Build Petalinux 2021.2 with Vitis AI 2.0 and Smartcam

首先需要的環境有

1. Ubuntu 18.04
2. PetaLinux Tools - Installer - 2021.2
3. Kria K26 SOM Board Support Package - 2021.2

建置開始

- Step 1:

  下載完 PetaLinux Tools – Installer 後

  安裝 Dependencies

  <command> sudo apt-get install gcc g++ libncurses5-dev libncursesw5-dev libtool net-tools autoconf xterm texinfo gcc-multilib gawk zlib1g libz1:i386 zlib1g-dev build-essential

  <command> ./petalinux-v2021.1-final-installer.run -d <自訂安裝的路徑>

  - 不能執行請先 sudo chmod -R 777

  <command> source <自訂安裝的路徑>/settings.sh

  以上便安裝完 PetaLinux Tools 與設定好環境變數

(optional)有時候同個版本像是 petalinux 2021.1 有 update 1，有加入新的 Vitis ai layers，會影響到使用，因此會建議更新 petalinux tool

---> From Network:

<command> petalinux-upgrade -u http://petalinux.xilinx.com/sswreleases/rel-v2021/sdkupdate/2021.1_update1/ -p "aarch64" --wget-args "--wait 1 -nH --cut-dirs=4"

---> From Local:

<command> petalinux-upgrade -f <Local eSDK Directory Path> -p "aarch64"

- Step 2: 創建 petalinux project

  <command> petalinux-create -t project -s /<放 kv260 BSP 的路徑>/xilinx-k26-starterkit-v2021.1-final.bsp -n kv260_os(此為專案名稱與資料夾)

  <command> cd ./kv260_os

  <command> ls 後可看見下圖:

```
parallels@parallels-Parallels-Virtual-Platform:~/kria_kv260$ petalinux-create -t project -s ../kria_bsp/xilinx-k26-starterkit-v2
021.1-final.bsp -n kv260_os
INFO: Create project: kv260_os
INFO: New project successfully created in /home/parallels/kria_kv260/kv260_os
parallels@parallels-Parallels-Virtual-Platform:~/kria_kv260$ cd ./kv260_os/
parallels@parallels-Parallels-Virtual-Platform:~/kria_kv260/kv260_os$ ls
components  config.project  hardware  pre-built  project-spec  README  README.hw
parallels@parallels-Parallels-Virtual-Platform:~/kria_kv260/kv260_os$
```

要先將基本的 Petalinux build 起來，後面再加入 accelerated application 和 AI 等等之類的項目

<command> petalinux-build

- 看電腦性能，我筆電 build 了兩小時

◆ Step 3: 將 Vitis AI 2.0 跟一些 application 加入到 Project 中

1. Vitis AI 2.0

原始在 petalinux 2021.2 的 vitis ai 為 1.4，因此我們要先將原始的刪除

<command> cd components/yocto/layers/
<command> sudo rm -r meta-vitis-ai

然後再裝新的 meta-vitis-ai

<command> git clone -b rel-v2021.2 https://github.com/jlamperez/meta-vitis-ai.git meta-vitis-ai

再來要去將預設的 meta-vitis-ai config 路徑刪掉

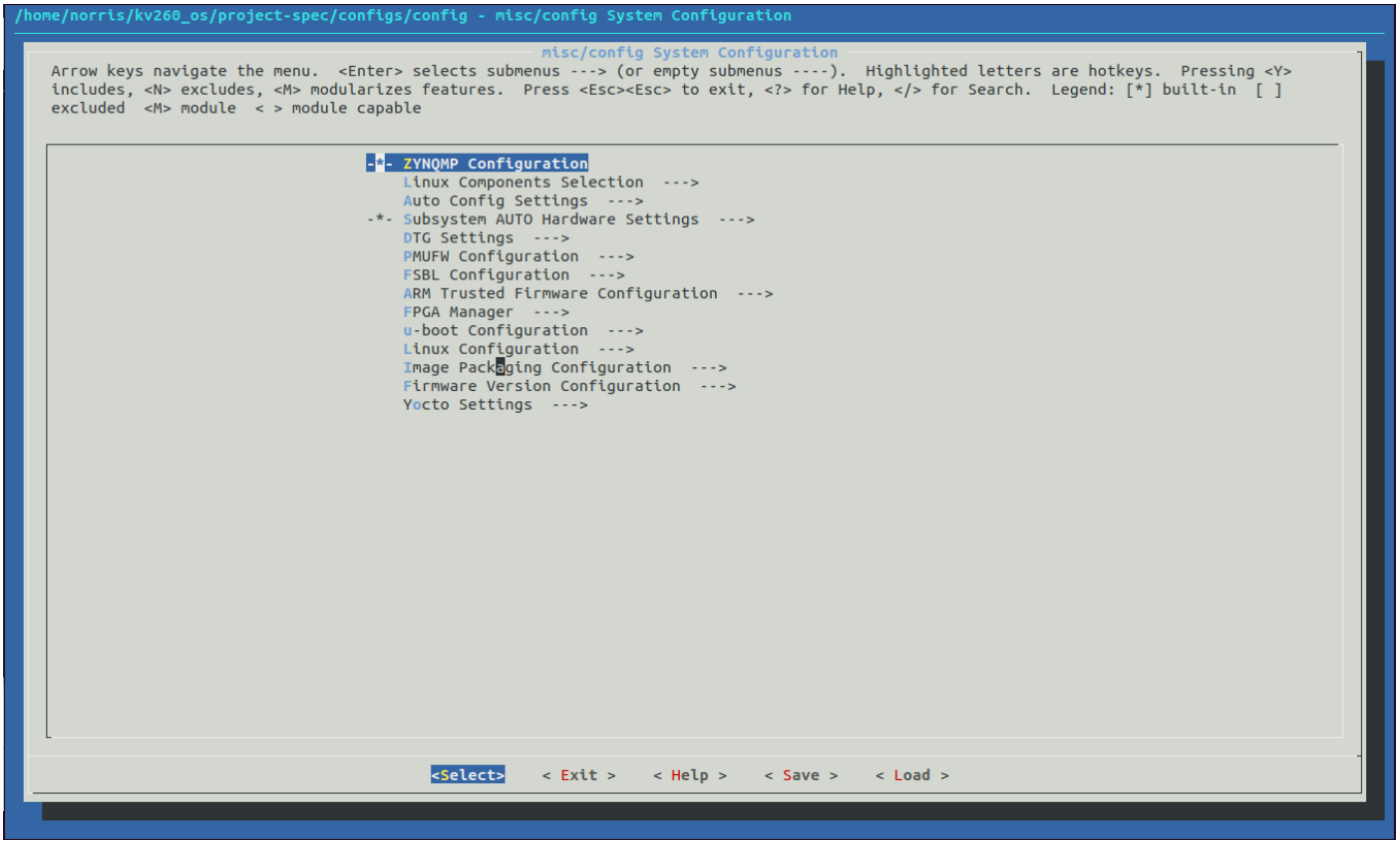
<command> cd ~/kv260_os
<command> vi build/conf/bblayers.conf

找到 ${SDKBASEMETAPATH}/layers/meta-vitis-ai 刪掉

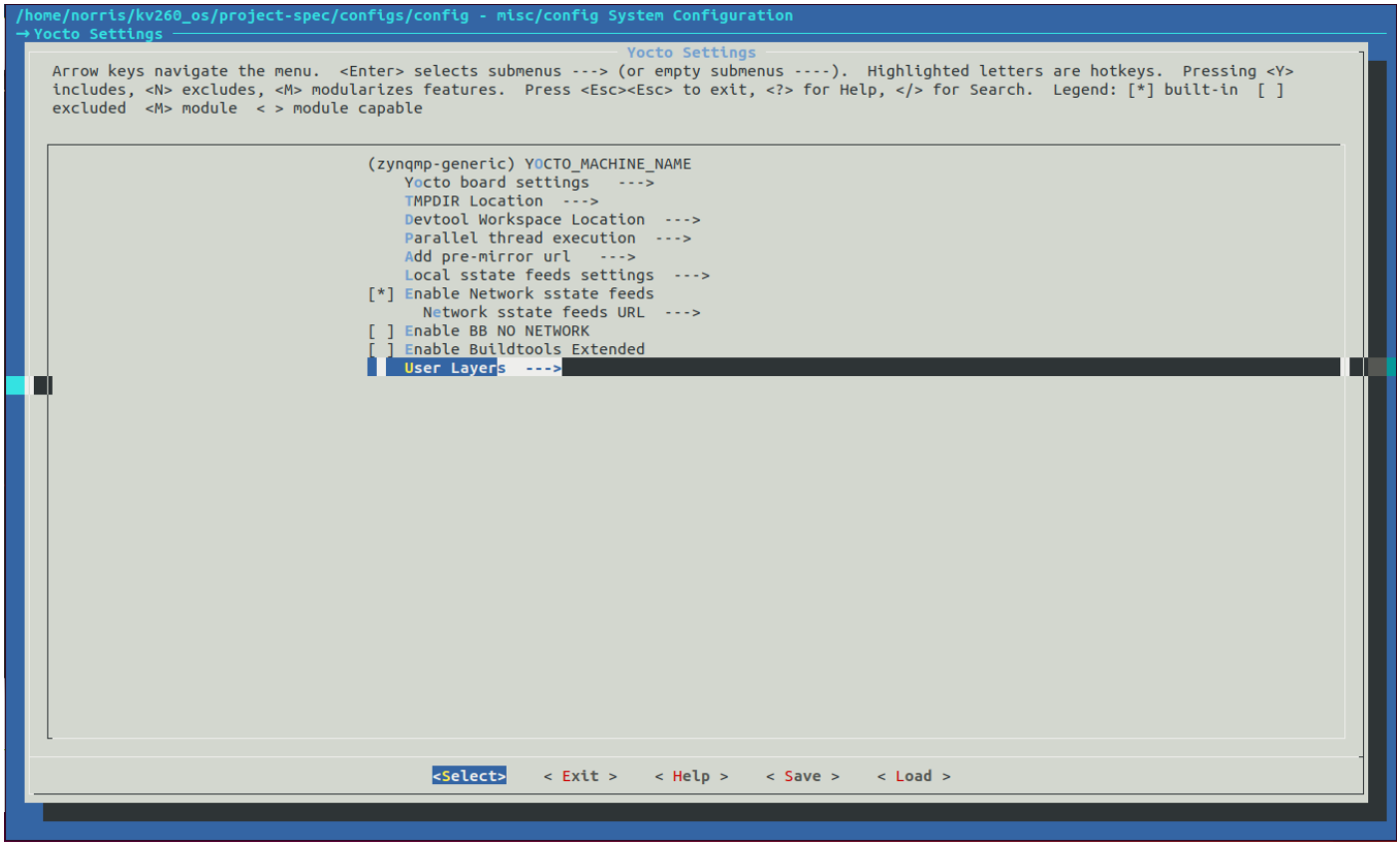
再來需要將新的 meta-vitis-ai 加入到 config 裡面

<command> petalinux-config

出現下圖

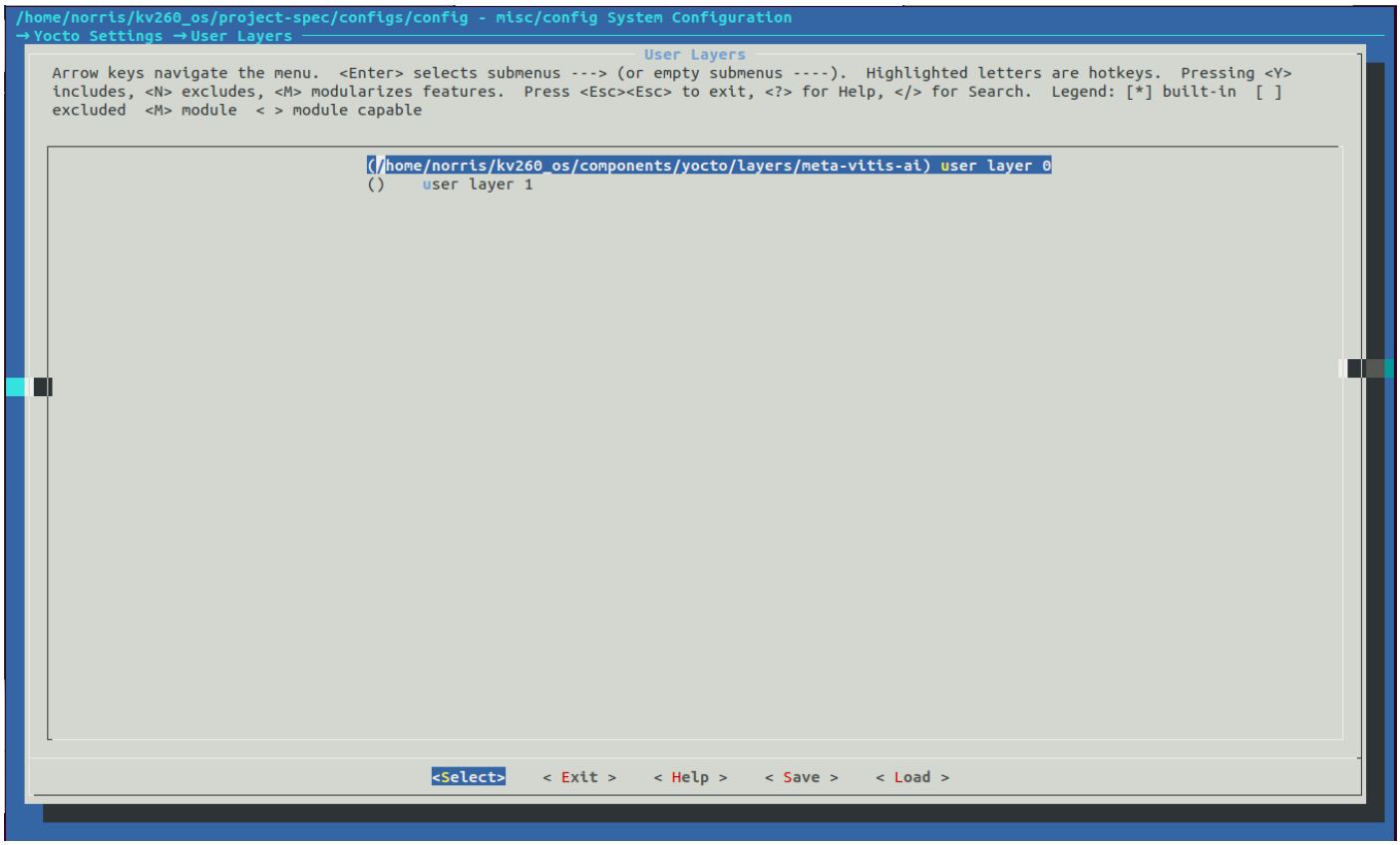


選擇 **Yocto Settings ---> User Layers**



打上如下圖中的這串文字，但前面需依照你自己的路徑

eg. /home/xxx/<project_name>components/yocto/layers/meta-vitis-ai



Save ---> Exit 直到退出整個 GUI

成功會看到下圖

```
norris@ubuntu:~/kv260_os/build/conf$ petalinux-config
[INFO] Sourcing buildtools
[INFO] Menuconfig project


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

[INFO] Sourcing build environment
[INFO] Generating kconfig for Rootfs
[INFO] Silentconfig rootfs
[INFO] Generating plnxtool conf
[INFO] Generating workspace directory
[INFO] Successfully configured project
```

再來，petalinux build 時預設是使用 petalinux-image-minimal

你可以在 petalinux-image-minimal.bbappend 或是 petalinux-image-

full.bbappend 中透過

<command> vi components/yocto/layers/meta-petalinux/recipes-core/images/petalinux-image-full.bb

添加下列文字

```
IMAGE_INSTALL_append = "packagegroup-petalinux-vitisai-dev \
                        packagegroup-petalinux-vitisai"
```

如此 Vitis ai 2.0 的部分就結束了

2. Applications

將 kv260 加入到 BOARD_VARIANT 中

```
<command> echo 'BOARD_VARIANT = "kv"' >>  project-spec/meta-user/conf/petalinuxbsp.conf
```

接下來再將每個 kv260 四個 demo 加入到 user-rootfsconfig 中

- 這步應該是確保在用 xmutil getpkgs 時找得到相對應的 app 包

```
<command> echo 'CONFIG_packagegroup-kv260-smartcam' >> project-spec/meta-user/conf/user-rootfsconfig
<command> echo 'CONFIG_packagegroup-kv260-aibox-reid' >> project-spec/meta-user/conf/user-rootfsconfig
<command> echo 'CONFIG_packagegroup-kv260-defect-detect' >> project-spec/meta-user/conf/user-rootfsconfig
<command> echo 'CONFIG_packagegroup-kv260-nlp-smartvision' >> project-spec/meta-user/conf/user-rootfsconfig
```

3. Add FPGA Firmware

直接加入現有的 Firmware 即可

```
<command> cd ~/kv260_os
<command> git clone -b xlnx_rel_v2021.2 https://github.com/Xilinx/kv260-firmware
```

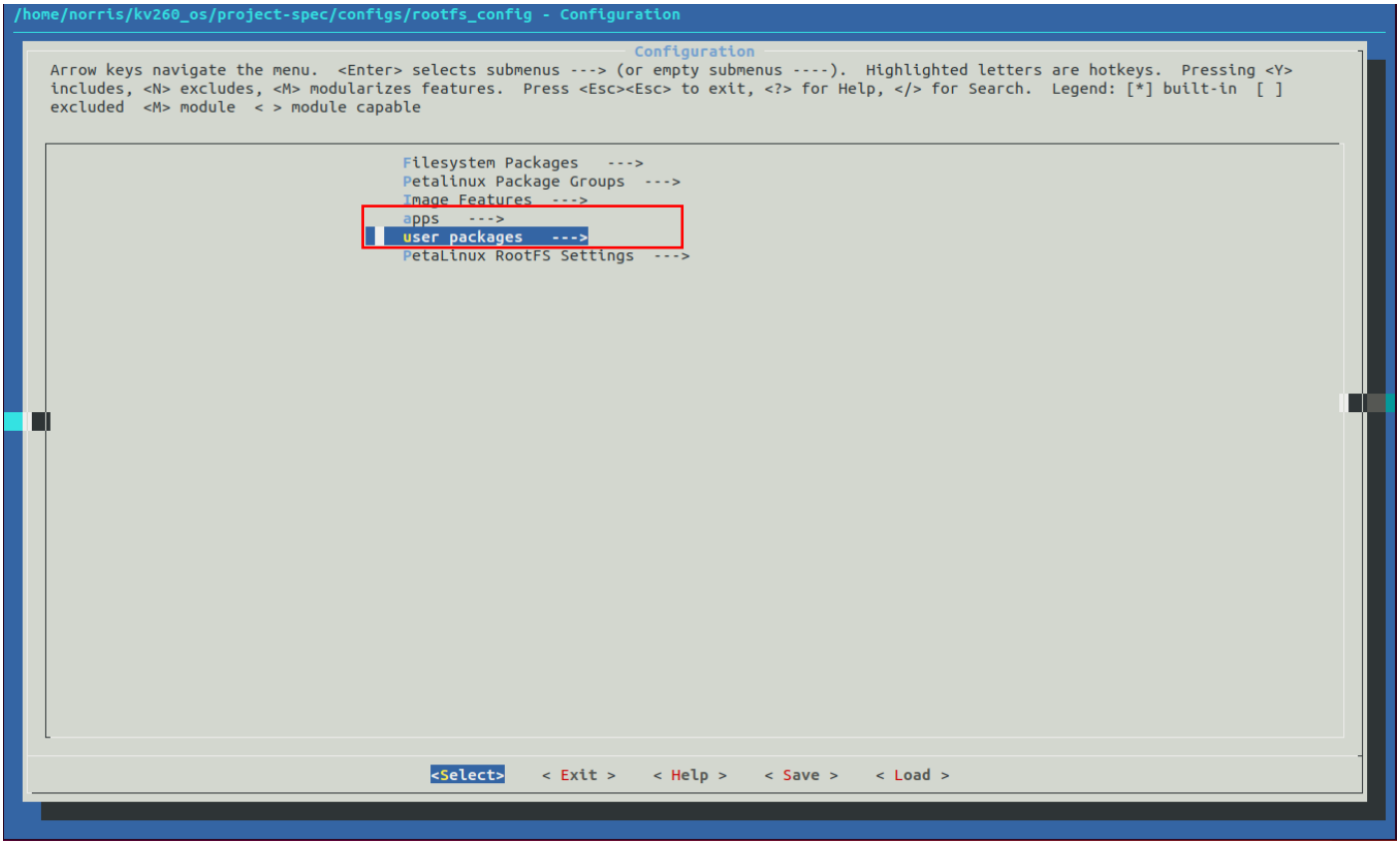
再來各自加入 firmware

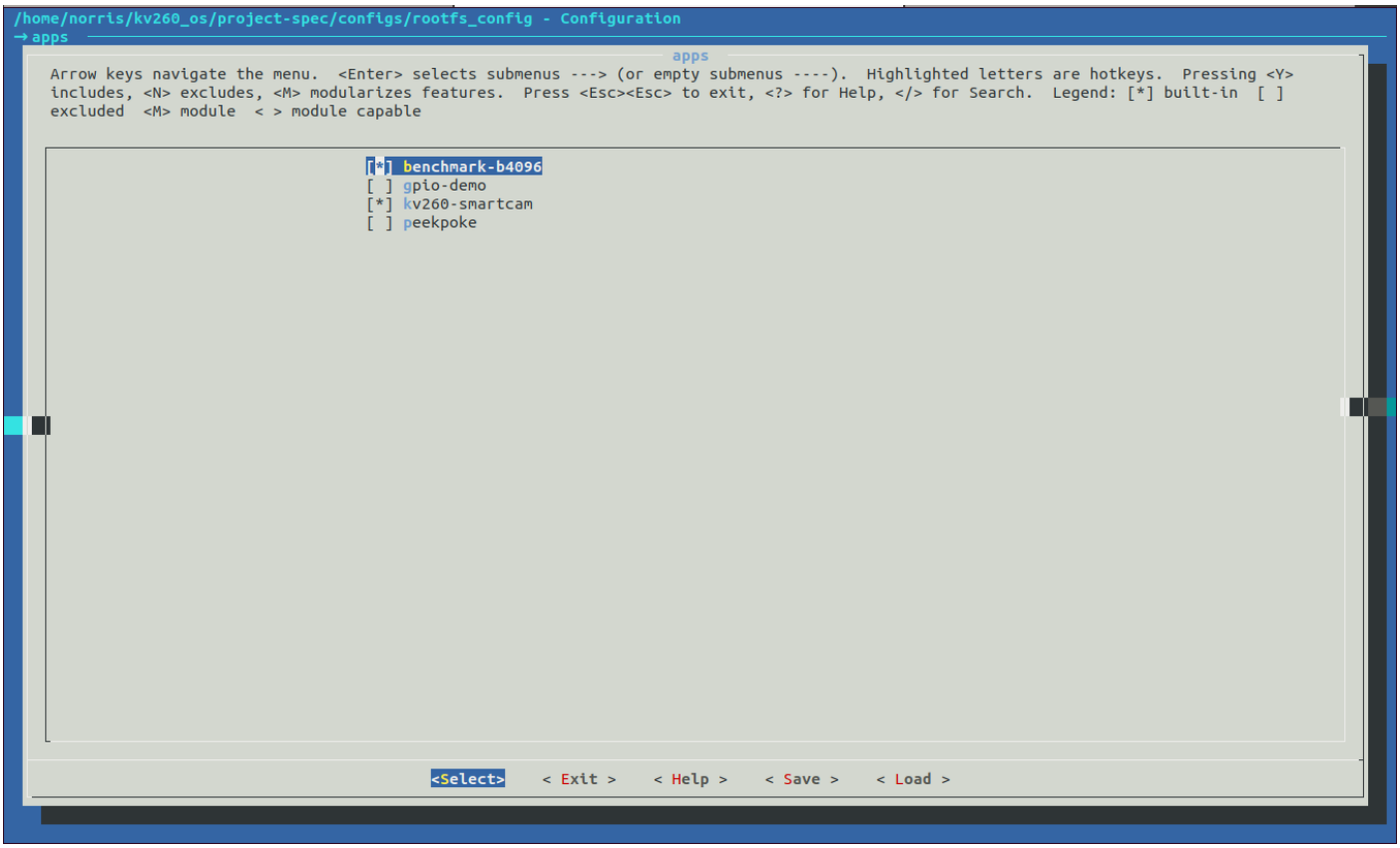
```
# benchmark-b4096.bb
<command> petalinux-create -t apps --template fpgamanager -n benchmark-b4096 --enable --srcuri "./kv260-firmware/benchmark-b4096/kv260-benchmark-b4096.bit ./kv260-firmware/benchmark-b4096/kv260-benchmark-b4096.xclbin ./kv260-firmware/benchmark-b4096/shell.json ./kv260-firmware/benchmark-b4096/kv260-benchmark-b4096.dtsi"

# smartcam.bb
<command> petalinux-create -t apps --template fpgamanager -n karp-smartcam --enable --srcuri "./kv260-firmware/smartcam/kv260-
```

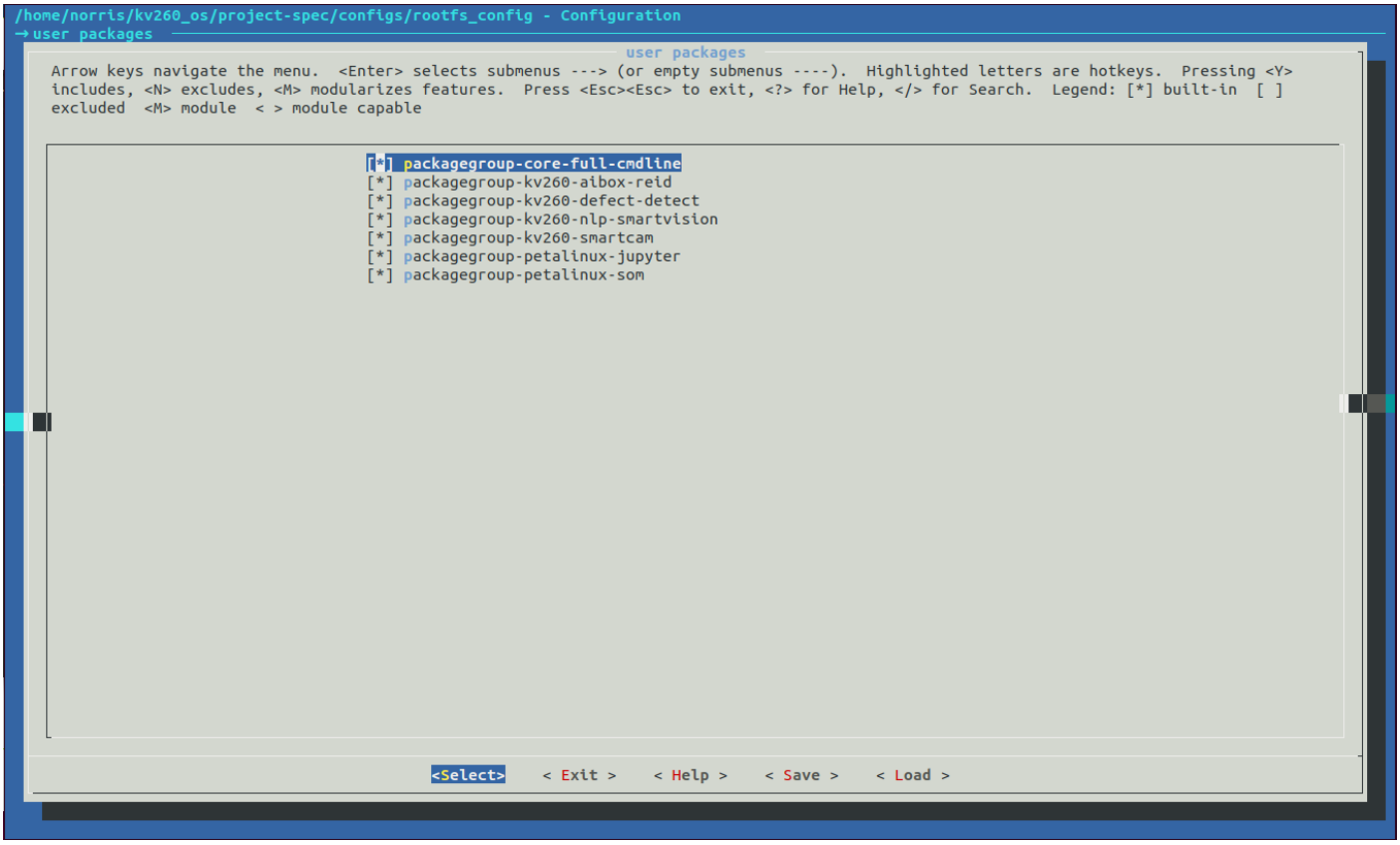
```
smartcam.bit ./kv260-firmware/smartcam/kv260-
smartcam.xclbin ./kv260-firmware/smartcam/shell.json ./kv260-
firmware/smartcam/kv260-smartcam.dtsi"
```

最後啟用 `root config` 來勾選 `image` 要使用的應用

```
<command> petalinux-config -c rootfs
```




/home/norris/kv260_os/project-spec/configs/rootfs_config - Configuration
→ apps
apps
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 [ ] excluded <M> module < > module capable
[*] benchmark-b4096
[ ] gpio-demo
[*] kv260-smartcam
[ ] peekpoke
<Select> < Exit > < Help > < Save > < Load >


/home/norris/kv260_os/project-spec/configs/rootfs_config - Configuration
→ user packages
user packages
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 [ ] excluded <M> module < > module capable
[*] packagegroup-core-full-cmdline
[*] packagegroup-kv260-aibox-reid
[*] packagegroup-kv260-defect-detect
[*] packagegroup-kv260-nlp-smartvision
[*] packagegroup-kv260-smartcam
[*] packagegroup-petalinux-jupyter
[*] packagegroup-petalinux-som
<Select> < Exit > < Help > < Save > < Load >

一樣儲存後退出，會看到

```
norris@ubuntu:~/kv260_os$ petalinux-config -c rootfs
[INFO] Sourcing buildtools
[INFO] Silentconfig project
[INFO] Generating kconfig for Rootfs
[INFO] Menuconfig rootfs


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

[INFO] Generating plnxtool conf
[INFO] Successfully configured rootfs
```

- **Step 4: Build Petalinux Image**

  <command> petalinux-build -c petalinux-image-full

- **Step 5: Create SD Card Image**