Embedded Design with PetaLinux Tools

Embedded Software 4

EMBD-PLNX-ILT (v1.0)

Course Specification

Course Description

This intermediate-level, two-day course provides embedded systems developers with experience in creating an embedded Linux system targeting a Zynq® All Programmable System on a Chip (SoC) processor development board using PetaLinux Tools. The course offers students hands-on experience with building the environment and booting the system using a Zynq All Programmable SoC design with PetaLinux Tools on the ARM® Cortex™-A9 processor.

This course also introduces embedded Linux components, use of open-source components, environment configurations, network components, and debugging options for embedded Linux platforms. The primary focus is on embedded Linux development in conjunction with the Xilinx tool flow.

Level – Embedded Software 4

Price –

Course Duration – 2 days

Course Part Number – EMBD-PLNX-ILT

Who Should Attend? – Embedded software developers interested in customizing the PetaLinux kernel on an ARM processor design for a Xilinx Zynq All Programmable SoC

Prerequisites

- *Essentials of FPGA Design* (introductory FPGA design course)
- *Embedded Systems Software Development* course (software development for FPGA embedded systems course)

Software Tools

- Vivado® Design or System Edition 2015.2
- PetaLinux Tools 2015.2

Hardware

- Architecture: Zynq-7000 All Programmable SoC*
- Demo board: ZedBoard*

* This course focuses on the Zynq-7000 All Programmable SoC architecture. Check with your local Authorized Training Provider for the specifics of the in-class lab board or other customizations.

After completing this comprehensive training, you will have the necessary skills to:

- Explain what an embedded Linux kernel is
- Describe the Linux device driver architecture
- Create a PetaLinux project to configure and build an image using PetaLinux tools
- Create a working ARM Cortex-A9 MPCore Linux system using the Vivado Design Suite and PetaLinux tools
- List various hardware interfacing options available for the ARM Cortex-A9 MPCore
- Build custom hardware cores and device drivers using the user space I/O (UIO) framework

Course Outline

Day 1

- Embedded Linux Overview
- **Lab 1:** A First Look
- Introduction to the PetaLinux Tools
- **Lab 2:** Build and Boot an Image
- Application Development and Debugging
- **Lab 3:** Application Development and Debugging
- Networking and TCP/IP
- **Lab 4:** Networking and TCP/IP
- Device Drivers, User Space I/O, and Loadable Modules
- **Lab 5:** Accessing Hardware Devices from User Space

Day 2

- Board Bring Up with the Vivado Design Suite and PetaLinux Tools
- **Lab 6:** Basic Hardware Design with the Vivado Design Suite and PetaLinux Tools
- Custom Hardware Development and Interfacing
- **Lab 7:** Custom Hardware Development
- Custom Driver Development (quick review)
- **Lab 8:** Custom Driver Development

Lab Descriptions

- **Lab 1:** A First Look – Log in to the ARM processor Linux system and make comparisons between the embedded Linux and desktop Linux environments.
- **Lab 2:** Build and Boot an Image – Explore the Linux configuration menus and build the ARM processor Linux kernel and applications. Download the resulting system image to the development board.
- **Lab 3:** Application Development and Debugging – Create a simple user application with PetaLinux Tools and debug the application with System Debugger.
- **Lab 4:** Networking and TCP/IP – Explore the kernel configuration menu. Log in to the ARM processor Linux system by using telnet. Transfer files to and from Linux by using FTP. Build and experiment with web-based applications under Linux.
- **Lab 5:** Accessing Hardware Devices from User Space – Access a hardware device directly from user space. Use the UIO framework to access a hardware device. Experience loading and unloading kernel modules.
- **Lab 6:** Basic Hardware Design with the Vivado Design Suite and PetaLinux Tools – Use the Vivado IP integrator (IPI) to create a basic hardware design with the ARM Cortex-A9 MPCore. Use PetaLinux Tools to create a new embedded Linux target for the hardware design.
- **Lab 7:** Custom Hardware Development – Design a customized IP core. Integrate the IP core with the AXI interface and debug.
- **Lab 8:** Custom Driver Development – Write a UIO program to access the PWM AXI IP core. Boot from Flash and verify it on the target board.

Register Today

Xilinx's network of Authorized Training Providers (ATP) delivers public and private courses in locations throughout the world. Please contact your closest ATP for more information, to view schedules, or to register online.

Visit **www.xilinx.com/training** and click on the region where you want to attend a course.

Americas, contact your training provider at www.xilinx.com/training/atp.htm#NA or send your inquiries to registrar@xilinx.com.

Europe, contact your training provider at www.xilinx.com/training/atp.htm#EU or send your inquiries to eurotraining@xilinx.com.

Asia Pacific, contact your training provider at www.xilinx.com/training/atp.htm#AP, or send your inquiries to education_ap@xilinx.com, or call +852-2424-5200.

Japan, contact your training provider at www.xilinx.com/training/atp.htm#JP, or send your inquiries to education_kk@xilinx.com, or call +81-3-6744-7970.