

# ZigBee1085 无线数据采集卡

## 使用说明书

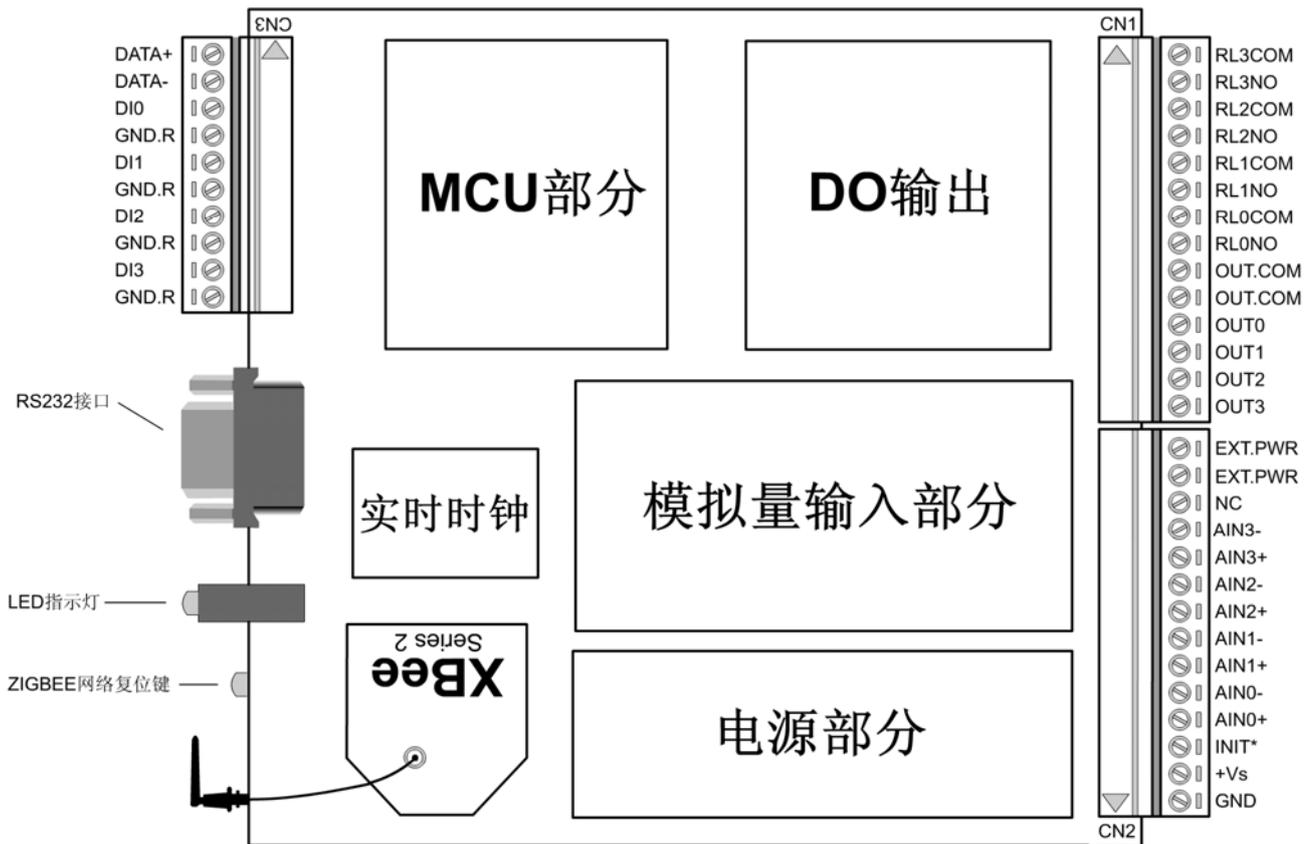


阿尔泰科技发展有限公司

产品研发部修订

# 一、模块结构及接线端子

## ZIGBEE1085模块结构及端子分布图



### 说明:

#### 1. 接线端子

GND: 电源负端

+Vs: 电源正端

INIT: 复位管脚，模块上电前，将此脚与 GND 短接，然后上电，则模块恢复为默认配置

(地址为 1，波特率为 9600)

AN0+ ~ AIN3-; 4 路模拟量信号输入端子

EXT.PWR: 集电极开路输出时，外接电源的正端

OUT.COM: 集电极开路输出时，外接电源的负端

OUT0~OUT3: 4 路集电极开路输出端

RL0NO~RL3NO: 4 路继电器输出正端

RL0COM~RL3COM: 4 路继电器开路输出负端

DATA+, DATA-: 485 总线，不使用 ZIGBEE 网络时，可以使用

DI0~DI3: 数字量输入正端

GND.R: 数字量输入地端

## 2. LED 指示灯

ZIGBEE1085 模块有 2 个指示灯, 分别为

- ① 黄灯: 网络指示灯, ZIGBEE1085 模块加入网络后, 此灯闪烁
- ② 红灯: 通讯指示灯, ZIGBEE1085 模块与数据中心通讯时, 此灯闪烁

## 3. 网络复位键

点击此键可复位本机的 ZIGBEE 网络, 使模块重新搜寻网络

## 4. RS232 接口

对 XBEE 模块进行配置时, 使用此口与计算机连接

# 二、模块主要性能指标

- 输入类型: 0~5V, 0~10V,  $\pm 5V$ ,  $\pm 10V$ , 0~20mA, 4~20mA,  $\pm 20mA$
- 通道输入: 4 路差分
- 采样频率: 10Hz
- 分辨率: 16 bit
- 精度: 见 TABEL2
- 满量程漂移: 25ppm/ $^{\circ}C$
- 隔离电压: 3000V
- 数字量输入: TTL 电平, 可测量高、低电平状态和频率 (Max 1K Hz)
- 数字量隔离电压: 3750V
- 集电极开路输出: 外部供电电压: +5V ~ +30V, 每路输出电流最大 100mA
- 继电器输出: 4 路 A 型功率继电器
- 继电器触点容量: 125VAC @0.5A, 30VDC @2A
- 内置看门狗
- 输入阻抗: 20M
- 电源: 未处理 +10V ~ +30VDC
- 功耗: 1W @24VDC (模块未使用 XBEE 网络传送数据, 且 DO 未输出时)

### 三、模块相关表格

■ TABLE1 波特率配置代码表

代码	00	01	02	03	04	05	06	07
波特率	1200	2400	4800	9600	19200	38400	57600	115200

■ TABLE2 模拟量输入类型配置代码及误差表

Input Type	Input Range	Typical Accuracy (%)	Maximum Error (%)	Code (Hex)
±5V	±5V	±0.03	±0.05	08
±10V	±10V	±0.03	±0.05	09
±20mA	±20mA	±0.05	±0.08	0A
0~20mA	0~20mA	±0.05	±0.08	0B
4~20mA	4~20mA	±0.05	±0.08	0C
0~5V	0~5V	±0.03	±0.05	0D
0~10V	0~10V	±0.03	±0.05	0E

### 四、MODBUS 协议地址分配

#### 1、协议说明

ZIGBEE1085 遵循 ZIGBEE 网络数据传输方式，包括网络 API 方式传输协议和 MODBUS 协议部分。其中，API 方式为阿尔泰 ZIGBEE 系列产品通用通讯方式，MODBUS 部分为标准 MODBUS 协议，方便用户与组态软件进行连接。

#### 2、MODBUS 协议部分

模拟量输入(AD)模块支持的功能码包括：

##### 1)、读开关量输出状态

功能码：01

数据地址：00001~00008

说明：读取输出继电器的状态

数据说明：

地址	描述	说明
00001	第 01 路开关量输出状态	=1 高电平 =0 低电平

00002	第 02 路开关量输出状态	=1 高电平 =0 低电平
00003	第 03 路开关量输出状态	=1 高电平 =0 低电平
00004	第 04 路开关量输出状态	=1 高电平 =0 低电平
00005	第 05 路开关量输出状态	=1 高电平 =0 低电平
00006	第 06 路开关量输出状态	=1 高电平 =0 低电平
00007	第 07 路开关量输出状态	=1 高电平 =0 低电平
00008	第 08 路开关量输出状态	=1 高电平 =0 低电平
保留		

**MODBUS 请求**

功能码	1 BYTE	0x01
起始地址	2 BYTE	0x0000 TO 0xFFFF
读取数量	2 BYTE	1 TO 8

**MODBUS 响应**

功能码	1 BYTE	0x01
字节计数	1 BYTE	N
线圈状态	n BYTE	n =N or N+1

N =读取数量/8 如果余数不为 0 则 N=N+1

**错误 响应**

功能码	1 BYTE	0x01+ 0x80
错误代码	1 BYTE	0x1 or 0x2

**举例**

请求		响应	
模块地址	数据 (hex)	模块地址	数据 (hex)
功能码	01	功能码	01
起始地址高(字节)	00	字节计数	01
起始地址低(字节)	01	00 (h) ~07 状态	01
读取数量高(字节)	00		
读取数量低(字节)	01		

**2)、读开关量输入**

功能码: 02

数据地址: 10001~10004

说明: 读取输入开关量的状态

**数据说明:**

地址	描述	说明
10001	第 01 路开关量输入状态	=0 没有通电 =1 接通电源
10002	第 02 路开关量输入状态	=0 没有通电 =1 接通电源
10003	第 03 路开关量输入状态	=0 没有通电 =1 接通电源
10004	第 04 路开关量输入状态	=0 没有通电 =1 接通电源
保留		

**MODBUS 请求**

功能码	1 BYTE	0x02
起始地址	2 BYTE	0x0000 TO 0xFFFF



读取数量	2 BYTE	1 TO 2000(0x7D0)
------	--------	------------------

**MODBUS 响应**

功能码	1 BYTE	0x02
字节计数	1 BYTE	N
输入状态	n BYTE	n =N or N+1

N =读取数量/8 如果余数不为 0 则 N=N+1

**错误 响应**

功能码	1 BYTE	0x02+ 0x80
错误代码	1 BYTE	0x1 or 0x2

**举例**

请求		响应	
模块地址	数据 (hex)	模块地址	数据 (hex)
功能码	02	功能码	02
起始地址高(字节)	00	字节计数	01
起始地址低(字节)	01	00(h)~03 状态	01
读取数量高(字节)	00		
读取数量低(字节)	04		

**3)、读保持寄存器**

功能码: 03

数据地址: 40129~40857

说明: 读取保持寄存器的值

**数据说明:** 读取的是十六位整数或无符合整数

地址	描述	说明
40129	模块类型寄存器	如: 1085 (HEX)
40130	模块类型后缀寄存器	如: 4244 (HEX) - 'BD'(ASC II)
40131	模块 MODBUS 协议标识	'+' : 2B20(HEX) - ASC II
40132	模块版本号	如: 0600 (HEX)
40133	模块地址	如: 01
40134	模块波特率	如: 03-9600bit/s
保留		
40257	第 1 路模拟量输入量程	Bit15_Bit 8 必须输入为 0。 Bit7_Bit 0 采集量程。 AD 模块支持的量程需参照模块说明书。 另外, AD 模块量程如果不支持单通道配置, 那么读取和写入只能对第 1 路输入量程进行读取和写入
40258	第 2 路模拟量输入量程	
40259	第 3 路模拟量输入量程	
40260	第 4 路模拟量输入量程	
保留		
40513	秒	Bit15~Bit8: 无效, 写 0 Bit7 Bit6~4 Bit3~0 0 秒的十位 秒的个位
40514	分	Bit15~Bit8: 无效, 写 0 Bit7 Bit6~4 Bit3~0 0 分的十位 分的个位

40515	时	Bit15~Bit8: 无效, 写 0 Bit7 Bit6 Bit5 0:24 时制 0 24 时制: 时的十位 1:12 时制 12 时制: 0:AM 1:PM Bit4 Bit3~0 时的十位 时的个位 例如: 要求设置时间为 24 时制的 15:32:54, 则各个地址的数据如下: 513: 0x54 514: 0x32 515: 0x15 (24 时制) 0xA3 (12 时制)
40516	日	Bit15~Bit8: 无效, 写 0 Bit7~6 Bit5~4 Bit3~0 0 日的十位 日的个位
40517	月	Bit15~Bit8: 无效, 写 0 Bit7~5 Bit4 Bit3~0 0 月的十位 月的个位
40518	年	Bit15~Bit8: 无效, 写 0 Bit7~4 Bit3~0 年的十位 年的个位
40519	周	Bit15~Bit8: 无效, 写 0 Bit7~3 Bit2~0 0 周的个位
保留		
40851	零点和满度恢复为默认值	1: 恢复为默认值, 其他值无效
40852	零点校准	1: 模块进行零点校准, 其他值无效 (只写)
40853	校准四个通道满度值	(只写)
保留		
40857	频率的周期值	程序中定义 DI 频率扫描周期为 400uS, 因此该值在 2500 左右, 由于时间存在偏差, 要对该值略作修正 (只写)

**MODBUS 请求**

功能码	1 BYTE	0x03
起始地址	2 BYTE	0x0000 TO 0xFFFF
读取数量	2 BYTE	1 TO 125(0x7D)

**MODBUS 响应**

功能码	1 BYTE	0x03
字节计数	1 BYTE	N*2
输入状态	N*2 BYTE	

**错误 响应**

功能码	1 BYTE	0x03+ 0x80
错误代码	1 BYTE	0x1 or 0x2

**举例**

请求	响应
----	----



模块地址	数据 (hex)	模块地址	数据 (hex)
功能码	03	功能码	03
起始地址高(字节)	01	字节计数	02
起始地址低(字节)	00	保持寄存器高	01
读取数量高(字节)	00	保持寄存器低	00
读取数量低(字节)	01		

#### 4)、读输入寄存器

功能码: 04

数据地址: 30101~30400

说明: 读取输入数据

**数据说明:** 读取的是十六位整数或无符合整数

地址	描述	说明
30101	第 01 路开关量频率	低 16 位计数数值
30102	第 01 路开关量频率	高 16 位计数数值
30103	第 02 路开关量频率	低 16 位计数数值
30104	第 02 路开关量频率	高 16 位计数数值
30105	第 03 路开关量频率	低 16 位计数数值
30106	第 03 路开关量频率	高 16 位计数数值
30107	第 04 路开关量频率	低 16 位计数数值
30108	第 04 路开关量频率	高 16 位计数数值
保留		
30257	第 1 路模拟量输入低 16 位	
30258	第 2 路模拟量输入低 16 位	
30259	第 3 路模拟量输入低 16 位	
30260	第 4 路模拟量输入低 16 位	
保留		
30400	环境温度	仅 DAM-3052BD 和 DAM-3039 模块支持 环境温度值 = 读回的值 * 0.1
保留		

#### MODBUS 请求

功能码	1 BYTE	0x04
起始地址	2 BYTE	0x0000 TO 0xFFFF
读取数量	2 BYTE	1 TO 125(0x7D)

#### MODBUS 响应

功能码	1 BYTE	0x04
字节计数	1 BYTE	N*2
输入状态	N*2 BYTE	

#### 错误 响应

功能码	1 BYTE	0x04+ 0x80
错误代码	1 BYTE	0x1 or 0x2

#### 举例

请求		响应	
模块地址	数据 (hex)	模块地址	数据 (hex)

功能码	04	功能码	04
起始地址高(字节)	01	字节计数	02
起始地址低(字节)	00	输入寄存器高(9)	00
读取数量高(字节)	00	输入寄存器低(9)	01
读取数量低(字节)	01		

## 5)、设置单个继电器

功能码: 05

### MODBUS 请求

功能码	1 BYTE	0x05
设置地址	2 BYTE	0x0000 TO 0xFFFF
设置内容	2 BYTE	0x0000 OR 0xFF00 0x0000 释放继电器 0xff00 吸合继电器

### MODBUS 响应

功能码	1 BYTE	0x05
设置地址	2 BYTE	0x0000 TO 0xFFFF
设置内容	2 BYTE	0x0000 OR 0xFF00

### 错误 响应

功能码	1 BYTE	0x05+ 0x80
错误代码	1 BYTE	0x1 or 0x2

### 举例

请求		响应	
模块地址	数据(hex)	模块地址	数据(hex)
功能码	05	功能码	05
设置地址高(字节)	00	设置地址高(字节)	00
设置地址低(字节)	05	设置地址低(字节)	05
设置内容高(字节)	FF	设置内容高(字节)	FF
设置内容低(字节)	00	设置内容低(字节)	00

## 6)、设置单个保持寄存器

功能码: 06

### MODBUS 请求

功能码	1 BYTE	0x06
设置地址	2 BYTE	0x0000 TO 0xFFFF
设置内容	2 BYTE	0x0000 to 0xFFFF

### MODBUS 响应

功能码	1 BYTE	0x06
设置地址	2 BYTE	0x0000 TO 0xFFFF
设置内容	2 BYTE	0x0000 to 0xFFFF

### 错误 响应

功能码	1 BYTE	0x06+ 0x80
-----	--------	------------



错误代码	1 BYTE	0x1 or 0x2
------	--------	------------

**举例**

请求		响应	
模块地址	数据 (hex)	模块地址	数据 (hex)
功能码	06	功能码	06
设置地址高(字节)	00	设置地址高(字节)	00
设置地址低(字节)	08	设置地址低(字节)	08
设置内容高(字节)	00	设置内容高(字节)	00
设置内容低(字节)	09	设置内容低(字节)	09

**7)、设置多个保持寄存器**

功能码: 10

**MODBUS 请求**

功能码	1 BYTE	0x10
设置起始地址	2 BYTE	0x0000 TO 0xFFFF
设置长度	2 BYTE	0x0000 TO 0x7B0
字节计数	1 BYTE	N*2
设置内容	N*2 BYTE	

**MODBUS 响应**

功能码	1 BYTE	0x10
设置起始地址	2 BYTE	0x0000 TO 0xFFFF
设置长度	2 BYTE	0x0000 TO 0x7B0

**错误 响应**

功能码	1 BYTE	0x10+ 0x80
错误代码	1 BYTE	0x1 or 0x2

**举例**

请求		响应	
模块地址	数据 (hex)	模块地址	数据 (hex)
功能码	10	功能码	10
设置地址高(字节)	00	设置地址高(字节)	00
设置地址低(字节)	01	设置地址低(字节)	01
设置数量高(字节)	00	设置数量高(字节)	00
设置数量低(字节)	02	设置数量低(字节)	02
字节计数	04		
设置内容高(字节)	00		
设置内容低(字节)	0A		
设置内容高(字节)	01		
设置内容低(字节)	02		

**8)、设置多个继电器**

功能码: 0F

**MODBUS 请求**

功能码	1 BYTE	0x0F
-----	--------	------

设置起始地址	2 BYTE	0x0000 TO 0xFFFF
设置长度	2 BYTE	0x0000 TO 0x7B0
字节计数	1 BYTE	N
设置内容	N BYTE	

**MODBUS 响应**

功能码	1 BYTE	0x0F
设置起始地址	2 BYTE	0x0000 TO 0xFFFF
设置长度	2 BYTE	0x0000 TO 0x7B0

**错误 响应**

功能码	1 BYTE	0x0F+ 0x80
错误代码	1 BYTE	0x1 or 0x2

**举例**

请求		响应	
模块地址	数据 (hex)	模块地址	数据 (hex)
功能码	0F	功能码	0F
设置地址高(字节)	00	设置地址高(字节)	00
设置地址低(字节)	13	设置地址低(字节)	13
设置数量高(字节)	00	设置数量高(字节)	00
设置数量低(字节)	0A	设置数量低(字节)	0A
字节计数	02		
设置内容高(字节)	CD		
设置内容低(字节)	01		

### 3、使用 MODBUS 协议模块注意事项

#### 1)、RTU 帧

使用 RTU 模式，每发一串完整的数据信息，称为一个 RTU 帧。每帧发送至少要以 3.5 个字符时间的间隔开始（如下表中的 T1-T4），在最后一个有效数据传输完成后，以一个 3.5 个字符时间的间隔作为该帧的结束。

消息帧格式表：

起始间隔	设备地址	功能代码	数据	CRC 校验	结束间隔
T1-T4	1 字节	1 字节	N 字节	2 字节	T1-T4

#### 2)、CRC 校验

使用 RTU 模式，消息帧包括了 CRC 校验值。整个消息帧中，除 CRC 校验的 2 个字节外的所有数据均参与 CRC 运算。

CRC 校验值长度为 2 个字节，消息帧内数据计算完成后，将 CRC 运算结果放到本消息的最后。接收设备收到消息帧后，重新计算 CRC 校验值，如和收到的校验值相同，则说明数据包传输正常；反之，则有误。

CRC 校验的 C 语言函数如下：



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  
};

### 3)、发送数据帧时的寄存器地址填写

在发送的数据帧中，需要指定要访问的寄存器地址。在本公司的 MODBUS 协议地址分配表中，寄存器的地址 = 数据帧中寄存器的地址 + 1。举例来说，需要访问输入寄存器的 30257 地址，其中“3”为输入寄存器地址的前缀，在某些组态软件中会用到，但是“3”不作为寄存器地址。“0257”是十进制数，表示寄存器地址，如果客户自己编写程序，则在发送的数据帧中，寄存器地址应填写“256”（即 0x0100）；如果在组态软件中，则直接填写“257”即可。