ZedBoard Linux-FreeRTOS AMP Board Bringup Guide.

Posted on February 26, 2014 by d9#idv-tech#com — 11 Comments ↓

One of the many nice features of Xilinx Zynq is ability to run it in Asymmetric MultiProcessing or AMP configuration. Xilinx released version v2013.10 of a UG980(Petalinux Board Bringup) and UG978(Zynq Linux-FreeRTOS AMP) guides for Xilinx ZC702 board. Today I will follow those guides to build Linux-FreeRTOS solution for ZedBoard.
I already have Vivado and Xilinx SDK packages installed along with required, but initially missing packages on my Ubuntu 13.10 64 bit Virtual Machine.

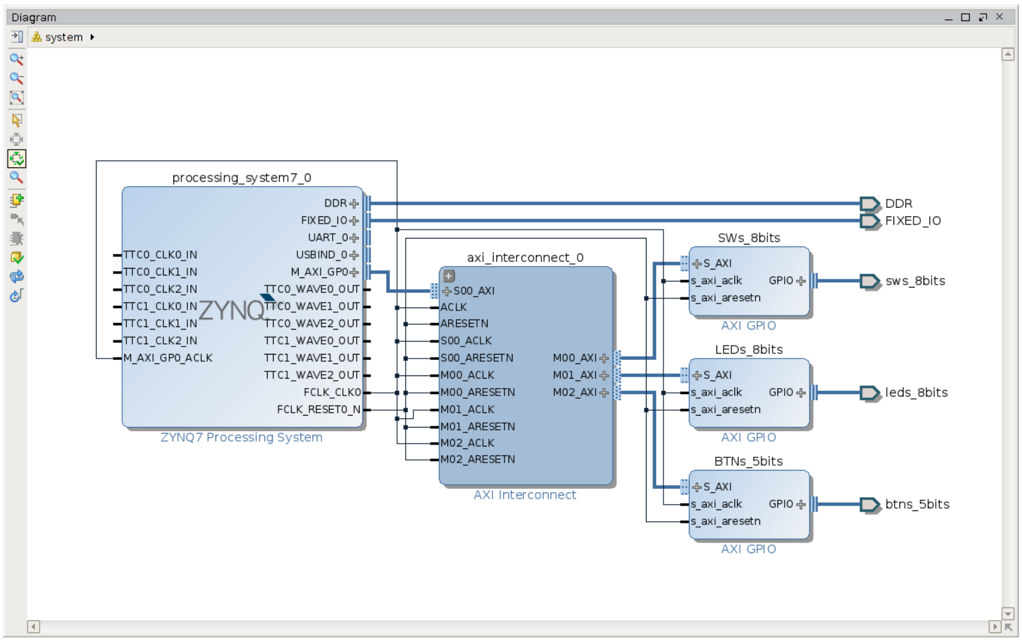
1. First of all we will need so called Zynq 'Hardware Platform' which will satisfy both Linux and FreeRTOS minimum requirements.
   Linux requires one UART and at least one storage peripheral, for example SD Card. And FreeRTOS also requires one UART and also a Timer.
   So, I will modify ZedBoard CTT hardware design I created using ZedBoard_CTT_v2013_2_130807 tutorial. That 'Hardware Platform', in addition to base system, consist of 8 switches, 8 led's and 5 push buttons and I will activate 1 more UART and 1 more timer for FreeRTOS.

   Open Vivado ZedBoard CTT project or create it from scratch using ZedBoard CTT tutorial. I called my project 'ZedBoard-AMP' and made next 2 modifications to CTT design:

   - Zynq7 Processing System: MIO Configuration: I/O Peripherals: Enable UART0 and set its IO as 'EMIO'.

- Zynq7 Processing System: MIO Configuration: Application Processor Unit: Enable Timer1 and set its IO as 'EMIO'.

As a result I got the system with the such 'Block Diagram':



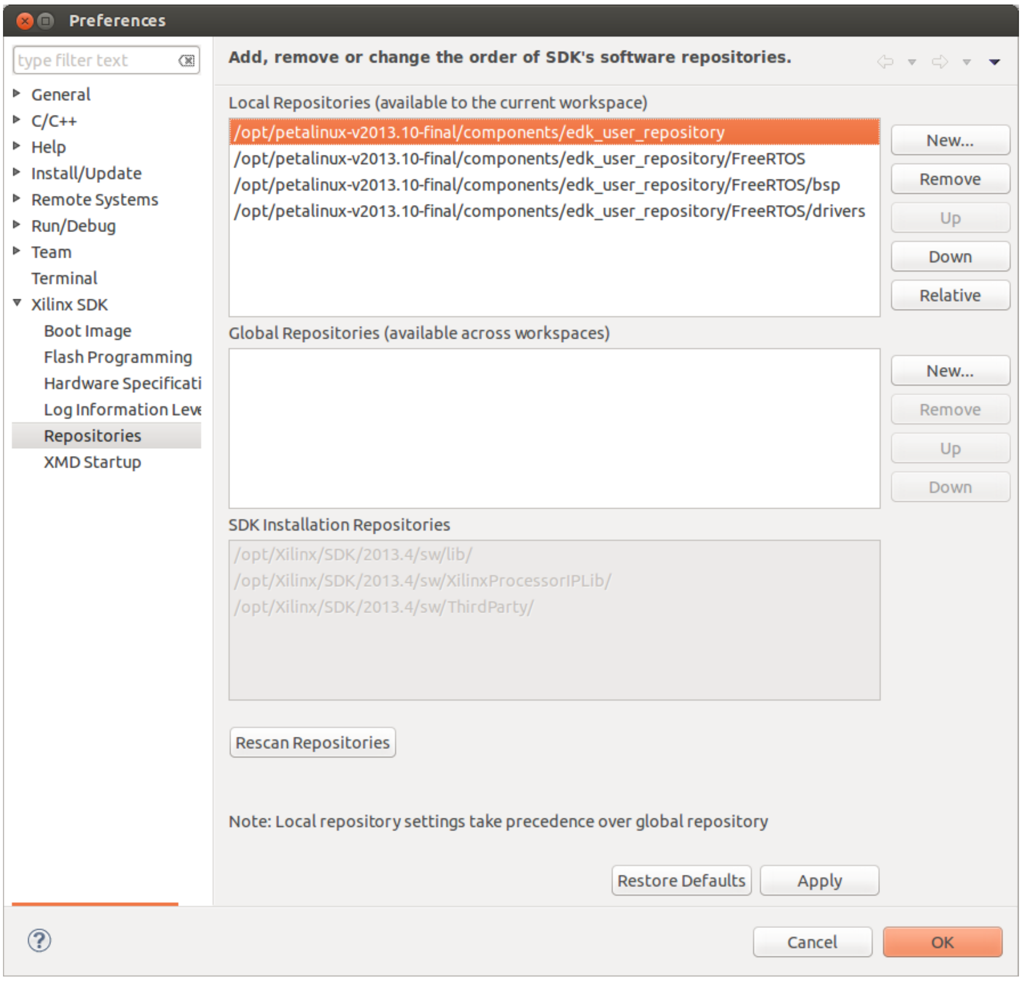
2. Run Synthesis, Implementation, generate new BitStream and Export new 'Hardware' to SDK. I prefer not to lauch SDK right away for a good reason - you have to source Xilinx SDK settings and/or PetaLinux settings before running XSDK. After export finished we can close Vivado.
3. Lets set required settings and run Xilinx SDK (XSDK). Set 'Eclipse' workspace to our ~/Projects/ directory.

```
mkdir Projects
cd Projects
source /opt/Xilinx/Vivado/2013.4/settings64.sh
xsdk
```

4. We need to add PetaLinux and FreeRTOS repositories to XSDK. In XSDK -> Xilinx Tools -> 'Xilinx SDK' -> 'Repositories' -> add 'Local Repositories' from your PetaLinux 13.10 components folder. In my case PetaLinux installed into '/opt/petalinux-v2013.10-final/'. So added repo's are:
   - /opt/petalinux-v2013.10-final/components/edk_user_repository
   - /opt/petalinux-v2013.10-final/components/edk_user_repository/FreeRTOS
   - /opt/petalinux-v2013.10-final/components/edk_user_repository

/FreeRTOS/bsp

- /opt/petalinux-v2013.10-final/components/edk_user_repository/FreeRTOS/drivers



Preferences

Add, remove or change the order of SDK's software repositories.

Local Repositories (available to the current workspace)

/opt/petalinux-v2013.10-final/components/edk_user_repository
/opt/petalinux-v2013.10-final/components/edk_user_repository/FreeRTOS
/opt/petalinux-v2013.10-final/components/edk_user_repository/FreeRTOS/bsp
/opt/petalinux-v2013.10-final/components/edk_user_repository/FreeRTOS/drivers

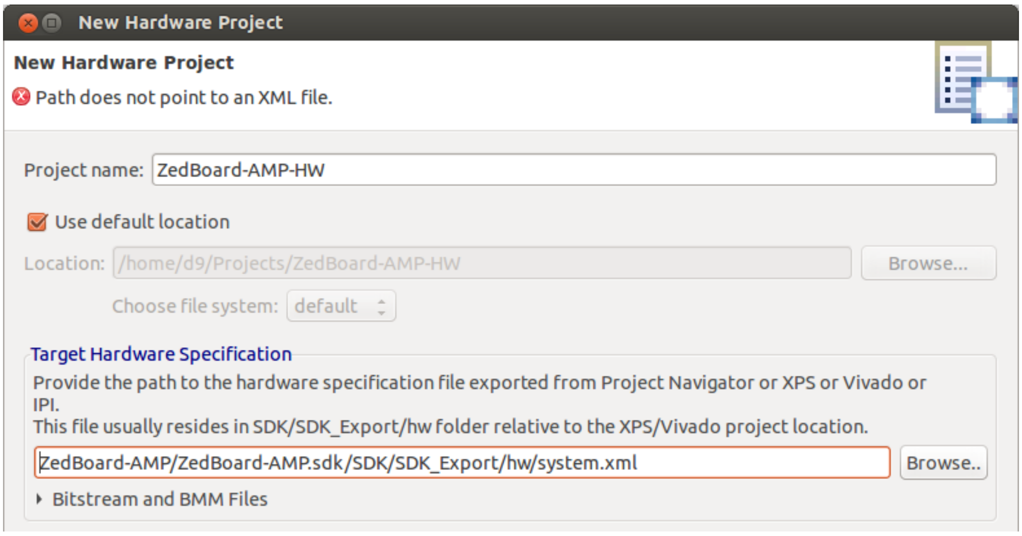
Global Repositories (available across workspaces)

SDK Installation Repositories

/opt/Xilinx/SDK/2013.4/sw/lib/
/opt/Xilinx/SDK/2013.4/sw/XilinxProcessorIPLib/
/opt/Xilinx/SDK/2013.4/sw/ThirdParty/

Note: Local repository settings take precedence over global repository

5. Create 'Hardware Platform Specification' project using 'Hardware Platfrom' exported from our ZedBoard-AMP Vivado project. I named it 'ZedBoard-AMP-HW'.



New Hardware Project

New Hardware Project

Path does not point to an XML file.

Project name: ZedBoard-AMP-HW

☑ Use default location

Location: /home/d9/Projects/ZedBoard-AMP-HW

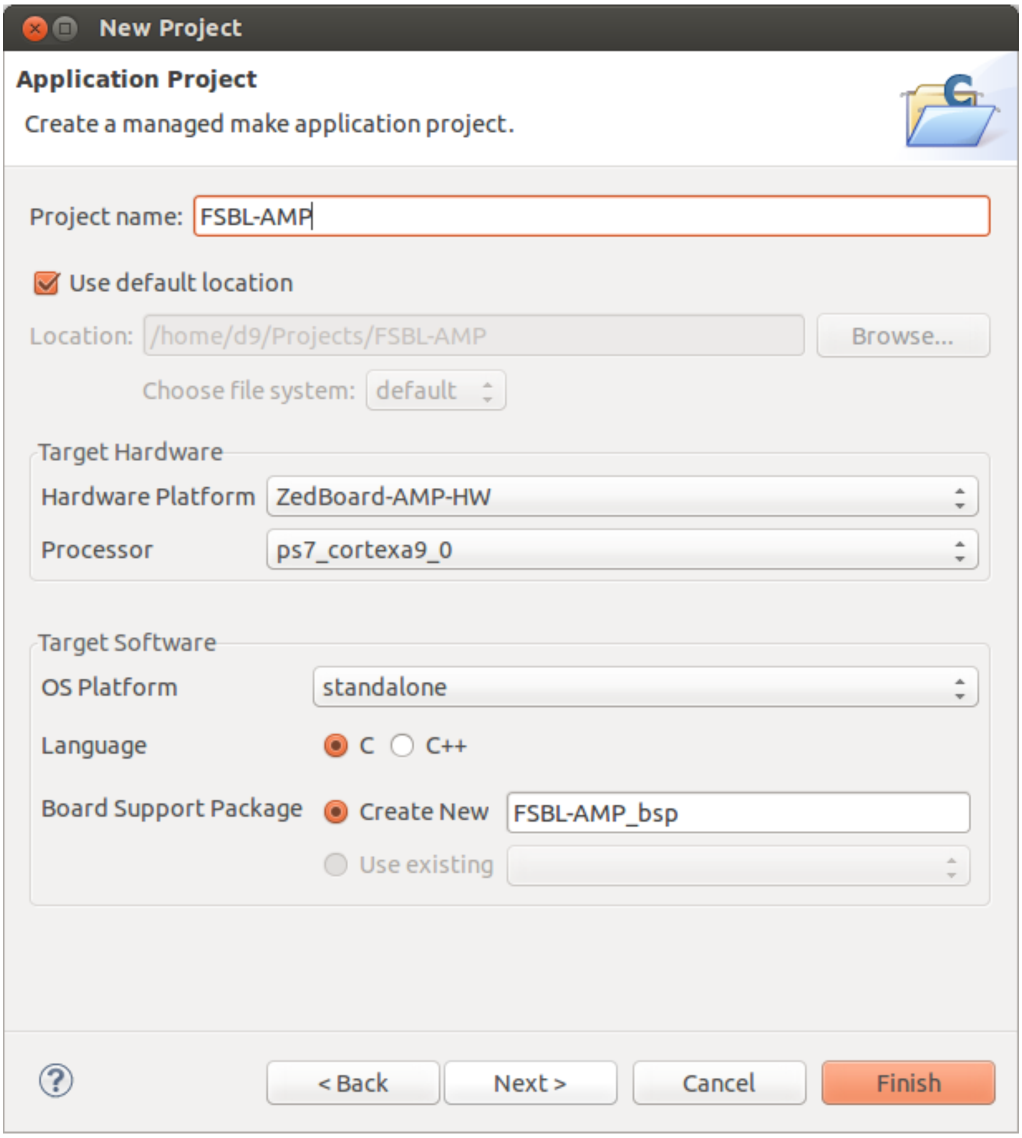
Choose file system: default

Target Hardware Specification

Provide the path to the hardware specification file exported from Project Navigator or XPS or Vivado or IPI.
This file usually resides in SDK/SDK_Export/hw folder relative to the XPS/Vivado project location.

ZedBoard-AMP/ZedBoard-AMP.sdk/SDK/SDK_Export/hw/system.xml

6. Create FSBL for AMP configuration using File -> New Project -> Application Project. I named it FSBL-AMP and selected 'ZedBoard-AMP-HW' as Hardware Platform, 'ps7_cortex9_0' as a Processor, 'standalone 'OS Platfrom ', after clicking 'Next' select 'Zynq FSBL' template and finish project creation. Compile both 'FSBL-AMP_bsp' and 'FSBL-AMP' projects if it didnt autobuilt.



Application Project

Create a managed make application project.

Project name: FSBL-AMP

☑ Use default location

Location: /home/d9/Projects/FSBL-AMP

Choose file system: default

Target Hardware

Hardware Platform: ZedBoard-AMP-HW

Processor: ps7_cortexa9_0

Target Software

OS Platform: standalone

Language: ● C ○ C++

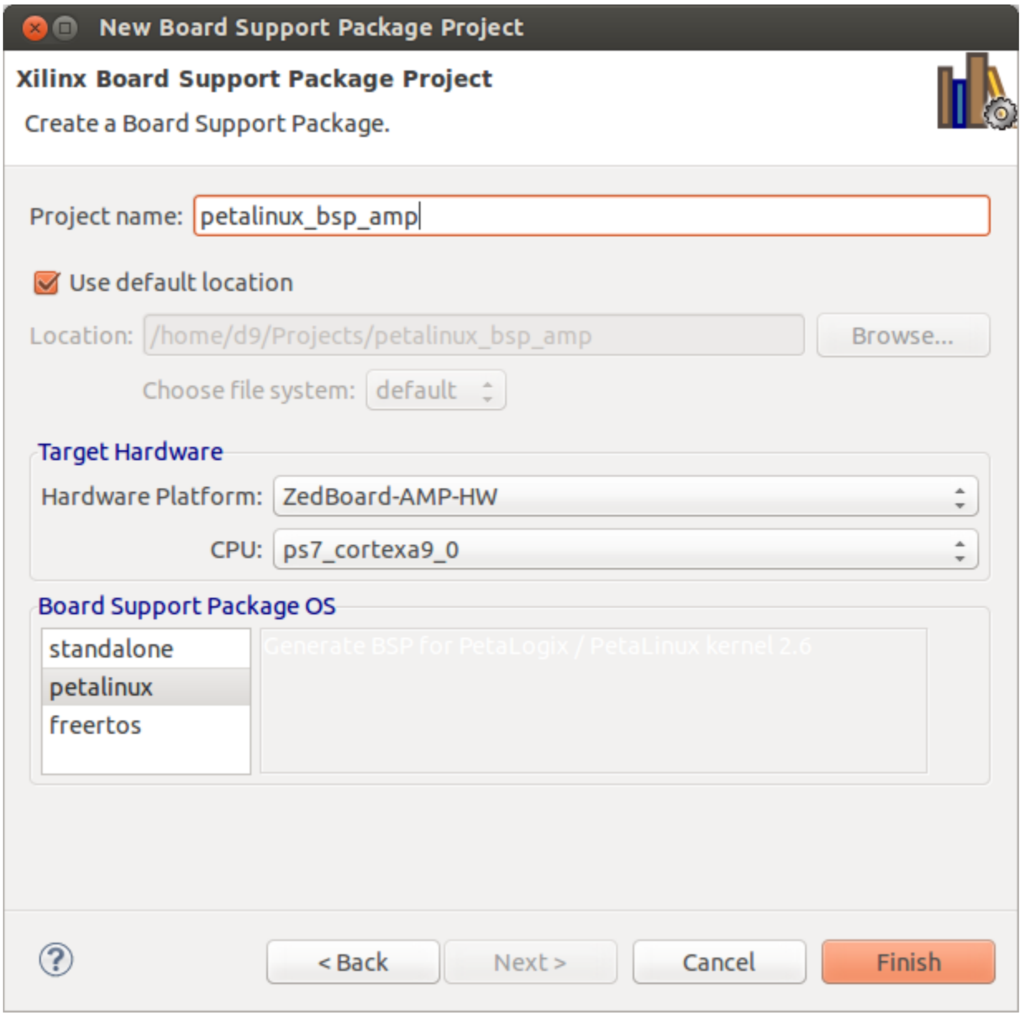
Board Support Package: ● Create New FSBL-AMP_bsp

○ Use existing

7. Now we will need to build a few PetaLinux projects, but before we can do this, we have to quit XSDK and source some PetaLinux settings. So, close XSDK, apply settings in next order and launch XSDK again:

```
export CROSS_COMPILE=arm-xilinx-linux-gnueabi-
source /opt/Xilinx/Vivado/2013.4/settings64.sh
source /opt/petalinux-v2013.10-final/settings.sh
xsdk
```

8. Create PetaLinux BSP project. In SDK select File -> New -> Project -> 'Board Support Package'. Name it 'petalinux_bsp_amp'. Select 'ZedBoard-AMP-HW' as Hardware Platform, CPU 'ps7_cortexa9_0'. Board Support Package OS 'petalinux'. Then Finish.



New Board Support Package Project

Xilinx Board Support Package Project

Create a Board Support Package.

Project name: petalinux_bsp_amp

☑ Use default location

Location: /home/d9/Projects/petalinux_bsp_amp

Choose file system: default

Target Hardware

Hardware Platform: ZedBoard-AMP-HW

CPU: ps7_cortexa9_0

Board Support Package OS

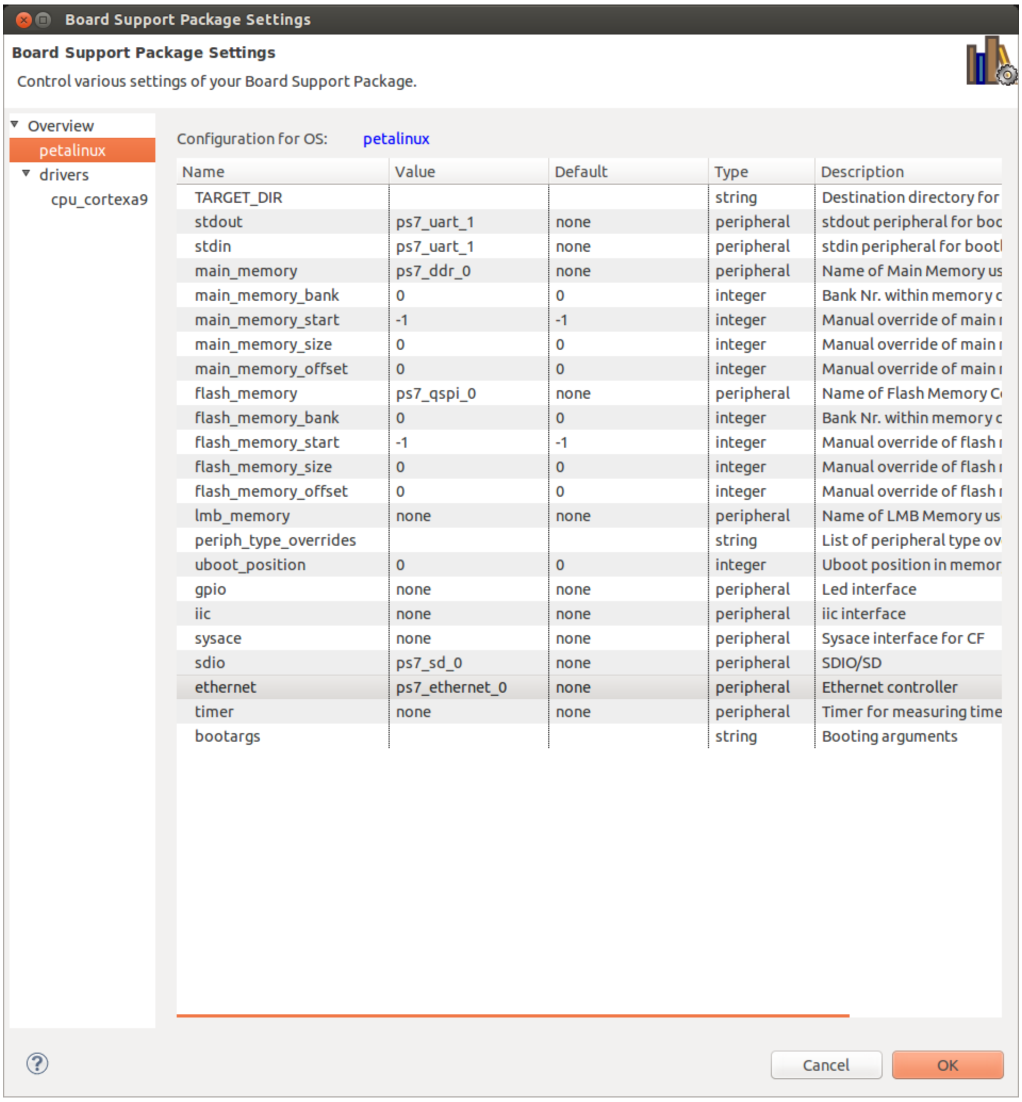
standalone
petalinux
freertos

Generate BSP for PetaLogix / PetaLinux kernel 2.6

XSDK should automatically open 'Board Support Package Settings' page. Go to 'Overview' -> 'petalinux' menu of that page and set:

- 'ps7_uart_1' for stdout and stdin.
- 'ps7_ddr_0' for main memory.
- 'ps7_qspi' for flash memory.

- 'ps7_sd_0' for sdio.
- 'ps7_ethernet' for ethernet.



Board Support Package Settings

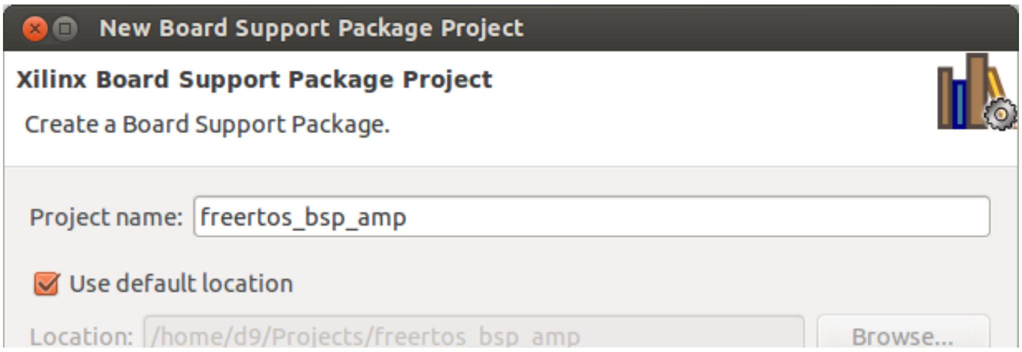
Control various settings of your Board Support Package.

Configuration for OS: petalinux

Name	Value	Default	Type	Description
TARGET_DIR			string	Destination directory for
stdout	ps7_uart_1	none	peripheral	stdout peripheral for boo
stdin	ps7_uart_1	none	peripheral	stdin peripheral for bootl
main_memory	ps7_ddr_0	none	peripheral	Name of Main Memory us
main_memory_bank	0	0	integer	Bank Nr. within memory c
main_memory_start	-1	-1	integer	Manual override of main r
main_memory_size	0	0	integer	Manual override of main r
main_memory_offset	0	0	integer	Manual override of main r
flash_memory	ps7_qspi_0	none	peripheral	Name of Flash Memory C
flash_memory_bank	0	0	integer	Bank Nr. within memory c
flash_memory_start	-1	-1	integer	Manual override of flash r
flash_memory_size	0	0	integer	Manual override of flash r
flash_memory_offset	0	0	integer	Manual override of flash r
lmb_memory	none	none	peripheral	Name of LMB Memory us
periph_type_overrides			string	List of peripheral type ov
uboot_position	0	0	integer	Uboot position in memor
gpio	none	none	peripheral	Led interface
iic	none	none	peripheral	iic interface
sysace	none	none	peripheral	Sysace interface for CF
sdio	ps7_sd_0	none	peripheral	SDIO/SD
ethernet	ps7_ethernet_0	none	peripheral	Ethernet controller
timer	none	none	peripheral	Timer for measuring time
bootargs			string	Booting arguments

Build this project.

9. Create FreeRTOS BSP project. In SDK select File -> New -> Project -> 'Board Support Package'. Name it 'freertos_bsp_amp'. Select 'ZedBoard-AMP-HW' as Hardware Platform, CPU 'ps7_cortexa9_1'. Board Support Package OS 'freertos'. Then Finish.



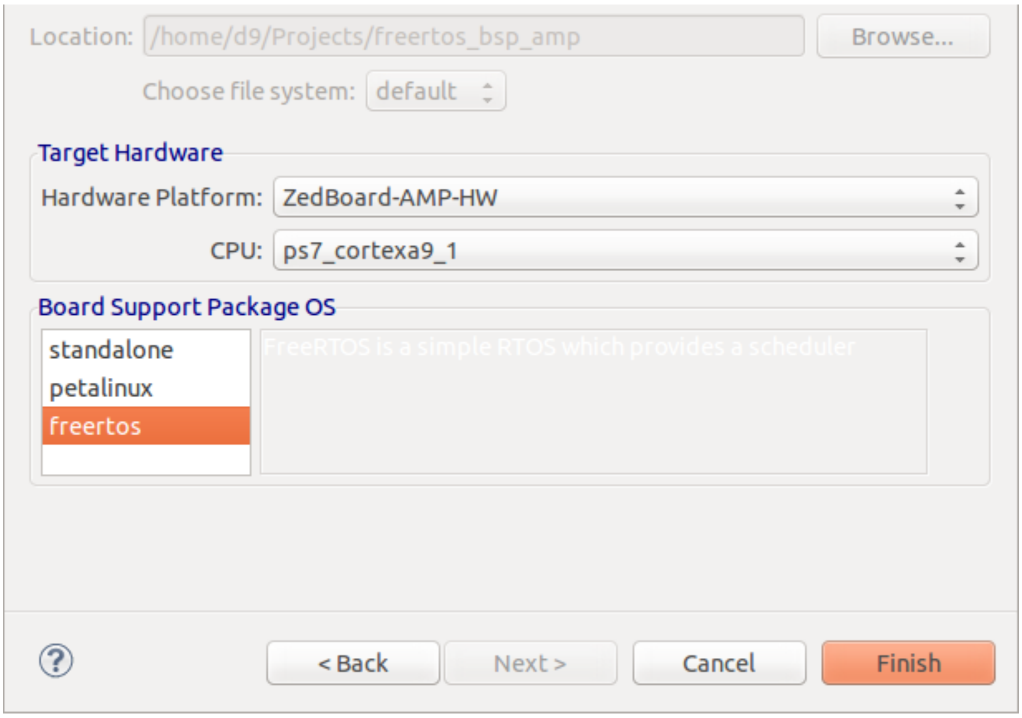
Xilinx Board Support Package Project

Create a Board Support Package.

Project name: freertos_bsp_amp

☑ Use default location

Location: /home/d9/Projects/freertos_bsp_amp



Location: /home/d9/Projects/freertos_bsp_amp

Choose file system: default

Target Hardware

Hardware Platform: ZedBoard-AMP-HW

CPU: ps7_cortexa9_1

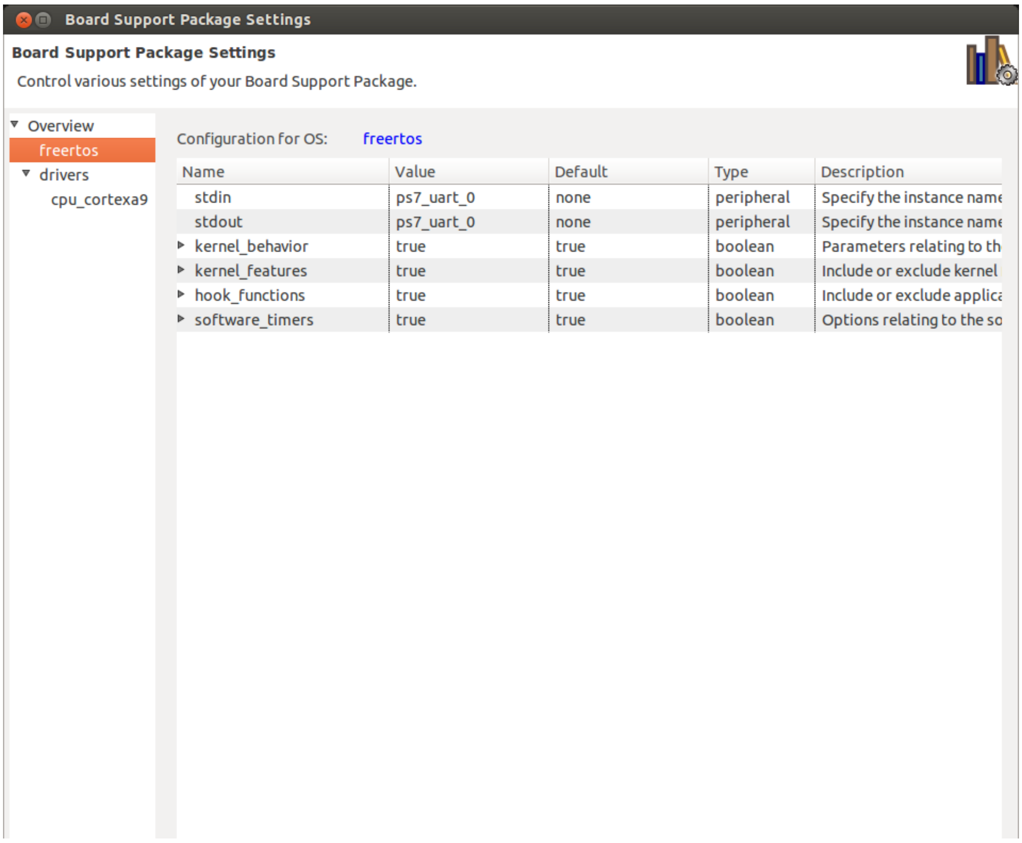
Board Support Package OS

- standalone
- petalinux
- freertos

FreeRTOS is a simple RTOS which provides a scheduler

XSDK will automatically open 'Board Support Package Settings' page now for FreeRTOS project. Then:

- 'Overview' -> 'freertos' -> Set 'ps7_uart_0' for both stdin and stdout.
- 'drivers' -> 'cpu_cortexa9' -> Set 'extra_compiler_flags' to '-g -DUSE_AMP=1'.



Board Support Package Settings

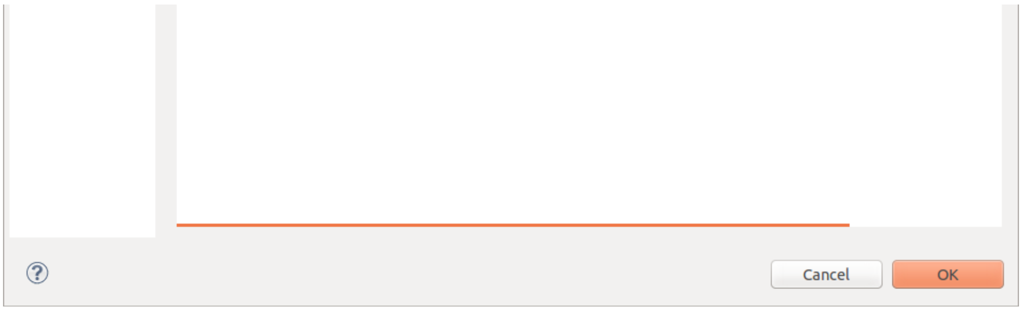
Board Support Package Settings

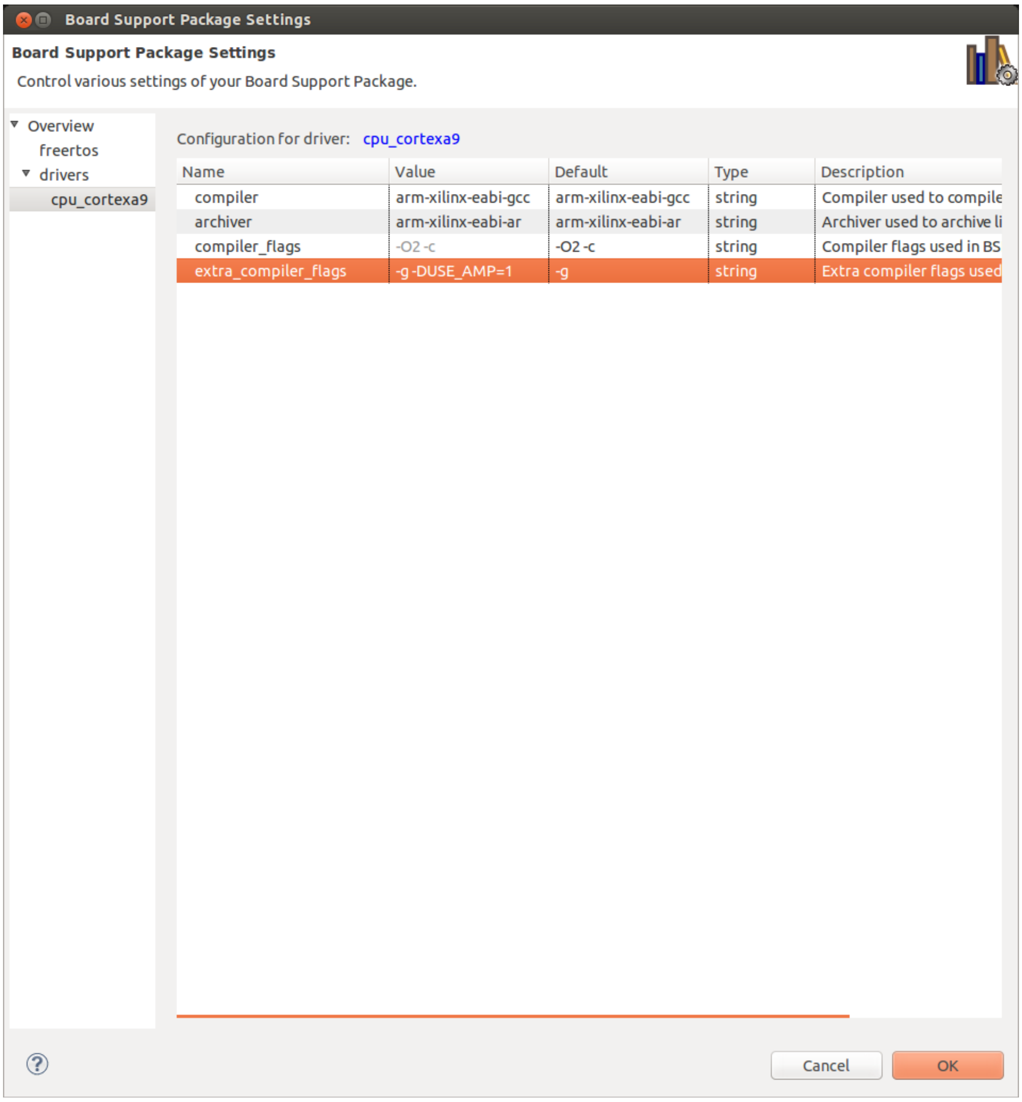
Control various settings of your Board Support Package.

- Overview
  - freertos
  - drivers
    - cpu_cortexa9

Configuration for OS: freertos

Name	Value	Default	Type	Description
stdin	ps7_uart_0	none	peripheral	Specify the instance name
stdout	ps7_uart_0	none	peripheral	Specify the instance name
kernel_behavior	true	true	boolean	Parameters relating to th
kernel_features	true	true	boolean	Include or exclude kernel
hook_functions	true	true	boolean	Include or exclude applica
software_timers	true	true	boolean	Options relating to the so





Board Support Package Settings

Control various settings of your Board Support Package.

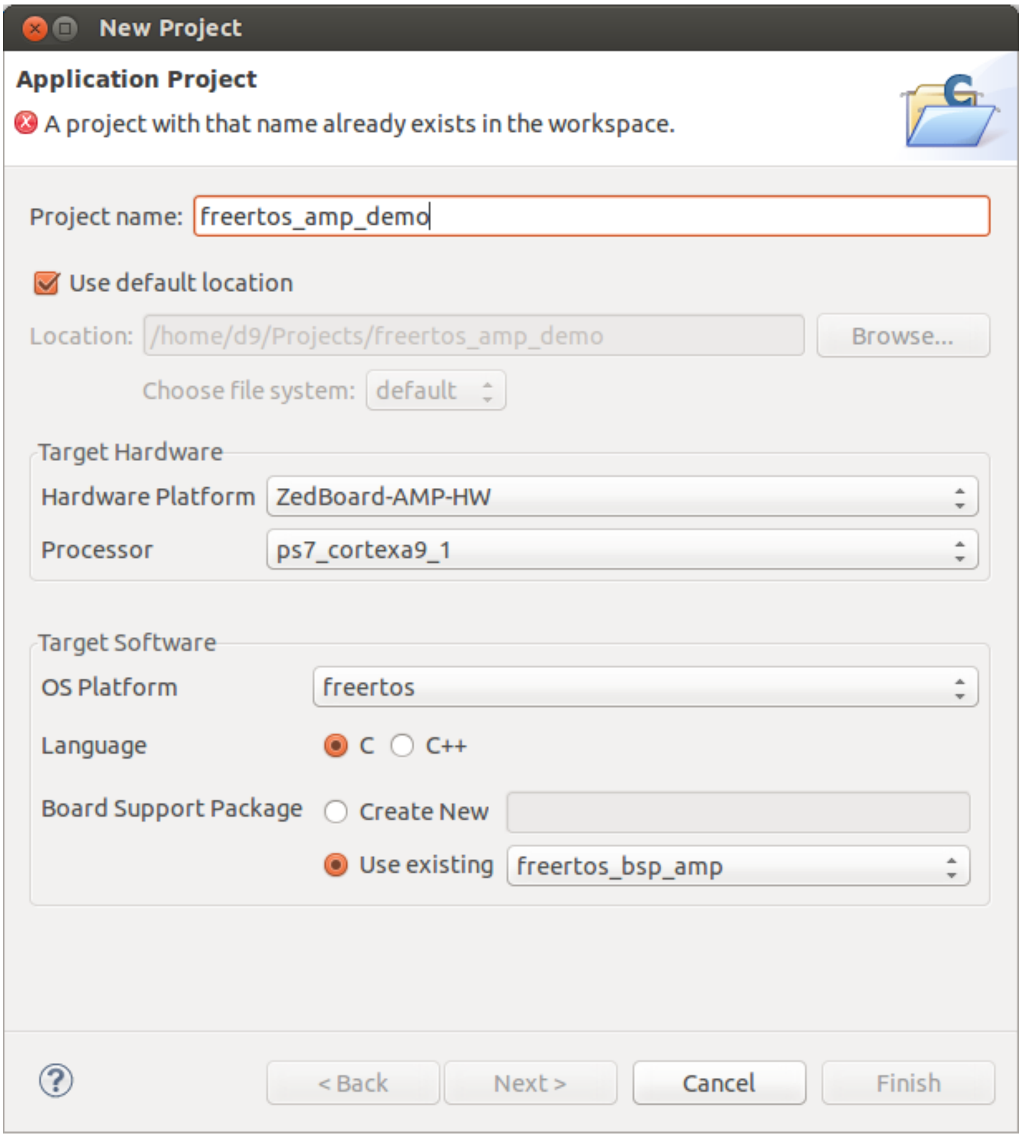
Configuration for driver: cpu_cortexa9

Name	Value	Default	Type	Description
compiler	arm-xilinx-eabi-gcc	arm-xilinx-eabi-gcc	string	Compiler used to compile
archiver	arm-xilinx-eabi-ar	arm-xilinx-eabi-ar	string	Archiver used to archive li
compiler_flags	-O2 -c	-O2 -c	string	Compiler flags used in BS
extra_compiler_flags	-g -DUSE_AMP=1	-g	string	Extra compiler flags used

This flag will enable AMP specific features in the FreeRTOS firmware. Hit Ok. And build this project too if it not built automatically.

10. Create FreeRTOS test applications project. In SDK select File -> New -> Project -> 'Application Project'. Name it 'freertos_amp_demo'. Select 'ZedBoard-AMP-HW' as Hardware Platform, Processor 'ps7_cortexa9_1', OS Platfrom 'freertos', for Board Support Package select 'Use existing' -> then our 'freertos_bsp_amp' project. Click 'Next' and select 'FreeRTOS AMP' template.

Click 'Finish'. And build it.



A project with that name already exists in the workspace.

Project name: freertos_amp_demo

☑ Use default location

Location: /home/d9/Projects/freertos_amp_demo

Choose file system: default

Target Hardware

Hardware Platform: ZedBoard-AMP-HW

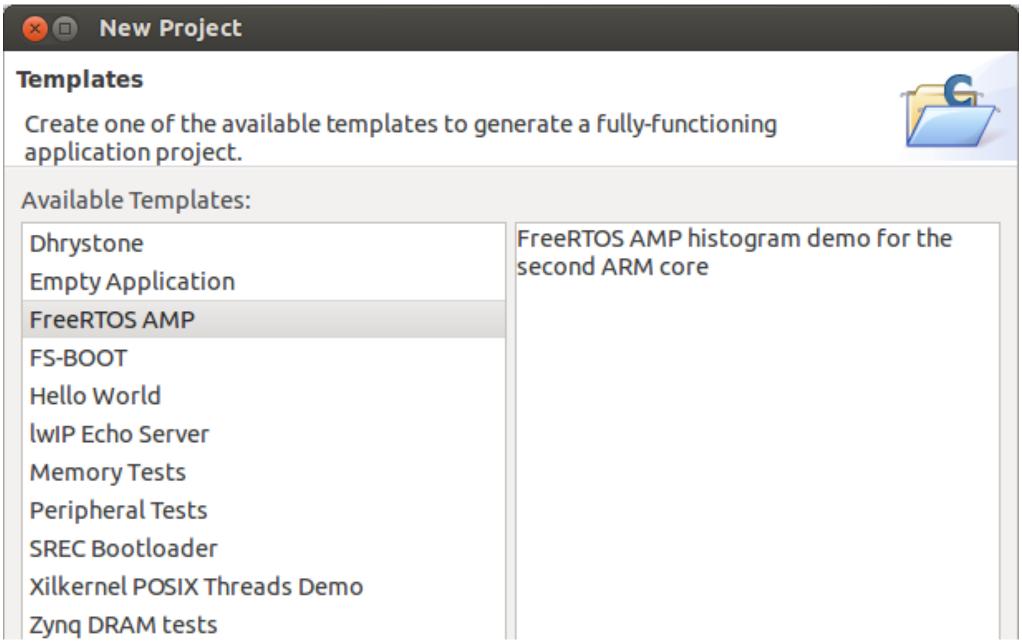
Processor: ps7_cortexa9_1

Target Software

OS Platform: freertos

Language: ● C ○ C++

Board Support Package: ○ Create New ● Use existing freertos_bsp_amp



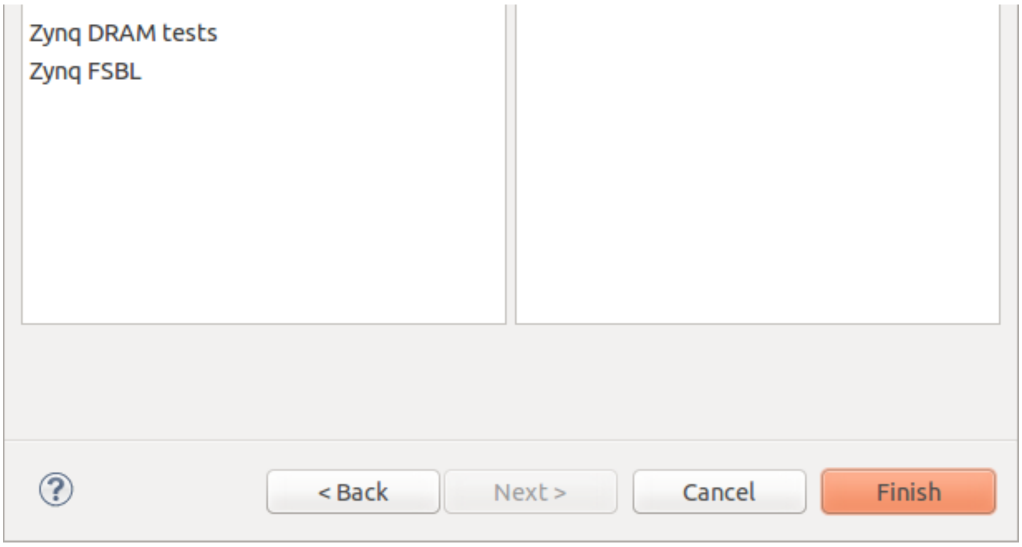
Templates

Create one of the available templates to generate a fully-functioning application project.

Available Templates:

- Dhrystone
- Empty Application
- FreeRTOS AMP
- FS-BOOT
- Hello World
- lwIP Echo Server
- Memory Tests
- Peripheral Tests
- SREC Bootloader
- Xilkernel POSIX Threads Demo
- Zynq DRAM tests

FreeRTOS AMP histogram demo for the second ARM core


Zynq DRAM tests
Zynq FSBL

11. We are done with XSDK for now. Close it.
12. Now we have to create PetaLinux 'project'. And I will name it 'AMP-Demo'.

```
cd ~/Projects
export CROSS_COMPILE=arm-xilinx-linux-gnueabi-
source /opt/Xilinx/Vivado/2013.4/settings64.sh
source /opt/petalinux-v2013.10-final/settings.sh
petalinux-create -t project -n AMP-Demo
```

```
INFO: Create project: AMP-Demo
INFO: New project successfully created in /home/d9
/Projects/AMP-Demo
```

13. Next step is to adjust our PetaLinux configuration to match our unique 'Hardware Platform' we created using Vivado - 'ZedBoard-AMP'. But we have to do it using BSP we created for our 'Hardware Platform' - 'petalinux_bsp_amp'. So, in a ~/Projects directory:

```
cd petalinux_bsp_amp/
petalinux-config --get-hw-description -p ../AMP-Demo/
```

```
INFO: Checking component...
INFO: Getting hardware description...
INFO: Using MSS file /home/d9/Projects/petalinux_bsp_amp/system.mss and XML file /home/d9/Projects/petalinux_bsp_amp/../ZedBoard-AMP-HW/system.xml
INFO: Copy autoconfig for PetaLinux project: /home/d9/Projects/AMP-Demo
INFO: Merging platform settings into kernel configuration
Auto-config file successfully updated for PetaLinux project: /home/d9/Projects/AMP-Demo
[INFO ] generate /home/d9/Projects/AMP-Demo/subsystems/linux/hw-description/system.dts
```

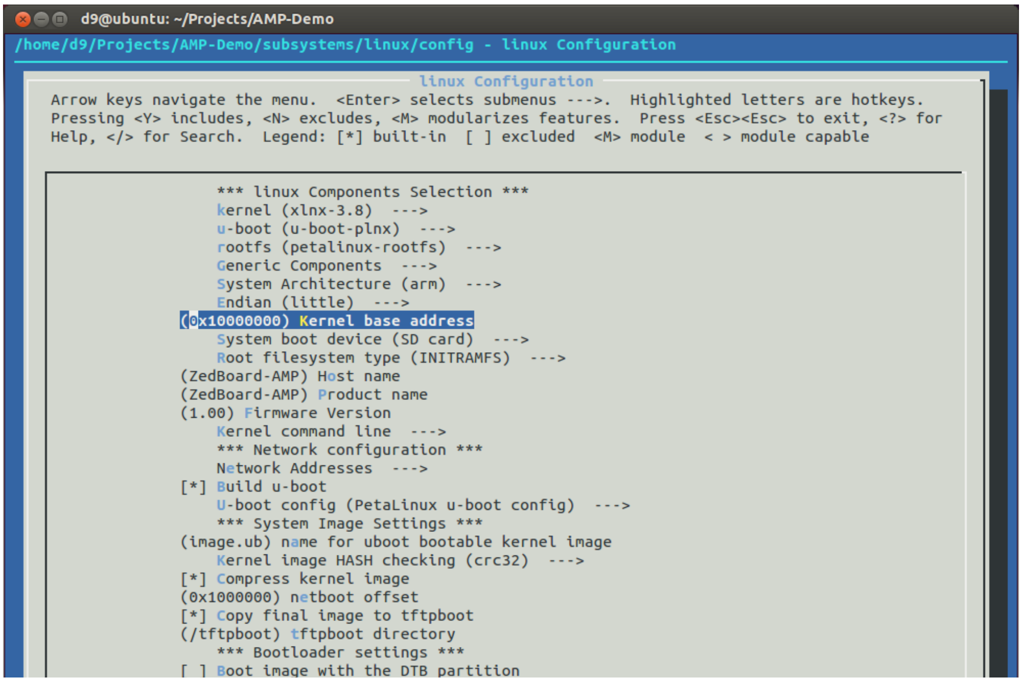
14. Verify, that we got the right configuration. In particular, amount of DDR memory - it should be 512M or 0x20000000. Below 'subsystems/linux/hw-description/xparameters.h' file generated in my case:

```
/*
 * (C) Copyright 2007-2008 Michal Simek
 *
 * Michal SIMEK <monstr@monstr.eu>
 *
 * CAUTION: This file is automatically generated by libgen.
 * Version: Xilinx EDK 2013.4 EDK_2013.4.20131205
 * Generate by U-BOOT v4.00.c
 * Project description at http://www.monstr.eu/uboot/
 */

#define XILINX_BOARD_NAME "AMP-Demo"

/* ARM is ps7_cortexa9_0 */
#define XPAR_CPU_CORTEXA9_CORE_CLOCK_FREQ_HZ    666666687

/* Interrupt controller is ps7_scugic_0 */
#define XILINX_PS7_INTC_BASEADDR        0xf8f00100

/* System Timer Clock Frequency */
#define XILINX_PS7_CLOCK_FREQ   333333343

/* Uart console is ps7_uart_1 */
#define XILINX_PS7_UART
#define XILINX_PS7_UART_BASEADDR    0xe0001000
#define XILINX_PS7_UART_CLOCK_HZ    50000000
```

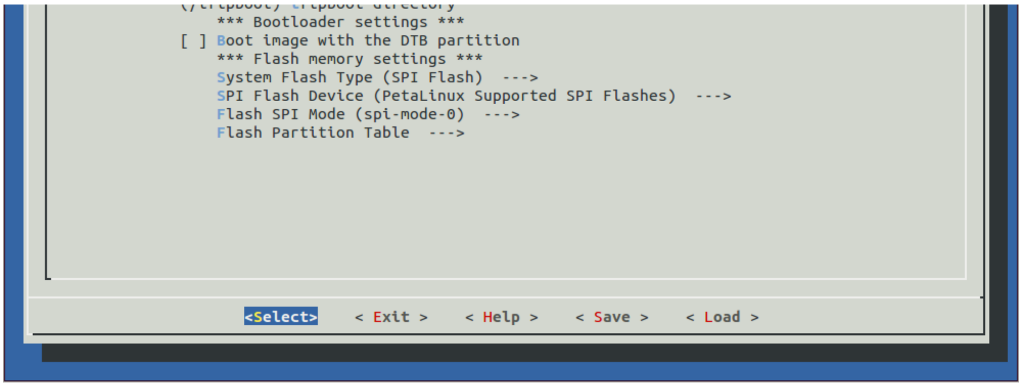
```
/* IIC doesn't exist */

/* GPIO doesn't exist */

/* SDIO controller is ps7_sd_0 */
#define XILINX_PS7_SDIO_BASEADDR        0xe0100000

/* Main Memory is ps7_ddr_0 */
#define XILINX_RAM_START    0x00000000
#define XILINX_RAM_SIZE     0x20000000

/* Flash Memory is ps7_qspi_0 */
#define XILINX_PS7_QSPI_FLASH_BASEADDR  0xE000D000
#define XILINX_SPI_FLASH_MAX_FREQ   50000000
#define XILINX_SPI_FLASH_CS 0

/* Sysace doesn't exist */

/* Ethernet controller is ps7_ethernet_0 */
#define XILINX_PS7_GEM_BASEADDR         0xe000b000
```

15. Next step is to configure our PetaLinux project. AMP system share memory between Linux Kernel and FreeRTOS, so PetaLinux project must be configured to segment the memory and lets split it 256M/256M since our ZedBoard have 512M total of DDR3 memory. I will also change boot media type to SD Card, Host and Product names. So, run petalinux-config and make next this changes:

> *cd ~/Projects/AMP-Demo/*
> *petalinux-config*



```
d9@ubuntu: ~/Projects/AMP-Demo
/home/d9/Projects/AMP-Demo/subsystems/linux/config - linux Configuration
                               linux Configuration
  Arrow keys navigate the menu.  <Enter> selects 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

                 *** linux Components Selection ***
                 kernel (xlnx-3.8)  --->
                 u-boot (u-boot-plnx)  --->
                 rootfs (petalinux-rootfs)  --->
                 Generic Components  --->
                 System Architecture (arm)  --->
                 Endian (little)  --->
             (0x10000000) Kernel base address
                 System boot device (SD card)  --->
                 Root filesystem type (INITRAMFS)  --->
             (ZedBoard-AMP) Host name
             (ZedBoard-AMP) Product name
             (1.00) Firmware Version
                 Kernel command line  --->
                 *** Network configuration ***
                 Network Addresses  --->
             [*] Build u-boot
                 U-boot config (PetaLinux u-boot config)  --->
                 *** System Image Settings ***
             (image.ub) name for uboot bootable kernel image
                 Kernel image HASH checking (crc32)  --->
             [*] Compress kernel image
             (0x1000000) netboot offset
             [*] Copy final image to tftpboot
             (/tftpboot) tftpboot directory
                 *** Bootloader settings ***
             [ ] Boot image with the DTB partition
```



```
(/tftpboot) tftpboot directory
    *** Bootloader settings ***
[ ] Boot image with the DTB partition
    *** Flash memory settings ***
    System Flash Type (SPI Flash)  --->
    SPI Flash Device (PetaLinux Supported SPI Flashes)  --->
    Flash SPI Mode (spi-mode-0)  --->
    Flash Partition Table  --->
```

16. Time to configure Linux Kernel for AMP. In a project directory:

    > *petalinux-config -c kernel*

    In a main page:

    - Make sure that 'Enable loadable module support' is selected.
    - In 'Kernel Features' -> make sure that 'High Memory Support' is enabled.
    - In 'Kernel Features' -> change 'Memory split' to '2G/2G'.
    - In 'Device Drivers' -> 'Generic Driver Options' -> make sure 'Userspace firmware loading support' is enabled.
    - In 'Device Drivers' -> 'Remoteproc drivers(EXPERIMENTAL)' -> change 'ZYNQ remoteproc' to (module) and disble Microblaze support.
    - In 'Device Drivers' -> 'Rpmsg drivers(EXPERIMENTAL)' -> set all three modules to ('An rpmsg server sample', 'rpmsg OMX driver' and 'An FreeRTOS statistic')

    Save changes and quit from menuconfig.

17. Configure PetaLinux project ROOTFS. All we need to change now is to in 'Apps-->' menu add/enable 'latencystat' app.

    > *petalinux-config -c rootfs*

18. Next very important step is to update 'Device Tree Source'(DTS). We need it to do because 'remoteproc' driver instantianated and configured by its node in a device

tree.
DTS file we need to modify located in a our petalinux project 'AMP-Demo/subsystems/linux/hw-description/system.dts'. We have to add 'remoteproc' node to 'ps7_axi_interconnect_0' device. Below a portion of my system.dts file. More details regarding 'remoteproc' you can find in a Xilinx UG978 v2013.10:

```
ps7_axi_interconnect_0: amba@0 {
    #address-cells = <1>;
    #size-cells = <1>;
    compatible = "xlnx,ps7-axi-interconnect-1.00.a", "simple
    ranges ;
    test: remoteproc-test@0 {
        compatible = "xlnx,zynq_remoteproc";
        reg = < 0x0 0x10000000 >;
        interrupt-parent = <&ps7_scugic_0>;
        interrupts = < 0 37 4 0 38 4>;
        firmware = "freertos";
        ipino = <6>;
        vring0 = <2>;
        vring1 = <3>;
    } ;
    ps7_spi_0: ps7-spi@e0006000 {
        clock-names = "ref_clk", "aper_clk";
        clocks = <&clkc 25>, <&clkc 34>;
        compatible = "xlnx,ps7-spi-1.00.a";
        interrupt-parent = <&ps7_scugic_0>;
        interrupts = <0 26 4>;
        num-chip-select = <3>;
        reg = <0xe0006000 0x1000>;
        bus-num = <0>;
        speed-hz = <1000000>;         //1000kHz
        xlnx,has-ss0 = <0x1>;
        xlnx,has-ss1 = <0x1>;
        xlnx,has-ss2 = <0x1>;
        xlnx,spi-clk-freq-hz = <0xF4240>;
        device@0{
            compatible="linux,spidev";
            reg =<0>;         //CS0
            spi-max-frequency= <1000000>;
        };
        device@1{
            compatible="linux,spidev";
            reg =<1>;         //CS1
            spi-max-frequency= <1000000>;
        };
    } ;
    ps7_spi_1: ps7-spi@e0007000 {
        clock-names = "ref_clk", "aper_clk";
        clocks = <&clkc 26>, <&clkc 35>;
        compatible = "xlnx,ps7-spi-1.00.a";
        interrupt-parent = <&ps7_scugic_0>;
        interrupts = <0 49 4>;
        num-chip-select = <2>;
        reg = <0xe0007000 0x1000>;
        bus-num = <1>;
        speed-hz = <1000000>;         //1000kHz
```

```
            xlnx,has-ss0 = <0x1>;
            xlnx,has-ss1 = <0x1>;
            xlnx,spi-clk-freq-hz = <0xF4240>;
            device@0{
                compatible="linux,spidev";
                reg =<0>;         //CS0
                spi-max-frequency= <1000000>;
            };
            device@1{
                compatible="linux,spidev";
                reg =<1>;         //CS1
                spi-max-frequency= <1000000>;
            };
        } ;
        ps7_ethernet_0: ps7-ethernet@e000b000 {
            #address-cells = <1>;
            #size-cells = <0>;
            clock-names = "ref_clk", "aper_clk";
            clocks = <&clkc 13>, <&clkc 30>;
            compatible = "xlnx,ps7-ethernet-1.00.a";
            interrupt-parent = <&ps7_scugic_0>;
            interrupts = <0 22 4>;
            local-mac-address = [ 00 0a 35 00 18 e0 ];
            phy-handle = <&phy0>;
            phy-mode = "rgmii-id";
            reg = <0xe000b000 0x1000>;
            xlnx,enet-reset = "";
            xlnx,eth-mode = <0x1>;
            xlnx,has-mdio = <0x1>;
            xlnx,ptp-enet-clock = <111111115>;
            mdio {
                #address-cells = <1>;
                #size-cells = <0>;
                phy0: phy@0 {
                    compatible = "marvell,88e1116r";
                    device_type = "ethernet-phy";
                    reg = <0>;
                } ;
            } ;
        } ;
```

19. Now, with PetaLinux configured, we can add our FreeRTOS demo app we created and compiled using XSDK into PetaLinux rootfs image. In order to do it we can create PetaLinux app using template.

> *cd ~/Projects/AMP-Demo/*
> *petalinux-create -t apps --template install -n freertos_fw*

> *INFO: Create apps: freertos_fw*

> *INFO: New apps successfully created in /home/d9/Projects/AMP-Demo/components/apps/freertos_fw*

20. Copy our FreeRTOS compiled app into PetaLinux apps directory:

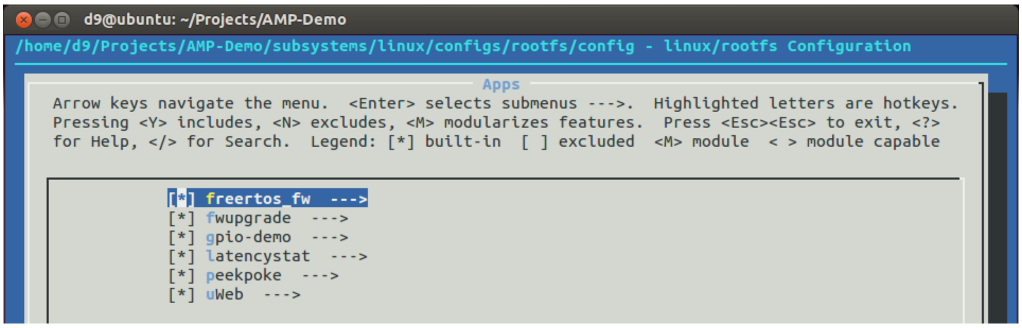
> *cd ~/Projects/AMP-Demo/components/apps/freertos_fw/*
> *cp ../../../../freertos_amp_demo/Debug/freertos_amp_demo.elf data/freertos*

21. Modify 'install' section of Makefile in the app folder:

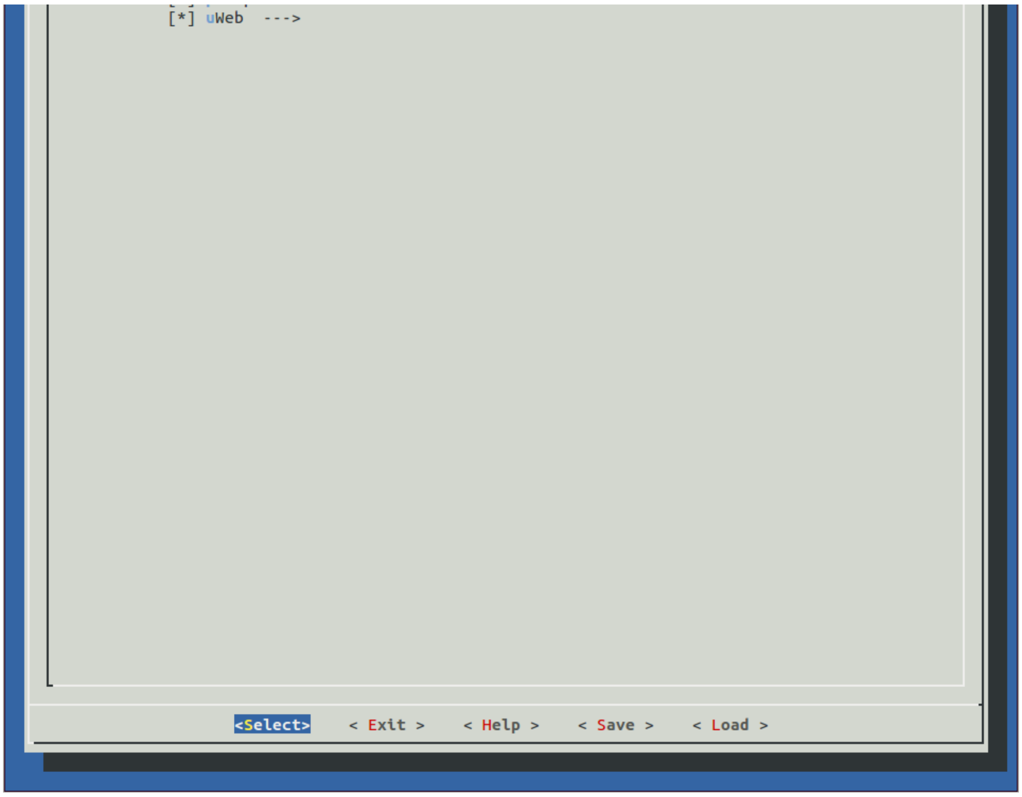
```
ifndef PETALINUX
$(error "Error: PETALINUX environment variable not set.  Cha
endif

include $(PETALINUX)/components/apps/apps.common.mk

all: build install

build:

clean:

.PHONY: install image

install:
	$(TARGETINST) -d data/freertos /lib/firmware/freerto
```

22. Configure PetaLinux 'rootfs' to include this new 'freertos_fw' in 'Apps-->' menu:

> *cd ~/Projects/AMP-Demo/*
> *petalinux-config -c rootfs*



```
d9@ubuntu: ~/Projects/AMP-Demo
/home/d9/Projects/AMP-Demo/subsystems/linux/configs/rootfs/config - linux/rootfs Configuration
                                     Apps
  Arrow keys navigate the menu.  <Enter> selects 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

              [*] freertos_fw  --->
              [*] fwupgrade  --->
              [*] gpio-demo  --->
              [*] latencystat  --->
              [*] peekpoke  --->
              [*] uWeb  --->
```



23. Build PetaLinux project, create BOOT.BIN image using our 'FSBL-AMP' and 'ZedBoard-AMP' bitstream file. Create/update prebuilt configuration.

```
cd ~/Projects/AMP-Demo/
petalinux-build
petalinux-package --boot --fsbl ../FSBL-AMP/Debug/FSBL-AMP.elf --fpga ../ZedBoard-AMP-HW/system_wrapper.bit --uboot --force -o images/linux/BOOT.BIN
petalinux-package --prebuilt --fpga ../ZedBoard-AMP-HW/system_wrapper.bit --force
```

24. Copy BOOT.BIN and image.ub to sd card, set ZedBoard MIO3, MIO4 and MIO5 to 'SD Boot' configuration and turn ZedBoard on.
25. Login as 'root'/'root' and verify our kernel version and build timestamp.

```
root@ZedBoard-AMP:~# uname -a
```

> *Linux ZedBoard-AMP 3.8.11 #2 SMP PREEMPT Wed Feb 26 15:39:37 EST 2014 armv7l GNU/Linux*
> *root@ZedBoard-AMP:~#*

26. Because we included remoteproc as a module and it not loaded yet - Linux must be using both CPU's in a conventional SMP way. Lets verify it:

> *more /proc/cpuinfo*

```
processor       : 0
model name      : ARMv7 Processor rev 0 (v7l)
BogoMIPS        : 1332.01
Features        : swp half thumb fastmult vfp edsp neon vfpv
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x3
CPU part        : 0xc09
CPU revision    : 0

processor       : 1
model name      : ARMv7 Processor rev 0 (v7l)
BogoMIPS        : 1332.01
Features        : swp half thumb fastmult vfp edsp neon vfpv
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x3
CPU part        : 0xc09
CPU revision    : 0

Hardware        : Xilinx Zynq Platform
Revision        : 0000
Serial          : 0000000000000000
```

27. Now, lets load remoteproc driver:

> *modprobe zynq_remoteproc*

```
CPU1: shutdown
 remoteproc0: 0.remoteproc-test is available
 remoteproc0: Note: remoteproc is still under development an
 remoteproc0: THE BINARY FORMAT IS NOT YET FINALIZED, and ba
```

> *modprobe rpmsg_freertos_statistic*

```
remoteproc0: powering up 0.remoteproc-test
 remoteproc0: Booting fw image freertos, size 2130820
 remoteproc0: remote processor 0.remoteproc-test is now up
virtio_rpmsg_bus virtio0: rpmsg host is online
virtio_rpmsg_bus virtio0: creating channel rpmsg-timer-stati
rpmsg_freertos_statistic rpmsg0: new channel: 0x400 -> 0x50!
```

So, second Processor unloaded from Linux and is setup to execute the FreeRTOS firmware.

28. Now, we can run 'latencystat' FreeRTOS demo app.

> *latencystat -b*

```
Linux FreeRTOS AMP Demo.
   0: Command 0 ACKed
   1: Command 1 ACKed
Waiting for samples...
   2: Command 2 ACKed
   3: Command 3 ACKed
   4: Command 4 ACKed
-----------------------------------------------------------
Histogram Buckvirtio_rpmsg_bus virtio0: msg received with no
et Values:
    Bucket 341 ns (38 ticks) had 14813 frequency
    Bucket 431 ns (48 ticks) had 1 frequency
    Bucket 521 ns (58 ticks) had 1 frequency
    Bucket 593 ns (66 ticks) had 1 frequency
    Bucket 692 ns (77 ticks) had 1 frequency
-----------------------------------------------------------
Histogram Data:
    min: 341 ns (38 ticks)
    avg: 341 ns (38 ticks)
    max: 692 ns (77 ticks)
    out of range: 0
    total samples: 14817
-----------------------------------------------------------
root@ZedBoard-AMP:~#
```

29. Access the TraceBuffer - a section of shared memory which is only written to by the FreeRTOS application. This
TraceBuffer used as a logging console to transfer information to Linux and 'latencystat' app uses it. So, lets check it out:

> *more /sys/kernel/debug/remoteproc/remoteproc0/trace0*

Congratulation! We got Linux-FreeRTOS AMP configuration running on our ZedBoard.

‹ Use SDCard to boot ZedBoard using PetaLinux 13.10

The easiest way to add screen to ZedBoard. ›

Posted in Linux, Xilinx Zynq, ZedBoard **Tagged with:** 13.10, 64bit, AMP, FreeRTOS, JTAG, Linux, microSD, MicroZed, MultiProcessing, PetaLinux, RealTime, sdcard, ubuntu, Vivado, xilinx, XSDK, ZedBoard

11 comments on “ZedBoard Linux-FreeRTOS AMP Board Bringup Guide.”

omcconnell@innovative-dsp.com says:

April 17, 2014 at 13:44

I have followed all of these instructions exactly, but I'm afraid there is something wrong with my environment. When I get to the step petalinux-build, I get an error that says "no architecture is defined" and the build fails. Does anyone know what the source of this problem is?

Log in to Reply

omcconnell@innovative-dsp.com says:

April 17, 2014 at 13:50

I should also note that I am using CentOS 6.5 64 bit and not Ubuntu

Log in to Reply

d9#idv-tech#com says:

April 17, 2014 at 14:03

Can you post whole log?

Log in to Reply

omcconnell@innovative-dsp.com says:

April 17, 2014 at 17:06

Is this what you wanted to see?

```
[root@localhost AMP-Demo]# petalinux-build -c all
WARNING: Your PetaLinux project was last modified by PetaLinux SDK version "2013.04",
WARNING: however, you are using PetaLinux SDK version "2013.10".
Please input "y" to continue. Otherwise it will exit![n]y
INFO: Checking component...
INFO: Generating make files and build linux
INFO: Generating make files for the subcomponents of linux
INFO: Building linux
[ERROR] /opt/petalinux-v2013.10-final/etc/build/common.mk:17: *** "No architecture is defined!". Stop.
[ERROR] make[1]: *** [sub_pre-build_component_/none/apps/multi/freertos_fw] Error 2
ERROR: Failed to build linux
```

Log in to Reply

d9#idv-tech#com says:

April 18, 2014 at 09:04

Oh boy.
Remove Petalinux 2013.4, and recreate AMP-Demo project using 2013.10. They changed a lot of thing between this two versions of Petalinux.

Log in to Reply

omcconnell@innovative-dsp.com says:

April 21, 2014 at 13:27

I did create the AMP-DEMO with 2013.10. I did everything with 2013.10. I installed 2013.4 after the fact just as a test to see if I could get that one working. The problem is with 2013.10

Log in to Reply

d9#idv-tech#com says:

April 22, 2014 at 12:41

Heh, can you post original then?
In general, good place to start digging into the problem is petalinux build log. It's located in petalinux project 'build' directory.

Log in to Reply

omcconnell@innovative-dsp.com says:

April 22, 2014 at 12:57

I followed your instructions on Ubuntu 13.10 64 bit and it works. I'm not sure what the deal was with CentOS 6.5 64 bit.

Log in to Reply

omcconnell@innovative-dsp.com says:

April 22, 2014 at 12:59

I would have preferred to use CentOS since that is where all of my other development is done. If a solution to that problem is ever found, please post it.

Log in to Reply

omcconnell@innovative-dsp.com says:

April 28, 2014 at 16:43

Nevermind that problem. This stemmed from the fact that I didn't use the right board from Vivado. I chose the ZC702 instead of the avnet version 7Z020. Once I recreated a project using the right board I had no problem making it work. Beautiful post. One thing I had to do that I noticed wasn't in here though is that I had to make a symbolic link from make to gmake so that the Vivado tools would build the projects.

Log in to Reply

TheHinky says:

August 2, 2014 at 10:19

Thanks for this guide, works perfectly.

Log in to Reply

Leave a Reply

You must be logged in to post a comment.

WordPress