



节段位移计

Modbus RTU 协议说明书

版 本 V1.3
更新日期 2022-12-01

www.aiterich.com

目录

前言	1
一、 通讯协议	1
二、 通讯格式	1
2.1、 读取输入寄存器指令	1
2.1.1、请求单个寄存器	1
2.1.2、请求多个寄存器	2
2.2、 读取保持寄存器指令	3
2.2.1、读取单个寄存器	3
2.2.2、读取多个寄存器	3
2.3、 修改保持寄存器指令	4
2.3.1、修改单个寄存器	4
2.3.1、修改多个寄存器	5
三、 节段位移计参数配置和数据获取流程	6
3.1、 确定监测方向和 X-Mark 方向的夹角	6
3.2、 获取井口第一节 X 偏转角度 α	6
3.3、 设置每个节段的监测方向偏转角 β	6
3.4、 获取 X/Y/Z 三个方向位移	7
附录 1	8
附录 2	8
附录 3	9

前言

感谢您选用我公司产品，如果您有什么疑问或需要请联系我们。

在进行操作前，请仔细阅读本手册，如不遵照本手册操作造成的一切严重后果用户自担。

一、通讯协议

默认 2400bps, 8b N1 None Parity 通讯采用标准 Modbus RTU RS485 协议；数据采用 16 进制，校验采用 CRC-16 (Modbus)，低位在前，高位在后。

二、通讯格式

2.1、读取输入寄存器指令

功能码：04

数据起始地址

X 角度	Y 角度	Z 角度	X 方向偏转角	X 方向位移	Y 方向位移	Z 方向位移
0x06	0x08	0x0A	0x26	0x2A	0x2C	0x2E

角度单位：°

方向偏转角单位：°

位移单位：mm

2.1.1、请求单个寄存器

请求两个寄存器，返回四个字节。浮点型简析参考附录 2。

上位机发送：

地址码	功能码	起始地址 高位	起始地址 低位	寄存器数 量高位	寄存器数 量低位	CRC16 低位	CRC16 高位
Address	04	00	0E	00	02	CRC16L	CRC16H

下位机回复：

地址码	功能码	数据 个数	数据位	数据位	数据位	数据位	CRC16 低位	CRC16 高位
Address	04	04	数据 1	数据 2	数据 3	数据 4	CRC16L	CRC16H

例子：请求获取 1 号设备 X 角度

上位机发送：01 04 00 06 00 02 91 CA

下位机回复：01 04 04 41 F0 00 00 EF 8B

其中 41 F0 00 00 为 X 角度读取数据值，换算后为 30，单位为°，数值换算参考附录 2。

例子：请求获取 1 号设备 Y 角度

上位机发送：01 04 00 08 00 02 F0 09

下位机回复：01 04 04 42 34 00 00 AE 32

其中 42 34 00 00 为 Y 角度读取数据值，换算后为 45，单位为°，数值换算参考附录 2。

例子：请求获取 1 号设备 Z 角度

上位机发送：01 04 00 0A 00 02 51 C9

下位机回复：01 04 04 42 70 00 00 EE 27

其中 42 70 00 00 为 Z 角度读取数据值，换算后为 60，单位为°，数值换算参考附录 2。

2.1.2、请求多个寄存器

请求 X/Y/Z 角度，请求六个寄存器 返回十二个字节，前四个字节表示 X 角度，中间四个字节表示 Y 角度，后四个字节表示 Z 角度。浮点型简析参考附录 2。

上位机发送：

地址码	功能码	起始地址 高位	起始地址 低位	寄存器数 量高位	寄存器数 量低位	CRC16 低位	CRC16 高位
Address	04	00	06	00	06	CRC16L	CRC16H

下位机回复：

地址码	功 能 码	数 据 个 数	数 据 位	数 据 位	数 据 位	数 据 位	数 据 位	数 据 位	数 据 位	数 据 位	数 据 位	数 据 位	数 据 位	CRC16 低位	CRC16 高位
Address	04	0C	数 据 1	数 据 2	数 据 3	数 据 4	数 据 5	数 据 6	数 据 7	数 据 8	数 据 9	数 据 10	数 据 11	CRC16L	CRC16H

例子：请求获取 1 号设备 X 角度、Y 角度和 Z 角度

上位机发送：01 04 00 06 00 06 90 09

下位机回复：01 04 0C 41 F0 00 00 42 34 00 00 42 70 00 00 8F C1

其中：

41 F0 00 00 为 X 角度读取数据值，换算后为 30，单位为°。

42 34 00 00 为 Y 角度读取数据值，换算后为 45，单位为°。

42 70 00 00 为 Z 角度读取数据值，换算后为 60，单位为°。

2.2、读取保持寄存器指令

功能码：03

设备配置起始地址对应表：地址码 00，数据位 01，停止位 02，校验位 03，波特率 04，详情参考附录 1。

地址码	数据位	停止位	校验位	波特率
00	01	02	03	04

2.2.1、读取单个寄存器

请求一个寄存器，返回二个字节。

上位机发送：

地址码	功能码	起始地址 高位	起始地址 低位	寄存器数 量高位	寄存器数 量低位	CRC16 低位	CRC16 高位
Address	03	00	00	00	01	CRC16L	CRC16H

下位机回复：

地址码	功能码	数据 个数	数据位 高位	数据位 低位	CRC16 低位	CRC16 高位
Address	03	02	00	01	CRC16L	CRC16H

例子 1：请求获取 1 号设备的波特率

上位机发送：01 03 00 04 00 01 C5 CB

下位机回复：01 03 02 00 02 39 85 读取到的波特率为 2400（对应关系参考附录 1）

例子 2：请求获取 1 号设备的停止位

上位机发送：01 03 00 02 00 01 25 CA

下位机回复：01 03 01 02 71 89 读取到的停止位为 2（对应关系参考附录 1）

2.2.2、读取多个寄存器

上位机发送：读取设备数据位和停止位

地址码	功能码	起始地址 高位	起始地址 低位	寄存器数 量高位	寄存器数 量低位	CRC16 低位	CRC16 高位
-----	-----	------------	------------	-------------	-------------	-------------	-------------

Address	03	00	01	00	02	CRC16L	CRC16H
---------	----	----	----	----	----	--------	--------

下位机回复：

地址码	功能码	数据 个数	数据 1 高位	数据 1 低位	数据 2 高位	数据 2 低位	CRC16 低位	CRC16 高位
Address	03	04	00	09	00	01	CRC16L	CRC16H

例子 1：请求获取 1 号设备数据位和停止位

上位机发送：01 03 00 01 00 02 95 CB

下位机回复：01 03 04 00 09 00 01 EB F1

00 09 读取到的设备 1 数据位为 9

00 01 读取到的停止位为 1

例子 2：请求获取设备 1 停止位，校验位，波特率

上位机发送：01 03 00 02 00 03 A4 0B

下位机回复：01 03 06 00 01 00 02 00 09 7D 73

00 01 读取到的停止位为 1

00 02 读取到的校验位为 Even

00 09 读取到的波特率为 115200

2.3、修改保持寄存器指令

注意：修改停止位、数据位、校验位可能导致数据接收不正常，一般只修改波特率和地址码。

2.3.1、修改单个寄存器

功能码：06

上位机发送：

地址码	功能码	起始地址 高位	起始地址 低位	数据 高位	数据 低位	CRC16 低位	CRC16 高位
Address	06	00	00	00	03	CRC16L	CRC16H

下位机回复：

地址码	功能码	起始地址 高位	起始地址 低位	数据 高位	数据 低位	CRC16 低位	CRC16 高位
Address	06	00	00	00	03	CRC16L	CRC16H

如果发送的是合法指令，发送指令和回复指令相同；如果发送的是非法指令，回复错误码 Address ERROR(从机不用关心) 06。地址码(Address)为 00，则表示所有从机接收对应指令，从机不回复。

地址码起始地址 00

例子：1 号设备的地址码修改为 03

上位机发送：01 06 00 00 00 03 C9 CB

下位机回复：01 06 00 00 00 03 C9 CB

数据位起始地址 01

例子：1 号设备的数据位修改为 9

上位机发送：01 06 00 01 00 09 18 0C

下位机回复：01 06 00 01 00 09 18 0C

停止位起始地址 02

例子：1 号设备的停止位修改为 2

上位机发送：01 06 00 02 00 02 A9 CB

下位机回复：01 06 00 02 00 02 A9 CB

校验位起始地址 03

例子：1 号设备的校验位修改为 Even

上位机发送：01 06 00 03 00 02 F8 0B

下位机回复：01 06 00 03 00 02 F8 0B

波特率起始地址 04

例子：1 号设备的波特率修改为 2400

上位机发送：01 06 00 04 00 02 49 CA

下位机回复：01 06 00 04 00 02 49 CA

2.3.1、修改多个寄存器

功能码：10

上位机发送：

地址码	功能码	起始地址高位	起始地址低位	数量高位	数量低位	数据1	数据1	数据2	数据2	...	CRC16低位	CRC16高位
						1	1	2	2			

						高位	低位	高位	低位			
Address	10	00	01	00	02	00	09	00	02	...	CRC16L	CRC16H

下位机回复：

地址码	功能码	起始地址 高位	起始地址 低位	数量 高位	数量 低位	CRC16 低位	CRC16 高位
Address	10	00	01	00	02	CRC16L	CRC16H

例子：1号设备的停止位修改为 01，校验位修改为 Even，波特率修改为 9600

上位机发送：01 10 00 02 00 03 00 01 00 03 00 04 2B 62

00 01 表示修改停止位 01

00 03 表示修改校验位为 Even

00 04 表示修改波特率为 9600，详情参看附录 1

下位机回复：01 10 00 02 00 03 21 C8

三、节段位移计参数配置和数据获取流程

确定监测方向和 X-Mark 方向的夹角 θ ，获取井口第一节 X 偏转角度 α ，则监测方向的偏转角为 $\beta = \alpha + \theta$ ，设置每个节段的监测方向偏转角，获取 X/Y/Z 三个方向位移。

按照如下示例设置。

3.1、确定监测方向和 X-Mark 方向的夹角

第一步：将手机和设计监测方向平行放置，读取度数 $\theta_1 = 95^\circ$

第二步：将手机和 X-Mark 方向平行放置，读取度数 $\theta_2 = 72^\circ$

第三步：计算两次角度差值 $\theta = \theta_1 - \theta_2 = 22^\circ$

3.2、获取井口第一节 X 偏转角度 α

发送：01 04 00 26 00 02 90 00

返回：01 04 04 42 90 00 00 EF 8B

其中：42 90 00 00 为角度 $\alpha = 72^\circ$

则监测方向的偏转角度 $\beta = \alpha + \theta = 72^\circ + 22^\circ = 94^\circ$ 。此例中默认第一节的地址码为 01。

3.3、设置每个节段的监测方向偏转角 β

单节设置：

发送: 01 06 00 08 24 B8 13 7A

返回: 01 06 00 08 24 B8 13 7A

例如 $\beta=94 \rightarrow 9400 \rightarrow 24\text{ B8}$, 角度 $\times 100$ 转为 16 进制的整数。

单节设置, 根据系统中的节段的地址码一一进行设置。

广播设置:

发送: 00 06 00 08 42 90 39 15

返回: 无

3.4、获取 X/Y/Z 三个方向位移

X/Y 为监测方向平面位移, Z 为深度。获取位移可以单个获取也可以多个获取, 多个获取参考请求多个数据例子。

获取第一节段地址码 01 的 X/Y/Z 方向位移

读取 X 方向位移发送: 01 04 00 2A 00 02 50 03

读取 Y 方向位移发送: 01 04 00 2C 00 02 B0 02

读取 Z 方向位移发送: 01 04 00 2E 00 02 11 C2

获取第二节段地址码 02 的 X/Y/Z 方向位移

·
·

获取第 n 节段地址码 n 的 X/Y/Z 方向位移

附录 1

保持寄存器起始地址对应表：

地址码	数据位	停止位	校验位	波特率
00	01	02	03	04

地址码范围 1-254。

数据位有 7、8、9，默认 8。

停止位有 1 和 2，默认 1。

校验位有 0, 1, 2。0 表示无，1 表示 Odd，2 表示 Even。默认无。

波特率对应选项

数据位	1	2	3	4	5	6
波特率	1200	2400	4800	9600	14400	19200
数据位	7	8	9	10		
波特率	38400	57600	115200	194000		

附录 2

1、单精度浮点数至 IEEE754 标准 4 字节转换函数(C 语言)

```
//float 转化为 IEEE754 4 个字节 big_endian
//若编译器采用 little endian 模式，请先逆转 bdat 数组
void float2byte(float fdat,unsigned char *bdat)
{
    unsignedchar i;
    //获得 float 数据所在四个字节地址
    unsignedchar*tmp =(unsigned char *)&fdat;
    //间接寻址，获得 float 所在 4 字节地址中的数值
    for(i = 0;i < (sizeof(float));i++)
        *(bdat + i) =*(tmp+ i);
}
```

2、IEEE754 标准 4 字节至单精度浮点数转换函数(C 语言)

```
//IEEE754 4 个字节转化为 float big_endian
//若编译器采用 little endian 模式，请先逆转 bdat 数组
```

```
float byte2float(unsigned char *bdat)
{
    return *((float*)bdat); //强制转换成 float
}
```

附录 3

数据格式:

读取的数据为 4 个字节 32 位单精度浮点数, 表示格式采用 IEEE754 标准。