# Lecture 1
# Introduction

CE346 – Microprocessor System Design

Branden Ghena – Spring 2021

Some slides borrowed from:
Josiah Hester (Northwestern), Prabal Dutta (UC Berkeley)

# Welcome to CE346!

- Focus on hardware/software systems and their design
  - Hardware/Software co-design
    - How do you write software that interacts with hardware?
    - How do you choose hardware to support software needs?
  - Sensors and Sensing
    - What can sensors do and how do they work?
    - How do you write applications that sense the world?

# Today’s Goals

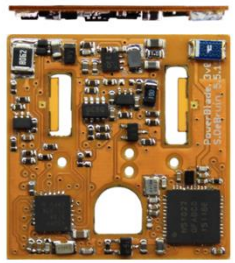
- What are the goals of this course?
- Why do I think embedded systems are so important?
- How is the course going to operate?

# Outline

- **Who and Why**
- Embedded Systems
- Course Overview

# Branden Ghena (he/him)

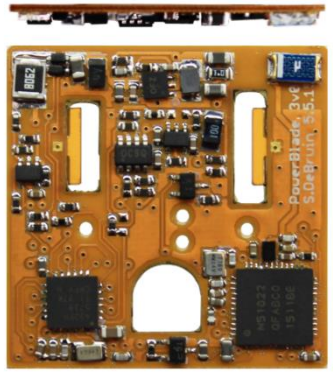
- Assistant Faculty of Instruction
- Education
  - Undergrad: Michigan Tech
  - Master's: University of Michigan
  - PhD: University of California, Berkeley
- Research
  - Resource-constrained sensing systems
  - Low-energy wireless networks
  - Embedded operating systems
- Teaching
  - Computer Systems
    - Intro to Computer Systems
    - Operating Systems
    - Wireless Protocols for the IoT

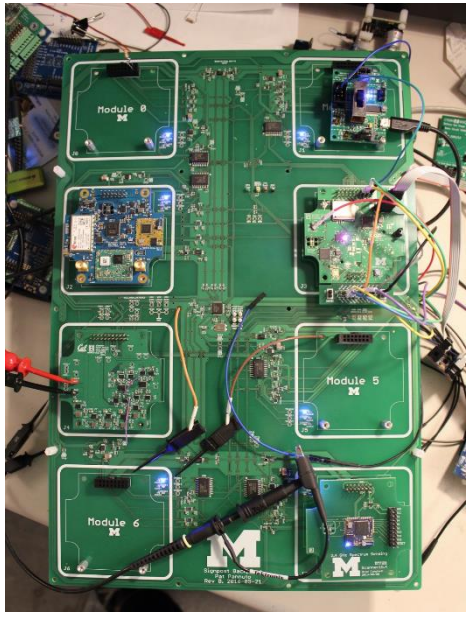



Things I love

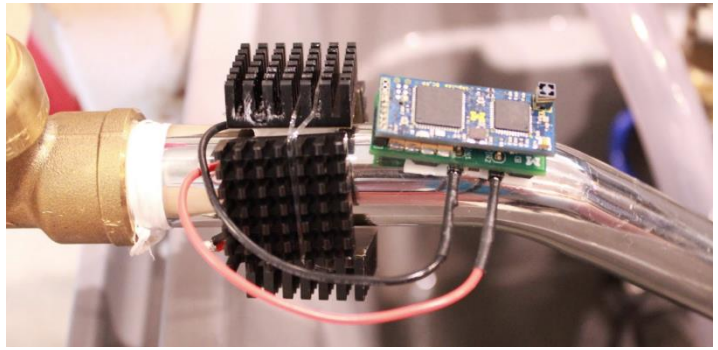



# Research area: resource-constrained embedded systems






- Most interesting to me: the interfaces
  - Hardware and software
  - Applications and OS
  - Communication



# Faculty: now I can choose what to teach!

- Goal: provide classes that teach more advanced embedded systems topics
  - Hopefully, generally useful to other nearby domains of CS and ECE too!

- An immediate result: this course!
  - Really would be much nicer in person, but 
  - Course goal: introduce students to hardware-software interactions
    - Practical hands-on experience with microcontrollers and sensors
    - Open-ended project where students can choose their specific focus

# Outline

- Who and Why
- **Embedded Systems**
- Course Overview

# What is an embedded system?

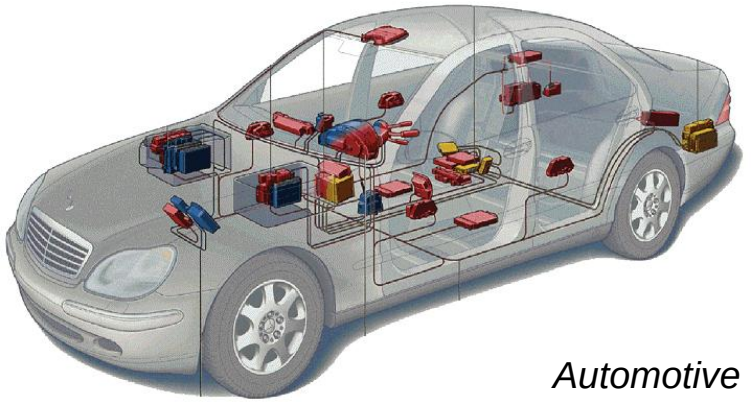
- A computer built into a device
  such that the device is interacted with, not the computer
  - Not a desktop, laptop, server, smartphone, smartwatch
  - (although many of those deal with overlapping hardware/software issues)

- Many domains
  - Robotics
  - Industrial processes
  - Smart home
  - Smart city
  - Wearables and health sensing
  - Generally: Internet of Things

# Related area: Cyber-Physical Systems

- Systems that are part computational and part real-world
  - Example: autonomous vehicles

- Combines multiple fields to handle this problem
  - Embedded Systems
  - Electronics
  - Controls
  - Software Engineering
  - Computer Theory



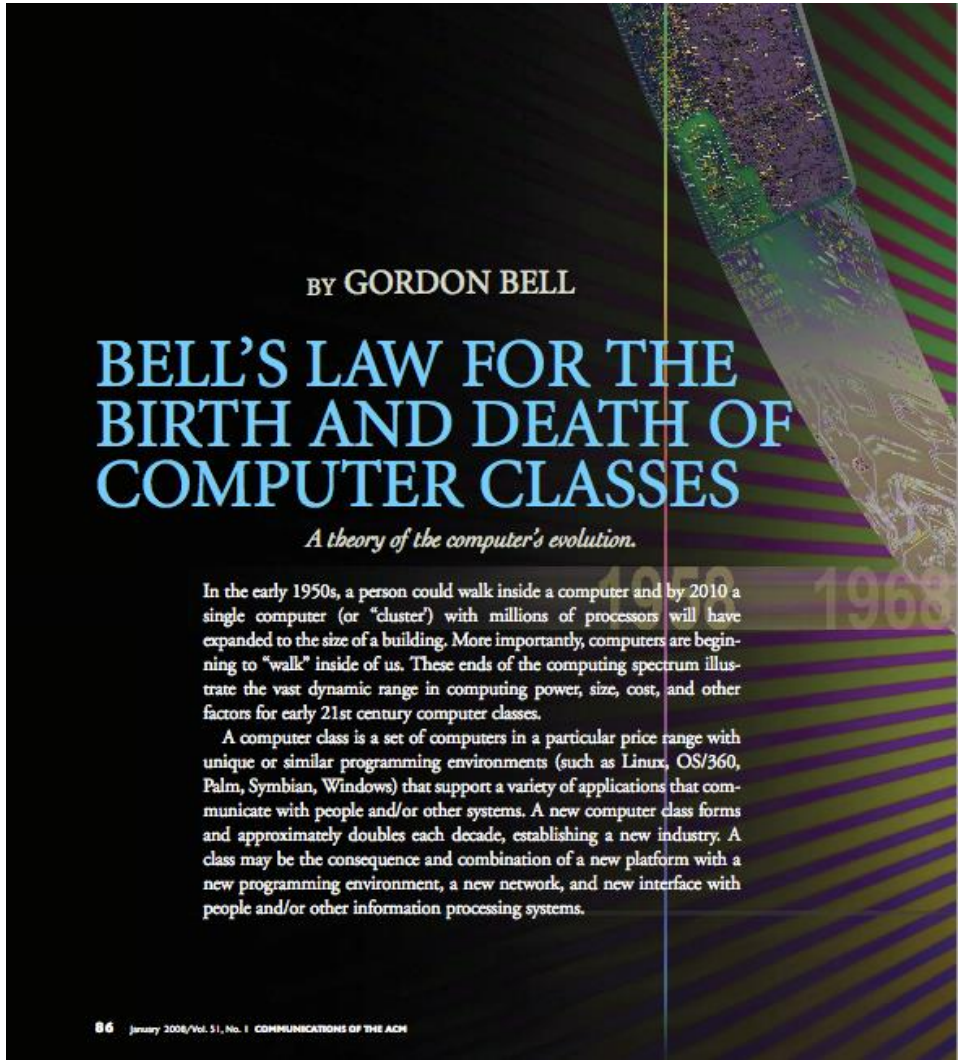
*Automotive*

# The Internet of Things

# Bell's Law: A new computer class every decade

*"Roughly every decade a new, lower priced computer class forms based on a new programming platform, network, and interface resulting in new usage and the establishment of a new industry."*

- Gordon Bell [1972,2008]



BY GORDON BELL

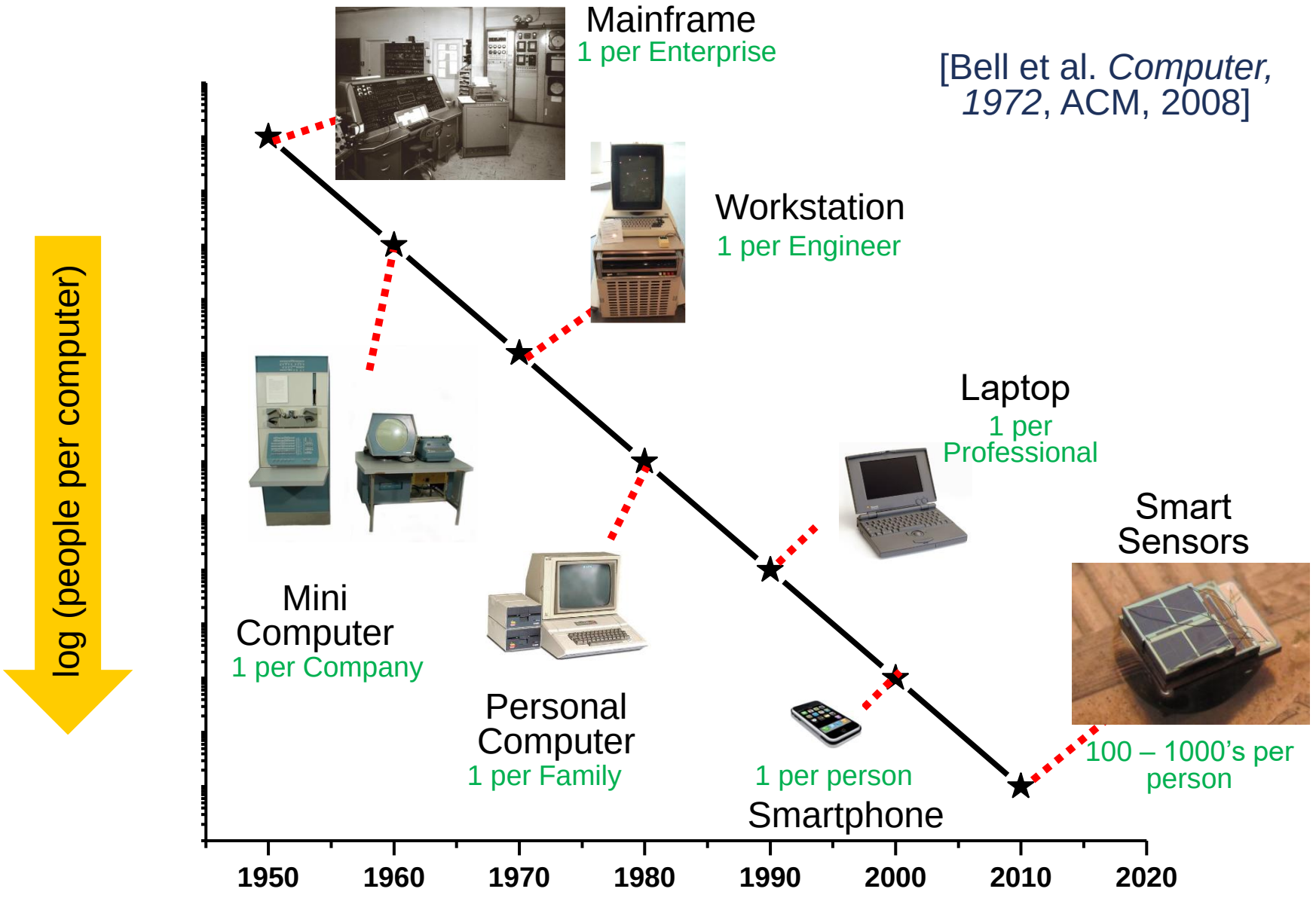
BELL'S LAW FOR THE BIRTH AND DEATH OF COMPUTER CLASSES

*A theory of the computer's evolution.*

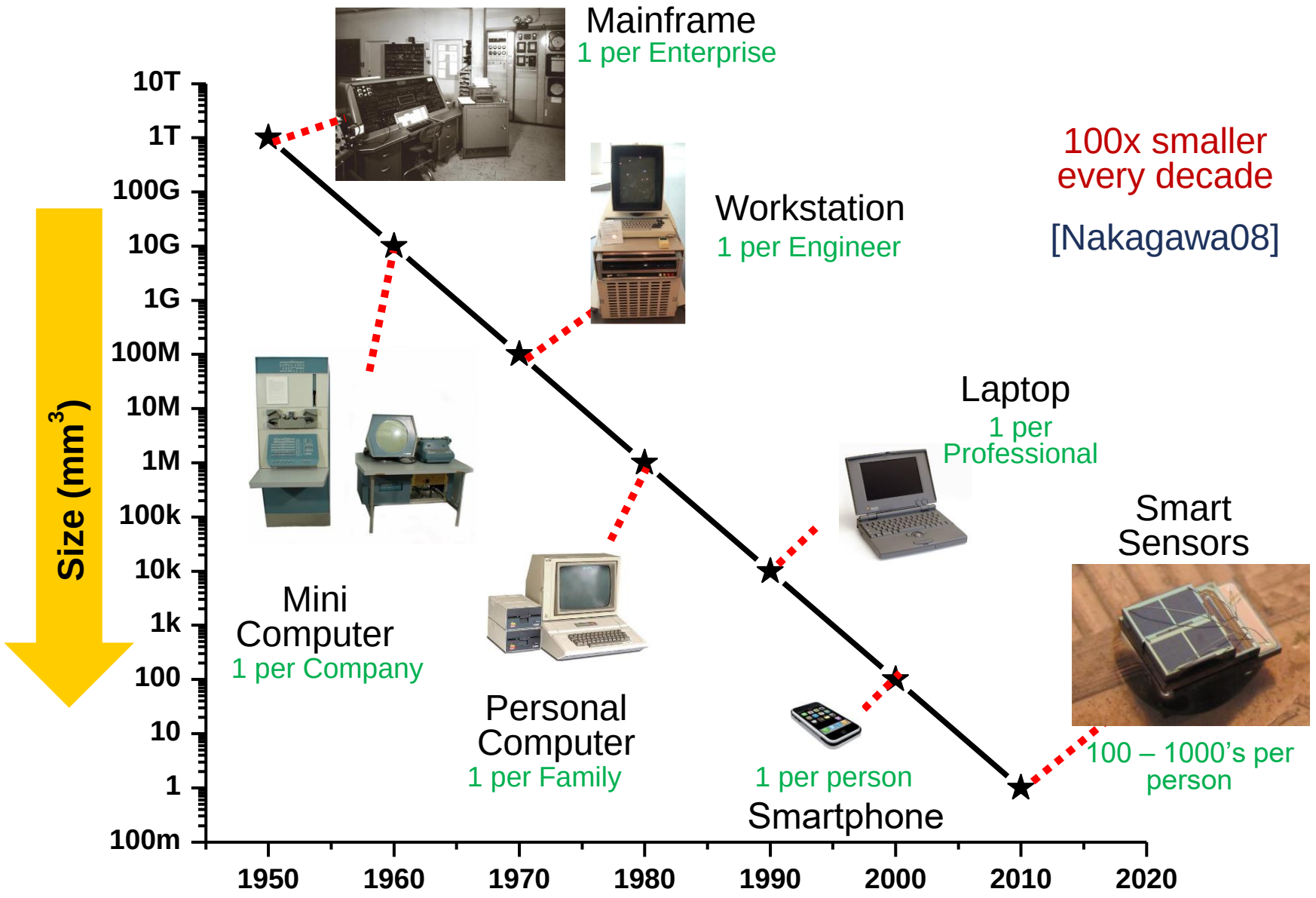
In the early 1950s, a person could walk inside a computer and by 2010 a single computer (or "cluster") with millions of processors will have expanded to the size of a building. More importantly, computers are beginning to "walk" inside of us. These ends of the computing spectrum illustrate the vast dynamic range in computing power, size, cost, and other factors for early 21st century computer classes.

A computer class is a set of computers in a particular price range with unique or similar programming environments (such as Linux, OS/360, Palm, Symbian, Windows) that support a variety of applications that communicate with people and/or other systems. A new computer class forms and approximately doubles each decade, establishing a new industry. A class may be the consequence and combination of a new platform with a new programming environment, a new network, and new interface with people and/or other information processing systems.

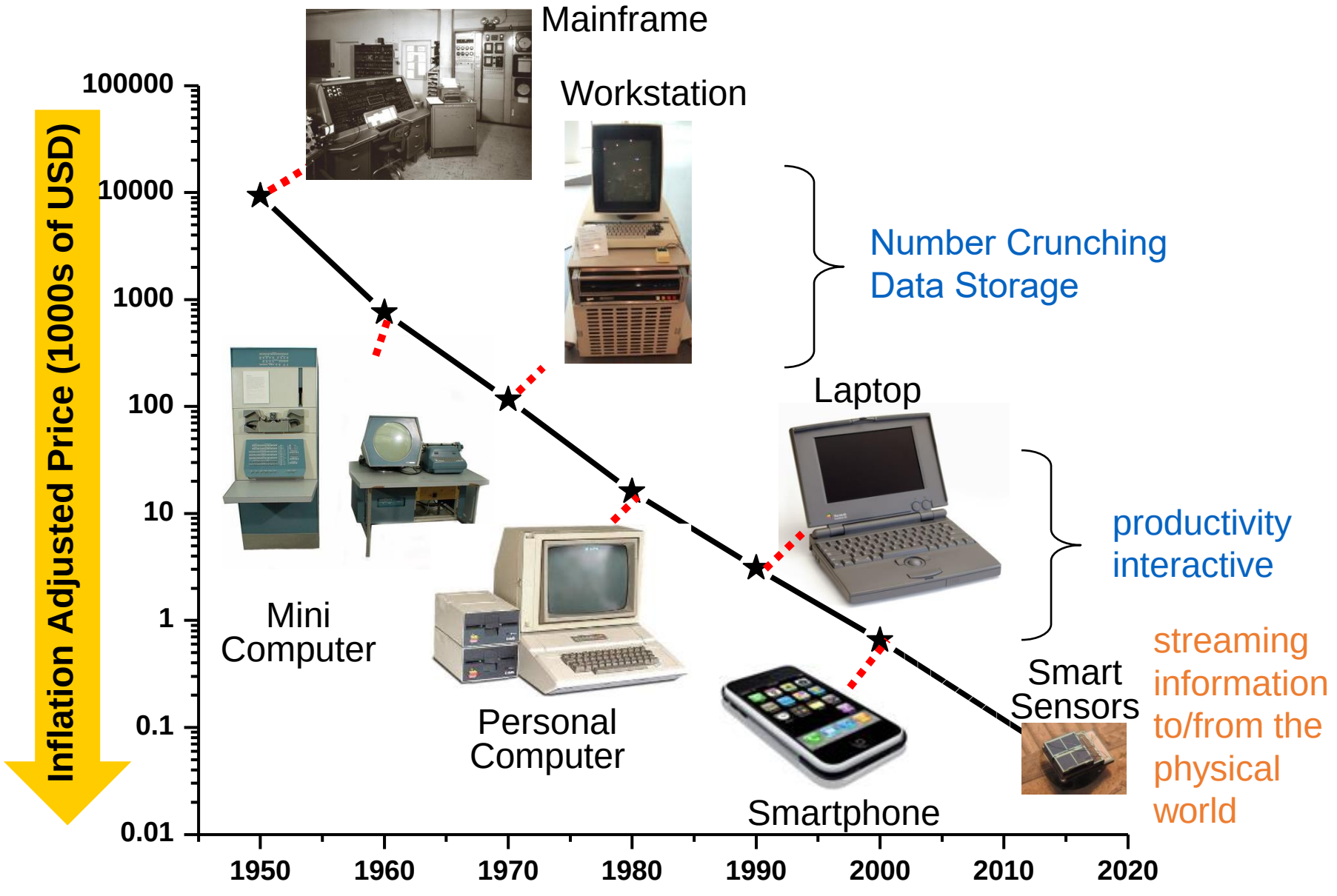
# Number of computers per person grows over time



# Computer volume shrinks by 100x every decade



# Price falls dramatically, enabling new applications



# Discussion: what is the Internet of Things?

1. Name a few Internet of Things devices

2. What are the qualities that designate those devices at “IoT”?

# Discussion: what is the Internet of Things?

1. Name a few Internet of Things devices

2. What are the qualities that designate those devices at “IoT”?





# Thought experiment on capabilities

- What if the Nest thermostat was powered by an entire desktop?
  - 8-core x86-64 processor, 32 GB RAM, 1 TB SSD
- Would that still count as IoT?
- Could that still be an embedded system?
- Why don’t we see that in practice?

# Thought experiment on capabilities

- What if the Nest thermostat was powered by an entire desktop?
  - 8-core x86-64 processor, 32 GB RAM, 1 TB SSD
- Would that still count as IoT?
  - **Doesn’t really feel right. Built in assumption of limitations.**
- Could that still be an embedded system?
  - **Yes**
- Why don’t we see that in practice?
  - **Cost**

# Thought experiment on energy

- IoT devices include a mix of batteries, wall power, (and energy-harvesting)

- Why do we put so much focus on systems with batteries?
  - Why do they need batteries?

# Thought experiment on energy

- IoT devices include a mix of batteries, wall power, (and energy-harvesting)

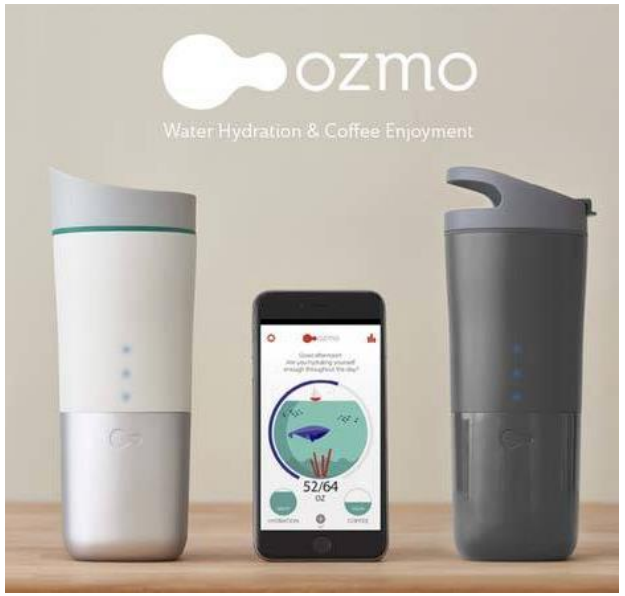
- Why do we put so much focus on systems with batteries?
  - Why do they need batteries?

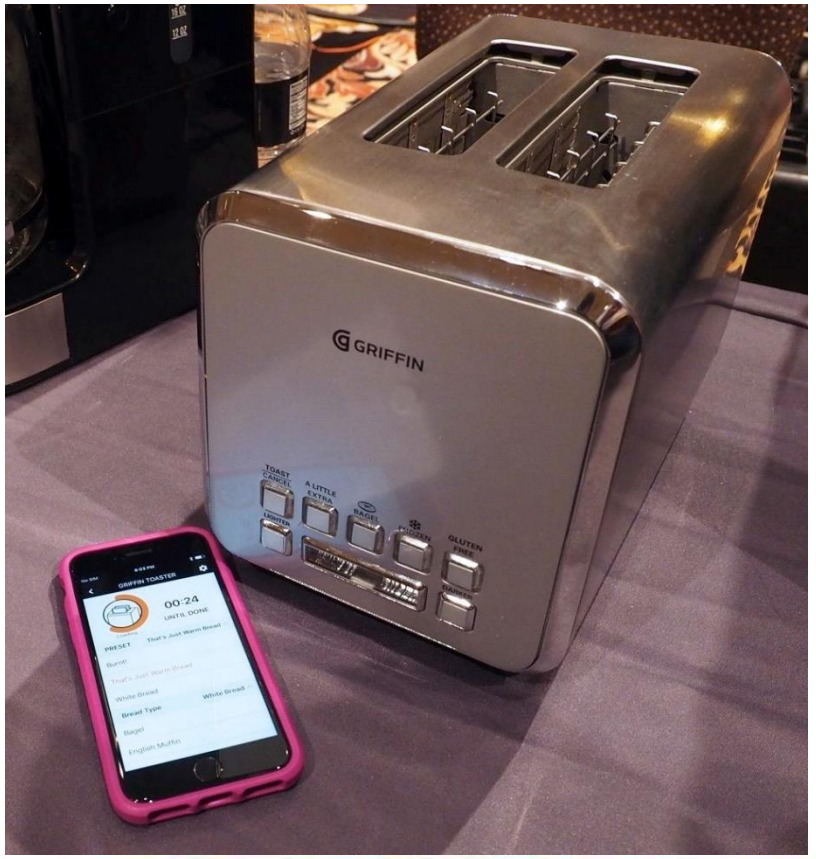
**Deployability**

# Branden's take on the Internet of Things

- Key features
  - Computation
    - Local to the device
    - With some capability for arbitrary compute and storage
  - Connectivity
    - Almost certainly wireless
    - Likely Internet, possibly local
  - Interaction
    - Sensing or Actuation
- Secondary features
  - Low energy
  - (Relatively) Low cost

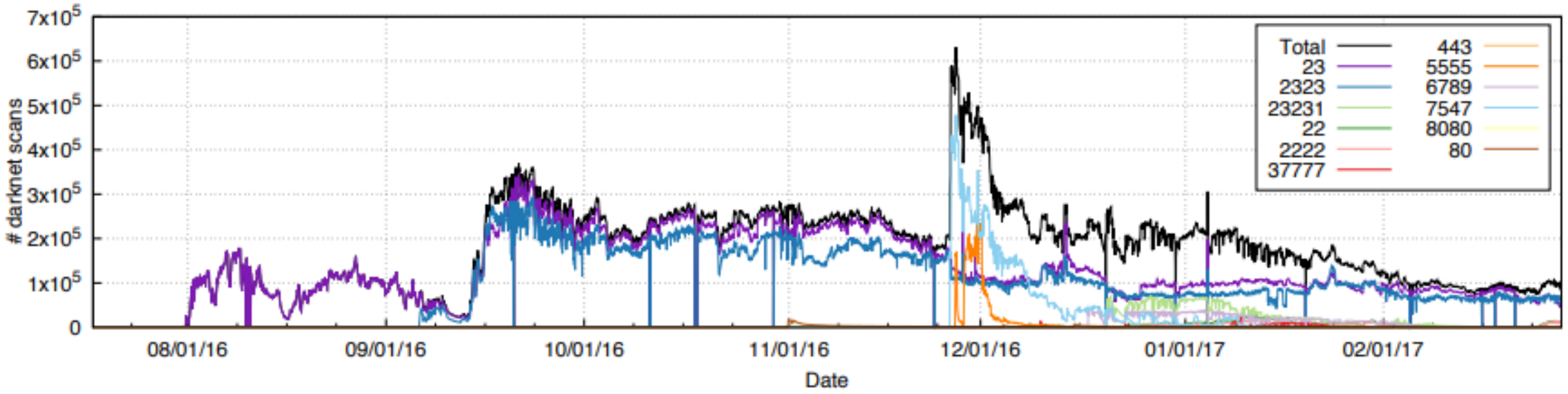
# Warning: Internet of Crap



# Internet of Insecure Crap




- Mirai botnet (2016)
- Takes control of up to 600,000 insecure connected devices
  - IP-attached cameras, DVRs, routers, printers
- Used to DoS websites

# What makes resource-constrained embedded systems interesting?

- Focus on the real world
  - You can actually see the purpose and effects of your applications
  - Easily explainable to non-engineer humans

- Challenging limitations
  - Limited memory and processing
  - Energy concerns

# What makes resource-constrained embedded systems frustrating?

- Challenging limitations
  - Limited memory and processing
  - Energy concerns

- Full-stack development means problems could be *anywhere*
  - Hardware problems
  - Firmware problems
  - Software problems

# Outline

- Who and Why
- Embedded Systems
- **Course Overview**

# Course Time

- M/W/F 10:00-10:50am (not using the Tuesday slot)

- This week: Lecture on Tuesday, Wednesday, and Friday

- Starting next week:
  - Lecture: Monday and Wednesday
  - Lab: Friday
    - Attempt to recreate lab environment over zoom
    - Breakout groups to encourage chatting and asking questions

# Course Staff and Office Hours (starting next week)

- Abu Bakar

- Office hours
  - Tuesday 11:00–1:00pm
  - Thursday 9:00–11:00am

- Julian Richey

- Office hours
  - Tuesday 3:00–5:00pm
  - Thursday 3:00–5:00pm

- My office hours: (starting this week)
  - Monday 11:00-12:00 pm
  - Wednesday 1:00-2:00 pm
  - Or by request!

# Course Communication

- Campuswire
  - You should all already be registered on it
  - If you aren't, let me know! (or if you want to change emails)

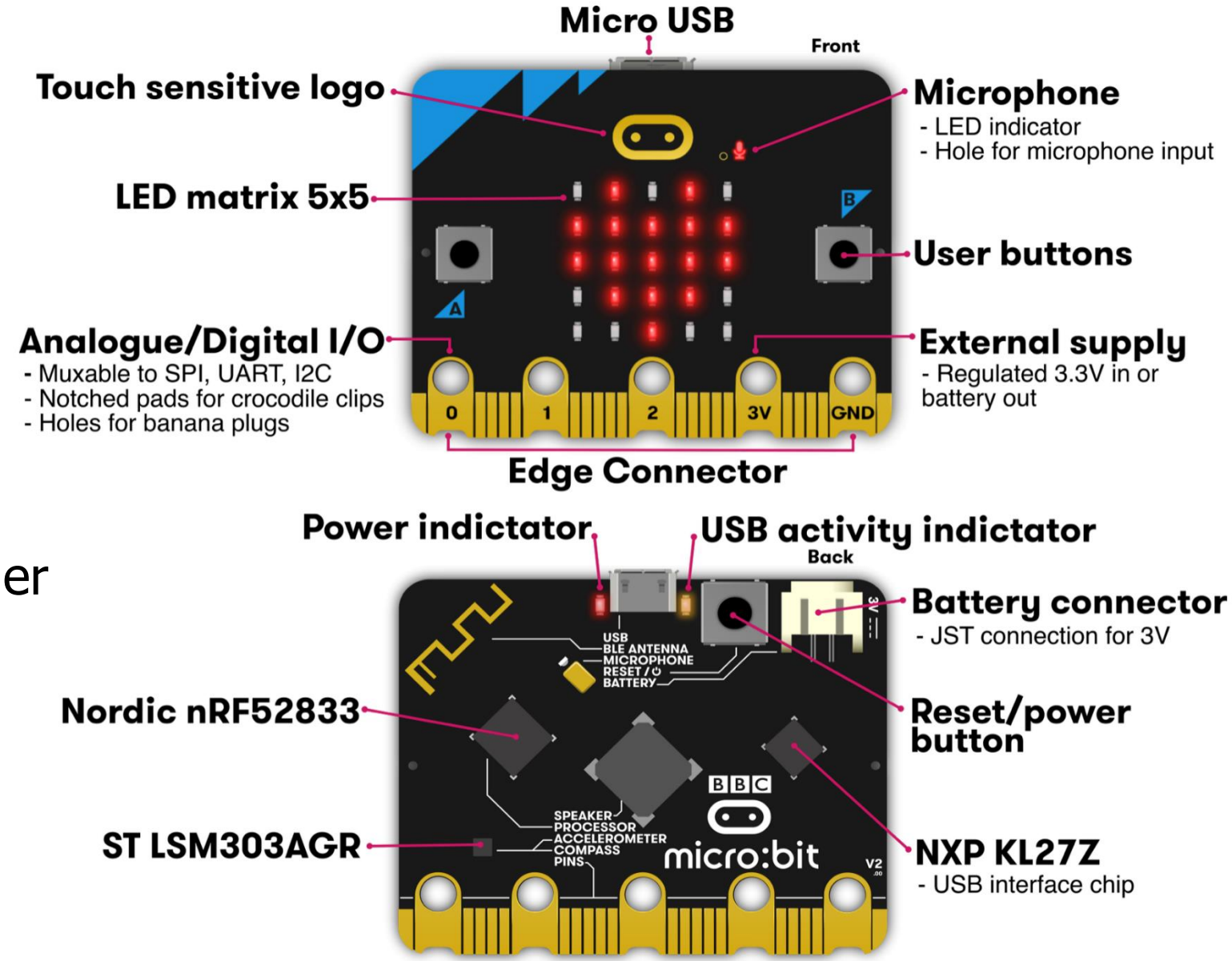
- All course communication goes through Campuswire, not email
  - Multiple people can respond to you
  - Messages are kept in one place and stay "unanswered"
  - You can post directly to "Instructors & TAs" if it is private

# Course grade components

- 45% Labs
  - 6 labs at 7% each, plus 3% for “Getting Started Lab”
  - Guided exploration of course concepts
- 30% Quizzes
  - 4 timed quizzes at 7.5% each
  - Covers lecture material from last two weeks
- 25% Final Project
  - Proposal due 2/3s of the way through class
  - Use knowledge from lab to create your own project

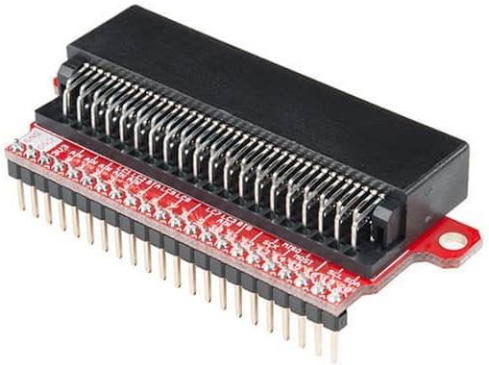
# Micro:bit v2

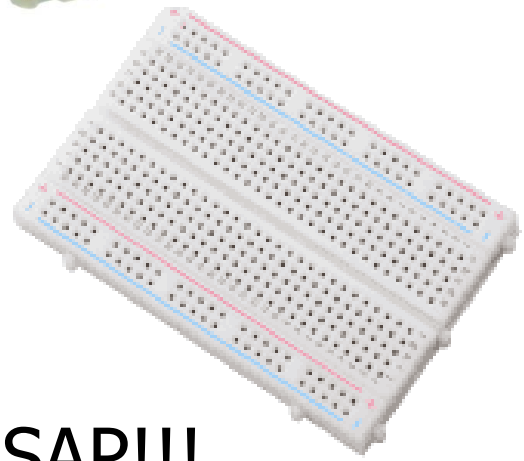
- Legacy from 1980s "BBC Computer Literacy Project"
- Under $20
  - Modern microcontroller AND sensors
- Plan for class:
  - Explore most of its functionality



# Lab kit

- Everyone gets
  - Microbit
  - Microbit-to-breadboard interface
  - Breadboard
  - Various components
  - USB micro cable



- Fill out survey from Campuswire ASAP!!!
  - I need your addresses to send you stuff
  - I can wait a few days if you're not sure if you're staying in the course
  - Let me know if you're international, and we'll figure it out

# Labs

0. Getting Started (later this week)
1. MMIO and Interrupts
2. Virtual Timers
3. LED Matrix
4. Breadboarding
5. Audio Input/Output
6. I2C Accelerometer/Magnetometer

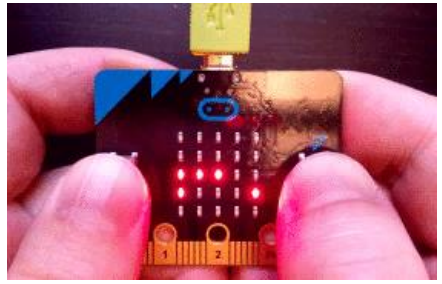
- Labs will be individual work
- Due end-of-day the next Friday (a week plus 14 hours after assigned)

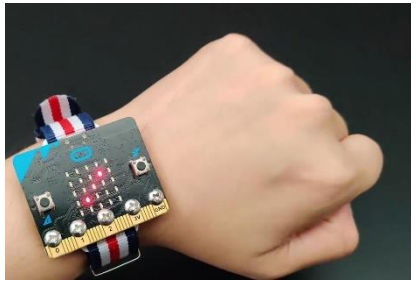
# Final projects

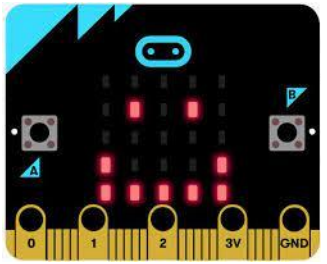
- Opportunity for you to apply your interests to this course
  - In groups of 1-2 students

- Demonstrate course knowledge through any application
  - Limited by the parts in the kit, unless you have other things on hand

- Some ideas (just with kit parts)
  - Video game with motion controls
  - Music pad controller
  - Morse code decoder
  - Smartwatch
  - Tamagotchi





# Deadlines

- Labs are due a week from release date
  - Require checkoffs from course staff during Lab or Office Hours
- Quizzes will also have a due date
  - And a limited time window for completion once started
- Late penalty: 10% deduction from max score per day

# Flexibility

- Sometimes stuff just doesn’t work
  - Especially when we’re working with hardware

- We can be flexible about those deadlines
  - If you’re having problems and **tell us**
  - Less flexible if you don’t communicate or if you started late

- Takeaway: let us know if you’re having problems
  - Especially being remote, it’s hard for us to spot this

# Outline

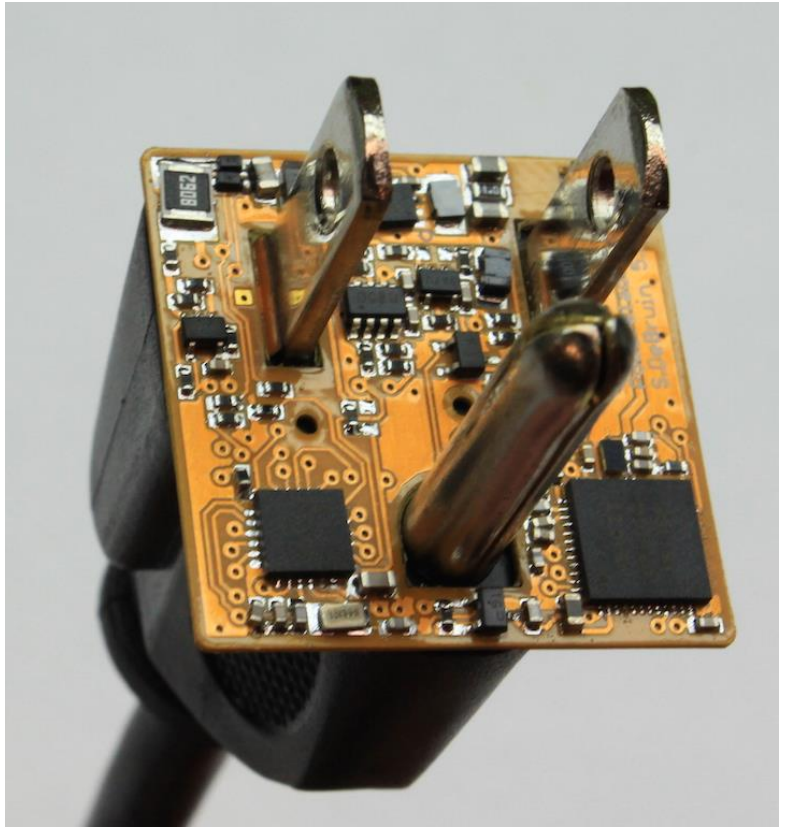
- Who and Why
- Embedded Systems
- Course Overview

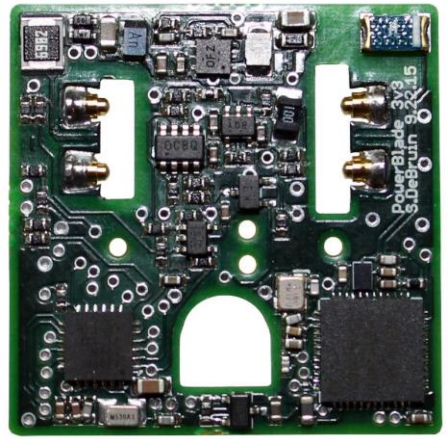
# Outline

- Bonus: embedded systems research

# PowerBlade – Smart Home

- Plug-load power meter
  - How do we measure *every* device in a home?
- Challenges
  - Deployability
  - Powering it
  - Sensing AC current and voltage
  - Reporting measurements

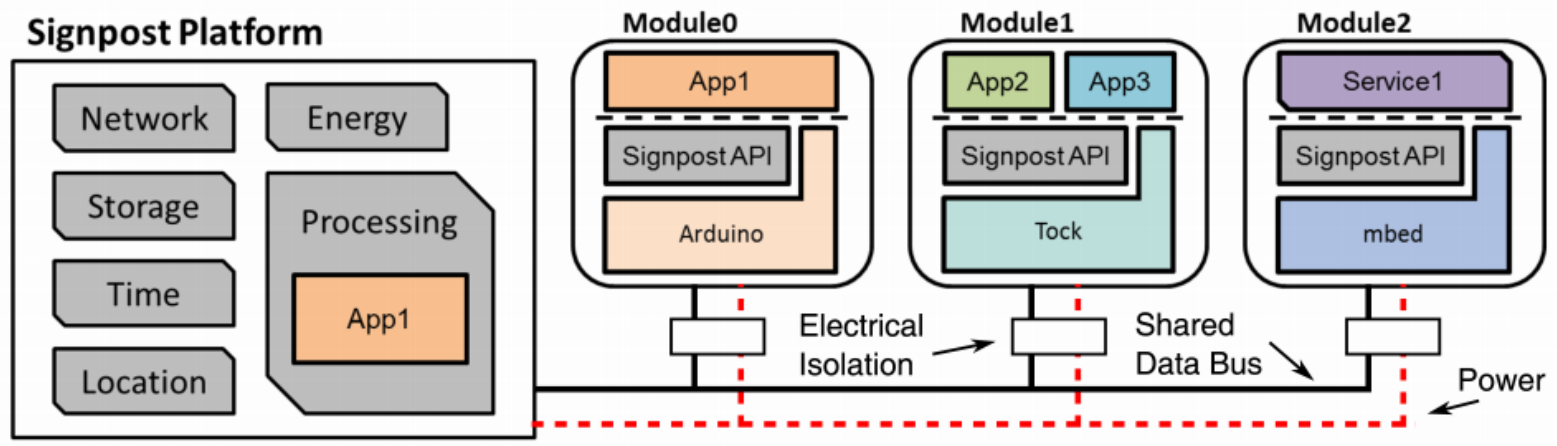


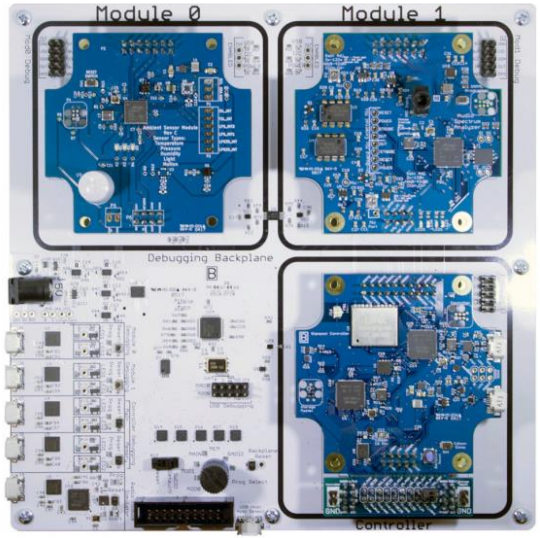


https://www.youtube.com/watch?v=oNUXhCDnHoE

# Signpost – City-Scale Sensing

- How do we reduce the burden of city-scale sensing experimentation?
- Platform provides resources
  - Modules provide sensor and application



# Tock – Reliable, security-conscious embedded OS in Rust