

DB14

山 西 省 地 方 标 准

DB 14/T 1731—2018
2024年2月28日确认有效

矿用安全监控系统传感器基于 RS485 的有线传输协议

2018 - 12 - 05 发布

2019 - 02 - 05 实施

山西省市场监督管理局 发布

目 次

前言 II

1 范围 1

2 规范性引用文件 1

3 术语定义 1

4 传输接口 3

5 通信规范 3

6 传输过程 3

7 协议格式 4

附录 A（规范性附录） 传感器寄存器定义..... 9

附录 B（规范性附录） 传感器类型编码..... 11

附录 C（规范性附录） 传感器状态说明..... 12

附录 D（规范性附录） 测量值类型说明..... 13

附录 E（规范性附录） 参数测量值说明..... 14

前 言

本标准依据GB/T 1.1-2009 《标准化工作导则 第1部分 标准的结构和编写》给出的规则起草。

本标准由山西省计量标准化技术委员会提出并归口。

本标准起草单位：山西省计量科学研究院、山西阳光三极科技股份有限公司。

本标准主要起草人：韩建书、李梅、卫睿华、史增福、高耀杰、王强、姚永强、卫蔚、陈少祥。

矿用安全监控系统传感器基于 RS485 的有线传输协议

1 范围

本标准规定了矿用安全监控系统传感器基于RS485的有线传输协议的传输接口、通信规范、传输过程及协议格式。

本标准适用于点对点及一点对多点的通信方式。

本标准适用于矿用传感器生产、销售、检测和使用的相关单位。

2 规范性引用文件

下列文件对于本标准的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本标准。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本标准。

GB/T 7665-2005 传感器通用术语

GB/T 16733-1997 国家标准制定程序的阶段划分及代码

GB/T 17710-1999 数据处理校验码系统

GB/T 19582.1-2008 基于Modbus协议的工业自动化网络规范 第1部分：Modbus应用协议

GB/T 19582.2-2008 基于Modbus协议的工业自动化网络规范 第2部分：Modbus协议在串行链路上的实现指南

AQ 1029 煤矿安全监测监控系统及检测仪器使用管理规范

MT/T 1116 煤矿安全生产监控系统联网技术要求

3 术语定义

下列术语和定义适用于本标准。

3.1

传感器

能感受被测量并按照一定的规律转换成可用输出信号的器件或装置，通常由敏感元件和转换元件组成。

注1：敏感元件，指传感器中能直接感受或响应被测量的部分。

注2：转换元件，指传感器中能将敏感元件感受或响应的被测量转换成适于传输或测量的电信号部分。

注3：当输出为规定的标准信号时，则称为变送器。

3.2

多参数传感器

能感受两种或两种以上被测物理量的传感器。

3.3

总线

总线是传送数据和地址信号的公共通道。

3.4

寄存器

寄存器是中央处理器内的组成部分。寄存器是有限存贮容量的高速存贮部件，它们可用来暂存指令、数据和地址。

3.5

ModBus 协议

ModBus 协议是 OSI 模型第 7 层上的应用层报文传送协议，在不同类型总线或网络的设备之间提供客户机/服务器通信，将串行链路上的协议标准化，实现一个主站同一个或多个从站之间交换 Modbus 请求。

3.6

485 总线通信

一种采用平衡发送和差分接收的平衡数字多点系统中的驱动器和接收器的电气特性的标准，以两线间的电压差为+（2—6）V 表示逻辑“1”；以两线间的电压差为-（2—6）V 表示逻辑“0”。

3.7

主从式通信

主从式通信是指主机发送的信息可以传送到各个从机或指定的从机，而各个从机的信息只能发送给主机，总线上只有主机可以主动请求与从机通信，而从机不能主动请求与主机或其他从机通信。

3.8

主机

在总线上能主动发起通信，对总线信号发出控制、接收、查询的设备。本标准中主机指主动发出读数据命令并接收从机（传感器）数据的设备。

3.9

从机

不能在总线上主动发起通信，只能挂接在总线上，对总线信号进行接收、回复的设备。本标准中从机即传感器。

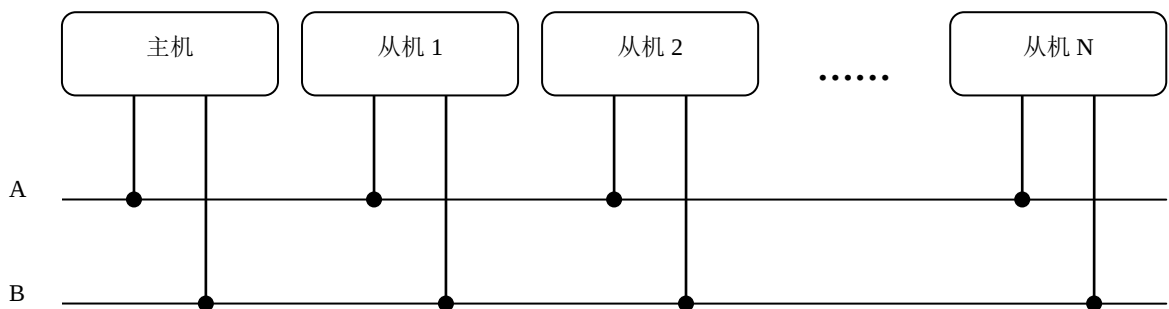
3.10

终端地址

系统中终端设备的地址编码，简称终端地址（以下简称地址）。

4 传输接口

4.1 信号线采用二线制串行总线结构，主从式传输方式，传输模型见图 1：



注：上图A、B表示信号的正负，A表示信号正，B表示信号负。

图1 传输模型

4.2 典型的传输方式为总线上有一台检测主机，N 台从机（即传感器），每台传感器地址唯一。主机采用轮巡地址方式呼叫从机，并接收在线从机的回复信息。

5 通信规范

- 字节传输按异步方式进行；
- 波特率：1200bps，2400bps，4800bps，9600bps 可选；
- 起始位：1 位；
- 数据位：8 位；
- 奇偶校验位：无；
- 停止位：1 位；

6 传输过程

一次完整的通信过程分为 2 个阶段：主机询问、从机应答。

在通信开始之前，主从机应事先设置好通信波特率（主从机的波特率必须一致）、地址编号等参数。

主机询问阶段：主机向指定地址的从机发送命令，主机发送完毕后等待一定时间获取从机应答，在

等待时间结束时，如果主机收到从机的正确应答信息，则说明主机与该从机通信正常，如果主机未收到任何信息或者收到错误信息或者应答信息尚未传递完毕，均视为主机与从机通讯异常。

从机应答阶段：从机解析接收到的命令，并组织相应信息回复到主机，从机清除接收缓冲区及相关变量，等待主机的下次命令。

任何一次完整的通信过程都是由主机方发起的，从机在被主机寻址前只能处于侦听状态，从机在接收命令结束后，判断指令是否寻址自己，如果是，从机对主机给予相应的应答；否则，从机不予应答。

7 协议格式

7.1 MODBUS 协议的指令格式为：

主机（请求）：

表1 主机（请求）指令格式

地址域	功能码	数据域	校验
-----	-----	-----	----

从机（响应）：

表2 从机（响应）指令格式

地址域	功能码	数据域	校验
-----	-----	-----	----

地址域：被主机请求的从机地址。

功能码：主机对从机请求的操作。

数据域：

- a) 主机（请求）：主机对从机执行请求的内容；
- b) 从机（响应）：从机对主机响应后的返回值；

校验：从地址域开始，到数据域结束之间的数据采用 CRC16 校验方式。

注：当从机对主机响应时，它使用功能码来指示正常（无差错）响应或者出现某种差错（称为异常响应）。对于一个正常响应来说，从机仅对原始功能码响应。对于异常响应，从机返回一个与原始功能码等同的码，设置该原始功能码的最高有效位为逻辑 1。

表3 主机（请求）指令格式

地址域	传感器地址	1 个字节	0X01 至 0X7F
功能码	功能码	1 个字节	0X03（读取从机寄存器的数值）
数据域	起始寄存器地址	2 个字节	0X0000-0XFFFF （起始寄存器地址高字节在前，低字节在后）
	寄存器数量	2 个字节	0X01-0X7D （寄存器数量：高字节在前，低字节在后）
校验	CRC 校验码 Lo	1 个字节	CRC 校验码低字节
	CRC 校验码 Hi	1 个字节	CRC 校验码高字节
注1：寄存器地址含义详见附录A 传感器寄存器定义。 注2：CRC校验码的计算方法见7.2 CRC校验码计算。			

表4 从机（正常响应）指令格式

地址域	传感器地址	1 个字节	0X01 至 0X7F
功能码	功能码	1 个字节	0X03
数据域	字节数	1 个字节	2×N
	寄存器值 Hi	1 个字节	起始寄存器值高字节
	寄存器值 Lo	1 个字节	起始寄存器值低字节
	...	...	...
	寄存器值 Hi	1 个字节	起始寄存器后第（N-1）个 寄存器值高字节
	寄存器值 Lo	1 个字节	起始寄存器后第（N-1）个 寄存器值低字节
校验	CRC 校验码 Lo	1 个字节	CRC 校验码低字节
	CRC 校验码 Hi	1 个字节	CRC 校验码高字节
注：N=主机请求寄存器的数量。			

表5 （异常响应）指令格式

地址域	传感器地址	1 个字节	0X01 至 0X7F
功能码	差错码	1 个字节	0X83
数据域	异常码	1 个字节	01 或 02 或 03 或 04
校验	CRC 校验码 Lo	1 个字节	CRC 校验码低字节
	CRC 校验码 Hi	1 个字节	CRC 校验码高字节

表6 异常码列表

代码	名称	含义
01	非法功能	对于传感器来说，询问中接收到的功能码是不可允许的操作。
02	非法数据地址	对于传感器来说，询问中接收到的数据地址是不可允许的地址。特别是，参考号和传输长度的组合是无效的。对于带有 100 个寄存器的控制器来说，带有偏移量 96 和长度 4 的请求会成功，带有偏移量 96 和长度 5 的请求将产生异常码 02。
03	非法数据值	对于传感器来说，询问中包括的值是不可允许的值。这个值指示了组合请求剩余结构中的故障，例如：隐含长度是不正确的。并不意味着，MODBUS 协议不知道任何特殊寄存器的任何特殊值的重要意义，寄存器中被提交存储的数据项有一个应用程序期望之外的值。
04	从站设备故障	当传感器正在设法执行请求的操作时，产生不可重新获得的差错。

7.2 CRC 校验码计算

校验码采用 CRC 循环冗余校验码。校验码是一个 16 位二进制数，占 2 个字节。CRC 校验码计算从通讯信息帧的地址开始、到数据区结束（包含数据区）。循环冗余校验码计算方法如下：

- c) 将一个 16 位寄存器装入十六进制 FFFF(全 1)。将之称作 CRC 寄存器。
- d) 将报文的第一个 8 位字节与 16 位 CRC 寄存器的低字节异或，结果置于 CRC 寄存器。
- e) 将 CRC 寄存器右移 1 位(向 LSB 方向)，MSB 充零，提取并检测 LSB。
- f) 如果 LSB 为 0：重复步骤 3(另一次移位)。(如果 LSB 为 1：对 CRC 寄存器异或多项式值 0xA001(1010 0000 0000 0001)。
- g) 重复步骤 3 和 4，直到完成 8 次移位。当做完此操作后，将完成对 8 位字节的完整操作。
- h) 对报文中的下一个字节重复步骤 2 到 5，继续此操作直至所有报文被处理完毕。
- i) CRC 寄存器中的最终内容为 CRC 值。

j) 当放置 CRC 值于报文时，先发送低字节，再发送高字节。

函数名称: unsigned short int MBCRC16(unsigned char *pucFrame, unsigned short int usLen)

函数功能: 计算 CRC 校验值

输入参数:

unsigned char *pucFrame /*报文数组的首地址*/

unsigned short int usLen /*报文数组的字节数*/

返回值: unsigned short 类型的 CRC 校验值

unsigned short int MBCRC16(unsigned char *pucFrame, unsigned short int usLen)

```
{
    unsigned char    ucCRCHi = 0xFF;    /* CRC 的高字节初始化 */
    unsigned char    ucCRCLo = 0xFF;    /* CRC 的低字节初始化 */
    unsigned int      iIndex;            /* CRC 查询表索引 */

    while(usLen--)                        /* 计算 CRC */
    {
        iIndex = ucCRCLo ^ *( pucFrame++ );
        ucCRCLo = ( unsigned char )( ucCRCHi ^ aucCRCHi[iIndex] );
        ucCRCHi = aucCRCLo[iIndex];
    }
    return ( unsigned short int )( ucCRCHi << 8 | ucCRCLo );
}
```

高字节表

/* 高位字节的 CRC 值 */

```
static const unsigned char aucCRCHi[] = {
    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, 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
};
低字节表
/* 低位字节的 CRC 值 */
static const unsigned char aucCRCLo[] = {
    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
};

```

附 录 A
(规范性附录)
传感器寄存器定义

A.1 传感器寄存器定义见表A.1。

表A.1 传感器寄存器定义

参数	寄存器地址	参数定义		备注
参数 0	0000H	高字节	传感器类型	详见附录 B 传感器类型编码
		低字节	参数个数	传感器测量参数的个数:单参数传感器为 1, 两参数传感器为 2, 以此类推
参数 1	0001H	高字节	传感器状态	详见附录 C 传感器状态说明
		低字节	参数子类型	当为单参数传感器时,子类型与传感器类型相同; 当为多参数传感器时,子类型根据实际情况,详见附录 B 传感器类型编码
	0002H	高字节	测量值类型	详见附录 D 量值类型说明
		低字节	测量值 1	详见附录 E 数测量值说明
	0003H	高字节	测量值 2	
		低字节	测量值 3	
参数 2	0004H	高字节	传感器状态	
		低字节	参数子类型	
	0005H	高字节	测量值类型	
		低字节	测量值 1	
	0006H	高字节	测量值 2	
		低字节	测量值 3	
...	...			...

A.2 单参数传感器示例见表A.2。其中,单参数传感器为甲烷传感器,测量值为 1.20%,传感器状态为三级报警。

表A.2 单参数传感器

参数序号	寄存器地址	参数定义	数据值（十六进制）	解析
参数 0	0000H	传感器类型	0X01	甲烷传感器
		参数个数	0X01	测量参数个数为 1
参数 1	0001H	传感器状态	0X83	传感器三级报警
		参数子类型	0X01	参数类型为甲烷
	0002H	测量值类型	0X02	正数且 1 位小数
		测量值 1	0X00	最高两位分别：0，0
	0003H	测量值 2	0X01	中间两位分别：0，1
		测量值 3	0X20	最低两位分别：2，0

A.3 多参数传感器示例见表A.3。其中，多参数传感器的测量参数个数为 2，第 1 个参数为温度，测量值为 22℃，传感器状态正常；第 2 个参数为风速，测量值-5.2m/s，传感器状态为报警。

表A.3 多参数传感器

参数序号	寄存器地址	参数定义	备注	解析
参数 0	0000H	传感器类型	0X14	多参数传感器
		参数个数	0X02	测量参数个数为 2
参数 1	0001H	传感器状态	0X00	传感器正常
		参数子类型	0X03	参数 1 类型为温度
	0002H	测量值类型	0X00	正数且无小数部分
		测量值 1	0X00	最高两位分别：0，0
	0003H	测量值 2	0X00	中间两位分别：0，0
		测量值 3	0X22	最低两位分别：2，2
参数 2	0004H	传感器状态	0X80	传感器报警
		参数子类型	0X09	参数 2 类型为风速
	0005H	测量值类型	0X11	负数且 1 位小数
		测量值 1	0X00	最高两位分别：0，0
	0006H	测量值 2	0X00	中间两位分别：0，0
		测量值 3	0X52	最低两位分别：5，2

附 录 B
(规范性附录)
传感器类型编码

表B.1 传感器类型编码

类型编号	传感器分类	单位	备注
0x01	甲烷类	%CH ₄	
0x02	一氧化碳类	μ mol/mol	
0x03	温度类	℃	
0x04	氧气类	%O ₂	
0x05	二氧化碳类	%CO ₂	
0x06	压力类	kPa	
0x07	压力类	MPa	
0x08	液位类	m	
0x09	风速类	m/s	
0x0a	粉尘类	mg/m ³	
0x0b	硫化氢	μ mol/mol	
0x0c	氢气	%H ₂	
0x0d	二氧化氮	μ mol/mol	
0x0e	煤仓煤位	m	
0x0f	温湿度	℃、%RH	
0x10	瞬时流量	m ³ /h	
0x11	累计流量	m ³	
0x12	开关量类		
0x13	控制量		
0x14	多参数		

附 录 C
(规范性附录)
传感器状态说明

表C.1 传感器状态

传感器状态		
传感器状态数值	传感器状态含义	备注
0	表示传感器正常	
1	表示传感器故障	
8	表示传感器参数断线	
128	表示传感器报警	
129	表示传感器一级报警	
130	表示传感器二级报警	
131	表示传感器三级报警	
132	表示传感器四级报警	

附 录 D
(规范性附录)
测量值类型说明

表D.1 测量类型说明

测量值类型							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	0	符号位： =0 表示正 数； =1 表示负 数	0	小数点后的位数： =0 表示无小数位； =1 表示小数点后 1 位； =2 表示小数点后 2 位； =3 表示小数点后 3 位；		

附 录 E
(规范性附录)
参数测量值说明

参数测量值采用 BCD 码表示，共占用 3 个字节，参数测量值 1 为高位，参数测量值 3 为低位，表示数值范围为 0~999999。

表E.1 参数测量值说明

参数测量值 1 (BCD 码)		参数测量值 2 (BCD 码)		参数测量值 3 (BCD 码)	
Bit7~Bit4	Bit3~Bit0	Bit7~Bit4	Bit3~Bit0	Bit7~Bit4	Bit3~Bit0
高位 ----->-----低位					