

# AW824P2EF

## ZigBee 二次开发模块

UM01010101 V1.00 Date: 2016/01/27

产品用户手册

类别	内容
关键词	无线通讯、FastZigBee、二次开发
摘要	开发 ZigBee 通信例程



## 修订历史

版本	日期	原因
V1.00	2016/01/27	创建文档

## 目 录

1. 产品简介.....	1
1.1 产品概述.....	1
1.2 产品特征.....	1
1.3 模块尺寸.....	1
2. FastZigBee 组网协议.....	2
2.1 特点.....	2
2.2 节点类型说明.....	4
2.3 软体基本配置参数.....	4
3. 配置工具.....	5
4. 参数配置协议.....	6
4.1 临时参数配置协议与命令.....	7
4.2 永久参数配置协议.....	8
4.3 命令详细介绍.....	9
5. 应用函数.....	14
5.1 ZigBee.h.....	14
5.2 ZigBee.c.....	15
5.3 ZigBeeFor824.c.....	16
6. 免责声明.....	19

## 1. 产品简介

### 1.1 产品概述

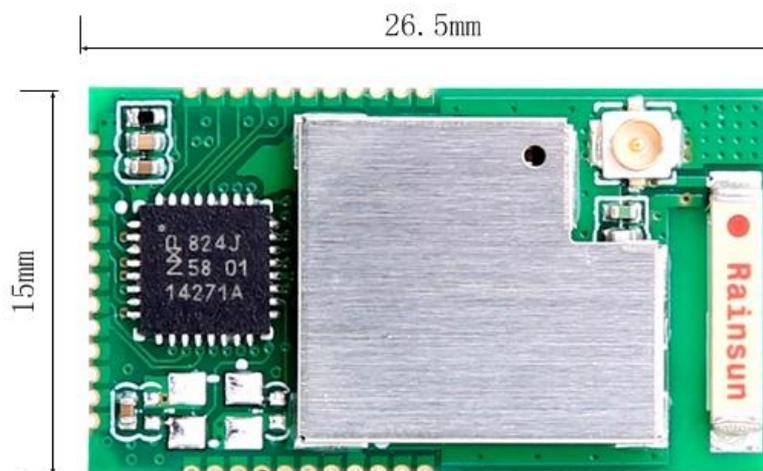
AW824P2EF 系列无线开发模块是广州致远电子股份有限公司基于 NXP 的 JN5161 (ZigBee 芯片)、LPC824 芯片开发的低功耗、高性能型 ZigBee 二次开发模块。JN5161 采用我司研发的 FastZigBee 协议固件，主控芯片 LPC824 通过串口即可轻松控制 JN5161 的 ZigBee 实现无线透传以及组网。集合这两款芯片的优势，AW824P2EF 无线开发模块可快速应用于工业控制、工业数据采集、农业控制、矿区人员定位、智能家居，智能遥控器等场合，大幅减小开发难度，使用户产品更快的投入市场，增加用户产品的竞争力，更好的把握住先机。

### 1.2 产品特征

- ◆ 工作电压 2.1V ~ 3.6V;
- ◆ 采用 NXP 基于 ARM Cortex - M0 + 的 LPC824 的处理器;
- ◆ 32KB Flash, 8KB SRAM;
- ◆ 3 路 USART(可分配给任意 IO 管脚), 4 路 I2C, 2 路 SPI, 12 路 ADC, 6 路 PWM;
- ◆ 支持 FastZigBee 固件, 轻松实现串口透明传输;
- ◆ 最大发射功率 20dbm;
- ◆ 最大接收灵敏度 -95dbm;
- ◆ 掉电模式下, 工作电流为 1.18  $\mu$  A;
- ◆ 使用陶瓷天线, 模块尺寸小;
- ◆ 支持外置天线, 增强信号覆盖范围;
- ◆ 邮票孔焊接方式;

### 1.3 模块尺寸

图 1.3 模块尺寸图



## 2. FastZigBee 组网协议

### 2.1 特点

标准 ZigBee 网络协议包括协调器、路由器和终端节点，而建立一个 ZigBee 网络除了必须要有协调器之外，仅需加上路由器或终端节点即可。在启动标准 ZigBeePro 网络通信前，如果没有建立存储跳转路径的路由表，则节点无法通信，所以必须通过发送网络报文查找节点建立路由表。当节点之间没有数据传输时，同样需要定时地发送网络报文检查节点是否异常。由此可见，ZigBee 不仅启动速度慢，而且定时发送网络报文占用大量的带宽。

如果网络上的任意节点都具备对等的数据传输功能，且无须协调器管理网络，那么网络中的任何一个节点都可以主动传输数据，这样一来用户就无需理会具体的网络结构，显然使用起来会更加方便，且施工人员也无需了解相关配置等。基于此，广州致远电子股份有限公司根据多年的实践经验积累，在 ZigBee 协议栈的基础上开发了 FastZigBee 透传对等网络协议。其特点如下：

- 启动零延时

FastZigBee 无需等待上电即可工作，该特点在低功耗应用领域尤其重要。

- 节点容量大

FastZigBee 采用 16bit 短地址管理方式，在轮询机制下，理论上节点容量高达 65535 个。

- 完全透传

FastZigBee 采用 3 线制串口全透明传输方式，即发送接收数据的长度和内容完全一致。其优势在于用户可以创建自己的协议格式，不再局限于固定第三方协议。

- 无需二次开发

如果节点数量小，那么使用配套的软件将参数配置好，并下载到 ZigBee 模块内置的 Flash 中保存起来，即可通过完全透传组网。当节点数量多到一定的程度时，如果还是一个一个节点地配置，则显然非常麻烦。那么使用 ZLG 提供的 AMetal 软件包，即可开发现场实时动态配置软件。

- 快速添加路由

FastZigBee 采用了即放即用的智能路由算法，当两个节点之间的距离超出通信范围时，那么只要在两个节点之间加入路由设备，其它任何网络参数都不要修改即可恢复通信，显然此路由方式特别便于施工。

当 A 节点向 C 节点发送数据时，无需知道是否存在 B 节点，只需将目标节点指向 C，则 B 节点会根据源地址和目标地址进行转发，详见图 2.1。当扩展到多级通讯时，同样是 A 节点，只需设置好目标节点即可与该节点通讯，节点 B 和节点 C 负责转发和通讯，详见图 2.2。

显然，采用 FastZigBee 全透传组网通讯，即可构建多种型态的网络拓扑结构。即：

- P2P 结构

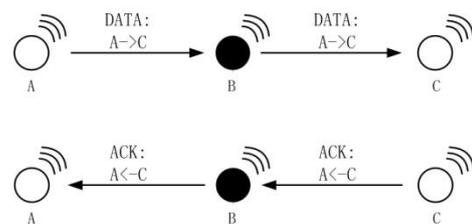


图 2.1 FastZigBee 通讯示意图

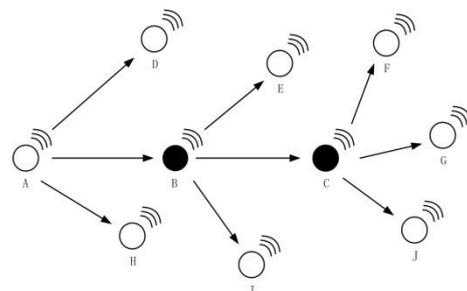


图 2.2 FastZigBee 多级通讯示意图

在 P2P (Peer to Peer) 点对点等网络环境中, 成千上万台彼此连接的计算机都处于对等的地位, 整个网络不依赖专用的集中服务器。网络中的每一台计算机既能充当网络服务的请求者, 又对其它计算机的请求作出响应提供资源和服务。

如图 2.3 所示的 ZigBee 点对点结构是最基本的拓扑结构, 专用于构建两个系统之间的通信, 该方式节点参数固定, 只要将两个节点的目标互相指向即可实现通信。

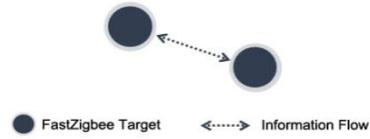


图 2.3 FastZigBee P2P 结构

### ● 星型网络拓扑结构

星型结构是最古老的一种连接方式, 大家每天都使用的电话就属于这种结构, 一般网络环境都被设计成星型拓扑结构, 因此星型网是广泛而又首选的网络拓扑设计之一。

在星型拓扑结构中, 网络中的各节点通过点对点的方式连接到一个中央节点上, 由该中央节点向目的节点传送信息。中央节点执行集中式通信控制策略, 因此中央节点相当复杂, 负担比各节点重得多, 在星型网中任何两个节点要进行通信都必须经过中央节点控制。

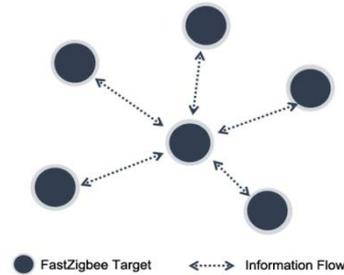


图 2.4 FastZigBee 星型网络拓扑结构

如图 2.4 所示的星型拓扑结构也称为主从结构, 该拓扑网络属于集中控制型网络, 整个网络由中心节点执行集中式通信控制管理, 各节点之间的通信都要通过中心节点, 一般由主控制中心不断切换通信目标进行轮询控制。

### ● 中继路由结构

中继 (Relay) 是两个交换中心之间的一条传输通路, 中继线是承载多条逻辑链路的一条物理连接。在日常生活中, 我们经常需要通过家里的电话和朋友聊天, 或者通过办公室的电话和公司外的客户联系, 要实现这些通话都离不开中继。在无线通信中, 中继的概念是指允许大量的用户在一个小区内共享相对较小数量的信道, 即从可用信道库中给每个用户按需分配信道。

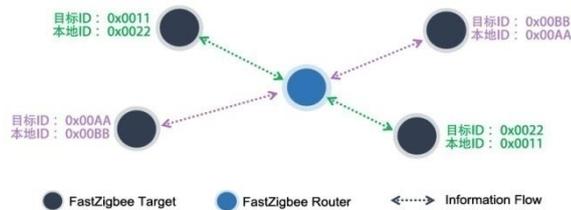


图 22.5 FastZigBee 中继路由结构

如图 22.5 所示的是最基础的中继路由拓扑图, 且终端可任意切换通信目标, 实现任意节点互相通信。

### ● 混合型网络拓扑结构

如图 2.6 所示的是将两种或几种网络拓扑结构混合起来构成的一种网络拓扑结构, 又称为混合型网络, 其不仅具备星型网络的简洁与低功耗, 而且兼备树形网络以及 MESH 网络的超远距离传输能力和自修复能力。在混合型网络中, 路由器组成网状结构, 而无线终端则在其周围呈现星型分布。路由中继扩展了网络的传输距离, 同时提供了容忍故障的能力, 在某些路由出现问题或强干扰时, 通信路径会进行自动调整, 以确保信息到达。

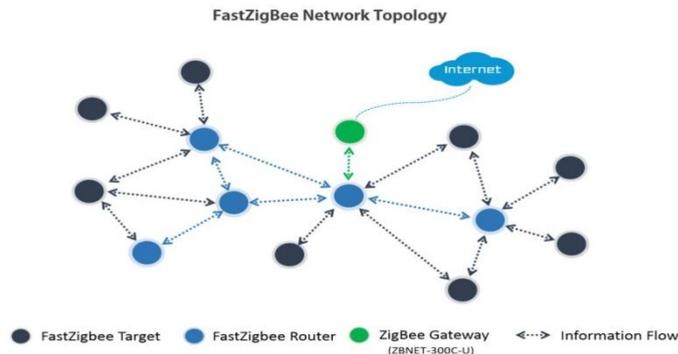


图 2.6 FastZigBee 网络拓扑结构

## 2.2 节点类型说明

FastZigBee 设备分为终端设备 (Target) 和路由设备 (Router), 且网络可通过 ZigBee 网关 (如 ZBNET-300C-U) 接入互联网, 详见表 2。

表 2.2 节点类型说明

节点类型	说明
终端设备 (Target)	终端设备的主要任务是发送和接收消息, 不允许其它节点与终端设备相连, 当没有数据收发时, 则进入休眠状态; 当需要收发数据时, 则通过 MCU 唤醒进入工作状态
路由设备 (Router)	允许其它节点与路由设备相连, 以扩大网络的覆盖范围, 其主要任务为转发报文, 起到中继路由作用, 并具备终端设备的所有功能。注意, 路由器必须保持活动状态, 保证终端报文实时转发, 因此不允许进入休眠状态。如果一个节点通往另一个节点存在多条路径时, 即便其中一条路径出现故障, 则网络会自动调整到其它路径传输, 以确保数据到达。注意, ZigBee 通信效率会随着路由级数的增加而下降, 所以路由器必须按需布局

## 2.3 软体基本配置参数

FastZigBee 协议提供了丰富的可配置参数, 用户可根据实际的应用需求灵活运用, 以构建不同形式的网络, 详见表 2., 以下所有配置参数均可通过 FastZigBee 配置工具或 AT 指令进行配置。

表 2.3 FastZigBee 主要配置参数

配置信息	参数范围	功能说明
PanID	0x0000~0xFFFF	PanID 即 ZigBee 局域网 ID, 用于判断自身所属的网络的标识。可互相通信的节点, PanID 必须相同, 且必须保证在同一工作区域内的相邻网络的 PanID 不同
本地网络地址	0x0000~0xFFFF	用于区分网络中各个节点的节点短地址即为本地网络地址, 在同一 PanID 下, 本地网络地址必须是唯一的, 引入短地址的目的是为了提高 ZigBee 的通信效率
目标网络地址	0x0000~0xFFFF	当前的通信目标 (地址), 可通过 AT 指令随时切换
本地物理地址	64bit MAC	本地物理地址即为模块的 MAC 地址, 不可修改的全球唯一标识, 但 FastZigBee 协议不使用
目标物理地址	64bit MAC	FastZigBee 协议不使用
设备类型	0、1	设备类型为 0, 即为终端设备; 设备类型为 1, 即为路由设备
通道号	CH 11~26	ZigBee 提供 16 个物理信道, 必须在同一通道下的节点才可能互相通信。在同一工作区域内的相邻网络, 建议使用不同的通道, 以免相互干扰导致通信效率降低。比如, 工作区域内存在大量的 2.4G Wi-Fi 热点, 可能会降低 ZigBee 的通信效率, 这时可选择 CH15、20、25、26, 达到有效避开干扰的目的
发送模式	0、1	单播模式就是在单个发送者和单个接受者之间的通信, 广播模式就是一个发送者和多个接受者之间的通信。发送模式参数为 0, 即为单播模式; 发送模式参数为 1, 即为广播模式
传输速率	250K	ZigBee 无线通信速率固定为 250K
发送功率	0~3 级	FastZigBee 提供 4 级功率可调, 该参数仅 P0(小功率)模块有效 0 = -32.0 dBm, 1 = -20.5 dBm, 2 = -9.0 dBm, 3 = +2.5 dBm
发送重试次数	0x00~0xFF	当数据发送失败后, 可通过多次发送保证数据的有效传递, 最多可设置 255 次。由于模块提供了 ACK 引脚, 因此用户可通过检测该引脚的数据判断是否成功发送
重试间隔时间	0x00~0xFF(ms)	当数据发送失败后, 可每隔一定的时间再发送保证数据的有效传递, 最大可设置为 255ms

### 3. 配置工具

AW824P2EF 系列模块提供了简易的图形配置工具, 用户通过该配置工具可以方便地对模块的运行参数进行配置, 配置的参数掉电可保存。配置步骤如下:

(1) 将模块的串口 (19 和 20 引脚, ZB\_TXD、ZB\_RXD) 通过 RS-232 电平转换后, 连接到电脑的串口, 模块上电, 打开配置软件的【串口】标签页, 波特率 (115200)、数据位 (8)、校验位 (无)、停止位 (1) 等根据模块的串口参数进行设定, 超时时间设置为 2000ms,

设定好串口参数后，点击【打开串口】按钮，详见图 3.1。（进行参数配置之前最好先把主控芯片 LPC824 的 PIO0\_26（19 号引脚）、PIO0\_27（20 号引脚）两个引脚设置为输入状态或擦除程序）。

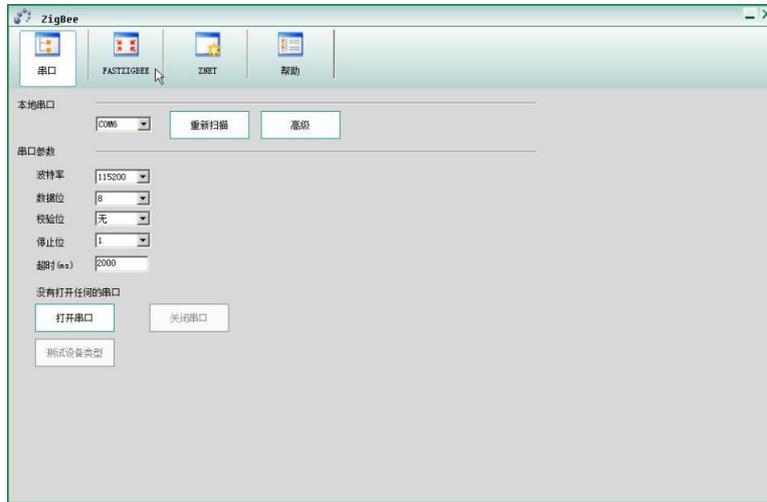


图 3.1 设定配置串口

(2) 选择【FastZigBee】标签页，点击【获取信息】按钮，获取模块的配置信息，修改模块各参数后点击【更改配置】，提交模块的配置信息，需要输入配置密码，配置密码默认为：88888，详见图 3.2。更改好模块的参数后，模块即可投入使用。



图 3.2 模块配置

## 4. 参数配置协议

AW824P2EF 的 ZigBee 串口协议包括临时参数配置协议和永久参数配置协议。用户可根据不同的需求使用不同的参数修改方式。

- 临时的参数配置协议是用于即时修改模块参数，不写入到模块的 Flash 存储器，掉电不保存；
- 永久的参数配置协议会将模块的参数保存到模块的 Flash 内，掉电保存，使用配置工具进行的配置是修改永久的参数配置。

## 4.1 临时参数配置协议与命令

AW824P2EF 的 ZigBee 临时参数配置协议命令帧格式详见表 4.1.1，协议标志占用 3 个字节，即：DE DF EF，功能码占用 1 个字节，其相应的参数占用 N 个字节。

表 4.1.1 临时参数配置命令格式 (CMD)

3 字节 (协议标志)	1 字节	N 字节
DE DF EF	功能码	参数

临时参数配置帧应答返回详见表 4.1.2，协议标志占用 3 个字节，即：DE DF EF，功能码占用 1 个字节，其返回值占用 1 个字节。

表 4.1.2 临时配置命令应答格式 (RSP)

3 字节 (协议标志)	1 字节	1 字节
DE DF EF	功能码	返回值

### 1. 修改通道号

其功能码为 0xD1，命令长度为 5 个字节，其参数为 1 个字节通道号 0x0B~0x1A。返回值为 0x00，表示修改成功；返回值为 0x06，表示超出范围。即：

```
CMD: DE DF EF D1 1A // 设置网络通道号为 0x1A
RSP: DE DF EF D1 00
```

### 2. 修改目的网络地址

其功能码为 0xD2，命令长度为 6 个字节，其参数为 2 个字节网络地址 0x0000~ 0xFFFF，返回值为 0x00，表示设置成功；返回其它值，表示设置失败。即：

```
CMD: DE DF EF D2 20 01 // 设置目标网络地址为 0x2001
RSP: DE DF EF D2 00
```

### 3. 包头显示源地址

其功能码为 0xD3，命令长度为 5 个字节，其参数为 0x00，表示不显示；其参数为 0x01，表示显示。返回值为 0x00，表示设置成功；返回其它值，表示设置失败。当设置成功后，则收到的数据包前 2 个字节为数据包源节点的网络地址。即：

```
CMD: DE DF EF D3 01 // 设置包头显示源地址
RSP: DE DF EF D3 00
```

在设置显示成功后，当模块收到一帧数据时，则数据包的前 2 个字节为数据包源节点的网络地址。即：

```
20 01 31 32 33 34 35 36 37 38 39
```

20 01 为源节点的网络地址，31 32 33 34 35 36 37 38 39 为接收到的数据。

### 4. 设置 I/O 输入输出

其功能码为 0xD4，命令长度为 7 个字节，其参数为 2 字节地址 + 1 字节 I/O 输入输出，bit0~bit7 为 IO0~IO7，其相应位为 1，表示 I/O 为输出；其相应位为 0，表示 I/O 为输入。返回值为 0x00，表示设置成功；返回其它值，表示设置失败。可设置本地 I/O 或远程 I/O。即：

```
CMD: DE DF EF D4 20 01 01 // 设置 I/O 输入输出
RSP: DE DF EF D4 20 01 00
```

将目标节点 2001 的 IO0 设置成输出，IO1~IO7 设置成输入。

### 5. 读取 I/O 状态

其功能码为 0xD5，命令长度为 6 个字节，其参数为被读节点的地址，1 字节 I/O 状态，bit0~bit7 位为 IO0~IO7。可读取本地 I/O 或远程 I/O，需先将 I/O 口设置成输入状态。即：

```
CMD: DE DF EF D5 20 01 // 读取 I/O 状态
RSP: DE DF EF D5 20 01 01
```

读取目标节点 2001 的 IO 状态。返回状态值 IO0 为高电平，IO1~IO7 为低电平。

## 6. 设置 I/O 状态

其功能码为 0xD6，命令长度为 7 个字节，其参数为 2 字节地址+1 字节 I/O 状态值，bit0~bit7 位分别代表 IO0~IO7。返回值为 0x00，表示设置成功；返回其它值，表示设置失败。可设置为本地 I/O 或远程 I/O，需先将 I/O 口设置成输出状态。即：

```
CMD: DE DF EF D6 20 01 01 // 设置 I/O 状态
RSP: DE DF EF D6 20 01 00
```

将目标节点 2001 的 IO0 设置成高电平，IO1~IO7 设置成低电平。

## 7. 读取 AD

其功能码为 0xD7，命令长度为 7 个字节，其参数为 2 字节地址+AD 通道号 (0~1)，返回值为 2 字节 AD 值，可设置本地或远程 10 位 AD，参考电压 2.47V。即：

```
CMD: DE DF EF D7 20 01 00 // 读取 AD 值
RSP: DE DF EF D7 20 01 00 9E
```

读取目标节点 2001 的 CH0 的 AD 值

## 8. 进入休眠

其功能码为 0xD8，命令长度为 5 个字节，其参数为 0x01，表示进入深度休眠状态；命令参数为其它值，则表示无效。当接收到此命令后，模块进入休眠状态，无返回值。即：

```
CMD: DE DF EF D8 01 // 模块进入深度休眠
```

休眠命令无返回，进入休眠后不保存临时参数配置，通过复位模块或管脚 19 拉低唤醒。

## 9. 设置通讯模式

其功能码为 0xD9，命令长度为 5 个字节，其参数为 0x00，表示单播模式；命令参数为 0x01，表示广播模式。返回值为 0x00，表示设置成功；返回值为其它值，表示设置失败。即：

```
CMD: DE DF EF D9 00 // 设置通讯模式为单播发送模式
RSP: DE DF EF D9 00
```

## 10. 查询节点的信号强度

其功能码为 0xDA，命令长度为 6 个字节，其参数为 0x0000，节点信号强度的返回值为 0x00~0xFF，0xFF 表示信号最强，经过路由无意义。即：

```
CMD: DE DF EF DA 20 02 // 获取模块的信号强度
RSP: DE DF EF DA 20 02 BA // 获取到的信号强度为 0xBA
```

获取到的信号强度是本地模块与目标模块 2002 之间的信号强度。

注：所获信号强度为带符号值，例如：0xBA=10111010（二进制）→取反+1=010001101+1=10001110=70（十进制）→（-70dbm）；如果获取的信号强度<-92dbm，通信就不可靠了。

## 4.2 永久参数配置协议

AW824P2EF 的 ZigBee 永久参数配置除了可以使用配置工具进行配置外，也可以使用命令的方式进行配置。永久参数配置协议命令帧格式详见表 4.1。

表 4.2.1 配置协议命令

3 字节 (协议标志)	1 字节	N 字节	1 字节
AB BC CD	命令标识符	命令实体	字节校验

字节校验为整条命令除校验位外所有字节相加的和 (1 字节)。

永久参数配置协议共有 7 条命令, 命令标识符详见表 4.。

表 4.2.2 配置协议命令标识

命令类型	命令标识符	备注
读取本地配置	0xD1	
设置通道号	0xD2	该指令掉电暂不保存
搜索	0xD4	
获取远程配置信息	0xD5	
修改配置	0xD6	设置成功需复位
复位	0xD9	
恢复出厂设置	0xDA	设置成功需复位

各配置命令帧返回的应答帧中包含有各种操作的响应状态, 各响应状态详见表 4.。

表 4.2.3 配置命令响应状态

响应状态	错误码
OK	0x00
LENGTH_FAUSE	0x01
ADDRESS_FAUSE	0x02
CHECK_FAUSE	0x03
WRITE_FAUSE	0x04
OTHER_FAUSE	0x05

### 4.3 命令详细介绍

#### 1. 读取本地配置命令

表 4.3.1 本地配置命令

3 字节 (协议标志)	1 字节	1 字节
AB BC CD	D1	校验

读取成功应答如下报文:

表 4.3.2 本地配置命令应答

3 字节 (协议标志)	1 字节	65 字节	1 字节	2 字节	2 字节
AB BC CD	D1	DEV_INFO 结构信息	运行状态	设备类型	固件版本

DEV\_INFO 结构信息详见表。

运行状态: 0xAA 该参数保留

固件类型: 0x0001 该参数保留

表 4.3.3 DEV\_INFO 结构信息

信息	偏移地址	长度 (字节)	备注	默认值
DevName	0	16	设备名称	ZLG Device
DevPwd	16	16	设备密码	88888
DevMode	32	1	设备类型 0: 终端设备 1: 路由设备	终端设备
Chan	33	1	通道号	0x19 (CH 25)
PanID	34	2	网络 ID (PanID)	0x1001
MyAddr	36	2	本地网络地址	0x2001
MyIEEE	38	8	本地物理地址(MAC)	每个模块的具有唯一的 MAC 地址, 不可修改
DstAddr	46	2	目标网络地址	0x2002
DstIEEE	48	8	目标物理地址 (保留)	0x00 00 00 00 00 00 00 00
Reserve	56	1	保留	0x00
PowerLevel	57	1	发射功率: 0x00: -32.0 dBm 0x01: -20.5dBm 0x02: -9.0 dBm 0x03: 2.5dBm	0x03
RetryNum	58	1	发送数据重试次数	0x05
TranTimeout	59	1	发送数据重试时间间隔 (单位: 10ms)	0x0A
Serial_Rate	60	1	串口波特率 <sup>[1]</sup>	0x07
Serial_DataB	61	1	串口数据位 <sup>[2]</sup>	0x08
Serial_StopB	62	1	串口停止位 <sup>[3]</sup>	0x01
Serial_Parity B	63	1	串口校验位 <sup>[4]</sup>	0x00
Reserve	64	1	发送模式: 0x00 = 单播 0x01 = 广播	0x00

(1) 串口波特率: 值为 1~7, 对应波特率: 2400、4800、9600、19200、38400、57600、115200

(2) 数据位: 5~8

注意: 数据位如果设置为 5、6、7 位, 则不可以获取配置信息。

(3) 停止位: 1~2

(4) 校验位: 校验位为 0, 表示无校验; 校验位为 1, 表示奇校验; 校验位为 2, 表示偶校验。

命令示例: 读取本地配置

```
CMD: AB BC CD D1 05
```

```
RSP: AB BC CD D1 5A 4C 47 20 44 65 76 69 63 65 00 00 00 00 00 00 38 38 38 38 38 00 00 00 00 00 00 00 00 00 00 01 19 10 01 20 01 00 38 1C 25 00 15 8D 00 20 02 00 00 00 00 00 00 00 00 00 03 03 0A 07 08 01
```

00 00 AA 00 01 01 00

## 2. 设置通道号

模块可工作在 16 个物理通道上，载波频率不同，不同通道的模块彼此间物理不通，可以实现物理上划分网段的效果。

表 4.3.4 设置通道命令

3 字节 (协议标志)	1 字节	1 字节	1 字节
AB BC CD	D2	通道号(11~26)	校验

设置成功回应报文如下：

表 4.3.5 设置通道命令应答

3 字节 (协议标志)	1 字节	1 字节
AB BC CD	D2	响应状态

命令示例：修改通道号

CMD: AB BC CD D2 0B 11

RSP: AB BC CD D2 00

## 3. 搜索

模块接收到本命令后，会向本网段所有通道的其他模块发出广播搜索包，运行本公司固件的 ZigBee 模块会应答此广播，将自己的相关基本信息返回到搜索发起节点。

表 4.3.6 搜索命令

3 字节 (协议标志)	1 字节	1 字节
AB BC CD	D4	校验

搜索成功回应报文如下：

表 4.3.7 搜索命令应答

3 字节 (协议标志)	1 字节	2 字节	1 字节	1 字节	2 字节	2 字节	1 字节
AB BC CD	D4	设备类型	通道号	速率	网络号	本地网络地址	运行状态

命令示例：搜索命令

CMD: AB BC CD D4 08

RSP: AB BC CD D4 00 01 0B 00 10 01 20 02 AA

## 4. 获取远程配置信息

为了获取其它节点的信息，可以通过向本机模块发送此命令。

表 4.3.8 获取远程配置命令

3 字节 (协议标志)	1 字节	2 字节	1 字节
AB BC CD	D5	目标网络地址	校验

注意：数据位如果设置为 5、6、7 位，则不可以获取配置信息。

远程的节点返回包含自己所有信息的数据包，回应报文如下：

表 4.3.9 获取远程配置命令应答

3 字节 (协议标志)	1 字节	65 字节	1 字节	2 字节	2 字节
AB BC CD	D5	DEV_INFO 结构信息	运行状态	设备类型	固件版本

命令示例：获取远程配置信息

```

CMD:   AB BC CD D5 20 02 2B
RSP:   AB BC CD D5 5A 4C 47 20 44 65 76 69 63 65 00 00 00 00 00 00 38 38 38 38 38 00 00 00 00 00
00 00 00 00 00 00 00 00 0B 10 01 20 02 00 38 1C 09 00 15 8D 00 20 01 00 00 00 00 00 00 00 00 03 03 0A 07 08
01 00 00 AA 00 01 01 00
  
```

## 5. 修改配置命令

表 4.3.10 修改配置命令

3 字节 (协议标志)	1 字节	2 字节	65 字节	1 字节
AB BC CD	D6	网络地址	DEV_INFO 结构信息	校验

修改本机配置时，只需在命令中填本地网络地址即可。设置成功回应如下报文：

表 4.3.11 修改配置命令应答

3 字节 (协议标志)	1 字节	2 字节	1 字节
AB BC CD	D6	网络地址	响应状态

响应状态详见表 4。

命令示例：修改配置命令

```

CMD:   AB BC CD D6 20 01 5A 4C 47 20 44 65 76 69 63 65 00 00 00 00 00 00 38 38 38 38 38 00 00 00 00 00
00 00 00 00 00 00 00 00 19 10 01 20 01 00 5B 28 61 00 15 8D 00 20 02 00 00 00 00 00 00 00 00 03 05 0A 07 08
01 00 00 78
RSP:   AB BC CD D6 20 01 00
  
```

## 6. 复位

表 4.3.12 复位命令

3 字节 (协议标志)	1 字节	2 字节	2 字节	1 字节
AB BC CD	D9	网络地址	设备类型	校验

命令示例：复位

```

CMD:   AB BC CD D9 20 01 00 01 2F          /* 复位帧无应答 */
  
```

## 7. 恢复出厂设置

表 4.3.13 恢复出厂命令

3 字节 (协议标志)	1 字节	2 字节	2 字节	1 字节
AB BC CD	DA	网络地址	设备类型	校验

恢复出厂设置帧应答如下报文：

表 4.3.14 恢复出厂命令应答

3 字节 (协议标志)	1 字节	2 字节	2 字节	1 字节
AB BC CD	DA	网络地址	设备类型	响应状态

命令示例：恢复出厂设置

CMD: AB BC CD DA 20 01 00 01 30

RSP: AB BC CD DA 20 01 00 01 00

## 5. 应用函数

为了更好地控制 ZigBee，我们提供了已经包装好的函数，调用这些函数，可以高效地获取和修改 ZigBee 的参数。

### 5.1 ZigBee.h

```
struct ZigBee_configuration
{
    uint8 DevName[16] ;
    uint8 DevPwd[16] ;
    uint8 DevMode ;
    uint8 Chan ;
    uint8 PanID[2] ;
    uint8 MyAddr[2] ;
    uint8 MyIEEE[8] ;
    uint8 DstAddr[2] ;
    uint8 DstIEEE[8] ;
    uint8 Reserve1 ;
    uint8 PowerLevel ;
    uint8 RetryNum ;
    uint8 TranTimeout ;
    uint8 Serial_Rate ;
    uint8 Serial_DataB ;
    uint8 Serial_StopB ;
    uint8 Serial_Parity;
    uint8 Reserve2 ;
};
extern struct ZigBee_configuration zb_config;
```

ZigBee 的配置结构，修改其各个字段可以修改 ZigBee 的各项参数

```
struct ZigBee_search_result zsr;
```

ZigBee 的搜索结果结构，用来存储各个远端 ZigBee 设备的通道号、网络号、本地地址。

```
enum temp_configuration_command
{
    set_channel=1, //设置通道号
    set_destinyaddress, //设置本机的目的地址
    set_displaysource, //设置是否显示数据源地址
    set_iodirection, //设置远程端或本机的 IO 方向
    get_iostate, //获取远程端或本机的 IO 方向
    set_iostate, //设置远程端或本机的 IO 状态
    get_advalue, //获取 AD 的数值
    set_sleep, //设置本机进入休眠模式
};
```

```

set_broadcast;//设置本机
get_signalstrength;//获取远程端的信号，获取本机信号会返回 0
};

```

临时指令的枚举

```

enum perm_configuration_command
{
    Pget_localconfig=1,//获取本机配置信息
    Pset_channel=2,//设置本机通道号
    Psearch=4,//搜索同频道、同网络号的设备
    Pget_remoteconfig=5,//获取远程端的配置信息
    Pset_config=6,//设置本机或远程端的配置信息（设置后需复位才能有效）
    Preset=9,//复位本机或远程端
    Pset_default=10,//使本机或远程端恢复出厂设置（设置后需复位才能有效）
};

```

永久指令的枚举

## 5.2 ZigBee.c

```
uint8 uart_buffer[500];
```

处理指令的缓冲区，既做发送缓冲区，也做接收缓冲区。

```

void zb_init_default(void)
{
    uint8 name[16] = "ZLG Device";
    uint8 password[16] = "88888";
    uint8 byte8[8]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,};
    memcpy(zb_config.DevName,name,16);
    memcpy(zb_config.DevPwd,password,16);
    zb_config.DevMode = 0x00;
    zb_config.Chan = 0x19;
    zb_config.PanID[0] = 0x10; zb_config.PanID[1] = 0x01;
    zb_config.MyAddr[0] = 0x20; zb_config.MyAddr[1] = 0x51;
    zb_config.DstAddr[0] = 0x20; zb_config.DstAddr[1] = 0x02;
    zb_config.Reserve1 = 0x00;
    zb_config.PowerLevel = 0x01;
    zb_config.RetryNum = 0x05;
    zb_config.TranTimeout = 0x0a;
    zb_config.Serial_Rate = 0x07;
    zb_config.Serial_DataB = 0x08;
    zb_config.Serial_StopB = 0x01;
    zb_config.Serial_Parity = 0x00;
    zb_config.Reserve2 = 0x00;
    memcpy(zb_config.MyIEEE, byte8,8);
    memcpy(zb_config.DstIEEE,byte8,8);
}

```

```
}
```

zb\_config 的初始化函数，此设置是出厂配置。

```
uint8 temp_command(uint8 command, uint16 addr, uint8 value);
```

临时指令函数：把临时指令和指令参数打包成需要发送的串口数据包，并把数据包保存在 uart\_buffer 里，函数的返回值指示数据包的大小。

```
uint8 temp_command_ask(uint8 command, uint16 addr, uint16 *returnvalue, uint8 *buffer);
```

应答临时指令函数：函数会对 buffer 里的数据（需要通过硬件串口把数据保存在 buffer 里）进行判断以确认 ZigBee 是否响应临时指令，判断成功，函数将返回 1，否则返回 0。

```
uint8 perm_command(uint8 command, uint16 addr, uint8 value);
```

永久指令函数：把永久指令和指令参数打包成需要发送的串口数据包，并把数据包保存在 uart\_buffer 里，函数的返回值指示数据包的大小。

```
uint8 perm_command_ask(uint8 command, uint16 addr, uint8 *buffer);
```

应答永久指令函数：对 buffer 里的数据（需要通过硬件串口把数据保存在 buffer 里）进行判断以确认 ZigBee 是否响应永久指令，判断成功，函数将返回 1，否则返回 0。

```
uint16 zb_write_buffer(uint16 destinyaddrss);
```

缓冲区写入函数：destinyaddrss 指示要被修改配置信息的设备，若是本机则传入本地地址，若是远程端，则传入远程端的本地地址。函数会把 zb\_config 结构体按照指令格式，写入到 uart\_buffer，方便用户直接发送数据。返回值为缓冲区的大小。

```
void zb_read_buffer(const uint8* buffer);
```

缓冲区读入函数：buffer 指向缓冲区的地址，一般设为串口接收缓冲区的地址。函数用于永久参数修改指令 Pget\_localconfig，可以把缓冲区内的数据写入 zb\_config 结构体中，方便用户直接访问 zb\_config 结构体来获取 ZigBee 设备的各项设置信息。

```
uint8* zb_buffer_addr(void);
```

获取缓冲区地址函数：返回 uart\_buffer 的地址。

### 5.3 ZigBeeFor824.c

```
void send_temp_command(uint8 command, uint16 addr, uint8 value)
```

发送临时指令函数：会把参数传递进 temp\_command 函数，并且会通过硬件串口把数据发送至 ZigBee。

```
uint8 rece_temp_command(uint8 command, uint16 addr, uint16 *returnvalue, uint16  
timeout)
```

接收临时指令函数：调用硬件的串口接收程序，并且设定超时值，如果在超时时间内正

常接收到 ZigBee 的响应信息会继续调用 temp\_command\_ask 函数，判断 ZigBee 的响应信息是否正确并把相应的数值赋值到 returnvalue。超时或反馈信息有误，函数返回 0。无超时且响应信息正确，函数返回 1。

```
void send_perm_command(uint8 command, uint16 addr, uint8 value)
```

发送永久指令函数：把参数传递给 perm\_command 函数，并且调用 zb\_write\_buffer 函数计算缓冲区，最后把数据送至串口，配置 ZigBee。

```
uint8 rece_perm_command(uint8 command, uint16 addr, uint16 *returnvalue, uint16 timeout)
```

接收永久指令函数：调用硬件的串口接收函数，并且设定超时值，如果在超时时间内正常接收到 ZigBee 的响应信息会继续调用 perm\_command\_ask 函数，判断 ZigBee 的响应信息是否正确并把相应的数值赋值到 returnvalue。超时或反馈信息有误，函数返回 0。无超时且反馈信息正确，函数返回 1。

```
void temp_command_debug(void)
```

临时参数指令调试函数：会依次给 ZigBee 模块发送所有的临时指令，并且把测试结果输出到另外一路调试串口上（ZigBee 串口通信使用 P0.26 Rx，P0.27 Tx，调试串口使用 P0.6 Rx，P0.7 Tx），用户通过串口助手可以看到主控 IC 反馈回来的调试信息。

```
void perm_command_debug(void)
```

永久参数指令调试函数：会依次给 ZigBee 模块发送所有的永久指令，并且把测试结果输出到另外一路调试串口上（ZigBee 串口通信使用 P0.26 Rx，P0.27 Tx，调试串口使用 P0.6 Rx，P0.7 Tx），用户通过串口助手可以看到主控 IC 反馈回来的调试信息。

```
void zb_init(uint16 addr)
{
    uint16 returnvalue,myaddr;
    print_string_uint16("Random number:",addr);
    /* 设置通道号为 0x10, 以避免配置同通道的其他远程节点 */
    send_temp_command(set_channel,0x0000,0x10);
    if(rece_temp_command(set_channel,0x0000,&returnvalue,50))
    {
        print_string_uint16("\r\n ...set channel to:",0x10);
    }
    /* 获取本机配置信息 */
    if(get_zb_config())
    {
        print_string_string("\r\nPget_localconfig:");
        /* 保存本机的本地地址 */
        myaddr = combine_2byte(zb_config.MyAddr);
        print_string_uint16("\r\nMyAddr:",myaddr);
    }
}
```

```
}
else
{
    print_string_string("\r\nPget_localconfig:failed:","");
}
/* 初始化 zb_config 结构体变量 */
zb_init_default();
/* 修改本地地址 */
//zb_config.MyAddr[0] = addr>>8;
//zb_config.MyAddr[1] = addr;
/* 修改网络号 */
zb_config.PanID[0] = 0x10;
zb_config.PanID[1] = 0x81;
/* 修改本地地址和目标地址 */
if(addr == 0x2001)
{
    zb_config.MyAddr[0]= 0x20;
    zb_config.MyAddr[1]= 0x01;
    zb_config.DstAddr[0] = 0x20;
    zb_config.DstAddr[1] = 0x02;
}
else
{
    zb_config.MyAddr[0]= 0x20;
    zb_config.MyAddr[1]= 0x02;
    zb_config.DstAddr[0] = 0x20;
    zb_config.DstAddr[1] = 0x01;
}
/* 以原本地地址，把 zb_config 的配置信息发送出去 */
send_perm_command(Pset_config,myaddr,0x00);
if(rece_perm_command(Pset_config,myaddr,&returnvalue,50))
{
    print_string_uint16("\r\nPset_config:",myaddr);
}
else
{
    print_string_string("\r\nPset_config:failed:","");
}
/* 硬件复位本机设备 */
zb_reset();
print_string_uint16("\r\nRrset local device :",myaddr);
if(get_zb_config())
{
    /* 获取本机配置信息，本地地址已设置成 addr */
```

```
print_string_string("\r\nPget_localconfig:");
//myaddr = combine_2byte(zb_config.MyAddr);
//print_string_uint16("\r\nMyAddr:",myaddr);
print_zb_config();
}
else
{
    print_string_string("\r\nPget_localconfig:failed:");
}
```

ZigBee 的初始化函数的工作流程:

(1) 修改参数之前首先要知道本机 ZigBee 的本地地址, 因此要调用 `get_zb_config()` 函数, 获取本机的 ZigBee 参数, 函数返回正确时, ZigBee 参数将保存在 `zb_config` 结构体中, 访问 `zb_config.MyAddr` 字段把地址保存在 `myaddr` 中。

(2) 调用 `zb_init_default()` 函数把 `zb_config` 恢复默认值, 随后可以修改 `zb_config` 各个字段来修改想要改变的参数。

(3) 调用 `send_perm_command(Pset_config,myaddr,0x00)` 函数把 `zb_config` 打包成数据包发通过串口发送至 ZigBee, 其中 `Pset_config` 指示使用的是配置参数指令, `myaddr` 指示要的设备地址也就是之前保存的地址, `0x00` 无作用。

(4) 调用 `rece_perm_command(Pset_config,myaddr,&returnvalue,50)` 函数用于判断 ZigBee 是否正确响应, 保存好参数。

(5) 调用 `zb_reset()` 函数硬件复位设备, 因为永久修改配置参数需要复位才能生效。

## 6. 免责声明

AW824P2EF 系列无线开发模块及相关资料版权均属广州致远电子股份有限公司所有，其产权受国家法律绝对保护，未经本公司授权，其它公司、单位、代理商及个人不得非法使用和拷贝，否则将受到国家法律的严厉制裁。

本档提供有关致远电子产品的信息。本档并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。除致远电子在其产品的销售条款和条件中声明的责任之外，致远电子概不承担任何其它责任。并且，致远电子对致远电子产品的销售和 / 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。致远电子产品并非设计用于医疗、救生或维生等用途。致远电子可能随时对产品规格及产品描述做出修改，恕不另行通知。

AW824P2EF 系列无线开发模块可能包含某些设计缺陷或错误，一经发现将收入勘误表，并因此可能导致产品与已出版的规格有所差异。如客户索取，可提供最新的勘误表。

在订购产品之前，请您与当地的致远电子销售处或分销商联系，以获取最新的规格说明。本档中提及的含有订购号的文档及其它致远电子文献可通过访问广州致远电子股份有限公司的万维网站点获得，网址是：[www.zlg.cn](http://www.zlg.cn)

广州致远电子股份有限公司保留在任何时候修订本用户手册且不需通知的权利。