Programming Linux to QSPI/eMMC on the Smart Vision Development Kit

Version 2014.4.01

LIT# 5255-SVDK-Programming-Linux-to-QSPI-eMMC-2014-4-01

Document Control

Document Version: 2014_4.01

Document Date: 08/27/2015

Prior Version History

Version	Date	Comment
2014.4.01	08/27/2015	Initial Release

Contents

1 Overview

This document describes how to program Linux to the PicoZed 7015T SOM's QSPI/eMMC devices for the Smart Vision Development Kit (SVDK).

As a real world example, the Linux binaries for the MVTEC HALCON design will be used.

These binaries consist of the following files:

File	Description	Size
BOOT.BIN	Boot files, including: First Stage Boot Loader (FSBL) application Bitstream U-Boot application	3.763 MBytes
Image.ub	Linux files, including: Linux kernel Device tree blob Root file system	29.862 MBytes

Figure 1 – MVTEC HALCON Linux binaries

The PicoZed 7015T SOM has the following two non-volatile storage devices:

Storage Device	Description / Notes	Size
QSPI	Spansion S25GL128S : - 4-bit SPI (quad-SPI) serial NOR flash Can be used to boot Zynq device	16 MBytes (128 Mbit)
eMMC	Micron MTFC4GMDEA-4M IT eMMC - Multi Media Controller and NAND Flash Cannot be used to boot Zynq device Can be used to store larger secondary files	4 GBytes

Figure 2 – PicoZed 7015T non-volatile storage devices

At this point, two important points should be obvious:

- The BOOT.BIN boot file must be programmed to the QSPI firmware, since the Zynq device only support this device during initial boot
- The image.ub linux file cannot be programmed to QSPI, since it is too large for this device. It must be programmed to eMMC

The following procedure will be used to program these linux binaries to the PicoZed 7015T SOM using the SVDK:

1. Program an intermediate version of BOOT.BIN (configured to boot entirely from QSPI) and image.ub (smaller is size, which fits in QSPI) to the QSPI device

2. Using this intermediate version of Linux, format the eMMC device and program the final image.ub file to the eMMC device

3. Update the QSPI with the final BOOT.BIN (configured to boot the linux files from eMMC)

For the intermedia (QSPI only) version of Linux, the BOOT_QSPI.BIN and image.ub files will be taken from the 2014.4 PetaLinux BSP, available on picozed.org. The files we are interested in can be found in the following directory:

Avnet-PicoZed-7z015-v2014.4/pz_7z015/pre-built/images/

2 Programming the Intermediate version of Linux to the QSPI device

The contents of the QSPI device, as interpreted by Linux, is defined by the XXX. In our case, the QSPI device is divided into four partitions:

Partition	Name / Usage	Start	Size
mtd0	Boot - used for BOOT.BIN	0x00000000	0x00500000 (5.242 MBytes)
mtd1	Bootenv - used to store u-boot environment variables	0x00500000	0x00020000 (0.131 MBytes)
mtd2	Kernel - used to store image.ub	0x00520000	0x00A80000 (11.010 MBytes)
mtd3	Spare - additional storage (if needed)	0x00FA0000	0x00060000 (0.393 MBytes)

Figure 3 – QSPI partitions defined by Linux

We want to program the intermediate BOOT_QSPI.BIN at offset 0x00000000 in the QSPI device.

We want to program the intermediate image.ub at offset 0x00520000 in the QSPI device. Also, the image.ub must have a size less than 11MBytes, which is the case for the intermediate image.ub file (8MBytes).

2.1 Programming BOOT_QSPI.BIN to the QSPI device

The intermediate BOOT_QSPI.BIN will be programmed at offset 0x00000000 in the QSPI device, as described below.

1. Connect a JTAG Programming Cable (Platform Cable, Digilent HS1 or HS2 cable), not included with the kit, to the PC using a USB cable and then plug the 14-Pin PC4 header or cable into the PC4 connector on the SVDK carrier card.

2. Launch **SDK 2014.4** (or greater)

3. Specify the following SDK workspace
   ..\SVDK_Programming_Linux_to_QSPI_eMMC\sdk_workspace

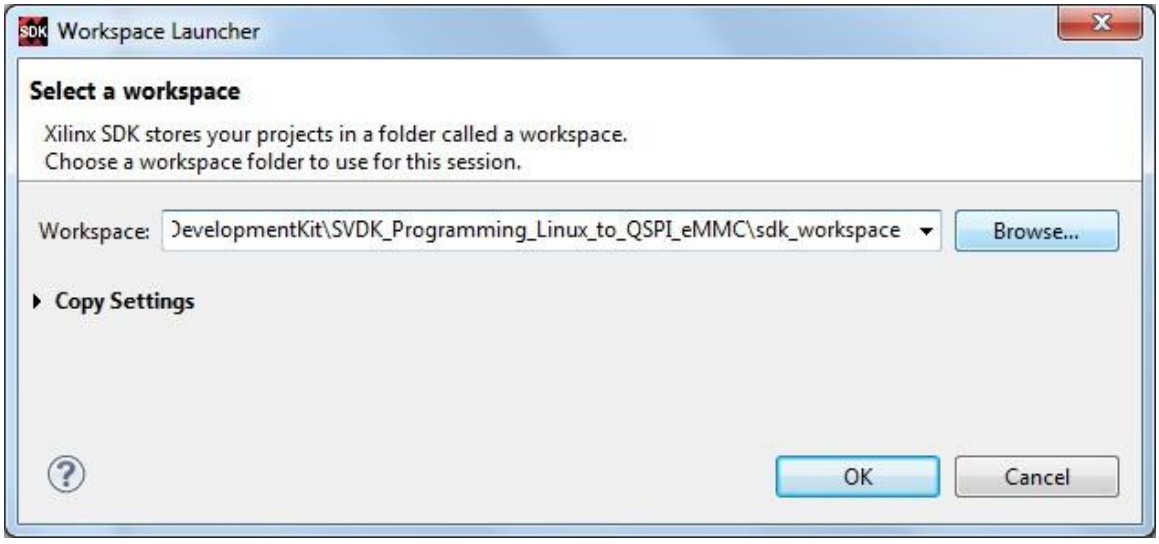

Figure 4 – SDK - Specify SDK workspace

4. Click **OK**

5. In the Project Explorer, you should see the "design_1_wrapper_hw_platform_0" hardware projects, which is required to use the "Program Flash" feature

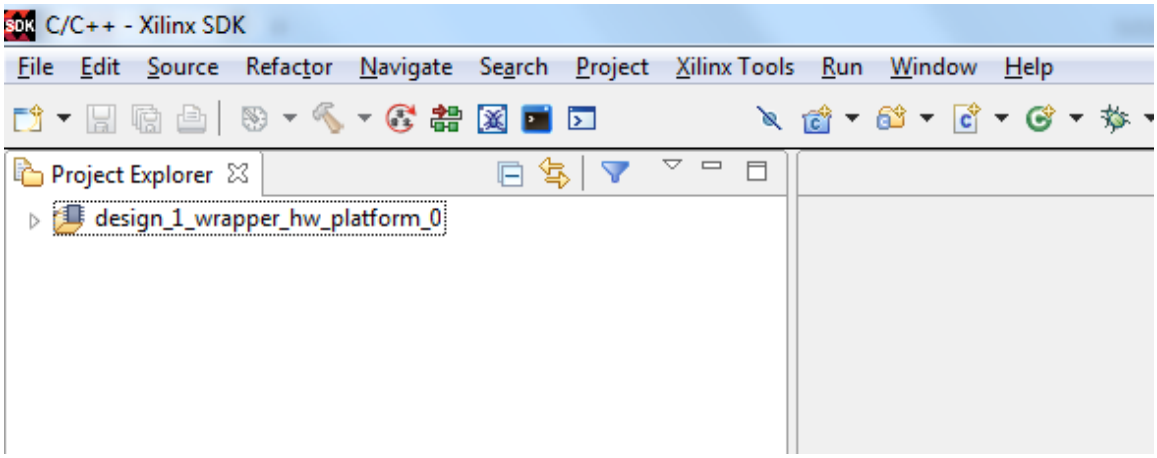

Figure 5 – SDK - "design_1_wrapper_hw_platform_0" hardware project

6. In the SDK menu, select **Xilinx Tools => Program Flash**

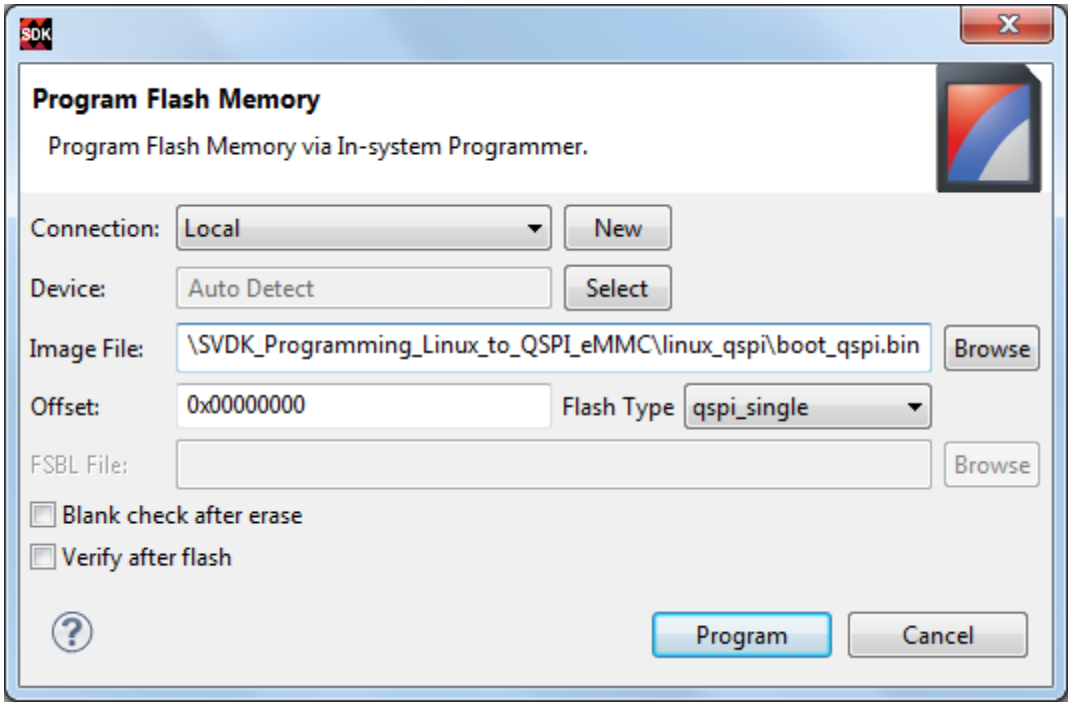

Figure 6 – SDK – Program boot_qspi.bin to QSPI at offset 0x00000000

7. Program offset 0x00000000 of the QSPI device with the following boot file **..\SVDK_Programming_Linux_to_QSPI_eMMC\linux_qspi\boot_qspi.bin**

8. Click **Program**

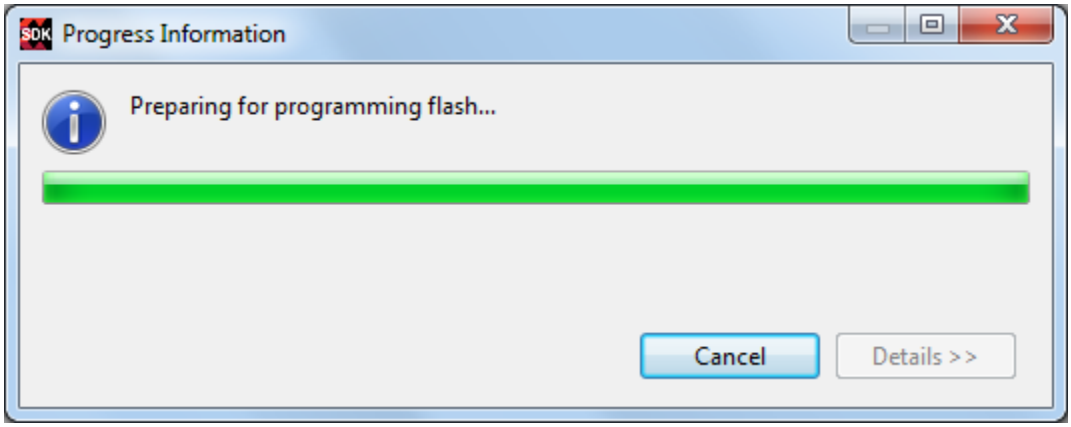

Figure 7 – SDK – Programming flash …

2.2 Programming IMAGE.UB to the QSPI device

The intermediate image.ub will be programmed at offset 0x00520000 in the QSPI device, as described below.

1. In the SDK menu, select **Xilinx Tools => Program Flash**

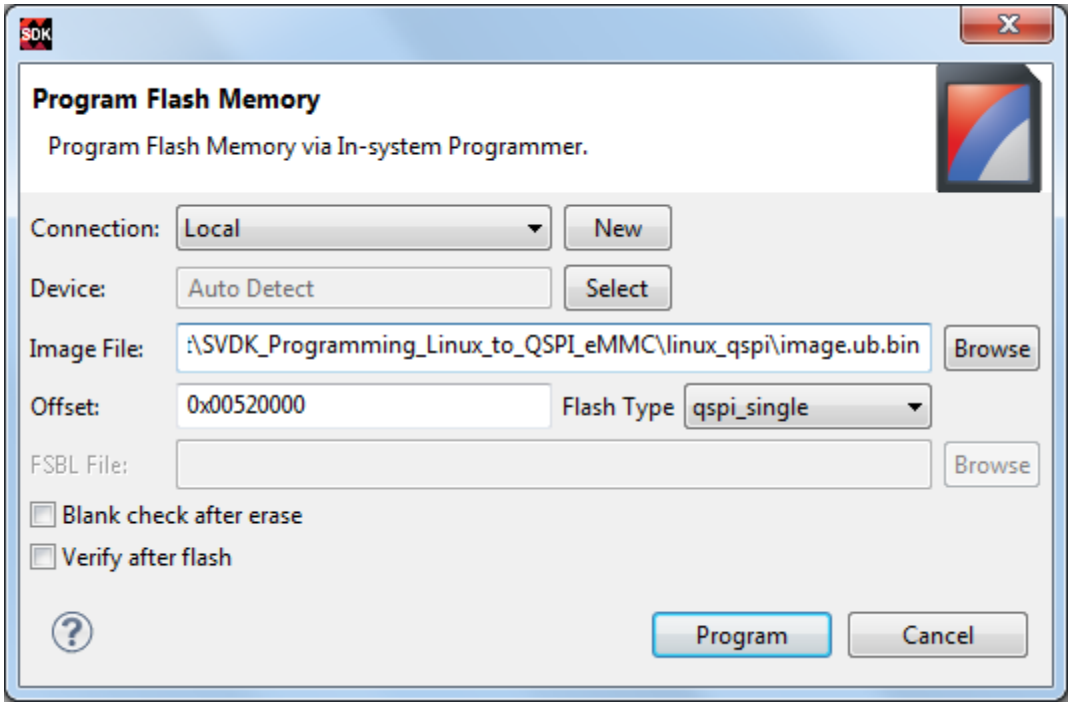

Figure 8 – SDK – Program image.ub to QSPI at offset 0x00520000

2. Program offset 0x00520000 of the QSPI device with the following boot file **..\SVDK_Programming_Linux_to_QSPI_eMMC\linux_qspi\image.ub.bin**

   NOTE: the image.ub was renamed as image.ub.bin since the wizard only accepts .mcs or .bin files

3. Click **Program**

2.3 Booting the intermediate (QSPI only) version of Linux

At this point, we have successfully programmed the intermediate version of Linux that will boot entirely from QSPI flash.

1. Turn the power switch on the SVDK to the ON position. After 1-2 seconds, you will notice several LEDs that are lit, including the FPGA DONE LED (DS1).

2. Now plug in the mini-USB-B to USB-A cable between the PC and the USB Serial (J17) connector on SVDK.

3. On the PC, open a serial terminal program. Tera Term is used to show the example output for this lab document. Follow the instructions in the CP210x Setup Guide to set the terminal as shown in Figure 9, using the appropriate COM port that you discover on your own machine.

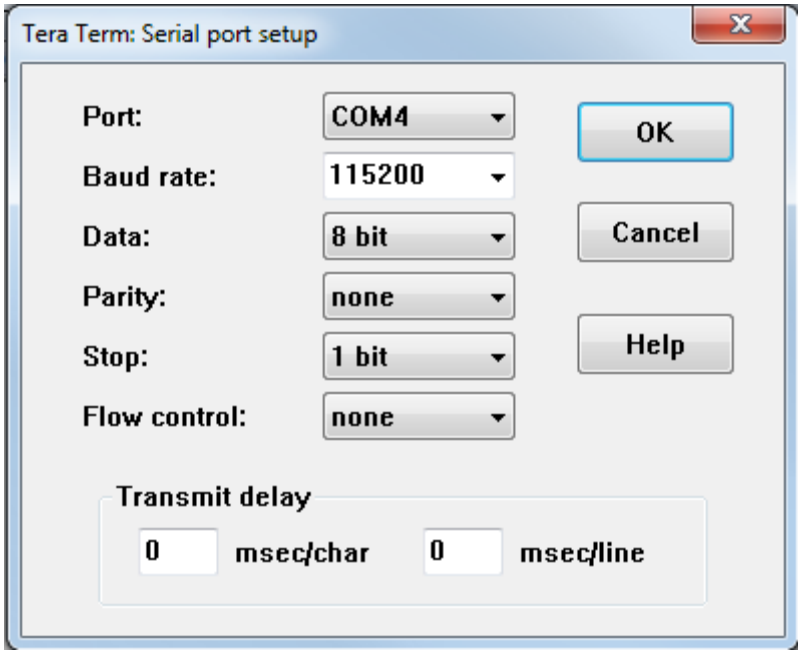

Figure 9 – Connect Tera Term to the proper COMx port

4. Cycle power on the SVDK. The terminal output displays feedback from the linux console.

```
U-Boot 2014.07 (Apr 22 2015 - 16:53:32)

DRAM:ECC disabled 1 GiB
MMC: zynq_sdhci: 0
```

```
SF: Detected S25FL128S_64K with page size 256 Bytes, erase size
64 KiB, total 16 MiB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   Gem.e000b000
U-BOOT for pz-7z015

Gem.e000b000 Waiting for PHY auto negotiation to complete.....
done
BOOTP broadcast 1
DHCP client bound to address 192.168.0.111
Hit any key to stop autoboot:  0
SF: Detected S25FL128S_64K with page size 256 Bytes, erase size
64 KiB, total 16 MiB
SF: 11010048 bytes @ 0x520000 Read: OK
## Loading kernel from FIT Image at 01000000 ...
   Using 'conf@1' configuration
   Trying 'kernel@1' kernel subimage
     Description:  PetaLinux Kernel
     Type:         Kernel Image
     Compression:  gzip compressed
     Data Start:   0x010000f0
     Data Size:    6848045 Bytes = 6.5 MiB
     Architecture: ARM
     OS:           Linux
     Load Address: 0x00008000
     Entry Point:  0x00008000
     Hash algo:    crc32
     Hash value:   3b35f7d2
   Verifying Hash Integrity ... crc32+ OK
## Loading fdt from FIT Image at 01000000 ...
   Using 'conf@1' configuration
   Trying 'fdt@1' fdt subimage
     Description:  Flattened Device Tree blob
     Type:         Flat Device Tree
     Compression:  uncompressed
     Data Start:   0x01688004
     Data Size:    13529 Bytes = 13.2 KiB
     Architecture: ARM
     Hash algo:    crc32
     Hash value:   69a80c82
   Verifying Hash Integrity ... crc32+ OK
   Booting using the fdt blob at 0x1688004
   Uncompressing Kernel Image ... OK
   Loading Device Tree to 07ff9000, end 07fff4d8 ... OK

Starting kernel ...

Booting Linux on physical CPU 0x0
Linux version 3.17.0-xilinx (training@VBCentOS6) (gcc version
4.8.3 20140320 (prerelease) (Sourcery CodeBench Lite 2014.05-23)
) #2 SMP PREEMPT Wed Apr 22 11:27:31 MDT 2015
```

```
CPU: ARMv7 Processor [413fc090] revision 0 (ARMv7), cr=18c5387d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing
instruction cache
Machine model: pz-7z015
cma: Reserved 128 MiB at 38000000
Memory policy: Data cache writealloc
PERCPU: Embedded 8 pages/cpu @7779c000 s8704 r8192 d15872 u32768
Built 1 zonelists in Zone order, mobility grouping on. Total
pages: 260096
Kernel command line: console=ttyPS0,115200 earlyprintk
PID hash table entries: 4096 (order: 2, 16384 bytes)
Dentry cache hash table entries: 131072 (order: 7, 524288 bytes)
Inode-cache hash table entries: 65536 (order: 6, 262144 bytes)
Memory: 897840K/1048576K available (4623K kernel code, 239K
rwdata, 1588K rodata, 3576K init, 210K bss, 150736K reserved, 0K
highmem)
Virtual kernel memory layout:
    vector  : 0xffff0000 - 0xffff1000   (   4 kB)
    fixmap  : 0xffc00000 - 0xffe00000   (2048 kB)
    vmalloc : 0x80800000 - 0xff000000   (2024 MB)
    lowmem  : 0x40000000 - 0x80000000   (1024 MB)
    pkmap   : 0x3fe00000 - 0x40000000   (   2 MB)
    modules : 0x3f000000 - 0x3fe00000   (  14 MB)
      .text : 0x40008000 - 0x406190f8   (6213 kB)
      .init : 0x4061a000 - 0x40998200   (3577 kB)
      .data : 0x4099a000 - 0x409d5de0   ( 240 kB)
       .bss : 0x409d5de0 - 0x40a0a6ec   ( 211 kB)
Preemptible hierarchical RCU implementation.
        Dump stacks of tasks blocking RCU-preempt GP.
        RCU restricting CPUs from NR_CPUS=4 to nr_cpu_ids=2.
RCU: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=2
NR_IRQS:16 nr_irqs:16 16
L2C: platform provided aux values match the hardware, so have no
effect.  Please remove them.
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 0x76360001
slcr mapped to 80804000
zynq_clock_init: clkc starts at 80804100
Zynq clock init
sched_clock: 64 bits at 333MHz, resolution 3ns, wraps every
3298534883328ns
timer #0 at 80806000, irq=43
Console: colour dummy device 80x30
Calibrating delay loop... 1332.01 BogoMIPS (lpj=6660096)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 2048 (order: 1, 8192 bytes)
Mountpoint-cache hash table entries: 2048 (order: 1, 8192 bytes)
CPU: Testing write buffer coherency: ok
CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
Setting up static identity map for 0x45eb00 - 0x45eb58
```

```
CPU1: Booted secondary processor
CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
Brought up 2 CPUs
SMP: Total of 2 processors activated.
CPU: All CPU(s) started in SVC mode.
devtmpfs: initialized
VFP support v0.3: implementor 41 architecture 3 part 30 variant 9
rev 4
regulator-dummy: no parameters
NET: Registered protocol family 16
DMA: preallocated 256 KiB pool for atomic coherent allocations
cpuidle: using governor ladder
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 @ 0x80880000
VCCPINT: 1000 mV
vgaarb: loaded
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
media: Linux media interface: v0.10
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
Advanced Linux Sound Architecture Driver Initialized.
Switched to clocksource arm_global_timer
NET: Registered protocol family 2
TCP established hash table entries: 8192 (order: 3, 32768 bytes)
TCP bind hash table entries: 8192 (order: 4, 65536 bytes)
TCP: Hash tables configured (established 8192 bind 8192)
TCP: reno registered
UDP hash table entries: 512 (order: 2, 16384 bytes)
UDP-Lite hash table entries: 512 (order: 2, 16384 bytes)
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.
hw  perfevents:  enabled  with  armv7_cortex_a9  PMU  driver,  7
counters available
futex hash table entries: 512 (order: 3, 32768 bytes)
jffs2: version 2.2. (NAND) (SUMMARY)  © 2001-2006 Red Hat, Inc.
msgmni has been set to 2009
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
dma-pl330 f8003000.dmac: Loaded driver for PL330 DMAC-2364208
dma-pl330  f8003000.dmac:            DBUFF-128x8bytes  Num_Chans-8
Num_Peri-4 Num_Events-16
```

```
e0001000.serial: ttyPS0 at MMIO 0xe0001000 (irq = 82, base_baud =
3125000) is a xuartps
console [ttyPS0] enabled
xdevcfg f8007000.devcfg: ioremap 0xf8007000 to 8086a000
[drm] Initialized drm 1.1.0 20060810
brd: module loaded
loop: module loaded
m25p80 spi32766.0: found s25fl129p1, expected n25q128a13
m25p80 spi32766.0: s25fl129p1 (16384 Kbytes)
4 ofpart partitions found on MTD device spi32766.0
Creating 4 MTD partitions on "spi32766.0":
0x000000000000-0x000000500000 : "boot"
0x000000500000-0x000000520000 : "bootenv"
0x000000520000-0x000000fa0000 : "kernel"
0x000000fa0000-0x000001000000 : "spare"
CAN device driver interface
e1000e: Intel(R) PRO/1000 Network Driver - 2.3.2-k
e1000e: Copyright(c) 1999 - 2014 Intel Corporation.
libphy: XEMACPS mii bus: probed
xemacps e000b000.ethernet: pdev->id -1, baseaddr 0xe000b000, irq
54
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
ehci-pci: EHCI PCI platform driver
ULPI transceiver vendor/product ID 0x0424/0x0007
Found SMSC USB3320 ULPI transceiver.
ULPI integrity check: passed.
zynq-ehci zynq-ehci.0: Xilinx Zynq USB EHCI Host Controller
zynq-ehci  zynq-ehci.0:  new  USB  bus  registered,  assigned  bus
number 1
zynq-ehci zynq-ehci.0: irq 53, io mem 0x00000000
zynq-ehci zynq-ehci.0: USB 2.0 started, EHCI 1.00
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
usbcore: registered new interface driver usb-storage
mousedev: PS/2 mouse device common for all mice
i2c /dev entries driver
cdns-wdt  f8005000.watchdog:  Xilinx  Watchdog  Timer  at  80872000
with timeout 10s
zynq-edac f8006000.memory-controller: 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
sdhci-arasan e0100000.sdhci: No vmmc regulator found
sdhci-arasan e0100000.sdhci: No vqmmc regulator found
mmc0:  SDHCI  controller  on  e0100000.sdhci  [e0100000.sdhci]  using
ADMA
sdhci-arasan e0101000.sdhci: No vmmc regulator found
sdhci-arasan e0101000.sdhci: No vqmmc regulator found
mmc1:  SDHCI  controller  on  e0101000.sdhci  [e0101000.sdhci]  using
ADMA
ledtrig-cpu: registered to indicate activity on CPUs
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
TCP: cubic registered
```

```
NET: Registered protocol family 17
can: controller area network core (rev 20120528 abi 9)
NET: Registered protocol family 29
can: raw protocol (rev 20120528)
can: broadcast manager protocol (rev 20120528 t)
can: netlink gateway (rev 20130117) max_hops=1
Registering SWP/SWPB emulation handler
/opt/pkg/petalinux-v2014.4-final/components/linux-kernel/xlnx-
3.17/drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
ALSA device list:
  No soundcards found.
Freeing unused kernel memory: 3576K (4061a000 - 40998000)
INIT: version 2.88 booting
Creating /dev/flash/* device nodes
random: dd urandom read with 2 bits of entropy available
mmc1: BKOPS_EN bit is not set
mmc1: new high speed MMC card at address 0001
mmcblk0: mmc1:0001 MMC04G 3.52 GiB
mmcblk0boot0: mmc1:0001 MMC04G partition 1 16.0 MiB
mmcblk0boot1: mmc1:0001 MMC04G partition 2 16.0 MiB
mmcblk0rpmb: mmc1:0001 MMC04G partition 3 128 KiB
 mmcblk0: p1
 mmcblk0boot1: unknown partition table
 mmcblk0boot0: unknown partition table
starting Busybox inet Daemon: inetd... done.
Starting uWeb server:
NET: Registered protocol family 10
update-rc.d:  /etc/init.d/run-postinsts  exists  during  rc.d  purge
(continuing)
 Removing any system startup links for run-postinsts ...
INIT: Entering runlevel: 5
Configuring network interfaces... udhcpc (v1.22.1) started
Sending discover...
Sending discover...
xemacps e000b000.ethernet: Set clk to 124999999 Hz
xemacps e000b000.ethernet: link up (1000/FULL)
Sending discover...
Sending select for 192.168.0.111...
Lease of 192.168.0.111 obtained, lease time 86400
/etc/udhcpc.d/50default: Adding DNS 192.168.0.1
done.

Built with PetaLinux v2014.4 (Yocto 1.7) pz-7z015 /dev/ttyPS0
pz-7z015 login: root
Password: root
login[878]: root login on 'ttyPS0'
root@pz-7z015:~#
```

Figure 10 – SVDK – Linux Boot Console

5. Enter the login credentials (login = **root**, password = **root**).

3 Programming the final version of Linux to the QSPI and eMMC devices

The intermediate version of Linux was required in order to format the eMMC device and program the larger image.ub file to the eMMC device. We will first format/program the eMMC device. Then, we will program the final BOOT.BIN file to the QSPI device.

3.1 Formatting the eMMC device

Before we can use the eMMC device, it must be partitioned and formatted, as described below:

1. To view the Multi-Media card devices attached to the SVDK system by listing the **/dev** directory for mmc devices. The /dev/mmcblk0 entry corresponds to the controller designated for the eMMC memory device **(ls /dev/mmc*)**

```
root@pz-7z015:~# ls /dev/mmc*
/dev/mmcblk0                /dev/mmcblk0boot0     /dev/mmcblk0boot1
/dev/mmcblk0rpmb
```

2. The Linux fdisk utility is used to create a partition on the storage media for use with a file system. Enter the commands as shown below:

   a. Start the fdisk utility for the eMMC controller. **(fdisk <device name>)**

```
root@pz-7z015:~# fdisk /dev/mmcblk0
Device contains neither a valid DOS partition table, nor Sun,
SGI, OSF or GPT disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that the previous content
won't be recoverable.


The number of cylinders for this disk is set to 115456.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):
```

b. List the existing partition information. If the storage media has never been used, there should be no partitions shown.

```
Command (m for help): p

Disk /dev/mmcblk0: 3783 MB, 3783262208 bytes
4 heads, 16 sectors/track, 115456 cylinders
Units = cylinders of 64 * 512 = 32768 bytes

   Device Boot      Start         End      Blocks   Id System

Command (m for help):
```

c. Create a new primary partition #1 starting at the first cylinder and extending for 128 MB

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-115456, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-115456, default
115456): +128M

Command (m for help):
```

d. Write the new partition to the eMMC device and exit fdisk

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table
 mmcblk0: p1
root@pz-7z015:~#
```

3. Before the new partition can be used, it must be formatted with a FAT32 file system. Use the Linux mkdosfs utility to perform this action. **(mkdosfs –F 32 <device name>)**

```
root@pz-7z015:~# mkdosfs -F 32 /dev/mmcblk0p1
root@pz-7z015:~#
```

4. Create a mount point for the eMMC partition…

```
root@pz-7z015:~# mkdir /temp
root@pz-7z015:~# mount /dev/mmcblk0p1 /temp
root@pz-7z015:~# ls /temp
```

3.2 Copying IMAGE.UB to the SVDK via FTP

The intermediate version of Linux has a FTP server, allowing large files to be transferred over the network.

1. Connect an Ethernet cable from a DHCP network to the SVSK's **J11** connector.

2. Query the embedded linux for the IP address using the **ifconfig** command. In the example below, the SVDK has the IP address 192.168.0.111.

```
root@pz-7z015:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0A:35:00:E2:8A
          inet addr:192.168.0.111 Bcast:0.0.0.0 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2828 errors:0 dropped:142 overruns:0 frame:0
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:367972 (359.3 KiB)  TX bytes:2514 (2.4 KiB)
          Interrupt:54 Base address:0xb000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@pz-7z015:~#
```

3. FTP will be used to transfer the large image.ub file to the SVDK's linux file system. File transfers will be placed in the **/var/ftp** directory, which should be empty at this time.

```
root@pz-7z015:~# ls /var/ftp
root@pz-7z015:~#
```

4. On the PC, launch a Command Prompt from the Start menu :

   Start => All Programs => Accessories => Command Prompt

5. Change to the following directory using the cd command.

   ..\SVDK_Programming_Linux_to_QSPI_eMMC\linux_halcon

6. Start a FTP session with the ftp command (**ftp <IP address>**)
7. Login as the **root** user.
8. Enter the **binary** command to indicate that we want to transfer a binary file (non text/ascii).
9. Enter the **put image.ub** command to transfer the image.ub file to the SVDK's linux file system.
10. When done, type the **quit** command

```
C:\SVDK_Prog_Linux_to_QSPI_eMMC\linux_halcon>ftp 192.168.0.111
Connected to 192.168.0.111.
220 Operation successful
User (192.168.0.111:(none)): root
230 Operation successful
ftp> binary
200 Operation successful
ftp> put image.ub
200 Operation successful
150 Ok to send data
226 Operation successful
ftp: 30578008 bytes sent in 0.54Seconds 56313.09Kbytes/sec.
ftp> quit
```

11. On the SVDK, verify that the image.ub file was correctly transferred, using the **ls command.**

```
root@pz-7z015:~# ls /var/ftp
image.ub
root@pz-7z015:~# ls -la /var/ftp
drwxrwxr-x    2 root     root            60 Jan  1 00:28 .
drwxrwxr-x    8 root     root           240 Apr 22  2015 ..
-rw-r--r--    1 root     root      30578008 Jan  1 00:28 image.ub
root@pz-7z015:~#
```

3.3 Programming IMAGE.UB to the eMMC device

In order to program the image.ub file to the eMMC device, a simple file copy is all that is required.

1. Copy the image.ub file from the /var/ftp directory to the /temp directory

```
root@pz-7z015:~# ls /temp
root@pz-7z015:~# cp /var/ftp/image.ub /temp/.
root@pz-7z015:~# ls /temp
image.ub
root@pz-7z015:~# ls -la /temp
drwxr-xr-x    2 root     root           512 Jan  1 00:42 .
drwxrwxrwt   19 root     root           420 Jan  1 00:40 ..
-rwxr-xr-x    1 root     root      30578008 Jan  1 00:42 image.ub
root@pz-7z015:~#
```

3.4 Programming BOOT.BIN to the QSPI device

The final BOOT.BIN will be programmed at offset 0x00000000 in the QSPI device, as described below.

1. In the SDK menu, select **Xilinx Tools => Program Flash**

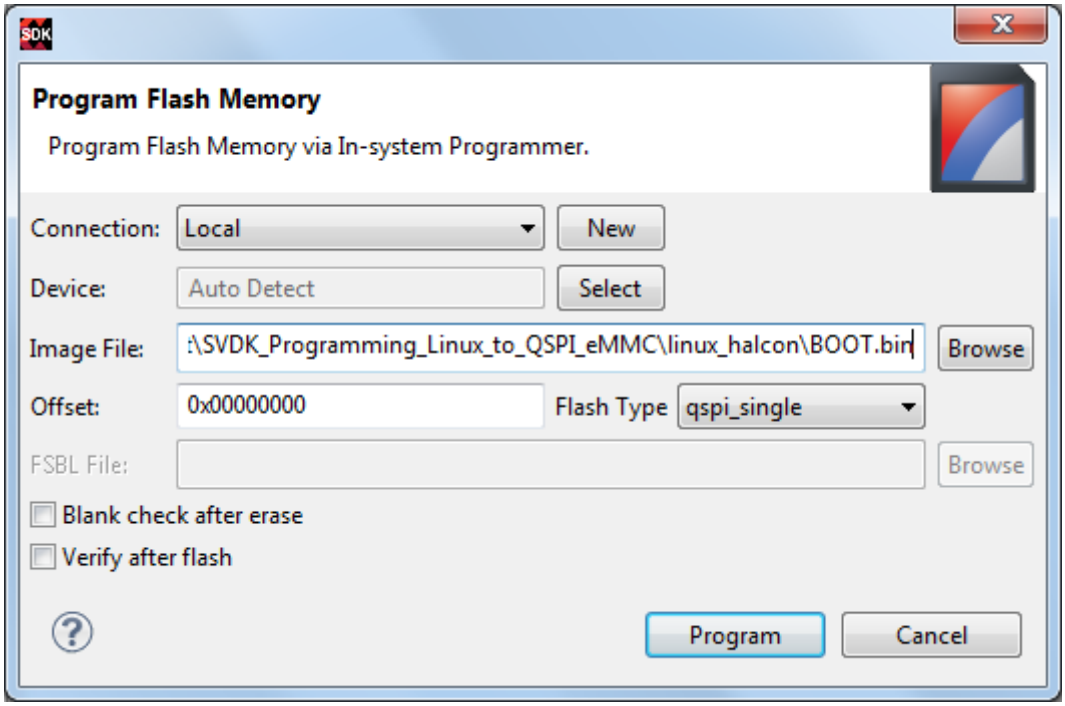

Figure 11 – SDK – Program BOOT.bin to QSPI at offset 0x00000000

2. Program offset 0x00000000 of the QSPI device with the following boot file

   ..\SVDK_Programming_Linux_to_QSPI_eMMC\linux_halcon\BOOT.bin

NOTE*: the BOOT.BIN was renamed BOOT.bin, since the wizard only accepts (case sensitive) .mcs or .bin files*

3. Click **Program**

3.5 Booting the final version of Linux

At this point, we have successfully programmed the final version of Linux to the QSPI device (BOOT.BIN) and eMMC device (image.ub).

1. Cycle power on the SVDK. The terminal output displays feedback from the linux console.

```
U-Boot 2014.01 (Jun 23 2015 - 11:14:42)

Memory: ECC disabled
DRAM:  1 GiB
MMC:   zynq_sdhci: 0
SF: Detected S25FL128S_64K with page size 256 Bytes, erase size
64 KiB, total 16 MiB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   Gem.e000b000
U-BOOT for gvrd

Gem.e000b000 Waiting for PHY auto negotiation to complete.....
done
BOOTP broadcast 1
DHCP client bound to address 192.168.0.115
Hit any key to stop autoboot:  0
Device: zynq_sdhci
Manufacturer ID: fe
OEM: 14e
Name: MMC04
Tran Speed: 52000000
Rd Block Len: 512
MMC version 4.41
High Capacity: Yes
Capacity: 3.5 GiB
Bus Width: 4-bit
```

```
reading image.ub
30578008 bytes read in 1806 ms (16.1 MiB/s)
## Loading kernel from FIT Image at 04000000 ...
   Using 'conf@1' configuration
   Trying 'kernel@1' kernel subimage
     Description:  PetaLinux Kernel
     Type:         Kernel Image
     Compression:  gzip compressed
     Data Start:   0x040000f0
     Data Size:    30561512 Bytes = 29.1 MiB
     Architecture: ARM
     OS:           Linux
     Load Address: 0x00008000
     Entry Point:  0x00008000
     Hash algo:    crc32
     Hash value:   fe10e335
   Verifying Hash Integrity ... crc32+ OK
## Loading fdt from FIT Image at 04000000 ...
   Using 'conf@1' configuration
   Trying 'fdt@1' fdt subimage
     Description:  Flattened Device Tree blob
     Type:         Flat Device Tree
     Compression:  uncompressed
     Data Start:   0x05d256bc
     Data Size:    15200 Bytes = 14.8 KiB
     Architecture: ARM
     Hash algo:    crc32
     Hash value:   26cfd7a0
   Verifying Hash Integrity ... crc32+ OK
   Booting using the fdt blob at 0x5d256bc
   Uncompressing Kernel Image ... OK
   Loading Device Tree to 07ff9000, end 07fffb5f ... OK

Starting kernel ...

Booting Linux on physical CPU 0x0
Linux version 3.14.2-xilinx (user@rw-ubuntu-12) (gcc version
4.8.1 (Sourcery CodeBench Lite 2013.11-53) ) #34 SMP PREEMPT Wed
Jun 24 08:32:17 CEST 2015
CPU: ARMv7 Processor [413fc090] revision 0 (ARMv7), cr=18c5387d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing
instruction cache
Machine model: gvrd
bootconsole [earlycon0] enabled
Memory policy: Data cache writealloc
PERCPU: Embedded 8 pages/cpu @ef0d5000 s10752 r8192 d13824 u32768
Built 1 zonelists in Zone order, mobility grouping on. Total
pages: 227856
Kernel command line: console=ttyPS0,115200 earlyprintk
PID hash table entries: 4096 (order: 2, 16384 bytes)
Dentry cache hash table entries: 131072 (order: 7, 524288 bytes)
Inode-cache hash table entries: 65536 (order: 6, 262144 bytes)
Memory: 870472K/917504K available (4843K kernel code, 310K
rwdata, 1776K rodata, 26598K init, 5337K bss, 47032K reserved,
139264K highmem)
```

```
Virtual kernel memory layout:
    vector  : 0xffff0000 - 0xffff1000   (   4 kB)
    fixmap  : 0xfff00000 - 0xfffe0000   ( 896 kB)
    vmalloc : 0xf0000000 - 0xff000000   ( 240 MB)
    lowmem  : 0xc0000000 - 0xef800000   ( 760 MB)
    pkmap   : 0xbfe00000 - 0xc0000000   (   2 MB)
    modules : 0xbf000000 - 0xbfe00000   (  14 MB)
      .text : 0xc0008000 - 0xc067f088   (6621 kB)
      .init : 0xc0680000 - 0xc2079a00   (26599 kB)
      .data : 0xc207a000 - 0xc20c7918   ( 311 kB)
       .bss : 0xc20c7924 - 0xc25fde48   (5338 kB)
Preemptible hierarchical RCU implementation.
        RCU lockdep checking is enabled.
        Dump stacks of tasks blocking RCU-preempt GP.
        RCU restricting CPUs from NR_CPUS=4 to nr_cpu_ids=2.
RCU: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=2
NR_IRQS:16 nr_irqs:16 16
ps7-slcr mapped to f0004000
zynq_clock_init: clkc starts at f0004100
Zynq clock init
sched_clock: 64 bits at 333MHz, resolution 3ns, wraps every
3298534883328ns
ps7-ttc #0 at f0006000, irq=43
Console: colour dummy device 80x30
Lock dependency validator: Copyright (c) 2006 Red Hat, Inc., Ingo
Molnar
... MAX_LOCKDEP_SUBCLASSES:  8
... MAX_LOCK_DEPTH:          48
... MAX_LOCKDEP_KEYS:        8191
... CLASSHASH_SIZE:          4096
... MAX_LOCKDEP_ENTRIES:     16384
... MAX_LOCKDEP_CHAINS:      32768
... CHAINHASH_SIZE:          16384
 memory used by lock dependency info: 3695 kB
 per task-struct memory footprint: 1152 bytes
Calibrating delay loop... 1325.46 BogoMIPS (lpj=6627328)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 2048 (order: 1, 8192 bytes)
Mountpoint-cache hash table entries: 2048 (order: 1, 8192 bytes)
CPU: Testing write buffer coherency: ok
missing device node for CPU 0
missing device node for CPU 1
CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
Setting up static identity map for 0x498908 - 0x498960
L310 cache controller enabled
l2x0: 8 ways, CACHE_ID 0x410000c8, AUX_CTRL 0x72760000, Cache
size: 512 kB
CPU1: Booted secondary processor
CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
Brought up 2 CPUs
SMP: Total of 2 processors activated.
CPU: All CPU(s) started in SVC mode.
devtmpfs: initialized
VFP support v0.3: implementor 41 architecture 3 part 30 variant 9
rev 4
```

```
regulator-dummy: no parameters
NET: Registered protocol family 16
DMA: preallocated 256 KiB pool for atomic coherent allocations
cpuidle: using governor ladder
cpuidle: using governor menu
syscon f8000000.ps7-slcr: regmap [mem 0xf8000000-0xf8000fff]
registered
hw-breakpoint: found 5 (+1 reserved) breakpoint and 1 watchpoint
registers.
hw-breakpoint: maximum watchpoint size is 4 bytes.
zynq-ocm f800c000.ps7-ocmc: ZYNQ OCM pool: 256 KiB @ 0xf0080000
bio: create slab <bio-0> at 0
vgaarb: loaded
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
media: Linux media interface: v0.10
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
DMA-API: preallocated 4096 debug entries
DMA-API: debugging enabled by kernel config
Switched to clocksource arm_global_timer
NET: Registered protocol family 2
TCP established hash table entries: 8192 (order: 3, 32768 bytes)
TCP bind hash table entries: 8192 (order: 6, 294912 bytes)
TCP: Hash tables configured (established 8192 bind 8192)
TCP: reno registered
UDP hash table entries: 512 (order: 3, 40960 bytes)
UDP-Lite hash table entries: 512 (order: 3, 40960 bytes)
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.
hw perfevents: enabled with ARMv7 Cortex-A9 PMU driver, 7
counters available
futex hash table entries: 512 (order: 3, 32768 bytes)
bounce pool size: 64 pages
jffs2: version 2.2. (NAND) (SUMMARY) © 2001-2006 Red Hat, Inc.
msgmni has been set to 1428
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
dma-pl330 f8003000.ps7-dma: Loaded driver for PL330 DMAC-267056
dma-pl330 f8003000.ps7-dma: DBUFF-128x8bytes Num_Chans-8
Num_Peri-4 Num_Events-16
e0001000.serial: ttyPS0 at MMIO 0xe0001000 (irq = 82, base_baud =
6249999) is a xuartps
console [ttyPS0] enabled
console [ttyPS0] enabled
```

```
bootconsole [earlycon0] disabled
bootconsole [earlycon0] disabled
xdevcfg f8007000.ps7-dev-cfg: ioremap 0xf8007000 to f0064000
brd: module loaded
loop: module loaded
cdns-spi e0006000.ps7-spi: pclk clock not found.
cdns-spi: probe of e0006000.ps7-spi failed with error -2
m25p80 spi32766.0: found s25fl128s1, expected n25q128
m25p80 spi32766.0: s25fl128s1 (16384 Kbytes)
4 ofpart partitions found on MTD device spi32766.0
Creating 4 MTD partitions on "spi32766.0":
0x000000000000-0x000000500000 : "boot"
0x000000500000-0x000000520000 : "bootenv"
0x000000520000-0x000000fa0000 : "kernel"
0x000000fa0000-0x000001000000 : "spare"
e1000e: Intel(R) PRO/1000 Network Driver - 2.3.2-k
e1000e: Copyright(c) 1999 - 2013 Intel Corporation.
libphy: XEMACPS mii bus: probed
xemacps e000b000.ps7-ethernet: pdev->id -1, baseaddr 0xe000b000,
irq 54
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
ehci-pci: EHCI PCI platform driver
usbcore: registered new interface driver usb-storage
mousedev: PS/2 mouse device common for all mice
i2c /dev entries driver
cdns-i2c e0004000.ps7-i2c: 400 kHz mmio e0004000 irq 57
vivi-000: V4L2 device registered as video0
Video Technology Magazine Virtual Video Capture Board ver 0.8.1
successfully loaded.
zynq-edac f8006000.ps7-ddrc: ecc not enabled
cpufreq_cpu0: failed to find cpu0 node
cpufreq-cpu0: probe of cpufreq-cpu0.0 failed with error -2
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: no vqmmc regulator found
mmc0: no vmmc regulator found
mmc0: SDHCI controller on e0101000.ps7-sdio [e0101000.ps7-sdio]
using ADMA
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
TCP: cubic registered
NET: Registered protocol family 17
Registering SWP/SWPB emulation handler
regulator-dummy: disabling
/home/user/petalinux-v2014.2-final/components/linux-kernel/xlnx-
3.14/drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
mmc0: BKOPS_EN bit is not set
mmcblk0: mmc0:0001 MMC04G 3.52 GiB ss 0001
mmcblk0boot0: mmc0:0001 MMC04G partition 1 16.0 MiB
mmcblk0boot1: mmc0:0001 MMC04G partition 2 16.0 MiB
mmcblk0rpmb: mmc0:0001 MMC04G partition 3 128 KiB
 mmcblk0: p1
Freeing unused kernel memory: 26596K (c0680000 - c2079000)
```

```
 mmcblk0boot1: unknown partition table
 mmcblk0boot0: unknown partition table
INIT: version 2.88 booting
Starting Bootlog daemon: bootlogd.
Creating /dev/flash/* device nodes
random: dd urandom read with 26 bits of entropy available
starting Busybox inet Daemon: inetd... done.
Starting uWeb server:
NET: Registered protocol family 10
update-rc.d: /etc/init.d/run-postinsts exists during rc.d purge
(continuing)
 Removing any system startup links for run-postinsts ...
  /etc/rcS.d/S99run-postinsts
INIT: Entering runlevel: 5
Configuring network interfaces... udhcpc (v1.22.1) started
Sending discover...
Sending discover...
xemacps e000b000.ps7-ethernet: Set clk to 124999998 Hz
xemacps e000b000.ps7-ethernet: link up (1000/FULL)
Sending discover...
Sending select for 192.168.0.115...
Lease of 192.168.0.115 obtained, lease time 86400
/etc/udhcpc.d/50default: Adding DNS 192.168.0.1
done.
Stopping Bootlog daemon: bootlogd.

Built with PetaLinux v2014.2 (Yocto 1.6) gvrd /dev/ttyPS0
gvrd login: root
Password: root
login[799]: root login on 'ttyPS0'
root@gvrd:~#
```

2. Enter the login credentials (login = **root**, password = **root**).