
无锡谷雨电子有限公司

GY-Z15 系列软件设计手册

适用型号 Z150 / Z151 / Z150PA / Z151PA

V1.2.02

目录

一、前言.....	2
二、使用介绍.....	2
2.1 模块的连接.....	2
2.2 配置模块的设备类型.....	2
2.3 配置模块的网络参数.....	3
三、功能介绍.....	4
3.1 数据透传.....	5
3.2 点对点传输数据.....	6
3.3 点对多点数据传输.....	7
3.4 广播发送数据.....	7
3.5 外设控制.....	7
3.6 属性设置.....	10
3.6 设置数据轮训时间.....	11
3.7 终端模式串口唤醒模块.....	11
3.8 终端模式下数据接收与发送.....	11
3.9 网络中更换协调器.....	12
四、AT 指令说明.....	12
附录 1: 联系方式.....	28
附录 2: 版本记录.....	29

一、前言

Z15 系列 ZigBee 模块（以下简称 Z15x 模块）是谷雨物联网开发的一款串口数据无线透传模块。支持点对点数据传输，广播数据传输等数据发送方式。在串口数据无线透传的基础上，我们加入了一些特色功能，其中包括远程 ADC 采样，远程 PWM 波输出控制，远程 GPIO 控制。用户不需要了解复杂的无线通信原理，根据手册，发送简单的控制指令，即可实现数据的无线控制与传输。通过该手册，您将了解该模块的具体使用方法。

手册通过三部分介绍该模块。第一部分介绍了模块的参数以及各参数的意义。第二部分介绍了模块不同的功能，包括数据传输和无线控制。第三部分介绍了 AT 指令及详细使用方法。

二、使用介绍

2.1 模块的连接

Z15x 模块需要通过串口和用户的设备连接。如果用户需要将模块和电脑连接，需要使用 TTL 转 USB 转接线或者使用 TTL 转 RS-232 转接线。和电脑连接后，即可使用配套的设置软件，设置其参数并且调试。

2.2 配置模块的设备类型

Z15x 系列 ZigBee 模块具有三种设备类型，分别为协调器，路由器和终端。出厂时设备类型默认路由器。用户可根据需要，更改设备类型（使用 AT+ROLE），详情参考 AT 指令集章节。也可使用上位机软件修改更多参数。

需要说明的是，一个网络中有且只能有一个协调器，同一个空间环境中可以存在多个协调器。但是，这些协调器的 PANID 和信道不能完全相同，否则，会产生冲突。路由器和终端设备通过设置其 PANID 和信道两个参数与协调器相同，来加入该协调器网络。

◆ 协调器：

协调器（Coordinator）也可称作中心设备，用于**创建** ZigBee 网络，与 WiFi 网络名 SSID 类似，用于标识 ZigBee 网络的称作 PANID，其他设备：路由器及终端节点，需要设置与协调器相同的 PANID 及信道参数，即可加入到该网络中。

在同一个 ZigBee 网络中，有且只有一个协调器设备，路由器和终端节点无限制。

◆ 路由器：

路由器（Router）是 ZigBee 网络中最常用的设备类型，不但自身可以收发数据，还可以将数据转发给相邻的设备，并且可以为处于休眠状态的终端设备缓存数据。除此之外，路由器还具有保持网络的能力，即便协调器由于某种原因掉线，或者损坏，其他设备仍然可以通过路由器，继续加入到网络中来。

◆ 终端：

终端（End Device）是为低功耗而设计，降低了通信的实时性，因此终端设备无法立刻收到数据，而是定期向父设备轮训数据，再配合休眠配置，可最大程度降低终端设备的功耗。

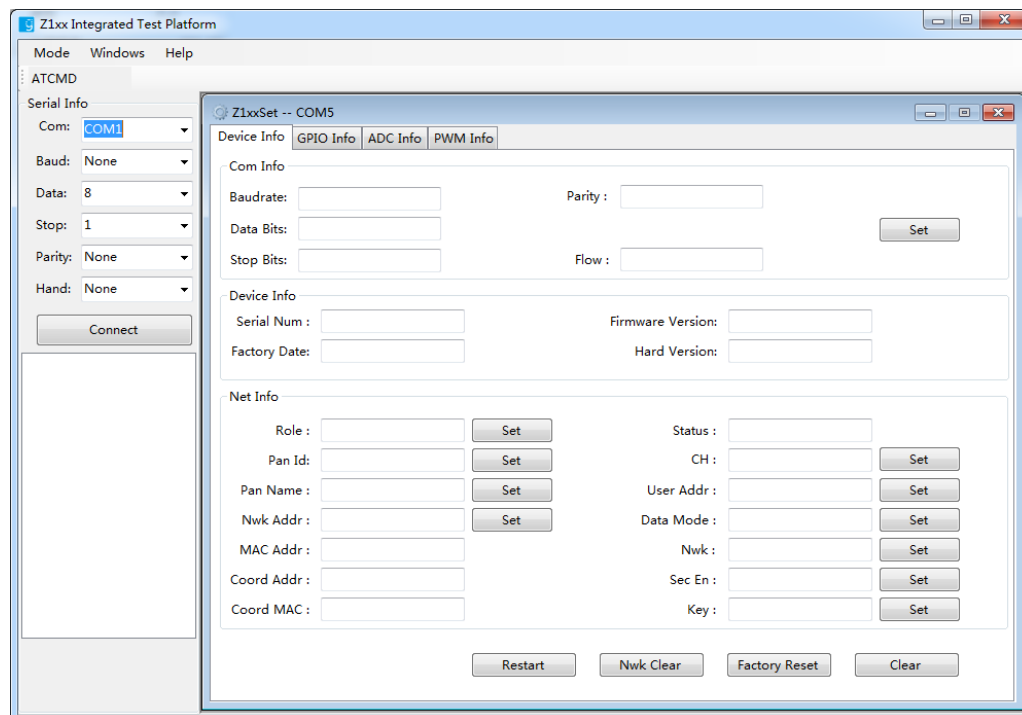
两节 7 号电池，可工作可达 10 个月之久

终端轮训周期使用指令 AT+POLL 设置；休眠参数，使用指令 AT+SLEEP 设置。

2.3 配置模块的网络参数

模块参数的配置方式有两种。

- 1: 进入 AT 指令模式下，通过 AT 指令配置模块的参数。
- 2: 使用配套的设置软件进行配置。



信道 (CH): 信道是决定天线发射的频率，Z15x 模块的设置信道的范围是 11 到 26，即 2.4GHz 频率附近，欲互相通信的 Z15x 模块，需要保证在相同的信道上。用户在设置了 Z15x 模块的信道后，在模块没有加入网络之前，信道可能变到其他信道，用户不必再次修改，在设备发现网络后，会跳回该信道。

PANID (Pan Id): PANID 用来区分同一信道中的 ZigBee 网络，在协调器模式下设置的 PANID，代表用户欲创建该网络，在路由器模式或者终端模式下设置的 PANID，代表该设备欲加入的网络。如果一个空间环境中有多台协调器，其 PANID 需要不相同，这样，网络才能够被区分，路由器或者终端通过设置其 PANID 来加入目标网络，在同一个网络中的设备即可互相传输数据。在不知道目标网络的 PANID 的情况下，协调器或者路由器想加入目标网络，可以设置其 PANID 为 0xFF FF（一般应用不要设置成 0xFF FF）。重启这个节点，这个节点就会加入附近 ZigBee 的网络。

扩展 PANID (Pan Name):

网络地址 (Nwk Addr): 网络地址又称短地址，在节点设备加入网络的时候，会自动获得一个网络地址，该网络地址可以人为修改，但是，在特定的环境下，网络地址存在变化的可能。

自定义地址 (User Addr): 自定义地址默认是 0xFF FF，用户修改自动地址后，无论在什么情况下，都不会改变（包括恢复出厂设置），在一个网络中，相同的自定义地址不会发生冲突。

父地址 (Coord Addr): 在了解父地址时，需要先了解父设备，一个节点设备连接的路由器或者协调器称为该节点的父设备，父地址即是父设备的网络地址。

传输方式 (Data Mode): Z15x 模块具有 5 种数据传输方式，具体的对应关系请参照下表：

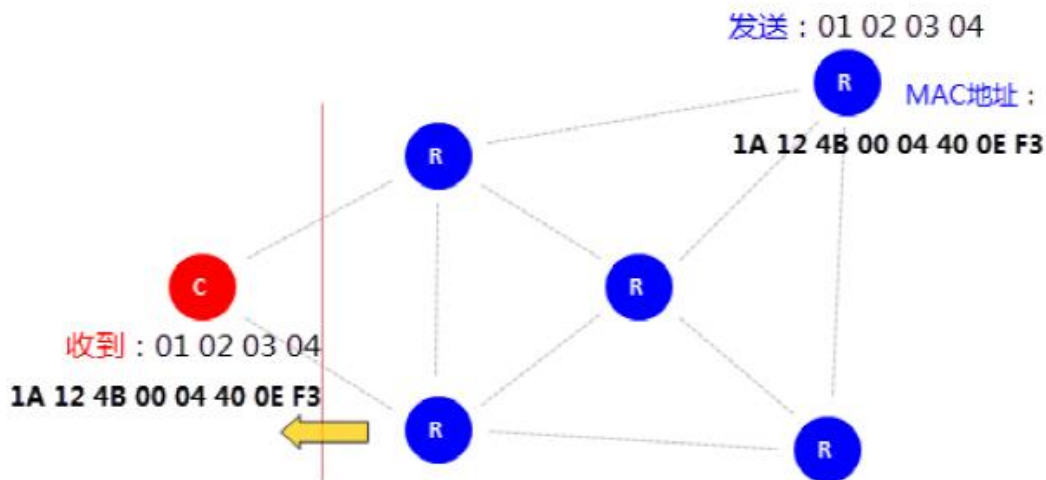
传输模式值 (十六进制)	数据透传方式下	点对点传输方式下
00	数据透传	Zigbee 网络地址寻址，含包头包尾
01	数据透传+Zigbee 网络地址	Zigbee 网络地址寻址，含包头包尾
02	数据透传+MAC 地址	Zigbee 网络地址寻址，含包头包尾
03	数据透传+自定义地址	Zigbee 网络地址寻址，含包头包尾
04	数据透传	Zigbee 网络地址寻址，不含包头包尾
05	数据透传	Zigbee 自定义地址寻址，不含包头包尾

传输方式只对发送设备有效，用户不用关心接收设备的传输方式。

- 上述中网络地址寻址或者自定义地址寻址，是决定点对点数据传输或者点对多点数据传输中的目的地址填写。如下图黄色部分：



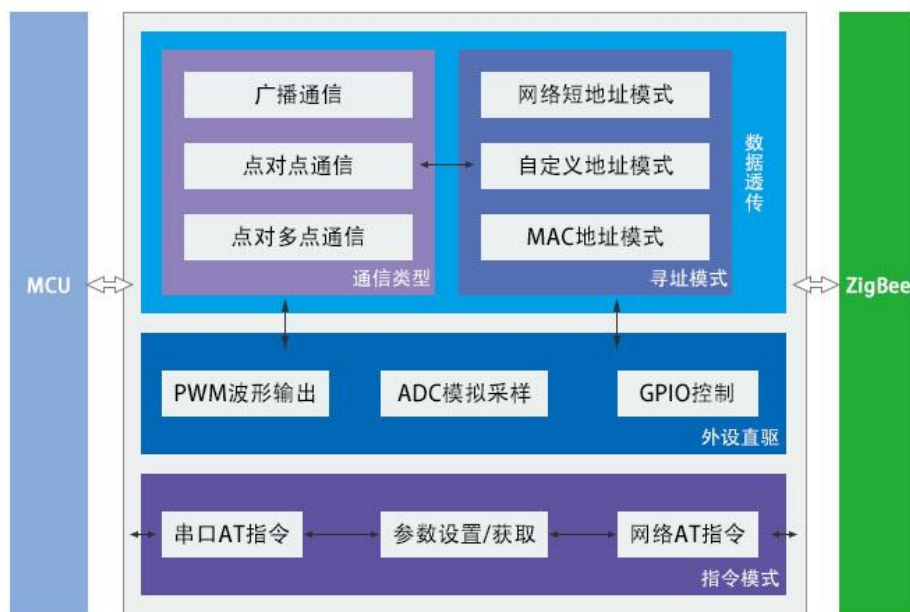
- 上述中的数据透传+MAC 地址是在数据透传的基础上将发送设备的网路地址附加在数据的末尾。如下图：（上述中的数据透传+ZigBee 网络地址和数据透传+自定义地址与其相同）。



- 上述中的是否包含包头包尾是在点对点或者点对多点数据传输中，接收方接收到的数据是否保留 FD，数据长度，目的地址等相关信息。

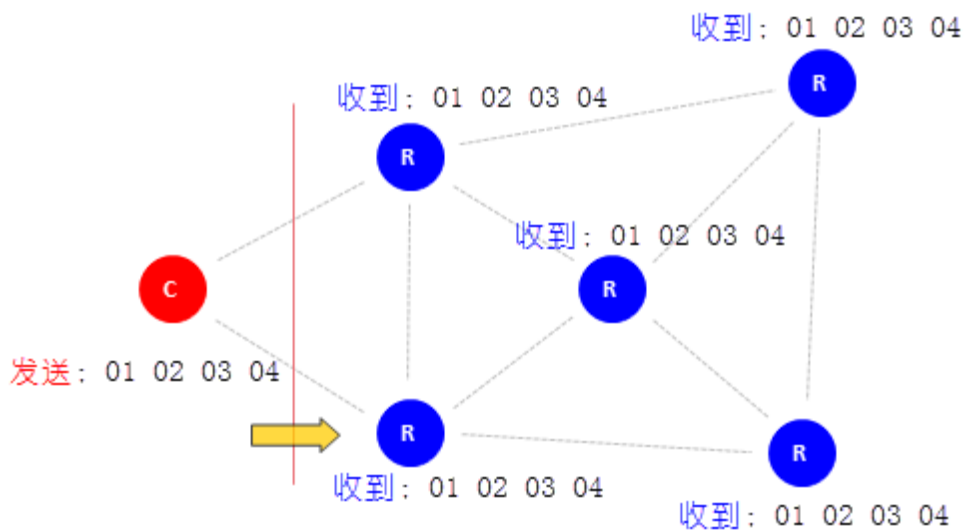
三、功能介绍

模块的功能主要是两方面，1：数据透传，2：无线控制。以下为模块功能框图。

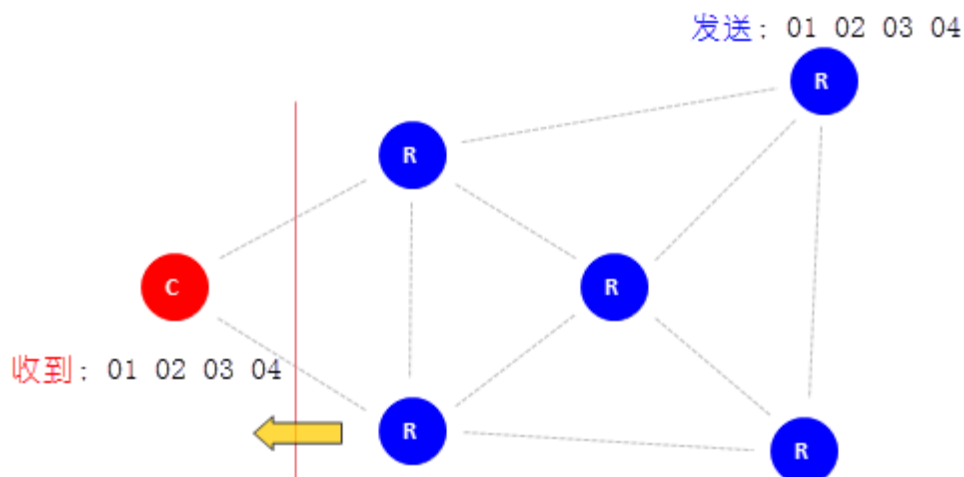


3.1 数据透传

数据包的第一个字节不是 0XFC,0XFD,0XFE, 则模块自动进入串口透传模式。协调器从串口接收到的数据, 会自动发送给所有在网络中的节点, 网络中某个节点从串口接收到数据, 会自动发送给协调器。我们建议用户在自己的数据包前面加上包头。在数据透传模式下, 最大数据包长度不能超过 80 个字节。但建议每个数据包应在 32 个字节内。



由协调器发送的数据, 自动发送到网络中的所有设备



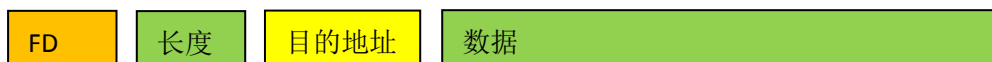
由路由器发送的数据，自动发送到协调器

3.2 点对点传输数据

点对点数据传输即网络中任意两个设备之间互相发送数据。点对点数据传输需要通过寻址的方式确定目标节点。可以通过自定义地址寻址，也可以通过网络地址寻址。即使协调器离开网络，只要网络中有路由设备存在，点对点数据传输依然可以在节点之间进行。以下详细讲解通过网络地址寻址的方式发送数据。

发送数据格式：

点对点帧头（FD）+ 发送数据长度（1 个字节）+ 目标地址（zigbee 网络地址，低字节在前，高字节在后）+ 数据（最多 32 个字节，超出部分丢弃）



例如：

发送：FD 0A 4C CB 01 02 03 04 05 06 07 08 09 0A

FD：点对点数据传输指令

0A：数据域长度

4C CB：目标地址

01 02 03 04 05 06 07 08 09 0A：要发送的数据

接收数据格式：

点对点帧头（FD）+ 发送数据长度（1 个字节）+ 目标地址（zigbee 短地址，低字节在前，高字节在后）+ 数据（最多 32 个字节）+ 发送端短地址（低字节在前，高字节在后）



例如：FD 0A 4C CB 01 02 03 04 05 06 07 08 09 0A 04 1B

FD：点对点数据传输指令

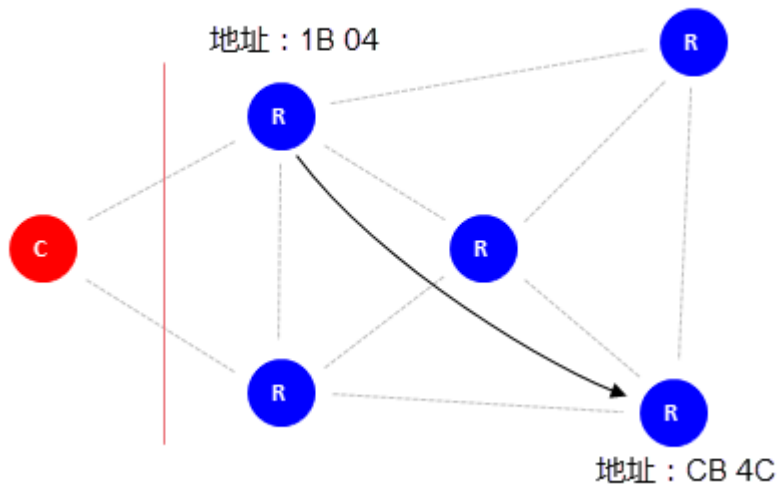
0A: 数据域长度

4C CB: 目的地址（对接收方来说，即是自己短地址）

01 02 03 04 05 06 07 08 09 0A : 接收到的数据

04 1B: 数据发送端地址

如下图：展示了 Zigbee 短地址寻址的点对点数据传输模型。



3.3 点对多点数据传输

点对多点发送即网络中一个节点向网络中某几个设备发送数据，点对多点发送需要这些设备有相同的自定义地址。发送设备的发送模式需要选择 05（Zigbee 自定义地址寻址，不含包头包尾）。

3.4 广播发送数据

广播发送即网络中的一个节点发送一个数据包，网络中的所有设备都接收到。一个设备想广播发送一个数据包，只要按照 3.2 节点点对点传输数据的格式发送，只是，目的地址是 FFFF 即可。

如：发送：FD 0A FF FF 01 02 03 04 05 06 07 08 09 0A

3.5 外设控制

Z15x 系列模块，在串口无线透传数据的基础上，我们加入了一些特色功能，在某些特定的场合下，用户可以省去自己的 MCU，只要通过简单的 AT 指令，即可实现一些控制。外设控制主要包括：

- 本地 GPIO 控制
- 远程 GPIO 控制
- 本地 ADC 采集

- 远程 ADC 采集
- 本地 PWM 波输出
- 远程 PWM 波输出

本地 GPIO 控制

本地 GPIO 控制，是通过串口输入 AT 指令，控制该模块的 GPIO 引脚的状态。每个模块共有 8 路 GPIO，分别是 GPIO0~GPIO7，控制包括：

- 读取模块 GPIO 引脚的输入输出状态
- 设置模块 GPIO 引脚的输入输出状态
- 读取模块 GPIO 引脚的电平状态
- 设置模块 GPIO 引脚的电平状态（注：需要先将模块引脚设置为输出状态。）

以下做详细的设置演示。测试模块的 GPIO0 引脚。（用户需要将模块和上位机连接。此处通过 XShell 4 串口调试软件发送 AT 指令）。

1：打开 XShell 4，将串口连接到模块。

2：连续输入+++，将模块进入 AT 指令模式。

3：通过 AT+IODIR 指令，设置 GPIO0 口为输出状态：AT+IODIR=0,1<CR>（此时，GPIO0 引脚为输出状态，默认电平状态为低电平）。

4：通过 AT+GPIO 指令，设置 GPIO0 口电平为高电平：AT+GPIO=0,1<CR>

```
Connection closed.
Disconnected from remote host(C1COM25) at 16:25:13.
Type `help' to learn how to use Xshell prompt.
[c:\~]$
Connecting to COM25...
Connected.
+OK
AT+IODIR=0,1
+OK
AT+GPIO=0,1
+OK
```

GPIO0已经设置成输出

GPIO0已经输出高电平

远程 GPIO 控制

远程 GPIO 控制，是通过网络中的某个节点 A，对网络中其他的节点 B 的 GPIO 进行控制的一个过程，控制指令从节点 A 的串口输入，作用在网络中节点 B 上。这样，实现了远程的 GPIO 无线控制。控制包括：

- 读取远程模块引脚的输入输出状态
- 设置远程模块引脚的输入输出状态（输入或者输出）

- 读取远程模块引脚的电平状态
- 设置远程模块引脚的电平状态（高电平或者低电平）

注：在设置远程模块引脚的电平状态之前，需要先将模块的 GPIO 引脚设置为输出状态。
具体请参考指令：AT+RGPIO。

本地 ADC 采集

本地 ADC 采集，是通过串口发送固定的指令，来采集该模块指定引脚上的模拟电压值，模块的 ADC 引脚和 PWM 引脚存在公用，默认情况下 PWM 波处于关闭状态。用户可以直接进行 ADC 采集。采集的数值及转换成电压值，请参考下文 AT 指令说明中的 AT+ADC。

远程 ADC 采集

远程 ADC 采集，是通过网络中的某个节点 A，对网络中另外一个节点 B 指定引脚的电压采集，采集指令从节点 A 的串口输入，作用在网络中的节点 B 上，再从节点 A 的串口返回节点 B 的被采集引脚的电压值。

本地 PWM 输出

本地 PWM 波输出，是通过串口发送固定的指令，控制本模块对应的引脚输出 PWM 波，每个模块有 3 路 PWM 波输出，分别是 PWM1，PWM2，PWM3。对 PWM 波的操作有：

- 开启 PWM 波（注：需要先关闭该模块的 485 流控制。）
- 关闭 PWM 波
- 读取 PWM 波信息（周期和占空比）
- 设置 PWM 波信息（周期和占空比）

远程 PWM 输出

远程 PWM 波输出，是通过网络中某个节点 A，控制网络中另外一个节点 B 的 PWM 波输出。控制指令从节点 A 串口输入，作用在网络中节点 B 上。控制包括：

- 开启节点 B 的 PWM 波。（注：需要先关闭节点 B 的 485 流控制。否则，开启会失败。暂不支持远程关闭流控制）。
- 关闭节点 B 的 PWM 波。
- 读取节点 B 的 PWM 波信息（周期和占空比）
- 设置节点 B 的 PWM 波信息（周期和占空比）

具体操作请参考下文 AT 指令之 AT+RPWM。

本地设置 GPIO 口中断触发参数

GPIO 口中断触发是指 ZIGBEE 网络中，节点 A 的某个 GPIO（GPIO0~GPIO7 任意一个）口

有电平变化时，其能通知网络中的另外一个节点 B，在节点 B 的串口打印出触发信息。

其中，触发参数包括：

- en: IO 口中断触发使能。
- pin: 具体 GPIO 引脚。GPIO0~GPIO7
- trigger: 引脚触发方式。有上升沿触发和下降沿触发。
- send: 触发信息发送方式。有目标地址发送。广播发送两种。
- fixedaddr: 目标地址(网络地址)。当 send 为广播方式发送时。Fixedaddr 固定为 0xffff。不可修改。

不论设置哪种触发方式（上升沿或者下降沿），GPIO 口默认高电平，如果是下降沿触发时，将 GPIO 口电平拉低，其立刻发出触发信息。如果是上升沿触发时，将 GPIO 口拉大，其不会发出触发信息，等到拉低动作抬起时，其发出触发信息。

具体配置请参考 AT+IOINT 指令。

远程设置 GPIO 口中断触发参数

远程设置 GPIO 口中断触发参数是：在 ZIGBEE 网络中任意一个节点 A 上，设置另外一个节点 B 的 GPIO 口触发参数。包括：en,pin,trigger,send。(注：不可以设置 fixedaddr)。在设置之前，需要通过 AT+DSTADDR 指令，确定 A 节点欲设置的 B 节点的网络地址。

当用户通过 A 节点设置 B 节点 send 参数后，如果设置目标地址发送，则 B 节点的 fixedaddr 将被默认修改成 A 节点的网络地址。也就是当 B 节点的 GPIO 口有触发时，将向 A 节点发送触发信息。

触发信息报告格式

当配置了网络中某个节点 B，其 GPIO 口有电平触发时，其向节点 A 报告。报告格式如下：（以下信息在节点 A 的串口打印出来。）

+RIOINT=GPIO, 0, MID<CR><LF>

GPIO 表示节点 B 的触发 GPIO 引脚。可能的值有 0~7，对应表示 GPIO0~GPIO7。

0: 自定义标识。暂时保留。

MID: 节点 B 的 MID。MID 的值可以通过 AT+MID 修改。具体请参考 AT+MID 指令。

3.6 属性设置

Z15x 模块网络的打开与关闭功能。

Z15x 模块工作在协调器或者路由器模式下，可以设置打开或者关闭网络。默认情况下，网络是打开的，用户可以通过 AT 指令关闭网络。对于路由器关闭网络之后，新的设备不能挂载到该设备上，关闭网络之前已经挂载到该设备上的设备依然可以和该设备正常通信。对于协调器，关闭其网络后，会自动连带网络中所有的路由器一同关闭网络。这个功能主要是用于保护固定的网络。防止其他人员恶意将新设备加入网络。

Z15x 模块网络环境清除

Z15x 模块在加入网络后，都会保存当前网络参数，这样带来的好处是模块断电重启后，入网速度非常快，因为其省去了扫描网络，加入网络这一个过程。是直接恢复保存的网络环境。如果需要清除网络环境，模块以新的身份扫描加入网络，可以在 AT 指令模式下，发送 AT+CLE 来清除网络环境。

Z15X 模块恢复出厂设置

Z15X 模块有两种方式恢复出厂设置，第一种是在命令行模式下，通过串口发送 AT+Z 恢复出厂设置。第二种是外部手动恢复出厂设置，将 nReload 引脚短接到 GND 超过 3 秒即可恢复出厂设置。

3.6 设置数据轮训时间

终端模式下，模块通过休眠的方式来降低功耗。模块睡眠时，如果有数据发送给该模块，数据将被缓存在该模块的父节点中，模块根据用户设定数据轮训时间，周期性的从父节点轮训数据，该轮训的时间间隔是可以设置的。终端节点接收到数据的延时时间小于设定的轮训时间。

3.7 终端模式串口唤醒模块

终端模式下，模块通过休眠的方式来降低功耗。休眠状态中，串口也是不能正常通信的，用户只需要发送任意一个数据包到串口，模块立即从睡眠状态中唤醒。该帧数据包将被丢掉。唤醒后，串口在串口超时时间内（该时间长度可设置，请参考 AT 指令 AT+SLEEP）无数据传输，模块再次进入休眠状态。

3.8 终端模式下数据接收与发送

当设备的器件类型被设置成终端时，设备就进入休眠状态，休眠状态下，模块是不可以收发数据的，当一个设备向该终端发送数据时，数据先缓存在终端设备的父设备中。终端设备定时轮询父设备（轮询设置请参考 3.7 节）。将数据读取下来。

如果外部 MCU 需要通过终端节点向网络中发送数据，首先需要唤醒该模块，我们抛弃以往使用 GPIO 唤醒的常规做法，采用更加便捷的串口唤醒，客户 MCU 只需要在发送数据前，先发送任意的一帧数据到串口，即可唤醒模块。在设定的时间内无串口输入数据时，又会自动进入休眠状态，无需外部 MCU 干预，用户可根据需要设定超时时间（参考指令表中的 AT+SLEEP），以达到不同的功耗级别。

3.9 网络中更换协调器

(等待更新)。

3.10 查询节点的信号强度

Z15 系列模块支持查询节点之间的信号强度。假设节点 A 和节点 B，节点 A 向节点 B 发送一条消息，该无线消息到达节点 B 处还剩余的信号强度支持可查询。信号强度单位 dbm。其绝对值越大，信号强度越小。

查询指令：AT+RSSI

查询方法：

将上述 A，B 两个节点组在一个网络内。

上述节点 B 进入 AT 模式下，输入 AT+RSSI=1<CR>。然后退出 AT 模式。

从节点 A 向节点 B 发送一条任意消息，这个时候，在节点 B 的串口打印出来的不是几点 A 发送的信息，而是+RSSI=value<CR><LF>。其中，value 是信号强度值。value 绝对值越大，信号强度越小。

Z15 系列模块接收灵敏度为-93dbm 用户需要保证 RSSI 值在-85 以上，可以保证数据的接收质量。

四、AT 指令说明

用户对模块进行配置参数时，首先需要进入命令行模式。进入命令行模式后，即可发送 AT 指令，进行读取或者设置模块的属性和参数。设置完毕后，需要退出命令行模式。发送 AT+Q 或者重启模块，即会退出命令行模式。

+++	<u>进入命令行模式</u>
AT+Q	<u>退出命令行模式</u>
AT+R	<u>重启模块</u>
AT+Z	<u>恢复出厂设置</u>
AT+E	<u>查询/设置命令回显</u>
AT+MID	<u>查询/设置模块自定义地址</u>
AT+VER	<u>查询固件版本号</u>
AT+SN	<u>查询序列号</u>
AT+MAC	<u>查询 MAC 地址</u>
AT+UART	<u>查询/设置串口</u>
AT+ROLE	<u>查询/设置设备类型</u>
AT+MODE	<u>查询/设置通信模式</u>
AT+POLL	<u>查询/设置终端节点轮训时间</u>
AT+NWK	<u>查询/设置网络地址及网络开关</u>
AT+CLR	<u>清除网络环境</u>
AT+STAT	<u>查询模块状态</u>
AT+PRT	<u>查询父设备信息</u>

AT+PAN	查询/设置 ZigBee 网络标识
AT+CH	查询/设置信道
AT+CHEX	查询/设置信道（扩展）
AT+PWM	查询/设置本地 PWM
AT+RPWM	查询/设置远程 PWM
AT+ADC	查询/设置本地 ADC
AT+RADC	查询/设置远程 ADC
AT+GPIO	查询/设置本地 GPIO
AT+RGPIO	查询/设置远程 GPIO
AT+SLEEP	查询/设置休眠状态及串口超时时间
AT+SCR	查询/设置安全网络
AT+KEY	查询/设置密钥
AT+ERR	查询/清除模块错误码
AT+DL	查询/设置唤醒用户 MCU 等待时间
AT+DSTADDR	查询/设置远程目标地址
AT+IODIR	查询/设置本地 GPIO 输入输出状态
AT+R IODIR	查询/设置远程 GPIO 输入输出状态
AT+HDT	查询/设置终端数据缓存时间
AT+IOINT	查询/设置本地 GPIO 口中断触发参数
AT+RIOINT	查询/设置远程 GPIO 口中断触发参数
AT+RSSI	查询信号强度

进入命令行模式

输入：+++

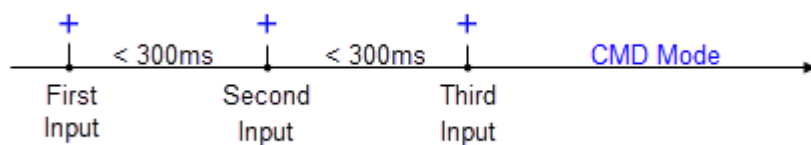
返回：<CR><LF>+OK <CR><LF>

说明：AT 指令是指在命令行模式下用户可通过 UART 与模块进行参数设定的命令集合。

模块的缺省 UART 接口参数为：波特率 115200，无检验，8 位数据位，1 位停止位。

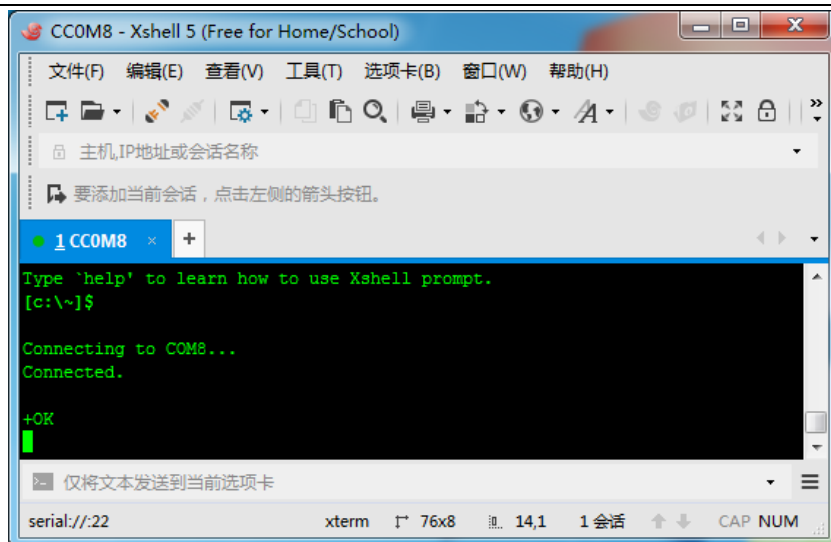
从透传模式下切换到命令模式需要按照依据如下规则：

UART 上连续输入：“+++”，每个符号“+”时间间隔小于 300ms，即可进入命令行模式



“+”输入示意

推荐使用 XShell5 串口调试工具，或者其他专业的串口调试工具



XShell 进入命令行界面

指令格式

AT 指令采用 ASCII 码格式的字符串

设置命令格式如下：

AT+CMD=param-1,param-2,...,param-n<CR>

- ◇ AT+: 指令前缀
- ◇ CMD: 具体的指令字符串，全部大写。
- ◇ param-1: 参数 1
- ◇ param-2: 参数 2
- ◇ param-n: 参数 n
- ◇ <CR>: 命令结束符，对应回车符，ASCII 码为 0x0A

查询格式如下：

AT+CMD<CR>

查询命令返回

<CR><LF>+CMD=param-1, param-2,..., param-n<CR><LF>

- ◇ param-1: 参数表示的意义，见每个指令的说明。

注：远程信息查询，为非同步返回，需要一点时间之后，才可能有返回。

设置命令返回（执行正确）：

<CR><LF>+OK<CR><LF>

设置命令返回（执行错误）：

<CR><LF>+ERR=errorcode<CR><LF>

- ◇ <CR><LF>: 换行符，ASCII 码为 0x0A,0x0D，<CR><LF>共同组成换行符
- ◇ errorcode: 错误码，-1 表示命令错误，-2 表示参数错误，-3 表示不允许的操作，-4 表示命令执行超时。

注意事项：

英文符号：加号'+', 等号 '=', 逗号 ',' 是 AT 命令特殊符号，自定义字符串参数不能携带。

小技巧：

设置参数时，灰色方括号中的参数([param-n])，意为可以忽略不设置的参数。

例如串口参数命令：AT+UART=baudrate,databits,stopbits,parity,flowcontrol

如果只设置波特率，可写成：

AT+UART=baudrate<CR>

后续的参数可忽略，如果只想设置 parity 校验位参数，可写成：

AT+UART=,,,parity

parity 前的参数可不写，只携带对应数量的逗号","即可，使用非常灵活。

退出命令行模式

指令：AT+Q

输入：AT+Q<CR>

返回：<CR><LF>

说明：输入此指令，模块就会从 AT 模式下退出到透传模式。

补充：在 AT 指令模式下，模块可以正常接收数据。但是，不能发送数据。模块不会自动退出 AT 指令模式，除非发送退出指令，或者发送重启模块指令，或者，硬件上断电重启模块。

重启模块

指令：AT+R

输入：AT+R <CR>

返回：<CR><LF>+OK <CR><LF>

说明：模块将在一秒后重新启动。

补充：设置完某些需要重启后才会生效的参数后，可以通过该指令控制其重新启动。

恢复出厂设置

指令：AT+Z

输入：AT+Z <CR>

返回：<CR><LF>+OK <CR><LF>

效果：一秒后，以出厂参数重新启动。此操作将擦除所有客户设置的参数。

查询/设置命令回显

指令：AT+E

[查询命令回显](#)

输入：AT+E <CR>

返回：<CR><LF>+E=status <CR><LF>

说明：status 为回显状态，取值范围：0 和 1。0 表示关闭。1 表示打开。

解释：命令回显是输入时及时显示输入的内容即为回显，关闭回显，则看不到串口输入的内容，只能看到命令执行后返回的数据。

[设置串口回显（默认打开，此处示例关闭）](#)

输入：AT+E= status <CR>

返回：<CR><LF>+OK <CR><LF>

说明：status：取值范围：0 和 1。0 表示关闭。1 表示打开。

查询/设置模块自定义地址

指令：AT+MID

[查询模块自定义地址](#)

输入：AT+MID <CR>

返回：<CR><LF>+MID=id <CR><LF>

说明: id 为 ZigBee 自定义地址, 长度为两个字节的十六进制数据。如 0XA768, 取值范围: 0x0000~0xFFFF。

设置模块自定义地址

输入: AT+MID =1234<CR>或者 AT+MID=0x5678<CR>

返回: <CR><LF>+OK<CR><LF>

说明: 1234 是 10 进制数据, 对应的 16 进制数据为 0x04D2, 若加前缀 0x, 则 0x5678 则为 16 进制数据

补充: 模块默认的自定义地址是 0xFFFF, 模块的自定义地址是非易失性的, 不会因为重启而改变。相同的自定义地址的两个模块在网络中不会产生冲突。

查询固件版本号

指令: AT+VER

输入: AT+VER <CR>

返回: <CR><LF>+VER=version<CR><LF>

说明: version 是固件的版本号。此指令只读。

查询序列号

指令: AT+SN

输入: AT+SN<CR>

返回: <CR><LF>+SN=serialnumber<CR><LF>

说明: serialnumber 模块序列号, 具有唯一性。

查询 MAC 地址

指令: AT+MAC

输入: AT+MAC<CR>

返回: <CR><LF>+MAC=mac<CR><LF>

说明: mac 是模块的 MAC 地址。带分号的 16 进制数据, 例如: 00:12:4B:00:0B:F7:1A:31。

查询/设置串口

指令: AT+UART

查询串口参数

输入: AT+UART<CR>

返回: <CR><LF>+UART=[baudrate],[databits],[stopbits],[parity],[flowcontrol] <CR><LF>

说明: baudrate:波特率

databits:数据位 (仅支持 8 位, 不可修改)

stopbits:停止位

parity:校验位: NONE (无), ODD (奇校验), EVEN (偶校验)

flowcontrol:流控制: OFF (关闭), ON (开启), 485 (485 控制)

设置串口参数

输入: AT+UART= [baudrate],[databits],[stopbits],[parity],[flowcontrol] <CR>

返回: <CR><LF>+OK<CR><LF>

说明: baudrate: 串口的波特率。取值范围: 9600,19200,38400,57600,115200。

databits:数据位 (仅支持 8 位, 不可修改)

stopbits: 停止位。取值范围: 1, 2。

parity: 校验位。取值范围: NONE(无校验), ODD(奇校验), EVEN(偶校验)。

flowcontrol: 流控制和 485 传输开关。取值范围: ON(流控制开), OFF(流控制关), 485(485 通信开)。默认开启 485 通信。

补充: 更改串口参数后, 需要重启模块才会生效。

查询/设置设备类型

指令: AT+ROLE

查询设备类型

输入: AT+ROLE<CR>

返回: <CR><LF>+ROLE=type<CR><LF>

说明: type 代表设备类型。取值范围: 0 (协调器), 1 (路由器), 2 (终端节点)

设置设备类型 (以协调器为例, 出厂默认是路由器)

输入: AT+ROLE=type <CR>

返回: <CR><LF>+OK<CR><LF>

说明: 需要重启后生效。

查询/设置通信模式

指令: AT+MODE

查询通信模式

输入: AT+MODE<CR>

返回: <CR><LF>+MODE=type<CR><LF>

说明: type 是模块当前工作模式, 其值存在 6 种情况。模块默认的通信模式是 0, 其他通信模式请参照下表:

传输模式值	数据透传方式下	点对点传输方式下
0	数据透传	Zigbee 网络地址寻址, 含包头包尾
1	数据透传+Zigbee 网络地址	Zigbee 网络地址寻址, 含包头包尾
2	数据透传+MAC 地址	Zigbee 网络地址寻址, 含包头包尾
3	数据透传+自定义地址	Zigbee 网络地址寻址, 含包头包尾
4	数据透传	Zigbee 网络地址寻址, 不含包头包尾
5	数据透传	Zigbee 自定义地址寻址, 不含包头包尾

设置通信模式 (以传输模式 1 为例)

输入: AT+MODE=1<CR><LF>

返回: <CR><LF>+OK<CR><LF>

补充: 传输模式只对本设备有效, 如: 一个网络中有 A,B,C 三台设备, A 的传输模式是 1, B 和 C 设备传输模式是 0, 那么通过 A 设备发送出去的数据是在数据的基础上加上 A 设备的 ZigBee 短地址, 通过 B 和 C 设备发送出去的数据依然是纯粹的数据。

查询/设置终端节点轮询时间

指令: AT+POLL

查询终端节点轮询时间

输入: AT+POLL<CR>

返回: <CR><LF>+POLL=time<CR><LF>

说明: time 是终端节点轮询的时间, 单位是秒, 默认 5 秒。

设置终端节点轮询时间

输入: AT+POLL=time<CR>

返回: <CR><LF>+OK<CR><LF>

说明: time 是轮询的时间间隔, 取值范围是[1,60]。轮询时间是终端定时到父设备中查询数据。当一个设备发送一帧数据给终端设备, 该帧数据先缓存在该终端设备的父设备中。终端通过定时轮询的方式, 读取父设备中的数据。

查询/设置网络地址及网络开关

指令: AT+NWK

查询网络地址和状态。

输入: AT+NWK<CR>

返回: <CR><LF>+NWK=addr,status<CR><LF>

说明: addr 是网络地址 (也叫短地址)。2 个字节长的 16 进制数据。

status 是网络开关。取值范围: 1 (打开) 和 0 (关闭)。

设置网络地址

输入: AT+NWK=addr,status<CR>

返回: <CR><LF>+OK<CR><LF>

补充: 我们建议网络地址不要修改, 如果网路中已经存在新修改的地址时, 将会发生冲突。模块加入网络后, 网络会自动给其分配网络地址, 特定情况下, 网络地址存在改变的可能。路由器网络的状态打开或者关闭是该设备是否接受挂载新的设备。协调器网络状态打开或者关闭是整个网络是否接受新的设备加入。在关闭网络之前已经加入网络的设备, 将不受影响。

清除网络环境

指令: AT+CLR

输入: AT+CLR<CR>

返回: <CR><LF>+OK<CR><LF>

说明: 模块将在一秒后自动重启重新搜索网络。

查询模块状态

指令: AT+STAT

输入: AT+STAT<CR>

返回: <CR><LF>+STAT=status<CR><LF>

说明: status 是状态。其值有 11 种可能。该指令为只读指令。返回参数与状态请对照下表。

数值	描述
0	初始化
1	初始化
2	网络查找
3	正在加入一个网络
4	再次加入 (只对终端设备有效)
5	终端启动, 但没有被认证 (只对安全网络)
6	终端启动
7	路由设备启动
8	协调器启动

9	协调器启动
10	孤点（丢失父设备）

查询父设备信息

指令：AT+PRT

输入：AT+PRT<CR>

返回：<CR><LF>+PRT= addr,mac <CR><LF>

说明：addr 为父设备网络地址。2 个字节的 16 进制数据，例如 0x4581

（如果父设备地址是 0xFFFE,表示该父设备没有加入到网络中）。

mac, 父设备的 MAC 地址，带分号的 16 进制数据，例如：00:12:4B:00:0B:F7:1A:31

查询/设置 ZigBee 网络标识

指令：AT+PAN

查询模块的 PANID 和 expandid

输入：AT+PAN<CR>

返回：<CR><LF>+PAN= panid,expandid<CR><LF>

说明：panid: ZigBee 网络标识 PANID，2 个字节长的 16 进制数据。

expandid: ZigBee 的扩展网络标识 ExtendPANID，字符串形式。

设置模块的网络标识

输入：AT+PAN=[panid],[expandid] <CR>

返回：<CR><LF>+OK<CR><LF>

说明：panid: ZigBee 网络标识 PANID，2 个字节长的 16 进制数据，例如 0xA76B，取值范围：0x0000~0xFFFE

expandid: ZigBee 的扩展网络标识 ExtendPANID，字符串形式。长度为 1~7 之间的任意字符串。取值为 NONE 时表示忽略此参数，仅使用 PANID 作为网络标识

补充：参数重启后生效，若模块为协调器，表示建立以此参数为网络标识的 ZigBee 网络，该参数类似 WIFI 路由器的网络名 SSID。；若模块为路由器或者终端，表示要加入以此为网络标识的 ZigBee 网络中去。

查询/设置信道

指令：AT+CH

查询无线通信信道

输入：AT+CH<CR>

返回：<CR><LF>+CH= channel <CR><LF>

说明：channel 为当前工作信道，默认 14 信道。

设置信道

输入：AT+CH= channel <CR>

返回：<CR><LF>+OK<CR><LF>

说明：channel 为指定的工作信道，取值范围：11~26。

补充：重启之后，就会工作在该信道上。如果设备类型是协调器就会在该信道上创建网络，如果是路由器就会在该信道上寻找网络。

查询/设置信道(扩展)

指令：AT+CHEX

查询无线通信信道

输入: AT+CHEX<CR>

返回: <CR><LF>+CHEX= channel_bits,channel <CR><LF>

说明: channel_bits 为当前设定的信道比特位, 默认 0x00004000 信道。

channel 为前工作的信道。为 0, 表示当前还没有加入网络。

设置信道

输入: AT+CH= channel_bits <CR>

返回: <CR><LF>+OK<CR><LF>

说明: channel_bits 为指定的工作信道, 可以设定多个信道 (一个信道占一个比特位)。

补充: 重启之后, 就会工作在该信道上。如果设备类型是协调器就会在该信道上创建网络, 如果是路由器就会在该信道上寻找网络。

查询/设置本地 PWM

指令: AT+PWM

查询本地模块当前 PWM 信息

输入: AT+PWM<CR>

返回: <CR><LF>+PWM=status,t,pwm1,pwm2,pwm3<CR><LF>

说明: status: 表示当前 PWM 状态。取值范围: 1(打开),0(关闭)。

t:表示当前 PWM 周期。

pwm1:表示 PWM1 通道高电平时间, PWM1 除以 T, 得到 PWM1 通道的占空比。

pwm2:表示 PWM2 通道高电平时间, PWM2 除以 T, 得到 PWM2 通道的占空比。

pwm3:表示 PWM3 通道高电平时间, PWM3 除以 T, 得到 PWM3 通道的占空比。

设置本地模块 PWM 参数

输入: AT+PWM=[status],[t],[pwm1],[pwm2],[pwm3] <CR>

返回: <CR><LF>+OK<CR><LF>

说明: status:PWM 的状态。取值范围: 0(关闭 PWM), 1(开启 PWM)。

t:PWM 波的周期。取值范围: [0x0002,0xFFFF]。

pwm1: PWM1 通道高电平的时间。取值范围: [0x0001,0xFFFF]。值要小于 T。

pwm2: PWM2 通道高电平的时间。取值范围: [0x0001,0xFFFF]。值要小于 T。

pwm3: PWM3 通道高电平的时间。取值范围: [0x0001,0xFFFF]。值要小于 T。

补充: 如果设置开启 PWM 时返回<CR><LF>+ERR=-2<CR><LF>, 请检查是否已经关闭了 UART 的 485 控制。

t,pwm1,pwm2,pwm3 所对应的值转换成具体的时间方法。这里以计算周期 t 为例。

$\text{Time(s)} = (1/32M) * t$ 。(t 的值需要转换成 10 进制)。

实算: 默认情况下, $t = 0x0fff$, 转换成 10 进制为 4095。所以周期为 $(1/32000000) * 4095 = 0.0001279$ (秒)。约等于 128uS。

查询/设置远程 PWM

指令: AT+RPWM

查询远程模块当前 PWM 信息

输入: AT+RPWM<CR>

返回: <CR><LF>+RPWM=status,t,pwm1,pwm2,pwm3<CR><LF>

说明: status: 远程模块 PWM 的状态。1 (PWM 已经打开), 0 (PWM 关闭状态)。

t: 远程模块 PWM 的周期。

pwm1: PWM1 通道的高电平时间。

pwm2: PWM2 通道的高电平时间。

pwm3: PWM3 通道的高电平时间。

补充: 在查询或者设置远程 PWM 之前, 需要先通过 AT+DSTADDR 指令指定远程的目标地址。否则, 模块按照默认地址查询。

设置远程模块 PWM 信息

输入: AT+RPWM=[status],[t],[pwm1],[pwm2],[pwm3] <CR>

返回: <CR><LF>+OK<CR><LF>

说明: status: PWM 的状态。取值范围: 0(关闭远程 PWM), 1(开启远程 PWM)。

t: 远程模块 PWM 的周期。取值范围[0x0002,0xFFFF]。

pwm1: PWM1 通道的高电平时间。取值范围: [0x0001,0xFFFF]。值要小于 T。

pwm2: PWM2 通道的高电平时间。取值范围: [0x0001,0xFFFF]。值要小于 T。

pwm3: PWM3 通道的高电平时间。取值范围: [0x0001,0xFFFF]。值要小于 T。

补充: 设置远程 PWM 之前, 需要确保远程目标节点 UART 的 485 控制是关闭状态。
该远程指令作用在由 AT+DSTADDR 所指定的目标节点上。

查询/设置本地 ADC

指令: AT+ADC

读取模块指定通道 ADC 的值

输入: AT+ADC=com<CR>

返回: <CR><LF>+ADC= com,value<CR><LF>

说明: com: 是 ADC 通道, 取值范围是[0,7]。对应 ADC0~ADC7。

value: 是 ADC 的值, 值是两个字节的十六进制表现形式。用户需要根据返回数值计算出引脚的电压值。参考电压为电源电压, 默认为 3.3V。电压值=返回的数值(转换成十进制)/8191*3.3。

举例: 上述中, 假设 value 的值是 0x0231, 转换成十进制是 561, 所以, 电压值=561/8191*3.3=0.226V。

查询/设置远程 ADC

指令: AT+RADC

读取指定模块指定通道 ADC 值

输入: AT+RADC=com<CR>

返回: <CR><LF>+RADC=com,value<CR><LF>

说明: com: 是 ADC 通道, 取值范围是[0,7]。对应 ADC0~ADC7。

value: 是 ADC 的值, 值是两个字节的十六进制表现形式。转换成电压参考本地。

补充: 该远程指令作用在由 AT+DSTADDR 所指定的目标节点上。

查询/设置本地 GPIO 输入输出状态

指令: AT+ IODIR

查询本地全部 GPIO 引脚输入输出状态

输入: AT+ IODIR <CR>

返回: <CR><LF>+IODIR=status<CR><LF>

说明: status: 是引脚输入输出状态参量, 其值为一个字节长度的十六进制数据, 每一

位代表一个 GPIO 引脚，从右向左依次是 GPIO0 到 GPIO7。如果该位值是 0，则代表该位对应的 GPIO 引脚是输入，反之，输出。例如：status 为 0x07，0x07 转换成二进制是 0000 0111，其意义是 GPIO0,GPIO1,GPIO2 为输出，其他 GPIO 为输入。

查询本地指定 GPIO 引脚输入输出状态

输入：AT+IODIR=pin<CR>

返回：<CR><LF>+IODIR=status<CR><LF>

说明：pin：具体的 GPIO 口，取值范围：[0,7]。对应 GPIO0~GPIO7。

status：当前该 GPIO 口的输入输出状态。0 表示输入，1 表示输出。

设置本地指定 GPIO 引脚输入输出状态

输入：AT+IODIR=pin, status<CR>

返回：<CR><LF>+OK<CR><LF>

说明：pin：具体的 GPIO 口，取值范围：[0,7]。对应 GPIO0~GPIO7。

status：当前该 GPIO 口的输入输出状态。0 表示输入，1 表示输出。

补充：模块引脚默认是输入状态。

查询/设置远程 GPIO 输入输出状态

指令：AT+R IODIR

查询远程全部 GPIO 引脚输入输出状态

输入：AT+R IODIR <CR>

返回：<CR><LF>+RIODIR=status<CR><LF>

说明：status：是引脚输入输出状态参量，其值为一个字节长度的十六进制数据，每一位代表一个 GPIO 引脚，从右向左依次是 GPIO0 到 GPIO7。如果该位值是 0，则代表该位对应的 GPIO 引脚是输入，反之，输出。例如：status 为 0x07，0x07 转换成二进制是 0000 0111，其意义是 GPIO0,GPIO1,GPIO2 为输出，其他 GPIO 为输入。

查询远程指定 GPIO 引脚输入输出状态

输入：AT+R IODIR =pin<CR>

返回：<CR><LF>+RIODIR=pin,status<CR><LF>

说明：pin：远程模块具体的 GPIO 口，取值范围：[0,7]。对应 GPIO0~GPIO7。

status：引脚的输入输出状态。0 表示输入，1 表示输出。

设置远程指定 GPIO 引脚输入输出状态

输入：AT+R IODIR =pin,status<CR>

返回：<CR><LF>+OK<CR><LF>

说明：pin：远程模块具体的 GPIO 口，取值范围：[0,7]。对应 GPIO0~GPIO7。

status：引脚的输入输出状态。0 表示输入，1 表示输出。

补充：模块引脚默认是输入状态。

查询/设置本地 GPIO

指令：AT+GPIO

查询本地全部 GPIO 引脚电平状态

输入：AT+GPIO<CR>

返回：<CR><LF>+OK=status<CR><LF>

说明：status：表示引脚的电平状态，其值是一个字节长度的十六进制数据，每位代表一个 GPIO 引脚，从右向左依次是 GPIO0 到 GPIO7。如果该位值是 0，则代表该位对应的 GPIO 口电平是低电平。反之，是高电平。如：status 为 0x07,0x07 转换成二进制是 0000 0111，其

意义是 GPIO0, GPIO1, GPIO2 为高电平。其他 GPIO 为低电平。

查询本地指定 GPIO 引脚电平状态

输入: AT+GPIO=pin<CR>

返回: <CR><LF>+GPIO=pin,status<CR><LF>

说明: pin: 远程模块具体的 GPIO 口, 取值范围: [0,7]。对应 GPIO0~GPIO7。

status: 引脚的电平状态。0 表示低电平, 1 表示高电平。

设置本地指定 GPIO 引脚高低电平

输入: AT+GPIO=pin,value<CR>

返回: <CR><LF>+OK<CR><LF>

说明: pin: 是对应的 GPIO 引脚号, 取值范围是[0,7]。对应 GPIO0~GPIO7。

value: 欲设置的电平值, 取值范围: 0 和 1。0 代表低电平, 1 代表高电平。

补充: 设置高低电平状态之前, 要确保引脚被设置成输出状态。设置成输出状态后, 默认高电平。

查询/设置远程 GPIO

指令: AT+RGPIO

查询远程模块全部 GPIO 引脚电平状态

输入: AT+RGPIO<CR>

返回: <CR><LF>+RGPIO=status<CR><LF>

说明: status: 表示引脚的电平状态, 其值是一个字节长度的十六进制数据, 每位代表一个 GPIO 引脚, 从右向左依次是 GPIO0 到 GPIO7。如果该位值是 0, 则代表该位对应的 GPIO 口电平是低电平。反之, 是高电平。如: status 为 0x07, 0x07 转换成二进制是 0000 0111, 其意义是 GPIO0, GPIO1, GPIO2 为高电平。其他 GPIO 为低电平。

查询远程模块指定 GPIO 引脚电平状态

输入: AT+RGPIO=pin<CR>

返回: <CR><LF>+RGPIO=pin,status<CR><LF>

说明: pin: 远程模块具体的 GPIO 口, 取值范围: [0,7]。对应 GPIO0~GPIO7。

status: 引脚的电平状态。0 表示低电平, 1 表示高电平。

设置远程指定 GPIO 引脚高低电平

输入: AT+RGPIO=pin,value<CR>

返回: <CR><LF>+OK<CR><LF>

说明: pin: 是对应的 GPIO 引脚号, 取值范围是[0,7]。对应 GPIO0~GPIO7。

value: 欲设置的电平值, 取值范围: 0 和 1。0 代表低电平, 1 代表高电平。

补充: 设置高低电平状态之前, 要确保引脚被设置成输出状态。设置成输出状态后, 默认低电平。(如果返回+ERR..., 很有可能没有将目标节点 GPIO 口设置成输出)。

查询/设置本地 GPIO 口中断触发参数

指令: AT+IOINT

查询本地 GPIO 口中断触发参数

输入: AT+IOINT<CR>

返回: <CR><LF>+IOINT=en,pin,trigger,send, fixedaddr<CR><LF>

说明: en: IO 口中断触发使能。0(表示失能), 1(表示使能)。

pin: 具体 GPIO 引脚。取值范围: 0~7。对应 GPIO0~GPIO7。

trigger: 引脚触发方式。0(表示上升沿触发), 1(表示下降沿触发)。

send: 触发信息发送方式。1(向 fixedaddr 地址发送)。0(广播发送, fixedaddr 自动变成 0xFFFF)。

fixedaddr: 目标地址。该地址是网络地址 (也就是短地址)。

设置本地 GPIO 口中断触发参数

输入: AT+IOINT=[en],[pin],[trigger],[send],[fixedaddr]<CR>

返回: <CR><LF>+OK<CR><LF>

说明: en: IO 口中断触发使能。取值范围: 0(失能), 1(使能)。

pin: 具体 GPIO 引脚。取值范围: 0~7。对应 GPIO0~GPIO7。

trigger: 引脚触发方式。0(表示上升沿触发),1(表示下降沿触发)。

send: 触发信息发送方式。1(向 fixedaddr 地址发送)。0(广播发送, fixedaddr 自动变成 0xFFFF)。

fixedaddr: 目标地址。该地址是网络地址 (也就是短地址)。

查询/设置远程 GPIO 口中断触发参数

指令: AT+RIOINT

查询远程 GPIO 口中断触发参数

输入: AT+RIOINT<CR>

返回: <CR><LF>+RIOINT=en,pin,trigger,send,fixedaddr<CR><LF>

说明: en: IO 口中断触发使能。0(表示失能), 1(表示使能)。

pin: 具体 GPIO 引脚。取值范围: 0~7。对应 GPIO0~GPIO7。

trigger: 引脚触发方式。0(表示上升沿触发),1(表示下降沿触发)。

send: 触发信息发送方式。1(向 fixedaddr 地址发送)。0(广播发送, fixedaddr 自动变成 0xFFFF)。

fixedaddr: 目标地址。该地址是网络地址 (也就是短地址)。

补充: 在该指令输入之前, 需要确保 AT+DSTADDR 指向正确的目标节点。默认情况下, 协调器的 DSTADDR 地址是自身的 0x0000。所以, 用户在没有指定目标节点时, 直接输入 AT+RIOINT 会返回错误。

该指令输入后, 返回的是 AT+DSTADDR 指令指定的节点的 GPIO 口中断触发参数。

设置远程 GPIO 口中断触发参数

输入: AT+RIOINT=[en],[pin],[trigger],[send]<CR>

返回: <CR><LF>+OK<CR><LF>

说明: en: IO 口中断触发使能。0(表示失能), 1(表示使能)。

pin: 具体 GPIO 引脚。取值范围: 0~7。对应 GPIO0~GPIO7。

trigger: 引脚触发方式。0(表示上升沿触发),1(表示下降沿触发)。

send: 触发信息发送方式。1(向 fixedaddr 地址发送)。0(广播发送, fixedaddr 自动变成 0xFFFF)。

补充: 该指令作用在有 AT+DSTADDR 指定的目标节点上。当设置 send 参数为 1 时, 目标节点的 fixedaddr 自动变成该节点的网络地址。

查询/设置休眠状态及串口超时时间

指令: AT+SLEEP

查询休眠状态及串口超时时间

输入: AT+SLEEP<CR>

返回: <CR><LF>+SLEEP=status,time <CR><LF>

说明: status: 休眠状态。取值范围: 0(关闭休眠),1(开启休眠)。

time: 为串口超时休眠时间, 单位为 100ms, 取值范围[0,255]。串口在 time 时间内无任何数据输入, 模块将自动进入休眠状态。该指令只对终端设备有效, 协调器和路由器设备不存在休眠。

[设置休眠状态及串口超时时间。](#)

输入: AT+SLEEP=[status],[time]<CR>

返回: <CR><LF>+OK<CR><LF>

说明: status:休眠状态。取值范围: 0(关闭休眠), 1(开启休眠)。

time: 为串口超时休眠时间, 单位是 100ms,取值范围[0,255]。当串口在 time 时间内无任何数据输入, 模块将自动进入休眠状态。SLEEP 与 PWM 功能相冲突。当 PWM 功能开启时, 不会进入睡眠。

查询/设置安全网络

指令: AT+SCR

[查询模块当前状态是否为加密状态](#)

输入: AT+SCR<CR>

返回: <CR><LF>+SCR=status<CR><LF>

说明: status 是加密状态, 取值范围: 0(关闭加密), 1(开启加密)。

模块默认为非加密状态, 加密的设备和非加密的设备不能同时使用。

[设置模块加密状态。](#)

输入: AT+SCR=status<CR>

返回: <CR><LF>+OK<CR><LF>

说明: status: 加密状态, 取值范围: 0(关闭加密),1(开启加密)。设置加密后, 需要重新启动才能生效。用户需要通过 AT+KEY 指令为在网络中的所有模块配置相同的密钥。

查询/设置安全密钥

指令: AT+KEY

[查询安全网络的密钥](#)

输入: AT+KEY<CR>

返回: <CR><LF>+KEY=key<CR><LF>

说明: key 为安全网络密钥, 字符串形式。长度为 1~15 之间的任意字符串。

[设置密钥](#)

输入: AT+KEY=key<CR>

返回: <CR><LF>+OK<CR><LF>

说明: 只有加密状态开启情况下, 该指令才有效。

查询/清除模块错误码

指令: AT+ERR

[查询模块错误码](#)

输入: AT+ERR<CR>

返回: <CR><LF>+ERR=value<CR><LF>

说明: value: 错误码的值。错误码是模块在执行具体操作失败时, 返回的信息, 给用户提示执行过程中遇到的问题, 方便用户及时解决问题。具体的错误码表如下:

错误码	说明
-----	----

0	没有错误
1	系统参数读取失败
20	更改网络地址失败
30	读远程 ADC 超时
40	远程 PWM 超时
50	远程 GPIO 超时

清除错误码

输入: AT+ ERR=0<CR>

返回: <CR><LF>+OK<CR><LF>

查询/设置唤醒用户 MCU 等待时间

指令: AT+DL

查询唤醒用户 MCU 等待时间

输入: AT+DL<CR>

返回: <CR><LF>+DL=time<CR><LF>

说明: time: 延时时间, 单位 MS。值范围[0,255], 当模块串口有数据输出时, 先将 nWakeUp 引脚电平置低, 延时 time 毫秒之后, 再输出数据。该功能用于用户 MCU 存在休眠时, 通过 nWakeUp 引脚先唤醒用户的 MCU。再向用户 MCU 传输串口数据。默认情况下, time=0。用户需要根据自己的 MCU 唤醒速度, 来设置 time 的值。

设置唤醒用户 MCU 等待时间

输入: AT+DL=time<CR>

返回: <CR><LF>+OK<CR><LF>

说明: time: 延时时间, 单位 MS, 取值范围[0,255]。

查询/设置远程目标地址

指令: AT+DSTADDR

查询远程控制中, 目标地址

输入: AT+DSTADDR<CR>

返回: <CR><LF>+DSTADDR=addr type,addr<CR><LF>

说明: addr type: 地址类型, 0 代表自定义地址方式通信, 1 代表网络地址方式通信。

addr: 欲控制的远程目标模块地址。

在远程查询/设置 GPIO,ADC,PWM 波时, 不需要每次都填写目标模块的地址。通过该指令, 保存远程目标节点地址。

设置远程控制中, 目标地址

输入: AT+DSTADDR=addr type,addr<CR><LF>

返回: <CR><LF>+OK<CR><LF>

说明: 用户在进行远程控制(远程 GPIO 控制, 远程 ADC 采集, 远程 PWM 波控制)前, 需要先通过该指令设置目标节点。在目标节点不变的情况下, 该指令只需要操作一次。

查询/设置终端数据缓存时间

指令: AT+HDT

查询发送给终端的数据缓存在父设备中的时间

输入: AT+HDT<CR>

返回: <CR><LF>+HDT=time<CR><LF>

说明：当网络中的某个设备向终端设备发送数据时，由于终端可能在休眠，所以，其数据是先缓存在终端的父设备中的，**time** 的值即为该数据缓存在父设备中的时间。单位是 **S**。默认 7 秒。终端设备被内部定时器定时唤醒。唤醒之后，主动去轮训父设备。将数据读取下来。

设置发送给终端的数据缓存在父设备中的时间

输入：AT+HDT=time<CR>

返回：<CR><LF>+OK<CR><LF>

说明：time： 值为缓存时间。取值范围是[1,10]。注：该值应该大于等于 POLL 的值。

查询信号强度

指令：AT+RSSI

查询是否开启查询信号强度模式

输入：AT+RSSI<CR>

返回：<CR><LF>+RSSI=status<CR><LF>

说明：status： RSSI 检测状态。0(表示 RSSI 检测关闭)，1(表示 RSSI 检测开启)。

设置是否开启查询信号强度模式

输入：AT+RSSI= status <CR>

返回：<CR><LF>+OK<CR><LF>

说明：status： RSSI 检测状态。取值范围：0(关闭)，1(开启)。

补充：具体操作请参考 3.10 章节。

附录 1：联系方式

公司：无锡谷雨电子有限公司

地址：江苏无锡市滨湖区锦溪路恒华科技园 21 号楼

网址：<http://www.ghostyu.com>

固话：0510-8518-7650

企业 QQ：400-670-7650

客服电话：400-670-7650

附录 2：版本记录

- V1.0 2016-07-07 第一次创建
- V1.1 2016-09-07 完善本地/远程控制 GPIO 指令，新增 AT+DL 指令。
- V1.2 2016-12-16 新增加远程查询 PWM 状态，远程查询 GPIO 口状态指令
增加 AT+IODIR 指令
增加 AT+RIODIR 指令
更改远程 GPIO，ADC，PWM 的查询和设置方法。
加入 AT+DHT。查询和更改终端数据在父设备中的缓存时间。
更改 ON OFF 为 1,0。
更改协调器，路由器，终端表示符号 C，R，E
查询返回值中，改成+CMD。
- V1.2.01 2016-12-30 更改协调器，路由器，终端表示符号 0，1，2
串口默认为 115200，8，1，NONE，485
- V1.2.02 2018-03-21