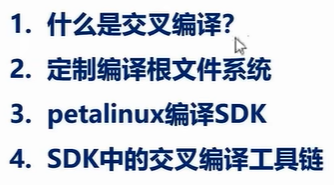
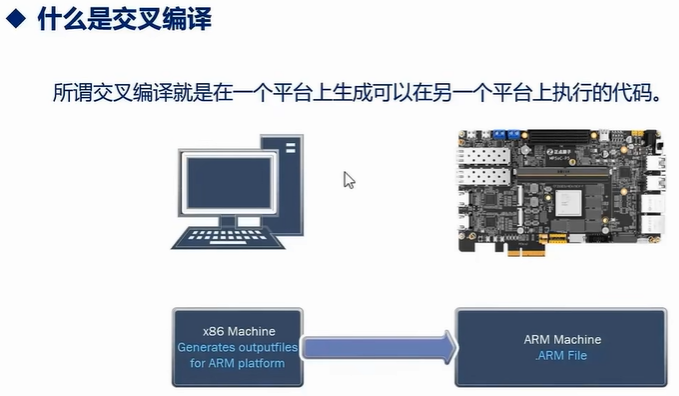
IMG_256

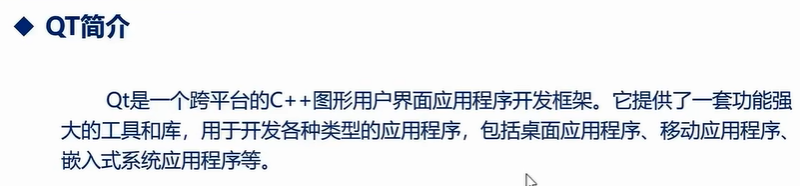


1. 什么是交叉编译？



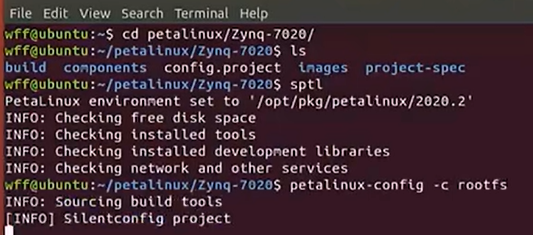


1. 定制编译根文件系统：

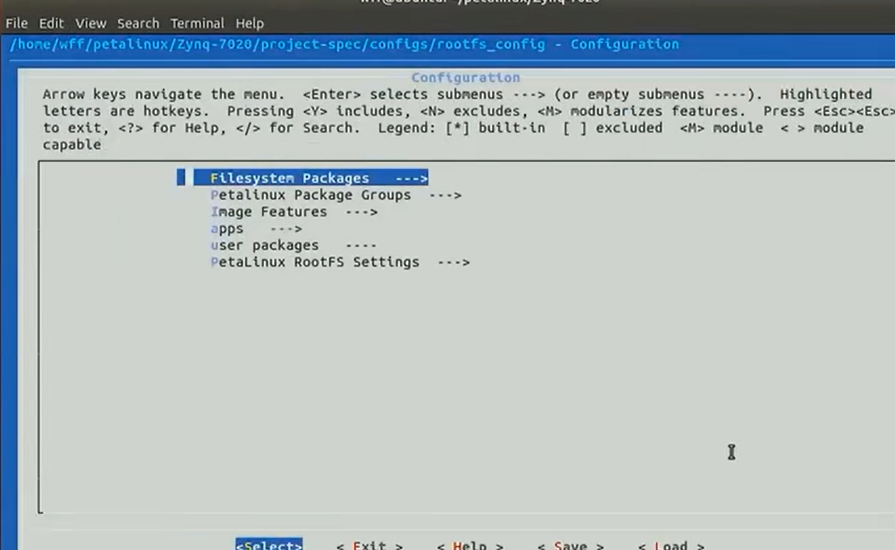






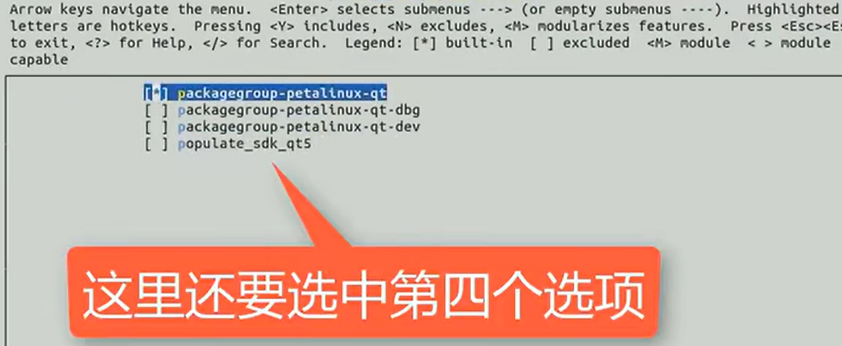


我们在如下图形界面配置工具库：



添加qt库：

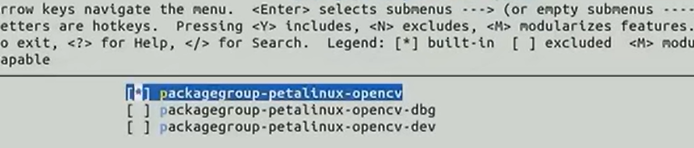
Petalinux package Groups -> packagegroup-petalinux-qt -> 选中packagegroup-petalinux-at:



返回上一级：

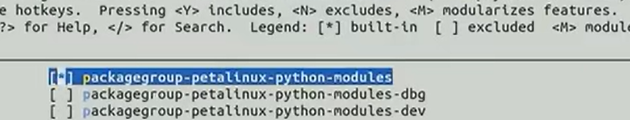
添加opencv库：

Petalinux package Groups -> packagegroup-petalinux-opencv -> 选中packagegroup-petalinux-opencv：



添加python库：

Petalinux package Groups -> packagegroup-petalinux-python-modules -> 选中packagegroup-petalinux-python-modules：



/\*

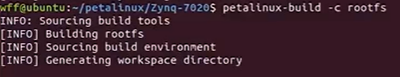
简单介绍以下其他的工具库：

Petalinux package Groups -> packagegroup-petalinux-v4lutils(使用开发板进行视频的采集，就需要添加这个工具库) ->packagegroup-petalinux-v4lutils

\*/

保存退出。

执行单独的编译需要使用 -c 参数，单独编译rootfs。



编译生成SDK:

Petalinux-build --sdk

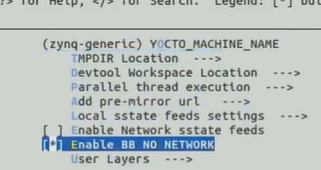
IMG_256

/\*

如果编译报错：使能了BB，则可以在config文件中取消使能：

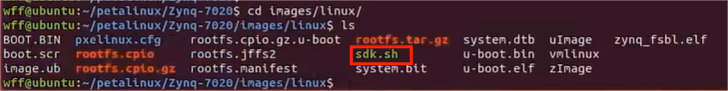
执行petalinux-config

Yocto Settings -> 取消使能Enable BB NO NETWORK

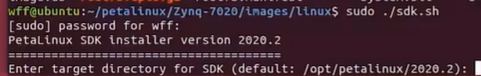


\*/

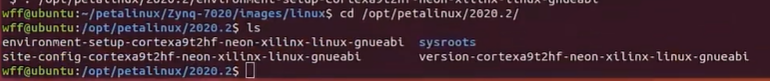
生成的产物在images/linux/ 目录下：



执行sudo ./sdk.sh 脚本。安装SDK。



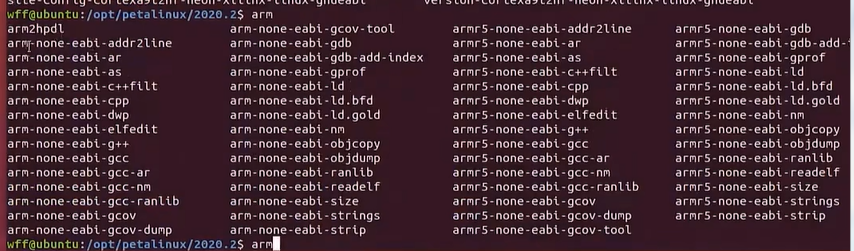
4.SDK中的交叉编译工具链和工具库：



Env文件：是用来配置交叉编译工具链的环境变量。

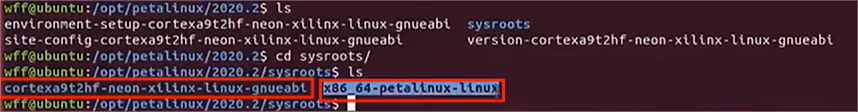
先使用source env+Tab

此时输入arm 可以看到许多交叉编译链：



进入sysroots:

里面有两个文件夹：



Cortex里面模拟了一个适用在arm环境下的根文件系统。

IMG_256

当source 环境变量之后 可以执行echo $CC 输出当前交叉编译器：

IMG_256

由于每次使用交叉编译工具链都需要先在/opt/petalinux/2020.2目录下执行 source + Tab更新环境变量。

并且另开终端环境变量又需要重新配置。

所以我们将操作放入文件之中：

vim ~/.bashrc

在最后一行加入如下：

IMG_256