



YL-300T

NB-IOT, 超低功耗

电话: 0755-26031631

E-mail: yl-link@rf-module.cn

地址: 深圳市南山区科技园中区科智西路 1 号科苑西工业区南 23 栋 6 楼

产品概述

YL-300T是一款NB-IOT的标准通信模块，直接与通信运营商的基站进行通信，可定制带定位功能的NB模组，实时上报定位信息;支持多种物联网平台协议。

❖ OneNET 中国移动物联网开放平台

❖ 消息队列遥测传输

IBM开发的一个即时通讯协议，有可能成为物联网的重要组成部分。该协议支持所有平台，几乎可以把所有联网物品和外部连接起来，被用来当做传感器和致动器（比如通过Twitter让房屋联网）的通信协议

❖ 超文本传输协议

互联网上应用最为广泛的一种网络传输协议，

所有的WWW文件都必须遵守这个标准。设计HTTP最初的目的是为了提供一种发布和接收HTML页面的方法。

❖ 支持多种工作状态

✧ Connected(连接态)

模块注册入网后处于该状态，可以发送和接收数据，无数据交互超过一段时间后会进入Idle模式，时间可配置

✧ Idle(空闲态)

可收发数据，且接收下行数据会进入Connected状态，无数据交互超过一段时会进入PSM模式，时间可配置。空闲状态可配置执行DRX或eDRX模式

✧ 不连续接收 (DRX)

discontinuous reception, 不连续接收模式。对下行业务时延要求高，如路灯。

✧ 扩展不连续接收 (eDRX)

Extended idle mode DRX, 扩展不连续接收模式。对下行业务时延有较高要求，可根据设备是否处于休眠状态缓存消息或者立即下发消息，如智能穿戴。

✧ 5PSM(节能模式)

此模式下终端关闭收发信号机，不监听无线侧的寻呼，因此虽然依旧注册在网络，但信令不可达，无法收到下行数据，功率很小。



产品特点

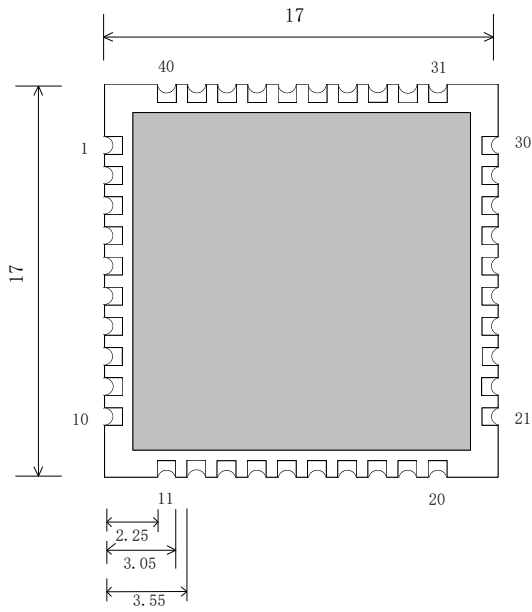
- 贴片式封装，方便嵌入客户端。
- 多种快速休眠，快速唤醒机制。
- 直接和服务端通信。
- 支持标准的 TCP 和 UDP 通信。
- 支持 HTTP 超文本传输协议
- 支持 MQTT 消息订阅服务
- 多种工作模式灵活切换

应用领域

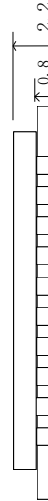
- ✓ 智能烟感设备
- ✓ 水、电、气、暖等计量表自动集中抄表系统
- ✓ 水利、油田、矿井、气象等设备信息采集
- ✓ 路灯控制、电网监测、风光互补系统
- ✓ 移动定位设备

尺寸结构

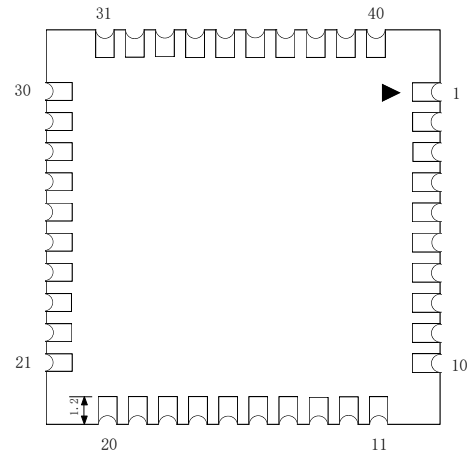
Top View



Side View



Bottom View

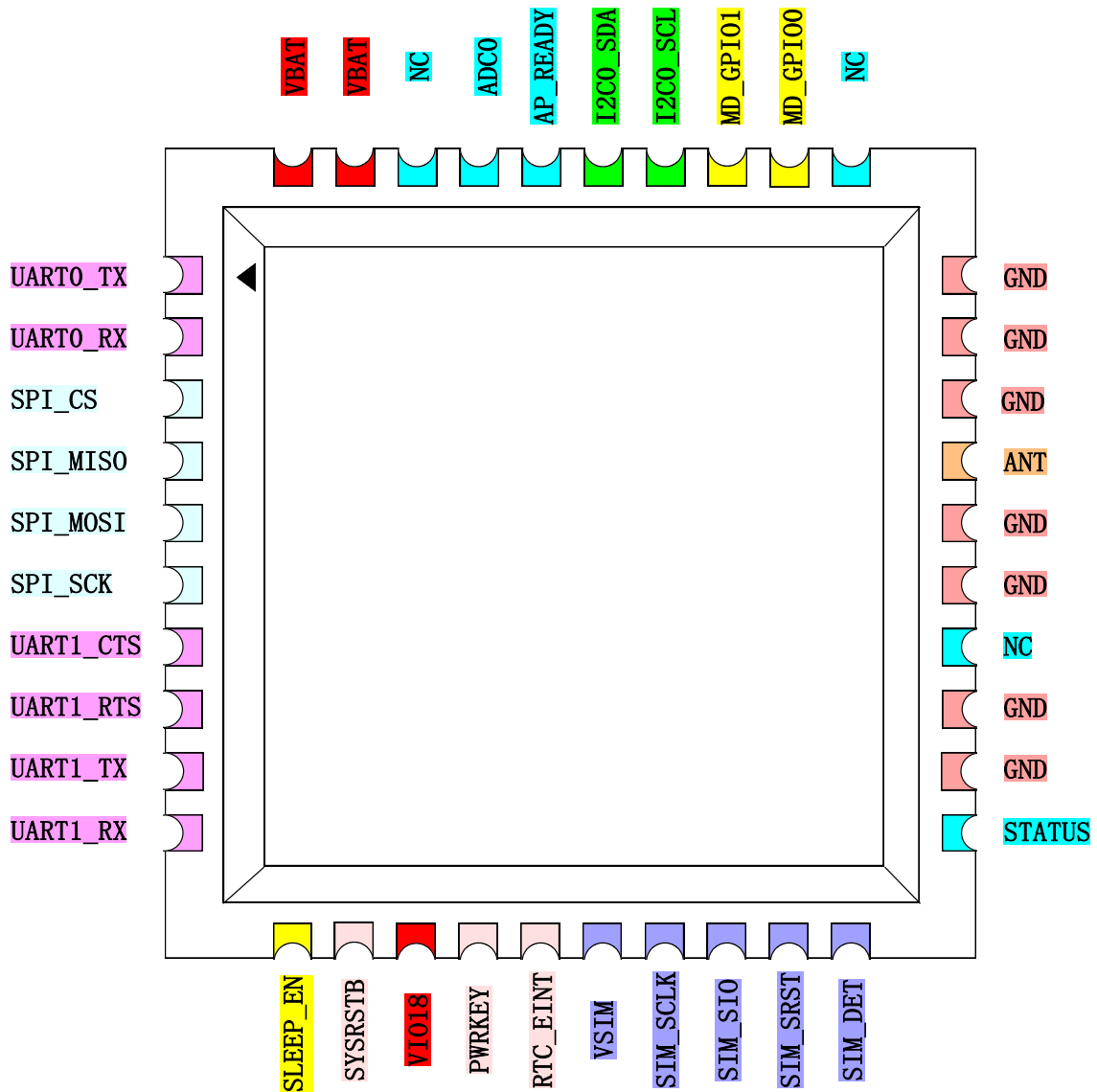


Unit:mm

引脚定义

YL-300T 模块是 SMT 封装，外部预留了多种通信接口，如串口，SPI ， I2C 等,还有供电电源接口，SIM 卡接口，外部控制接口。

引脚分布



引脚描述

引脚符号	引脚功能	电压域
UART0_TX	UART0数据输出引脚, 接用户的输入引脚	1.8V
UART0_RX	UART0数据输入引脚, 接用户的输出引脚	1.8V
SPI_CS	SPI片选引脚	1.8V
SPI_MISO	SPI主入从出引脚	1.8V
SPI_MOSI	SPI主出从入引脚	1.8V
SPI_SCK	SPI时钟引脚	1.8V
UART1_CTS	UART1清除发送引脚	1.8V
UART1_RTS	UART1准备接收引脚	1.8V
UART1_TX	UART1数据输出引脚, 接用户的输入引脚	1.8V
UART1_RX	UART1数据输入引脚, 接用户的输出引脚	1.8V
SLEEP_EN	模块休眠使能脚	1.8V
SYSRSTB	系统重启	1.8V
VIO18	1.8V输出电压引脚	1.8V
PWRKEY	开机键	0—3.6V
RTC_EINT	内部时钟中断引脚, 拉低可退出PSM模式	1.8V
VSIM	SIM电源引脚, 由模块输出	1.8V/3V
SIM_SCLK	SIM时钟引脚, 时钟信号由模块输出	1.8V/3V
SIM_SIO	SIM数据引脚, 双向信号	1.8V/3V
SIM_SRST	SIM复位引脚, 复位信号由模块输出	1.8V/3V
SIM_DET	SIM检测引脚	1.8V/3V
STATUS	模块状态	1.8V

GND	接地，模拟信号地	
GND	接地，模拟信号地	
NC	无	
GND	接地，射频信号地	
GND	接地，射频信号地	
ANT	天线	
GND	接地，射频信号地	
GND	接地，射频信号地	
GND	接地，模拟信号地	
NC	无	
MD_GPIO0	通用IO口0引脚	1.8V
MD_GPIO1	通用IO口1引脚	1.8V
I ² CO_SCL	I ² CO的时钟引脚	1.8V
I ² CO_SDA	I ² CO的数据引脚	1.8V
AP_READY	预留IO口引脚	1.8V
ADCO	模数转换口引脚	1.4V
NC	无	
VBAT	电源	2.2V—3.6V
VBAT	电源	2.2V—3.6V

技术参数

供电范围：2.2V—3.6V
省电电流：支持PSM&40uA, eDRX&12uA两种低功耗模式
支持协议：支持3GPP R13/R14协议
支持频段：B1/B2/B3/B5/B8/B11/B12/B13/B17/B18/B19/ B20/B21/B25/B26/B28/B31/B66/B70/B71
数据带宽：理论数据UL 20 kbps/DL 25 kbps
发射功率：23dBm
SIM卡接口：支持SIM卡1.8V, 3.0V
数据格式：支持UTP和TCP。
射频输出：50Ω
固件升级：支持串口升级。
稳定范围：常规温度-35℃~75℃, 扩展温度-40℃~85℃

电气特性

模块功耗

模式	测试条件	最小值	平均值	最大值
IDLE	VBAT=3.3V, DRX CYCLE=2.56s	-	233uA	-
ACTIVE	VBAT=3.3V, DRX CYCLE==2.56s	-	288uA	-
PSM	VBAT=3.3V	-	3.3uA	-
eDRX	VBAT=3.3V, DRX CYCLE=81.92s	-	66.2uA	-
POWER OFF	VBAT=3.3V	-	2.3uA	-
DEEP SLEEP	VBAT=3.3V	-	3.2uA	-
LIGHT SLEEP	VBAT=3.3V	-	49uA	-

绝对最大额定参数

名称	说明	最小值	推荐值	最大值
VBAT	工作电压	2.2V	3.3V	3.63V
VI018	数字 IO 口电源	1.7V	1.8V	1.9V
IDDP	峰值电流	-0.3	-	1A

数字接口特性

名称	说明	最小值	典型值	最大值
VIL	输入低电平	0V	-	0.7V
VIH	输入高电平	1.62V	1.8V	1.98V
VOL	输出低电平	0V	-	0.3V
VOH	输出高电平	1.62V	1.8V	1.98V

ESD 防护

在生产、装配模块时需要注意 ESD 防护，避免静电对模块造成损坏。模块 ESD 性能参数如下：

引脚	接触放电	空气放电
VBAT	±4KV	±8KV
GND	±4KV	±8KV
ANT	±4KV	±8KV

注：（温度：25℃，湿度：45%）

硬件设计电路

YL-300T 是一款高度集成模块,所以在设计硬件时要考虑周全,特别是在电平转换,电源的处理以及 ESD 保护电路的设计。

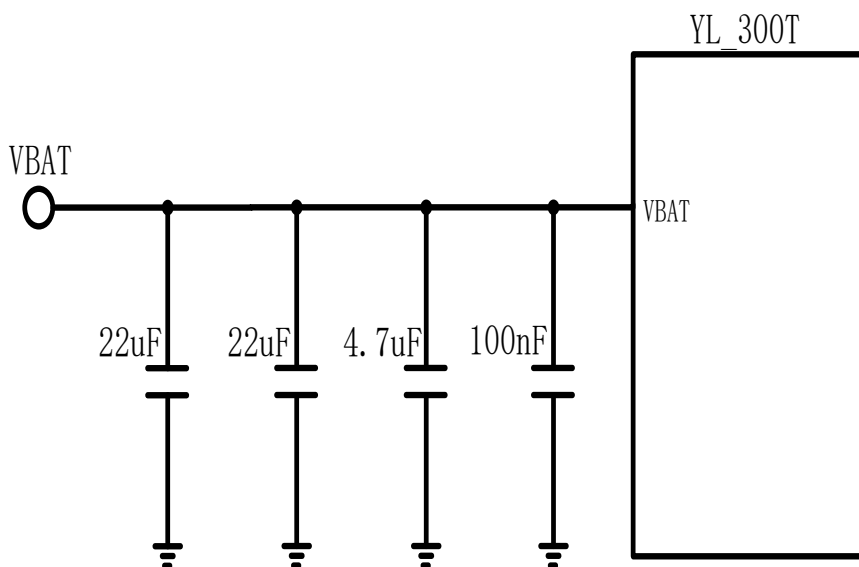
电源的设计

VBAT 为模块的主电源,其电压输入范围是 2.2V—3.6V,推荐电压为 3.3V。在网络较差环境下,天线会以最大功率发射,模块瞬态最大峰值电流可能达到 1A。电源的峰值电流供电能力要达到 1A 以上,建议靠近 VBAT 使用至少 2 个 22 μ F 以上的电容,以提高电源的瞬间大电流续流能力。PCB 布局时,电容尽可能靠近模块的电源引脚,VBAT 走线宽度为 1mm 左右,电源部分的地平面尽量完整,且多打地孔。

※ VBAT 接口管脚定义

名称	管脚号	说明
VBAT	39, 40	模块电源正极输入端
GND	22, 23, 30	模块电源负极输入端

※ VBAT 接口参考设计



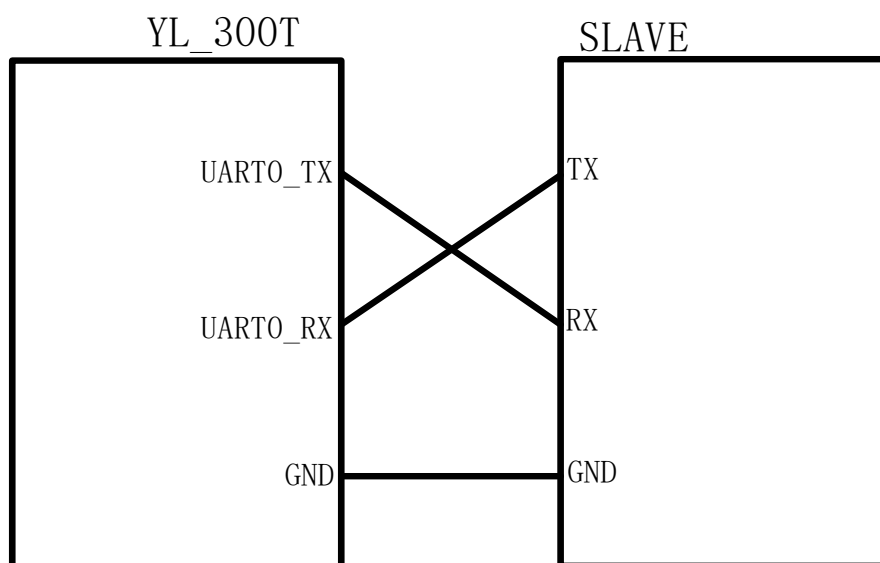
UART 接口设计

YL-300T 有 2 组 UART 接口，最大波特率为 921600bps。其中，UART0 为模块的 AT 指令串口，用来处理 AT 指令信息。UART1 为模块的调试串口，在开机后串口会自动向外面输出 log 信息。

※ UART 接口管脚定义

名称	管脚号	说明
UART0_TX	1	模块串口数据发射，接其他设备接收脚
UART0_RX	2	模块串口数据接收，接其他设备发射脚

※ UART0 接口连接示意图



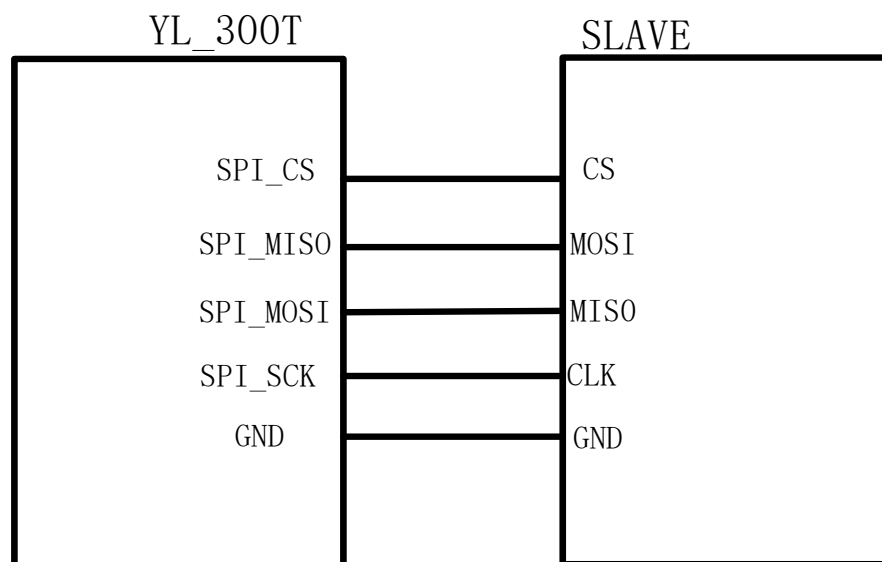
SPI 接口设计

YL-300T 带有 1 组标准 4 线 SPI 接口，可以与 SPI 接口的外设通讯。

※ SPI 接口管脚定义

名称	管脚号	说明
SPI_CS	3	SPI 片选，接从设备片选脚
SPI_MISO	4	SPI 数据输入，接从设备输出脚
SPI_MOSI	5	SPI 数据输出，接从设备输入脚
SPI_SCK	6	SPI 时钟，接从设备时钟

※ SPI 接口连接示意图



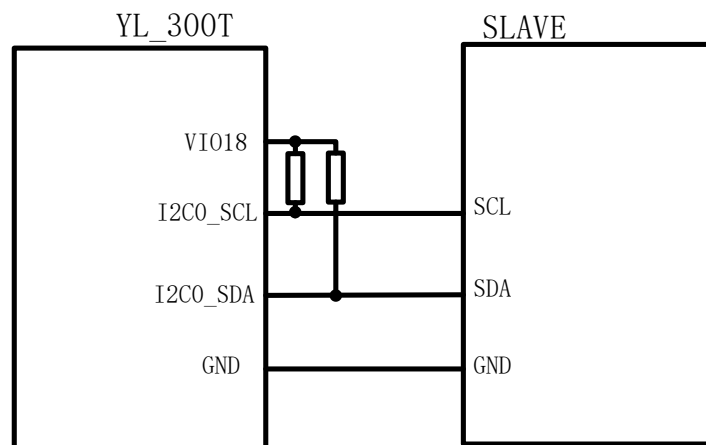
IIC 接口设计

YL-300T 带有 1 组 IIC 接口，可以与 IIC 接口的外设通讯，外部需要加 4.7K 电阻上拉到 VI018。

※ I2C0 接口管脚定义

名称	管脚号	说明
I2C0_SCL	34	I2C 时钟
I2C0_SDA	35	I2C 数据

※ I2C0 接口连接示意图



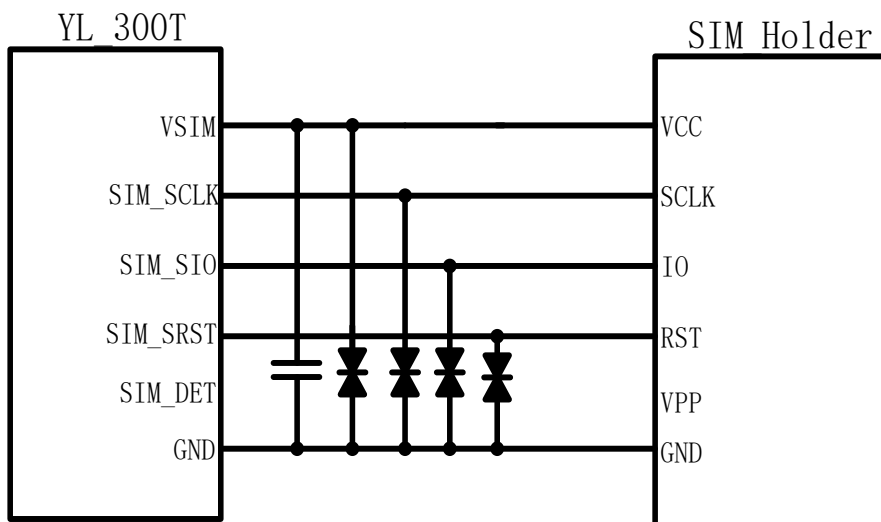
SIM 卡接口设计

SIM 卡通过模块内部的电源供电，支持 1.8V 和 3.0V 供电。在 PCB LAYOUT 时 SIM 卡座应靠近模块引脚摆放，VSIM 引脚需对地加 1uF 滤波电容，电源走线尽量短而粗。信号线最好加 ESD 保护器件，且 ESD 器件的寄生电容不超过 22pF。SIM_SCLK 与 SIM_SIO 走线必须包地保护。

※ SIM 卡接口管脚定义

名称	管脚号	说明
VSIM	26	SIM 卡电源
SIM_SCLK	27	SIM 卡时钟
SIM_SIO	28	SIM 卡数据
SIM_SRST	29	SIM 卡复位
SIM_DET	30	保留，不需要连接

※ SIM 卡接口连接示意图



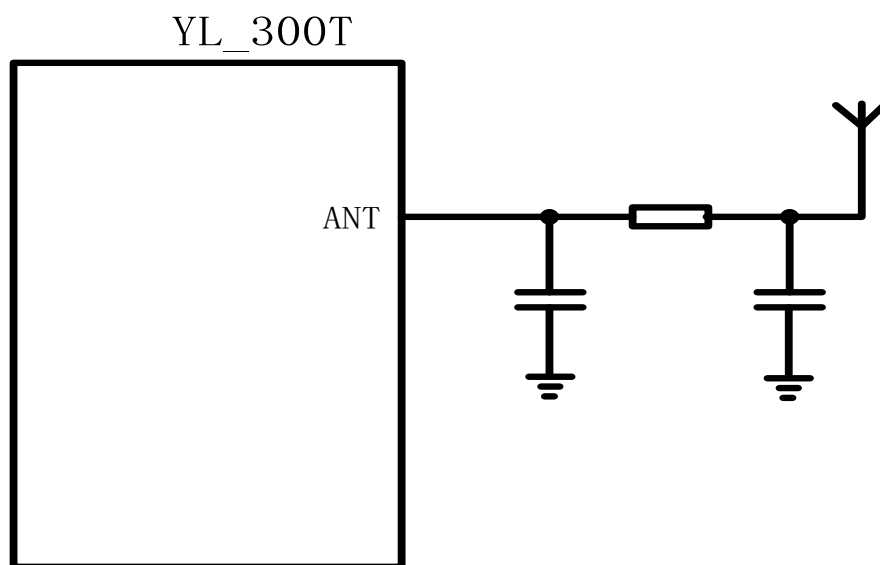
输出天线的设计

对于天线接口的外围电路设计，为了能够更好地调节射频性能，建议预留 π 型匹配电路。射频线的阻抗应该控制在 $50\ \Omega$

※ 天线接口管脚定义

名称	管脚号	说明
ANT	27	RF 信号输入输出
GND	25, 26, 28, 29	RF 地

※ 天线接口参考设计



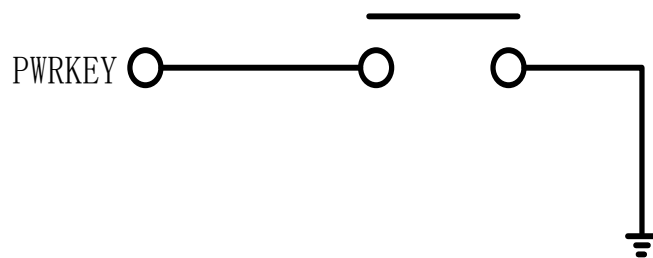
开机键 PWRKEY

PWRKEY 为模块的硬件开机脚。模块上电后，将 PWRKEY 拉低至少 1S 后释放，可以使模块开机。若模块一直拉低不释放，模块将无法进入 PSM 模式。模块内部有上拉电阻，外部不需要上拉。

※ PWRKEY 管脚定义

名称	管脚号	说明
PWRKEY	14	模块硬件开机输入脚

※ PWRKEY 参考设计



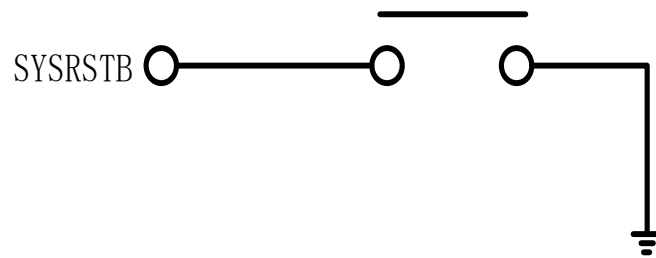
复位键 SYSRSTB

SYSRSTB为模块的硬件复位脚。给SYSRSTB管脚一个持续32mS的低电平可以复位模块。模块内部有上拉电阻，外部不需要上拉。

※ SYSRSTB 管脚定义

名称	管脚号	说明
SYSRSTB	12	模块硬件复位输入脚

※ SYSRSTB 参考设计



RTC_EINT 脚

在 PSM 模式下，给 RTC_EINT 管脚一个持续 10ms 的低电平可以使模块退出 PSM 模式。

※ RTC_EINT 管脚定义

名称	管脚号	说明
RTC_EINT	15	RTC 外部中断输入脚

休眠使能脚 SLEEP_EN

模块 WAKEUP 脚悬空或者上拉至 1.8V，模块允许进入休眠模式。模块 WAKEUP 脚下拉，模块则不会进入休眠模式。

※ SLEEP_EN 管脚定义

名称	管脚号	说明
SLEEP_EN	11	模块休眠使能脚

模数转换脚 ADC0

模块提供一路 ADC，可以用来检测需要 ADC 的外设。其最大检测电压为 1.4V，精度为 12bit。

※ ADC0 管脚定义

名称	管脚号	说明
ADC0	37	模数转换脚

AT 命令介绍

本节详细介绍 AT 命令接口支持的标准和专有 AT 命令。

ATCI 支持以下规范：

- ITU V. 250
- 3GPP TS 27.007 发布 14
- 3GPP TS 27.005 发布 14

注：对于大多数 AT 命令，ATCI 软件都符合标准，但是在某些情况下可能不符合标准，由于协议栈的限制，与这些标准稍有偏差。

在下面的 AT 命令表中，每个 AT 命令都有一个特定于每个通道的 27.010 MUX 的作用域命令语法定义。对于一些 AT 命令，一些参数是可选的。在这种情况下，它们被指定为：AT+CRLP=[<iws>[, <mws>[, <T1>[, <N2>[, <ver>[, <T4>]]]]]]，所有参数都是可选的。如果遗漏了一个可选参数，那么逗号必须存在，如果输入其他可选参数，则插入。例如：

AT+CRLP=61,61,,,1,3，如果没有进一步输入可选参数，则不需要逗号。例如：

AT+CRLP=61,61

注：此命令仅作为示例给出

常用的 AT 命令列表

AT 命令	AT 命令描述
ATI	显示产品的表示信息
AT+GSN	显示唯一序列号标识 IMEI
AT+EREG	读取或写入 NVDM 上的数据
AT+EWD T	触发 WDT 重启系统
AT+ERTC	从 RTC 寄存器读取或写入数据
AT+EGPIO	询问和改变 GPIO 的配置
AT+SM	控制浅睡/深睡模式
AT+ERORT	应用程序的显示和修改串口分配
AT+EVERINFO	询问 SDK 的版本信息
AT+SYSTEM	询问任务和内存信息
AT+UTILIZATION	CPU 利用率信息分析
AT+EGACT	CPU 利用率信息分析
AT+EGACT	激活或者关闭指定的 PDN 环境

AT+ESOC	创建一个 TCP 或者 UDP 套接字
AT+ESOB	设置 TCP 或者 UDP 本地端口
AT+ESOCON	建立 TCP 或者 UDP 连接
AT+ESOSEND	发送 TCP 或者 UDP 数据
AT+ESOCL	关闭连接
AT+ESONMI	表示从网络接收了一些的数据
AT+ESOERR	表示有一些错误

常用的 AT 命令详细描述

※ AT+EREG

描述：读取寄存器数据或者将数据写入寄存器

格式：

命令	可能得到的回复
+EREG=<op>,<address>[,<value>]	[+EREG:<address_r>,<value_r>] OK/ERROR

参数说明：

<op>: 整型
 0 read register
 1 write register
 <address>:地址是十六进制，加前缀 0x
 <value>:寄存器数据（4 个字节）是十六进制，加前缀 0x

举例：

```
AT+EREG=1,0x12345678,0xFEDCBA98 // 将 0xFEDCBA98 写入到寄存器地址 0x12345678
OK
AT+EREG=0,0x12345678 //读取寄存器地址为 0x12345678 的数据
+EREG:0x12345678,0xFEDCBA98 //读到寄存器地址为 0x12345678 的数据 0xFEDCBA98
OK
```

※ AT+EWDT

描述：触发 WDT 重启系统

格式：

命令	可能得到的回复
+EWDT=<OP>	操作失败返回：ERROR 操作成功：系统复位

参数说明:

字段	
<OP>: 整型	
1	马上触发 WDT 重启
其他值	无效, 返回 “ERROR”

举例:

```
AT+EWDT=1 //触发 WDT 重启, 系统会马上重启
```

※ AT+ERTC

描述: 从 RTC 寄存器读取或写入数据; 用户应该知道软件负载中的 RTC 操作 (如向寄存器写入数据流)。然后用户可以使用这个 AT 命令和正确的信息来访问 RTC 模块

格式:

命令	可能得到的回复
+ERTCREG=<OP><Register_address>	返回: OK/ERROR
+ERTC=<alarm_value>	返回: OK/ERROR
+ERTC32K=	RTC 32K 时钟频率值 返回: OK/ERROR

参数说明:

<OP>: 整型
0 读取寄存器操作
1 写入寄存器操作
<Register_address>: 整型
RTC 寄存器偏移量, 从 0x 0 到 0x9c
<alarm_value>: 整型
用于表示 RTC 警报将发生多少秒之后

举例:

```
AT+ERTCREG=0,74 //从 RTC 寄存器读取数值, 偏移量为 0x74
+ERTCREG:0x74,0x200 //从偏移量 0x74 读取的数值为 0x200
OK
AT+ERTCREG=1,74,1 //写入 0x1 这个数据到偏移量为 0x74 的寄存器中
OK
AT+ERTC=3 //RTC 警报将在三秒后发生
OK
```

```
AT+ERTC32K=          //读取 RTC32K 的时钟频率
32767                //读取时钟频率为 32767Hz
OK
```

※ AT+EGPIO

描述：询问和改变 GPIO 的配置

格式：

命令	可能得到的回复
+EGPIO=?	返回：OK/ERROR
+EGPIO=<op>:...	ATCI_CMD_MODE_EXECUTION OK 返回：OK/ERROR

参数说明：

```
GPIO_SET_MODE: 将某个 IO 进入设置模式
GPIO_SET_DIR:  设置管脚的方向
GPIO_SET_PULL: 设置管脚上拉
GPIO_SET_OD:  设置管脚的高低电平(0 或 1)
```

举例：

```
AT_EGPIO=GPIO_GET:5~10 //获取 GPIO_5~GPIO_10 的状态；0x0d, 0x0a 位于字符串尾部
ATCI_CMD_MODE_EXECUTION OK
OK
AT+EGPIO=GPIO_SET_OD:11,1 //设置 I011 为输出数据高
ATCI_CMD_MODE_EXECUTION OK
OK
```

※ AT+SM

描述：控制浅睡/深睡模式，询问睡眠模式

格式：

命令	可能得到的回复
+SM=<op>	返回：OK/ERROR
+SM=SHOWINFO, <show>, <freq>	返回：OK/ERROR

参数说明:

<op>: 字符串, 取决于下表中的参数

<op>	描述
STATUS	显示全部的当前的睡眠锁状态
LOCK	锁住系统不进入休眠, 掉电无效
UNLOCK	释放临时锁, 系统休眠, 掉电无效
LOCK_FOREVER	锁住系统不进入休眠, 掉电有效
UNLOCK_FOREVER	释放系统锁, 释放后马上休眠, 掉电有效

举例:

```

AT+SM=STATUS //询问睡眠状态和锁信息, 日志将会显示在 GKI 日志中
OK
AT+SM=LOCK //要求一个临时的锁去保护系统进入睡眠
OK
AT+SM=UNLOCK // 释放 AT+SM=LOCK 这条命令要求的那个临时的锁
OK
AT+SM=LOCK_FOREVER //要求一个锁去保护系统进入睡眠。这个锁会被写入 NVDM, 所以即使是
重启也依然有效
OK
AT+SM=UNLOCK_FOREVER //释放 AT+SM=LOCK_FOREVER 这条命令要求的那个锁
OK
AT+SM=SHOWINFO, 1, 1000 //当空闲任务被调度 1000 次之后会自动显示锁的状态
OK
AT+SM=SHOWINFO, 0, 0 //关闭锁状态显示
OK
    
```

※ AT+ERORT

描述: 修改应用程序串口

格式:

命令	可能得到的回复
+EPORT=<op>[, <param1...>]	返回: OK/ERROR
+EPORT=0	返回: OK/ERROR
+EPORT=1, <owner_name>, <port_id>	返回: OK/ERROR

+EPORT=2, <owner_name>, <port_id>	返回: OK/ERROR
+EPORT=3, <port_id>, <baudrate>	返回: OK/ERROR
+EPORT=4	返回: OK/ERROR

参数说明:

<op>: 整型

- 0: 显示应用程序或者用户的端口分配
- 1: 对特定的应用程序或者用户重新分配串口并写入 NVDM
- 2: 对特定的应用程序或者用户切换原先的串口到指定的串口
- 4: 修改串口服务设备的参数
- 5: 显示串口服务设备的参数

<param1...>: AT 命令中更多的参数

如果<op>是 1 或者 2, <param1...>在 AT 命令中是
<owner_name>, <port_id>

<owner_name>: 字符串 使用串口服务的应用程序名称, 例如, "uls" 表示 HSL 日志模块, "emmi" 表示 genie 日志模块, "conn1" 表示 COM 串口的第一个调制解调器, "conn12" 表示 COM 串口的第二个调制解调器

<port_id>: 整型, 串口服务支持的指定设备。参考结构—serial_port_type_t

例如: AT+EPORT=1, <owner_name>, <port_id>

AT+EPORT=2, <owner_name>, <port_id>

如果<op>是 3, <param1...>在 AT 命令中是<port_id>, <baudrate>

<baudrate>: 整型, 串口指定支持的指定波特率, 参考结构—hal_uart_baudrate_t

例如 +EPORT=3, <port_id>, <baudrate>

<param2...>: AT 命令中更多的参数

如果<op>是 0, <param2...>在 AT 命令中是<owner_name>=<port_id>

<owner_name>: 字符串, 使用串口服务的应用程序名称。例如, "uls" 表示 HSL 日志模块, "emmi" 表示 genie 日志模块, "conn1" 表示 COM 串口的第一个调制解调器, "conn12" 表示 COM 串口的第二个调制解调器。

<port_id>: 整型, 串口服务支持的指定设备。参考结构—serial_port_type_t

例如 +EPORT:<owner_name>=<port_id> \r\n OK

如果<op>是 4, <param2...>在 AT 命令中是<port_id> \r\n
baudrate=<baudrate>

<port_id>: 整型, 串口服务支持的指定设备。参考结构—serial_port_type_t

<baudrate>: 整型, 串口服务支持的指定波特率。参考结构—hal_uart_baudrate_t
例如 +EPORT:<port_id> \r\n baudrate=<baudrate> \r\n OK

举例:

```
AT+EPORT=0 //显示应用程序或者用户的串口分配
+EPORT:uls=2
+EPORT:emmi=1
OK
AT+EPORT=1,uls,3 //对 ULS 重新分配串口并写入 NVDM
OK
AT+EPORT=2,uls,2 //对 ULS 重新分配串口但不写入 NVDM
OK
AT+EPORT=3,0,12 //设置波特率到 921600bps
OK
AT+EPORT=4 //显示存储在 NVDM 中最新的参数
+EPORT:0
baudrate=12
+EPORT:1
baudrate=12
+EPORT:2
baudrate=12
+EPORT:3
baudrate=9
+EPORT:4
none
+EPORT:5
none
OK
```

※ AT+EVERINFO

描述: 询问 SDK 的版本信息

格式:

命令	可能得到的回复
+EVERINFO=<type>	+EVERINFO:<version info.> OK

参数说明:

<type>: 整型
 0—SDK 版本
 如果想要询问更多信息 (HW 信息, 创建时间...), 可以使用这条命令
 <version info>字符串

举例:

询问 SDK 版本
 AT+EVERINFO=0
 +EVERINFO:SDK_VER_IOT_SDK_DEMO
 OK

※ AT+SYSTEM

描述: 询问任务和内存信息

格式:

命令	可能得到的回复
+SYSTEM=<module>	+SYSTEM:<system info.> OK

参数说明:

<module>: 字符串
 task 显示所有的 freeRTOS 的任务信息
 mem 显示堆栈状态
 crash 强制触发系统崩溃以转储系统信息

举例:

AT+SYSTEM=task //询问任务信息
 参数含义
 1: pcTaskName
 2: cStatus (Ready/Blocked/Suspended/Deleted)
 3: uxCurrentPriority
 4. usStackHighWaterMark
 5. xTaskNumber
 OK

※ AT+UTILIZATION

描述：CPU 利用率信息分析

格式：

命令	可能得到的回复
+UTILIZATION=<op> [, <param>]	+UTILIZATION:<profiling info.> OK

参数说明：

<op>: 字符串
 start 开始 CPU 利用率性能分析
 stop 停止 CPU 利用率性能分析
 在特定的持续时间内执行<param>的 CPU 利用率分析
 <param>: 整型 分析持续时间

举例：

```
AT+UTILIZATION=start
+UTILIZATION:
cpu utilization profiling begin, please use AT+UTILIZATION=stop to end profiling...
OK

AT+UTILIZATION=stop
+UTILIZATION:
parameter meaning :
1:PcTaskName
2:count (unit is 32 k of gpt)
3:ratio
OK
```

※ AT+EGACT

描述：激活或者关闭指定的 PDN 环境

格式：

命令	可能得到的回复
+EGACT=<op>, <pdp_type/cid>[, <apn> , <user_name>, <pwd>[, <bearer_type> [, <sim_id>]]]	+EGACT=<cid>, <type>, <result>[, <activate d_pdp_type>] OK /

	+EGACT=<cid> OK/ ERROR
	+EGACT=<cid>, <type>, <result>, [, activate d_pdp_type>]
+EGACT?	Not Support
+EGACT=?	Not Support

参数说明:

< op >: 整型
 0 失活需求
 1 激活需求

< pdp_type/ cid >: 整型
 如果<op>为 0, 则为 pdp_type, 否则它是 cid
 pdp_type: 激活的 pdp_type。
 1 IPv44
 2 IPV6
 3 IPy4v6
 4 Non-ip
 cid: 它是指定特定 PDP 环境的数字参数。这里它应该等于激活响应返回的<cid>。

< apn >: 字符串
 它是访问点名称, 对于激活需求是必需的, 对于失活需求应该省略。

<user_name>: 字符串
 它是访问 IP 网络的用户名, 该网络是激活需求的强制要求, 并且应该忽略掉激活需求。

< pwd >: 字符串
 它是访问 IP 网络的密码, 对于激活需求是必需的, 对于失活需求应该省略。

<bearer_type>: 整型
 需要激活的载体类型是激活需求的可选类型, 对于失活的需要应该忽略。
 1 NBIOT(目前只支持 NBIOT)

< sim_id >: 整型
 这是想要使用的 SIM 卡的 id, 对于激活需求是可选的, 对于失活需求

<p>应该省略。</p> <p>1 SIM卡1(目前只支持SIM卡1)</p> <p>< cid >:整型 它是指定特定的PDP环境一个数值参数</p> <p><type>:整型</p> <table border="0"> <tr> <td>0</td> <td>Result/URC for deactivation requirements</td> <td>失活请求的结果或者 URC</td> </tr> <tr> <td>1</td> <td>Result/URC for activation requirements</td> <td>激活请求的结果或者 URC</td> </tr> <tr> <td>2</td> <td>URC for passive deactivate</td> <td>被动失活的 URC</td> </tr> </table> <p><result>:整型</p> <table border="0"> <tr> <td>0</td> <td>失败</td> </tr> <tr> <td>1</td> <td>成功</td> </tr> </table> <p>< activated_pdp_type >:整型 它是实际激活的 pdp_type</p> <table border="0"> <tr> <td>1</td> <td>IPv44</td> </tr> <tr> <td>2</td> <td>IPV6</td> </tr> <tr> <td>3</td> <td>IPy4v6</td> </tr> <tr> <td>4</td> <td>Non-ip</td> </tr> </table>	0	Result/URC for deactivation requirements	失活请求的结果或者 URC	1	Result/URC for activation requirements	激活请求的结果或者 URC	2	URC for passive deactivate	被动失活的 URC	0	失败	1	成功	1	IPv44	2	IPV6	3	IPy4v6	4	Non-ip
0	Result/URC for deactivation requirements	失活请求的结果或者 URC																			
1	Result/URC for activation requirements	激活请求的结果或者 URC																			
2	URC for passive deactivate	被动失活的 URC																			
0	失败																				
1	成功																				
1	IPv44																				
2	IPV6																				
3	IPy4v6																				
4	Non-ip																				

举例:

```
+EGACT=1,1," apn_example", " username_example", "password_example" //激活 PDN 环境+EGACT=1
//如果没有错误返回 cid
OK //没有错误可以立即返回 OK
+EGACT=1,1,1,1 //通过 URC 通知激活结果
AT+EGACT=0,1 //失活 PDN 环境
+ EGACT=1 //如果没有错误返回 cid
OK //没有错误立即返回 OK
+EGACT=1,0,1 //通过 URC 通知失活结果
```

注: 这个命令是临时的, 可能在以后的版本中更改。

※AT+ESOC

描述: 这条命令创建一个 TCP 或者 UDP 套接字

格式:

命令	可能得到的回复
+ESOC=<domain>,<type>,<protocol>[,cid]	+ESOC=<socket_id> OK/ERROR

+ESOC?	Not Support
+ESOC=?	Not Support

参数说明:

<p><domain>整型</p> <p>1-IPy4</p> <p>2-IPv6</p> <p><type> 整型</p> <p>1-TCP</p> <p>2-UDP</p> <p>3-RAW</p> <p><protocol>整型</p> <p>1-IP</p> <p>2-ICMP</p> <p>3-UDP_LITE</p> <p><cid>整型, PDP 环境 ID, AT+EGACT 的回复[可选]</p>

※ AT+ESOB

描述: 这条命令用来连接本地地址和本地串口。主要用来设置本地端口

格式:

命令	可能得到的回复
+ESOB=<socket_id>,<local_port>[,<local_address>]	OK/ERROR
+ESOB?	Not Support
+ESOB=?	Not Support

参数说明:

<p><socket_id> 整型, 套接字 ID, AT+ESOC 的回复</p> <p><local_port> 整型, 本地串口</p> <p><local_address>字符串, 本地地址[可选]</p>

※AT+ESOCON

描述: 此命令用于将套接字连接到远程地址和端口; 如果套接字是 TCP, 则开始发送 TCP SYN 数据包; 如果套接字是 UDP, 则不会发送数据包。

格式:

命令	可能得到的回复
+ESOCON=<socket_id>,<remote_port> ,<remote_address>	OK ERROR
+ESOCON?	Not Support
+ESOCON=?	Not Support

参数说明:

<socket_id> 整型，套接字 ID, AT+ESOC 的回复
 <remote_port> 整型, 远程端口
 <remote_address> 字符串, 远程地址

※AT+ESOSEND

描述: 这条命令用于向网络发送数据; 回复是“OK”意味着 ACTMD 格式正确, 数据已被放入套接字中, 等待发送

格式:

命令	可能得到的回复
+ESOSEND=<socket_id>,<data_len>,<data>[,<flag>]	OK ERROR
+ESOSEND?	Not Support
+ESOSEND=?	Not Support

参数说明:

<socket_id> 整型 套接字 ID, AT+ESOC 的回复
 <data_len>整型，数据的长度
 <data>raw_data, 数据内容
 <flag>整型, 发送标志
 1—ACK 无延迟
 2—no nagle

※ AT+ESOCL

描述：这条命令用来关闭连接

格式：

命令	可能得到的回复
+ESOCL=<socket_id>	OK ERROR
+ESOCL?	Not Support
+ESOCL=?	Not Support

参数说明：

<socket_id>:整型，套接字 ID，AT+ESOC 的回复

※ AT+ESONMI

描述：表示从网络接收了一些的数据

格式：

命令	可能受到的回复
	+ESOCL=<socket_id>,<data_len>,<data>

参数说明：

<socket_id>:整型，套接字，AT+ESOC 回复

<data_len> 整型 数据的长度

<data> 原始数据

※ AT+ESOERR

描述：表示有一些错误

格式：

命令	可能得到的回复
	+ESOERR=<socket_id>,<error_code>

参数说明:

<socket_id>整型, 套接字 ID, AT+ESOC 的回复
<error_code>整型, 错误编码
1—通过 peer point 重启
2—网络不连接

TCP 和 UDP 应用举例

※ 创建一个 TCP 连接示例

AT+EGACT=1,1 "apn", "user_name", "pwd" ————激活 apn
+EGACT=<cid>, 1, 1, 1
OK
AT+ESOC=1, 1, 1 ————创建套接字
+ESOC=0
OK
AT+ESOCON=0, 1026, " 172. 26. 109. 69" ————连接套接字
OK
AT+ESOSEND=0, 10, 1234562112 ————发送数据
OK
+ESONMI=0, 3, 3BX ————接收数据
AT+ESODIS=0 ————不连接套接字
OK
AT+ESOCL=0 ————关闭套接字
OK

※创建一个 UDP 连接示例

AT+EGACT=1,1,"apn","user_name","pwd" —— 激活 apn
+EGACT=<cid>,1,1,1
OK
AT+ESOC=1,2,1 —— 创建套接字
+ESOC=0
OK
AT+ESOCON=0,1026,"172.26.109.69" —— 连接套接字
OK
AT+ESOSEND=0,10,1234562112 —— 发送数据
OK
+ESODIS=0,3,3BX
AT+ESODIS=0 —— 不连接套接字
OK
AT+ESOCL=0 —— 关闭套接字
OK