

项目一	Wi-Fi 串口第 2 部分.....	2
2.1	添加 ser2net 应用程序.....	2
2.2	配置 Wi-Fi 串口.....	3
2.3	测试 Wi-Fi 串口.....	4
2.4	开机自启动 Wi-Fi 串口.....	10

www.f403tech.cn

项目一 Wi-Fi 串口第 2 部分

本项目目标

- 掌握 RT5350 串口 2 的使用方法
- 实现 Wi-Fi 串口

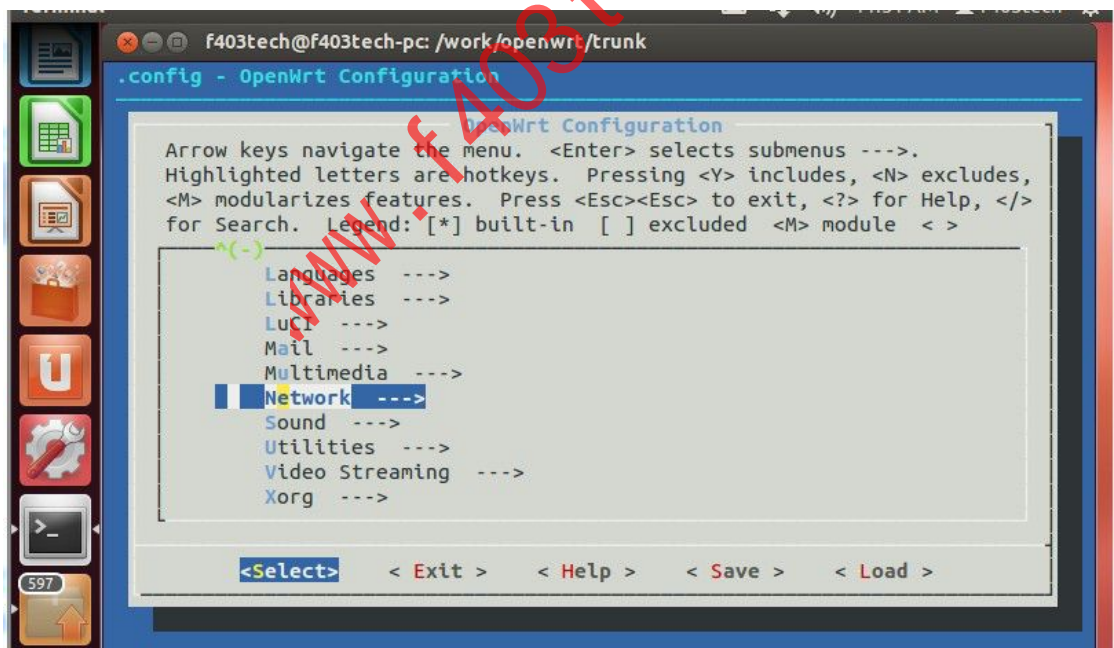
2.1 添加 ser2net 应用程序

在上一章中，我们成功的让驱动程序支持了串口 2，并且做了简单的测试。接下来，我们就为串口 2 添加一个应用程序，从而实现 Wi-Fi 串口。

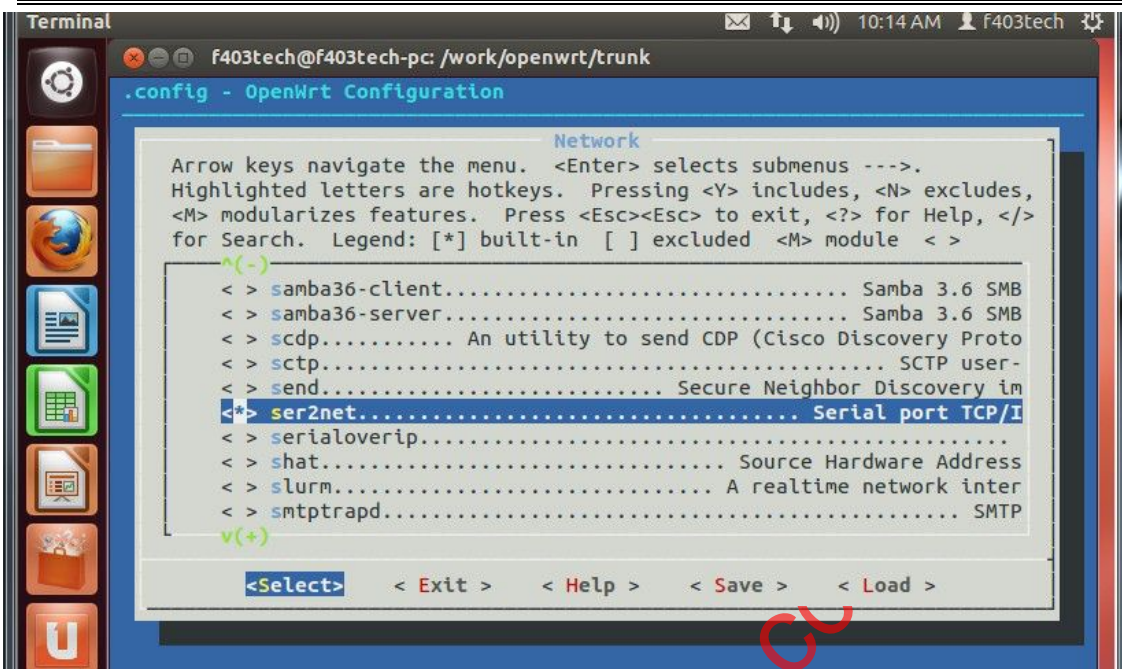
实现过程非常简单，首先进入 OpenWrt 系统源码，然后运行 `make menuconfig` 命令进行配置。

```
# cd /work/openwrt/trunk/  
# make menuconfig
```

在弹出的配置菜单中，首先进入 Network，如图所示。



然后选中 ser2net 这个选项，如果所示。



最后退出保存，再编译。

```
# make V=99
```

2.2 配置 Wi-Fi 串口

既然是 Wi-Fi 串口，那么我们就可以理解为它由两部分组成，一部分是网络编程器的 TCP 服务器，用于接收客户端通过 Wi-Fi 传来的数据；另一部分就是串口，用于将 TCP 服务器接收到的数据通过串口发送出去。

既然是 TCP 服务器，那么我们需要设置它的 IP 地址和端口号。

既然是串口，那么我们需要设置它的波特率等参数。

那么，这些参数，我们怎么来设置呢？

通过修改 ser2net.conf 这个配置文件来实现，位于开发板上的/etc 目录下。

将上面编译出来的固件，重新烧写到开发板上，然后上电启动，进入/etc 目录。

```
# cd /etc
```

```
# vi ser2net.conf
```

我们打开 ser2net.conf 这个配置文件，再它的最下面，我们可以看到以下这些内容。

```
2001:raw:600:/dev/ttyS0:9600 NONE 1STOPBIT 8DATABITS XONXOFF LOCAL -RTSCTS
#2002:raw:600:/dev/ttyS1:9600 NONE 1STOPBIT 8DATABITS XONXOFF LOCAL -RTSCTS
2003:raw:5:/dev/ttyS2:9600
2004:raw:5:/dev/ttyS3:115200
2005:raw:5:/dev/ttyS4:9600
2006:raw:5:/dev/ttyS5:9600
```

```

2007:raw:5:/dev/ttyS6:9600 tw=twl tr=trl
3001:telnet:0:/dev/ttyS0:19200 remctl banner1
3011:telnet:3:/dev/ttyS0:19200 banner2
#3002:telnet:0:/dev/ttyS1:9600
3003:telnet:0:/dev/ttyS2:9600 banner3
3003:telnet:0:/dev/ttyS2:9600 signature1
3004:telnet:0:/dev/ttyS3:115200
3005:telnet:0:/dev/ttyS4:9600
3006:telnet:0:/dev/ttyS5:9600 open1
3007:telnet:0:/dev/ttyS6:9600 close1
5001:rawlp:10:/dev/lp0

```

那么这些内容是什么意思呢？我们通过看 ser2net.conf 这个配置文件前面的说明注释部分，可以知道，这些参数的格式如下：

```
# <TCP port>:<state>:<timeout>:<device>:<options>
```

<TCP port>：用于设置 TCP 服务器要监听的端口号；

<state>：用于设置 TCP 服务器的状态；

<timeout>：用于设置超时时间，如果设置为 0，表示不开启超时时间；

<device>：用于选择使用哪个串口设备；

<options>：用于设置波特率、校验位等参数。

更多详细的介绍，就请大家看 ser2net.conf 这个配置文件前面的说明注释部分。

那么，如果我想让我们的 Wi-Fi 串口实现监听 8002 这个端口，然后通过串口 2 输出，波特率是 115200，应该怎么做呢？

如果我们看懂了前面的内容，就发现，要实现这个功能非常简单，只需要在 ser2net.conf 这个配置中的最后面添加一行以下内容即可。

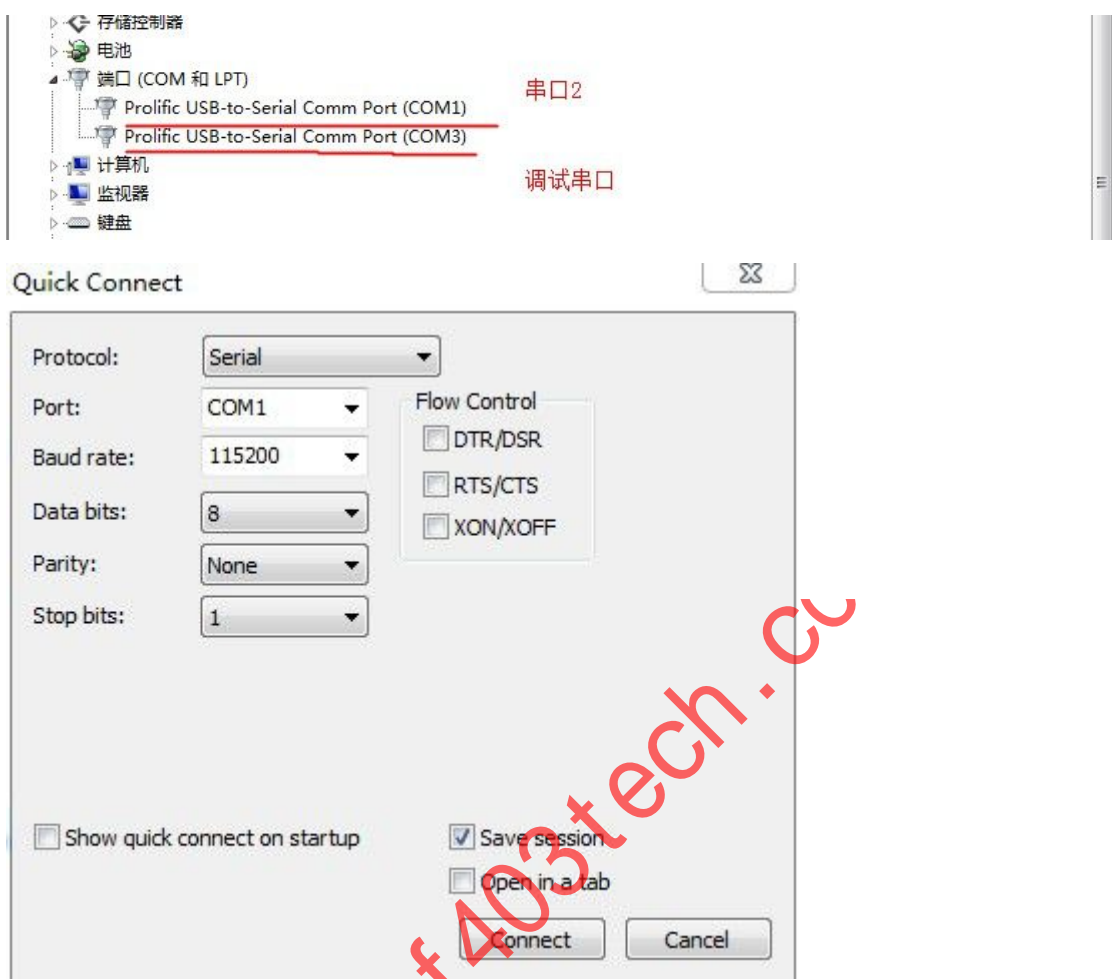
```
8002:raw:0:/dev/ttyS0:115200
```

最后保存退出，然后重启开发板，即可。

2.3 测试 Wi-Fi 串口

通过前面的那么多的努力，我们终于把 Wi-Fi 串口这个系统给做好了，接下来我们就来测试它。

1). 将开发板的串口 2 和电脑的串口连接起来，然后打开 SecureCRT，设置好相应的串口。



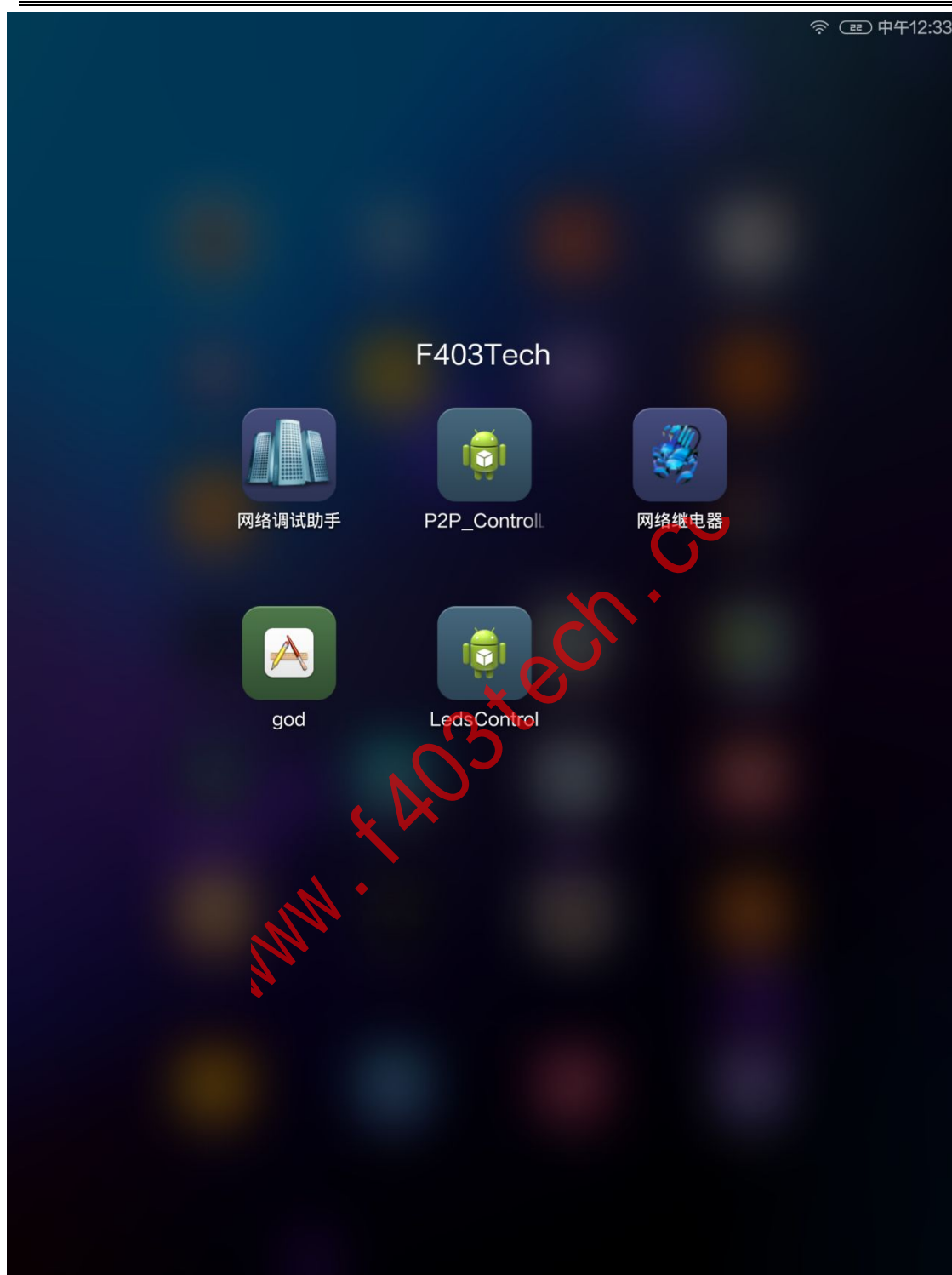
2). 启动开发。

3). 启动 ser2net 软件，只需要在调试串口测，输入如下命令即可。

```
# ser2net
```

4). 手机连接上开发板发出的 WiFi 信号 f403tech_ap。

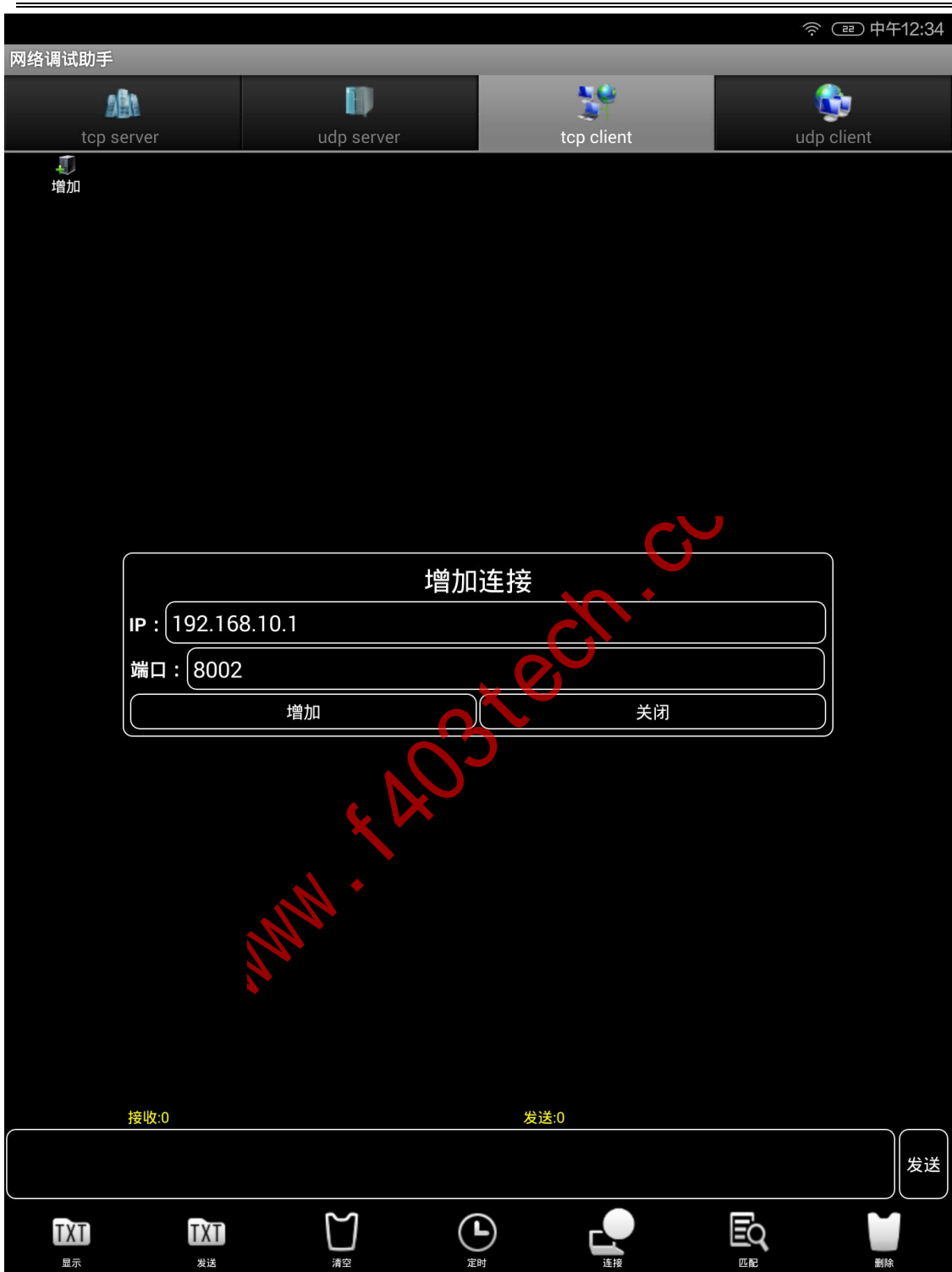
5). 启动“网络调试助手”软件。



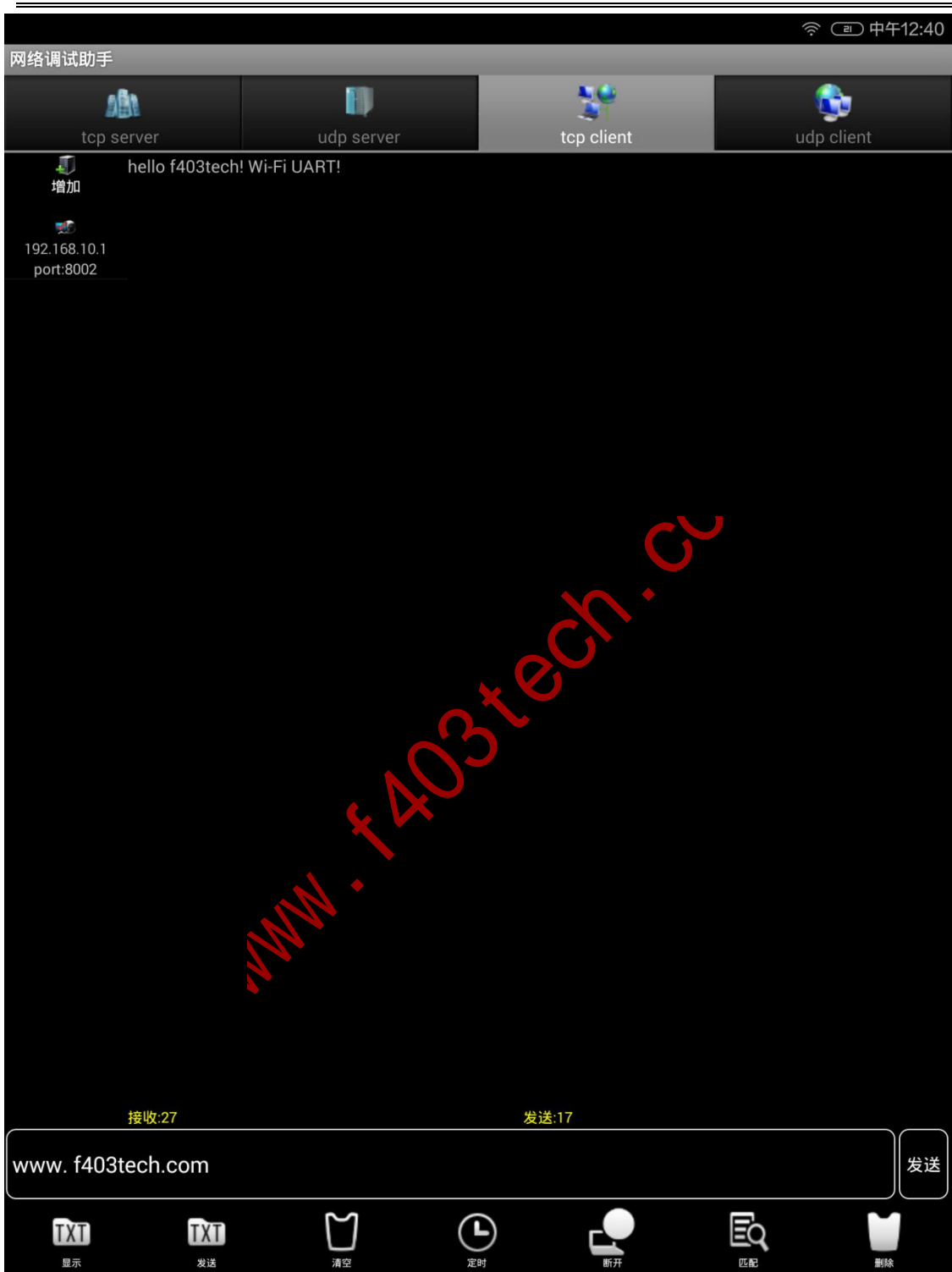
6). 选择 tcp client。



7). 点添加按钮。然后在弹出的对话框中填入开发板的 IP 地址，还有就是 Wi-Fi 串口的端口号 8002.



8). 接下来，我们的手机就可以通过 Wi-Fi 串口和电脑进行传数据了。



2.4 开机自启动 Wi-Fi 串口

我们的最终目的，肯定是要做出一个产品来，而产品，不可能还要用户通过调试串口来输入命令，从而才能启动 Wi-Fi 串口。

那么，接下来，我们就来实现开机自启动 Wi-Fi 串口。

首先打开开发板上的 /etc 目录下的 rc.local 这个文件。

```
# vi /etc/rc.local
```

然后在 exit0 前面加入一条 ser2net，详情如下。

```
# Put your custom commands here that should be executed once
# the system init finished. By default this file does nothing.

ser2net
exit 0
```

最后保存退出，重启开发板，然后通过 ps 命令查看一下，发现 ser2net 已经启动好了。

```
root@Openwrt:/#
root@Openwrt:/#
root@Openwrt:/#
root@Openwrt:/# ps
  PID USER      VSZ STAT COMMAND
    1 root        0 S    /sbin/procd
    2 root        0 SW   [kthreadd]
    3 root        0 SW   [ksoftirqd/0]
    4 root        0 SW   [kworker/0:0]
    5 root        0 SW<  [kworker/0:0H]
    6 root        0 SW   [kworker/u2:0]
    7 root        0 SW<  [khelper]
    8 root        0 SW   [kworker/u2:1]
   65 root        0 SW<  [writeback]
   67 root        0 SW<  [bioset]
   69 root        0 SW<  [kblockd]
   92 root        0 SW   [kworker/0:1]
   98 root        0 SW   [kswapd0]
  143 root        0 SW   [fsnotify_mark]
  169 root        0 SW   [sp132766]
  209 root        0 SW<  [deferwq]
  213 root        0 SW<  [kworker/0:1H]
  214 root        0 SW   [kworker/0:2]
  258 root        0 SW   [khubd]
  287 root        0 SWN   [jffs2_gcd_mtd5]
  337 root        888 S    /sbin/ubusd
  338 root       1480 S    /bin/ash --login
  571 root        0 SW<  [cfg80211]
  658 root       1316 S    /sbin/logd -s 16
  690 root       1536 S    /sbin/netifd
  718 root       1184 S    /usr/sbin/odhcpd
  756 root       1148 S    /usr/sbin/dropbear -F -P /var/run/dropbear.1.pid -p
  775 root       1472 S    /usr/sbin/telnetd -F -l /bin/login.sh
  789 root       1528 S    /usr/sbin/uhttpd -f -h /www -r Openwrt -x /cgi-bin -
  805 root        864 S    ser2net
  820 root       1480 S    /usr/sbin/ntpd -n -p 0.openwrt.pool.ntp.org -p 1.ope
  869 root       1480 S    udhcpd -p /var/run/udhcpd-eth0.2.pid -s /lib/netifd/
  962 root       1724 S    /usr/sbin/hostapd -P /var/run/wifi-phy0.pid -B /var/
  989 nobody     980 S    /usr/sbin/dnsmasq -C /var/etc/dnsmasq.conf -k
 1044 root       1476 R    ps
root@Openwrt:/#
root@Openwrt:/#
root@Openwrt:/#
```

这样，我们的 Wi-Fi 串口就这样做好了。

注意：

- 1). 该教程为我司(www.f403tech.com)原创教程，版权所有；
- 2). 该教程会不断更新、不断深入，详情请咨询我司客服；
- 3). 针对该教程，我们还有 QQ 群和论坛，专门负责技术答疑，详情请咨询我司客服。

www.f403tech.com