

# LSD4WN-2R717M90

## 产品使用说明书

文件版本: Rev01

文件状态: RC

# 目录

<b>1</b>	<b>概述</b>	<b>- 1 -</b>
1.1	LoRaWAN 简介	- 1 -
1.1.1	LoRaWAN 设备类型	- 2 -
1.2	文档适用范围	- 2 -
1.3	参考	- 3 -
1.3.1	LoRa Alliance	- 3 -
1.3.2	协议标准	- 3 -
<b>2</b>	<b>工作机制</b>	<b>- 4 -</b>
2.1	系统状态	- 5 -
2.2	工作模式	- 5 -
2.3	上电/复位初始化	- 6 -
2.4	参数掉电存储	- 6 -
2.5	串口通信流	- 6 -
2.6	典型应用电路	- 8 -
<b>3</b>	<b>指令模式</b>	<b>- 9 -</b>
3.1	AT 协议	- 9 -
3.2	网络激活配置	- 10 -
<b>4</b>	<b>透传模式</b>	<b>- 11 -</b>
4.1	引脚配置	- 11 -
4.2	搜索网络	- 12 -
4.2.1	热启动	- 12 -
4.3	用户数据发送	- 13 -
4.3.1	流控机制	- 13 -
4.3.2	物理分包	- 14 -
4.4	服务器响应	- 14 -
4.5	空口通信流	- 16 -
<b>5</b>	<b>引脚时序</b>	<b>- 17 -</b>
5.1	上电（复位）时序	- 17 -
5.2	唤醒时序	- 18 -
5.3	串口帧分割时序	- 18 -
5.4	指令操作时序	- 18 -

5.5	数据发送时序.....	- 19 -
5.6	数据接收时序.....	- 19 -
5.7	数据通信时序.....	- 20 -
<b>6</b>	<b>AT 指令集 .....</b>	<b>- 21 -</b>
6.1	指令处理时间.....	- 21 -
6.2	系统指令 .....	- 22 -
6.2.1	空指令 NOP .....	- 22 -
6.2.2	恢复出厂设置 FACTORY .....	- 22 -
6.2.3	设置 IO 口状态 GPIO .....	- 23 -
6.2.4	设置外设 PERIPH.....	- 23 -
6.2.5	读取版本 VER .....	- 24 -
6.2.6	设置日志信息输出 DEBUG .....	- 25 -
6.2.7	参数保存 SAVE .....	- 25 -
6.2.8	系统复位 RESET .....	- 25 -
6.2.9	设置串口波特率 BAUD .....	- 26 -
6.2.10	读取串口波特率 BAUD.....	- 26 -
6.2.11	设置串口分帧时间 TIMEOUT .....	- 27 -
6.2.12	读取串口分帧时间 TIMEOUT .....	- 27 -
6.2.13	设置实时时间 RTC .....	- 28 -
6.2.14	读取实时时间 RTC .....	- 28 -
6.2.15	设置系统参数 SYSTEM .....	- 29 -
6.2.16	读取系统参数 SYSTEM.....	- 29 -
6.3	MAC 指令 .....	- 30 -
6.3.1	设置上行传输类型 CONFIRM .....	- 30 -
6.3.2	读取上行传输类型 CONFIRM .....	- 30 -
6.3.3	设置激活模式 OTAA.....	- 31 -
6.3.4	读取激活模式 OTAA.....	- 31 -
6.3.5	设置 DEVEUI .....	- 32 -
6.3.6	读取 DEVEUI .....	- 32 -
6.3.7	设置 APPEUI .....	- 33 -
6.3.8	读取 APPEUI .....	- 33 -
6.3.9	设置 APPKEY .....	- 34 -
6.3.10	读取 APPKEY .....	- 34 -
6.3.11	设置 DEVADDR.....	- 35 -

6.3.12	读取 DEVADDR.....	- 35 -
6.3.13	设置 APPSKEY .....	- 36 -
6.3.14	设置 NWKSKEY .....	- 36 -
6.3.15	设置 PORT.....	- 37 -
6.3.16	读取 PORT.....	- 37 -
6.3.17	设置设备类型 CLASS .....	- 38 -
6.3.18	读取设备类型 CLASS .....	- 38 -
6.3.19	设置 ADR .....	- 39 -
6.3.20	读取 ADR .....	- 39 -
6.3.21	设置发送功率 POWER.....	- 40 -
6.3.22	读取发送功率 POWER.....	- 40 -
6.3.23	设置通信速率 DATARATE.....	- 41 -
6.3.24	读取通信速率 DATARATE.....	- 41 -
6.3.25	设置 BAND .....	- 42 -
6.3.26	读取 BAND .....	- 42 -
6.3.27	设置 FREQ .....	- 43 -
6.3.28	读取 FREQ .....	- 44 -
6.3.29	设置注册入网信息 JOIN .....	- 46 -
6.3.30	读取注册入网信息 JOIN .....	- 47 -
6.3.31	设置状态寄存器 Status_Output_Reg.....	- 48 -
6.3.32	读取状态寄存器 Comm_Status_Reg .....	- 49 -
6.3.33	设置 Link Check.....	- 50 -
6.3.34	读取 Link Check.....	- 51 -
6.3.35	设置 RX2 .....	- 52 -
6.3.36	读取 RX2 .....	- 52 -
6.3.37	设置 CSMA .....	- 53 -
6.3.38	读取 CSMA .....	- 53 -
6.3.39	时间同步 TIMESYNC .....	- 54 -
6.3.40	Class C 组播 .....	- 54 -
7	<b>低功耗处理 .....</b>	<b>- 55 -</b>
8	<b>应用配置说明.....</b>	<b>- 56 -</b>
8.1	出厂参数说明.....	- 56 -
8.2	OTAA 快速配置示例 .....	- 58 -
8.3	Class C 应用 .....	- 59 -

8.3.1	应用场景 .....	- 59 -
8.3.2	典型配置 .....	- 59 -
8.3.3	组播配置 .....	- 59 -
<b>9</b>	<b>日志信息 .....</b>	<b>- 60 -</b>
9.1	注册入网过程.....	- 61 -
9.2	数据传输过程.....	- 62 -
<b>10</b>	<b>配套资源 .....</b>	<b>- 63 -</b>
10.1	LoRaWAN 开发板 .....	- 63 -
10.2	开发板配套程序 .....	- 63 -
10.3	资源下载.....	- 63 -
<b>11</b>	<b>产品变更记录.....</b>	<b>- 64 -</b>
<b>12</b>	<b>文件修订历史.....</b>	<b>- 65 -</b>
<b>13</b>	<b>敬告用户 .....</b>	<b>- 66 -</b>

## 图片索引

图 1-1	LoRaWAN 设备类型.....	- 1 -
图 1-2	Class A 设备通信机制.....	- 2 -
图 1-3	Class C 设备通信机制.....	- 2 -
图 2-1	LSD4WN-2R717M90 固件功能框图.....	- 4 -
图 2-3	模块与用户 MCU 的串口通信流.....	- 7 -
图 2-4	LSD4WN-2R717M90 应用电路.....	- 8 -
图 4-1	用户数据发送流程.....	- 13 -
图 4-2	模块的空口通信流.....	- 16 -
图 5-1	开机时序.....	- 17 -
图 5-2	模块唤醒时序.....	- 18 -
图 5-3	帧超时分割时序.....	- 18 -
图 5-4	指令操作时序.....	- 18 -
图 5-5	串口发送时序.....	- 19 -
图 5-6	数据接收时序.....	- 19 -
图 5-7	模块透传模式的工作时序.....	- 20 -
图 6-1	指令处理时间.....	- 21 -

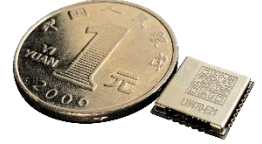
## 表格索引

表 1-1	LoRaWAN 的终端类型.....	- 2 -
表 2-1	模块的系统状态控制.....	- 5 -
表 2-2	激活状态下的模式控制.....	- 5 -
表 2-3	串口通信的消息类型.....	- 6 -
表 3-1	AT 协议.....	- 9 -
表 3-2	AT 指令返回值.....	- 10 -
表 3-3	网路激活方式及其所需参数.....	- 10 -
表 4-1	透传模式的引脚功能.....	- 11 -
表 4-2	模块重新入网的三种方式.....	- 12 -
表 4-3	不同速率对应的最大负载值 N.....	- 14 -
表 4-5	数据帧内容释义.....	- 15 -
表 6-1	Status_Output_Reg 的功能索引.....	- 48 -
表 6-2	Comm_Status_Reg 内容释义.....	- 49 -
表 7-1	模块睡眠的引脚状态.....	- 55 -

表 8-1	模块出厂缺省参数.....	- 56 -
表 9-1	日志信息.....	- 60 -
表 9-2	状态定义.....	- 60 -
表 9-3	模块注册信息.....	- 61 -

# 1 概述

LSD4WN-2R717M90 模块是利尔达科技集团股份有限公司研制的一款“极”系列 LoRaWAN 终端模块。本模块集成了 LoRaWAN™ 协议栈，符合 LoRa Alliance 发布的 LoRaWAN™ Specification 1.0.3 Class A/C 标准。硬件支持 470-510MHz 频段。



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通信等业务。

本说明书描述了 LSD4WN-2R717M90 模块的工作机制，以及模块与用户端交互的 AT 指令集。

## 1.1 LoRaWAN简介

LoRaWAN 属于一种低功耗广域网网络(LPWAN)规范，适用于多种物联网低功耗(IoT)应用场合，如自动抄表、智慧城市、智能家居、工业自动化等。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(Backend central server)。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，所有的节点与网关之间均是双向通信。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。

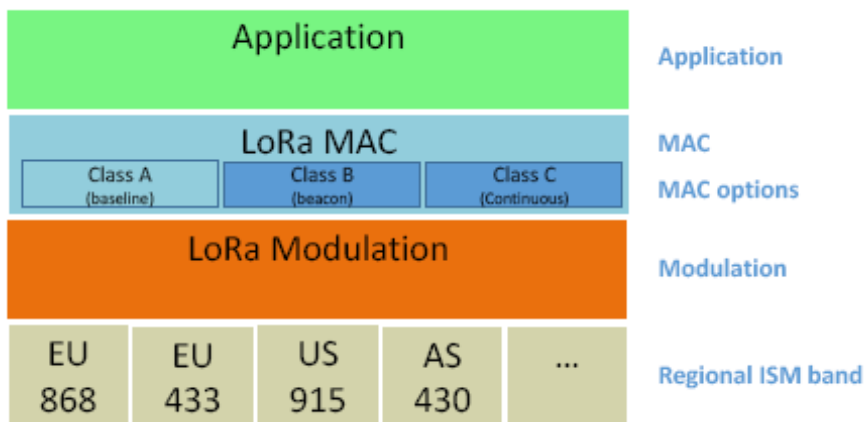


图1-1 LoRaWAN设备类型



### 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, 具体如表 1-1 所示; 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, 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设备功耗最低, 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, 终端设备会从网关接收到一个Beacon, 以此获得同步, 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 1-2 所示:

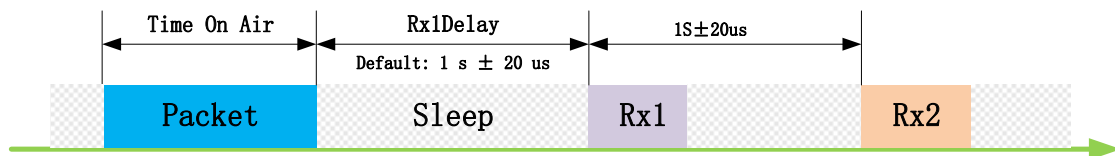


图1-2 Class A设备通信机制

Class C 设备通信机制, 如图 1-3 所示:

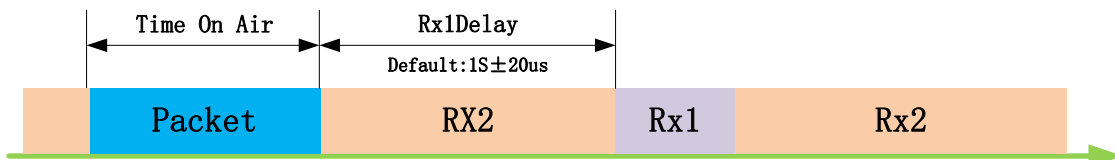


图1-3 Class C设备通信机制

## 1.2 文档适用范围

本说明书适用的固件版本: LoRaWAN1.0.3\_CN470P.21-02.22.01 及后续版本

## 1.3 参考

### 1.3.1 LoRa Alliance

LoRa™ Alliance 成立于 2015 年 3 月，是一个由行业内领先厂商发起的开放、非盈利组织。LoRa 联盟致力于建立一个安全的、运营级的低功耗广域网(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。自 2015 年 3 月成立以来，截止 2017 年年底，全球目前已有 500+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。

由 LORA 联盟开发并支持的低功耗广域(LPWAN)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。

### 1.3.2 协议标准

1. 《LoRaWAN\_Specification\_-v1.0.3》由 LoRa 联盟发布，详见 <http://www.lora-alliance.org>
2. 《LoRaWAN Regional Parameter V1.1rb》由 LoRa 联盟发布，详见 <http://www.lora-alliance.org>

## 2 工作机制

LSD4WN-2R717M90 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 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传输等功能。模块的固件功能框图, 如图 2-1 所示。

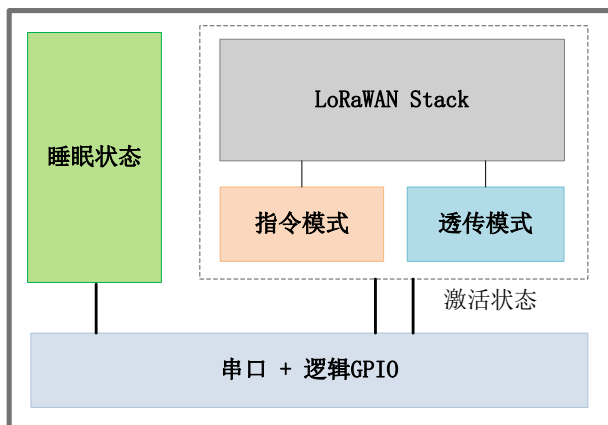


图2-1 LSD4WN-2R717M90固件功能框图

模块引脚的功能定义如图 2-2 所示, 具体引脚定义详见《DS\_LSD4WN-2R717M90 ( SX1268 LoRaWAN End Node ) 产品规格书》。

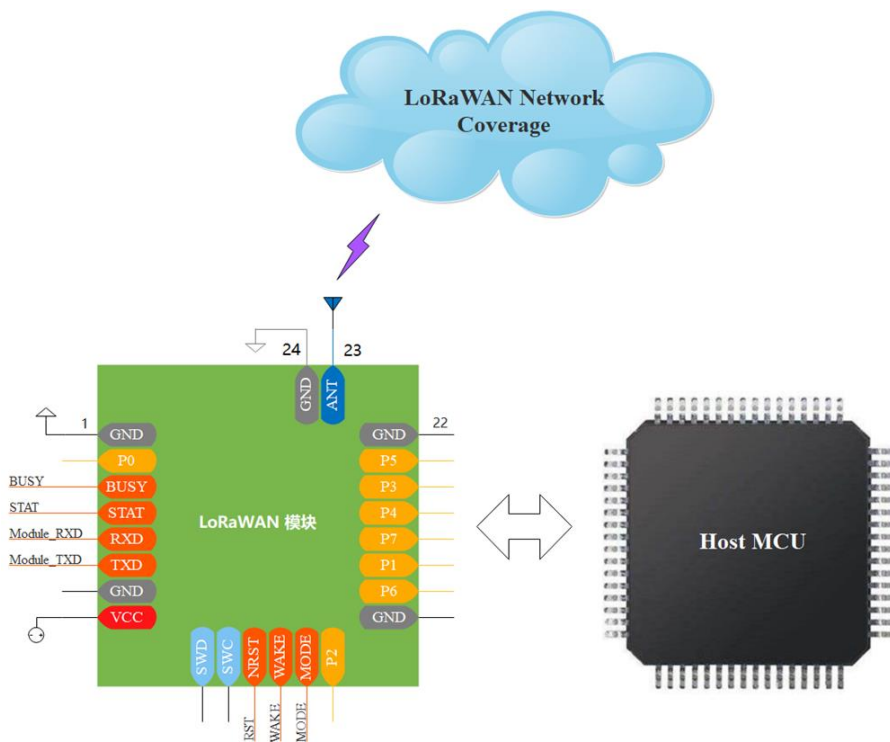


图 2-2 模块应用框图

## 2.1 系统状态

模块定义了两种系统状态，分别是睡眠状态与激活状态。

(1) WAKE 引脚被拉低，模块进入睡眠状态。此状态下用户无法执行任何指令、数据通信等操作，但模块会保存系统当前的运行状态、配置参数等。

(2) WAKE 引脚被拉高，模块进入激活状态。此状态下用户可进行 LoRaWAN 数据收发、参数配置等操作。

用户通过 WAKE 引脚来选择模块的系统状态，如表 2-1 所示。

表2-1 模块的系统状态控制

系统状态控制脚	I/O 方向	描述
WAKE = 1 保持高电平	Input	激活状态
WAKE = 0 保持低电平		睡眠状态

## 2.2 工作模式

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：

(1) 指令模式主要用于参数配置、状态读取等。在指令模式下，用户发送完一条参数配置或状态读取的指令时，都需等待模块返回“OK”后，再进行下一条指令操作，否则可能会导致本条指令无效。

(2) 透传模式主要用于发送与接收应用数据。在透传模式下，若模块被选定为 OTAA 激活，在未加入网络的情况下，默认会自动执行加入网络 (JOIN) 操作。模块正常加入网络，才可以提供有效的 LoRaWAN 数据服务。在与服务器通信过程中，根据模块与服务器两者之间的通信状态，可分为连接状态与连接丢失状态，详细介绍在第 4 章“[4.5 网络通信状态](#)”中说明。

用户通过 MODE 引脚<sup>注 2-1</sup>来选择模块的工作模式，如表 2-2 所示。

表2-2 激活状态下的模式控制

功能引脚	I/O 方向	描述
MODE = 1 保持高电平	Input	指令模式
MODE = 0 保持低电平		透传模式

注 2-1：

非法操作！睡眠状态下，即 WAKE 引脚为低电平，用户不允许切换 MODE 引脚状态。

## 2.3 上电\复位初始化

当模块上电\复位后，模块会进行初始化操作，此时 BUSY 引脚输出低电平，表示模块正忙，用户不能对模块进行操作；当初始化完成后，BUSY 引脚输出高电平，表示模块在空闲状态，用户可对模块进行相应操作。

若模块为首次上电，建议用户配置 MODE 引脚为高电平，使模块先进入指令模式进行参数配置，然后再切换至透传模式进行注册入网。

## 2.4 参数掉电存储

用户在指令模式下完成系统层以及 MAC 层参数配置后，若用户需要对设置的参数进行掉电存储，则可执行“AT+SAVE”指令，将配置的参数更新到模块 EEPROM 中。

模块重新上电\复位（软复位\硬复位\POR）后，将按照新参数运行（支持掉电存储的参数，详见“第 6 章 AT 指令集”）。

## 2.5 串口通信流

模块与用户 MCU 的串口通信流，主要包含了指令、应答、事件三种消息类型。其中指令、应答类消息主要用于参数配置等操作，事件类消息主要用于上行\下行数据通信等，如表 2-3 所示。

表2-3 串口通信的消息类型

消息类型	通信方向	主要功能	描述
指令	模块 <-- 用户MCU	参数配置	用户MCU发送指令给模块
应答	模块 --> 用户MCU		模块返回对应指令的应答给用户MCU
事件	模块 <-> 用户MCU	数据通信服务	任意时刻发生的异步事件： 用户发送上行数据 <sup>注2-1</sup> 给模块； 模块发送下行数据 <sup>注2-2</sup> 给用户MCU； 模块发送失败报告等

注 2-1：

上行数据指的是终端通过 LoRaWAN 网络发送的用户数据

注 2-2：

下行数据指的是服务器通过 LoRaWAN 网络下发的用户数据

模块与用户 MCU 之间的串口通信流详细流程如图 2-3 所示。

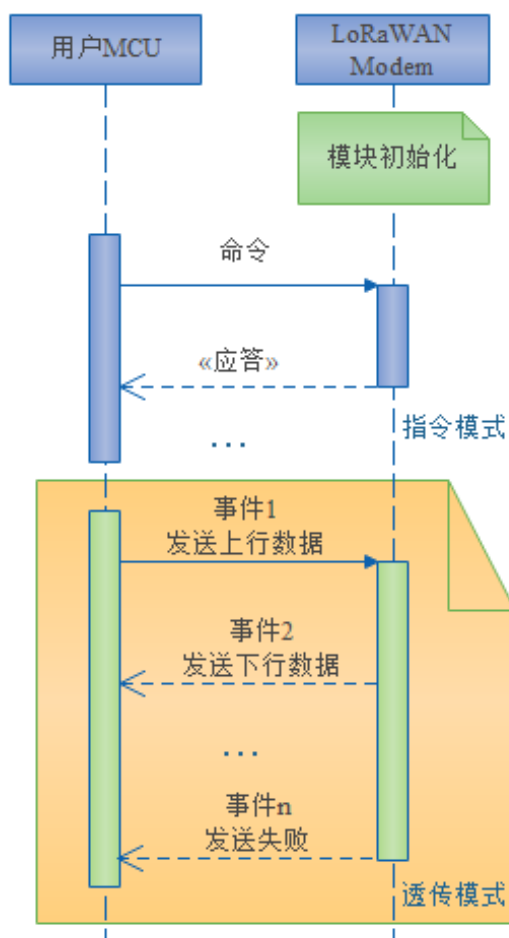


图2-3 模块与用户MCU的串口通信流

## 2.6 典型应用电路

典型应用电路包含了用户端 (Host MCU)、LoRaWAN 模块端 (LSD4WN-2R717M90 (ANT)), 具体如图 2-4 所示。

用户接口：串口、GPIO、电源等。

天线接口：50Ω 邮票孔输出。

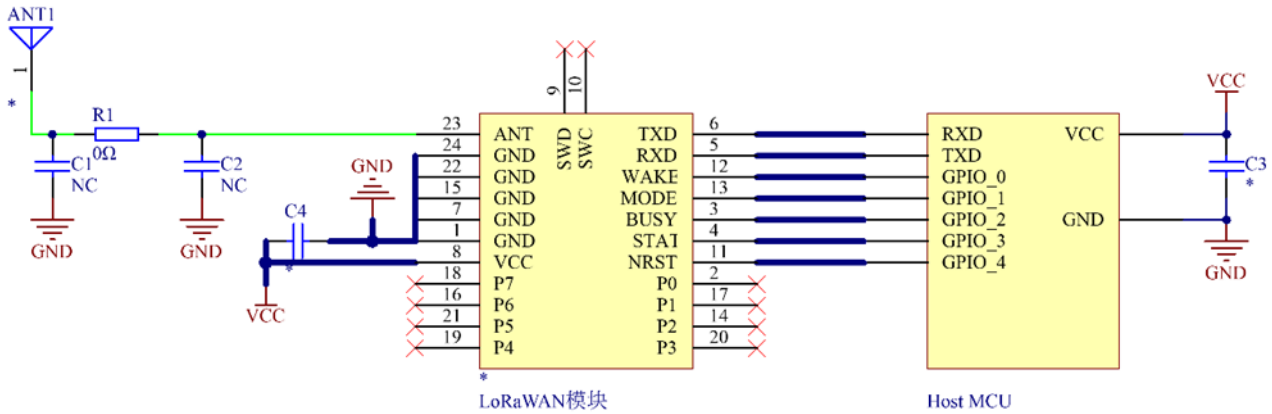


图2-4 LSD4WN-2R717M90应用电路

以下是本应用电路的部分说明。

说明：

1. 该应用电路为显示方便，图中模块引脚封装与模块本身不同
2. 加粗 Trace 为系统所需连接(推荐)
3. 天线出口 (ANT<->PIN22) 的绿色 Trace 要求 50Ω 阻抗匹配
4. 缺省情况下，R1 为 0Ω，C1，C2 为空贴；R1、C1、C2 参数的具体取值，由产品进行天线匹配后确定
5. 天线部分的 Layout 设计，可参考应用文档《射频 PCB LAYOUT 设计规则(适用 sub-1GHZ 及蓝牙模块)》

## 3 指令模式

指令模式主要用于模块参数配置、状态读取等。用户通过串口发送 AT 指令来访问模块，从而进行寄存器的读写配置、状态寄存器的查询等操作。

若模块为首次上电，建议用户先配置 MODE 为高电平，此种情形下，模块上电初始化完成后，BUSY 引脚将输出高电平，指示用户可以进行 AT 指令操作；一般而言，在指令模式下，BUSY 引脚的电平变化不影响用户对模块的操作，仅当执行个别指令需要对 BUSY 引脚做出判断，具体请参考“[第 6 章 AT 指令集](#)”。

在指令模式下，串口参数固定为 9600N81 注 3-1。

用户发送 AT 指令给模块，模块进行指令解析，解析完成后立即返回一个指令响应，表征指令的执行结果；若模块长时间无回应，表示模块状态异常。

在 BUSY 为低电平(忙)的状态下，用户可切换至指令模式，读取系统状态寄存器等有效信息，查看模块当前状态。

注 3-1:

串口参数详细说明：波特率 9600bps，无校验，数据位 8 位，停止位 1 位。

### 3.1 AT协议

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 3-1 所示。

表3-1 AT协议

符号	描述
指令格式	设置指令：AT+[...x] = Y 或 AT+[...x] 读取指令：AT+[...x]?
AT	每个AT指令，都以AT开头。不区分大小写
+X1	+X1为指令
= or ?	‘=’，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?
Para1,Para2...	参数分隔符[逗号]，可能带多个参数，比如AT+CLASS=<param>,<param>,<param>
<CR>	回车符，指令结束

AT 指令返回值根据指令结果的不同，具有不同返回值，如表 3-2 所示。



表3-2 AT指令返回值

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

注意:

在指令模式下, 输入 AT 指令, 必须以回车为结束符, 即("\r")或者 (0x0D)。

## 3.2 网络激活配置

为了能够与 LoRaWAN 网络通信, 首次使用模块前, 需进行必要的参数配置与激活操作。有两种激活方式 (当前主要以 OTAA 激活方式为主) 可供选择, 每种方式所需配置的参数有所不同, 如表 3-3 所示。

表3-3 网路激活方式及其所需参数

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:

出厂固件已烧好 DevEui, AppEui 和 AppKey 供用户自行烧写。

## 4 透传模式

透传模式主要用于发送与接收应用数据。在透传模式下，模块若未加入 LoRaWAN 网络，首先执行注册入网，直至成功加入网络；在成功加入网络后，模块可正确处理用户数据，进行有效的空口数据交互与串口数据交互。

### 4.1 引脚配置

模块在透传模式下，相关引脚控制如表 4-1 所示。

表4-1 透传模式的引脚功能

引脚	定义	I/O	行为	电平	描述
13	MODE	Input	透传模式	低电平	当前为透传模式
6	TXD	Output	数据发送	-	模块串口发送端(TX)
5	RXD	Input	数据接收	-	模块串口接收端(RX)
12	WAKE	Input	激活\睡眠	高电平	模块在激活状态，若BUSY引脚为高电平，用户可发送数据 (注：在发送数据过程中不允许将wake引脚拉低)
				低电平	模块在睡眠状态，将停止所有服务
3	BUSY	Output	注册入网	高电平	模块已注册入网成功
				低电平	模块在注册入网过程中
			数据通信	高电平	模块空闲，指示用户MCU可向模块写入数据
				低电平	模块忙，指示用户MCU不允许向模块写入数据
4	STAT	Output	注册入网	高电平	模块已注册入网成功
				低电平	模块在注册入网过程中
			数据通信	高电平	单次空口数据通信发\收成功
				低电平	单次空口数据通信发\收失败

## 4.2 搜索网络

若模块未加入网络（首次入网）并且被设定为 OTAA 激活（缺省），当模块上电/复位后切换至透传模式，会自动执行入网（JOIN）操作。未成功加入网络前，BUSY 与 STAT 引脚始终保持为低电平<sup>注 4-1</sup>；直至成功加入网络，BUSY 与 STAT 引脚输出高电平，指示用户模块已成功加入网络。

模块入网方式有两种：收发同频和收发异频。模块默认采用收发同频的方式，以 475.3MHz 为起始频点、200kHz 为间隔的 8 个连续信道进行入网。入网阶段，模块上下行频率需要和服务端保持一致，否则会出现无法入网的情况。频段配置请参考“[6.3.25 设置 BAND](#)”和“[6.3.27 设置 FREQ](#)”。

在入网阶段，入网速率以初始速率为起点，每 2 次逐级降速至 SF11；若 SF11 还未入网，则回到初始速率，以相同的方式循环入网，直至入网成功。如果模块入网 176 次（默认值）后仍无法加入网络（网络异常等），模块将停止入网操作，同时 BUSY 和 STAT 引脚将保持为低电平。入网参数配置请参考“[6.3.29 设置注册入网信息 JOIN](#)”。

### 4.2.1 热启动

模块默认关闭热启动，成功加入网络后，如果模块被复位，需要重新 JOIN 以加入网络；相反，若模块开启热启动，成功加入网络后，注册上下文被保存，复位后无需重新 JOIN。热启动配置请参考“[6.3.3 设置激活模式 OTAA](#)”。

模块有以下 3 种重新入网方式，如表 4-2 所示。

表4-2 模块重新入网的三种方式

入网操作	功能描述	引脚状态描述	备注
手动执行 AT+JOIN=1	清除注册上下文， 重启模块后生效	注册过程中BUSY和STAT引脚均拉低；注册结束后， 若注册失败，BUSY和STAT引脚保持为低电平，若注册成功，BUSY和STAT引脚均拉高（同首次入网）	热启动开启或关闭均有效
手动执行 AT+JOIN=6	不清除注册上下文， 执行后立即生效	注册过程中BUSY和STAT引脚均拉低；注册结束后， 无论注册成功与否，BUSY和STAT引脚均会拉高	热启动开启或关闭均有效
复位	复位模块	BUSY和STAT引脚状态同“AT+JOIN=1”	热启动关闭有效

#### 注意：

##### 4-1：在入网成功后的通信阶段：

1. STAT 引脚用于表示一次数据通信正常与否，不再作为模块是否在线的依据。
2. 入网成功后若需查询模块是否在线，需执行指令“AT+JOIN?”，具体参考“[6.3.30 读取注册入网信息 JOIN](#)”。
3. 若终端需要同步本地 RTC 时间，可执行指令“AT+TIMESYNC”，具体参考“[6.3.39 时间同步 TIMESYNC](#)”。

##### 4-2：模块入网阶段功耗消耗较大，非 class C 应用场景，建议用户增加入网超时判断。

## 4.3 用户数据发送

如图 4-1 所示,在透传模式下,用户发送串口数据给模块前,需要判断 BUSY 引脚电平状态。仅当 BUSY 引脚为高电平（空闲状态）后,用户才能发送串口数据给模块。

模块收到用户数据后, BUSY 引脚将拉低, 指示模块将进行数据通信, STAT 引脚将拉高, 指示通信异常状态被清除; 模块对用户数据不做任何数据解析, 直接通过 LoRaWAN 空口发送。在数据通信期间, 即 **busy 为低电平时不允许用户休眠模块**, 否则会导致本次数据传输失败。

数据通信结束后, BUSY 引脚重新拉高, 指示用户本次数据通信已完成。若数据包为确认帧, STAT 引脚拉高指示本次通信成功, 相反, STAT 引脚拉低指示本次通信失败; 若数据包为非确认帧, STAT 引脚为拉高指示本次数据包发送成功（发送成功不代表通信成功）, STAT 引脚拉低指示本次数据包发送失败。

BUSY 和 STAT 引脚状态指示可参考“[表 4-1 透传模式的引脚功能](#)”。

如果本次通信成功并且模块有接收到服务器的下行数据, 则通过串口立即返回给用户。



图4-1 用户数据发送流程

### 4.3.1 流控机制

为了保证数据传输可靠性与完整性, 模块引入了一种简单的流控机制, 解决因空口数据包最大数据长度的动态变化而带来的数据丢失等问题。

用户自行决定一帧数据的长度, 当模块串口超过 10ms 时间未接收到新的串口数据或者达到物理分包最大负载值（如表 4-2 所示）时, 判定一帧串口数据传输完成, 模块立即拉低 BUSY 引脚（忙）, 同时关闭串口接收, 模块启动 LoRaWAN 空口发送服务。空口发送完成后（成功或失败）, BUSY 引脚重新拉高。如果本次空口服务中, 服务器有下行数据, 模块在 BUSY 拉高后, 通过串口发送给用户。

### 4.3.2 物理分包

模块的物理分包参考《LoRaWAN Regional Parameter V1.0.3》相关定义,用户可以通过”AT+DATARATE?”或者”AT+STATUS?”指令查询模块当前的速率,以获得当前模块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 ( $N = \text{用户数据包字节数} + \text{MAC 命令字节数}$ ),如表 4-3 所示:

表4-3 不同速率对应的最大负载值N

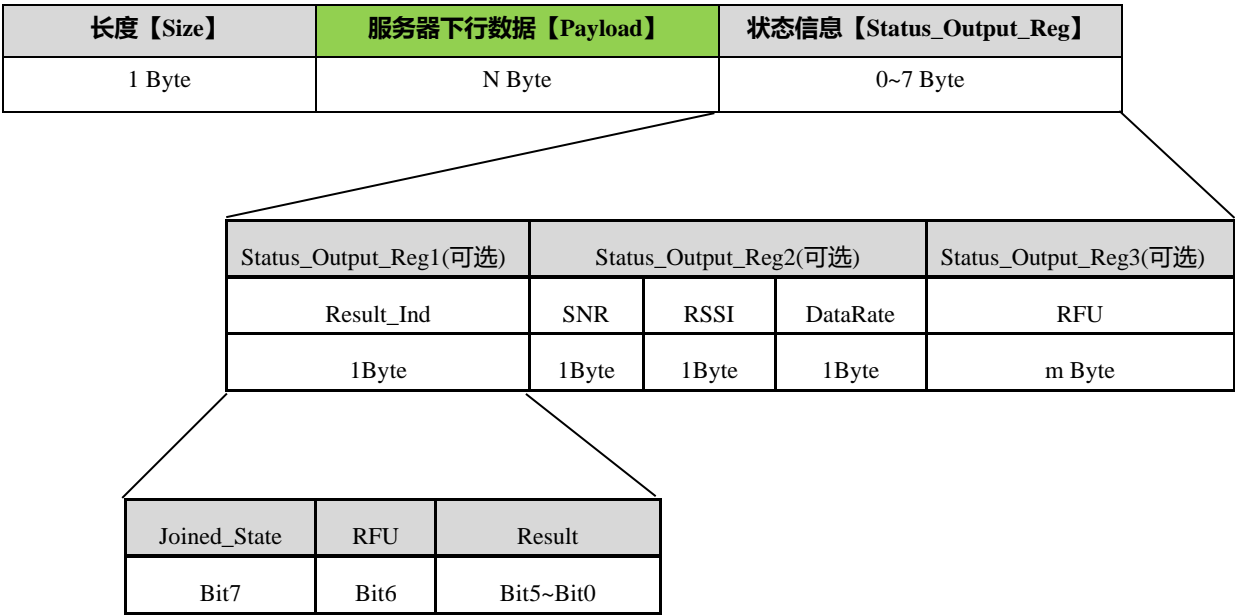
Data Rate	SF\BW	N (MAX)	Time On Air (ms)
DR5	SF7\BW125	222	368.9
DR4	SF8\BW125	222	655.9
DR3	SF9\BW125	115	676.9
DR2	SF10\BW125	51	698.4
DR1	SF11\BW125	51	1314.8
DR0	SF12\BW125	51	2465.8

## 4.4 服务器响应

根据 LoRaWAN 网络 Class A 的运行特点,终端每发送一次上行数据后,服务器可根据实际需要,发送一个下行响应数据包。如果模块收到用户服务器的下行数据,会通过串口输出给用户。

缺省情况下,模块直接转发接收到的服务器下行数据【Payload】给用户 MCU;可选的,用户可通过”AT+STATUS?”指令开启输出对应状态信息的功能。其中下行数据前 1 字节为下行数据与状态信息的大小【Size】,下行数据后的 n ( $n \geq 1$ ) 字节为状态信息【Status\_Output\_Reg】。具体的数据帧格式及内容释义如表 4-4、表 4-5 所示。

表4-4 串口响应数据帧格式 (模块->Host MCU)



注:  用户数据  
 可选项

表4-5 数据帧内容释义

字段	说明
Size	用户数据[Payload]+状态信息[Status_Output_Reg]的大小，可选字段， 缺省无该字段
Payload	服务器给用户的下行数据， N <= 222
Status_Output_Reg	Status_Output_Reg=<Status_Output_Reg1>+<Status_Output_Reg2>+<Status_Output_Reg3> 每次通信完成后， 模块更新Status_Output_Reg信息 用户可通过 <a href="#">AT+STATUS=X1.X2</a> 指令， 设置状态字节的输出等级。 可选字段， 缺省无该字段。
	Result_Ind[Bit7~Bit0]： 表示本次通信状态信息。 BIT7： 模块注册状态 0 - 表示模块未注册 1 - 表示模块已经注册（在线） BIT6： 参数保留 BIT5~BIT0： 本次通信状态信息 具体参数含义， 参考 “ <a href="#">表9-2 状态定义</a> ”
	SNR:表示本次通信的SNR， 有符号数 RSSI:表示本次通信的RSSI， 有符号数 DataRate: 表示下次通信上行的速率， 低四位有效， 高四位保留
	RFU:保留

## 4.5 空口通信流

模块的空口通信流 (LoRaWAN 网络接入与数据通信服务), 如图 4-2 所示。

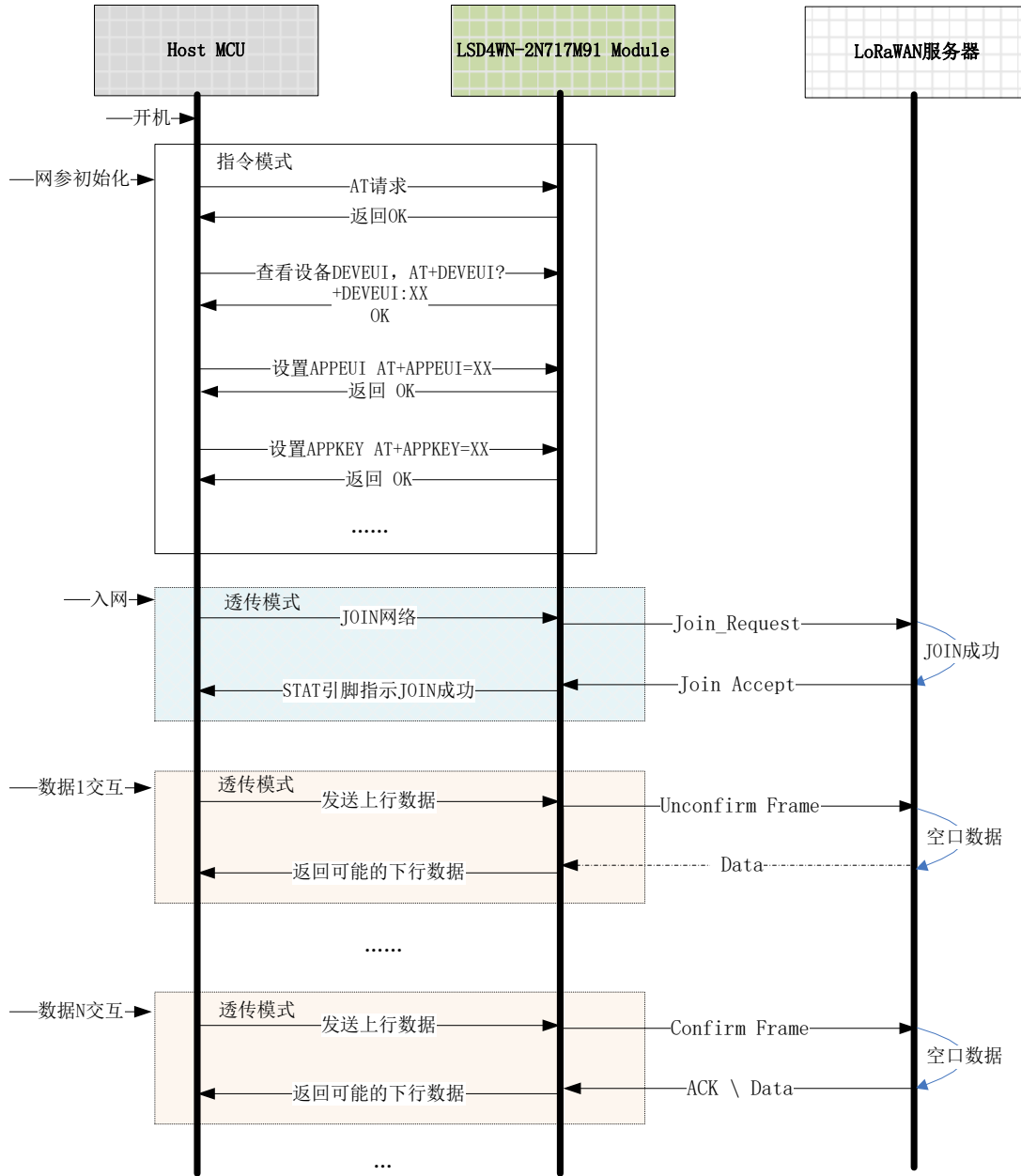


图4-2 模块的空口通信流

若网络环境等已经相对固定, 则模块只需要在首次入网的时候, 设置相关网络参数即可。随后用户 MCU 与模块直接进行用户数据的交互。

在目前市场中的 LoRaWAN 网络中, 模块成功入网后, 除了异常情况外 (比如基站异常, 终端\模块异常), 逻辑上可认为, 模块始终保持与服务器的连接状态; 可选的一种处理方式是用户层维护一个心跳包, 来周期性监测 LoRaWAN 网络通信是否发送异常 (比如连续多次通信失败)。

## 5 引脚时序

本章节描述模块与 MCU 串口通信过程中所遵循的控制时序，包含了开机、唤醒、指令及串口数据发送等。用户操作模块时，在对应的阶段必须遵循本章给出的时序以减少因时序问题而导致的意外情况。

### 5.1 上电（复位）时序

模块每次上电（复位）需要一段初始化时间，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 5-1 所示。

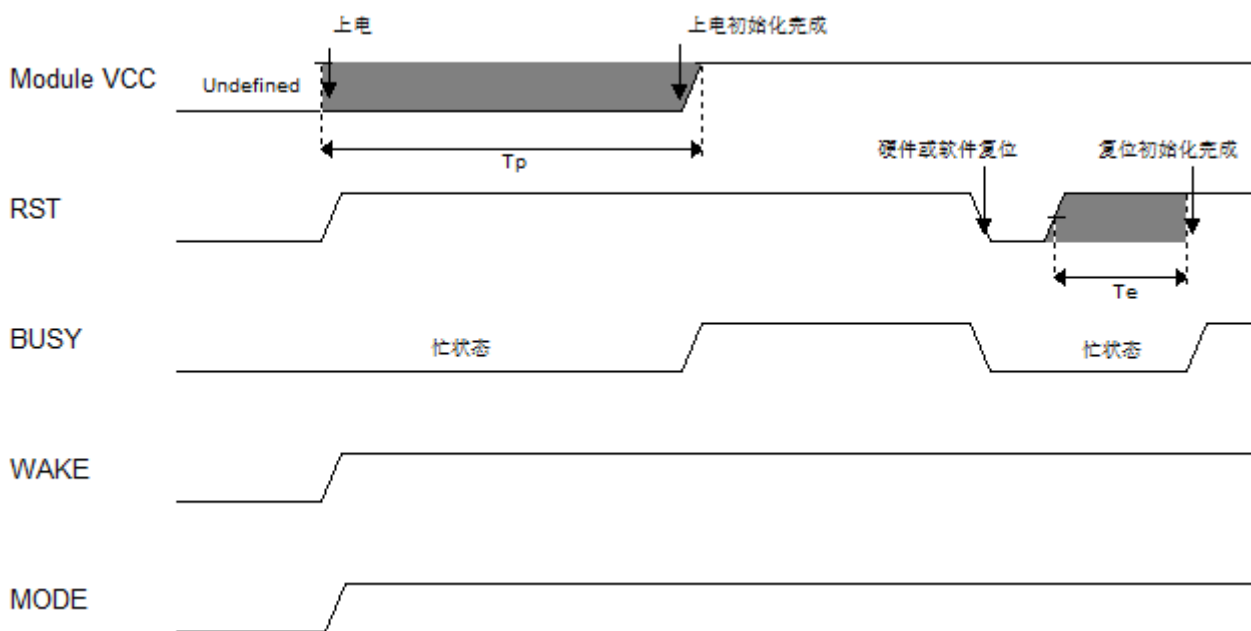


图5-1 开机时序

模块上电后，用户需要等待一段时间  $T_p$ ，典型值为 500ms；在此时间段内，模块会拉低 BUSY 引脚，内部进行上电初始化操作，不响应用户任何操作（比如发送串口数据或者 I/O 操作等）。

上电初始化完成后，若用户手动执行复位（硬/软复位）操作，同样需要初始化一段时间  $T_e$ ，典型值为 150ms，该期间模块会拉低 BUSY 引脚，不响应用户任何操作（比如发送串口数据或者 I/O 操作等）。

若是首次上电，用户需要给模块 WAKE 和 MODE 引脚提供一个固定的电平，建议配置为高电平。

当模块初始化完成后，BUSY 引脚会拉高，指示用户可操作模块。



## 5.2 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  $T_w$  后才能完全被唤醒, 一般  $T_w = 10\text{ms}(\text{max})$ 。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 5-2 所示。

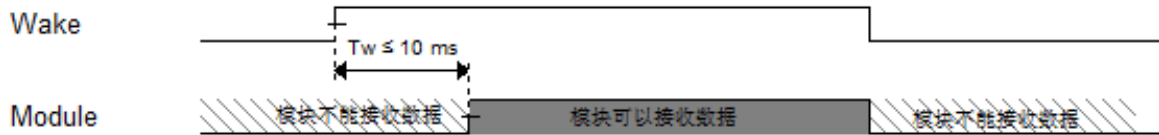


图5-2 模块唤醒时序

## 5.3 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, 模块有两种分帧机制。如下描述的是串口超时分帧机制 (另一种见“[4.3.2 物理分包](#)”介绍的最大负载值限制), 模块在串口分帧间隔  $T_g$  内,  $T_g \geq 10\text{ms} + X$  ( $X$  的单位为  $1/2048\text{s}$ ), 若没有接收到任何新的串口数据, 模块则认为该帧已经结束并开始处理此帧数据。帧分割时序如图 5-3 所示。

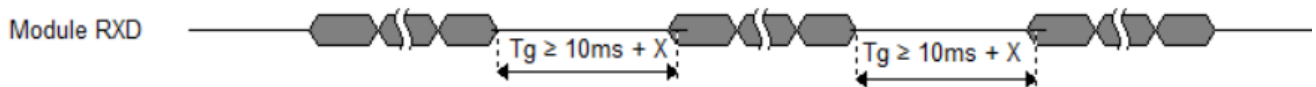


图5-3 帧超时分割时序

## 5.4 指令操作时序

在指令模式下, 从用户发出指令到收到模块响应需要一定的响应时间  $T_r$ , 若用户发送指令的间隔小于  $T_r$ , 则会出现指令无效的情况。  $T_r$  值与指令内容有关, 不是定值, 详细可参考第 6 章“[6.1 指令处理时间](#)”。指令操作的时序如图 5-4 所示。

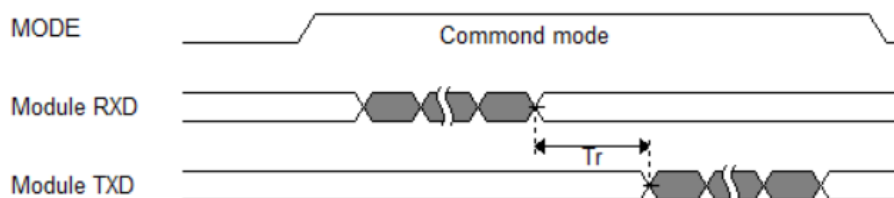


图5-4 指令操作时序

指令操作是用户操作模块很重要的一个环节, 为了更有效判断所发送的指令是否生效, 建议用户通过获取模块返回的响应信息(“OK”、“ERROR”等)进行判断, 同时进行超时管理。一般来讲, 指令响应时间不会大于  $2\text{s}$ 。

## 5.5 数据发送时序

用户发送串口数据，需要判断 BUSY 状态，只有在 BUSY 为高的情况，写入数据才是安全的。

数据发送的时序如图 5-5 所示。



图5-5 串口发送时序

## 5.6 数据接收时序

当模块发送完上行数据后，若有来自服务器的下行数据，则模块会在 BUSY 引脚拉高的发送迟滞时间  $T_d$  后，将数据发送给用户。用户可以设定  $T_d$  值，缺省为 10ms。数据接收时序如图 5-6 所示。

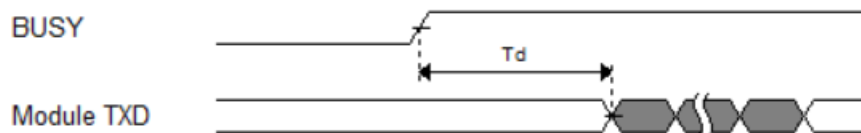


图5-6 数据接收时序

在 CLASS A 低功耗应用中，若串口无法提供唤醒 MCU 功能，则用户可利用 BUSY 引脚来唤醒。当模块接收完服务器数据后，BUSY 引脚拉高，用户可利用  $T_d$  这段时间将 MCU 完全唤醒，接下来便可接收模块发送的数据。

## 5.7 数据通信时序

图 5-7 详细描述了当模块成功加入网络后，用户操作模块进行一次完整用户数据交互的流程。

- 1) 用户拉高 WAKE 引脚，以唤醒模块；
- 2) MODE 引脚保持为低电平（透传模式）；
- 3) 用户等待模块准备好，即等待 BUSY 为高电平；
- 4) 用户发送串口数据，等待串口分帧超时（用户一次传输的数据包小于 LoRaWAN 最大数据包限制），即等待用户数据发送完成，发送结束后 BUSY 引脚拉低；
- 5) 模块发送空口数据（上行），BUSY 保持为低，如果是 Confirm 帧发送失败，模块自动重发；
- 6) 若当前是 Confirm 帧或者服务器端有下行数据，则模块会接收到（空口）数据；
- 7) 模块完成数据（上行与下行）处理，BUSY 引脚被拉高。如果本次通信数据出现异常，则 STAT 引脚输出为低，指示用户本次通信有异常。用户可进入指令模式，通过 AT+STATUS? 等查询相关寄存器，获取更详细的状态信息。如果无异常，STAT 引脚保持为高电平；
- 8) 如果服务器有下行数据，则模块首先完成数据解析，然后在  $T_d$ （缺省为 10ms）时间后，通过串口发送给用户；
- 9) 一次用户数据交互完成。用户可根据实际需要，让模块重新进入休眠。

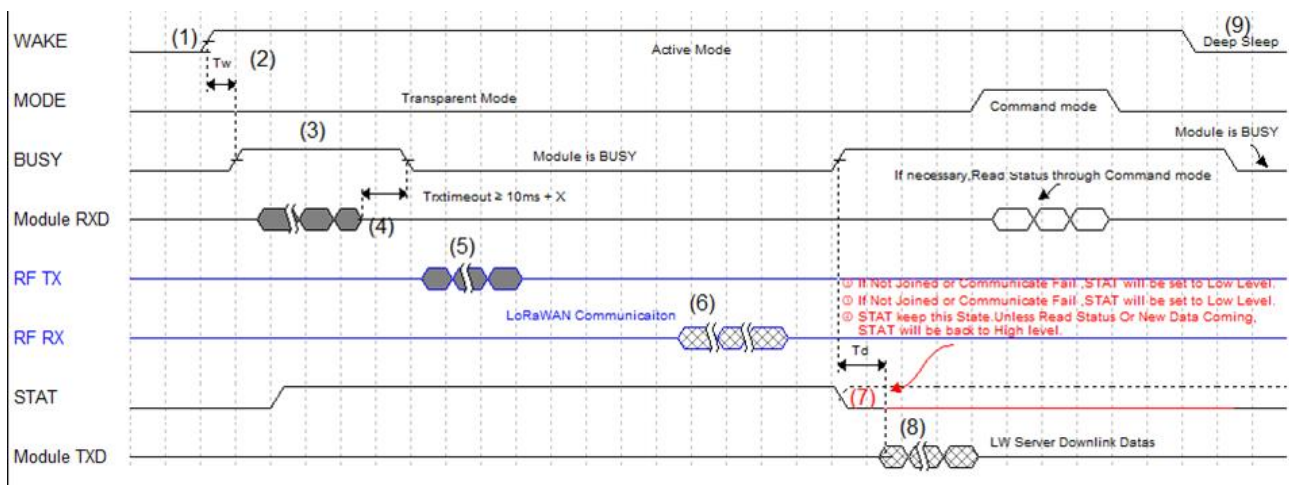


图5-7 模块透传模式的工作时序

## 6 AT指令集

在本章节中，AT 指令集细分为系统指令与 MAC 指令<sup>注6-1</sup>。

系统指令主要配置串口、RTC 等与 LoRaWAN 网络无关的应用参数，MAC 指令主要配置 LoRaWAN 相关的网络参数等。

### 6.1 指令处理时间

用户执行指令时，需等待模块返回响应结果（如“OK”，“ERROR”等）后，才能进行下一条指令操作<sup>注6-2</sup>；除此之外，用户可设置指令超时时间，以此判断模块是否出现异常。用户可根据单指令的处理时间作为超时参考。

一般来讲，一条 AT 指令的处理时间（ $T_h$ ）包含串口传输时间（ $T_s$ ）以及模块响应时间（ $T_r$ ），以指令“AT+RTC?”为例，如图 6-1 所示。

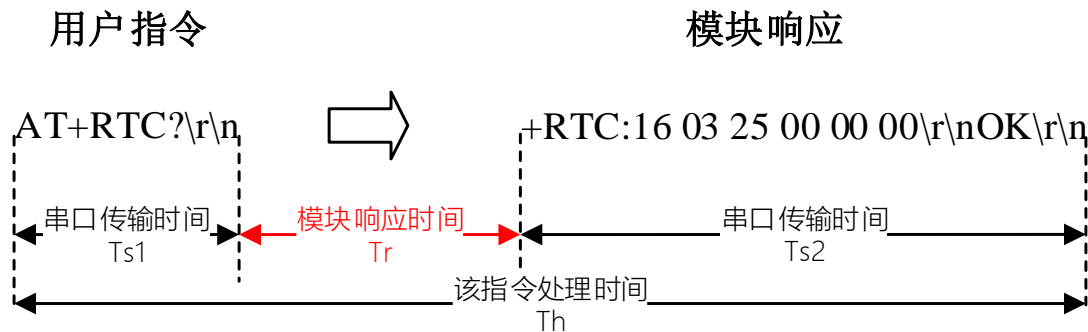


图6-1 指令处理时间

串口传输时间  $T_s$ ：包含用户指令的传输时间  $T_{s1}$  和指令响应的传输时间  $T_{s2}$ 。在指令模式下，模块串口通信参数固定为 9600N81。用户可根据指令本身及其响应结果两者的实际长度计算得到串口传输时间  $T_s$  ( $T_{s1} + T_{s2}$ )。

模块响应时间  $T_r$ ：以模块接收完用户指令最后一个字节为起点，模块发送响应第一个字节为终点，起点到终点这段时间即为模块响应时间  $T_r$ 。目前模块响应时间依据指令的不同主要分为以下两种类别。

1. 响应时间  $T_r \leq 10\text{ms}$ ：适用于大部分指令，如“AT+RTC=X”、“AT+OTAA=X”等；这类响应时间在具体的指令描述中将不再给出。
2. 响应时间  $T_r > 10\text{ms}$ ：适用于某些特殊指令，如“AT+DEVEUI=X”、“AT+SAVE”等；这类响应时间在具体的指令描述中将给出。

同样以指令“AT+RTC?”为例，该指令的串口传输时间  $T_s = T_{s1} + T_{s2} = 9\text{ms} + 28\text{ms} = 37\text{ms}$ ，故指令处理时间  $T_h = T_s + T_r = 37\text{ms} + 10\text{ms} = 47\text{ms}$ 。用户可利用该值做指令超时管理。

注 6-1:

1. 如下说明中，“响应 1、2..”中可能出现的具体数值，仅供示例参考；
2. 指令  $AT+X1,<X2>=Y$  中， $X1$  代表必选参数， $<X2>$  代表可选参数。

注 6-2:

个别指令在模块返回响应结果后，同时还需判断 BUSY 引脚状态，请参考具体的 AT 指令，如“AT+FACTORY”等。

## 6.2 系统指令

### 6.2.1 空指令 NOP

空指令 NOP		
描述	该指令可以用于测试模块与用户MCU的串口通信是否正常	
指令	AT	NOP指令
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 OK	串口通信正常

### 6.2.2 恢复出厂设置 FACTORY

恢复出厂设置		
描述	该指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 ◆ 该指令生效需满足模块返回“OK”响应以及BUSY引脚拉高 ◆ 该指令生效时间最大值为1500ms	
指令	AT+FACTORY	执行指令
响应 1	OK	设置成功
响应 2	ERROR	未知错误
示例	AT+FACTORY OK	恢复出厂设置成功，模块内部自动重启，BUSY引脚拉高后生效

### 6.2.3 设置 IO 口状态 GPIO

设置 io 口状态		
描述	该指令为一扩展功能，针对P0、P1、P2、P3有效 ◆ 该指令当前不可读	
指令	AT+GPIO=X1,X2	X1: 模块扩展引脚 0 - 表示 P0 1 - 表示 P1 2 - 表示 P2 3 - 表示 P3 255 - 表示 P0 P1 P2 P3 X2: 扩展引脚输出电平状态 1 - 输出高电平 0 - 输出低电平
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	设置P0为输出高电平

### 6.2.4 设置外设 PERIPH

设置模块外设功能		
描述	该指令用于设置模块外设功能选择	
指令	AT+PERIPH=X1,X2	X1: 外设功能选择 0 - RTC RTC默认是打开状态，关闭RTC将进一步优化低功耗，但是RTC时间将不会更新 1 - P1\ P2引脚脉冲输出 P1\ P2引脚常态为高电平，每次发送\接收后，输出一个25ms的低电平，可用于发送\接收通信指示(eg:LED) P1默认为输出高电平，每次发送完成后，P1输出25ms的低电平脉冲 P2默认为输出高电平，每次接收完成后，P2输出25ms的低电平脉冲 X2: 对应外设功能的开启或关闭，默认值为0 0 - 关闭 1 - 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 1	AT+PERIPH=0,1 OK	使能RTC实时时钟
示例 2	AT+PERIPH=1,1 OK	使能P1\P2引脚脉冲输出功能 每次发送完成后, P1输出25ms的低电平脉冲。每次接收完成后, P2输出25ms的低电平脉冲

## 6.2.5 读取版本 VER

读取版本																		
描述	该指令返回产品版本号																	
指令	AT+VER?	读取版本号信息																
		返回 LoRaWAN协议版本+硬件版本+固件版本+时间版本																
		硬件版本：CN470P+1Byte，区分硬件，比如CN470P.01																
		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 OK	读取成功，返回结果 Y1 为当前固件版本																
示例	AT+VER?  +VER:LoRaWAN1.0.3  _CN470P.21-  02.22.01_Nov 8 2019  15:30:22  OK	当前版本的LoRaWAN协议版本为LoRaWAN1.0.3，硬件版本为CN470P.21，固件版本为02.22.01，时间版本为： Nov 8 2019 15:30:22																

## 6.2.6 设置日志信息输出 DEBUG

设置日志信息输出		
描述	该指令为配置日志信息输出 ◆ 关于日志信息的详细内容请参考“ <a href="#">第9章 日志信息</a> ” ◆ DEBUG默认是关闭的，在调试阶段，可打开DEBUG便于分析问题	
指令	AT+DEBUG=X1	X1: 打开/关闭日志信息输出，出厂值为0 0 – 关闭日志输出 1 – 打开日志输出
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+DEBUG=1 OK	开启日志输出功能

## 6.2.7 参数保存 SAVE

保存配置参数		
描述	该指令将配置参数保存到内部EEPROM ◆ 该指令最大响应时间为1000ms	
指令	AT+SAVE	执行指令
响应 1	OK	执行完成
响应 2	ERROR	未知错误
示例	AT+SAVE OK	保存成功

## 6.2.8 系统复位 RESET

系统复位		
描述	该指令复位模块 ◆ 系统重启后，模块自动装载EEPROM中的配置参数，并运行 ◆ 该指令最大响应时间为150ms，即模块初始化所需时间	
指令	AT+RESET	执行指令
响应 1	OK	设置成功
响应 2	ERROR	未知错误
示例	AT+RESET OK	设置成功，系统随后立即启动复位



### 6.2.9 设置串口波特率 BAUD

设置串口波特率		
描述	<p>该指令设置透传模式下的串口波特率，执行该指令后，用户从指令模式切换到透传模式后，将立即生效</p> <ul style="list-style-type: none"> <li>◆ 指令模式下的串口配置固定为9600N81，无法通过此指令进行配置</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+BAUD=X1	<p>X1：透传模式下支持的波特率， 出厂值：9600，取值范围：</p> <p>1200\ 2400\ 4800\ 9600\ 19200\ 38400\ 57600\ 115200</p>
响应 1	OK	设置成功
响应 2	BAD PARM	指令参数错误
响应 3	ERROR	未知错误
示例	<p>AT+BAUD=9600</p> <p>OK</p>	设置模块透传模式下的串口波特率为9600bps

### 6.2.10 读取串口波特率 BAUD

读取串口波特率		
描述	该指令读取透传模式下的串口波特率	
指令	AT+BAUD?	返回透传模式使用的波特率值：1200\ 2400\ 4800\ 9600\ 19200\ 38400\ 115200
响应 1	<p>+BAUD: X1</p> <p>OK</p>	读取成功，返回结果
响应 2	ERROR	未知错误
示例	<p>AT+BAUD?</p> <p>+BAUD: 9600</p> <p>OK</p>	模块透传模式的串口波特率为9600

## 6.2.11 设置串口分帧时间 TIMEOUT

设置串口分帧时间		
描述	<p>该指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</p> <ul style="list-style-type: none"> <li>◆ <a href="#">串口分帧间隔Tg</a>，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</li> <li>◆ <a href="#">串口发送迟滞时间Td</a>，默认为0，允许用根据应用设置迟滞时间</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+TIMEOUT=X1 或者 AT+TIMEOUT=X1,X2	X1: 串口接收分帧间隔Tg，单位：1/2048s，出厂值为0(Tg最小值固定为10ms)，取值范围0~65535 X2: 延时发送用户数据给串口的时间Td，单位为ms，默认值为0，有效范围：0~255
响应 1	OK	设置成功
响应 2	BAD PARM	指令参数错误
响应 3	ERROR	未知错误
示例 1	AT+TIMEOUT=20 OK	串口分帧超时时间 $T_g = (20 / 2048s + 10ms) \approx 20ms$
示例 2	AT+TIMEOUT=0,10 OK	串口分帧超时时间 $T_g = 10ms$ (缺省最小值) 串口发送迟滞时间 $T_d = 10ms$

## 6.2.12 读取串口分帧时间 TIMEOUT

读取串口分帧时间		
描述	该指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	
指令	AT+TIMEOUT?	返回串口接收分帧间隔Tg与串口延时给用户发送数据的迟滞时间Td
响应 1	+TIMEOUT: X1,X2 OK	读取成功，返回结果
响应 2	ERROR	未知错误
示例	AT+TIMEOUT? +TIMEOUT: 0,0 OK	串口分帧超时时间 $T_g = 10ms$ (缺省最小值) 串口发送迟滞时间 $T_d = 0$

### 6.2.13 设置实时时间 RTC

设置 RTC 时间																		
描述	该指令设置模块的RTC时间 ◆ 模块复位后，需要再次执行该指令，不支持掉电保存																	
指令	AT+RTC=X1,X2...X6	X1,X2...X6 为RTC时间，时间格式如下 <table><tr><td>年</td><td>月</td><td>日</td><td>时</td><td>分</td><td>秒</td></tr><tr><td>2000-2100</td><td>1-12</td><td>1-31</td><td>0-59</td><td>0-59</td><td>0-59</td></tr></table>					年	月	日	时	分	秒	2000-2100	1-12	1-31	0-59	0-59	0-59
年	月	日	时	分	秒													
2000-2100	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 OK	设置年、月、日、时、分、秒																

### 6.2.14 读取实时时间 RTC

读取 RTC 时间							
描述	该指令读取模块的RTC时间						
指令	AT+RTC?	返回RTC时间X1,X2...X6					
		年	月	日	时	分	秒
		2000-2100	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						

### 6.2.15 设置系统参数 SYSTEM

设置系统参数		
描述	该指令配置模块杂项(MISC)功能，比如回显等功能 ◆ 该参数修改后，执行AT+SAVE指令可掉电保存	
指令	AT+SYSTEM=X1	X1：回显功能，出厂值为0 0 – 关闭回显 1 – 开启回显
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+SYSTEM=1 OK	开启模块回显功能

### 6.2.16 读取系统参数 SYSTEM

读取系统参数		
描述	该指令查看当前系统配置	
指令	AT+SYSTEM?	读取相关的系统配置
响应 1	+SYSTEM: X1,X2...X6 OK	读取成功，返回结果，详细参数说明同 <a href="#">“6.2.15 设置系统参数SYSTEM”</a>
响应 2	ERROR	未知错误
示例	AT+SYSTEM? +SYSTEM: 1, 0, 1, 1, 0, 0 OK	开启回显功能，使能Class C自动应答下行确认帧 X4、X5、X6参数保留

## 6.3 MAC指令

### 6.3.1 设置上行传输类型 CONFIRM

设置上行链路 (upLink ) 传输类型		
描述	该指令设置上行链路的传输类型 ◆ 该参数修改后，执行AT+SAVE指令可掉电保存	
指令	AT+CONFIRM=X1,<X2>	X1: 上行链路的传输类型，默认值为0，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 X2: 重发次数+1，仅当X1=1时有效。可选参数，取值范围：1~8，默认值为8
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=1,4 OK	设置上行帧为Confirm (确认)帧，重发次数为3

### 6.3.2 读取上行传输类型 CONFIRM

读取上行链路 (upLink ) 传输类型		
描述	该指令读取上行链路的传输类型	
指令	AT+CONFIRM?	读取上行帧类型 (Confirm or Unconfirm)
响应 1	+CONFIRM:X1,X2 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? +CONFIRM:1,4 OK	模块当前帧传输类型为Confirm帧，重发次数为3

### 6.3.3 设置激活模式 OTAA

设置设备的激活模式		
描述	该指令用于设置加入网络的激活模式 ◆ 该参数修改后，执行AT+SAVE指令可掉电保存	
指令	AT+OTAA=X1,<X2>	X1: 激活类型，默认值为1，取值范围： 0 – ABP激活 1 – OTAA激活 X2: 关闭/开启热启动，该参数为可选参数，默认值为1 0 – 关闭热启动，模块上电\复位后，已入网设备注册上下文被清除，需重新JOIN 1 – 开启热启动，模块上电\复位后，已入网设备注册上下文被保存，无需重新JOIN
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 1	AT+OTAA=1 OK	设置为OTAA激活
示例 2	AT+OTAA=1,0 OK	设置为OTAA激活，关闭热启动
示例 3	AT+OTAA=1,1 OK	设置为OTAA激活，开启热启动

### 6.3.4 读取激活模式 OTAA

读取设备的激活模式		
描述	该指令读取设备当前的激活模式	
指令	AT+OTAA?	读取设备的激活模式。
响应 1	+OTAA:X1,X2,X3,X4 OK	读取成功，返回结果。详细参数说明见“ <a href="#">6.3.3 设置激活模式OTAA</a> ” X3、X4参数保留
响应 2	BAD PARM	参数错误
响应 3	ERROR	未知错误
示例 1	AT+OTAA? +OTAA:1,0,1,1 OK	模块当前为OTAA激活方式，关闭热启动
示例 2	AT+OTAA? +OTAA:1,1,1,1 OK	模块当前为OTAA激活方式，开启热启动

### 6.3.5 设置 DEVEUI

设置 DevEui		
描述	<p>该指令设置模块的DevEui</p> <ul style="list-style-type: none"> <li>◆ DevEui以16进制表示，共8个字节，它为全网唯一的设备标识符，类似MAC地址的意义</li> <li>◆ 该指令只用于OTAA激活模式</li> <li>◆ 该指令最大响应时间为100ms</li> <li>◆ 该指令配置后自动保存至EEPROM中，模块复位后无需再配置</li> </ul>	
指令	AT+DEVEUI= X1n,X2m,X3	<p>X1n: DevEUI, 以16进制数表示，共8个字节: X11 X12...X18</p> <p>X2m: 校验值，共8个字节，固定为 D3 91 01 02 20 10 28 16</p> <p>X3: 参数保留，固定为1</p>
响应 1	OK	设置成功.如示例说明
响应 2	ERROR	未知错误
示例	AT+DEVEUI=00 95 69 00 00 00 1A 0D, D3 91 01 02 20 10 28 16, 1 OK	设置模块DevEui为0x00, 0x95, 0x69, 0x00, 0x00, 0x00, 0x1A, 0x0D

### 6.3.6 读取 DEVEUI

读取DevEui		
描述	<p>该指令读取模块的DevEui</p> <ul style="list-style-type: none"> <li>◆ DevEui为全网唯一的设备标识符.类似MAC地址的意义</li> </ul>	
指令	AT+DEVEUI?	读取DevEui，格式详见 “ <a href="#">6.3.5 设置DEVEUI</a> ”
响应1	+DEVEUI: X11 X12...X18 OK	读取成功.返回结果
响应2	ERROR	未知错误
示例	AT+DEVEUI? +DEVEUI: 00 95 69 00 00 00 1A 0D OK	模块DevEui为 0x00, 0x95, 0x69, 0x00, 0x00, 0x00, 0x1A, 0x0D

### 6.3.7 设置 APPEUI

设置 AppEui		
描述	该指令设置应用服务标识符 ◆ 该指令只用于OTAA激活模式 ◆ 该参数修改后，执行AT+SAVE指令可掉电保存	
指令	AT+APPEUI=X1n	X1n: AppEui, 以16进制数表示, 共8个字节: X1 <sub>1</sub> X1 <sub>2</sub> ...X1 <sub>8</sub>
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

### 6.3.8 读取 APPEUI

读取 AppEui		
描述	该指令读取应用标识符	
指令	AT+APPEUI?	返回值X1 <sub>1</sub> X1 <sub>2</sub> ...X1 <sub>8</sub> , 描述同“ <a href="#">6.3.7 设置APPEUI</a> ”, 数据格式为16进制, 共8个字节
响应1	+APPEUI: X1 <sub>1</sub> X1 <sub>2</sub> ...X1 <sub>8</sub> OK	读取成功, 则返回AppEui
响应2	ERROR	未知错误
示例	AT+APPEUI? +APPEUI: FF FF FE 00 00 00 00 01 OK	模块AppEui为0xFF, 0x FF,0xFE, 0x00, 0x00, 0x00, 0x00, 0x01



### 6.3.9 设置 APPKEY

设置 AppKey		
描述	<p>该指令设置AppKey，数据格式为16进制，共16个字节</p> <ul style="list-style-type: none"> <li>◆ 该指令只用于OTAA激活模式</li> <li>◆ 该指令最大响应时间为100ms</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+APPKEY= X1n,<X2>	<p>X1n: AppKey，以16进制数表示，共16个字节：X1<sub>1</sub> X1<sub>2</sub>...X1<sub>16</sub></p> <p>X2: 设置密钥是否可读，可选参数，出厂值为1</p> <p>0 - 密钥可读模式，用户可以通过AT指令读取相关密钥</p> <p>1 - 密钥不可读模式，用户无法读取相关密钥</p> <p>出于安全性考虑，建议用户使用“密钥不可见模式”</p>
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1	AT+APPKEY= 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C	设置APPKEY为 0x2B 0x7E 0x15 0x16 0x28 0xAE 0xD2 0xA6 0xAB 0xF7 0x15 0x88 0x09 0xCF 0x4F 0x3C，密钥不可读模式
示例2	AT+APPKEY= 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C,0	设置APPKEY为 0x2B 0x7E 0x15 0x16 0x28 0xAE 0xD2 0xA6 0xAB 0xF7 0x15 0x88 0x09 0xCF 0x4F 0x3C，密钥可读模式

### 6.3.10 读取 APPKEY

读取 AppKey		
描述	<p>该指令读取AppKey</p> <ul style="list-style-type: none"> <li>◆ 该指令只对“密钥可读模式”有效，具体可参考“<a href="#">6.3.9 设置APPKEY</a>”</li> </ul>	
指令	AT+APPKEY?	返回应用密钥值：X1 <sub>1</sub> X1 <sub>2</sub> ...X1 <sub>16</sub> ，数据格式为16进制，共16个字节。
响应1	+APPKEY: X1 <sub>1</sub> X1 <sub>2</sub> ...X1 <sub>16</sub> OK	读取成功，返回结果
响应2	ERROR	未知错误。或者模块处于“密钥不可见模式”
示例	AT+APPKEY? +APPKEY: 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C OK	<p>如果配置了模块为密钥可读模式，则返回模块当前应用密钥为：</p> <p>0x2B 0x7E 0x15 0x16 0x28 0xAE 0xD2 0xA6 0xAB 0xF7 0x15 0x88 0x09 0xCF 0x4F 0x3C</p>

### 6.3.11 设置 DEVADDR

设置 DevAddr		
描述	<p>该指令设置DevAddr，数据格式为16进制，共4个字节</p> <ul style="list-style-type: none"> <li>◆ 该指令仅适用于ABP激活方式</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+DEVADDR=X1n	<p>X1n: DevAddr，以16进制数表示，共4个字节: X1<sub>1</sub> X1<sub>2</sub> X1<sub>3</sub> X1<sub>4</sub></p> <p>DevAddr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，共4个字节</p> <p>DevAddr根据入网激活方式，区分两种情况：</p> <ol style="list-style-type: none"> <li>1. 通过本地激活的节点(ABP)，必须设置DevAddr</li> <li>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）</li> </ol>
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

### 6.3.12 读取 DEVADDR

读取 DevAddr		
描述	<p>该指令读取DevAddr</p> <ul style="list-style-type: none"> <li>◆ 该指令对OTAA与ABP都有效</li> </ul>	
指令	AT+DEVADDR?	读取设备地址，返回具体DevAddr值，数据格式为16进制，共4个字节
响应1	+DEVADDR: X1 <sub>1</sub> X1 <sub>2</sub> X1 <sub>3</sub> X1 <sub>4</sub> OK	<p>读取成功，返回结果</p> <p>X1<sub>1</sub> X1<sub>2</sub> X1<sub>3</sub> X1<sub>4</sub> 为DevAddr值</p>
响应2	ERROR	未知错误
示例	AT+DEVADDR? +DEVADDR: FF 00 00 01 OK	模块DevAddr为 0xFF000001

### 6.3.13 设置 APPSKEY

设置AppSKey		
描述	<p>该指令设置AppSKey</p> <ul style="list-style-type: none"> <li>◆ AppSKey只用于ABP激活方式。数据格式为16进制，共16个字节。</li> <li>◆ 该指令最大响应时间为100ms</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+APPSKEY= X1n	<p>设置应用会话密钥Application Session Key</p> <p>X1n: AppSKey, 以16进制数表示, 共16个字节: X11 X12...X116</p>
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 D2 A6 AB F7 15 88 09 CF 4F 3C OK	<p>设置APPSKEY为0x2B 0x7E 0x15 0x16 0x28 0xAE</p> <p>0xD2 0xA6 0xAB 0xF7 0x15 0x88 0x09 0xCF 0x4F 0x3C</p>

### 6.3.14 设置 NWKSKEY

设置 NwkSKey		
描述	<p>该指令设置NwkSKey</p> <ul style="list-style-type: none"> <li>◆ NwkSKey只用于ABP激活方式；数据格式为16进制，共16个字节</li> <li>◆ 该指令最大响应时间为100ms</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+NWKSKEY=X1n	<p>设置网络会话密钥Network Session Key</p> <p>X1n: NwkSKey, 以16进制数表示, 共16个字节: X11 X12...X116</p>
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 D2 A6 AB F7 15 88 09 CF 4F 3C	<p>设置NwkSKey为0x2B 0x7E 0x15 0x16 0x28 0xAE</p> <p>0xD2 0xA6 0xAB 0xF7 0x15 0x88 0x09 0xCF 0x4F 0x3C</p>

### 6.3.15 设置 PORT

设置PORT		
描述	该指令设置端口号[PORT] ◆ 对于LoRaWAN透模块，建议采用缺省值，用户不需要设置该参数 ◆ 该参数修改后，执行AT+SAVE指令可掉电保存	
指令	AT+PORT=X1	X1为所使用的port，数据格式为16进制，出厂值为0x0A，取值范围：0x15~0xDF 注：Port: 0x00，LoRaWAN MAC指令指定端口
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

### 6.3.16 读取 PORT

读取PORT		
描述	该指令读取port端口号	
指令	AT+PORT?	返回值如“6.3.17 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:0A OK	模块当前PORT为0x0A

### 6.3.17 设置设备类型 CLASS

设置设备类型		
描述	该指令设置模块的设备类型 ◆ 该参数修改后，执行AT+SAVE指令可掉电保存	
指令	AT+CLASS=X1	X1：设备类型，出厂值为0，取值范围： 0 - Class A 2 - Class C
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

### 6.3.18 读取设备类型 CLASS

读取设备类型		
描述	该指令读取模块的设备类型	
指令	AT+CLASS?	返回模块的设备类型 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

### 6.3.19 设置 ADR

设置 ADR使能		
描述	<p>该指令设置使能/失能自适应数据速率ADR(Adaptive data rate)机制</p> <ul style="list-style-type: none"> <li>◆ ADR功能开启后，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，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+ADR=X1,<X2>,<X3>	<p>X1: ADR使能控制，默认值为1</p> <p>0 - ADR不使能</p> <p>1 - ADR使能</p> <p>X2: ADR_ACK_LIMIT，默认值为64，可选参数</p> <p>X3: ADR_ACK_DELAY，默认值为32，可选参数</p> <p>X2、X3用于非确认帧调速，具体可参考《LoRaWAN_Specification_-v1.0.3》</p>
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	开启ADR功能

### 6.3.20 读取 ADR

读取ADR使能		
描述	该指令读取ADR使能控制	
指令	AT+ADR?	读取ADR使能控制
响应1	+ADR: X1,X2,X3,X4,x5 OK	<p>读取成功，返回结果，详细说明见“<a href="#">6.3.19 设置ADR</a>”</p> <p>X4, X5参数保留</p>
响应2	ERROR	未知错误
示例	AT+ADR? +ADR: 1,64,32,32,16384 OK	模块当前开启ADR功能，ADR_ACK_LIMIT=64，ADR_ACK_DELAY=32

### 6.3.21 设置发送功率 POWER

设置发送功率		
描述	<p>该指令设置模块的发送功率</p> <ul style="list-style-type: none"> <li>◆ 如果ADR功能被开启，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，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+POWER=X1	<p>X1：发送功率大小，出厂值为0，取值范围：</p> <ul style="list-style-type: none"> <li>0 - 20dBm</li> <li>1 - 18dBm</li> <li>2 - 16dBm</li> <li>3 - 14dBm</li> <li>3 - 12dBm</li> <li>5 - 10dBm</li> <li>6 - 8dBm</li> <li>7 - 6dBm</li> <li>8 - 4dBm</li> <li>9 - 2dBm</li> </ul>
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最大功率（20dBm）

### 6.3.22 读取发送功率 POWER

读取发送功率		
描述	读取模块的当前发送功率	
指令	AT+POWER?	返回模块当前的功率值，详细说明见 <a href="#">“6.3.21 设置发送功率POWER”</a>
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? +POWER:0 OK	当前模块功率为17dBm

### 6.3.23 设置通信速率 DATARATE

设置通信速率		
描述	<p>该指令设置模块的通信速率</p> <ul style="list-style-type: none"> <li>◆ 如果ADR功能被开启，模块在运行过程中，服务器会根据ADR机制动态调整模块的上行速率</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+DATARATE=X1	<p>X1：速率值，出厂值为3，取值范围：</p> <p>0 - SF12, BW125</p> <p>1 - SF11, BW125</p> <p>2 - SF10, BW125</p> <p>3 - SF9, BW125</p> <p>4 - SF8, BW125</p> <p>5 - SF7, BW125</p>
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

### 6.3.24 读取通信速率 DATARATE

读取通信速率		
描述	读取模块通信速率	
指令	AT+DATARATE?	返回模块当前通信速率与缺省速率
响应1	+DATARATE:X1,X2,X3,X4,X5 OK	<p>读取成功，返回结果</p> <p>X1：当前速率</p> <p>X2：出厂使用的缺省速率</p> <p>X3：当前设备的速率范围：高四位为最小速率，低四位为最大速率</p> <p>X4：保留</p> <p>X5：保留</p>
响应2	ERROR	未知错误
示例	AT+DATARATE? +DATARATE:0, 3, 50, 80, 20 OK	模块当前速率为DR0(SF12, BW125)，缺省速率为DR3(SF9, BW125)，速率范围DR0~DR5



### 6.3.25 设置 BAND

设置BAND		
描述	<p>该指令设置模块的入网扫描模式</p> <ul style="list-style-type: none"> <li>◆ 该指令一般配合 AT+FREQ=X 使用</li> <li>◆ 成功设置该指令后，会清除模块注册上下文信息</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+BAND=X1	<p>X1：同频或异频，默认值为7，取值范围：</p> <p>6 – 异频</p> <p>7 – 同频</p>
响应1	OK	设置成功
响应2	BAD PARM	指令参数错误
响应3	ERROR	未知错误
示例1	AT+BAND=6 OK	设置为入网扫描为异频
示例2	AT+BAND=7 OK	设置为入网扫描为同频

### 6.3.26 读取 BAND

读取BAND		
描述	该指令读取模块当前BAND	
指令	AT+BAND?	返回当前扫描模式
响应1	+BAND:X1,X2,X3,X4 OK	<p>读取成功，返回结果。详细说明见 “<a href="#">6.3.25 设置BAND</a>”</p> <p>X2, X3, X4参数保留</p>
响应2	ERROR	未知错误
示例	AT+BAND? +BAND:7,1,0,0 OK	当前入网扫描模式为同频

### 6.3.27 设置FREQ

设置FREQ		
描述	<p>该命令设置模块频点表，用户可以设置模块使用自定义的频率表</p> <ul style="list-style-type: none"> <li>◆ 该指令一般配合 AT+BAND=X 使用</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	<p>AT+FREQ=X1,X2,X3</p> <p>或者</p> <p>AT+FREQ=X1,X2,X3,X4</p>	<p>X1: 频率设置方式</p> <p>0 - 清除之前频率配置，重启模块后，以出厂默认参数运行</p> <p>1 - 设置上行频率表，频率表按照起始频率、信道带宽和信道个数，自动生成</p> <p>2 - 设置上行频率表，单独设置逻辑信道对应的指定频率</p> <p>3 - 设置下行频率表，频率表按照起始频率、信道带宽和信道个数，自动生成。若未设置该项，则默认下行频率 = 上行频率</p> <p>X2: 信道个数，有效范围1~16。注意与基站侧协同配合</p> <p>X3: 取决于X1参数设置</p> <p>若X1=1或者3，则X3为起始频率，单位Hz</p> <p>若X1=2，则X3为实际使用频点，可能有多参数，频点个数取决于X2，单位Hz</p> <p>X4: 信道带宽，缺省为200KHz，可选参数。X4仅对X1=1或者3有效</p>
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1	<p>AT+FREQ=1, 8, 470300000</p> <p>AT+FREQ=3, 8, 500300000</p>	<p>设置模块上行频率为470300000开始的8个信道（470300000、470500000、470700000、470900000、471100000、471300000、471500000、471700000）；</p> <p>设置模块下行频率为500300000开始的8个信道（500300000、500500000、500700000、500900000、501100000、501300000、501500000、501700000）</p>
示例2	AT+FREQ=0	清除之前配置，模块重启后，以出厂默认参数运行

### 6.3.28 读取 FREQ

读取FREQ		
描述	该指令读取模块当前可用的频率	
指令	AT+FREQ?	<p>查询得到当前频率表：</p> <p>nY1 - 频率，单位Hz</p> <p>nY2 - 有效速率范围。高四位为最大速率，低四位为最小速率</p>
响应1	+FREQ: CHn : nY1, nY2 OK	读取成功，返回结果
响应2	ERROR	未知错误或者当前无可用频率
示例1	AT+FREQ? +FREQ: CH0: 475300000, 50 CH1: 475500000, 50 CH2: 475700000, 50 CH3: 475900000, 50 CH4: 476100000, 50 CH5: 476300000, 50 CH6: 476500000, 50 CH7: 476700000, 50 OK	<p>模块配置：</p> <p>设置同频, 上下行频率为475300000为起始点, 200kHz依次递增的8个频点;</p> <p>指令配置：</p> <p>AT+BAND = 7</p> <p>AT+REQ = 1, 3, 475300000</p> <p>如上配置，用户查询频率得到的参数如左侧所示</p>

<p><b>示例2</b></p>	<p>AT+FREQ?</p> <p>+FREQ:</p> <p>CH0: 470300000, 50</p> <p>CH1: 470500000, 50</p> <p>CH2: 470700000, 50</p> <p>CH3: 470900000, 50</p> <p>CH4: 471100000, 50</p> <p>CH5: 471300000, 50</p> <p>CH6: 471500000, 50</p> <p>CH7: 471700000, 50</p> <p>DNCH0: 500300000</p> <p>DNCH1: 500500000</p> <p>DNCH2: 500700000</p> <p>DNCH3: 500900000</p> <p>DNCH4: 501100000</p> <p>DNCH5: 501300000</p> <p>DNCH6: 501500000</p> <p>DNCH7: 501700000</p> <p>OK</p>	<p>模块配置:</p> <p>设置为异频, 上行频率为470300000为起始点, 200kHz依次递增的的8个频点,; 下行频率为500300000为起始点, 200kHz依次递增的的8个频点,;</p> <p>指令配置:</p> <p>AT+BAND = 6</p> <p>AT+FREQ = 1, 3, 470300000</p> <p>AT+FREQ = 3, 8, 500300000</p> <p>如上配置, 用户查询频率得到的参数如左侧所示</p>
-------------------	---	--

### 6.3.29 设置注册入网信息 JOIN

设置JOIN信息		
描述	<p>该指令配置注册入网相关参数</p> <ul style="list-style-type: none"> <li>◆ 以下说明中JOIN表示注册入网</li> <li>◆ 该指令的最大响应时间为50ms</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	<p>格式1:</p> <p>AT+JOIN=X1</p> <p>格式2:</p> <p>AT+JOIN=X1,&lt;X2&gt;,&lt;X3&gt;,&lt;X4&gt;</p>	<p>X1: 执行JOIN操作, X1取值范围:</p> <p>1 - 清除注册上下文, 重启后切换至透传模式开始新一轮扫描JOIN</p> <p>6 - 不清除注册上下文, 立即开始新一轮JOIN</p> <p>X2: 参数保留, 默认值为1</p> <p>X3: 单次JOIN周期, X3取值范围: 11~255, 单位为s, 默认值: 13</p> <p>X4: JOIN最大尝试次数, X4取值范围: 1~255, 默认值: 176</p>
响应1	OK	设置成功
响应2	ERROR	未知错误
示例1	<p>AT+JOIN=1</p> <p>OK</p>	清除注册上下文, 手动重启模块 (或执行AT+RESET指令) 后切换至透传模式开始新一轮JOIN
示例2	<p>AT+JOIN=6</p> <p>OK</p>	不清除注册上下文, 立即开始JOIN
示例3	<p>AT+JOIN=6,1,15,20</p> <p>OK</p>	清除注册上下文, 立即开始JOIN, JOIN周期为15s, 最大尝试次数为20

### 6.3.30 读取注册入网信息 JOIN

读取JOIN信息		
描述	该指令读取注册入网信息 ◆ 以下说明中JOIN表示注册入网	
指令	AT+JOIN?	读取JOIN信息Y0,Y1,...Y11 Y0 表示当前注册入网结果 0 – 模块未入网 1 – 模块已入网 Y1 保留，默认为1 Y2 保留，默认为1 Y3 表示单次JOIN周期，单位为s Y4 JOIN最大尝试次数 Y5 当前JOIN信道，0xFF为无效值 Y6 首次JOIN退避机制 0 - 连续JOIN，无退避 1 - 保留 2 – CLAA首次JOIN退避机制，指数退避 Y7 固定值，默认为57600 Y8 默认值，固定为3600 Y9 参数保留 Y10 参数保留，默认为0 Y11 已发送的JOIN请求次数-1，从0开始计数，该参数动态变化，重启模块（硬重启或者软重启）或清除注册上下文可刷新该值
响应1	+JOIN:Y0,Y1,...,Y11 OK	读取成功，返回结果
响应2	ERROR	未知错误
示例	AT+JOIN? +JOIN: 1, 1, 1, 13, 176, 3, 0, 57600, 3600, 0, 0, 0 OK	Y0=1，模块已入网 Y3 = 13，模块单次入网扫描周期为13s Y4 = 176，模块JOIN最大尝试次数为176次 Y11= 0，表示模块已发送1次JOIN请求

### 6.3.31 设置状态寄存器 Status\_Output\_Reg

设置状态寄存器		
描述	<p>该指令配置模块在发送给用户MCU的服务器下行数据包中，是否增加状态信息输出</p> <ul style="list-style-type: none"> <li>◆ Status_Output_RegX (X=1,2,3) 的详细内容如 “<a href="#">表4-5 数据帧内容释义</a>” 所示</li> <li>◆ 使用该指令设置的参数当前不可读</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+STATUS=X1,X2	<p>X1: 功能索引值，取值范围：</p> <p>1 – 服务器下行数据长度输出</p> <p>2 – 状态信息输出</p> <p>X2: 值与X1有关，具体定义如“<a href="#">表6-1 Status_Output_Reg的功能索引</a>”所示</p>
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1	AT+STATUS=1,1	<p>使能服务器下行数据长度输出；</p> <p>模块发送的串口数据 = 服务器下行数据长度 + 服务器数据</p>
示例2	AT+STATUS=2,1	<p>使能状态寄存器Status_Output_Reg1输出；</p> <p>模块发送的串口数据 = 服务器下行数据 + Status_Output_Reg1</p>
示例3	AT+STATUS=2,2	<p>使能状态寄存器Status_Output_Reg1 + Status_Output_Reg2输出；</p> <p>模块发送的串口数据 = 服务器下行数据 + Status_Output_Reg1 + Status_Output_Reg2</p>

表6-1 Status\_Output\_Reg的功能索引

X1	X2	描述
1	<p>0 - 关闭</p> <p>1 - 使能</p>	<p>X1: 服务器下行数据长度输出功能索引</p> <p>X2: 使能/失能服务器下行数据长度输出指示(UART_LEN_IND)，默认值为0</p> <p>使能该功能后，模块每接收到一个有效的服务器下行数据后，</p> <p>串口发送给用户MCU数据 = 长度字节+数据</p>
2	<p>0 - 0字节</p> <p>1 - 1 Byte</p> <p>2 - 3 Bytes</p>	<p>X1: 状态信息输出功能索引</p> <p>X2: 配置是否输出相关Status_Output_Reg，默认值为0</p> <p>配置该功能后，模块每接收到一个有效的服务器下行数据后，模块串口发送给用户MCU数据 = 数据+[X2]状态字节</p> <p>注：状态字节具体含义请参考 “<a href="#">表4.4 串口响应数据帧格式</a>” (Pages of 18 )</p>

### 6.3.32 读取状态寄存器 Comm\_Status\_Reg

读取状态信息		
描述	该指令读取模块的Comm_Status_Reg信息 ◆ 该指令返回值包含上行结果与下行结果部分内容	
指令	AT+STATUS?	返回模块通信状态信息，内容如“ <a href="#">表6-2 Comm_Status_Reg内容释义</a> ”所示
响应1	+STATUS0:Y0,Y1,...Y39 OK	读取状态成功
响应2	ERROR	未知错误

模块每完成一次数据服务后，模块更新通信状态寄存器值（Comm\_Status\_Reg），模块返回的状态信息如表 6-2 所示。

表6-2 Comm\_Status\_Reg内容释义

寄存器Offset	寄存器功能	字节	描述
0	当前通信结果	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） 06 – 网络可能异常 07 – 发送数据成功，无下行 08 – 发送数据成功，有下行
1	上行信道	1	本次上行所使用的信道
2	上行速率	1	本次上行所使用的速率
3	上行功率	1	本次上行所使用的功率
4	下行信道	1	本次下行所使用的信道
5	下行速率	1	本次下行所使用的速率
6	RFU	1	保留
7	RXSLOT	1	本次下行的接收窗口
8	PORT	1	本次下行PORT
9	RFU	1	保留
10	RFU	1	保留



11	SNR	1	本次下行的SNR，以补码表示。 i.e.:SNR = +6dB, 则 Data[11] = 0x06 i.e.:SNR = -1dB, 则 Data[11] = 0xFF
12	RFU	1	保留
13~14	RSSI	2	本次下行的RSSI，补码表示。仅低字节有效，高字节保留 i.e.:RSSI = -87dBm, 则 Data[13] = 0xA9, Data[14]=保留
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

### 6.3.33 设置 Link Check

设置 Link Check			
描述	该指令用来检测网络无线链路质量，类似以太网中的ping功能。 ◆ 该指令在模块成功join网络后才有效		
指令	AT+LINK CHECK=X1		X1: Link Check使能控制，出厂值为0 0 - 关闭Link Check 1 - 执行一次Link Check 2 - 模块自动在每次上行数据包中携带Link Check指令 注1: 如果X1=1, 该指令执行成功后, 会先后产生两条响应
响应1	若X1=1		返回OK, 设置成功。 若X1=1, 等待一段时间后, 会返回第二条响应信息: Y0: Link Check结果 0 - 表示本次Link Check执行成功 非0 - 表示本次Link Check执行失败 Y1: DemodMargin Y2: NbGateways Y3: 本次下行的RSSI Y4: 本次下行的SNR
	第一条响应	OK	
	第二条响应	+LINK INFO: Y0, Y1, Y2, Y3, Y4	
响应2	BAD PARM		参数错误
响应3	ERROR		未知错误

示例1	AT+LINK CHECK=1 OK +LINK INFO: 0, 17, 1, -52, 8	模块执行一次Link Check操作(等待网络交互) 模块返回Link Check结果。
示例2	AT+LINKCHECK=2 OK	设置模块自动携带Link Check指令到每个上行包中。 便于用户前期系统测试，检查终端产品的上下行无线链路质量

### 6.3.34 读取 Link Check

读取 Link Check配置		
描述	该指令读取Link Check配置参数	
指令	AT+LINK CHECK?	读取Link Check参数配置
响应1	+LINK CHECK: X1,X2 OK	读取成功，返回结果 X1：是否已执行Link Check，取值范围： 0 – 未执行Link Check 1 – 已执行Link Check X2：参数保留
响应2	ERROR	未知错误
示例	AT+LINK CHECK? +LINK CHECK: 1,100 OK	读取成功，已执行Link Check

### 6.3.35 设置 RX2

读取RX2		
描述	该命令设置模块RX2参数 ◆ 注意该参数需要与服务器端配置保持一致 ◆ 该参数修改后，若需要掉电存储，需要执行AT+SAVE指令	
指令	AT+RX2=X1,X2	X1: 速率 0 - DR0 (SF12 BW125) 1 - DR1 (SF11 BW125) 2 - DR2 (SF10 BW125) 3 - DR3 (SF9 BW125) 4 - DR4 (SF8 BW125) 5 - DR5 (SF7 BW125) X2: RX2通信频率, 单位Hz
响应1	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+RX2=0,505300000	设置模块RX2速率为DR0, 频率为505300000

### 6.3.36 读取 RX2

读取RX2		
描述	该指令模块读取当前RX2参数	
指令	AT+RX2?	返回RX2的当前参数 X1: 速率 X2: RX2使用频率
响应1	+RX2:X1,X2 OK	读取成功, 返回结果
响应2	ERROR	未知错误
示例	AT+RX2? +BAND:0,502500000 OK	当前模块RX2速率为DR0, 频率为502500000

### 6.3.37 设置 CSMA

设置CSMA		
描述	<p>该指令设置模块的CSMA功能</p> <ul style="list-style-type: none"> <li>◆ 若开启模块CSMA功能，模块每次发送数据前，会先检查空口是否空闲，若检查到空口空闲，则发送数据，否则随机退避后，重新执行上述流程。最大退避次数缺省为4</li> <li>◆ 对于时延不敏感的应用，开启CSMA功能，可以在一定程度上提高模块的通信成功率</li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+CSMA=X1	<p>X1: 关闭/开启CSMA功能，默认值为0</p> <p>0 - 关闭</p> <p>1 - 开启</p>
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+CSMA=1	开启CSMA

### 6.3.38 读取 CSMA

设置CSMA		
描述	该指令设置模块的CSMA功能	
指令	AT+CSMA?	读取CSMA当前配置
响应1	+CSMA:X1 OK	读取成功
响应2	ERROR	未知错误
示例	AT+CSMA? +CSMA: 0 OK	当前未开启CSMA

### 6.3.39 时间同步 TIMESYNC

时间同步TIMESYNC		
描述	<p>该指令获取服务器实时时间，并更新本地RTC时间</p> <ul style="list-style-type: none"> <li>◆ 该指令需模块入网成功后执行有效</li> <li>◆ 模块复位后，需要再次执行该指令，不支持掉电保存</li> </ul>	
指令	AT+TIMESYNC	获取服务器实时时间
响应1	第一条响应	OK
	第二条响应	+RTC INFO: Y0, Y1, Y2, Y3, Y4, Y5
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	<p>AT+TIMESYNC</p> <p>OK</p> <p>+RTC INFO: 20 01 17 11 05 13</p>	

### 6.3.40 Class C 组播

Class C组播		
描述	<p>该指令用于配置模块的组播参数</p> <ul style="list-style-type: none"> <li>◆ 前提：模块的设备类型为Class C</li> <li>◆ 模块端配置的组播地址与服务器端的组播地址<b>大小端相反</b></li> <li>◆ 该参数修改后，执行AT+SAVE指令可掉电保存</li> </ul>	
指令	AT+DEVADDR=X1, X2, X3, X4	<p>X1: MultiDevAddr, 组播地址，从服务器获取，大小端与服务器相反</p> <p>X2: 参数保留，固定为1</p> <p>X3: MultiAppSKey, 组播AppSKey, 从服务器获取</p> <p>X4: MultiNwkSKey, 组播NwkSKey, 从服务器获取</p>
响应1	OK	返回OK，设置成功
响应2	BAD PARM	参数错误
响应3	ERROR	未知错误
示例	<p>AT+DEVADDR=BD3E6369,1,6EB161870C05EA29C348200DF7C22A38,0151DF65AFA4FFFC091CD7CB4D342F02</p>	

## 7 低功耗处理

在**透传模式**下，用户可在 BUSY 引脚为高电平（空闲状态）时拉低 WAKE 引脚，使模块进入睡眠。用户拉低 WAKE 引脚之前，模块相关引脚的处理建议，如表 7-1 所示。

表7-1 模块睡眠的引脚状态

模块端			用户端
引脚	引脚功能	端口类型	用户MCU引脚处理建议
12	WAKE	INPUT	用户MCU输出并拉低WAKE引脚，并保持
13	MODE	INPUT	用户MCU输出并拉低MODE引脚，并保持
6	TXD	OUTPUT	用户MCU浮空输入
5	RXD	INPUT	用户MCU输出并拉高
3	BUSY	OUTPUT	用户MCU浮空输入
4	STAT	OUTPUT	用户MCU浮空输入
11	NRST	INPUT	用户MCU输出并拉高NRST引脚，或者浮空输入

## 8 应用配置说明

模块出厂默认烧入 DevEui，用户需自行烧写 AppEui (JoinEui)、Appkey。

### 8.1 出厂参数说明

表8-1 模块出厂缺省参数

序号	类型	主要参数	缺省值	说明
1	系统配置	Baud	9600	透传模式的通信波特率为9600bps，指令模式固定为9600bps
2		RX Timeout	0	串口分帧超时时间为10ms
3		工作模式	-	透传模式，用户通过MODE引脚控制
4		回显	0	关闭回显
5		日志 (DEBUG)	0	默认关闭日志输出
1	LoRaWAN	DEVEUI <sup>注1</sup>	-	出厂已烧写固化，保证全网唯一
2		APPEUI	-	当前无有效出厂值，保存为用户最新写入参数
3		APPKEY	-	当前无有效出厂值，保存为用户最新写入参数
4		OTAA	1	OTAA入网激活
5		Band	7	入网扫描模式为同频
6		Class	0	设备类型为Class A
7		Join Accept Delay1	5000	Join Accept delay1默认为5000ms
8		Join Accept Delay2	6000	Join Accept delay2默认为6000ms
9		Rx1Delay	1000	RX1DELAY默认为1000ms，Rx2Delay = Rx1Delay + 1
10		Unconfirm	0	非确认帧，非确认帧默认发送次数为1
11		Nbtrials	4	确认帧每次通信失败，最大发送次数为4次
12		Ack Fail	32	确认帧连续通信失败阈值，超过32次模块将进入连接丢失状态
13		Datarate	3	初始无线通信速率为DR 3 (SF9 BW125)。即首次JOIN速率为DR 3 (SF9 BW125)
14		DR Range	0x50	最高速率为DR5，最低速率为DR0
15		Output Power	0	初始功率为最大值 (20dBm)
16		MAX_FCNT_GAP	16384	下行Fcnt丢包门限值
17		ADR	1	开启自动调速功能，允许模块与服务器调速
18		ADR_ACK_LIMIT	64	ADR开始调速门限值

19		ADR_ACK_DELAY	32	ADR持续调速门限值
20		ADR_ACK_TIMEOUT	32	非确认帧掉网门限值
21		RX2 Freq	505.3M	RX2 frequency = 505300000Hz
22		RX2 Datarate	0	CLAA D模式 RX2 Datarate = SF12 BW125
23		FPort	0x0A	透传模块缺省值为0x0A
24		热启动 <sup>注</sup>	0	关闭热启动，已入网的设备重新上电或者复位，需要重新JOIN
25		JOIN interval	13	单次JOIN周期为13s
26		JOIN number	176	JOIN最大尝试次数，默认为176次

**注：**

由于热启动的设备，一旦注册成功，上电将不会执行注册操作。

如果终端为热启动产品，用户在出厂前若进行终端产品的联网测试，建议用户在工厂联网测试完成后，执行一次清除注册上下文操作(AT+JOIN=1)，以避免产品实际使用时，工厂产测网络参数与实际商用网络参数不一致，避免设备不能正常使用。



## 8.2 OTAA快速配置示例

下文描述了出厂模块（已含 DevEui）配置为 OTAA 激活模式最小需要配置/修改的参数（如果未配置，将按照表 8-1 默认参数运行），用户可以根据实际需求设置其他特性参数。

(1) 进入指令模式

拉高MODE引脚，延时10ms

(3) 配置AppEUI，(该参数FF 00 66 FF FF FF 01仅作为示例说明，以实际使用为准)

AT+APPEUI= FF 00 66 FF FF FF 01

(4) 配置AppKey，(该密钥00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF仅作为示例说明，以实际使用为准)

AT+APPKEY= 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

(5) 保存参数到EEPROM

AT+SAVE

(6) 复位，重启模块

AT+RESET或者拉低RESET引脚，拉高后至少延时150ms

(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，延时10ms

(8) 等待BUSY和STAT引脚变为高电平，表明模块加入网络成功，此时用户可以通过串口收发数据。

(以上任何参数发生变化，需要执行SAVE，并复位后，处于透传模式的模块将使用新参数JOIN网络)

注：

正常应用中，用户配置并保存入网参数到EEPROM后，若不需要改变，则用户只需要在产品中设置一次即可。

## 8.3 Class C应用

### 8.3.1 应用场景

若用户应用场景符合以下条件之一：

1. 终端无低功耗要求，且希望获得更实时的下行控制。
2. 模块一直保持接收。

用户可以通过 AT+CLASS=2 指令设置模块的设备类型为 Class C；该类型的设备，除了发送数据外，将一直保持接收，入网（JOIN）机制与 Class A 设备相同。

**注意：**

对于 Class C 的设备，用户需要线下与服务器平台充分沟通，参数保持一致，主要有以下两点：

1. 服务器平台需要登记该设备为 Class C
2. RX2 的频点与速率均也需要与服务器保持一致。

### 8.3.2 典型配置

Class C 典型配置方式如下：

(1) 进入指令模式

WAKE引脚拉高，MODE引脚拉高

(2) 设置为Class C（服务器登记该设备的设备类型需要为Class C）

AT+CLASS = 2

(3) 配置RX2（频点、速率需要和服务器保持一致）

AT+RX2 = 0,505300000

(5) 保存

AT+SAVE

(6) 复位，重启模块

AT+RESET或者拉低RESET引脚，拉高后至少延时150ms

### 8.3.3 组播配置

若设备应用于为 Class C 组播，设备需配置组播命令，指令配置参考“[6.3.40 Class C 组播](#)”，终端详细配置文档请参考手册《利尔达自主 LoRaWAN 组播测试操作步骤》。

## 9 日志信息

用户可以通过指令“AT+DEBUG=1”来开启模块日志信息输出。

如表 9-1 所示，若使能模块的日志输出功能，模块在注册、每次空口数据发送（unconfirm 帧）、每次空口数据发送和接收（confirm 帧），都会输出相应的日志信息，辅助用户调试与测试。

日志信息主要包含注册入网过程与数据传输过程相关的状态信息。

表9-1 日志信息

日志类型		描述
网络管理	注册	+JOIN INFO: Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8 详细信息请参考“ <a href="#">9.1 注册入网过程</a> ”
数据通信	数据上行	[UP] Cfm_Stat: <b>状态</b> , UpCnt: 上行帧计数, CH[信道]: 频点, DR[速率]: 上行SF, TxPower: 发射功率, AckCnt: 保留, PER: 保留, TOA: 空口传输时间, nbTrials: 发送次数, BackOff: 保留, cRssi: 保留, Nb: 保留, Band: 保留 详细内容请参考“ <a href="#">9.2 数据传输过程</a> ”
	数据下行	[DN] Ind_Stat: <b>状态</b> , DnCnt: 下行帧计数, CH[信道]: 频点, DR[速率]: 下行SF, Port: 应用端口, RxSlot: RX窗口, pRSSI: 下行rssi, SNR: 下行SNR, Pend:服务器有更多数据, SIZE: 下行数据包大小, Multi: xx, Margin: xx, nGW: xx 详细内容请参考“ <a href="#">9.2 数据传输过程</a> ”

表 9-1 中的“状态”，如表 9-2 所示。

表9-2 状态定义

STATUS	描述
0	成功（发送成功\接收成功）
1	未知错误
2	发送超时
3	RX2接收超时
4	RX2接收出现错误情况
5	JOIN出现错误情况
6	无效的下行计数值
7	节点丢 失应答帧超过MAX_FCNT_GAP
8	地址校验错误
9	消息校验错误
11	RX1超时

## 9.1 注册入网过程

模块在 JOIN 网络过程中，每执行完一次空口注册操作，会输出如表 9-3 所示信息。

表9-3 模块注册信息

注册入网过程中的日志信息
+JOIN INFO:Y0, Y1, [Y2]Y3, Y4, Y5, Y6, Y7, Y8
说明:
Y0 注册结果
0 – 注册成功
3 – 本次注册失败
注: 其他可能出现的结果, 可参考 “ <a href="#">表9-2 状态定义</a> ”
Y1 表示当前入网扫描模式
6 – 异频
7 – 同频
Y2 表示当前通信所使用信道
Y3 表示当前通信所使用频点
Y4 表示当前第几次注册-1 (从0开始计数)
Y5 表示当前通信所使用速率
Y6 表示当前通信所使用功率
Y7 表示当前下行通信的RSSI(dBm) , 通信成功时有效
Y8 表示当前下行通信的SNR, 通信成功时有效
示例:
+JOIN INFO: 0, 7, [3]475900000, 0, 12, 11, -58, 5
表示注册成功, 同频, 信道3, 频率475900000Hz, 第1次注册, SF12, 功率11dBm, RSSI-66dBm, SNR+8dBm

## 9.2 数据传输过程

每一次数据通信结束后，模块输出对应的上行（或上行和下行）日志信息。

### 数据传输过程中的日志信息

示例：

[UP] Cfm\_Stat: 0,UpCnt: 0,CH4: 476100000,DR0: SF12,TxPower: 11,AckCnt: 1,PER: 0,TOA: 1155072,nbTrials: 1,BackOff:

0,cRssi: 0,Nb: 0,Band: 7

[DN] Ind\_Stat: 0,DnCnt: 0,CH4: 476100000,DR0: SF12,Port: 0,RxSlot: 0,pRSSI: -50,SNR: +5,Pend: 0,SIZE: 0,Multi: 0,Margin:

0,nGW: 0

说明：

若设置AT+DEBUG=1，则用户每发送一个新数据包后，模块输出类似上述信息段；

- 1、若没有接收到下行数据包，则只输出【UP】相关信息；
- 2、若有接收下行数据包（纯MAC层的应答或者用户数据），则输出【UP】和【DN】相关信息。

## 10 配套资源

### 10.1 LoRaWAN开发板

LoRaWAN 开发板 LSD4WN-2NTEST0A 是一款配合 LoRaWAN 各系列模组使用的配套开发工具。该开发板有丰富的外设资源以供开发与前期评估。

1. 集成了温湿度传感器、光照传感器、GPS等各类传感器资源，可模拟实现实际中的相关应用；
2. 集成了丰富的调试接口。若进行模块功能测试，用户可通过MCU操作模块或者直接通过PC串口操作模块，若进行功耗测试，用户可获取整板功耗或者模块功耗。

### 10.2 开发板配套程序

开发板 LSD4WN-2NTEST0A 配套程序结合了 LoRaWAN 各系列模块的特性，用户可利用该代码来操作模块并且进行开发、移植等。该份配套程序包含了指令模式、透传模式、定时发送模式和休眠定时发送模式四种模式。

1. 指令模式：用户可通过PC串口发送AT指令，进行指令配置，参数读取等操作。
2. 透传模式：用户进行注册入网或通过PC发送数据包。
3. 定时发送模式：MCU定时发送数据包给模块，无功耗优化。
4. 休眠定时发送模式：MCU定时发送数据包给模块，包含功耗优化，功耗优化主要体现在数据包发送到接收之间MCU休眠、数据包发送结束后MCU和模块均休眠。

### 10.3 资源下载

LoRaWAN 开发板以及开发板配套程序相关资料可登录网址：<http://rf.lierda.com/> 资料下载板块进行下载；或登录我司官方论坛 <http://bbs.lierda.com/> LoRaWAN\RF 版块进行下载。

# 11 产品变更记录

模块固件版本	描述

# 12 文件修订历史

版本	修订日期	修订说明



## 13 敬告用户

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。

浙江利尔达物联网技术有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范，参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，利尔达公司有权对该文档进行更新。

浙江利尔达物联网技术有限公司始终以为客户提供最及时、最全面的服务为宗旨，如需任何帮助,请随时联系我司各地分部或浙江总部。

版权所有 © 利尔达科技集团，保留一切权利。

**Copyright © Lierda Science & Technology Group Co.,Ltd**

编制：利尔达物联网技术有限公司