

Special Topics Assignment

*Instructor: Prabal Dutta**Due: Thu, Nov 23, 2015 – 1:30 PM EDT***Background**

Embedded systems cover a wide range of dynamic topics and being aware of the technologies, trends, and standards is necessary to ensure a successful project (and career). While this course exposes you to many of the fundamentals of embedded system design, including the ARM architecture, assembly/C and the ABI, software toolchains, interrupts, timers, memory, buses, serial communications, and data converters, there are many topics that we are not able to cover (or discuss in sufficient detail).

This assignment will give you an opportunity to learn about a topic in some depth using a quantitative methodology, work with other classmates to distill your findings, prepare a short talk, and share your learnings with the class. This process will give you a very real-world experience since almost every embedded systems design requires some research, some evaluation, and some compelling arguments for others that you are making a good choice in the midst of tradeoffs.

Logistics

The logistics for this assignment are as follow:

- **Team.** This assignment is to be carried out in teams of three (except for one team of two). To encourage broader interactions between members of the class, your team members *must not exclusively* be your lab partner or your project partner(s).
- **Content.** The content of the presentations must be crisp, clear, critical, comparative, and quantitative. The presentation should highlight tradeoffs where present (e.g. the best battery for a miniaturization application is Li+ because of its energy density, or ANT is a better choice for a battery-powered body sensors than BLE because of its lower energy/bit transmitted; i.e. your presentations must argue quantitatively *why* a claim is true). Every presentation must have at least *two* graphs, and ideally several, that present a set of figures of merit and a set of tradeoffs that you generate from the data you collect on your own. For example, for microcontrollers, you might choose to create a graph that shows sleep current (y-axis) vs memory size, e.g. RAM (x-axis). Pay careful attention to ensuring that the independent variable, if there is a dependency, is the x-axis, You *must* submit the data that you collect/generate as a spreadsheet and you *must* include citations to any sources that you used.
- **Length.** The presentation to the class should be *12* minutes long. The presentation will be followed by *5* minutes of Q&A. All team members must participate in the presentation and Q&A.
- **Template.** Use the PowerPoint templates we use for most lectures:
<http://web.eecs.umich.edu/~prabal/teaching/eecs373/slides/lec3.ppt>.
- **Slides.** Make your slides after you structure your thinking. Have something useful to say *before* you begin your slides. Check out Jean-Luc Doumont's suggestions on how to create great slides:
<http://pcs.ieee.org/podcast-creating-effective-presentation-slides/>.
- **Schedule.** Presentations will occur on the following days: 11/23, 11/30, 12/7 (if needed). The presentation schedule will be decided at random at the beginning of each day. If a team is randomly selected to present but a team member is not present, then the entire team's score will be reduced by 25%. Make sure you have your teammates' cell phone numbers so you can call them if they don't show up on time. If a student cannot attend class on a particular date, the instructor must be notified

in writing at least 24 hours prior to the start of that day's class. Plan accordingly if there is some variability in your commute.

- **Exams.** Material from the presentations may be included in the final exam.

Presentation Topics

The assignment of students to project topics will be made based roughly on expressed preference and coverage of topics. Those who express a preference and have priority will be awarded one of their top choices, and ideally their first or second choice. The assignments of students to topics is (or will be) available in the Google Spreadsheet that will be (was) shared with the class via email and/or Piazza. The instructor will make the final choice of topics to ensure a sufficiently diverse yet cohesive assortment.

Presentation Coverage

To provide a sense of the kind of things that should be covered in the presentations, read over the following topics and suggestions for things that would be appropriate in a presentation on that topic. Not all topics are covered here, and you certainly should not be constrained by the suggestions below. However, reading them should give you a better sense of what is expected.

- **Imager Sensors:** Imagers are now inexpensive and widely available. Analyze imagers from a variety of sources (Aptina, Centeye, Foveon, etc.) to understand their size, cost, power, pixel count and density, hardware interface, frame rate, and other properties. Capture this data in a spreadsheet. Read over datasheets to understand the key figures of merit. Present these findings in a quantitative framework. Identify useful relationships by plotting data from multiple devices (e.g. power draw vs pixel count, or frame rate vs output rate, or cost vs pixel count).
- **Energy Harvesting** Battery replacement has been identified by many as a problem inhibiting the growth of wireless embedded sensors. To address this problem, some have proposed using environmental energy harvesting to power devices. Enumerate the chips that have been recently released by companies like Linear Technology, Maxim Integrated, or Texas Instruments. Identify their tradeoffs. What kinds of energy sources do they work with? What is their minimum startup voltage? What is their quiescent current?
- **Project Soli:** Google's Project Soli is attempting to change the way we interact with small devices. This presentation will cover the motivations for Project Soli will cover how Soli works, what kinds of signal data Soli produces, and what sort of algorithms might be used to make sense of these signals to determine what gestures are being performed.
- **Capacitive Sensors:** Capacitive sensors are all around us: they are used for touch-controlled lighting, in some multi-touch interfaces, and in various other systems (like fingernail based touchpads). How do they work? What chips are available to integrate them into a system? How much power do these chips draw? What kinds of data do they produce? What is their spatial resolution? Explore these and other questions.
- **FRAM:** Ferro-electric Random Access Memory is relatively new memory technology available from Texas Instruments. Describe how FRAM works and what it's good for. How does it compare in memory density to other technologies like SRAM, EEPROM, and NAND and NOR flash in terms of power, cost, read times, write times, power, and energy per operation? What are some drawbacks and benefits of using FRAM in an application? Are there new applications enabled by FRAM?
- **Microcontrollers:** We studied the ARM Cortex-M3 in this class but there are many other micro-controllers available. Identify a handful of these based on different architectures (e.g. Atmel AVR, TI

MSP430, Microchip PIC18/PIC32, or the 8051). Identify key metrics of merit: cost, memory, peripherals, etc. Collect key performance data like $\mu\text{A}/\text{MHz}$. What kind of toolchain support is available? Why would a system designer pick one of these over the others? Argue for why certain MCUs are better options than other MCUs, and show how you'd convince your colleagues that your MCU choice is the best one.

- **Radios:** Wireless technologies have enabled mobile communications and computing. Looking ahead, low-power wireless is ushering in an age of connected devices. Research several wireless standards (e.g. ANT, Bluetooth, Bluetooth Low-Energy, ZigBee, 802.15.4, etc.) and describe what these various standards are, what modulation schemes they use, why there are so many different protocols, which applications they target, which vendors provide radios, and how a designer should go about choosing the right radio (or standard) for a particular application. Is it cost? Power draw? Standard vs proprietary? In stock at Digikey or Mouser? Other concerns?
- **Software Toolchains** We studied the GNU GCC toolchain for ARM processors. What other toolchains are available from third-parties? What are the tradeoffs among the various compiler options? What is the difference in cost and code generation efficiency? What are some of the reasons that one would choose a particular toolchain over a different toolchain?
- **Efficient Algorithms:** Many embedded systems employ signal processing and other algorithm for doing their work. Identify at least two of the most common algorithms (e.g. the FFT/DFT, ZIP, etc.), describe why the algorithm is useful, how it works, and ways in which it might be implemented. How should a system designer decide between rolling their own or using an off the shelf library? What are the tradeoffs? Cost? Code Size? Runtime? What's available as a library? Describe a library that you downloaded (e.g. FFT) and then tried to use. How did that go?
- **Motor Control:** Controlling motors from computers requires interface electronics and driver software. Research how various different kinds of actuators can be attached to embedded systems including servo motors, step motors, high-power DC motors, and perhaps even other actuators (e.g. pumps, solenoids, etc.). What are the digital signals that the processor must generate to drive these motors, like PWM. How are these signals generated? Are driver electronics needed, like power MOSFETs or an H-bridge chip, for certain motors but not others?
- **Gyroscopes:** Gyroscopes have not gotten as much attention as their accelerometer brethren, but they are quite useful for both tracking (e.g. people walking around) and control (e.g. helicopter or quadcopter stability). Research the range of embedded gyroscopes and identify how they trade off on a range of parameters, e.g. power, size vs power, cost vs power, integration (e.g. number of axes), etc. How are gyro data combined with accelerometer and magnetometer data for inertial navigation? What sort of processing algorithms do gyros use when used for doing inertial navigation?
- **Internet of Things:** The Internet of Things (IoT) marks the age of networked machines – when the number of embedded devices connecting to the Internet outnumbers the number of people connected to the Internet. It is now here, and we are connecting devices more quickly to the net than we are people. Learn about the trends underlying the IoT, identify the kinds of devices that are being connected (e.g. FitBit), and what are the methods that are being employed to connect them (e.g. wired, wireless)? What are some of the challenges that people face in connecting devices to the Internet? What are the protocols being employed (802.15.4, ZigBee, IPv6, CoAP, etc.)?

Checklist

Make sure that you:

- Remove any non-essential content from your slides. Keep it focused.

- Schedule at least one meeting with me to review your slides and content.
- Use the class template so all of the slides have the same look and feel.
- Submit both your PowerPoint slides and Excel spreadsheet (of the data underlying your graphs) via email to `prabal@eecs.umich.edu` with the subject line: [EECS 373] Special Topics Presentation by the due date/time.
- Attend class on the following days: 11/23, 11/30, 12/6 (if needed), ready to present from **your** laptop for your groups slides, and be sure to participate in the discussions of other group's presentations. The goal is to learn something new, share it with others, and have fun!