新购了一套高性能单板计算开发板，大名鼎鼎的Beaglebone Black国内版。鼓捣了两天，记下一些流水账，以供参考。

一、版本和升级

预装在eMMC中的系统版本比较老，是5月的版本， 一个最大的问题是插上Micro-SD 卡后无法正常启动。根据网上找到的一些资料，尝试在Ubuntu虚拟机下格式化SD卡，在SD卡启动分区上编辑修改uEnv.txt文件 ，试图让U-Boot重定位启动文件。但不幸的是，多次尝试均告失败。

随后浏览相关网站，发现新版的系统中有提到，支持系统启动后Micro-SD卡的挂载，遂感觉还是得选择正确的路线，就是要先升级到尽量新的系统版本。

系统版本的说明可以在下面找到

 http://elinux.org/Beagleboard:Updating\_The\_Software

镜像文件下载后，解压出来一个.img文件，用推荐的烧写工具[DiskImager](https://wiki.ubuntu.com/Win32DiskImager)，烧写到Micro-SD卡中即可使用。

其中的 [Production](https://s3.amazonaws.com/angstrom/demo/beaglebone/BBB-eMMC-flasher-2013.09.04.img.xz" \t "_blank)

[Image 2013.09.04](https://s3.amazonaws.com/angstrom/demo/beaglebone/BBB-eMMC-flasher-2013.09.04.img.xz" \t "_blank)  和[Updated Image 2013.06.20](https://s3.amazonaws.com/angstrom/demo/beaglebone/BBB-eMMC-flasher-2013.06.20.img.xz) 这类的映像是eMMC-Flasher, 是烧写到Micro-SD卡 用来升级板载eMMC中的系统的。 升级过程约45分钟，4个user-LED全部亮起来就算升级结束了。 如果中途断电了，那么eMMC不能正常启动了，但eMMC-FlasherMicro-SD卡是可以启动运行的，重新升级就可以，不用太顾虑。

其中[SD Boot 2013.08.21](https://s3.amazonaws.com/angstrom/demo/beaglebone/Angstrom-Cloud9-IDE-GNOME-eglibc-ipk-v2012.12-beaglebone-2013.08.21.img.xz) 这类的映像文件是 CLoud9 GNOME Image,是直接用来烧写到4G Micro-SD 卡中启动系统的。这个不会修改板载eMMC的内容。

二、启动与终端

    购置的板子，附送的USB转串口线，是非常有用的。连接usb串口的GND、RXD、TXD，对应到 BBB板上J1的1脚、5脚、4脚。打开串口终端，设置速率115200。 然后连接上BBB板上的供电兼多接口线缆--就是那条USB-otg线。很有成就感吧，串口终端就会出来引导的过程了。 启动过程中，BBB会启动若干的设备连接到电脑上，按要求安装驱动程序即可。 启动完成后也会出现一个移动盘，其中有需要安装的驱动程序，非常方便。

 系统启动完成后，可以用root登陆，默认密码为空。 这样就可以进入字符界面操作了。 比如看看网络接口，

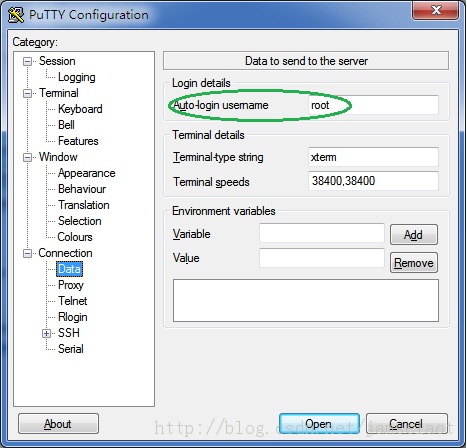
# ifconfig  
eth0      Link encap:Ethernet  HWaddr 90:59:AF:65:F4:76  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
          Interrupt:56  
  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:354 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:354 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:21912 (21.3 KiB)  TX bytes:21912 (21.3 KiB)  
  
  
usb0      Link encap:Ethernet  HWaddr AA:E2:B4:0C:28:FD  
          inet addr:192.168.7.2  Bcast:192.168.7.3  Mask:255.255.255.252  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:42402 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:35212 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:3008175 (2.8 MiB)  TX bytes:18764279 (17.8 MiB)

三、网络与ssh终端

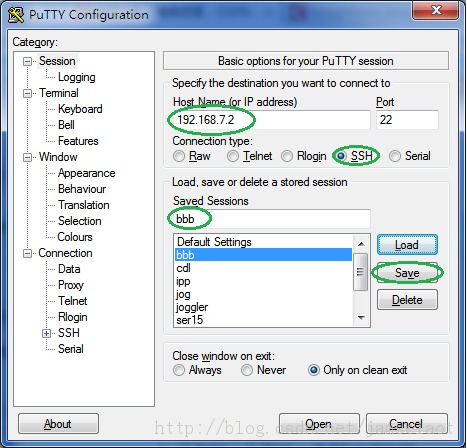
通过串口终端我们可以看到启动的过程，也能做一些交互操作，比如我们看到了网络接口的状况。其中的usb0已经在随板的系统中配置好了，有一个固定的IP地址 192.168.7.2 。

    在主机这边，应该也安装好了跟BB-Black相关的驱动程序，其中就有一个网络设备 Linux USB Ethernet/RNDIS Gadget。为了和BB-Black连接方便，可以把主机的这个网卡也配置一个固定的IP地址 192.168.7.1

    这里使用一个非常小巧的终端软件 PuTTY 作为BB-Black的ssh 终端，为了登陆方便，可以做一些简单配置比如



配置地址，名称等，要记得保存。

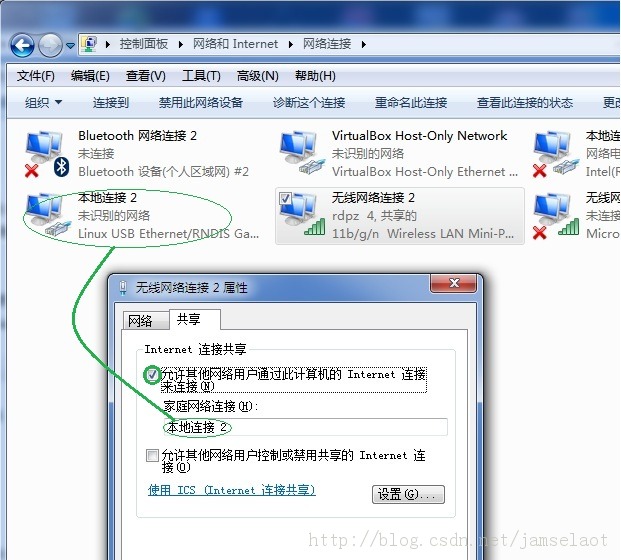


    启动PuTTY后，就可以登入 BB-Black的ssh终端界面。

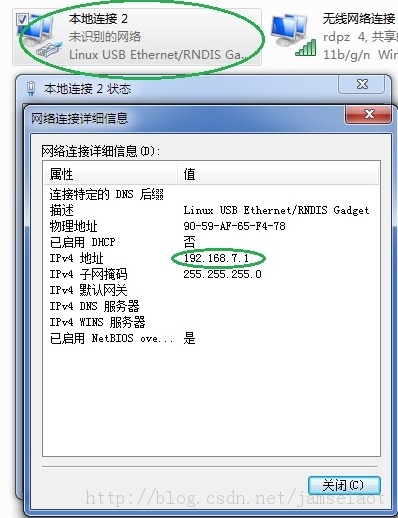
四、让BB-Black通过usb0上网

既然我们已经用usb0作为主机和BB-Black之间的网络通道了，再进一步，就可以让BB-Black通过usb0共享主机的互联网连接，使BB-Black有外网的连接。

    先配置主机连接外网的网卡，此处是 无线网络连接2，让它可以用来被其他网卡共享上网，这里需要共享网络连接的是本地连接2 就是BB-Block的usb网络接口。



注意配置 本地连接2 的网络地址：



再配置BB-Black的路由和DNS等，可以在PuTTY中，登陆ssh终端，输入指令：

**[plain]** [view plaincopy](http://blog.csdn.net/jamselaot/article/details/17080011)

1. root@beaglebone:~# route add default gw 192.168.7.1

然后查看路由设置

**[plain]** [view plaincopy](http://blog.csdn.net/jamselaot/article/details/17080011)

1. root@beaglebone:~# route
2. Kernel IP routing table
3. Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
4. default         192.168.7.1     0.0.0.0         UG    0      0        0 usb0
5. 192.168.7.0     \*               255.255.255.252 U     0      0        0 usb0

还需要配置域名解析，编辑文件  /etc/resolv.conf

**[plain]** [view plaincopy](http://blog.csdn.net/jamselaot/article/details/17080011)

1. root@beaglebone:~# nano /etc/resolv.conf

增加几条nameserver的记录，并保存修改。比如 

**[plain]** [view plaincopy](http://blog.csdn.net/jamselaot/article/details/17080011)

1. nameserver 127.0.0.1
2. nameserver 8.8.8.8
3. nameserver 192.168.1.1

测试一下，结果如下：

**[plain]** [view plaincopy](http://blog.csdn.net/jamselaot/article/details/17080011)

1. >root@beaglebone:~# ping bbs.csdn.net
2. PING bbs.csdn.net (117.79.157.198) 56(84) bytes of data.
3. 64 bytes from 117.79.157.198: icmp\_req=1 ttl=48 time=16.9 ms
4. 64 bytes from 117.79.157.198: icmp\_req=2 ttl=48 time=9.00 ms
5. ^C64 bytes from 117.79.157.198: icmp\_req=3 ttl=48 time=8.51 ms

这样，BB-Black就可以通过usb0这个网卡，共享主机的外网连接上网了。 可以执行升级等操作了。

五、vnc图形终端

虽然 BB-Black带有一个Micro-HDMI接口，不过那么名片不到的一个小板，连接到一个20来寸的显示器上，还是有些不协调。索性我们使用图形界面远程登陆，实现BB-Black的图形显示。这样的话，仍然是使用那条连接到USB-OTG上面的多接口USB线缆，就可以实现一线通的目的。功能完备又十分经济，和BeagleBone 开发板的设计套路也是非常吻合的。

在BBB上安装VNC server 很简单

**[plain]** [view plaincopy](http://blog.csdn.net/jamselaot/article/details/17081421)

1. opkg update
2. opkg install x11vnc

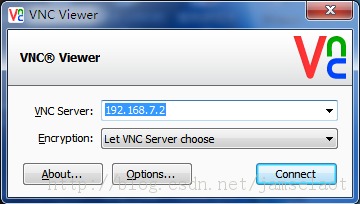
（在烧写了最新版的eMMC映像后，发现已经用不着自己在BB-Black上面升级添加 vnc-server了。---我是不是记错了，有可能啊）

在PuTTY的ssh终端中，启动X11vnc-server,格式如下：

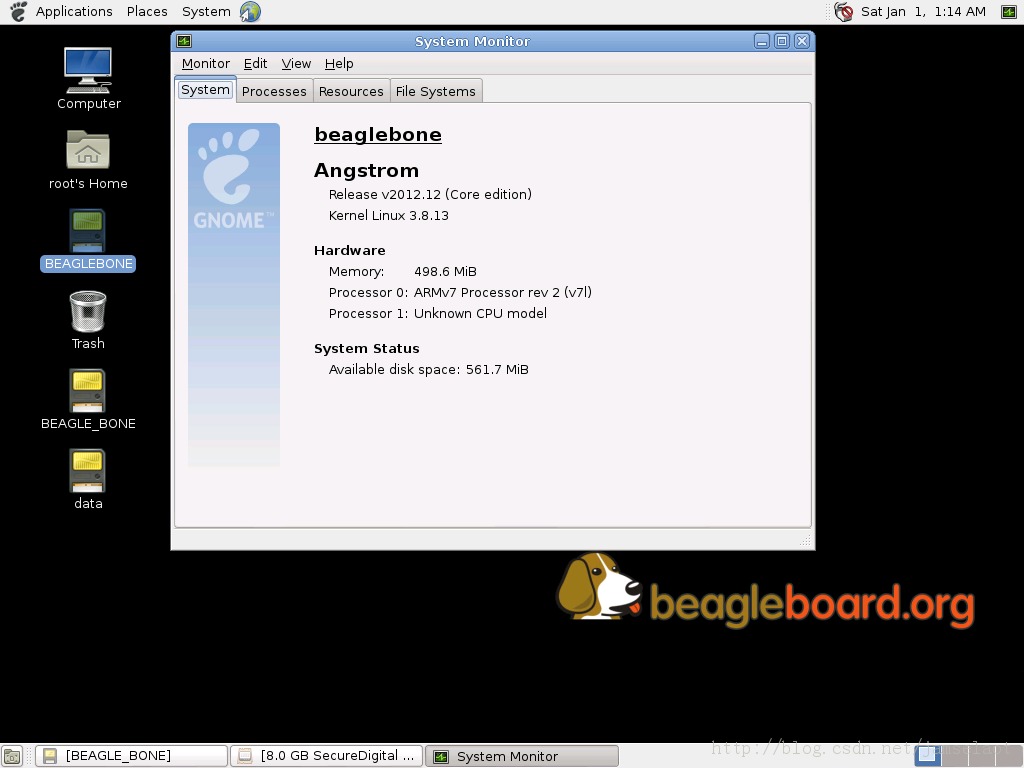
**[plain]** [view plaincopy](http://blog.csdn.net/jamselaot/article/details/17081421)

1. x11vnc -bg -o %HOME/.x11vnc.log.%VNCDISPLAY -auth /var/run/gdm/auth-for-gdm\*/database -display :0  -forever

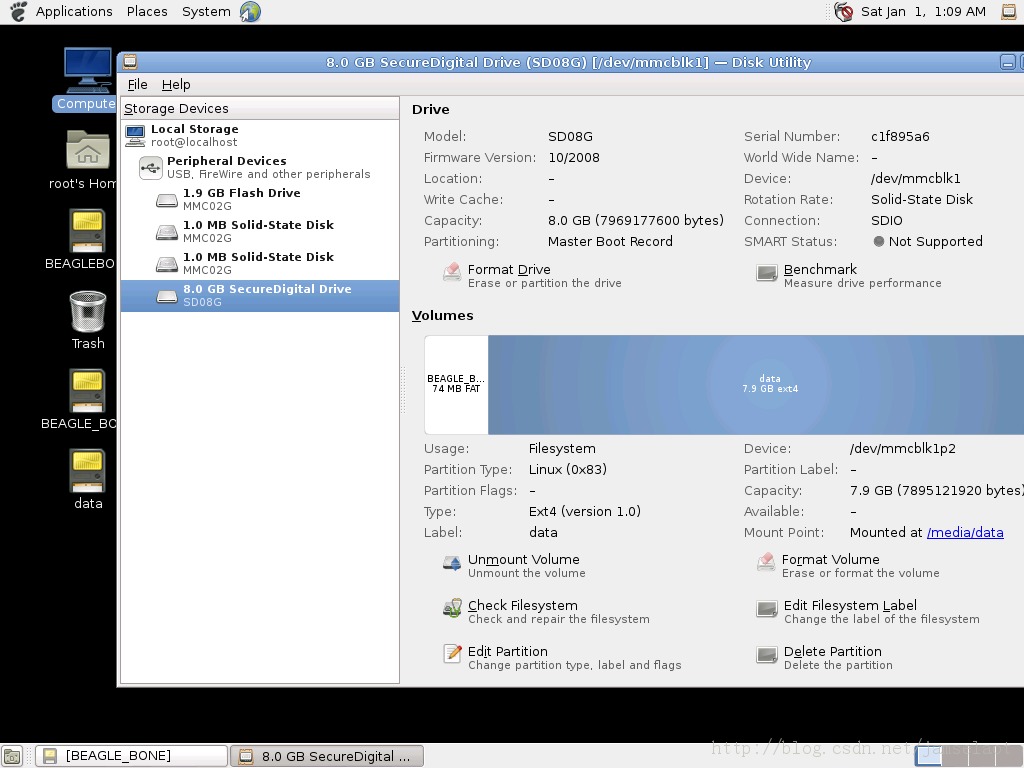
window主机可以使用免费的  vnc-viewer。非常简单。



 连接后就可以接入BB-Black的图形界面了，请看



再看



到此为止，我们就可以利用一条 USB连接线，通过BB-Black的USB-OTG接口，使用这个优秀的开发板提供的多个逻辑接口，在主机和开发板之间建立了丰富的联系。为BB-Black开发板配齐了网络连接、远程图形终端、虚拟键盘鼠标等设备。