IEEE Boston

Serving Eastern Massachusetts, and throughout New England

Embedded Linux BSPs and Device Drivers – On-Line Course

You will have 180 days to access the on-line course.

Register Now

Embedded Linux Board Support Packages and Device Drivers (EL805)

Lecturer – Mike McCullough is President and CEO of RTETC, LLC. Mike has a BS in Computer Engineering and an MS in Systems Engineering from Boston University. A 20-year electronics veteran, he has held various positions at LynuxWorks, Tilera, Embedded Planet, Wind River Systems,

IEEE Boston Sponsored Conferences & Events

Lockheed Sanders, Stratus Computer and Apollo Computer. RTETC, LLC is a provider of Eclipse-based software development tools, training and consulting services for the embedded systems market.

Course Summary – This video course provides advanced training in the development of Embedded Linux Board Support Packages (BSPs) and Device Drivers. The first part of the course focuses on BSP and Software Development Kit (SDK) development in an Embedded Linux context with a focus on application performance measurement and improvement. The latter part of the course covers Embedded Linux Device Driver development including key device driver decisions and deployment considerations for Embedded Linux BSPs.

Who Should Attend – The course is designed for real-time engineers who are developing Embedded Linux BSPs and Device Drivers for Embedded Linux distributions. It is also targeted at experienced developers requiring a refresher course on Linux BSP and Device Driver development.

Course Objectives

• To gain an understanding of the complexities of BSP and SDK development and their uses in Embedded Linux systems.
• To provide a basic understanding of the Linux I/O Subsystem and the Device Driver Models provided with Embedded Linux distributions.
• To gain an in-depth understanding of character-based device drivers in Embedded Linux
• To understand key device driver subsystems including relatively slow I/O interconnects such as I2C, SPI and USB as well as high-speed interfaces such as Ethernet, USB 3.0 and PCIe
• To give students the confidence to apply these concepts to their next Embedded Linux project.

Course Schedule
Getting Started with Embedded Linux

Online Courses

Fundamentals of Real-Time Operating Systems (RT201) On-Line Course

Design Thinking For Technical Work

Verilog 101: Verilog Foundations

SystemVerilog 101 (SV101): Design Constructs

SystemVerilog 102 (SV102): Verification Constructs

High Performance Project Management

Introduction to Embedded Linux

Software Development for Medical Device Manufacturers

Fundamental Mathematics Concepts Relating to Electromagnetics

Reliability Engineering for the Business World

Embedded Linux Optimization

Embedded Linux Training Overview
Linux Terminology, History and the GPL
Building the Kernel Source Code
Embedded Linux Kernels
BSPs and SDKs
Linux References (Books and Online)
BSP Requirements
U-Boot and Bootloader Development
Embedded Linux BSP Development Basics
Basic BSP Development
Files and Filesystem Support
The I/O Subsystem: Talking to Hardware
Memory Management and Paging
Error Handling in Embedded Linux BSPs
Timing and Timers
Interrupt and Exception Handling in BSPs
BSP Deployment Issues and Practices
Embedded Linux SDK Basics
The 3 Pieces of an SDK
Embedded Linux Distributions and the GNU Compiler Collection (GCC)
Other Embedded Linux Development Tools
Library Support, Glibc and Alternatives
SDK Deployment and Support
Debugging
GDB, GDB Server and the GDB Server Debugger
Other Debug and Test Tools
An Eclipse Remote Debug Example
Advanced Debug with printk and syslogd
System-Level Debug
System-Level Debug Tools
The /proc and sys Filesystems
Advanced Logging Methods
KGDB and KDB
Crash Dumps
Debugging Embedded Linux Systems
Configuring Embedded Linux
Config Methods
Config Syntax

Embedded Linux BSPs and Device Drivers

All Chapter Meetings

JUN 16 Tue (http://ieeeboston.org/calendar/action~oneday/exact_date~6-16-2020/)

7:00 pm How Does Covid-19 Impact the Lif... @ Webinar (http://ieeeboston.org/event/how-does-covid-19-impact-the-life-science-start-up/?instance_id=2887)

JUL 28 Tue (http://ieeeboston.org/calendar/action~oneday/exact_date~7-28-2020/)

6:30 pm Digital Signal Processing for So... @ Webinar (http://ieeeboston.org/event/digital-signal-processing-for-software-radio-live-webinar-course/?instance_id=2888)

JUL 30 Thu (http://ieeeboston.org/calendar/action~oneday/exact_date~7-30-2020/)

6:30 pm Digital Signal Processing for So... @ Webinar (http://ieeeboston.org/event/digital-signal-processing-for-software-radio-live-webinar-course/?instance_id=2898)

AUG 4

6:30 pm Digital Signal Processing for So... @

Adding Code to the Linux Kernel
Booting Embedded Linux
Processor Startup
Initial Functions
The initcalls
Using __init Functions
NFS Booting
Root File Systems
RAMdisk Booting with initrd
RAMdisk Booting with initramfs
initrd vs initramfs
Root File System Development
Busybox Development
Building a RAMdisk for an initrd
Building a RAMdisk for an initramfs
Flash File System Development

Testing and Debug of Embedded Linux BSPs
Kernel Debug and Kernel Probes
Kexec and Kdump
The Linux Test Project (LTP)
Performance Tuning Embedded Linux BSPs
Virtualization
Measuring Embedded Linux BSP Performance
Common Considerations
Uncommon Considerations
BootLoader Optimizations
Boot Time Measurements
Effective Memory and Flash Usage
Filesystem Performance Measurement
Some Ideas on Performance Measurement

The Original UNIX Device Driver Model
The fops and file structs
The inode and dentry structs
Major and Minor Numbers
Embedding Channel Information
Deferring Work
The /proc Filesystem

Tue (http://ieeeboston.org/calendar/action~oneday/exact_date~8-4-2020/)
Webinar (http://ieeeboston.org/event/digital-signal-processing-for-software-radio-live-webinar-course/?instance_id=2890)

AUG 6 Thu (http://ieeeboston.org/calendar/action~oneday/exact_date~8-6-2020/)
6:30 pm Digital Signal Processing for So... @ Webinar (http://ieeeboston.org/event/digital-signal-processing-for-software-radio-live-webinar-course/?instance_id=2899)

View Calendar → (http://ieeeboston.org/calendar/cat_ids~11,9/)
Add

Configuring the Device Driver
A Simulated Device Driver
Modularization Revisited
The Evolution of a New Driver Model
The Initial Object-Oriented Approach
Platform Devices and Drivers
A Generic Subsystem Model
The Generic Subsystem Model in Detail
Subsystem Registration
The Probe and Init Functions
The Show and Store Functions
User Access via the /sys Filesystem
Configuring the New Device Driver
The udev Linux Application
Comparing the Two Driver Models
The Flattened Device Tree (FDT)
openBoot and its Effect on Embedded Linux
The Device Tree Script (dts) File
The Device Tree Compiler (dtc)
The Device Tree Blob (dtb) File
Building a dtb File
Hybrid Device Drivers
Other fops Functions
The Need for Ioctl
Linux Device Driver Subsystems
Direct Connect Device Drivers
Serial/Console Drivers, I2C & SPI
Real-Time Clocks and Watchdogs
GPIO and the Pinmux
Flash MTDs and Direct Memory Access
USB, Power and CPU Management
Video and Audio
PCI and VME
Block Devices
RAMdisk and Flash Filesystems
MMCs and SD Cards
Network Device Drivers
MAC and PHY Device Drivers
net_device and net_device_stats

Network Device Initialization
Device Discovery and Dynamic Initialization
Network Interface Registration
Network Interface Service Functions
Receiving and Transmitting Packets
Notifier Chains and Device Status Notification
Unwired Device Drivers
Wireless Device Drivers (WiFi, WLAN)
Bluetooth and BlueZ
Infrared and IrDA
Cellular from 2G to 5G
Drivers in User Space
Accessing I/O and Memory Regions
User Mode SCSI, USB and I2C
UIO
High-Speed Interconnects
PCIe
iSCSI
Infiniband
FibreChannel
Debugging Device Drivers
kdb, kgdb and JTAG
Kernel Probes
Kexec and Kdump
Kernel Profiling
User Mode Linux
Performance Tuning Device Drivers

Some Final Recommendations

Course Fee: $350.00

Blogroll

Blog

Recent Posts

What is Wrong with the IEEE Boston Section

IEEE Boston Drones…. GDPR

IEEE Boston Blockchain and ICOs