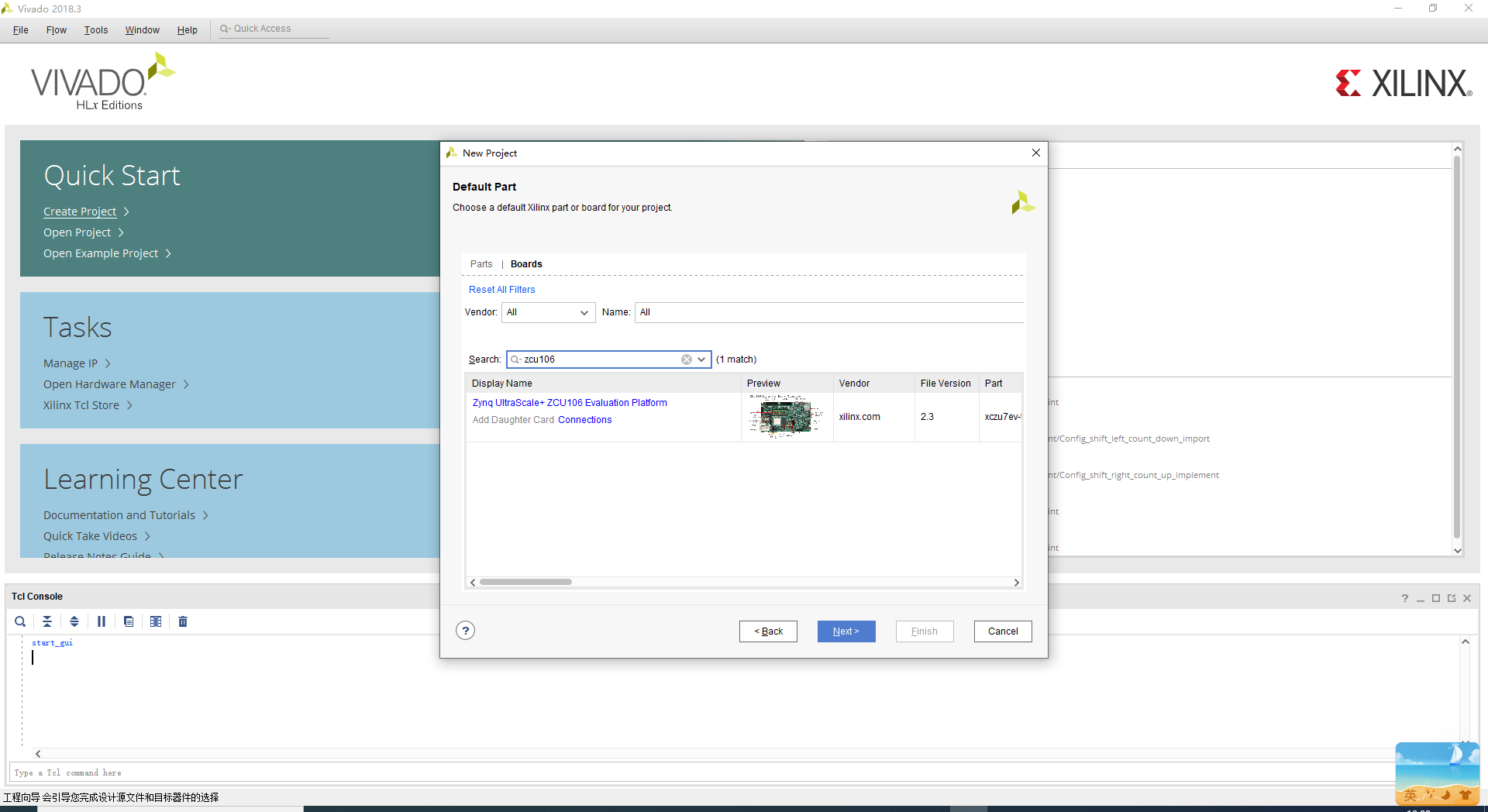
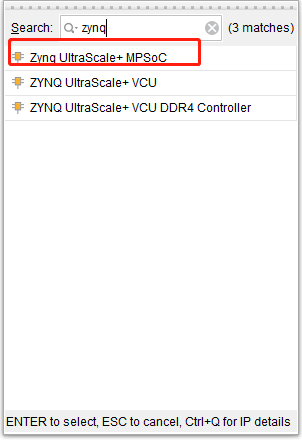
1. 新建bd

选一个幸运路径新建一个工程，没啥要注意的，选下图所示套板即可

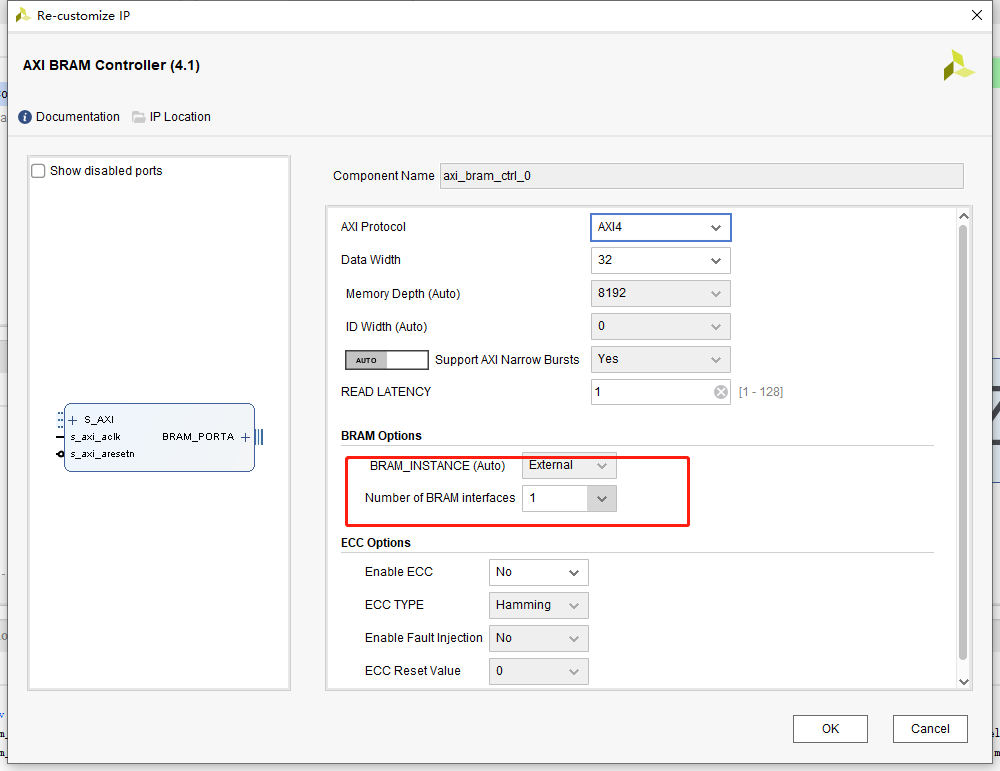


创建一个bd，加入下图所示大宝贝

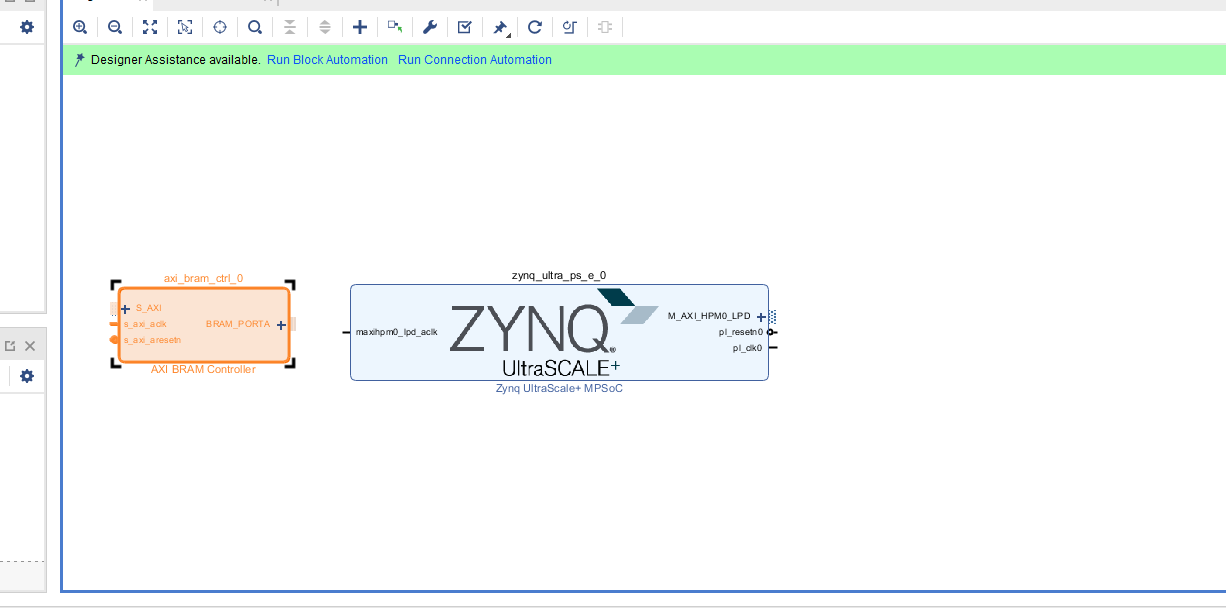


以及一个平平无奇的axi bram controller

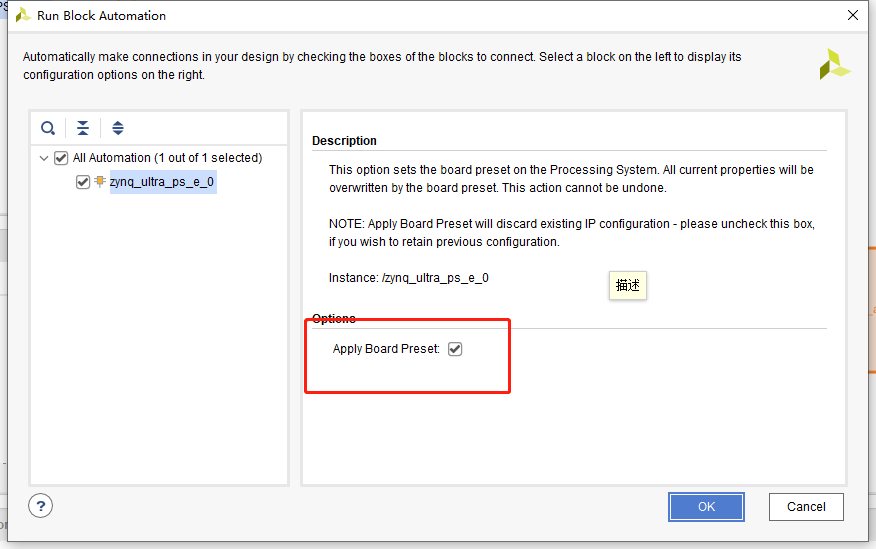
先设置一下bramctrl里面，改为单通道，如下所示



之后退出来能看到bd正上方有两个东西在诱惑你点击它

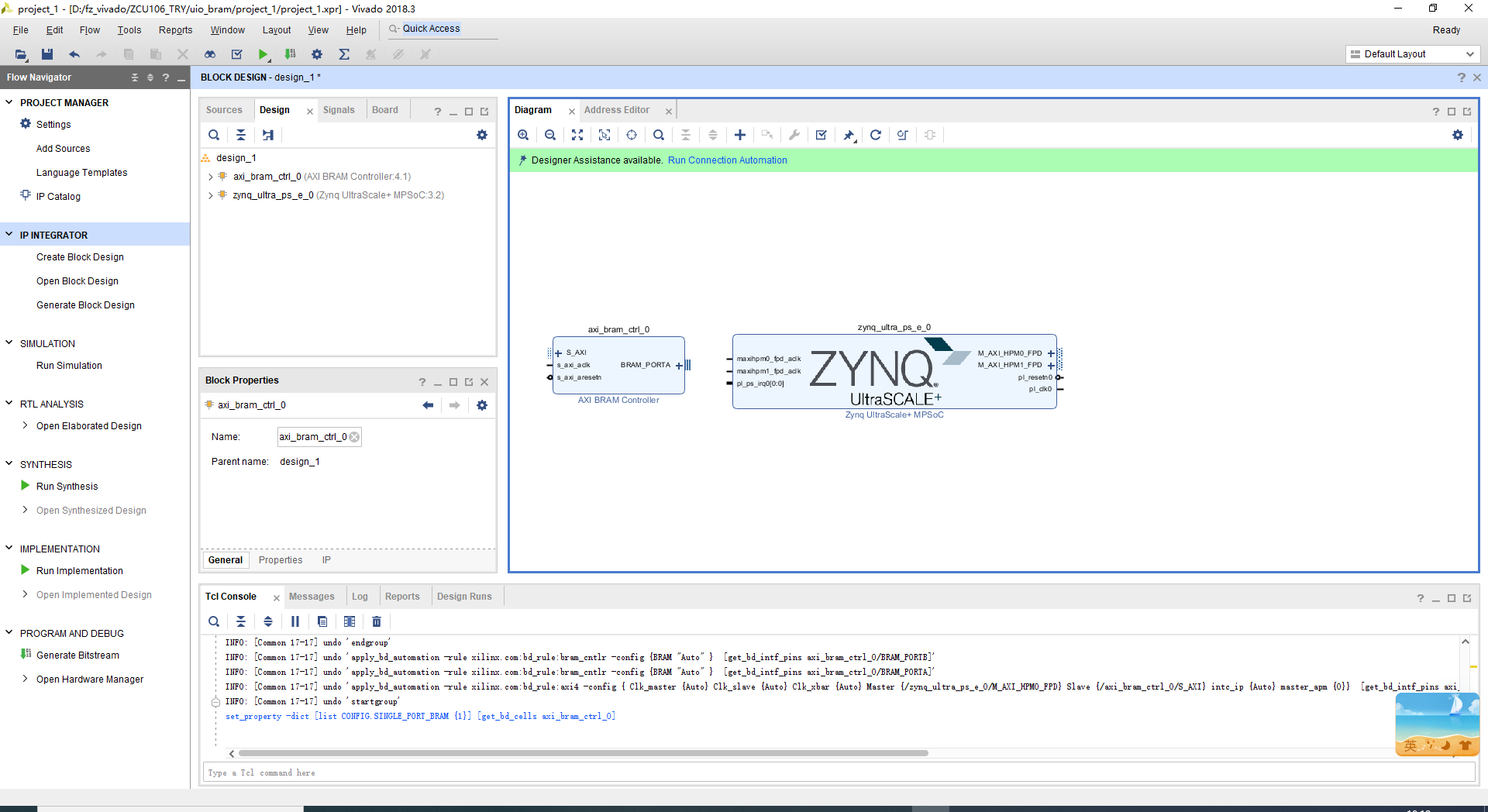


先点左边的，这个是为板子提供预设的，可以理解为帮咱们设置好了一些琐碎的参数并打开一些常用的管脚

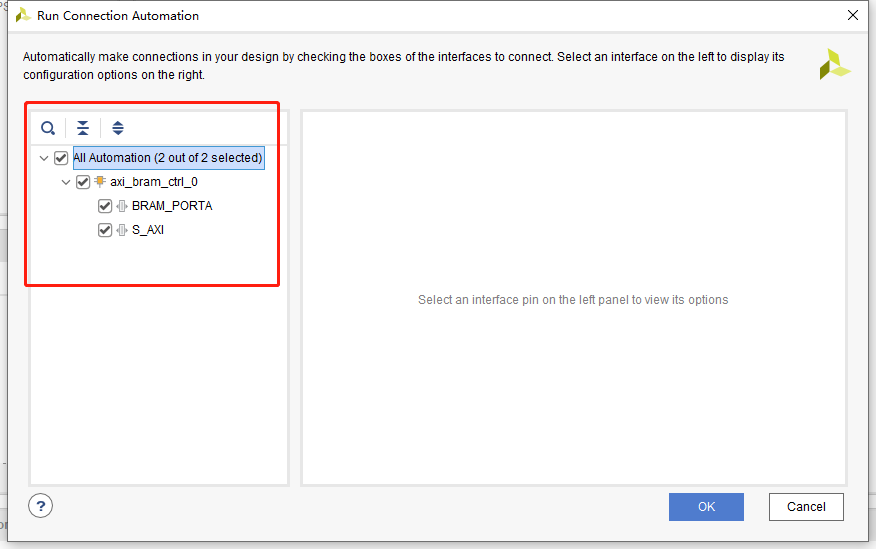


确保这个钩是点着的就行

之后zynq板有了些许变化，如下所示

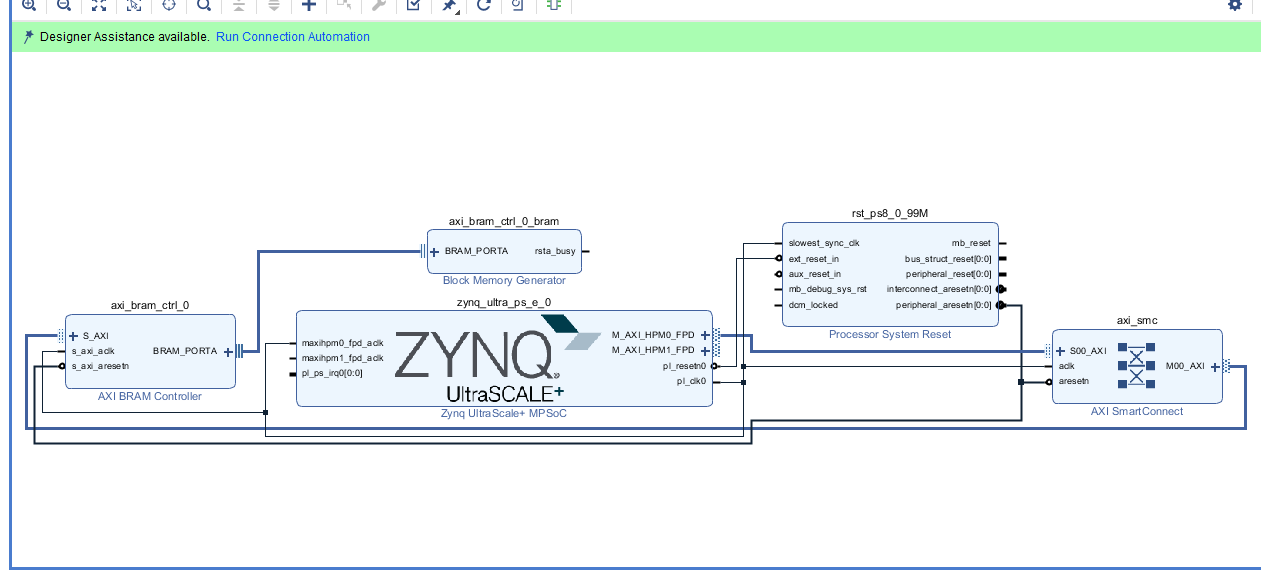


此时再点击剩下的那个，这个会揣摩我们的意图并且按照他觉得我们想做的给我们连线。其实平时是不太建议用这个的，可能会有不同的想法，但是本次实验就一个bram，他没有错误空间，用就行了

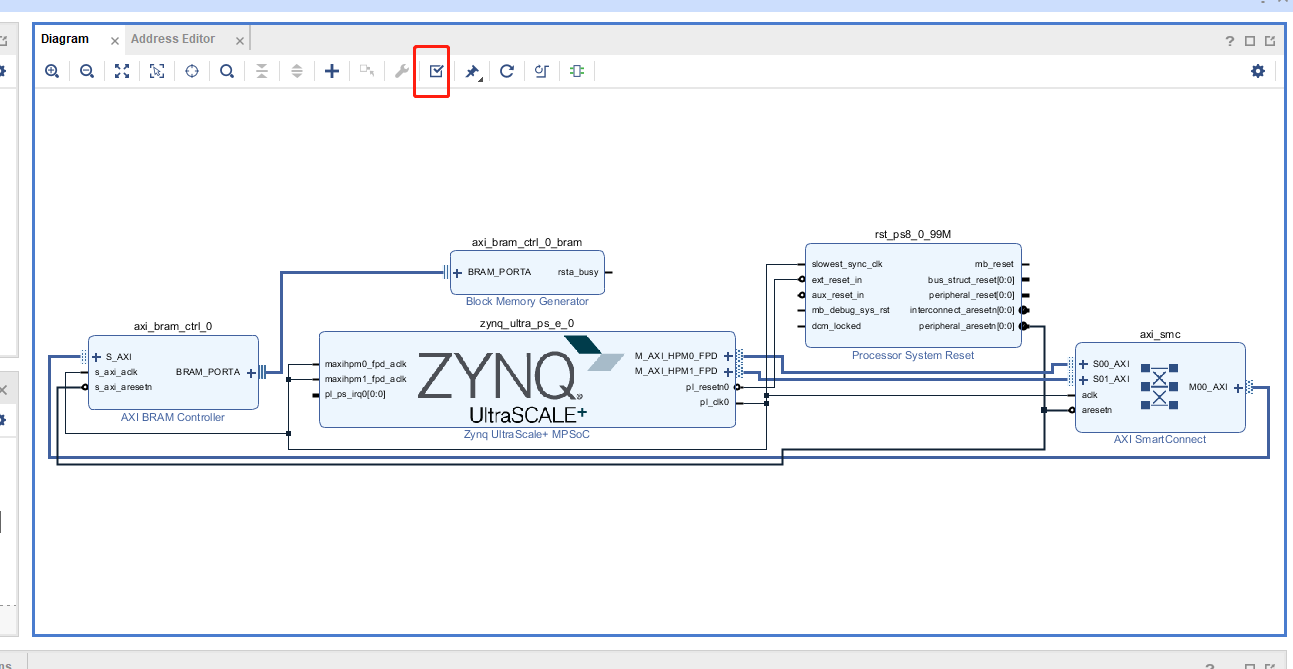


记得把钩都选上

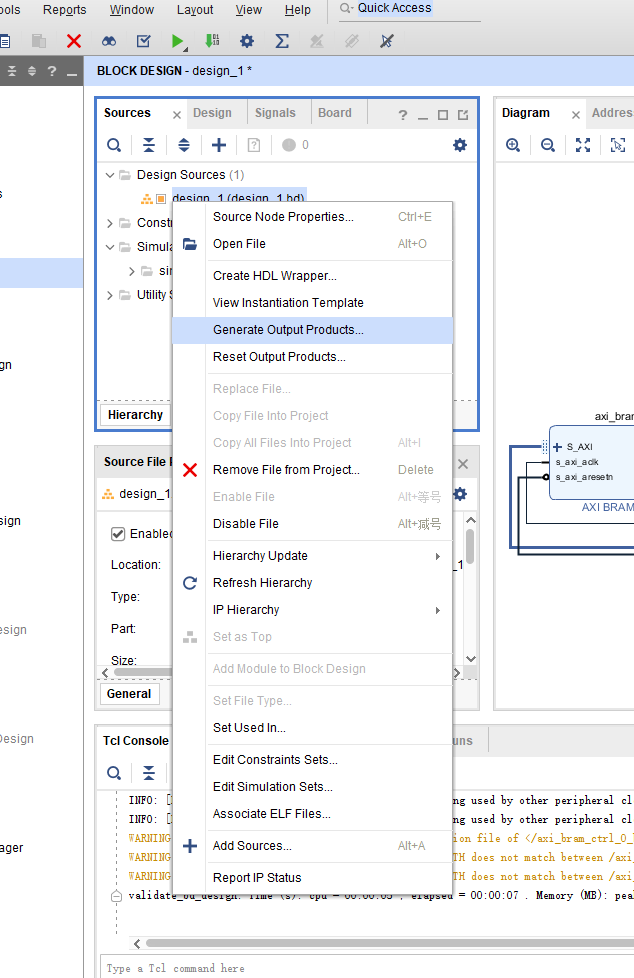
这时可能还会出现一个新的



没事，再点一下就行

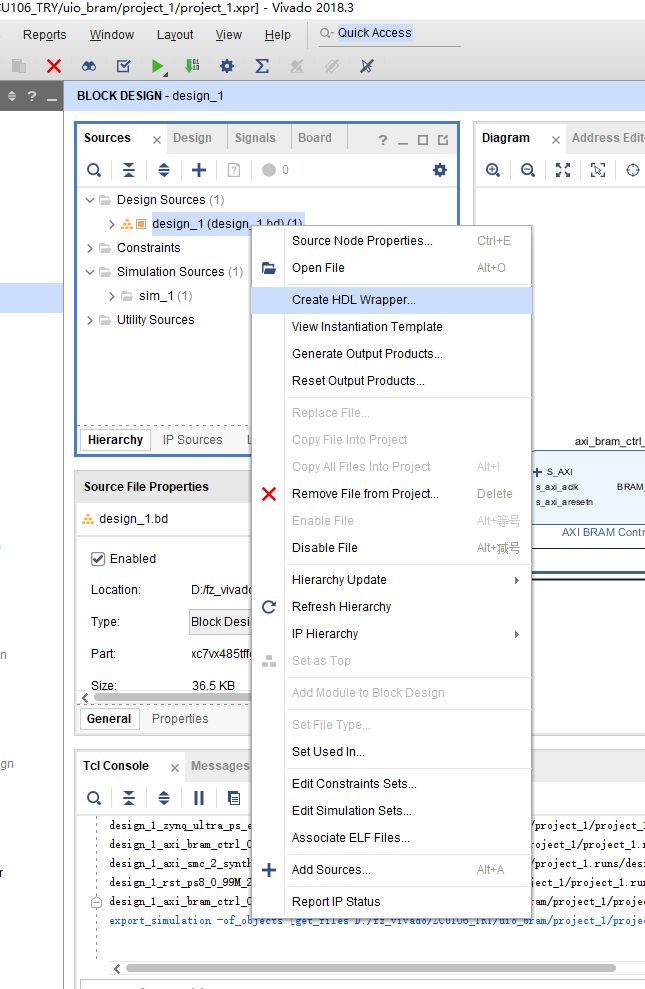


点击验证bd是否正确（当然我们是自动连的，一般是正确的，不然vivado可以倒闭了）

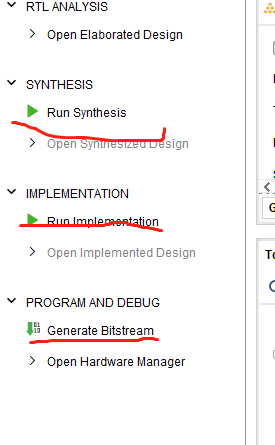


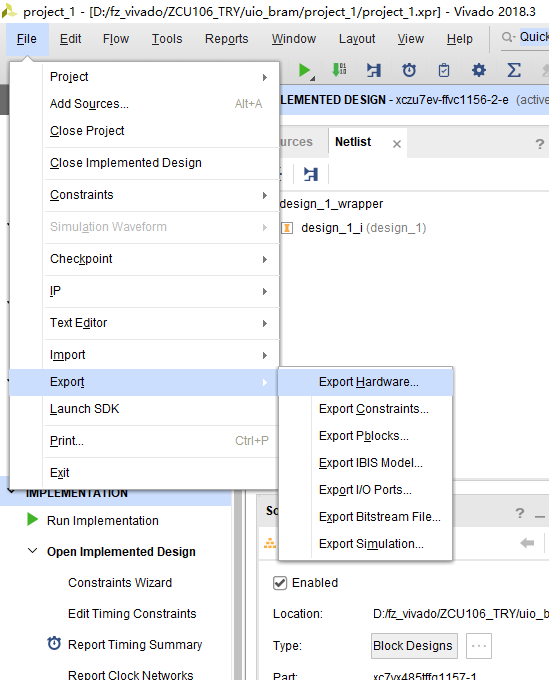
选到source界面右键你的bd，选择这个generate output products

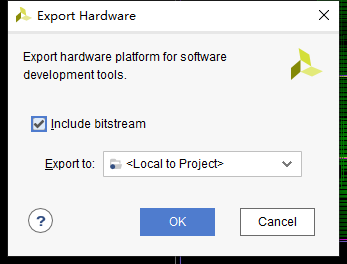
再选如图所示



记得等右上角转完

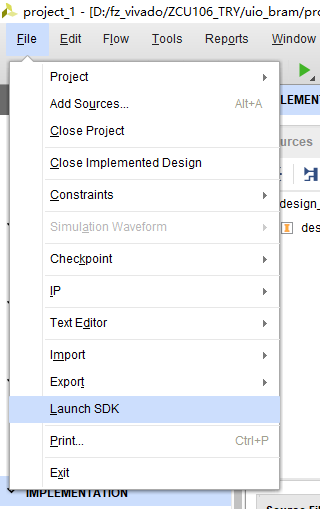
依次点击这三兄弟（也要等右上角转完）





一定包括bit流嗷

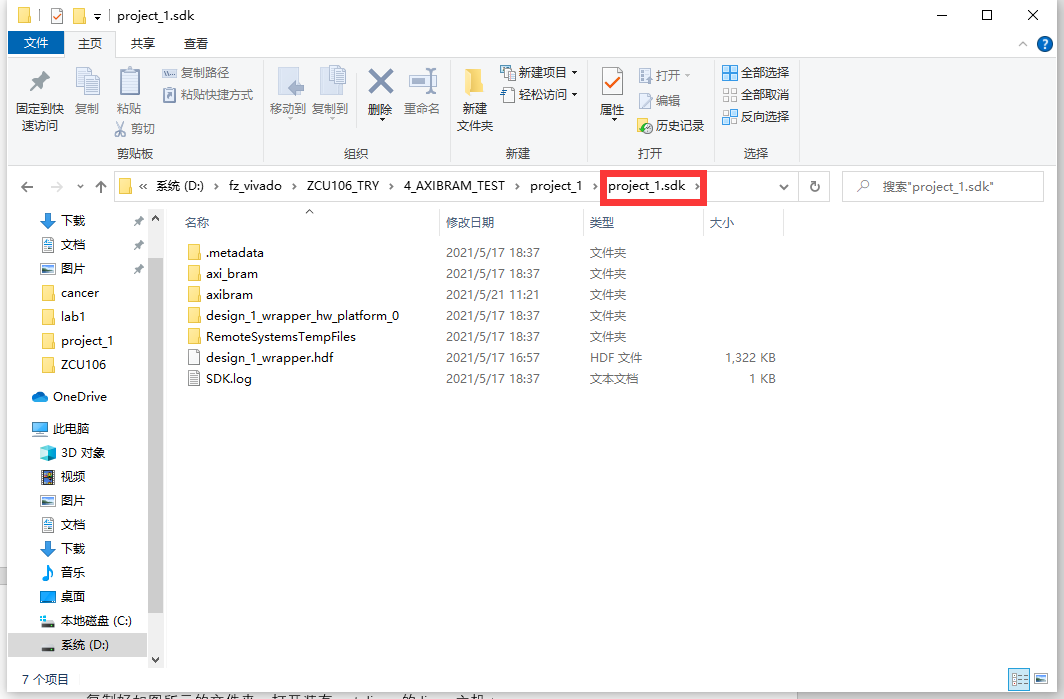
最后



找到工程路径，把下面这个文件夹复制出来，后面的活要在linux主机上做了

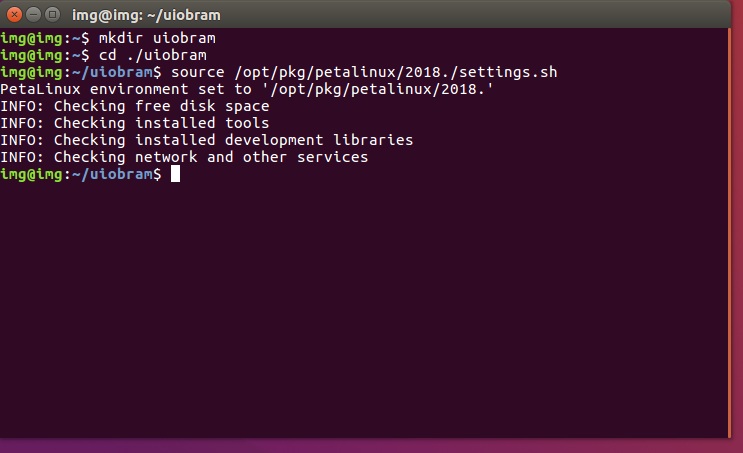
1. Petalinux

复制好如图所示的文件夹，打开装有petalinux的linux主机



（其实需要的是里面的output hardware那步中输出的hdf文件）

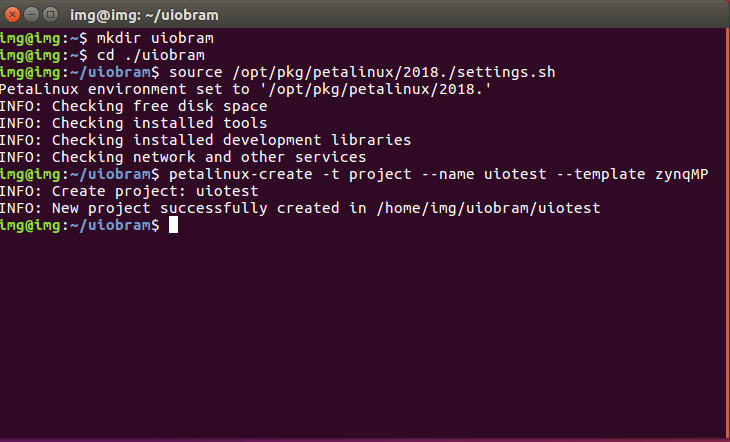
从桌面打开终端，mkdir一个你喜欢的路径名，我是uiobram，并且找到对应的petalinux脚本路径，启动petalinux，我的是：source /opt/pkg/petalinux/2018./settings.sh(其实好像是2018.3版本的petalinux)

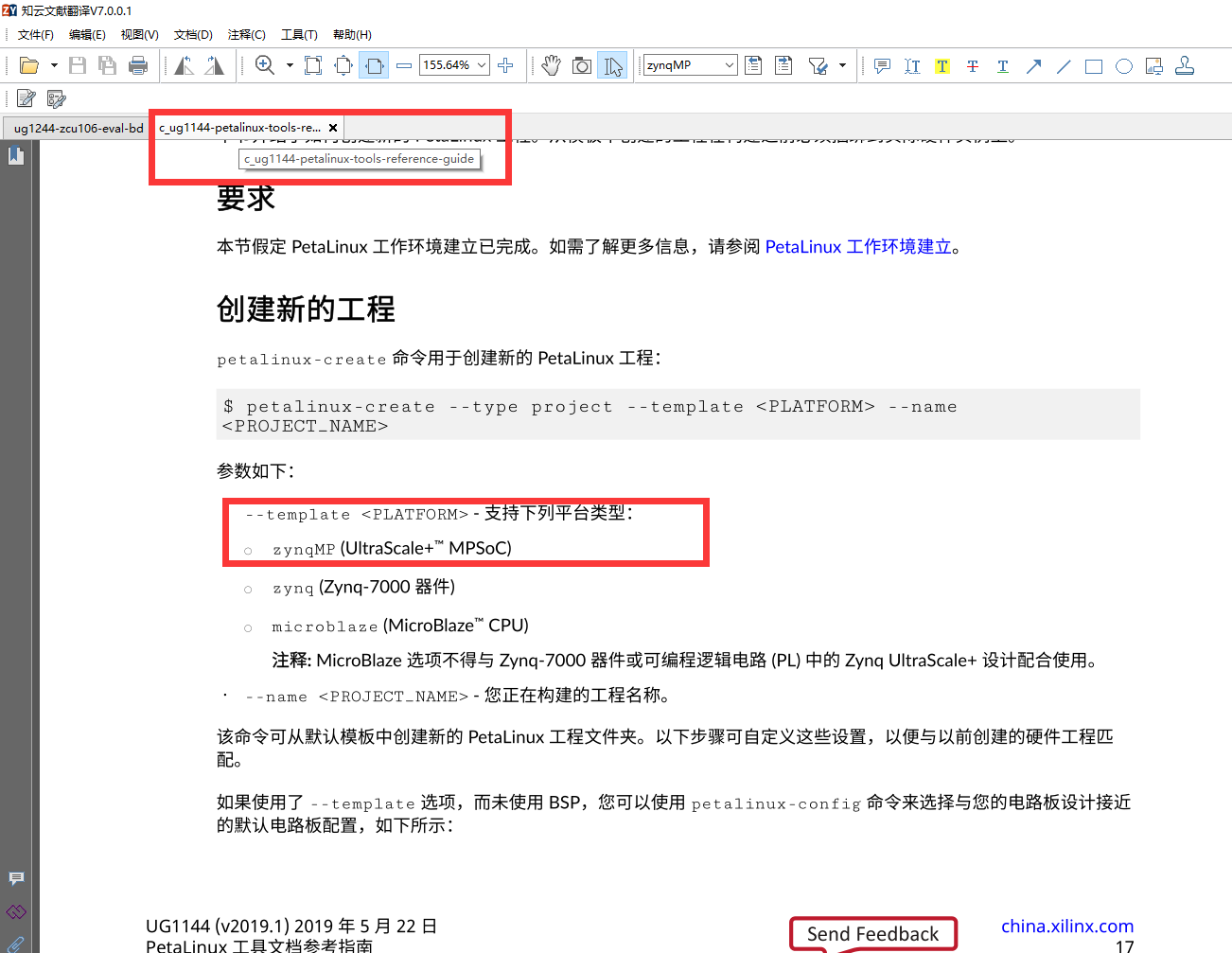


在这个路径下建立一个你喜欢的工程，命令是：

petalinux-create -t project --name uiotest --template zynqMP

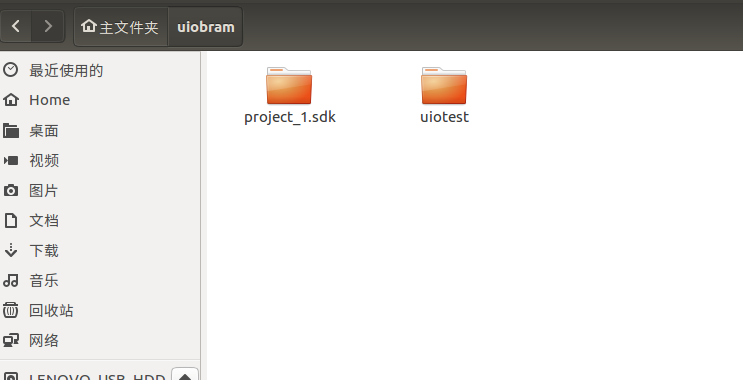
(其中的uiotest那个位置是工程名字可以更改，-t project 表示建立的是工程，template zynqMP是因为我们选用的平台类型（就在vivado中搭建bd时选择的）是UltraScale+ MPSoC那个东西，要对应，完成后看到提示





上图是zcu106使用petalinux官方手册中关于这部分的写法

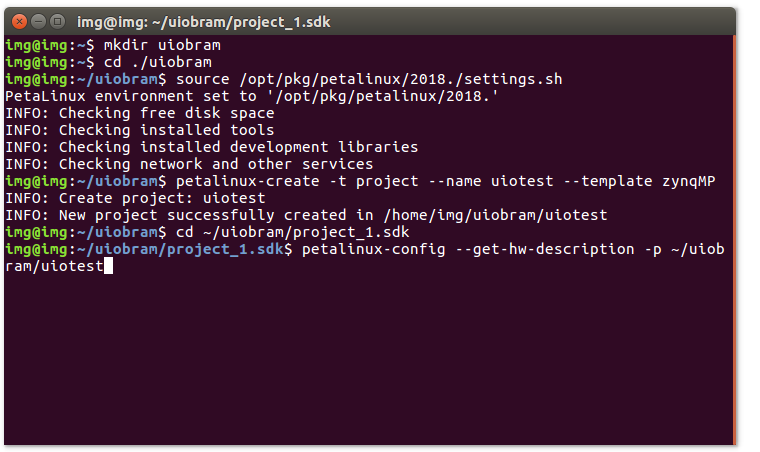
将之前拿出来的sdk文件夹（不改vivado工程名字的话就是project\_1.sdk），复制到和工程名字文件夹并列的路径下（如图所示）



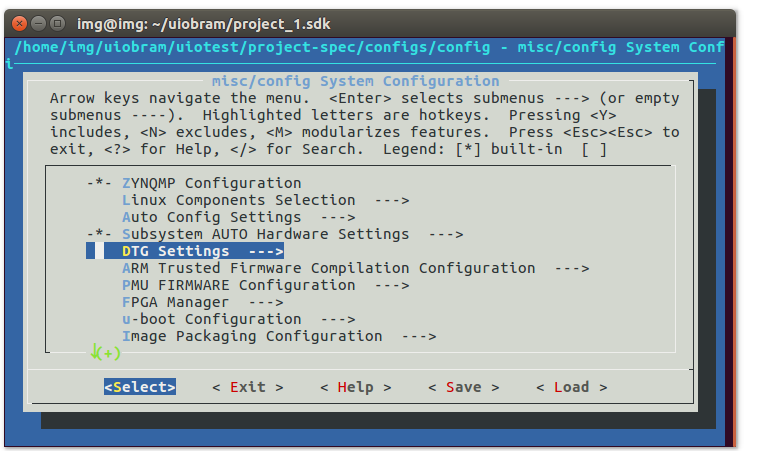
cd进入project\_1.sdk文件夹

输入：petalinux-config --get-hw-description -p ~/uiobram/uiotest

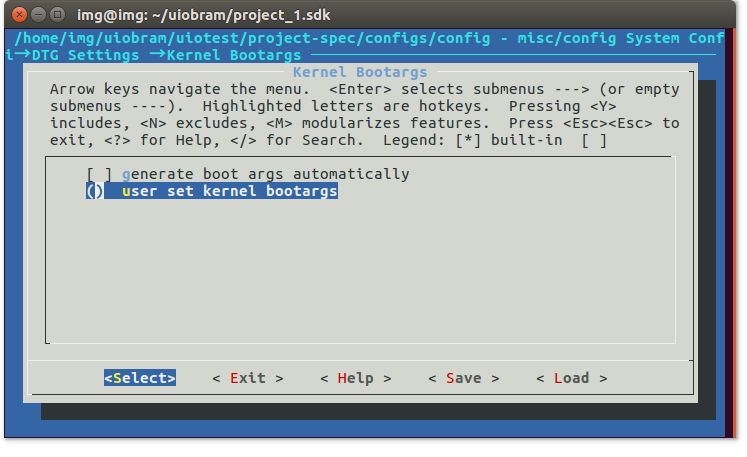
指令的意思是从该路径下选择硬件描述（hdf文件），后面的-p要指到你的工程名路径



完成后弹出如图所示配置界面

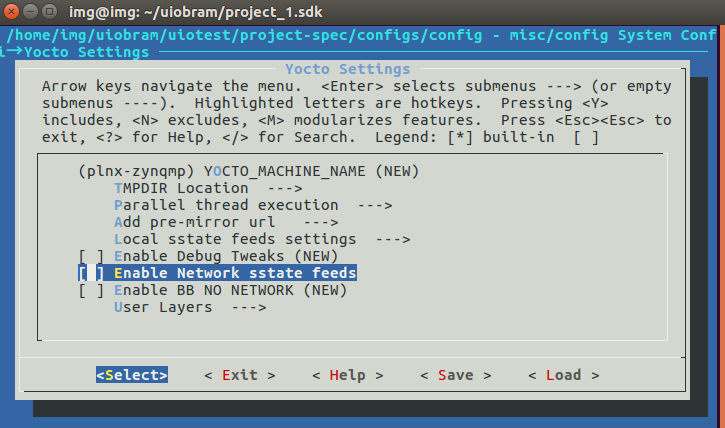


先进入DTG Settings



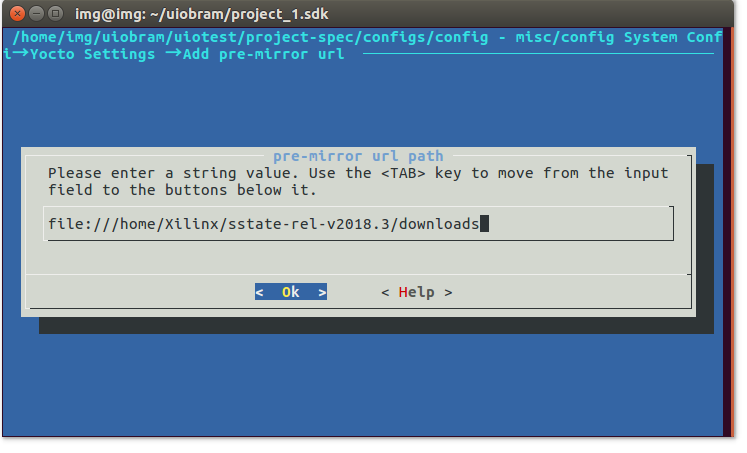
先进入上图所示进到kernel bootargs 按n取消generate boot args automatically,移动到下面这个并进入，输入：console=ttyPS0,115200 earlyprintk uio\_pdrv\_genirq.of\_id=”generic-uio”

切到exit退出到外层并重新选择如下的设置（在图的上方蓝色字迹显示路径）



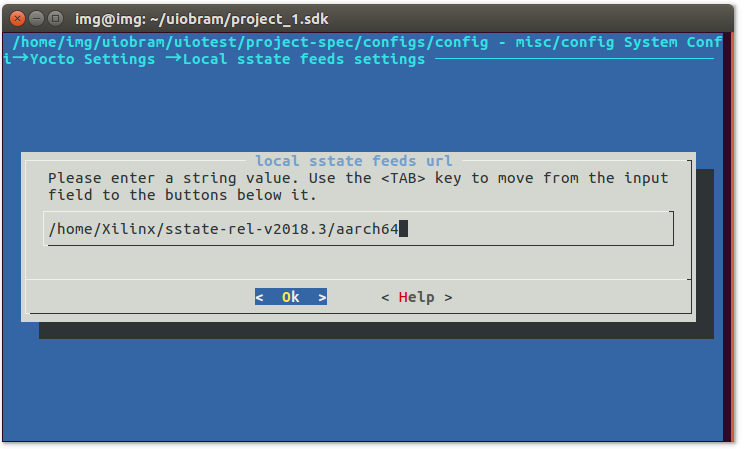
按n取消掉enable network sstate feeds

进入add pre-mirror url，输入：file///home/Xilinx/sstate-rel-v2018.3/downloads

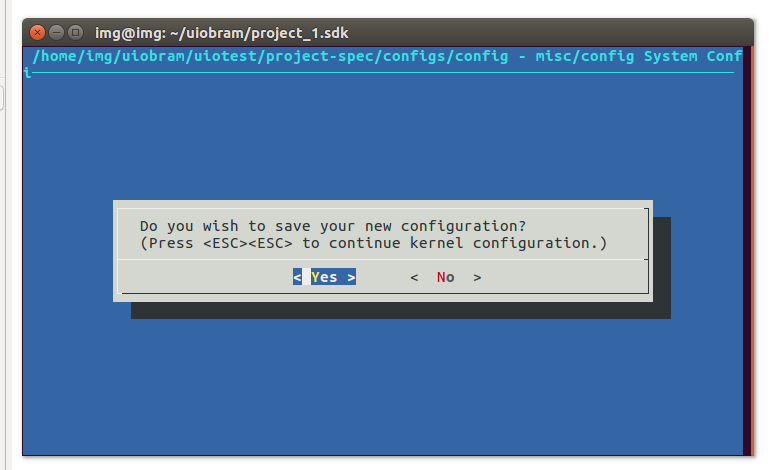


进入local sstate feeds settings, 输入： /home/Xilinx/sstate-rel/v2018.3/aarch64

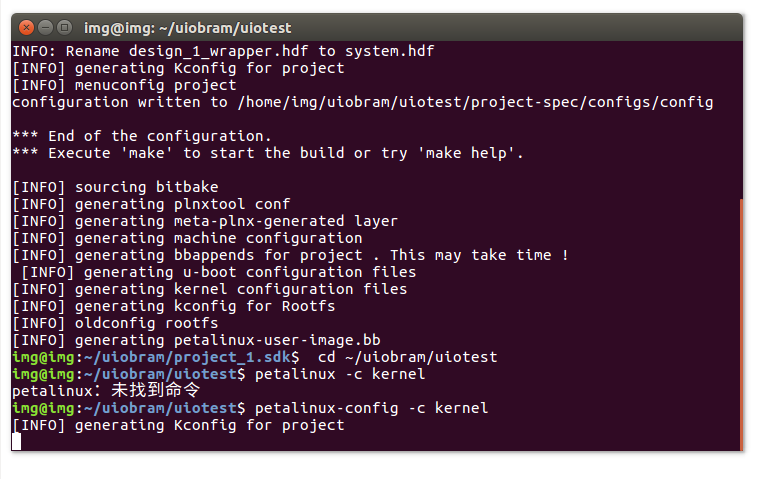
以上两条是由于实验室的那台linux主机的网络状况不佳，只能改用本地的一些库来进行编译



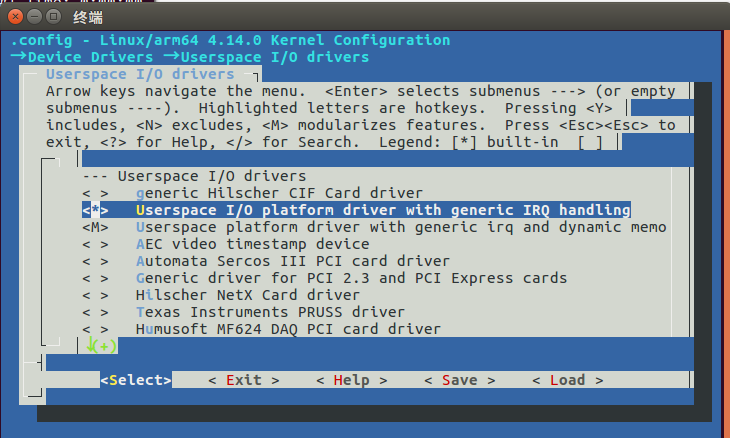
连续退出到如下界面，选择yes保存配置



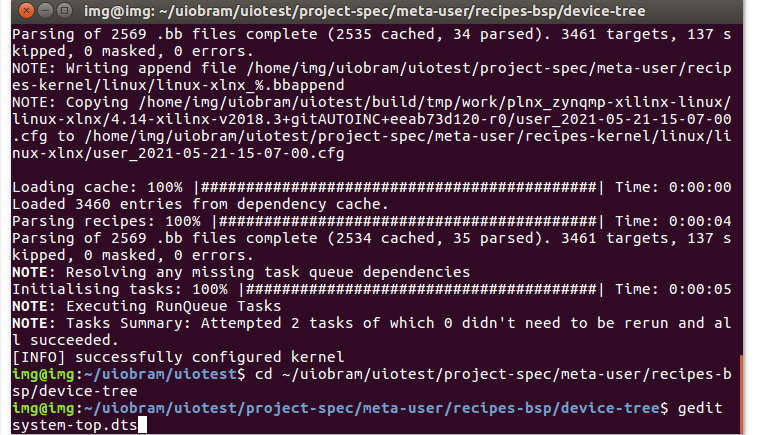
进入工程路径下，输入petalinux-config -c kernel进行进一步配置



在生成的配置界面中选择图片上方所示蓝色路径中的设置并使能（按y）光标所在选项

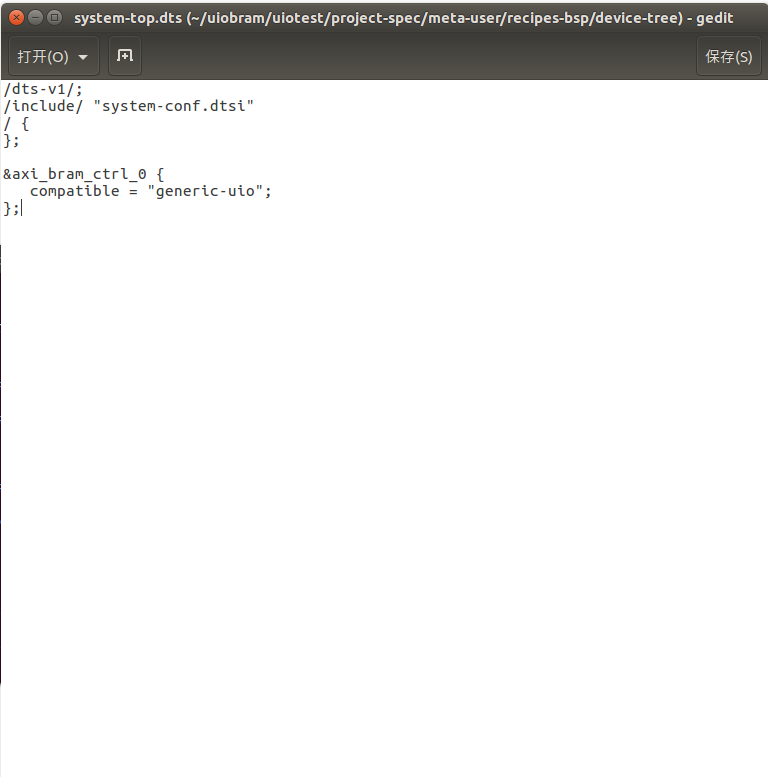


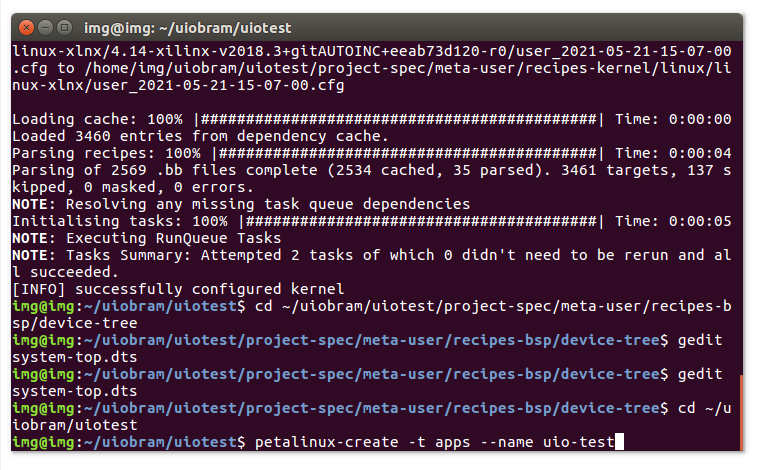
依旧是退出并且选yes保存设置



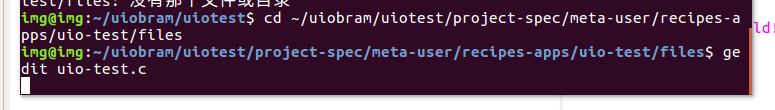
进入设备树路径并且用 gedit system-top.dts打开配置文件，输入如下图所示代码

其中axi\_bram\_ctrl\_0是bd起主要作用的器件名称

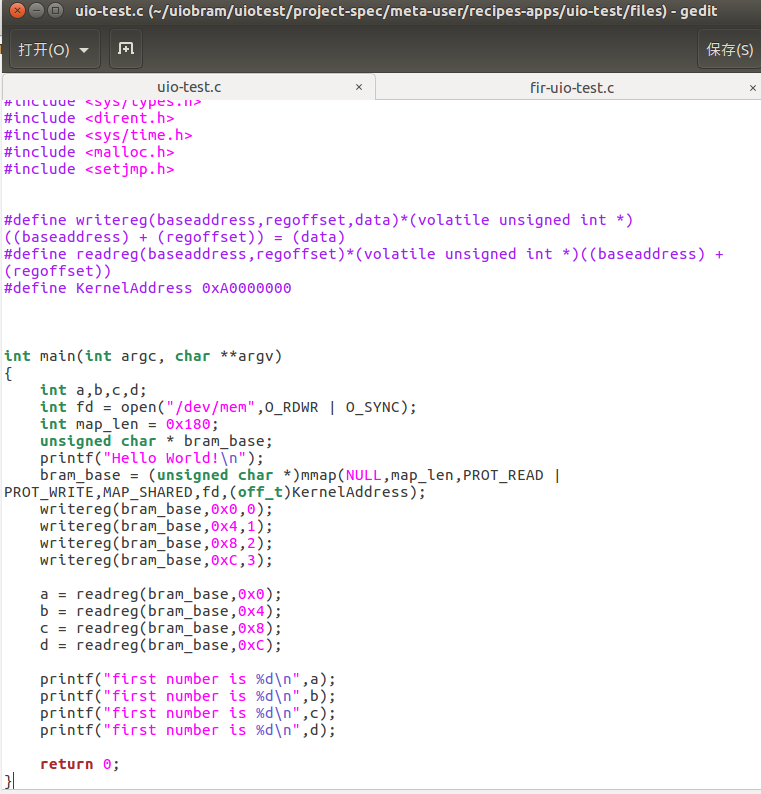
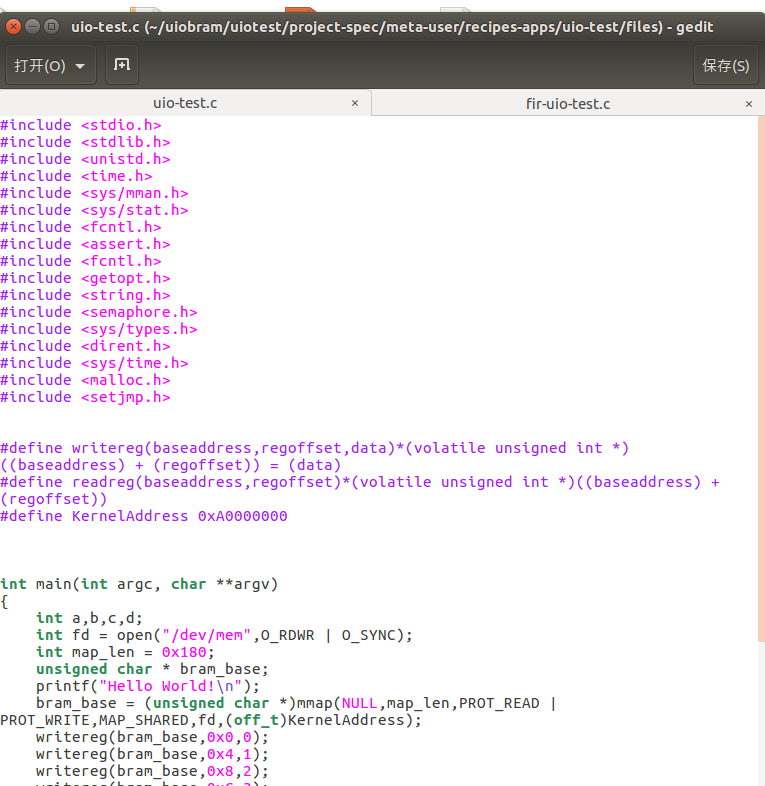




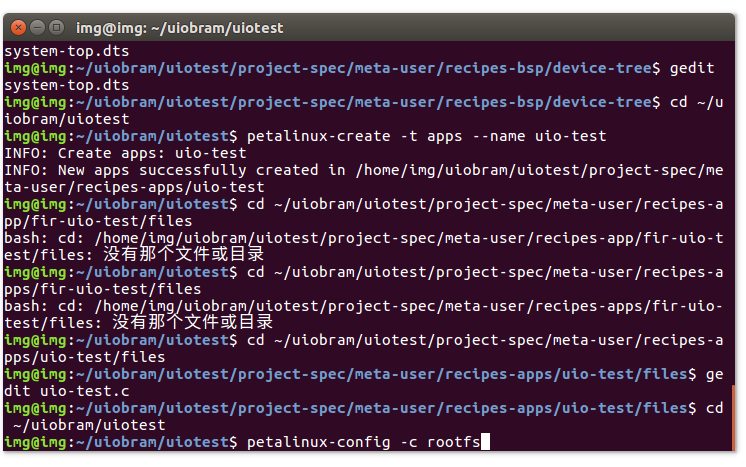
如上图所示建立一个新的app，名字中不可以有下划线！！！



载入这个app对应的路径并且打开app的c函数，进行编辑

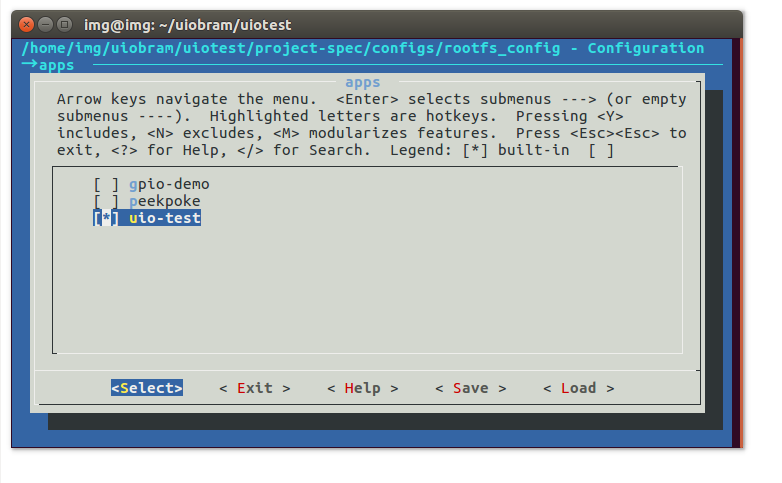


我这边就是随便写了个读写功能的，保存关闭



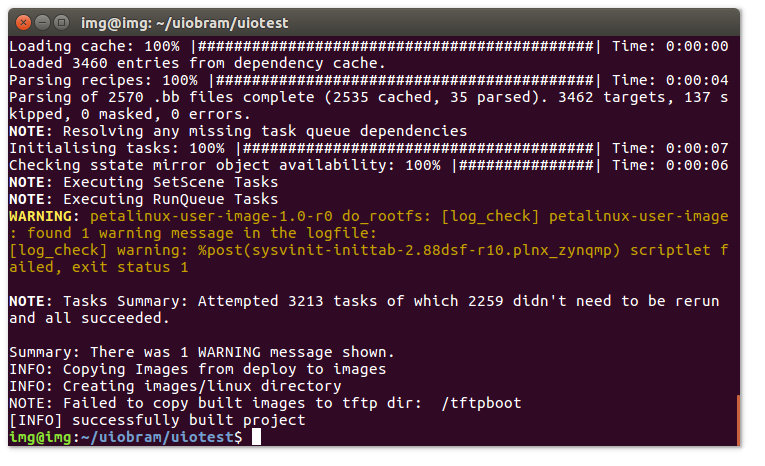
回到项目路径，输入：petalinux-config -c rootfs

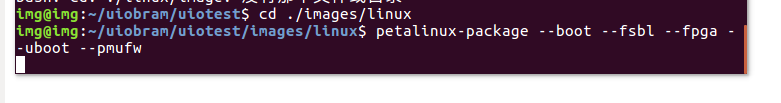
进去后载入如下所示设置，在刚才新建的app前面点y选择并退出保存



之后输入petalinux-build



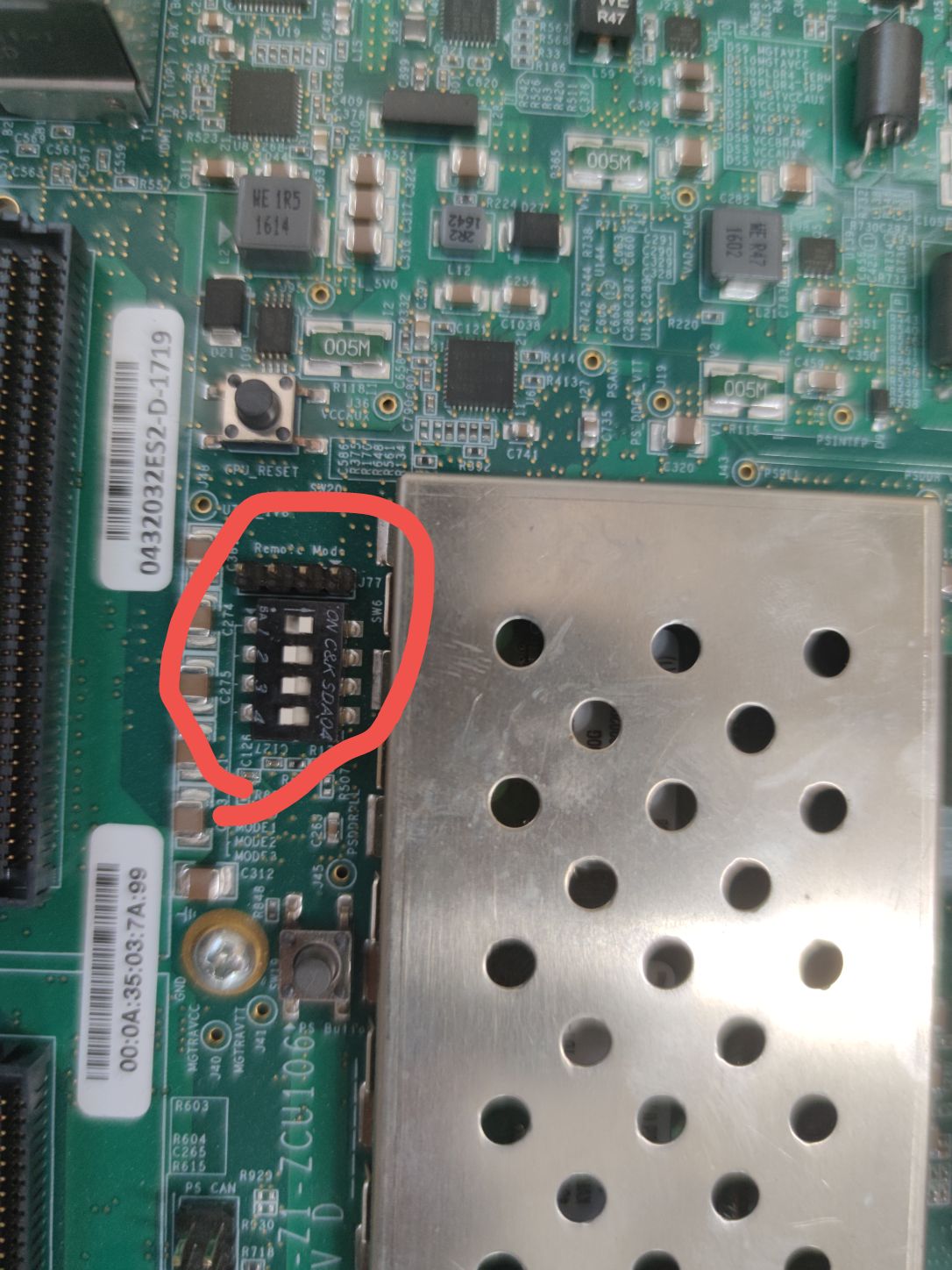




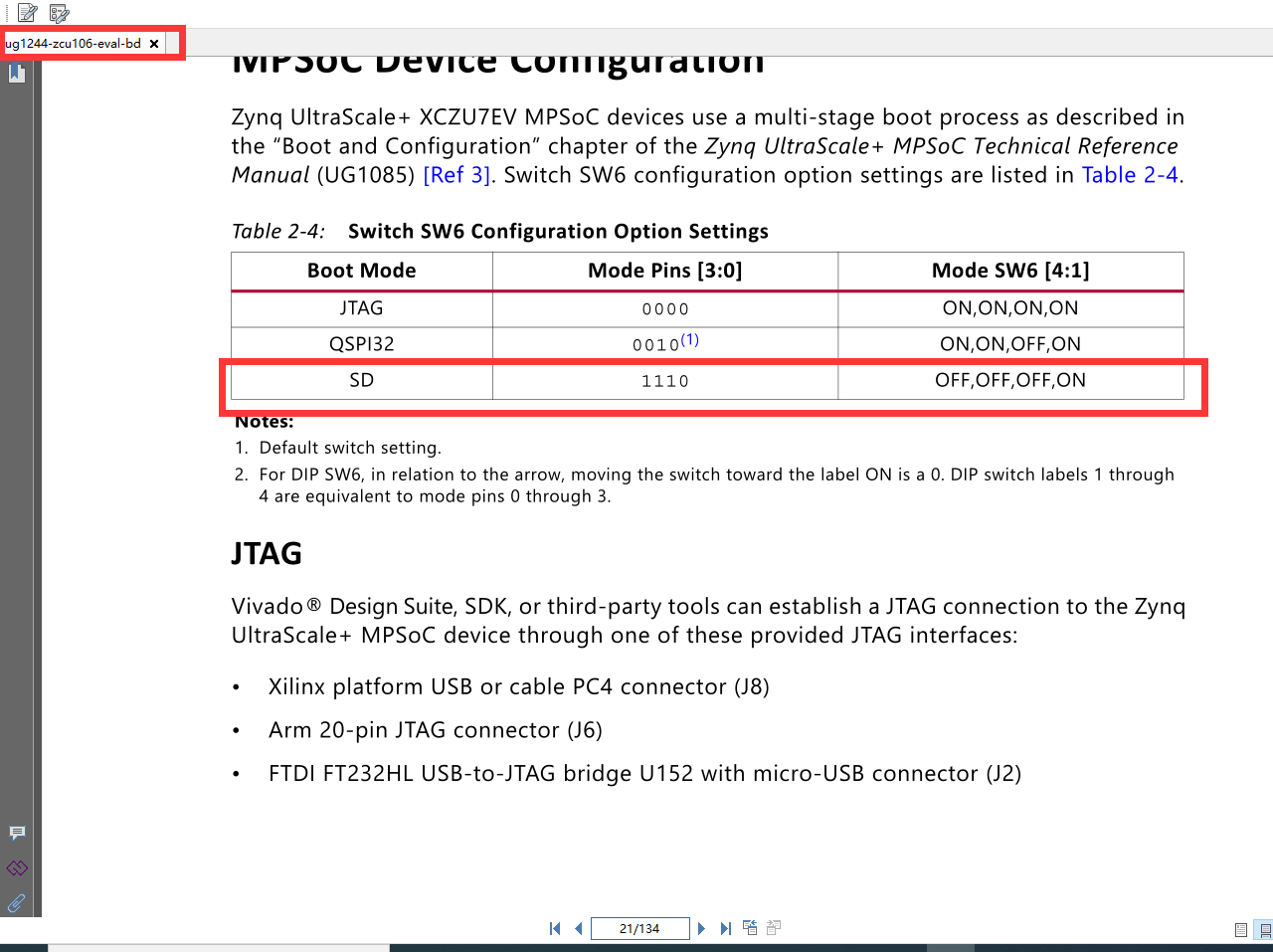
Build后输入如上所示指令打包，在该路径下生成boot.bin 和image.ub，拷到sd卡里，上板

1. Sd卡

首先记得把板子调到sd卡模式



这个玩意很坑爹，一定要记得换

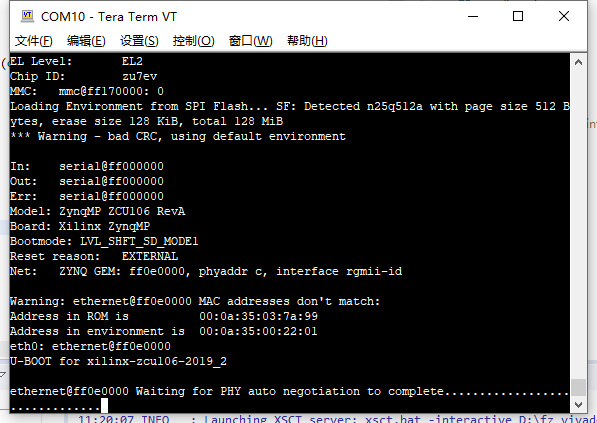


Zcu106的是在左上角红框内所示文档中的一个地方写的，切记他是从高管脚（3

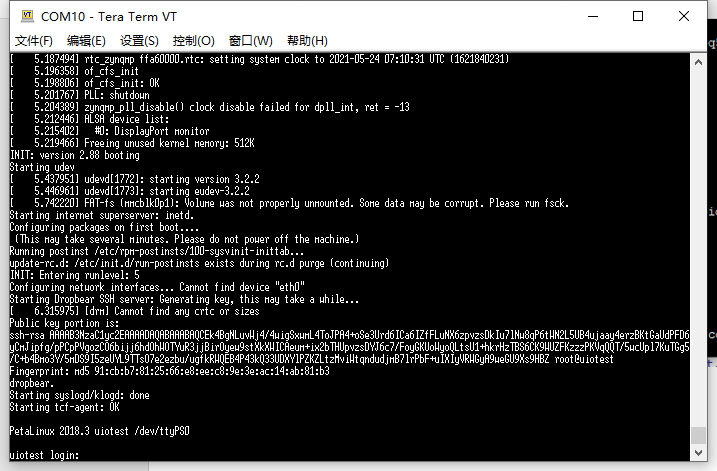
）到低管脚（0）的顺序为1110，别弄反了

然后先链接串口，调整波特率到115200

开电



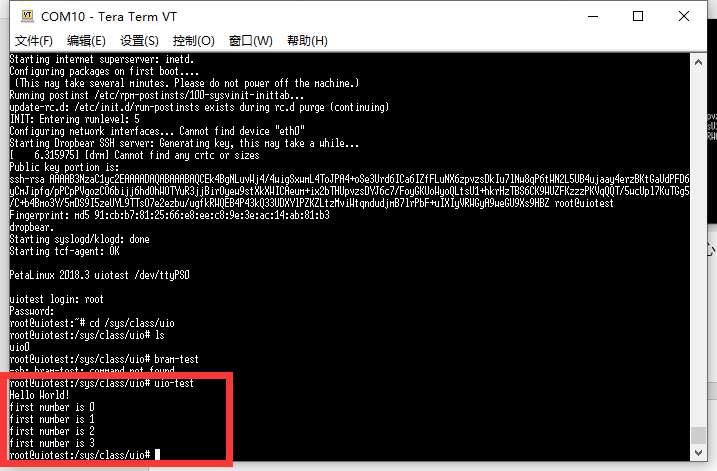
等一会之后变成这个



用户名密码都是默认root（输入密码时候不会出现“\*”，放心大胆输就行）

输入 cd /sys/class/uio ,再输入ls即可看到当前是哪个uio在工作

Cd 到初始路径，输入app名称，即可直接启动程序，看到如下所示输出



大功告成