

AN1809 ATK-SIM800C GSM/GPRS 模块使用说明

本应用文档(AN1809,对应 **NANO STM32F1 开发板扩展实验 8**)将教大家如何在 ALIENTEK STM32 开发板上使用 ATK-SIM800C GSM/GPRS 模块。

本文档分为如下几部分：

- 1, ATK-SIM800C GSM/GPRS 模块简介
- 2, 硬件连接
- 3, 软件实现
- 4, 验证

1、ATK-SIM800C GSM/GPRS 模块简介

ATK-SIM800C-V15(V15 是版本号, 下面均以 ATK-SIM800C 表示该产品)是 ALIENTEK 推出的一款高性能工业级 GSM/GPRS 模块(开发板)。ATK-SIM800C 模块板载 SIMCOM 公司的工业级四频 GSM/GPRS 模块: SIM800C, 工作频段四频: 850/900/1800/1900MHz, 可以低功耗实现语音、SMS(短信)、MMS(彩信)、蓝牙数据信息的传输。

ATK-SIM800C 模块支持 RS232 串口和 LVTTTL 串口(即支持 3.3V/5V 系统), 并带硬件流控制, 支持 5V~24V 的超宽工作范围, 使得本模块可以非常方便的与您的产品进行连接, 从而给您的产品提供包括语音、短信、彩信、蓝牙和 GPRS 数据传输等功能。

1.1 模块资源简介

ATK-SIM800C 模块是 ALIENTEK 开发的一款高性能工业级 GSM/GPRS 模块(开发板), 功能完善, 尤其适用于需要语音/短信/GPRS 数据/蓝牙通信服务的各种领域, 其资源图如图 1.1.1 所示:

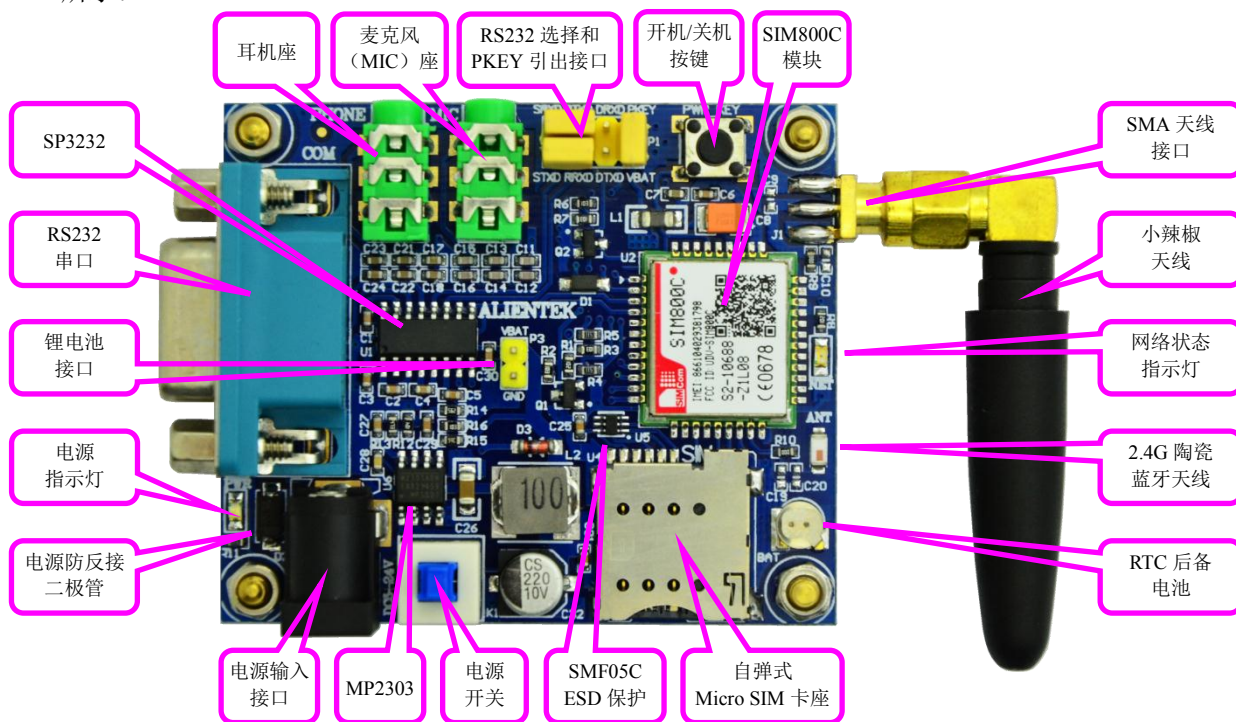


图 1.1.1 ATK-SIM800C 模块资源图

从图 1.1.1 可以看出, ATK-SIM800C 模块不但外观漂亮, 而且功能齐全, 模块尺寸(不

算天线部分)为 62mm*52.5mm,并带有安装孔位,非常小巧,并且利于安装,可方便应用于各种产品设计。

ALIENTEK ATK-SIM800C 模块(开发板)板载资源如下:

- ◆ GSM 模块: SIM800C
- ◆ 1 个 RTC 后备电池
- ◆ 1 个麦克风接口
- ◆ 1 个耳机接口
- ◆ 1 个 RS232 选择和 PKEY 引出接口
- ◆ 1 个 RS232 串口
- ◆ 1 个锂电池接口
- ◆ 1 个电源输入接口
- ◆ 1 个电源指示灯(蓝色)
- ◆ 1 个电源开关
- ◆ 1 个自弹式 Micro SIM 卡座
- ◆ 1 个 SMA 天线接口并配套小辣椒天线
- ◆ 1 个 2.4G 蓝牙陶瓷天线
- ◆ 1 个开机/关机按键
- ◆ 1 个网络状态指示灯(红色)

ATK-SIM800C 模块(开发板)采用工业级标准设计,特点包括:

- 板载 RS232 串口(支持硬件流控制),方便与 PC/工控机等设备连接;
- 板载 3.5mm 耳机和麦克风座,方便进行语音通信开发;
- 板载高效同步降压电路,转换效率高达 90%,支持超宽电压工作范围(5~24V),非常适合工业应用;
- 板载电源防反接保护, SIM 卡 ESD 保护,保护功能完善;
- 板载 RTC 后备电池(XH414H-IV01E),无需担心掉电问题;
- 板载小辣椒天线和陶瓷天线,能有效提高信号接收能力;
- 采用国际 A 级 PCB 料,沉金工艺加工,稳定可靠;
- 采用全新元器件加工,纯铜镀金排针,坚固耐用;
- 人性化设计,各个接口都有丝印标注,使用起来一目了然;接口位置设计安排合理,方便顺手。
- PCB 尺寸为 62mm*52.5mm,并带有安装孔位,小巧精致;

ATK-SIM800C 模块的资源介绍,我们就介绍到这里,详细的介绍,请看《ATK-SIM800C GSM(GPRS)模块用户手册_V1.0》相关章节。

1.2 模块使用

本文档,我们将介绍大家如何通过 ALIENTEK NANO STM32 开发板连接 ATK-SIM800C 模块,实现:电话通信(拨打、接听)、短信通信(英文短信发送、中英文短信发送、中英文短信接收)、GPRS 通信(TCP 和 UDP 通信)、蓝牙通信(SPP 通信)、以及 TTS 文本转语音。本节我们将介绍要实现这些功能所需要的相关知识。

1.2.1 AT 指令简介

AT 即 Attention, AT 指令集是从终端设备(Terminal Equipment, TE)或数据终端设备(Data Terminal Equipment, DTE)向终端适配器(Terminal Adapter, TA)或数据电路终端设备(Data Circuit Terminal Equipment, DCE)发送的。通过 TA, TE 发送 AT 指令来控制移动台(Mobile Station, MS)的功能,与 GSM 网络业务进行交互。用户可以通过 AT 指令进行呼叫、短信、电话本、

数据业务、传真等方面的控制。

AT 指令必须以"AT"或"at"开头，以回车 (<CR>) 结尾。模块的响应通常紧随其后，格式为：<回车><换行><响应内容><回车><换行>。

我们通过串口调试助手 XCOM 来测试一下，打开：ATK-SIM800C 模块配套资料\3，配套软件\串口调试助手\XCOM V2.0.exe，选择正确的 COM 号（连接到 ATK-SIM800C 模块的 COM 端口，我电脑是 COM8），然后设置波特率为 115200，**勾选发送新行（必选！即 XCOM 自动添加发送回车换行功能）**，然后发送 AT 到 ATK-SIM800C 模块，如图 1.2.1.1 所示：



图 1.2.1.1 AT 指令测试

图 1.2.1.1 中，我们发送了 3 次 AT 指令，第一次看到有乱码，这是因为模块上电后，还没有实现串口同步，在收到第一次数据（不一定要 AT 指令）后，模块会自动实现串口同步（即自动识别出了通信波特率），后续通信就不会出现乱码了。因为 SIM800C 具有自动串口波特率识别功能（识别范围：1200-115200），所以我们的电脑（或设备）可以随便选择一个波特率（不超过识别范围即可），来和模块进行通信，这里我们选择最快的 115200。

从图 1.2.1.1 可以看出，我们现在已经和 SIM800C 模块进行通信了，我们通过发送不同的 AT 指令，就可以实现对 SIM800C 的各种控制了。

SIM800C 模块提供的 AT 命令包含符合 3GPP TS 27.005、3GPP TS 27.007 和 ITU-T Recommendation V.25ter 的指令，以及 SIMCOM 自己开发的指令。接下来我们介绍几个常用的 AT 指令：

1, AT+CPIN?

该指令用于查询 SIM 卡的状态，主要是 PIN 码，如果该指令返回：+CPIN:READY,则表明 SIM 卡状态正常，返回其他值，则有可能是没有 SIM 卡。

2, AT+CSQ

该指令用于查询信号质量，返回 SIM800C 模块的接收信号强度，如返回：+CSQ: 24,0，

表示信号强度是 24（最大的有效值是 31）。如果信号强度过低，则要检查天线是否接好了？

3, AT+COPS?

该指令用于查询当前运营商，该指令只有在连上网络后，才返回运营商，否则返回空，如返回：+COPS:0,0,"CHINA MOBILE",表示当前选择的运营商是中国移动。

4, AT+CGMI

该指令用于查询模块制造商，如返回：SIMCOM_Ltd，说明 SIM800C 模块是 SIMCOM 公司生产的。

5, AT+CGMM

该指令用于查询模块型号，如返回：SIMCOM_SIM800C，说明模块型号是 SIM800C。

6, AT+CGSN

该指令用于查询产品序列号（集 IMEI 号），每个模块的 IMEI 号都是不一样的，具有全球唯一性，如返回：866104023267696，说明模块的产品序列号是：866104023267696。

7, AT+CNUM

该指令用于查询本机号码，必须在 SIM 卡在位的时候才可以查询，如返回：+CNUM:"","136*****","129",7,4，则表明本机号码为：136*****。另外，不是所有的 SIM 卡都支持这个指令，有个别 SIM 卡无法通过此指令得到其号码。

8, ATE1

该指令用于设置回显模式（默认开启），即模块将收到的 AT 指令完整的返回给发送端，启用该功能，有利于调试模块。如果不需要开启回显模式，则发送 ATE0 指令即可关闭，这样收到的指令将不再返回给发送端，这样方便程序控制。

9, AT+CGMR

该指令用于查询固件版本序列号，如返回：Revision:1418B02SIM800C32_BT,说明模块的固件版本序列号是 1418B02SIM800C32_BT，flash 大小是 32Mbit、支持蓝牙通信功能。

以上就是我们介绍的几个常用的 AT 指令，当然还有其他一些常用的 AT 指令，比如 ATD/ATA/ATH 等，我们在后面的章节会慢慢介绍。关于 SIM800C 详细的 AT 指令介绍，请参考：[ATK-SIM800C 模块配套资料\4, SIM800C 模块资料\SIM800 Series_AT Command Manual_V1.09.pdf](#) 这个文档。

发送给模块的指令，如果执行成功，则会返回对应信息和“OK”，如果执行失败/指令无效，则会返回“ERROR”。

1.2.2 拨打/接听电话

使用 ATK-SIM800C 模块可以非常方便的进行拨打和接听电话。实现拨号和接听电话，常用的指令有：ATE0/ATD/ATA/ATH/AT+CLIP/AT+VTS 等 6 条 AT 指令。

ATE0，用于关闭回显，在通过电脑串口调试助手调试的时候，我们发送：ATE1，开启回显，可以方便调试，但是我们通过单片机程序控制的时候，用不到回显功能，所以发送：ATE0，将其关闭。

ATD，用于拨打任意电话号码，格式为：ATD+号码+；，末尾的‘；’一定要加上，否则不能成功拨号，如发送：ATD10086；，即可实现拨打 10086。

ATA，用于应答电话，当收到来电的时候，给模块发送：ATA，即可接听来电。

ATH，用于挂断电话，要想结束正在进行的通话，只需给模块发送：ATH，即可挂断。

AT+COLP，用于设置被叫号码显示，这里我们通过发送：AT+COLP=1，开启被叫号码显示，当成功拨通的时候（被叫接听电话），模块会返回被叫号码。

AT+CLIP，用于设置来电显示，通过发送：AT+CLIP=1，可以实现设置来电显示功能，模块接收到来电的时候，会返回来电号码。

AT+VTS, 产生 DTMF 音, 该指令只有在通话进行中才有效, 用于向对方发送 DTMF 音, 比如在拨打 10086 查询的时候, 我们可以通过发送: AT+VTS=1, 模拟发送按键 1。

以上就是在拨打/接听电话时经常用到的几条指令, 通过这几条指令, 就可以实现电话的拨打和接听了, 不过首先要保证模块成功接入到 GSM 网络, 通过发送: AT+COPS?, 如果返回: +COPS: 0,0,"CHINA MOBILE", 则说明模块成功连接到了 GSM 网络, 可以正常使用了, 网络运营商为"CHINA MOBILE"(中国移动)。这些指令的使用示例可以参考《ATK-SIM800C GSM (GPRS) 模块用户手册_V1.0》2.3.3 节。

1.2.3 短信的读取与发送

使用 ATK-SIM800C 模块, 我们可以很方便的进行中英文短信的读取与发送。短信的读取与发送将用到的指令有: AT+CNMI/ AT+CMGF / AT+CSCS / AT+CSMP / AT+CMGR/AT+CMGS/ AT+CPMS 等 7 条 AT 指令。

AT+CNMI, 用于设置新消息指示。发送: AT+CNMI=2,1, 设置新消息提示, 当收到新消息, 且 SIM 卡未满的时候, SIM800C 模块会通过串口输出数据, 如: +CMTI: "SM",2, 表示收到接收到新消息, 存储在 SIM 卡的位置 2。

AT+CMGF, 用于设置短消息模式, SIM800C 支持 PDU 模式和文本 (TEXT) 模式等 2 种模式, 发送: AT+CMGF=1, 即可设置为文本模式。

AT+CSCS, 用于设置 TE 字符集, 默认的为 IRA, 国际标准字符集, 在发送纯英文短信的时候, 发送: AT+CSCS="GSM", 设置为缺省字符集即可。在发送中英文短信的时候, 需要发送: AT+CSCS="UCS2", 设置为 16 位通用 8 字节倍数编码字符集。

AT+CSMP, 用于设置短消息文本模式参数, 在使用 UCS2 方式发送中文短信的时候, 需要发送: AT+CSMP=17,167,2,25, 设置文本模式参数。

AT+CMGR, 用于读取短信, 比如发送: AT+CMGR=1, 则可以读取 SIM 卡存储在位置 1 的短信。

AT+CMGS, 用于发送短信, 在"GSM"字符集下, 最大可以发送 180 个字节的英文字符, 在"UCS2"字符集下, 最大可以发送 70 个汉字 (包括字符/数字)。

AT+CPMS, 用于查询/设置优选消息存储器, 通过发送: AT+CPMS?, 可以查询当前 SIM 卡最大支持多少条短信存储, 以及当前存储了多少条短信等信息。如返回: +CPMS:"SM_P",1,50,"SM_P",1,50,"SM_P",1,50, 表示当前 SIM 卡最大存储 50 条信息, 目前已经有 1 条存储的信息。

以上就是短信读取与发送需要用到的一些 AT 指令, 这些指令的使用示例可以参考《ATK-SIM800C GSM(GPRS)模块用户手册_V1.0》2.3.4 节。

为方便实现中英文短信的读取与发送, 本文档例程采用文本模式 (AT+CMGF=1)、UCS2 编码字符集 (AT+CSCS="UCS2"), 这样电话号码和短信内容, 全部是采用 UNICODE 编码的字符串。在读取短信的时候, 需要将模块返回的 UNICODE 编码字符串转换为 GBK/ASCII 码, 以便显示。而在发送短信的时候, 需要将 GBK/ASCII 编码的电话号码和短信内容转换为 UNICODE 编码的字符串, 发送给 ATK-SIM800C 模块, 实现中英文短信的发送。

在《ATK-SIM800C GSM(GPRS)模块用户手册_V1.0》2.3.4 节里面, 我们使用了一个汉字 Unicode 互换工具的软件来实现汉字和 UNICODE 的互换。

1.2.4 GPRS 通信

ATK-SIM800C 模块内嵌了 TCP/IP 协议, 通过该模块, 我们可以很方便的进行 GPRS 数据通信。本文档例程我们将实现模块与电脑的 TCP 和 UDP 数据传输。将要用到的指令有: AT+CGCLASS/AT+CGDCONT/ AT+CGATT/AT+CIPCSGP/AT+CIPHEAD /AT+CLPORT/AT+CIPSTART/ AT+CIPSEN/AT+CIPSTATUS/AT+CIPCLOSE/AT+CIPSHUT 等 11 条 AT 指令。

AT+CGCLASS, 用于设置移动台类别。SIM800C 模块支持类别"B"、"CG"和"CC", 发送: AT+CGCLASS="B", 设置移动台类别为 B。即, 模块支持包交换和电路交换模式, 但不能同时支持。

AT+CGDCONT, 用于设置 PDP 上下文。发送: AT+CGDCONT=1,"IP","CMNET", 设置 PDP 上下文标志为 1, 采用互联网协议 (IP), 接入点为"CMNET"。

AT+CGATT, 用于设置附着和分离 GPRS 业务。发送: AT+CGATT=1, 附着 GPRS 业务。

AT+CIPCSGP, 用于设置 CSD 或 GPRS 链接模式。发送: AT+CIPCSGP=1, "CMNET", 设置为 GPRS 连接, 接入点为"CMNET"。

AT+CIPHEAD, 用于设置接收数据是否显示 IP 头。发送: AT+CIPHEAD=1, 即设置显示 IP 头, 在收到 TCP/UDP 数据的时候, 会在数据之前添加如: +IPD:28, 表示是 TCP/UDP 数据, 数据长度为 28 字节。通过这个头, 可以方便我们在程序上区分数据来源。

AT+CLPORT, 用于设置本地端口号。发送: AT+CLPORT="TCP","8888", 即设置 TCP 连接本地端口号为 8888。

AT+CIPSTART, 用于建立 TCP 连接或注册 UDP 端口号。发送: AT+CIPSTART="TCP","219.137.88.114","8086", 模块将建立一个 TCP 连接, 连接目标地址为: 219.137.88.114, 连接端口为 8086, 连接成功会返回: CONNECT OK。

AT+CIPSEND, 用于发送数据。在连接成功以后发送: AT+CIPSEND, 模块返回: >, 此时可以输入要发送的数据, 最大可以一次发送 1352 字节, 数据输入完后, 同发短信一样, 输入十六进制的: 1A (0X1A), 启动发送数据。在数据发送完成后, 模块返回: SEND OK, 表示发送成功。

AT+CIPSTATUS, 用于查询当前连接状态。发送: AT+CIPSTATUS, 模块即返回当前连接状态。

AT+CIPCLOSE, 用于关闭 TCP/UDP 连接。发送: AT+CIPCLOSE=1, 即可快速关闭当前 TCP/UDP 连接。

AT+CIPSHUT, 用于关闭移动场景。发送: AT+SHUT, 则可以关闭移动场景, 关闭场景后连接状态为: IP INITIAL, 可以通过发送: AT+CIPSTATUS, 查询。另外, 在连接建立后, 如果收到: +PDP: DEACT, 则必须发送: AT+CIPSHUT, 关闭场景后, 才能实现重连。

以上就是 GPRS 通信 (TCP/UDP) 将要用到的一些 AT 指令的简介, 这些指令的使用示例可以参考《ATK-SIM800C GSM(GPRS)模块用户手册_V1.0》2.3.5 节。

另外, 要实现模块与电脑的 GPRS 通信, 需要**确保所用电脑具有公网 IP, 否则无法实现通信**, 推荐在 ADSL 网络下进行测试, 并最好关闭防火墙/杀毒软件。

对于 ADSL 用户 (没用路由器), 直接拥有 1 个公网 IP, 你可以通过百度, 搜索: IP, 第一个条目, 就是本机 IP, 如图 1.2.4.1 所示:



图 1.2.4.1 百度得到的本机公网 IP

该 IP 将与你的电脑 IP（双击本地连接图标→支持选项卡，即可查看）是一致的。

对与使用了路由器的 ADSL 用户，那么电脑 IP 与你百度到的公网 IP 是不一样的，如图 1.2.4.2 所示：



图 1.2.4.2 经过路由器后的电脑 IP

可以看到，我们电脑 IP 为 192.168.1.140，与公网 IP 不一致，此时我们需要对路由器进行一下转发规则设置：登录路由器控制页面，然后选择→LAN 接口配置→DMZ 配置，如图 1.2.4.3 所示：



图 1.2.4.3 DMZ 设置值

然后设置启用 DMZ 主机，并设置 DMZ 主机 IP 地址为所用电脑的 IP 地址，本机 IP 为：192.168.1.140，如图 1.2.4.4 所示：

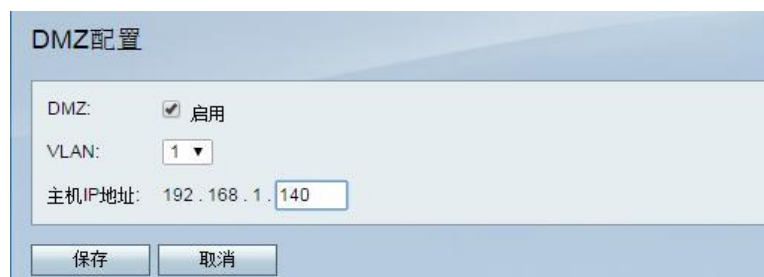


图 1.2.4.4 设置 DMZ 主机

然后保存。这样，我们就把内网 IP（192.168.1.140）映射到了外网，相当于经过路由器的电脑，拥有了一个公网 IP。

最后，我们在电脑上，还需要用到一个软件：网络调试助手，来协助验证 GPRS 通信，该软件启动界面如图 1.2.4.5 所示：



图 1.2.4.5 网络调试助手启动界面

该软件的使用非常简单，我们将在第四节配合我们的例程向大家介绍该软件的使用。

1.2.5 蓝牙通信

ATK-SIM800C 模块集成了蓝牙 3.0，通过该模块，我们可以很方便的进行蓝牙数据通信。本文档例程我们将实现模块与手机蓝牙数据传输。将要用到的指令有：AT+BTPOWER/AT+BTHOST/AT+BTSCAN/AT+BTUNPAIR/AT+BTPAIR/AT+BTACPT/AT+BTSPSEND/AT+BTDISCONN 等 8 条 AT 指令。

AT+BTPOWER,用于设置开启或关闭蓝牙电源，当发送 AT+BTPOWER=1，返回 OK，表示开启蓝牙电源；发送 AT+BTPOWER=0,返回 OK，表示关闭蓝牙电源。

AT+BTHOST，用于查询和设置当前模块蓝牙设备名，当发送 AT+BTHOST? 时，返回该设备的蓝牙名字和地址，设置当前模块蓝牙设备名时，命令格式为 AT+BTHOST=<name>,name 为你要设置的设备名。

AT+BTSCAN,用于设置蓝牙搜索参数，发送 AT+BTSCAN=1,10，开启扫描设备，时间为 10s，搜索到设备返回 例如：+BTSCAN: 0,1,"Meizu MX4 Pro",22:22:5f:b8:e9:af,-79，表示设备 1, 名称：Meizu MX4 Pro，地址：22:22:5f:b8:e9:af，信号：-79。

AT+BTUNPAIR 用于删除蓝牙设备配对信息，发送 AT+BTUNPAIR=0，删除所有已配对的蓝牙设备信息。（**注意：上次配对过的设备，下次进行配对前必须删除配对信息**）

AT+BTPAIR 用于实现蓝牙配对，发送:AT+BTPAIR=0,1，向设备 1 发起配对请求。

AT+BTACPT 用于接收配对的蓝牙设备的连接请求，发送 AT+BTACPT=1，接收连接请求，发送 AT+BTACPT=0，拒绝连接请求。

AT+BTSPSEND 用于蓝牙串口发送数据，发送数据有两种方式，定长与非定长。在连接成功以后发送：AT+CIPSEND，模块返回：>，即非定长模式下，此时可以输入要发送的数据，最大可以一次发送 1024 字节，数据输入完后，同发短信一样，输入十六进制的：1A(0X1A)，启动发送数据。在数据发送完成后，模块返回：SEND OK，表示发送成功。**关于定长模式，请参考文档《ATK-SIM800C 蓝牙功能_AN1603C.pdf，文件路径：ATK-SIM800C 模块资料\ATK-SIM800C 蓝牙功能_AN1603C.pdf**

AT+BTDISCONN 用于断开已连接的蓝牙设备服务，发送 AT+BTDISCONN=1，断开与设备 1 服务的连接。

以上就是蓝牙通信将要用到的一些 AT 指令的简介，这些指令的使用示例可以参考《ATK-SIM800C 蓝牙功能_AN1603C.pdf》或《ATK-SIM800C GSM(GPRS)模块用户手册_V1.0》2.3.6 节。

另外，要实现模块与手机的蓝牙通信，需要在手机端安装一个软件：蓝牙串口助手增强版_Bluetooth_spp_pro,该软件你可以在豌豆荚市场搜索到，也可以在我们提供的 ATK-SIM800C 配套软件资料中找到。该软件启动界面如图 1.2.5.1 所示：

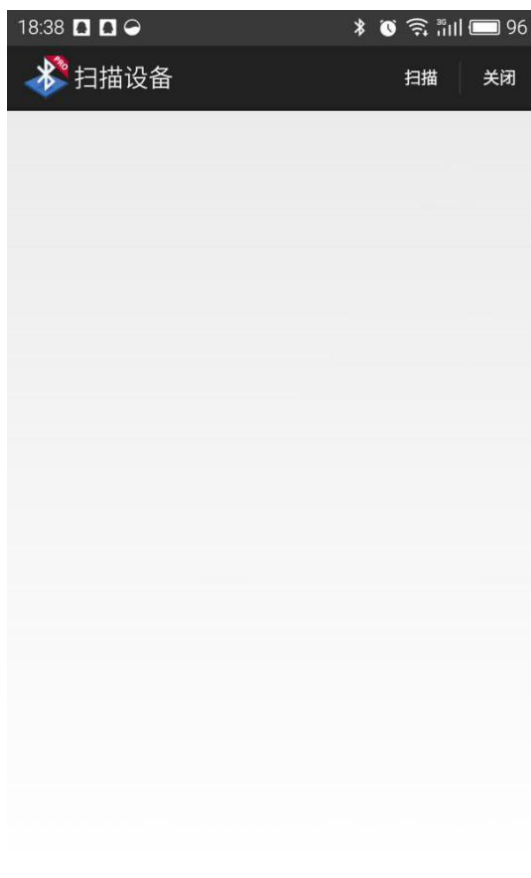


图 1.2.5.1 蓝牙串口助手增强版启动界面

1.2.6 TTS 文本转语音

TTS 文本转语音的使用说明请查看《ATK-SIM800C TTS 功能_AN1603B.pdf》，文件路径：ATK-SIM800C 模块资料\ATK-SIM800C TTS 功能_AN1603B.pdf，在这里我们就不做出介绍了。

2、硬件连接

上面我们分别介绍了拨打/接听电话，短信的读取与发送，GPRS 通信、蓝牙通信、以及 TTS 文本转语音，下面将测试以上的每个功能，例程总分为 10 个，如下：

- 1) 拨打电话
- 2) 接听电话
- 3) 英文短信发送
- 4) 中英文短信发送
- 5) 中英文短信接收
- 6) GPRS TCP 通信
- 7) GPRS UDP 通信
- 8) 蓝牙 被动连接通信
- 9) 蓝牙 主动连接通信
- 10) TTS 文本语音

例程功能讲解：

- 1) 拨打电话：上电后，模块会自动拨通 10086，插上耳机到模块可以听到在通话。
- 2) 接听电话：上电后，用手机拨打电话到插在模块上的手机卡号码，当识别到有来电

时，模块会自动接听来电。

- 3) 英文短信发送：上电后，模块会往例程设定的手机号码发送一条英文短信。
- 4) 中英文短信发送：上电后，模块会往例程设定的手机号码发送一条中英文短信，短信中对方的手机号码和短信内容是 UNICODE 码，UNICODE 码是经过汉字 UNICODE 互换工具软件转换而成的。
- 5) 中英文短信接收：上电后，模块会等待短信的接收，用手机往插在模块上的手机号码发送短信，模块收到短信会内容输出，若短信内容是纯数字、纯英文、或数字和英文的，会以字符串输出，若内容夹带中文，则是输出 UNICODE 码,这时请用汉字 UNICODE 互换工具将短信内容 UNICODE 码转换成中文。另外，当接收到短信内容是“LED1ON”时，开发板 LED1 亮，若短信内容时“LED1OFF”，开发板 LED1 灭。
- 6) GPRS TCP 通信：上电后，模块会往例程设定的 IP 和端口进行 TCP 通信连接，连接成功，对方 IP 往模块发送数据会串口输出，当发送“LED1ON”，开发板 LED1 亮，发送“LED1OFF”时，开发板 LED1 灭。开发板收到模块返回的 TCP 通信数据，会往对方 IP 发送数据应答，若 10 秒内开发板没收到数据，会向服务器发送心跳帧“OK”。
- 7) GPRS UDP 通信：上电后，模块会往例程设定的 IP 和端口进行 UDP 通信连接，连接成功，对方 IP 往模块发送数据会串口输出，当发送“LED1ON”，开发板 LED1 亮，发送“LED1OFF”时，开发板 LED1 灭。开发板收到模块返回的 UDP 通信数据，会往对方 IP 发送数据应答，若 10 秒内开发板没收到数据，会向服务器发送心跳帧“OK”。
- 8) 蓝牙被动连接通信：上电后用带蓝牙的安卓手机开启蓝牙，搜索“sim800c”设备并进行连接。连接成功过后，手机端打开蓝牙串口助手 PRO APP 搜索“sim800c”设备进行 spp 连接，直到连接成功。SPP 连接成功后，开发板以 500ms 时间向手机发送数据，同时手机端发送的数据会串口输出。
- 9) 蓝牙主动连接通信：上电后，会搜索周边的蓝牙设备，当搜索到设备后，会向搜索列表第一个设备发起连接申请（这里需保证搜索列表的第一个设备是你的手机），这时手机端会收到蓝牙配对申请，点击确定进行连接，连接成功后，手机端打开蓝牙串口助手 PRO APP，搜索“sim800c”设备进行 spp 连接，直到连接成功。SPP 连接成功后，开发板以 500ms 时间向手机发送数据，同时手机端发送的数据会串口输出。
- 10) TTS 文本语音：上电后，在串口助手输入需要播放的文本，然后勾选“发送新行”，点击发送，模块会将文本内容转换程语音输出。当模块正在播放语音，在串口助手输入:暂停，然后勾选“发送新行”，点击发送，模块会停止当前的语音播放。

本实验所需的硬件资源如下：

- 1, ALIENTEK STM32 开发板 1 个
- 2, ATK-SIM800C GSM/GPRS 模块一个（包含连接杜邦线 3 跟）
- 3, 直流稳压电源 1 个（推荐 12V 1A 电源）
- 4, 中国移动/联通 GSM SIM 卡一张（未停机，开通 GPRS 业务）
- 5, 耳机一副（带麦克风功能，用于通话测试）
- 6, 一台支持蓝牙的手机设备（安卓系统）

要完成本文档例程的所有功能测试，请大家务必准备好以上硬件，否则有些功能可能无法完成。ATK-SIM800C 所有的控制与数据，都是通过串口来传输的，所以我们的开发板与模块连接，只需要连接串口即可（当然还需要共地）。接下来，我们看看 ALIENTEK NANO STM32 开发板与 ATK-SIM800C 模块的连接方式。

ATK-SIM800C 模块需要把 P1 的两个跳线帽拔了。然后，用 3 根杜邦线，按表 2.1 所示关系连接：

ATK-SIM800C GSM 模块与开发板连接关系			
ATK-SIM800C GSM 模块	GND	STXD	SRXD
NANO V1 开发板	GND	PA3	PA2

表 2.1 ATK-SIM800C 模块同 NANO STM32F103 开发板连接关系表

注意，表中的 GND，大家可以在开发板和 ATK-SIM800C 模块上面，随便找一个 GND 标号的排针，连接在一起即可。

最后，特别提醒：ATK-SIM800C 模块必须由单独的电源供电（推荐 12V1A 电源），开发板通过 USB 插电脑供电，不过切记要共地哦！！

3、软件实现

全部的实验，我们都是以扩展实验 1-HC05 上进行修改，在 HARDWARE 文件里面新建 USART2 文件夹，存放 usart3.c 和 usart3.h 两个文件。并在工程 HARDWARE 组里面添加 uasrt3.c，并添加 usart3 文件夹到头文件包含路径。

在工程目录添加 SIM800C 文件夹，并在工程里面再添加 SIM800C 分组，新建 sim800c 和 sim800c 两个文件，存放再 SIM800C 文件夹内，将 sim800c.c 加入 SIM800C 分组，并添加 SIM800C 文件夹到头文件包含路径。

我们去掉原工程的一些未用到的.c 文件，最终的工程如图 3.1 所示：

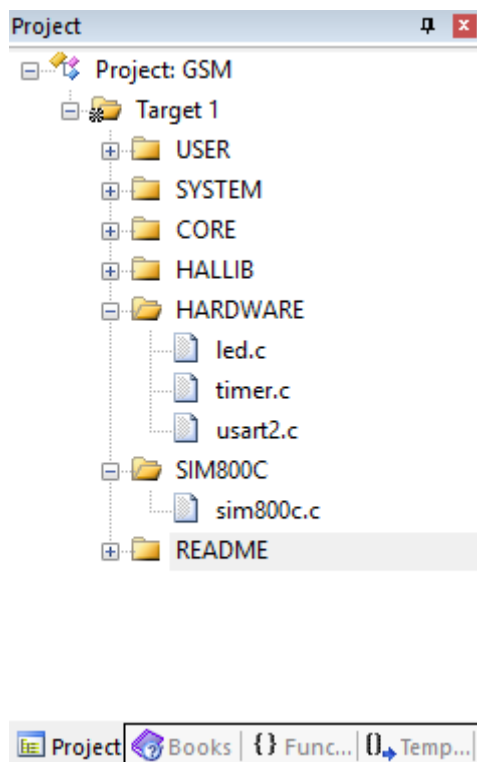


图 3.1 ATK-SIM800C 模块测试实验工程截图

usart2.c 在之前的例程（详见：ATK-HC05 蓝牙串口模块使用说明）已经有介绍过，下面我们将每个例程进行讲解。

1) 拨打电话：

我们主要看 sim800c.c 和 main.c 的代码，首先 sim800c，该文件是 ATK-SIM800C 模块的驱动代码，sim800c 里面的代码如下：

```
//sim800C 发送命令后,检测接收到的应答
//str:期待的应答结果
```

```
//返回值:0,没有得到期待的应答结果
//其他,期待应答结果的位置(str 的位置)
u8* sim800c_check_cmd(u8 *str)
{
    char *strx=0;
    if(USART2_RX_STA&0X8000)    //接收到一次数据了
    {
        USART2_RX_BUF[USART2_RX_STA&0X7FFF]=0;//添加结束符
        strx=strstr((const char*)USART2_RX_BUF,(const char*)str);
    }
    return (u8*)strx;
}

//向 sim800C 发送命令
//cmd:发送的命令字符串(不需要添加回车了),当 cmd<0XFF 的时候,发送数字(比如发送 0X1A),大于的时候发送字符串.
//ack:期待的应答结果,如果为空,则表示不需要等待应答
//waittime:等待时间(单位:10ms)
//返回值:0,发送成功(得到了期待的应答结果)
//      1,发送失败
u8 sim800c_send_cmd(u8 *cmd,u8 *ack,u16 waittime)
{
    u8 res=0;
    USART2_RX_STA=0;
    if((u32)cmd<=0XFF)
    {
        USART2->DR=(u32)cmd;
    }else u2_printf("%s\r\n",cmd);//发送命令

    if(ack&&waittime)    //需要等待应答
    {
        while(--waittime)    //等待倒计时
        {
            delay_ms(10);
            if(USART2_RX_STA&0X8000)//接收到期待的应答结果
            {
                if(sim800c_check_cmd(ack))break;//得到有效数据
                USART2_RX_STA=0;
            }
        }
        if(waittime==0)res=1;
    }
    return res;
}
```



```
//GSM 检测(SIM 卡准备和是否注册成功)
//返回值:2 正常
//其他,错误代码
u8 sim800c_gsminfo_show(void)
{
    static u8 sim_flag=0;

    USART2_RX_STA=0;

    switch(sim_flag)
    {
        case 0:
            if(sim800c_send_cmd("AT+CPIN?","OK",20)) //查询 SIM 卡是否在位
            {
                printf("SIM 卡不存在!\r\n");
            }else
            {
                sim_flag=1;
                printf("SIM 卡存在!\r\n");
            }break;
        case 1:
            if(sim800c_send_cmd("AT+CREG?","+CREG: 0,1",20))
                //查询 SIM 卡网络是否已注册
            {
                printf("网络正在注册!\r\n");
            }else
            {
                sim_flag=2;
                printf("网络注册成功!\r\n");
            }break;
    }
    USART2_RX_STA=0;
    return sim_flag;
}

//电话号码
#define CALL_NUM 10086

//sim800C 主测试程序
void sim800c_test(void)
{
    u8 timex=0;
    u8 sim_ready=0;
```

```

u8 call_ok=0;

while(sim800c_send_cmd("AT","OK",100))//检测是否应答 AT 指令
{
    printf("未检测到模块!!!\r\n");delay_ms(800);
    printf("尝试连接模块...\r\n");delay_ms(400);
    LED0=!LED0;
}
sim800c_send_cmd("ATE0","OK",200);//指令不回显
printf("\r\nSIM800C 拨打电话测试\r\n");
while(1)
{
    delay_ms(10);
    if(sim_ready)//SIM 卡就绪.
    {
        if(call_ok)
        {
            call_ok=0;
            delay_ms(50);
            printf("拨打%d\r\n",CALL_NUM);
            u2_printf("ATD%d;\r\n",CALL_NUM);//拨号
        }
    }
    if(!sim_ready)//SIM 卡没就绪
    {
        if(sim800c_gsminfo_show()==2)
        {
            sim_ready=1;
            call_ok=1;
        }
        else sim_ready=0;
    }
    if((timex%20)==0)LED0=!LED0;//200ms 闪烁
    timex++;
}
}

```

sim800c的代码比较简单就几个函数，下面简单讲解下。

首先，是检测模块应答函数：u8* sim800c_check_cmd(u8 *str)，该函数用于检测 ATK-SIM800C 模块发送回来的应答/数据，其中 str 为期待应答字符串，返回值如果为 0，则表示没有收到期待应答字符串，否则为期待应答字符串所在的位置。

第二个函数是：u8 sim800c_send_cmd(u8 *cmd,u8 *ack,u16 waittime)，该函数用于向 ATK-SIM800C 模块发送命令。cmd 为命令字符串，当 cmd<=0xFF 的时候，则直接发送 cmd，比如短信发送结束的时候，需要发送 0x1A，也可以通过该函数发送。ack 为期待应答字符串，waittime 为等待时间（单位：10ms），等待时间中，我们使用了定时器定义 10ms 更新中断，

用来实现两个字符接收间距以 10ms 接收超时标准(具体请看 timer.c 中定时器 3 的中断服务函数和 usart2.c 中 USART2_IRQHandler 串口 2 的中断服务函数)

第三个函数是: u8 sim800c_gsminfo_show(void), 该函数用于检测 ATK-SIM800C 模块 SIM 卡是否插上, 以及运营商网络是否注册成功, 返回值如果为 0, 则表示 SIM 卡已就绪。

最后要介绍的函数是: void sim800c_test(void), 该函数是本 ATK-SIM800C 模块测试的主函数, 先检测 SIM 卡是否就绪成功, 就绪了就拨打 CALL_NUM 宏定义的电话, 这里我们拨打的是 10086 号码。

sim800c.c 我们就介绍到这里, 我们再来看看 main.c, 该文件里面就一个 main 函数, main 函数代码如下:

```
int main(void)
{
    HAL_Init();                //初始化 HAL 库
    Stm32_Clock_Init(RCC_PLL_MUL9); //设置时钟,72M
    delay_init(72);            //初始化延时函数
    LED_Init();                //初始化 LED
    uart_init(115200);          //初始化串口 115200
    USART2_Init(115200);        //串口 2 初始化
    printf("ALIENTEK NANO STM32\r\n");
    while(1)
    {
        sim800c_test(); //GSM 测试
    }
}
```

此部分代码比较简单, 先对外设进行初始化, 然后就调用 sim800c_test()进行 GSM 的测试。

以上就是拨打电话的代码实现, 接下来讲解下接听电话测试。

2) 接听电话:

接听电话测试代码与拨打电话大致相同, 仅多了一个 call_test()函数, 该函数实现配置模块来电提醒功能, 等待来电接听来电的处理, 代码如下:

```
//来电检测函数
void call_test(void)
{
    u8 timex=0;
    u8 *p,*p2;
    u8 p1[20]={0}; //申请 20 直接用于存放号码

    if(sim800c_send_cmd("AT+CLIP=1","OK",200)); //设置来电显示
    if(sim800c_send_cmd("AT+COLP=1","OK",200)); //设置被叫号码显示
    printf("等待来电...\r\n");
    while(1)
    {
        delay_ms(10);
        if(USART2_RX_STA&0X8000)//接收到串口的数据
        {
```

```

USART2_RX_BUF[USART2_RX_STA&0X7FFF]=0;//添加结束符

if(sim800c_check_cmd("+CLIP:"))//接收到来电
{
    p=sim800c_check_cmd("+CLIP:");
    p+=8;
    p2=(u8*)strstr((const char *)p,"\n");
    p2[0]=0;//添加结束符
    strcpy((char*)p1,(char*)p);//获取电话号码

    printf("接听来电: %s\r\n",p1);//打印提示
    sim800c_send_cmd("ATA","OK",200);//接听来电
}
USART2_RX_STA=0;
}
if((timex%20)==0)LED0=!LED0;//200ms 闪烁
timex++;
}
}

```

函数中先设置模块的来电显示和被叫电话显示功能,然后在程序 while 中一直等待来电,当有来电时会立即接听,并且将来电的号码通过串口打印显示。

main.c 的代码跟之前一样,同样先初始化外设,然后进入 sim800c 测试。

以上就是接听电话的代码实现,接下来讲解下英文短信发送测试。

3) 英文短信发送:

英文短信发送测试代码与之前大致相同,仅多了一个 Text_SMS()函数,该函数实现配置模块发送短信功能的参数,以及启动短信的发送,代码如下:

```

/*****参数配置*****/

static char *phone="\13670613084\0";    //短信接收方手机号
static char *content="ATK-SIM800C MSG SEND TEST";//发送短信内容

/*****
//设置 TEXT 文本模式发送英文短信
void Text_SMS(void)
{
    u8 timex=0;
    char temp[50]="AT+CMGS=";

    sim800c_send_cmd("AT+CMGF=1","OK",200);    //设置文本模式
    sim800c_send_cmd("AT+CSCS=\\"GSM\\","OK",200);//设置 TE 字符集为 GSM
    strcat(temp,(char*)phone);                //字符串拼接函数
    sim800c_send_cmd((u8*)temp,">",200);      //发送短信命令+电话号码
    u2_printf("%s",content);                  //发送短信内容到 GSM 模块
}

```

```

delay_ms(50); //必须增加延时,否则接收方接收信息不全
if(sim800c_send_cmd((u8*)0X1A,"+CMGS:",1000)==0)
//发送结束符,等待发送完成(最长等待 10 秒钟,因为短信长了的话,
//等待时间会长一些)
{
    printf("短信发送成功!!\r\n");
}else
{
    printf("短信发送失败!!\r\n");
}
while(1)
{
    delay_ms(10);
    if((timex%20)==0)LED0=!LED0;//200ms 闪烁
    timex++;
}
}

```

函数中先配置短信功能的参数,然后调用 phone 和 content 变量向目标手机发送一条英文短信,测试时请勿使用程序中的手机号码,请修改自己的手机号码,以免短信收不到。

main.c 的代码跟之前一样,同样先初始化外设,然后进入 sim800c 测试。

以上就是英文短信发送的代码实现,接下来讲解下中英文短信发送测试。

4) 中英文短信发送:

中英文短信发送测试代码与英文短信发送大致相同,但在 Text_SMS()函数中会有所区别,同样该函数实现配置模块发送短信功能的参数,以及启动短信的发送,代码如下:

```

/*****参数配置*****/

static char *phone="\00310033003600370030003600310033003000380034\0";
//短信接收方手机号“13670613084” UNICODE 码
static char *content="00410054004B002D00530049004D003
800300030004300204E2D82F1658777ED4FE153D190016D4B8BD5";
//发送短信内容“ATK-SIM800C 中英文短信发送测试” UNICODE 码

/*****/

//设置 TEXT 文本模式发送英文短信
void Text_SMS(void)
{
    u8 timex=0;
    char temp[50]="AT+CMGS=";

    sim800c_send_cmd("AT+CMGF=1","OK",200); //设置文本模式
    sim800c_send_cmd("AT+CSCS=\"UCS2\"", "OK",200); //设置 TE 字符集为 UCS2
    sim800c_send_cmd("AT+CSMP=17,0,2,25","OK",200); //设置短消息文本模式参数
    strcat(temp,(char*)phone); //字符串拼接函数
}

```



```

sim800c_send_cmd((u8*)temp,">",200);//发送短信命令+电话号码
u2_printf("%s",content);//发送短信内容到 GSM 模块
delay_ms(50);//必须增加延时,否则接收方接收信息不全
if(sim800c_send_cmd((u8*)0X1A,"+CMGS:",1000)==0)//发送结束符,
等待发送完成(最长等待 10 秒钟,因为短信长了的话,等待时间会长一些)
{
    printf("短信发送成功!!\r\n");
}else
{
    printf("短信发送失败!!\r\n");
}
while(1)
{
    delay_ms(10);
    if((timex%20)==0)LED0=!LED0;//200ms 闪烁
    timex++;
}
}

```

函数中先配置短信功能的参数,然后调用 phone 和 content 变量向目标手机发送一条中英文短信,注意:电话号码和短信内容是 UNICODE 码,测试时请勿使用程序中的手机号码,请修改自己的手机号码,以免短信收不到。

main.c 的代码跟之前一样,同样先初始化外设,然后进入 sim800c 测试。

以上就是中英文短信发送的代码实现,接下来讲解下中英文短信接收测试。

5) 中英文短信接收:

中英文短信接收代码与中英文短信发送大致相同,但在 Text_SMS()函数中会有所区别,该函数实现配置模块接收短信功能的参数,配置当新信息来时,直接信息内容输出,代码如下:

```

//中英文短信接收
void Text_SMS(void)
{
    u8 timex=0;

    sim800c_send_cmd("AT+CMGF=1","OK",200); //设置文本模式
    sim800c_send_cmd("AT+CSCS=\"GSM\"","OK",200);//设置 TE 字符集为 GSM
    sim800c_send_cmd("AT+CNMI=3,2,0,0,0","OK",200);//设置新短信直接输出
    USART2_RX_STA=0;
    printf("等待短信接收..\r\n");
    while(1)
    {
        delay_ms(10);

        if(USART2_RX_STA&0X8000)//接收到串口的数据
        {
            USART2_RX_BUF[USART2_RX_STA&0X7FFF]=0;//添加结束符

```

```

        if(sim800c_check_cmd("+CMT"))//接收到新的信息
        {
            printf("\r\n 收到新短息:");
            printf("%s\r\n",USART2_RX_BUF);//模块收到的信息原样输出
            if(sim800c_check_cmd("LED1ON"))
            {
                LED1=0;//LED1 亮
            }
            if(sim800c_check_cmd("LED1OFF"))
            {
                LED1=1;//LED1 灭
            }
        }
        USART2_RX_STA=0;
    }
    if((timex%20)==0)LED0=!LED0;//200ms 闪烁
    timex++;
}
}

```

函数中配置好短信接收的参数后，会等待短信的接收，用手机往插在模块上的手机号码发送短信，收到短信会内容输出，若短信内容是纯数字、纯英文、或数字和英文的，会以字符串输出，若内容夹带中文，则是输出 UNICODE 码，这时请用汉字 UNICODE 互换工具将短信内容 UNICODE 码转换成中文。另外，当接收到短信内容是“LED1ON”时，开发板 LED1 亮，若短信内容时“LED1OFF”，开发板 LED1 灭。

main.c 的代码跟之前一样，同样先初始化外设，然后进入 sim800c 测试。

以上就是中英文短信接收的代码实现，接下来讲解下 GPRS_TCP 通信测试。

6) GPRS TCP 通信:

GPRS TCP 通信中，调用 GPRS_ServerCnt 函数，该代码如下：

```

/*****

u8 Heartbeat_cnt=0;//心跳累加数
u8 Heartbeat_OK=0;//心跳包发送 1:发送 0:未发送
const char *Server = "AT+CIPSTART=\"TCP\", \"119.129.58.199\",8086"; //IP 登录服务器

/*****

//定时器 2 中断服务函数
void TIM2_IRQHandler(void)
{
    if(__HAL_TIM_GET_FLAG(&TIM2_Handler,TIM_FLAG_UPDATE)!=RESET) //更新中断
    {
        __HAL_TIM_CLEAR_IT(&TIM2_Handler,TIM_IT_UPDATE); //清除中断
        Heartbeat_cnt++;
    }
}

```

```

        if(Heartbeat_cnt>9)//10 秒发送一次心跳包
        {
            Heartbeat_cnt=0;
            Heartbeat_OK=1;
        }
    }
}

//应答服务器
//功能:
//1,数据接收应答
//2,心跳包发送
void Send_OK(void)
{
    sim800c_send_cmd("AT+CIPSEND",>",100);
    sim800c_send_cmd("OK\32\0","SEND OK",100);//回复 OK
}

//GPRS 服务器连接
void GPRS_ServerCnt(void)
{
    u8 timex=0;

    sim800c_send_cmd("AT+CIPCLOSE=1","CLOSE OK",100);//关闭连接
    sim800c_send_cmd("AT+CIPSHUT","SHUT OK",100); //关闭移动场景
    sim800c_send_cmd("AT+CGCLASS=\"B\"","OK",100);
        //设置 GPRS 移动台类别为 B,支持包交换和数据交换
    sim800c_send_cmd("AT+CGDCONT=1,\"IP\", \"CMNET\"","OK",100);
        //设置 PDP 上下文,互联网接协议,接入点等信息
    sim800c_send_cmd("AT+CGATT=1","OK",100); //附着 GPRS 业务
    sim800c_send_cmd("AT+CIPCSGP=1,\"CMNET\"","OK",100);//设置为 GPRS 连接模式
    sim800c_send_cmd("AT+CIPHEAD=1","OK",100);
        //设置接收数据显示 IP 头(方便判断数据来源)
    if(!sim800c_send_cmd((u8*)Server,"OK",300)) //发起目标地址连接
    {
        printf("TCP 连接成功!\r\n");
    }
    else
    {
        printf("TCP 连接失败!\r\n");
        while(1);
    }
    TIM2_Init(9999,7199);//TIM2 1S 中断一次
    USART2_RX_STA=0;
}

```

```

while(1)
{
    delay_ms(10);

    if(USART2_RX_STA&0X8000)//接收到串口的数据
    {
        if(sim800c_check_cmd("+IPD"))//接收到新的信息
        {

            if(strstr((char*)USART2_RX_BUF,"LED1ON")!=NULL)
            {
                LED1=0;//LED1 亮
            }else if(strstr((char*)USART2_RX_BUF,"LED1OFF")!=NULL)
            {
                LED1=1;//LED1 灭
            }
            printf("\r\n 收到新信息:\r\n");
            printf("%s",USART2_RX_BUF);
            Heartbeat_OK=1;//接收应答
        }

        USART2_RX_STA=0;
    }
    if(Heartbeat_OK)//启动发送
    {
        Send_OK();
        Heartbeat_OK=0;
    }
    if((timex%20)==0)LED0=!LED0;//200ms 闪烁
    timex++;
}
}

```

函数中先设置 GPRS TCP 通信的功能参数,然后连接定义好的 IP 地址,建立其 TCP 通信,使能 1 秒周期的定时器,用于定时心跳包的发送,若无数据接收,当定时到 10 秒,会发送一次心跳包给对方 IP,以保持 TCP 的连接。当接收到对方 IP 发送的数据会串口输出打印,并且回复应答给对方 IP。当收到对方 IP 数据是“LED1ON”,开发板 LED1 亮,收到“LED1OFF”时,开发板 LED1 灭。注意,测试时请修改自己的 IP 地址,以免通信不正常。

main.c 的代码跟之前一样,同样先初始化外设,然后进入 sim800c 测试。

以上就是 GPRS TCP 通信的代码实现,接下来讲解下 GPRS_UDP 通信测试。

7) GPRS UDP 通信:

GPRS UDP 通信跟上一讲基本一致,同样也调用 GPRS_ServerCnt 函数,该代码如下:

```

/*****

```

```

u8 Heartbeat_cnt=0;//心跳累加数
u8 Heartbeat_OK=0;//心跳包发送 1:发送 0:未发送
const char *Server = "AT+CIPSTART=\"UDP\", \"119.129.58.199\",8086"; //IP 登录服务器

/*****/

//定时器 2 中断服务函数
void TIM2_IRQHandler(void)
{
    if(__HAL_TIM_GET_FLAG(&TIM2_Handler,TIM_FLAG_UPDATE)!=RESET) //更新中断
    {
        __HAL_TIM_CLEAR_IT(&TIM2_Handler,TIM_IT_UPDATE); //清除中断

        Heartbeat_cnt++;

        if(Heartbeat_cnt>9)//10 秒发送一次心跳包
        {
            Heartbeat_cnt=0;
            Heartbeat_OK=1;
        }
    }
}

//应答服务器
//功能:
//1,数据接收应答
//2,心跳包发送
void Send_OK(void)
{
    sim800c_send_cmd("AT+CIPSEND",>",100);
    sim800c_send_cmd("OK\32\0","SEND OK",100); //回复 OK
}

//GPRS 连接服务器函数
void GPRS_ServerCnt(void)
{
    u8 timex=0;

    sim800c_send_cmd("AT+CIPCLOSE=1","CLOSE OK",100); //关闭连接
    sim800c_send_cmd("AT+CIPSHUT","SHUT OK",100); //关闭移动场景
    sim800c_send_cmd("AT+CGCLASS=\"B\", \"OK\",1000);
        //设置 GPRS 移动台类别为 B,支持包交换和数据交换
    sim800c_send_cmd("AT+CGDCONT=1,\"IP\", \"CMNET\", \"OK\",1000);
        //设置 PDP 上下文,互联网接协议,接入点等信息

```



```

sim800c_send_cmd("AT+CGATT=1","OK",500); //附着 GPRS 业务
sim800c_send_cmd("AT+CIPCSGP=1,\"CMNET\"","OK",500); //设置为 GPRS 连接模式
sim800c_send_cmd("AT+CIPHEAD=1","OK",500);
//设置接收数据显示 IP 头(方便判断数据来源)
if(!sim800c_send_cmd((u8*)Server,"OK",300)) //发起目标地址连接
{
    printf("UDP 连接成功!\r\n");
}
else
{
    printf("UDP 连接失败!\r\n");
    while(1);
}
TIM2_Init(9999,7199); //TIM2 1S 中断一次
USART2_RX_STA=0;

while(1)
{
    delay_ms(10);

    if(USART2_RX_STA&0X8000)//接收到串口的数据
    {
        if(sim800c_check_cmd("+IPD"))//接收到新的信息
        {
            if(strstr((char*)USART2_RX_BUF,"LED1ON")!=NULL)
            {
                LED1=0; //LED1 亮
            }else if(strstr((char*)USART2_RX_BUF,"LED1OFF")!=NULL)
            {
                LED1=1; //LED1 灭
            }
            printf("\r\n 收到新信息:\r\n");
            printf("%s",USART2_RX_BUF);
            Heartbeat_OK=1; //接收应答
        }

        USART2_RX_STA=0;
    }
    if(Heartbeat_OK)//启动发送
    {
        Send_OK();
        Heartbeat_OK=0;
    }
    if((timex%20)==0)LED0=!LED0; //200ms 闪烁
}

```

```

        timex++;
    }
}

```

函数中先设置 GPRS UDP 通信的功能参数，然后连接定义好的 IP 地址，建立其 UDP 通信，使能 1 秒周期的定时器，用于定时心跳包的发送，若无数据接收，当定时到 10 秒，会发送一次心跳包给对方 IP，以保持 TCP 的连接。当接收到对方 IP 发送的数据会串口输出打印，并且回复应答给对方 IP。当收到对方 IP 数据是“LED1ON”，开发板 LED1 亮，收到“LED1OFF”时，开发板 LED1 灭。注意，测试时请修改自己的 IP 地址，以免通信不正常。

main.c 的代码跟之前一样，同样先初始化外设，然后进入 sim800c 测试。

以上就是 GPRS UDP 通信的代码实现，接下来讲解下蓝牙被动连接通信测试。

8) 蓝牙被动连接通信：

蓝牙被动连接测试代码中 sim800c_test()函数与之前的有所区别，函数如下：

```

//sim800C 测试程序
void sim800c_test(void)
{
    char *p1,*p2;
    u8 timex=0;
    u8 sendbuf[20];
    u8 sendcnt=0;

    while(sim800c_send_cmd("AT","OK",100))//检测是否应答 AT 指令
    {
        printf("未检测到模块!!!\r\n");delay_ms(800);
        printf("尝试连接模块...\r\n");delay_ms(400);
        LED0=!LED0;
    }
    printf("\r\nSIM800C 蓝牙被动连接通信测试\r\n");

    sim800c_send_cmd("ATE1","OK",200);//指令回显
    sim800c_send_cmd("AT+BTPOWER=1","AT",300);
    //打开蓝牙电源 不判断 OK，因为电源原本开启再发送打开的话会返回 error
    sim800c_send_cmd("AT+BTUNPAIR=0","AT",120);
    //删除已配对的蓝牙设置,不判断 OK，因为没有已配对的设备时返回 ERROR
    printf("等待连接请求.....\r\n");
    do
    {
        if(!sim800c_wait_request("+BTPAIRING:",30))//等待手机端蓝牙连接请求
        {
            delay_ms(10);
            sim800c_send_cmd("AT+BTPAIR=1,1","BTPAIR:",20);//响应连接
        }
        LED1=!LED1;
    }while(strstr((const char*)USART2_RX_BUF,"+BTPAIR: 1")!=NULL);//判断是否匹配成功
    LED1=1;
}

```

```

USART2_RX_STA=0;
printf("蓝牙连接成功!\r\n");
printf("等待 SPP 连接.....\r\n");
do
{
    if(!sim800c_wait_request("SPP",50))//等待 SPP 连接请求
        break;
    LED1=!LED1;
}while(1);
LED1=1;
if(!sim800c_send_cmd("AT+BTACPT=1","+BTCONNECT:",200))
//应答手机端 spp 连接请求 2S
{
    printf("SPP 连接成功\r\n");
}
else
{
    printf("SPP 连接失败\r\n");
}
while(1)
{
    delay_ms(10);

    if(USART2_RX_STA&0x8000)
    {
        USART2_RX_BUF[USART2_RX_STA&0x7FFF]=0;//添加结束符
        USART2_RX_STA=0;

        p1=strstr((const char*)USART2_RX_BUF,"DATA: ");
        if(p1)
        {
            p2=(char*)strstr((const char*)p1,",");
            if(p2)
            {
                p2=strstr((const char *)p1,",");//搜索第 1 个,
                p1=strstr((const char *)p2+1,",");//搜索第 2 个,
                printf("接收数据: ");
                printf("%s\r\n",p1+1);//数据打印到串口
            }
        }
    }
    if(timex%50==0)//500ms 发送数据
    {
        sim800c_send_cmd("AT+BTSPSEND", ">", 100);//发送数据
        sprintf((char*)sendbuf, "Bluetooth test %d \r\n\32", sendcnt);
    }
}

```

```

        sendcnt++;
        if(sendcnt>99) sendcnt = 0;
        sim800c_send_cmd((u8*)sendbuf,"OK",100);//发送数据
        LED1=!LED1;
    }

    if((timex%20)==0)LED0=!LED0;//200ms 闪烁
    timex++;
}
}

```

函数中模块先配置好蓝牙, 然后等待蓝牙连接, 这时可以用带蓝牙的安卓手机开启蓝牙, 搜索“sim800c”设备并进行连接, 连接成功后。手机端打开蓝牙串口助手 PRO APP 搜索“sim800c”设备进行 spp 连接, 知道连接成功。连接成功后, 开发板会以 500MS 的时间向手机发送数据, 手机端向开发板发送的数据会在串口助手输出显示。

main.c 的代码跟之前一样, 同样先初始化外设, 然后进入 sim800c 测试。

以上就是蓝牙被动连接通信的代码实现, 接下来讲解下蓝牙主动连接通信测试。

9) 蓝牙主动连接通信:

蓝牙主动连接测试代码 sim800c_test() 与被动连接的差不多, 唯独主动需要去扫描连接手机, sim800c_test() 函数如下:

```

//sim800C 测试程序
void sim800c_test(void)
{
    char *p1,*p2;
    u8 timex=0;
    u8 sendbuf[20];
    u8 sendcnt=0;

    while(sim800c_send_cmd("AT","OK",100))//检测是否应答 AT 指令
    {
        printf("未检测到模块!!!\r\n");delay_ms(800);
        printf("尝试连接模块...\r\n");delay_ms(400);
        LED0=!LED0;
    }
    printf("\r\nSIM800C 蓝牙主动连接通信测试\r\n");

    sim800c_send_cmd("ATE1","OK",200);//指令回显
    sim800c_send_cmd("AT+BTPOWER=1","AT",300);
    //打开蓝牙电源 不判断 OK, 因为电源原本开启再发送打开的话会返回 error
    sim800c_send_cmd("AT+BTUNPAIR=0","AT",120);
    //删除已配对的蓝牙设置,不判断 OK, 因为没有已配对的设备时返回 ERROR
    LED1=0;
    do
    {
        printf("搜索设备中.....\r\n");
    }
}

```

```

sim800c_send_cmd("AT+BTSCAN=1,11","+BTSCAN: 1",1100);
//搜索附近的蓝牙设备,搜索 11s(重新配置定时器分频系数, 定时为 1S 中断,
//蓝牙扫描结束后重新配置为 10ms 定时中断)

LED1=!LED1;
}while(strstr((const char*)USART2_RX_BUF,"+BTSCAN: 0")!=NULL);
//判断是否扫描到设备
LED1=1;
USART2_RX_STA=0;
printf("发现设备!\r\n");
do
{
    printf("正在连接中.....\r\n");
    if(!sim800c_send_cmd("AT+BTPAIR=0,1","+BTPAIRING:",500))
        //连接搜索到的第一个设备
    {
        sim800c_send_cmd("AT+BTPAIR=1,1","BTPAIR:",500);
        //响应连接, 如果手机长期不确认匹配, SIM800C 端蓝牙 30S 后才会上
        报配对失败
    }
    LED1=!LED1;
}while(strstr((const char*)USART2_RX_BUF,"+BTPAIR: 1")!=NULL);
//判断是否 匹配成功
LED1=1;
USART2_RX_STA=0;
BT_Scan_mode=0;
printf("蓝牙连接成功!\r\n");
printf("等待 SPP 连接.....\r\n");
do
{
    if(!sim800c_wait_request("SPP",50))//等待 SPP 连接请求
        break;
    LED1=!LED1;
}while(1);
LED1=1;

if(!sim800c_send_cmd("AT+BTACPT=1","+BTCONNECT:",200))
//应答手机端 spp 连接请求 2S
{
    printf("SPP 连接成功!!\r\n");
}else
{
    printf("SPP 连接失败!!\r\n");
}

```



```

while(1)
{
    delay_ms(10);

    if(USART2_RX_STA&0x8000)
    {
        USART2_RX_BUF[USART2_RX_STA&0x7FFF]=0;//添加结束符
        USART2_RX_STA=0;

        p1=strstr((const char*)USART2_RX_BUF,"DATA: ");
        if(p1)
        {
            p2=(char*)strstr((const char*)p1,",");
            if(p2)
            {
                p2=strstr((const char *)p1,",");//搜索第 1 个,
                p1=strstr((const char *)p2+1,",");//搜索第 2 个,
                printf("接收数据: ");
                printf("%s\r\n",p1+1);//数据打印到串口
            }
        }
    }

    if(timex%50==0)//500ms 发送数据
    {
        sim800c_send_cmd("AT+BTSPSEND",>,100);//发送数据
        sprintf((char*)sendbuf,"Bluetooth test %d \r\n\32",sendcnt);
        sendcnt++;
        if(sendcnt>99) sendcnt = 0;
        sim800c_send_cmd((u8*)sendbuf,"OK",100);//发送数据
        LED1=!LED1;
    }

    if((timex%20)==0)LED0=!LED0;//200ms 闪烁
    timex++;
}
}

```

函数中模块先配置好蓝牙，然后会启动搜索周边的蓝牙设备，当搜索到设备后，会向搜索列表的第一个设备发起连接申请（这里需保证搜索列表的第一个设备时你的手机），这时手机端会收到蓝牙配对的申请，点击确定进行连接，连接成功后，手机端打开蓝牙串口助手 PRO APP，搜索“sim800c”设备进行 spp 连接，直到连接成功。spp 连接成功后，连接成功后，开发板会以 500MS 的时间向手机发送数据，手机端向开发板发送的数据会在串口助手输出显示。

main.c 的代码跟之前一样，同样先初始化外设，然后进入 sim800c 测试。

以上就是蓝牙主动连接通信的代码实现，接下来讲解下 TTS 文本转语音测试。

10) TTS 文本转语音：

TTS 文本转语音代码比较简单，主要是 TTS_Test()函数实现，代码如下：

```

/*****
#define MAX_LEN 200 //最多播放字节数 100 个汉字
/*****
//TTS 文本语音测试
void TTS_Test(void)
{
    u16 len=0;
    u8 timex=0;
    u8 temp[MAX_LEN+15];
    u8 loc=0;
    sim800c_send_cmd("AT+CTTSPARAM=5,0,51,61,0","OK",200);
    //设置 TTS 语调音量大小参数设置
    strcpy((char*)&temp,(const char*)"AT+CTTS=2,\");
    loc=sizeof("AT+CTTS=2,\");
    printf("\r\nSIM800C TTS 文本语音测试\r\n\r\n");
    printf("语音播放: 输入\"播放文本内容+回车\"\r\n");
    printf("退出测试: 输入\"暂停+回车\"\r\n");
    printf("注意:最大支持%d 字节(中文占 2 个字节)\r\n\r\n",MAX_LEN);
    while(1)
    {
        if(USART_RX_STA&0x8000)
        {
            len = USART_RX_STA&0X3FFF;

            if((len==4)&&strstr((char*)USART_RX_BUF,"暂停"))
            {
                printf("暂停播放!\r\n");
                sim800c_send_cmd("AT+CTTS=0","OK",10);
            }else
            {
                strcpy((char*)&temp[loc-1],(const char*)USART_RX_BUF);
                temp[loc+len-1]='\';
                temp[loc+len]='\0';
                USART_RX_BUF[len]='\0';
                if(sim800c_send_cmd((u8*)temp,"OK",100))
                {
                    printf("上一文本未播完!\r\n");
                }else
                {
                    printf("播放文本: %s\r\n",USART_RX_BUF);
                }
            }
        }
    }
}

```

```
    }  
    USART_RX_STA=0;  
}  
delay_ms(10);  
if((timex%20)==0)LED0=!LED0;//200ms 闪烁  
timex++;  
}  
}
```

函数中先对模块进行 TTS 参数的配置，然后等待串口 1 数据的接收，当在串口助手输入需要播放的文本，然后勾选“发送新行”，点击发送，模块会将文本内容转换语音输出。当模块正在播放语音，在串口助手输入:暂停，然后勾选“发送新行”，点击发送，模块会停止当前的语音播放。注意，例程中最大支持 200 字节文本内容播放(中文占 2 个字节)。

main.c 的代码跟之前一样，同样先初始化外设，然后进入 sim800c 测试。

至此，软件实现部分就介绍完了，我们接下来看代码验证。

4、验证

首先，请先确保硬件都已经连接好了：

- 1，给 ATK-SIM800C 模块装上 SIM 卡，并插好耳机和麦克风。
- 2，连接 ATK-SIM800C 模块与 ALIENTEK NANO STM32F103 开发板（连接方式见表：2.1）
- 3，给 ATK-SIM800C 模块上电（按 K1，蓝色电源指示灯亮）
- 4，ATK-SIM800C 模块开机（**长按 PWR_KEY 键开机** 或 **用跳线帽短接 P1 口的 PKEY 与 VBAT 实现上电自动开机**，红色 NET_STA 指示灯闪烁）

4.1 拨打电话

下载程序后会拨通 10086，插上耳机可以听到在通话，如图 4.1.1 所示：

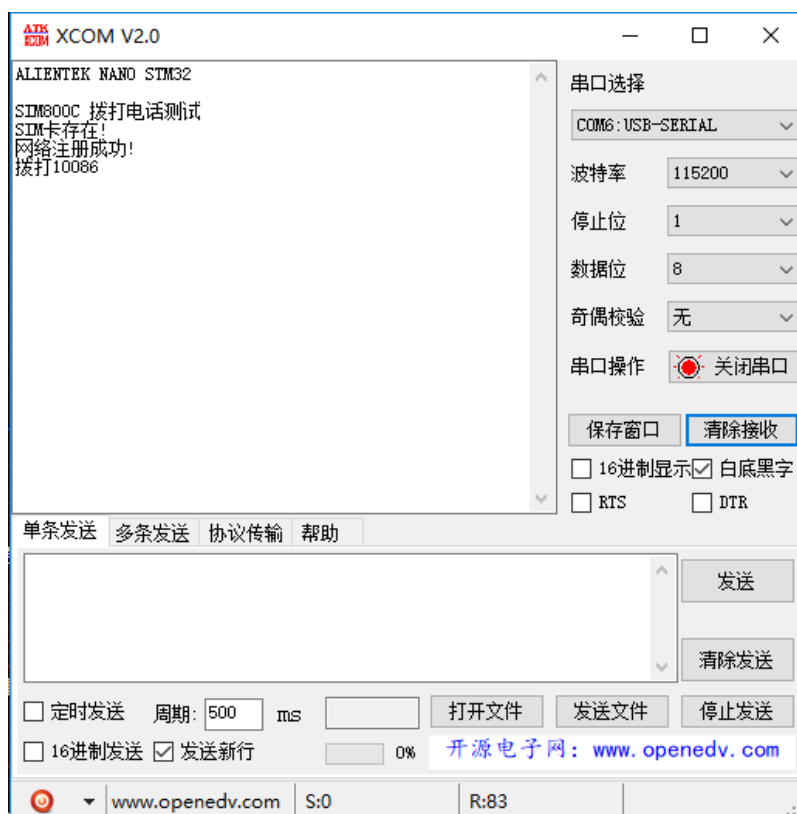


图 4.1.1 拨打电话

4.2 接听电话

用手机拨打电话到模块上的手机卡，当识别到有来电时模块会自动接听来电，如图 4.2.1 所示：

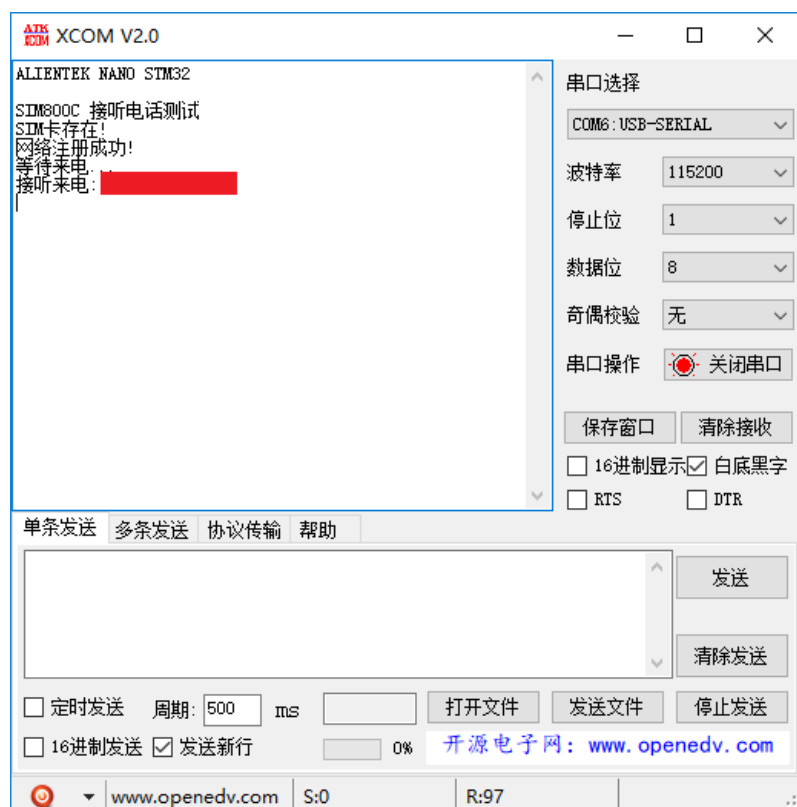


图 4.2.1 接听电话

4.3 英文短信发送

下载程序后会向目标手机号发送英文短信，如图 4.3.1 所示：

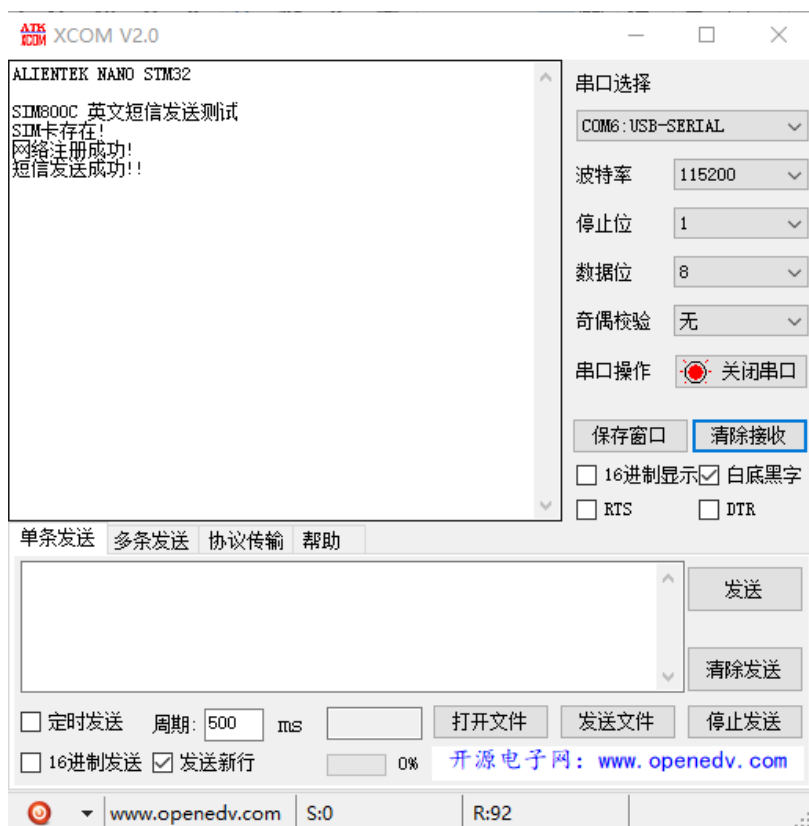


图 4.3.1 英文短信发送

4.4 中英文短信发送

下载程序后会向目标手机号发送中英文短信，如图 4.4.1 所示：

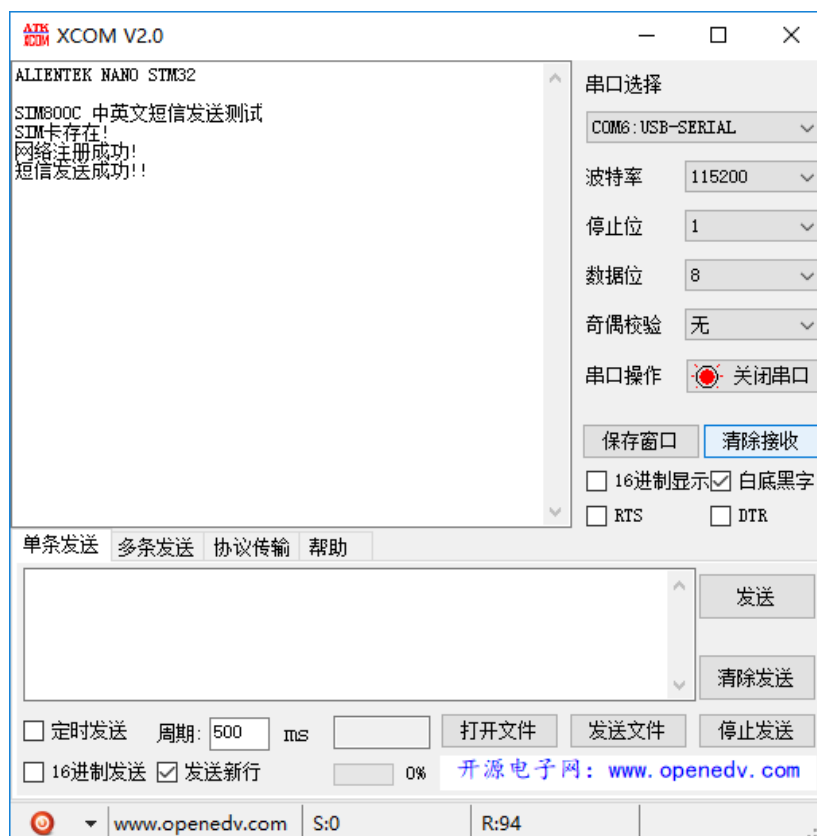


图 4.4.1 中英文短信发送

4.5 中英文短信接收

下载程序后开发板等待短信的接收,用手机往模块的手机号码发送短信,模块收到短信会内容输出,若短信内容是纯数字、纯英文、或数字和英文的,会以字符串输出,如图 4.5.1 所示,若内容夹带中文,则是输出 UNICODE 码,如图 4.5.2 所示,这时请用汉字 UNICODE 互换工具将短信内容 UNICODE 码转换成中文。当接收到短信内容是“LED1ON”,开发板 LED1 亮;短信内容是“LED1OFF”,开发板 LED1 灭。

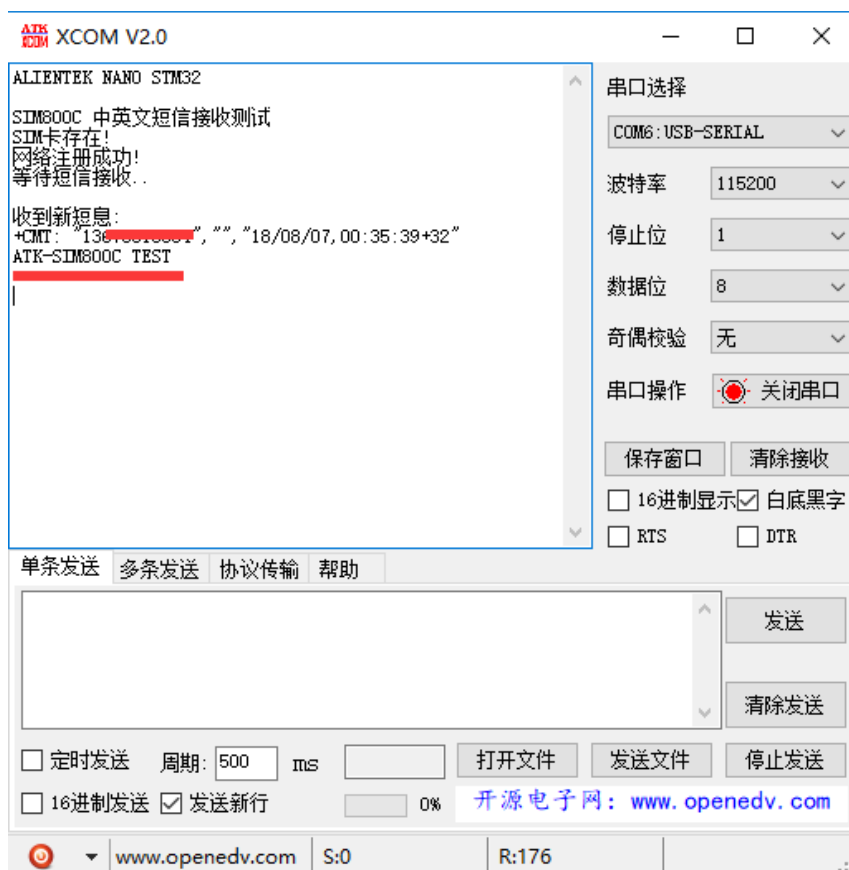


图 4.5.1 非中文短信接收



图 4.5.2 中英文短信接收

4.6 GPRS TCP 通信

下载程序后，模块会往例程设定的 IP 和端口进行 TCP 通信连接，连接成功，对方 IP 往模块发送数据会串口输出，如图 4.6.1 所示，当发送“LED1ON”，开发板 LED1 亮，发送“LED1OFF”时，开发板 LED1 灭。开发板收到模块返回的 TCP 通信数据，会往对方 IP 发送数据应答，若 10 秒内开发板没收到数据，会向服务器发送心跳帧“OK”，如图 4.6.2 所示。



图 4.6.1 接收 TCP 通信数据



图 4.6.2 TCP 服务器接收数据

4.7 GPRS UDP 通信

下载程序后，模块会往例程设定的 IP 和端口进行 UDP 通信连接，连接成功，对方 IP 往模块发送数据会串口输出，如图 4.7.1 所示，当发送“LED1ON”，开发板 LED1 亮，发送“LED1OFF”时，开发板 LED1 灭。开发板收到模块返回的 UDP 通信数据，会往对方 IP 发送数据应答，若 10 秒内开发板没收到数据，会向服务器发送心跳帧“OK”，如图 4.7.2 所示。



图 4.6.1 接收 UDP 通信数据



图 4.6.2 电脑端 UDP 接收数据

4.8 蓝牙被动连接通信

下载程序后，用带蓝牙的安卓手机开启蓝牙，搜索“sim800c”设备并进行连接。连接成功过后，如图 4.8.1 所示，手机端打开蓝牙串口助手 PRO APP 搜索“sim800c”设备进行 spp 连接，直到连接成功。SPP 连接成功后，开发板以 500ms 时间向手机发送数据，如图 4.8.2，同时手机端发送的数据会串口输出，如图 4.8.3 所示。

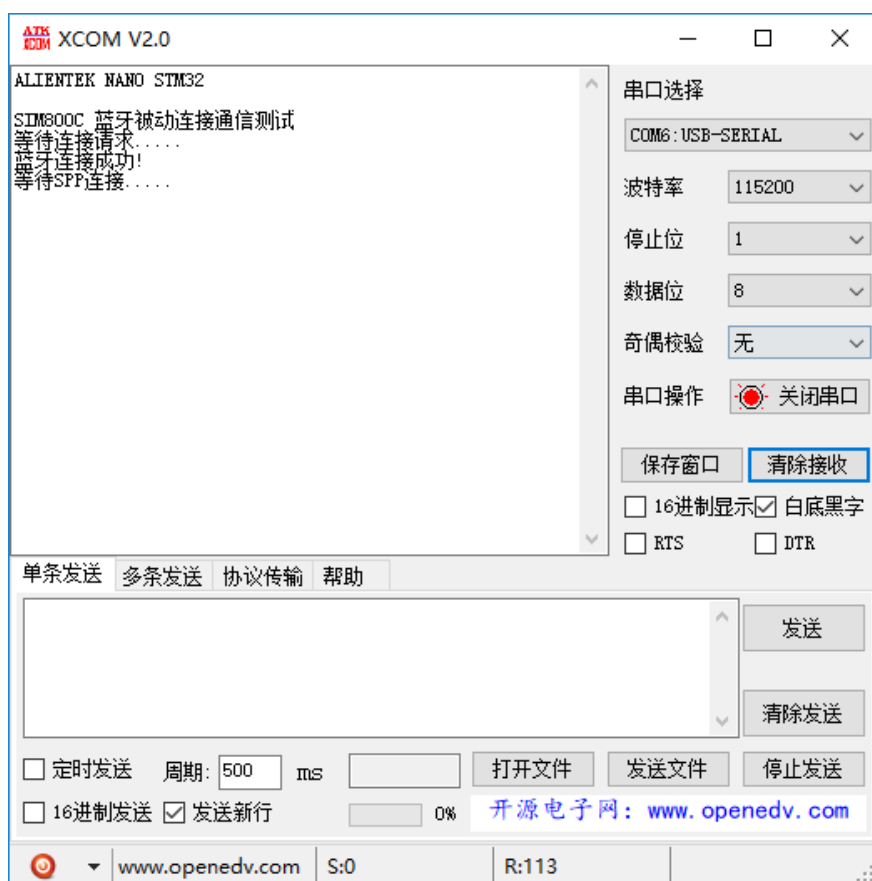


图 4.8.1 等待 SPP 连接



图 4.8.2 手机端接收数据

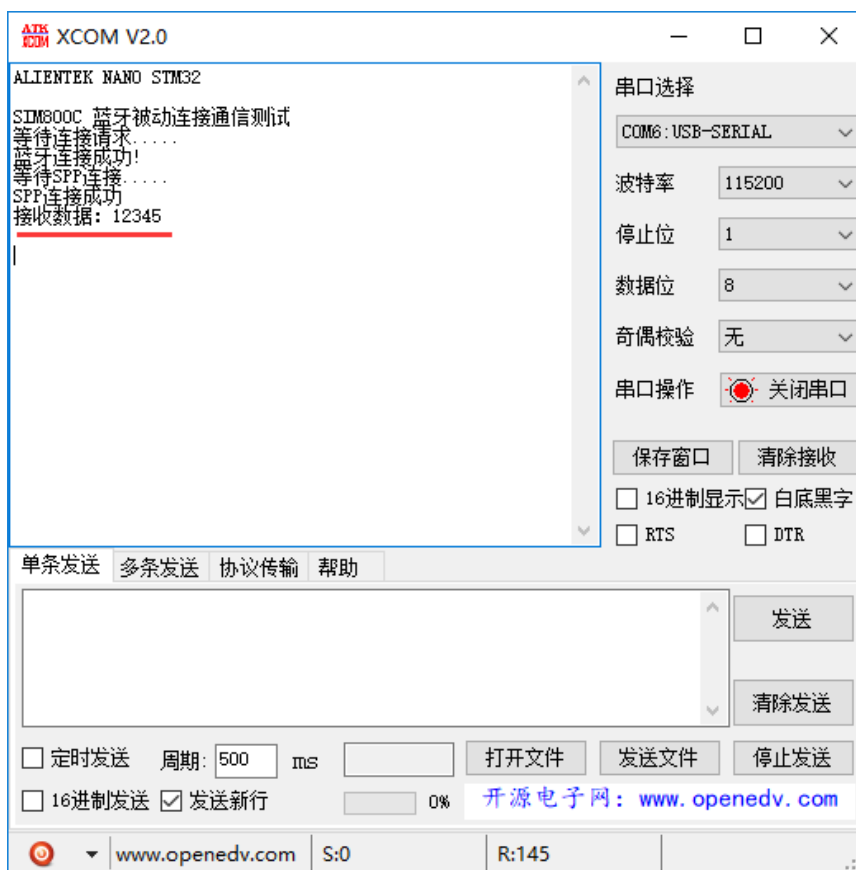


图 4.8.3 开发板接收手机发送的数据

4.9 蓝牙主动连接通信

下载程序后，模块会搜索周边的蓝牙设备，当搜索到设备后，会向搜索列表第一个设备发起连接申请（这里需保证搜索列表的第一个设备是你的手机），这时手机端会收到蓝牙配对申请，点击确定进行连接，连接成功后，如图 4.9.1 所示，手机端打开蓝牙串口助手 PRO APP，搜索“sim800c”设备进行 spp 连接，直到连接成功。SPP 连接成功后，开发板以 500ms 时间向手机发送数据，如图 4.9.2 所示，同时手机端发送的数据会串口输出，如图 4.9.3 所示。

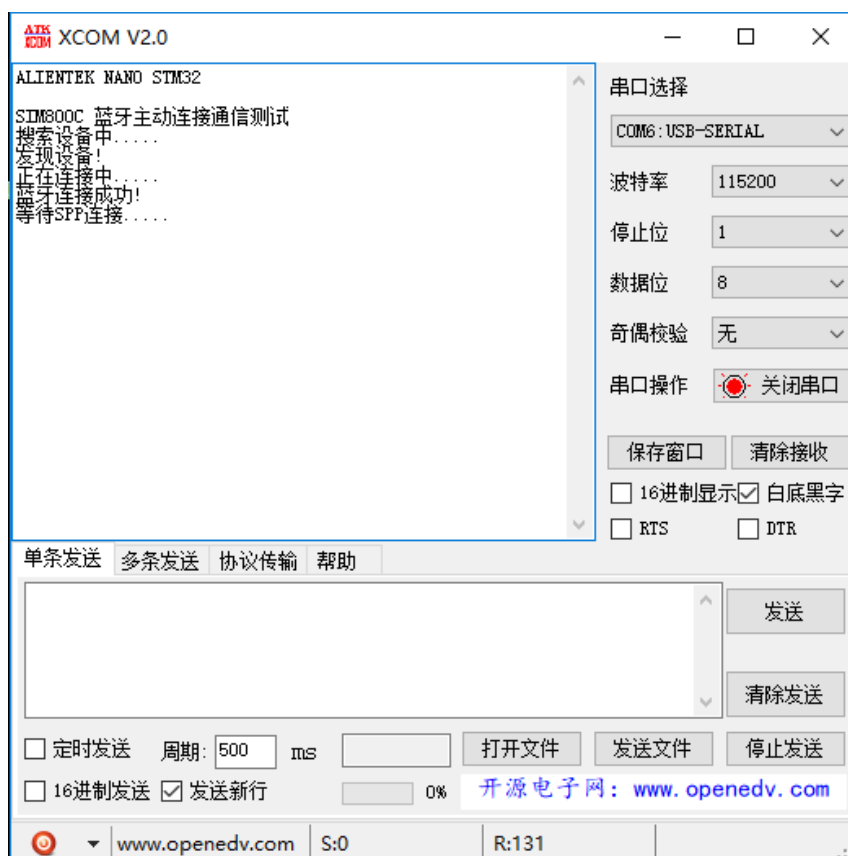


图 4.9.1 等待 SPP 连接



图 4.9.2 手机端接收数据

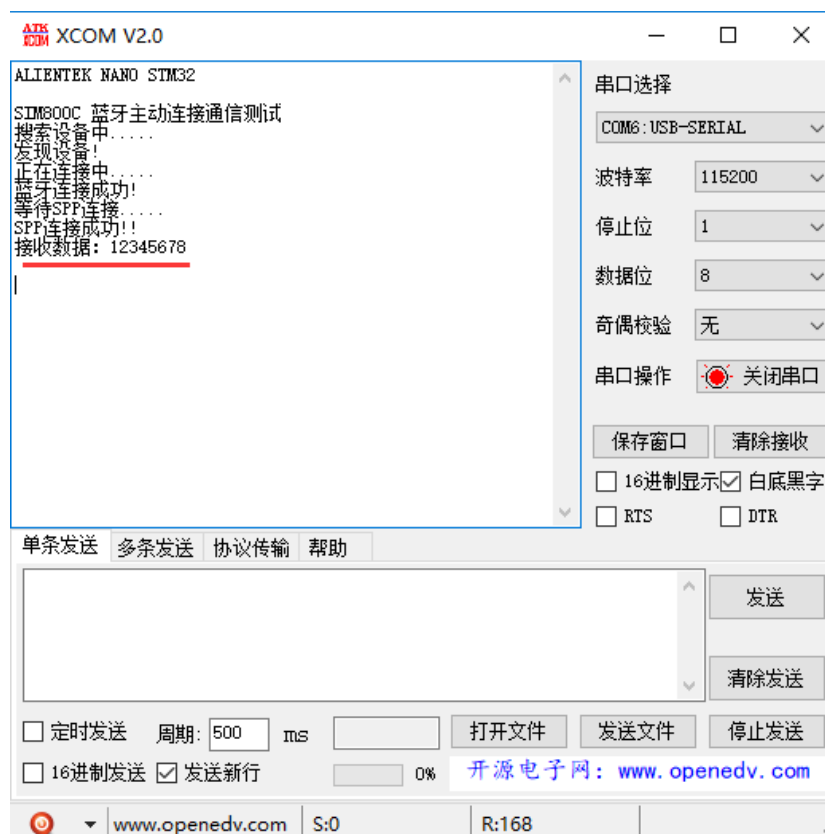


图 4.9.3 开发板接收手机发送的数据

4.10 TTS 文本转语音

下载程序后，开发板等待串口 1 数据的接收，当在在串口助手输入需要播放的文本，然后勾选“发送新行”，点击发送，模块会将文本内容转换程语音输出。如图 4.10.1 所示，当模块正在播放语音，在串口助手输入:暂停，然后勾选“发送新行”，点击发送，模块会停止当前的语音播放，如图 4.10.2 所示。注意，例程中最大支持 200 字节文本内容播放(中文占 2 个字节)。

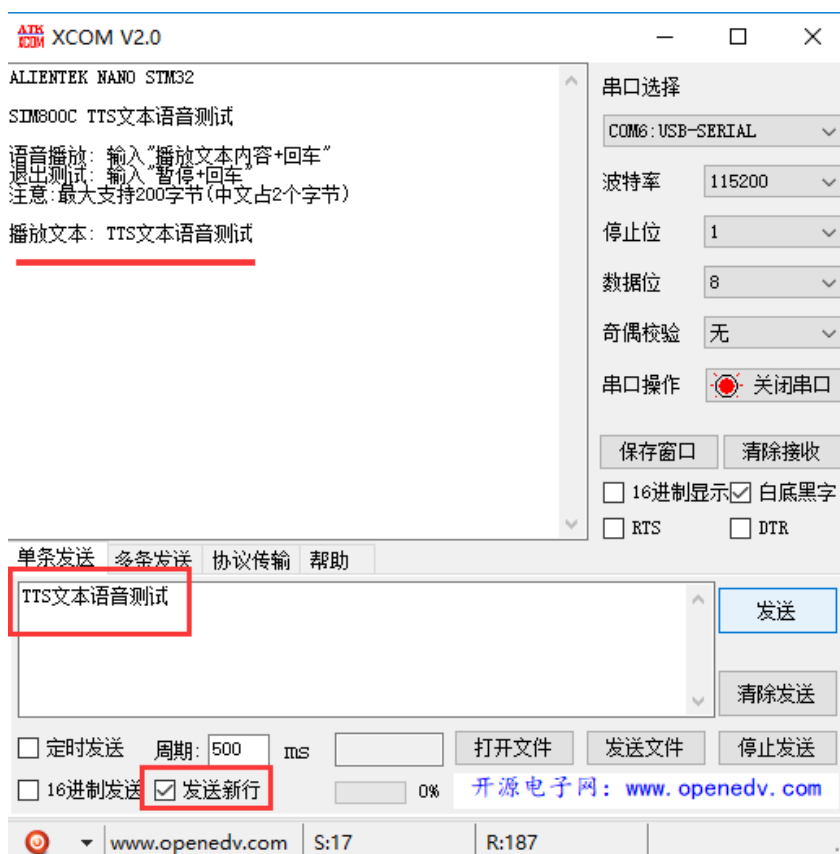


图 4.10.1 TTS 文本语音测试

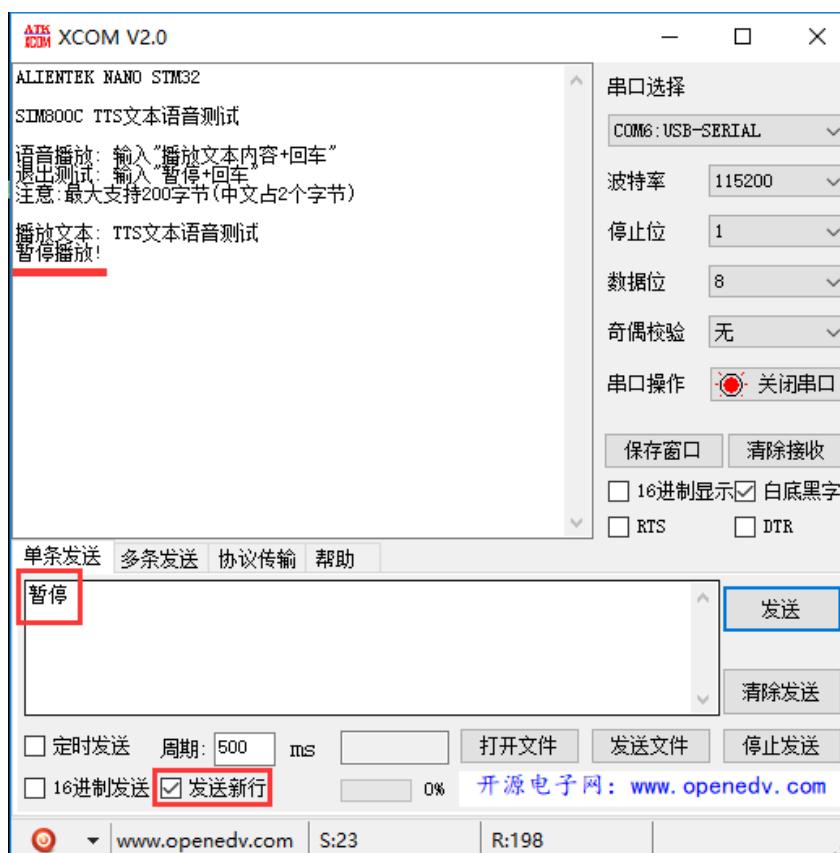


图 4.10.2 暂停语音播放

至此，关于 ATK-SIM800C GSM/GPRS 模块的使用介绍，我们就讲完了，本文档详细介绍了 ATK-SIM800C 模块的使用，有助于大家快速学会 ATK-SIM800C 模块的使用。

正点原子@ALIENTEK

2018-8-7

公司网址: www.alientek.com

技术论坛: www.openedv.com

电话: 020-38271790

传真: 020-36773971

