**串口连接BBB**  
使用usb线可以连接BBB和电脑，用ssh就可以登录BBB来进行操作。但有时候万一系统配置出现故障，或ssh用不了了，那就只能用串口连接了。首先要有一个串口转USB模块，用GND, TXD, RXD这三根线连接到板子上。具体连接方法见下图。  
  
理论上串口是随时都可以连接的，但我为了看到更全的输出信息，我在给BBB上电之前就先把它和电脑连好，在电脑端打开串口调试助手（用linux或mac的话可以打开终端的screen）然后用5V电源供电或者用usb连到电脑上供电。  
  
mac打开screen的命令：screen /dev/cu.usbserial 115200  
linux打开screen的命令：sudo screen /dev/ttyUSB0 115200  
  
供电以后，串口马上就会开始输出启动信息。过了一会就会出现一个Angstrom的字符画logo，并要求登录，输入用户名和密码（用户名是root，如果你没改过的话）回车就登录了。

1. .---O---.
2. |       |                  .-.           o o
3. |   |   |-----.-----.-----.| |   .----..-----.-----.
4. |       |     | \_\_  |  ---'| '--.|  .-'|     |     |
5. |   |   |  |  |     |---  ||  --'|  |  |  '  | | | |
6. '---'---'--'--'--.  |-----''----''--'  '-----'-'-'-'
7. -'  |
8. '---'
10. The Angstrom Distribution beaglebone ttyO0
12. Angstrom v2012.12 - Kernel 3.8.13
14. beaglebone login:

复制代码

启动完成以后，在这里可以像之前用ssh连接一样直接输入shell命令。可见串口是比usb+ssh更可靠的连接方式，但是因为串口的传输速度比usb线还是要慢很多的（你可以输入dmesg感受一下），所以如果能用ssh的话，一般我们还是用ssh。  
  
这里有一篇文章可以参考《用串口连接BBB的2.5种方法》[http://dave.cheney.net/2013/09/2 ... ur-beaglebone-black](http://dave.cheney.net/2013/09/22/two-point-five-ways-to-access-the-serial-console-on-your-beaglebone-black)  
  
**用usb联网**  
beaglebone如果用usb连接到了可以上网的电脑，可以利用分享电脑网络的办法让BBB也联网。  
Windows7下的配置方法可以参考[http://lanceme.blogspot.com/2013 ... et-sharing-for.html](http://lanceme.blogspot.com/2013/06/windows-7-internet-sharing-for.html)  
  
**电脑系统是Ubuntu的话：**  
BBB上的操作：

1. route add default gw 192.168.7.1
2. vim /etc/resolv.conf

复制代码

在最后添加一行nameserver 8.8.8.8  
  
在Ubuntu上：  
通过ifconfig命令得知eth0是本机连接网络的接口，eth1是usb连接BBB的接口

1. sudo su
2. iptables --table nat --append POSTROUTING --out-interface eth0 -j MASQUERADE
3. iptables --append FORWARD --in-interface eth1 -j ACCEPT
4. echo 1 > /proc/sys/net/ipv4/ip\_forward

复制代码

然后在BBB上ping [www.baidu.com](http://www.baidu.com/)检测一下。  
但重启电脑或BBB就无效了，须重新配置。  
  
**若是mac OS X系统：**插上usb以后，网络连接里会出现一个到BBB的连接，记住这个连接的名字。  
然后打开系统设置里的“共享”，先取消勾选左边的“互联网共享”，然后在右边勾选刚才的连接名字，再重新勾选左边的互联网共享，电脑端就好了。然后保持usb线连接，再插上串口线，用串口连接到BBB（此时ssh是连接不通的），输入 udhcpc -i usb0，就完成配置了。然后就可以拔掉串口线了，如果本次开机过程中断开了usb线（比如同时使用5v电源在供电）也没关系，重新插上还可以恢复网络连接。  
  
  
  
联网以后第一件事就是 opkg update。  
  
**本地编译内核模块**  
因为买来的BBB里没有内核源文件，所以没法编译。现在联网了，可以不必重新交叉编译内核，直接在BBB上下载配置一下内核源文件。

1. opkg update
2. opkg install kernel-headers
3. opkg install kernel-dev
4. cd /usr/src/kernel
5. make scripts

复制代码

就配置完成了。这要耗费约200M空间，可以先输入 df -h 看看eMMC剩余空间够不够。  
测试一下。  
  
源文件mykernelmod.c

1. #include
2. #include
4. static int \_\_init enable\_usermode(void)
5. {
6. printk(KERN\_INFO "Usermode enabled.\n");
7. return 0;
8. }
10. static void \_\_exit disable\_usermode(void)
11. {
12. printk(KERN\_INFO "Usermode disabled.\n");
13. }
15. module\_init(enable\_usermode);
16. module\_exit(disable\_usermode);

复制代码

Makefile（M必须大写，注意make语句前面必须是tab键，直接从网页上复制过去可能会变成空格）

1. obj-m += mykernelmod.o
2. KDIR = /usr/src/kernel
3. PWD := $(shell pwd)
5. all:
6. make -C $(KDIR) M=$(PWD) ARCH=arm modules
7. clean:
8. make -C $(KDIR) M=$(PWD) ARCH=arm clean

复制代码

然后执行make，成功生成mykernelmod.ko  
  
insmod mykernelmod.ko 之后 dmesg | tail 可以看到输出了 “Usermode enabled.” 然后 rmmod mykernelmod，再dmesg | tail，看到输出了"Usermode disabled."  
  
参考资料：[http://elene.dahners.com/blog/20 ... ule-for-beaglebone/](http://elene.dahners.com/blog/2012/10/22/compiling-a-kernel-module-for-beaglebone/)