ECE/CS 5780-6780 : Embedded Systems Design
Lect. 01: Introduction to Embedded Systems

Pierre-Emmanuel Gaillardon
Department of Electrical and Computer Engineering – University of Utah

Spring 2017
Salt Lake City, UT, USA

A Few Words About Me

- ECE professor
- Second year at the U

- Undergraduate studies in Lyon, FR
- Graduate studies at CEA-LETI in Grenoble, FR
- Post-doc at EPFL in Lausanne, CH
- Visiting researcher at Stanford

- Research activities in “Nanosystems”
  (Digital VLSI design exploiting emerging technologies)

A Few Words About You

- Who is from EE? Who is from CS?
- Who registered at the 5000 level?
- Who registered at the 6000 level?

- What is your motivation for this class?
- Who loves hacking stuffs?

- Who has limited knowledge in C – HDL?
- Who has limited EE knowledge

A Few Words About the Class

- A pretty complicated class : ECE/CS 5780/6780
- 2nd Edition of the Course: Should have less kinks but still…
- Mixed Goals:
  - You have yours - I have mine
  - We'll make it work!!
    - → *Input/feedback from you will help*
- I want this class to be as interactive as possible

Open-Mindedness

- Some topics may seem familiar
  - Be open to seeing them in a new light
- It is embedded systems: **NOT** CS or ECE
- It is **NOT** only a class-based course
  - There is a STRONG lab component
  - Labs to help with applying concepts
  - Mini-project to stimulate interest and creativity
- It is **NOT** about learning C
  - Though this will be our tool of choice

Prerequisites

- Official prerequisites
  - ECE/CS 5780
    - ECE/CS 3810 (Computer Organization)
    - CS 2000 (Introduction to Program Design in C), OR CS 4400 (Computer Systems), OR CS 1410-30 (Introduction to Object-Oriented Programming)
  - ECE/CS 6780
    - Must be enrolled as a CS or ECE graduate student.
- You are expected to have knowledge of the following subjects:
  - Programming and data structures in C (queues, stacks, linked lists...)
  - Microcomputer programming (i.e., assembly language programming)
  - Digital logic (multiplexers, tristate logic, finite state machines, etc.)
  - Hardware description languages (VHDL, Verilog)
  - Test equipment (multimeters, oscilloscopes, etc.)
  - Discrete analog electrical circuits (resistors, capacitors, inductors, transistors, opamps, etc.)

Contact Information

- Webpage: access using Canvas
- Instructor: Prof. Pierre-Emmanuel Gaillardon

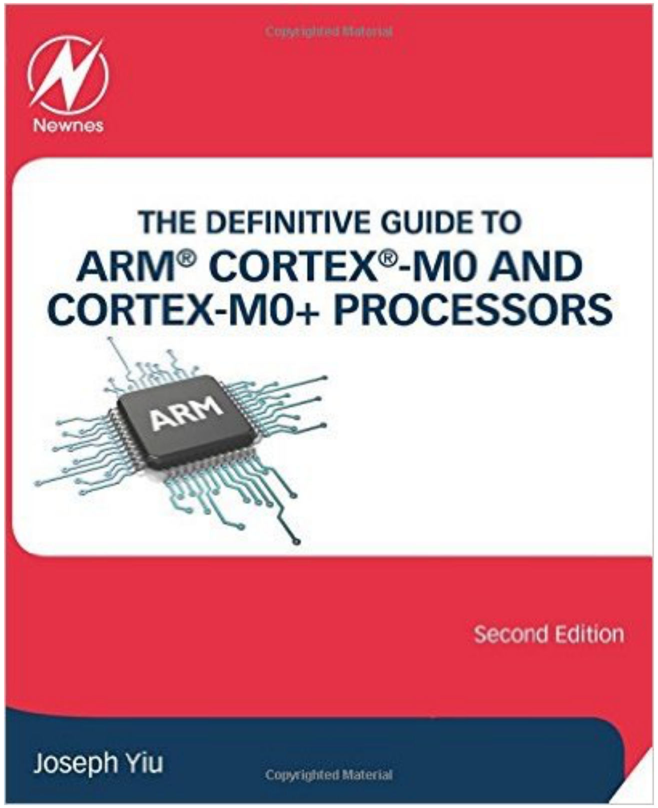
  pierre-emmanuel.gaillardon@utah.edu
- Office: MEB 2126
- Office hours: T 2:15-3:30pm / Th 11:15-12:15am
- Teaching Assistants:
  – Brent Mellon (Labs): brent.r.mellor@utah.edu
  – Thomas Becnel (Labs): thomas.becnel@utah.edu
  – Mehedy Hasan (Lead Grader): mehedy.hasan@utah.edu
  – Tengda Shi (Grader): tengda.shi@utah.edu
- - Send questions to:

  teach-ece5780@list.eng.utah.edu

Textbook (Optional)

The Definitive Guide to ARM® Cortex®-M0 and Cortex®-M0+ Processors, 3rd Edition

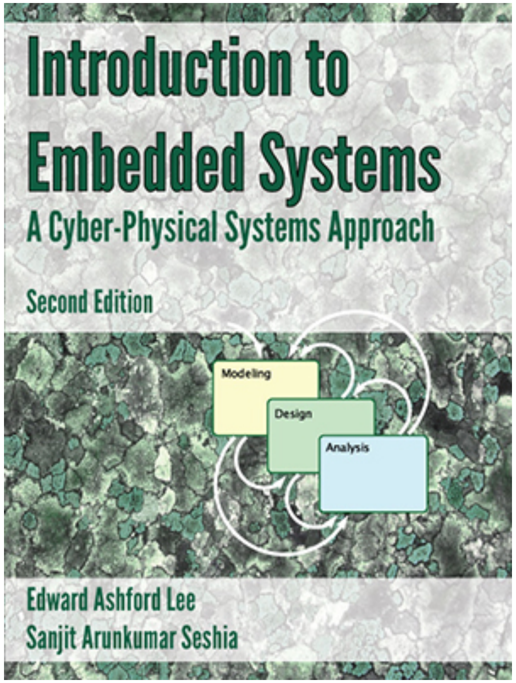
Joseph Yiu – Publisher: Newnes



Textbook (Optional)

Introduction to Embedded Systems: A Cyber-Physical Systems Approach

Edward Lee / Sanjit Seshia



Available for free download or purchase from:
http://leeseshia.org

Course Description (Traditional)

- Introduction to issues in embedded system design using microcontrollers.
- Topics include:
  - Microcontroller architecture,
  - Embedded software design,
  - Interrupt synchronization,
  - Timing generation and measurement,
  - Serial and parallel I/O interfacing, and
  - Analog interfacing.

Course Description (CPS-Focused)

- Introduction to issues in *Cyber-Physical Design* (CPS)
- Topics include:
  - Embedded software design
  - Interrupts and hardware interfacing
  - Model-based design
  - Representations for continuous and discrete dynamics
  - Reasoning about concurrency and timing
  - System analysis and verification

What We Will Do

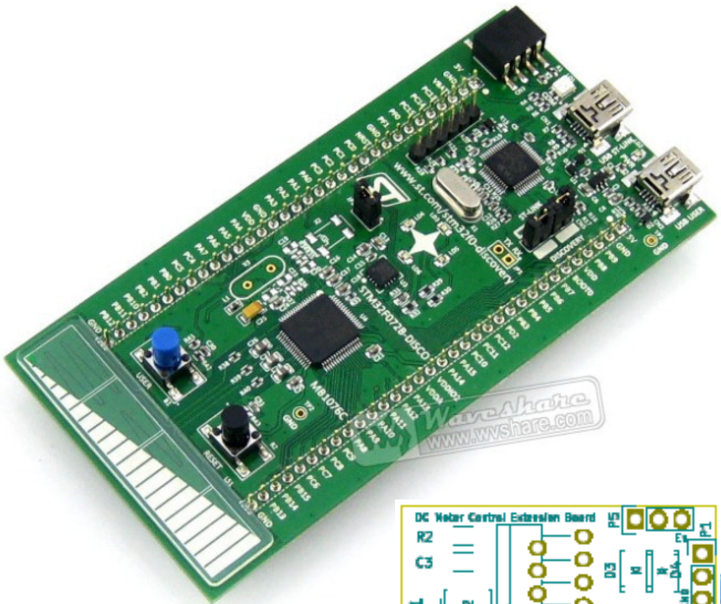
- Learn embedded systems:
  - Through a hands-on approach
  - Quite a bunch of hacking (even though people do not believe in that)
  - Augmented with design methodologies

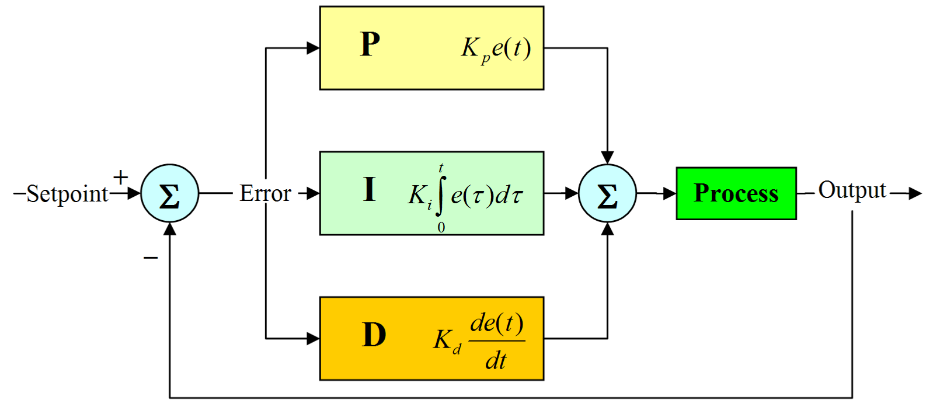
- 2 parts in the class:
  - Lectures, Lab sessions and HW
  - Graduate presentations and mini-project

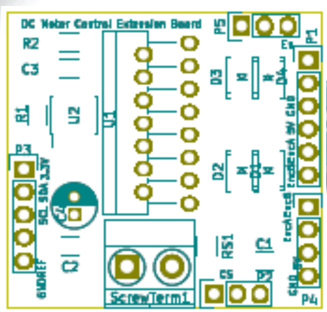
Course Objectives

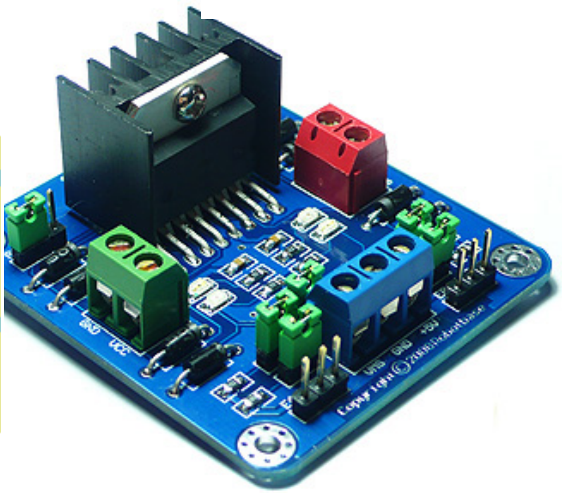
- Get familiar with the ARM M processor architecture
- Understand the architecture of a microcontroller
- Acquire the fundamentals of the hardware/software interface
- Acquire the fundamentals for sensing and controlling the physical world
- Learn modeling techniques for embedded system design
- Understand the usage of several peripherals through labs
- Design a complete embedded system – from specs to realization
  - Realization of a PCB
  - Behavioral modeling of the system
  - Complete SW realization

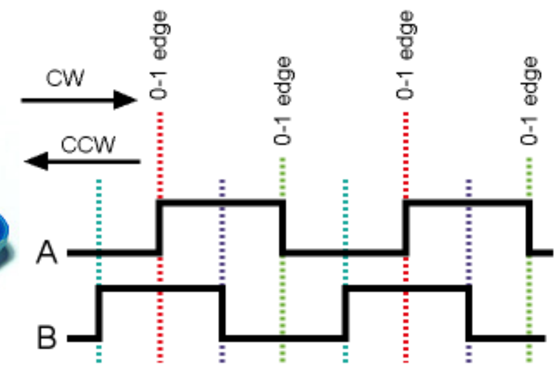
Directed Labs / Homework Mini-Project

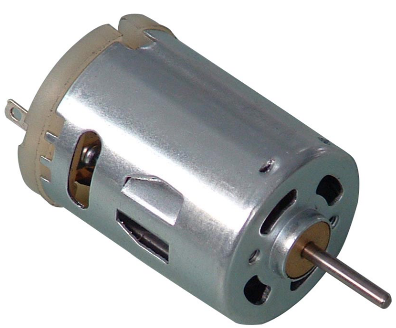












Build an autonomous embedded system:
- with an ARM M0 processor
- with a DC motor H-bridge
- with a quad. encoder interface

What We Will Do: Labs

- Work in pairs
  - You **must** partner with someone in your lab section
  - Try and form cross discipline pairs
  - Must have a CADE account.
- TA will be present in labs
- Labs Topics:
  - System modeling
  - ARM Toolchain and Hello world
  - Interrupts / Timers / PWM
  - DAC / ADC
  - Communications: UART, I^2C
  - DC motor control on MCU
  - DC motor control on MCU+ RTOS
  - Mini-project

What We Will Do: Homework

- Design and build a PCB
  - This board will be used during the lab sessions
  - Contains analog interface for the DC motor
  - Contains I^2C temperature sensor
  - It will part of a real embedded system you'll work with in the mini-project
  - You have to pay for the board fabrication and components
- Answer questions on graduate presentations
- Most other activities will be folded into the labs

What We Will Do: Mini-projects

- *Goal*: learn how to build embedded systems
  - By **expanding** our motor control system
  - Work in teams
  - Pick a problem of your own interest
  - Meet with instructors to discuss other ideas
- Should be related to the class and emphasized topics
- Grad. Students: Can relate to your research
- Scope of project must grow with size of team

What We Will Do: Presentations

- For Graduate students only

- Prepare a 15-20 minute in-class presentation
- Should be aimed at “teaching” something

- Prepare a question related to the material (with solution)
- They will serve to prepare a graded homework

What We Will Do: Exams

- A Unique Midterm
  - Covering the most important notions of embedded systems and software design
  - Emphasis on problem solving fundamentals
  - Planned on Feb. 21

Grading Policy

Item	Weight 5780	Weight 6780	Description
Labs (including mini project)	50	45	8 labs (+ mini project)
Exam	30	30	
Homeworks	20	15	Weighted equally
Presentation		10	15-20 min. related topic

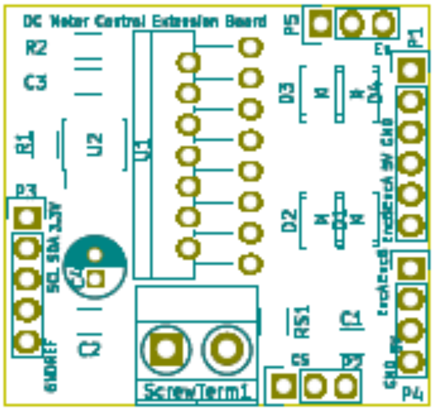
Lab Assignments

Sect. 2	07:30am-10:30am	W	Becnel	MEB 2265
Sect. 3	03:05pm-06:05pm	W	Mellor	MEB 2265
Sect. 4	07:30am-10:30am	F	Mellor	MEB 2265
Sect. 5	03:05pm-06:05pm	M	Mellor	MEB 2265
Sect. 6	07:30am-10:30am	T	Becnel	MEB 2265

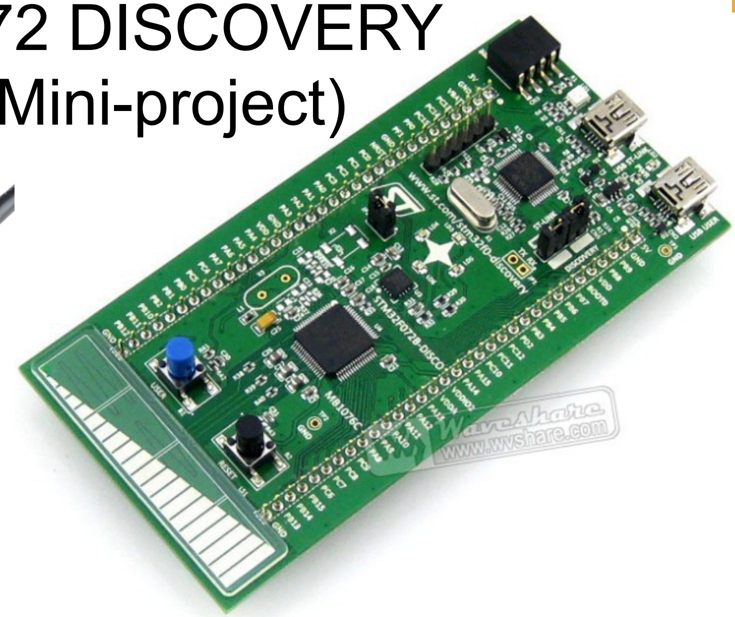
- Groups will be made on canvas for you to form you pair
- If you do not find a partner before the first lab, you will form your group at the first lab
- **Labs begin the week of January 17th**
- Lab schedule has been adapted for Monday group to accommodate public holidays

Hardware (to buy)

Custom PCBs
(HW, Labs,
Mini-project)



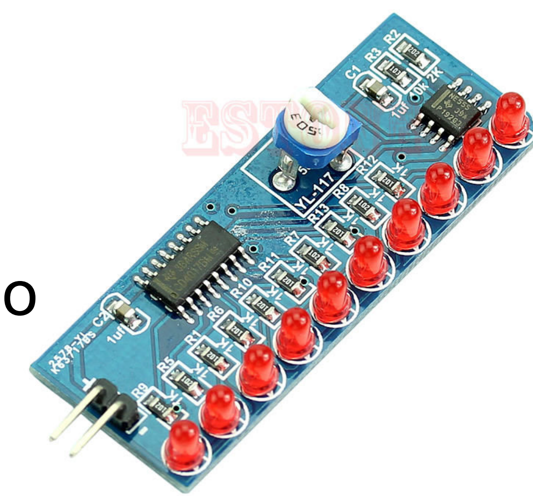
STM32F072 DISCOVERY
(Labs, Mini-project)



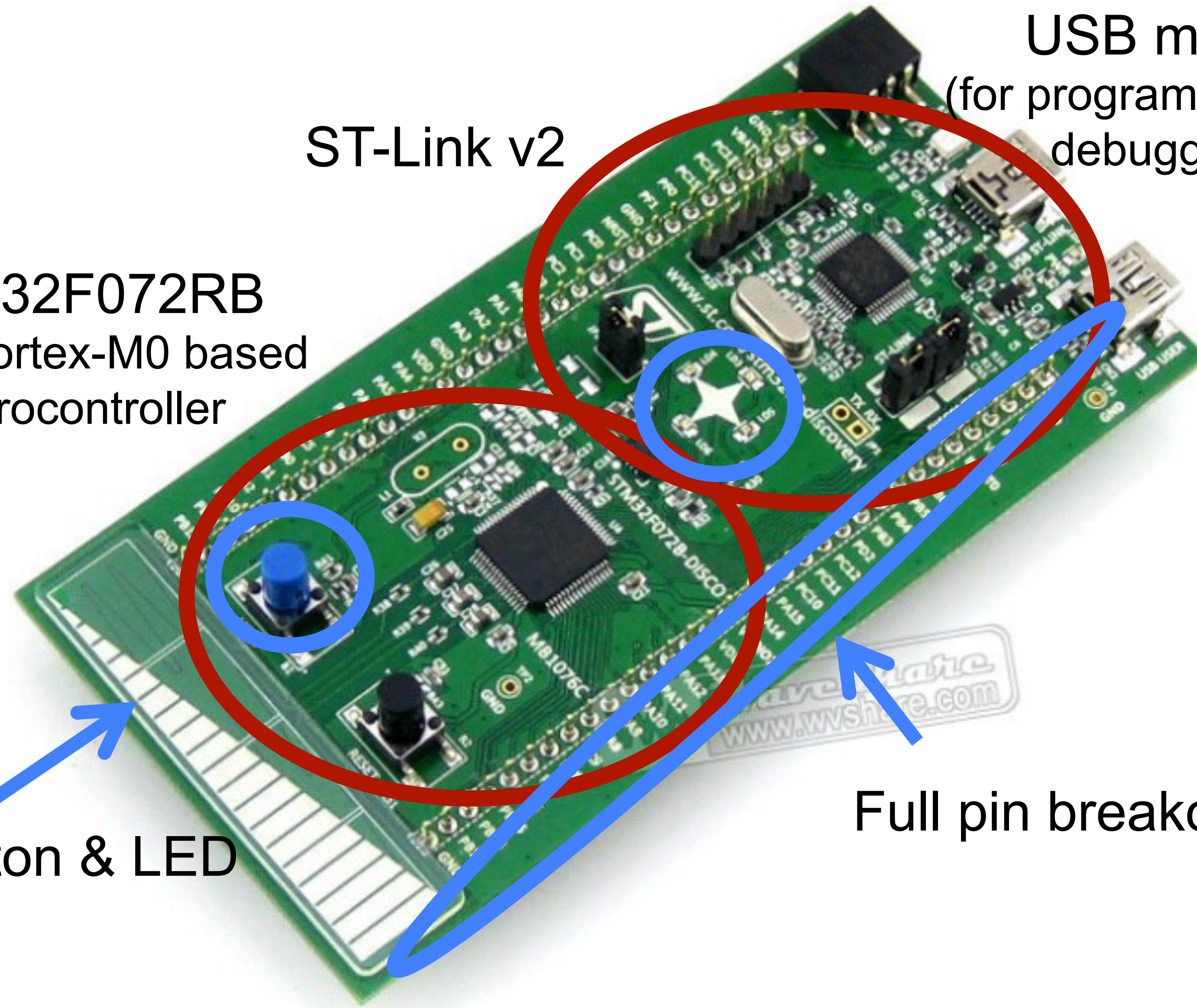
Adafruit
USB-UART Cable
(Labs, Mini-project)

Saleae Logic 4
analyzer
(Labs, Mini-project)

Soldering demo
board



STM32F072 Discovery Board



STMF0 Discovery Kit

- STM32F072RBT6 microcontroller
- On-board ST-LINK/V2 with selection mode switch to use the kit as a standalone ST-LINK
- Board power supply: through USB bus or from an external 5 V supply voltage
- External application power supply: 3 V and 5 V
- Four user LEDs:
  - LD3 (orange)
  - LD4 (green)
  - LD5 (red)
  - LD6 (blue)
- One user button
- Extension header for all LQFP64 I/Os for quick connection to prototyping board and easy probing

STM32F072RB Microcontroller

- Core: ARM®32-bit Cortex®-M0 CPU, frequency up to 48 MHz
- Memory: 128 KB Flash, 168 KB RAM
- Peripherals:
  - Up to 68 I/Os with 5 V tolerant capability – All mappable on external interrupt vectors
  - One 12-bit, 1.0 µs ADC (up to 16 channels) - Conversion range: 0 to 3.6 V
  - One 12-bit DAC channel (with 2 channels)
  - Up to 11 timers
  - One 16-bit advanced-control timer for 6 channels PWM output
- Independent and system watchdog timers
- SysTick timer: 24-bit downcounter
- Communication interfaces
- Two I^{2}C interfaces, Four USARTs, Two SPIs, CAN, USB

Kits Availability

- **Every student has to buy an STMF0 Discovery kit and a USB-UART cable.**
- We also recommend to buy a soldering demo kit to practice before you solder you own board
- **The kits are available for purchase from the ECE stockroom**

  ($15.25 MCU board - $12 USB-UART cable - $1.35 Soldering demo)

- **We strongly recommend to buy a Saleae Logic4 logic analyzer** ($54 with EDU discount from the manufacturer)

  Please get in touch with Brent to get the discount code!

- A few older Saelae logic analyzer can be checkout during lab sessions only

STMF0 kits: Things to Remember

- **Be careful not to short the board**
- Use antistatic mats and connectors in lab when handling the boards

  (ask TA if you don't understand this)
- Ground yourself before touching the board
- Handle boards by their edges
- Avoid tasks that produce static before handling the board

Lab Reports – Homework Policy

- Working lab should be checked off during your lab section
- **Lab reports due on Canvas by the time of your next lab section**
- Put your lab section number on all lab reports
- One late check off – One late lab report
- Further reports will lose 20% per week for 2 weeks; then a 0%
- Lab reports will be returned via Canvas by your next lab section
- All labs reports and exams are assumed to be correctly graded one week after they are returned
- After the one week has lapsed, no changes will be considered

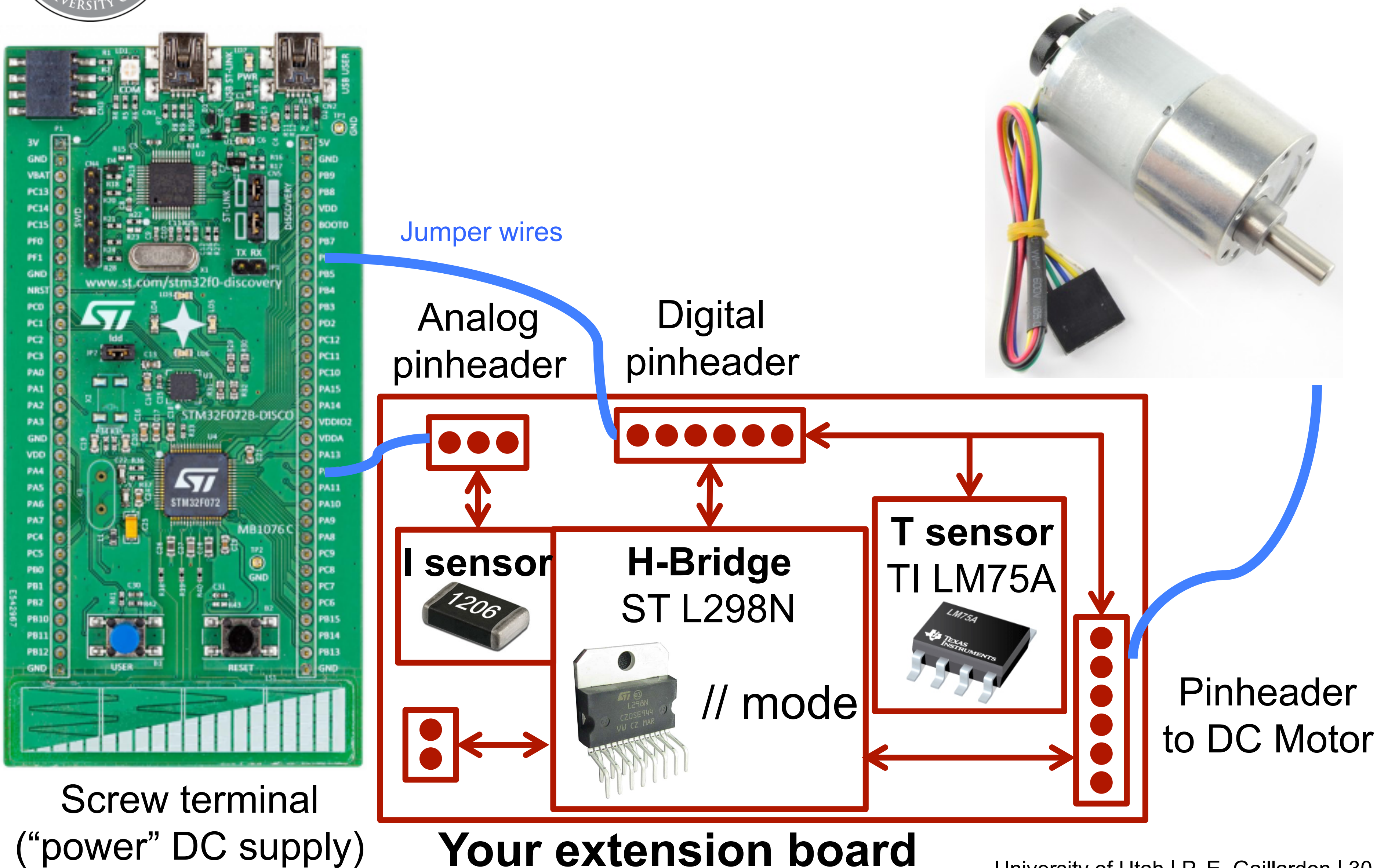
- **No homework will be accepted late**

- If you have a question regarding your grade, please contact the grader for that assignment to address the issue

Cheating Policy

- **Cheating will not be tolerated in this course**
- First offense results in a zero on the lab/exam
- Second offense results in a failure of the course
- Cheating includes but is not limited to:
  - Examining or using solutions for labs from previous years or other students in the class
  - Copying a solution from another student during an exam
  - Making ones solution to a lab or exam question available to another student
  - etc.
- Use your common sense. Anything that gives you an unfair advantage over other students is likely considered cheating

HW1: Extension board PCB design



Homework 1

- Introduction to PCB Design
- Check-out canvas for details and submission
- Four-part homework
  1- Choose a MCU and make a schematic using Eagle
  2- Layout PCB and verify it
  3- Get your board manufactured by Oshpark
  4- Assemble your board
- *First part due*: Jan. 26th at 11:59pm
- You have to **pay** for manufacturing & parts
- Eagle is available as a **freeware** or with **EDU license**

What are Embedded Systems?

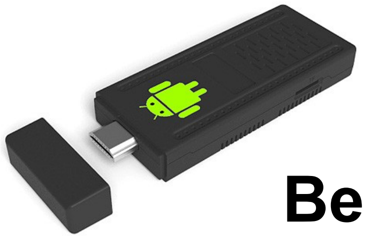
What are Embedded Systems?

Computers whose job is not primarily information processing, but rather is interacting with physical processes

- A few attributes:
  - Application-specific
  - Resource-constrained
  - Real-time operations
  - Software runs “forever”
  - More and more: Networked, Integrated with physical processes

A broader view is that of *Cyber-Physical Systems (CPS)*

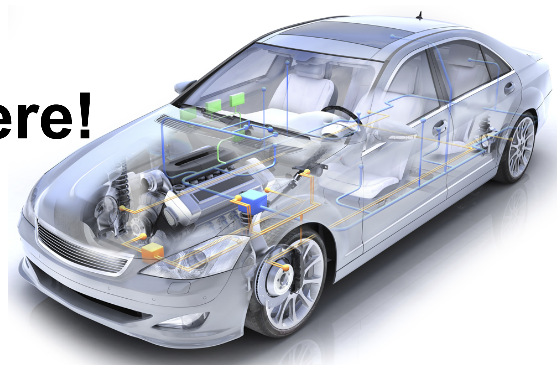
Why should we care?

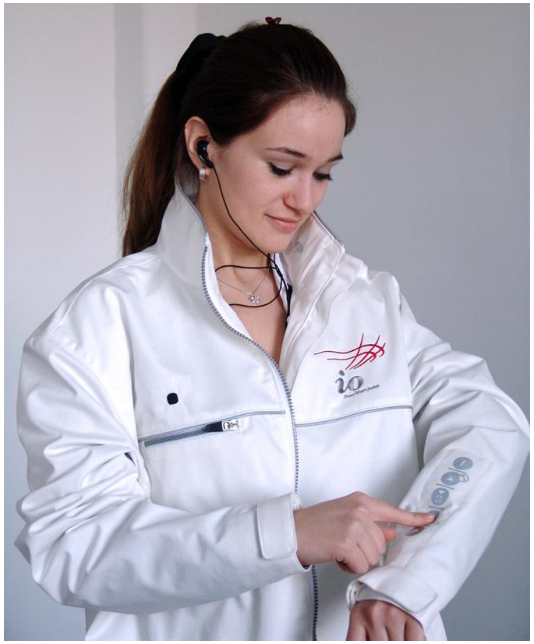


To pass the course?

No!

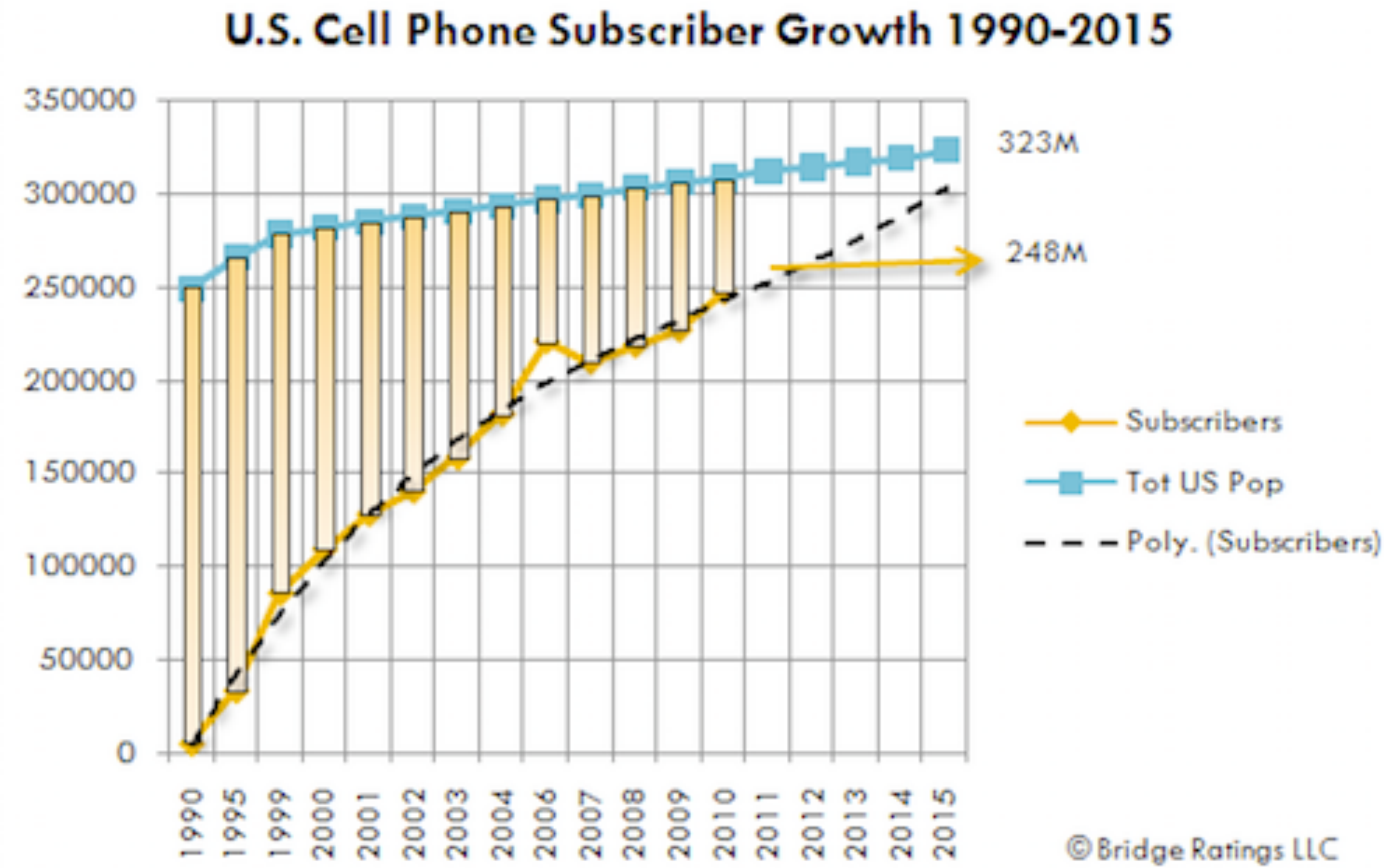
Because they are everywhere!



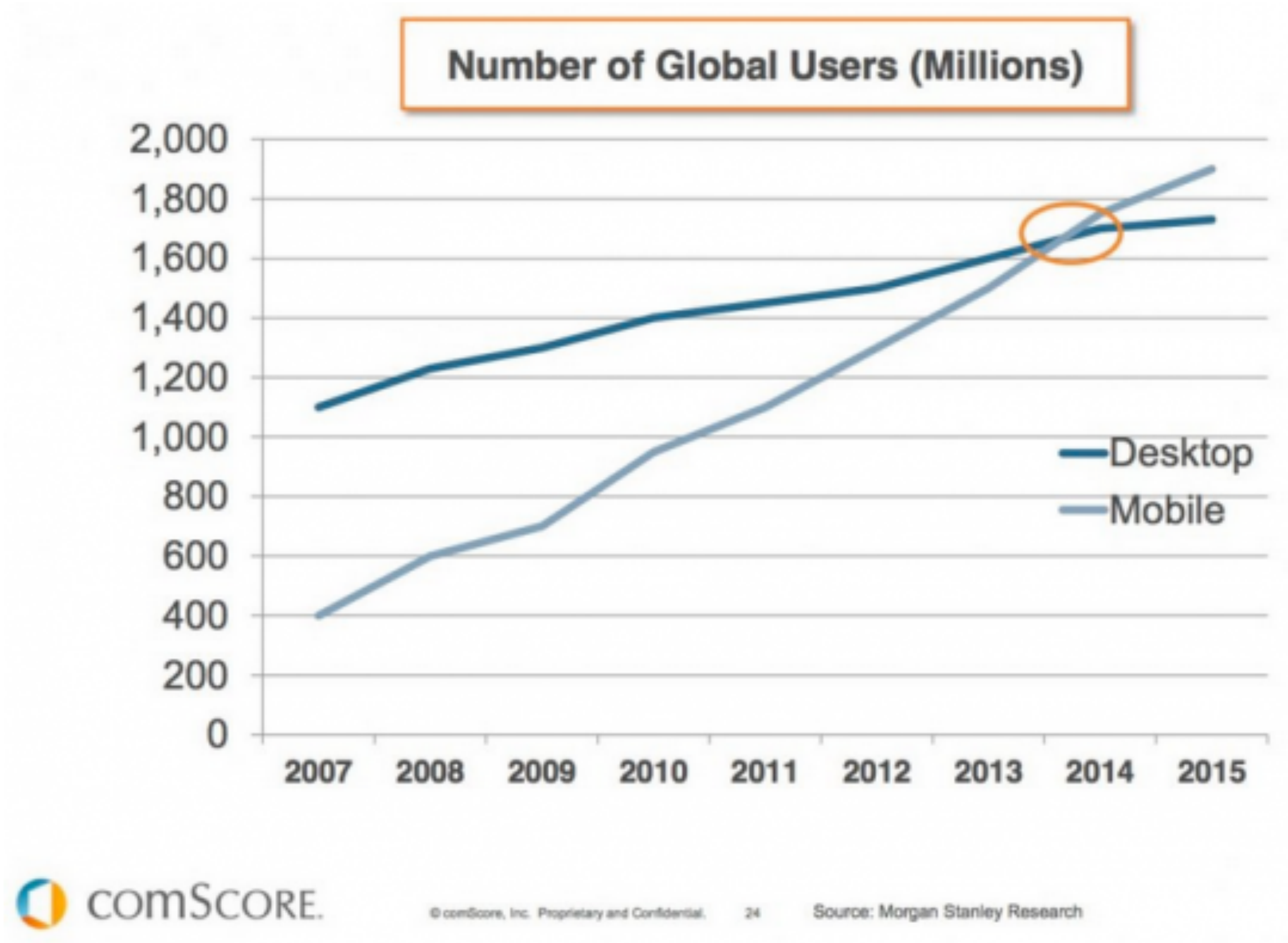


An ever-growing trend!

Mobile phones: The most successful technology **ever**!



Keep up with the Future!



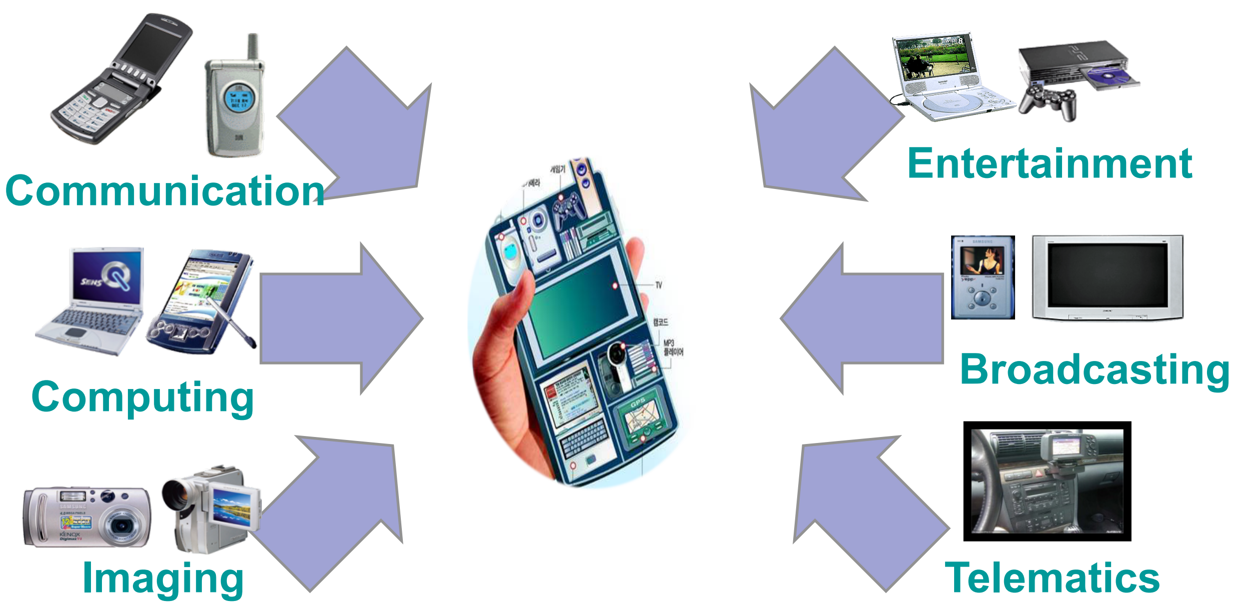
This is not going to stop soon! → Internet of Things

Differences between Emb. Systems and General Computing

- Time matters:
  - As fast as possible" is not always good enough
- Concurrency is intrinsic:
  - It's not an illusion (as in time sharing), and
  - It's not (necessarily) about exploiting parallelism
- Processor requirements can be specialized:
  - Predictable, repeatable timing.
  - Support for common operations (e.g. FIR filters)
  - Need for specialized data types (fixed point, bit vectors)
- Programs need to run (essentially) forever:
  - Memory usage has to be bounded (no leaks!!).
  - Rebooting is not acceptable.

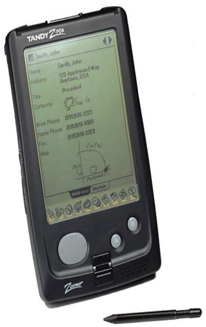
Embedded Systems Convergence

- Until 2000, embedded systems spread everywhere
  - **Cheap electronics** for cameras, portable gaming, etc.
- But since late 2000 all tend to converge
  - Everything in one device...



The difference vanishes…

Embedded System



General-Purpose Computer

The difference vanishes…

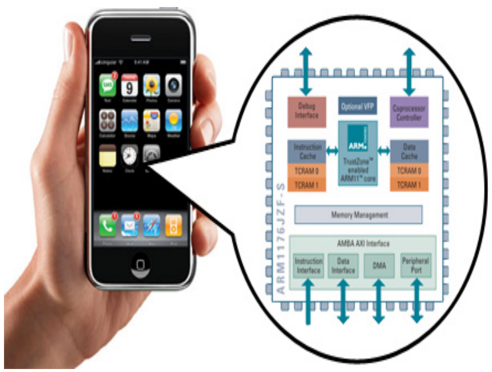
Embedded System?

General-Purpose Computer?

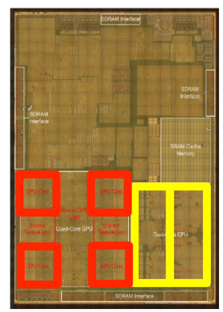
Maybe!?!

The New Reality of Embedded Systems

- New embedded systems are “ultra-light computers”
  - Complex computing systems, multiple uses (**15%** time calling [TechCrunch ‘15])



iPhone: ARM 1176JZF-S



iPhone 6: A8 MPCore (Dual-core CPU, 4-core GPU)

Apple Watch: APL 0778 processor (with 4GB, 5 sensors, loudspeaker, …)

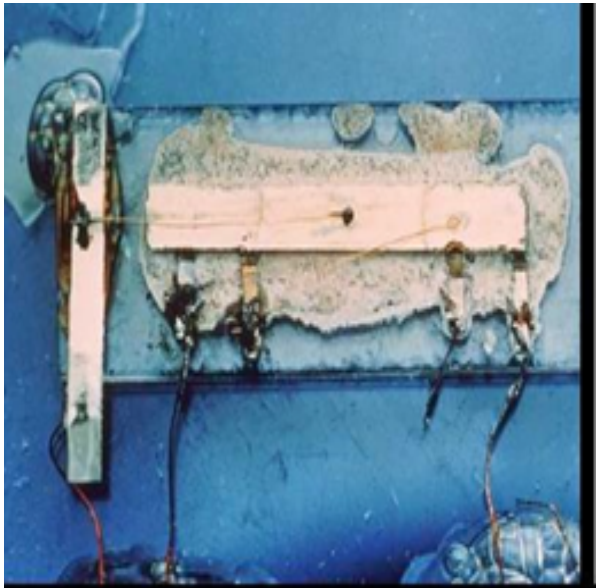
System-level co-design:

hardware-software (HW-SW)

interaction is key in new embedded systems

History Begins Some Time Ago…

- Linked to 1st Integrated Circuit (IC or “chip”): 1958



First Integrated Circuit (1958)
10 mm x 1.5 mm
1 transistor

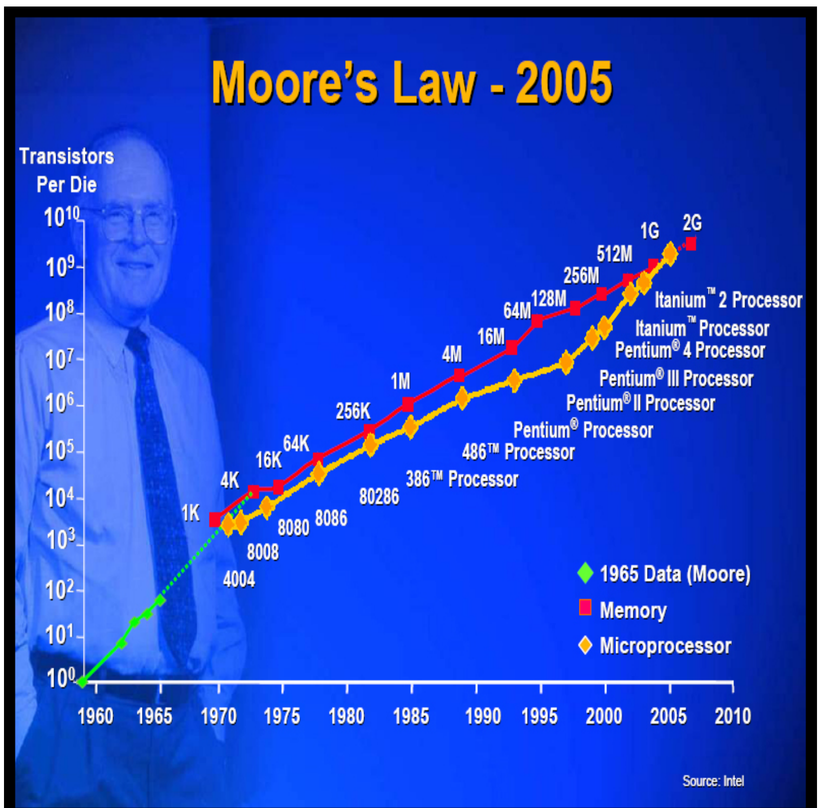
Jack Kilby (Texas Instruments)
Novel Prize in Physics (2000)
"for his part in the invention of the integrated circuit”

Since Then… 50 Years of Growth

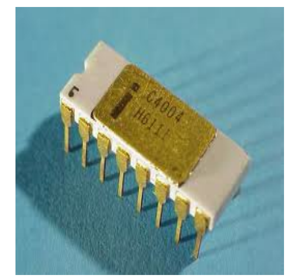
Since 1965 - Moore's Law:

Doubling the number of transistors in the same surface every 12 (or 24) months

Doubling chip performance every 18 months



Intel 4004,1971



92,000 ops/sec

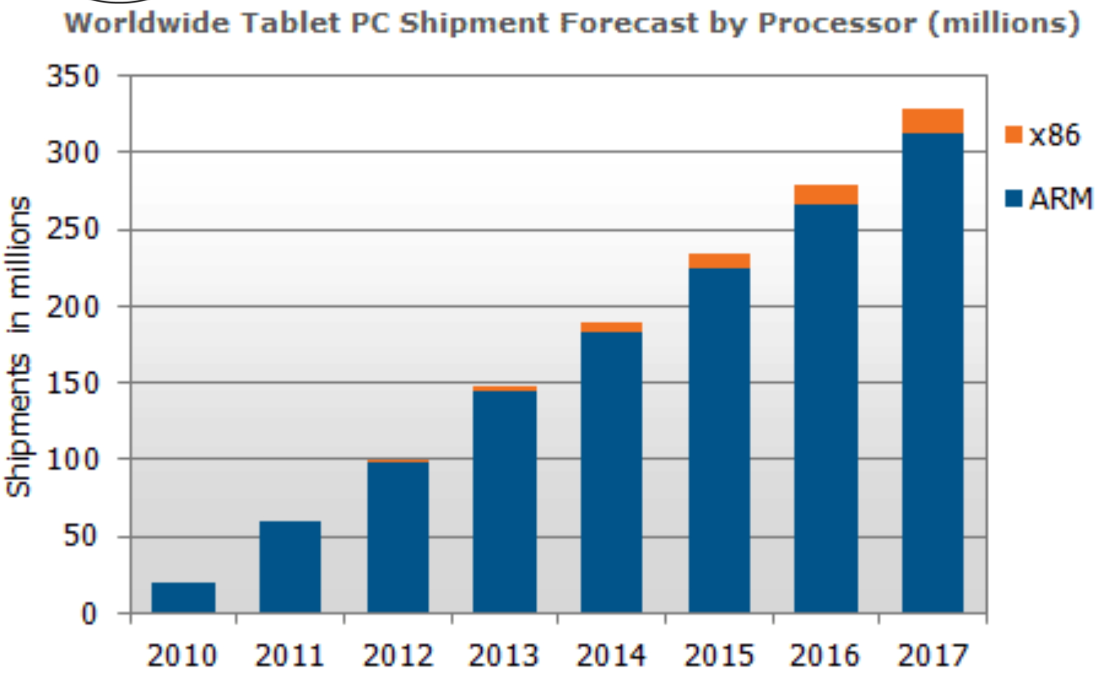
Intel Xeon, 2011

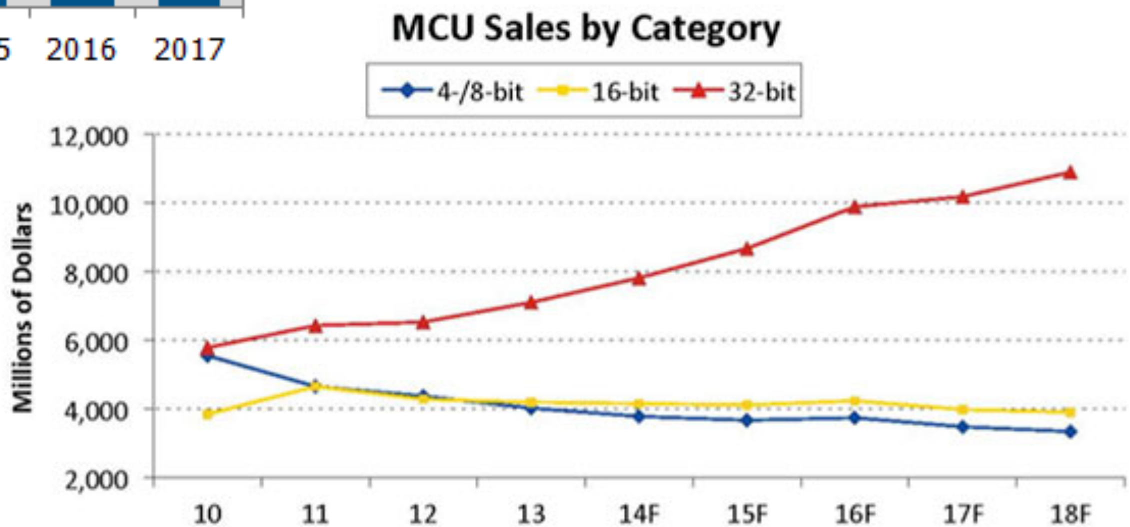
96,000,000,000 ops/sec

What is pushing Embedded Everywhere?

- Moore's Law
- Flash Memory Scaling
- MEMS Accelerometers: falling price
- Energy harvesting
- Big data: Need for gigantic quantity of data

Why study the ARM Cortex-M arch. ?





Lots of manufacturers propose ARM cores

- Microsemi
- Xilinx
- Atmel
- Cypress
- Dust networks
- Ember
- NXP
- ST
- Texas instruments
- …

How to differentiate the products?

Peripherals

- FPGA
- Sensors
- Memory
- Processor architecture
- ADC/DAC
- Sensors
- ...

Choosing the right MCU is a critical phase in the design of an embedded systems

Think twice for your project

During the first part of the class,
we'll use a STM microcontroller based on an ARM Cortex-M0

Things to do

- Labs
  - No labs this week
  - Get access to the Digital Lab MEB 2265
  - Form teams: One student per pair, email teach list
  - Buy the different hardware elements Discovery kits
  - Get a CADE account, if you don't have one

- Homework 1
  - Start working on your PCB design!
  - Read the datasheets
  - Prepare questions for Thursday and for your labs next week!

Thank you for your attention

Questions?

Integrated Nanosystems Research Group
Department of Electrical and Computer Engineering
MEB building – University of Utah – Salt Lake City – UT – USA