

# 温湿度数据采集模块

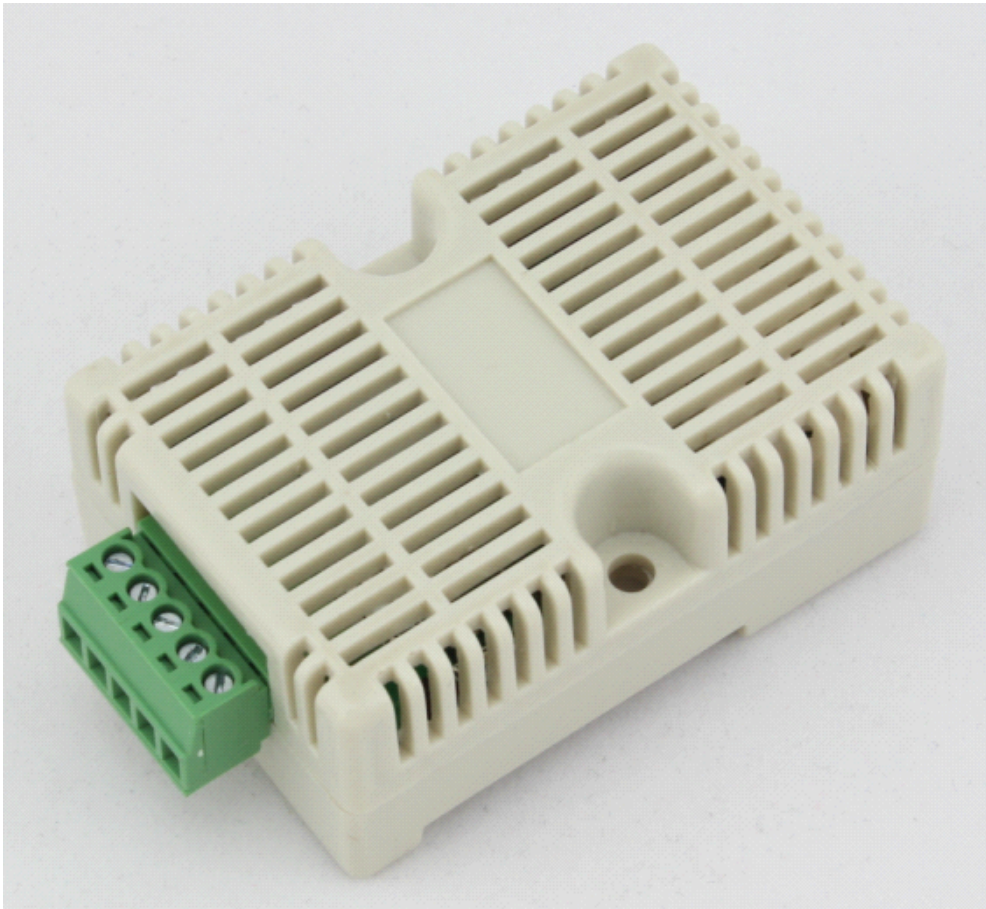
USR-SENS-WSD-T

USR-SENS-WSD-2

USR-SENS-WSD-4

说明书版本：V1.2

日期：2015-6-11



## 目录

温湿度数据采集模块.....	1
第一章 产品介绍.....	3
1.1 产品简介.....	3
1.2 功能特点.....	3
1.3 产品特性.....	3
1.4 接口定义.....	4
1.5 出厂设置.....	5
1.6 使用前需阅读的部分: .....	5
第二章 工作模式.....	6
2.1 系统框图.....	6
2.2 命令格式.....	6
2.2.1、MODBUS 命令帧.....	6
2.2.2、MODBUS 响应帧.....	7
2.2.3、MODBUS 帧举例.....	8
2.2.4、MODBUS 异常响应: .....	10
2.2.5、MODBUS 异常响应帧举例.....	11
2.3 简单 ASCII 码指令.....	12
2.3.1 READ.....	12
2.3.2 AUTO.....	12
2.3.3 STOP.....	13
附录一 CRC16 校验算法的 C 语言实现: .....	14
有人联系方式.....	15
免责声明.....	15
附录 版本历史.....	15

# 第一章 产品介绍

## 1.1 产品简介

USR-SENS-WSD 温湿度数据采集模块是一款高性能温湿度数据采集传输模块，可选的 RS485/RS232/TTL 不同电平输出，MODBUS 工业控制总线协议，可以多个模块共同接入总线组网，也可以与其他符合 MODBUS 协议的设备共同组网，实时、稳定地监控多个生产现场环境变化，以使用户准确把握不同应用现场的环境变化，快速做出反应，保障生产现场环境的稳定性。

如果在产品使用中需要技术支持，可以在“客户支持中心”提交问题，工程师会及时给您解答。

<http://h.usr.cn/index.php?c=frontTicket&m=sign>

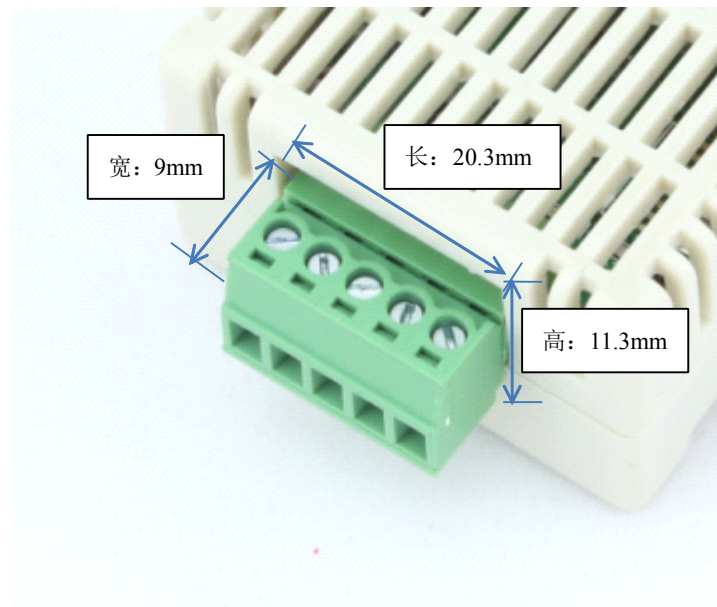
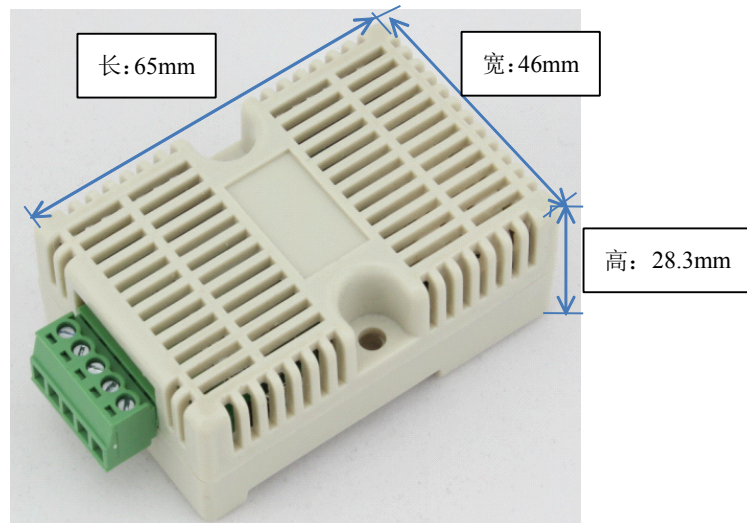
## 1.2 功能特点

- DATA/VCC/GND/RXD(RS85-A)/TXD(RS485-B)五线接口，简单可靠，便于扩展；
- 可选的 RS232/RS485/TTL 信号传输方式；
- 基于 MODBUS 工业控制总线协议的数据传输，性能可靠，兼容性好，易于组网；
- 支持简单字符指令，便于单个模块简易应用场合使用
- 体积小，易于安装；
- 针对农业应用环境做防潮处理，适用于大棚等潮湿环境；
- 串口参数配置范围：
  - 波特率：1200 / 4800 / 9600 / 19200 / 38400 / 57600 ；
  - 数据位：8 位，不可配置；
  - 校验：无校验 / 奇校验 / 偶校验；
  - 停止位：1 / 2 位；
  - 配置完成后断电并重新通电后配置生效；
  - 具体配置命令请参考“第二章 工作模式”的“2.2.3.7 设置串口参数”

## 1.3 产品特性

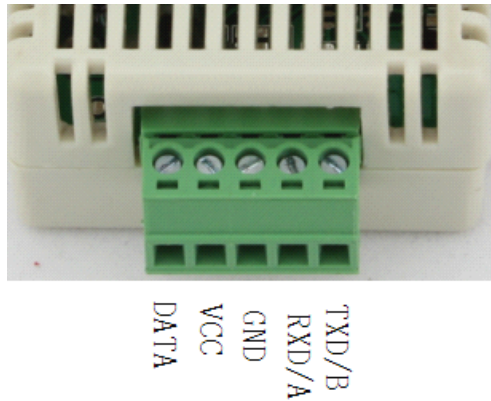
- 测量温度范围：-40℃~+80℃；分辨率：0.1℃；精度±0.2℃
- 工作温度范围：-40℃~+80℃；
- 测量湿度范围：0%~100%（RH）相对湿度；分辨率：0.1%RH；精度：环境温度 25℃ 下 ±2%RH；
- 工作湿度范围：0%~100%（RH）相对湿度；
- 电源输入：5V DC；
- 支持数据格式：串行 MODBUS 协议，简易 ASCII 码字符指令：“READ”，“AUTO”，“STOP”；
- 模块 ID 可配置，范围为：1~247；

■ 外形尺寸:



## 1.4 接口定义

■ 接口排列:



- DATA : 保留;
- VCC : 电源正, 5V DC;
- GND : 电源负, 信号地;
- RXD/A: -T 版本为串行 TTL RXD 信号线, -2 版本为 RS232 RXD 信号线, -4 版本为 RS485 A 端信号线;
- TXD/B: -T 版本为串行 TTL TXD 信号线, -2 版本为 RS232 TXD 信号线, -4 版本为 RS485 B 端信号线;

## 1.5 出厂设置

- 串行接口: 波特率 9600, 数据位 8 位, 停止位 1 位, 无校验位;
- 模块 ID (“0x\*\*”表示十六进制数, 下同): 0x11;

## 1.6 使用前需阅读的部分:

1、如果您希望正常使用本模块, 需要具有一定的 MODBUS 协议基础, 或者可以参照本说明书“第二章 工作模式”的“2.2 命令格式”部分中对命令格式的讲解, 并参照其中的举例来设计您所使用的命令, 并使用您根据本说明书设计的命令来编写上位机程序。本模块的数据传输部分采用的是串行 MODBUS 协议标准, 与之对应的, 我们为您提供基于串行 MODBUS 协议的技术支持。您可以使用网络转串口模块通过 TCP MODBUS 协议向模块发送指令并得到数据, 然而这方面的知识和应用方案我们暂未支持, 请您参考相关文辑或者资料。

另外, 为便于简易应用场合用户的使用, 本模块支持三条 ASCII 码指令, 用于读取当前环境温度湿度数据(注意, 当使用 ASCII 码指令时, 线路上只允许有一个模块, 如果有多个模块会导致数据传输混乱), 指令的具体格式及参数请参考“第二章 工作模式”的“2.3 简单 ASCII 码指令”。

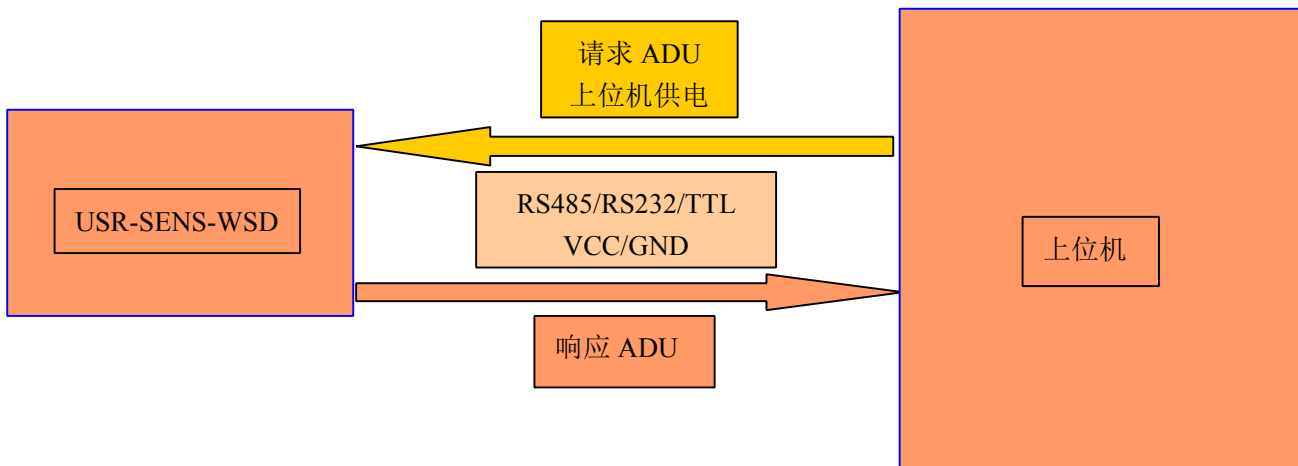
2、模块发送和接收的指令中校验码域是必须的, 本模块采用 CRC16 校验, 本模块在解析您所发送的指令时会执行 CRC16 校验并且校验正确才会响应您的指令, 这个机制保证了数据传输的准确性, 保障了您所构建的系统的健壮性。我们所采用的 CRC 校验算法的实现源代码在“附录二”中提供给您。

3、模块目前所接收的指令长度为 8 个字节 (8Byte), 暂不支持可变长度的指令, 虽然模块返回的指令可能会大于或者小于 8 个字节, 这是因为模块根据指令判断返回数据的多少来决定应该输出多少数据。

## 第二章 工作模式

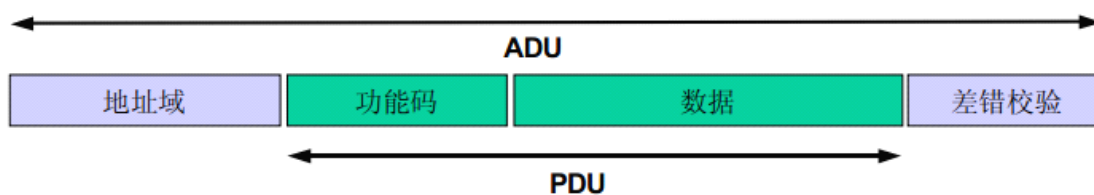
### 2.1 系统框图

USR-SENS-WSD 温湿度数据采集模块（如无特别说明，以下统称模块）是生产现场温度、湿度数据采集终端，通过接收上位机发送的 MODBUS 协议帧，对 MODBUS 帧进行解析，并根据指令返回数据，该模块采用 5 线接口（其中 DATA 线保留），同时需要上位机对其进行供电。结构框图如下：



### 2.2 命令格式

模块支持的 MODBUS 帧结构如下所示：



#### 2.2.1、MODBUS 命令帧

模块上电后约 2S 内执行初始化操作，之后可以正常接收上位机 MODBUS 帧并根据上位机指令执行操作。

MODBUS 命令帧包括：1 Byte 地址域（指示响应上位机命令的模块地址），1Byte 功能码（指示符合指定地址的模块作何响应），4Bytes 数据（指示 读取/写入 的起始地址、结束地址 或 用于写入的数据）以及 2Bytes CRC（用于差错校验）；

命令帧地址域取值为“0x00~0xF7”，取值为 0 时，表示广播功能，这时所有支持 MODBUS 协议的设备均按照指令执行操作，但不返回数据；

支持的功能码取值为



“0x03”: 读取环境温湿度数据 (可选读取 湿度/温度 或湿度&温度);

“0x04”: 读取环境温湿度数据 (可选读取 湿度/温度 或湿度&温度);

“0x06”: 设置模块 ID, 设置模块串口参数;

特别的, 当上位机发送广播命令时, 模块支持一个自定义功能码“0x6E”, 该功能码用于使模块重置设备 ID 为出厂默认, 即“0x11”, 定义该功能码的用意是当用户遗忘或者不确定模块 ID 时, 可以使用该功能码通过广播方式来使模块恢复出厂设置;

数据部分 4Bytes 数据, 在读取环境温湿度数据时, 前 2Bytes 表示读取起始地址, 高字节在前, 低字节在后 (“0x00 0x00” 表示从第一个数据开始读取, “0x00 0x01” 表示从第二个数据开始读取); 后 2Byte 表示读取数据的个数, 高字节在前, 低字节在后 (“0x00 0x01” 表示读取一个数据, “0x00 0x02” 表示读取两个数据);

差错校验采用 CRC16 校验, 2Bytes, 低字节在前, 高字节在后, 为 MODBUS 协议所推荐的校验方法。

## 2.2.2、MODBUS 响应帧

MODBUS 响应帧包括: 1 Byte 地址域 (向上位机返回模块 ID), 1Byte 功能码/异常码 (返回模块所接收到的功能码/返回差错码), 2Bytes~5Bytes 数据 (1Byte 表示返回字节数, 1Byte~4Bytes 可选的数据) 以及 2Bytes CRC 校验码 (用于差错校验);

响应帧地址域=命令帧地址域, 模块在接收到上位机命令帧后提取地址域和 CRC 校验码与自己 ID 和计算所得的 CRC 校验码比较, 如命令帧目的 ID 为模块自身 ID 且 CRC 校验正确, 则模块读取传感器数据并发送响应帧, 如命令帧目的 ID 与模块自身 ID 不相符或 CRC 校验错误, 则不响应命令帧;

响应功能码域: 支持的功能码取值为

“0x03”: 读取环境温湿度数据 (可选读取 湿度/温度 或湿度&温度);

“0x04”: 读取环境温湿度数据 (可选读取 湿度/温度 或湿度&温度);

“0x06”: 设置模块 ID, 设置串口参数;

“0x6E”: 重置模块 ID;

异常码返回: 模块向上位机返回异常码分为两部分, 各 1Byte, 第一部分占用功能码域, 在接收到的功能码基础上加 0x80 标示具体是哪个功能码出错, 第二部分指示出错的具体原因;

数据部分 2Bytes~5Bytes 数据, 在返回数据时, 第一个字节表示之后返回数据区字节数, 后 1~4Bytes 表示返回的数据, 高字节在前, 低字节在后; 返回温湿度数据时, 湿度数据在前, 温度数据在后; 高字节在前, 低字节在后; 温度最高位为 0 时为正温度, 温度最高位为 1 时为负温度; 返回数据均为实际测量数据的 10 倍, 例如: 返回温度数据是 185, 则实际温度为 18.5℃;

差错校验采用 CRC16 校验, 2Bytes, 低字节在前, 高字节在后, 为 MODBUS 协议所推荐的校验方法。

## 2.2.3、MODBUS 帧举例

以下列举了读取模块数据的 MODBUS 命令帧格式及响应帧格式，其中命令帧的 ID 是不固定的，举例中采用出厂默认 ID 进行设置，当用户重新配置 ID 后，所有的 MODBUS 帧 ID 都应该是配置 ID 后的新 ID；响应帧数据是不固定的，但是格式是确定的。

### 2.2.3.1、读湿度数据:

命令帧:	0x11	0x04	0x00	0x00	0x00	0x01	0x33	0x5A
	ID	功能码	起始地址		读取数据个数		CRC 校验码	
响应帧:	0x11	0x04	0x02		0x01	0xC3	0x39	0x32
	ID	响应功能码	返回字节数		返回湿度数据		CRC 校验码	

返回湿度数据为 0x01C3，换算为十进制为 451，表示当前相对湿度为 45.1%。

### 2.2.3.2、读温度数据:

命令帧:	0x11	0x04	0x00	0x01	0x00	0x01	0x62	0x9A
	ID	功能码	起始地址		读取数据个数		CRC 校验码	
响应帧:	0x11	0x04	0x02		0x00	0xEE	0xF8	0xBF
	ID	响应功能码	返回字节数		返回湿度数据		CRC 校验码	

返回温度数据为 0x00EE，换算为十进制为 238，表示当前温度为 23.8℃。

### 2.2.3.3、读湿度、温度数据:

命令帧:	0x 11	0x04	0x00	0x00	0x00	0x02	0x73	0x5B
	ID	功能码	起始地址		读取数据个数		CRC 校验码	
响应帧:	0x11	0x04	0x04		0x01	0xC8	0x00	0xED 0xAA 0x0A
	ID	响应功能码	返回字节数		湿度数据		温度数据	CRC 校验码

返回湿度数据为 0x01C8，换算为十进制为 456，表示当前相对湿度为 45.6%；返回温度数据为 0x00ED，换算为十进制为 237，表示当前温度为 23.7℃。

### 2.2.3.4、设置模块 ID:

模块 ID 为模块内部地址为 0x0000 的寄存器变量，可以使用功能码“0x06(写单个寄存器)来进行配置。

命令帧:	0x11	0x06	0x00	0x00	0x00	0x02	0x0A	0x9B
	ID	功能码	地址		配置 ID		CRC 校验码	
响应帧:	0x02	0x06	0x00	0x00	0x00	0x02	0x08	0x38



ID	响应功能码	返回地址	返回数据	CRC 校验码
----	-------	------	------	---------

### 2.2.3.5、重置模块 ID:

命令帧: 0x00	0x6E	0x00	0x00	0x00	0x00	0xE9	0x D2
ID	功能码	地址	配置 ID			CRC 校验码	

该命令是一条指定命令, 当模块接收到该命令后会重置模块 ID 为“0x11”, 因为该命令采用广播模式, 为避免与系统其它设备的命令有所冲突, 所以采用了一个自定义的功能码“0x6E”, 并且必须当功能码后面的四个字节数据全为 0 时, 模块才执行重置 ID 的动作。

### 2.2.3.6、设置串口参数

串口参数为模块内部地址为 0x0001, 可使用功能码“0x06(写单个寄存器)”来进行配置。其中, 地址 0x0001 高字节用于存储波特率参数, 低字节用于配置校验位/停止位。配置格式如下:

波特率 参数:	0	1200
	1	4800
	2	9600
	3	19200
	4	38400
	5	57600

校验位/停止位 参数:

位 7: 位 6	空	高位
位 5: 位 4	STOP : 停止位, 用来设置停止位的位数 00: 1 个停止位; 10: 2 个停止位;	
位 3	空	
位 2	奇偶校验控制使能 0 禁用/1 使能	
位 1	奇偶校验选择 0 偶校验/1 奇校验	
位 0	空	低位

**注: 当停止位设置为 2 位时, 校验位只能配置为 0; 当校验位为奇校验/偶校验时, 停止位只能设置为 1 位; 用简单 ASCII 指令时, 对串口的参数设置同样必须符合该规则。**

命令帧: 0x11	0x06	0x00	0x01	0x02	0x04	0xDA	0x39
ID	功能码	地址	波特率	校验位/停止位	CRC 校验码		
响应帧: 0x11	0x06	0x00	0x01	0x02	0x04	0xDA	0x39
ID	响应功能码	返回地址	返回数据			CRC 校验码	

该命令设置模块波特率为 9600, 停止位 1 位, 偶校验。

## 2.2.4、MODBUS 异常响应:

当上位机设备向模块发送请求时, 上位机希望一个正常响应。从主站询问中出现下列四种可能事件之一:

- 1、 如果模块接收到无通信错误的请求, 并且可以正常地处理询问, 那么模块将返回一个正常响应 (一个正常 MODBUS 帧);
- 2、 如果由于通信错误, 模块没有接收到请求, 那么不能返回响应。上位机程序将最终处理请求的超时状态;
- 3、 如果模块接收到请求, 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。上位机程序将最终处理请求的超时状态;
- 4、 如果模块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), 模块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, 模块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 (它们的值都低于十六进制 80)。在异常响应中, 模块设置功能码的 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 0x80。

**数据域:** 通过设置功能码的 MSB, 上位机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。在正常响应中, 模块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, 模块返回数据域中的异常码。这就定义了产生异常的模块状态。

客户机请求和服务器异常响应的实例:

请求		响应	
域名	(十六进制)	域名	(十六进制)
功能	01	功能	81
起始地址 Hi	04	异常码	02
起始地址 Lo	A1		
输出数量 Hi	00		
输出数量 Lo	01		

在这个实例中, 客户机对服务器设备寻址请求。功能码(01)用于读输出状态操作。它将请求地址 0x04A1 的输出状态。值得注意的是, 像输出域(0001)号码说明的那样, 只读出一个输出。

如果在服务器设备中不存在输出地址, 那么服务器将返回异常码(02)的异常响应。这就说明从站的非法数据地址。

以下是模块支持的 MDOBUS 异常码表:

MODBUS异常码		
代码	名称	含义
0x01	非法功能	对模块来说, 询问中接收到的功能码是不可允许的操作。同时, 它还指出上位机在错误的状态下发送请求, 例如: 因为它是未配置的功能, 并且要求返回数据
0x02	非法数据地址	对模块来说, 询问中接收到的数据地址是不可允许的地址, 特别是, 起始地址和数据个数的组合是无效的。对于模块来说, 起始地址0x0001而请求数据个数是0x0004显然是无效的, 将产生异常码0x02
0x03	非法数据值	对于模块来说, 询问中包括的值是不被允许的。这个值指示了组合请求中剩余结构的故障。例如: MODBUS协议中地址0x00用于广播, 如果试图设置模块ID为0x00则返回该异常码
0x0C	模块故障	该异常码不是MODBUS协议规定的标准异常码, 这是一个自定义的异常码, 当模块向上位机返回“0x0C”时, 产生该异常码的原因需根据上位机所发送的指令判断, 当上位机指令为设置模块ID时, 该异常码表示无法正常设置ID, 模块需要重启以设置ID; 当上位机指令为读取温湿度数据时, 该异常码表示无法读取传感器数据, 此时需要上位机延时几秒(应该大于2秒)再发送指令, 如果仍然返回该异常码, 表示模块上的传感器损坏, 需要维修

## 2.2.5、MODBUS 异常响应帧举例

### 2.2.5.1、非法功能命令:

```

命令帧: 0x 11   0x01           0x00   0x00   0x00   0x02           0xBF   0x5B
          ID     功能码         起始地址   读取数据个数   CRC 校验码
响应帧: 0x11   0x81           0x01           0x80  0x55
          ID     响应功能码     异常码       CRC 校验码
    
```

该命令帧指示模块读取首地址 0x0000 起两个线圈(离散量)的值, 因为模块内无离散量, 所以没有支持该功能码(0x01)。模块返回的响应帧为一帧异常帧, 响应功能码在接收到的功能码的基础上加 0x80, 返回的响应功能码为 0x81。由于功能码 0x01 是不受支持的, 所以异常码域为 0x01, 之后为差错校验域。

### 2.2.5.2、非法数据地址:

```

命令帧: 0x 11   0x04           0x00   0x01   0x00   0x02           0x22   0x9B
          ID     功能码         起始地址   读取数据个数   CRC 校验码
响应帧: 0x11   0x84           0x02           0xC3  0x04
          ID     响应功能码     异常码       CRC 校验码
    
```

该命令帧指示模块读取首地址 0x0001 起两个寄存器的值, 因为模块内只有两个寄存器, 地址分别为

0x0000(湿度数据)、0x0001(温度数据), 所以该命令帧所请求的起始地址和数据个数的组合是无效的, 模块将返回如上所示的异常响应帧。响应帧功能码域的数据为接收到的功能码的基础上加 0x80, 返回的响应功能码为 0x84。由于命令帧请求的数据超出范围, 所以异常码域为 0x02, 之后为差错校验。

### 2.2.5.3、非法数据值:

命令帧:	0x 11	0x06	0x00	0x00	0x00	0x00	0x8B	0x5A
	ID	功能码	设置地址	设置 ID			CRC 校验码	
响应帧:	0x11	0x86	0x0C	0x03	0xA4			
	ID	响应功能码	异常码	CRC 校验码				

该命令帧指示模块设置 ID 为 0x00, 然而 0x00 在 MODBUS 协议中是用于广播的, 所以该命令帧所要求写入的数据是非法的, 模块将返回一个异常响应帧。响应帧的功能码域的数据为接收到的功能码基础上加 0x80, 返回的响应功能码为 0x86, 由于命令帧请求的数据非法的, 所以异常码域为 0x03, 之后为差错校验。

### 2.2.5.4、模块故障:

命令帧:	0x 11	0x04	0x00	0x00	0x00	0x02	0x73	0x5B
	ID	功能码	起始地址	读取数据个数			CRC 校验码	
响应帧:	0x11	0x84	0x0C	0x42	0xC0			
	ID	响应功能码	异常码	CRC 校验码				

该命令帧请求模块返回湿度、温度数据, 如果模块传感器出现故障, 则无法正确获得数据, 此时模块将返回一个异常响应帧。响应帧的功能码域的数据为接收到的功能码基础上加 0x80, 返回的响应功能码为 0x84, 由于模块本身工作在异常状态, 所以异常码域为 0x0C, 之后为差错校验。

## 2.3 简单 ASCII 码指令

简单 ASCII 码指令由四个 ASCII 码字符组成, 数据传输均为 ASCII 码。

### 2.3.1 READ

该指令用于单次读取环境温湿度数据

发送指令格式: READ

返回数据格式: 29.8,56.5%

### 2.3.2 AUTO

该指令用于以约 2 秒为周期返回当前环境温湿度数据

发送指令格式: AUTO

返回数据格式: 29.8,56.5%

### 2.3.3STOP

该指令用于停止返回环境温湿度数据

发送指令格式：STOP

返回数据格式：无

## 附录一 CRC16 校验算法的 C 语言实现:

```
const unsigned int ccitt_table[256] =
    {0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241, 0xC601, 0x06C0,
    0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440, 0xCC01, 0x0CC0, 0x0D80, 0xCD41,
    0x0F00, 0xCF41, 0xCE81, 0x0E40, 0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0,
    0x0880, 0xC841, 0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
    0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41, 0x1400,
    0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641, 0xD201, 0x12C0, 0x1380,
    0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040, 0xF001, 0x30C0, 0x3180, 0xF141, 0x3300,
    0xF3C1, 0xF281, 0x3240, 0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0,
    0x3480, 0xF441, 0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
    0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840, 0x2800, 0xE8C1,
    0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41, 0xEE01, 0x2EC0, 0x2F80, 0xEF41,
    0x2D00, 0xEDC1, 0xEC81, 0x2C40, 0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1,
    0xE681, 0x2640, 0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
    0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240, 0x6600, 0xA6C1,
    0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441, 0x6C00, 0xACC1, 0xAD81, 0x6D40,
    0xAF01, 0x6FC0, 0x6E80, 0xAE41, 0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1,
    0xA881, 0x6840, 0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
    0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40, 0xB401, 0x74C0,
    0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640, 0x7200, 0xB2C1, 0xB381, 0x7340,
    0xB101, 0x71C0, 0x7080, 0xB041, 0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0,
    0x5280, 0x9241, 0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
    0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40, 0x5A00, 0x9AC1,
    0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841, 0x8801, 0x48C0, 0x4980, 0x8941,
    0x4B00, 0x8BC1, 0x8A81, 0x4A40, 0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0,
    0x4C80, 0x8C41, 0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
    0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040, };

unsigned int cal_crc16(unsigned char *ptr, unsigned char len)
{
    unsigned int crc = 0xffff;
    while (len-- > 0)
        crc = ccitt_table[(crc ^ *ptr++) & 0xff] ^ (crc >> 8);
    return crc;
}
```



# 有人联系方式

---

公 司: 济南有人物联网技术有限公司

地 址: 济南市高新区会展国际城北塔 1-724~729 室

网 址: <http://www.usr.cn>

客户支持中心: <http://h.usr.cn>

邮 箱: sales@usr.cn

企 业 QQ: 8000 25565

电 话: 4000-255-652 或者 0531-88826739

有人愿景: 国内联网通讯第一品牌

公司文化: 有人在认真做事!

产品理念: 简单 可靠 价格合理

有人信条: 天道酬勤 厚德载物 共同成长

---

# 免责声明

本文档提供有关 USR-WM1 产品的信息, 本文档未授予任何知识产权的许可, 并未以明示或暗示, 或以禁止发言或其它方式授予任何知识产权许可。除在其产品的销售条款和条件声明的责任之外, 我公司概不承担任何其它责任。并且, 我公司对本产品的销售和/或使用不作任何明示或暗示的担保, 包括对产品的特定用途适用性, 适销性或对任何专利权, 版权或其它知识产权的侵权责任等均不作担保。本公司可能随时对产品规格及产品描述做出修改, 恕不另行通知。

# 附录 版本历史

V1.0 2014-06-23 第一版成立

V1.1 2014-07-08 更新产品特点

V1.2 2015-06-11 增加工单精灵操作系统简介、链接, 更改部分文字说明