

ENEE759V
EMBEDDED SYSTEMS (graduate level)
Summer 2015

The first decade of 21st century was marked by the emergence of smart devices that are used in everyday life. Smart phones, smart cars, smart TV's, smart thermostats, smart vacuum cleaners, to name just a few. These developments are powered in large part by the embedded systems. This course will provide students with the essential knowledge base that will enable them to tackle complex problems encountered in embedded systems design. In addition to the overview of associated hardware components and software methodologies and tools used in the development of modern embedded systems, and theory behind them, the course will include a carefully selected collection of hands-on Lab exercises that would help students get a sense of how the presented theoretical concepts connect with the real-world embedded systems applications.

PREREQUISITES: Required – ENEE 350 or equivalent course on Computer Architecture, ENEE 244 or equivalent course on Digital Logic Design, ENEE 150 or CMSC 216 or equivalent course on programming (preferably C)

TEXTBOOK: There is no required textbook.

REFERENCES: TBD

INSTRUCTOR: Bogdan Kosanovic, Ph.D. EE
Manager/SMTS, Embedded Processing, Texas Instruments, Inc.
20450 Century Boulevard, Germantown, MD 20874
Phone: (301) 407-9579
E-mail: bogdank@ti.com

GRADING: HW (20%), LABS (40%), EXAM (40%)

DESCRIPTION: **PART I.** Introduction to Embedded Systems

- What are the embedded systems? (multi-disciplinary aspects, e.g. electrical engineering, mechanical engineering, bio-engineering, physics, robotics, environment, general system theory)
- typical HW components (clocks, timers, peripherals, transducers)
- SW tools (compilers, debuggers, version control, bug tracking)
- SW Engineering (requirements, design, coding standards, configuration management, static analysis, testing considerations, portability, re-use, real-time operating systems, object oriented abstractions, templating and generalizations, design/code reviews, optimizing code size, power consumption, and performance)
- embedded processors (general purpose, microcontrollers, DSPs, GPUs)
- embedded systems programming (interrupts, memory management, real-time programming, control vs. computation, code optimization, scheduling, state machines, event processing, generality vs. execution performance, precision and correctness vs. latency and throughput)
- interfacing with the real-world through signals (sound, image/video, sensors)

Labs: Basic microcontroller programming (e.g. GPIO, LED's/buttons, debugging/tracing, polling vs. interrupt handling)

PART II. Multithreaded Programming with RTOS

- Multitasking fundamentals
- Scheduling
- Inter-task communication
- Memory management

Labs: Endless (Idle) loops, many ways to toggle LED (bare bone, HW interrupt, SW interrupt, RTOS clock, tasks/semaphores, mailbox/queue), dynamic creation of tasks/semaphores.