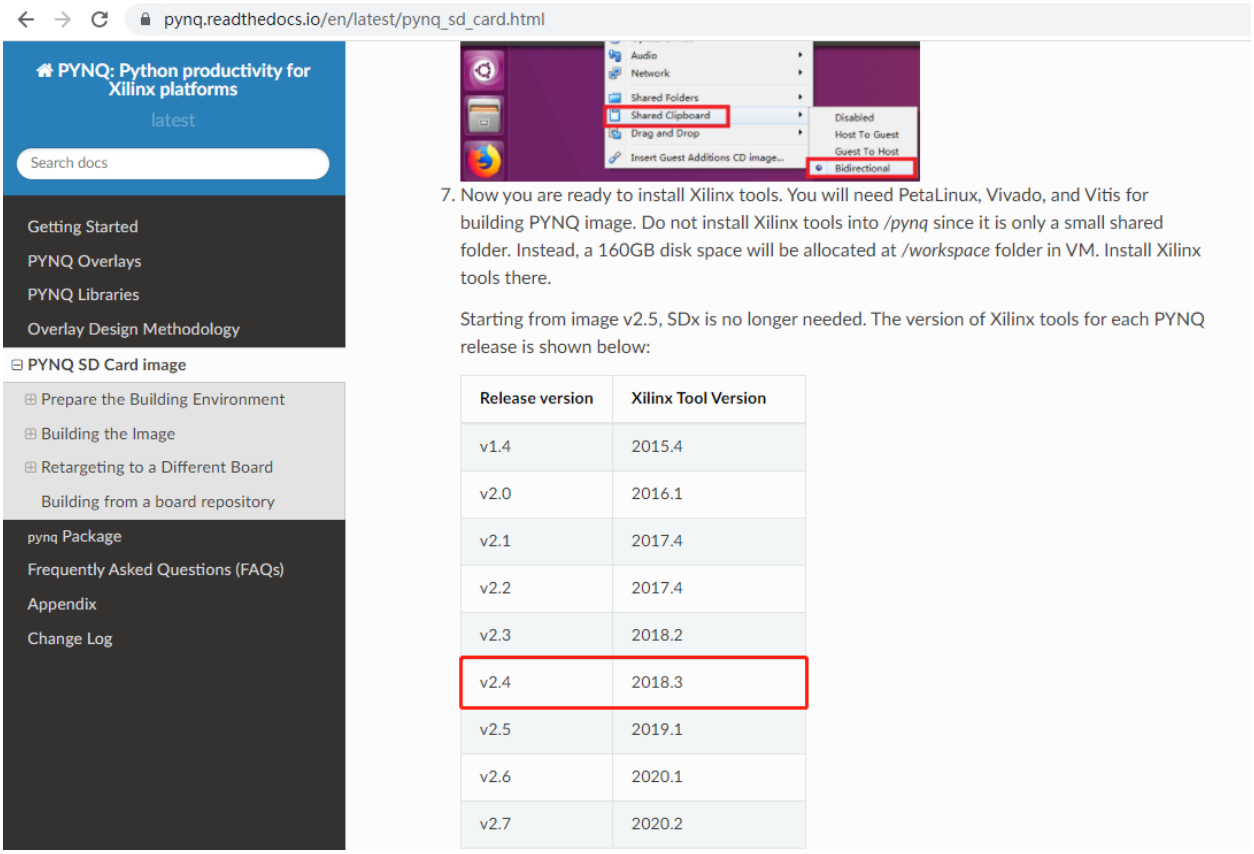
[L16]PYNQ上手

PYNQ实际是基于ZYNQ PetaLinux上做的Python环境,所以还是沿用老的bd文件生成的bitstream就可以了,内核也没变化,对于之前有困难的,务必熟练掌握,再继续以下内容,生成bitstream过程,当然,PYNQ需要的环境比较复杂,需要在指定版本的Ubuntu(比如目前支持的最高版本Ubuntu 16.04 LTS)安装Vivado,PetaLinux SDK等等.

PYNQ官方文档页:https://pynq.readthedocs.io/en/v2.4/pynq_sd_card.html (请注意选择合适的版本!)

查表得知,我们要用2.4版本的PYNQ.



可是找Rootfs Image就没那么简单了,链接如下:

https://www.xilinx.com/member/forms/download/xef.html?filename=pynq_rootfs_arm_v2.4.zip

接着还要克隆PYNQ的库,当然记得切换分支.

```
git clone https://github.com/xilinx/PYNQ.git
cd PYNQ
git checkout v2.4
```

```
taterli@taterli-VirtualBox:~$ git clone https://github.com/xilinx/PYNQ.git
Cloning into 'PYNQ'...
remote: Enumerating objects: 26310, done.
remote: Total 26310 (delta 0), reused 0 (delta 0), pack-reused 26310
Receiving objects: 100% (26310/26310), 164.40 MiB | 12.98 MiB/s, done.
Resolving deltas: 100% (15700/15700), done.
Checking connectivity... done.
taterli@taterli-VirtualBox:~$ cd PYNQ
taterli@taterli-VirtualBox:~/PYNQ$ git checkout v2.4
Note: checking out 'v2.4'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at d9c7521... Add register map text to the overlay tutorial (#821)
taterli@taterli-VirtualBox:~/PYNQ$
```

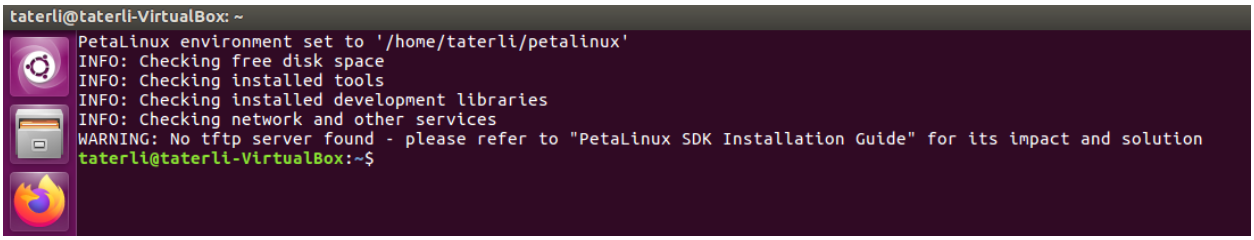
检查并安装一些环境相关的包:

```
HEAD is now at d9c7521... Add register map text to the overlay tutorial (#821)
taterli@taterli-VirtualBox:~/PYNQ$ cd sdbuild/scripts/
taterli@taterli-VirtualBox:~/PYNQ/sdbuild/scripts$ ls
check_Env.sh     cleanup.sh     create_mount_img.sh   create_rootfs.sh    image_from_prebuilt.sh  kill_chroot_processes.sh  qemu.patch
check_mounts.sh  create_bsp.sh  create_partitions.sh  git_clone_checkout  install_packages.sh     mount_image.sh            resize_umount.s
taterli@taterli-VirtualBox:~/PYNQ/sdbuild/scripts$ ./setup_host.sh
# ... 省略大量内容
# ... 省略大量内容
# ... 省略大量内容
# ... 省略大量内容
# ... 省略大量内容
strip "/opt/qemu/bin/qemu-aarch64"
+ cd /opt/qemu/bin
+ sudo rm -rf qemu-arm-static qemu-aarch64-static
+ sudo ln -s qemu-arm qemu-arm-static
+ sudo ln -s qemu-aarch64 qemu-aarch64-static
+ cd /home/taterli
+ cd /usr/bin
+ which gmake
+ sudo ln -s make gmake
+ echo 'PATH=/opt/qemu/bin:/opt/crosstool-ng/bin:$PATH'
+ echo 'Now install Vivado, SDx, and Petalinux.'
Now install Vivado, SDx, and Petalinux.
+ echo 'Re-login to  ensure the enviroment is properly set up.'
Re-login to  ensure the enviroment is properly set up.
taterli@taterli-VirtualBox:~/PYNQ/sdbuild/scripts$
```

按照他要求设置一下bash环境,把以下内容塞到~/.bashrc文件中,当然要根据实际改变哦.

```
source /home/taterli/petalinux/settings.sh # PetaLinux SDK
source /home/taterli/Xilinx_Vivado_SDK/Vivado/2018.3/settings64.sh # Vivado 本体
source /home/taterli/Xilinx_Vivado_SDK/SDK/2018.3/settings64.sh # Vivado 本体
export PATH=/opt/qemu/bin:/opt/crosstool-ng/bin:$PATH # 调试工具
```

设置好后重启一下Shell,这时候启动Shell按道理就会自动执行一些检查和环境配置.



接着我们做编译前准备工作:

1. 首先是在PYNQ/sdbuild/下面新建一个prebuilt文件夹.
2. 将bionic.arm.2.3.img文件拷贝至该文件夹,这个由pynq_rootfs_arm_v2.4.zip解压得到.
3. 在PYNQ/boards/目录新建一个文件夹TATERLI-Z7.
4. TATERLI-Z7中再新建两个文件夹分别为base和petalinux_bsp.
5. 在petalinux_bsp中再建立文件夹hardware_project.
6. 修改bitstream文件为base.bit并放到刚才创建的base目录里.
7. 复制system.hdf到刚才创建的hardware_project目录里.
8. 复制PYNQ/boards/Pynq-Z2中的Pynq-Z2.spec到TATERLI-Z7并改名为TATERLI-Z7.spec.
9. 编辑TATERLI-Z7.spec,并改成如下内容.

```
ARCH_TATERLI-Z7 := arm
BSP_TATERLI-Z7 :=
BITSTREAM_TATERLI-Z7 := base/base.bit
STAGE4_PACKAGES_TATERLI-Z7 := pynq ethernet
```

然后回到sdbuild目录开始构建.

```
make BOARDS=TATERLI-Z7 PREBUILT=./prebuilt/bionic.arm.2.4.img
```

如果出现类似如下问题,可以对Makefile动刀一下.

```
taterli@taterli-VirtualBox:~/PYNQ/sdbuild$ make BOARDS=TATERLI-Z7 PREBUILT=./prebuilt/bionic.arm.2.4.img
/opt/qemu/bin/qemu-arm-static -version | fgrep 2.8.0
qemu-arm version 2.8.0
which vivado | fgrep 2018.3
/home/taterli/Xilinx_Vivado_SDK/Vivado/2018.3/bin/vivado
which sdx | fgrep 2018.3
Makefile:309: recipe for target 'checkenv' failed
make: *** [checkenv] Error 1
taterli@taterli-VirtualBox:~/PYNQ/sdbuild$
```

确定所有包没问题,可以把环境检查关闭掉,因为有时候他的判断就有毛病,当然确保版本安装都要正确无误.

```
checkenv: $(patsubst %, qemu_check_%, $(USED_ARCH))
        which arm-linux-gnueabihf-gcc
        which microblaze-xilinx-elf-gcc
        which ct-ng
        which python | fgrep /usr/bin/python
        sudo -n mount > /dev/null
        bash $(SCRIPT_DIR)/check_Env.sh
        bash $(SCRIPT_DIR)/check_mounts.sh

boot_files: checkenv $(BOOT_FILES)

images: checkenv $(IMAGE_FILES)

nocheck_images: $(IMAGE_FILES)

real_all: checkenv $(BOOT_FILES) $(IMAGE_FILES)

sdx_sw: $(patsubst %, sdx_sw_%, $(BOARDS))

sysroot: $(patsubst %, sysroot_%, $(USED_ARCH))

bsp: $(patsubst %, bsp_%, $(BOARDS))

# Default package clean target
PACKAGE_CLEAN_%: ;

clean: $(PACKAGE_CLEAN)
        -rm -rf $(BUILD_ROOT)
        -rm -rf $(IMAGE_ROOT)

.PHONY: bsp boot_files images all clean real_all checkenv sdx_sw sysroot nocheck_images

~
"Makefile" 339L, 11835C written
```

最后得到IMG文件是可以直接写入到TF卡的.

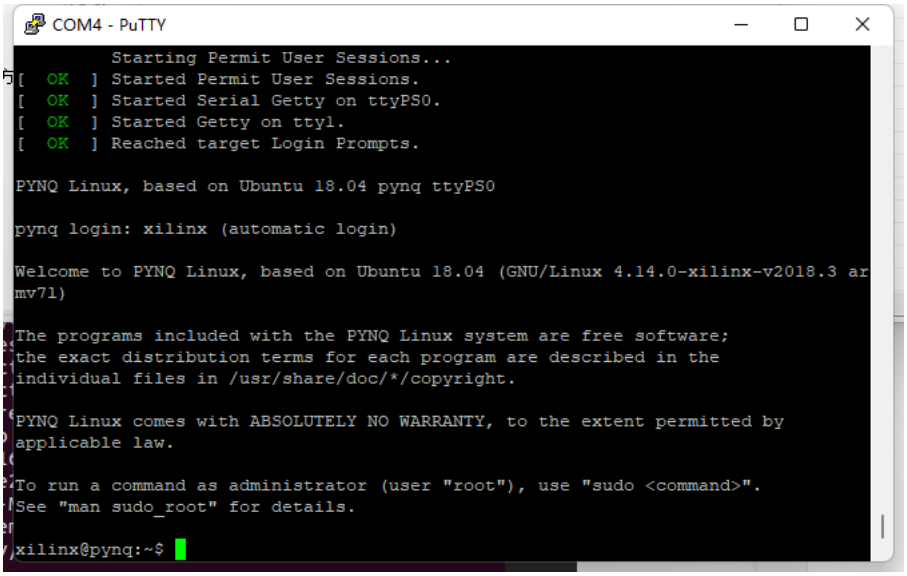
```
Command (m for help): Disk /home/taterli/PYNQ/sdbuild/output/TATERLI-Z7-2.4.img: 7 GiB, 7516192768 bytes, 14680064 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xa51ca6d9

Device                                          Boot  Start      End  Sectors  Size Id Type
/home/taterli/PYNQ/sdbuild/output/TATERLI-Z7-2.4.img1 *      2048   206847   204800  100M  c W95 FAT
/home/taterli/PYNQ/sdbuild/output/TATERLI-Z7-2.4.img2      206848 10742887 10536040    5G 83 Linux

Command (m for help): The partition table has been altered.
Syncing disks.

+ total_size=5371444
+ truncate -s 5371444K /home/taterli/PYNQ/sdbuild/output/TATERLI-Z7-2.4.img
taterli@taterli-VirtualBox:~/PYNQ/sdbuild$ cp output/^C
taterli@taterli-VirtualBox:~/PYNQ/sdbuild$ sudo cp output/TATERLI-Z7-2.4.img /media/sf_ZYNQ-FILES/
```

可以启动但是似乎一些外设无法使用,如果不能启动先检查前面步骤.

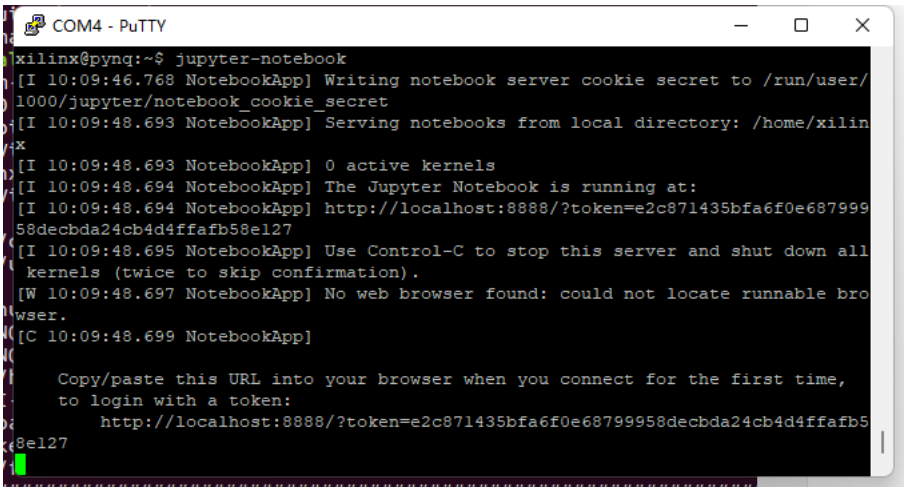


切换到目录~/PYNQ/sdbuild/build/TATERLI-Z7/petalinux_project然后按照之前替换内核,dts等内容.

```
petalinux-config --get-hw-description=../petalinux_bsp/hardware_project/
petalinux-config -c kernel
vim ~/PYNQ/sdbuild/build/TATERLI-Z7/petalinux_project/project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi
```

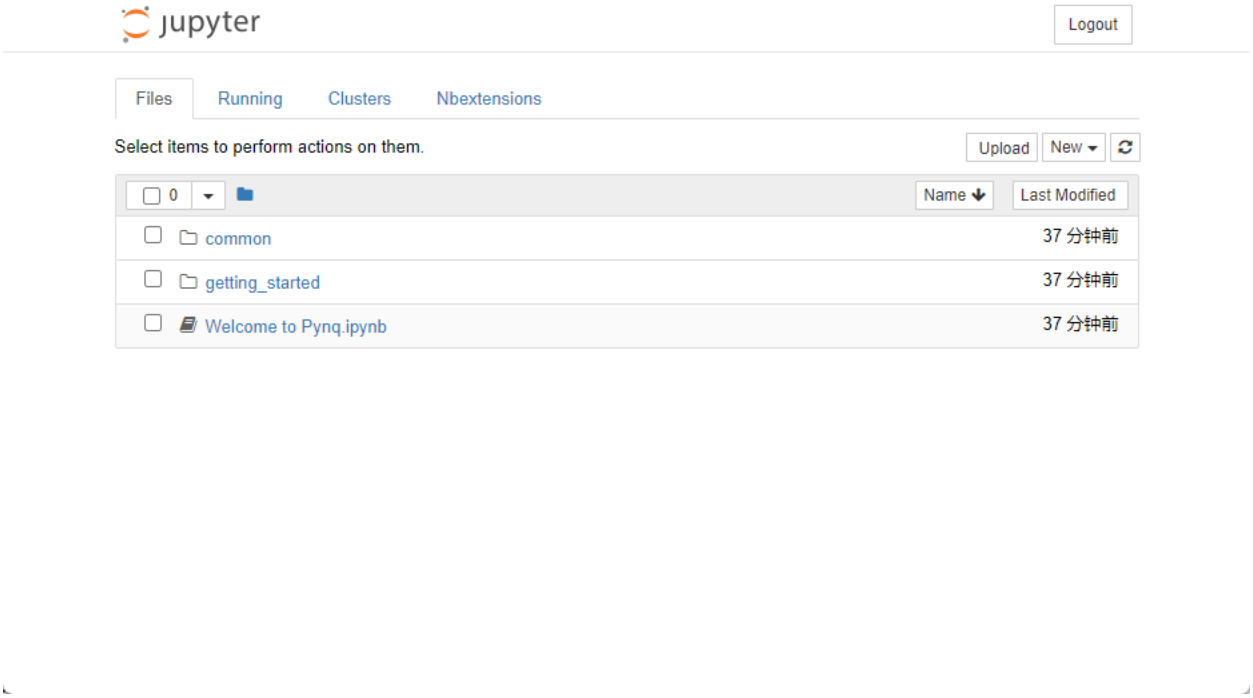
再次编译后网卡USB都好用了,默认已经有jupyter-notebook,我们启动它.

```
jupyter-notebook
```



比如上面地址我要换成这个格式,替换成实际IP地址以及端口换成9090.默认密码xilinx.

```
http://192.168.31.32:9090/?token=e2c871435bfa6f0e68799958decbda24cb4d4ffafb58e127
```



截止到这一节,对ZYNQ的基础已经入门了,之后就会深化每个细节展开学习.