C Language Programming with SDK

Course Description

This course is broken into a day of C language review, including variable naming, usage, and modifiers as well as an introduction to the Software Development Kit (SDK) environment, an explanation of the use of the preprocessors, program control, and proper use of functions. The second day consists of common issues and techniques employed by embedded programmers in the Xilinx SDK environment. This comprehensive course equally balances lecture modules with practical hands-on lab work.

Level: Embedded 1

Training Duration: 2 days

Who Should Attend?

Programmers and software engineers looking to reinforce their C skills for the embedded environment and hardware engineers interested in software engineering basics

Prerequisites:

- Basic familiarity with embedded systems
- Basic background in programming

Software Tools:

- Xilinx ISE® Design Suite: Embedded Edition 14.2

Hardware:

- Architecture: N/A*
- Demo board: Zynq™7000 All Programmable SoC ZC702 or Zed board*

Skills Gained: After completing this training, you will be able to:

- Recognize C language symbology
- Design an effective C language program for the embedded environment
- Identify the nuances between functions and macros
- Effectively utilize numeric techniques
- Debug software using the GNU debugging tool in the SDK software environment

Course Outline

1. The C Language

2. SDK Environment
Lab 1: SDK Environment

3. C Preprocessor

4. Variables

5. Control Structures
Lab 2: Writing a Simple Program

6. Functions and Libraries

7. Program Design

8. Common Errors

9. Debugging Strategies

10. Dynamic Memory
Lab 3: Debugging Dynamic Memory

11. The Stack
Lab 4: Debugging Stack Issues

12. Numeric Techniques

13. The Xilinx Embedded Environment
Lab 5: Driving Xilinx Hardware

Lab Description

Lab 1: SDK Environment – Walks you through the process of configuring the hardware through SDK, building a simple application, and verifying that it works.
Lab 2: Writing a Simple Program – Examine a piece of existing code, then complete the program using the skills developed in the previous lecture modules.
Lab 3: Debugging Dynamic Memory – Guides you through the phases of debugging a program with memory leaks.
Lab 4: Debugging Stack Issues – Debug stack issues, another common problem.
Lab 5: Driving Xilinx Hardware – Combine the abstraction of programming with actual hardware to drive the LEDs on the demo board.