/8+1
CRC 校验	2 个字节	

注: 输入状态的字节数 N=输入数量 n/8, 如果余数不等于 0，则 N=n/8+1

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x82 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读离散量输入 DI1-DI4 的实例

请求		响应	
地址	64	地址	64
功能码	02	功能码	02
起始地址高 H	00	字节数	01
起始地址低 L	10	DI1-DI4 状态	0C
输入数量高 H	00	CRC 校验高 H	BF
输入数量低 L	04	CRC 校验低 L	41
CRC 校验高 H	71		
CRC 校验低 L	F9		

发送：64020010000471F9　　DTU 响应：6402010CBF41

DI1-DI4 的状态字节为 0C，二进制 00001100，DI1 是这个字节的 LSB(第 0 位)为 0 表示低，DI2 是第 1 位为 0 表示低，DI3 是第 2 位为 1 表示高，DI4 是第 3 位为 1 表示高，用 0 填充剩余的 4 位。

3.1.3 03 读保持寄存器

3.1.4 04 读输入寄存器

使用该功能码可以读取所有寄存器包括 AI1-AI8、DI1-DI4、DO1-DO4 的状态。

请求 PDU 详细说明了起始寄存器地址和寄存器数量，从零开始寻址寄存器，因此寻址寄存器 1-16 为 0-15。

响应报文中的寄存器数据每个寄存器有 2 个字节，对于每一个寄存器，第一个字节代表寄存器值的高位，第二个字节代表寄存器值的低位。字节数为寄存器数量乘以 2。对于 AI1-AI7，一个通道占用 2 个寄存器，4 个字节的值使用浮点数表示，对于 DI1-DI4，2 个字节的值 0000 代表输入量为低，0001 代表输入量为高，对于 DO1-DO4，2 个字节的值 0000 代表继电器断开，0001 代表继电器闭合。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x03 或 04
起始地址	2 个字节	0x0000 至 0x0017
寄存器数量	2 个字节	n(1 至 24)
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x03 或 0x04
字节数	1 个字节	N=2*n
寄存器值	N 个字节	N=2*n, n 为寄存器数量
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x83 或 0x84 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读模拟量输入 AI1-AI8、离散量输入 DI1-DI4、继电器状态 DO1-DO4 的实例

请求		响应	
地址	64	地址	64
功能码	03	功能码	03
起始地址高 H	00	字节数	30
起始地址低 L	00	AI1-AI8 值	32 个字节
寄存器数量高 H	00	DI1-DI4 状态	8 个字节
寄存器数量低 L	18	DO1-DO4 状态	8 个字节
CRC 校验高 H	4C	CRC 校验高 H	
CRC 校验低 L	35	CRC 校验低 L	

发送：6403000000184C35

3.1.5 05 写单个线圈

可以使用该功能码写单个继电器 DO1-DO4 为断开或闭合

请求数据域中的常量说明请求的 ON/OFF 状态，十六进制值 0xFF00 请求输出为 ON(闭合)，十六进制值 0x0000 请求输出为 OFF(断开)，其他所有值都是非法的，对输出不起作用，DTU 返回错误响应。

请求域中的输出地址规定了要写入线圈的地址。

正常响应是请求的应答，在写入线圈状态后返回这个正常响应。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x05
输出地址	2 个字节	0x0014 至 0x0017
输出值	2 个字节	0x0000 或 0xFF00
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x05
输出地址	2 个字节	0x0014 至 0x0017
输出值	2 个字节	0x0000 或 0xFF00
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x85 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04

CRC 校验	2 个字节	

这是一个请求写线圈 DO2 为 ON(闭合)的实例

请求		响应	
地址	64	地址	64
功能码	05	功能码	05
输出地址高 H	00	输出地址高 H	00
输出地址低 L	15	输出地址低 L	15
输出值高 H	FF	输出值高 H	FF
输出值低 L	00	输出值低 L	00
CRC 校验高 H	94	CRC 校验高 H	94
CRC 校验低 L	0B	CRC 校验低 L	0B

发送：64050015FF00940B　　DTU 响应：64050015FF00940B

3.1.6　06 写单个寄存器

可以使用该功能码写单个继电器 DO1-DO4 为断开或闭合。

请求数据域中的寄存器值说明请求的 ON/OFF 状态，十六进制值 0001 请求输出为 ON(闭合)，十六进制值 0x0000 请求输出为 OFF(断开)。

请求域中的寄存器地址规定了要写入线圈的地址。

正常响应是请求的应答，在写入线圈状态后返回这个正常响应。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0014 至 0x0017
寄存器值	2 个字节	0x0000 至 0xFFFF
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0014 至 0x0017
寄存器值	2 个字节	0x0000 至 0xFFFF
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x86 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求写线圈 DO2 为 ON(闭合)的实例

请求		响应	
地址	64	地址	64
功能码	06	功能码	06
寄存器地址高 H	00	寄存器地址高 H	00
寄存器地址低 L	15	寄存器地址低 L	15
寄存器值高 H	00	寄存器值高 H	00
寄存器值低 L	01	寄存器值低 L	01
CRC 校验高 H	50	CRC 校验高 H	50
CRC 校验低 L	3B	CRC 校验低 L	3B

发送：640600150001503B　　DTU 响应：640600150001503B

3.1.7　0F 写多个线圈

可以使用此功能码写多个继电器 DO1~DO4 为断开或闭合。

请求 PDU 详细说明了起始地址，即指定第一个线圈的地址和线圈数量，从零开始寻址线圈，因此寻址线圈 1-4 为 0-3。

请求数据域中的内容说明了被请求的 ON/OFF 状态，域比特位中的逻辑“1”请求相应输出为 ON，域比特位中的逻辑“0”请求相应输出为 OFF。从数据域中第一个字节的 bit0 开始到 bit7，然后到第二个字节的 bit0，依次表示第一个线圈到第 n 个线圈的 ON/OFF 值。

正常响应返回功能码、起始地址和线圈数量。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x0F
起始地址	2 个字节	0x0014 至 0x0017
线圈数量	2 个字节	n(1 至 4)
字节数	1 个字节	N=n/8,　或 N=n/8+1
输出值	N 个字节	
CRC 校验	2 个字节	

注: 线圈输出字节数 N=线圈数量 n/8, 如果余数不等于 0，则 N=n/8+1

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x0F
起始地址	2 个字节	0x0014 至 0x0017
线圈数量	2 个字节	n(1 至 4)
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x8F (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求从线圈 DO1 开始写入 4 个线圈的实例

请求		响应	
地址	64	地址	64
功能码	0F	功能码	0F
起始地址高 H	00	起始地址高 H	00
起始地址低 L	14	起始地址低 L	14
线圈数量高 H	00	线圈数量高 H	00
线圈数量低 L	04	线圈数量低 L	04
字节数	01	CRC 校验高 H	1D
输出值	0A	CRC 校验低 L	F9
CRC 校验高 H	48		
CRC 校验低 L	85		

发送：640F00140004010A4885　　DTU 响应：640F001400041DF9

DO1-DO4 的输出值为 0A，二进制 00001010，DO1 是这个字节的 LSB(第 0 位)为 0 表示断开，DO2 是第 1 位为 1 表示闭合，DO3 是第 2 位为 0 表示断开，DO4 是第 3 位为 1 表示闭合，用 0 填充剩余未使用的 4 位。

3.1.8　10 写多个寄存器

使用该功能码可以写连续寄存器 DO1-DO4 的状态。

请求 PDU 详细说明了起始寄存器地址、寄存器数量、字节数和寄存器值，从零开始寻址寄存器，因此寻址寄存器 1-16 为 0-15。

寄存器数据中每个寄存器有 2 个字节，对于每一个寄存器，第一个字节代表寄存器值的高位，第二

个字节代表寄存器值的低位。字节数为寄存器数量乘以 2，2 个字节的值 0000 代表继电器断开，0001 代表继电器闭合。

正常响应返回功能码、起始地址和被写入寄存器的数量。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x10
起始地址	2 个字节	0x0014 至 0x0017
寄存器数量	2 个字节	n(1 至 4)
字节数	1 个字节	N=2*n
寄存器值	N 个字节	N=2*n, n 为寄存器数量
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x10
起始地址	2 个字节	0x0014 至 0x0017
寄存器数量	2 个字节	n(1 至 4)
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x90 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个控制继电器 DO1-DO4 的实例

请求		响应	
地址	64	地址	64
功能码	10	功能码	10
起始地址高 H	00	起始地址高 H	00
起始地址低 L	14	起始地址低 L	14
寄存器数量高 H	00	寄存器数量高 H	00
寄存器数量低 L	04	寄存器数量低 L	04
字节数	08	CRC 校验高 H	88
DO1 寄存器值高 H	00	CRC 校验低 L	3B
DO1 寄存器值高 L	01		
DO2 寄存器值高 H	00		

DO2 寄存器值高 L	00		
DO3 寄存器值高 H	00		
DO3 寄存器值高 L	00		
DO4 寄存器值高 H	00		
DO4 寄存器值高 L	01		
CRC 校验高 H	F2		
CRC 校验低 L	61		

发送：64100014000408000100000000000 1 DTU 响应：641000140004883B

DO1 寄存器值为 0001 表示闭合，DO2 寄存器值为 0000 表示断开，DO3 寄存器值为 0000 表示断开，DO4 寄存器值为 0001 表示闭合。

3.2 错误码描述

错误码含义：当 DTU 收到错误的 Modbus 指令时，会返回功能码为请求功能码+0x80，紧随着一个字节的错误码代表出错原因。

错误码 01：表示不支持的功能码，众山 DTU 支持上述 8 种功能码，除此之外的功能码都会返回错误码为 01 的错误。

错误码 02：表示起始地址不存在或者起始地址加上寄存器数量后的地址不存在。总的来说表示访问的寄存器不存在。

错误码 03：表示寄存器数量不符合规范或者寄存器值非法。

错误码04：表示读写寄存器错误。

3.3 CRC 校验算法

CRC即循环冗余校验码（Cyclic Redundancy Check ）：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查（CRC）是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

附CRC校验算法代码：

```
uint16_t mb_crc(uint8_t *snd,uint16_t num)
{
    uint8_t CRC_Lb, CRC_Hb;
```

```
uint8_t CRC_L,CRC_H;
uint16_t crc;

uint16_t i, j;

CRC_H = 0xFF;
CRC_L = 0xFF;
for(i = 0; i < num; i++)
{
    CRC_L = CRC_L ^ snd[i];
    for(j = 0; j < 8; j++)
    {
        CRC_Lb = CRC_L;
        if((CRC_L&1)==1)
        {
            CRC_L = (CRC_L - 1) / 2;
            CRC_Lb = CRC_L;
            CRC_Hb = CRC_H;
            if((CRC_H&1)==1)
            {
                CRC_L = CRC_L + 128;
                CRC_Lb = CRC_L;
                CRC_H = (CRC_H - 1) / 2;
                CRC_Hb = CRC_H;
            }
            else
            {
                CRC_H = CRC_H / 2;
                CRC_Hb = CRC_H;
            }
            CRC_L = CRC_L ^ 1;
            CRC_Lb = CRC_L;
            CRC_H = CRC_H ^ 0xA0;
            CRC_Hb = CRC_H;
        }
        else
        {
            CRC_L = CRC_L / 2;
            CRC_Lb = CRC_L;
            CRC_Hb = CRC_H;
            if((CRC_H&1)==1)
            {
```

```
                    CRC_L = CRC_L + 128;
                    CRC_Lb = CRC_L;
                    CRC_H = (CRC_H - 1) / 2;
                    CRC_Hb = CRC_H;
                }
                else
                {
                    CRC_H = CRC_H / 2;
                    CRC_Hb = CRC_H;
                }
            }

        }
    }
    crc = CRC_L;
    crc <<= 8;
    crc |= CRC_H;
    return crc;
}
```