DPU-Petalinux指南

什么是Peatlinux?

Peatlinux能产生ZYNQ7000、ZYNQ7MP和MicroBlaze嵌入式芯片Linux操作系统的工具，包括编译U-Boot，Kernel，Root File System，生成嵌入式芯片的fsbl.elf，uImage，zImage等文件。

创建系统的基本流程

1. Vivado创建硬件平台
2. 建PetaLinux工程

   `petalinux-create -t project -n your_project_name`
3. 据硬件平台描述文件（.XSA/.BSP）创建工程

   `petalinux-config --get-hw-description /path/to/XSA/or/BSP`
4. 配置系统选项

   `petalinux-config`
5. 建用户组件

   `petalinux-create -t COMPONENT`
6. 配置Linux内核

   `petalinux-config -c kernel`
7. 配置rootfs

   `peatlinux-config -c rootfs`
8. Build系统

   `petalinux-build`
9. 为部署系统打包

   `petalinux-package`
10. 启动系统和测试

    `petalinux-boot`

Create a Project

1、通过BSP来创建工程

Board Support Package, BSP是对硬件电路的描述文件，通过它能够快速生成相应的Linux系统。一般Xilinx的开发板都有对应的BSP文件，通过这些文件可以快速上手Petalinux，但是如果用户自己设计的开发板，官方BSP就失效了。

```
petalinux-create -t project -s /path/to/bsp
```

2、通过Vivado输出的硬件描述文件创建工程

2.1 准备阶段

你需要熟练使用Vivado工具，完成Block Design等设计，生成Bitstream，导出**硬件描述文件（.XSA）**。在使用Vivado工具进行设计时，需要对PS模块做如下配置

Zynq-7000 Devices

1.添加一个TTC模块，当有多个TTC模块时，petalinux默认使用第一个。

2.至少32MB的内存（DDR3 RAM > 32MB）

3.PS模块的串口

4.外部存储器，例如QSPI Flash和SD/MMC

5.PS模块的以太网（一般选上）

Zynq UltraScale+ MPSoC

1.至少64MB的内存（DDR3/4 RAM > 64MB）

2.PS模块的串口

3.外部存储器，例如QSPI Flash和SD/MMC

4.PS模块的以太网（一般选上）

2.2 根据模板创建工程

```
petalinux-create --type project --template <PLATFORM> --name <your_prj_name>
```

<PLATFORM>

-----zynqMP (Zynq UltraScale+MPSoC)

-----zynq (Zynq-7000 devices)

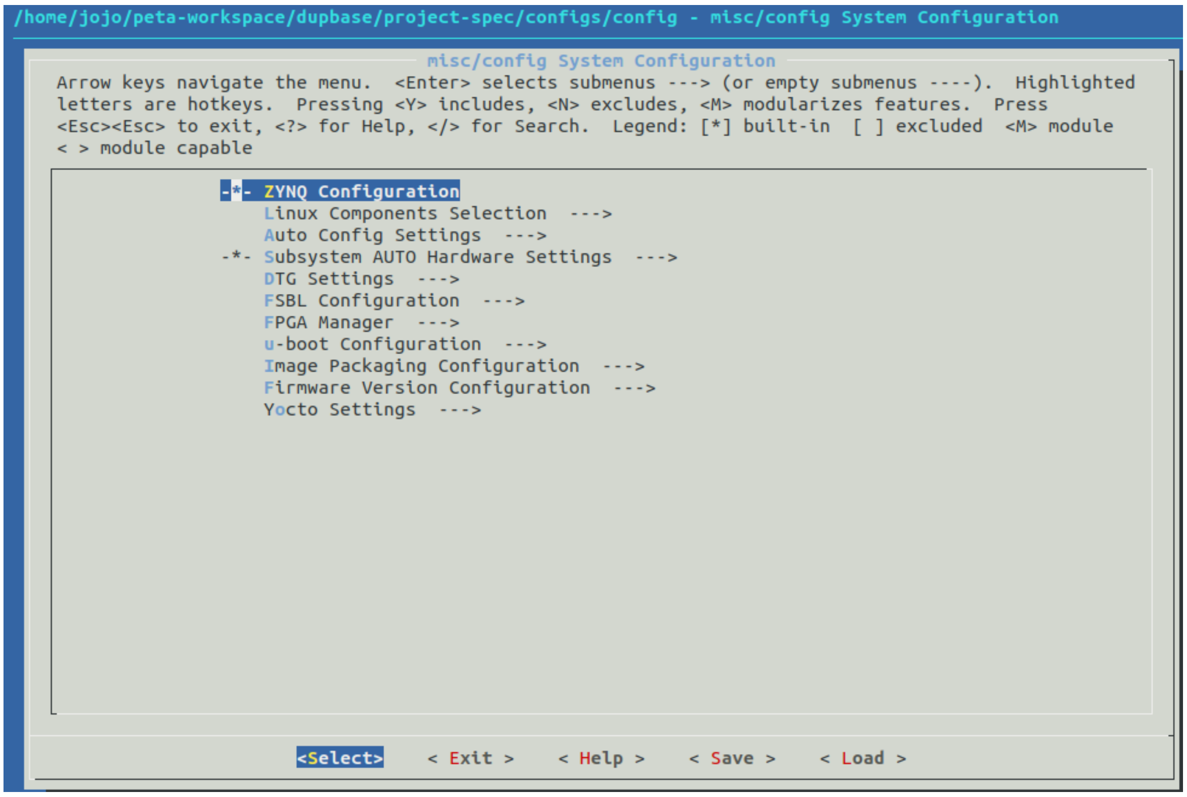
-----microblaze (MicroBlaze processor)

<your_prj_name>

petalinux会在你的workspace下创建一个<your_prj_name>文件夹，所有的工程文件都在里面。

2.3 导入硬件信息初始化工程

```
petalinux-config --get-hw-description /path/to/XSA
```



```
/home/jojo/peta-workspace/dupbase/project-spec/configs/config - misc/config System Configuration
                         misc/config System Configuration
  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

                -*- ZYNQ Configuration
                    Linux Components Selection  --->
                    Auto Config Settings  --->
                -*- Subsystem AUTO Hardware Settings  --->
                    DTG Settings  --->
                    FSBL Configuration  --->
                    FPGA Manager  --->
                    u-boot Configuration  --->
                    Image Packaging Configuration  --->
                    Firmware Version Configuration  --->
                    Yocto Settings  --->

              <Select>    < Exit >    < Help >    < Save >    < Load >
```

需要注意的是，确保***Subsystem AUTO Hardware Setting***被选上。该选项中包括了处理器、内存、串口、以太网、Flash、SD、RTC硬件信息，以及BOOT.BIN，U-Boot，内核，rootfs和dtb软件信息。**根据你的需要，通过这些子选项可以对系统进行修改，一定要记得save，再退出。**

补充：选择***DTG Settings*** 出现***(template)MACHINE_NAME***

如果你使用的是Xilinx的开发板，template参数可以是

ac701-full, ac701-lite, kc705-full, kcu105, zcu1275-revb,
zcu1285-reva, zc1751-dc1, zc1751-dc2, zc702, zc706, avnet-ultra96-rev1, zcu100-revc,
zcu102-rev1.0, zcu104-revc, zcu106-reva, zcu111-reva

2.4 进一步配置工程

2.4.1 修改Rootfs的类型

`petalinux-config`

选择 ***Image Packaging Configuration*** 修改 ***Root filesystem type***为***EXT4(SD/eMMC/SATA/USB)***

Root filesystem type可以是

---INITRAMFS---INITRD---JFFS2---NFS---EXT4(SD/eMMC/SATA/USB)

2.4.2 修改启动镜像的存储方式

选择***Subsystem AUTO Hardware Settings***   ---> ***Advanced bootable images storages settings***

boot image settings 该选项配置BOOT.BIN文件的存储方式，可选为**primary flash**或者**primary sd**

u-boot env parition settings 该选项配置u-boot文件的存储方式，可选为**primary flash**或者**primary sd**

kernel image settings 该选项配置linux内核的存储方式，可选为**primary flash**、**primary sd**和**ethernet**

jffs2 rootfs image settings和***dtb image settings***默认就好

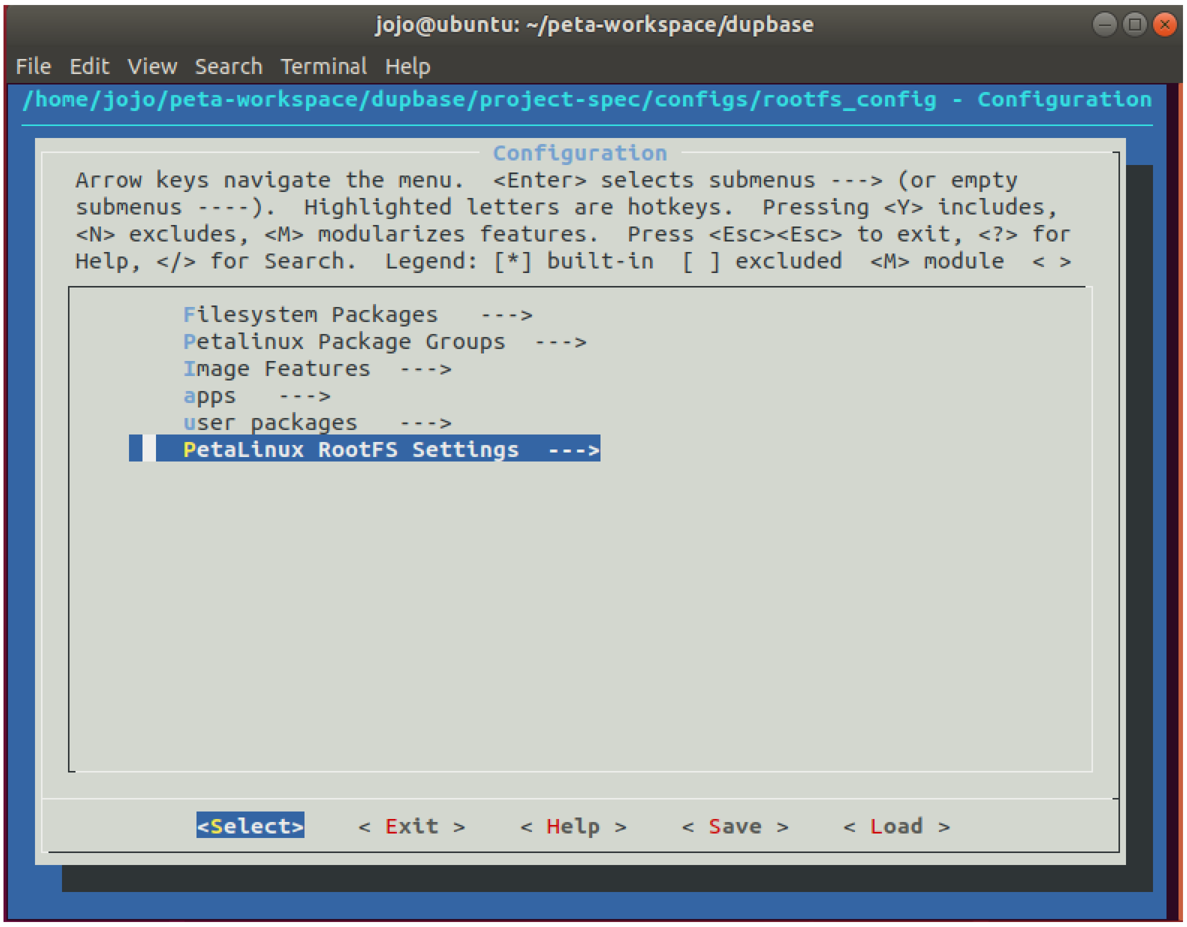
2.4.3 为Rootfs添加依赖库和包

复制extra opencv文件夹到***/path/to/your/prj/dir/project-spec/meta-user/recipes-ai/***，如果没有***recipes-ai***文件夹可以新创建一个。

找到*/path/to/your/prj/dir/project-spec/meta-user/conf/****user-rootfsconfig***文件，添加下列内容

```
# Xilinx Run Time, XRT support
CONFIG_xrt
CONFIG_xrt-dev
CONFIG_zocl
CONFIG_opencl-clhpp-dev
CONFIG_opencl-headers-dev
CONFIG_packagegroup-petalinux-opencv
CONFIG_packagegroup-petalinux-opencv-dev

# DPU support
CONFIG_glog
CONFIG_gtest
CONFIG_json-c
CONFIG_protobuf
CONFIG_python3-pip
CONFIG_apt
CONFIG_dpkg

CONFIG_packagegroup-petalinux-x11
CONFIG_packagegroup-petalinux-v4lutils
CONFIG_packagegroup-petalinux-matchbox

# Vitis AI packages
CONFIG_gtest-staticdev
CONFIG_json-c-dev
CONFIG_protobuf-dev
CONFIG_protobuf-c
CONFIG_libeigen-dev

# Native compiling support
CONFIG_packagegroup-petalinux-self-hosted
CONFIG_cmake

# Extra opencv support
CONFIG_opencv
```

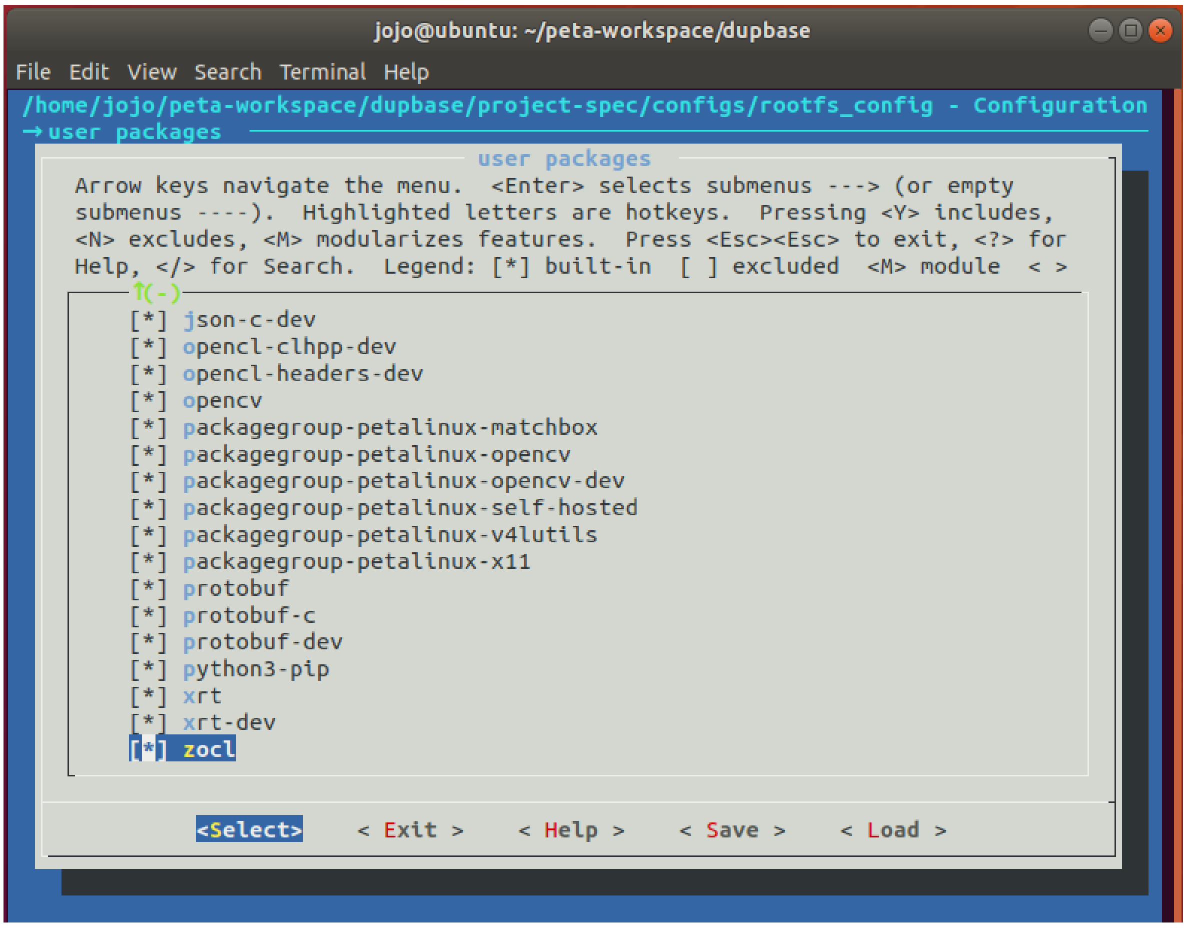
`petalinux-config -c rootfs`



```
jojo@ubuntu: ~/peta-workspace/dupbase
File  Edit  View  Search  Terminal  Help
/home/jojo/peta-workspace/dupbase/project-spec/configs/rootfs_config - Configuration
                          Configuration
  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  < >

            Filesystem Packages  --->
            Petalinux Package Groups  --->
            Image Features  --->
            apps  --->
            user packages  --->
            PetaLinux RootFS Settings  --->

          <Select>    < Exit >    < Help >    < Save >    < Load >
```

*Tips：在**PetaLinux Rootfs Settings**中可以修改登陆的密码**默认为root**，其他选项顾明思意，不清楚的选择进入后搜索每个packages的功能即可。*

选择***user packages***勾选出现的文件包，在***Image Features***目录中开启***package-management***和***debug_tweaks***。



```
jojo@ubuntu: ~/peta-workspace/dupbase
File  Edit  View  Search  Terminal  Help
/home/jojo/peta-workspace/dupbase/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  < >
      ↑(-)
        [*] json-c-dev
        [*] opencl-clhpp-dev
        [*] opencl-headers-dev
        [*] opencv
        [*] packagegroup-petalinux-matchbox
        [*] packagegroup-petalinux-opencv
        [*] packagegroup-petalinux-opencv-dev
        [*] packagegroup-petalinux-self-hosted
        [*] packagegroup-petalinux-v4lutils
        [*] packagegroup-petalinux-x11
        [*] protobuf
        [*] protobuf-c
        [*] protobuf-dev
        [*] python3-pip
        [*] xrt
        [*] xrt-dev
        [*] zocl

          <Select>    < Exit >    < Help >    < Save >    < Load >
```

（可选）2.4.4 shell的模式从dropbear替换为openssh

a)选择 ***Image Features***

b)关闭***ssh-server-dropbear***，开启***ssh-server-openssh***

c)前往目录***Filesystem Packages->misc->packagegroup-core-ssh-dropbear***，关闭***packagesgroup-core-ssh-dropbear***

d)前往目录***Filesystem Packages->console->network->openssh***，开启***openssh，openssh-sftp-server，openssh-sshd，openssh-scp***

2.4.5 添加DPU的设备树

修改**system-ueser.dtsi**，文件的路径为***path-to-your-prj/project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi***。可参考的设计是#AR73058和ZED BSP

```
/include/ "system-conf.dtsi"
/ {
    amba_pl: amba_pl {
        #address-cells = <1>;
        #size-cells = <1>;
        compatible = "simple-bus";
        ranges ;
        dpu: dpu_eu@40000000 {
            clock-names = "s_axi_aclk", "dpu_2x_clk", "m_axi_dpu_aclk";
            clocks = <&clkc 15>, <&misc_clk_0>, <&misc_clk_1>;
            compatible = "xlnx,dpu-eu-3.0";
            interrupt-names = "dpu_interrupt";
            interrupt-parent = <&intc>;
            interrupts = <0 29 4>;
            reg = <0x40000000 0x1000000>;
        };
        misc_clk_0: misc_clk_0 {
            #clock-cells = <0>;
            clock-frequency = <400000000>;
            compatible = "fixed-clock";
        };
        misc_clk_1: misc_clk_1 {
            #clock-cells = <0>;
            clock-frequency = <200000000>;
            compatible = "fixed-clock";
        };
    };
};
```

上面的设备树中可以修改的是***clock-frequency***和***interrupts***。

此外这个设备树信息也可以放在**pl-custom.dtsi**中:

```
/ {
    amba_pl: amba_pl {
        #address-cells = <1>;
        #size-cells = <1>;
        compatible = "simple-bus";
        ranges ;
        dpu: dpu_eu@40000000 {
            clock-names = "s_axi_aclk", "dpu_2x_clk", "m_axi_dpu_aclk";
            clocks = <&clkc 15>, <&misc_clk_0>, <&misc_clk_1>;
            compatible = "xlnx,dpu-eu-3.0";
            interrupt-names = "dpu_interrupt";
            interrupt-parent = <&intc>;
            interrupts = <0 29 4>;
            reg = <0x40000000 0x1000000>;
        };
        misc_clk_0: misc_clk_0 {
            #clock-cells = <0>;
            clock-frequency = <400000000>;
            compatible = "fixed-clock";
        };
        misc_clk_1: misc_clk_1 {
            #clock-cells = <0>;
            clock-frequency = <200000000>;
            compatible = "fixed-clock";
        };
    };
};
```

然后在**system.dtsi**中添加：

```
/include/ "system-conf.dtsi"
/ {
    amba{
        dpu{
            #address-cells = <1>;
            #size-cells = <1>;
            compatible = "xilinx,dpu";
            base-addr = <0x40000000>;
            dpucore {
                compatible = "xilinx,dpucore";
                interrupt-parent = <&intc>;
                interrupts = <0x0 29 0x4>;
                core-num = <0x1>;
            };
        };
    };
    amba_pl@0 {
        /delete-node/ dpu_eu@40000000;
    };
};
```

这样做的好处应该是增加驱动的兼容性，不知道这么想对不对。

注意：DPU的版本更迭比较频繁，网络上已有的设备树的写法多种多样，不管怎么改都要和驱动相匹配，所以修改设备树的时候不能依葫芦画瓢，个人感觉第二种写法好一些，因为删除了dpu_eu这种过于抽象性不好的节点，添加了dpu和dpucore这种高度抽象的节点和子节点，兼容性好一些，不知道驱动会不会也改了。驱动和设备树的关系我还不是很清楚

Build your Project

1.Build

在console中输入 `petalinux-build` 进行编译，离线编译大概15分钟，在线编译会从github上和petalinux.xilinx.com中下载离线包，耗时大概2小时（网速差），1小时（网速好）。推荐使用离线编译。

完成编译后在***/path/to/your/prj/images/linux***下会有以下文件

- 生成u-boot镜像**uImage**，**u-boot.elf**
- linux内核镜像**zImage**，**vmlinux**
- 设备树**system.dtb**
- 根文件系统**rootfs.tar.gz**，**rootfs.cpio**等，
- zynq第一阶段启动文件**fsbl.elf**，
- fpga比特流文件**system.bit**，
- petalinux2020.1多出来的**boot.src文件**

2.打包生成boot文件

前往***/path/to/your/prj/images/linux***目录，在console中输入下列指令，会得到**BOOT.INI**和**image.ub**文件。

`petalinux-package --boot --fsbl --u-boot --fpga`

Deploy your project

- 准备一张大于4GB的SD卡，将SD分为两个部分。第一个部分格式化为FAT32，大小为1GB，作为启动盘**BOOT**。第二个部分为EXT4，>3GB，作为文件系统**ROOTFS**
- 将**BOOT.INI**、**image.ub**和**boot.src**文件拷贝至SD卡的**BOOT**
- 使用下列指令将**rootfs.tar.gz**解压到SD卡的**ROOTFS**

  `sudo tar -zxf rootfs.tar.gz /media/user/ROOTFS/`

  `sync`
- 插卡，设置板子的启动方式，上电，启动。下列是U-boot logfile

```
U-Boot 2020.01 (Jul 31 2020 - 04:01:28 +0000)

CPU:   Zynq 7z030
Silicon: v3.1
DRAM:  ECC disabled 512 MiB
Flash: 0 Bytes
NAND:  512 MiB
MMC:   mmc@e0100000: 0
Loading Environment from SPI Flash... Invalid bus 0 (err=-19)
*** Warning - spi_flash_probe_bus_cs() failed, using default environment

In:    serial@e0000000
Out:   serial@e0000000
Err:   serial@e0000000
Net:
ZYNQ GEM: e000b000, mdio bus e000b000, phyaddr -1, interface rgmii-id

Warning: ethernet@e000b000 using MAC address from DT
eth0: ethernet@e000b000
Hit any key to stop autoboot:  0
switch to partitions #0, OK
```

```
mmc0 is current device
Scanning mmc 0:1...
Found U-Boot script /boot.scr
2010 bytes read in 15 ms (130.9 KiB/s)
## Executing script at 03000000
4347500 bytes read in 260 ms (15.9 MiB/s)
## Loading kernel from FIT Image at 10000000 ...
   Using 'conf@system-top.dtb' configuration
   Verifying Hash Integrity ... OK
   Trying 'kernel@1' kernel subimage
     Description:  Linux kernel
     Type:         Kernel Image
     Compression:  uncompressed
     Data Start:   0x100000f8
     Data Size:    4325856 Bytes = 4.1 MiB
     Architecture: ARM
     OS:           Linux
     Load Address: 0x00200000
     Entry Point:  0x00200000
     Hash algo:    sha256
     Hash value:
c3f1da3c801cf4fe6404aa08803112a95490d9204d34e1de12044579c29a9410
   Verifying Hash Integrity ... sha256+ OK
## Loading fdt from FIT Image at 10000000 ...
   Using 'conf@system-top.dtb' configuration
   Verifying Hash Integrity ... OK
   Trying 'fdt@system-top.dtb' fdt subimage
     Description:  Flattened Device Tree blob
     Type:         Flat Device Tree
     Compression:  uncompressed
     Data Start:   0x104203e4
     Data Size:    19748 Bytes = 19.3 KiB
     Architecture: ARM
     Hash algo:    sha256
     Hash value:
2d24734607098a51f4a5e42e8e2f05a656a7aa1e476b87753c3a635dbbda7c4b
   Verifying Hash Integrity ... sha256+ OK
   Booting using the fdt blob at 0x104203e4
   Loading Kernel Image
   Loading Device Tree to 1eaff000, end 1eb06d23 ... OK

Starting kernel ...

Booting Linux on physical CPU 0x0
Linux version 5.4.0-xilinx-v2020.1 (oe-user@oe-host) (gcc version 9.2.0 (GCC)) #1
SMP PREEMPT Fri Jul 31 03:53:16 UTC 2020
CPU: ARMv7 Processor [413fc090] revision 0 (ARMv7), cr=18c5387d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
OF: fdt: Machine model: xlnx,zynq-7000
earlycon: cdns0 at MMIO 0xe0000000 (options '115200n8')
printk: bootconsole [cdns0] enabled
Memory policy: Data cache writealloc
cma: Reserved 16 MiB at 0x1f000000
percpu: Embedded 15 pages/cpu s31948 r8192 d21300 u61440
Built 1 zonelists, mobility grouping on.  Total pages: 129920
Kernel command line: console=ttyPS0,115200 earlycon root=/dev/mmcblk0p2 rw
rootwait
Dentry cache hash table entries: 65536 (order: 6, 262144 bytes, linear)
```

```
Inode-cache hash table entries: 32768 (order: 5, 131072 bytes, linear)
mem auto-init: stack:off, heap alloc:off, heap free:off
Memory: 491568K/524288K available (6144K kernel code, 217K rwdata, 1840K rodata,
1024K init, 131K bss, 16336K reserved, 16384K cma-reserved, 0K highmem)
rcu: Preemptible hierarchical RCU implementation.
rcu:    RCU restricting CPUs from NR_CPUS=4 to nr_cpu_ids=2.
        Tasks RCU enabled.
rcu: RCU calculated value of scheduler-enlistment delay is 10 jiffies.
rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=2
NR_IRQS: 16, nr_irqs: 16, preallocated irqs: 16
efuse mapped to (ptrval)
slcr mapped to (ptrval)
L2C: platform modifies aux control register: 0x72360000 -> 0x72760000
L2C: DT/platform modifies aux control register: 0x72360000 -> 0x72760000
L2C-310 erratum 769419 enabled
L2C-310 enabling early BRESP for Cortex-A9
L2C-310 full line of zeros enabled for Cortex-A9
L2C-310 ID prefetch enabled, offset 1 lines
L2C-310 dynamic clock gating enabled, standby mode enabled
L2C-310 cache controller enabled, 8 ways, 512 kB
L2C-310: CACHE_ID 0x410000c8, AUX_CTRL 0x76760001
random: get_random_bytes called from start_kernel+0x260/0x440 with crng_init=0
zynq_clock_init: clkc starts at (ptrval)
Zynq clock init
sched_clock: 64 bits at 333MHz, resolution 3ns, wraps every 4398046511103ns
clocksource: arm_global_timer: mask: 0xffffffffffffffff max_cycles: 0x4ce07af025,
max_idle_ns: 440795209040 ns
Switching to timer-based delay loop, resolution 3ns
Console: colour dummy device 80x30
Calibrating delay loop (skipped), value calculated using timer frequency.. 666.66
BogoMIPS (lpj=3333333)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 1024 (order: 0, 4096 bytes, linear)
Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes, linear)
CPU: Testing write buffer coherency: ok
CPU0: Spectre v2: using BPIALL workaround
CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
Setting up static identity map for 0x100000 - 0x100060
rcu: Hierarchical SRCU implementation.
smp: Bringing up secondary CPUs ...
CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
CPU1: Spectre v2: using BPIALL workaround
smp: Brought up 1 node, 2 CPUs
SMP: Total of 2 processors activated (1333.33 BogoMIPS).
CPU: All CPU(s) started in SVC mode.
devtmpfs: initialized
VFP support v0.3: implementor 41 architecture 3 part 30 variant 9 rev 4
clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns:
19112604462750000 ns
futex hash table entries: 512 (order: 3, 32768 bytes, linear)
pinctrl core: initialized pinctrl subsystem
NET: Registered protocol family 16
DMA: preallocated 256 KiB pool for atomic coherent allocations
cpuidle: using governor menu
hw-breakpoint: found 5 (+1 reserved) breakpoint and 1 watchpoint registers.
hw-breakpoint: maximum watchpoint size is 4 bytes.
zynq-ocm f800c000.ocmc: ZYNQ OCM pool: 256 KiB @ 0x(ptrval)
```

```
e0000000.serial: ttyPS0 at MMIO 0xe0000000 (irq = 25, base_baud = 6249999) is a
xuartps
printk: console [ttyPS0] enabled
printk: console [ttyPS0] enabled
printk: bootconsole [cdns0] disabled
printk: bootconsole [cdns0] disabled
e0001000.serial: ttyPS1 at MMIO 0xe0001000 (irq = 26, base_baud = 6249999) is a
xuartps
vgaarb: loaded
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
mc: Linux media interface: v0.10
videodev: Linux video capture interface: v2.00
pps_core: LinuxPPS API ver. 1 registered
pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo Giometti
<giometti@linux.it>
PTP clock support registered
EDAC MC: Ver: 3.0.0
FPGA manager framework
Advanced Linux Sound Architecture Driver Initialized.
clocksource: Switched to clocksource arm_global_timer
thermal_sys: Registered thermal governor 'step_wise'
NET: Registered protocol family 2
tcp_listen_portaddr_hash hash table entries: 512 (order: 0, 6144 bytes, linear)
TCP established hash table entries: 4096 (order: 2, 16384 bytes, linear)
TCP bind hash table entries: 4096 (order: 3, 32768 bytes, linear)
TCP: Hash tables configured (established 4096 bind 4096)
UDP hash table entries: 256 (order: 1, 8192 bytes, linear)
UDP-Lite hash table entries: 256 (order: 1, 8192 bytes, linear)
NET: Registered protocol family 1
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
PCI: CLS 0 bytes, default 64
hw perfevents: no interrupt-affinity property for /pmu@f8891000, guessing.
hw perfevents: enabled with armv7_cortex_a9 PMU driver, 7 counters available
workingset: timestamp_bits=14 max_order=17 bucket_order=3
jffs2: version 2.2. (NAND) (SUMMARY)  ? 2001-2006 Red Hat, Inc.
io scheduler mq-deadline registered
io scheduler kyber registered
zynq-pinctrl 700.pinctrl: zynq pinctrl initialized
dma-pl330 f8003000.dmac: Loaded driver for PL330 DMAC-241330
dma-pl330 f8003000.dmac:        DBUFF-128x8bytes Num_Chans-8 Num_Peri-4
Num_Events-16
brd: module loaded
loop: module loaded
libphy: Fixed MDIO Bus: probed
CAN device driver interface
libphy: MACB_mii_bus: probed
Marvell 88E1116R e000b000.ethernet-ffffffff:07: attached PHY driver [Marvell
88E1116R] (mii_bus:phy_addr=e000b000.ethernet-ffffffff:07, irq=POLL)
macb e000b000.ethernet eth0: Cadence GEM rev 0x00020118 at 0xe000b000 irq 28
(00:0a:35:00:1e:53)
e1000e: Intel(R) PRO/1000 Network Driver - 3.2.6-k
e1000e: Copyright(c) 1999 - 2015 Intel Corporation.
```

```
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
ehci-pci: EHCI PCI platform driver
usbcore: registered new interface driver usb-storage
i2c /dev entries driver
cdns-i2c e0004000.i2c: 400 kHz mmio e0004000 irq 22
cdns-wdt f8005000.watchdog: Xilinx Watchdog Timer with timeout 10s
EDAC MC: ECC not enabled
Xilinx Zynq CpuIdle Driver started
sdhci: Secure Digital Host Controller Interface driver
sdhci: Copyright(c) Pierre Ossman
sdhci-pltfm: SDHCI platform and OF driver helper
mmc0: SDHCI controller on e0100000.mmc [e0100000.mmc] using ADMA
ledtrig-cpu: registered to indicate activity on CPUs
clocksource: ttc_clocksource: mask: 0xffff max_cycles: 0xffff, max_idle_ns:
537538477 ns
timer #0 at (ptrval), irq=41
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
nand: device found, Manufacturer ID: 0x2c, Chip ID: 0xdc
nand: Micron MT29F4G08ABADAWP
nand: 512 MiB, SLC, erase size: 128 KiB, page size: 2048, OOB size: 64
nand: WARNING: MT29F4G08ABADAWP: the ECC used on your system is too weak compared
to the one required by the NAND chip
pl353_nand_calculate_hwecc status failed
pl353_nand_calculate_hwecc status failed
pl353_nand_calculate_hwecc status failed
pl353_nand_calculate_hwecc status failed
Bad block table not found for chip 0
pl353_nand_calculate_hwecc status failed
pl353_nand_calculate_hwecc status failed
pl353_nand_calculate_hwecc status failed
pl353_nand_calculate_hwecc status failed
Bad block table not found for chip 0
Scanning device for bad blocks
mmc0: new high speed SDHC card at address 59b4
mmcblk0: mmc0:59b4 USD00 7.42 GiB
 mmcblk0: p1 p2
Bad block table written to 0x00001ffe0000, version 0x01
Bad block table written to 0x00001ffc0000, version 0x01
fpga_manager fpga0: Xilinx Zynq FPGA Manager registered
NET: Registered protocol family 10
Segment Routing with IPv6
sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
NET: Registered protocol family 17
can: controller area network core (rev 20170425 abi 9)
NET: Registered protocol family 29
can: raw protocol (rev 20170425)
can: broadcast manager protocol (rev 20170425 t)
can: netlink gateway (rev 20190810) max_hops=1
Registering SWP/SWPB emulation handler
of-fpga-region fpga-full: FPGA Region probed
hctosys: unable to open rtc device (rtc0)
of_cfs_init
of_cfs_init: OK
ALSA device list:
  No soundcards found.
random: fast init done
EXT4-fs (mmcblk0p2): recovery complete
```

```
EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (null)
VFS: Mounted root (ext4 filesystem) on device 179:2.
devtmpfs: mounted
Freeing unused kernel memory: 1024K
Run /sbin/init as init process
INIT: version 2.88 booting
Starting udev
udevd[70]: starting version 3.2.8
random: udevd: uninitialized urandom read (16 bytes read)
random: udevd: uninitialized urandom read (16 bytes read)
random: udevd: uninitialized urandom read (16 bytes read)
udevd[71]: starting eudev-3.2.8
FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt.
Please run fsck.
EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
hwclock: Cannot access the Hardware Clock via any known method.
hwclock: Use the --verbose option to see the details of our search for an access
method.
Wed Jul 29 09:42:09 UTC 2020
hwclock: Cannot access the Hardware Clock via any known method.
hwclock: Use the --verbose option to see the details of our search for an access
method.
urandom_read: 2 callbacks suppressed
random: dd: uninitialized urandom read (512 bytes read)
INIT: Entering runlevel: 5
Configuring network interfaces... udhcpc: started, v1.31.0
udhcpc: sending discover
udhcpc: sending discover
udhcpc: sending discover
udhcpc: no lease, forking to background
done.
Starting system message bus: random: dbus-daemon: uninitialized urandom read (12
bytes read)
random: dbus-daemon: uninitialized urandom read (12 bytes read)
dbus.
Starting random number generator daemon
Initializing available sources

Failed to init entropy source hwrng

Initializing AES buffer

Enabling JITTER rng support

Initializing entropy source jitter

.
random: crng init done
Starting haveged: haveged: listening socket at 3
haveged: haveged starting up


Starting OpenBSD Secure Shell server: sshd
done.
Starting Xserver
Starting rpcbind daemon...
done.
starting statd:
```

```
X.Org X Server 1.20.5
X Protocol Version 11, Revision 0
Build Operating System: Linux 3.10.0-693.el7.x86_64 x86_64
Current Operating System: Linux zing2dpu 5.4.0-xilinx-v2020.1 #1 SMP PREEMPT Fri
Jul 31 03:53:16 UTC 2020 armv7l
Kernel command line: console=ttyPS0,115200 earlycon root=/dev/mmcblk0p2 rw
rootwait
Build Date: 26 May 2020  04:13:44PM

Current version of pixman: 0.38.4
        Before reporting problems, check http://wiki.x.org
        to make sure that you have the latest version.
Markers: (--) probed, (**) from config file, (==) default setting,
        (++) from command line, (!!) notice, (II) informational,
        (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(==) Log file: "/var/log/Xorg.0.log", Time: Wed Jul 29 09:42:39 2020
(==) Using system config directory "/usr/share/X11/xorg.conf.d"
(EE)
Fatal server error:
done
(EE) no screens found(EE)
(EE)
Please consult the The X.Org Foundation support
         at http://wiki.x.org
 for help.
(EE) Please also check the log file at "/var/log/Xorg.0.log" for additional
information.
(EE)
(EE) Server terminated with error (1). Closing log file.
Starting bluetooth: bluetoothd.
hwclock: Cannot access the Hardware Clock via any known method.
hwclock: Use the --verbose option to see the details of our search for an access
method.
Starting internet superserver: inetd.
exportfs: can't open /etc/exports for reading
NFS daemon support not enabled in kernel
Starting syslogd/klogd: done
Starting internet superserver: xinetd.
 * Starting Avahi mDNS/DNS-SD Daemon: avahi-daemon                       [ ok ]
Starting watchdog daemon...done
Starting tcf-agent: OK

PetaLinux 2020.1 zing2dpu ttyPS0



Login incorrect
zing2dpu login: root
Password:
root@zing2dpu:~#
```