

HDR30-S01-485 数据采集器

使用配置手册

北京云天创科技有限公司

内部讨论版-V1.0

目录

一、采集器简介.....	2
1.1 特点.....	2
1.2 结构尺寸及技术规格.....	2
1.3 技术规格.....	4
二、采集器使用流程及相关指令说明.....	5
2.1、采集器使用流程.....	5
2.2、相关指令说明.....	6
2.2.1 采集器可操作 Modbus 寄存器定义	6
2.2.2 采集器可操作 Modbus 指令	8
2.2.3 采集器可操作 Modbus 寄存器操作注意事项	17
2.2.4 其他指令说明.....	19
三、采集器安装拓扑结构图.....	20
四、附录.....	21
4.1 传感器数据结构及计算公式.....	21
4.1.1 采集器下发传感器数据格式.....	21
4.1.2 温湿度传感器计算公式.....	21
4.2 CRC16 算法.....	21

一、采集器简介

HDR30-S01-485型采集器是针对工业现场及仓库等室内现场专门设计的采集器。

1.1 特点

- 集成 modbus 协议
- 隔离 485 通信接口
- 安装方便，布点简单，无需复杂设置
- 直流 5V 供电，配套直流电源适配器
- 可进行大规模组网

1.2 结构尺寸及技术规格

(1) 内部结构接口说明

接口说明如图所示：

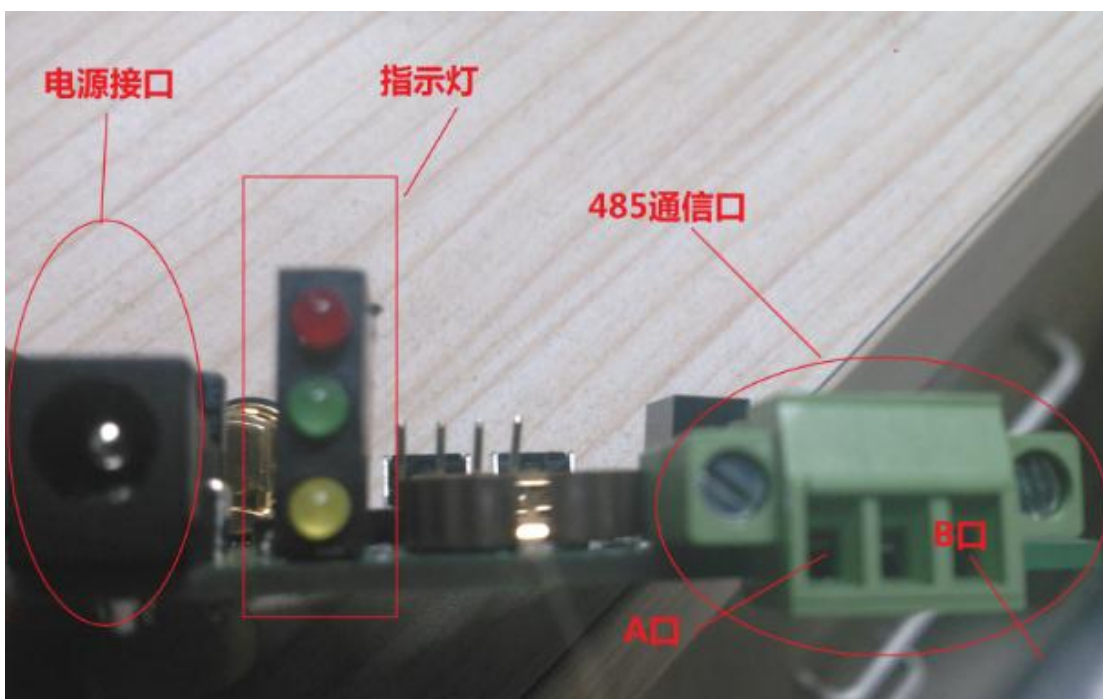


图 1 采集器后面板接口图

- 电源接口输入电压为 5V—12V，标准配置为 5V 适配器
- 指示灯：红灯：有串口数据或无线数据闪烁
绿灯：采集器进入扫频功能闪烁
进入对码状态长亮
黄灯：电源正常，系统初始化完成点亮
- 485 通信接口：用于 485 通信（注意 A、B 线序）



图 2 采集器前面板接口图

- 拨码开关：
 - 1)、拨码开关 1 OFF 位置为 485 接口不选择 120 欧姆的匹配电阻；处于 ON 选择 120 欧姆的匹配电阻接入 485 接口（默认为 OFF）
 - 2)、拨码开关 1 OFF 位置为 485 接口波特率为 9600，处于 ON 位置波特率为 4800（默认为 OFF）。

注意：重新选择好波特率必须重新上电或复位采集器才能生效
- 面板按键：1 按键为采集器复位按键。
2 按键为设置采集器进入配对模式，长按 3 秒绿灯点亮进入配对模式。
- 射频连接头：连接天线

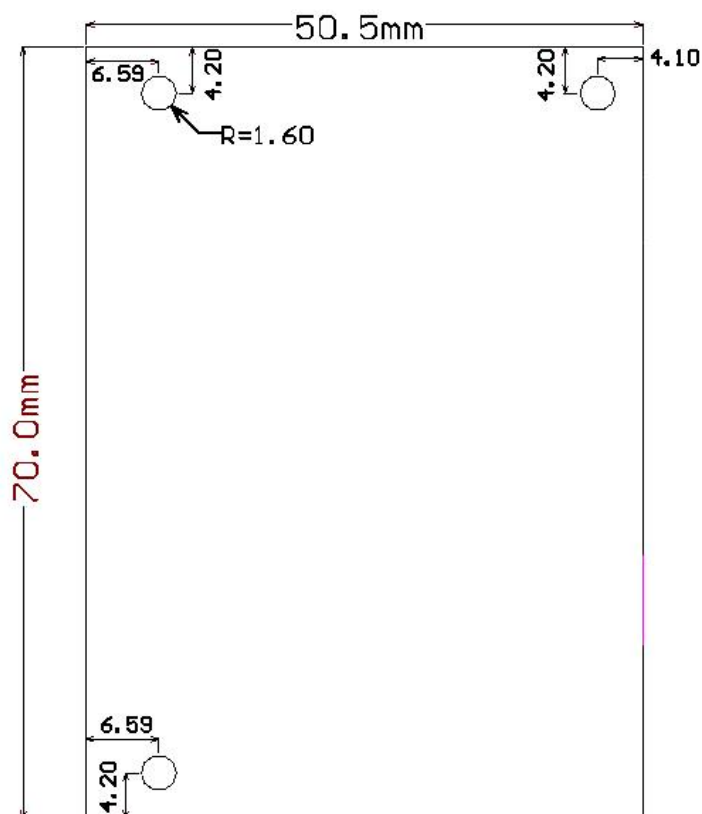


图 3 内部结构尺寸图

1.3 技术规格

(一) 射频指标

769M~926MHz

参数名称	数值范围	单位	说明
工作频率范围	769~936	MHz	步进 1M 或 0.1MHz
频段数量	4	个	中国 WPAN 标准
	1	个	欧盟 SRD 标准
	10	个	北美 ISM 标准
频段间隔	2	MHz	中国 WPAN 标准
发射功率	-11~+10	dBm	可进行 16 步调整
接收灵敏度	-110	dBm	PER = 1%
无线速率	20~1000	kbps	
射频接口阻抗	50	Ohm	单端非平衡输出

(二) 工作特性指标

769M~926MHz 频段

参数名称	数值范围	单位	说明
工作电源电压 VCC	5~12	V	适配器输入工作电压
485 总线波特率	9600/4800	bps	通过拨码开关指定

(三) 结构指标

参数名称	数值范围	单位	说明
PCB 板外型尺寸	70*50.5*1.5	mm	
外壳外形尺寸		mm	
工作温度范围	-40~85	°C	
工作湿度范围	0~95%		非凝结

二、采集器使用流程及相关指令说明

2.1、采集器使用流程

第一步：将采集器的通信接口接上 485 通信线缆，电源接口接上配送的适配器，采集器网络指示灯黄灯点亮。

采集器默认参数为：

- Modbus 地址：01
- 采集器网络参数：
MAC 地址与 PANID 参数见所贴标签
正常工作信道：10
对码信道：10
- 采集器 485 通信口参数
波特率：9600（见拨码开关说明）
120 欧匹配电阻不接入 485 通信口（见拨码开关说明）

将 485 通信线缆接到 PC 机上，设置采集器的 Modbus 地址参数。

(1) 方法一：通过串口调试助手设置：

发送命令如下：（16 进制发送）

01 06 00 13 xx xx CRC16(高 16 位在前)

或 00 06 00 13 xx xx CRC16(高 16 位在前)

说明：(1)、00 为广播地址

(2)、xx xx 为设置的 Modbus 地址范围为（0001 至 00FA）

(3)、指令具体格式见（二）指令说明

设置成功返回值如下：（16 进制）

01 06 00 13 xx xx CRC16(高 16 位在前)

(2)、方法二：通过单机采集器配置软件设置（具体使用方法见单机版软件说明）

第二步：设置好采集器的 Modbus 地址，设置采集器网络参数。

(1)、方法一：通过串口调试助手发送采集器扫频命令：

命令如下：（16 进制发送）

AA 80 E3 91 FC EA FC EA 00 EE DD XOR 其中 XOR 为异或校验

采集器开始扫频操作返回命令为：

AA 80 E3 92 00 EE DD XOR 其中 XOR 为异或校验

此时工作指示灯绿灯开始闪烁，在扫频期间绿灯会闪烁，直至扫频测试完成，采集器 485 通信口会输出一组扫频到的空间频点参数（具体格式见（二）指令说明）。选择其中一个频点作为工作频点，并按照（二）指令说明格式设置。

注意：

- (1) 一般情况只要设置正常工作频点。若配对频点 10 有干扰可以更改配对频点，更改配置频点后需要更改与之组网传感器节点配对频点
- (2)、尽量保持相邻采集器的正常工作频点不同

- (2)、方法二：通过单机采集器配置软件设置（具体使用方法见单机版软件说明）

第三步：设置好采集器网络参数，进行采集器与传感器节点绑定操作

具体如下：

- (1)、按住采集器对码键 3 秒，工作指示灯绿灯点亮，采集器进入绑定配对模式。
- (2)、操作传感器节点进入配对模式与采集器绑定，具体操作见传感器操作说明。
- (3)、若无传感器节点加入网络，采集器将在 90 秒后退出配对模式，工作指示灯绿灯熄灭，进入正常工作模式；若有节点加入采集器将重新计时，直到没有节点加入 90 秒超时退出。

第四步：将组好网络的传感器节点布置在合适位置

要求如下：（信号强度值必须大于 16，详细操作见传感器操作说明），采集器组成 485 通信网络，建议布置好 485 网络后需要在 PC 机端发送查询采集器 Modbus 地址指令确保此采集器已经在网络中。

2.2、相关指令说明

Modbus 格式指令说明

2.2.1 采集器可操作 Modbus 寄存器定义

地址	寄存器名称	寄存器说明	读写	取值范围
采集器网络参数设置寄存器				
10	PANID	无线通信网络地址	RW	0x0001~0xFEFE
11	MAC	无线通信节点地址	RW	0xF001~0xFEFE
12	CHNL (NORMAL/PAIRUP)	通信信道(正常/配对模式)	RW	10~250
13	ModAddr	ModBUS 设备地址	RW	1~250
采集器内部功能参数设置寄存器				
20	UTC_NOW_H	当前 UTC 时间高半字	RW	0x0000~0xFFFF
21	UTC_NOW_L	当前 UTC 时间低半字	RW	0x0000~0xFFFF
22	RfScanEnable	自动空间扫描功能使能(待定)	RW	0x0000/0xEABE
23	RfBindEnable	无线绑定功能使能(待定)	RW	0x0000/0xEABE
24	ConfigSave	配置信息写入使能(待定)	W	0x0000/0xCFAE
27	NewDataClearH	传感器最新数据清除使能	W	0x0000/0xFCEA
28	NewDataClearL	传感器最新数据清除使能	W	0x0000/0xFCEA
29	HistoryClearH	收集器历史数据清除使	W	0x0000/0xFCEA

		能		
2A	HistoryClearL	收集器历史数据清除使 能	W	0x0000/0xFCEA
2B	SensorClearH	传感器离线数据清除使 能(待定)	W	0x0000/0xFCEA
2C	SensorClearL	传感器离线数据清除使 能(待定)	W	0x0000/0xFCEA
采集器信息参数寄存器				
30	ProtocolVersion	接口协议版本号	R	0x0000~0xFFFF
31	ConfigVersion	参数配置版本号	R	0x0000~0xFFFF
32	SensorClearVersion	传感器离线数据清除 Ver	R	0x0000~0xFFFF
33	NewDataCount	最新节点数据个数	R	
34	HistoryMax	历史记录索引最大值	R	0x0000~0xFFFF
35	WritePos	历史记录写入索引	R	0x0000~0xFFFF
36	ReadPos	历史记录读取索引	R	0x0000~0xFFFF
37	WorkMode	当前的工作模式(待定)	R	0~2
网络设置传感器参数寄存器				
40	TempSampleIntv	温度采集间隔(秒)	RW	60~1800
41	TempAlaramIntv	温度报警间隔(秒)	RW	30~1800
42	TempThresholdHigh	温度上限阈值(°C)	RW	-40~120
43	TempThres holdLow	温度下限阈值(°C)	RW	-40~120
44	TempAlarmAddr	温度报警节点地址	RW	0xF001~0xFF9F
45	HumiditySampleIntv	湿度采集间隔(秒)	RW	60~1800
46	HumidityAlarmIntv	湿度报警间隔(秒)	RW	30~1800
47	HumidityThresholdHigh	湿度上限阈值(%)	RW	0~100
48	HumidityThresholdLow	湿度下限阈值(%)	RW	0~100
49	HumidityAlarmAddr	湿度报警节点地址	RW	0xF001~0xFF9F

2.2.2 采集器可操作 Modbus 指令

2.2.2.1 读取寄存器指令：功能码 03

请求 PDU:

功能码	1 字节	03 (0x03)
起始地址	2 字节	见采集器可操作 Modbus 寄存器
寄存器数据量	2 字节	见采集器可操作 Modbus 寄存器

响应 PDU:

功能码	1 字节	03(0x03)
字节数	1 字节	2*N
寄存器值	N*2 字节	

N=寄存器数量

错误码:

错误功能码	1 字节	(0x83)
异常字节数	1 字节	01 或 02 或 03 或 04

01: 功能码异常

02: 寄存器地址或寄存器地址+寄存器数量 异常

03: 寄存器数量 异常

实例:

读取采集器网络参数设置

发送命令如下:

01 03 00 10 00 04 CC 45

说明: 01 为 Modbus 地址
03 为功能码
00 10 为起始地
00 04 为寄存器数量
CC 45 为 CRC16 校验 (高位在前)

返回值:

01 03 08 E3 01 F0 01 0A 0A 00 01 60 00

说明: 01 为 Modbus 地址
03 为功能码
08 字节数量
E3 01 采集器 PANID
F0 01 采集器 MAC 地址
0A 0A 正常工作信道与配对信道
00 01 为 Modbus 地址
60 00 为 CRC16 校验 (高位在前)

2.2.2.2 写单个寄存器指令：功能码 06

请求 PDU:

功能码	1 字节	06 (0x06)
起始地址	2 字节	见采集器可操作 Modbus 寄存器
寄存器值	2 字节	见采集器可操作 Modbus 寄存器

响应 PDU:

功能码	1 字节	03 (0x06)
寄存器地址	2 字节	见采集器可操作 Modbus 寄存器
寄存器值	2 字节	见采集器可操作 Modbus 寄存器

错误码:

错误功能码	1 字节	(0x86)
异常字节数	1 字节	01 或 02 或 03 或 04

- 01: 功能码异常
- 02: 寄存器地址异常
- 03: 寄存器值 异常

实例:

设置采集器 MAC 地址

发送命令如下:

01 06 00 10 F0 02 CE 40

说明: 01 为 Modbus 地址

06 为功能码

00 10 为寄存器地址

F0 02 为寄存器值

CE 40 为 CRC16 校验 (高位在前)

返回值:

01 06 00 10 F0 02 CE 4D

说明: 01 为 Modbus 地址

06 为功能码

00 10 为寄存器地址

F0 02 采集器 MAC 地址

CE 4D 为 CRC16 校验 (高位在前)

2.2.2.3 写多寄存器指令：功能码 16

请求 PDU:

功能码	1 字节	(0x10)
起始地址	2 字节	见采集器可操作 Modbus 寄存器
寄存器数量	2 字节	见采集器可操作 Modbus 寄存器
字节数	1 字节	2*N
寄存器值	N*2 字节	值

N=寄存器数量

响应 PDU:

功能码	1 字节	(0x10)
起始地址	2 字节	见采集器可操作 Modbus 寄存器
寄存器数量	2 字节	见采集器可操作 Modbus 寄存器

错误码:

错误功能码	1 字节	(0x86)
异常字节数	1 字节	01 或 02 或 03 或 04

01: 功能码异常

02: 寄存器地址或寄存器地址+寄存器数量 异常

03: 寄存器数量 异常

实例:

设置采集器采集器网络参数

发送命令如下:

01 10 00 10 00 04 08 8A 6B F0 02 0A 0A 00 02 77 EA

说明:

- 01 为 Modbus 地址
- 10 为功能码
- 00 10 为寄存器起始地址
- 00 04 为寄存器数量
- 08 为字节数
- 8A 6B 采集器 PANID
- F0 02 为采集器 MAC 地址
- 0A 0A 正常工作信道与配对信道
- 00 02 为新设置的 Modbus 地址
- 77 EA 为 CRC16 校验 (高位在前)

返回值:

01 10 00 10 00 04 0F C0

说明: 01 为 Modbus 地址
 10 为功能码
 00 10 为寄存器起始地址
 00 04 为寄存器数量
 0F C0 为 CRC16 校验 (高位在前)

2.2.2.4 读取当前传感器节点在线地址列表: 功能码 65

请求 PDU:

功能码	1 字节	65 (0x41)
-----	------	-----------

响应 PDU:

功能码	1 字节	65(0x41)
节点数量	1 字节	0 至 30
节点 1 地址	2 字节	0x0001 至 0xEF EF
节点 2 地址	2 字节	0x0001 至 0xEF EF
...
节点 N 地址	2 字节	0x0001 至 0xEF EF

注意: 当前在线的传感器列表是指从转发器本次上电后所有有数据上报的传感器节点地址列表。每个转发器下可容纳的传感器节点数量为 30 个。每个转发器下的传感器地址必须唯一确定, 否则无法区分两个不同的传感器设备, 但这些无需人工分配, 全部由传感器节点与转发器的配对操作实现。

错误码:

错误功能码	1 字节	(0xC1)
异常字节数	1 字节	01

01: 功能码异常

实例:

假设现在采集器已经收到节点 MAC 地址为 0122 的最新数据。

发送命令:

01 41 10 C0

说明: 01 为 Modbus 地址
 41 为功能码
 10 C0 为 CRC16 校验 (高位在前)

回复命令:

01 41 01 01 22 B5 DD

说明: 01 为 Modbus 地址
 41 为功能码

- 01 为节点数量
 01 22 为节点 MAC 地址
 B5 DD 为 CRC16 校验（高位在前）

2.2.2.5 读取传感器最新数据：功能码 66

该命令支持两种读取方式：读取单点的传感器最新数据和读取所有节点的传感器最新数据。读取单点的最新数据，就是将请求 PDU 中的节点地址设置为对应的节点地址即可，取值范围 0x0001 至 0xEFEF；读取所有节点的传感器数据可将请求 PDU 中的节点地址设置为 0x0000。

请求 PDU:

功能码	1 字节	66 (0x42)
最大数据长度	1 字节	36 至 250
节点地址	2 字节	0x0000 至 0xEFEF
读取标识符	2 字节	0x0000 至 0xFFFF
上次读取序列号	1 字节	0x01~0xFF
上次数据读取确认	1 字节	0x00~0xFF

注意:

(1)、读取标识符可使用 0x0000 表示，收集器将其解析为从头开始读取，并自动产生一个非零的读取标识符，该标识符将包含在上报的数据包中，如果无法一包数据所有的最新数据读取完成，则在读取剩余数据的时候，上位机的请求 PDU 中的读取标识符应该使用与上报数据包中的读取标识符相同的数据值进行读取，否则收集器将会认为是新一轮数据读取的操作而从头开始上报数据。

(2)、请求 PDU 中包含了最大数据长度字段，表示通信发起方可接收的最大数据长度，最小值为 36 (0x24)，最大值为 255 (0xFF)

(3)、“上次读取序列号”和“上次数据读取确认”用于确认上次读取的数据是否得到了正确的回复，上次数据包接收正确，则本次 PDU 中的“上次数据读取确认”数值为 0，

响应 PDU:

功能码	1 字节	66(0x42)
本包数据长度	1 字节	36 至 250
节点地址	2 字节	0x0000 至 0xEFEF
读取标识符	2 字节	0x0001 至 0xFFFF
读取序列号	1 字节	0x01 至 0xFF
获取时间请求	1 字节	0xA5 表示请求最新时间
本地 UTC 时间	4 字节	
最新数据总数量	1 字节	0 至 120
本次上报数量	1 字节	0 至 25
上报完成剩余数量	1 字节	0 至 120
节点 1 数据	N1 字节	传感器有效数据内容
分隔符	1 字节	0xA7
节点 2 数据	N2 字节	

分隔符	1 字节	0xA7
...
节点 n 数据	Nn 字节	
结束符	1 字节	0xED

注意：

- (1)、读取序列号用于区分一轮数据上报过程中的不同数据包，也表示本包数据是本轮读取的第几次上报数据，每次上报增 1，如果上报失败，下次上报依然加 1，在下次请求 PDU 中将包含对本次序列号的回复信息。
- (2)、获取时间请求为 0xA5 时，表示请求通信发起方下发最新的 UTC 时间；0x00 表示不用下发最新的 UTC 时间。
- (3)、最新数据总数量表示目前 RAM 中存储的最新数据的数量，发起方可通过配置相应的寄存器清除收集器 RAM 缓存中的所有最新数据（参考采集器内部功能参数设置寄存器使用）。
- (4)、本次上报数量为 0 时，表示没找到对应的节点地址或没有传感器数据；上报完成剩余数量表示本轮读取（以读取标识字区分）在本次上报完成后，剩余的更新数据数量还有多少，当该值为 0 时，表示本轮读取完成。
- (5)、每包数据最多可容纳 14 个节点的数据（一般小于 14 个节点的数据），节点数据之间使用分隔符 0xA7，最后一个节点数据使用结束符。
- (6)、本包数据长度为：此字节后开始到 0xED 的长度。

错误码：

错误功能码	1 字节	(0xC2)
异常字节数	1 字节	01 或 02

01：功能码异常

02：最大数据长度 异常

实例：

假设采集器已经收到最新节点 A001、A002、A003、A004 的最新数据，采集器分两次读取，过程如下：

(1) 发送命令：

01 42 46 00 00 00 00 01 00 DD A4

说明： 01 为 Modbus 地址
 42 为功能码
 46 最大数据长度（70 字节）
 00 00 节点地址（全部顺序读取）
 00 01 读取标识符
 00 数据读取确认标志
 DD A4 为 CRC16 校验（高位在前）

2.2.2.6 读取传感器历史数据：功能码 67

请求 PDU:

功能码	1 字节	67 (0x43)
节点地址	2 字节	0x0000 至 0xEFEF
起始索引值	2 字节	0 至 0xFFFF
读取记录数量	1 字节	1 至 15

注意：当节点地址为 0x0000 时表示读取所有节点的传感器历史数据；当节点地址为非零时，仅读取一个节点的历史数据。请注意在读取历史数据前，尤其是第一次读取历史数据的时候请首先读取历史数据存储相关信息的寄存器，用于决定如何读取历史数据，每次读取的记录的最大数量为 15。

响应 PDU:

功能码	1 字节	67(0x43)
节点地址	2 字节	0x0000 至 0xEFEF
起始索引值	2 字节	0 至 0xFFFF
终止索引值	2 字节	0 至 0xFFFF
实际读取记录数量	1 字节	1 至 15
节点数据 1	N1 字节	传感器有效数据内容
分隔符	1 字节	0xA7
节点数据 2	N2 字节	
分隔符	1 字节	0xA7
...
节点数据 N	Nn 字节	
结束符	1 字节	0xED

实例:

1、顺序读取历史数据过程:

(1)、发送读取历史索引命令如下:

00 03 00 35 00 02 D4 D5

回复数据为:

01 03 06 00 1C A0 00 18 00 B7 D8

说明： 01 为 Modbus 地址
 03 为功能码
 06 字节数
 00 1C A0 历史数据写指针位置
 00 18 00 历史数据读指针位置
 B7 D8 为 CRC16 校验（高位在前）

历史数据存放的条数为=(历史数据写指针位置-历史数据读指针位置)/16
 （此条命令是用来估算历史数据的总数方便读取）

(2)、发送读取历史数据命令

00 43 00 00 00 01 05 60 D4

说明：

00	为广播的 Modbus 地址
43	为功能码
00 00	顺序读
00 01	第一条历史数据开始
05	5 条历史数据
60 D4	为 CRC16 校验（高位在前）

回复值为：

01 43 00 00 00 01 00 06 05 01 23 00 13 00 0A 53 4B 85 37 01 36
 00 82 00 1F A7 01 22 00 13 00 0A 53 4B 85 37 00 F9 01 33 00 1F
 A7 01 23 00 13 00 0A 53 4B 85 42 01 32 00 8C 00 20 A7 01 22 00
 13 00 0A 53 4B 85 42 00 F9 01 33 00 1F A7 01 23 00 13 00 0A 53
 4B 85 4D 01 2F 00 8C 00 20 A7 ED 68 DA

说明：

- 1)、 01 43 00 00
 Modbus 地址 功能码 顺序读
- 2)、 00 01 00 06 05
 起始索引值 终止索引值 实际读取记录数量
- 3)、 有效数据
 01 23 00 13 00 0A 53 4B 85 37 01 36 00 82 00 1F A7
 01 22 00 13 00 0A 53 4B 85 37 00 F9 01 33 00 1F A7
 01 23 00 13 00 0A 53 4B 85 42 01 32 00 8C 00 20 A7
 01 22 00 13 00 0A 53 4B 85 42 00 F9 01 33 00 1F A7
 01 23 00 13 00 0A 53 4B 85 4D 01 2F 00 8C 00 20 A7
 （有效数据具体格式见附录）
- 4)、 ED 68 DA
 结束符 为 CRC16 校验（高位在前）

2、读取特定 MAC 地址的历史数据

过程与顺序读取历史数据过程类似

2.2.3 采集器可操作 Modbus 寄存器操作注意事项

2.2.3.1 采集器网络参数设置寄存器（可读写）

地址	寄存器名称	寄存器说明	取值范围
10	PANID	无线通信网络地址	0x0001~0xFEFE
11	MAC	无线通信节点地址	0xF001~0xFEFE
12	CHNL (NORMAL/PAIRUP)	通信信道(正常/配对模式)	10~250
13	ModAddr	ModBUS 设备地址	1~250

注意：操作以上范围寄存器按照标准操作，注意取值范围

2.2.3.2 采集器内部功能参数设置寄存器

地址	寄存器名称	寄存器说明	取值范围
20	UTC_NOW_H	当前 UTC 时间高半字	0x0000~0xFFFF
21	UTC_NOW_L	当前 UTC 时间低半字	0x0000~0xFFFF
22	RfScanEnable	自动空间扫描功能使能(待定)	0x0000/0xEABE
23	RfBindEnable	无线绑定功能使能(待定)	0x0000/0xEABE
24	ConfigSave	配置信息写入使能(待定)	0x0000/0xCFAE
27	NewDataClearH	传感器最新数据清除使能	0x0000/0xFCEA
28	NewDataClearL	传感器最新数据清除使能	0x0000/0xFCEA
29	HistoryClearH	收集器历史数据清除使能	0x0000/0xFCEA
2A	HistoryClearL	收集器历史数据清除使能	0x0000/0xFCEA
2B	SensorClearH	传感器离线数据清除使能(待定)	0x0000/0xFCEA
2C	SensorClearL	传感器离线数据清除使能(待定)	0x0000/0xFCEA

注意：1)、地址为 20、21 (UTC 时间寄存器) 为可读写寄存器，要同步采集器时间必须对这两个寄存器都操作才能生效采集器的 UTC 时间；若监测到读取新数据包中获取时间请求字段为 0xA5 时就必须给采集器同步一次 UTC 时间；建议 30 分钟与采集器同步一次时间

2)、地址 22、23、24、2B、2C 功能暂时不开放

3)、地址 27、28、29、2A 为只写寄存器。要清除传感器最新数据，必须对 27、28 这两个寄存器都写 0xFCEA 才能生效，生效完成 27、28 寄存器里的值自动变为 0；要清除历史数据，必须对 29、2A 这两个寄存器都写 0xFCEA 才能生效，生效完成 29、2A 寄存器里的值自动变为 0；

2.2.3.3 采集器信息参数寄存器（只读寄存器）

地址	寄存器名称	寄存器说明	取值范围
30	ProtocolVersion	接口协议版本号	0x0000~0xFFFF
31	ConfigVersion	参数配置版本号	0x0000~0xFFFF
32	SensorClearVersion	传感器离线数据清除 Ver	0x0000~0xFFFF
33	NewDataCount	最新节点数据个数	
34	HistoryMax	历史记录索引最大值	0x0000~0xFFFF
35	WritePos	历史记录写入索引	0x0000~0xFFFF
36	ReadPos	历史记录读取索引	0x0000~0xFFFF
37	WorkMode	当前的工作模式(待定)	0~2

注意: 1)、地址 31 为最新节点数据个数寄存器为当前采集器收集到节点的 MAC 地址，最大值为 30，最小值为 0。

2)、地址 34 为历史记录索引最大值寄存器为采集器能保存离线数据的条数，其值为 62848 条 (0xF580)。

3)、地址 35 为历史记录写入索引指针寄存器，地址 36 为历史记录读取索引指针寄存器，读出寄存器值为 3 个字节，如：0x00 0x18 0x00。

情况一：当写入索引指针大于读取索引指针时

历史数据存放的条数为=(历史数据写指针位置-历史数据读指针位置)/16

情况二：当写入索引指针小于读取索引指针时

历史数据存放的条数为= ((历史数据写指针位置-存储的起始指针位置) +(存储的末端指针位置-历史数据读指针位置)) /16

其中：存储的起始指针位置=0x01800

存储的末端指针位置=0xF77FF

4)、地址 37 为当前的工作模式寄存器，功能暂时不开放

2.2.3.4 网络设置传感器参数寄存器（可读写）

地址	寄存器名称	寄存器说明	取值范围
40	TempSampleIntv	温度采集间隔(秒)	60~1800
41	TempAlarmIntv	温度报警间隔(秒)	30~1800
42	TempThresholdHigh	温度上限阈值(°C)	-40~120
43	TempThres holdLow	温度下限阈值(°C)	-40~120
44	TempAlarmAddr	温度报警节点地址	0xF001~0xFF9F
45	HumiditySampleIntv	湿度采集间隔(秒)	60~1800
46	HumidityAlarmIntv	湿度报警间隔(秒)	30~1800
47	HumidityThresholdHigh	湿度上限阈值(%)	0~100
48	HumidityThresholdLow	湿度下限阈值(%)	0~100
49	HumidityAlarmAddr	湿度报警节点地址	0xF001~0xFF9F

注意：1)、地址 40、41、45、46 寄存器按照 16 进制设置参数，

如：60（秒）=0x003C； 1800（秒）=0x0708

2)、地址 42、43 寄存器按照 16 进制设置参数

如：-40 =0xFFD8； 120=0x0078

2.2.4 其他指令说明

2.2.4.1 采集器空间频点扫频命令（用于初始化网络）

命令格式：

AA 80 E3 91 FC EA FC EA 00 EE DD 6B

说明：6B 为异或校验值

自动扫描开始回复：

AA 80 E3 92 ED EE DD 85

说明：85 为异或校验值

扫描完成回复：

AA 80 E3 93 0A 0A CC 0A AA CC ED 00 01 ED EE DD XOR

说明：

- 1)、两个 0A 分别表示自动选择的工作信道和绑定信道值，两个 CC 之间为按照 ED 值从小到大的顺序排列的信道值，两个 ED 之间为信道对应的 ED 值，排在前面的说明信道越好，可做为工作信道优选。
- 2)、XOR 为异或校验值
- 3)、在扫描过程中指示灯绿灯闪烁，此时 485 通信口不接收任何命令
- 4)、扫描时采集器必须接上天线，否则扫描值会出错

实例：

2.2.4.2 采集器恢复出厂设置命令

命令格式：

AA 80 E3 C7 00 EE DD 3D

说明：3D 为异或校验值

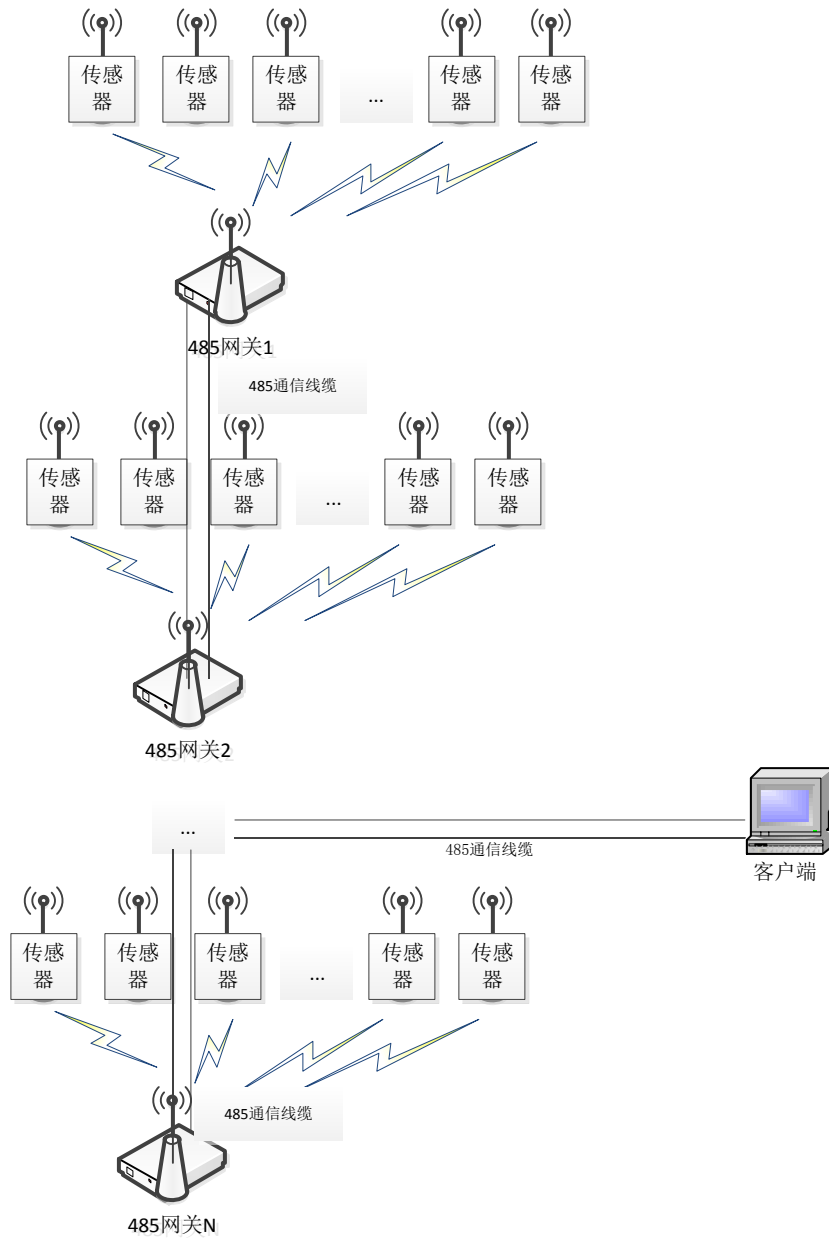
命令响应回复值为：

AA 80 E3 C8 ED EE DD DF

说明：DF 为异或校验值

此时采集器重启，恢复出厂默认配置

三、采集器安装拓扑结构图



四、附录

4.1 传感器数据结构及计算公式

4.1.1 采集器下发传感器数据格式

传感器节点数据的一般格式为：

节点 MAC 地址	数据类型	传感器类型	数据长度	UTC 时间	温度数据	湿度数据	电池电量
2 字节	1 字节	1 字节	2 字节	4 字节	2 字节	2 字节	2 字节

1)、数据类型： 00 为最新数据 01 为历史数据

2)、传感器类型： 温度传感器 0x11

湿度传感器 0x12

温湿度传感器 0x13

3)、UTC 时间：为采集器的 RTC 时间

4.1.2 温湿度传感器计算公式

温度实际值 = $-39.6 + 0.01 * \text{温度数据} (\text{°C})$

湿度实际值 = $-4.00 + 0.0405 * \text{湿度数据} + (-2.8000E-6) * \text{湿度数据} (\%)$

4.2 CRC16 算法

```
static const uint8_t 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, 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, 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
    } ;
    /* CRC 低位字节值表*/
    static const uint8_t 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
    };

    uint16_t modbus_CRC16(uint8_t *buffer, uint16_t usDataLen)
    {
        unsigned char uchCRCHi = 0xFF ; /* 高 CRC 字节初始化 */
        unsigned char uchCRCLo = 0xFF ; /* 低 CRC 字节初始化 */
        unsigned char  uIndex;
        uint16_t i ;

        i=0 ; /* CRC 循环中的索引 */

```

```
while (usDataLen-- ) /* 传输消息缓冲区 */
{
    uIndex = uchCRChi ^ buffer[i++] ; /* 计算 CRC */
    uchCRChi = uchCRCLo ^ auchCRChi[uIndex] ;
    uchCRCLo = auchCRCLo[uIndex] ;
}
return (uchCRCLo<< 8 |uchCRChi ) ;
}
```

五、技术支持



公司地址：北京市海淀区上地七街国际创业园 2 号 2 号楼 20B

工作时间：上午 9：00 - 下午：18：00

电 话：+86(10)82176393/58417092

传 真：+86(10)82176393

电子邮箱：sales@atzgb.com

公司网站：www.atzgb.com/www.atzgb.net/www.atzgb.org

由北京云天创科技有限公司（ATZGB）提供全面的技术支持：

电子邮箱：support@atzgb.com

支持电话：+86(10)58417092

响应时间：上午 9：00 - 下午：18：00

支持方式：电话、Email