MPSoc Petalinux

1. 创建一个硬件平台

使用Vivado建立基础的硬件平台，如使用SDsoc请按照SDsoc标准来建立硬件平台。

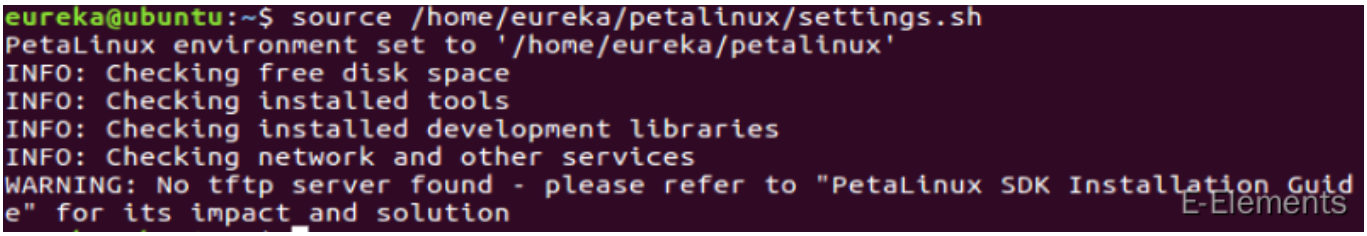
2. 导出硬件描述文件

在Vivado生成Bit文件后，执行 **Export Hardware** 后在工程目录下的 XXX.sdk 下会有类似于 system_wrapper.hdf 的硬件描述文件生成，将 xxx.sdk 拷贝到linux下之后会用到.

3. 启动PetaLinux工作环境

PetaLinux的启动非常简单，进入到petalinux的安装目录后，在命令窗口输入以下命令:即可启动

```
$ source ./settings.sh
```



4. 创建一个PetaLinux工程

使用 petalinux-create 命令来创建petalinux工程.

```
$ petalinux-create --type project --template <PLATFORM> --name <PROJECT_NAME>
```

其中的变量说明:

- --template -是和你使用的硬件平台有关
    - zynqMP (for UltraScale+ MPSoC)
    - zynq (for Zynq)
- microblaze (for MicroBlaze).
- --name <PROJECT_NAME> -这是你所创建PetaLinux工程的名字

```
$ petalinux-create --type project --template zynqMP --name MPSoc
```

```
eureka@ubuntu:~/Desktop$ petalinux-create --type project --template zynqMP --nam
e MPSoc
INFO: Create project: MPSoc
INFO: New project successfully created in /home/eureka/Desktop/MPSoc
```


使用这个命令来创建一个默认模板,之后通过命令来导入硬件信息.

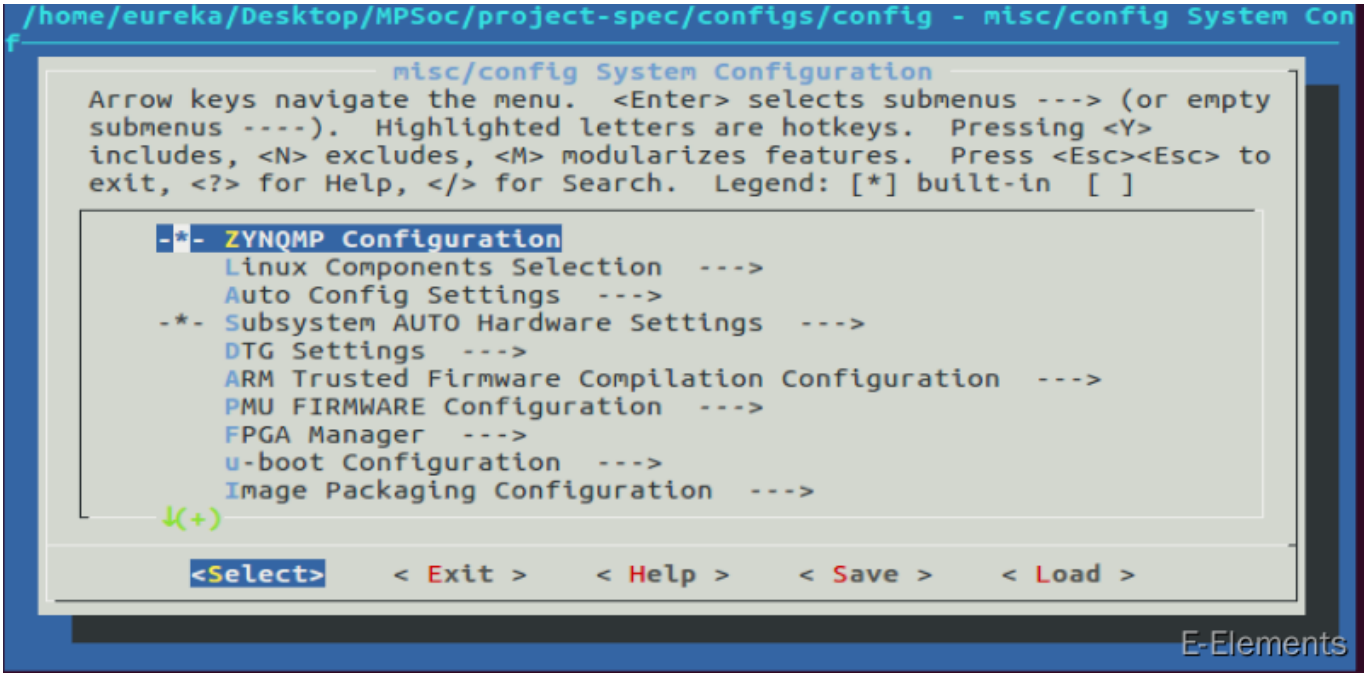
5. 获取硬件信息

使用 petalinux-config 命令来获取硬件信息

1. 将xxx.SDK拷贝到linux
2. 在命令行输入命令 **($ cd test)** 进入工程文件夹,此步非常关键,如果不进入会提示找不到指令.
3. 进入工程文件夹后,在窗口中输入以下命令,petalinux会自动读取硬件信息,定位到 HDF的文件路径就可以了.

```
$ petalinux-config --get-hw-description=<path-to-directory-containing-hardware
description-file>
```

等待一段时间后会进入如下图的配置界面(下面是Petalinux的系统配置菜单,例如"**Subsystem AUTO Hardware Settings**""**Auto Config Settings**"他们直接影响包含设备树、内核配置和uboot等配置步骤,这里可以直接悬着Exit退出并保存,退出后可以使用 **petalinux-config** 命令来进入这个界面)



- 设置第二分区启动,如果使用RAMdisk启动，就选择 **INITRD**

PetaLinux Configuration and Build System Image

Steps to configure PetaLinux for SD card ext filesystem boot and build the system image are as follows:

1. Change to root directory of your PetaLinux project.

   ```
   $ cd <plnx-proj-root>
   ```

2. Launch top level system configuration menu.

   ```
   $ petalinux-config
   ```

3. Select **Image Packaging Configuration ---> Root filesystem type**.
4. Select **SD card** as the RootFS type.
5. Exit menuconfig and save configuration settings.

- PetaLinux Auto Login(自动登录)

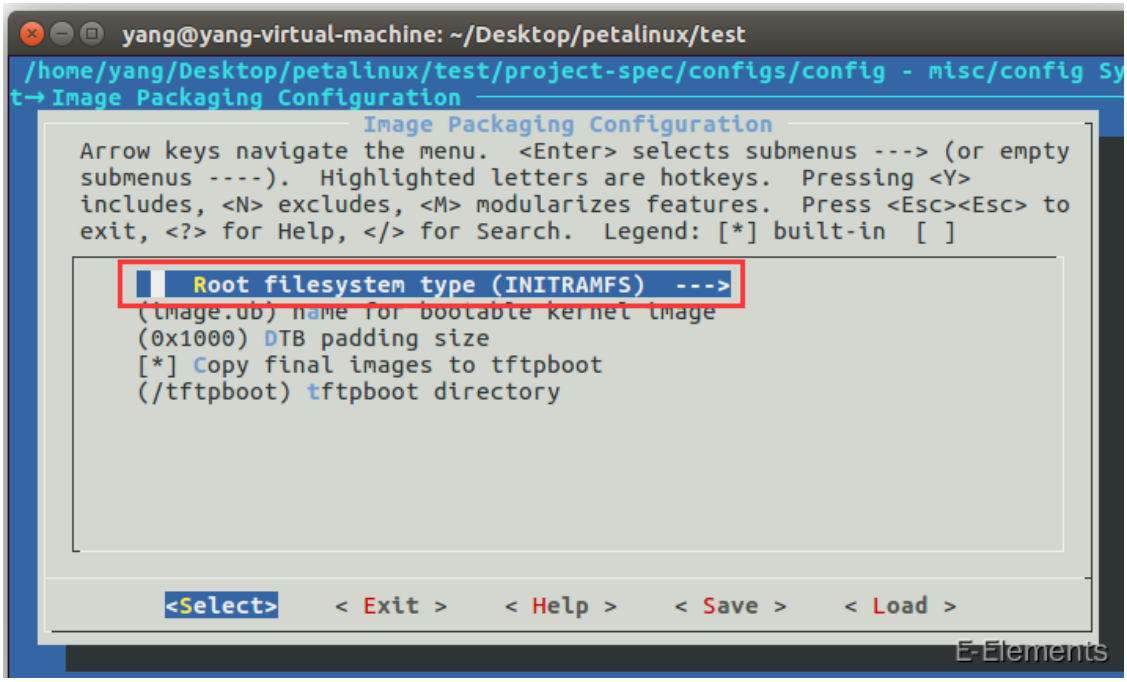
  1. Change the root directory of your PetaLinux project

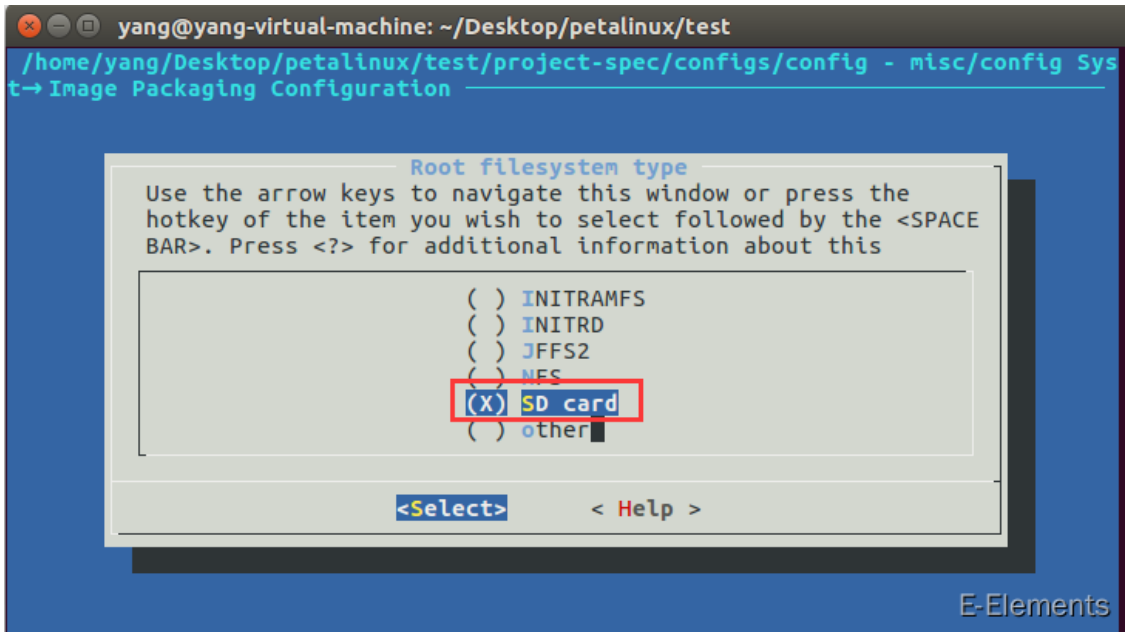
     ```
     cd <plnx-proj-root>
     ```

  2. Run `petalinux-config`
  3. Select **Yocto-settings** > **Enable debug-tweaks**
  4. Save the configuration and exit
  5. Run `petalinux-build`.

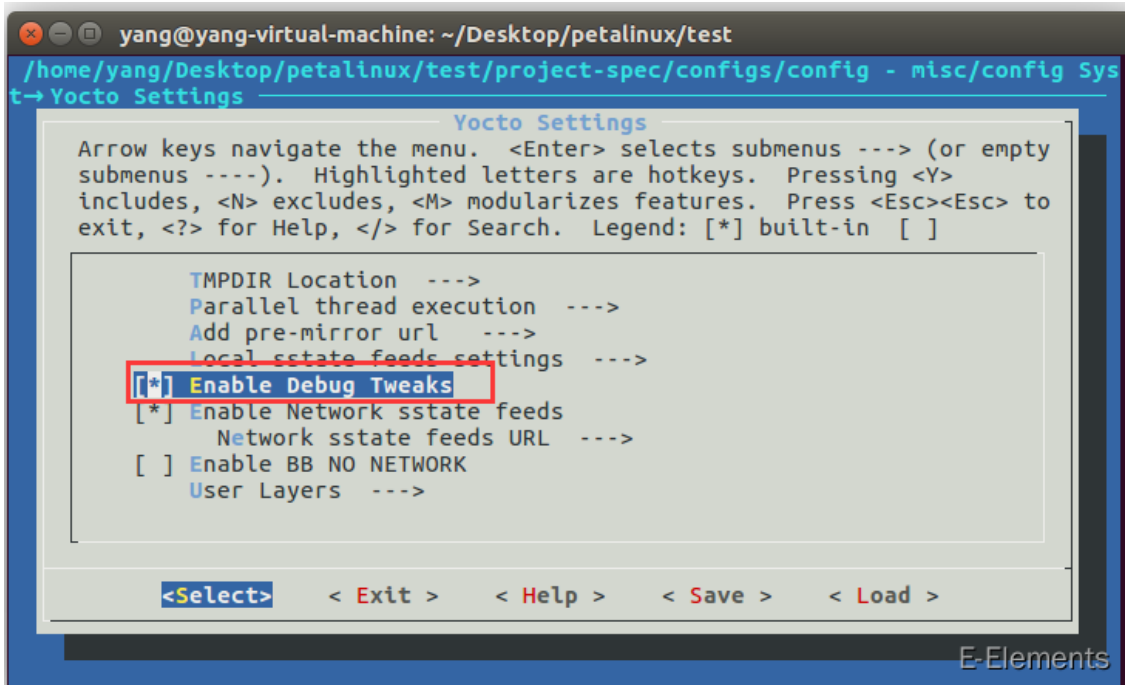
- 具体操作

  - 设置SD卡





- 设置自动登录



4. 配置linux内核
   在获取硬件信息完成后，输入以下指令后还可以继续修改内核配置，进入内核配置界面。

```
petalinux-config -c kernel
```

在 **SDSoc** 使用的PateLinux中需要配置如下：

Set CMA size to be larger, for SDS-alloc buffers:

- for Zynq UltraScale+ MPSoC: Device Drivers→ Generic Driver Options → Size in Mega Bytes(1024)
- for Zynq-7000 SoC: Device Drivers→ Generic Driver Options → Size in Mega Bytes(256)

E-Elements

Enable staging drivers:

- Device Drivers → Staging drivers (ON)

Enable APF management driver:

- Device Drivers → Staging drivers → Xilinx APF Accelerator driver (ON)

Enable APF DMA driver:

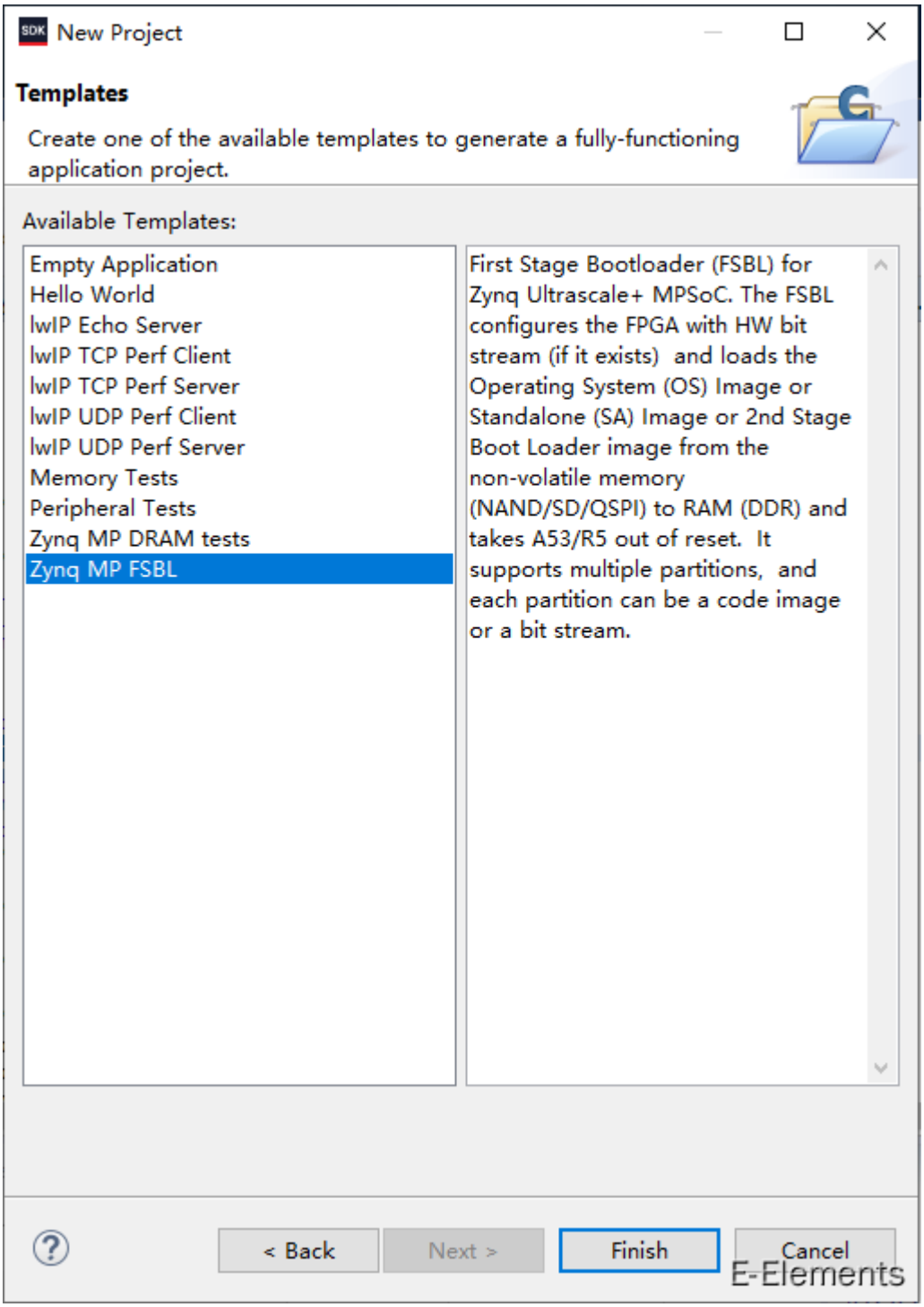
- Device Drivers → Staging drivers → Xilinx APF Accelerator driver → Xilinx APF DMA engines support (ON)

Note:

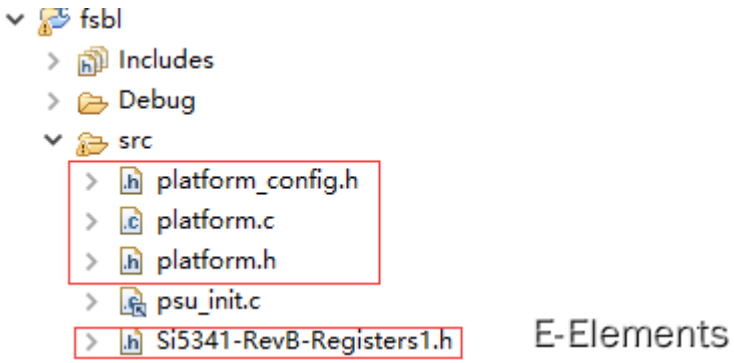
For Zynq UltraScale+ MPSoC, you must turn off CPU idle and frequency scaling. To do so, mark the following options:

- **CPU Power Management → CPU idle → CPU idle PM support (OFF)**
- **CPU Power Management → CPU Frequency scaling → CPU Frequency scaling (OFF)**

5. 修改设备树信息
   在工程目录下的 ~/project-spec/meta-user/recipes-bsp/files/system-user.dtsi 文件中添加设备树文件。
   具体参考文件在本目录中的 system-user.dtsi 文件中
   system-usr.dtsi
6. 在FSBL中添加 GTR 时钟的 IIC 配置
   在导出到SDK后打开SDK新建一个FSBL文件



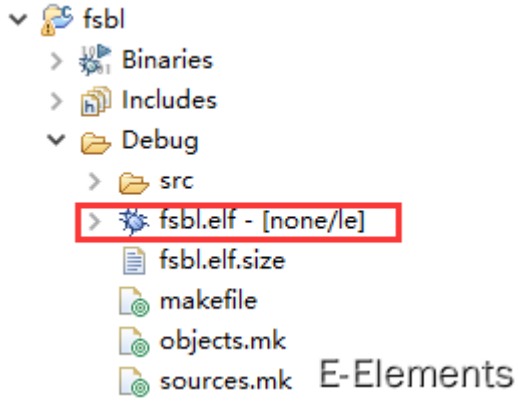
创建完成后在src文件夹中添加以下文件



添加完成后在 xfsbl_main.c 中添加代码，具体代码在本目录中的 xfsbl_main.c 中.

xfsbl_main.c

添加完成后进行编译，编译完成后在在SDK工程目录下的Debug文件夹中找到 **fsbl.elf** 文件将其拷贝出备用.



6. 编译Petalinux工程

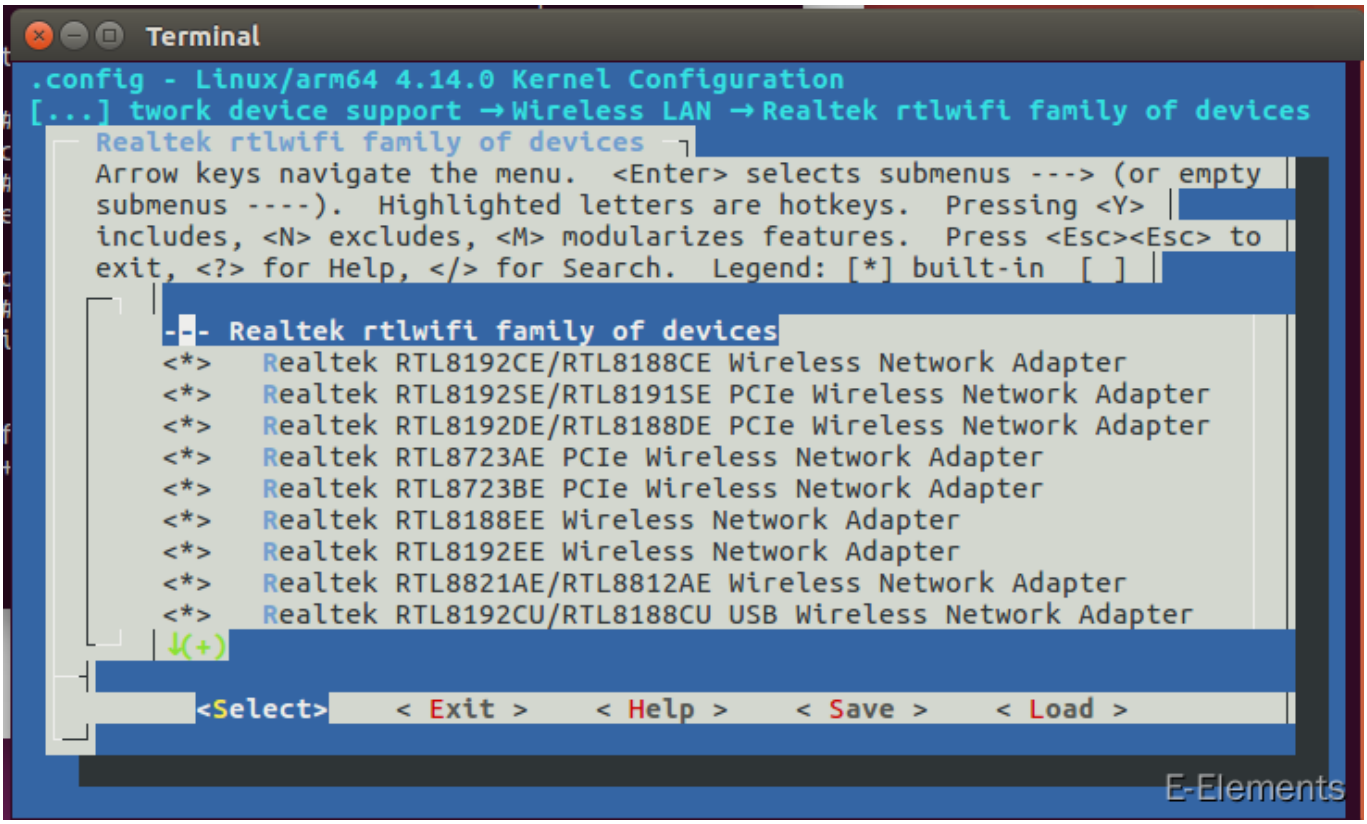
```
$ petalinux-build
```

编译完成后将上一步骤中的第六步中生成的 fsbl.elf 拷贝到 petalinux 工程目录下的 ~/images/linux 目录下。然后在工程的根目录下输入以下指令来生成 BOOT.BIN 文件

```
petalinux-package --boot --format BIN --fsbl images/linux/fsbl.elf --u-boot
images/linux/u-boot.elf --pmufw images/linux/pmufw.elf --fpga images/linux/*.bit -
-force</p>
```

7. 附录

usbWifi配置


Terminal
.config - Linux/arm64 4.14.0 Kernel Configuration
[...] twork device support → Wireless LAN → Realtek rtlwifi family of devices
Realtek rtlwifi family of devices
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
--- Realtek rtlwifi family of devices
<*> Realtek RTL8192CE/RTL8188CE Wireless Network Adapter
<*> Realtek RTL8192SE/RTL8191SE PCIe Wireless Network Adapter
<*> Realtek RTL8192DE/RTL8188DE PCIe Wireless Network Adapter
<*> Realtek RTL8723AE PCIe Wireless Network Adapter
<*> Realtek RTL8723BE PCIe Wireless Network Adapter
<*> Realtek RTL8188EE Wireless Network Adapter
<*> Realtek RTL8192EE Wireless Network Adapter
<*> Realtek RTL8821AE/RTL8812AE Wireless Network Adapter
<*> Realtek RTL8192CU/RTL8188CU USB Wireless Network Adapter
<Select> < Exit > < Help > < Save > < Load >
E-Elements