	文档编号	文档版本	文档密级
	05130010	A/5	内部公开
协议名称	U3-EC 空调 MODBUS 协议		

U3-EC 空调 MODBUS 协议

Revision Record 修订记录

Date 日期	Revision Version 修订版本	Software Version 软件版本	Change Description 修改描述	Author 作者

目　录

空调与上位机 Modbus 通信协议

1 说明

本协议描述了空调与其专用上位机监控模块进行命令控制和数据交换的协议。

《协议》中规定的功能主要有：

1) 上位机通过发读取命令获取空调的相关信息；

2) 上位机通过发写命令设置相关参数和动作控制；

通讯过程以上位机为主节点，通过一问一答的方式进行信息交互；从节点中的各种信息和参数均以目标寄存器作为存储地址，主节点通过访问寄存器的方式完成读写命令。本协议支持一个主节点、多个从节点组网，从节点以地址来区分，地址设置范围为1-127,不同的从节点对应不同的地址，不能有相同地址的从节点挂在同一条通讯总线上。

2 专用术语

主节点：后台监控系统。

从节点：空调设备控制器。

RS485：一种串行通讯标准，可支持半双工串行近程通讯；

读命令：由主节点发向从节点，使从节点返回对应的寄存器的内容；

写命令：由主节点打包相关参数，发向从节点，完成对应参数的设置；

寄存器地址：从节点的每个信号和参数均对应一个 2 字节的地址，主节点获取相关信息或设置相关参数均是以访问这些寄存器的方式来完成的，这个地址就称为寄存器。

3 物理接口

3.1 串行通信口电气标准

从节点以 RS485 方式通过串口与主节点通讯。

3.2 信息传输方式

通讯传输采用异步方式，并以字节帧(数据帧)为单位。在主节点和从节点之间传递的每

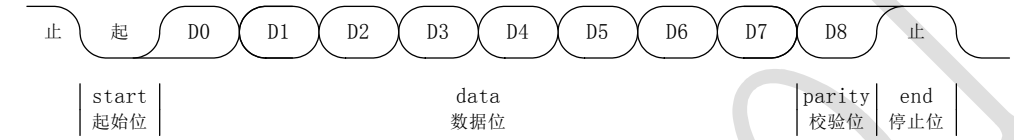
一个数据帧都是11位的串行数据流。

数据帧格式：

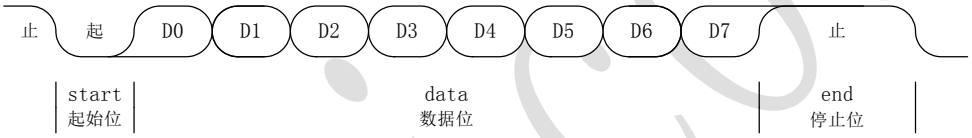
起始位	1位
数据位	8位(低位在前、高位在后)
奇偶校验位	无：本协议不采用奇偶校验位；
停止位	1位 （即实际的奇偶校验位强制为高电平）

参考：

有校验位的时序图：



无校验位的时序图：



3.3 数据传输速率

缺省波特率使用9600bps

4 物理层通信方式

4.1 基本过程

从节点上电或复位，稳定运行后，即可响应主节点的读写命令；当从接点接收到相关命令后，正常情况下返回主节点所需的信息，异常情况返回具体错误类型对应的错误码。

5 应用层命令类型及格式

当通讯命令发送至仪器时，符合相应的地址码的设备接收通讯命令，读取信息，如果没

有出错，则执行相应的任务；然后把执行结果返送给发送者。返送的信息中包括地址码、执行动作的功能码、执行动作后的数据以及错误校验码(CRC)。如果出错就不发送任何信息。

5.1 信息帧格式

START	ADDR	CMD	DATA	CRC	END
起始	地址码	功能码	数据	错误校验	结束
延时(>=3.5个字符的时间)	1字节 8位	1字节 8位	N字节 N×8位	2字节 16位	延时(>=3.5个字符的时间)

注1：最大帧长不大于255个字节；

注2：CRC校验码低字节在前，高字节在后；

5.2 功能码

功能码（CMD）是每次通讯传送的信息帧中的第二个数据帧。ModBus通讯规约定义功能码为1～127(01H～7FH)。本协议利用其中的一部分功能码。作为主节点请求发送，通过功能码告诉从节点执行什么动作。作为从节点响应，从节点发送的功能码与主节点发送来的功能码一样，并表明从节点已响应主节点进行操作。如果从节点发送的功能码的最高位是1(功能码＞127)，则表明从节点没有响应或出错。

命令编码	含义	备注
0x03	读命令	支持单个和多个寄存器连续读取
0x10	写多个寄存器命令	支持多个寄存器连续写动作
0x06	写单个寄存器命令	支持单个寄存器连续写动作

5.3 读命令格式

注3：MSB表示高字节；LSB表示低字节。

注4：每个寄存器存放两个字节；对寄存器数据类型为一个字节的数据，要求存放在低字节（LSB）。

主节点发送帧格式：

序号	0	1	2	3	4	5	6	7

字段定义	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
解释	控制器地址	命令类型	寄存器起始地址		寄存器个数 n		CRC 校验	

从节点正常应答帧格式：

序号	0	1	2	3	4	5	6	…	L+1	L+2	L+3	L+4
字段定义	ADDR	CMD	Length	MSB	LSB	MSB	LSB	…	MSB	LSB	LSB	MSB
解释	控制器地址	命令类型	发送字节数 L=n*2	第一个寄存器的值		第二个寄存器的值		…	最后一个寄存器的值		CRC 校验	

从节点异常应答格式：

序号	0	1	2	3	4
字段定义	ADDR	CMD + 128	ErrCode	LSB	MSB
解释	控制器地址	命令类型 +128	Error Code	CRC 校验	

5.4 写单个寄存器命令格式

主节点发送帧格式：

序号	0	1	2	3	4	5	6	7
字段定义	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
解释	控制器地址	命令类型	寄存器地址		数据		CRC 校验	

从节点正常应答帧格式：

序号	0	1	2	3	4	5	6	7
字段定义	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
解释	控制器地址	命令类型	寄存器地址		数据		CRC 校验	

从节点异常应答格式：

序号	0	1	2	3	4
字段定义	ADDR	CMD +128	ErrCode	LSB	MSB
解释	控制器地址	命令类型 +128	Error Code	CRC 校验	

5.5 写多个寄存器命令格式

主节点发送帧格式：

序号	0	1	2	3	4	5	6	7	8	9	10	…	L+5	L+6	L+7	L+8
字段定义	ADDR	CMD	MSB	LSB	MSB	LSB	Length	MSB	LSB	MSB	LSB	…	MSB	LSB	LSB	MSB
解释	控制器地址	命令类型	起始寄存器地址		寄存器数 n		发送字节数 L = n*2	第一个寄存器的值		第二个寄存器的值		…	最后一个寄存器的值		CRC 校验	

从节点正常应答帧格式：

序号	0	1	2	3	4	5	6	7
字段定义	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
解释	控制器地址	命令类型	起始寄存器地址		寄存器个数		CRC 校验	

从节点异常应答格式：

序号	0	1	2	3	4
字段定义	ADDR	CMD +128	ErrCode	LSB	MSB
解释	控制器地址	命令类型 +128	Error Code	CRC 校验	

说明：CRC 校验范围为 CRC 字段前所有字节的校验。

5.6 错误码定义

当从节点设备向主节点设备发送请求时，从节点希望一个正常响应。从主节点询问中出现下列四种可能事件之一：

1)如果从节点设备接收到无通信错误的请求，并且可以正常地处理询问，那么从节点设备将返回一个正常响应;

2) 如果由于通信错误，从节点没有接收到请求，那么不能返回响应。主节点程序将最终处理请求的超时状态;

3) 如果从节点接收到请求，但是检测到一个通信错误（奇偶校验、LRC、CRC、...），那么不能返回响应。主节点程序将最终处理请求的超时状态;

4) 如果从节点接收到无通信错误的请求，但不能处理这个请求（例如，如果请求读一个不存在的输出或寄存器），从节点将返回一个异常响应，通知用户错误的本质特性;

异常响应报文有两个与正常响应不同的域：

功能码域：在正常响应中，从节点利用响应功能码域来应答最初请求的功能码。所有功能码的最高有效位（MSB）都为0（它们的值都低于128）。在异常响应中，从节点设置功能码的MSB为1。这使得异常响应中的功能码值比正常响应中的功能码值高128。

通过设置功能码的MSB，主节点的应用程序能够识别异常响应，并且能够检测异常码的数据域。

数据域：在正常响应中，从节点可以返回数据域中数据或统计表（请求中要求的任何报文）。在异常响应中，从节点返回数据域中的异常码。这就定义了产生异常的从节点状态。

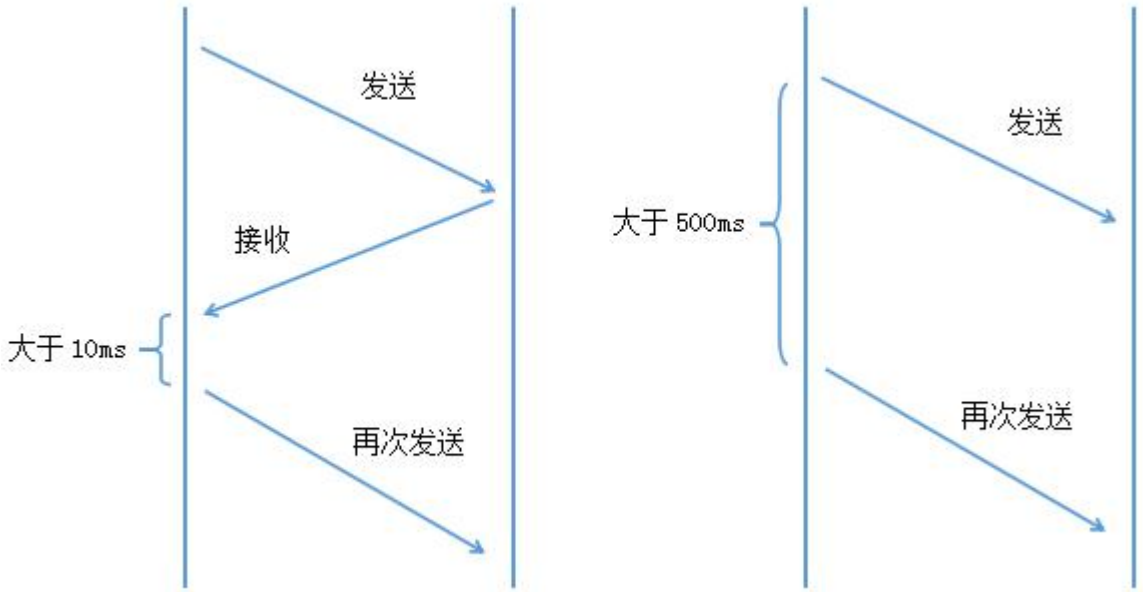
MODBUS 异常码：

异常码名称	说　　明

01 非法功能码	对于从节点来说，询问中接收到的功能码是不可允许的操作。这也许是因为功能码仅仅适用于新设备而在被选单元中是不可实现的。同时，还指出从节点在错误状态中处理这种请求，例如：因为它是未配置的，并且要求返回寄存器值。
02 非法数据地址	对于从节点来说，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	对于从节点来说，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，因为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。

5.7 间隔时间

发送数据并接收数据后，在发送下一帧时，引脚可能会切换不过来，所以统一收发间隔时间大于 10ms，超时时间大于 500ms。



6 CRC 校验算法

6.1 CRC 算法

```
unsigned short count_CRC(unsigned char *addr, int num)
{
    unsigned short CRC = 0xFFFF;
    int i;
    while (num--)
    {
        CRC ^= *addr++;
        for (i = 0; i < 8; i++)
        {
            CRC = (CRC & 0x0001) ? ((CRC >> 1)^0xa001) : (CRC >> 1);
        }
    }
    return CRC;
}
```

7 寄存器列表

约定：

1、 每个寄存器存放两个字节。

2、 数据传输方式：高字节在前，低字节在后

序号	数据	单位	寄存器地址	属　性（读/写）	变比	备注
版本信息						
1	软件版本		0x0000	只读	x1	
运行状态（0：停止，1：运行，2：未选配）						
1	整机状态		0x0100	只读	x1	

2	内风机		0x0101	只读	x1	
3	外风机		0x0102	只读	x1	
4	压缩机		0x0103	只读	x1	
5	电加热		0x0104	只读	x1	
6	应急风机		0x0105	只读	x1	
传感器状态（温度传感器失效上送值为 2000，湿度传感器失效值为 120）						
1	盘管温度	℃	0x0500	只读	x 10	
2	室外温度	℃	0x0501	只读	x 10	
3	冷凝温度	℃	0x0502	只读	x 10	
4	室内温度	℃	0x0503	只读	x 10	
5	湿度	%	0x0504	只读	x 1	
6	排气温度	℃	0x0505	只读	x 10	
7	电流	A	0x0506	只读	x 1000	
8	交流电压	V	0x0507	只读	x 1	
9	直流电压	V	0x0508	只读	x 10	
告警状态（0X00 正常　0X01 告警）						
1	高温告警		0x0600	只读	x1	
2	低温告警		0x0601	只读	x1	
3	高湿告警		0x0602	只读	x1	
4	低湿告警		0x0603	只读	x1	
5	盘管防冻		0x0604	只读	x1	
6	排气高温		0x0605	只读	x1	
7	盘管温感失效		0x0606	只读	x1	
8	室外温感失效		0x0607	只读	x1	
9	冷凝温感失效		0x0608	只读	x1	
10	内温感失效		0x0609	只读	x1	
11	排气温感失效		0x060A	只读	x1	
12	湿感失效		0x060B	只读	x1	
13	内风机故障		0x060C	只读	x1	

14	外风机故障		0x060D	只读	x1	
15	压缩机故障		0x060E	只读	x1	
16	电加热故障		0x060F	只读	x1	
17	应急风机故障		0x0610	只读	x1	
18	高压告警		0x0611	只读	x1	
19	低压告警		0x0612	只读	x1	
20	水浸告警		0x0613	只读	x1	
21	烟感告警		0x0614	只读	x1	
22	门禁告警		0x0615	只读	x1	
23	高压锁定		0x0616	只读	x1	
24	低压锁定		0x0617	只读	x1	
25	排气锁定		0x0618	只读	x1	
26	交流过压		0x0619	只读	x1	
27	交流欠压		0x061A	只读	x1	
28	交流掉电		0x061B	只读	x1	
29	缺相		0x061C	只读	x1	
30	频率异常		0x061D	只读	x1	
31	逆相		0x061E	只读	x1	
32	直流过压		0x061F	只读	x1	
33	直流欠压		0x0620	只读	x1	
系统参数						
1	制冷点	℃	0x0700	读/写	x1	15~50℃
2	制冷回差	℃	0x0701	读/写	x1	1~10℃
3	加热点	℃	0x0702	读/写	x1	-15~25℃
4	加热回差	℃	0x0703	读/写	x1	1~10℃
5	保留		0x0704	读/写	x1	
6	保留		0x0705	读/写	x1	
7	高温点	℃	0x0706	读/写	x1	25~80℃
8	低温点	℃	0x0707	读/写	x1	-20~15℃

9	高湿点	%	0x0708	读/写	x1	0~100%
10	内风机停止点	℃	0x070A	读/写	x1	-20~50℃
遥控参数						
1	保留		0x0800			
1	监控开关机		0x0801	读/写		0x01 开机 0x00 关机

举例：

①读取单个寄存器的值-版本信息：

发送指令：01 03 00 00 00 01 84 0A

返回指令：01 03 02 02 10 B8 E8

解析指令：01 控制器地址

03 命令类型-读命令

02 发送字节数

02 10 返回寄存器的值

B8 E8 校验

②读取多个寄存器的值-盘管温度-室外温度-冷凝温度：

发送指令：01 03 05 00 00 03 05 07

返回指令：01 03 06 01 1F 01 16 01 12 D4 C3

解析指令：01 控制器地址

03 命令类型-读命令

06 发送字节数

01 1F 返回的盘管温度：十进制287→28.7℃

01 16 返回的室外温度：十进制278→27.8℃

01 12 返回的冷凝温度：十进制274→27.4℃

D4 C3 校验

③写单个寄存器的值-制冷点30℃：

发送指令：01 06 07 00 00 1E 08 B6

返回指令：01 06 07 00 00 1E 08 B6

解析指令：01 控制器地址

06 命令类型-写单个寄存器

07 00 寄存器地址

00 1E 数据：十进制30→30℃

08 B6 校验

④写多个寄存器的值-高温点40℃-低温点15℃

发送指令：01 10 07 06 00 02 04 00 28 00 0F 95 B9

返回指令：01 10 07 06 00 02 A0 BD

解析指令：01 控制器地址

10 命令类型-写多个寄存器

07 06 起始地址

00 02 寄存器个数

A0 BD 校验

1 Instructions

This protocol describes the protocol of air conditioner and its special upper machine monitoring module for command control and data exchange.

The functions specified in the *Protocol* mainly include:

1) The upper machine obtains the relevant information of the air conditioner through reading commands;

2) The upper machine sets relevant parameters and action control by writing commands;

The communication process takes the upper machine as the master node, through the mode of question and answer for the information interaction; Various information and parameters of the slave node take the target register as the memory address, and the master node completes the read and write commands by accessing the register. This protocol support a master node and multiple slave nodes networking, and the slave node is distinguished by the address, where the address range is set as 1-128; Different slave nodes correspond to different addresses, while the slave nodes with the same address cannot be on the same communication bus.

2 Special terms

Master node: background monitoring system.

Slave node: air conditioning controller.

RS485: a serial communication standard that supports semi-duplex serial short-range communication;

Read command: sent from the master node to the slave node, so that the slave node can return to the contents of the corresponding register.

Write command: the relevant parameters packaged by the master node, sent to the corresponding slave node, to complete the corresponding parameter settings;

Register address: as each signal and parameter of the slave node is corresponding to an address of 2 bytes, the master node to obtain relevant information or to set related parameters is completed by having access to these registers, where this address is called the register.

3 Physical interface

3.1 Serial communication port electrical standard

The slave node communicates with the master node in RS485 way via the serial port.

3.2 Information transmission mode

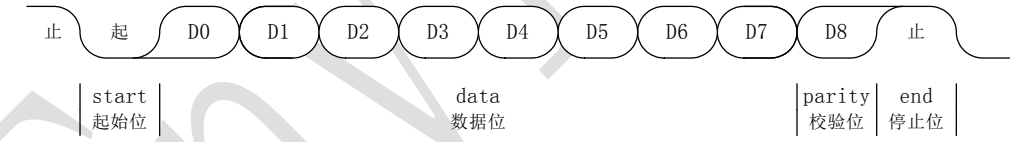
Communication transmission is asynchronous and is based on byte frame (data frame). Each data frame passed between the master node and slave node is a serial data stream of 11 bits.

Data frame format:

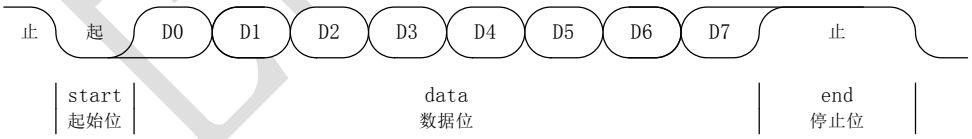
Start bit	1 bit
Data bits	8 bits (low bit in the front, high bit in the back)
Parity check bit	None: parity check bit is not adopted in this protocol;
End bit	1 bit (namely the real parity check bit is mandatory for high level)

Reference:

Sequence chart with check bit:



Sequence chart without check bit:



3.3 Data transmission rate

Default Baud rate uses 9600bps

4 Physical layer communication mode

4.1 Basic process

When the slave node is electrified or reset in stable operation, it can respond to the reading and writing commands of the master node. When the relevant command is received by the slave node, the necessary information will return to the master node in normal conditions, and it will return to the error code corresponding to the specific error type in abnormal conditions.

5 Application layer command type and format

When the communication command is sent to the instrument, the device with the corresponding address code will receive the communication command and read the information, and if there is no error, it performs the corresponding task; The execution result is then returned to the sender. The returned information includes the address code, the function code of the execution action, the data after execution and the error check code (CRC). If there is an error, no information will be sent.

5.1 nformation frame format

START	ADDR	CMD	DATA	CRC	END
Start	Address code	Function code	Data	Error check	End
Delay (time for >=3.5 characters)	1 byte 8 bits	1 byte 8 bits	N byte N X 8 bits	2 bytes 16 bits	Delay (time for >=3.5 characters)

Note 1: maximum frame length is not greater than 255 bytes;

Note 2: CRC check code low byte is in the front, and high byte is in the back;

5.2 Function code

The function code (CMD) is the second data frame in the information frame sent in

each communication. Modbus communication protocol defines the function code as 1～127(01H～7FH). This protocol utilizes some of the function codes. It is sent as the master node request, telling the slave node what action to perform through the function code. As the slave node response, the function code sent by the slave node is the same as the function code sent by the master node, indicating that the slave node has responded to the master node for operation. If the highest bit of the function code sent by the slave node is 1 (function code > 127), it indicates that the slave node has no response or there is an error.

Command code	Implication	Notes
0x03	Read command	Support single and multiple registers to read continuously
0x10	Write multiple register commands	Support continuous write action for multiple registers
0x06	Write single register command	Support continuous write action for single register

5.3Read command format

Note 3: MSB indicates high byte; LSB represents low byte.

Note 4: each register stores two bytes; The data of one byte for the register data type is required to store low byte (LSB).

Master node sending frame format:

Serial number	0	1	2	3	4	5	6	7
Field definition	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
Explanation	Controller address	Command type	Register start address		Register number n		CRC check	

Slave node normal reply frame format:

Serial number	0	1	2	3	4	5	6	…	L+1	L+2	L+3	L+4
Field definition	ADDR	CMD	Length	MSB	LSB	MSB	LSB	…	MSB	LSB	LSB	MSB
Explanation	Controller address	Command type	Number of bytes sent L=n*2	The value of the first register		The value of the second register		…	The value of the last register		CRC check	

Slave node abnormal reply format:

Serial number	0	1	2	3	4

Field definition	ADDR	CMD + 128	ErrCode	LSB	MSB
Explanation	Controller address	Command type +128	Error Code	CRC check	

5.4 Write single register command format

Master node sending frame format:

Serial number	0	1	2	3	4	5	6	7
Field definition	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
Explanation	Controller address	Command type	Register address		Data		CRC check	

Slave node normal reply frame format:

Serial number	0	1	2	3	4	5	6	7
Field definition	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
Explanation	Controller address	Command type	Register address		Data		CRC check	

Slave node abnormal reply format:

Serial number	0	1	2	3	4
Field definition	ADDR	CMD +128	ErrCode	LSB	MSB
Explanation	Controller address	Command type +128	Error Code	CRC check	

5.5Write multiple register command format

Master node sending frame format:

Serial number	0	1	2	3	4	5	6	7	8	9	10	…	L+5	L+6	L+7	L+8
Field definition	ADDR	CMD	MSB	LSB	MSB	LSB	Length	MSB	LSB	MSB	LSB	…	MSB	LSB	LSB	MSB
Explanation	Controller read	Command type	Register address		Register number n		Number of bytes sent L = n*2	The value of the first register		The value of the second register		…	The value of the last register		CRC check	

dre ss									

Slave node normal reply frame format:

Serial number	0	1	2	3	4	5	6	7
Field definition	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
Explanation	Controller address	Command type	Start Register address		Register number		CRC check	

Slave node abnormal reply format:

Serial number	0	1	2	3	4
Field definition	ADDR	CMD +128	ErrCode	LSB	MSB
Explanation	Controller address	Command type +128	Error Code	CRC check	

Note: the CRC check range is to check for all bytes before the CRC field.

5.6 Error code definition

When the slave node device is sending request to the master node device, the slave node expects a normal response. In the slave and master node inquiry, one of the following four possible events occurs:

1) If the slave node device receives a request without communication error and can normally process the inquiry, then the slave node device will return a normal response.

2) If the request is not received by the slave node due to a communication error, the response cannot be returned. The master node program will eventually process the timeout status of the request;

3) If the slave node receives a request, but detects a communication error (parity check, LRC, CRC, ...), then the response cannot be returned. The master node program will eventually process the timeout status of the request;

4) If the slave node receives the request without communication error, but it can't process this request (for example, if it is requested to read a nonexistent output or register), the slave node will return an abnormal response and notify the essential characteristic of error to the user;

The abnormal response message has two domains that are different from normal responses:

Function code domain: In normal response, the slave node uses the response function code domain to answer the function code of the initial request. The highest effective bits (MSB) of all functional codes are 0 (all of these values are below 128). In the abnormal response, the MSB of the slave node function code is 1. This enables the function code value in the abnormal response to be higher than the function code value in the normal response.

By setting the MSB of the function code, the application of the master node can recognize the abnormal response, and can detect the data domain of the abnormal code.

Data domain: In the normal response, the slave node can return the data or statistics table in data domain (any message required in the request). In the abnormal response, the slave node can return the abnormal code in the data domain. This defines the slave node state that produces an anomaly.

MODBUS abnormal code:

Name of abnormal code	Instructions
01 Illegal function code	For the slave node, the function code received in the query is an unallowable operation. This may be because the function code is only applicable to the new devices and is not implemented in the selected cell. It is also pointed out that the slave node can process such a request in the error state, for example, because it is unconfigured, it is required to return the register value.
02 Illegal data address	For the slave node, the data address received in the query is an unallowable address. In particular, the combination of reference number and transmission length is invalid. For the controller with 100 registers, the request with offset 96 and length 4 will be successful, while the request with offset 96 and length 5 will produce the abnormal code 02.

03 Illegal data value	For the slave node, the values included in the query are unallowable values. This value indicates the failure of the combination request remaining structure, for example: the implicit length is incorrect. This does not mean that, because the MODBUS protocol does not know the significance of any particular value of any special register, the data item that is submitted for storage in the register has a value that is beyond the expectation of the application.

6 CRC check algorithm

6.1 CRC algorithm

```
unsigned short count_CRC(unsigned char *addr, int num)
{
    unsigned short CRC = 0xFFFF;
    int i;
    while (num--)
    {
        CRC ^= *addr++;
        for (i = 0; i < 8; i++)
        {
            CRC=(CRC & 0x0001) ? ((CRC>>1) ^0xA001):(CRC>>1);
        }
    }
    return CRC；
}
```

7 Register List

The protocol is normal modbus communication protocol：

1、Each register deposit two bytes.

2、Data transmission mode: high byte in the front, and the low byte.

NO.	Item	unit	Register Address	Attribute (Read/Write)	Ratio	Remark
Version Information						
1	Software version		0x0000	Read Only	x1	
Running Status (0:Stop,1:Running, 2:Invalid)						
1	Unit running status		0x0100	Read Only	x1	
2	Internal fan status		0x0101	Read Only	x1	
3	External fan status		0x0102	Read Only	x1	
4	Compressor status		0x0103	Read Only	x1	
5	Heater status		0x0104	Read Only	x1	
6	Emergency fan status		0x0105	Read Only	x1	
Sensor Status (The invalid value of Temp. is 2000. The invalid value of humidity is 120. The invalid value of humidity is 32767.)						
1	Evaporator Temp.	℃	0x0500	Read Only	x 10	
2	Outdoor Temp.	℃	0x0501	Read Only	x 10	
3	Condenser Temp.	℃	0x0502	Read Only	x 10	
4	Indoor Temp.	℃	0x0503	Read Only	x 10	
5	Humidity	%	0x0504	Read Only	x 1	
6	Discharge Temp.	℃	0x0505	Read Only	x 10	
7	AC running current	A	0x0506	Read Only	x 1000	
8	AC input voltage	V	0x0507	Read Only	x 1	
9	DC input voltage	V	0x0508	Read Only	x 10	
Alarm Status (Normal:0, Fault:1)						
1	High Temp. alarm		0x0600	Read Only	x1	
2	Low Temp. alarm		0x0601	Read Only	x1	
3	High humidity alarm		0x0602	Read Only	x1	

4	Low humidity alarm		0x0603	Read Only	x1	
5	Coil freeze protection		0x0604	Read Only	x1	
6	High exhaust Temp. alarm		0x0605	Read Only	x1	
7	Evaporator Temp. sensor failure		0x0606	Read Only	x1	
8	Outdoor Temp. sensor failure		0x0607	Read Only	x1	
9	Condenser Temp. sensor failure		0x0608	Read Only	x1	
10	Indoor Temp. sensor failure		0x0609	Read Only	x1	
11	Exhaust Temp. sensor failure		0x060A	Read Only	x1	
12	Humidity sensor failure		0x060B	Read Only	x1	
13	Internal fan failure alarm		0x060C	Read Only	x1	
14	External fan failure alarm		0x060D	Read Only	x1	
15	Compressor failure alarm		0x060E	Read Only	x1	
16	Heater failure alarm		0x060F	Read Only	x1	
17	Emergency fan failure alarm		0x0610	Read Only	x1	
18	HP. alarm		0x0611	Read Only	x1	
19	LP. alarm		0x0612	Read Only	x1	
20	Water alarm		0x0613	Read Only	x1	
21	Fire alarm		0x0614	Read Only	x1	
22	Gating alarm		0x0615	Read Only	x1	
23	HP. lock		0x0616	Read Only	x1	
24	LP. lock		0x0617	Read Only	x1	
25	High exhaust Temp. lock		0x0618	Read Only	x1	
26	AC over voltage alarm		0x0619	Read Only	x1	

27	AC under voltage alarm		0x061A	Read Only	x1	
28	AC power supply failure		0x061B	Read Only	x1	
29	Lose phase alarm		0x061C	Read Only	x1	
30	Freq.fault		0x061D	Read Only	x1	
31	Anti phase alarm		0x061E	Read Only	x1	
32	DC over voltage alarm		0x061F	Read Only	x1	
33	DC under voltage alarm		0x0620	Read Only	x1	
Parameter Setting						
1	Refrigeration stop point	℃	0x0700	Read/Write	x1	15~50℃
2	Refrigeration band	℃	0x0701	Read/Write	x1	1~10℃
3	Heating stop point	℃	0x0702	Read/Write	x1	-15~25℃
4	Heating band	℃	0x0703	Read/Write	x1	1~10℃
5	Reserve		0x0704			
6	Reserve		0x0705			
7	High Temp.point	℃	0x0706	Read/Write	x1	25~80℃
8	Low Temp. point	℃	0x0707	Read/Write	x1	-20~15℃
9	High humidity point	%	0x0708	Read/Write	x1	0~100%
10	Internal fan stop point	℃	0x070A	Read/Write	x1	-20~50℃
Remote Control Parameter						
1	Reserve		0x0800			
2	Remote control		0x0801	Read/Write		Open:1, Close:0

For example:

①Read the value of a single register-Software version:

Send command: 01 03 00 00 00 01 84 0A

Return instruction: 01 03 02 02 10 B8 E8

Parsing instructions:

01 ADDR

03 CMD

02 Length

02 10 Data

B8 E8 CRC

②Read the value of multiple registers-Coil temperature-outdoor temperature-Condensation temperature:

Send command: 01 03 05 01 00 03 54 C7

Return instruction: 01 03 06 01 1F 01 16 01 12 D4 C3

Parsing instructions:

01 ADDR

03 CMD

06 Length

01 1F Data:Coil temperature,Decimal 287→28.7℃

01 16 Data:outdoor temperature,Decimal 278→27.8℃

01 12 Data:Condensation temperature,Decimal 274→27.4℃

D4 C3 CRC

③Write Single Register-Refrigeration point 30℃:

Send command: 01 06 07 00 00 1E 08 B6

Return instruction: 01 06 07 00 00 1E 08 B6

Parsing instructions:

01 ADDR

06 CMD

07 00 Register address

00 1E Data: Decimal 30→30℃

08 B6 CRC

④write the value of multiple registers-High Temp set point40℃-Low Temp set point15℃

Send command: 01 10 07 06 00 02 04 00 28 00 0F 95 B9

Return instruction: 01 10 07 06 00 02 A0 BD

Parsing instructions:

01 ADDR

10 CMD

07 06 Start register address

00 02 Register number

A0 BD CRC