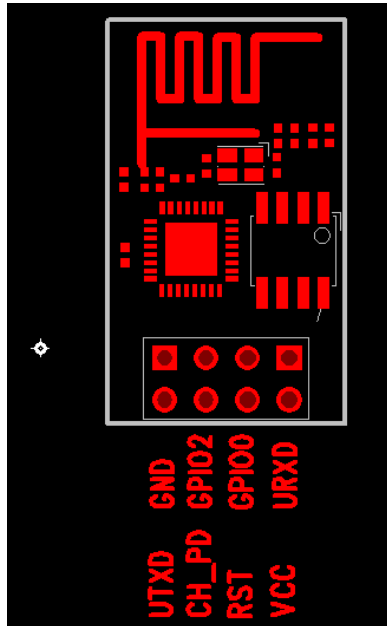


## 一 接线

**注意：本系统是 3.3V！请勿接 5V！**

新版(全 IO 口引出版):

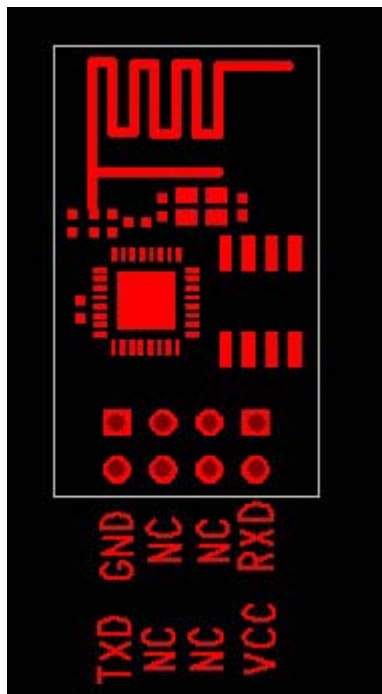


此版本若想从 FLASH 启动进入 AT 系统，只需 CH-PD 引脚接 VCC 或**接上拉**（不接上拉的情况下，串口可能无数据），其余三个引脚可选择**悬空**或接 VCC(但群里 (@云海之梦) 反映接 VCC 后导致无法正常启动)

（GPIO0 为高电平代表从 FLASH 启动，GPIO0 为低电平代表进入系统升级状态，此时可以经过串口升级内部固件 RST（GPIO16）可做外部硬件复位使用）

测试系统不同，接线方法也选择多多，请各位根据自己的情况进行选择，推荐接法：在 CH-PD 和 VCC 之间焊接电阻后，将 UTXD,GND,VCC,URXD 连上 USB-TTL(两者的 TXD 和 RXD 交叉接)即可进行测试

旧版(AT 指令版):



该版本**中间四根线**悬空即可。

群里 (@ZR(小 R)) 反应：接线正确的情况下，连上 USB 转 TTL，红灯蓝灯闪烁一下以后全灭，此现象为模块电流过大，超过了 USB 转 TLL 模块的承受范围。解决办法：1. CH-PD 换一个阻值更大的上拉电阻；2.换个承受电流大一点 USB 转 TLL 模块(群众们也可以自己改造)

（@ZR(小 R)）反应：在供电不足的情况下，收到的响应为乱码。请各位检查下供电电压是否足够。

## 二 上电：正常工作验证

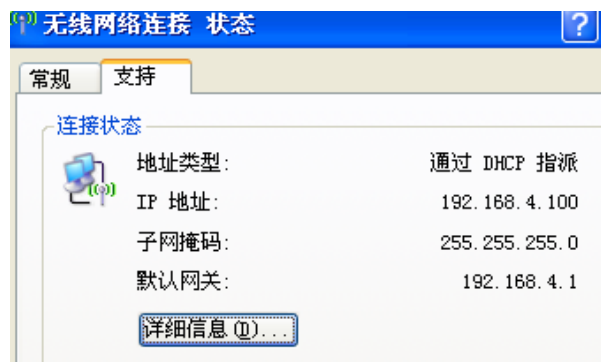
本模块可以工作在三种模式：1. STA 2. AP 3. AP+STA，出厂设置为第三种

上电后，蓝色灯微弱闪烁后熄灭，红灯长亮

1. 搜索无线网络，可见 ESP\_XXXXXX 已经处于列表中(后面的数字是 MAC 地址后几位)



连接该网络以后，查看连接状态：



手机搜索该网络，也可连接上：

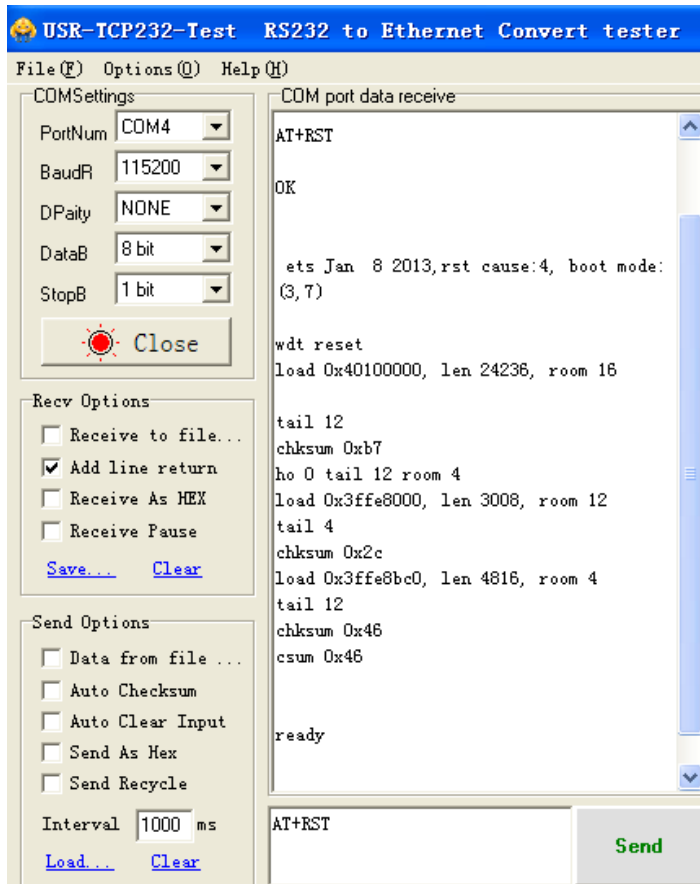


2. 使用 USR-TCP232-Test.exe 进行测试：

软件配置如下图：

注意：(1)波特率一般在出厂情况下默认的是 115200。如果在 115200 情况下收到的是乱码可以试试其他波特率（旧版本的默认波特率可能是其他数值）。

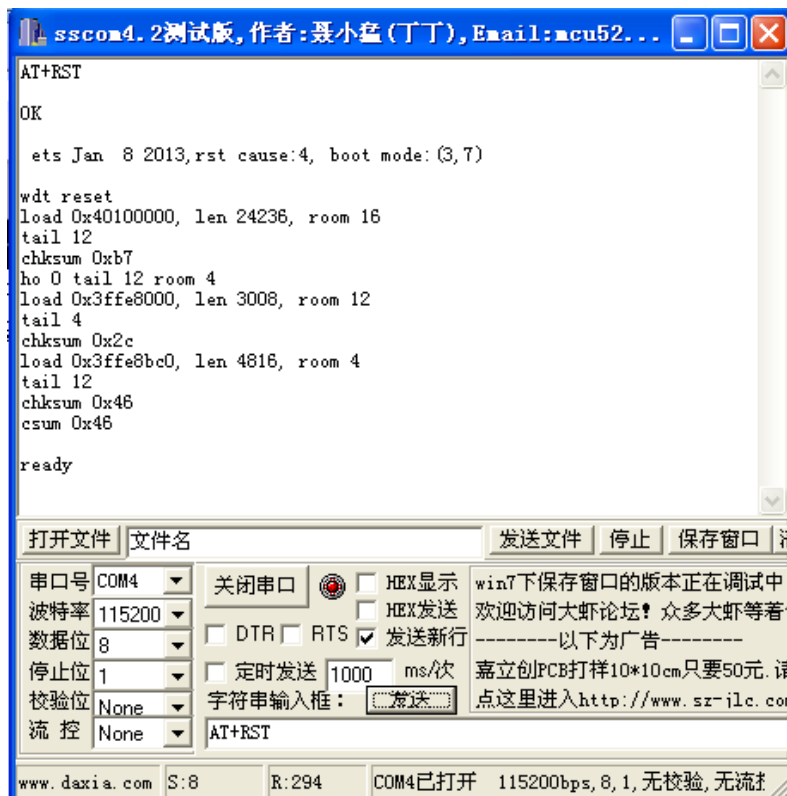
**(2)在输入命令后必须再按一下回车键，然后再按发送！**



### 3. 使用 sscom42.exe 进行测试

**注意：勾选上“发送新行”**

发送命令 **AT+RST** (重启模块),返回如下:



### 三 测试

测试前先阅读: (1)Espressif IoT AT 指令集\_v0.1.5.pdf  
<http://www.electrodragon.com/w/Wi07c>  
(2) ESP8266 测试方法汇总.pdf  
测试过程参照: ESP8266 作为 TCP 服务器端使用心得.pdf

描述了该模块支持的 AT 命令  
该网页命令描述更清晰一些  
测试的具体流程  
出错的原因分析及处理

## 四 测试例程

原来推荐的测试例程比较粗略，这里放上我自己的测试过程，理解可能有误，仅供参考。  
**请勿直接复制指令，部分格式经过 WORD 编辑后有误！可能导致出错！**

### (一) AP 模式

#### 1. 建立 AP

##### (1)重启模块

发送命令：AT+RST(执行指令)

指令：AT+RST

响应：OK

##### (2)设置模块

发送命令：AT+CWMODE=3 或 AT+CWMODE=2(设置指令)

指令：AT+CWMODE=<mode>

说明：<mode>:1-Station模式，2-AP模式，3-AP兼Station模式

响应：OK

说明：需重启后生效(AT+RST)

```
AT+CWMODE=3
no change

AT+CWMODE=2

OK

AT+RST

OK
```

##### (3)配置 AP 参数

发送命令：AT+CWSAP="TEST","123456123456",1,3(设置指令)

指令：AT+ CWSAP= <ssid>,<pwd>,<chl>,<ecn>

说明：指令只有在AP模式开启后有效

<ssid>:字符串参数，接入点名称

<pwd>:字符串参数，密码最长64字节，ASCII

<chl>:通道号

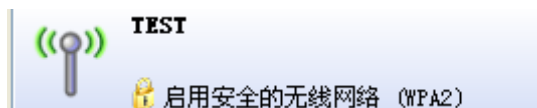
<ecn>:0-OPEN，1-WEP，2-WPA\_PSK，3-WPA2\_PSK，4-WPA\_WPA2\_PSK

响应：OK

```
AT+CWSAP="TEST","123456123456",1,3

OK
```

刷新无线网络列表，可见到 SSID 为 TEST 的无线网络列于其中：



**注意：此时连接网络会可能出现连接不上的情况，请发送 AT+RST 命令并等待几分钟之后再连接**

(4)查看已接入设备的 IP

连接上 TEST 后，发送命令：AT+CWLIF(执行指令)

指令：AT+CWLIF

说明：查看已接入设备的 IP

响应：<ip addr>

OK

说明：<ip addr>:已接入设备的 IP 地址

```
AT+CWLIF
192.168.4.100
OK
```

如果返回命令如下：

```
AT+CWLIF
OK
```

表示网络成功建立，目前无设备连入。  
如果无线网络实际已连接上，请等待几分钟后再发送 AT+CWLIF 命令进行查询。

(4)查询本机 IP 地址

发送命令：AT+CIFSR(执行指令)

指令：AT+CIFSR

说明：查看本模块的 IP 地址

注意：AP 模式下无效！会造成死机现象！

响应：<ip addr>

说明：<ip addr>:本模块 IP 地址

```
AT+CIFSR
192.168.4.1
```

但实际测试中并未发生死机现象。

查看本机配置模式：CWMODE=2，为 AP 模式。

```
AT+CWMODE?
+CWMODE:2
OK
```

发送命令：AT+CIFSR=?(测试指令)，返回响应如下：

```
AT+CIFSR=?
OK
```

同样未出现死机现象。

## 2. Server 方法收发

(0)查询此时模块状态(该步骤可省略)

发送命令 AT+CWMODE?(查询指令)

指令：AT+CWMODE?

说明：查看本模块的 WIFI 应用模式

响应：+CWMODE:<mode>

OK

说明: <mode>:1-Station 模式, 2-AP 模式, 3-AP 兼 Station 模式

```
AT+CWMODE?  
+CWMODE:1  
OK
```

发送命令 AT+CIPMUX? (查询指令)

指令: AT+CIPMUX?

说明: 查询本模块是否建立多连接

响应: + CIPMUX:<mode>

OK

说明: <mode>:0-单路连接模式, 1-多路连接模式

```
AT+CIPMODE?  
+CIPMODE:0  
OK
```

发送命令 AT+CIPMODE? (查询指令)

指令: AT+CIPMODE?

说明: 查询本模块的传输模式

响应: + CIPMODE:<mode>

OK

说明: <mode>:0-非透传模式, 1-透传模式

```
AT+CIPSTO?  
+CIPSTO:0  
OK
```

发送命令 AT+CIPSTO? (查询指令)

指令: AT+CIPSTO?

说明: 查询本模块的服务器超时时间

响应: + CIPSTO:<time>

OK

说明: <time>:服务器超时时间, 0~2880, 单位为 s

```
AT+CIPSTO?  
+CIPSTO:180  
OK
```

#### (1)开启多连接模式

发送命令: AT+CIPMUX=1(设置指令)

指令: AT+CIPMUX=<mode>

说明: <mode>:0-单路连接模式, 1-多路连接模式

响应: OK

```
AT+CIPMUX=1  
OK
```

查询可知，设置成功

```
AT+CIPMUX?  
+CIPMUX:1  
OK
```

## (2)创建服务器

**发送命令：AT+CIPSERVER=1,8080(设置指令)**

指令：AT+CIPSERVER=<mode>[,<port>]

说明：<mode>:0-关闭 server 模式，1-开启 server 模式  
<port>:端口号，缺省值为 333

响应：OK

说明：(1) AT+ CIPMUX=1 时才能开启服务器；关闭 server 模式需要重启

(2)开启 server 后自动建立 server 监听,当有 client 接入会自动按顺序占用一个连接。

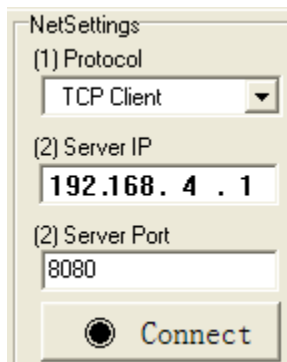
开启 server 服务如下图所示：

```
AT+CIPSERVER=1,8080  
OK
```

关闭 server 服务如下图所示：

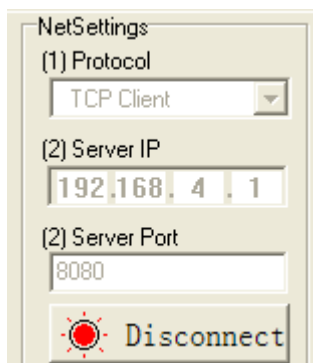
```
AT+CIPSERVER=0  
we must restart  
AT+RST  
OK
```

打开 USR-TCP232-Test.exe，点击 Connect 按钮连接不上，可知 server 服务未开启



重新开启 server 服务(注意：之前需要再发送一遍 AT+CIPMUX=1 以重新开启多连接模式)。

点击 Connect 按钮





连接成功后，串口收到模块返回的数据串：Link

180S(默认值)后，连接自动断开，返回 Unlink,客户端的相应按钮变成灰色。

全过程如下图：

```
AT+CIPMUX=1
OK
AT+CIPSERVER=1,8080
OK
Link
Unlink
```

### (3) 设置服务器超时时间

发送命令 **AT+CIPSTO=2880** (设置指令)

指令：AT+CIPSTO=<time>

说明：<time>:服务器超时时间，0~2880，单位为 s

响应：OK

```
AT+CIPSTO=2880
OK
```

### (4) 建立客户端



### (5) 查看当前连接

发送命令 **AT+CIPSTATUS** (执行指令)

指令：AT+CIPSTATUS

响应：STATUS:<stat>

+ CIPSTATUS:<id>,<type>,<addr>,<port>,<tetype>

OK

说明: <id>:连接的 id 号 0-4

<type>:字符串参数, 类型 TCP 或 UDP

<addr>:字符串参数, IP 地址

<port>:端口号

<tetype>: 0-本模块做 client 的连接, 1-本模块做 server 的连接

```
AT+CIPSTATUS
STATUS:3
+CIPSTATUS:0,"TCP","192.168.4.100",3195,1
+CIPSTATUS:1,"TCP","192.168.4.100",3792,1
OK
```

(之前电脑的 USB 无线网卡发烫厉害, 拔出后重新插上, 再次连接后自动分配的端口产生了变化, 所以刷出了两个客户端, 实际中存在的仅有 ID=1 的客户端)

#### (6) 向某个连接发送数据

**发送命令 AT+CIPSEND=1,6(设置指令)** (通过上一条指令 AT+CIPSTATUS 得知 ID=1)

指令: 1)单路连接时(+CIPMUX=0), 指令为: AT+CIPSEND=<length>

2)多路连接时(+CIPMUX=1), 指令为: AT+CIPSEND=<id>,<length>

响应: 收到此命令后先换行返回">", 然后开始接收串口数据

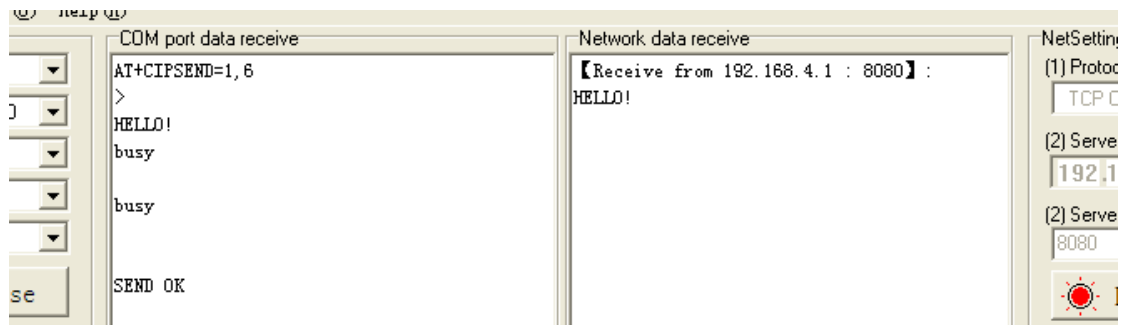
当数据长度满 length 时发送数据。

如果未建立连接或连接被断开, 返回 ERROR

如果数据发送成功, 返回 SEND OK

说明: <id>:需要用于传输连接的 id 号

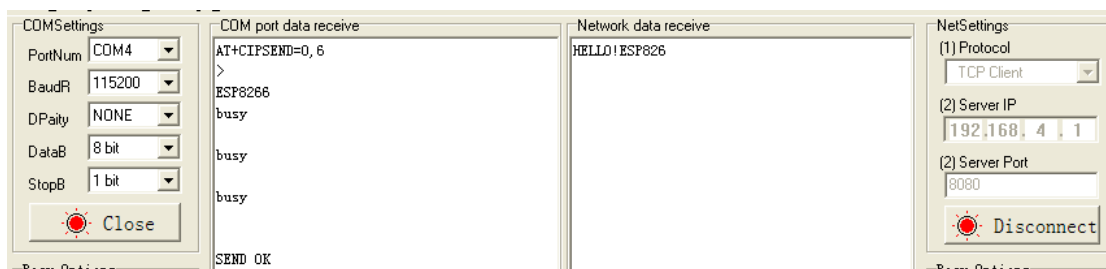
<length>:数字参数, 表明发送数据的长度, 最大长度为 2048



断开客户端后再发送一次, 得到以下响应:

```
HELLO!
Error
```

发送数据长度大于 LENGTH 时响应如下(HELLO!为之前一次发送的数据):



此时连接已建立，可以进行数据的双向收发。  
用客户端向 ESP8266 发送数据，正常。

```
+IPD,0,5:happy
OK

+IPD,0,4:测试
OK

+IPD,0,22:多次测试数据发送和接收
OK
```

### 3. Client 方法收发

(0) 关闭 server 服务 (如果没有开启 server 服务，可免除此步骤)

发送命令: **AT+CIPSERVER=0**(设置指令)

指令: **AT+CIPSERVER=<mode>[,<port>]**

说明: <mode>:0-关闭 server 模式, 1-开启 server 模式  
<port>:端口号, 缺省值为 333

响应: OK

说明: (1) **AT+ CIPMUX=1** 时才能开启服务器; 关闭 server 模式需要重启

(2)开启 server 后自动建立 server 监听,当有 client 接入会自动按顺序占用一个连接。

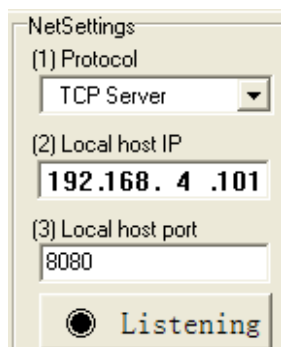
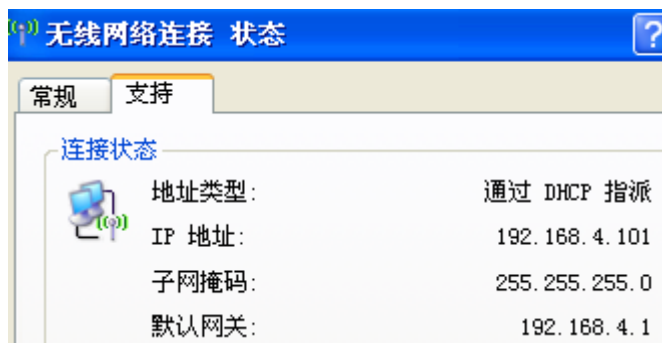
关闭 server 服务如下图所示:

```
AT+CIPSERVER=0
we must restart

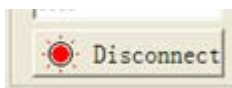
AT+RST

OK
```

(1) 创建服务器



点击 Listening，创建成功后，该按钮变为：



## (2) 开启多连接模式

**发送命令：AT+CIPMUX=1(设置指令)**

指令：AT+CIPMUX=<mode>

说明：<mode>:0-单路连接模式，1-多路连接模式

响应：OK

```
AT+CIPMUX=1
OK
```

## (3) 建立 TCP 连接

**发送命令 AT+CIPSTART=2,"TCP","192.168.4.101",8080 (设置指令)**

指令：1)单路连接时(+CIPMUX=0)，指令为：AT+CIPSTART= <type>,<addr>,<port>

2)多路连接时(+CIPMUX=1)，指令为：AT+CIPSTART=<id>,<type>,<addr>,<port>

响应：如果格式正确且连接成功，返回 OK，否则返回 ERROR

如果连接已经存在，返回 ALREAY CONNECT

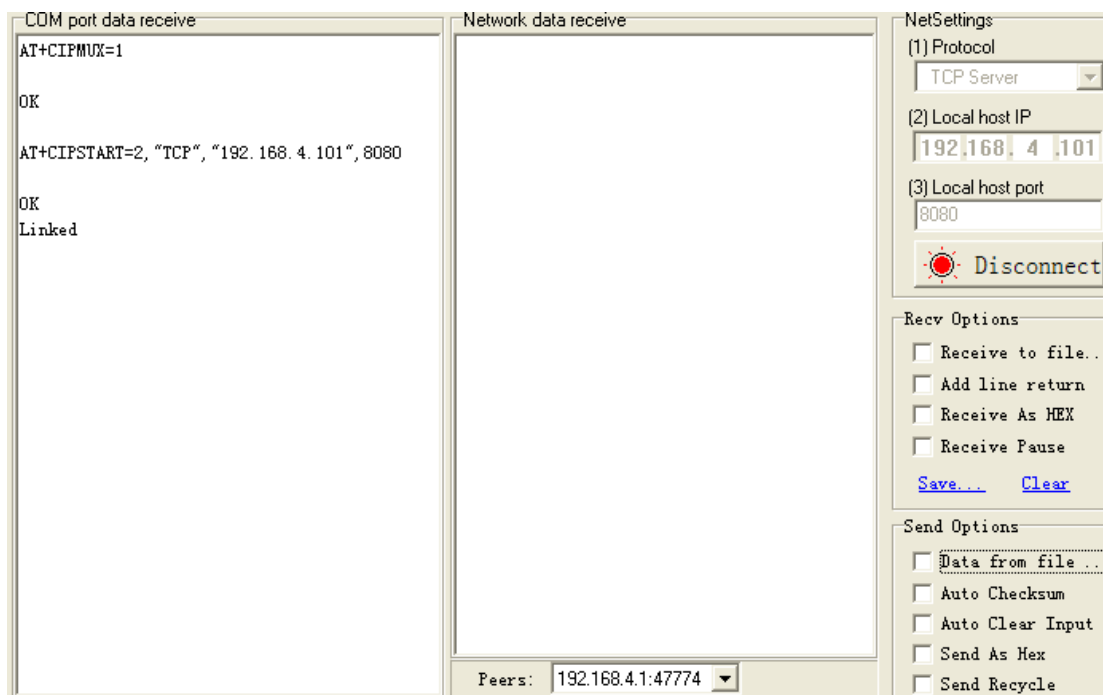
说明：<id>:0-4，连接的 id 号

<type>:字符串参数，表明连接类型，“TCP”-建立 tcp 连接，“UDP”-建立 UDP 连接

<addr>:字符串参数，远程服务器 IP 地址

<port>:远程服务器端口号

如下图所示：



## (4) 向服务器发送数据

**发送命令 AT+CIPSEND=2,8(设置指令)** （通过上一条指令 AT+CIPSTART 设置为 ID=2）

指令：1)单路连接时(+CIPMUX=0)，指令为：AT+CIPSEND=<length>

2)多路连接时(+CIPMUX=1)，指令为：AT+CIPSEND= <id>,<length>

响应：收到此命令后先换行返回">", 然后开始接收串口数据

当数据长度满 length 时发送数据。

如果未建立连接或连接被断开，返回 ERROR

如果数据发送成功，返回 SEND OK

说明：<id>:需要用于传输连接的 id 号

<length>:数字参数，表明发送数据的长度，最大长度为 2048

The screenshot shows a software interface for network communication. It is divided into several sections:

- COM port data receive:** Displays the AT command sequence: `AT+CIPSTART=2,"TCP","192.168.4.101",8080`, `OK`, `Linked`, `AT+CIPSEND=2,8`, `>`, `how are y`, `busy`, `o`, `busy`, `u`, `busy`, `busy`, `busy`, `SEND OK`.
- Network data receive:** Shows the received data: `【Receive from 192.168.4.1 : 40652】 :` followed by `how are`.
- NetSettings:** Includes a dropdown for Protocol (TCP Server), input fields for Local host IP (192.168.4.101) and Local host port (8080), and a Disconnect button.
- Recv Options:** A list of checkboxes for `Receive to file...`, `Add line return`, `Receive As HEX`, and `Receive Pause`, with Save and Clear buttons.
- Send Options:** A list of checkboxes for `Data from file...`, `Auto Checksum`, `Auto Clear Input`, `Send As Hex`, and `Send Recycle`.
- Interval:** A field set to 1000 ms.
- Peers:** A dropdown menu showing 192.168.4.1:40652.
- Send Buffer:** At the bottom, it shows `how are you` in a text field with a Send button.

接收正确，超过 8 个字符的部分不接收。

此时连接已建立，可以进行数据的双向收发。用服务器向 ESP8266 发送数据，正常。

```
+IPD,2,9:try again
OK

+IPD,2,10:happy hour
OK
```

## (二) STA 模式

### 1. 建立 STA

#### (1) 重启模块

发送命令: **AT+RST**(执行指令)

指令: **AT+RST**

响应: **OK**

```
AT+RST
OK

ets Jan 8 2013,rst cause:4, boot mode:
(3,7)

wdt reset

load 0x40100000, len 24236, room 16
tail 12
chksum 0xb7
ho 0 tail 12 room 4
load 0x3ffe8000, len 3008, room 12
tail 4
chksum 0x2c
load 0x3ffe8bc0, len 4816, room 4
tail 12
chksum 0x46
csum 0x46

ready
```

#### (2) 设置模块

发送命令: **AT+CWMODE=3** 或 **AT+CWMODE=1**(设置指令)

指令: **AT+CWMODE=<mode>**

说明: <mode>:1-Station模式, 2-AP模式, 3-AP兼Station模式

响应: **OK**

说明: 需重启后生效(**AT+RST**)

```
AT+CWMODE=1
OK

AT+RST
OK
```

将模块设置为模式 1,此时刷新网络列表可见由 ESP8266 建立的网络从列表中消失了

#### (3) 查看当前无线路由器列表

发送命令: **AT+CWLAP**(执行指令)

指令: **AT+CWLAP**

响应: 正确: (终端返回AP列表)

+ CWLAP: <ecn>,<ssid>,<rssi>

OK

错误: ERROR

说明: <ecn>:0-OPEN, 1-WEP, 2-WPA\_PSK, 3-WPA2\_PSK, 4-WPA\_WPA2\_PSK

<ssid>:字符串参数, 接入点名称

<rssi>:信号强度

```
AT+CWLAP
+CWLAP: (0, "", 0)
+CWLAP: (0, "CMCC-FREEGAME", -82)
+CWLAP: (4, "shiningwuxi", -91)
+CWLAP: (4, "TP-LINK_shr", -72)
+CWLAP: (0, "CMCC", -85)
+CWLAP: (1, "TP-LINK_lq", -79)

+CWLAP: (2, "ChinaNet-emrG", -53)
+CWLAP: (2, "iTV-emrG", -55)
+CWLAP: (4, "908", -89)
+CWLAP: (4, "AFD", -85)
+CWLAP: (4, "MERSAIN", -55)
+CWLAP: (4, "FAST_DACD2C", -94)
+CWLAP: (0, "CMCC-AUTO", -85)
+CWLAP: (2, "Tenda_330170", -83)
+CWLAP: (0, "CMCC-FREEGAME", -85)
+CWLAP: (2, "ChinaNet-P9Gt", -76)
+CWLAP: (4, "LINKSYS", -91)
+CWLAP: (4, "maxhome", -89)
+CWLAP: (4, "TP-LINK_FC566A", -75)
+CWLAP: (4, "tfdy", -88)
+CWLAP: (4, "liu", -91)

OK
```

如搜索不到信号, 响应:

```
AT+CWLAP
ERROR
```

请重新上电并严格按照 AT 命令再发送一遍。

没有列表返回的时候请耐心等待, 否则会出现“busy”

#### (4) 加入当前无线网络

**发送命令: AT+CWJAP="MERSAIN","XXXXXXXX"(设置指令)**

**指令: AT+CWJAP=<ssid>,<pwd>**

说明: <ssid>:字符串参数, 接入点名称

<pwd>:字符串参数, 密码, 最长64字节ASCII

响应: 正确: OK

错误: ERROR

```
AT+CWJAP="MERSAIN", "██████████"
OK
```

检测是否真的连上该路线网络

### 发送命令：AT+CWJAP?(查询指令)

指令：AT+CWJAP?

响应：返回当前选择的AP  
+ CWJAP:<ssid>

OK

说明：<ssid>:字符串参数，接入点名称

```
AT+CWJAP?  
+CWJAP:"MERSAIN"  
OK
```

**断电。**上电后发送 AT+CWJAP?，返回同上，系统保持上次状态。

查看模块 IP 地址

### 发送命令：AT+CIFSR(执行指令)

指令：AT+CIFSR

响应：正确：+CIFSR:<IP address>

OK

错误：ERROR

说明：<ssid>:字符串参数，接入点名称

```
AT+CIFSR  
192.168.1.102
```

**群里 (@云海之梦) 反映：错误的 SSID 也可以连接上**，测试后发现果然如此，输入实际不存在的账号密码返回也是 OK

```
ready  
AT+CWJAP="JOKER", "111111"  
OK  
AT+CWJAP?  
+CWJAP:"JOKER"  
OK  
AT+CIFSR  
ERROR
```

但发送 AT+CIFSR(查看模块地址)，返回 ERROR，可通过此命令来判断有没有真正连接上。

## 2. Server 方法收发

(0) 查询此时模块状态(该步骤可省略)

发送命令 AT+CWMODE?(查询指令)

指令：AT+CWMODE?

说明：查看本模块的 WIFI 应用模式

响应：+CWMODE:<mode>

OK

说明：<mode>:1-Station 模式，2-AP 模式，3-AP 兼 Station 模式

```
AT+CWMODE?  
+CWMODE:2  
OK
```



发送命令 AT+CIPMUX? (查询指令)

指令: AT+CIPMUX?

说明: 查询本模块是否建立多连接

响应: + CIPMUX:<mode>

OK

说明: <mode>:0-单路连接模式, 1-多路连接模式

```
AT+CIPMUX?  
+CIPMUX:0  
  
OK
```

发送命令 AT+CIPMODE? (查询指令)

指令: AT+CIPMODE?

说明: 查询本模块的传输模式

响应: + CIPMODE:<mode>

OK

说明: <mode>:0-非透传模式, 1-透传模式

```
AT+CIPMODE?  
+CIPMODE:0  
  
OK
```

发送命令 AT+CIPSTO? (查询指令)

指令: AT+CIPSTO?

说明: 查询本模块的服务器超时时间

响应: + CIPSTO:<time>

OK

说明: <time>:服务器超时时间, 0~2880, 单位为 s

```
AT+CIPSTO?  
+CIPSTO:180  
  
OK
```

#### (1)开启多连接模式

**发送命令: AT+CIPMUX=1(设置指令)**

指令: AT+CIPMUX=<mode>

说明: <mode>:0-单路连接模式, 1-多路连接模式

响应: OK

```
AT+CIPMUX=1  
  
OK
```

查询可知, 设置成功

```
AT+CIPMUX?  
+CIPMUX:1  
  
OK
```

## (2)创建服务器

发送命令: **AT+CIPSERVER=1,8080(设置指令)**

指令: **AT+CIPSERVER=<mode>[,<port>]**

说明: <mode>:0-关闭 server 模式, 1-开启 server 模式  
<port>:端口号, 缺省值为 333

响应: OK

说明: (1) **AT+CIPMUX=1** 时才能开启服务器; 关闭 server 模式需要重启

(2)开启 server 后自动建立 server 监听,当有 client 接入会自动按顺序占用一个连接。

开启 server 服务如下图所示:

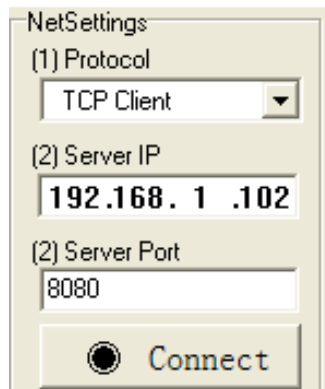
```
AT+CIPSERVER=1,8080
OK
```

关闭 server 服务如下图所示:

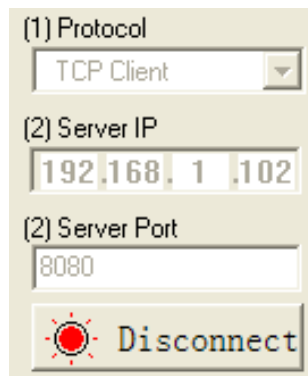
```
AT+CIPSERVER=0
we must restart

AT+RST
OK
```

打开 USR-TCP232-Test.exe, 点击 Connect 按钮连接不上, 可知 server 服务未开启



重新开启 server 服务(注意: 之前需要再发送一遍 **AT+CIPMUX=1** 以重新开启多连接模式)。点击 Connect 按钮



连接成功后, 串口收到模块返回的数据串: Link

180S(默认值)后, 连接自动断开, 返回 Unlink,客户端的相应按钮变成灰色。

全过程如下图：

```
AT+CIPMUX=1
OK
AT+CIPSERVER=1,8080
OK
Link
Unlink
```

(3) 设置服务器超时时间

发送命令 **AT+CIPSTO=2880** (设置指令)

指令：AT+CIPSTO=<time>

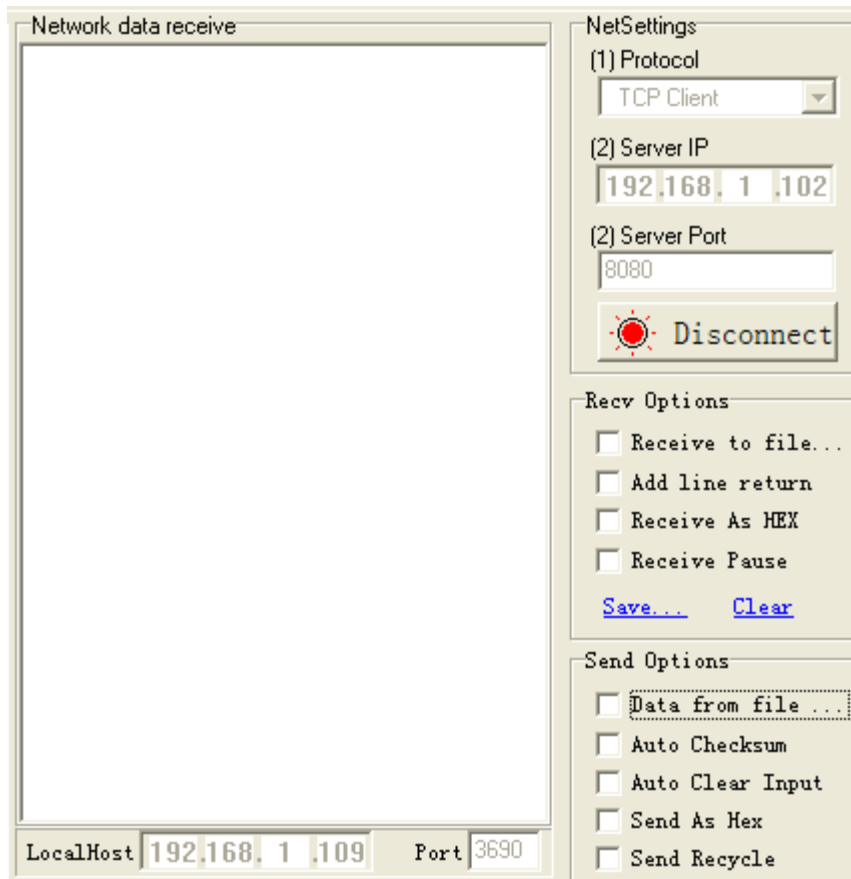
说明：<time>:服务器超时时间，0~2880，单位为 s

响应：OK

```
AT+CIPSTO=2880
OK
```

(4) 建立客户端

界面设置如下图所示：



(5) 查看当前连接

发送命令 **AT+CIPSTATUS** (执行指令)

指令: AT+CIPSTATUS

响应: STATUS:<stat>

+ CIPSTATUS:<id>,<type>,<addr>,<port>,<tetype>

OK

说明: <id>:连接的 id 号 0-4

<type>:字符串参数, 类型 TCP 或 UDP

<addr>:字符串参数, IP 地址

<port>:端口号

<tetype>: 0-本模块做 client 的连接, 1-本模块做 server 的连接

```
AT+CIPSTATUS
STATUS:3
+CIPSTATUS:0,"TCP","192.168.1.109",3690,
1
OK
```

#### (6) 向某个连接发送数据

**发送命令 AT+CIPSEND=0,10(设置指令)** (通过上一条指令 AT+CIPSTATUS 得知 ID=0)

指令: 1)单路连接时(+CIPMUX=0), 指令为: AT+CIPSEND=<length>

2)多路连接时(+CIPMUX=1), 指令为: AT+CIPSEND=<id>,<length>

响应: 收到此命令后先换行返回">", 然后开始接收串口数据

当数据长度满 length 时发送数据。

如果未建立连接或连接被断开, 返回 ERROR

如果数据发送成功, 返回 SEND OK

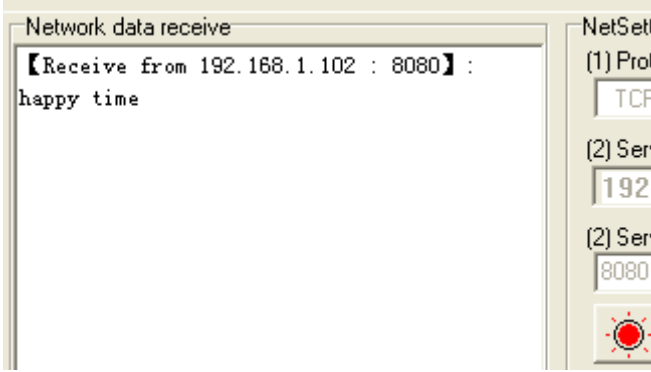
说明: <id>:需要用于传输连接的 id 号

<length>:数字参数, 表明发送数据的长度, 最大长度为 2048

```
AT+CIPSEND=0,10
>
happy time
busy

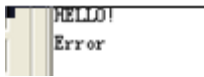
busy
发: happy time
```

收:



The screenshot shows a window titled "Network data receive" with a text area containing "【Receive from 192.168.1.102 : 8080】 : happy time". To the right, there is a "NetSet" panel with three entries: (1) Protocol set to "TCP", (2) Server IP set to "192", and (3) Server Port set to "8080". A red status indicator is visible at the bottom of the panel.

断开客户端后再发送一次，得到以下响应：



此时连接已建立，可以进行数据的双向收发。

### 3. Client 方法收发

(0) 关闭 server 服务 (如果没有开启 server 服务，可免除此步骤)

发送命令：**AT+CIPSERVER=0(设置指令)**

指令：**AT+CIPSERVER=<mode>[,<port>]**

说明：<mode>:0-关闭 server 模式，1-开启 server 模式

<port>:端口号，缺省值为 333

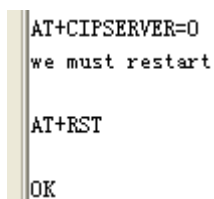
响应：OK

说明：(1) AT+CIPMUX=1 时才能开启服务器；关闭 server 模式需要重启

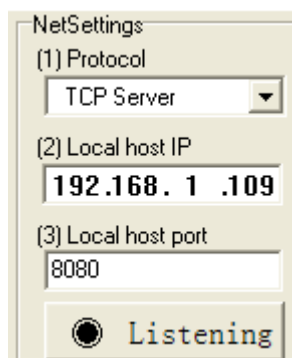
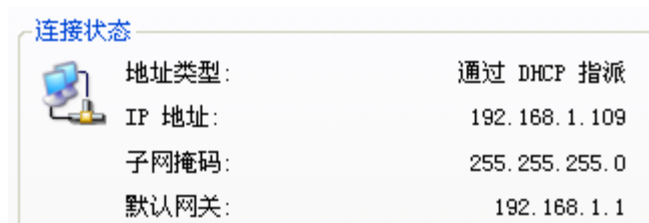
(2) 开启 server 后自动建立 server 监听, 当有 client 接入会自动按顺序占用一个连

接。

关闭 server 服务如下图所示：



(1) 创建服务器



点击 Listening，创建成功后，该按钮变为：



(2) 开启多连接模式

发送命令：**AT+CIPMUX=1(设置指令)**

指令：**AT+CIPMUX=<mode>**

说明：<mode>:0-单路连接模式，1-多路连接模式

响应: OK

```
AT+CIPMUX=1
OK
```

### (3) 建立 TCP 连接

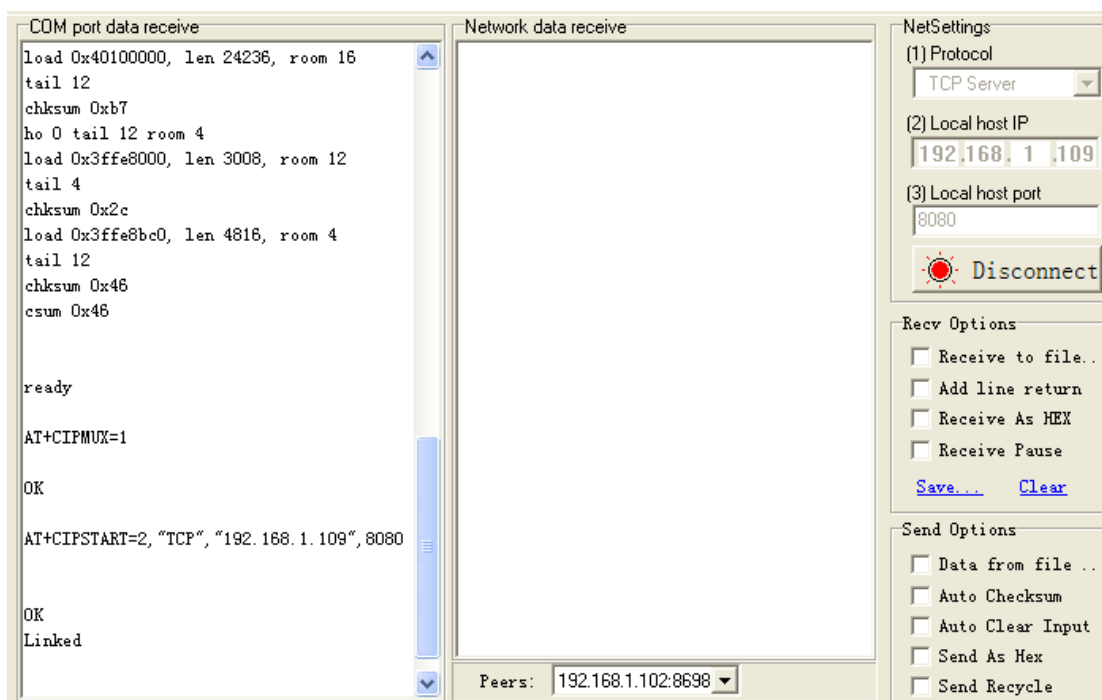
**发送命令 AT+CIPSTART=2,"TCP","192.168.1.109",8080 (设置指令)**

指令: 1)单路连接时(+CIPMUX=0), 指令为: AT+CIPSTART=<type>,<addr>,<port>  
2)多路连接时(+CIPMUX=1), 指令为: AT+CIPSTART=<id>,<type>,<addr>,<port>

响应: 如果格式正确且连接成功, 返回 OK, 否则返回 ERROR  
如果连接已经存在, 返回 ALREAY CONNECT

说明: <id>:0-4, 连接的 id 号  
<type>:字符串参数, 表明连接类型, "TCP"-建立 tcp 连接, "UDP"-建立 UDP 连接  
<addr>:字符串参数, 远程服务器 IP 地址  
<port>:远程服务器端口号

如下图所示:



### (4) 向服务器发送数据

**发送命令 AT+CIPSEND=2,10(设置指令)** (通过上一条指令 AT+CIPSTART 设置为 ID=2)

指令: 1)单路连接时(+CIPMUX=0), 指令为: AT+CIPSEND=<length>  
2)多路连接时(+CIPMUX=1), 指令为: AT+CIPSEND=<id>,<length>

响应: 收到此命令后先换行返回">", 然后开始接收串口数据  
当数据长度满 length 时发送数据。

如果未建立连接或连接被断开, 返回 ERROR

如果数据发送成功, 返回 SEND OK

说明: <id>:需要用于传输连接的 id 号  
<length>:数字参数, 表明发送数据的长度, 最大长度为 2048

The screenshot displays a software interface for configuring a network connection. It is divided into several sections:

- COM port data receive:** A text area showing the following sequence of messages:

```
ready
AT+CIPMUX=1
OK
AT+CIPSTART=2,"TCP","192.168.1.109",8080
OK
Linked
AT+CIPSEND=2,10
>
good luck!
busy
busy
SEND OK
```
- Network data receive:** A text area showing the received data:

```
【Receive from 192.168.1.102 : 23269】 :
good luck!
```
- NetSettings:** Configuration options for the network connection:
  - (1) Protocol: TCP Server
  - (2) Local host IP: 192.168.1.109
  - (3) Local host port: 8080
  - Disconnect button
- Recv Options:** Checkboxes for receiving data:
  - Receive to file...
  - Add line return
  - Receive As HEX
  - Receive Pause
  - Save... Clear
- Send Options:** Checkboxes for sending data:
  - Data from file...
  - Auto Checksum
  - Auto Clear Input
  - Send As Hex
  - Send Recycle
  - Interval: 1000 ms
- Peers:** A dropdown menu currently set to 192.168.1.102:8698.
- Send buttons:** Two 'Send' buttons are visible at the bottom, one of which is highlighted in green.

此时连接已建立，可以进行数据的双向收发。