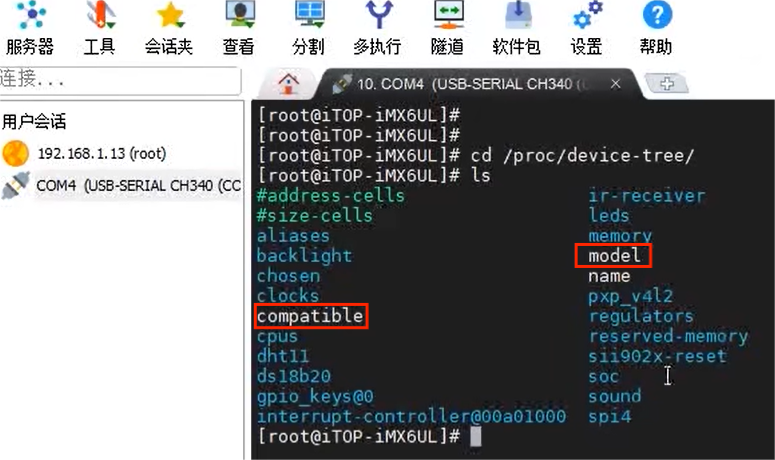
我们在添加完后怎么验证是否添加成功？

我们的设备树也可以在linux中查看：我们的子节点和节点的属性都可以在linux系统里面以文件或者目录的形式体现出来。

我们查看的方式有两种：

1. 在 /proc/device-tree/ 目录下面的都是一些节点：

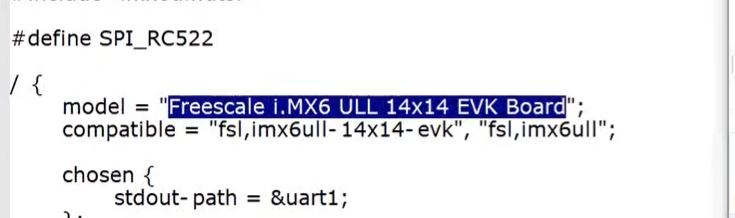


(比如这里有model和compatible这两个节点，那么我们想要在linux中查看这两个属性的话，如何操作？)

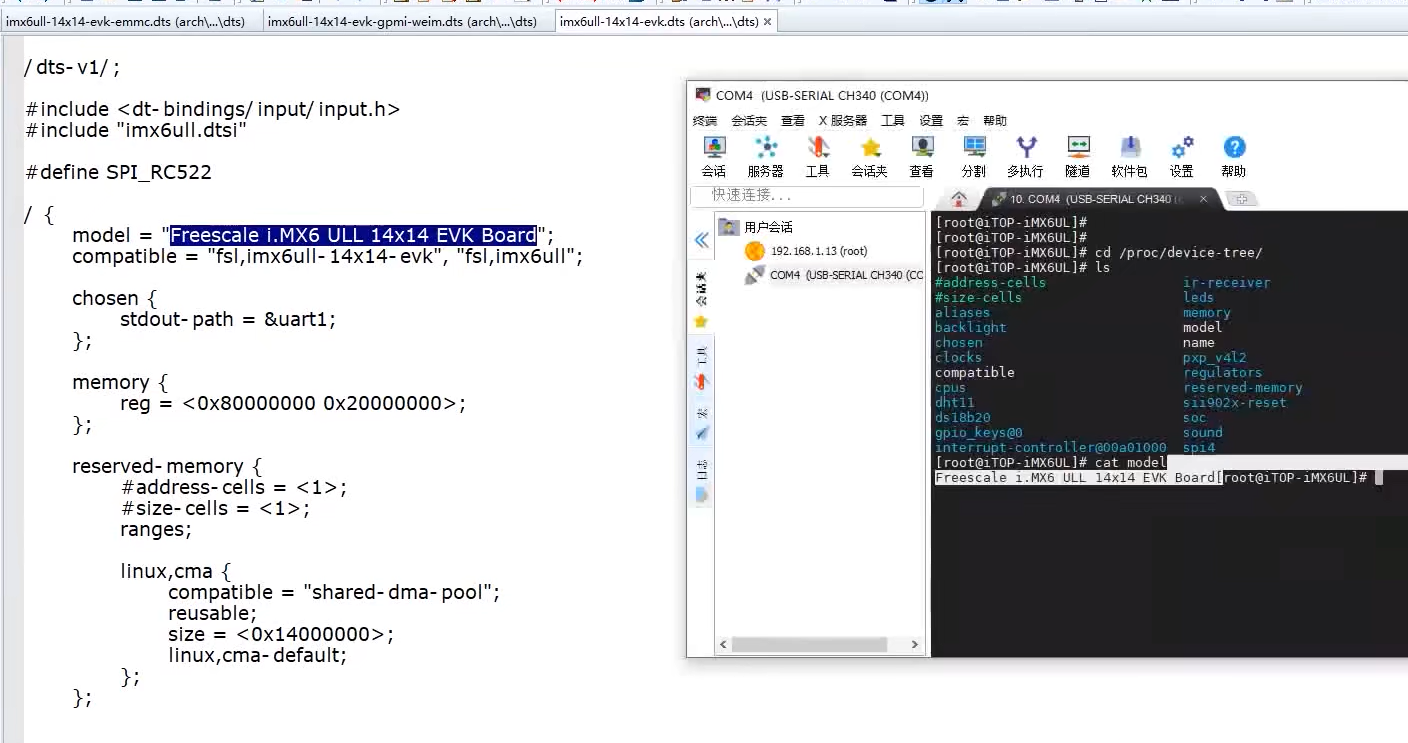
直接执行 cat model:

IMG_256

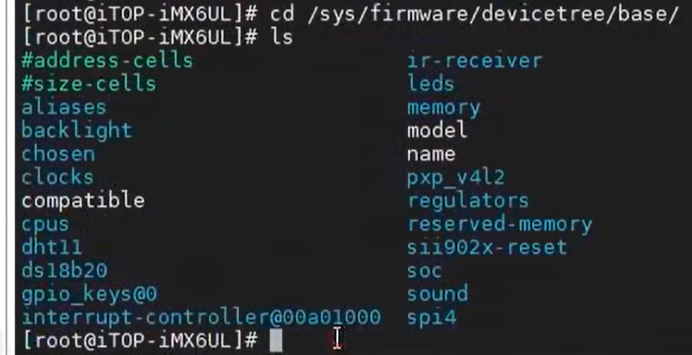
其中这个值就是我们设备树文件里面：



这两个值是一模一样的：



1. 进入 /sys/firmware/devicetree/base/ 目录：该目录和上面的/proc/device-tree/ 目录结构是一样的



————————————————————————————————

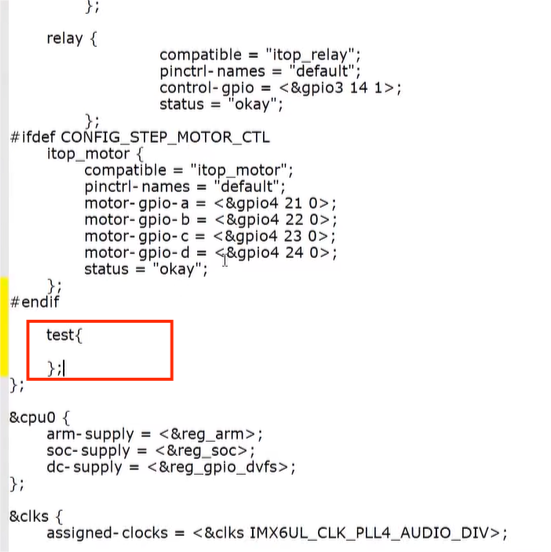
现在我们正式创建自己的设备树节点：

/\*

我们在kernel/arch/arm64/boot/dts/rockchip/rk3568-atk-atompi-ca1.dts里面编写

\*/

在根目录下面写如下test节点：



然后需要什么属性，就添加什么属性：

这里定义reg地址的数量为1

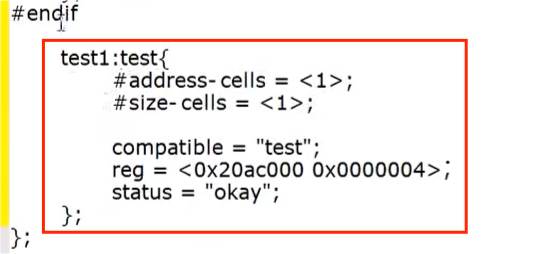
定义reg地址长度 为1

Compatible为“test”

reg寄存器的值为 DR寄存器的地址：0x20ac000 长度为0x00000004

Status为okay

并且，我们将test节点取别名为test1:



那么我们如何编译我们的设备树文件呢？

首先我们需要dtc编译工具。

如果环境没有dtc工具，则需要安装dtc工具。

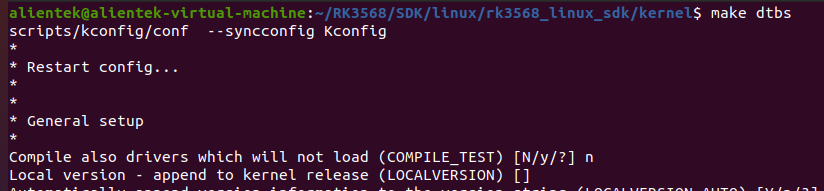
安装命令： apt-get install decive-tree-compiler

/\*

我们在kernel/arch/arm64/boot/dts/rockchip/rk3568-atk-atompi-ca1.dts里面添加test节点。

\*/

当我们执行make dtbs 命令显示：一般是环境没有配置好。

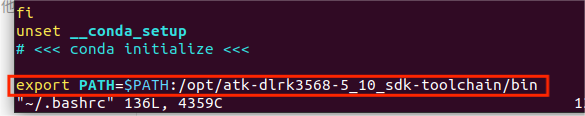


就需要配置编译环境：

/\*

首先需要注意：

在/.bashrc 文件末尾有无交叉编译工具链的路径：

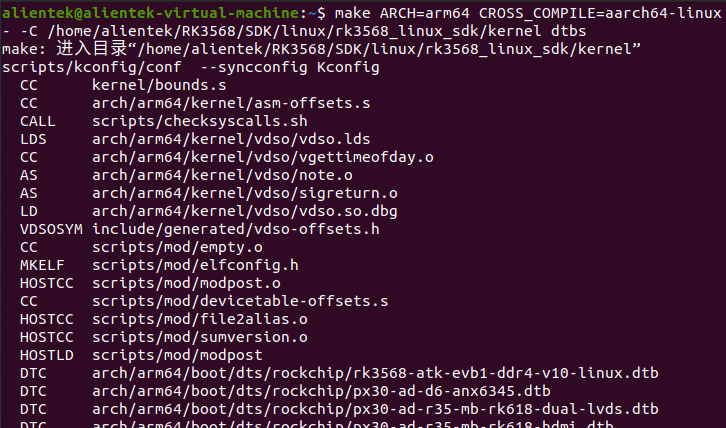


\*/

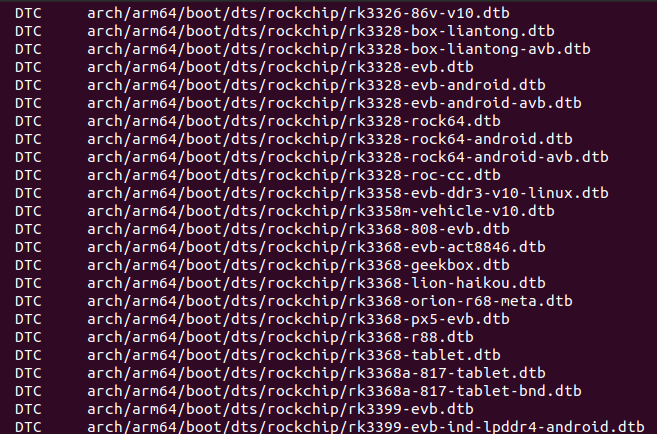
然后就可以在内核直接执行命令编译设备树：

make ARCH=arm64 CROSS\_COMPILE=aarch64-linux- -C /home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/kernel dtbs

我们的编译结果：



在编译过程中：



/\*

DTC arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10-android9.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10-linux.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10-linux-spi-nor.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb2-lp4x-v10.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb2-lp4x-v10-bt1120-to-hdmi.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb4-lp3-v10.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb5-ddr4-v10.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb6-ddr3-v10.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb6-ddr3-v10-linux.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb6-ddr3-v10-rk628-bt1120-to-hdmi.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb6-ddr3-v10-rk628-rgb2hdmi.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb6-ddr3-v10-rk630-bt656-to-cvbs.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-evb7-ddr4-v10.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-iotest-ddr3-v10.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-iotest-ddr3-v10-linux.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-nvr-demo-v10.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-nvr-demo-v10-linux.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-nvr-demo-v10-linux-spi-nand.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-nvr-demo-v12-linux.dtb

DTC arch/arm64/boot/dts/rockchip/rk3568-nvr-demo-v12-linux-spi-nand.dtb

DTC arch/arm64/boot/dts/rockchip/rk630-rk3568-ddr3-v10.dtb

\*/

前面的DTC表示我们在使用DTC工具进行编译设备树文件。

/\*

上面的命令编译的是整个设备树，

如果我们只需要编译设备树其中的某一个文件又怎么操作？

我们只需要将命令后面的dtbs 改成要编译的设备树文件的 文件名，即可。

例：

make ARCH=arm64 CROSS\_COMPILE=aarch64-linux- -C /home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/kernel rk3568-evb1-ddr4-v10.dtb

dtbs s就表示很多dtb嘛，所以单个文件就是dtb。

\*/

我们编译好的dtb文件就会在 arch/arm64/boot/dts/路径下：

在这个路径下面就可以找到我们编译成功的dtb文件。然后我们的开发板后续烧写即可。

但注意，以他的开发板为例，开发板烧写的并不是imx6ull-14x14-evk.dts文件，因为该文件被包含于topeet\_emmc\_4\_3.dts文件中。（他们是包含关系，当我们编译topeet\_emmc\_4\_3.dts文件时，就会去编译imx6ull-14x14-evk.dts文件）