

Lecture 01

Introduction

CE346 – Microcontroller System Design

Branden Ghena – Fall 2025

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

Welcome to CE346/CS346!

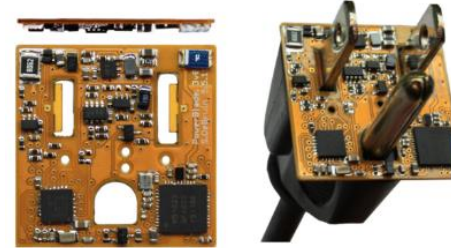
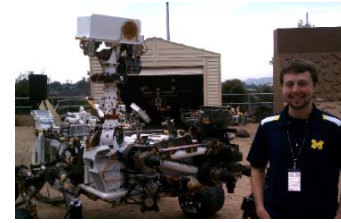
- Focus on hardware/software systems and their design
 - Hardware/Software co-design
 - How do you write software that interacts with hardware?
 - How do you choose hardware to support software needs?
 - Sensors and Sensing
 - What can sensors do and how do they work?
 - How do you write applications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

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: Tock
- Teaching
 - Computer Systems
 - CS211: Fundamentals of Programming II
 - CS213: Intro to Computer Systems
 - CS343: Operating Systems
 - CE346: Microcontroller System Design
 - CS433: Wireless Protocols for the IoT



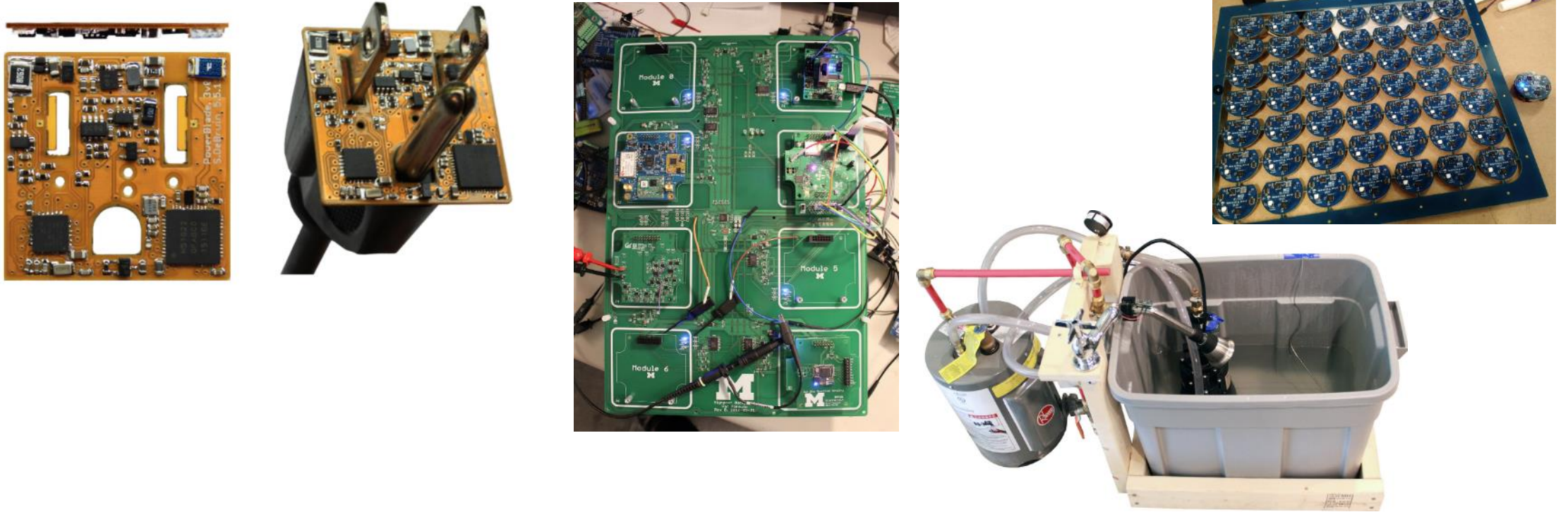
Things I love



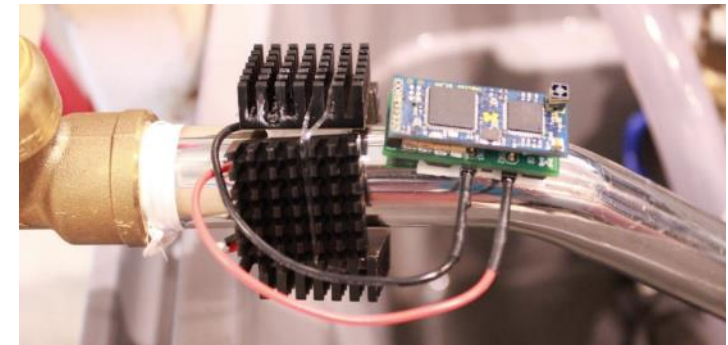
My summer reading list:

- Some Desperate Glory
- Stone and Sky
- Royal Gambit
- A Psalm for the Wild-Built
- The Tainted Cup

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

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.

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

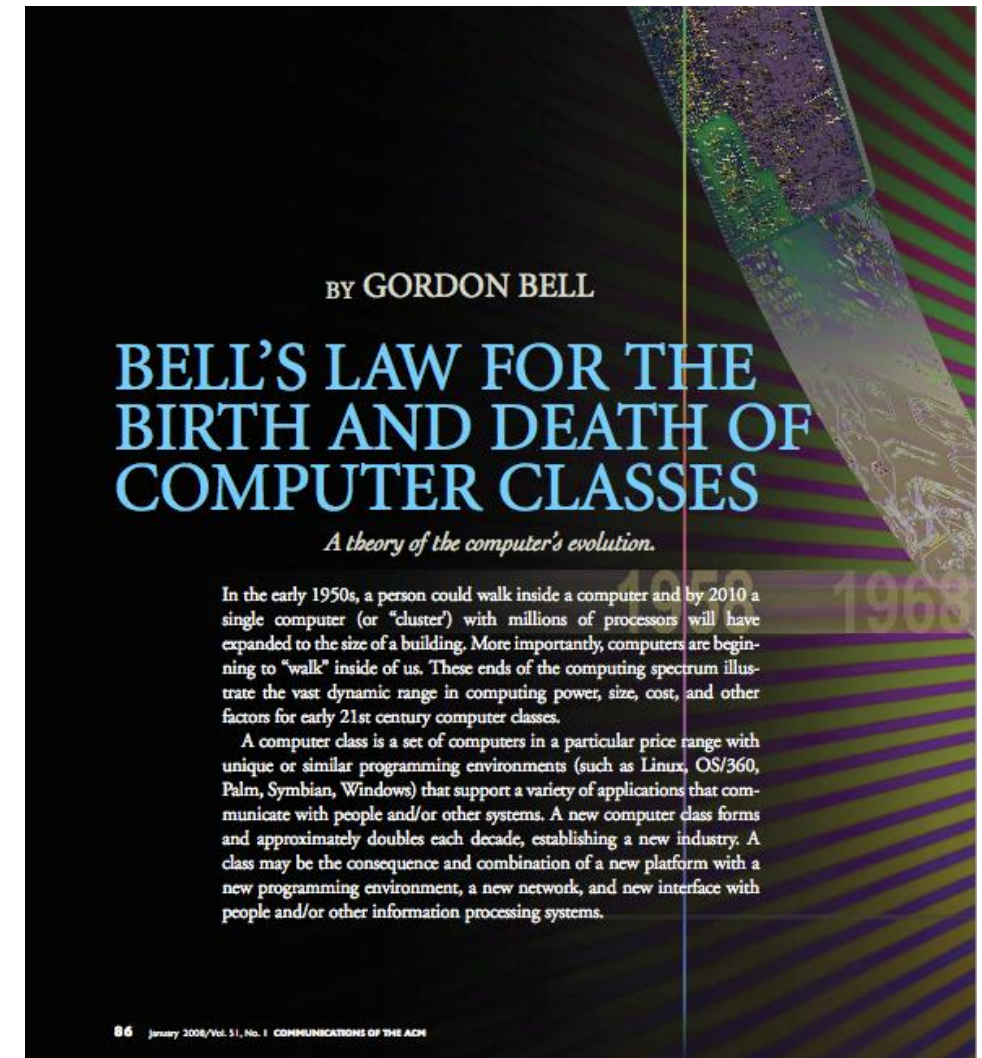
Outline

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

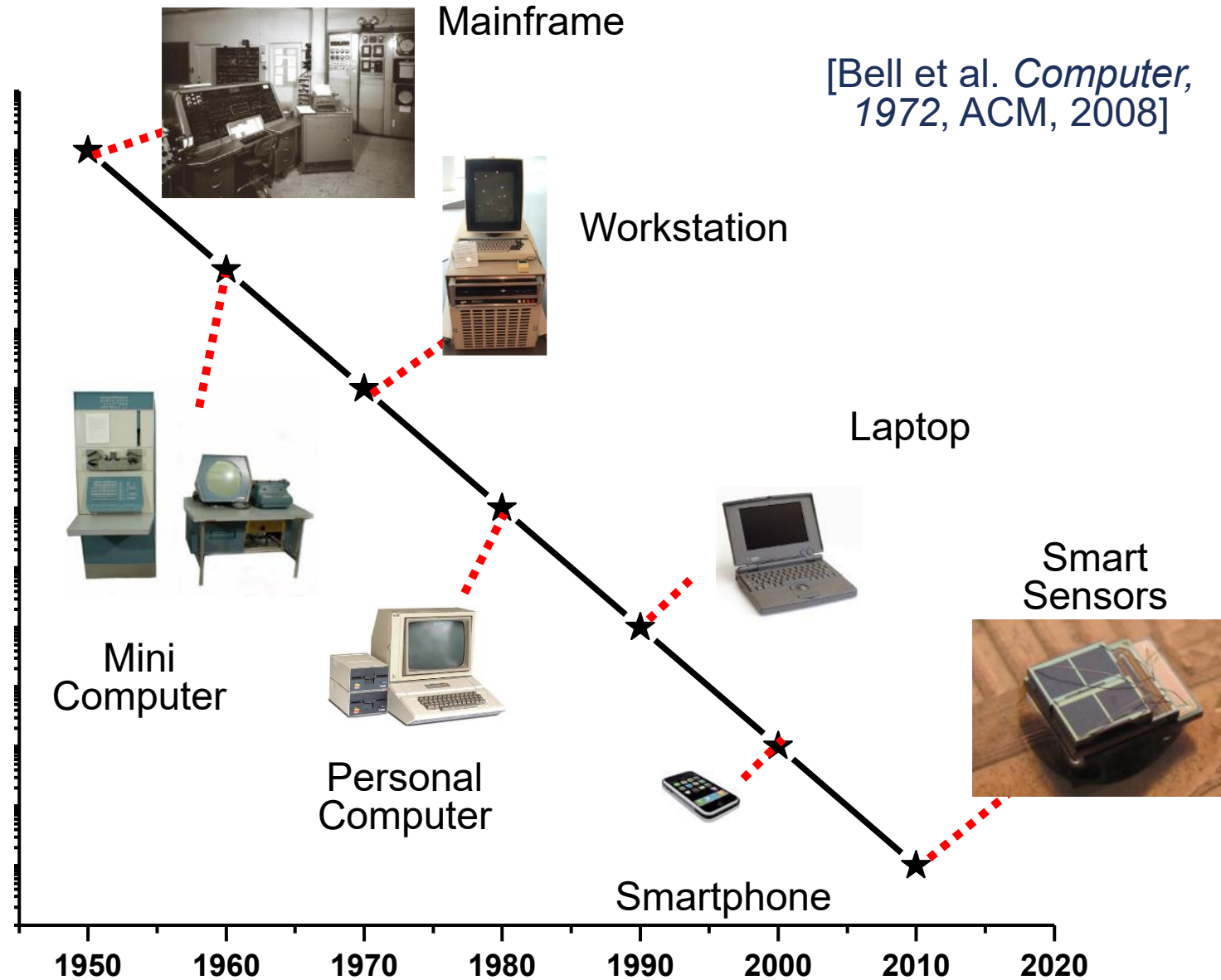
Bell's Law: A new computer class every decade

"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."

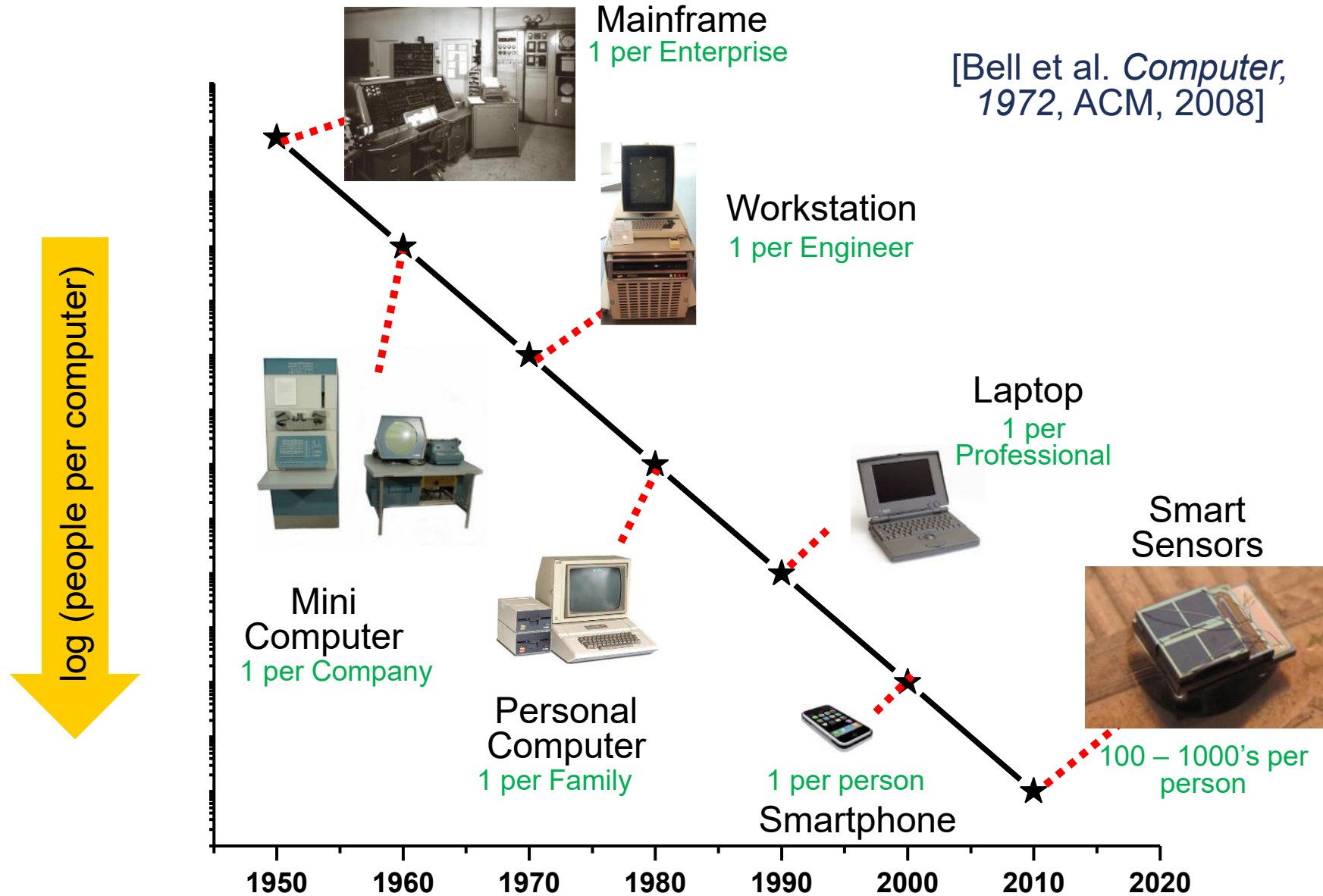
- Gordon Bell [1972,2008]



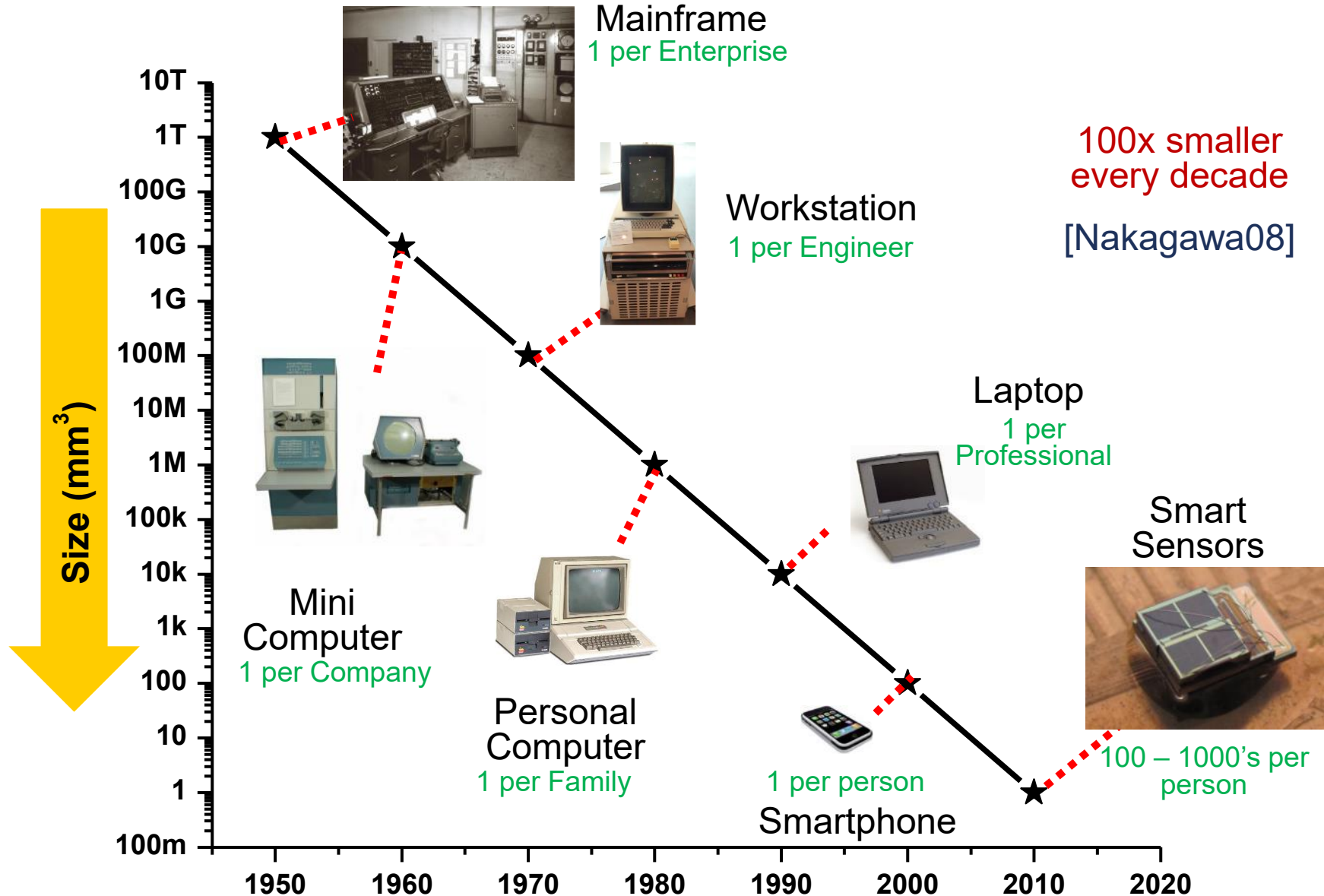
Classes of computation



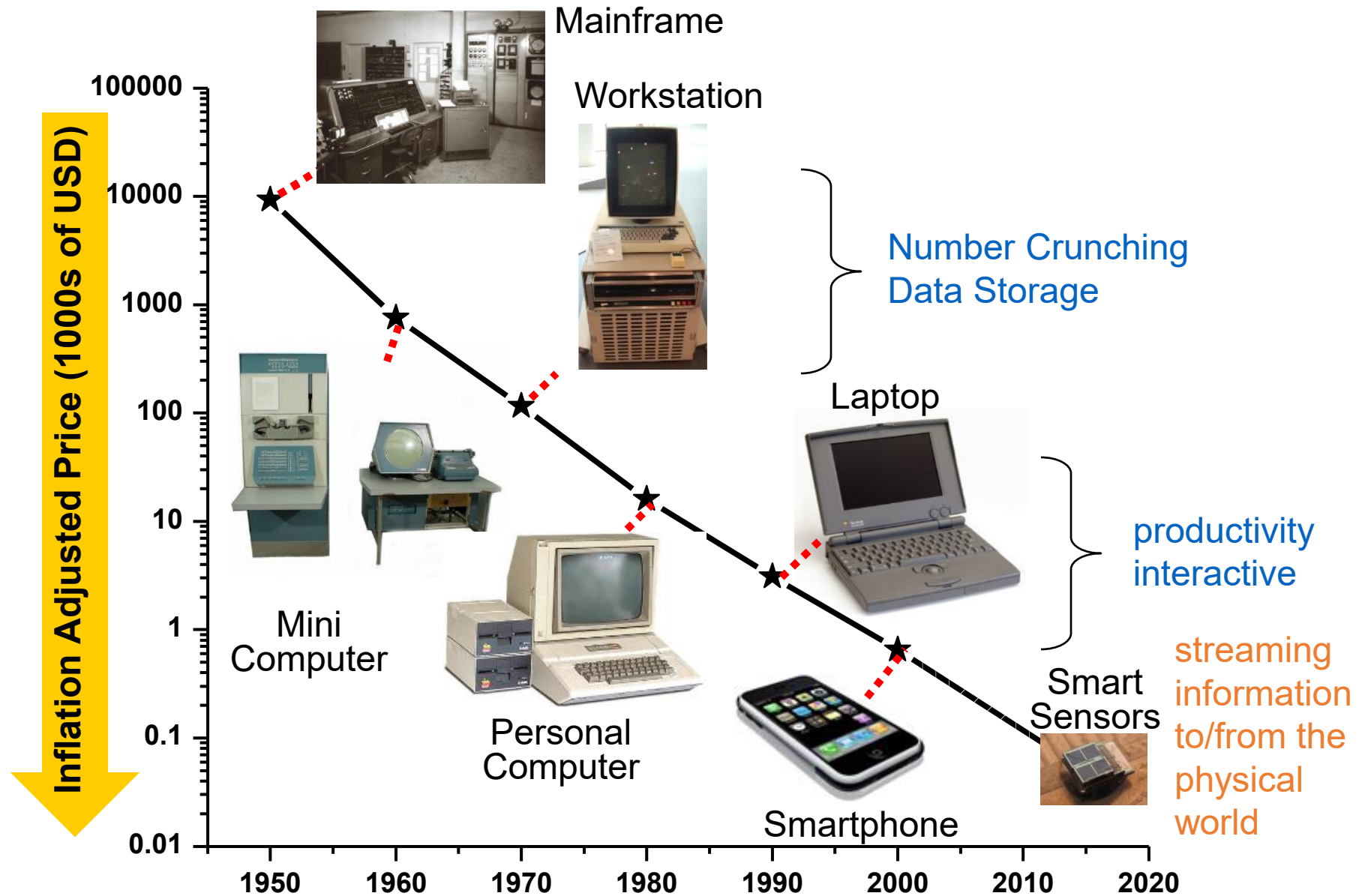
Number of computers per person grows over time



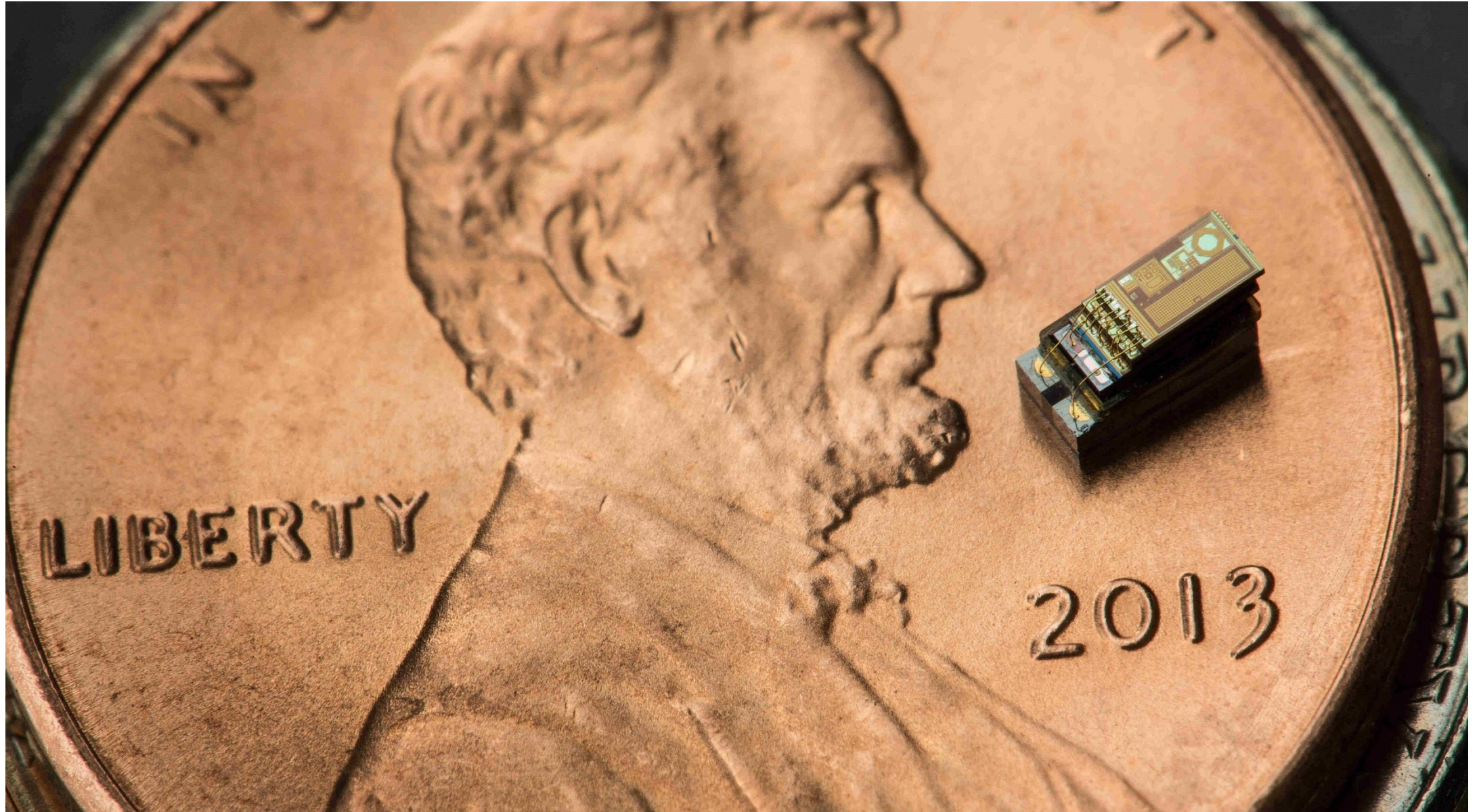
Computer volume shrinks by 100x every decade



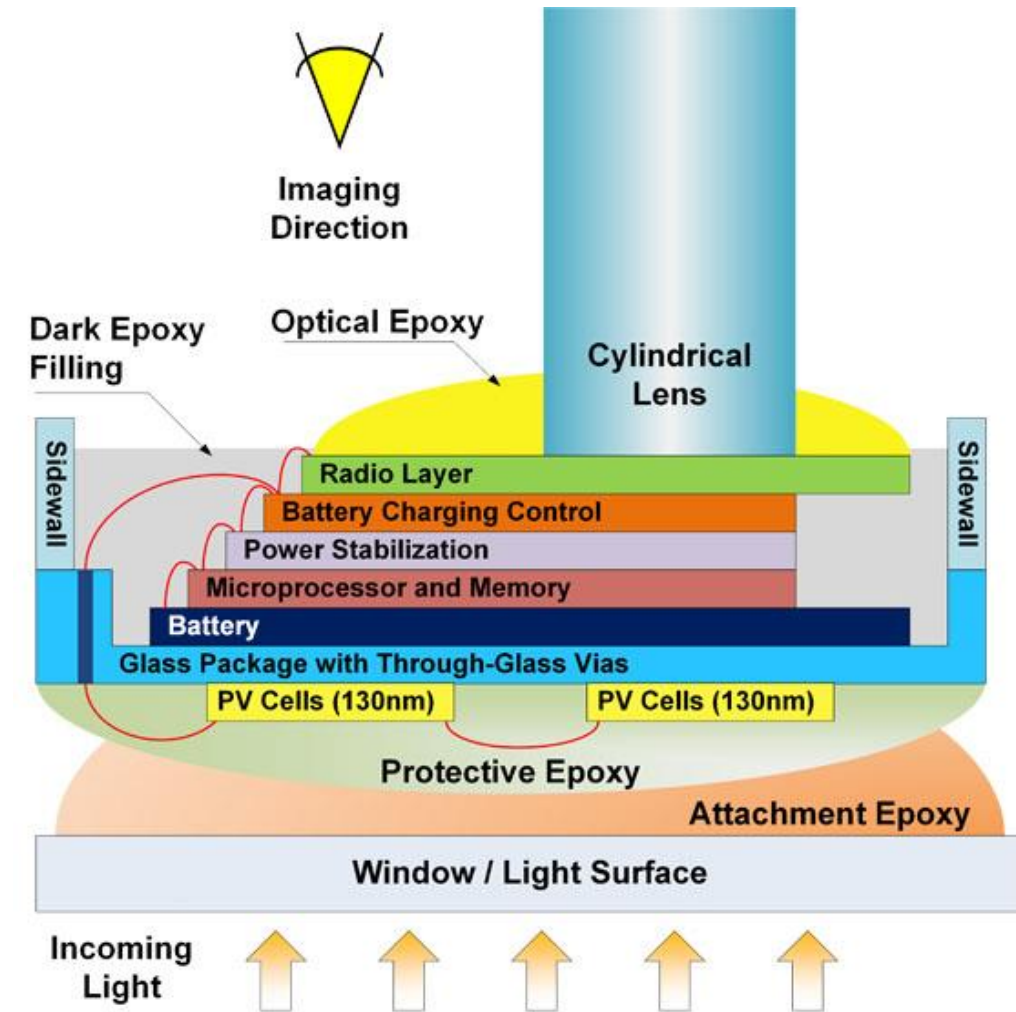
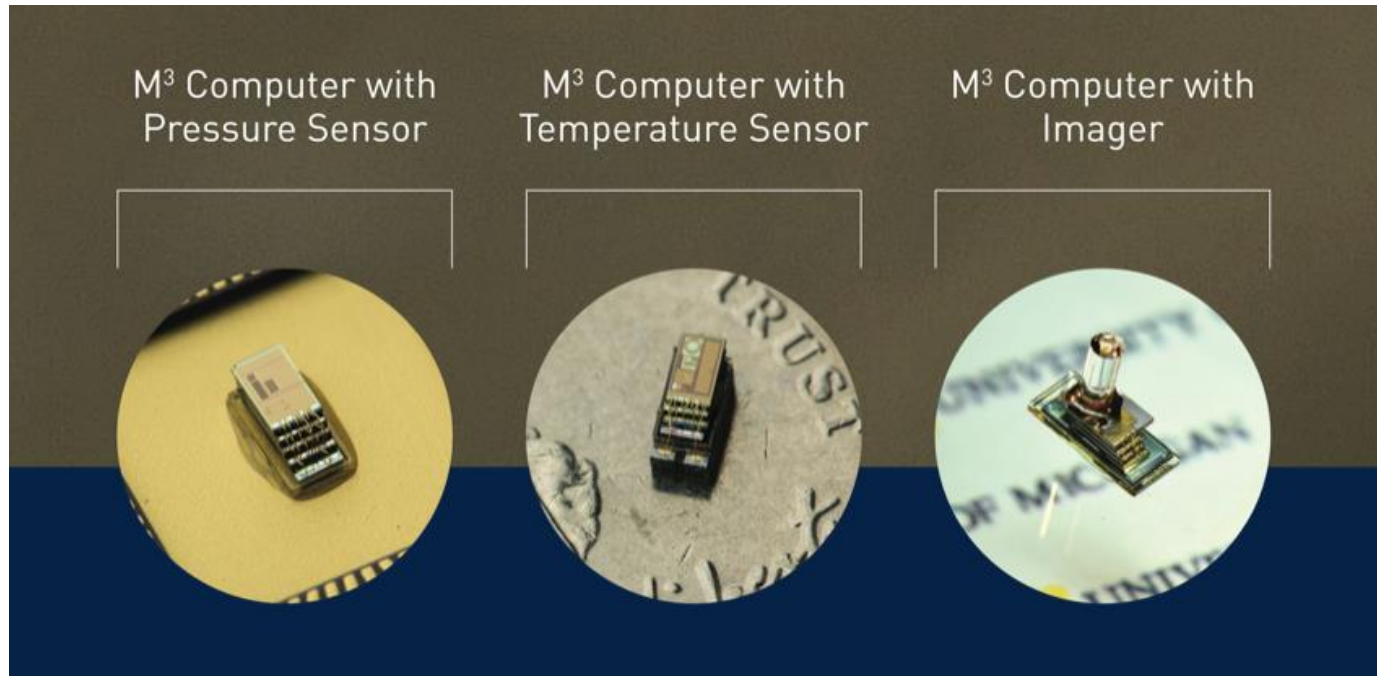
Price falls dramatically, enabling new applications



Michigan Micro Mote (2015)



M³ Use cases and design



<https://ece.engin.umich.edu/stories/michigan-micro-mote-m3-makes-history-as-the-worlds-smallest-computer>

Powering and Programming the Mote



Key idea: computers can be built into other things

- Computers have become small enough that they don't need their own rooms or even their own boxes to house them
- They can be built into other things we use in everyday life

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

What does and does not count

- Is _____ an embedded system? Why or why not?

- Smart lightbulb
- Roomba
- Autonomous vehicle
- Smartwatch
- Smartphone
- Voice-assistant (Alexa)
- Keyboard



Thought experiment: high-capability computing

- What if the smart lightbulb was powered by an entire desktop?
 - 16-core x86-64 processor, 64 GB RAM, 1 TB SSD
- Could that still be an embedded system?
- Why don't we see that in practice?

Thought experiment: high-capability computing

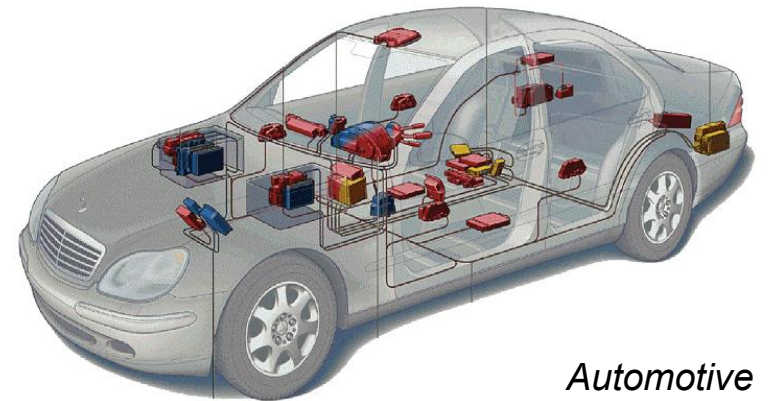
- What if the smart lightbulb was powered by an entire desktop?
 - 16-core x86-64 processor, 64 GB RAM, 1 TB SSD
- Could that still be an embedded system?
 - **Yes**
- Why don't we see that in practice?
 - **Cost**

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
- Example: thermostats existed without computers
 - But are cheaper to make *with* computers these days

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

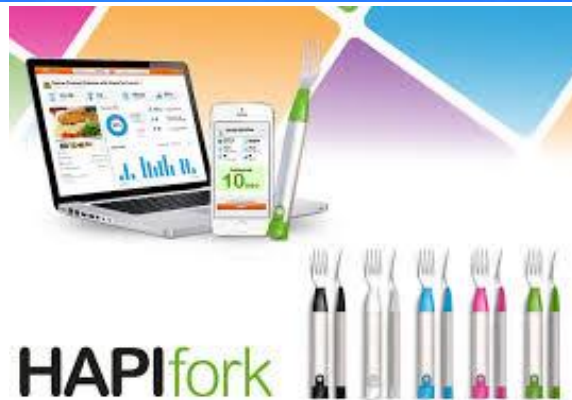


Related area: Internet-of-Things devices

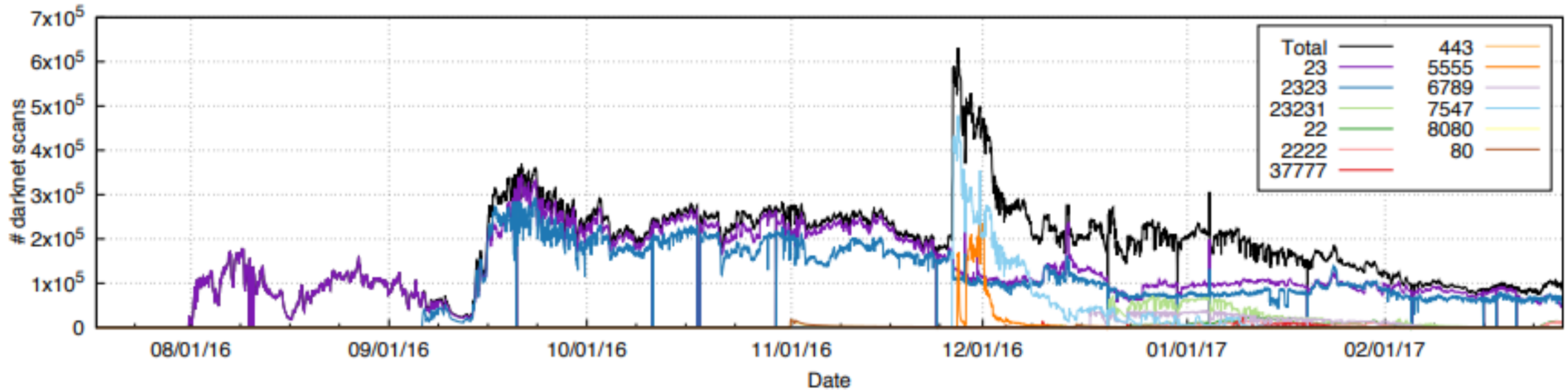


- Device + Computation + Networking

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?

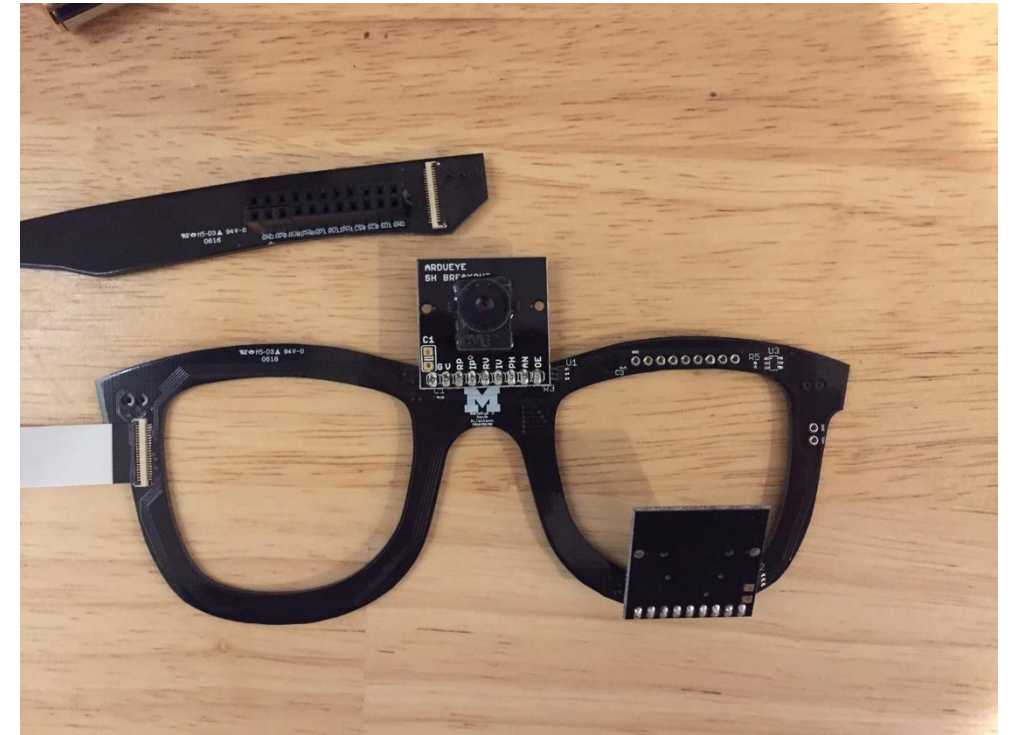
- Focus on the real world
 - You can actually see the purpose and effects of your applications
 - Easily explainable to non-engineer humans
- Challenging limitations
 - Limited memory and processing
 - Energy concerns

What makes resource-constrained embedded systems frustrating?

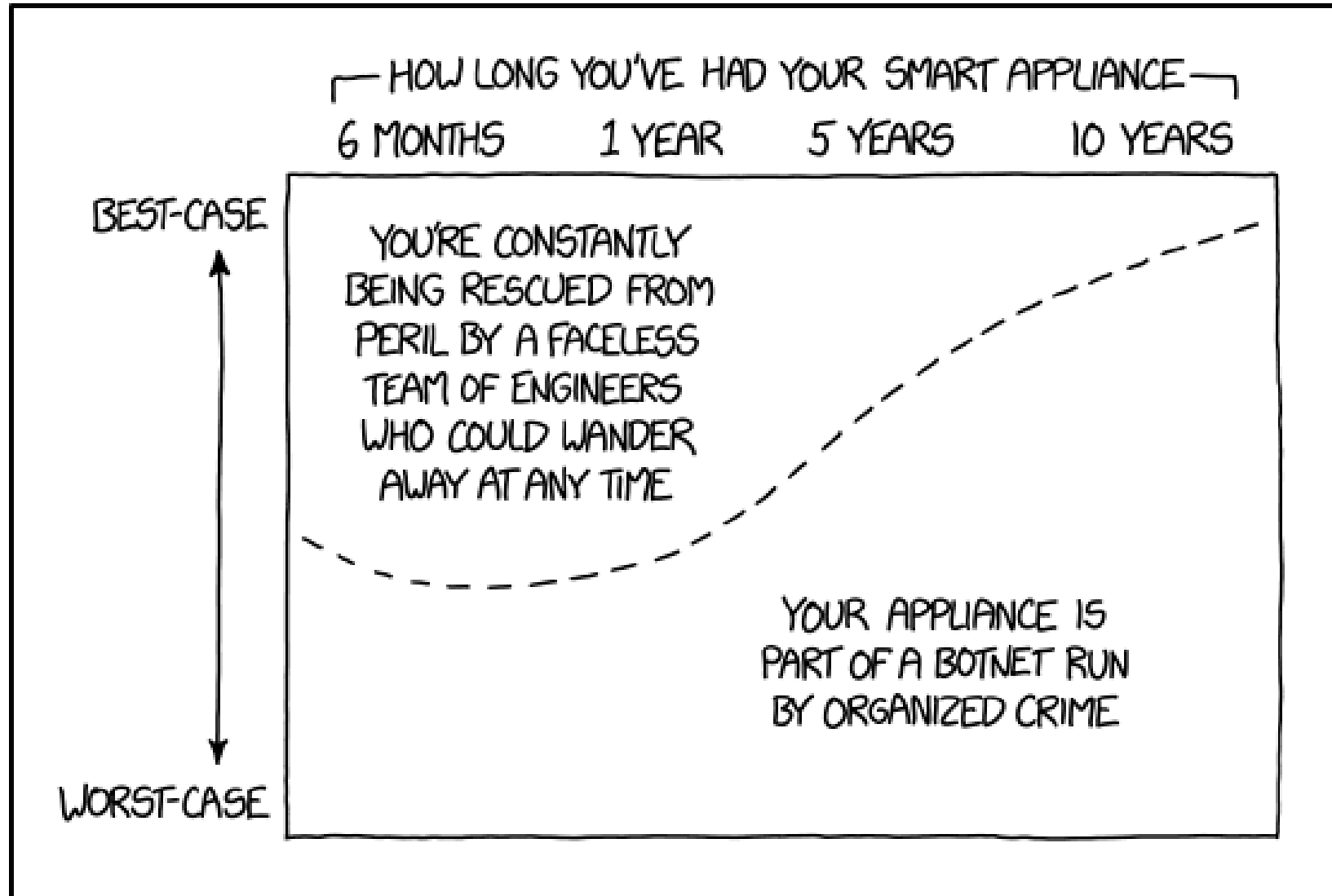
- Challenging limitations
 - Limited memory and processing
 - Energy concerns
- Full-stack development means problems could be *anywhere*
 - Hardware problems
 - Firmware problems
 - Software problems

What makes resource-constrained embedded systems frustrating?

- Example: my first grad project, eye-tracking glasses
 - Captured images of eyes and world, display on a nearby desktop
 - System stack-up:
 - Camera
 - > Analog-to-Digital Converter
 - > FPGA (hardware to manage camera)
 - > Linux driver
 - > Linux app
 - > Network
 - > Visualizer app
 - One morning, I came to lab and nothing worked...
 - This WILL happen to you



Break + xkcd



Outline

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

Course Staff and Office Hours

- Four PMs who previously took the class
 - Satviki Madaan
 - Jonathan Raport
 - Lucas Takayasu
 - Jayden Wool
- They will help out during labs and also provide lab office hours
- Office Hours: TBD
 - We'll post a schedule soonish (should be M, T, W, and R)
 - 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
 - Slides are posted to the Canvas homepage right before class
- 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:
 - 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
 - In-class at the end of class, schedule on Canvas
- 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 1483
 - Mandatory attendance for these (part of your lab grade)
 - Let me know ASAP if you're sick and will miss
- 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
 - Optional lab this Friday for software setup
- 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
 - You'll need to get checkoffs in lab or during office hours

Show up to lab session on time

- Labs start right away forming partnerships and starting work
- So showing up late is a problem
 - And I will deduct lab points for it if I have to

Late (assignment) Penalties

- 10% reduction in maximum points per day late
 - Just applies to the labs. Quizzes and Projects can't be late
- Sometimes stuff just doesn't work. Especially when we're working with hardware. We can be flexible about deadlines
 - If you're having problems and **tell us**
 - Less flexible if you don't communicate or if you started late
- In general, communicate for any problems outside of your control and we can provide flexibility

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 (give or take)
 - So make sure you're up-to-date on what we're talking about
- First quiz is Tuesday, September 30th (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)
 - Groups of three is encouraged
- Demonstrate course knowledge through any application
 - Microbit (99% required) in our C software system (99.9% required)
 - Various hardware I'll have on hand
 - Decent per-team budget for purchasing additional stuff (usually around \$40 per person in team)

Project Logistics

- Week 4: Proposals due
 - I'll get feedback to teams I'm concerned about
- 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 8-12: Labs are done and Fridays are used for project office hours
 - And a required project check-in, probably week 9
- Exam Week: Live project demos!!
 - Public demo session (invite friends!)
 - Tuesday of exam week (usually half of the class at a time for a 2-3 hour block)

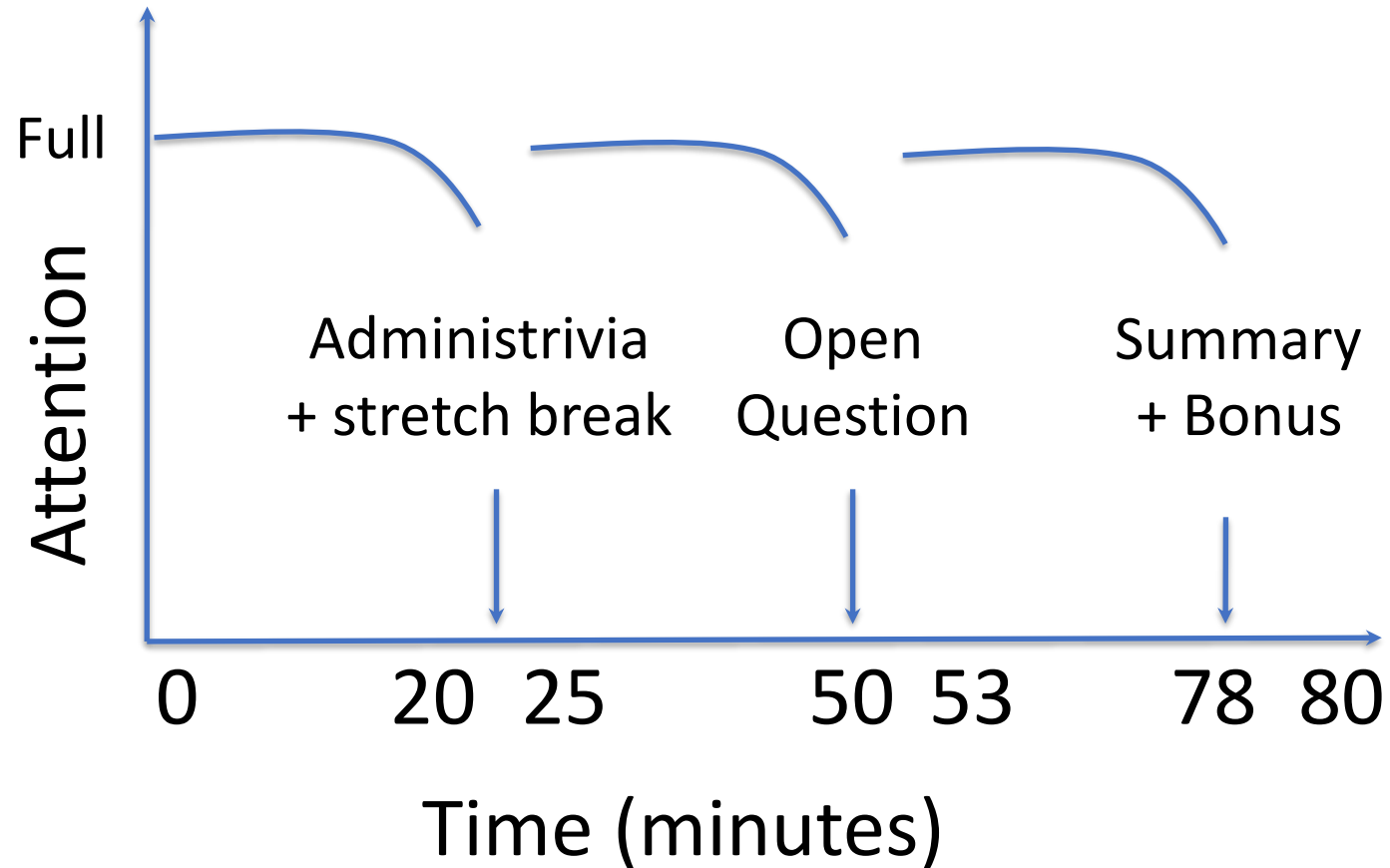
Let's talk about Generative AI

- Generative AI tools **MAY NOT** be used to develop **LAB** code
- Rationale:
 1. Want you to practice working on systems code
 2. Want you to be **significantly better** than AI tools
 3. We will help you complete the labs (in lab session and office hours)
- You may use AI tools for studying and understanding course material outside of labs
- You may use AI tools for the final project **as long as the source is properly documented**

Details about who is in the class

- 55 total students
 - Grad: 14
 - Undergrad: 42
- Majors
 - Computer Science: 26
 - Computer Engineering: 22
 - Data Science: 4
 - Electrical Engineering: 5
 - Robotics: 2
 - Mechanical Engineering: 1
- You don't all know the same things, and that's okay!!
- This class thrives on a variety of backgrounds
 - Software
 - Hardware
 - Data Interpretation
 - Applications
- The best projects often have a mix of backgrounds

Break + Architecture of a lecture

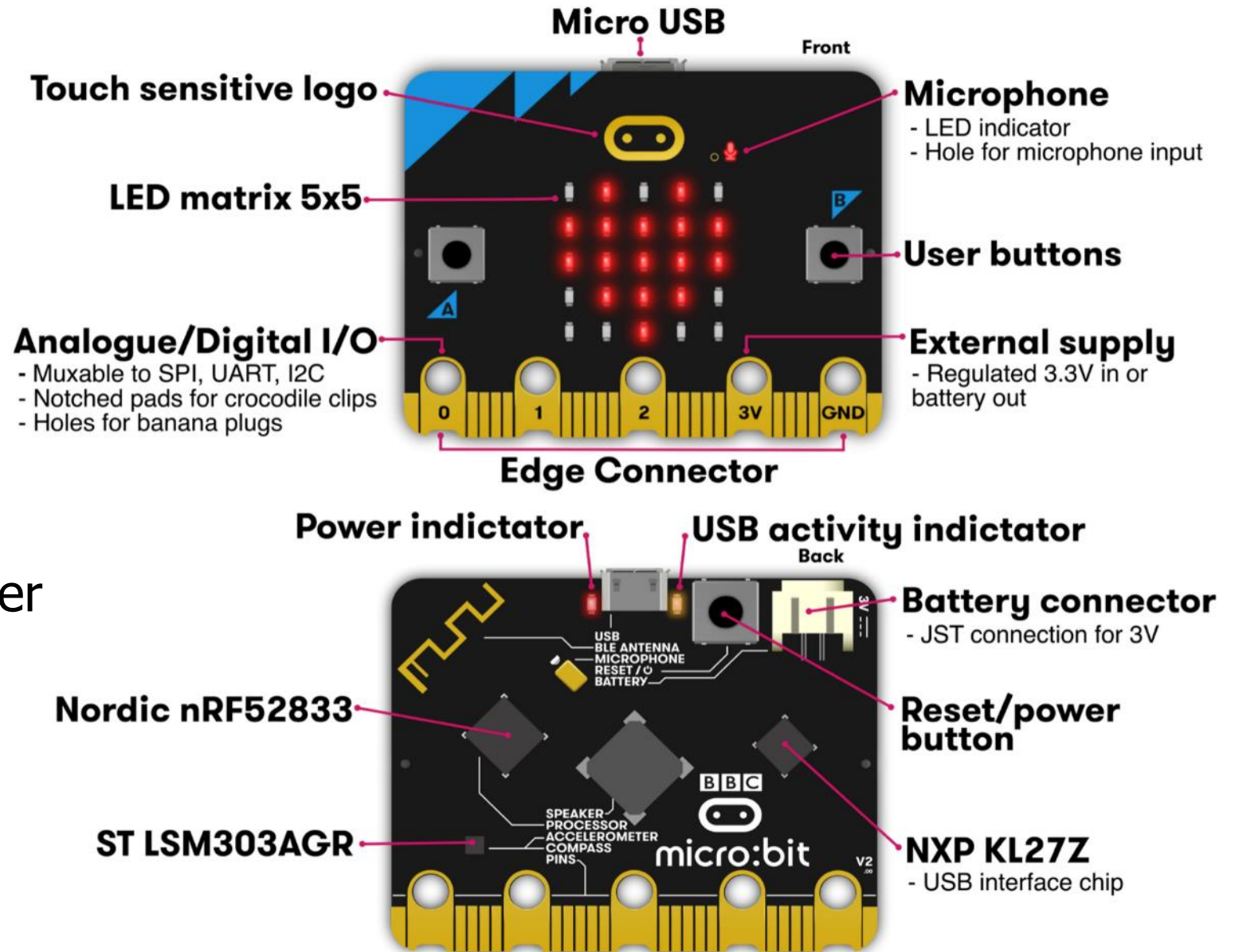


Outline

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

Micro:bit v2

- Legacy from 1980s "BBC Computer Literacy Project"
 - Reimagined today
- Around \$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>
 - \$19.95 Amazon: <https://www.amazon.com/BBC-Micro-Programming-Include-Battery/dp/B0BSF5KTLT>

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 cramped 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?

Lab0: Software Setup

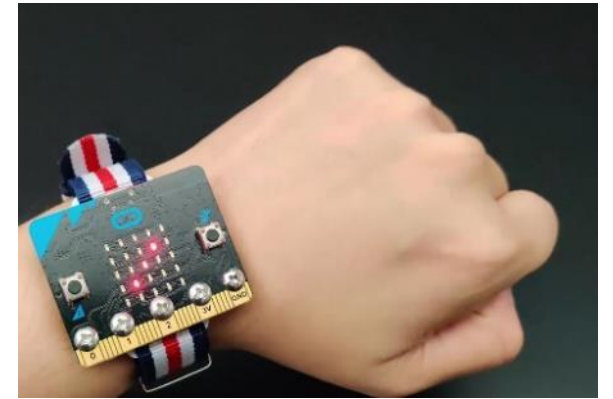
- Should release Thursday (maybe tomorrow)
- Optionally, I'll be at the lab sessions on Friday if you want help with your setup
 - If you have a weird computer setup
 - Or if you feel uncomfortable installing a bunch of tooling
- If you feel comfortable with it, you're welcome to do this first "lab" on your own
 - And you don't have to attend lab this Friday

Outline

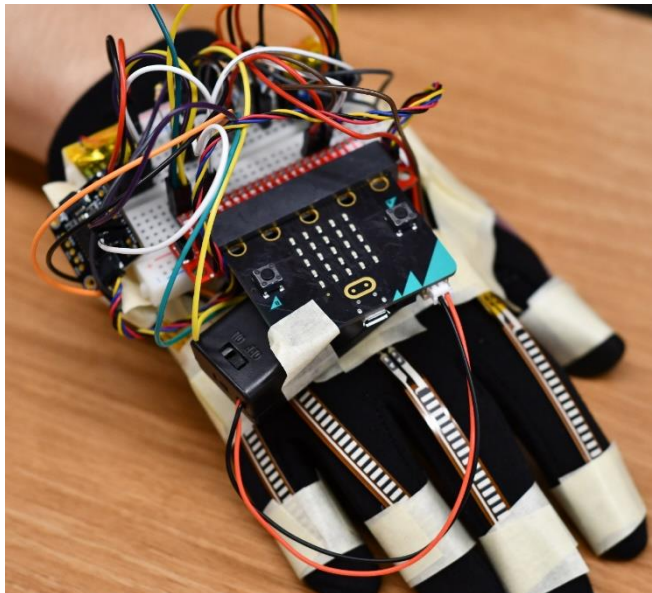
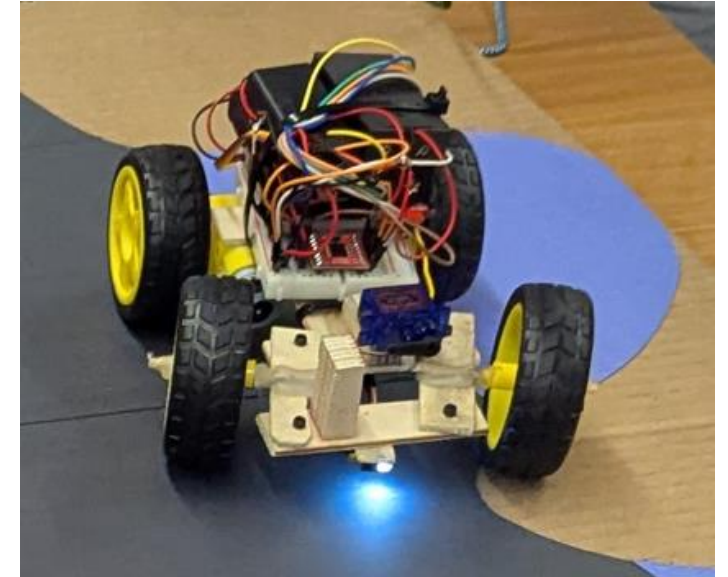
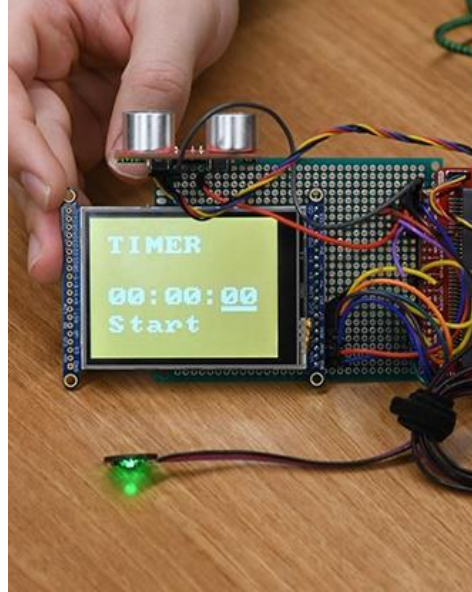
- Who and Why
- Embedded Systems
- Course Overview
- Class Hardware
- **Project Ideas**

Project Ideas

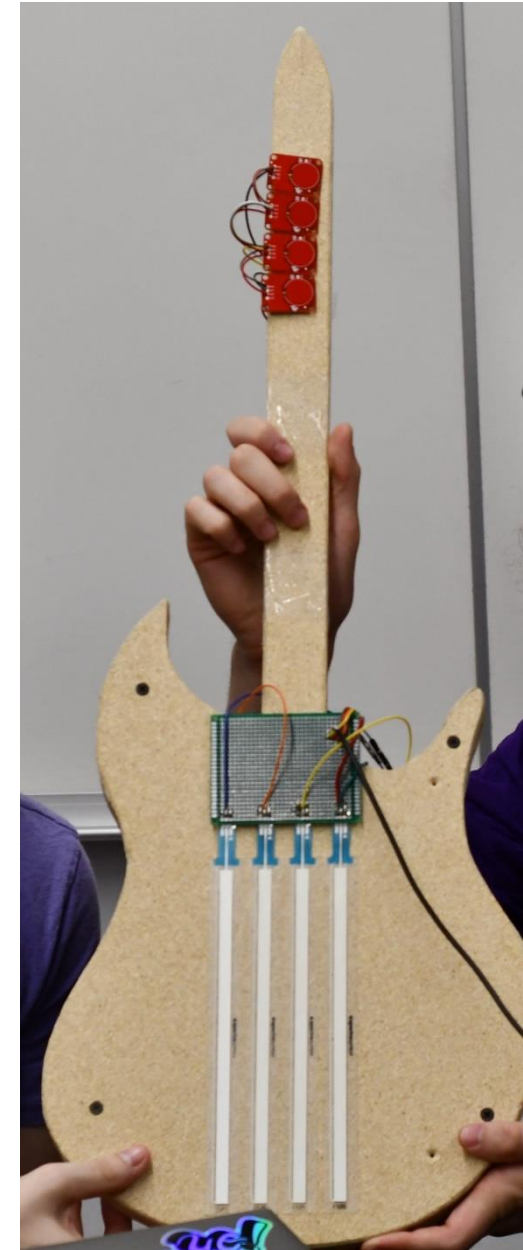
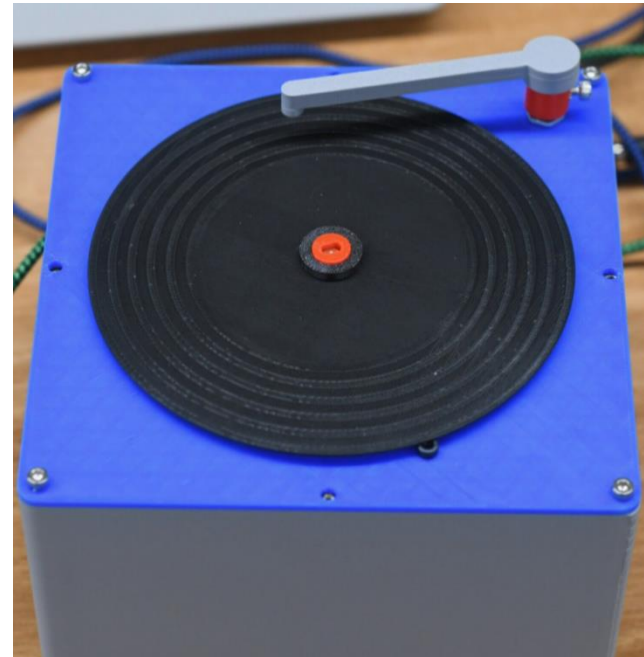
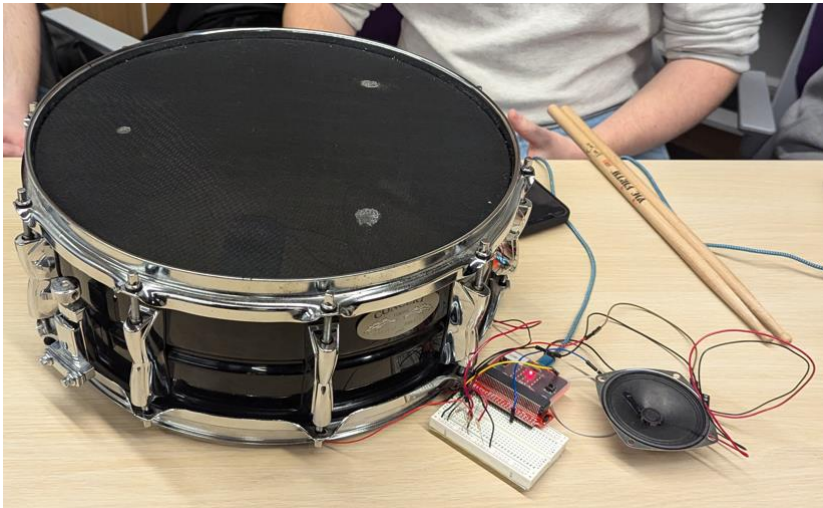
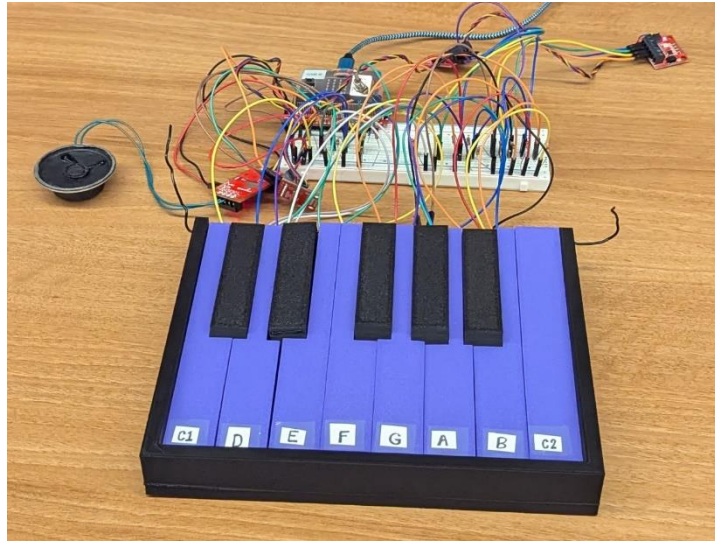
- Some ideas to get you thinking
 - Game with interesting control mechanism
 - Smart gloves
 - Smartwatch
 - Simple robotic systems
- Projects can use
 - Multiple Microbits
 - A personal computer for some amount of coordination
 - Small screens and displays
 - Lots of different sensors or actuators
 - Go explore [Sparkfun](#) and [Adafruit](#)



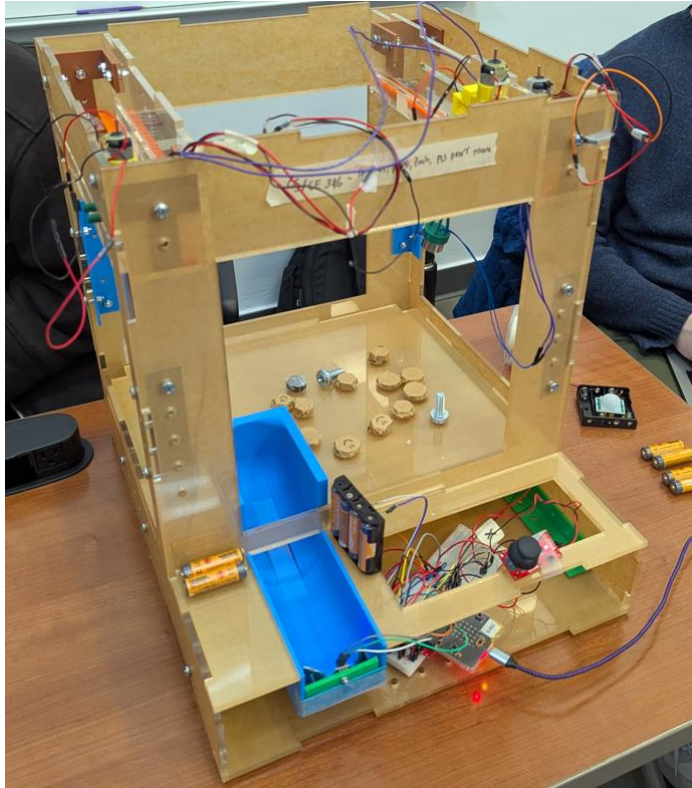
Some awesome projects – interactable objects



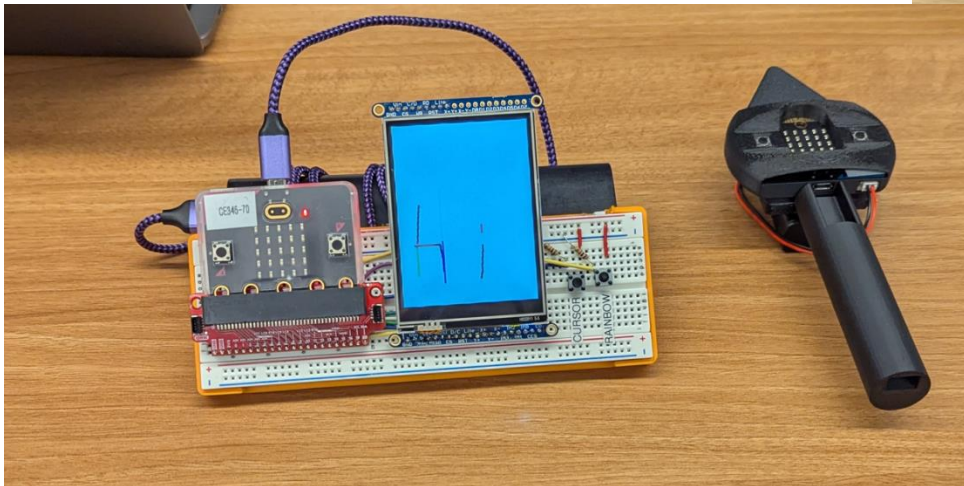
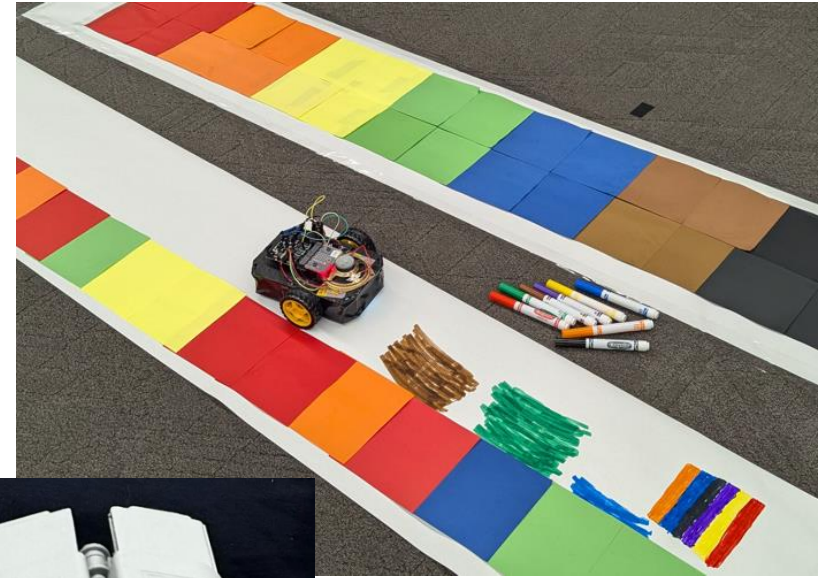
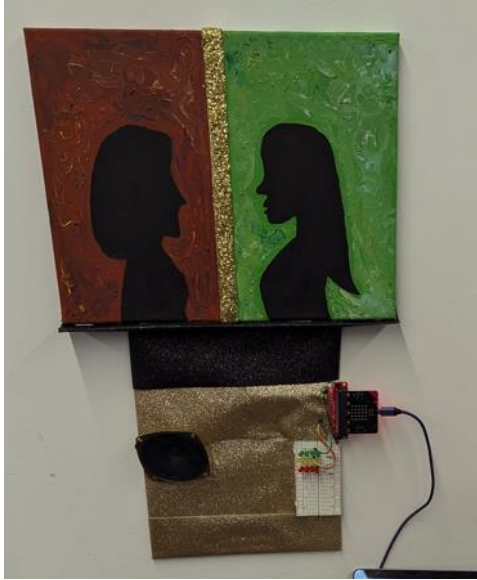
Some awesome projects – electronic music



Some awesome projects – playable games



Some awesome projects – interactable art and robotics



Outline

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