# 1. 环境准备

硬件准备

小熊派开发板

BearPi-IoT

NB-IoT通信模组（BC35-G）

BC35-G 是一款高性能、低功耗的多频段 NB-IoT 无线通信模块，支持 B1/B3/B8/B5/B20/B28 频段，在设计和AT指令上与BC95兼容。

BC35-G

小熊派NB-IoT通信扩展模组的波特率为9600bps。

软件准备

QCOM串口助手

QCOM串口助手

文档准备

BC35-G AT命令手册

此文档来自于移远官方！阅读本教程时，关于AT指令的详细说明请参考该文档！

切换开关

小熊派开发板右上角的开关拨到AT-PC一端，则模组直接与PC相连，方便调试。

所有工具和文档均在『小熊派开源社区』公众号回复“通信模组”获取！

# 2. NB-IoT入网AT指令

AT

指令：AT

功能：测试AT指令功能是否正常

示例：

AT

OK

获取信号强度指示

指令：AT+CSQ

功能：返回从 UE 接收到的信号强度指示 <rssi> 和信道误码率 <ber>，其中第一个值rssi应当在0-31之间，如果为99则表示信号无法检测，第二个参数ber因为模组当前不支持，所以始终为99。

示例：

+CSQ:26,99

OK

查询网络注册状态

指令：AT+CEREG?

功能：查询当前 EPS 网络注册状态，该指令返回的第一个参数为0则表示禁止网络注册URC，第二个参数表示网络注册状态，1表示已注册本地网，5表示已注册漫游网络，其余值则表示注册失败。

示例：

+CEREG:0,1

OK

查询网络是否被激活

指令：AT+CGATT?

功能：该命令用于查询当前是否将 UE 附着于 PS 域，返回值为1则表示已附着，即网络激活成功。

示例：

+CGATT:1

OK

查询模块的 IP 地址

指令：AT+CGPADDR

功能：该命令用于查询模组当前的ip地址。

示例：

+CGPADDR:0,10.44.108.10

OK

# 3. 基于 UDP 协议连接远程服务器通信实例

## 3.1. 禁用IoT平台的注册功能

由于NB-IoT模组可以直接对接IoT平台，所以在单独测试使用UDP连接时，需要在激活网络成功之后，在获取ip地址之前，关闭IoT平台注册功能。

使用如下命令禁止该功能：

AT+QREGSWT=2

## 3.2. 搭建远程UDP服务器

首先我们需要搭建一个UDP服务器，有两种方式：

在服务器上使用Python、Java、C#等语言自行编写服务器程序；

在本地PC上使用网络调试助手开启UDP服务器；

因为 NB-IoT 模组直接注册的是公网ip地址，所以这里我们使用第一种方式，在Linux服务器上运行一个Python编写的UDP测试服务器：

本地PC使用的是局域网，公网不可以直接根据ip地址访问到本PC，需要进行内网穿透，不推荐使用。

这里的Python程序如下：

# udp-server.py

from socket import \*

host = ''

port = 8000

# 创建server socket

server\_socket = socket(AF\_INET,SOCK\_DGRAM)

# 绑定socket监听地址

server\_addr = (host,port)

server\_socket.bind(server\_addr)

print('UDP Server Start...')

# 处理连接请求

while(True):

# 接收客户端的数据

data, addr = server\_socket.recvfrom(1024)

print("Receive from %s:%s" % addr % data)

if data == b"quit":

server\_socket.sendto(b"Good bye!\n", addr)

continue

server\_socket.sendto(b"Hello,udp client!\n", addr)

运行：

python3 udp-server.py

效果如下：

UDP服务器程序

## 3.3. 模组与服务器通信

创建Socket

使用AT命令连接UDP服务器，首先需要创建一个 UDP 类型的 Socket，创建socket的指令如下：

AT+NSOCR=<type>,<protocol>,<listen port>

其中第一个参数是socket类型，DGRAM表示UDP，STREAM表示UDP；第二个参数表示协议类型，UDP 为 17， UDP 为 6，最后一个参数指定socket使用的本地端口，如果为0则表示随机分配。

所以创建UDP socket的示例如下：

AT+NSOCR=DGRAM,17,0

1

OK

模组向服务器发送消息

指令：

AT+NSOST=<socket>,<remote\_addr>,<remote\_port>,<length>,<data>

其中第一个参数是由 AT+NSOCR 返回的 Socket 编号，第二个参数是UDP服务器ip地址，也可以使用域名，第三个参数是UDP服务器开启监听的端口，第四个是发送数据的长度，最后一个是要发送的十六进制数据。

注意：该命令仅限UDP scoket连接服务器使用，并且所有参数均无引号！

示例：

AT+NSOST=1,122.51.89.94,8000,4,30313233

1,4

OK

发送之后，在服务器端也可以看到：

服务器收到消息成功

模组接收服务器消息

模组发送数据到服务器后，服务器会自动发送消息，模组会打印出收到信息的提示：

+NSONMI:1,18

该信息表示编号为1的socket收到了18字节的数据。

可以使用如下命令查看收到的数据，第一个参数是socket编号，第二个参数是查询的数据长度：

AT+NSORF=<socket>,<req\_length>

查看刚刚收到的数据：

AT+NSORF=1,18

1,122.51.89.94,8000,18,48656C6C6F2C75647020636C69656E74210A,0

OK

其中收到的数据为倒数第二个参数，是十六进制格式：

48656C6C6F2C75647020636C69656E74210A

使用在线工具将数据转化为字符串即可：

转换结果

关闭scoket

通信完毕之后，可以使用下面的命令关闭最开始创建的socket：

AT+NSOCL=<socket>

示例：

AT+NSOCL=1

OK