Lecture 01
Introduction

CS433 – Wireless Protocols for IoT

Branden Ghena – Spring 2025

Materials in collaboration with Pat Pannuto (UCSD) and Brad Campbell (UVA)

Welcome to CS433!

- Goal: get a **feel** for wireless communication and protocols
  - How are they used?
  - Why are they different and in what ways?
  - Which applications are they most useful for?

- 25 students (17 undergrad, 8 grad)
  - Lots of different backgrounds and interests
    (CS, ECE, Robotics, Data Science, Music, Mathematics)

- This course is based on discussion and questions
  - Expect to attend classes, ask questions, and interact with others
  - You're hopefully here because you want to be and want to learn

Branden Ghena (he/him)

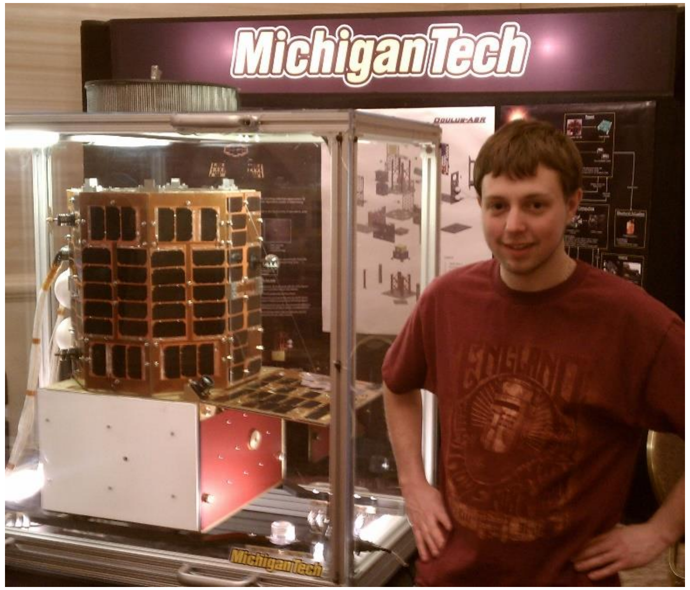
- Assistant Faculty of Instruction
- Education
  - Undergrad: Michigan Tech
  - Master’s: University of Michigan
  - PhD: University of California, Berkeley
- Research
  - Resource-constrained sensing systems
  - Low-energy wireless networks
  - Embedded operating systems
- Teaching
  - Computer Systems
    - CS211: Fundamentals of Programming II
    - CS213: Intro to Computer Systems
    - CS343: Operating Systems
    - CE346: Microprocessor System Design
    - CS433: Wireless Protocols for the IoT

Things I love

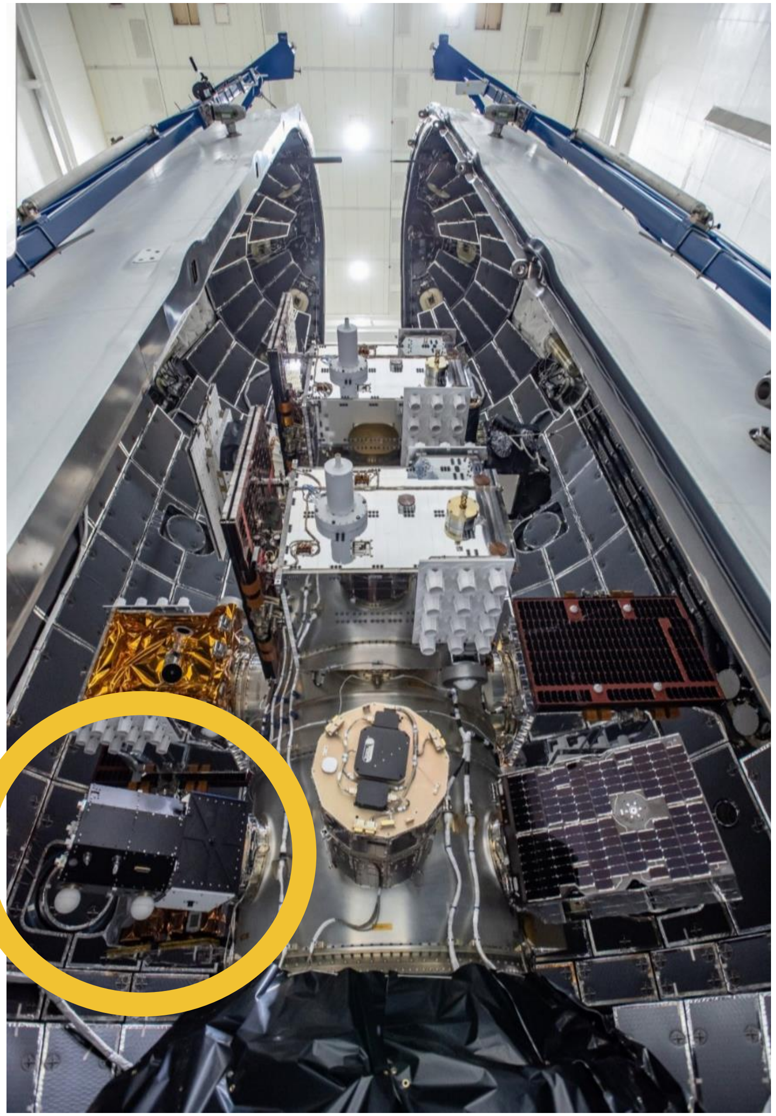
Undergraduate: satellite radios and computers

Oculus -Sat	NOTE:		Oculus -Sat	Date Data Last Modified:
Uplink Command Budget:			Version: 2.4.1	2011 October 12

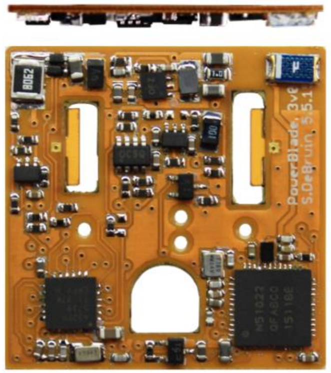
Parameter:	Value:	Units:	Comments:
Ground Station:			
Ground Station Transmitter Power Output:	50.0	watts	This value is transferred from "Transmitters" W/S, Cell [E15].
In dBW:	17.0	dBW	Transmitter power expressed in dB above one watt
In dBm:	47.0	dBm	Transmitter power expressed in dB above one milliwatt
Ground Stn. Total Transmission Line Losses:	3.3	dB	This value is transferred from "Transmitters" W/S, Cell [I33]
Antenna Gain:	16.3	dBi	This value is selected at "Antenna Gain" W/S, Cell [E11]
Ground Station EIRP:	30.0	dBW	Ground Station Effective Isotropic Radiated Power (EIRP) [EIRP=Pt x Ltl x Ga]
Uplink Path:			
Ground Station Antenna Pointing Loss:	0.3	dB	This value is calculated in the "Antenna Pointing Losses" W/S, and transferred f
Gnd-to-S/C Antenna Polarization Losses:	3.0	dB	This value is calculated in the "Polarization Loss" W/S and is transferred from C
Path Loss:	141.5	dB	Lp = 22 + 20LOG(D/l); Transferred from "Orbit & Frequency" W/S
Atmospheric Losses:	1.1	dB	This value is transferred from "Atmos. & Ionos. Losses" W/S, Cell [D23]
Ionospheric Losses:	0.7	dB	This value is transferred from "Atmos. & Ionos. Losses" W/S, Cell [D47:D50]
Rain Losses:	0.0	dB	This value should be estimated by the link model operator and place into Cell [
Isotropic Signal Level at Spacecraft:	-116.6	dBW	This is the signal level received in space in the vacinity of the spacecraft using

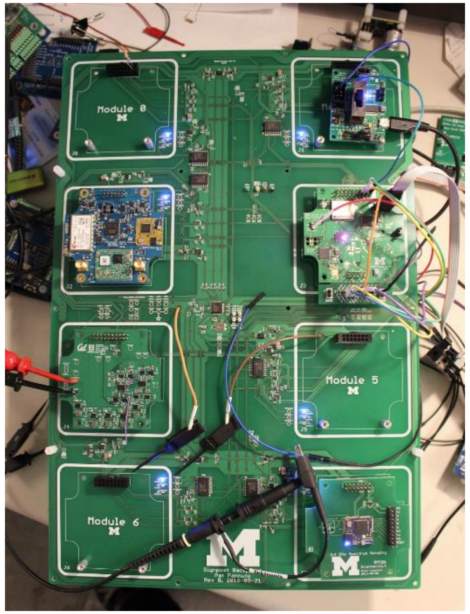


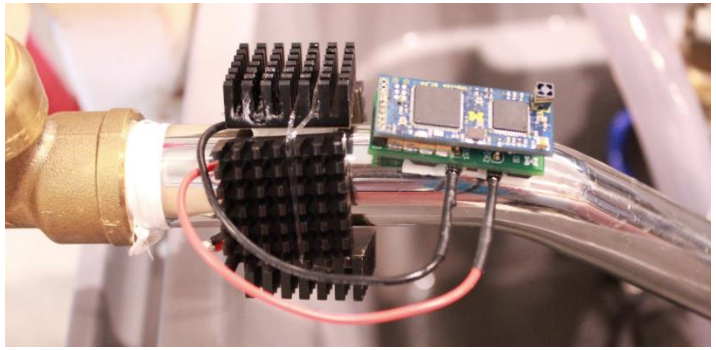
- How the heck are you supposed to learn this stuff?



Grad school: resource-constrained embedded systems







- Most interesting to me: the interfaces
  - Hardware and software
  - Applications and OS
  - Communication
- Again: learn by doing
  - With significant assistance from my peers

Faculty: now I can choose what to teach!

- Goal: provide classes that teach more advanced embedded systems topics
  - Overlaps strongly with CS and ECE, but hopefully useful to other engineering and sciences domains too

- Result: this course!
  - Course goals: make students familiar with a number of different wireless protocols and their tradeoffs
    - Practical hands-on experience with as many networks as possible
    - Open-ended design for final, chance to delve deeply into materials
  - Recently updated to a real course number: CS433
    - Let me know if you still see CS397 references around anywhere

Today's Goals

- Introduction to the Internet of Things and Embedded Systems
- Overview of the course
- Introduction to wireless communication

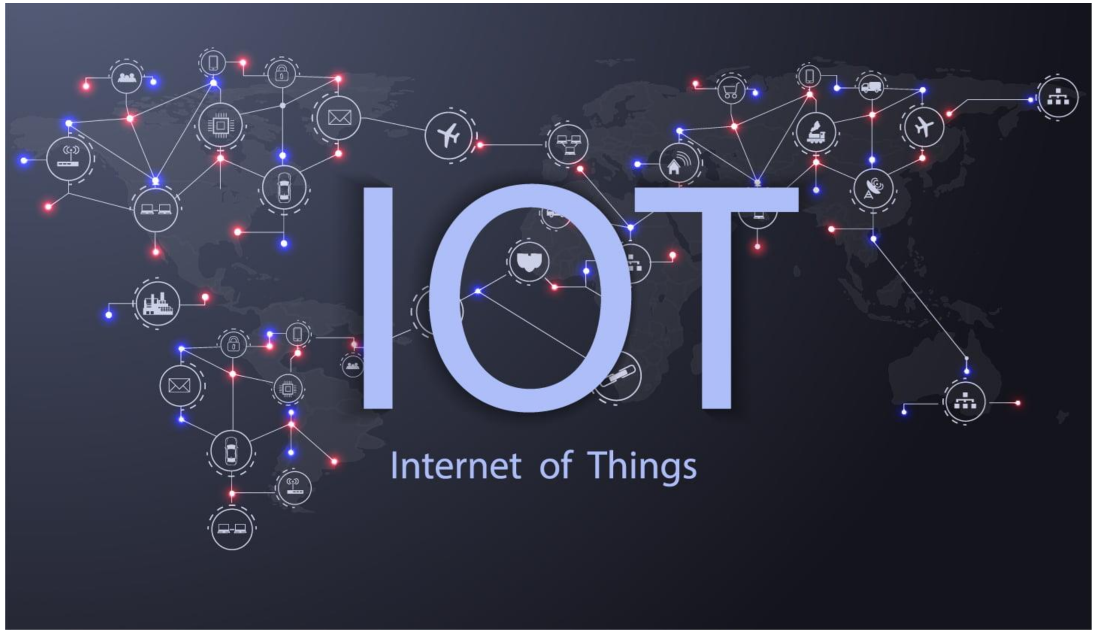
Outline

- Who and Why
- **Internet of Things**
- Embedded Systems
- Course Overview
- Overview of wireless networks

Perspective of this course

- This class is about wireless protocols
  - For a specific domain: the Internet of Things

- So we'll spend some amount of time discussing the Internet of Things and embedded systems



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”?

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: headphones

- What about battery powered wireless headphones?
- Would that still count as IoT?

Thought experiment: headphones

- What about battery powered wireless headphones?

- Would that still count as IoT?

  - Feels to me like a personal device rather than infrastructure
  - Something important about having many users

Thought experiment: desktop IoT

- 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?

- Why don't we see that in practice?

Thought experiment: desktop IoT

- 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?

- Why don't we see that in practice?

Cost

Focus on energy needs

- The origins of IoT are battery-powered sensing systems
  - And energy-harvesting devices

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

Focus on energy needs

- The origins of IoT are battery-powered sensing systems
  - And energy-harvesting devices

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

Deployability

Largest IoT challenges: *power* and *communication*

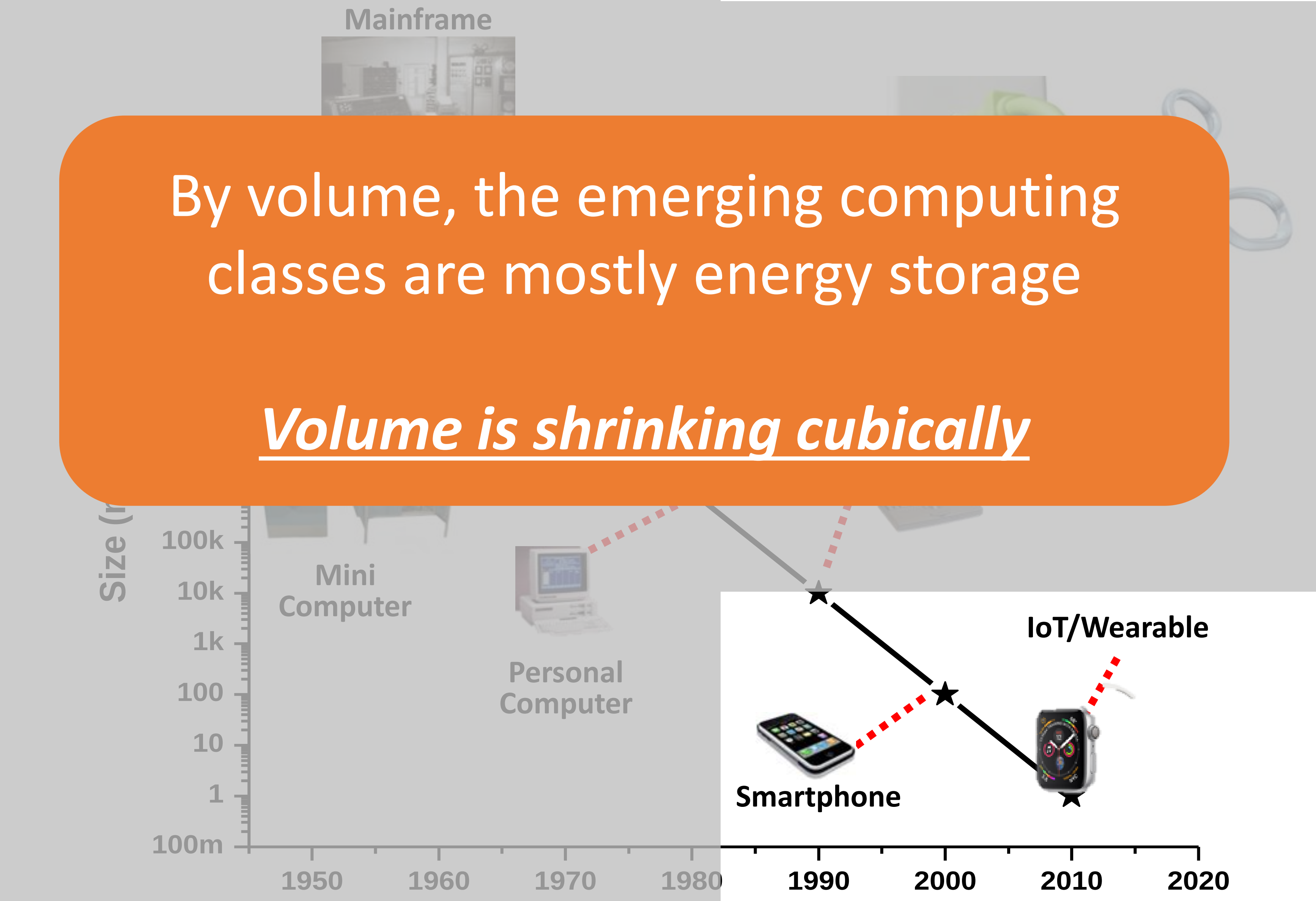
- This class is about wireless technologies
  - For resource-constrained systems, such as the IoT

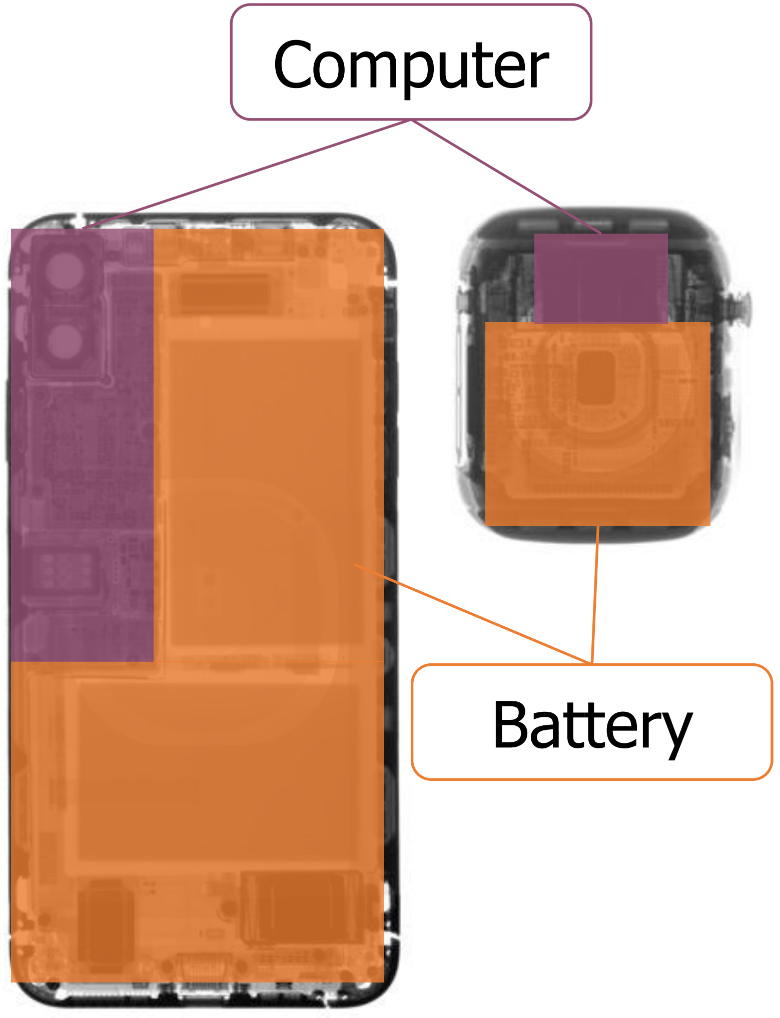
- We will focus on the tradeoffs between technologies
  - How they balance differing constraints
    - Power, spectrum, complexity, etc.
  - And the technical foundations of these designs and differences

Energy is *the* defining constraint of emerging technologies

By volume, the emerging computing classes are mostly energy storage

Volume is shrinking cubically





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

Pat Pannuto’s take on the Internet of Things

- His early grad school essays described the “last inch” problem
  - Now he often says “expanding the reach of digital world”

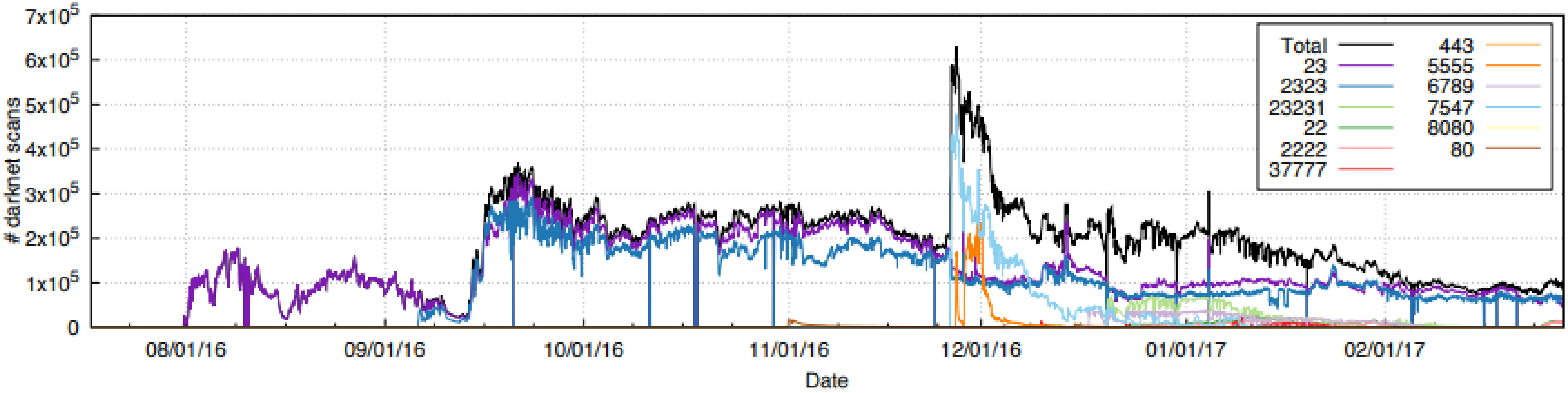
- For him, it is about ‘networked’ ‘things’
  - Which implicitly adds some computational capacity

Assistant Professor, UC San Diego
https://patpannuto.com/

He teaches a version of this course there, and we share materials back-and-forth!

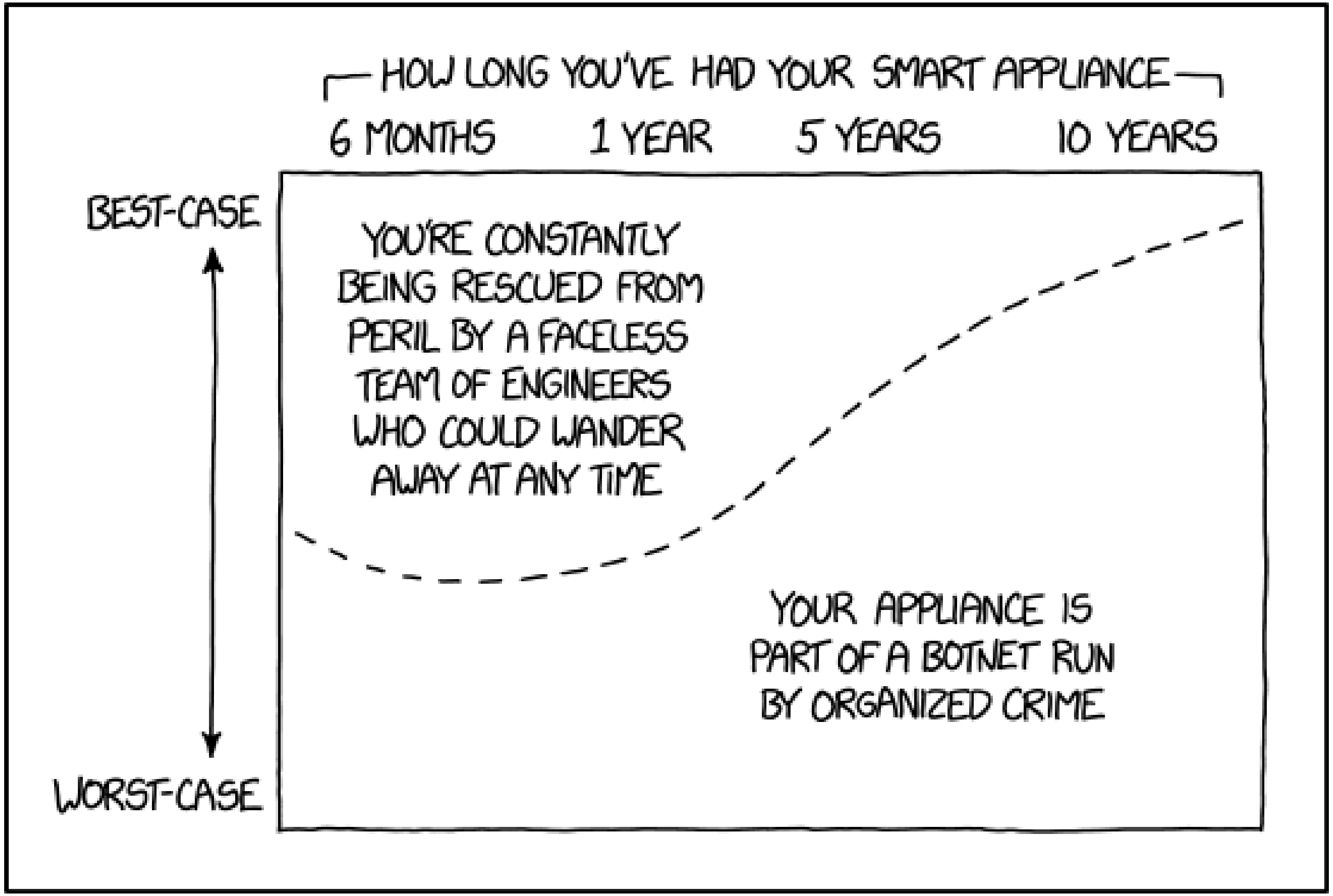
Warning: Internet of Crap

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

Break + xkcd



https://xkcd.com/1966/

Outline

- Who and Why
- Internet of Things
- **Embedded Systems**
- Course Overview
- Overview of wireless networks

What are embedded systems

- Embedded systems are devices containing computers that are used as a **device** rather than as a **computer**
  - Computers are **embedded** into the device

- Internet of Things devices are embedded devices

- But there are embedded systems that are not IoT
  - Smart vehicles and Robotics
  - Personal devices
  - Security or management chips inside other devices (like laptops/smartphones)

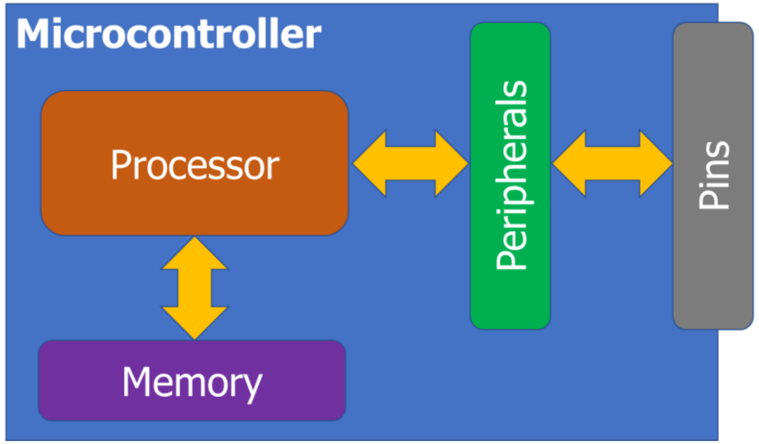
- Example: USB-C power brick has more compute than Apollo 11 did

https://forrestheller.com/Apollo-11-Computer-vs-USB-C-chargers.html
https://www.popularmechanics.com/technology/a30916315/usb-c-charger-apollo-11-computer/

Microcontrollers drive most embedded systems

- Microcontroller: entire computer in a single chip
  - Processor
  - Working memory: SRAM (like RAM)
  - Nonvolatile memory: Flash (like SSD)

  - Peripherals
    - I/O pins
    - Analog Inputs and Outputs
    - Timers
    - Wireless radios
    - Cryptography accelerators
  - Buses
    - UART
    - I2C
    - SPI
    - USB



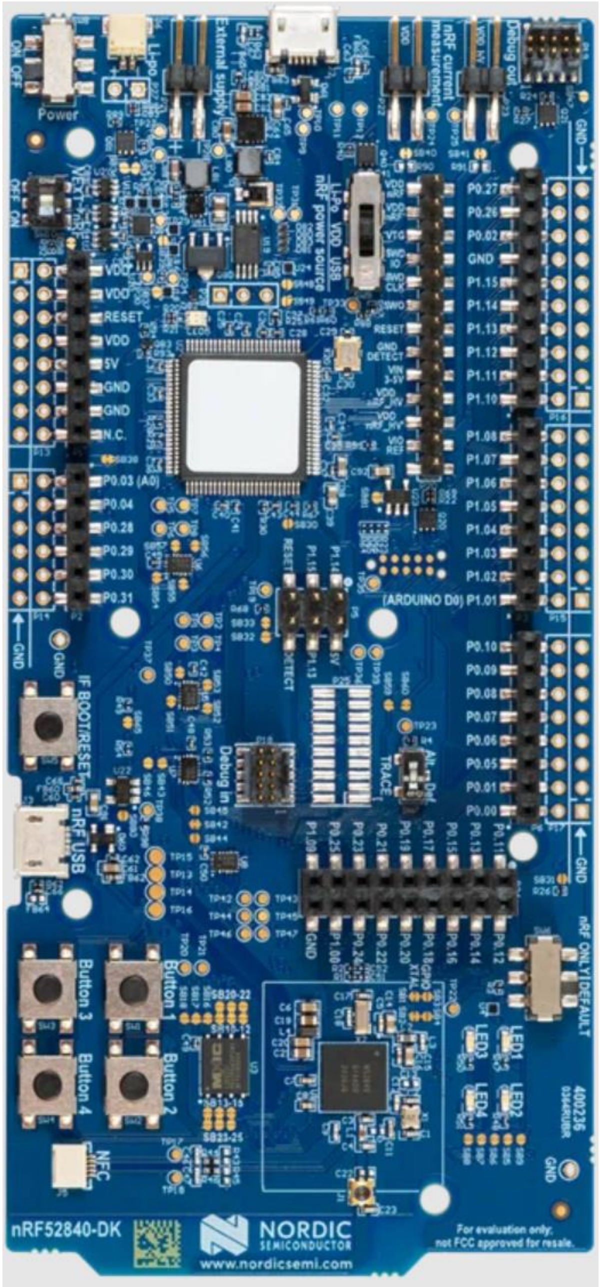
How is a microcontroller different?

- A very constrained computer
  - Simple processor
    - 16 or 32 bits (usually 32-bit these days)
    - Processor speed in MHz
    - Single core, pipelined processor
    - No cache, or maybe a very small instruction cache
  - Memory measured in kB
    - Code executes right from read-only Flash (which is part of the address space)
  - Sometimes no OS support at all
    - “bare-metal” programming

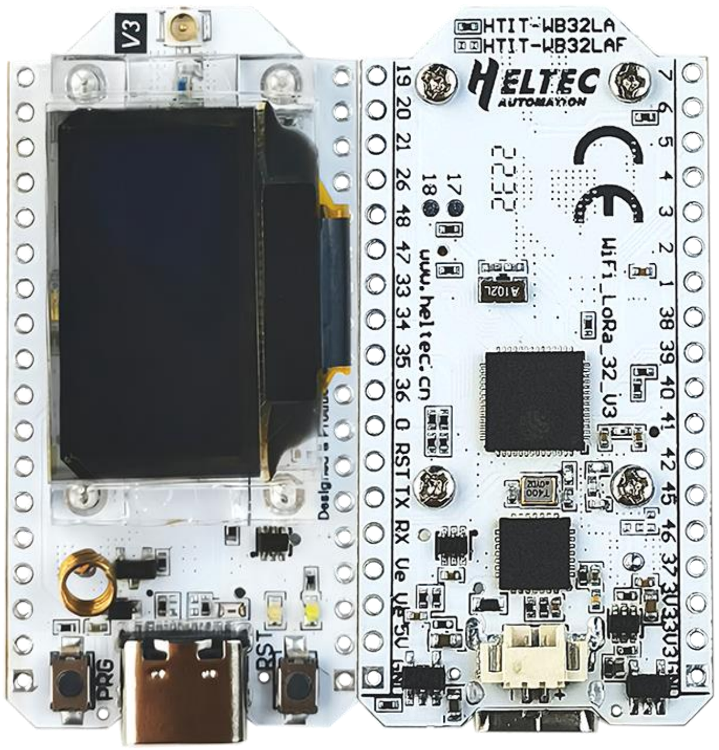
Nordic nRF52840DK - BLE & Thread

- nRF52840 microcontroller dev kit
  - Microcontroller is the bottom chip
  - Everything else is just to program it, measure power, and connect to external things

- nRF52840
  - 64 MHz, 32-bit ARM Cortex-M4F
  - 1 MB Flash, 256 KB RAM
  - A bunch of peripherals
  - Bluetooth Low Energy and 802.15.4 communication



Heltec WiFi LoRa 32 v3 - WiFi & LoRa



- ESP32 microcontroller
  - 240 MHz, 32-bit custom architecture
  - 384 kB Flash, 512 kB RAM
  - A bunch of peripherals
  - WiFi and Bluetooth Low Energy communication
- SX1262
  - LoRa transceiver (connected over SPI)
- OLED display

Embedded systems are programmed in C

- C++ also used
  - Occasionally assembly or other things (Rust, Lua, Python)

- But even the few things C gives you aren't necessarily available
  - Heap space possibly nonexistent
    - You have to choose some space in RAM to save as a heap
    - And then include the algorithm for allocating that memory

  - Printf could be nonexistent too!!
    - There's no STDIN/STDOUT/STDERR because there is no shell
    - Might be able to do serial output though – our boards can

Embedded software

- There are a multitude of embedded software systems
  - Every microcontroller vendor has their own
  - Popular platforms like Arduino

- Embedded OSes
  - Contiki, Riot, Zephyr, Mynewt, FreeRTOS, Tock

- nRF52840DK has support across all of these
- WiFi LoRa 32 v3 should work with some
  - Best support likely in Arduino

Break + Question

- What's the advantage of a microcontroller?
  Why not use a traditional computer processor instead?

Break + Question

- What's the advantage of a microcontroller?
  Why not use a traditional computer processor instead?

- Simplicity
  - Processor, memory, and peripherals are all in one chip
  - Makes the circuit board easier to design too

- Cheaper in practice
  - They don't sell as many low-end processors anymore
  - They still sell many low-end microcontrollers
  - (this is somewhat of a circular chain though)

Outline

- Who and Why
- Internet of Things
- Embedded Systems
- **Course Overview**
- Overview of wireless networks

Course Staff

- Instructor: Me!

- PM: Evan Bertis-Sample
  - Took the class last year!
  - Lots of Formula team experience on telemetry and data acquisition

- We’ll have weekly office hours to support you all
  - Schedule posted soon

Asking questions

- Class and office hours are always an option!

- Piazza:
  - Post questions
  - Answer each other's questions
  - Find lab partners
  - Information from the course staff
  - Post private info just to course staff

- Please do not email me! Post to Piazza instead!
  - Let me know if you don't have access

Lectures

- Tuesdays and Thursdays, in this room at this time
- No particular textbook makes sense for this class
  - Very few schools teach this area of communication/networks
- Automatically recorded (best-effort)
  - Panopto tab in Canvas
- Slides posted to Canvas homepage right before lecture

Grading

- 45% Lab projects - hands-on semi-guided activities
  - Small groups of three students (except for the first one)
  - Likely 5 of these in total
- 25% Homeworks - pencil-on-paper practice
  - Likely 4 of these, with the last worth double
  - Individual work
- 15% Quizzes - 15-minute, on-paper
  - Likely 3 of these
  - Tuesdays during the last part of class, dates on Canvas
- 15% Final design project - paper writeup
  - One of these due during exam week
  - Individual work

Labs

- Practice interacting with and considering networks
- Group work (teams of three) except the first one
- Should all include real hardware and real wireless communication
    1. Wireshark (worth a little less, individual starter lab)
    2. Bluetooth
    3. Thread
    4. WiFi
    5. LoRaWAN
- Graded partially (primarily?) on “checkoffs”

Friday lab sessions

- New this quarter: reserved time on Fridays to start labs
  - Time when you whole group is available to work together

- I'll be there to help answer questions and get you started

- You are expected to attend these when we're starting a new lab
  - Let me know in advance if you're sick and can't make it
  - Some Fridays are "Lab Office Hours", so I'll be there but you aren't required to be. Good time to get checkoffs though

Homeworks

- Analysis, designing, calculating
    1. Background
    2. BLE packets
    3. Matter
    4. Cellular (worth double)

- Usually take the form of some details on paper and you looking stuff up on your own
    - Looking through wireless specifications and searching on the internet

- Individual work

Late Policy

- You can submit homeworks and labs late

- 20% penalty to maximum grade per day late
  - Example: two days late means maximum grade is 60%

- There are exceptions to this:

1. We will be flexible with deadlines for problems outside of your control
   - Sick, family emergency, broken computer
   - Contact me (via Piazza) as soon as possible

Slip Days

2. Slip days let you turn in a homework late and receive no penalty

- Each student gets **3 slip days**
  - Apply to **homeworks and labs** (not final project or quizzes)
  - You don’t need to tell us you’re using them, we’ll just automatically apply them at the end of the year as best helps your grade
  - Be sure to coordinate about them on partner assignments

- Examples:
  - Turn in a homework three days late
  - Turn in a homework two days late and a lab one day late
  - Turn in a homework four days late with only a one-day penalty

Quizzes

- In-class, on-paper, closed notes individual quizzes
  - Usually about 15 minutes and held at the end of lecture
- Cover the last three weeks worth of material
  - So make sure you're up-to-date on what we're talking about
- First quiz is Tuesday, April 22^{th} (fourth week of classes)

Final design project

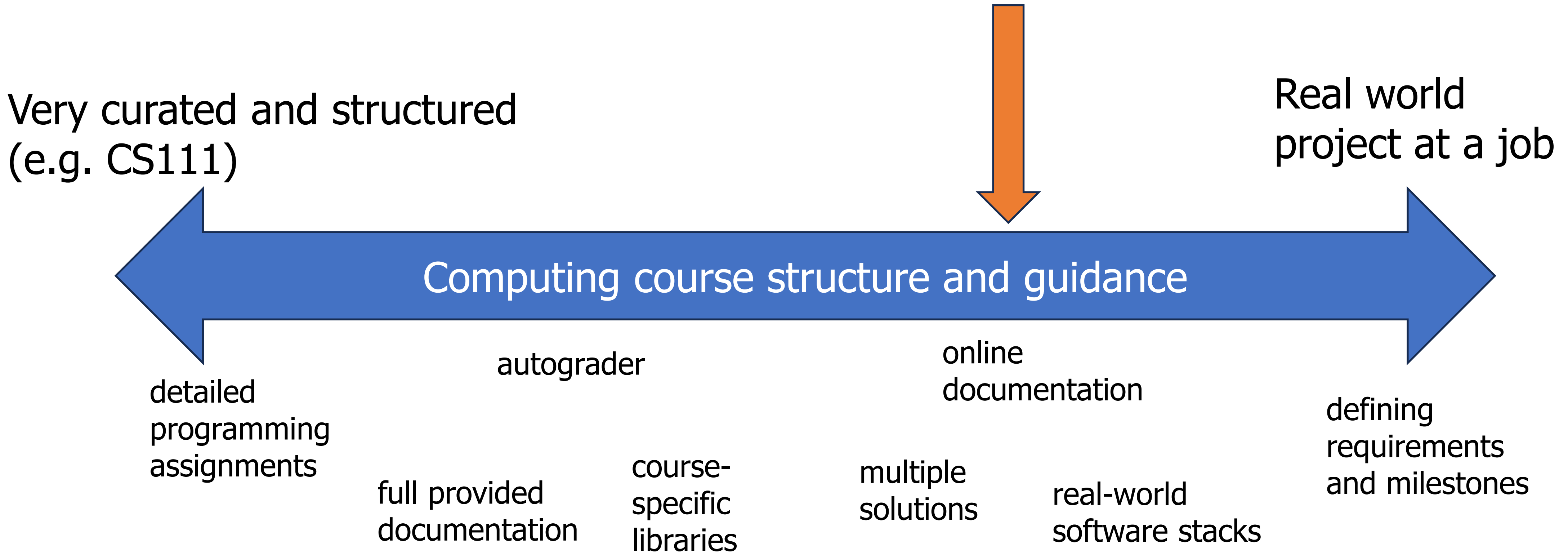
- Design a communication plan for an application of your choice
- Explain application and determine what its constraints are
  - Cost, power, scale, data, reliability
- Actually think through how it will work, citing real-world capabilities
  - Research and cite data you find
  - Explain and consider tradeoffs
- This is the “final project” for the course
  - Design rather than real hardware

On upper-level courses

- The point is to be here because you’re actually interested in the material
  - Come to lecture and discuss ideas with us
  - Focus on actually understanding stuff

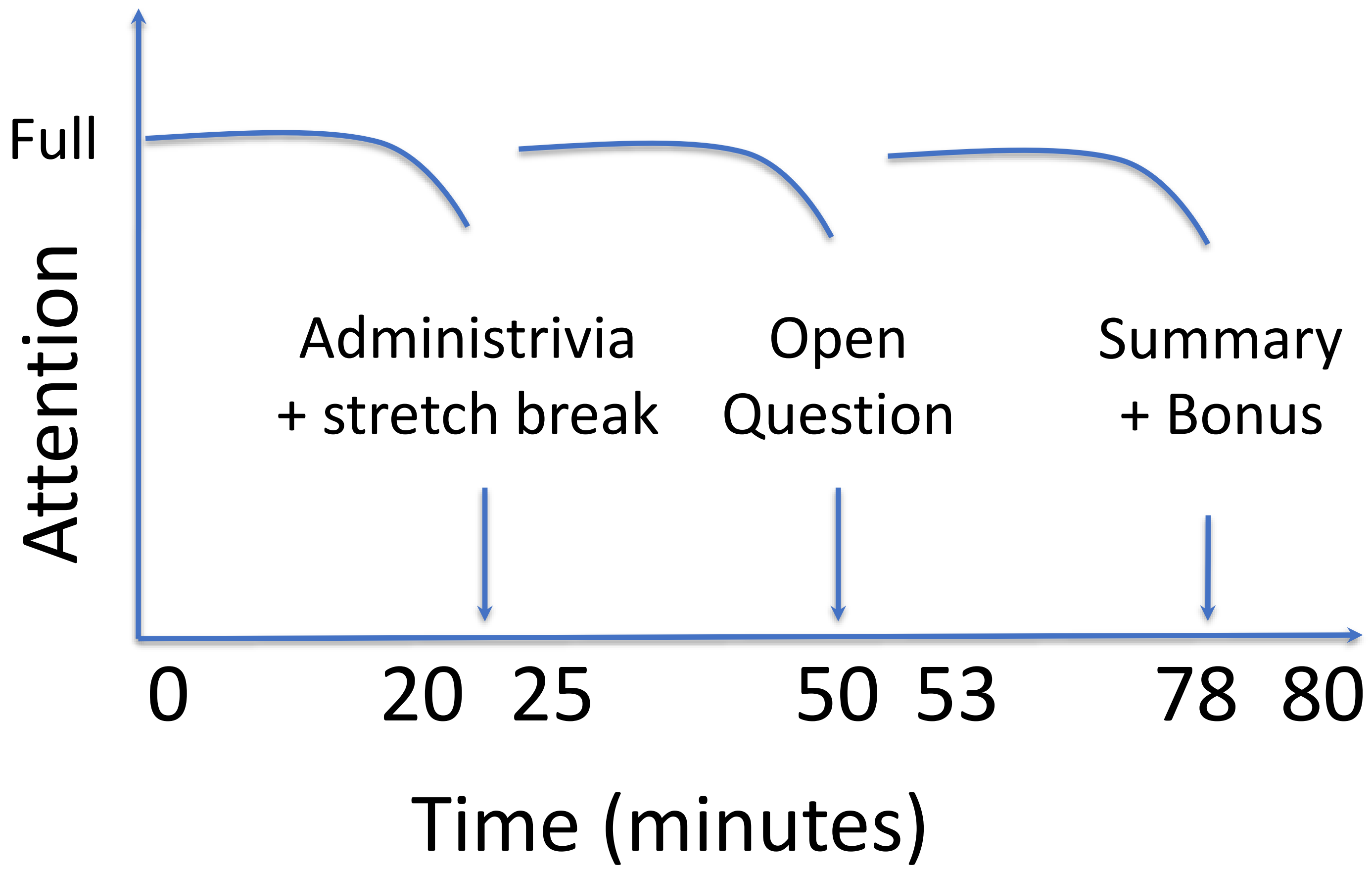
- You can help guide the course
  - It’s okay to derail lecture (sometimes) with interesting tangents
  - I’m happy to add material on other topics if you ask

What to expect



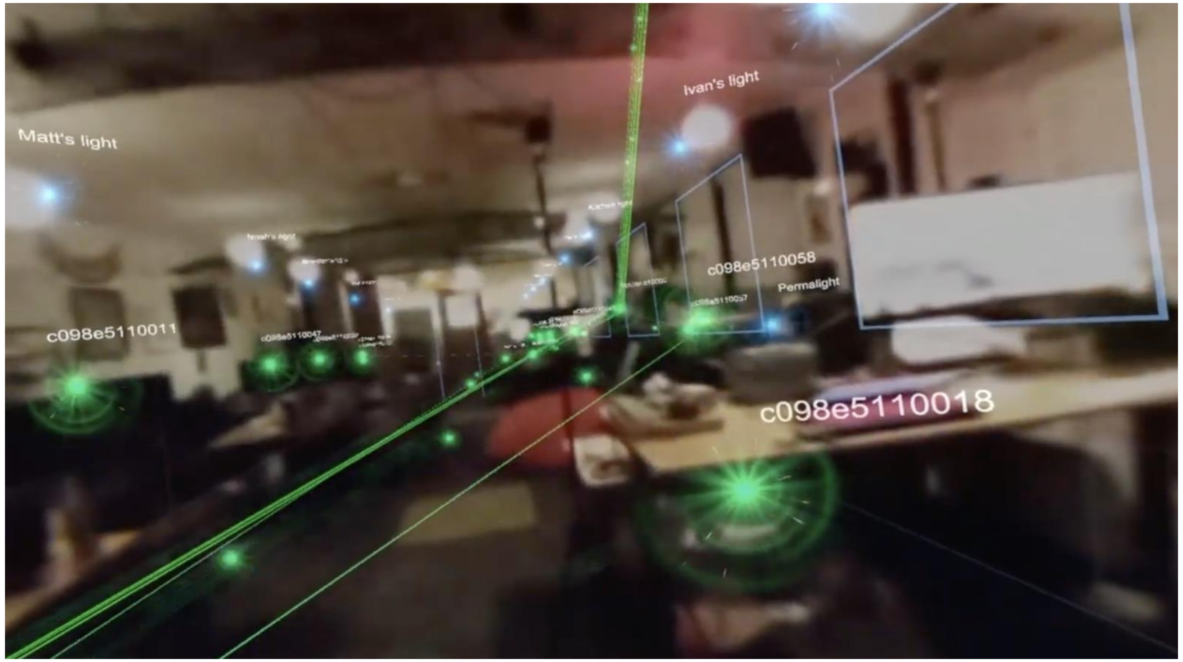
- Figure stuff out on your own! (or with a group)
- Goal is to help prepare you to engineer projects in the real world

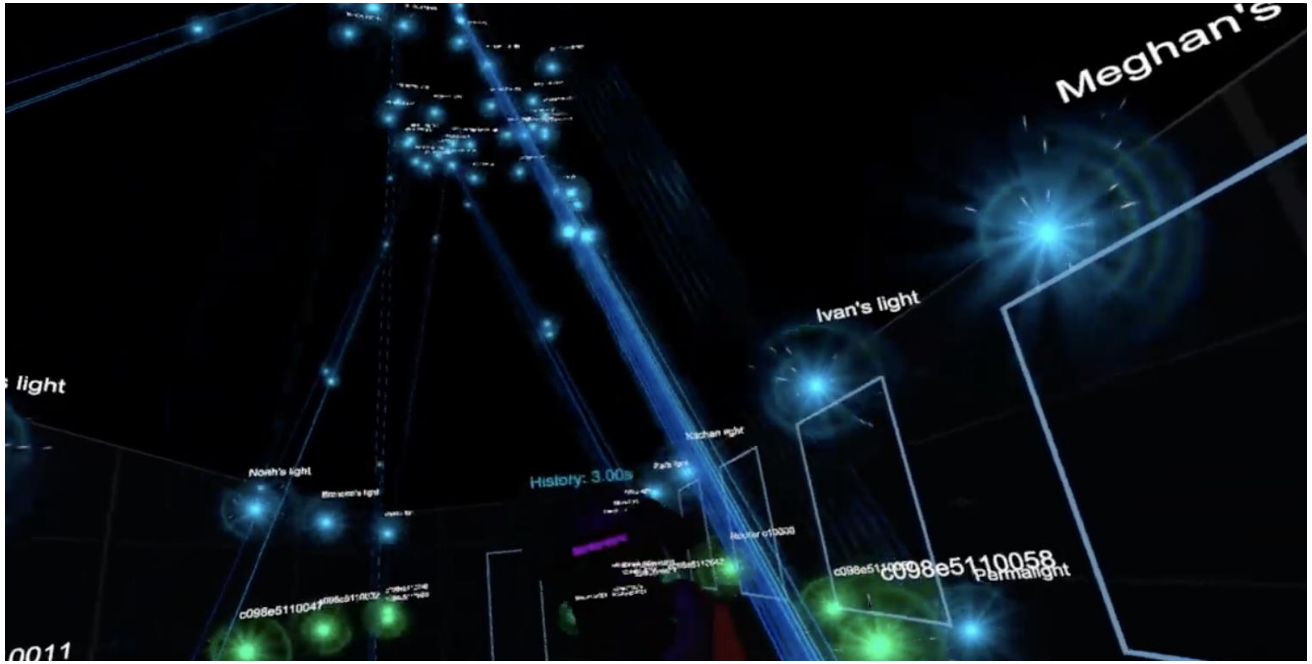
Architecture of a lecture



Break+Video

Wireless Network Visualization (Dr. Meghan Clark – UC Berkeley)
https://www.youtube.com/watch?v=KLOdp54_qJ4





Outline

- Who and Why
- Internet of Things
- Embedded Systems
- Course Overview
- **Overview of wireless networks**

Bluetooth Low Energy (BLE)

- Bluetooth Classic was good for enabling device to device communication
  - But not particularly low energy
  - And very bad at short-term, one-time connections
- Bluetooth Low Energy was developed to improve this
  - Focuses on low-energy interactions
  - Much lower throughput that Bluetooth
- Supported by hardware devices already in smartphones
  - Humans can interact directly with nearby devices!!

802.15.4 & Thread

- 802.15.4 is a low-energy physical layer
  - Radio chips have been widely available for 15-20 years

- *Significant* amounts of sensor network research have focused on building layers on top of 802.15.4
  - Access control layers
  - Network layers

- Thread is a selection of these possibilities to make a network
  - Uses IPv6 networking!!
  - Basis for modern smart-home standards: Matter

WiFi (802.11)

- Ubiquitous wireless communication
  - High energy requirements for high throughput communication

- Now accessible through relatively low power radios
  - ESP32 is dominant here
  - Still significantly more effort than BLE or Thread

- IoT devices can use the same WiFi that’s already available
  - No need for additional infrastructure!!

LPWANs (Low-Power Wide-Area Networks)

- How do we collect data from city-scale deployments?
  - There's an unmet need for long-range, but low-throughput networks
  - Existing cellular technologies focus on human requirements

- Still a brand new space (relatively)
  - Unlicensed-band technologies in last 10 years: Sigfox and LoRaWAN
  - Cellular technologies in last 4 years: LTE-M and NB-IoT

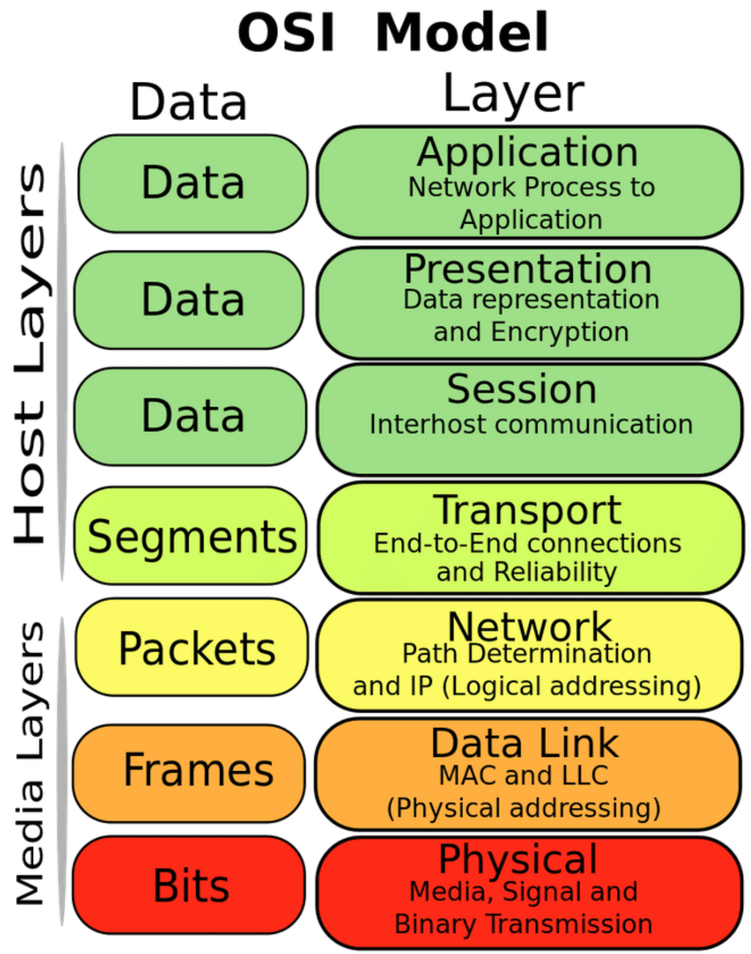
- Focus on long-range, low-energy, low-throughput
  - One gateway can cover an entire city!!

Extra topics

- Backscatter
  - Insanely low-energy communication
  - Enables energy-harvesting indoor devices
- Localization
  - How do we find all this stuff?
  - And how do devices determine where they are relative to each other?
- Satellite Communication
  - Growing use-case
- Other topics are possible if desired! (seriously!)
  - Reach out and tell me what you want to learn about (that's clearly a class topic)

OSI model of communication layers

- Transport
  - Sending data between applications
  - TCP and UDP
- Network
  - Sending data between networked computers
  - IP
- Data Link
  - Sending collections of bits
  - Ethernet, WiFi
- Physical
  - Sending individual bits
  - Ethernet, WiFi



Where does this class focus?

- Transport
  - Sending data between applications
  - TCP and UDP
- Network
  - Sending data between networked computers
  - IP
- Data Link
  - Sending collections of bits
  - Ethernet, WiFi
- Physical
  - Sending individual bits
  - Ethernet, WiFi

CS domain
CS340, CS440

This course!

EE domain
EE307, EE378, EE380

Outline

- Who and Why
- Internet of Things
- Embedded Systems
- Course Overview
- Overview of wireless networks