Lecture 1 — Introduction

Jeff Zarnett
jzarnett@uwaterloo.ca

Department of Electrical and Computer Engineering
University of Waterloo

December 29, 2014

Course Syllabus

As our first order of business, we'll go over the course syllabus.

About Embedded Systems

A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. In many cases, their "embeddedness" may be such that their presence is far from obvious to the casual observer. Even the more technically skilled might need to examine the operation of a piece of equipment for some time before being able to conclude that an embedded control system was involved in its functioning.

(Institute of Electrical Engineers)

Where Can You Find Embedded Systems?

- Cellphones/other communications systems
- Microwaves/thermostats/other appliances
- Industrial automation
- Medical devices
- Transportation: aviation and automobiles

Two Types of Embedded Systems

1. Embedded Systems
2. Embedded Computer Systems

Exhaust Gas Recirculator

Many vehicles have a device in the engine called the Exhaust Gas Recirculator (EGR).

- *Problem:* car engines produce oxides of nitrogen (NOx) when they burn too hot.
- *Solution:* Recirculate already-burned exhaust gases
- *When?* It's difficult to figure this out. Hence, embedded systems.

Exhaust Gas Recirculator

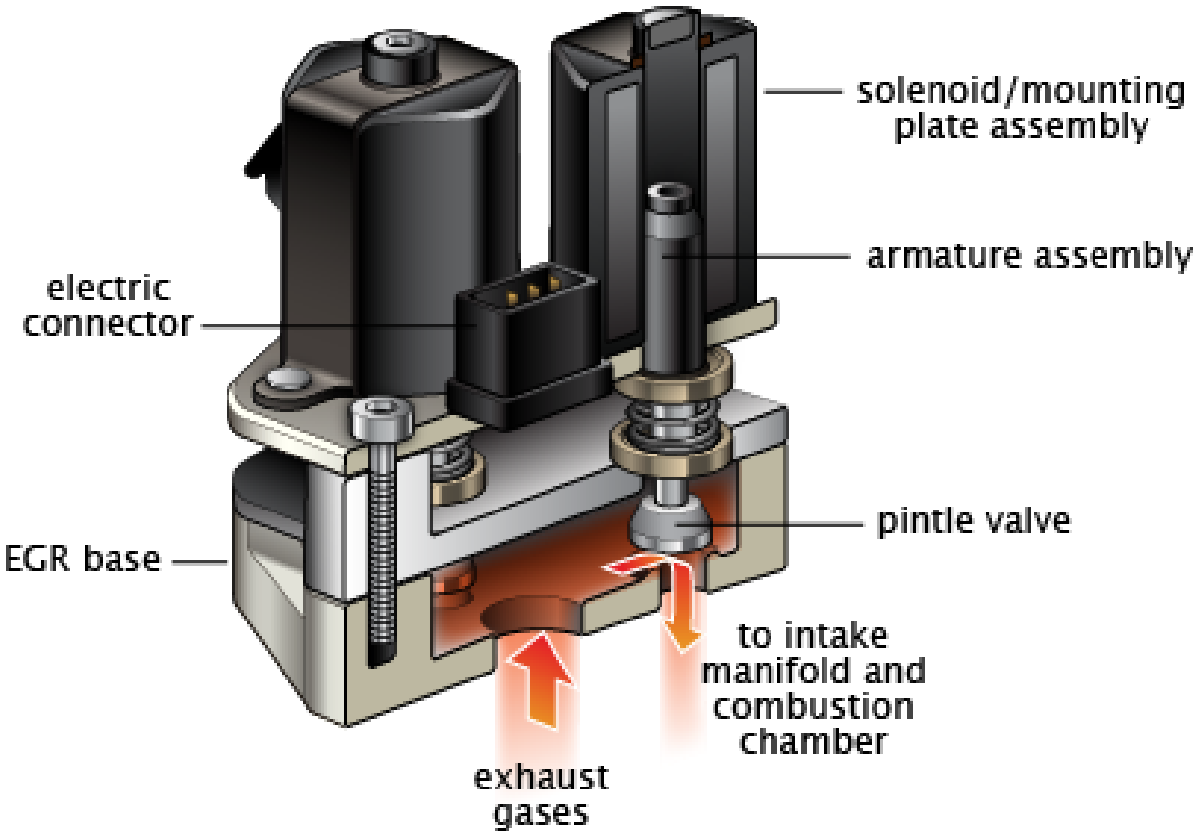


Image courtesy of ClearMechanic.com

http://repairpal.com/exhaust-gas-recirculation-system

Exhaust Gas Recirculator

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

Exhaust Gas Recirculator

First Approach: Fully mechanical system.

Problem: EGR is not needed in a cold engine.

Refinement: More complex mechanical system.

Problem: more complexity, more cost, more parts that can fail.

Now: Use embedded systems!

Exhaust Gas Recirculator: Inputs & Outputs

Inputs: RPM, throttle, temperature (sensors)

Outputs: signal to the valve; pulse-width modulation (actuators)

Exhaust Gas Recirculator: Design Constraints

1. Processor power
2. Environment

Embedded Systems Challenges

- Variability
- No/bad UI
- No API
- Hard to get at

Plan

Today we'll look inside a Samsung Galaxy S, with pictures from:

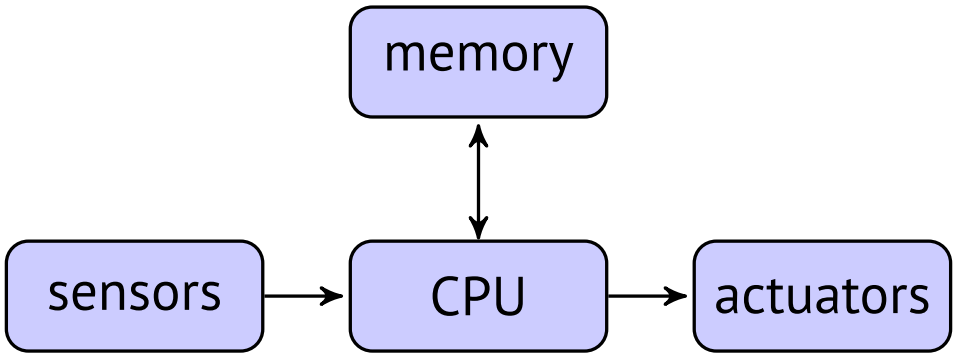
`http://www.ifixit.com/Teardown/Samsung+Galaxy+S+4G+Teardown/4977/1`

Thanks to `ifixit.com` for posting these pictures under the CC BY-NC-SA license.

Educational goal: be able to discuss sensors and actuators, plus embedded operating systems.

Sensors and Actuators

In particular, today we'll see how a typical phone integrates sensors and actuators, as typical for embedded systems.



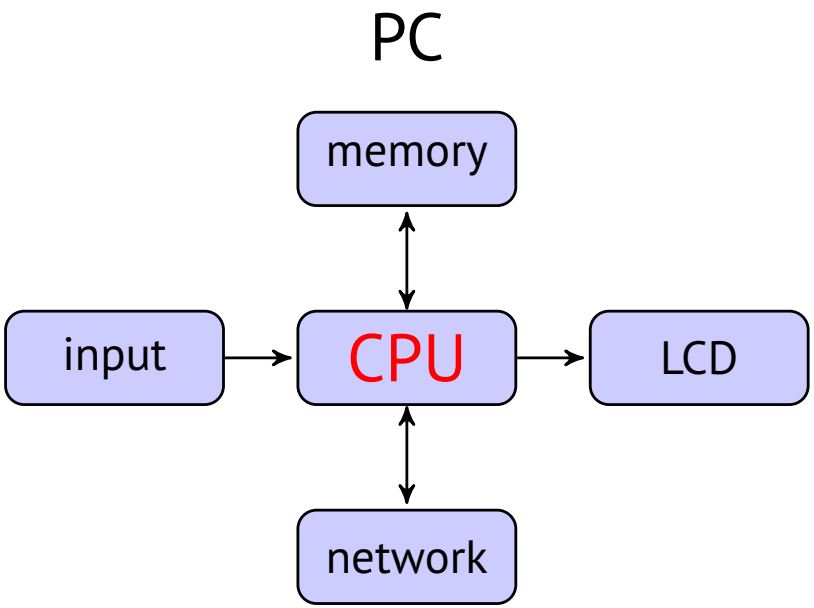
Specifications:

- 1GHz ARM "Hummingbird" processor
- 512MB RAM (plus 1GB storage & MicroSD slot)
- 480x800 Super AMOLED display
- 5.0MP and VGA cameras

What actuators can you see?

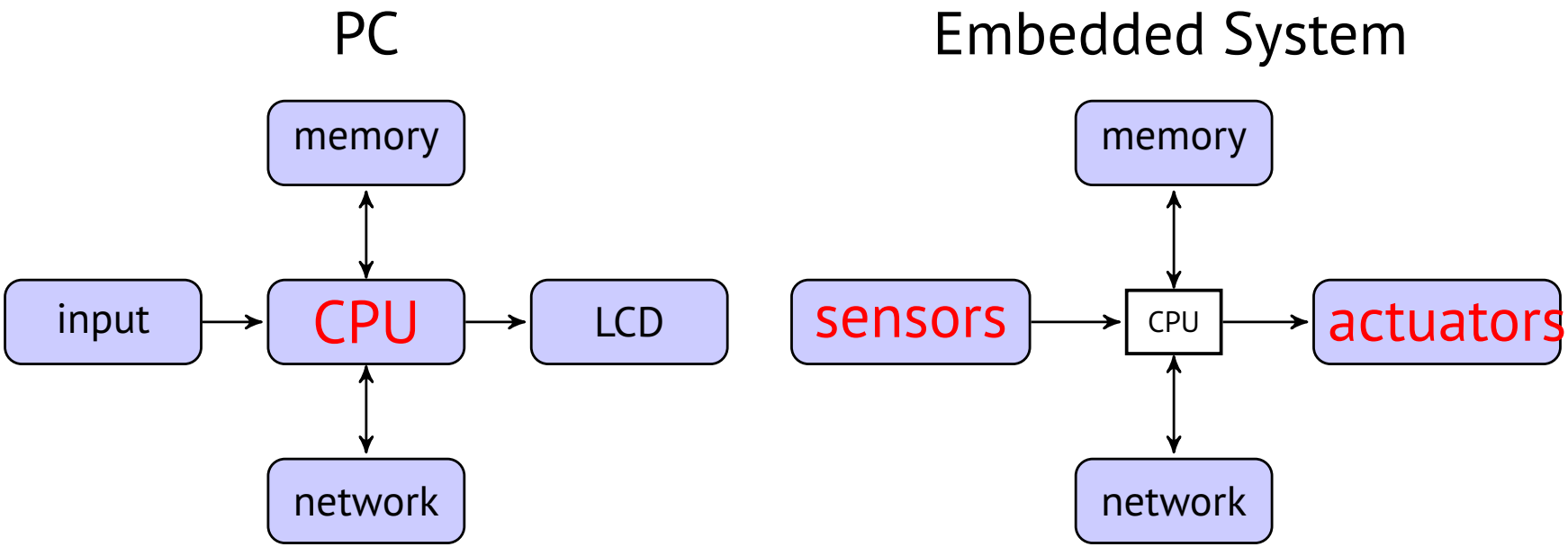
PCs vs Embedded Systems

Compare and contrast:



PCs vs Embedded Systems

Compare and contrast:



Opening the Phone

First Sensors and Actuators

Really Opening the Phone

This is as far as I'm willing to go with my phone. Thanks, Internet!

We're going to look at the components of the phone now.

Really Opening the Phone

This is as far as I'm willing to go with my phone. Thanks, Internet!

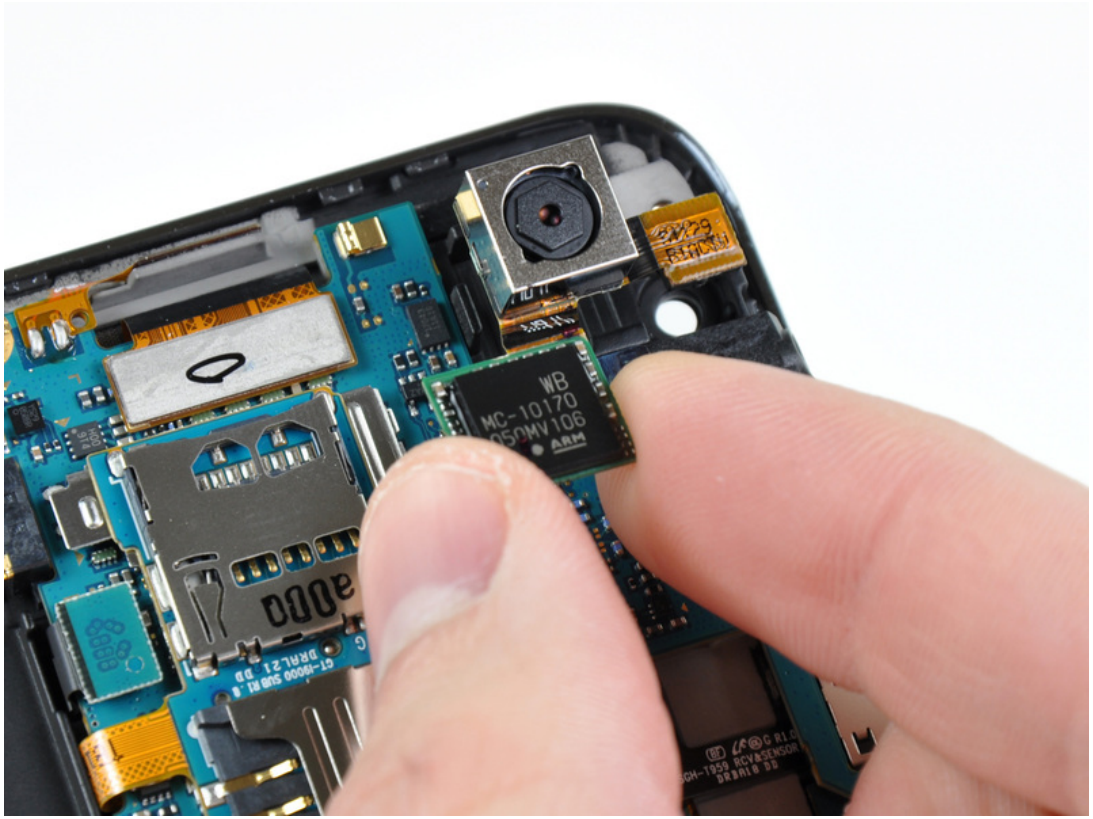
We're going to look at the components of the phone now.

Phone Speaker

Speakers are definitely actuators; they move the air.

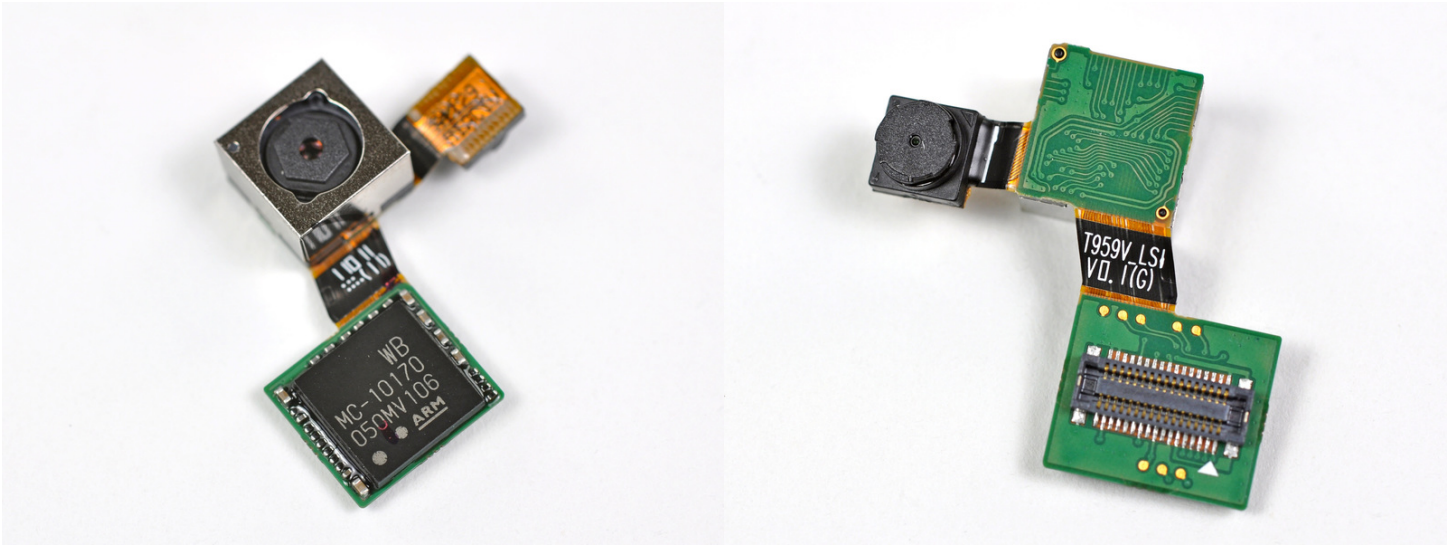
`http://www.howstuffworks.com/speaker.htm`

Next up, we have some sensors.

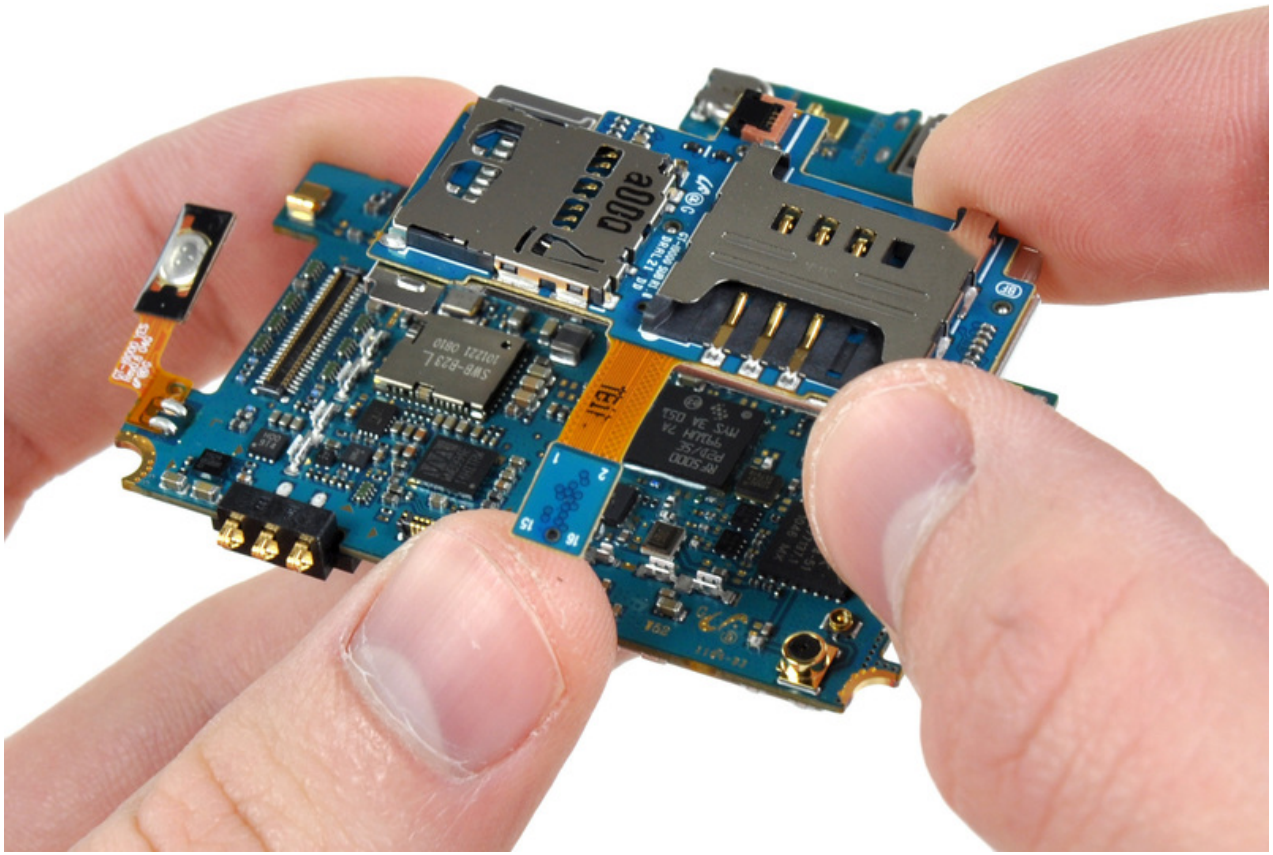


Cameras

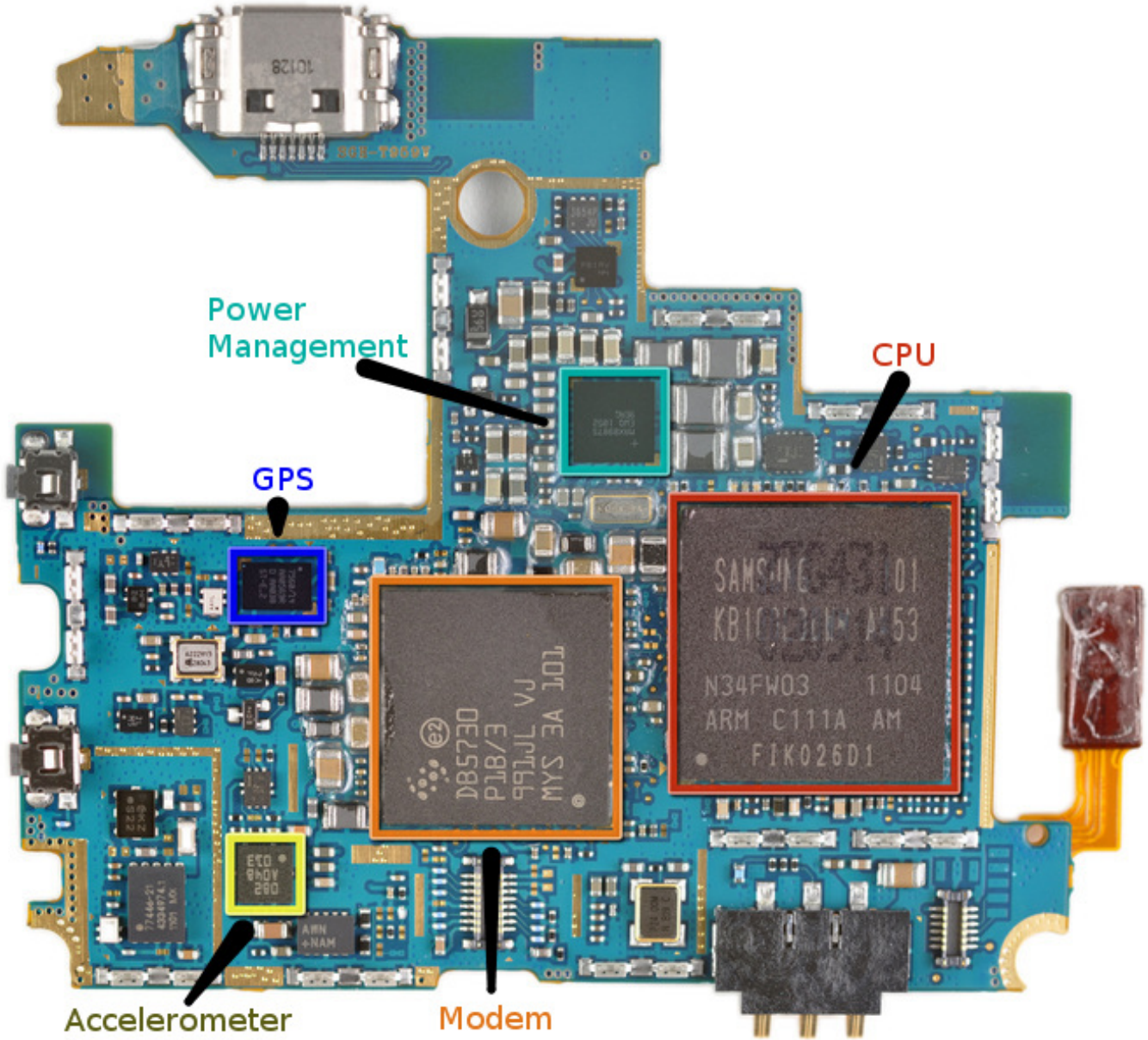
Next up, we have some sensors.



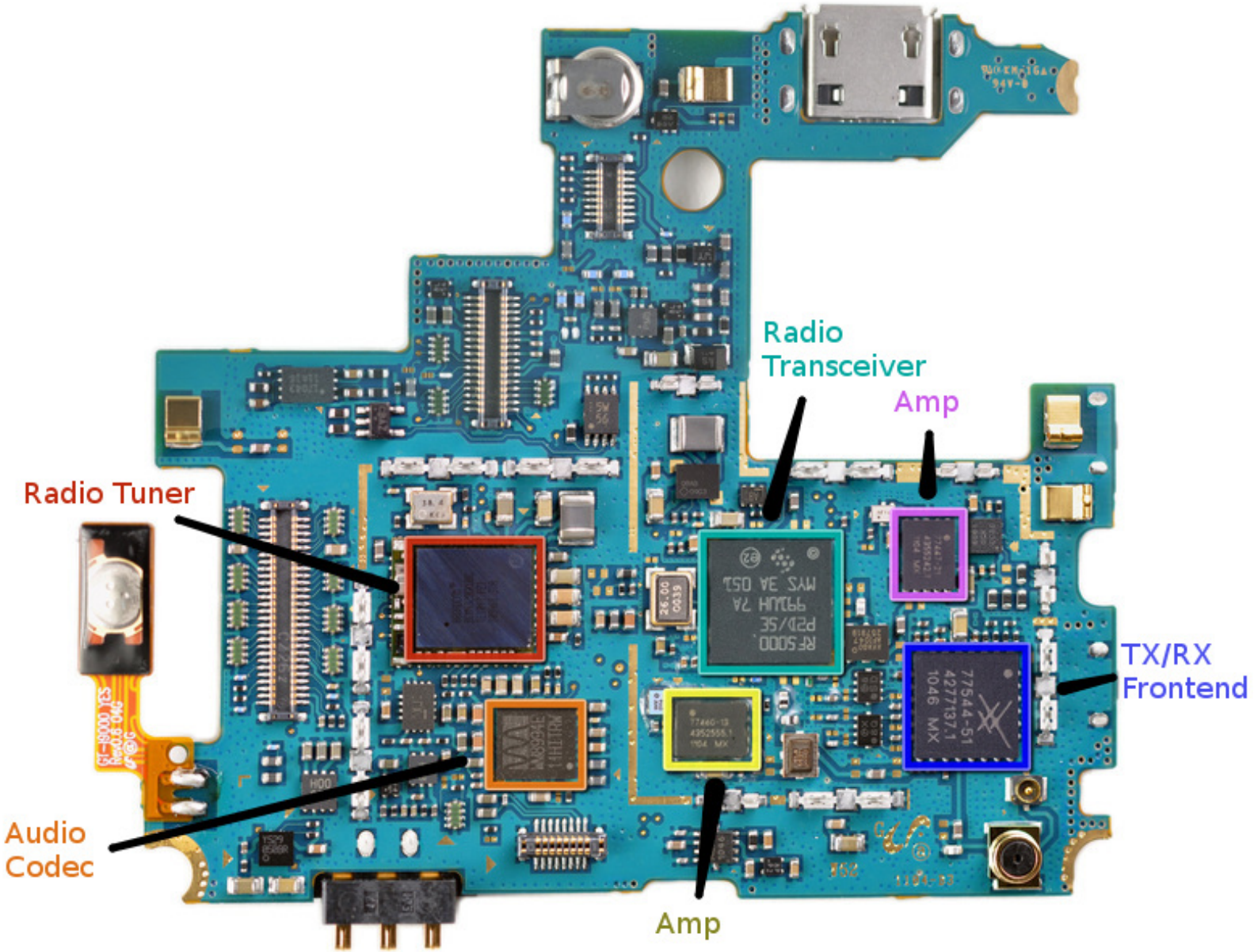
The Motherboard



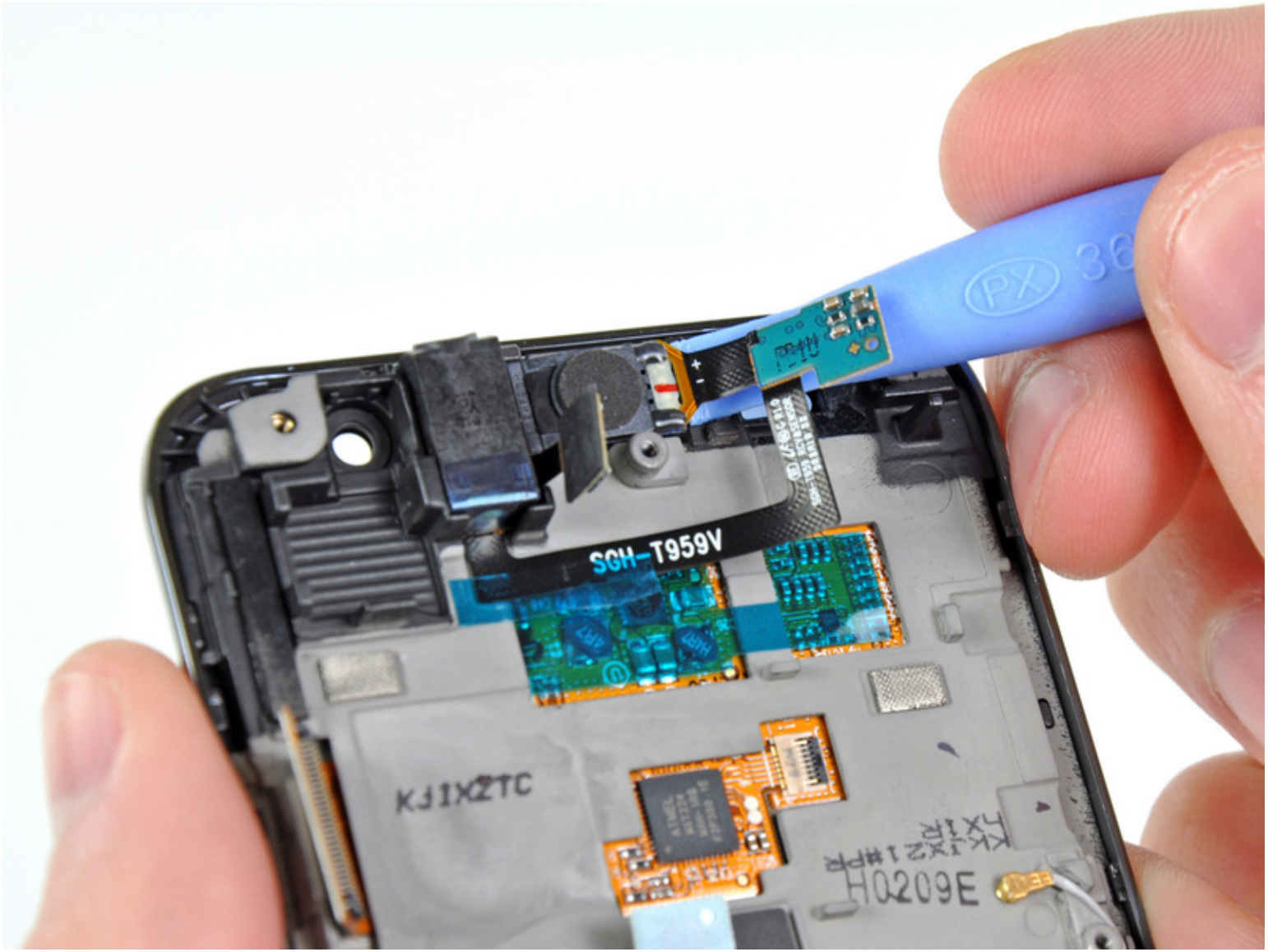
Touring the Motherboard Front



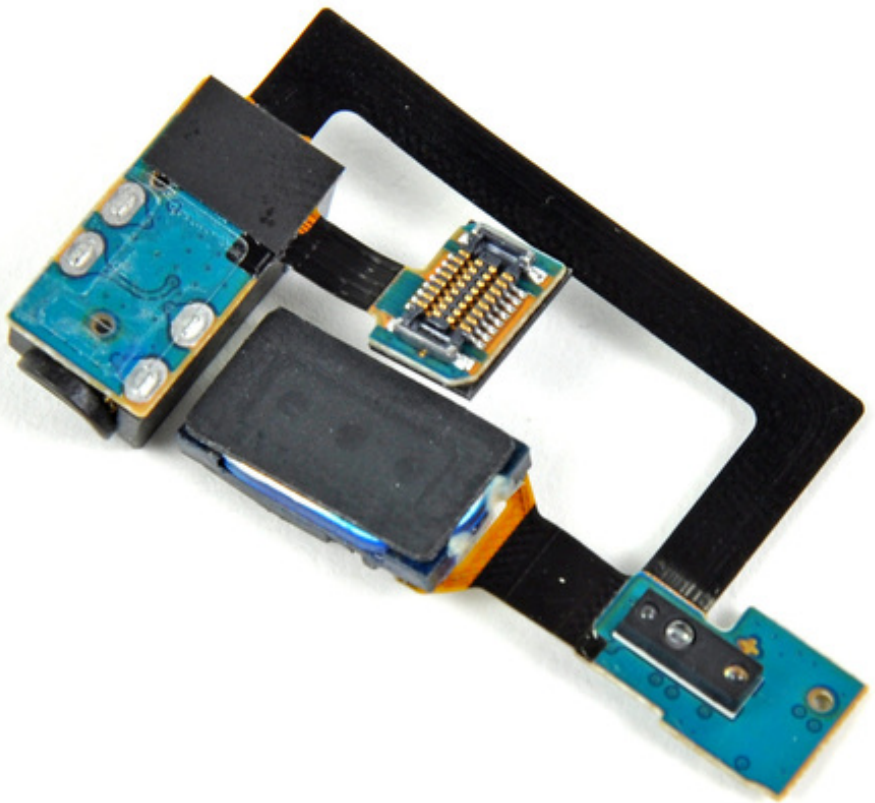
Motherboard Back: Mostly Radios



Headphone Jack, Earpiece Speaker, Proximity/Light Sensor

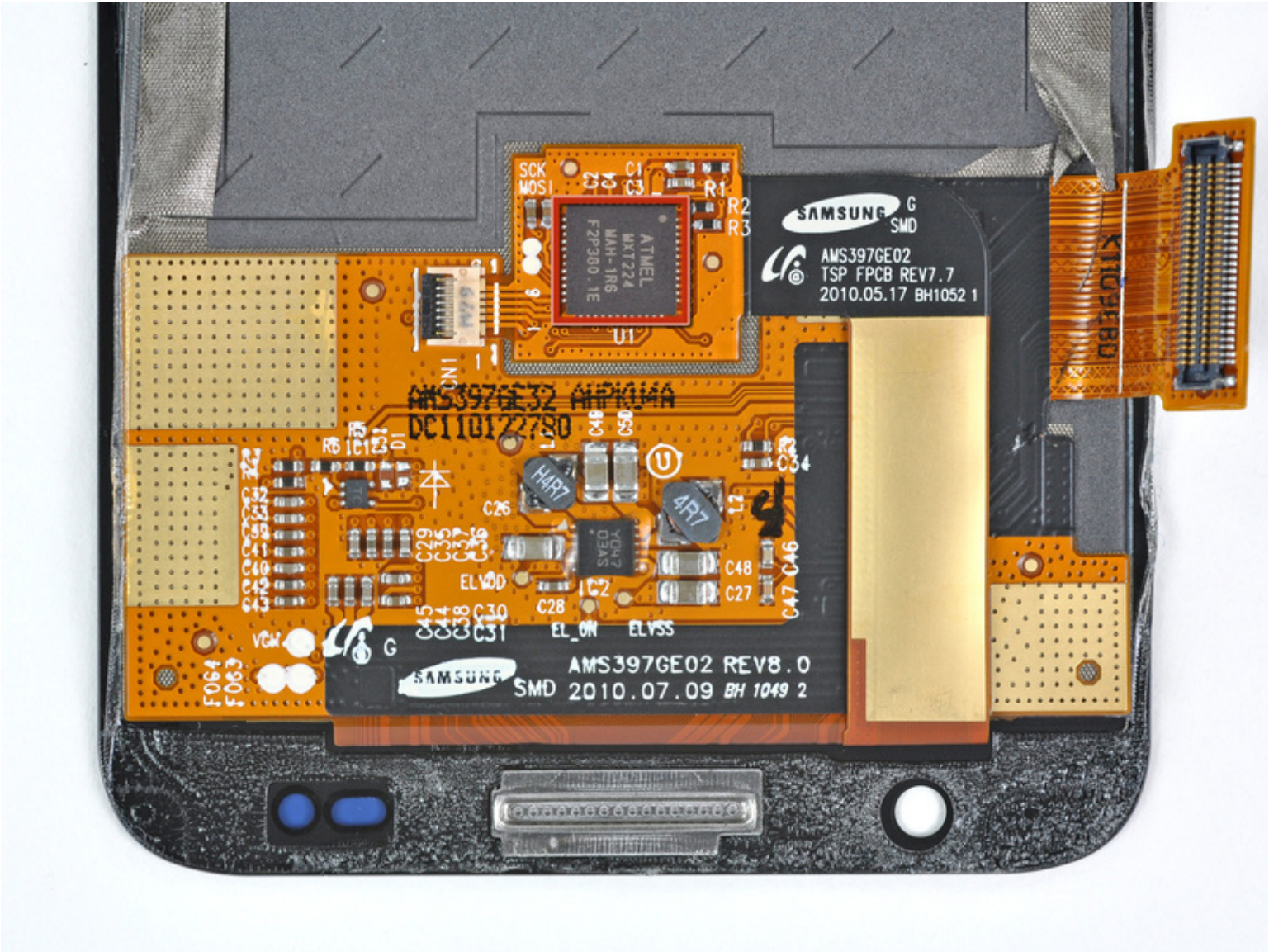


Headphone Jack, Earpiece Speaker, Proximity/Light Sensor

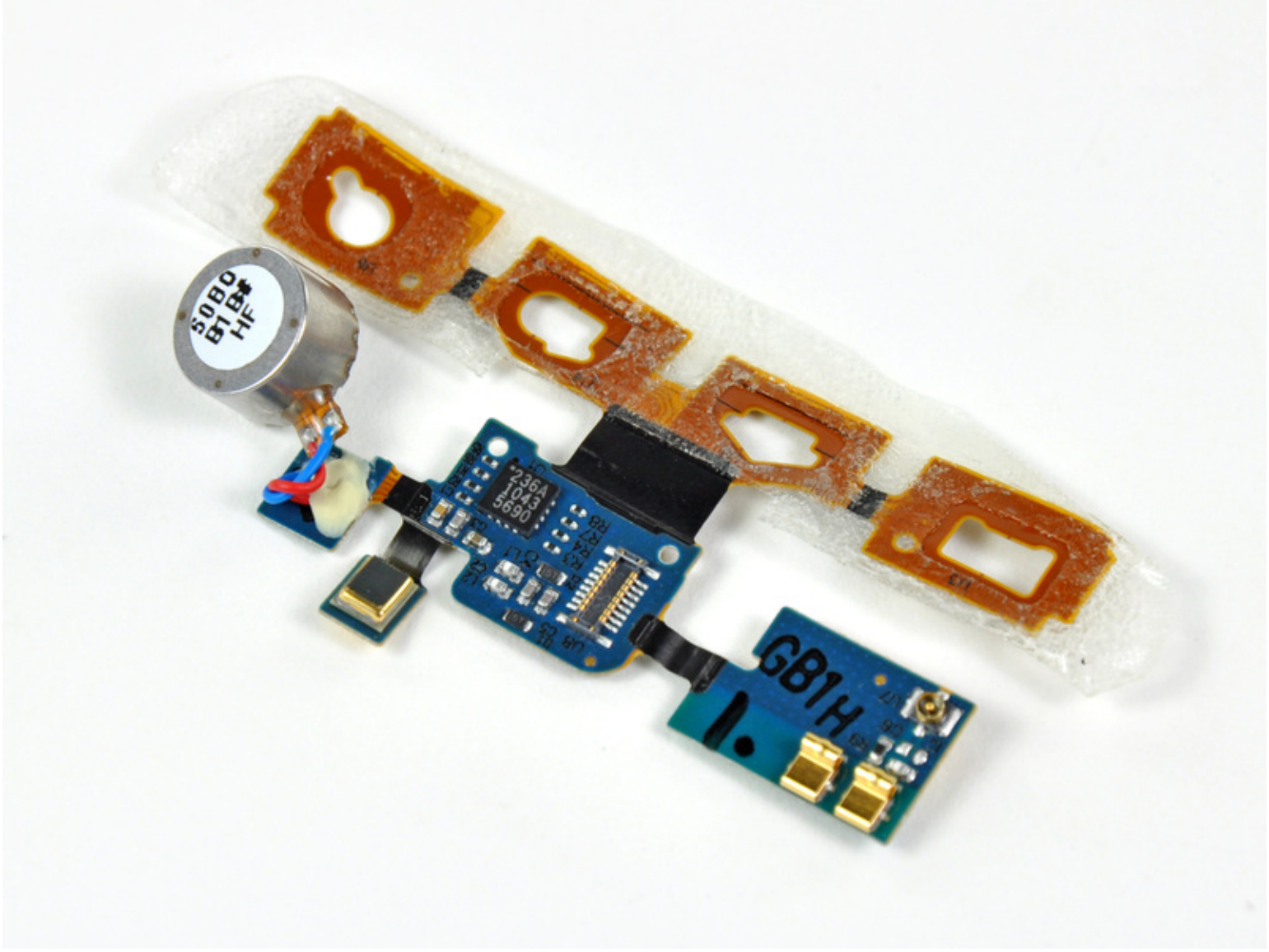


Heat Gun!

Touch Screen Controller



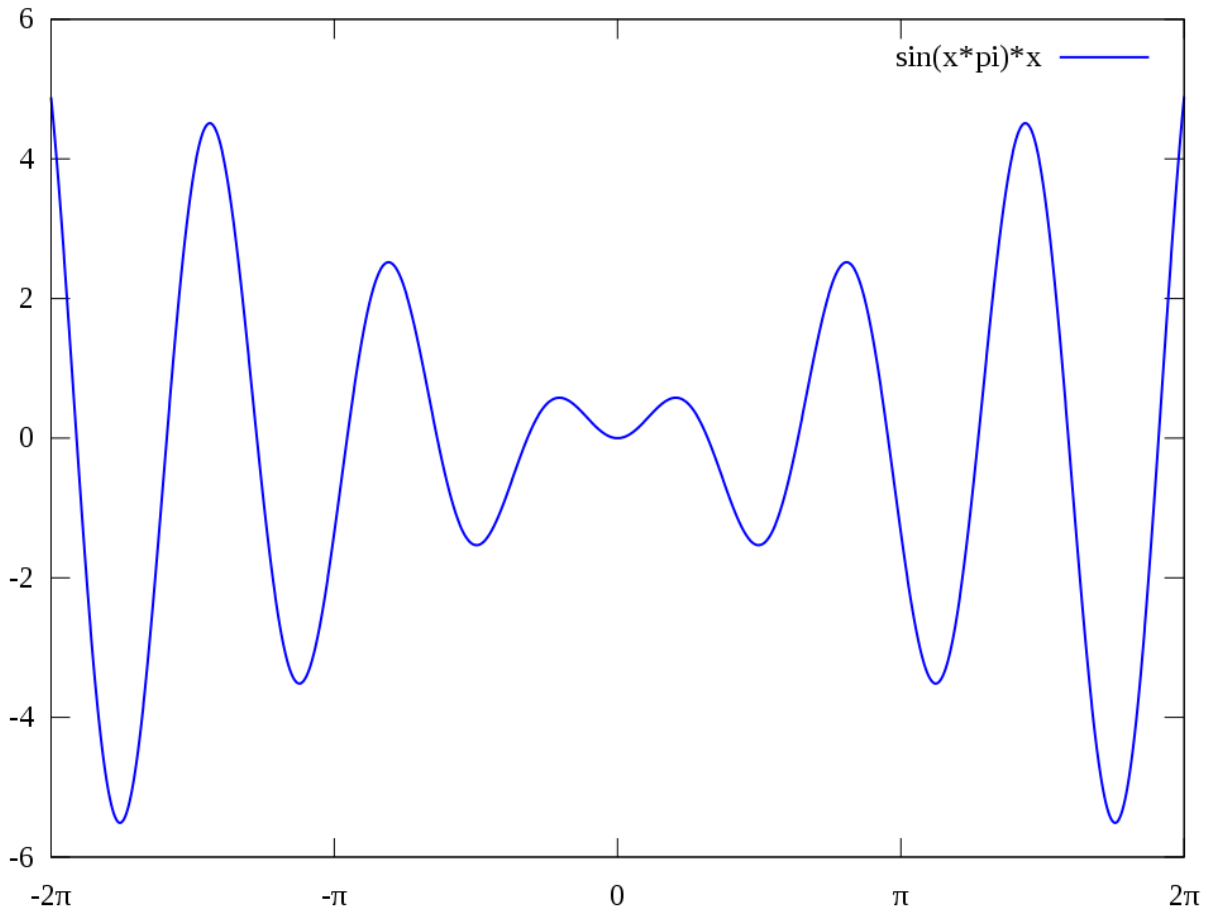
Touch Sensors, Vibrator, Microphone, Antenna Cable



Summing Up

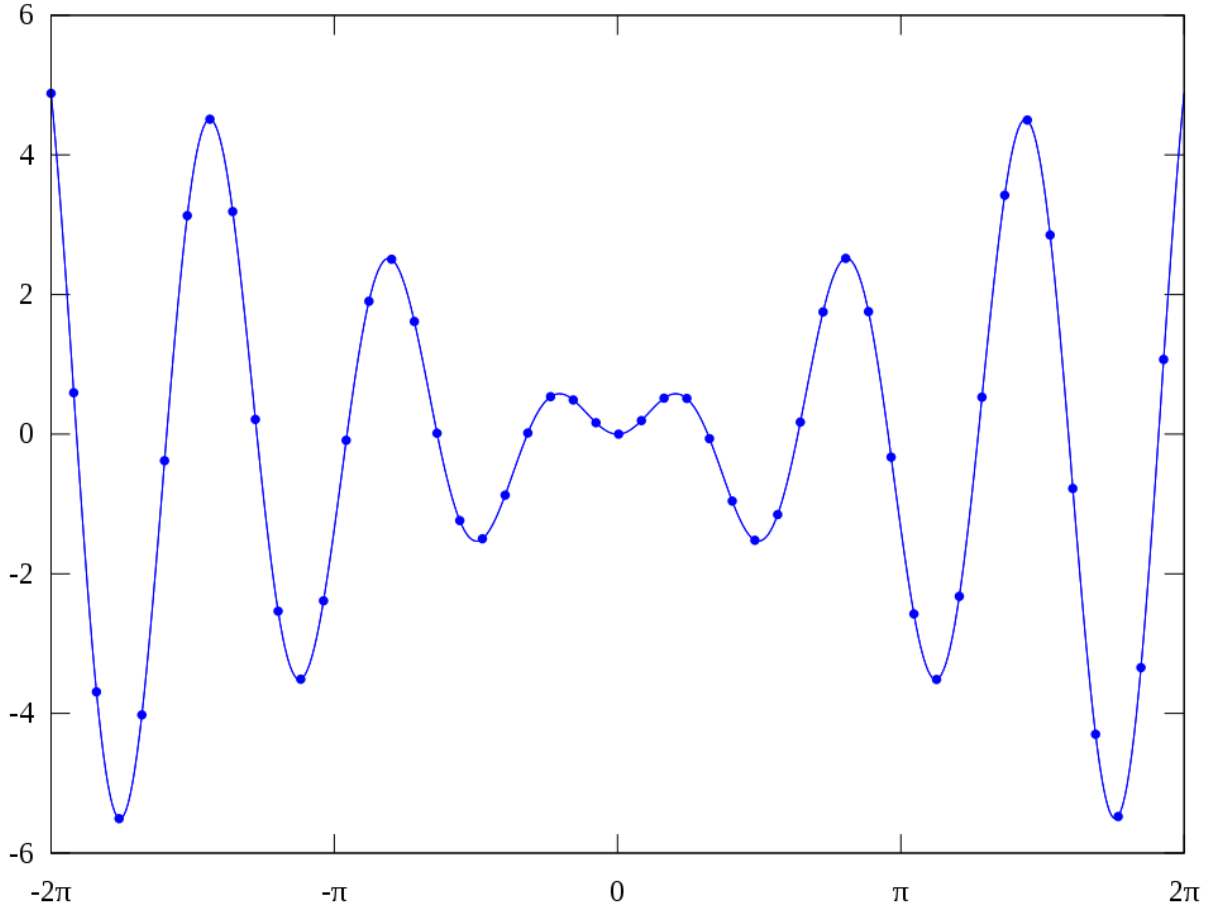
About Sensors

Consider a real-world phenomenon (e.g., light intensity):



Sensor results (in Volts, say) are continuous and analog.
Computers are discrete-time and digital. What to do?

Answer: Sampling



Report discrete values at given times, e.g.,

$$(0, 0), (0.2, 0.12), (0.4, 0.38), (0.6, 0.57), (0.8, 0.47).$$

Analog-to-Digital Converter

(Wolfson Microelectronics WM8994 Audio CODEC)

ADC part of the codec converts continuous-time analog signal to discrete-time digital signal.

ADC Output

We lose information if samples too infrequent[1], or if sample resolution too poor.

Time (ms)	Signal (V)
0.0	0.00
0.2	0.12
0.4	0.38
0.6	0.57
0.8	0.47

How would we increase sample frequency? Sample resolution?

[1] Nyquist sampling theorem.

About Actuators

Actuators convert output from the computer system into some effect on the environment.

What are some examples of actuators?

Some actuators require analog voltage signals;

- feed the discrete-time data to a digital-to-analog converter (**DAC**).

Combined Sensors and Actuators

(Wolfson Microelectronics WM8994 Audio CODEC)

This contains both an ADC and a DAC.

Also, e.g. piezoelectric sensors are both sensors and actuators.

– Android 2.3 embedded operating system.

About Android

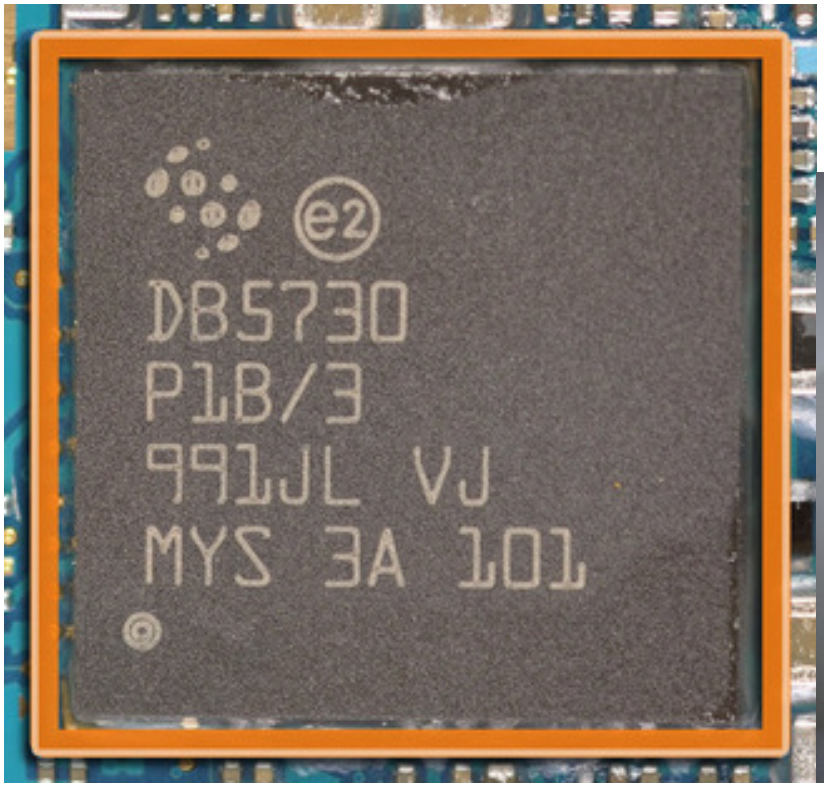
Example of an embedded operating system.

Target: smartphones and tablets.

- Runs Linux under the hood;
- Using Java, insulates applications from the hardware;
- Is compact and efficient;
- Cares about battery life;
- Favours portability;
- Available under an open-source license (Apache).

We'll learn Android programming in this course.

Consider the modem:



[2]

Converts encoded (radio) signals to and from bits.

[2] (credit Wilton Ramon de Carvalho Machado, http://en.wikipedia.org/wiki/File:Fax_modem_antigo.jpg)

Interacting with a Modem

```
Welcome to minicom 2.6.1

OPTIONS: I18n
Compiled on Feb 11 2012, 18:12:55.
Port /dev/ttyUSB0

Press CTRL-A Z for help on special keys

ATZ
OK
ATI
Manufacturer: huawei
Model: E1691
Revision: 11.126.15.03.562
IMEI: 355081032818180
+GCAP: +CGSM,+DS,+ES

OK
ATDT16172451343
NO CARRIER

 CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.6.1  | VT102 | Online 00:01
```

Running an Embedded Control Program

The modem is running an embedded control program.

- accepts high-level commands from the CPU;
- translates bits into analog signals; contains an ADC and a DAC.

More generally, the modem:

- boots automatically;
- never terminates;
- translates stream of sensor inputs and actuator outputs;
- cares about timing.

Therac-25 (1)

PATIENT NAME : TEST
TREATMENT MODE : FIX BEAM TYPE: X ENERGY (MeV): 25

	ACTUAL	PRESCRIBED	
UNIT RATE/MINUTE	0	200	
MONITOR UNITS	50 50	200	
TIME (MIN)	0.27	1.00	
GANTRY ROTATION (DEG)	0.0	0	VERIFIED
COLLIMATOR ROTATION (DEG)	359.2	359	VERIFIED
COLLIMATOR X (CM)	14.2	14.3	VERIFIED
COLLIMATOR Y (CM)	27.2	27.3	VERIFIED
WEDGE NUMBER	1	1	VERIFIED
ACCESSORY NUMBER	0	0	VERIFIED

DATE : 84-OCT-26 SYSTEM : BEAM READY OP. MODE : TREAT AUTO
TIME : 12:55: 8 TREAT : TREAT PAUSE X-RAY 173777
OPR ID : T25V02-R03 REASON : OPERATOR COMMAND:

Therac-25 (2)

- Radiation therapy machine
- Problem: many $\rightarrow$ race conditions, overflow, missing safety interlocks
- 3 patients died!
- Fix: software updates

Someone has to build these systems!

Your life will depend on them.

Someone has to build these systems!

Your life will depend on them.

Chances to learn about building safety-critical systems:

- ECE455 Embedded Software
- SE499: Independent project
- FYDP
- Undergraduate research assistant

Collaborative Course

The source material for the ECE 155 notes and slides is now open-sourced via Github.

If you find an error in the notes/slides, or have an improvement, go to `https://github.com/jzarnett/ece155` and open an issue.

If you know how to use `git` and LaTeX, then you can go to the URL and submit a pull request (changes) for me to look at and incorporate!