Professional Master's Program (PMP) in Electrical Engineering

Course: **Modern introduction to Embedded & Real-Time Systems**
Length: 10-week, 1-quarter course

History:

2016-2-11: updated the course work | Andrew N. Sloss
2015-7-20: updated & added equipment estimate | Andrew N. Sloss
2015-7-15: clean up and improve document | Andrew N. Sloss
2015-7-12: update and restructured the document | Andrew N. Sloss
2015-7-7: started modifying the course content | Andrew N. Sloss

Course

GOAL

Introduce Embedded & Real-Time Systems

MESAUREMENT

Students will gain an appreciation of Embedded & Real-Time Systems through learning how to characterize hardware and software. This will involve being introduced to a specific hardware platform, exploring real-time requirements, computation limits, analysis of different scheduling algorithms, and the configuration of memory. In addition, students will be introduced to some harder subjects in Embedded & Real-Time Systems including power management, reliability, safety-critical systems and finally simulation. Upon completion of the program, students will have a firm understanding of real world issues and characterization methods and techniques for Embedded & Real-Time Systems.

Topics to be covered during the course: -

1. Introduction & Characterization
2. Fundamentals
3. Memory
4. Power management
5. Critical Systems I - Reliability
6. Critical Systems II - Safety-Critical
7. Simulation
8. Course Work – Hardware Characterization

NOTES

The course assumes the student understand Computer Science concepts including the C language and fundamental computer architecture. The overarching aim is to provide the students with a set of useful problem-solving tools found in the Embedded Systems Industry. As well as revisit the scientific method as it pertains to Embedded.

Each lesson will comprise an interactive portion where the students will be encourage to provide questions especially around Embedded Systems and characterization.

1. Introduction & Characterization

ESTIMATED LESSON 1

GOAL

Introduce hardware-software characterization. Explain how that characterization is used to define the requirements of a final Embedded & Real-Time System.

MEASUREMENT

Given a requirement the students should be able to provide the correct hardware-software configuration. This lesson will introduce Embedded & Real-time, basic system design, exceptions and handling exceptions (including interrupts), and the type of kernel/executives required for a specific solution. Later part of the lesson will further introduce the concept Real-Time and the different methods to determine whether a problem can be schedulable through Rate Monotonic (RMS) and Earliest Deadline First scheduling (EDF).

STRUCTURE

- Introduction to Embedded & Real-Time Systems, and the Characterization Theme
- Applications, Tasks, Threads and Processes
- Exceptions, Interrupts, Stacks, Context Switch, Ready-List and Scheduler
- Deterministic (Real-Time) and non-deterministic behavior
- System Level Characterization
- Scheduling algorithms
- Re-entrant/non re-entrant kernel
- Nested and non-nested interrupt handling
- Introduce scheduling algorithms
    1. Shortest-job-first
    2. First-come-first-serve
    3. Round-robin
    4. Priority-scheduling
- Introduce real-time scheduling design
    1. Ask the question is the system schedulable?
    2. Rate Monotonic scheduling

3. Earliest Deadline First Scheduling

2. Fundamentals

ESTIMATED LESSON 1

GOAL

Fundamentals lesson will explain two important areas in Embedded, namely the boot/initialization process and the constraints on development tools and environments for Embedded & Real-Time systems.

MEASUREMENT

Students will be introduced to a hardware boot flow for a generic hardware platform. This will involve going through the order and the specific pieces of hardware that are required to be initialized before an Operating System/Executive is functioning. On the software side students will be re-introduced to compilers, debugger and introduced to hardware-assist debugging/trace for Embedded & Real-Time development.

STRUCTURE

- Cover a complete boot process of a simple device
- Re-visit compilers, debuggers and architectures
- Cover compiler optimizations
- Hardware assisted tools: JTAG Probes, Hardware Trace, Logical Analyzers and Oscilloscopes

3. Memory

ESTIMATED LESSONS 2

GOAL

Provide an overview of the use of caches and introduce the different forms of memory management with-respect-to Embedded & Real-Time Systems. *Covering simple microcontrollers to the more advanced multicore systems.*

MEASUREMENT

Students will be introduced to the complexities of memory systems. After these two lectures they will have a basic understanding of caches and in particular how different cache policy effect the overall performance of a system. Intertwined is the need to understand the different memory management techniques - linear memory, protected memory and full virtual memory. Students will have a good understanding on the tradeoffs of using different memory management methods.

STRUCTURE

- Memory layout
- Caches Revisited
    - Cache lines
    - Cache lockdown
    - Virtual and Physical caches
- Memory Protection
    - Protection regions
- Memory Management Unit
    - TLB – software and hardware operations
    - Virtual Memory
    - Demand paging
    - Loading and unloading applications
    - Shared resources
- Messaging Passing
    - Mailboxes (revisited with connection with memory design)
- Direct Memory Access (DMA)
- Implications on tools

4. Power Management

ESTIMATED LESSONS 1

GOAL

Introduce the complexity of power management at the hardware level. Power management is particularly important for Internet-of-Things, Mobile devices as well as dormant tethered devices.

MEASUREMENT

Students will be introduced to power management as a subject. And how power management covers everything from tethered to untethered devices. Explain the difference between power and energy efficiency equations and why frequency is fundamentally important. Introduce mobile phone power demands, different techniques used to save power consumption and finally some battery characteristics.

STRUCTURE

- Introduce power management
    - Tether and non-tethered (mobile)
    - Power state machine

- Mobile Power Management
- Introduce the Power and Energy equations
- Processor features that help with power control
    - Voltage scaling
    - Frequency scaling
    - Wait-on-interrupt
- Peripherals
    - Ethernet
    - LCD
    - Hard drives
- Basic Battery Characteristics
    - Charging/discharging of batteries
    - History effects
- Power Transmission
- Power Harvesting
- Discussion of the complexity of Power Management

5. Critical Systems I: Reliability

ESTIMATED LESSONS 1

GOAL

Introduce reliability for Embedded & Real-Time Systems and explain the difficulties hardware aging incurs.

MEASUREMENT

Student will be introduced to reliability as a subject and the interesting area of why technology fails. The lesson will include describing Occam's razor, hardware aging and the fundamental reliability equations. Explain different methods of recovery once a deadlock has occurred. Student will be aware of different embedded designs that can help in reliability such as, voting, watchdog timers, and basic reset.

STRUCTURE

- Introduce subject of Reliability and relationship with Safety-Critical
- Explain Occam's Razor
- Hardware aging and the bathtub failure curve
- Discuss why systems fail?
- Define the terms SLOC, MTTF, MTTR, and MTBF
- Availability (Up time)
- Fault Tolerance

- Revisit interrupt handling
- Describe deadlocks
    - Deadlock identification
    - Deadlock recovery
    - Priority inversion
- Hardware assistance
    - Watchdog timers (re-visit)
- Basic Reset

6. Critical System II: Safety-Critical

ESTIMATED LESSONS 1

GOAL

Introduce the aspects of safety-critical for Embedded & Real-Time Systems.

MEASUREMENT

Students will be introduced to Safety-Critical and the consequences of failure. This includes having a comprehensive understanding of the philosophy behind safety-critical and the consequences of bypassing the philosophy. Understand the meaning of the important terms and the different methods to design a safety-critical system. The students will be introduced to one safety-critical standard DO-178B, as well as the MISRA C standard for safety-critical C writing style.

STRUCTURE

- Introduce the concept of safety-critical
- Describe security, integrity, vulnerability, threat and risk
- Complexity analysis
- Consistency checking
- Instrumenting code
- Writing code to check for stack overflow and underflow
- Redundancy (Byzantine algorithms)
- MISRA C Standard
- Example: safety-critical standard DO-178B

7. Simulation

ESTIMATED LESSONS 1

GOAL

Introduce the aspects of Simulation for Embedded & Real-Time Systems and how simulation can be used as an important tool for design and development.

MEASUREMENTS

Introduce and explore the concept of simulation. Including the differences between simulation and emulation of systems. Be able to define cycle accurate, cycle approximate and instruction level modeling. Define the advantages and scope of simulation and hence discuss the problems that simulation introduces.

STRUCTURE

- Introduce simulation modeling
    - Why?
    - How?
    - Cost?
- Different types of simulation models
    - Cycle accurate modeling
    - Cycle approximate modeling
    - Instruction level modeling
- Co-design/Co-simulation

8 Course work – Hardware Characterization

GOAL

Characterize and use a hardware platform for Embedded & Real-Time Development.

MEASUREMENT

Take a development platform initialize the board, characterize/explore its features and create your own project (and present your project findings). This includes setting up a boot image, initializing the Wi-Fi and installing telnet/SSH/Samba. Measuring interrupt latency, memory performance, code performance, communication requirements and potentially the ability to connect sensors and interrupts.

EQUIPMENT

- Raspberry Pi 2
- Mini SD Card 16GB
- Wi-Fi USB
- Power Supply