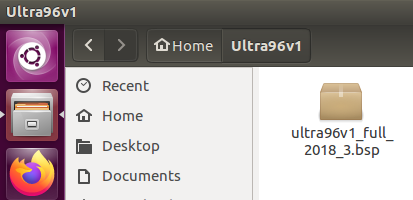
# Ultra96 BSP Petalinux镜像制作

### 第一部分 直接使用BSP镜像

#### 一、下载Ultra96的BSP

可以通过此链接下载| Zedboard <http://zedboard.org/support/design/24166/156> Ultra96\_BSP，其他开发板的BSP可以在官网下载。

打开虚拟机Ubutun 16.04, 创建文件夹，拷入BSP文件，如下图所示:



##### 二、建立环境变量，

source ~/petalinux2018.3/settings.sh

长久建立环境变量：

echo 'source ~/petalinux2018.3/settings.sh'>>~/.bashrc

或者编辑写入下面文件之一：

sudo vim ~/.bashrc

sudo vim /etc/profile

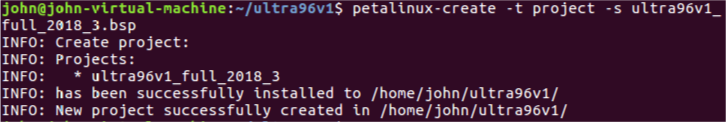
##### 三、创建工程

基于BSP，根据ug1144步骤操作即可, 创建一个基于官方硬件开发板的一个工程：

petalinux-create -t project -s ultra96v1\_full\_2018\_3.bsp

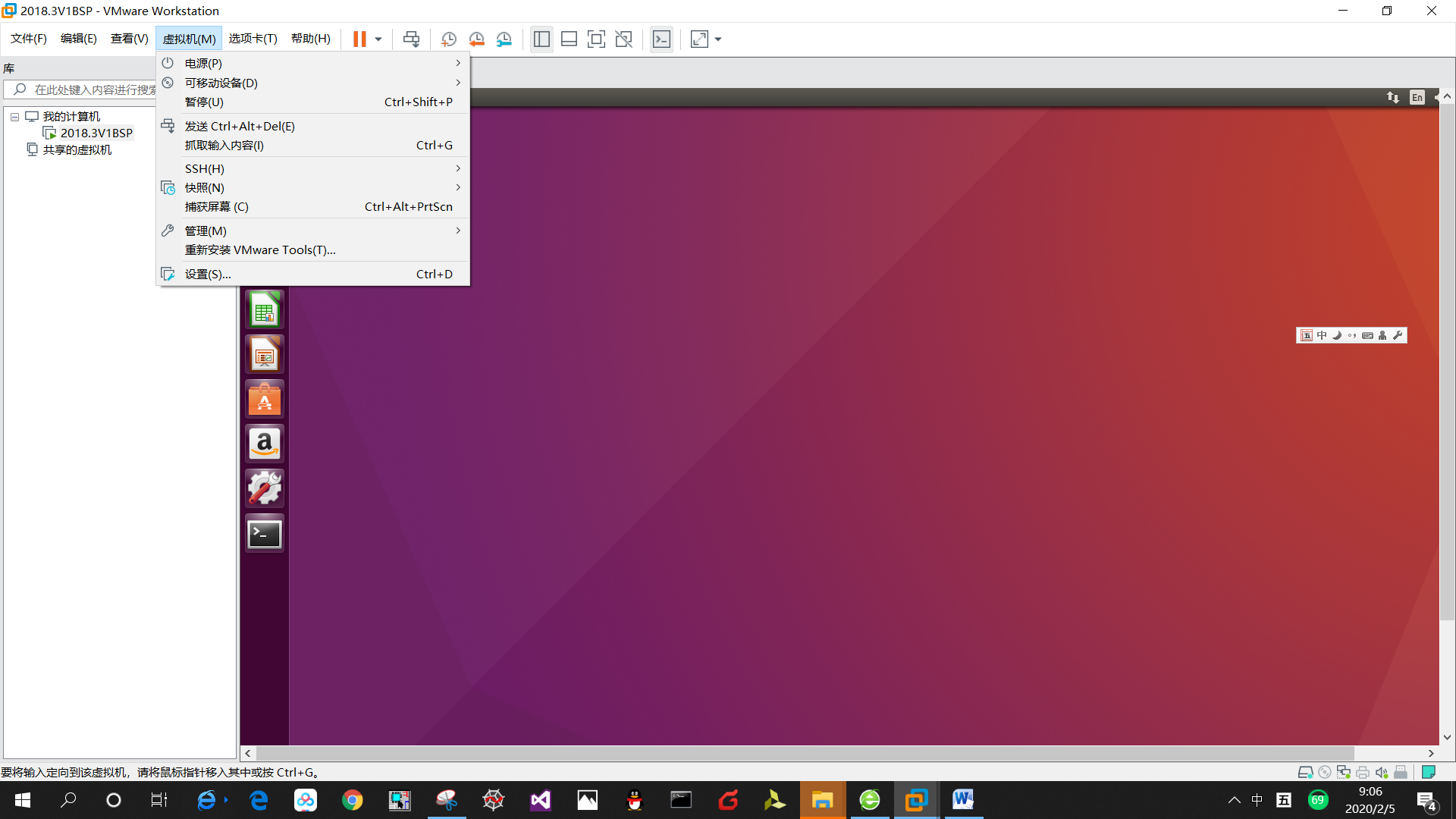
john@john-virtual-machine:~$ cd ultra96v1

john@john-virtual-machine:~/ultra96v1$ petalinux-create -t project -s /mnt/hgfs/xilinx/ultra96v1/ultra96v1\_full\_2018\_3.bsp



##### 四、虚拟机挂载TF卡

将TF卡连接虚拟机，在下图的可移动设备中选取TF卡，断开与主机连接，连接虚拟机



查看硬盘：

john@john-virtual-machine:~/ultra96v1$ sudo fdisk -l



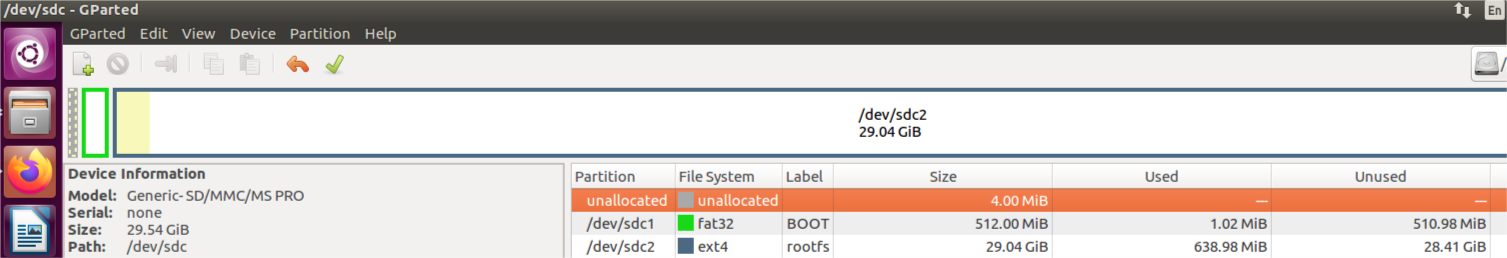
创建挂载文件夹：

john@john-virtual-machine:~/ultra96v1$ sudo mkdir /media/boot

john@john-virtual-machine:~/ultra96v1$ sudo mkdir /media/rootfs

##### 五、TF分区和拷贝

安装gparted软件，并进行分区，第一分区格式为FAT32，第二分区格式为ext4, 第一分区前预留4M。



挂载TF卡

john@john-virtual-machine:~/ultra96v1$ sudo mount /dev/sdc1 /media/boot

john@john-virtual-machine:~/ultra96v1$ sudo mount /dev/sdc2 /media/rootfs

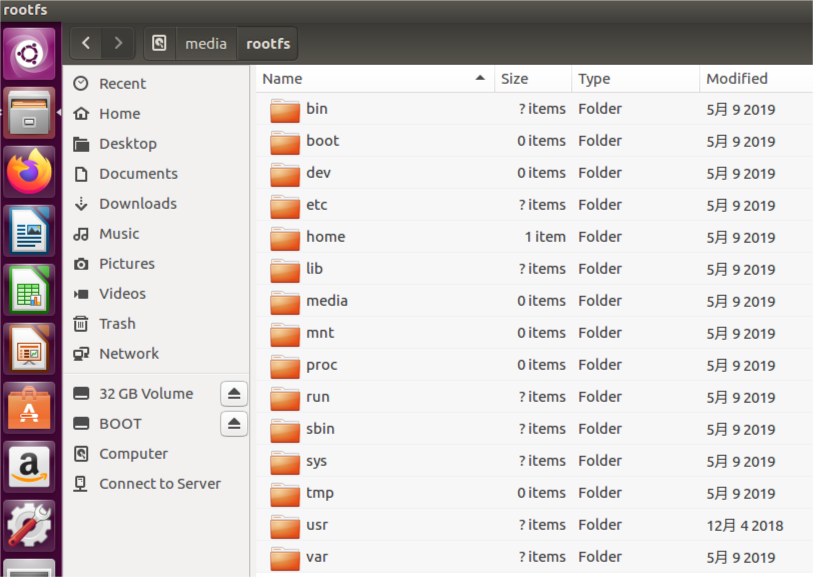
然后拷贝文件：

john@john-virtual-machine:~/ultra96v1$sudo cp /home/john/ultra96v1/ultra96v1\_full\_2018\_3/pre-built/linux/images/BOOT.BIN /media/boot

john@john-virtual-machine:~/ultra96v1$sudo cp /home/john/ultra96v1/ultra96v1\_full\_2018\_3/pre-built/linux/images/image.ub /media/boot

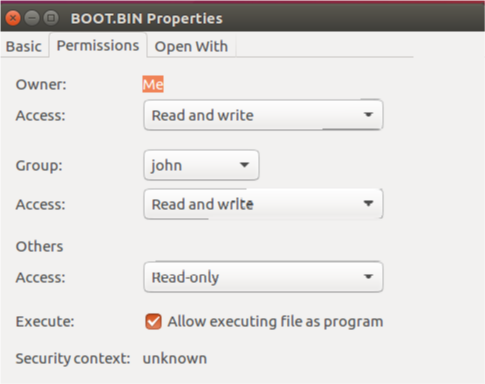
sudo tar zxvf /home/john/ultra96v1/ultra96v1\_full\_2018\_3/pre-built/linux/images/rootfs.ext4.gz -C /media/rootfs

解压后的TF目录根文件：



##### 五、检查BOOT.BIN文件属性

BOOT.BIN是否为可执行文件，若不是，请下面属性处打勾：



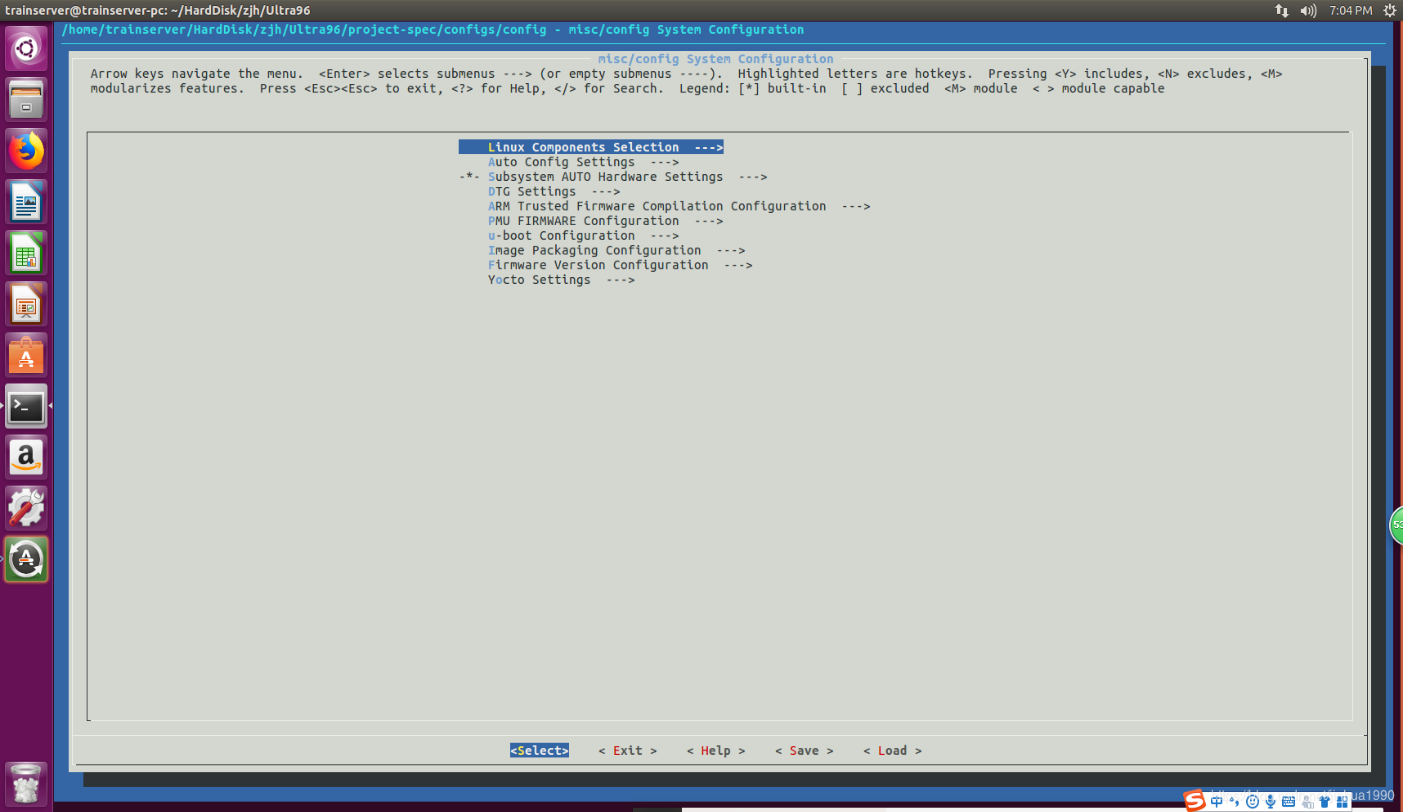
##### 六、卸载TF卡

##### john@john-virtual-machine:~/ultra96v1$ sudo umount /media/boot

### 第二部分 BSP生成镜像

##### 一、输入命令进行配置

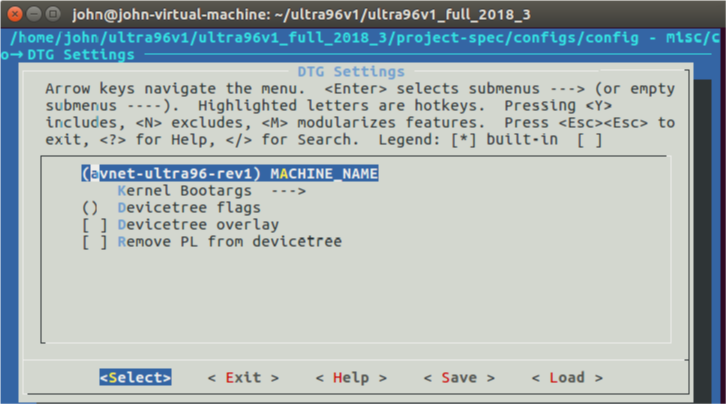
petalinux-config



图一

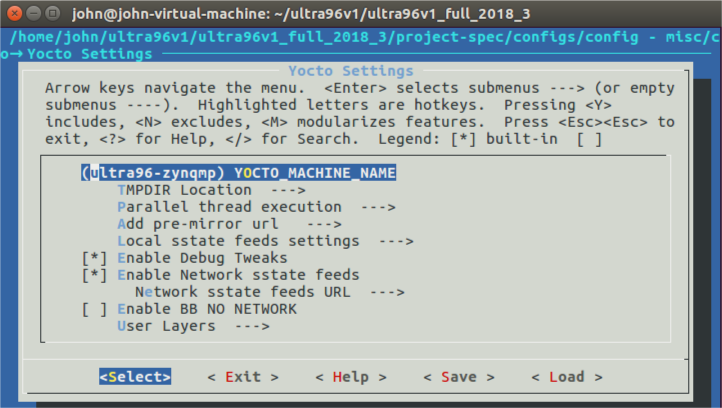
##### 二、DTG Setting

图一第三项DTG Setting, 输入avnet\_ultra96\_rev1，如果你用的是BSP，不用更改。



##### 三、Yocto Settings

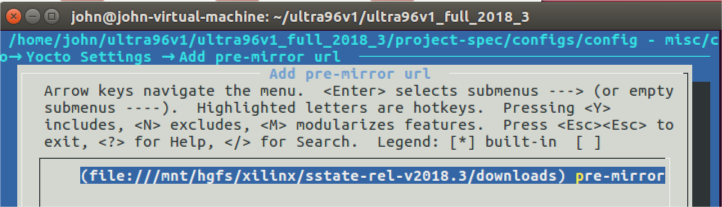
打开图一最后一项Yocto Settings,

其中第三项Add pre-mirror url：

http://petalinux.xilinx.com/sswreleases/rel-v${PETALINUX\_VER}/downloads

修改为file://<path>/downloads，<path>为sstate下载包解压后的地址，如：

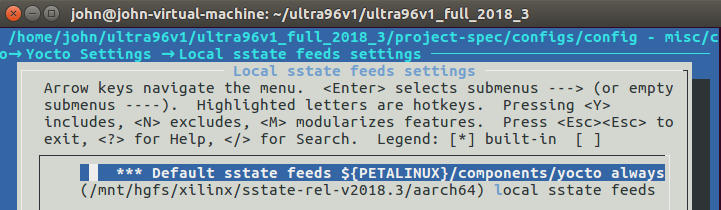
file:///mnt/hgfs/xilinx/sstate-rel-v2018.3/downloads



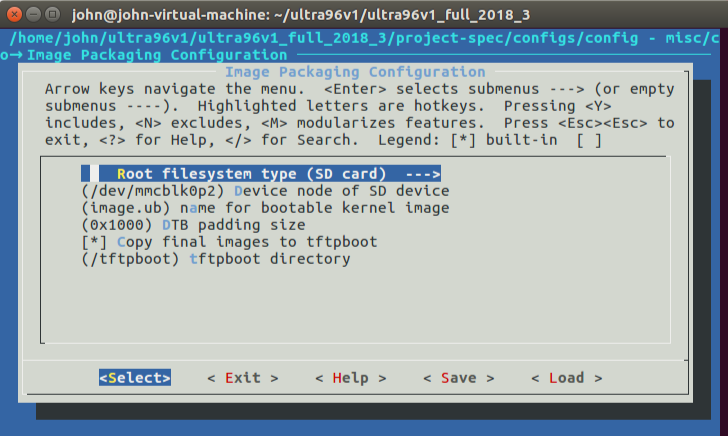
第四项Local sstate feeds settings：

修改为：<path>/aarch64(arm/mb-full)，<path>为sstate下载包解压后的地址，如：

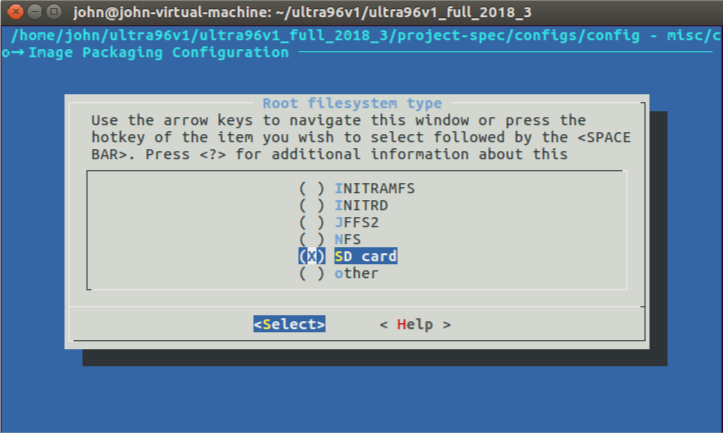
/mnt/hgfs/xilinx/sstate-rel-v2018.3/aarch64



##### 四、检查根文件格式设置，进入Image Packaging Configuration,



勾选SD card



##### 五、执行petalinux-build编译及问题分析

执行petalinux-build编译命令，若需要重新编译，可考虑是否需要执行以下两个清除命令：

Clean all build collaterals. It removes build/, ${TMPDIR} and images. This will bring the project to its initial state  
$ petalinux-build -x mrproper

Clear the build area of the PetaLinux project for archiving as a BSP or for revision  
control. This example retains the images directory of the project  
$ petalinux-build -x distclean

以下是编译过程中可能出现的问题：

###### 5.1 编译时间过长，内存溢出

请参考我的博客

xilinx ultra96 2018.3以上版本BSP制作镜像时运行时间过长、内存耗尽的分析和解决\_u010879745的博客-CSDN博客

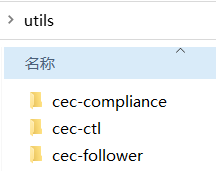
https://blog.csdn.net/u010879745/article/details/104136726

###### 5.2 找不到源文件

2018.3与2018.2版本在编译时在以下三个文件中均会出现缺少源文件cec-ctl.cpp, cec-compliance.cpp, cec-fellow.cpp, 解决方案见我在xilinx论坛上的post,



在/home/john/ultra96v2/v2/build/tmp/work/aarch64-xilinx-linux/v4l-utils/1.12.3-r0/文件夹下找到这些缺失的文件，



将其拷贝到文件缺失的以下三个目录中

/home/john/ultra96v2/v2/build/tmp/work/aarch64-xilinx-linux/v4l-utils/1.12.3-r0/build/utils/cec-ctl

/home/john/ultra96v2/v2/build/tmp/work/aarch64-xilinx-linux/v4l-utils/1.12.3-r0/build/utils/cec-compliance

/home/john/ultra96v2/v2/build/tmp/work/aarch64-xilinx-linux/v4l-utils/1.12.3-r0/build/utils/cec-follower

例如某个目录拷贝完的文件：

##### 

###### 5.3 无线驱动无法下载

V2 版从网上下载无线配置遇到这个问题，V1没有无线配置没有这个问题

##### 

##### 

WARNING: wilc-15.2-r0 do\_fetch: Failed to fetch URL git://github.com/avnet/u96v2-wilc-driver;protocol=http;branch=v15\_2, attempting MIRRORS if available

ERROR: wilc-firmware-15.2-r0 do\_fetch: Function failed: base\_do\_fetch

ERROR: Task (/home/john/ultra96v2/v2/project-spec/meta-user/recipes-bsp/wilc-firmware/wilc-firmware\_15.2.bb:do\_fetch) failed with exit code '1'

export HOME="/home/john"; git -c core.fsyncobjectfiles=0 ls-remote http://github.com/linux4wilc/firmware failed with exit code 128, output:

fatal: unable to access 'http://github.com/linux4wilc/firmware/': Recv failure: Connection reset by peer

找到下面的无线配置文件，

##### 

打开文件，找到下面的段落：

SUMMARY = "wilc-firmware: Firmware binaries for Microchip WILC1000/WILC3000"

SECTION = "PETALINUX/modules"

LICENSE = "GPLv3"

LIC\_FILES\_CHKSUM = "file://${COMMON\_LICENSE\_DIR}/GPL-3.0;md5=c79ff39f19dfec6d293b95dea7b07891"

SRC\_URI = "git://github.com/linux4wilc/firmware;protocol=http;tag=wilc\_linux\_15\_2"

**将文件中的协议protocol=http;改为protocol=https;**

##### 六、打包 BOOT.BIN

通过以上修改，petalinux-build应能编译通过，之后打包BOOT.BIN

petalinux-package --boot --fsbl zynqmp\_fsbl.elf --u-boot u-boot.elf --pmufw pmufw.elf --fpga system.bit –force

##### 七、拷贝文件至TF卡

请参考第一部分的第四章和第五章。

Ultra96基础学习篇——（5）PetaLinux创建BOOT.bin\_jinhua1990的博客-CSDN博客

<https://blog.csdn.net/jinhua1990/article/details/89567386>

<https://www.element14.com/community/community/design-challenges/path2programmable/blog/2019/12/04/p2p-petalinux-labs>

<https://www.96boards.org/documentation/consumer/ultra96/ultra96-v1/build/peta-linux.md.html>