

DME 优化 Modbus 通讯协议

发放范围: 项目组、评审组

本模板责任部门 (Owner): 研发技术管理处 (模板修订记录见最后一页)

项目名称 Project	DME 优化	项目代码 Code	7510044
拟制人 Author	周遵法/李垂君	日期 Date	2013-11-10
审核人 Reviewed by	郝建伟	日期 Date	2013-11-22
批准人 Approved by	郎炳莉	日期 Date	2013-11-22

修订记录 Chang Record:

版本 Version	日期 Date	修改内容及理由 Change and Reason	拟制人 Author	审核人 Reviewed by	批准人 Approved by
V1.0	2013-11-22	重新定义寄存器	周遵法/ 李垂君	郝建伟	郎炳莉
V1.1	2013-12-27	1、修改 CRC 字节传输描述 2、修正协议支持的最大长度和异常描述	周遵法	郝建伟	郎炳莉
V1.2	2014-4-22	1. 增加红外加湿高水位告警	李垂君	郝建伟	郎炳莉

自检表（CHECKLIST）

序号	内容	检视情况 (Y/N)
1	是否使用最新版本的通讯协议模版（从流程优化专栏拆离）	Y
2	封面内容是否填写完整——项目名称和项目编码	Y
3	封面内容是否填写完整——拟制人（项目经理）和拟制时间	Y
4	封面内容是否填写完整——审核人（开发经理）	Y
5	封面内容是否填写完整——批准人（技术管理处经理）	Y
6	是否使用统一字体（宋体/黑色）且无错别字	Y
7	是否按照《对外通讯硬件接口缺省规范》要求进行检视	Y
8	是否对协议中的通讯方式进行说明	Y
9	是否对协议中用到的数据类型进行说明	Y
10	是否根据查检表进行了自检	Y

目录

1. 概述	5
1.1. 协议概述.....	5
1.2. 适用范围.....	5
1.3. 引用标准.....	5
2. 底层协议.....	5
2.1. 物理接口.....	5
2.2. 数据传输速率.....	5
2.3. 字符格式.....	5
2.4. 传输方式.....	5
3. 数据类型.....	5
4. 通讯方式.....	6
4.1. 网络拓扑.....	6
4.2. 通讯的建立.....	6
5. 应用层数据包/帧格式定义	6
5.1. 帧格式.....	6
5.1.1. 地址	6
5.1.2. 功能码	6
5.1.3. 数据	6
5.1.4. 错误校验	7
5.1.5. RTU 帧举例.....	8
6. 命令/响应信息详解.....	8
6.1. 读取运行参数（0x03）	8
6.1.1. 主站格式	8
6.1.2. 子站响应格式.....	9
6.1.3. 寄存器列表.....	9
6.2. 设置运行参数（0x06）	1 1
6.2.1. 主站格式	1 1
6.2.2. 子站响应格式.....	1 1
6.2.3. 寄存器列表.....	1 2
7. 不正常响应.....	1 2
7.1. 异常回应格式如下:	1 2

8. 附录	1 3
-------------	-----

1. 概述

1.1. 协议概述

本协议定义DME优化空调与后台监控之间的Modbus通讯协议规范。

本协议采用Modbus标准的RTU协议。

1.2. 适用范围

DME优化空调软件开发工程师以及后台监控开发工程师使用本协议作为开发依据，测试工程师使用本协议指导测试。

1.3. 引用标准

《MODBUS-RTU》

2. 底层协议

2.1. 物理接口

RS-485

2.2. 数据传输速率

波特率可通过显示面板设置。

参数	默认	范围
波特率	19200	1200/2400/4800/9600/19200

2.3. 字符格式

十六进制

2.4. 传输方式

RTU 模式：起始位为 1 位，数据位 8 位，停止位 1 位，无奇偶校验。

CRC 循环冗余检验。

3. 数据类型

4. 通讯方式

4.1. 网络拓扑

在局站内的监控系统为分布式结构。局站监控制单元（SU）与设备监控模块（SM）的通信方式为主从方式，监控单元为上位机，监控模块为下位机。

在本系统中，DME空调为SM，上位机为SU。

4.2. 通讯的建立

SU呼叫SM并下发命令，SM收到命令后并返回响应信息。

5. 应用层数据包/帧格式定义

5.1. 帧格式

帧格式如下：

地址	功能码 FC	数据	CRC
1 byte	1 byte	N bytes	2 bytes

5.1.1. 地址

监控地址默认为 1，在菜单中可设。

参数	默认	下限	上限
通讯地址	1	1	247

5.1.2. 功能码

码	功能	说明
0x03	读运行参数	读设置值和运行参数，每个数据占 2 个字节，第 1 个字节为高字节，第 2 个字节为低字节，一次最多可读 125 个连续的寄存器。读 4xxxx 寄存器
0x06	运行参数设置	写入运行参数，每个参数占 2 个字节，第 1 个字节为高字节，第 2 个字节为低字节。写 4xxxx 寄存器。

5.1.3. 数据

数据域的长度随请求和响应的命令码不同而不同。

5.1.4. 错误校验

Modbus 错误校验采用 16 bit(2 字节)的 CRC (Cyclical Redundancy Check)校验。

CRC 校验算法如下:

/* 高位字节的 CRC 值*/

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

```
static char auchCRCLo[] = {  
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  
};
```

```

unsigned short CRC16 ( unsigned char *puchMsg,unsigned short usDataLen )
{
    unsigned char uchCRCHi = 0xFF ; /* CRC 的高字节初始化*/
    unsigned char uchCRCLo = 0xFF ; /* CRC 的低字节初始化*/
    unsigned ulIndex ; /* CRC 查询表索引*/
    while (usDataLen--) /* 完成整个报文缓冲区*/
    {
        ulIndex = uchCRCLo ^ *puchMsg++ ; /* 计算 CRC */
        uchCRCLo = uchCRCHi ^ auchCRCHi[ulIndex] ;
        uchCRCHi = auchCRCLo[ulIndex] ;
    }
    return (uchCRCHi << 8 | uchCRCLo) ;
}

```

5.1.5. RTU 帧举例

下面的例子说明了典型的 Q/R 帧交互过程，一次通讯由 SU 发起，SU 发送一个查询帧，SM 根据需求返回一个应答帧。

03 查询请求

子站地址	01H
FC	03H
起始寄存器高字节	--
起始寄存器低字节	--
读寄存器数高字节	--
读寄存器数低字节	--
CRC 低字节	--
CRC 高字节	--

03 请求响应

子站地址	01H
FC	03H
响应数据字节数	--
Byte 0	--
Byte 1	--
:	--
:	--
Byte n	--
CRC 低字节	--
CRC 高字节	--

6. 命令/响应信息详解

6.1. 读取运行参数 (0x03)

6.1.1. 主站格式

读取运行参数的查询帧如下：

内容	长度 (Byte)
地址	1
功能码 (0x03)	1
起始寄存器高字节	1
起始寄存器低字节	1
读寄存器数高字节	1
读寄存器数低字节	1
CRC 低字节	1
CRC 高字节	1

6.1.2. 子站响应格式

子站响应读取运行参数的个数如下：

内容	长度 (Byte)
地址	1
功能码 (0x03)	1
响应字节数	1
寄存器值	N*2
CRC 低字节	1
CRC 高字节	1

注：N 指的是查询寄存器的数量

6.1.3. 寄存器列表

可读运行参数列表如下：

序号	寄存器	参数名称	范围	默认值	变比	单位	备注
1	40001 ^{注1}	测量温度	[-400, 1000]	/	x10	°C	R
2	40002	测量湿度	[0, 1000]	/	x10	%	R
3	40003	高压压力	[0, 500]	/	x10	Bar	R
4	40004	温度设定	[150, 350]	240	x10	°C	R/W
5	40005	温度精度	[10, 50]	30	x10	°C	R/W
6	40006	湿度设定	[20, 80]	500	x10	%	R/W
7	40007	湿度精度	[10, 100]	50	x10	%	R/W
8	40008.0 ^{注2}	告警状态.高压告警	/	/	/	/	R
9	40008.1	告警状态.低压告警	/	/	/	/	R
10	40008.2	告警状态.排气温度告警	/	/	/	/	R
11	40008.3	告警状态.高温告警	/	/	/	/	R
12	40008.4	告警状态.低温告警	/	/	/	/	R
13	40008.5	告警状态.高湿告警	/	/	/	/	R
14	40008.6	告警状态.低湿告警	/	/	/	/	R
15	40008.7	告警状态.加湿器故障	/	/	/	/	R
16	40008.8	告警状态.回风温度传	/	/	/	/	R

		传感器故障告警					
17	40008.9	告警状态.高压锁定	/	/	/	/	R
18	40008.10	告警状态.低压锁定	/	/	/	/	R
19	40008.11	告警状态.排气锁定	/	/	/	/	R
20	40008.12	告警状态.地板溢水	/	/	/	/	R
21	40008.13	告警状态.电源丢失	/	/	/	/	R
22	40008.14	告警状态.电源过压	/	/	/	/	R
23	40008.15	告警状态.电源欠压	/	/	/	/	R
24	40009.0	告警状态.电源频偏	/	/	/	/	R
25	40009.1	告警状态.电源缺相	/	/	/	/	R
26	40009.2	告警状态.电源反相	/	/	/	/	R
27	40009.3	告警状态.用户自定义 1告警	/	/	/	/	R
28	40009.4	告警状态.节能卡故障	/	/	/	/	R
29	40009.5	告警状态.主风机维护 提示	/	/	/	/	R
30	40009.6	告警状态.加湿器维护	/	/	/	/	R
31	40009.7	告警状态.过滤网维护	/	/	/	/	R
32	40009.8	告警状态.气流丢失告 警	/	/	/	/	R
33	40009.9	告警状态.用户自定义 2告警	/	/	/	/	R
34	40009.10	告警状态.压力传感器 故障	/	/	/	/	R
35	40009.11	告警状态.回风湿度传 感器故障	/	/	/	/	R
36	40009.12	告警状态.气流传感器 故障	/	/	/	/	R
37	40009.13	红外加湿高水位告警	/	/	/	/	R
38	40009.14	保留	/	/	/	/	R
39	40009.15	保留	/	/	/	/	R
40	40010.0	机组状态.风机状态	/	/	/	/	R
41	40010.1	机组状态.制冷状态	/	/	/	/	R
42	40010.2	机组状态.加热状态	/	/	/	/	R
43	40010.3	机组状态.加湿状态	/	/	/	/	R
44	40010.4	机组状态.除湿状态	/	/	/	/	R
45	40010.5	机组状态.机组状态 ^{注3}	^{注3}	关机	/	/	R
46	40010.6						R
47	40010.7						R
48	40010.8	机组状态.机组属性 ^{注4}	^{注4}	单机	/	/	R
49	40010.9		/	/	/	/	R
50	40010.10	保留	/	/	/	/	R
51	40010.11	保留	/	/	/	/	R
52	40010.12	保留	/	/	/	/	R
53	40010.13	保留	/	/	/	/	R

54	40010.14	保留	/	/	/	/	R
55	40010.15	保留	/	/	/	/	R

注 1: 起始寄存器号 = 实际寄存器号 - 40001;

注 2: 40008.0-40009.15 Bit 位编码: 1: 告警 0: 无告警

注 3: 40010.5-40010.7 3 位二进制编码: 0 表示关机, 1 表示运行, 2 表示待机, 3 表示锁定, 其它无效;

注 4: 40010.8-40010.9 2 位二进制编码: 0 表示单机, 1 表示主机, 2 表示备机;

注 5: 读取超出范围寄存器返回地址异常。如果请求数据量超过最大长度, 则返回 03 (不合法数据) 错误。

6.2. 设置运行参数 (0x06)

6.2.1. 主站格式

设置运行参数格式如下:

内容	长度 (Byte)
地址	1
功能码 (0x06)	1
寄存器高字节	1
寄存器低字节	1
设置值高字节	1
设置值低字节	1
CRC 低字节	1
CRC 高字节	1

6.2.2. 子站响应格式

响应运行参数设置数据格式如下:

内容	长度 (Byte)
地址	1
功能码 (0x06)	1
寄存器高字节	1
寄存器低字节	1
设置值高字节	1
设置值低字节	1
CRC 低字节	1
CRC 高字节	1

6.2.3. 寄存器列表

可写运行参数如下表：

序号	寄存器	参数名称	范围	默认值	变比	单位	备注
1	40004 ^{注1}	温度设定 ^{注2}	[150, 350]	240	x10	°C	R/W
2	40005	温度精度 ^{注3}	[10, 50]	30	x10	°C	R/W
3	40006	湿度设定 ^{注4}	[20, 80]	500	x10	%	R/W
4	40007	湿度精度 ^{注5}	[10, 100]	50	x10	%	R/W
5	40011	遥控开关机 ^{注6}	/	/	/	/	W

注 1：起始寄存器号 = 实际寄存器号 - 40001；

注 2：温度设定范围：150~350，即 15C ~35C ；

注 3：温度精度范围：10~50，即 1C ~5C ；

注 4：湿度设定范围：200~800，即 20%~80%；

注 5：湿度精度范围：10~100，即 1%~10%；

注 6：10H，遥控空调开机；1FH，遥控空调关机。

7. 不正常响应

不正常响应时有 2 个与正常响应不同的区域：

- 1) 功能代码区：正常响应时，从机的响应功能代码区，带原查询的功能代码，所有功能代码的 MSB 为 0（其值低于 80H）；不正常响应时，从机把功能代码的 MSB 置为 1，使功能代码高于 80H，高于正常响应的值。
- 2) 数据区：正常响应中，数据区含有数据或统计值；在不正常响应中，数据区为一个不正常代码，说明从机产生不正常响应的条件和原因。
- 3) 不正常代码如下：

代码	名称	含义
1	不合法功能代码	从机接收的是一种不能执行功能代码。发出查询命令后，该代码指示无程序功能
2	不合法数据地址	接收的数据地址，是从机不允许的地址
3	不合法数据	查询或设置数据区的值是从机不允许的值
4	从机设备故障	从机执行主机请求的动作时出现不可恢复的错误（只对 06 功能码）

7.1. 异常回应格式如下：

内容	长度 (Byte)
地址	1
功能码+0x80	1
数据区代码	1
CRC 低字节	1
CRC 高字节	1

8. 附录

模板修订记录（可删除）

版本 Ver	日期 Date	修改内容及理由 Change and Reason	拟制人 Author	审核人 Reviewed by	批准人 Approved by
1.0	2009-03-02	初稿	杨爱泉	李其祥	丁浩华
1.1	2010-11-26	更改模板和密级	黄怡鸣	孙泽民	