ECE155: Engineering Design with Embedded Systems Winter 2013

Lecture 1 — January 7, 2013

Patrick Lam *version 1*

This course is about three things: software design, project management, and embedded systems.

Introduction to Embedded Systems

An *embedded system* is any system hidden inside of a larger system to perform a task or set of tasks without the user's knowledge of the embedded system's existence.

Where can you find embedded systems?

Two types of embedded systems. The term "embedded system" covers many systems.

- Simple embedded systems might be constructed out of electronics without a processor or control software.
- Complex embedded systems incorporate one or more processors along with control software.

Another term is "embedded computer system", which overlaps with complex embedded systems. We'll call an *embedded computer system* a special-purpose computer system designed to perform a set of tasks without the user's knowledge of its existence.

Digression: Talking about the Course. Consult the course syllabus, available on LEARN, for more information about the course logistics. It contains information about the course objectives; grading scheme; labs; (unmarked) assignments; readings; and Policy 71.

Course staff have office hours. We encourage you to use the office hours to get answers to your questions. That's what office hours are for.

Back to embedded systems: Exhaust Gas Recirculator example. The widget that's circulating around the class is an *Exhaust Gas Recirculator*.

- *Problem:* car engines produce oxides of nitrogen (NOx) when they burn too hot.
- *Soluion:* The best way to do that is to recirculate already-burned exhaust gases, which don't burn again, thus lowering the temperature.
- *When?* It's difficult to figure this out. Hence, embedded systems.

Mechanically, the EGR contains a valve which lets exhaust gases back into the combustion chamber.

First Approach. When you step on the gas, this opens, and then closes, the EGR valve. However, you don't need, or want, EGR on a cold engine, because it lowers the engine's performance. So, GM put a fully-mechanical thermal switch in its cars. Unfortunately, this didn't work well, because mechanical components often don't work the way you want them to, as you'll find out.

More Mechanical Components. Car manufacturers added more components, like vacuum amplifiers, delay valves, and solenoids, to fix the issues: "spaghetti" tubes.

Embedded Systems in EGRs. Today's solution is to use a small embedded system to control the EGR valve.

What are the inputs and outputs of the EGR system?

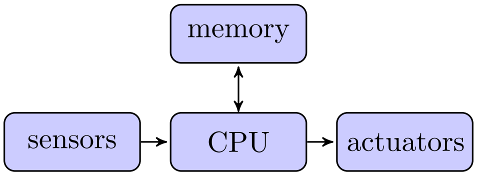
Design Constraints. There are two main constraints when choosing which processors (or electronics) to include in your embedded system.

- Processor power: The processor must be able to crunch enough bits. In particular, it needs to have good enough latency (response time) and bandwidth (processing power) to control the systems it's responsible for. Plus, you need to be able to model its performance characteristics.
- Environment: You have to be able to embed the processor. In particular, you have to respect size, power and connectivity constraints.

Challenges. Beyond meeting the design constraints, it's harder to write code for embedded systems than for PCs for a few additional reasons:

- Variability: Programming Windows systems is all the same, while programming cellphones is quite different from programming EGRs.
- No/bad UI: Can't necessarily put a print statement into a microwave oven's embedded system, nor can you put it in the state you'd want to.
- No API: An embedded system might not contain any operating system to speak of.
- Hard to get at: Have to load the software onto the system, which can be hard.

Block Diagram. Here are some parts of a typical embedded system.



Sensors provide input to the CPU, while actuators enable the CPU to affect the outside world.