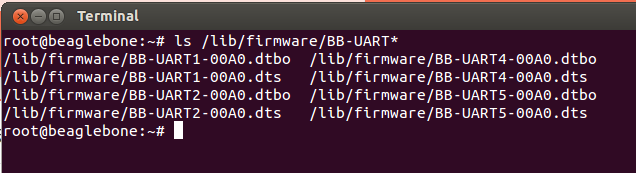
今天成功地进行了对串口的操作和配置（其实比较简单）， 想记录一下。

首先，由于Beaglebone Black采用了3.8的Linux内核，引进了设备树的概念。所以要想操作特定的外设，首先要先和设备树打交道。

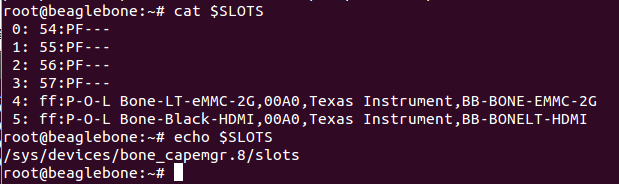
好在今天要操作的是串口，串口的设备树已经被包含在文件系统中了， 可以查看一下：



其中有UART1-UART5共5个uart串口，UART0供调试使用。

该目录下的\*dtbo是由\*dts编译之后生成的文件，所以，如果你想写自己的设备树文件，可以参考这里的\*,dts的写法。

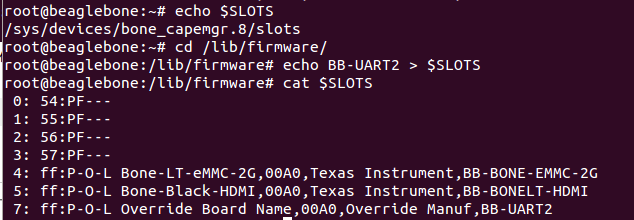
在操作UART（这里我们先尝试使用UART2）之前，我们先看看系统默认已经加载的一些设备：



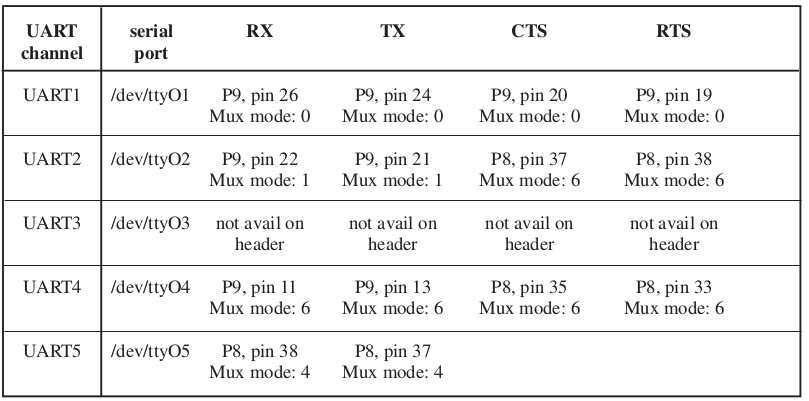
可以看到eMMC和HDMI是默认加载好的。注意我的$SLOTS的实际值，如果你想跟着我来一起操作，关注一下倒数第二行。

为了能够加载UART2，我们只需要在终端输入：

cd /lib/firmwareecho BB-UART2 > $SLOTS这时，我们看看是否加载成功了：



最后一行果然显示已经成功加载了。  
下面就可以操作一下串口了，不过首先要知道UART2在哪儿，可以参考一下下面这个表格：



好的，直接进入/dev/ttyO2。  
可是，应该如何来显示呢？  
我是通过BBB自带的cable来远程登录，用另一个USB转串口线来显示UART2接受到的信息的。  
连接时，只需要USB串口上的三根线：GND， RXD， TXD。GND接BBB的P9\_01， RXD接P9\_21， TXD接P9\_22。  
将另一端直接接在计算机的USB口上。  
有很多串口工具，我使用的是minicom。具体的设置可以参考[这篇文章](http://hi.baidu.com/zengzhaonong/item/a7452f3d8d50ecbf124b1449)， 不过要注意，要选择/dev/ttyUSB0， 波特率选择9600。  
下面，就可以测试了：

