Lecture 1
Introduction

CE346 – Microprocessor System Design

Branden Ghena – Fall 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?

COVID

- We’re all figuring this out together
  - Please be patient and empathetic, and we will be too
- Masks in class/office hours are **mandatory**
- If you are sick, do not come to class
  - We’ll be flexible with deadlines if we need to be
  - Lectures are being recorded automatically
- Contact me (via Campuswire) and we’ll figure it out

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

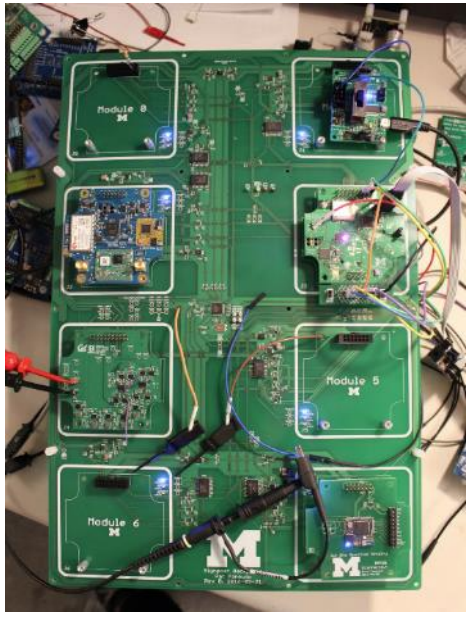
Branden Ghena (he/him)

- 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

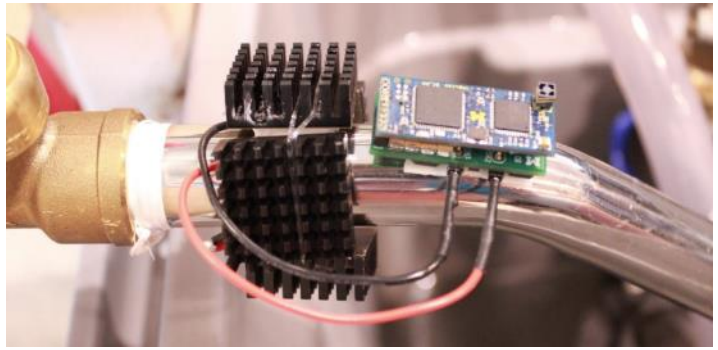
Things I love

Tock

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!
  - And this year we get to be in person!!
  - 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?

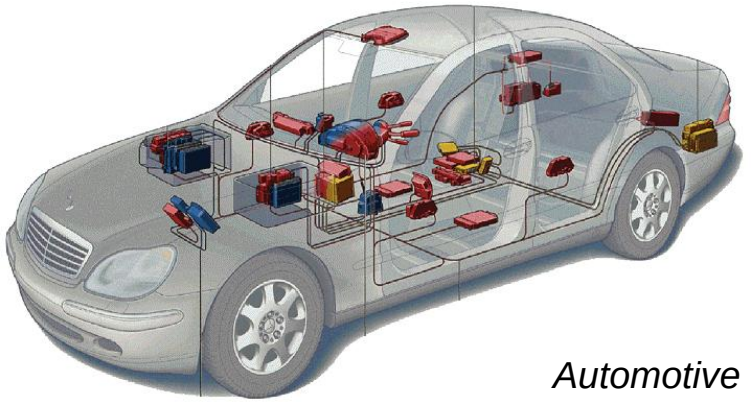
- A computer built into a device
  such that the device is interacted with, not the computer
  - Not a desktop, laptop, server, smartphone, smartwatch
  - (although many of those deal with overlapping hardware/software issues)

- 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



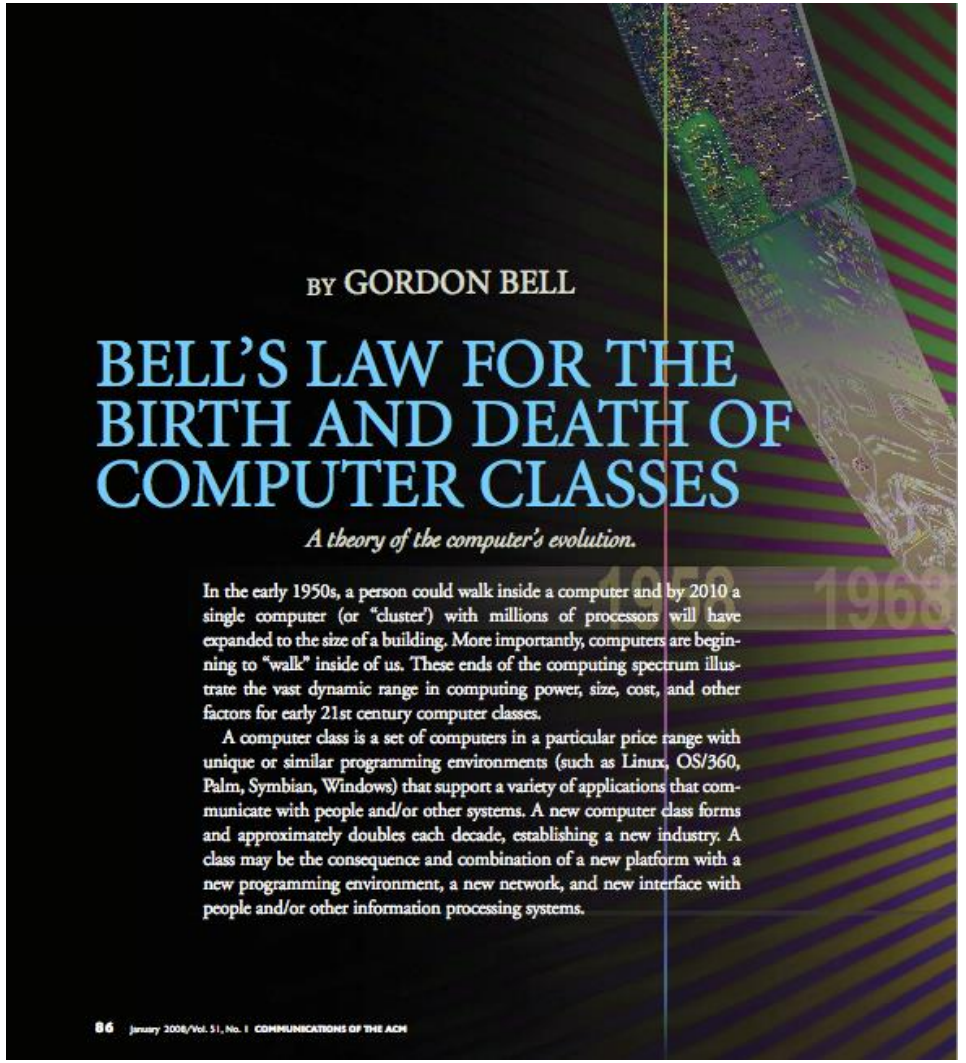
Automotive

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

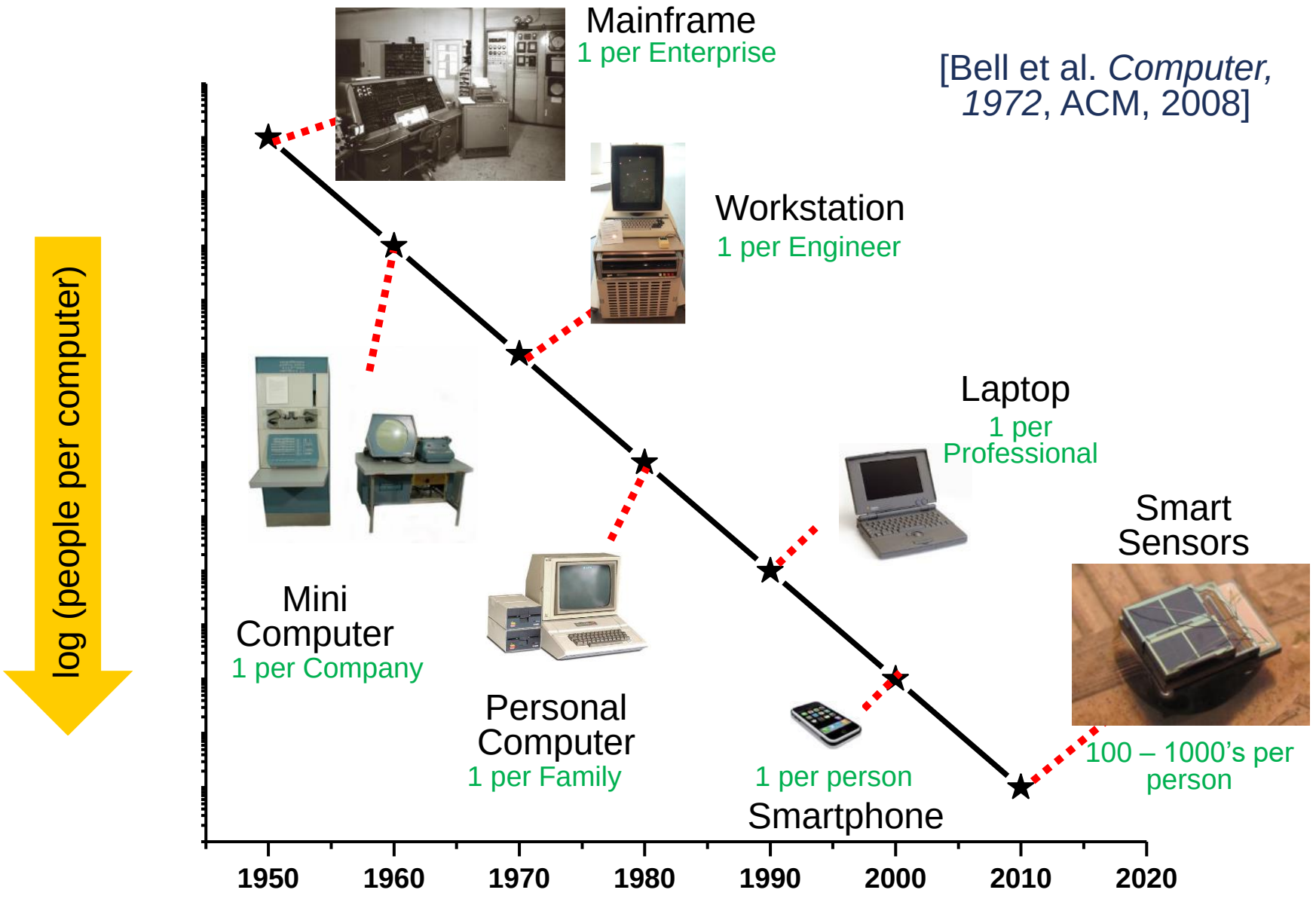
BELL'S LAW FOR THE BIRTH AND DEATH OF COMPUTER CLASSES

A theory of the computer's evolution.

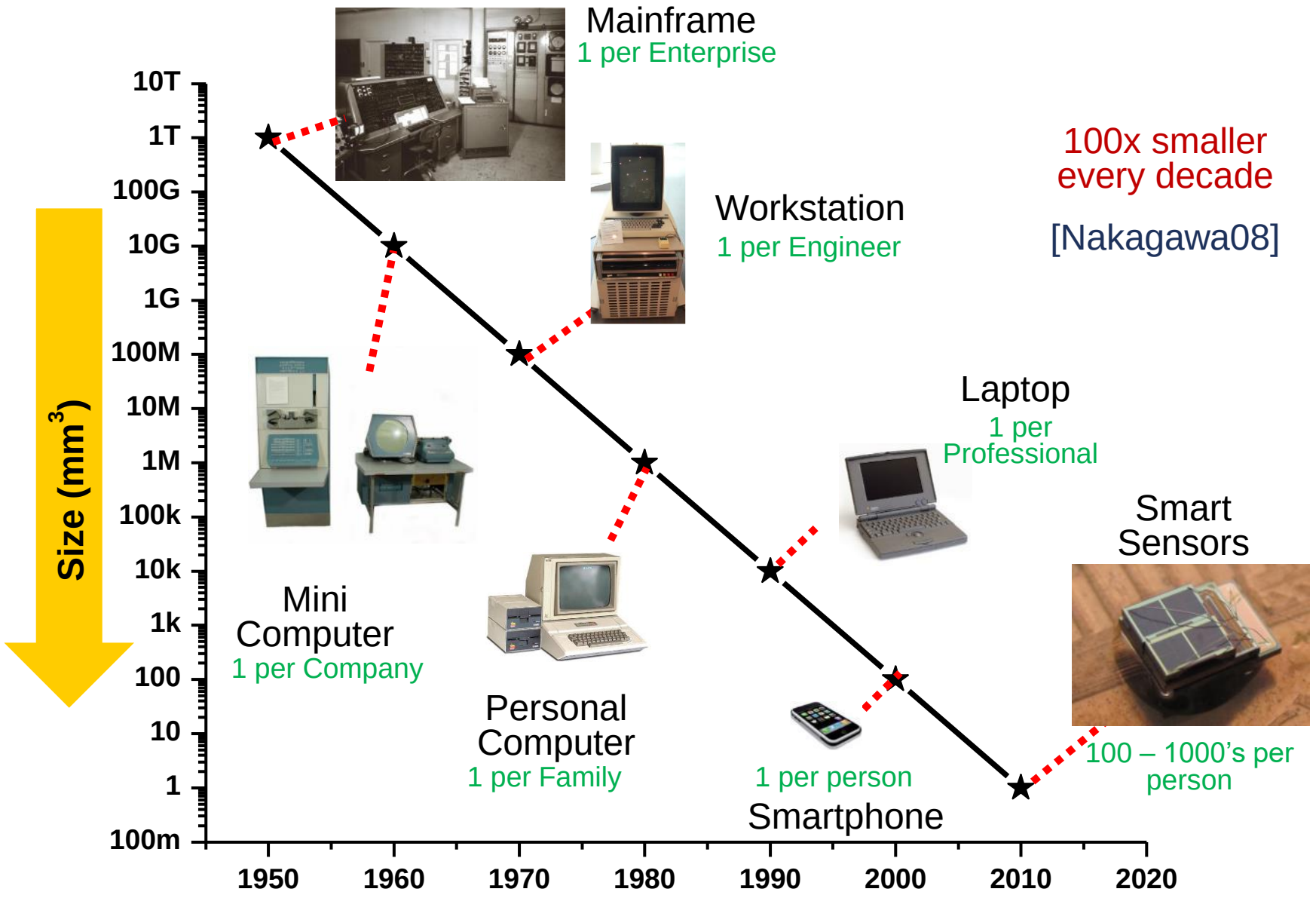
In the early 1950s, a person could walk inside a computer and by 2010 a single computer (or "cluster") with millions of processors will have expanded to the size of a building. More importantly, computers are beginning to "walk" inside of us. These ends of the computing spectrum illustrate the vast dynamic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

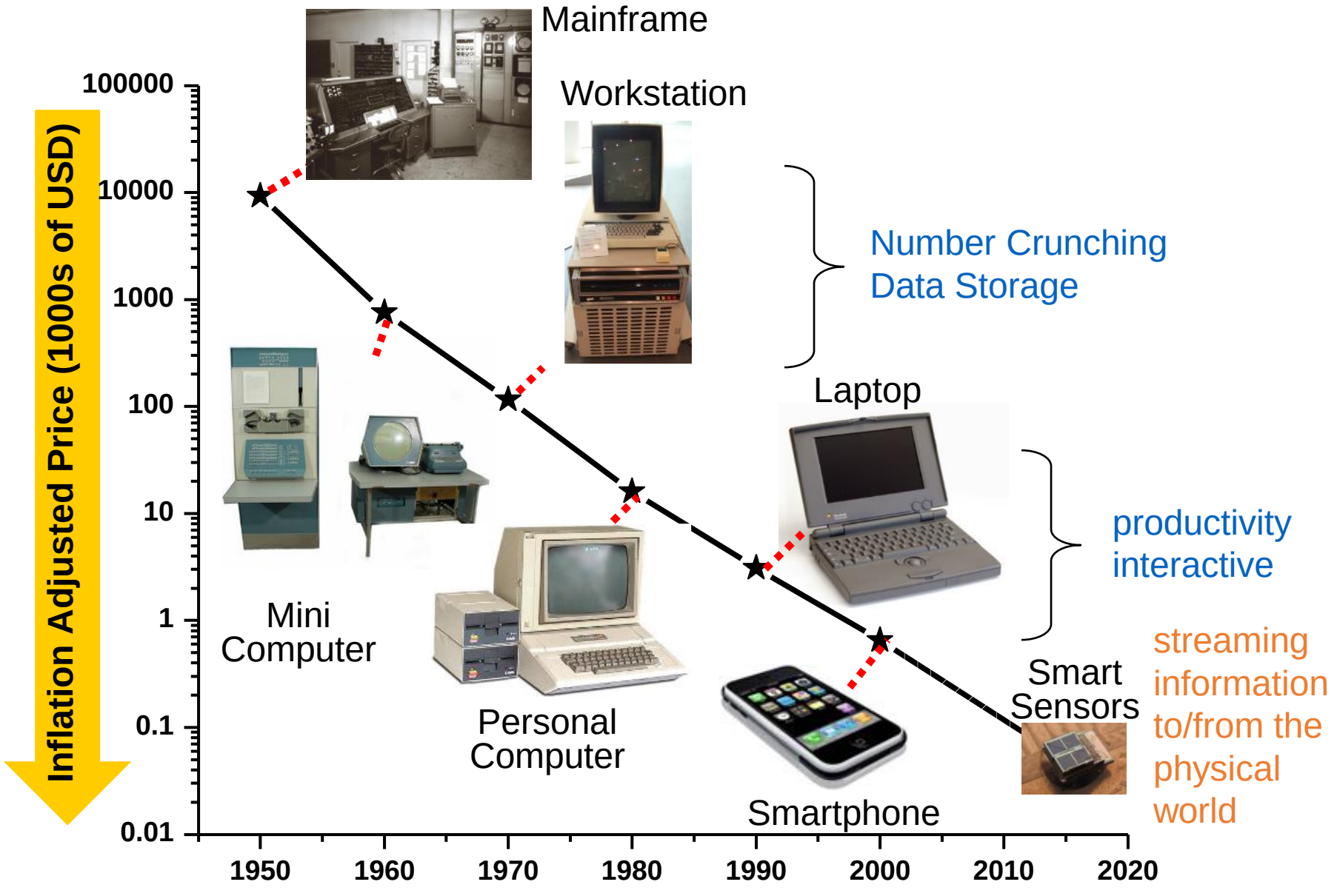
Number of computers per person grows over time



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)

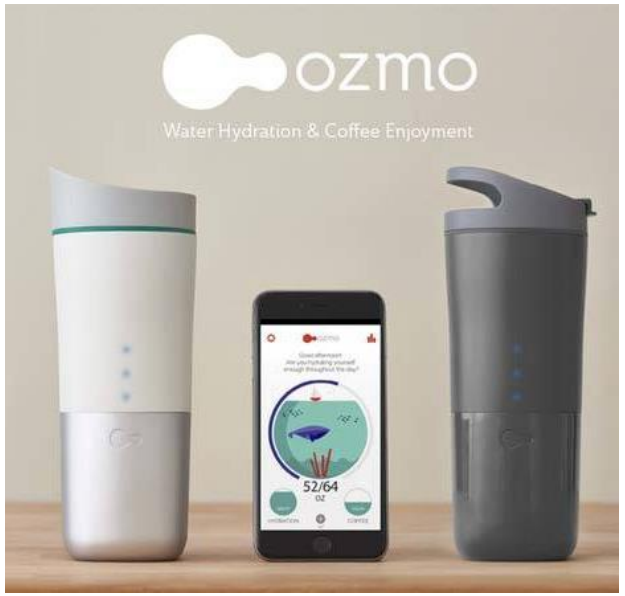
- Why do we put so much focus on systems with batteries?
  - Why do they need batteries?

Deployability

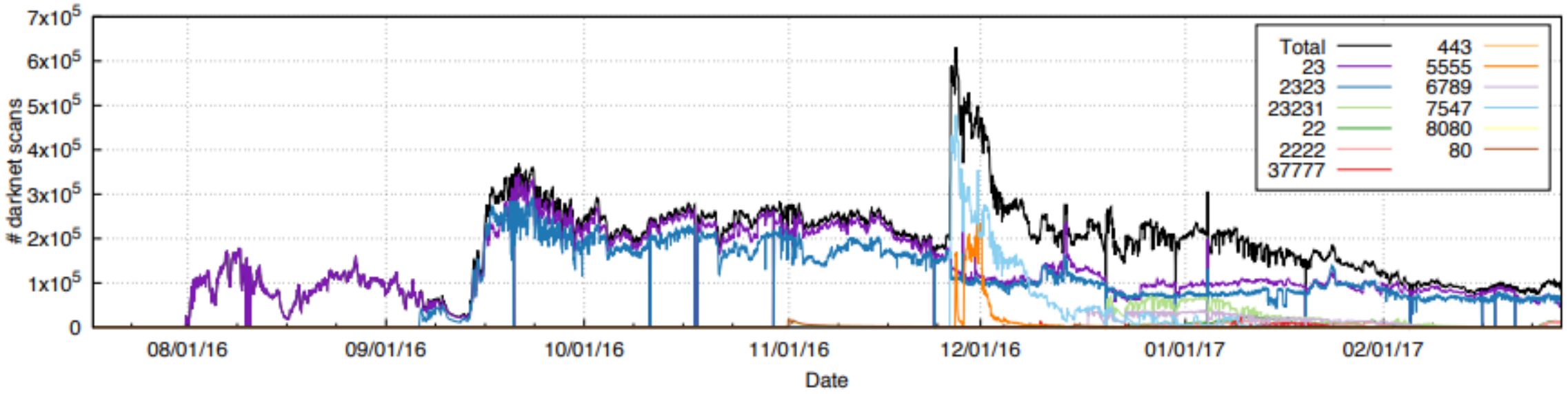
Branden's take on the Internet of Things

- 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

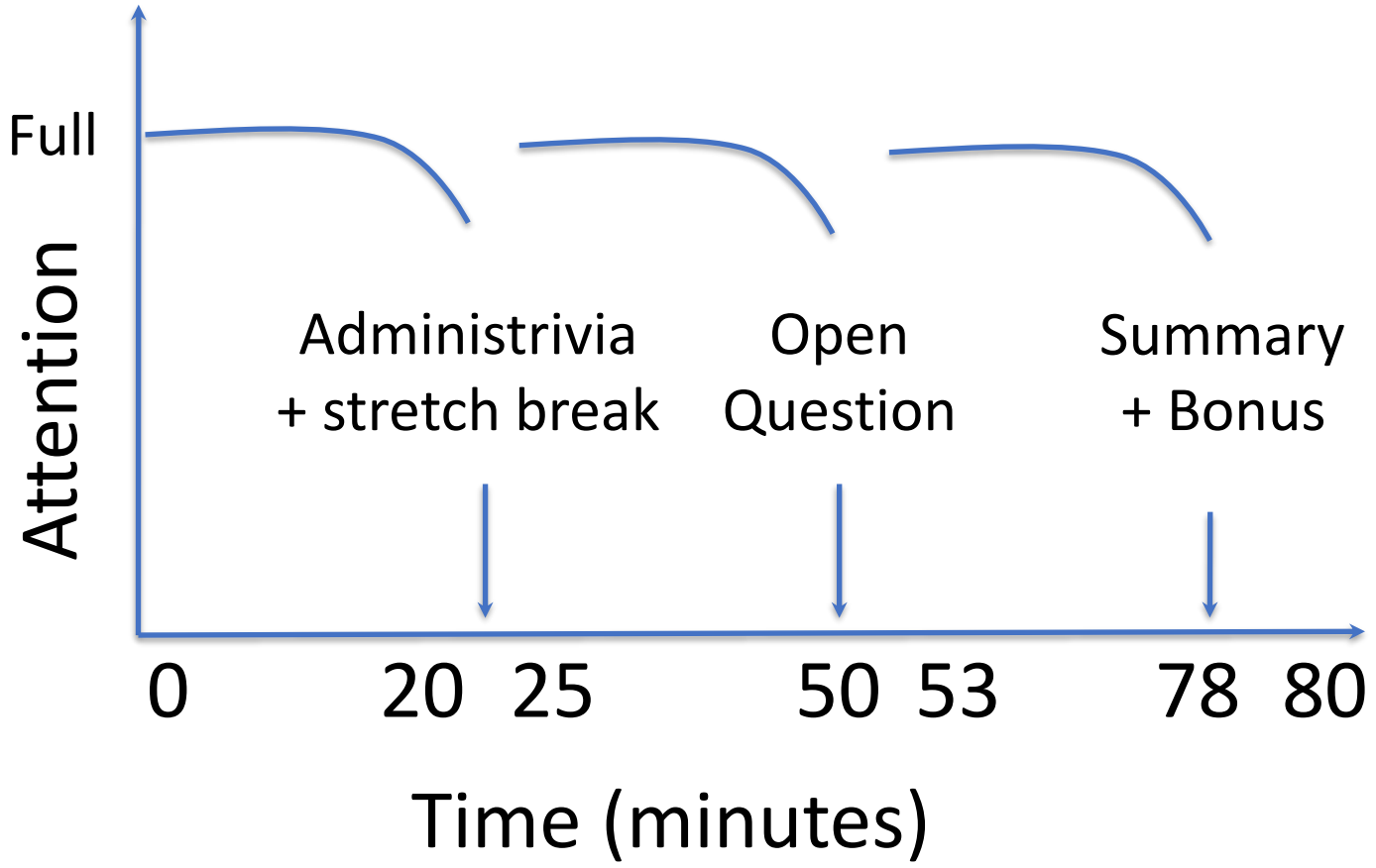
- Challenging limitations
  - Limited memory and processing
  - Energy concerns

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

Architecture of a lecture



Outline

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

Course Time

- Lecture: Tuesdays and Thursdays 3:30-4:50pm, Tech A110
- Lab: Fridays 10:00-11:50am, Tech CG50

- No lab this week!

- Labs start next week Friday and are weekly from there
  - Six labs total
  - When labs run out, I'll use the time for project meetings with groups

Course grade components

- 42% Labs
  - 6 labs at 7% each
  - Guided exploration of course concepts
  - Staff gives checkoffs as you complete parts

- 20% Quizzes
  - Four timed quizzes at 5% each
  - Covers lecture material from last two weeks
  - Probably in-class at the end of class, I'll update you in advance

- 38% Final Project
  - Open-ended group project (will explain in a minute)

Labs

1. MMIO and Interrupts
2. Virtual Timers
3. LED Matrix
4. Breadboarding
5. Audio Input/Output
6. I2C Accelerometer/Magnetometer

- Labs will be partner work
- Due one week from start of lab

Final projects

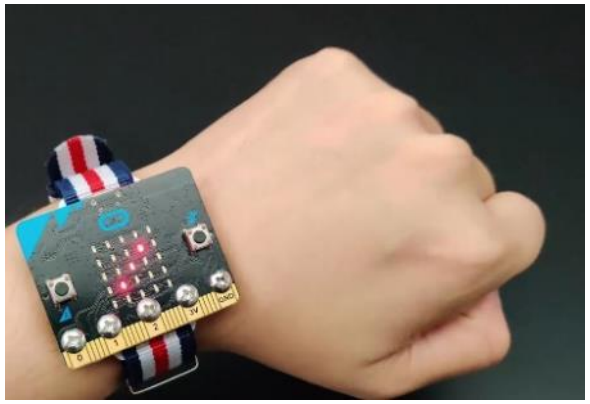
- Opportunity for you to apply your interests to this course
  - In groups of 2-3 students (maybe 4 for a really big idea)

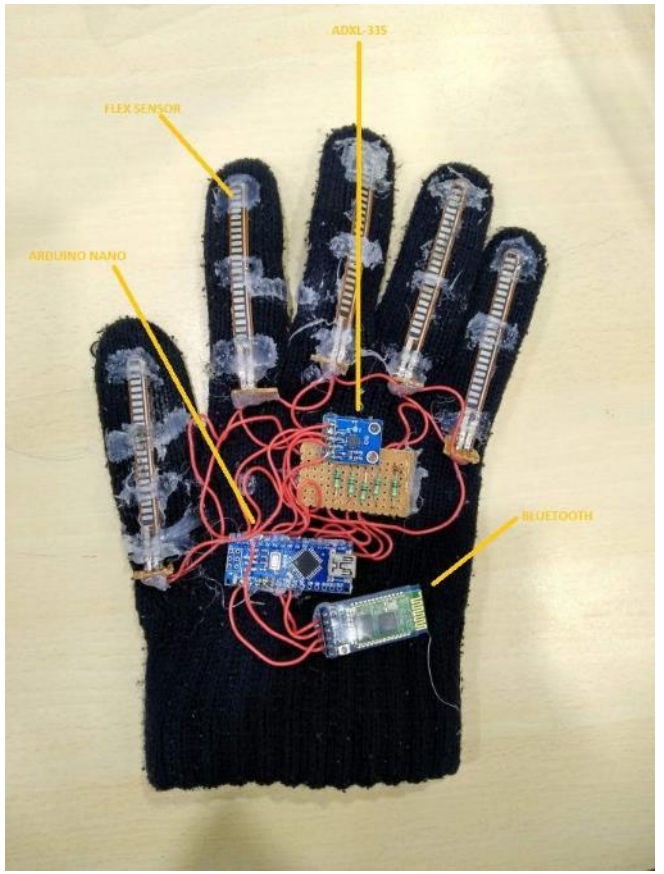
- Demonstrate course knowledge through any application
  - Microbit
  - Various hardware I'll have on hand
  - Small budget for purchasing additional stuff

Project Ideas

- Some ideas to get you thinking
  - Game with interesting control mechanism
  - Smart gloves
  - Smartwatch
  - Simple robotic systems (example: plotter)

- Projects can use
  - Multiple Microbits
  - A personal computer for some amount of coordination
  - Lots of different sensors or actuators
    - Go explore sparkfun.com





Project Logistics

- Week 4: Proposals due
  - I'll get you feedback in a week
- Week 6: Project Design Presentations
  - Short presentations in class about your proposed project and design
  - Chance to give each other useful feedback about how to proceed
- Week 8-9: Labs are done and Fridays are used for update meetings
- Week 11: Live project demos!!
  - Currently scheduled as in-class
  - Might have a public demo session as well

Course Staff and Office Hours

- Two PMs who took the course this past spring
  - Kevin Zhu
  - Will Phillips

  - They will help out during labs and also provide lab office hours

- Office Hours: TBD
  - We'll post a schedule soon
  - Also by request! (especially during projects)

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"

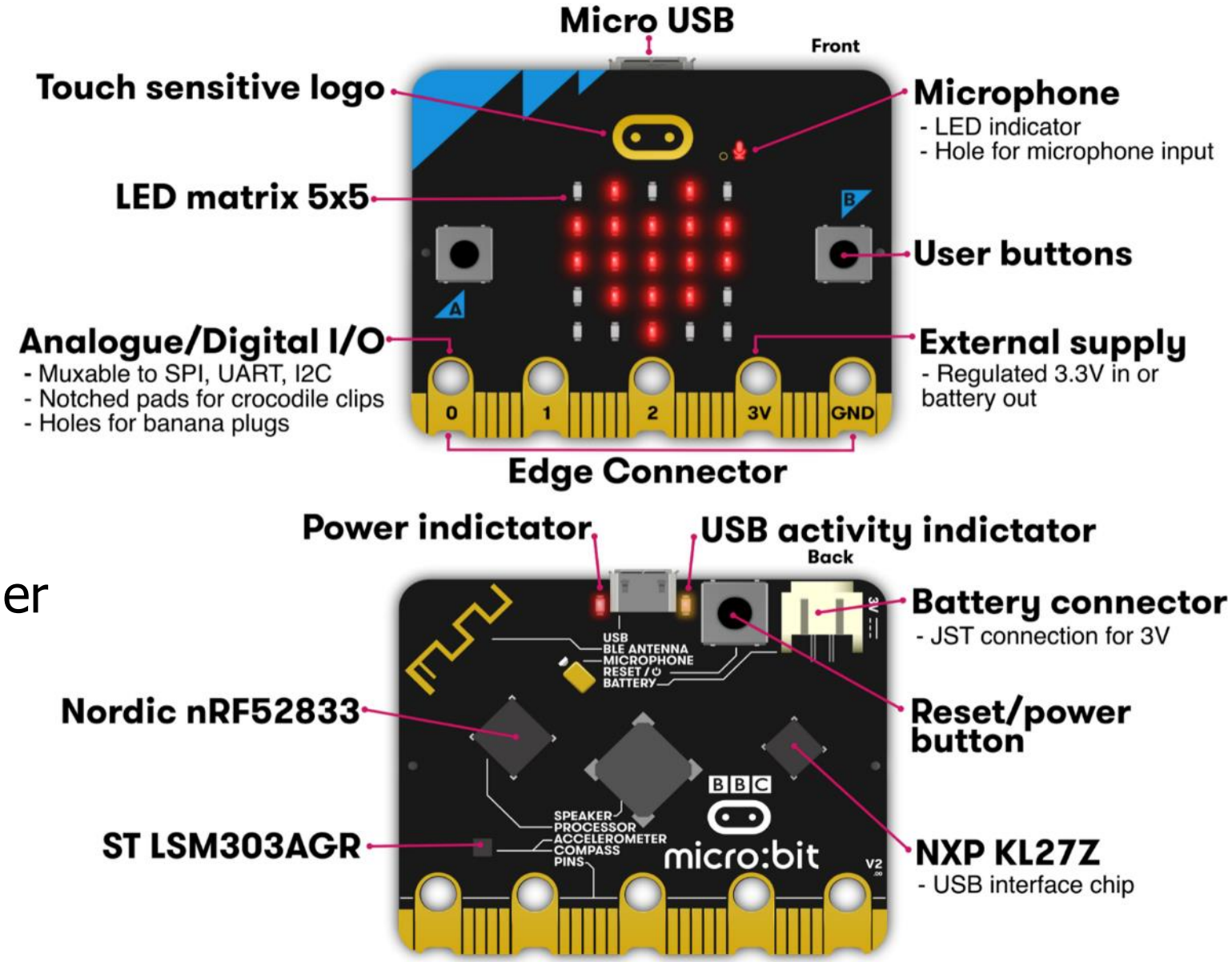
  - You can post directly to "Instructors & TAs" if it is private
    - Use that feature to request office hour appointments if desired

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

Micro:bit v2

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



Working on your own machine

- Getting Started Lab
  - https://docs.google.com/document/d/1dxjVgK7TlFbclEvBSpXaN4LK6Zrd_04iUUbbkXI8Vak/edit?usp=sharing

- Generally, CG50 should have plenty of availability outside of lab sessions
  - Plus staff will hold office hours to provide checkoffs

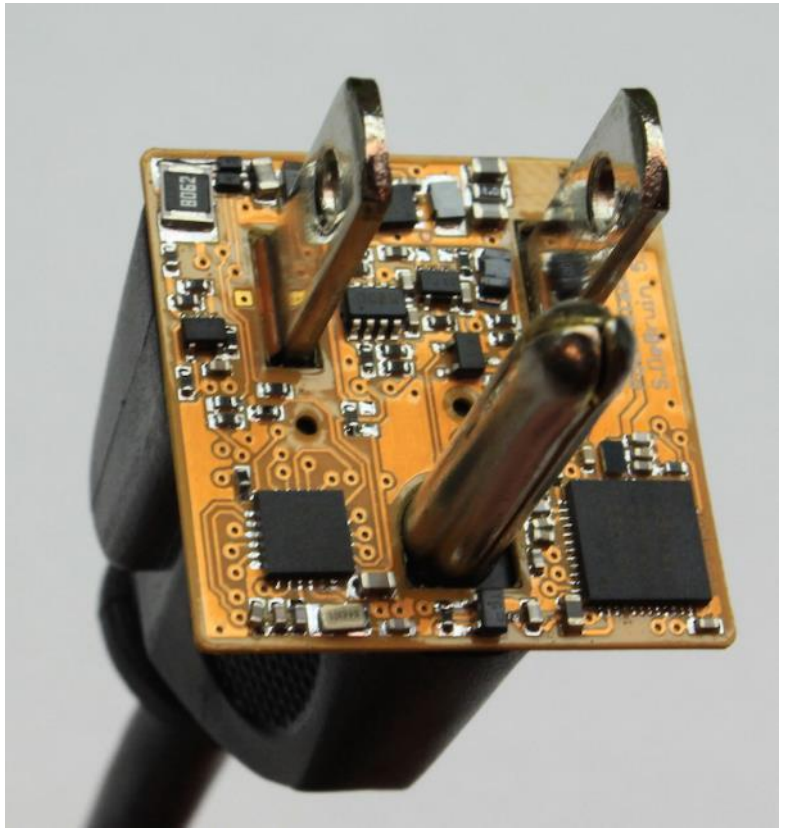
- But it might be nice to have your own personal setup too
  - If you've already got a Linux install, you can just install the programs

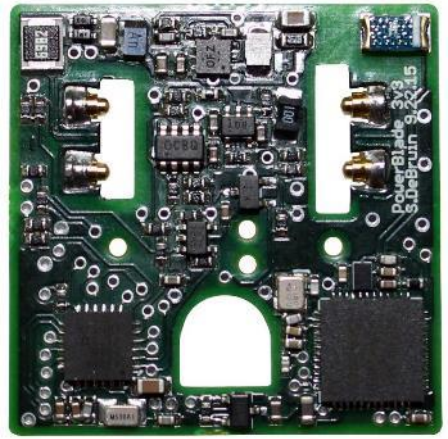
Outline

- Who and Why
- Embedded Systems
- Course Overview

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