**ENGN8537: Embedded Systems and Real Time Signal Processing**

**Offered In**: *2013 -- S2*

**Credit points**: *6*

**Academic Career:** *Masters*

**Course Subject:** *Engineering*

**Prerequisite**: *ENGN3213*

**Offered By:** *Research School of Engineering*

**Lectures**: *2 hours per week*

**Laboratories**: *3 hours per week, 8 weeks*

**Lecturer:** *Dr Brad Yu and Ben Nizette*

**Subject Leader:** *Dr Brad Yu*

**Contact:** Ben Nizette**; Email:** benjamin.nizette@anu.edu.au  **RSISE BLD115, Room-322.**

**Contact:** Brad Yu**;  Email:** brad.yu@anu.edu.au **RSISE BLD115, Room-353**

**Course Description:**

This course provides an introduction to the design and analysis of Embedded Systems.  Embedded Systems are at the heart of almost all modern technologies; Smart Phones to televisions, cars to intelligent light bulbs.  They are one of the most important disciplines in Electronic Engineering and require a unique set of technical and analysis skills to design effectively.  Students will learn how to critically analyse embedded design problems and present solutions that integrate effectively with the world in which they must operate.  Real-time concepts will be introduced with reference to FPGA and Microcontroller systems alongside high reliability methods.

**Learning Outcomes:**

* Analyse the strengths and limitations of Microcontrollers and FPGAs for Embedded Systems
* Construct real time constraints from problem specifications
* Effectively integrate sensor, actuator and interface devices with Microcontrollers and FPGAs
* Apply high reliability methods to safety-critical functions
* Optimise the interaction of Embedded hardware designs with their software

**Indicative Assessment:**

Labs (5%); Major Project (35%); Exam (60%)

**Recommended Reading:**

Burns and Wellings **Real-Time Systems and Programming Languages (Third Edition)**, Addison Wesley Longmain 2001

White, E. **Making Embedded Systems**, O'Reilly Media 2012

Noergaard, T **Embedded Systems Architecture,** Elsevier 2004

Readings specific to particular topics (such as a particular microcontroller, sensor etc) may be allocated during the course.

**Contents :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Week** | **Lectures** | **Ref** | **Labs** | **Mon** | **Wed** |
| **22/7** | 1 | Intro/ Microcontrollers and FPGAs | DSM Notes | None | BN |  |
| **29/7** | 2 | ES Architecture | NG1, NG3 | Intro to Altera and the DE2 | BN |  |
| **5/8** | 3 | Embedded Processors | NG 4 | Graphics and Hardware | BN |  |
| **12/8** | 4 | Embedded Processors 2 | NG 4 | Signals and Simulation | BN | Tute 1 |
| **19/8** | 5 | Real Time and Determinism | BW9, BW11 | Nios II | BN |  |
| **26/8** | 6 | RTDet 2, Embedded OS | NG9 | DSP on FPGA | BN |  |
| **2/9** | 7 | Embedded OS 2 | NG9 | Guided Project Session 1 | BN | Tute 2 |
|  |  | **Mid-Semester Break** |  |  |  |  |
| **23/9** | 8 | Sensors and Physical Interfacing | - | Guided Project Session 2 | BN | BN |
| **30/9** | 9 | Communications and Memories | NG5, 6, 7 | Guided Project Session 3 | - | BN |
| **7/10** | 10 | Physical Integration and Power | - | Guided Project Session 4 | - | BN |
| **14/10** | 11 | Reliability and Redundancy | BW2 | Final Project Demonstrations | BY |  |
| **21/10** | 12 | Effective ES | NG11, NG12 | Optional catch-up | BY | Tute 3 |
| **28/10** | 13 | Guest Lecture / Exam Prep |  |  | BN/BY |  |
|  |  | **Exam Period** |  |  |  |  |

NG := Noegaard, Embedded Systems Architecture

BW := Burns and Wellings, Real-Time Systems and Programming Languages

**Section Synopses:**

**Recap**: **Microcontrollers and FPGAs**

Review material from ENGN3213, particularly with a view to refreshing the student's memory regarding FPGA programming concepts and primitives, ready for the first set of labs.

**ES Architecture:**

Demonstrate that modern ES are excellent examples of a Systems approach to Engineering.  Show that they bring together many of the disciplines that the students may already be familiar with (e.g. Power, Digital Comms, Wireless Comms and one layer higher, Supply Chain, Sustainable Design etc).  Highlight the areas of typical ES that aren't covered elsewhere (e.g. sensors, Operating Systems etc) and motivate their teaching later in the course.

**Embedded Processors:**

Extend the student's knowledge of microprocessors and FPGAs to suitability for embedded applications.  Introduce the students to standard metrics for power consumption, processor performance etc.  Show the advantages of tightly integrated SoCs versus the PC paradigm of highly segmented and expandable processing.

**Real Time and Determinism:**

Define time as understood by computers.  Introduce time of delivery as a metric of correctness and describe, with examples, why this matters.  Particularly show situations where the time scales of even the fastest computers would violate RT constraints if those constraints weren't properly specified (i.e. why 'going faster' is no substitute for proper RT planning).

**Embedded Operating Systems:**

Define an operating system and motivate their role in Embedded Systems.  Give an overview of several commonly used alternatives and examine their performance, both in terms of throughput (the 'intuitive' definition of performance) and also RT performance, correctness, provability etc.  Briefly describe scheduling, locking, races and other areas where OSs can help or hinder RT performance.

**Sensors and Physical Interfacing:**

Reiterate the core fact that Embedded Systems, by their nature, have tight bindings with the environment in which they operate.  Discuss converter parameters and sensor and interface technologies.  Provide deeper insight in to a small set of sensing domains (e.g. temperature, position) in order for the students to understand the complexities of seemingly simple measurements.

**Communications and Memories:**

Describe the interfaces commonly used between embedded processors and their environment.  Talk about the strengths and weaknesses of familiar protocols (e.g. USB, Ethernet) and also low level communication channels such as SPI, I2C, PCI etc.  The discussion should be regarding not only their core functionality and use cases, but also their ease of integration with embedded processors and tasks.  This must cover both technical/electrical issues but also latency and RT performance.

All ES have memory for program and data storage and operation.  The different types of storage available should be discussed WRT their use cases, interfaces and, as always, latency and RT performance.

**Physical Integration and Power**:

All ES must exist within a larger framework.  Now that the pieces are in place, the integration of memories, communication channels, sensors, processors and software layers can be explored.

One often overlooked coupling between components is their power requirements and response.  Talk about decoupling, power rail sequencing and the software and hardware requirements around this.

**Reliability and Redundancy:**

Introduce fault mitigation methods including hardware and software redundancy, N-version design.  Discuss prominent embedded systems failures (e.g. Ariane 5) and reinforce the notion that every aspect of the system has to work in harmony with the intended operating environment.

**Effective ES:**

Follow the later chapters of Noergaard regarding specification, life cycles, debugging, testing and maintenance.