PETALINUX 2018.2 20/05/19

Project Creation and Configuration

Step by Step:

By Juliano Pimentel (github julianop99)

1. **Project Create**

petalinux-create -t project -s <home>/Downloads/uz3eg_iocc_sd_oob_2018_2.bsp

2. Move into just created directory

cd uz3eg_iocc_sd_oob_2018_2/

3. **Initially configure the PetaLinux project with the corresponding hardware file.**

petalinux-config --get-hw-description=<path to folder containing system.hdf file>

```
      # Import the hardware description into the PetaLinux project.
```

```
petalinux-config --oldconfig --get-hw-description=./hw_platform/ -p
${START_FOLDER}/${PETALINUX_PROJECTS_FOLDER}/${PETALINUX_PROJECT_NAME}
```

4. **System Configuration (memory, serial setting, etc…)** – can be skipped if nothing has been changed before…

petalinux-config

5. **Kernel Configuration (**Power Management API's, UART, and RTC, etc…)

petalinux-config -c kernel

6. **Create custom applications**

```
########################################
# Use: petalinux-create --type apps --name <app_name>--enable
petalinux-create --type apps --name hts221-sensor --enable
```

App generated at
<project_dir>/project-spec/meta-user/recipes-apps/hts221-sensor

```
# Copy the hts221-sensor folder over to the hts221-sensor
# application folder
```

Copy .c and edit (if required) the Makefile and the .bb at:
<project_dir>/project-spec/meta-user/recipes-apps/hts221-sensor/files

Before building the module, you will need to enable the module from PetaLinux menuconfig by running:

petalinux-config -c rootfs

To compile and install your APPs to the target file system copy on the host, simply run the

petalinux-build -c hts221 -x cleanall (or do_clean) → to cleanup, and then run
petalinux-build -c hts221" → to build the module command.

##

7. Create custom kernel modules (device drivers)

```
# Create a PetaLinux kernel module named hts221.
 petalinux-create --type modules --name hts221 --enable
```

App generated at
<project_dir>/ project-spec/meta-user/recipes-modules/hts221

```
# Copy the hts221 folder over to the hts221 module folder.
```

Copy .c and edit (if required) the Makefile at:
<project_dir>/project-spec/meta-user/recipes-modules/hts221/files

You will see your module in the "Kernel Modules --->" submenu.

To compile and install your module to the target file system copy on the host, simply run the
petalinux-build -c kernel" → to build kernel first, and then run
petalinux-build -c hts221" → to build the module command.

If need to remove apps and modules:

```
# To remove apps and modules, Edit the files below:
```

project-spec/meta-plnx-generated/recipes-core/images/petalinux-user-image.bb

project-spec/meta-user/recipes-core/images/petalinux-image.bbappend

8. Rootfs Configuration (To enable the just created apps)

petalinux-config -c rootfs

9. Change the DTS (device tree source) file

File to be changed is at {PROJECT}/

project-spec/meta-user/recipes-bsp/device-tree/files/**system-user.dtsi**

10. Build Kernel and DTB (device tree blob)

petalinux-build

11. **Generate BOOT.BIN** (and potentially image.ub). Bynary package will include
    a. Zynq_fsbl.elf
    b. Pmufw.efl
    c. Bl31.efl
    d. U-boot.efl
    e. Bit file

cd images/linux

petalinux-package --boot --fsbl zynqmp_fsbl.elf --fpga uz_petalinux_wrapper.bit --u-boot --force

12. **Copy BOOT.BIN and Image.ub to SD card**!

cp BOOT.BIN image.ub system.dtb /media/sf_Share/.

Important Documents and Notes

Document ug1144

Table 1-2: **Layers from Xilinx**

Layer	Recipes
meta-xilinx	Contains recipes of linux kernel, uboot and Arm Trusted Firmware (ATF)
meta-xilinx-tools	Contains recipes of all embeddedsw apps: fsbl, pmufw, fsboot, device-tree
meta-petalinux	Contains distro recipes and package groups petalinux-image-minimal --> minimal feature set petalinux-image-full ---> Full feature set
meta-openamp	Contains openamp recipes and configurations
meta-linaro-toolchain	Contains tool chain recipes for Zynq and ZynqMP

For example, for Zynq UltraScale+ MPSoC:
`$PETALINUX/components/yocto/source/aarch64/layers`

https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841618/PetaLinux+Getting+Started

ug1144

DeviceTrees

Configuration (User Level)
<plnx-projroot>/project-spec/meta-user/recipes-bsp/device-tree/files

- File to be modified: system-user.dtsi

Generated Files:
<plnx-projroot>/components/plnx_workspace/device-tree/

- Files generated:
  system-top.dts (automatically generated, includes system-user.dtsi)
  system-conf.dtsi (file is automatically generated by PetaLinux SDK)
  zynqmp.dtsi (dts file for Xilinx ZynqMP – contains all the hdwr mapping for the MPSoC)
  pcw.dtsi (file automatically generated by Xilinx, brings all interfaces enabled “okay”)

pl.dtsi (automatically generated, brings all PL interfaces enabled “okay”)

- Para compilar apenas o device-tree

petalinux-build -c device-tree

U-BOOT

PetaLinux u-boot configuration is associated with **config.mk and platform-auto.h**

configuration files
<plnxproj_root>/project-spec/ meta-plnx-generated/recipes-bsp/u-boot/configs

and **platform-top.h** located at
<plnxproj_root>/project-spec/meta-user/recipes-bsp/u-boot/files/.

KERNEL

PetaLinux project config - System Configuration

<plnxproj_root>/project-spec/configs/config

PetaLinux Kernel config - System Configuration

.config - Linux/arm64 4.14.0 Kernel Configuration

<plnxproj_root>/project-spec/configs/config

- Files generated:
  config (configuration file used to store user settings)
  roots_config (configuration file used for root filesystem)

Generated Files:
< <plnx-proj-root>/build/tmp/work/plnx_zynqmp-xilinx-linux/linux-xlnx/4.14-xilinx-v2018.2
+gitAUTOINC+ad4cd988ba-r0/linux-plnx_zynqmp-standard-build

Rootfs Config File

<plnxproj_root>/project-spec/configs/rootfs_config

Decompile DTB into DTS

```
$ dtc -I dtb -O dts -o system.dts system.dtb
```

```
dtc -O dtb -o system.dtb system.dts
```