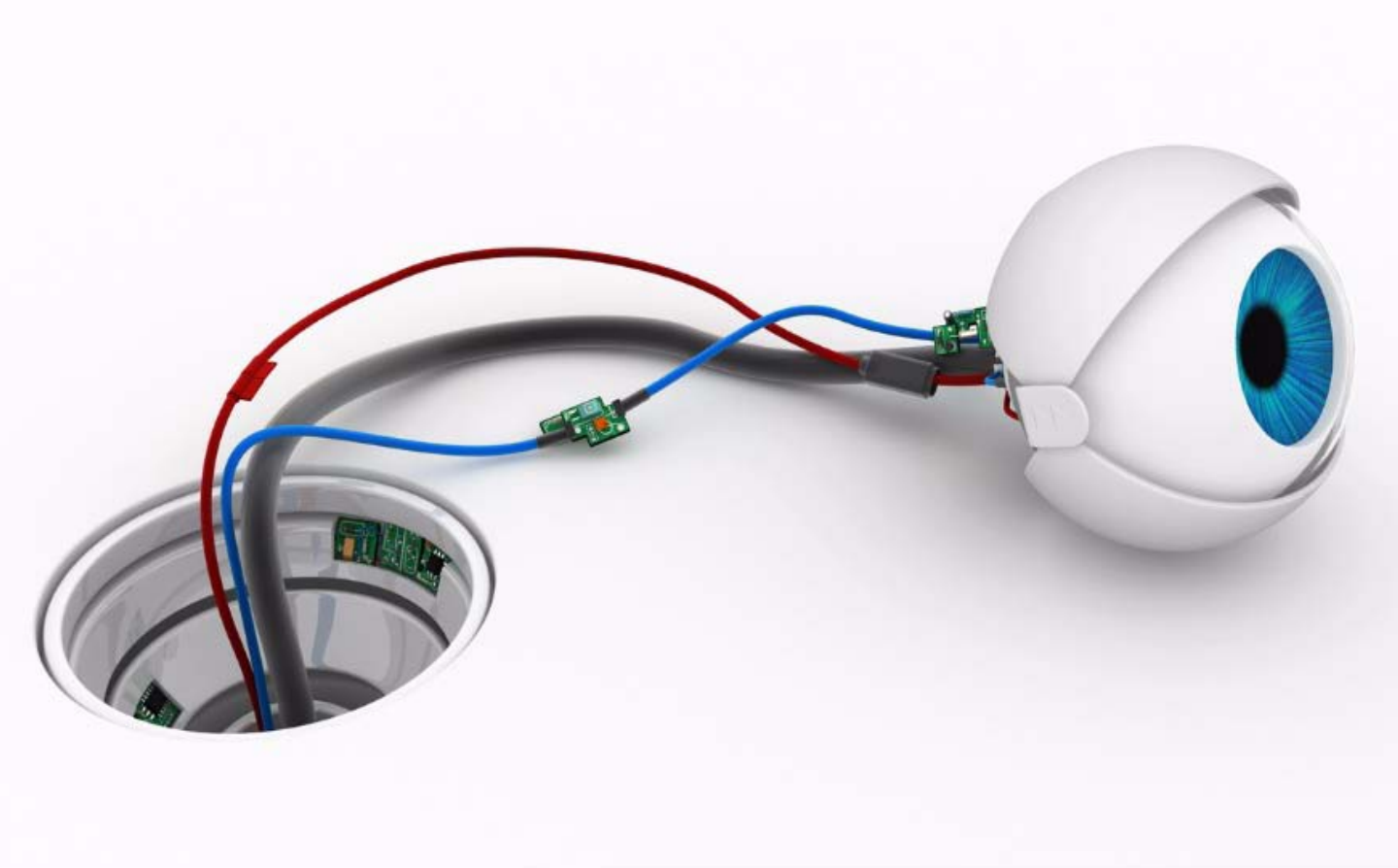
BOSTON

THE REFLECTOR

ISSUE #7
JULY 2018



IEEE
Advancing Technology for Humanity

TABLE OF CONTENTS

Already, Augmented Reality has a Positive Impact on the Workplace

Kevin Flavin, Chair, IEEE Boston Section Electronic Communications Team

Coincidentally, as I write this, GE has announced that they are selling their healthcare business as well as their portion of the oil & gas company Baker Hughes. Last month, they sold their railroad business. A couple of days ago, I read somewhere that they sold their industrial engines business, too. Some of this news is puzzling to me considering the sharp rise in manufacturing that the U.S. is experiencing right now. I'm not worried about the companies that are spun off and sold. I believe that a smaller, and implicitly more nimble, company could take advantage of new ideas being tested right now.

Well, back to the business here at Boston Section. Last month, we talked about Virtual Reality, which has some pretty cool implementations. However, the real exciting developments are with Augmented Reality (AR). Augmented Reality has slowly crept into some of our lives already.

When I first read about Augmented Reality, as a defined term, I immediately thought of the 'Heads Up Display' that I've seen pictures of in fighters.

Today, however, Augmented Reality is finding its way into common, and practical uses.

IKEA has made an app (https://www.wired.com/story/ikea-place-ar-kit-augmented-reality/
) for a mobile phone that modifies the image of the piece of furniture that you are looking to buy, and 'places' it into the space you are aiming your phone's camera at, in real-time. For example, to see what that chair looks like in your apartment, or home office, simply access the piece of furniture in their app, select the option to view in your home, and "voilà!", there it is against the wall.

On another level of Augmented Reality, one that doesn't involve ecommerce, but certainly involves apps and computers, imagine what the office of that future looks like. To quote from a Wired article about Augmented Reality in the Workplace (https://www.wired.com/story/augmented-reality-in-the-workplace/), this is what it might look like:

"Picture it: You get to the office, grab a keyboard off the shelf (because air typing still sucks), and find an open space. You log in to your glasses, and your entire workspace appears in front of you. To your right is a shelf stocked with all the apps and bookmarks you use every day. You reach over and grab one, place it on the floor, and the full-scale CAD model of the car you were designing pops into place. Pinned to the wall are all your digital notes, arranged in exactly the way you left them last night. To your left hover six virtual screens displaying browser windows, stock tickers, and Twitter. You ask [your digital assistant] to pull up your email, and your inbox appears. You can see everything, but all anyone else sees is you, wearing glasses that look like standard Warby Parkers, typing on a keyboard and reaching around in the air. (This is considered socially acceptable-ish, somehow.) When you really need to focus, you flip on Occluded Mode to turn off the world around you, diving into a black hole of virtual productivity."

The part that strikes me is how easy the solution is to a problem I have right now - desk space and the need for three monitors! For what it's worth, I need three for pragmatic and efficient reasons - one for the codebase, another for the output, and one for the data it's ingesting.
First-world-problem, I know, but the AR would make this perfectly useful ('Why is a marketer coding?' is a ques-

tion for another day). Besides, writing software is simply digital manufacturing. This is fine for the white-collar worker, Augmented Reality is a little more difficult to roll out to, say, manufacturing….right?

Not so fast, says this article from last year (!) from the Harvard Business Review, Augmented Reality Is Already Improving Worker Performance (https://hbr.org/2017/03/augmented-reality-is-already-improving-worker-performance). In the online version of this article, there's an embedded YouTube video showing real impact on a single worker installing wiring into a panel of a machine - with and without AR glasses to help - and the savings were 34% in this instance. Let's consider for a moment the potential reduction of errors and re-construction of complex machinery like that in this video. With a 'heads-up' display of the diagrams related to the task at hand, instead of in a 3-ring binder, the diagrams would be immediately served from source materials, in a timely manner, with updates.

Here's another statistic that will have a substantive impact on deciding to use AR - the U.S. labor department reports that the majority of new manufacturing jobs will require at least some advanced education. As the need for more advanced education and training increases the cost of the labor, time savings like this will become a serious economic factor - possibly making the difference between company survival or failure.

Near me, a couple of towns over, are the Lowell Woolen Mills historic district, where manufacturing arguably began, at least for cloth. We've come a long way from those climate-controlled rooms full of flailing looms, where quick hands and repetitive motion were the only required skills. We've come a long way.

What does our future of making things look like? Will the trend toward AR and advanced manufacturing, both the analog (engines, etc.) and the digital (software) play out for us?

I certainly hope so.

Call for Articles

Now that the Reflector is all electronic, we are expanding the content the publication. One of the new features we will be adding are technical and professional development articles of interest to our members and the local technology community. These will supplement the existing material already in our publication.

Technical submissions should be of reasonable technical depth and include graphics and, if needed, any supporting files. The length is flexible; however, a four to five page limit should be used as a guide. An appropriate guide may be a technical paper in a conference proceeding rather than one in an IEEE journal or transaction.

Professional development articles should have broad applicability to the engineering community and should not explicitly promote services for which a fee or payment is required. A maximum length of two to three pages would be best.
To ensure quality, technical submissions will be reviewed by the appropriate technical area(s). Professional articles will be reviewed by the publications committee for suitability.The author will be notified of the reviewers' decision.

The Reflector is published the first of each month. The target submission deadline for the articles should be five weeks before the issue date (e.g., June 1st issue date; article submission is April 27). This will allow sufficient time for a thorough review and notification to the author.

We are excited about this new feature and hope you are eager to participate!
Submissions should be sent to;
ieeebostonsection@gmail.com

IEEE Boston Section Online Courses:

(Students have 90 day access to all online, self-paced courses)

Verilog101:Verilog Foundations
Full course description and registration at ,
http://ieeeboston.org/verilog-101-verilog-foundations-online-course/

System Verilog 101: Design Constructs
Full course description and registration at ,
http://ieeeboston.org/systemverilog-101-sv101-design-constructs-online-course/

System Verilog 102: Verification Constructs
Full course description and registration at ,
http://ieeeboston.org/systemverilog-102-sv102-verification-constructs-online-course/

High Performance Project Management
Full course description and registration at ,
http://ieeeboston.org/high-performance-project-management-online-course/

Introduction to Embedded Linux Part I
Full course description and registration at ,
http://ieeeboston.org/introduction-to-embedded-linux-part-i-el201-online-course/

Embedded Linux Optimization - Tools and Techniques
Full course description and registration at ,
http://ieeeboston.org/embedded-linux-optimization-tools-techniques-line-course/

Embedded Linux Board Support Packages and Device Drivers
Full course description and registration at ,
http://ieeeboston.org/embedded-linux-bsps-device-drivers-line-course/

Software Development for Medical Device Manufacturers
Full course description and registration at ,
http://ieeeboston.org/software-development-medical-device-manufacturers-line-course/

Fundamental Mathematics Concepts Relating to Electromagnetics
Full course description and registration at ,
http://ieeeboston.org/fundamental-mathematics-concepts-relating-electromagnetics-line-course/

Reliability Engineering for the Business World
Full course description and registration at ,
http://ieeeboston.org/reliability-engineering-business-world-line-course/

Design Thinking for Today's Technical Work
http://ieeeboston.org/design-thinking-technical-work-line-course/

Fundamentals of Real-Time Operating Systems
http://ieeeboston.org/fundamentals-of-real-time-operating-systems-rt201-on-line-course/

Industry Applications Society – 6:00PM, Tuesday, 17 July

MV & LV Grounding and Ground Fault Protection Systems

Speaker: Jeff Glenney, P.E, BSEE - Product Manager, Bender, Inc. Ground Fault Protection Systems

Electrical Grounding and Grounding Systems are not always completely understood, with solidly grounded, ungrounded, high resistance grounded, low resistance grounded all present in our electrical systems. There are reasons for each type of system and Bender will highlight why each type of system is desirable and how each can achieve the highest levels of electrical safety for man and machine, all while keeping operations running in the event of and during an electrical ground fault. Consulting Engineers, Facility Managers, Maintenance Personnel all should attend to gain important information on increased safety, uptime, and decreased risk during electrical ground faults. Ground Fault location technology will also be presented.

Biography: Jeff Glenney, P.E. has a Bachelor of Science degree in Electrical Engineering from the University of Saskatchewan and is a registered Professional Engineer in Saskatchewan. He has worked in the electrical protection relay market with various manufacturers since 1995. His roles have included sales engineer, U.S. sales engineering manager, vertical market manager for fuses and relays and now as a product manager. Jeff is a member of IEEE and co-authored “Type B Ground-Fault Protection on Adjustable Frequency Drives“. He has worked with many end customers and design engineers to assist in applying protective relays to meet their system needs.

Time: Refreshments start at 6pm, talk commences at 6:30pm

Location: MathWorks, Harvard A Conference Room, 1 Apple Hill Drive, Natick, MA 01760

Free and Open to the Public; RSVP is appreciated

Registration Link: https://events.vtools.ieee.org/meeting_registration/register/174163

IEEE Boston Section Social Media Links:

Twitter: https://twitter.com/ieeeboston

Facebook: https://www.facebook.com/IEEEBoston

YouTube: https://www.youtube.com/user/IEEEBostonSection

Google+: https://plus.google.com/107894868975229024384/

LinkedIn: https://www.linkedin.com/groups/IEEE-Boston-Section-3763694/about

2018 MIT IEEE Undergraduate Research Technology Conference

Call for Submissions

CONFERENCE: Oct 5-7, 2018 | Massachusetts Institute of Technology, Cambridge MA, USA

SUBMISSION DEADLINE: June 30th, 2018

SUBMIT TO: https://ieee-r1-studentconference.myreviewroom.com

Envisioning a technical conference targeted towards undergraduate students all over the globe, the MIT IEEE Student Branch in 2015 inaugurated the IEEE MIT Undergraduate Research Technology Conference. This year we are organizing it again with the goal to make the conference a venue where undergraduate students can meet to present, discuss, and develop solutions advancing technology. Participants can attend a rich program with renowned speakers, technical sessions, a student design competition, exhibits, networking, and social activities, as well as a great opportunity for students to interact with leading industry experts. Our 2017 conference had 267 attendees, including several internationals and students from across the US, with a technical paper acceptance rate of 38%.

The conference theme is "Meet Innovative Technology," and the eight technical tracks include:

1. Machine Learning and Artificial Intelligence
2. Security and Communications
3. Human-Computer Interaction and Graphics
4. Robotics and Controls
5. BioEECS and Applied Physics
6. Computer Systems
7. Circuits, Materials, and Nanotechnologies
8. Theoretical Computer Science and Mathematics

Authors may submit content in the form of a technical paper, poster, or lightning talk.

All submissions must be written in English. Paper submissions must be no longer than 4 pages, single-spaced, with a minimum of 10 point font, and submissions may include figures, illustrations, and graphs. Abstract submissions for the poster and lightning talks are limited to 500 words.

All submissions will be peer-reviewed by faculty, graduate students, and industry professionals. Submissions are online, with a deadline of June 30th, 2018. Notification of acceptance will be sent via email by August 4th, 2018.

Please join the mailing list (MIT-Conference@ieee.org) for more information and updates on submission, the technical program, registration, and accommodation.

A conference proceeding of all the accepted papers that have been presented at the conference may be published and included in the IEEE Xplore journal. Electronic and online media containing all accepted submissions will be distributed to all registered attendees.

MEET INNOVATIVE TECHNOLOGY

Sponsored by the MIT IEEE Student Branch and IEEE Boston Section

http://ieee.scripts.mit.edu/conference

Call for Articles

Now that the Reflector is all electronic, we are expanding the content the publication. One of the new features we will be adding are technical and professional development articles of interest to our members and the local technology community. These will supplement the existing material already in our publication.

Technical submissions should be of reasonable technical depth and include graphics and, if needed, any supporting files. The length is flexible; however, a four to five page limit should be used as a guide. An appropriate guide may be a technical paper in a conference proceeding rather than one in an IEEE journal or transaction.

Professional development articles should have broad applicability to the engineering community and should not explicitly promote services for which a fee or payment is required. A maximum length of two to three pages would be best.

To ensure quality, technical submissions will be reviewed by the appropriate technical area(s). Professional articles will be reviewed by the publications committee for suitability.The author will be notified of the reviewers' decision.

The Reflector is published the first of each month. The target submission deadline for the articles should be five weeks before the issue date (e.g., June 1st issue date; article submission is April 27). This will allow sufficient time for a thorough review and notification to the author.

We are excited about this new feature and hope you are eager to participate!

Submissions should be sent to; ieeebostonsection@gmail.com

Call for Course Speakers/Organizers

IEEE's core purpose is to foster technological innovation and excellence for the benefit of humanity. The IEEE Boston Section, its dedicated volunteers, and over 8,500 members are committed to fulfilling this core purpose to the local technology community through chapter meetings, conferences, continuing education short courses, and professional and educational activities.

Twice each year a committee of local IEEE volunteers meet to consider course topics for its continuing education program. This committee is comprised of practicing engineers in various technical disciplines. In an effort to expand these course topics for our members and the local technical community at large, the committee is publicizing this CALL FOR COURSE SPEAKERS AND ORGANIZERS.

The Boston Section is one of the largest and most technically divers sections of the IEEE. We have over 20 active chapters and affinity groups.

If you have an expertise that you feel might be of interest to our members, please submit that to our online course proposal form on the section's website (www.ieeeboston.org) and click on the course proposal link (direct course proposal form link is **http://ieeeboston.org/course-proposals/** . Alternatively, you may contact the IEEE Boston Section office at ieeebostonsection@gmail.com or 781 245 5405.

- Honoraria can be considered for course lecturers
- Applications oriented, practical focused courses are best (all courses should help attendees expand their knowledge based and help them do their job better after completing a course
- Courses should be no more than 2 full days, or 18 hours for a multi-evening course
- Your course will be publicized to over 10,000 local engineers
- You will be providing a valuable service to your profession
- Previous lecturers include: Dr. Eli Brookner, Dr. Steven Best, Colin Brench, to name a few.

IEEE Advancing Technology for Humanity

CALL FOR PAPERS

2019 IEEE International Symposium on

Phased Array Systems and Technology

Revolutionary Developments in Phased Arrays

BOSTON 2019

Sponsors

Platinum

Gold

Silver

Bronze

Other Sponsors

15–18 October 2019

Westin Waltham Hotel, Greater Boston, Massachusetts, USA

www.array2019.org

About the Symposium

Phased array systems continue to be a rapidly evolving technology with steady advances motivated by the challenges presented to modern military and commercial applications. This symposium will present the most recent advances in phased array technology and provide a unique opportunity for members of the international community to interact with colleagues in the field of Phased Array Systems and Technology.

Suggested Topics

- System Architecture
- Aperture Design
- Antenna Elements
- Beamforming Techniques
- T/R Modules
- Signal Processing for Arrays
- Array Measurements
- Advanced Materials
- Packaging and Manufacturing Techniques
- Applied Computational Electromagnetics
- 5 G
- Metamaterials
- Radio Astronomy

See website: www.array2019.org for details

Special Sessions

Please provide suggestions for special sessions to the Technical Program Chair at info@array2019.org

Paper Template and Submission Procedures

Template and submission procedures are available at www.array2019.org/call-for-papers

General Paper Submission Information

All paper submissions will be peer reviewed and must be received in PDF format via the symposium web site on or before Friday, 15 March 2019.This is a firm deadline. Papers will not be accepted after this date. Papers must be in IEEE dual-column format and must be 2 pages (minimum) to 8 pages (maximum) in length including figures. Additional instructions are on the website www.array2019.org/call-for-papers

Technical Program Schedule

Please note: Our submission process and dates have been streamlined – plan accordingly.

15 March 2019 – Full paper submission deadline

- Submitted paper must be final and in IEEE dual-column format, not an abstract
- Submitted paper must be 2-8 pages in length including figures

30 April 2019 – Author notification of paper acceptance

1 Sept. 2019 – Conference registration deadline for accepted authors

Conference Committee

Conference Chair:
Jeffrey S. Herd, MIT LL

Vice Chair:
William Weedon, Applied Radar

Technical Program Chair:
Alan J. Fenn, MIT LL

Technical Program Vice Chair:
David Mooradd, MIT LL

Honorary Chair:
Eli Brookner, Raytheon (retired)

Secretary:
Duane J. Matthiesen, Technia

Publicity:
Glenn Meurer, MITRE
Don McPherson, SRC, Inc.

Publications:
Raoul Ouedraogo, MIT LL
Mark McClure, STR

Plenary Session:
Bradley T. Perry, MIT LL
Eli Brookner, Raytheon (retired)

Tutorials:
Will Moulder, MIT LL
Wajih Elsallal, MITRE

Student Program:
Justin Kasemodel, Raytheon
Honglei Chen, The MathWorks

International Liaison:
Alfonso Farina, Selex (retired)
Alberto Moreira

Sponsorships
Dan Culkin, NGC

Social Media:
Andrew Zai, Humatics Inc.

Special Sessions:
Sean Duffy, MIT LL

Web page:
Kathleen Ballos, Ballos Associates, Kenneth E. Kolodziej, MIT LL

5G Advisors:
Jonathan Williams, Teradyne Tony Fischetti, MACOM

Exhibits Chair:
Matt Facchine, NGC

Local Arrangements/Finance:
Robert Alongi, IEEE Boston

Poster Session:
Pierre Dufilie, MIT LL

Advisors:
Greg Charvat, Humatics Inc.,
Greg Arlow, Lockheed Martin

v.1.4

Technical Co-Sponsors

MTT-S

AP S

AESS

Our Blog

Call for Papers

Radar Conference 2019

Boston

IEEE

Revolutions in Radar

Sponsors

Platinum

NORTHROP GRUMMAN

Raytheon

LOCKHEED MARTIN

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Georgia Tech Research Institute

Silver

ARRC

About the Conference

22-26 April 2019

Westin Waterfront Hotel
Boston, Massachusetts, USA
www.radarconf19.org

A radar revolution is underway, made possible by the rapid evolution of digital electronics, and powered by new innovative architectures, advanced components, novel waveforms and sophisticated processing techniques. Please join us in historic Boston, birthplace of the original American Revolution, as we continue this new revolution in radar technology. The beautiful Westin Waterfront hotel, located in the heart of Boston's seaport district, will make the perfect venue for the international community as we meet to share the latest advances in radar. The conference will include three days of parallel technical sessions, and two days of tutorials with ample opportunity to interact with international radar experts from around the world.

Tutorials and Special Sessions

Please submit suggestions for tutorial topics or special sessions to the Technical Program Chair at info@radarconf2019.org

Suggested Topics

- Radar phenomenology
- Antenna technology
- Radar Electronics
- Over the horizon radar (OTHR)
- Bistatic, multistatic & passive radar
- Networked & distributed radar
- Commercial radar
- mm-wave & THz radar
- Environmental Sensing
- Airborne & space based
- SAR and ISAR imaging
- ATR and classification
- Tracking
- Cognitive methods
- Waveform diversity
- Spectrum sharing
- Electronic warfare
- Emerging Systems and Technology

Paper Submission Procedures

See website for details

Important Dates	
Special Session Proposals Due	17 Aug 2018
Tutorial Submissions Due	30 Aug 2018
Paper Submissions Due	17 Oct 2018
Notification of Acceptance	14 Jan 2019
Paper Submission Due	25 Feb 2019

Organizing Committee

General Chair
Eric Evans
MIT Lincoln Laboratory (MIT LL)

Deputy Chairs
Mark Russell, Raytheon
Eric Reinke, Northrup Grumman
Richard Buck, Lockheed Martin

Vice Chair
Jennifer Watson, MIT LL

Technical Chair
Dan Rabideau, MIT LL

Technical Vice Chair
Dan Bliss, Arizona State University

Exhibits
Pamela Evans, MIT LL

International
Alfonso Farina, Selex (ret.)
Hugh Griffiths, U. College of London

Plenary
Eli Brookner, Raytheon (ret.)
Frank Robey, MIT LL

Publications
Jeffrey Herd, MIT LL
Vito Mecca, MIT LL

Publicity
Mabel Ramirez, MIT LL
David Mooradd, MIT LL

Tutorials
James Ward, MIT LL
Katherine Rink, MIT LL

Student Program
Julie Jackson, AFIT

Sponsorships Chair
Jonathan Towle, Raytheon

Business Manager:
Robert Alongi, IEEE Boston

Website
Andrew Zai

Draft 02

Fundamentals of Real-Time Operating Systems (*Online Edition*)

Students have access to this self-paced course for 90 days!!

Registration Fee: $350

Course Summary - This course introduces the basics of Real-Time Operating Systems (RTOSes) using VxWorks and Linux as examples. The course focuses on the primary principles of RTOSes including determinism, real-time scheduling, interrupt latency and fast context switching as well as time and space partitioning in hard real-time environments. The first part of the course focuses on acquiring an understanding of microkernel and memory architectures for Real-Time including scheduling, signals, system calls, synchronization, inter-process communications and interrupt handling. The latter part of the course covers considerations for timing, memory management, device drivers, booting, debugging and deployment of Real-Time embedded systems.

Who Should Attend - The course is designed for real-time engineers who are using or intending to use a Real-Time Operating System. It is also targeted at experienced developers requiring a refresher course on RTOSes. This course will clearly demonstrate both the strengths and weaknesses of Real-Time Operating Systems used in Embedded Systems.

Course Objectives

- To provide a basic understanding of Real-Time Requirements and Design Decisions
- To understand the complexities of RTOS scheduling and synchronization
- To learn how to configure, boot, test and deploy Real-Time embedded systems
- To give students the confidence to apply these concepts to their next RTOS project

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. He has held a variety of software engineering positions at LynuxWorks, Embedded Planet, Wind River Systems and Lockheed Sanders. RTETC, LLC provides Real-Time embedded training and consulting to many embedded systems companies. RTETC focuses on Real-Time operating systems (RTOS), Linux and Android solutions for the embedded systems market.

Hardware and Software Requirements - The student should have a working Linux desktop environment either directly installed or in a virtualization environment or have access to a development environment for a Real-Time Operating System such as VxWorks. An Embedded Linux or VxWorks target hardware platform is useful but not absolutely required for this course.

Embedded Development Basics

Embedded Systems Characteristics
Embedded Real-Time Systems
Real-Time Enough
Embedded Linux and Real-Time

Real-Time Operating System Basics

Microkernel Architecture
Microkernel Scheduling
Determinism
Rate Monotonic Analysis and Fixed Priority Scheduling
Latency and Latency Measurements
Fast Context Switching
Real-Time Memory Architectures
Time and Space Partitioning and ARINC
Multiprocessor Basics
Amdahl's Law

RTOS Kernel Overview

Real-Time Scheduling and Task Management
Signals and System Calls
Synchronization

Inter-Process Communications (IPC)
Interrupt Handling
Error Handling
Timing and Timers
Real-Time Memory Management

Real-Time Scheduling
OS Scheduling Types
Pre-emptive Multitasking
Typical Scheduling Issues
Linux Scheduling
VxWorks Scheduling
VxWorks Tasks
VxWorks Real-Time Processes (RTPs)
Linux Threads
Task and Thread-Specific Data (TSD)
Measuring Task and Thread Performance

Signals in Embedded RTOSes

System Calls in Embedded RTOSes

Synchronization
Via Global Data
Via Semaphores, Files and Signals
Condition and Completion Variables in Linux
Mutexes in VxWorks and Linux
Linux Futexes
Software Watchdog Timers

Inter-Process Communications (IPC)
More Semaphores
Message Queues
Shared Memory
Pipes and FIFOs
Remote Procedure Calls
Networking

Interrupts and Exception Handling
Basic Interrupt Process
VxWorks intLib and excLib
Routines You Can Call From Interrupt Context

Interrupt Service Routines (ISRs)
VxWorks and Linux ISRs

Bottom Halves in Linux

Deferring Work
Tasklets and Work Queues in Linux
Helper Tasks

Error Handling
Error Handling Approaches in OS Design

Error Handling in VxWorks
Error Handling in Tasks and Interrupts
Error Number Format
Using errnoSet, errnoGet and printErrno
Creating Your Own Errors

Error Handling in Linux
Standard Error Defines
errno and perror
strerror and strerror_r
Resets, OOPS, Panics and Segmentation Faults

Error Logging Approaches

Timing and Timers
How RTOSes Tell Time
VxWorks tickLib and timerLib
Linux Kernel, POSIX and Interval Timers
Linux High-Resolution Timers (HRTs)
VxWorks taskDelay
Linux Sleeping, Sleep Waiting and Spinlocks
VxWorks Watchdog Timers (wdLib)
Periodic Execution Example
Deadline Miss Detection
Embedded Recommendations for Timing

Memory Management and Paging
The VxWorks Memory Model
Real-Time Memory Algorithms
VxWorks memLib and memPartLib
Linux, Memory and Demand Paging
Mapping Device Memory in Linux
The Slab Allocators in Linux
The Linux /proc Filesystem
Memory Barriers
The Linux OOM Killer
Reserving and Locking Memory
Memory in Embedded Systems

Device Drivers in VxWorks
File Descriptors
The VxWorks IO Subsystem
VxWorks ioLib, fioLib and iosLib

The 5 Basic Device Driver Types
Char, Block and Network Drivers
Virtual Drivers and Basic I/O Drivers
Other Device Drivers
The VxBus in VxWorks

Device Drivers in Linux
File Descriptors in Linux
The UNIX Device Driver Model
Major and Minor Numbers
The New Device Driver Model

The VxWorks Boot Process
VxWorks Boot Example
Configuration Files
Application Startup
VxWorks 7

The Linux Boot Process
The Root Filesystem in Linux
Bootloaders and U-Boot
Configuring Linux
Embedded Linux Boot Methods
Building and Booting from SD Cards and eMMC
Yocto and Poky

Debugging Basics
How Software Debuggers Work
Debuggers and Intrusion
Types of Debugging Approaches
Process-Level vs System-Level Debug

Process-Level Debug
GDB, GDB Server and the GDB Server Debugger
The VxWorks Debug Agent (WDB)
Other Debug Tools for Linux and VxWorks
A Remote Debug Example
Printing and Logging

System-Level Debug
LTTng and the VxWorks Systems Viewer (Windview)
System-Level Debug Tools
The /proc and /sys Virtual Filesystems in Linux
Linux Kernel Debug
Linux Crash Dumps

Deploying VxWorks Systems
VxWorks Systems Customization and Configuration
VxWorks Field Upgrades

Deploying Embedded Linux Systems
Linux Systems Customization and Configuration
Choosing and Building the Root Filesystem
Module Decisions
Final IT Work
Final deployment of Embedded Linux Field Systems

RTOS Trends

Some Final Recommendations
Getting Help
Measuring Performance
Managing Memory
Things To Remember

http://ieeeboston.org/fundamentals-of-real-time-operating-systems-rt201-on-line-course/

IEEE Boston Section Social Media Links:

Twitter: https://twitter.com/ieeeboston
Facebook: https://www.facebook.com/IEEEBoston
YouTube: https://www.youtube.com/user/IEEEBostonSection
Google+: https://plus.google.com/107894868975229024384/
LinkedIn: https://www.linkedin.com/groups/IEEE-Boston-Section-3763694/about

Embedded Linux Board Support Packages and Device Drivers (*Online Edition*)

Students have access to this self-paced course for 90 days!!

Registration Fee: $350

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.

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, 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 Schedule

Getting Started with Embedded Linux

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
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
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

http://ieeeboston.org/embedded-linux-bsps-device-drivers-line-course/

Embedded Linux Optimization - Tools and Techniques (*Online Edition*)

Students have access to this self-paced course for 90 days!!

Registration fee: $250

Summary - This video course provides advanced training in the debugging, testing, profiling and performance optimization of Embedded Linux software. The first part of the course focuses on advanced debugging, testing and profiling in an Embedded Linux context with a focus on using Eclipse, Backend Debuggers, JTAG and In-Circuit Emulators as well as Kernel Logging capabilities and Kernel Hacking. The latter part of the course covers performance measurement and optimization affecting boot, memory, I/O and CPU performance and key performance optimization tools for Embedded Linux software including the perf tool, advanced cache usage and compiler-based optimization.

Who Should Attend - The course is designed for real-time engineers who are developing high-performance Linux applications and device drivers using Embedded Linux distributions. It is also targeted at experienced developers requiring a refresher course on Advanced Embedded Linux optimization.

Course Objectives

- To understand debugging, profiling and testing high performance Embedded Linux software.
- To provide an overview of Linux application performance measurement and optimization.
- To understand the tools used for performance optimization of Embedded Linux software.
- To give students the confidence to apply these concepts to their next Embedded Linux project.

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. He has held a variety of software engineering positions at LynuxWorks, Embedded Planet, Wind River Systems and Lockheed Sanders. RTETC, LLC provides real-time embedded training and consulting to many embedded systems companies. RTETC focuses on real-time operating systems (RTOS), Linux and Android solutions for the embedded systems market.

Getting Started with Embedded Linux
Embedded Linux Training Overview
Terminology
Linux Versioning
The GPL
Building the Kernel Source Code
Embedded Linux Kernels
BSPs and SDKs
Linux References (Books and Online)
A Development Cycle Focused on Performance
A Basic Optimization Process

Basic Debugging Review
Embedded Applications Debug
GDB, GDB Server and the GDB Server Debugger
Other Debuggers
An Eclipse Remote Debug Example
Debugging with printk, syslog, syslogd and LTTng

System-Level Debug
System-Level Debug Tools
The /proc and /sys Filesystems

Basic Logging
KDB and KGDB
Crash Dumps and Post-Mortem Debugging
Debugging Embedded Linux Systems
Backend Debuggers
In-Circuit Emulators
Hardware Simulators
Analyzers
Requirements Development
Performance Requirements
Derived Requirements
Testability and Traceability
Reviewing Requirements
Designing for Performance
Design for Test (DFT)
Agile Software Design
Software and Linux Decomposition
Memory Management
CPU and OS Partitioning
Design Reviews
Coding for Performance
Coding Standards and Consistency
Languages, Libraries and Open Source Components
Learning Magic Numbers
Letting Compilers Work For You
Global, Static and Local Variables
Code Reviews

Software Testing
Unit-Level Testing
System-Level Testing
Code Coverage Tools
gcov
Automated Testing
Some Embedded Linux Test Recommendations
DebugFS
Configuring DebugFS
DebugFS Capabilities
Advanced Logging
LogFS
Using Logwatch and Swatch
Using syslogd and syslog-ng
Tracing
ptrace and strace
New Tracing Methods
SystemTap
Ftrace, Tracepoints and Event Tracing
Tracehooks and utrace
Profiling
Basic Profiling
gprof and Oprofile
Performance Counters
LTTng
Another DDD Example
Manual Profiling
Instrumenting Code
Output Profiling
Timestamping
Addressing Performance Problems
Types of Performance Problems
Using Performance Tools to Find Areas for Improvement
Application and System Optimization
CPU Usage Optimization
Memory Usage Optimization
Disk I/O and Filesystem Usage Optimization
Measuring Embedded Linux Performance
Some Ideas on Performance Measurement
Common Considerations
Uncommon Considerations
Using JTAG Methods
BootLoader Measurements
Boot Time Measurements
The Perf Tool
Origins of Perf
The Perf Framework
Perf Commands and Using Perf
Listing Events
Counting Events
Profiling with Perf
Static Tracing with Perf
Dynamic Tracing with Perf
Perf Reporting
Performance Tool Assistance
Recording Commands and Performance
System Error Messages and Event Logging
Dynamic Probes
Jprobes and Return Probes
Kernel Probes

Kexec and Kdump
- Improving Boot Performance
- Boot Time Optimization
- The Linux Fastboot Capability
- Building a Smaller Linux
- Building a Smaller Application
- Filesystem Tips and Tricks
- Some Notes on Library Usage
- Improving Kernel Performance

Kernel Hacking
CONFIG_EMBEDDED
Configuring printk
Test Code
Configuring Kernel and IO Scheduling

- Improving CPU Performance

Run Queue Statistics
Context Switches and Interrupts
CPU Utilization
Linux Performance Tools for CPU
Process-Specific CPU Performance Tools
Stupid Cache Tricks
- Improving System Memory Performance

Memory Performance Statistics
Linux Performance Tools for Memory
Process-Specific Memory Performance Tools
More Stupid Cache Tricks
- Improving I/O and Device Driver Performance
- Disk, Flash and General File I/O
- Improving Overall Performance Using the Compiler
- Basic Compiler Optimizations
- Architecture-Dependent and Independent Optimization
- Code Modification Optimizations
- Feedback Based Optimization
- Application Resource Optimization
- The Hazard of Trust
- An Iterative Process for Optimization
- Improving Development Efficiency
- The Future of Linux Performance Tools

Some Final Recommendations

http://ieeeboston.org/embedded-linux-optimization-tools-techniques-line-course/

IEEE Boston Section Social Media Links:

Twitter: https://twitter.com/ieeeboston

Facebook: https://www.facebook.com/IEEEBoston

YouTube: https://www.youtube.com/user/IEEEBostonSection

Google+: https://plus.google.com/107894868975229024384/

LinkedIn: https://www.linkedin.com/groups/IEEE-Boston-Section-3763694/about

Software Development for Medical Device Manufacturers *(Online Edition)*

Students have access to this self-paced course for 90 days!! ***Registration Fee : $125***

Course Description This course provides an introduction to the development of medical device software. The course is comprised of 4 modules that range from 30-45 minutes in duration. The focus is on complying with FDA Design Controls and IEC 62304 requirements.

This course is intended for software developers who are actively involved in developing medical device software.

Module 1

- Medical Device Definitions: FDA and European Union (EU)
- Regulatory Roadmap
- FDA/EU Device Classifications
- FDA QSR Regulation
- FDA Guidance Documents that pertain to medical device software

Module 2

- International Standards that pertain to medical device software
- Types of Software Regulated by FDA
- Quality System basics: Procedures, Work Instructions and Records
- ALL Software is Defective…

Module 3:

- Design Control Overview
- General Requirements
- Design and Development Planning
- Software Development Models
- Design Input
- About Requirements…
- Design Output
- Design Reviews

Module 4:

- Design Control (continued)
- Design Verification
- Software Verification Process
- Testing Overview
- Design Validation
- Software Validation Process
- Design Changes
- Design Transfer
- Design History File
- Course Summary

Speaker Bio:

Steven R. Rakitin has over 40 years experience as a software engineer including 25 years of experience in the medical device industry. He has worked with over 85 medical device manufacturers worldwide, from startups to Fortune 100 corporations. He has written several papers on medical device software risk management as well as a book titled: Software Verification & Validation for Practitioners and Managers.

He received a BSEE from Northeastern University and an MSCS from Rensselaer Polytechnic Institute. He earned certifications from the American Society for Quality (ASQ) as a Software Quality Engineer (CSQE) and Quality Auditor (CQA). He is a Senior Life member of IEEE and a member of MassMEDIC. He is on the Editorial Review Board for the ASQ Journal Software Quality Professional.

As President of Software Quality Consulting Inc., he helps medical device companies comply with FDA regulations, guidance documents, and international standards in an efficient and cost-effective manner.

http://ieeeboston.org/software-development-medical-device-manufacturers-line-course/

Fundamental Mathematics Concepts Relating to Electromagnetics (*Online Edition)*

Students have access to this self-paced course for 90 days!!

Registration Fee: $150

Course Summary This course is designed for people wishing to refresh or to learn the fundamental mathematical concepts that are used to describe electromagnetic wave behavior. The modules address all of the basic math concepts covered in a traditional undergraduate electromagnetics course in an ECE curriculum. These concepts include Vector Basics, Integral Vector Calculus, Differential Vector Calculus, Fundamental Coordinate Systems and Complex Numbers. After completing these modules, a person should have sufficient math skills to pursue graduate studies in electromagnetics and/or be able to decipher the math presented in an upper-level text on the subject.

Target audience: This course is designed for people wishing to refresh or to learn the fundamental mathematical concepts that are used to describe electromagnetic wave behavior.

Course chapters

1. Vector Basics
2. Dot Product
3. Cross Product
4. Contour Integration
5. Vector Algebra
6. Surface Integration
7. Metric Coefficients
8. Coordinate Systems
9. Vector Coordinate Conversion
10. Del Operator and the Gradient
11. The Curl
12. Divergence
13. Stokes Theorem
14. Divergence Theorem
15. Laplacian
16. Complex Numbers

Instructor's Bio:

Dr. Kent Chamberlin is the Chair and a Professor in the Department of Electrical and Computer Engineering. In his more than thirty-five years in academia, he has performed research for more than twenty sponsors, including the National Science Foundation. He has received two Fulbright awards, including the prestigious Fulbright Distinguished Chair, which he served in Aveiro, Portugal. He has also served as an Associate Editor for the Institute for Electrical and Electronics Engineers, and he continues to be active in performing and publishing in a range of research areas.

http://ieeeboston.org/fundamental-mathematics-concepts-relating-electromagnetics-line-course/

Reliability Engineering for the Business World *(Online Edition)*

Students have access to this self-paced course for 90 days!!

Registration Fee: $320

Course Description

This course is about becoming a leader in reliability engineering. While statistics are the tools of reliability engineering, it takes knowledge not only of these tools but also of the business. Developing knowledge of the business, from sales, engineering, customer service, to supply chain management can determine how effective you can be in improving reliability.

Never take anything for granted, even some rules of thumb in reliability can be misleading, this course will show you how to prove what truly happens in the real world and how to effect change in any part of the business where it is needed. We will explore the balance sheet, organizational structure, customers, service, and high volume manufacturing. It's not just about how often things fail, it is also about where the defect came from, what is the financial effect, the recovery, when should a business take field action, effect of human error, failure analysis/material science, reliability testing, and much more. I will also discuss how you develop executive buy in for change. The course assumes a basic knowledge in reliability statistics. There are 12 sessions that cover the following topics.

Course Outline

Basics – Measurements
Business Model
Design Model (HW and SW)
HALT/RDT/Predictions
Manufacturing Model
Early Life Failures
Wear Out and Mid Life Crisis
Advanced Reliability

Course Objective

To teach you how to become the go to person in your business for objective business sensed reliability answers and requirements.

Instructor's Bio

Kevin is an innovative leader in reliability methodologies with more than 30 years experience in the storage industry. In his latest role as Director of Engineering, he developed a top down reliability/availability management process for design organizations developing mission-critical storage systems. Kevin previously directed the most extensive HALT/HASS operation in the industry, with over 300 chambers worldwide. He has written several papers, consulted with many companies, 3 patents awarded and 2 pending related to systems reliability and test.

His most recent work has been performing system architectural analysis to optimize system availability, serviceability and costs. Providing guidance to development to maximize system reliability and reduce service costs. He has provided consultation to many large companies such as EMC, CISCO, AT+T, HP, Seagate and many others. His position and experience has enabled him to perform extensive field studies and design of experiments. Kevin has developed many

http://ieeeboston.org/reliability-engineering-business-world-line-course/

Introduction to Embedded Linux
(Online Edition)

Students have access to this self-paced course for 90 days!! ***Registration Fee: $350***

Course Summary:
This first of a 2-part series introduces the Linux Operating System and the use of Embedded Linux Distributions. The course focuses on the development and creation of applications in an Embedded Linux context using the Eclipse IDE. The first part of the course focuses on acquiring an understanding of the basic Linux Operating System, highlighting areas of concern for Embedded Linux applications development using Eclipse. The latter part covers the methods for booting Embedded Linux distributions including embedded cross-development and target board considerations.

Who Should Attend:
The course is designed for real-time engineers who are building Embedded Linux solutions. It is also targeted at experienced developers requiring a refresher course on Embedded Linux. This course will clearly demonstrate both the strengths and weaknesses of the Linux Operating System in Embedded Systems.

Course Objectives:
To provide a basic understanding of the Linux OS and the Eclipse IDE framework.
To gain an understanding of the complexities of Embedded Linux Distributions and their use in embedded systems.
To give students confidence to apply these concepts to their next Embedded Linux project
Hardware and Software Requirements

The student should have a working Linux desktop environment either directly installed or in a virtualization environment. The desktop Linux should have the GNU compiler and binary utilities (binutils) already installed. A working Eclipse C/C++ installation or prior knowledge of C-based Makefiles is useful for completion of lab exercises. Lab solutions are also provided with the course. An Embedded Linux target hardware platform is useful but not absolutely required for this course.

Additional Reference Materials
Linux Kernel Development by Robert Love
Linux System Programming by Robert Love
Linux Debugging and Performance Tuning by Steve Best
Optimizing Linux Performance by Phillip G. Ezolt
Embedded Linux Primer by Christopher Hallinan
Pro Linux Embedded Systems by Gene Sally
Embedded Linux Development Using Eclipse by Doug Abbott
Linux Device Drivers by Jonathan Corbet et al
Essential Linux Device Drivers by Sreekrishnan Venkateswaran
Course Downloadable Content:

Video Lecture
Hands-On Lab Instructions
Hands-On Lab Solutions
Additional Related Materials

The Basics
Linux Terminology, History and Versioning
The Linux Community: Desktop & Embedded
The GPL
Linux References (Books and Online)

Getting Started
Kernel Source Code
Building the Kernel
Embedded Linux Kernels
Linux 2.6

Basic Kernel Capabilities
Process and Threads Management
Signals and System Calls
Synchronization, IPC and Error Handling
Timing and Timers

Memory Management and Paging
The I/O Subsystem: A Tale of Two Models
Modularization

Debugging
Process-Level and System-Level Debug
GDB and KGDB
GDB Server and Remote Debugging

An Eclipse Debug Example
Other Debug and Test Tools
Other System-Level Debug Approaches
Process & Threads Management
What are Processes and Threads?
Virtual Memory Mapping
Creating and Managing Processes and Threads
Thread-Specific Data (TSD) POSIX
The Native POSIX Threading Library (NPTL)
Kernel Threads

Signals
System Calls
Scheduling
Linux 2.4 and 2.6 Scheduling Models
The O(1) Scheduler
The Completely Fair Scheduler (CFS)

Synchronization
Via Global Data
Via Semaphores, Files and Signals

Inter-Process Communications (IPC)
Message Queues
Semaphores Revisited
Shared Memory
Pipes, FIFOs and Futexes
Remote Procedure Calls
Networking

Error Handling
errno and perror
strerror and strerror_r
oops, panics and Segmentation Faults
Timing
How Linux Tells Time
Kernel, POSIX and Interval Timers
High-Resolution Timers (HRTs)

Memory Management and Paging
Demand Paging and Virtual Memory
Allocating User and Kernel Memory
Mapping Device Memory
The Slab Allocator
The OOM Killer
Memory in Embedded Systems

Modularization
Creating a Module and Module Loading
Dependency Issues
In Embedded Systems

Shared Libraries
A Shared Library Example
Static and Dynamic Libraries

The I/O Subsystem: A Tale of Two Models
The Original Device Driver Model
The Standard I/O Interface
The New Device Driver Model and Kernel Object Classes
Initialization
Platform Devices, Busses, Adapters and Drivers
Comparing the Two Models

Embedded Linux Trends
Development, Monitoring and Testing

Some Final Recommendations

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 Tilera, Embedded Planet, Wind River Systems, Lockheed Sanders, Stratus Computer and Apollo Computer. RTETC, LLC is a provider of Eclipse-based development tools, training and consulting for the embedded systems market.

http://ieeeboston.org/introduction-to-embedded-linux-part-i-el201-online-course/

Design Thinking for Today's Technical Work

New Online Course

Students have access to this self-paced course for 90 days!!

Registration Fee: $160

Course Description:

This course covers the principles of Design Thinking; the steps commonly used; how it enhances the likelihood of success in a wide variety of applications; and, in particular, how to apply it to technical work. Examples of its application to technical work are presented along with the successes that followed.

Design Thinking has garnered much attention in recent years mainly as a way to design consumer products that engage users, such as Apple's iPhone. But its use is spreading to situations ranging from how to provide medical care to planning one's career. This course explains what Design Thinking is about, but, most important, explains how an individual can apply Design Thinking to their own technical work. Care has been taken to focus the course content on using Design Thinking as a structured, practical process for the daily work of technical professionals. A specific technical example is carried through the teaching of the five stages of Design Thinking. The course covers applying Design Thinking to the range of tasks performed during a technical project, including design of: technical functions; user interactions (if applicable); factors for business success; solutions to problems that arise; and project presentations and reports to influence adoption of project outcomes, funding approval, and hiring for consulting. The content applies to employees of large to small companies, start-ups, consultants and contact work, and government organizations. The course is focused on an individual worker employing Design Thinking.

Course Objectives

Provide an understanding of Design Thinking and how an individual can apply it to their technical work:

- Understand the steps of Design Thinking (Understand, Define, Ideate, Prototype, and Test)
- Learn how to apply Design Thinking in technical work
- Understand where Design Thinking can be applied in project activities.

Who Would Benefit from this Course

Anyone who works on solutions to problems or designs hardware, software, products, services, and processes. This includes technical professionals, project managers, and organizational managers. Also, anyone who wants to learn what Design Thinking is about in a practical sense.

Course Modules

- Module 1 – How Design Thinking Can Help Technical Work (60 minutes)
- Module 2 – Understand: Explore the Problem (44 minutes)
- Module 3 – Define: Synthesize What Is Needed (23 minutes)
- Module 4 – Ideate: Generate Solutions (26 minutes)
- Module 5 – Prototype: Build Versions to Test (23 minutes)
- Module 6 – Test: Examine and Learn (28 minutes)
- Module 7 – Design Thinking for Presenting and Writing (23 minutes)

• Module 8 – Getting Started with Design Thinking (30 minutes)

Speaker biography

Speaker: James L. Poage, President/Owner JLP Performance Consulting

Dr. James L. Poage has been designing future concepts for Air Traffic Control for 25 years, first with the Volpe National Transportation Systems Center and then for the past dozen years as an independent consultant (JLP Performance Consulting). He has taught short courses on Benefit-Cost analysis to the FAA and NASA, as well as spoken at conferences and published in professional journals. Over the past 15 years, Dr. Poage has been applying Design Thinking to his project work; to marketing his consulting services; and to planning briefings, reports, and courses. His clients have included FAA, NASA, BAE Systems, Engility, Georgia Tech University, San Jose State University, and Saab Sensis. Dr. Poage has co-authored the book, Flair: Design Your Daily Work, Products, and Services to Energize Customers, Colleagues, and Audiences (Maven House Press, 2016), with his daughter, Jennifer Poage who works in fashion design. Dr. Poage has a Ph.D. in applied mathematics from the Harvard University School of Engineering and Applied Sciences and a M.S. and B.S. in electrical engineering from Stanford University.

Note: Course participants will receive a copy of the book, Flair.

Call for Course Speakers/Organizers

IEEE's core purpose is to foster technological innovation and excellence for the benefit of humanity. The IEEE Boston Section, its dedicated volunteers, and over 8,500 members are committed to fulfilling this core purpose to the local technology community through chapter meetings, conferences, continuing education short courses, and professional and educational activities.

Twice each year a committee of local IEEE volunteers meet to consider course topics for its continuing education program. This committee is comprised of practicing engineers in various technical disciplines. In an effort to expand these course topics for our members and the local technical community at large, the committee is publicizing this CALL FOR COURSE SPEAKERS AND ORGANIZERS.

The Boston Section is one of the largest and most technically divers sections of the IEEE. We have over 20 active chapters and affinity groups.

If you have an expertise that you feel might be of interest to our members, please submit that to our online course proposal form on the section's website (www.ieeeboston.org) and click on the course proposal link (direct course proposal form link is **http://ieeeboston.org/course-proposals/** . Alternatively, you may contact the IEEE Boston Section office at ieeebostonsection@gmail.com or 781 245 5405.

- Honoraria can be considered for course lecturers
- Applications oriented, practical focused courses are best (all courses should help attendees expand their knowledge based and help them do their job better after completing a course
- Courses should be no more than 2 full days, or 18 hours for a multi-evening course
- Your course will be publicized to over 10,000 local engineers
- You will be providing a valuable service to your profession
- Previous lecturers include: Dr. Eli Brookner, Dr. Steven Best, Colin Brench, to name a few.