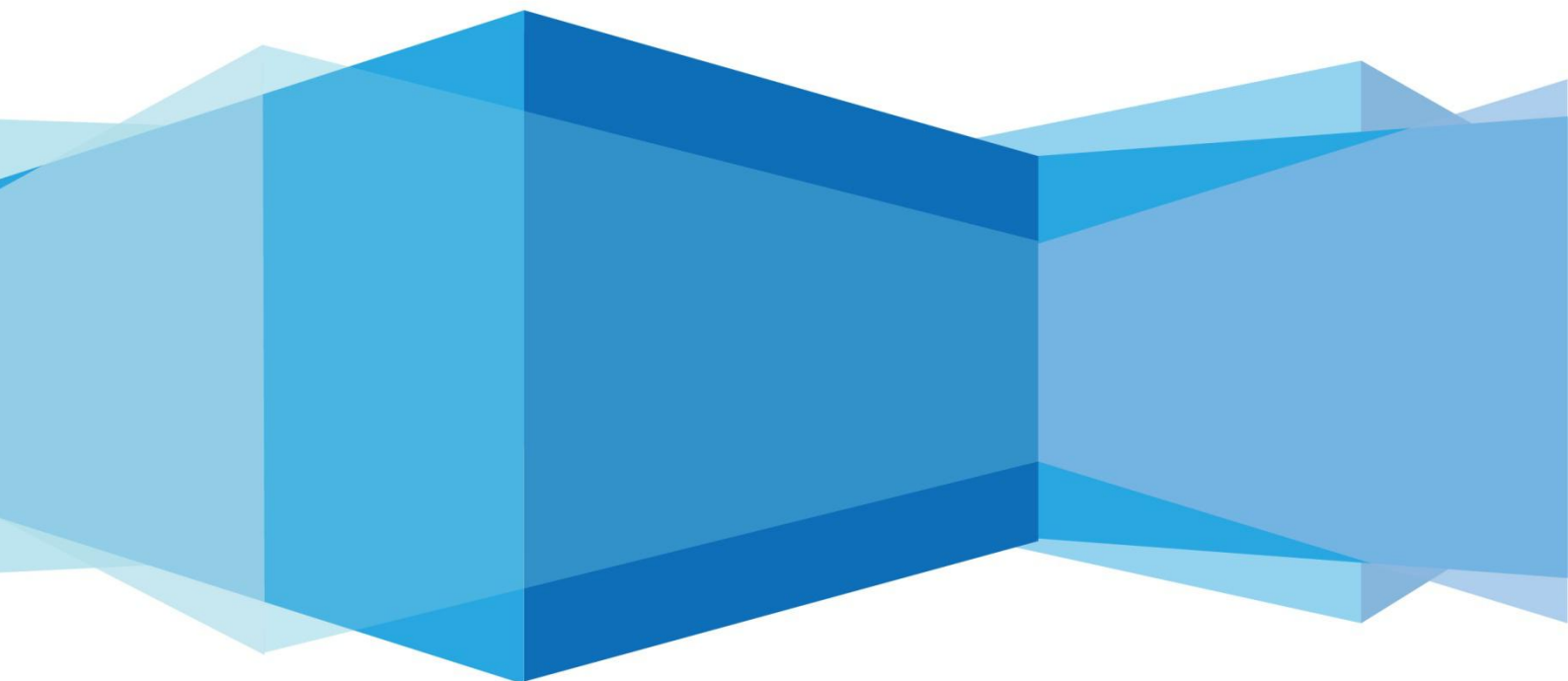


# **LSD4WN-2N717M91 产品使用说明书**

文件版本：Rev02

提交时间：2016 年 12 月 19 日



## 文件修订历史

版本	修订日期	修订说明
1.0	2016-06-07	初始版本
1.1	2016-06-14	细化功能描述，更正错误信息
1.2	2016-06-21	修正了快速配置范例章节，对 AppEUI 错误描述
1.2.1	2016-07-18	增加工作时序图
1.2.2	2016-07-21	修改了版本，删去提示符>>的说明
1.2.3	2016-08-25	删去命令间的换行，修改文档格式、参数错误的描述
1.2.4	2016-09-08	增加部分 AT 命令、增加子目录、增加 LoRaWAN、CLAA 等基础知识
2.0	2016-10-22	增加模块时序章节
2.0.1	2016-11-10	增加模块睡眠控制说明
2.0.2	2016-11-28	增加了使用场景的配置说明
2.0.3	2016-12-19	更新 AT 命令说明

# 目录

1 概述	7
1.1 文档使用范围	9
1.2 参考	9
1.2.1 LoRa Alliance	9
1.2.2 CLAA 联盟	10
1.2.3 协议标准	10
2 模块工作机制	11
2.1 指令模式	13
2.1.1 AT 协议约定	13
2.2 透传模式	14
2.3 参数配置	19
3 模块串口时序	20
3.1 开机时序	20
3.2 唤醒时序	20
3.3 串口帧分割时序	21
3.4 指令时序	21
3.5 串口发送时序	22
3.6 数据接收时序	22
3.7 模块透传时序	22
4 AT 命令集	24
4.1 系统命令	24

4.1.1	设置串口波特率	24
4.1.2	读取串口波特率	24
4.1.3	设置串口时间	25
4.1.4	读取串口时间	25
4.1.5	读取版本	26
4.1.6	系统复位	26
4.1.7	恢复出厂设置	27
4.1.8	设置 IO 口状态	27
4.1.9	设置 RTC 时间	28
4.1.10	读取 RTC 时间	28
4.2	MAC 命令	28
4.2.1	设置上行传输类型	28
4.2.2	读取上行传输类型	29
4.2.3	设置激活模式	29
4.2.4	读取激活模式	30
4.2.5	读取 DEVEUI	30
4.2.6	设置 APPEUI	31
4.2.7	读取 APPEUI	31
4.2.8	设置 DEVADDR	32
4.2.9	读取 DEVADDR	32
4.2.10	设置 APPKEY	33
4.2.11	读取 APPKEY	33

4.2.12	设置 APPSKEY.....	33
4.2.13	读取 APPSKEY.....	34
4.2.14	设置 NWKSKEY.....	34
4.2.15	读取 NWKSKEY.....	35
4.2.16	设置 ADR 使能.....	35
4.2.17	读取 ADR 使能.....	35
4.2.18	设置 PORT.....	36
4.2.19	读取 PORT.....	36
4.2.20	设置 CSQ.....	37
4.2.21	读取 CSQ.....	37
4.2.22	设置 Confirm 帧的发送次数.....	38
4.2.23	读取 Confirm 帧的发送次数.....	39
4.2.24	设置设备类型.....	39
4.2.25	读取设备类型.....	39
4.2.26	设置发送功率.....	40
4.2.27	读取发送功率.....	41
4.2.28	设置通信速率.....	41
4.2.29	读取通信速率.....	42
4.2.30	读取状态寄存器.....	42
4.2.31	读取上行结果寄存器.....	43
4.2.32	读取下行结果寄存器.....	43
4.2.33	读取 JOIN 信息.....	44

4.2.34	设置 Link Check.....	45
4.2.35	读取 Link Check.....	45
4.2.36	NOP 命令.....	47
4.2.37	参数保存.....	47
5	低功耗处理.....	49
6	应用配置范例.....	50
6.1	OTAA 模式配置.....	50
6.2	CLAA 模式配置.....	51
6.2.1	应用场景 1：单模模式.....	51
6.2.2	应用场景 2：多模模式.....	51
6.3	Class C 应用.....	52
	敬告用户.....	53

# 1 概述

LSD4WN-2N717M91 是利尔达科技集团股份有限公司研制的一款 LoRaWAN End Node 模块。本模块集成了 LoRaWAN™ 协议栈，符合 LoRa Alliance 发布的 LoRaWAN™ Specification 1.01 Class A\C 应用标准与中国 LoRa 应用联盟发布 CLAA 应用规范。本模块采用串行接口与用户设备进行数据、指令交互，可以方便地为用户提供快速 LoRaWAN 网络接入和无线数据等业务。

本说明书描述了 LSD4WN-2N717M91 模块的工作机制，以及模块与用户端交互的 AT 指令集。从空口支持的频段来看，模块功能包括：

- a) LoRaWAN CLAA CN470 应用。

## 1.1 LoRaWAN简介

LoRaWAN 属于一个低功耗广域网网络(LPWAN)规范，适用于在地区、国家或全球网络中，使用电池供电的无线设备，是一个支持 IoT 应用的通信网络。LoRaWAN 由 LoRa 联盟推动。

LoRaWAN Specification 定义了 PHY 与 MAC 协议，PHY 层主要使用 LoRa 调制技术（部分频段也使用了 FSK 调制）。LoRaWAN 网络架构是一个典型的星形拓扑结构，用于大容量远距离低功耗的网络应用，在这个网络架构中，LoRaWAN 网关可以看作是一个透明的“网桥”，连接前端终端设备(End Device)和后端中央服务器。网关与服务器通过标准 IP 连接，而终端设备采用单跳广播，与一个或多个网关通信，所有的节点均是双向通信。LoRaWAN 协议的星形拓扑结构消除了同步开销和跳数，因而降低了功耗。

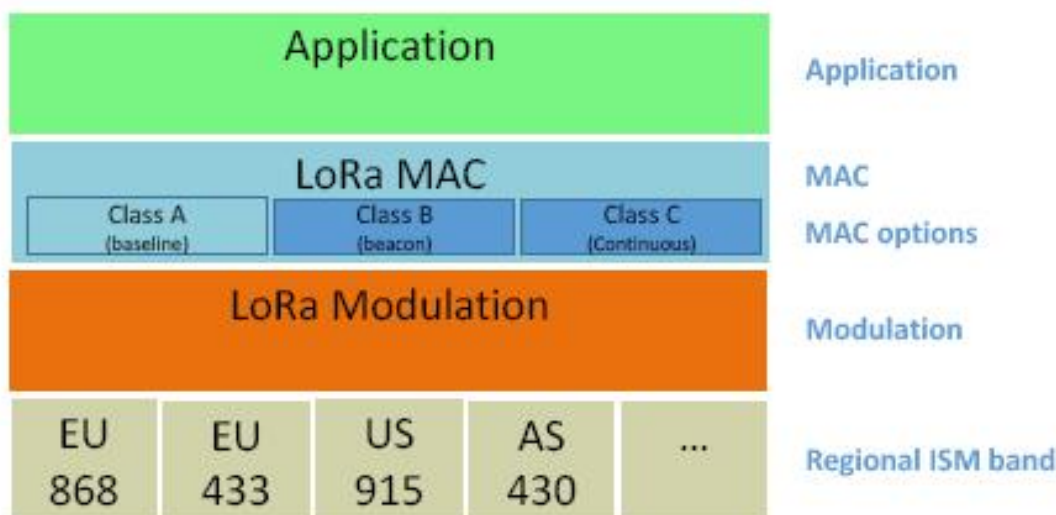


图 1.1 LoRaWAN 设备类型

### 1.1.1 LoRaWAN 设备类型

LoRaWAN 协议针对低功耗、电池供电的传感器等应用进行了不同等级的优化，分成不同级别的终端类型，以优化网络延迟和电池寿命间的平衡关系。同时为最大限度地提升终端设备的电池寿命和整体网络容量，LoRaWAN 网络服务器通过一种自适应数据速率（ADR）的方法管理每个终端速率和功率等。

表1-1 LoRaWAN的终端类型

设备类型	描述
Class A	Class A设备执行的是一个非对称的双向通信机制，每一终端设备上行发送后会伴随打开两个下行接收窗口。终端设备的传输窗口是基于其自身通信需求，其微调是基于一个随机的时间基准（ALOHA协议）。Class A设备应用中功耗最低，在终端发送一个上行传输信号后才能与服务器进行下行通信，与服务器任何时候的下行通信都只能是在上行通信之后。
Class B	Class B设备建立在Class A基础上，并会在预设时间中开放多余的接收窗口，为了达到这一目的，终端设备会同步从网关接收到一个Beacon，这一方式会让服务器知道终端设备正在“倾听”。
Class C	Class C设备同样基于Class A，在不需要发送数据的情况下，一直打开接收。
.....	更多的设备类型

Class A 设备通信机制，如图 1-1 所示：



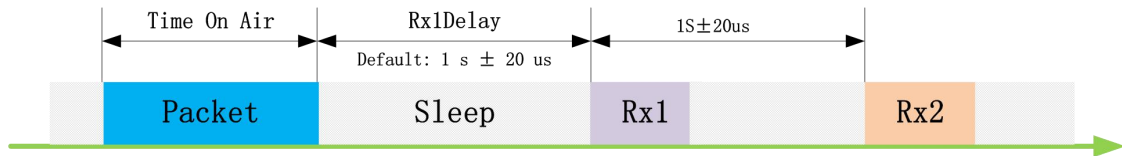


图 1-1 Class A 设备通信机制

Class C 设备通信机制，如图 1-2 所示：

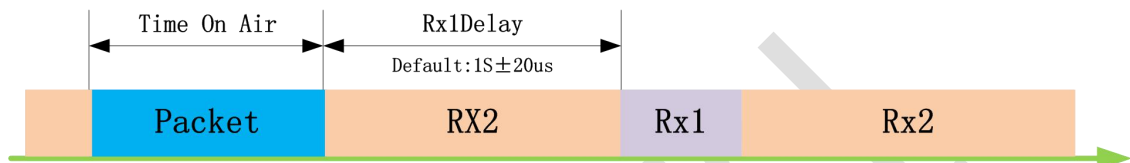


图 1-2 Class C 设备通信机制

## 1.2 文档使用范围

该说明书适用的固件版本：

LoRaWAN1.0.1 CLAA.02 02.09.06 Nov 25 2016 16:18:45 及其后续版本

## 1.2 参考

### 1.2.1 LoRa Alliance

LoRa™ Alliance 是一个开放、非盈利组织，由行业内领先厂商发起，成立于 2015 年 3 月。LoRa 联盟致力于建立一个安全的、运营等级、低功耗广域网(LPWAN)开放式标准规范，将低功耗广域网(LPWAN)推向全球。LoRa 联盟推广的 LoRaWAN™ 协议，可应用于 IoT、智慧城市、智慧工业、智慧生活等等众多领域。截止 2015 年的统计数据，全球目前已超过 350 个成员，成员领域涵盖电信商、设备商、物联网系统整合商、传感器制造商和半导体商、创新创业企业等等。由 LORA 联盟开发并支持的低功耗广域(LPWA)无线网络，力求针对低功耗、远距离、低成本 3 大物联网关键因素提供解决方案。

### 1.2.2 CLAA 联盟

CLAA 联盟是在 LoRa Alliance 支持下，由中兴通讯发起，各行业物联网应用创新主体广泛参与、合作共建的技术联盟，是一个跨行业、跨部门的全国性组织。其会员由国内外各类有低功耗、广覆盖物联网需求的企事业单位和专业社团组成，加强产业链厂家合作，构建 LoRa 技术应用生态圈。联盟的宗旨是推动 LoRa 产业链在中国的应用和发展，建设多业务共享、低成本、广覆盖、可运营的 LoRa 物联网。联盟口号：合作共赢，建设中国 LoRa 钻石联盟。

### 1.2.3 协议标准

- 1、LoRaWAN1.0.1 由 LoRa 联盟发布 <http://www.lora-alliance.org>
- 2、OMS-001-CLAA 码号命名规范及原则，由 CLAA 联盟发布
- 3、TES-003-CLAA 对中国 470M-510M 频段使用技术要求，由 CLAA 联盟发布
- 4、TES-004-CLAA 对 MAC 命令扩展规范，由 CLAA 联盟发布

## 2 模块工作机制

LSD4WN-2N717M91 模块(以下称模块) 集成了 LoRaWAN™ 协议栈, 模块采用串行接口、结合逻辑 GPIO 口与用户设备进行数据、指令交互, 可以方便地为用户提供快速 LoRaWAN 网络接入和无线数据等业务。

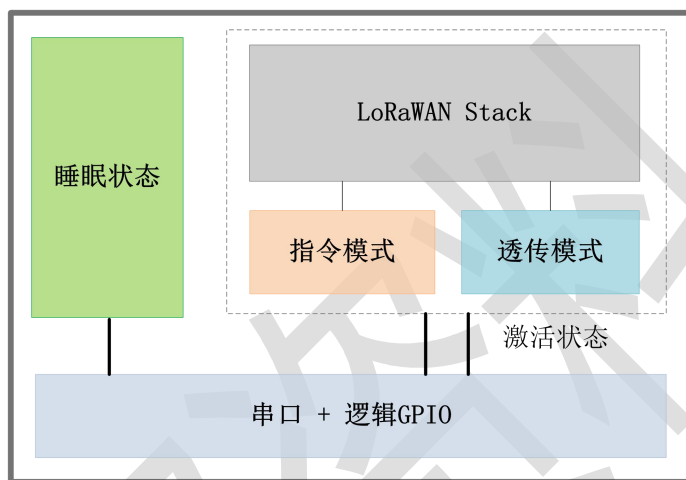


图 2.1 LSD4WN-2N717M91 固件功能框图

模块定义了两种工作状态, 分别是激活状态与睡眠状态, 用户通过 WAKE 引脚来选择, 如表 2-1 所示。

激活状态包含了两种工作模式, 分别是透传模式与指令模式。睡眠状态下, 模块将进入低功耗, 不会进行任何数据通信等操作, 但保存系统当前的运行状态, 模块重新被唤醒后, 用户即可进行 LoRaWAN 网络数据收发等操作。

表 2-1 状态控制

功能引脚	描述
WAKE= 1 保持高电平	激活状态
WAKE= 0 保持低电平	睡眠状态

用户通过 MODE 引脚来选择模块的工作模式, 如表 2-2 所示。

表 2-2 模式控制

功能引脚	描述
MODE= 1 保持高电平	指令模式
MODE= 0 保持低电平	透传模式

用户在指令模式，配置模块的相关串口通信参数、LoRaWAN 网络参数等，在完成参数配置后，若参数支持掉电存储，且用户需要，则发送保存命令，固件将配置参数写入模块 EEPROM 中。用户复位模块后，模块将按照新参数运行。

若模块被选定为 OTAA 激活，复位设备后默认会自动执行加入网络 ( JOIN ) 操作。

用户在透传模式，发送与接收应用数据。

## 2.1 指令模式

在指令模式下，串口参数固定为 9600N81，用户通过串口 AT 指令来访问模块，进行模块的配置寄存器的读取、查询状态寄存器等。

用户发送 AT 命令给模块，模块解析接收到的命令，执行完成后，立即返回一个命令响应，表征命令的执行结果。在 BUSY 为低电平(忙)的状态下，允许用户进入指令模式，此时可以读取系统状态寄存器等。

注意：

模块的LoRaWAN网络通信操作，只发生在透传模式，因此用户若完成指令模式下的相关操作，建议用户，立即切换回透传模式。

### 2.1.1 AT 协议约定

指令模式采用 AT 协议解析，传输 ASCII 格式数据。其中做如下约定，如表 2-3

表 2-3 AT 协议约定

符号	描述
AT	每个AT命令，都以AT开头。不区分大小写
+X1	+X1为命令
=	说明当前为设置操作，比如AT+DEVADDR=00112233
?	说明当前为读取操作，比如AT+DEVADDR?
,	参数分隔符，可能带多个参数，比如AT+MAC DATA=<type>,<port>,<data>
<CR><CF>	回车符,命令结束

### AT 命令说明

设置命令 AT+[...x] = Y

读取命令 AT+[...x] ?

执行 AT+[...x]

AT 命令返回值根据命令不同，具有不同返回值，如表 2-4 所示：

表 2-4 AT 指令返回值

返回结果	描述
OK	串口命令执行成功
ERROR	未知错误
BAD PARM	参数错误
+XX:YY..	命令XX对应的响应结果

注意：

AT命令以回车为结束符，即（'\r'）或者（0x0D）

## 2.2 透传模式

在透传模式下，未入网设备，首先执行加入网络操作，直至成功加入网络后，模块将直接转发用户数据。

模块只要在成功加入网络后，才能处理用户数据，进行有效的空口数据交互与串口数据交互。

### 2.2.1 搜索网络

模块首次上电或者复位后，若未执行过加入网络的操作，且模块被设定为 OTAA 激活（缺省），模块会自动执行加入网络（JOIN）操作。未成功加入网络前，STAT 引脚保持为低，直到成功加入网络，STAT 输出高电平。

模块默认会搜索 CLAA 五种模式（多模模式）（A-mode、B-mode、C-mode、D-mode、E-mode），按照一轮 D-E-B-C-A 的优先级循环扫描 2 次，由于 5 种模式所覆盖的信道众多，可能会出现耗时比较长，才成功加入网络的情况。缺省设置下，一轮扫描的最大时间约 12min。对于特定的应用，为了加快入网速度，用户可以通过 AT 指令来设定，模块固定在某个 CLAA 模式（单模模式）来扫描网络。

模块一旦成功加入网络后，模块将持久化网络上下文参数。因此重启模块后，将不会进行 JOIN 操作，可以直接进行数据服务。如果用户数据连续发送失败，达到重启 JOIN 条件后，模块会自动开启 JOIN 操作，直到 JOIN 成功，可进行正常的数据服务。如果模块 JOIN 一定次数后仍无法加入网络（网络异常等），模块将停止 JOIN 操作，此时 STAT 引脚将保持为低。

### 2.2.2 流控机制

用户给串口发送数据前，需要判断 BUSY 信号是否为高电平（不忙）。BUSY 信号为低电平（忙），用户停止发送串口数据，必须等待 BUSY 信号为高电平，用户才可以继续发送数据。当 BUSY 信号为低电平（忙），用户可以通过指令模式，读取当前系统状态。

一次数据包发送后，如果模块有接收到服务器下行数据，则通过串口立即返回给用户。

如果开启 ADR 机制，由于每个空口数据包的最大数据长度可能会动态变化（Network Server 可动态调整），为了保证数据传输可靠性与完整性，引入一种简单的流控机制。

用户自行决定一帧数据的长度。当串口超过 10ms 时间未接收到新的串口数据数据或者达到 FIFO 存储上限时，判定一帧串口数据传输完成，立即拉低 BUSY 引脚（忙），同时关闭串口接收，模块进行空口发送操作。空口发送完成后（成功或失败），BUSY 引脚重新拉高。

### 2.2.3 物理分包机制

实际的物理分包情况由 Network Server 决定，用户可以通过 AT 指令查询当前可支持的最大负载值，来获取分包情况。

通常情况下，不同速率对应的最大负载值 N，如表 2-5 所示：

表 2-5 不同速率对应的最大负载值 N

Data Rate	SF\BW	N (MAX)	Time On Air (ms)
DR5	SF7\BW125	222	102.7
DR4	SF8\BW125	222	215.6
DR3	SF9\BW125	115	390.1
DR2	SF10\BW125	51	698.4
DR1	SF11\BW125	51	1560.6
DR0	SF12\BW125	51	2793.5

## 2.2.4 服务器响应

根据 LoRaWAN 网络 Class A 运行特点，任何一包数据，用户服务器都可以给出响应，如果模块收到用户服务器数据，会立即通过串口输出。因此由于数据帧分包的原因，用户的一帧数据可能收到若干个响应数据包。模块默认直接转发下行数据给用户 MCU，用户可以通过指定的 AT 指令设置，第一个字节是否为数据包的大小。模块发送给用户 MCU 的串口数据格式，如表 2-6 所示：

表 2-6 发送用户 MCU 的串口数据格式

字段	说明
SIZE	用户数据的大小，该字段可选，默认情况下，无该字段
Payload	服务器给用户的下行数据

每次发送完空口数据，模块 BUSY 引脚会重新设置为高电平，同时模块通过 STAT 引脚输出当前网络状态，会保存系统状态，同时 STAT 引脚输出模块的网络状态。用户可以切换到指令模式，读取状态值（比如用户物理包的大小、FIFO 剩余数据、最近的 RSSI、网络状态等信息），系统状态寄存器如表 2-7：



表 2-7 模块状态寄存器

寄存器Offset	寄存器名	字节	描述
0	上行结果	1	当前上行结果 00 - 无数据操作 01 - 数据发送中 02 - 数据发送失败 03 - 数据发送成功 04 - JOIN成功（仅出现在首次JOIN过程中） 05 - JOIN失败（仅出现在首次JOIN过程中）
1	上行信道	1	当次上行所使用的信道
2	上行速率	1	当次上行所使用的速率
3	上行功率	1	当次上行所使用的功率
4	下行信道	1	当次下行所使用的信道
5	下行速率	1	当次下行所使用的速率
6	RFU	1	保留
7	RXSL0T	1	当次下行的接收窗口
8	PORT	1	当次下行PORT
9	RFU	1	保留
10	RFU	1	保留
11	SNR	1	当次下行的SNR
128	RFU	1	保留
13~14	RSSI	2	当次下行的RSSI
15~18	下行计数值	4	记录服务器侧下行的次数
19~20	剩余字节数	2	模块内存中待发送的上行数据
21~22	Confirm帧重发次数	2	针对Confirm帧，当前是第几次重发
23~26	RFU	4	保留
27	RFU	1	保留
28~31	RFU	4	保留
32~35	RFU	4	保留
36~39	上行计数值	4	记录模块的上行次数

模块每执行完一次上行数据交互后会更新上行结果寄存器,上行结果寄存器包含内容如

表 2-8 所示：

表 2-8 模块上行结果寄存器

寄存器Offset	寄存器名	字节	描述
0	上行结果	1	当前上行结果 00 - 无数据操作 01 - 数据发送中 02 - 数据发送失败 03 - 数据发送成功 04 - JOIN成功（仅出现在首次JOIN过程中） 05 - JOIN失败（仅出现在首次JOIN过程中）
1	上行信道	1	当次上行所使用的信道
2	上行速率	1	当次上行所使用的速率
3	上行功率	1	当次上行所使用的功率
4	非确认帧发送次数	1	UnConfirm帧有效，当次数据上行的发送次数
5	确认帧发送次数	1	Confirm帧有效，当次数据上行的发送次数
6~7	剩余字节数	2	模块内存中待发送的上行数据
8~11	上行计数值	4	记录模块的上行次数
12	RFU	1	保留

模块每执行完一次数据交互后，如果有接收到服务器的数据（Confirm 帧的 ACK 包或者下行数据包）会更新下行结果寄存器，下行结果寄存器包含内容，如表 2-9 所示：

表 2-9 模块下行结果寄存器

寄存器Offset	寄存器名	字节	描述
0	PORT	1	当次下行PORT
1	下行用户数据大小	1	当次下行接收到的用户数据大小
2	RXSLLOT	1	当次下行的接收窗口
3	SNR	1	当次下行的SNR
4~5	RSSI	2	当次下行的RSSI
6	下行信道	1	当次下行所使用的信道
7	下行速率	1	当次下行所使用的速率
8~11	下行计数值	4	记录服务器侧下行的次数
12	RFU	1	保留
14~17	RFU	4	保留

## 2.3 参数配置

为了能够与 LoRaWAN 网络通信，首次使用模块前，有部分网络参数必须被配置（若未设置 将按照缺省参数运行）。LoRaWAN Specification 提供了两种 End Device 激活方式，每次方式所需配置参数有所不同，如表 2-10 所示

表 2-10 两种激活方式所需网络参数

激活方式	描述	所需参数
Over the Air Activation (OTAA)	模块通过加入网络（JOIN）操作，来获得网络安全密钥	DevEUI、AppEUI、AppKey，同时设置为OTAA激活方式。
Activation by Personalization (ABP)	模块本地已经保存网络会话密钥、应用加密密钥等，可以加入指定的 LoRaWAN 服务器	DevAddr、NwkSkey、AppSkey，同时设置为ABP激活

注意：

当前模块在CLAA协议中，推荐使用的激活模式是OTAA。

## 3 模块串口时序

本章节描述了模块串口通信过程中所遵循的控制时序，用户参考如下时序，进行产品相关接口设计。

### 3.1 开机时序

模块每次上电需要一段的初始化时间，在这段时间内，用户不能对模块进行任何操作。模块上电时序，如图 3.1 所示。

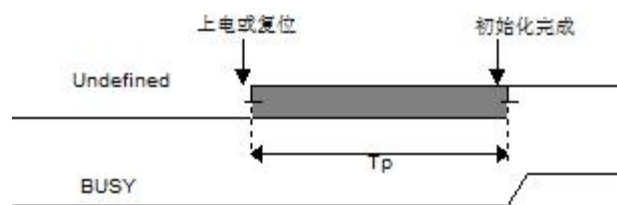


图 3.1 开机时序

模块上电或者复位后需要等待一段时间  $T_p$ ，典型值 150ms。在模块初始化时间内，用户需要给模块的输入 I/O 一个固定的电平（WAKE 引脚、MODE 引脚）。在初始化完成之前，模块不会响应用户任何操作（比如发送串口数据或者进行 I/O 操作等）。

模块上电，BUSY 缺省为低电平，待初始化完成后，BUSY 信号会输出高电平，这个 BUSY 信号用来指示用户，接下来模块可以进入指令模式，配置参数等。

注意  $T_p$  时间为模块硬件初始化时间，不包括模块搜索（JOIN）网络的时间。

### 3.2 唤醒时序

模块进入睡眠状态后，用户只能通过 WAKE 引脚将其唤醒（除非复位模块），模块接收到有效的唤醒信号后，需要一段时间（ $T_w \leq 10ms$ ）才能完全醒来。在模块被完全唤醒前，模块将不能接收来自用户数据，模块的唤醒时序如图 3.2 所示。



图 3.2 模块唤醒时序

当模块处于睡眠状态时，用户拉高 WAKE 引脚即可唤醒模块，模块完全唤醒后，将保持激活状态，直到 WAKE 引脚被再次拉低，模块将再次进入睡眠状态。模块进入激活状态前，需要一段恢复时间，这段时间内，模块不可以接收数据。

### 3.3 串口帧分割时序

在透传模式下，模块会将串口数据流分割为不同的帧，如第 3 章所述，模块有两种分帧机制。如下描述的是串口超时分帧机制，即模块在超过设定时间  $T_g$  内( $T_g \geq 10\text{ms} + X$ ，其中  $X$  的单位为  $1/2048\text{s}$ )，没有接收到任何新的串口数据，模块认为该帧已经结束，开始处理这个帧。帧超时分割时序，如图 3.3 所示。

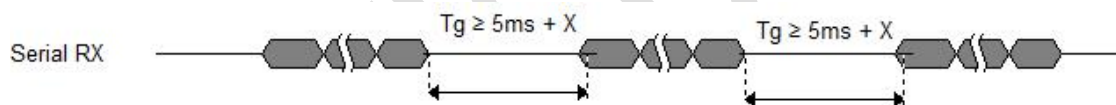


图 3.3 帧超时分割时序

### 3.4 指令时序

在指令模式下，用户发出指令，模块收到后，就会立即解析该指令，并给出相应的响应结果。从用户发出指令到收到模块响应需要一定时间 ( $T_r$ )，指令操作的时序如图 3.4 所示。

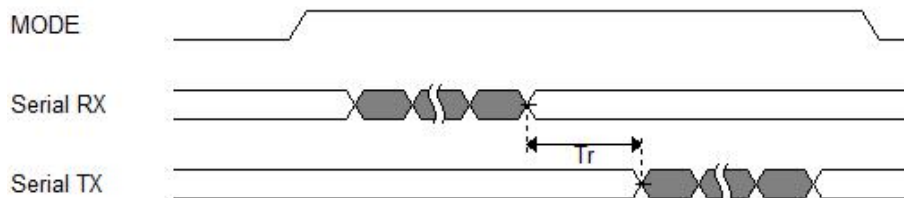


图 3.4 指令操作时序

命令响应的时间，跟命令带参数个数有关，除了 AT+SAVE，单参数指令响应时间

$Tr \leq 10ms$ 。

### 3.5 串口发送时序

用户发送串口数据，需要遵循 BUSY 信号，数据发送的时序如图 3.5 所示。

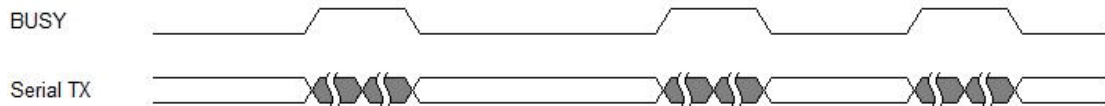


图 3.5 串口发送时序

只有在 BUSY 为高的情况，写入数据才是安全的。

### 3.6 数据接收时序

当模块接收到服务器的下行数据，模块会通过 BUSY 引脚通知 MCU，一定时间( 迟滞时间  $T_d$  )后，再将数据发送给用户，数据接收时序如图 3.6 所示。

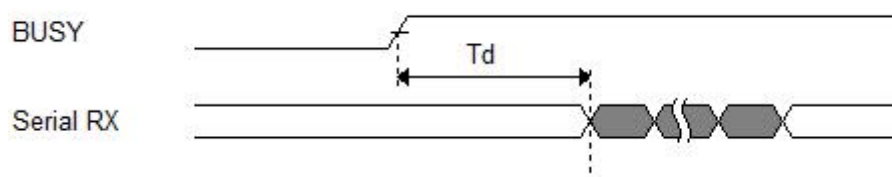


图 3.6 数据接收时序

当模块接收数据，且用户设定了模块串口延时发送，那么模块会先拉高 BUSY 引脚，一段时间后（ $T_d$ ）再发送数据给用户。这样，用户 MCU 有足够长的时间通过 BUSY 引脚将自己唤醒，准备好接收，这样不会丢失模块发送的数据，用户可以设定  $T_d$  时间。

$T_d$  缺省为 0ms。

### 3.7 模块透传时序

下图描述了在模块已经成功加入网络后，模块进行一次用户数据交互的流程，如图 3.7 所示。

0) 用户手动复位模块，该步在异常情况才需要，比如模块长时间通信异常等。

1) 用户拉高 WAKE，唤醒模块。MODE 引脚保持为低电平（透传模式）。

- 2) 用户等待模块准备好，等待 BUSY 为高电平。
- 3) 用户发送串口数据。
- 4) 串口分帧超时（如果用户一次传输的数据包小于当前 LoRaWAN 最大数据包限制）
- 5) 模块发送空口数据（上行），如果是 Confirm 帧，如果发送失败，模块自动重发。
- 6) 如果服务器端有下行数据或者当前是 Confirm 帧，则模块会接收到空口数据
- 7) 如果服务器有下行数据，则模块通过串口发送给用户。
- 8) 用户等待 BUSY 信号为高电平，模块完成空口数据（上行与下行）处理。如果当次通信数据出现异常，则 STAT 引脚输出为低，指示用户异常。用户可进入指令模式，通过 **AT+STATUS?** 等相关寄存器，获取更详细的状态信息。如果无异常，STAT 引脚保持为高电平。
- 9) 一次用户数据交互完成，用户根据需要，可以关闭模块，模块重新进入睡眠。

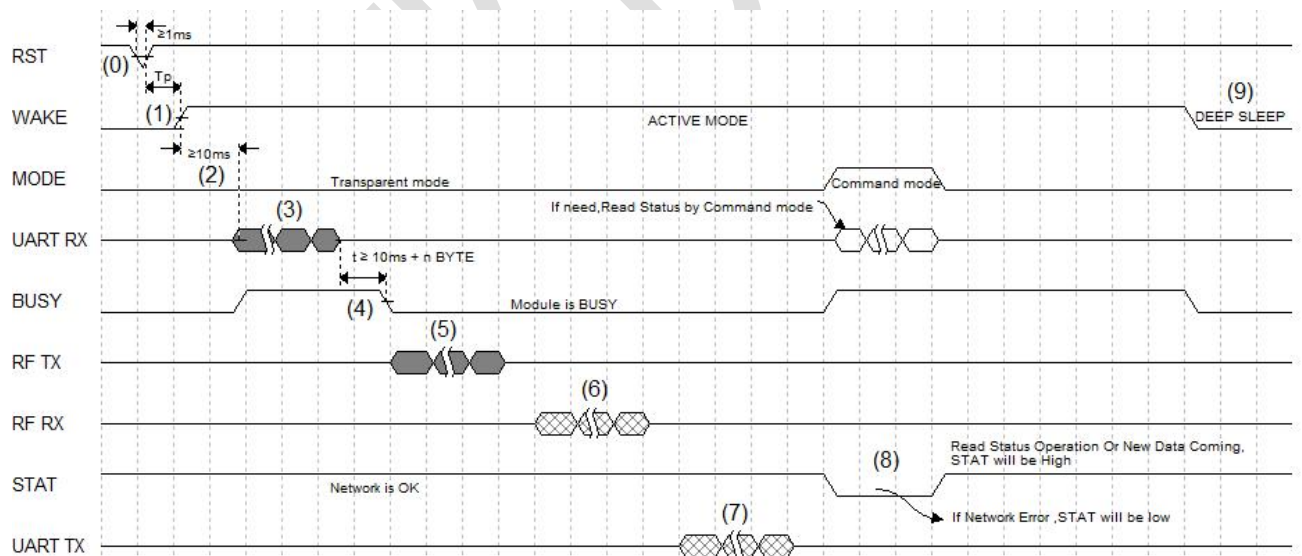


图 3.7 模块透传模式的工作时序



## 4 AT命令集

命令集细分为系统命令与 MAC 命令。系统命令主要配置串口、保存、恢复出厂参数等上层应用参数，MAC 命令主要配置 LoRaWAN 相关的网络参数。

注 1：“响应”中可能出现的具体数值，为示例所示。

### 4.1 系统命令

#### 4.1.1 设置串口波特率

设置串口波特率		
描述	<ul style="list-style-type: none"><li>该命令设置透传模式下的串口波特率。执行该设置后，用户从指令模式切换到透传模式后，将立即生效</li><li>命令模式固定为9600N81。</li><li>该参数修改后，若需要保存到EEPROM，需要执行SAVE指令</li></ul>	
命令	AT+BAUD=X1	X1为串口支持的波特率，出厂值：9600，取值范围： 1200\2400\ 4800\ 9600\ 19200\ 38400\57600\ 115200
响应1	OK	设置成功
响应2	BAD PARM	命令参数错误
响应3	ERROR	未知错误
示例	AT+BAUD=9600  OK	设置模块透传模式下的串口波特率为9600bps

#### 4.1.2 读取串口波特率

读取串口波特率		
描述	该命令读取透传模式下的串口波特率	
命令	AT+BAUD?	返回波特率，有效范围： 1200\2400\ 4800\ 9600\ 19200\ 38400\ 115200
响应1	+BAUD: X1  OK	读取成功，返回结果
响应2	ERROR	未知错误
示例	AT+BAUD?  +BAUD: 9600	模块透传模式的串口波特率为9600



OK

### 4.1.3 设置串口时间

#### 设置串口时间

描述	◆ 该命令设置透传模式下的接收超时时间（分帧间隔Tg）与串口的发送迟滞时间Td。	
	◆ 串口分帧间隔Tg，默认最小值为10ms，允许用户根据应用设置增加串口接收分帧间隔	
	◆ 串口的发送迟滞时间Td，默认为0，允许用户根据应用设置迟滞时间	
	◆ 该参数修改后，若需要保存到EEPROM，需要执行SAVE指令	
命令	格式1：	X1为串口接收分帧间隔Tg，单位：1/2048s，默认值：0 (Tg
	AT+TIMEOUT=X1	最小值固定为10ms)，取值范围0~65535，
	格式2：	X2为串口延时给用户发送数据的时间Td，单位为ms，默认
	AT+TIMEOUT=X1, X2	值为0，有效范围：0~255
响应1	OK	设置成功
响应2	BAD PARM	命令参数错误
响应3	ERROR	未知错误
示例1	AT+TIMEOUT=20	串口分帧超时时间Tg=10ms+~10ms
	OK	
示例2	AT+TIMEOUT=0, 10	串口分帧超时时间Tg=10ms (缺省最小值)
	OK	串口发送迟滞时间Td=10ms

### 4.1.4 读取串口时间

#### 读取串口接收

描述	该命令读取透传模式下的接收分帧间隔Tg与串口发送迟滞时间Td	
命令	AT+TIMEOUT?	读取串口接收分帧间隔Tg与串口延时给用户发送数据的时间Td
响应1	+TIMEOUT: X1, X2	读取成功。返回结果
	OK	
响应2	ERROR	未知错误
示例	AT+TIMEOUT?	串口分帧超时时间Tg=10ms (缺省最小值)
	+TIMEOUT: 0, 0	串口发送迟滞时间Td=0
	OK	

## 4.1.5 读取版本

读取版本														
描述	该命令返回产品版本号													
命令	AT+VER?	读取版本号												
		返回硬件版本+固件版本+时间版本												
		硬件版本：1Byte，区分硬件，比如0x01												
		固件版本： X. Y. Z.												
		(major.minor.build)												
		<table><tr><td>X</td><td>Y</td><td>Z</td></tr><tr><td>01</td><td>00</td><td>00</td></tr></table>	X	Y	Z	01	00	00						
X	Y	Z												
01	00	00												
		时间版本												
		<table><tr><td>月</td><td>日</td><td>年</td><td>时</td><td>分</td><td>秒</td></tr><tr><td>英文缩写</td><td>1-31</td><td>当前年</td><td>0-23</td><td>0-59</td><td>0-59</td></tr></table>	月	日	年	时	分	秒	英文缩写	1-31	当前年	0-23	0-59	0-59
月	日	年	时	分	秒									
英文缩写	1-31	当前年	0-23	0-59	0-59									
响应1	+VER: Y1	读取成功，返回结果												
	OK	Y1 为当前固件版本												
示例	AT+VER?	固件版本												
	+VER: LoRaWAN1.0.1 CLAA.02													
	01.00.01 Aug 24 2016													
	16:41:24													
	OK													

## 4.1.6 系统复位

系统复位		
描述	<p>该命令复位模块</p> <p>系统重启后自动装载EEPROM中的配置参数，并运行</p>	
命令	AT+RESET	执行命令
示例	AT+RESET	软件复位系统
响应1	OK	设置成功，
响应2	ERROR	未知错误
示例	AT+RESET	设置成功，系统随后立即启动复位

OK

### 4.1.7 恢复出厂设置

#### 恢复出厂设置

描述	该命令擦除用户对模块的参数配置，恢复模块配置参数为出厂值，并重启模块	
命令	AT+FACTORY	执行命令
响应1	OK	设置成功
响应2	ERROR	未知错误
示例	AT+FACTORY OK	

### 4.1.8 设置 IO 口状态

#### 设置IO口状态

描述	扩展功能，针对P1，P2，P3有效。 ps: 这只是当前一种可能的扩展功能。	
命令	AT+GPIO=pin, state pin: 0 表示 P0 1 表示 P1 2 表示 P2 3 表示 P3 255 表示 P0 P1 P2 P3 state 1 输出高 0 输出低	
响应1	OK	设置成功。
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+GPIO=0, 1 OK	

### 4.1.9 设置 RTC 时间

#### 设置RTC时间

##### 描述

该命令设置模块的RTC时间

RTC需要在模块不掉电的情况下才能保证有效

##### 命令

AT+RTC=X1, X2...X6

X1, X2...X6 为RTC时间，时间格式如下：

年	月	日	时	分	秒
X-2000	1-12	1-31	0-59	0-59	0-59

##### 响应1

OK

设置成功。

##### 响应2

BAD PARM

参数错误

##### 响应3

ERROR

未知错误

##### 示例

AT+RTC=16, 03, 25, 00, 00, 00

设置年、月、日、时、分、秒

OK

### 4.1.10 读取 RTC 时间

#### 读取RTC时间

##### 描述

该命令读取模块的RTC时间

##### 命令

AT+RTC?

返回RTC时间X1, X2...X6

年	月	日	时	分	秒
X-2000	1-12	1-31	0-59	0-59	0-59

##### 响应1

+RTC: X1, X2...X6

读取成功。返回结果

OK

##### 响应2

ERROR

未知错误

##### 示例

+RTC:16 03 25 00 00 00

当前RTC

OK

## 4.2 MAC命令

### 4.2.1 设置上行传输类型

设置上行链路（upLink）传输类型

描述	该命令设置上行链路的传输类型	
	该参数修改后，若需要保存到EEPROM，需要执行SAVE指令	
命令	AT+CONFIRM=X1	X1为上行链路的传输类型, 取值范围: 0 - UnConfirmed message, 服务器不会应答上行数据包 1 - Confirmed message, 服务器会应答上行数据包
响应1	OK	设置成功
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+CONFIRM=1	设置上行帧为Confirm (确认) 帧
	OK	

## 4.2.2 读取上行传输类型

读取上行链路 (upLink) 传输类型		
描述	该命令读取上行链路的传输类型	
命令	AT+CONFIRM?	读取上行帧类型 (Confirm or Unconfirm)
响应1	+CONFIRM:X1 OK	读取成功。返回结果
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+CONFIRM?	模块当前帧传输类型为Confirm帧
	+CONFIRM:1	
	OK	

## 4.2.3 设置激活模式

设置设备的激活模式		
描述	该命令加入网络的激活模式	
	该参数修改后，若需要保存到EEPROM，需要执行SAVE指令	
命令	AT+OTAA=X1	X1为激活类型, 取值范围: 0 - ABP激活 1 - OTAA激活
响应1	OK	设置成功. 示例说明
响应2	BAD PARM	参数错误

响应3	ERROR	未知错误
示例	AT+OTAA=1 OK	设置为OTAA激活

## 4.2.4 读取激活模式

读取设备的激活模式		
描述	该命令读取设备的激活模式	
命令	AT+OTAA?	读取设备的激活模式
响应1	+OTAA:X1 OK	读取成功。返回结果
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+OTAA? +OTAA:1 OK	模块当前为OTAA激活方式

## 4.2.5 读取 DEVEUI

读取模块的 DevEUI								
描述	该命令读取DevEui，设备端固化。 DevEUI 为全网唯一的设备标识符。类似MAC地址的意义							
命令	AT+DEVEUI?	读取DevEUI，返回Y1, Y2, ...Y8，16进制格式，取值8字节，CLAA定义如下所示 <table><tr><th>OUI</th><th>制造商识别号</th><th>制造商序列号</th></tr><tr><td>3 Byte</td><td>2 Byte</td><td>3 Byte</td></tr></table> <p>OUI是由电器和电子工程师协会(IEEE)分配的组织唯一标示符，当前为中兴公司在IEEE申请的OUI (0x004A77)</p> <p>制造商识别号为CLAA编号。</p> <p>制造商序列号为厂商自己定义。</p>	OUI	制造商识别号	制造商序列号	3 Byte	2 Byte	3 Byte
OUI	制造商识别号	制造商序列号						
3 Byte	2 Byte	3 Byte						
响应1	+DEVEUI: Y1, Y2, ...Y8 OK	读取成功。返回结果						
响应2	ERROR	未知错误						
示例	AT+DEVEUI?	模块DEVEUI为						

+DEVEUI: 00 4A 77 00 66 00 00 0x00, 0x4A, 0x77, 0x00, 0x66, 0x00, 0x00, 0x01  
01  
OK

## 4.2.6 设置 APPEUI

### 设置 AppEUI

**描述** 该命令设置应用服务标识符。

该参数修改后，若需要保存到EEPROM，需要执行SAVE指令

**命令** AT+APPEUI=X1 X2...X8

X1, X2, ...X8为全网唯一的应用标识符。服务器分配，用于区分具体应用，比如水表采集、传感器采集、灯类控制等等，数据格式为16进制，取值8个字节：

CLAA命名规则如下所示：

应用类别	应用地域	应用编号
3 Byte	2 Byte	3 Byte

**响应1** OK 设置成功

**响应2** BAD PARM 参数错误

**响应3** ERROR 未知错误

**示例** AT+APPEUI= FF FF FE 00 00 00 00 01 设置APPEUI为0xFF, 0xFF, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x01

## 4.2.7 读取 APPEUI

### 读取 AppEUI

**描述** 该命令读取应用标识符

**命令** AT+APPEUI? 返回值Y1, Y2...Y8, 描述同“设置AppEUI”，数据格式为16进制，取值8个字节

**响应1** +APPEUI: X1 X2...X8 设置成功，则返回Y1。  
OK Y1为APPEUI值

**响应2** ERROR 未知错误

**示例** AT+APPEUI? 模块的APPEUI为0xFF, 0xFF, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x01  
+APPEUI: FF FF FE 00 00 00 00 01  
01

OK

## 4.2.8 设置 DEVADDR

### 设置 DevAddr

描述	该命令设置DevAddr，取值4个字节。	
	该参数修改后，若需要保存到EEPROM，需要执行SAVE指令。	
命令	AT+DEVADDR= X1 X2 X3 X4	X1, X2, X3, X4为LoRaWAN全网唯一的设备地址, 取值： 0x00000000~0xffffffff，高字节在前，数据格式为16进制， 取值4个字节 DevAddr根据入网激活方式，区分两种情况： 1、通过本地激活的节点（ABP），必须设置devaddr。 2、通过空中激活（OTAA）方式，无需设置devaddr，一旦成功JOIN LoRaWAN网络，该参数会被自动重写（服务器分配）。
响应1	OK	设置成功.
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+DEVADDR=FF 00 00 01  OK	设置模块DEVADDR为0xFF000001

## 4.2.9 读取 DEVADDR

### 读取 DevAddr

描述	该命令读取DevAddr	
命令	AT+DEVADDR?	读取设备地址，返回具体DevAddr值，数据格式为16进制， 取值4个字节
响应1	+DEVADDR: X1 X2 X3 X4  OK	读取成功。返回结果  X1 X2 X3 X4 为DEVADDR值
响应2	ERROR	未知错误
示例	AT+DEVADDR?  +DEVADDR: FF 00 00 01  OK	模块DEVADDR为0xFF000001



## 4.2.10 设置 APPKEY

### 设置 AppKey

描述	该命令设置AppKey。AppKey只针对OTAA激活方式有效。	
	该参数修改后，若需要保存到EEPROM，需要执行SAVE指令。	
命令	AT+APPKEY= X1 X2...X16	设置应用密钥。X1, X2...X16为Application Key，数据格式为16进制。取值16个字节
响应1	OK	设置成功
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+APPKEY= 2B 7E 15 16 28 AE	设置APPKEY为0x2B 0x7E 0x15 0x16 0x28 0xAE
	D2 A6 AB F7 15 88 09 CF 4F 3C	0xD2 0xA6 0xAB 0xF7, 0x15 0x88 0x09 0xCF 0x4F 0x3C

## 4.2.11 读取 APPKEY

### 读取 AppKey

描述	该命令在OTAA激活方式中使用（ABP方式不需要这个参数）	
命令	AT+APPKEY?	读取应用密钥。返回X1 X2...X16，数据格式为16进制。取值16个字节
响应1	+APPKEY: X1 X2...X16	读取成功，返回结果
	OK	
响应2	ERROR	未知错误。
示例	AT+APPKEY?	模块当前应用密钥为0x2B 0x7E 0x15 0x16 0x28 0xAE 0xD2
	+APPKEY: 2B 7E 15 16 28 AE D2	0xA6 0xAB 0xF7, 0x15 0x88 0x09 0xCF 0x4F 0x3C
	A6 AB F7 15 88 09 CF 4F 3C	
	OK	

## 4.2.12 设置 APPSKEY

### 设置 AppSKey

描述	该命令设置AppSkey。AppSkey只对ABP激活方式有效。取值16个字节。	
	该参数修改后，若需要保存到EEPROM，需要执行SAVE指令	
命令	AT+APPSKEY= X1 X2...X16	设置应用会话密钥。16个参数。X1, X2...X16为Application session key，数据格式为16进制。
响应1	OK	设置成功

响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+APPSKEY = 2B 7E 15 16 28 AE    设置APPSKEY为0x2B 0x7E 0x15 0x16 0x28 0xAE D2 A6 AB F7 15 88 09 CF 4F 3C    0xD2 0xA6 0xAB 0xF7, 0x15 0x88 0x09 0xCF 0x4F 0x3C OK	

### 4.2.13 读取 APPSKEY

读取 AppSKey		
描述	该命令读取AppSKey。AppSKey仅对ABP激活方式有效。	
命令	AT+APPSKEY?	读取应用会话密钥。 返回X1 X2...X16为Application session key, 数据格式为16进制。取值16个字节。
响应1	+APPSKEY: X1 X2...X16	读取成功。返回结果
响应2	ERROR	未知错误
示例	AT+APPSKEY?  +APPSKEY: 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C  OK	

### 4.2.14 设置 NWKSKEY

设置 NwkSKey		
描述	该命令设置NwkSKey。NwkSKey对ABP激活方式有效。 该参数修改后, 若需要保存到EEPROM, 需要执行SAVE指令。	
命令	AT+NWKSKEY=X1 X2...X16	设置NWK会话密钥。X1,X2...X16为network session key, 数据格式为16进制。取值16个字节。
响应1	OK	设置成功
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+NWKSKEY = 2B 7E 15 16 28 AE    设置NWKSKEY为0x2B 0x7E 0x15 0x16 0x28 0xAE D2 A6 AB F7 15 88 09 CF 4F 3C    0xD2 0xA6 0xAB 0xF7, 0x15 0x88 0x09 0xCF 0x4F 0x3C	

### 4.2.15 读取 NWKSKKEY

读取 NwkSKey		
描述	该命令读取NwkSKey。NwkSKey对ABP激活方式有效。	
命令	AT+NWKSKEY?	读取NWK密钥，数据格式为16进制。取值16个字节。
响应1	+NWKSKEY: X1 X2...X16	读取成功。返回结果
响应2	ERROR	未知错误
示例	AT+NWKSKEY?  +NWKSKEY: 2B 7E 15 16 28 AE D2 0xD2 0xA6 0xAB 0xF7, 0x15 0x88 0x09 0xCF 0x4F 0x3C  A6 AB F7 15 88 09 CF 4F 3C  OK	

### 4.2.16 设置 ADR 使能

设置 ADR使能		
描述	<ul style="list-style-type: none"><li>◆ 该命令设置使能\失能ADR (Adaptive data rate) 机制。</li><li>◆ 模块的ADR状态体现在上行帧中，以便通知服务器。ADR使能后，服务器将根据当前网络状态，优化节点的速率与功率。</li><li>◆ 该参数修改后，若需要保存到EEPROM，需要执行SAVE指令</li></ul>	
命令	AT+ADR=X1	X1值定义如下： 0 ADR不使能 1 ADR使能
响应1	OK	设置成功
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+ADR=1  OK	

### 4.2.17 读取 ADR 使能

读取ADR使能		
描述	该命令读取ADR使能控制	
命令	AT+ADR?	读取ADR使能控制
响应1	+ADR: X1	读取成功。返回结果
	OK	

响应2	ERROR	未知错误
示例	AT+ADR?  +ADR: 1  OK	模块当前开启ADR功能

## 4.2.18 设置 PORT

设置PORT		
描述	<ul style="list-style-type: none"><li>◆ 该命令设置端口号[PORT],</li><li>◆ 对于透传模块, 建议采用缺省值。</li><li>◆ 该参数修改后, 若需要保存到EEPROM, 需要执行SAVE指令</li></ul>	
命令	AT+PORT=X1	X1为所使用port, 数据格式为16进制, 取值范围: 0x15~0xDF 注1: Port:0x00 LoRaWAN的MAC命令 注2: 0x01~0x14为CLAA保留使用, 用户不可使用 注3: CLAA规范中, 透传模块固定为0x0A
响应1	OK	设置成功
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+PORT=20  OK	设置应用端口为0x20

## 4.2.19 读取 PORT

读取PORT		
描述	读取port端口号, 数据格式为16进制	
命令	AT+PORT?	返回值如“设置PORT”
响应1	+PORT:X1  OK	读取成功。返回结果
响应2	ERROR	未知错误
示例	AT+PORT?  +PORT:20  OK	模块当前PORT为0x20

## 4.2.20 设置 CSQ

### 设置查询CSQ的频率\信道\全信道

描述	◆ 该命令设置模块执行RSSI扫描。返回值RSSI表示是当前环境的噪声	
	◆ 参数X1分为三种格式可选，分别为频率\信道\全信道	
	◆ CSQ结果通过执行AT+CSQ?获得	
命令	AT+CSQ=X1	
	格式1：  X1为频率，单位为Hz，取值范围： 433000000~525000000。	
	格式2  X1也可以为信道，取值范围：0~80。 (注：信道的实际最大值取决于当前CLAA模式)	
响应1	OK	
	设置成功	
	BAD PARM	
响应2	命令参数错误	
	ERROR	
	未知错误	
示例1	AT+CSQ=482300000	
	设置查询频率为482300000的RSSI	
	OK	
示例2	AT+CSQ=1	
	设置查询信道1的RSSI	
	OK	
示例3	AT+CSQ=-1	
	设置执行一次全信道扫描。	

## 4.2.21 读取 CSQ

### 读取CSQ

描述	该命令读取信道的信号质量强度。	
	该命令根据” 设置CSQ” 参数，如果命令正确，会产生两种不同的响应	
命令	AT+CSQ?	
	返回信道当前的RSSI，单位dBm。返回值取决于“设置CSQ信道”命令。	
响应1	+CSQ: [X1]: Y2	读取成功，如果执行AT+CSQ=X1，返回当前信道X1的RSSI

	OK	X1 为频率值, 单位Hz, 或者信道值 Y2 为RSSI值, 单位dBm
响应2	+CSQ: [0]: Y1, [1]: Y2, [2]: Y3, [3]: Y4, ... OK	读取成功, 如果执行AT+CSQ=-1, 则返回当前模块所有允许信道的RSSI, 单位dBm
响应3	ERROR	未知错误
示例1	AT+CSQ? +CSQ: [482300000]: -122 OK	执行AT+CSQ=482300000后, AT+CSQ?则返回当前频点482300000的RSSI
示例2	AT+CSQ? +CSQ: [0]: -122 OK	执行AT+CSQ=0后, AT+CSQ?则返回当前信道0的RSSI
示例3	AT+CSQ? +CSQ: [0]: -123, [1]: -118, [2]: -122, [3]: -122, ... OK	执行AT+CSQ=-1后, AT+CSQ?则返回当前模块所有允许信道的RSSI

## 4.2.22 设置 Confirm 帧的发送次数

### 设置Confirm的重发次数

描述	<ul style="list-style-type: none"><li>◆ 该命令设置Confirm帧的发送次数。</li><li>◆ 该参数仅针对Confirm帧有效。</li><li>◆ 模块每发送一次Confirm帧, 如果没有接收到服务器的ACK, 则自动重发(X1-1)次, 直至接收到ACK或者到达发送次数X1。</li></ul>	
命令	AT+NBTRIALS=X1	X1为发送次数, 取值范围: 1~8 当前缺省值为8

响应1	OK	设置成功,
响应2	PARAMETER ERROR	参数错误
响应3	ERROR	未知错误
示例	AT+NBTRIALS=8 OK	设置Confirm帧的发送次数为8

### 4.2.23 读取 Confirm 帧的发送次数

读取Confirm发送次数		
描述	该命令读取Confirm帧的发送次数	
命令	AT+NBTRIALS?	读取Confirm帧的重发次数。
响应1	+NBTRIALS:X1 OK	读取成功。返回结果
响应2	ERROR	未知错误
示例	AT+NBTRIALS? +NBTRIALS:8 OK	当前Confirm帧的发送次数为8次

### 4.2.24 设置设备类型

设置设备类型		
描述	该命令设置模块的设备类型	
命令	AT+CLASS=X1	X1为所使用设备类型, 取值范围: 0 Class A 2 Class C 出厂缺省值为0
响应1	OK	设置成功
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+CLASS=0 OK	设置设备类型为Class A

### 4.2.25 读取设备类型

读取设备类型

描述	读取设备类型	
命令	AT+CLASS?	返回模块当前所处的设备类型  0   Class A  2   Class C
响应1	+CLASS:X1  OK	读取成功. 返回结果
响应2	ERROR	未知错误
示例	AT+CLASS?  +CLASS:0  OK	当前模块设备类型为CLASS A

## 4.2.26 设置发送功率

### 设置发送功率

描述	<ul style="list-style-type: none"><li>◆ 该命令设置模块的缺省发送功率。</li><li>◆ 注意如果开启ADR功能，模块在运行过程中，服务器会根据ADR机制动态调整模块的发送功率</li><li>◆ 该参数修改后，若需要保存到EEPROM，需要执行SAVE指令</li></ul>	
命令	AT+POWER=X1	X1为发送功率大小, 取值范围：  0   20dBm  1   17dBm  2   16dBm  3   14dBm  4   12dBm  5   10dBm  6   7dBm  7   5dBm  8   2dBm  出厂缺省值为0 (20dBm)
响应1	OK	设置成功
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+POWER=0  OK	设置设备发送功率为20dBm



## 4.2.27 读取发送功率

读取发送功率		
描述	读取模块的当前发送功率	
命令	AT+POWER?	返回模块当前所处的设备类型
		0 20dBm
		1 17dBm
		2 16dBm
		3 14dBm
		4 12dBm
		5 10dBm
		6 7dBm
		7 5dBm
		8 2dBm
响应1	+POWER: X1 OK	读取成功。返回结果
响应2	ERROR	未知错误
示例	AT+POWER?	当前模块功率为20dBm
	+POWER: 0	
	OK	

## 4.2.28 设置通信速率

设置模块的通信速率		
描述	该命令设置模块的通信速率，该命令只有在关闭ADR的情况下，有效 该参数修改后，若需要保存到EEPROM，需要执行SAVE指令	
命令	AT+DATARATE=X1	X1为速率值，取值范围：
		0 SF12, BW125
		1 SF11, BW125
		2 SF10, BW125
		3 SF9, BW125
		4 SF8, BW125
		5 SF7, BW125
		出厂缺省值为0 (SF12)

响应1	OK	设置成功
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+ DATARATE =0 OK	设置模块通信速率为SF12, BW125

### 4.2.29 读取通信速率

读取通信速率		
描述	读取模块通信速率	
命令	AT+DATARATE?	返回模块当前通信速率 0 SF12, BW125 1 SF11, BW125 2 SF10, BW125 3 SF9, BW125 4 SF8, BW125 5 SF7, BW125
响应1	+DATARATE:X1 OK	读取成功. 返回结果
响应2	ERROR	未知错误
示例	AT+DATARATE?  +DATARATE:0  OK	模块当前速率为SF12, BW125

### 4.2.30 读取状态寄存器

读取状态寄存器		
描述	该命令读取模块的状态信息 AT+STATUS?返回值包含上行结果与下行结果部分内容	
命令	AT+STATUS?	返回模块状态信息。详细见表2-6
响应1	+STATUS0:Y0,Y1...Y39 OK	读取状态成功。
响应2	ERROR	未知错误

### 4.2.31 读取上行结果寄存器

#### 读取上行结果寄存器

描述	该命令读取模块的上行结果	
命令	AT+STATUS0?	返回模块上行结果。详细见表2-7
响应1	+STATUS0:Y0,Y1...Y12 OK	读取状态成功。
响应2	ERROR	未知错误

### 4.2.32 读取下行结果寄存器

#### 读取下行结果寄存器

描述	该命令读取模块的下行结果	
命令	AT+STATUS1?	返回模块下行结果。详细见表2-8
响应1	+STATUS1:Y0,Y1...Y17 OK	读取下行结果成功。
响应2	ERROR	未知错误

### 4.2.33 设置 JOIN

#### 设置JOIN信息

描述	该命令设置JOIN，该命令可选带多个命令参数	
命令	格式1 AT+JOIN=X1	X1 表示执行JOIN操作，X1取值范围： -1 - 该参数无效
	格式2 AT+JOIN=X1, X2	0 - 停止JOIN 1 - 重启JOIN，如果模块在此之前已成功JOIN网络， 志星该操作，将清除之前的JOIN上下文参数，重新开启 JOIN过程。
	格式3 AT+JOIN=X1, X2, X3	X2 表示是否开启自动JOIN功能。X2取值范围： 0 - 关闭自动JOIN 1 - 自动JOIN. 模块进入透传模式后，自动启动JOIN。 缺省值：1。
	格式4 AT+JOIN=X1, X2, X3, X4	X3 表示JOIN周期，X3取值范围：7~255，单位为s。 缺省值：8。 X4 表示JOIN最大尝试次数，X4取值范围：1~255。

		缺省值：176。该缺省值可以进行2次多模扫描。（如果是单模模式，建议最大尝试次数设置为每种模式一轮扫描次数的整数倍）
响应1	OK	设置成功
响应2	ERROR	未知错误
示例1	AT+JOIN=1 OK	重新开启JOIN，
示例2	AT+JOIN=-1, 1 OK	设置JOIN参数：使能自动JOIN
示例3	AT+JOIN=-1, 1, 10, OK	设置JOIN参数：使能自动JOIN，JOIN周期为10s
示例4	AT+JOIN=-1, 1, 8, 27, OK	使能自动JOIN，JOIN周期为8s，JOIN最大尝试次数为27（比如当前为CLAA A模式，则可以执行3轮扫描，总耗时216s）

#### 4.2.34 读取 JOIN 信息

##### 读取JOIN信息

描述	该命令读取JOIN信息	
命令	AT+JOIN?	<p>读取JOIN信息Y1, Y2</p> <p>Y1表示当前JOIN结果：</p> <p>0 - 模块未JOIN</p> <p>1 - 模块已JOIN</p> <p>X2 表示是否使能自动JOIN，单位为s</p> <p>X3 表示JOIN周期，单位为s</p> <p>X4 表示JOIN最大尝试次数</p> <p>Y5 表示当前为第几次发送JOIN</p> <p>Y6 表示成功加入网络的信道</p>
响应1	+JOIN: Y1, X2, X3, X4, Y5, Y6 OK	读取成功。返回结果
响应2	ERROR	未知错误
示例	AT+JOIN?  +JOIN: 1, 1, 8, 176, 10, 30  OK	表示模块在第10次，成功加入网络，对应的信道为30

## 4.2.35 设置 Link Check

设置 Link Check		
描述	<ul style="list-style-type: none"> <li>该命令指示模块进行Link Check。</li> <li>该功能可以用来检测网络是否正常，类型以太网中的ping功能。</li> <li>该命令会产生两条响应</li> </ul>	
命令	格式1:	X1 为Link Check使能控制
	AT+LINK CHECK=X1	0 - 不使能Link Check 1 - 执行一次Link Check
响应1	OK	设置成功。
	等待一段时间后，会返回第二条响应信息：	
	+LINK INFO:OK, Y1, Y2, Y3, Y4	OK 表示本次Link Check执行成功
	或者	Y1 为 DemodMargin
	+LINK INFO:FAIL	Y2 为 NbGateways
		Y3 为 当次下行的RSSI
		Y4 为 当次下行的SNR
	FAIL 表示本次Link Check执行失败，无返回值	
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	AT+LINK CHECK=1	模块执行一次Link Check操作
	OK	(等待网络交互)
	+LINK INFO:OK, 0, 1, -68, 8	模块返回LINK CHECK结果。

## 4.2.36 读取 Link Check

读取 Link Check配置		
描述	该命令读取Link Check配置	
命令	AT+LINK CHECK?	读取Link Check参数配置
响应1	+LINK CHECK: X1, X2	读取成功。返回结果
	OK	X1 为是否执行Link Check X2 自动执行Link Check周期，单位s
响应2	ERROR	未知错误
示例	AT+LINK CHECK?	
	+LINK CHECK: 1, 100	

OK

## 4.2.37 设置 BAND

### 设置BAND

- 描述**
- ◆ 该命令设置模块的入网扫描模式，
  - ◆ CLAA支持五种模式 (CLAA A\B\C\D\E)，不同CLAA模式对应频段不相同。模块支持两种扫描方式：单模与多模。
  - ◆ 模块出厂默认采用多模扫描网络。
  - ◆ 该参数修改后，若需要保存到EEPROM，需要执行SAVE指令

<b>命令</b>	格式1	X1对应CLAA工作模式, 取值范围:
	AT+BAND=X1	0 CLAA MODE A 1 CLAA MODE B 2 CLAA MODE C 3 CLAA MODE D 4 CLAA MODE E
	格式2	X2选择为单模或者多模模式 (MODE A\B\C\D\E), 取值范围:
	AT+BAND=X1, X2	0 单模模式, 固定在指定的模式扫描网络 1 多模模式, 开启将会自动搜索所有模式

<b>响应1</b>	OK	设置成功
<b>响应2</b>	BAD PARM	命令参数错误
<b>响应3</b>	ERROR	未知错误
<b>示例1</b>	AT+BAND=3 OK	设置为CLAA MODE D, 模块优先使用MODE D开始搜索网络（缺省为多模模式, 因此仍然执行多模扫描）。
<b>示例2</b>	AT+BAND=3, 0 OK	设置为CLAA MODE D, 模块只工作在CLAA MODE D扫描网络

## 4.2.38 读取 BAND 模式

### 读取BAND

<b>描述</b>	该命令模块读取当前BAND，对应CLAA模式	
<b>命令</b>	AT+BAND?	返回当前模式。X1对应CLAA MODE A\B\C\D\E
<b>响应1</b>	+BAND: X1, X2 OK	读取成功, 返回结果
<b>响应2</b>	ERROR	未知错误

示例	AT+BAND?	当前模块工作在CLAA D，多模扫描方式
	+BAND: 3, 1	
	OK	

### 4.2.39 NOP 命令

#### 读取JOIN信息

**描述** 该命令可以用于测试模块与用户MCU的串口通信是否正常

**命令** AT NOP命令

**响应1** OK 设置成功

**响应2** 无响应 未知错误，比如串口通信异常

**示例** AT 串口通信正常  
OK

### 4.2.40 参数保存

#### SAVE保存配置参数

**描述** 该命令保存配置参数到EEPROM中

在执行AT+RESET命令后，模块将使用新的配置参数进行网络初始化与运行

如下配置参数会保存到EEPROM：

类型	EEPROM参数	说明
SYS	Baud	
	Receive Timeout	
	Transmit Gap	
LORAWAN	APPEUI	
	APPKEY	
	APPSKEY	
	NWKSKEY	
	DEVADDR	
	ADR	
	OTAA	
	CONFIRM	
	BAND	
	CLASS	设备类型

		DATARATE		
		POWER		
		PORT		
命令	AT+SAVE	执行命令		
响应1	OK	执行完成		
响应2	ERROR	未知错误		
示例	AT+SAVE	保存成功		
	OK			

内部资料



## 5 低功耗处理

用户在透传模式，通过拉低 WAKE 引脚，可以使模块进入睡眠。模块进入睡眠后，各引脚状态，如表 5-1 所示。

表 5-1 模块睡眠的引脚状态

模块				用户 MCU
引脚	引脚功能	端口类型	引脚状态	用户 MCU 引脚处理建议
7	WAKE	Input	Float	用户 MCU 拉低 WAKE 引脚，并保持
16	MODE	Input	PULL-DOWN	用户 MCU 拉低 MODE 引脚，并保持
18	TXD	Output	HIGH	用户 MCU 浮空输入
19	RXD	Input	Float	用户 MCU 拉低 RXD 引脚，并保持
15	BUSY	Output	LOW	用户 MCU 浮空输入
8	STAT	Output	HIGH/LOW	用户 MCU 浮空输入，该引脚状态取决于模块的当前状态
14	NRST	Input	PULL-UP	用户拉高或者浮空输入
11	P0	Output	LOW/用户配置	如果用户 MCU 配置了该引脚，以实际为准
3	P1	Output	LOW/用户配置	如果用户 MCU 配置了该引脚，以实际为准
4	P2	Output	LOW/用户配置	如果用户 MCU 配置了该引脚，以实际为准
5	P3	Output	LOW/用户配置	如果用户 MCU 配置了该引脚，以实际为准

## 6 应用配置范例

用户需要根据不同应用场景，来配置不同的参数。其中 6.1 为基本配置，其他小节

用户根据需求选择。

### 6.1 OTAA模式配置

下文描述了激活模式最小需要配置\修改的参数（如果未配置，将按照默认参数运行），其他特性参数可以根据用户需求设置。

#### 1、进入指令模式

拉高 MODE 引脚

#### 2、配置 OTAA 模式

AT+OTAA=1

#### 3、配置 AppEui，(该参数 FF 00 66 FF FF FF 01 仅做为示例说明)

AT+APPEUI= FF 00 66 FF FF FF 01

4、配置 AppKey，(该密钥 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 仅做为示例说明)

AT+APPKEY= 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

#### 5、保存，(以上任何参数发生变化，需要执行 SAVE 并复位后，才能生效)

AT+SAVE

#### 6、复位

AT+RESET 或者拉低 Reset 引脚，并保持 1ms

#### 7、进入透传模式，(请注意：只有在透传模式下，模块才会执行 JOIN 操作)。

拉低 MODE 引脚

#### 8、用户等待 BUSY 引脚与 STAT 引脚为高电平，表明模块加入成功。用户可以通

过串口收发数据。

## 6.2 CLAA模式配置

模块出厂默认为多模，即模块首次入网会按照既定的优先级来搜索基站。目前优先级为 D->E->B->C->A。

### 6.2.1 应用场景 1：单模模式

如果用户确认所在区域的基站运行的 CLAA 模式(CLAA A\B\C\D\E 模式)，且基站模式不会永久不会改变，则可以设置模块永久工作在指定的模式。

特殊说明：2016 年 10 月份以前批次的基站与模块只工作在 CLAA A 模式。

例如：

当前基站确定永久工作在 CLAA A 模式，且该设备为非移动设备，相对应地，可以设置模块固定工作在 CLAA A 模式。配置方法，如下：

1、指令模式下，输入 AT+BAND=0,0

### 6.2.2 应用场景 2：多模模式

如果用户确认所在区域的基站运行的 CLAA 模式(CLAA A\B\C\D\E 模式)，但是基站后续可能升级成其他 CLAA 模式，则可以设置模块优先工作在指定模式，并允许模块进行全网扫描（多模模式）。

例如：

当前基站工作在 CLAA A 模式，但是根据业务需求，基站可能后续会升级成新的 CLAA 模式，或者该设备为移动设备，相对应地，可以设置模块优先工作在 CLAA A 模式。配置方法，如下：

2、指令模式下，输入 AT+BAND=0,0

## 6.3 Class C应用

对于终端不需要进行低功耗的应用，且希望获得更实时的下行控制，用户可以通过

设置模块的设备类型为 Class C。配置方法，如下：

- 1、保持 WAKE 引脚为高
- 3、指令模式下，输入 AT+CLASS=2

## 敬告用户

1、欢迎您使用利尔达科技有限公司的产品，在使用我公司产品前，请先阅读此敬告；如果您已开始使用说明您已阅读并接受本敬告。

利尔达科技有限公司保留所配备全部资料的最终解释和修改权，如有更改恕不另行通知。

编制：利尔达科技集团股份有限公司 无线传感网

2016 年 10 月