Lecture 01
Introduction

CE346 – Microcontroller System Design

Branden Ghena – Fall 2024

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

Northwestern

Welcome to CE346/CS346!

- 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?

Asking questions, four ways

1. You can always ask questions during lecture!
   - I'll let you know if I need to move on for now and answer you after class

2. We'll take breaks during lecture
   - I'll pause after each break to see if any questions came up

3. I will hang out after class for questions
   - Plenty of time to answer everyone

4. You can always ask questions on Piazza too

   The class message board app

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?

- Discuss hardware used for the course and some project ideas.

Outline

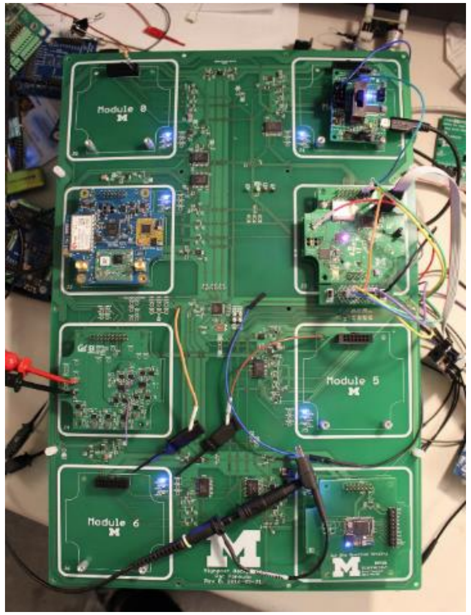
- **Who and Why**
- Embedded Systems
- Course Overview
- Class Hardware
- Project Ideas

Branden Ghena (he/him)

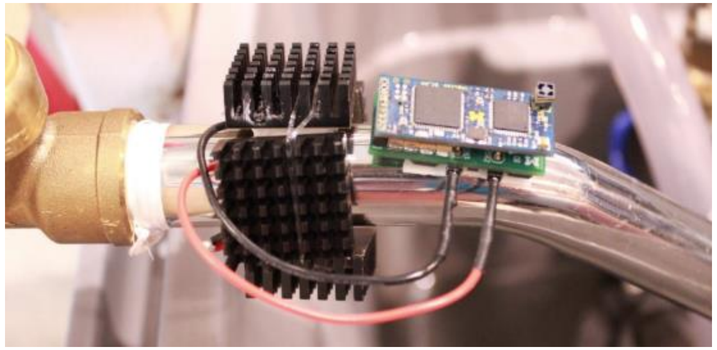
- Assistant Faculty of Instruction
- Education
  - Undergrad: Michigan Tech
  - Master’s: University of Michigan
  - PhD: University of California, Berkeley
- Research
  - Resource-constrained sensing systems
  - Low-energy wireless networks
  - Embedded operating systems
- Teaching
  - Computer Systems
    - CS211: Fundamentals of Programming II
    - CS213: Intro to Computer Systems
    - CS343: Operating Systems
    - CE346: Microcontroller System Design
    - CS397: 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!

- Result: this course!
    - 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
- Class Hardware
- Project Ideas

What is an embedded system?

- A computer built into a device
  such that the *device* is interacted with,
  not the computer
  - E.g., not a desktop or laptop
  - (although many of those deal with overlapping hardware/software issues)

- Includes many domains
  - Robotics
  - Industrial processes
  - Smart home
  - Smart city
  - Wearables and health sensing
  - Internet of Things

Discussion: identify some embedded systems

- What devices that you might not usually consider as computers actually have embedded computers in them?
  - Talk with the others around you
  - Goal: come up some unique ideas

  - We’ll share ideas with the class afterwards

Discussion: what does and does not count

- Share your thoughts
- Is ______ an embedded system? Why or why not?
  - Smart lightbulb
  - Roomba
  - Autonomous vehicle
  - Smartwatch
  - Smartphone
  - Voice-assistant (Alexa)
  - Keyboard

Trend: embedded computers instead of custom hardware

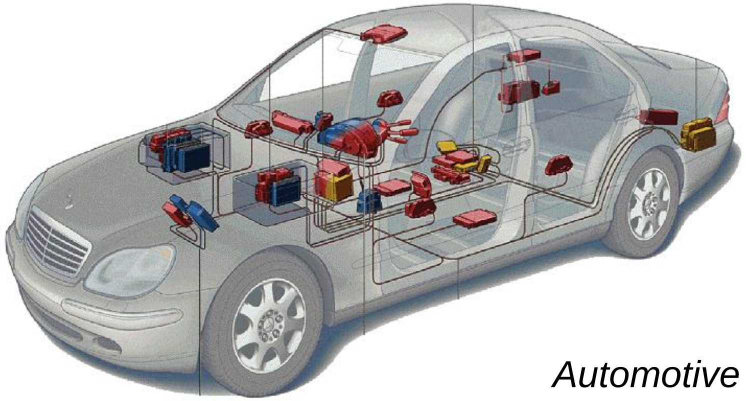
- Some embedded devices could be a state machine in custom hardware instead

- However, computers are increasingly common in those cases
  1. Embedded computers are increasingly cheap
  2. More software developers than hardware developers

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

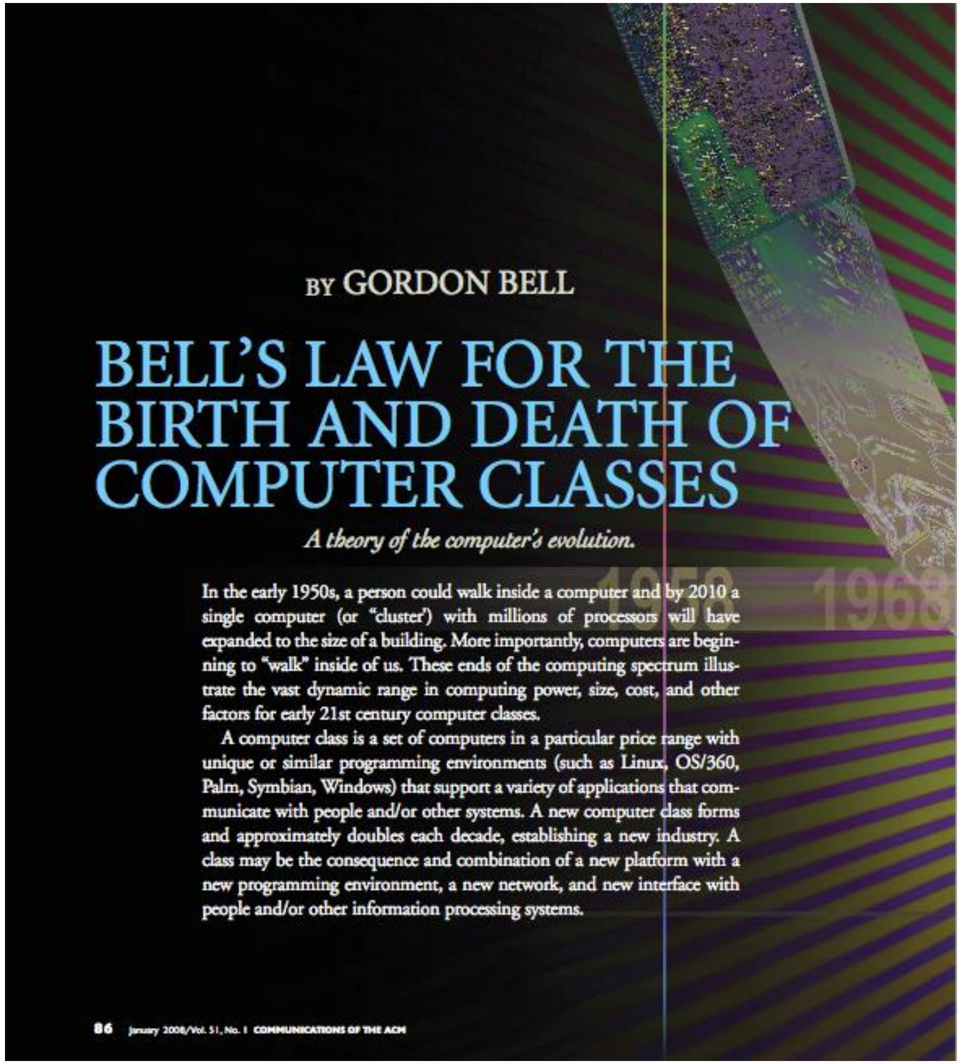


Automotive

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

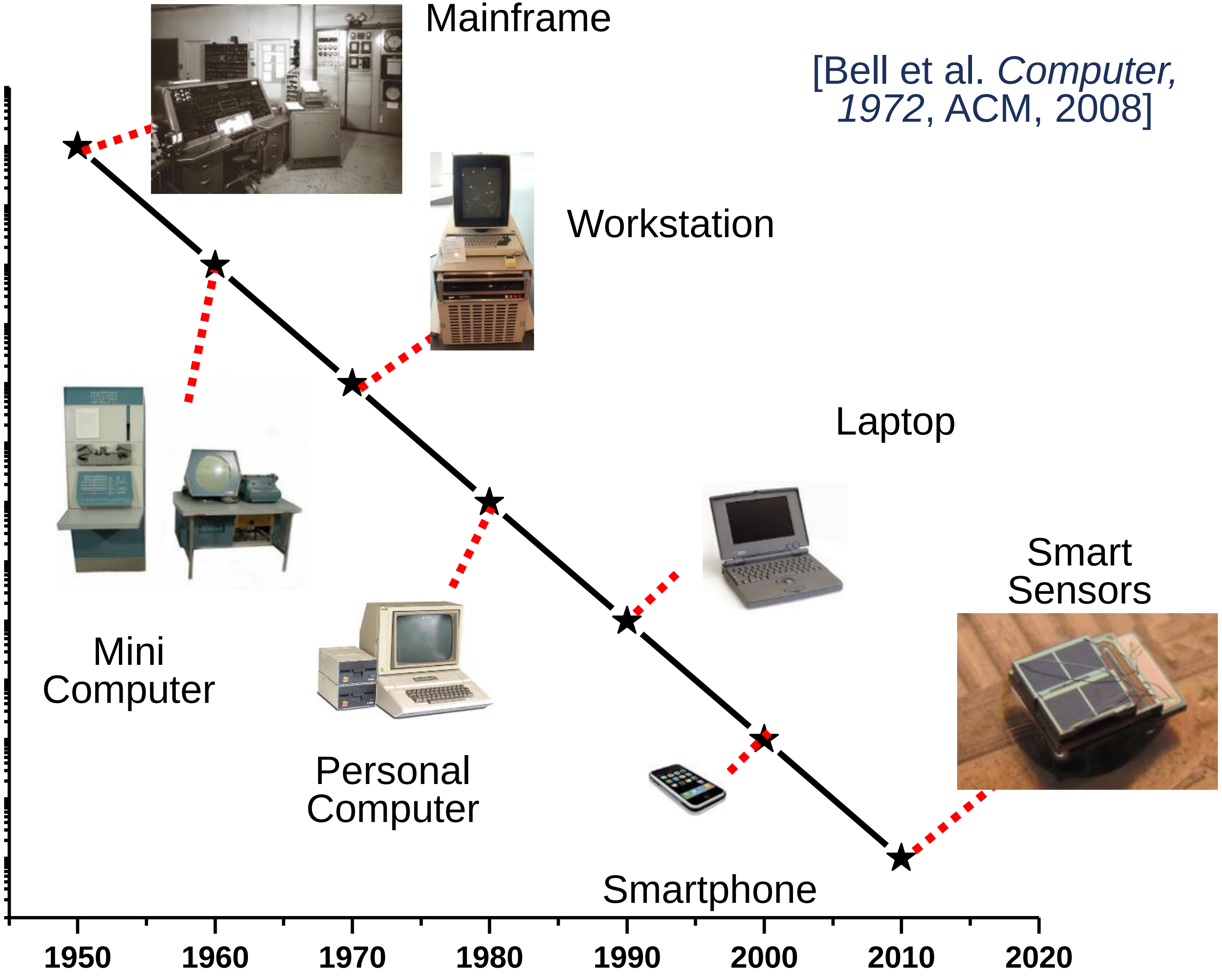
A theory of the computer’s evolution.

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.

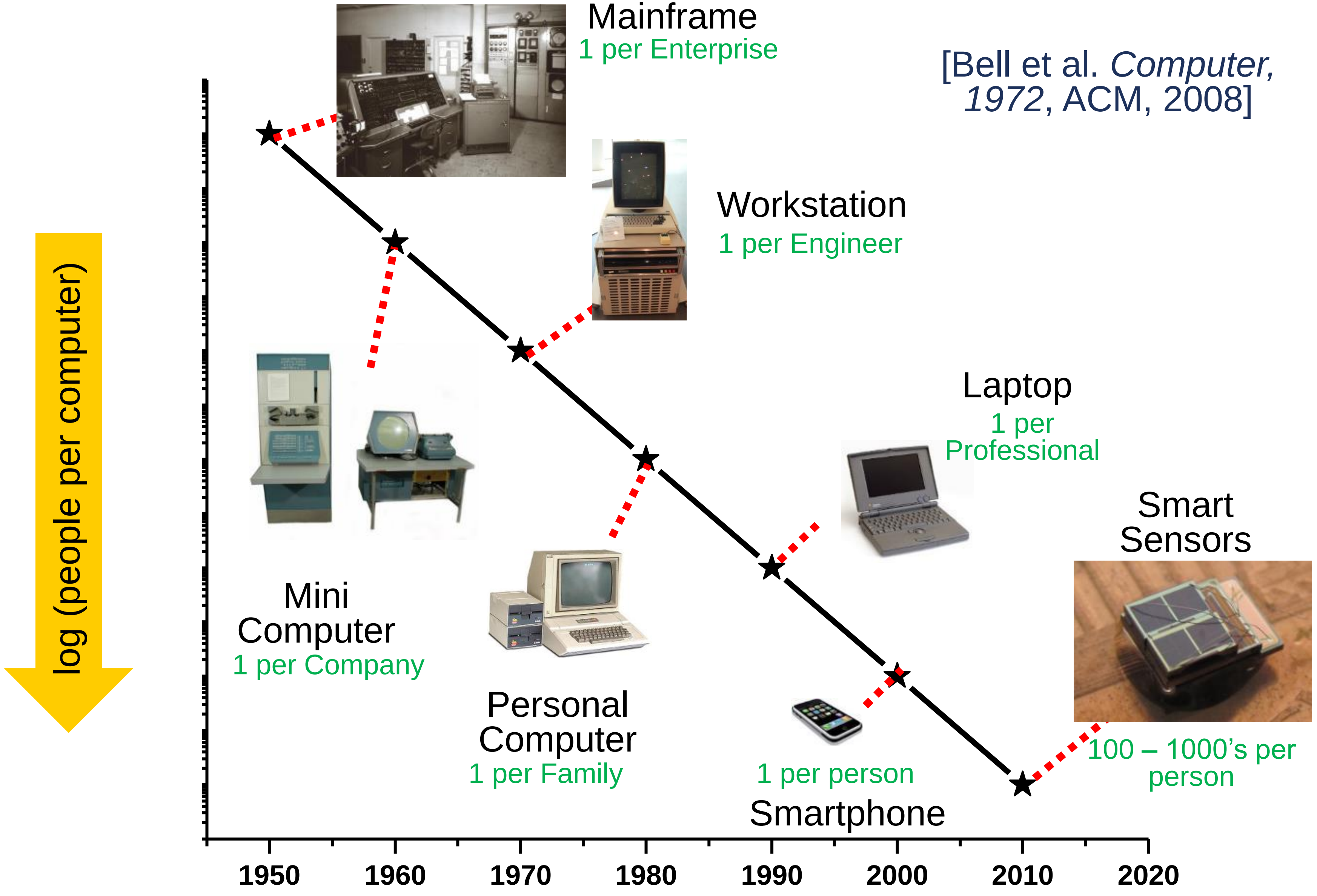
86 January 2008/Vol. 51, No. 1 COMMUNICATIONS OF THE ACM

Classes of computation



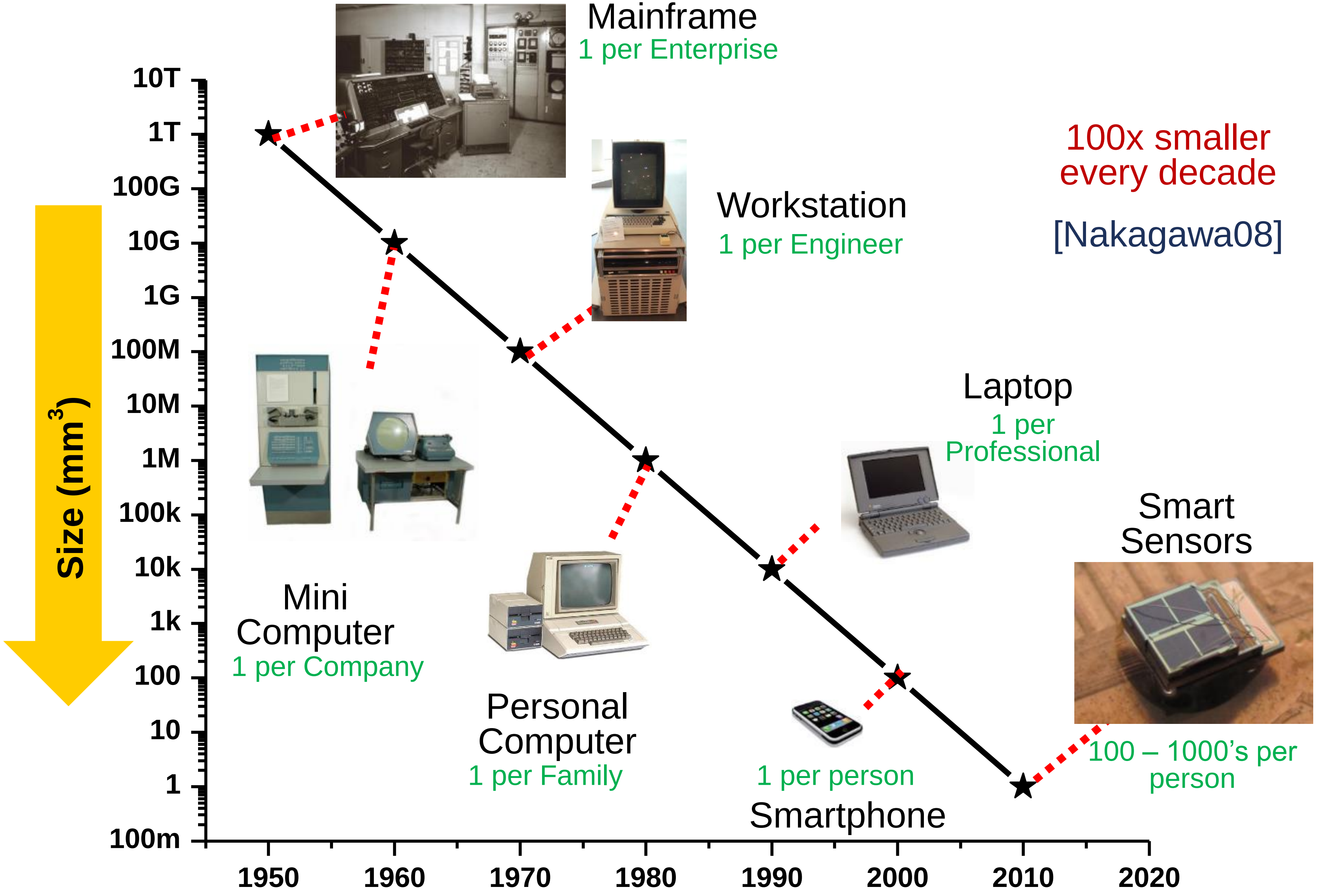
[Bell et al. *Computer, 1972*, ACM, 2008]

Number of computers per person grows over time



[Bell et al. *Computer, 1972*, ACM, 2008]

Computer volume shrinks by 100x every decade



Mainframe
1 per Enterprise

Workstation
1 per Engineer

Mini Computer
1 per Company

Personal Computer
1 per Family

Laptop
1 per Professional

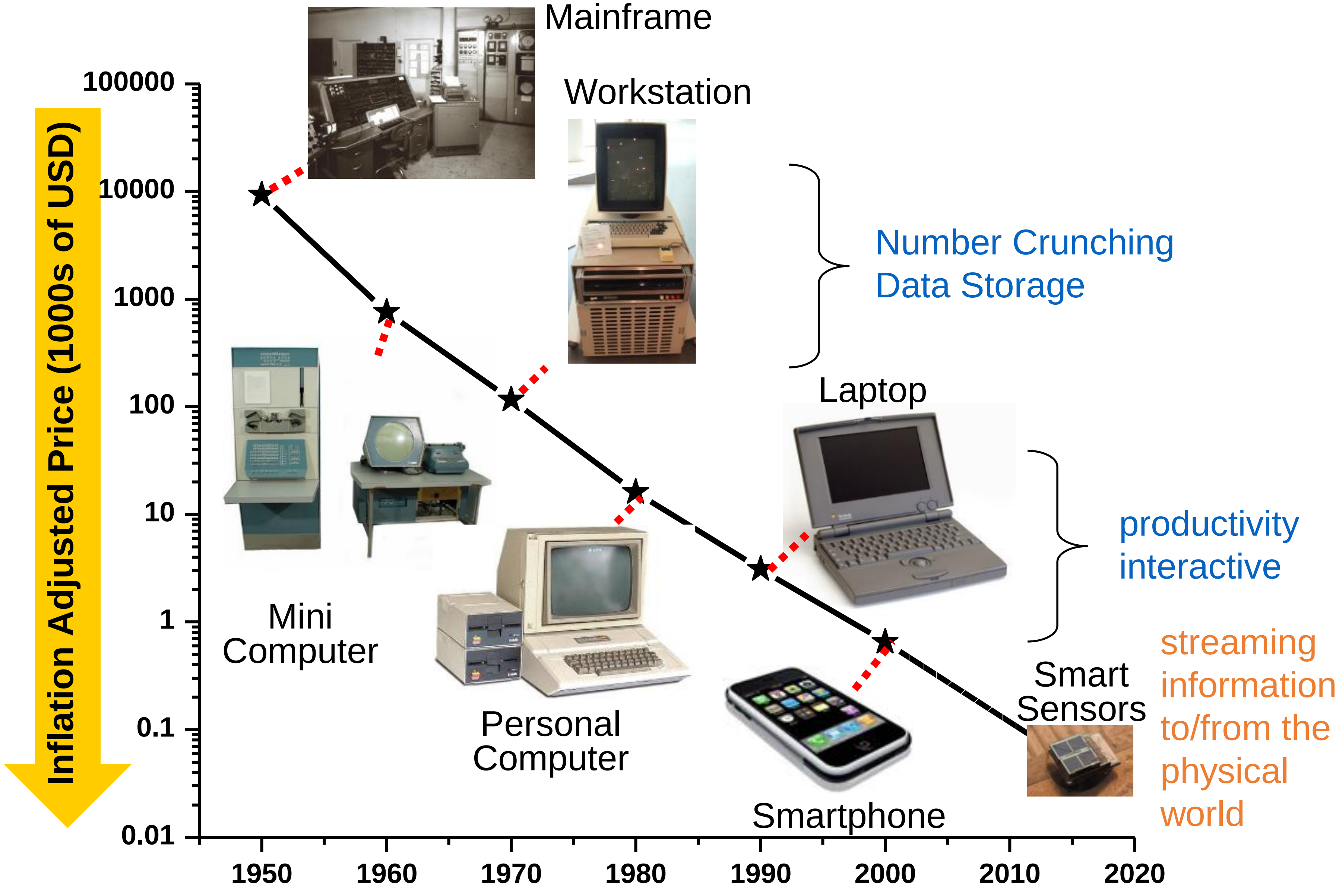
Smartphone
1 per person

Smart Sensors
100 – 1000's per person

100x smaller every decade

[Nakagawa08]

Price falls dramatically, enabling new applications



The Internet of Things (IoT)

REALTOR
REAL ESTATE AND THE INTERNET OF THINGS
HOW WILL SUPER CONNECTIVITY CHANGE REAL ESTATE
challenge
Say Hello to the INTERNET of THINGS
A FULL GUIDE FOR INTERNET OF THINGS! GET ONE FOR A REVIEW
DIGITAL REVIEW
INTERNET OF THINGS
What it means for the future?
DIGITAL LOCKS & SMART BUILDINGS
CAN'T ALL OUR THINGS JUST GET ALONG?
40.02 Percent
Business and marketing
30.3 Health Care
8.3 Retail
7.7 Security
Terror Central / The Antisocial Network / Summer Books
TIME
THE SMARTER HOME
Forbes INSIGHTS
HARNESSING THE INTERNET OF THINGS
HOW TO DERIVE BIG BUSINESS BENEFITS FROM THE CONNECTED WORLD
Volume
INTERNET OF THINGS
With Trust Design #2 and Tracing Concepts
The Price of Ubiquity 18 • Strategic Technology 27 • Hungry for Business 38
SIGNAL
Small Business Leads The Way in Innovation
THE INTERNET of THINGS
Disparate Elements, Unknown Outcomes
Harvard Business Review
NOVEMBER 2014
The HBR List of Best-Performing CEOs
The New Deal on Data
Do Business and Politics Mix?
THE INTERNET OF EVERYTHING
Smart, connected products will transform your business PAGE 64

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

Some example Internet-of-Things devices

Thought experiment: motion-controlled lighting

- What about motion-controlled lights?
  - Sensor detects motion and turns on the light
  - Each light is individually controlled by its sensor only

- Would that still count as IoT?

- Could that still be an embedded system?

Thought experiment: motion-controlled lighting

- What about motion-controlled lights?
  - Sensor detects motion and turns on the light
  - Each light is individually controlled by its sensor only

- Would that still count as IoT?
  - **No**
  - **Missing communication aspect for sure, probably computation**

- Could that still be an embedded system?
  - **Yes? Sensing + actuation packaged as a device**
  - **Again, computation *could* be lacking**

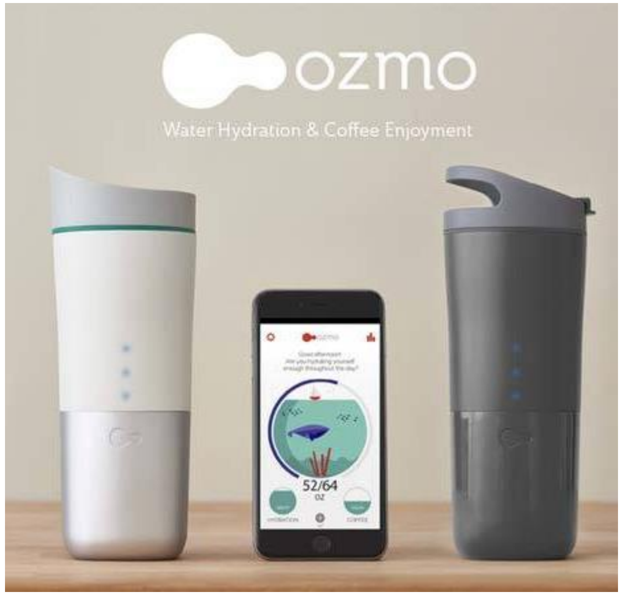
Thought experiment: high-capability computing

- 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?

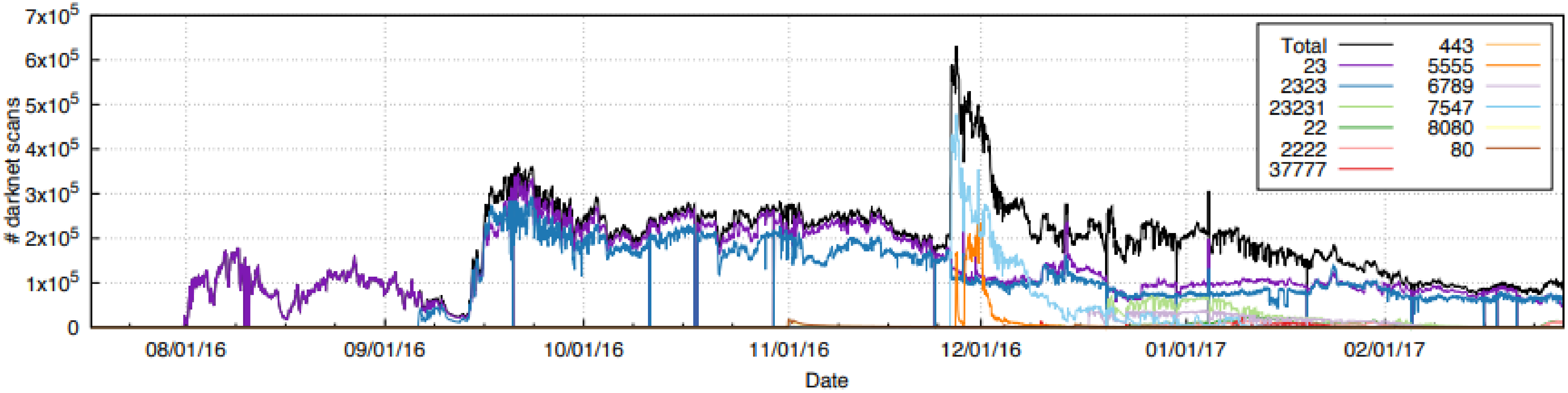
Thought experiment: high-capability computing

- 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**

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

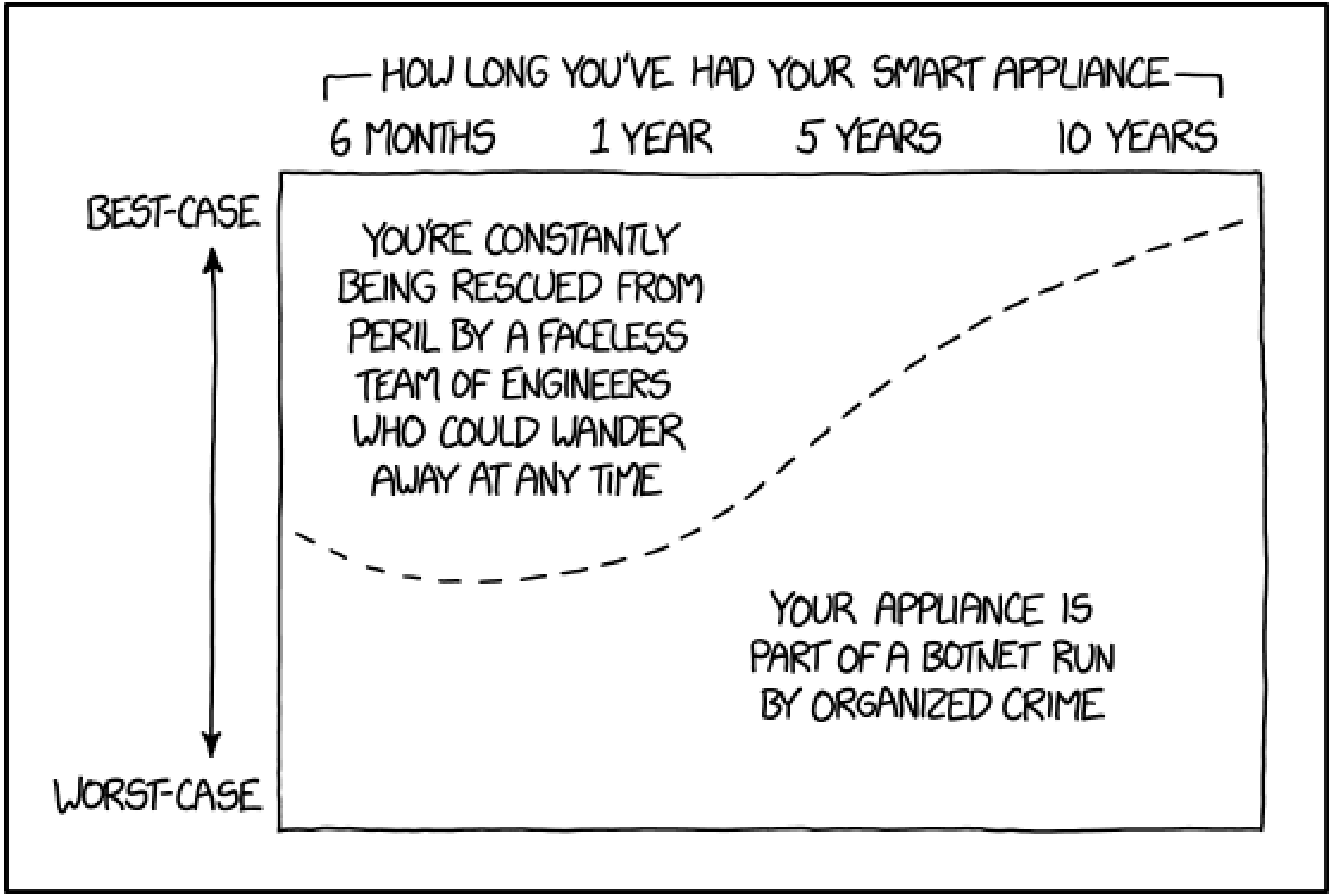
What makes resource-constrained embedded systems frustrating?

- Challenging limitations
  - Limited memory and processing
  - Energy concerns

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

  - Example: my first grad project eye-tracking glasses
    - Camera -> ADC -> FPGA -> Linux driver -> Linux app -> Network -> Visualizer app

Break + xkcd


HOW LONG YOU'VE HAD YOUR SMART APPLIANCE
6 MONTHS
1 YEAR
5 YEARS
10 YEARS
BEST-CASE
WORST-CASE
YOU'RE CONSTANTLY BEING RESCUED FROM PERIL BY A FACELESS TEAM OF ENGINEERS WHO COULD WANDER AWAY AT ANY TIME
YOUR APPLIANCE IS PART OF A BOTNET RUN BY ORGANIZED CRIME

https://xkcd.com/1966/

Outline

- Who and Why
- Embedded Systems
- **Course Overview**
- Class Hardware
- Project Ideas

Course Staff and Office Hours

- Four PMs who previously took the class
    - Alex Huggins
    - Isa Gonzalez
    - Joseph Grantham
    - Lucas Takayasu

    - 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 details – how to learn stuff

- Lecture: Tuesdays and Thursdays 3:30-4:50pm
    - 2122 Sheridan, Room 250

- Provides background on everything we’ll be doing in labs
    - Lectures are automatically recorded so you can review them

- No textbook for this class
    - Nobody seems to write a good one
    - The datasheet for our microcontroller (nRF52833) will be important though!

Asking Questions

- Class and office hours are always an option!
  - We can do extra questions right after class too

- Piazza: (similar to Campuswire)
  - Post questions
  - Answer each other’s questions
  - Find posts from the course staff
  - Post private info just to course staff

- Post on Piazza – do NOT send me emails
  - Messages are kept in one place and stay “unanswered”

  - You can post directly to “Instructors” if it is private
    - Use that feature to request office hour appointments if desired
    - Or to tell me that you’re sick and can’t attend lab

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)

Class lab sessions

- Lab: Fridays 1:00-2:50pm OR 3:00-4:50pm, Frances Searle 3220
  - Mandatory attendance for these
  - Let me know ASAP if you’re sick and will miss
- Labs start next week Friday and are weekly from there
  - No lab this week!
  - Six labs total
  - When labs run out, I’ll use the time for project meetings with groups
- Warning: labs won’t usually be finished during the lab sessions
  - You’ll have to work on them on your own time too
  - We’ll have office hours for checkoffs

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
    - You choose, but different partner each week
    - MUST work with a partner

- Due one week from start of lab
    - Complete **Checkoffs** plus some **Post-lab Questions** due online

Quizzes

- In-class, on-paper, closed notes quizzes
  - Usually about 15 minutes and held at the end of lecture

- Cover the last two weeks worth of material
  - So make sure you’re up-to-date on what we’re talking about

- First quiz is Tuesday, October 8th (third week of classes)

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 (99% required)
  - Various hardware I'll have on hand
  - Small budget for purchasing additional stuff (up to $40 per person in team)

Project Logistics

- Week 4: Proposals due
  - I’ll get you feedback in about 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 9-11: Labs are done and Fridays are used for update meetings

- Exam Week: Live project demos!!
  - Public demo session
  - Tuesday of exam week (usually half of the class at a time for a 2-3 hour block)

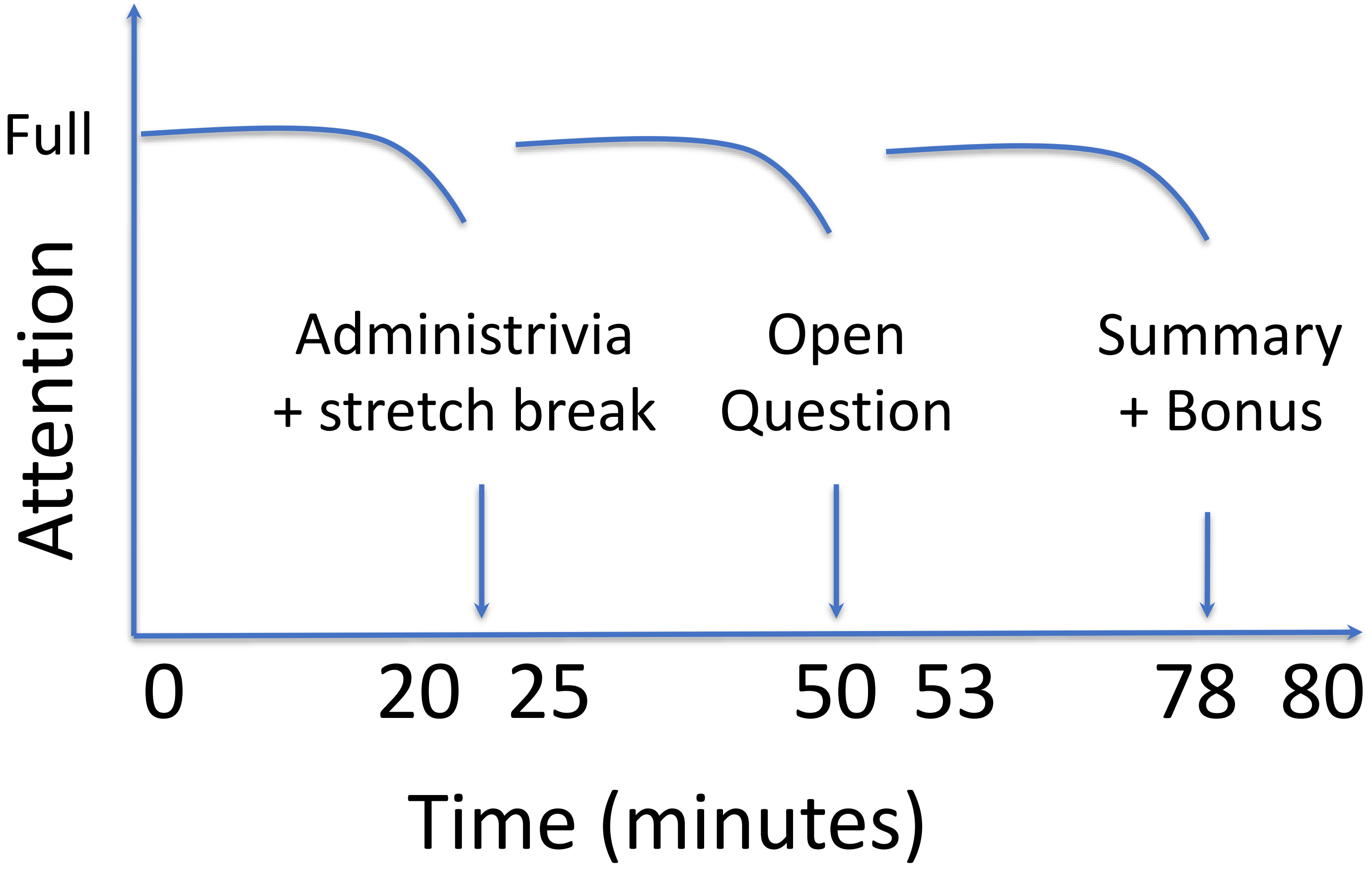
Flexibility

- 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

Architecture of a lecture



Outline

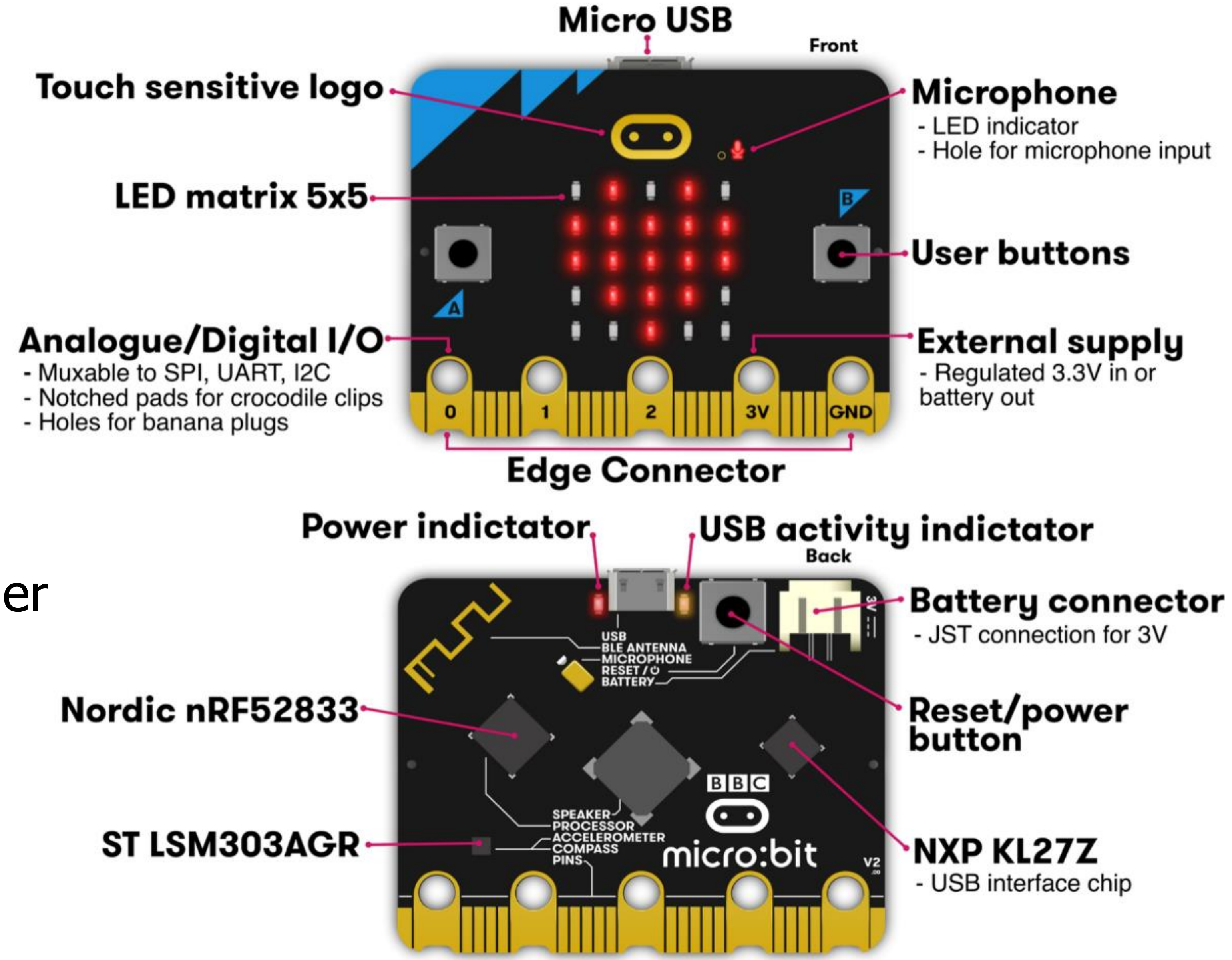
- Who and Why
- Embedded Systems
- Course Overview
- **Class Hardware**
- Project Ideas

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



Getting your own Microbit

- You do NOT need to buy your own Microbit
    - I have enough for everyone in the class to borrow one for the quarter

- If you want your own though, they're pretty cheap:
    - $17.95 Adafruit: https://www.adafruit.com/product/4781
    - $16.50 Sparkfun: https://www.sparkfun.com/products/17287
    - $20.99 Amazon: https://www.amazon.com/Seeed-Studio-BBC-Micro-Accelerometer/dp/B0BDFD1ZM1

Labs will use your own laptops

- All labs will use your own computers

- In the past we used the computers in CG50
  - About one computer would crash per lab session
  - Super crammed in there with no elbow room or walking room
  - And you had to physically go there to work on labs

- Setup for your own computers won’t be that hard
  - Native MacOS or Linux works great
  - For Windows, VMWare Workstation + Ubuntu is pretty easy, but requires ~20 GB

- Concern of mine: equal access to labs
  - If you don’t have a laptop or don’t think it’ll work, let me know!

Poll of the room

- MacOS users
- Linux users
- Windows users
  - WSL users?
- Other?

Reminder: No lab this Friday

- Labs are Fridays

- This Friday there are no lab sessions
  - You’ll be installing things on your machine to set it up for class

  - Try it on your own and see how it goes
    Even if you’ve never done it before, it’s not too difficult

  - We’ll have office hours next week in case you run into issues that you’re not sure how to resolve

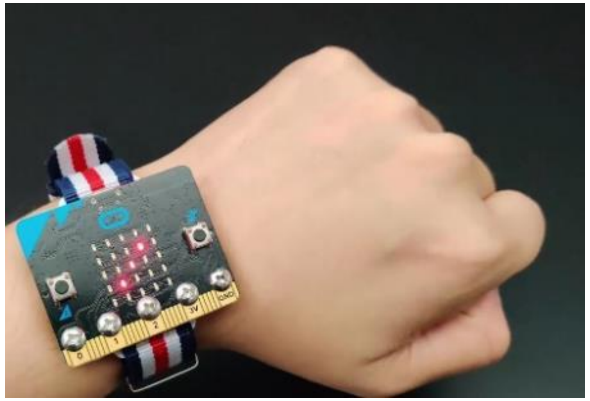
Outline

- Who and Why
- Embedded Systems
- Course Overview
- Class Hardware
- **Project Ideas**

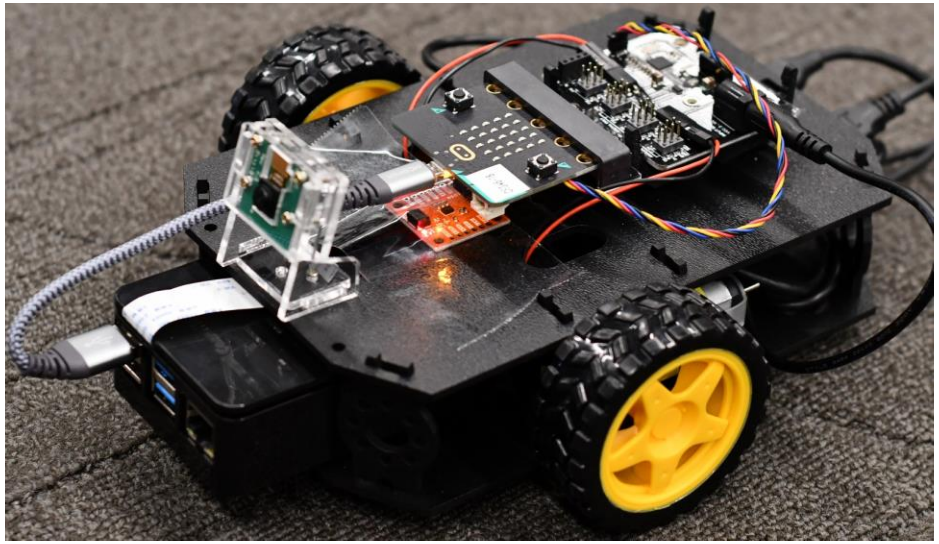
Project Ideas

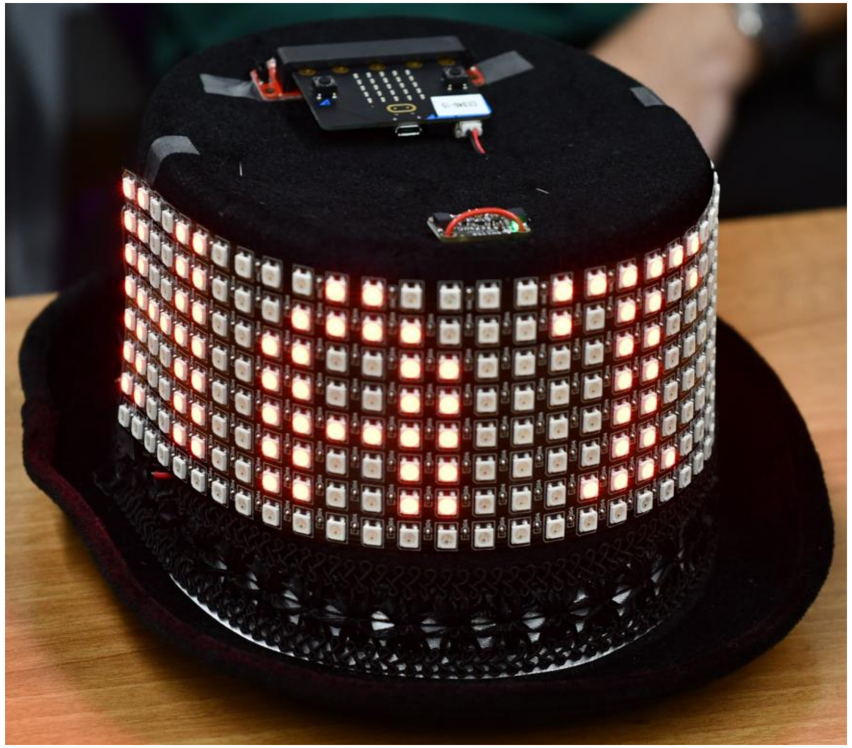
- Some ideas to get you thinking
  - Game with interesting control mechanism
  - Smart gloves
  - Smartwatch
  - Simple robotic systems

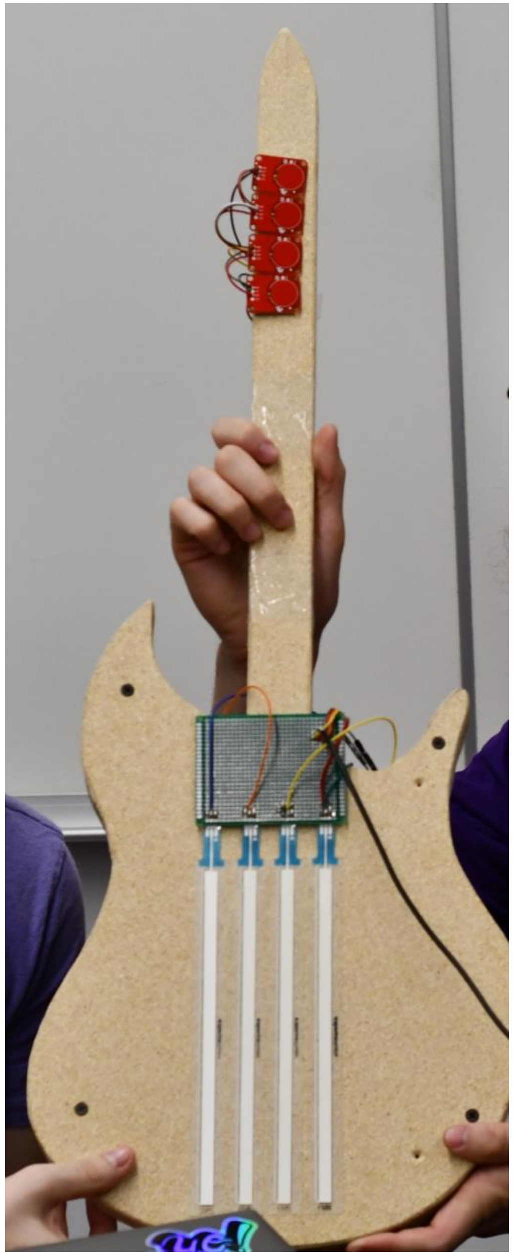
- Projects can use
  - Multiple Microbits
  - A personal computer for some amount of coordination
  - Lots of different sensors or actuators
    - Go explore sparkfun.com

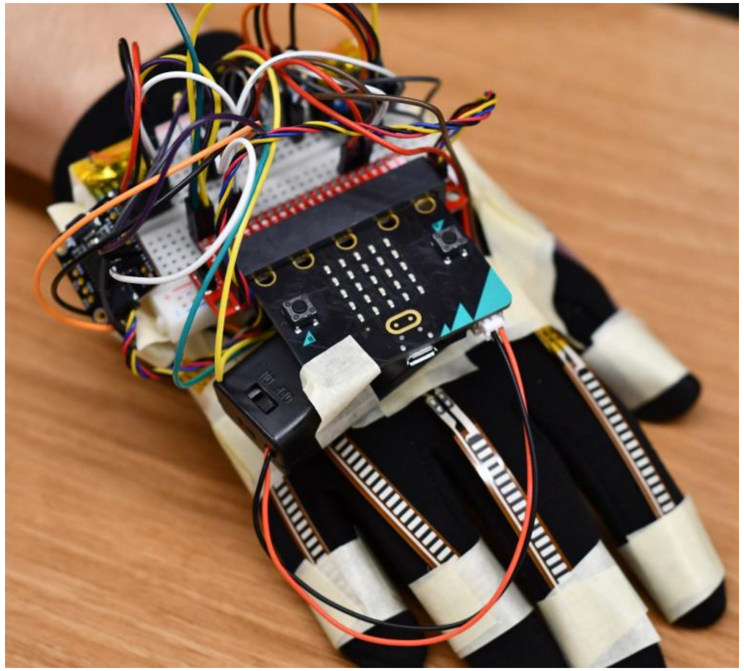


Some awesome Fall 2021 projects

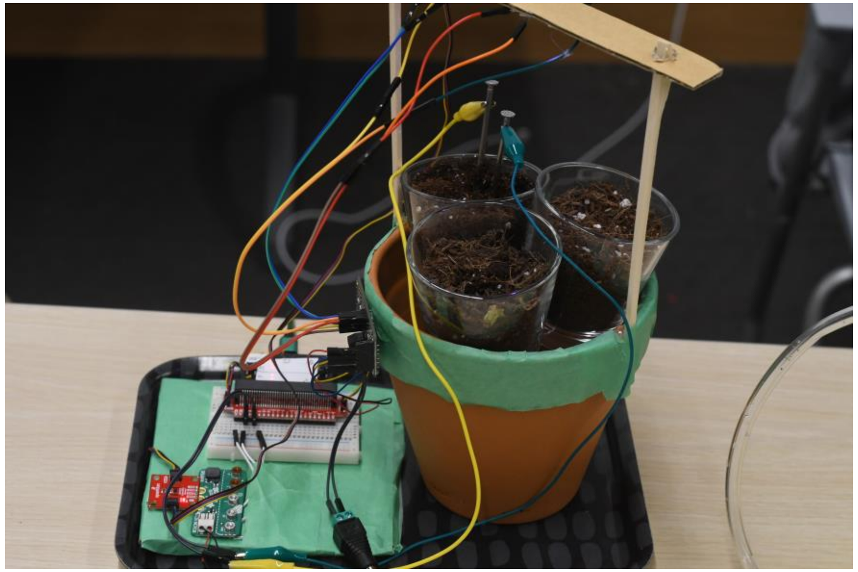


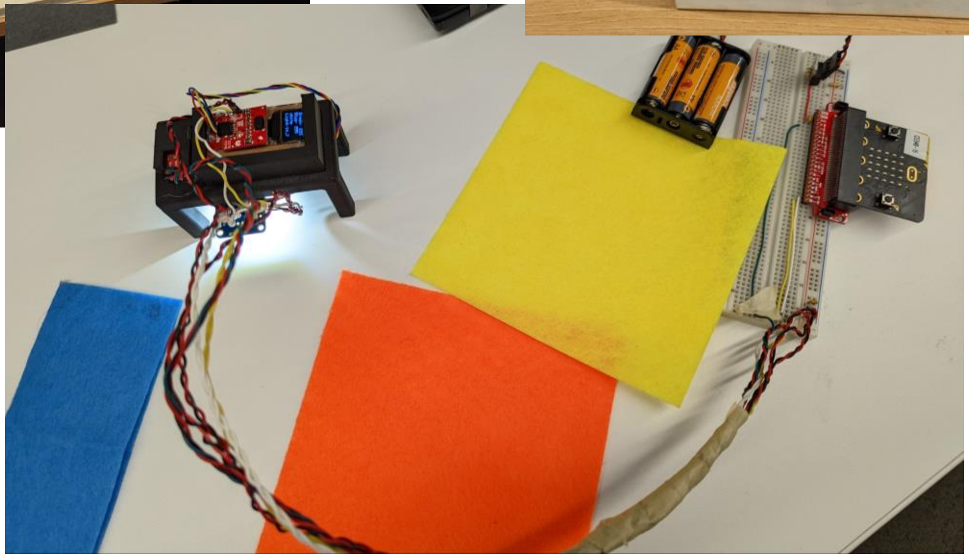




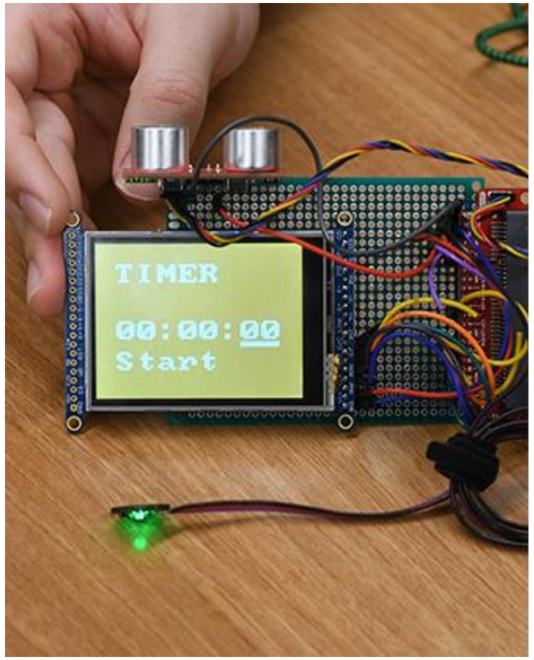


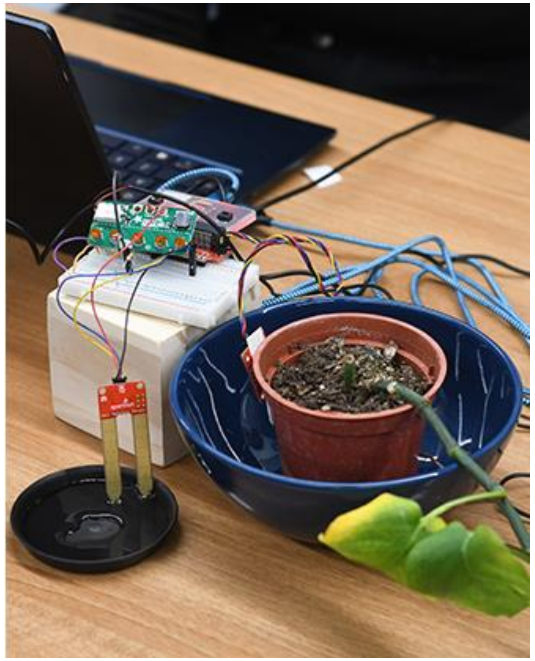
Some awesome Fall 2022 projects

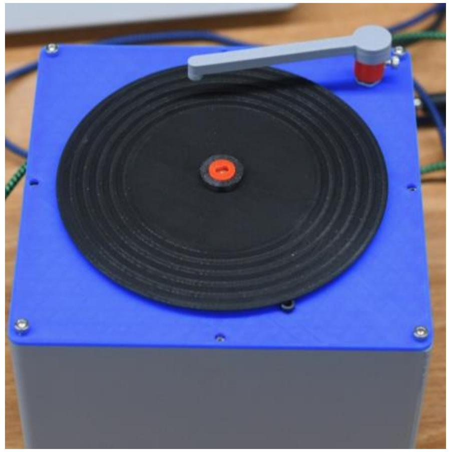




Some awesome Fall 2023 projects





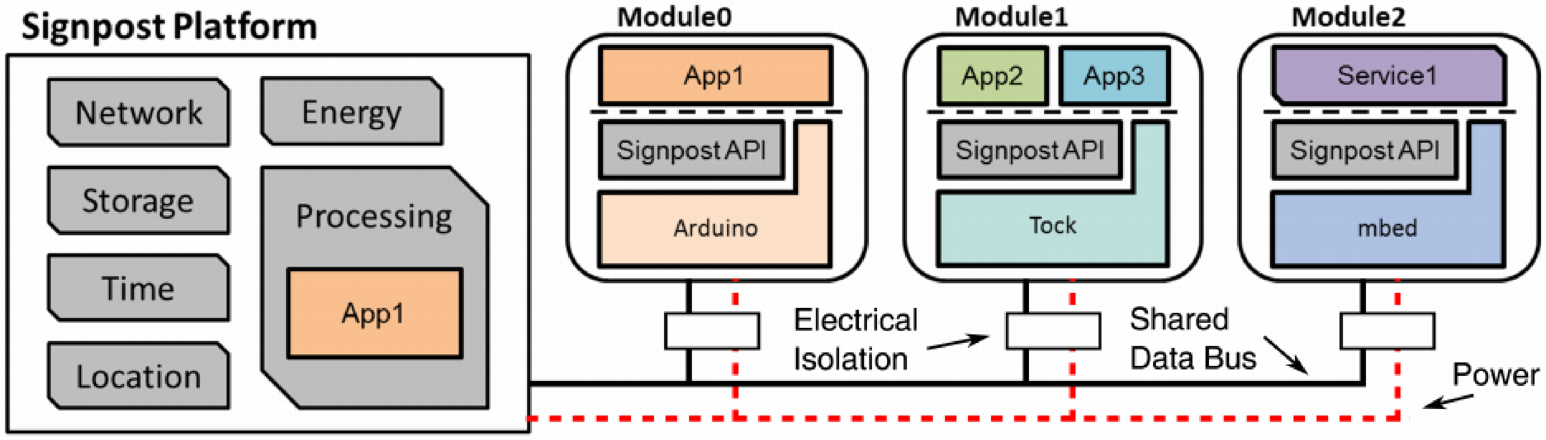


Outline

- Who and Why
- Embedded Systems
- Course Overview
- Class Hardware
- Project Ideas

Signpost – City-Scale Sensing

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



Signpost Platform

Network

Energy

Storage

Processing

App1

Time

Location

Module0

App1

Signpost API

Arduino

Module1

App2

App3

Signpost API

Tock

Module2

Service1

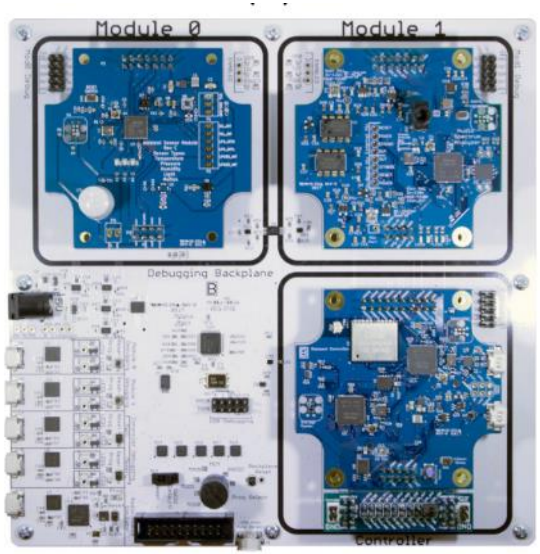
Signpost API

mbed

Electrical Isolation

Shared Data Bus

Power



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