Computer Organization & Architecture

Welcome

Brian Gormanly
CMPT 422N
Marist College

Agenda

- Syllabus
  - Topics
  - Labs / Content
  - Projects
- Org & Arch Introduction
- Me / Research

Quick Summary

- Course is an intense ‘deep dive’ into how computers actually work, bit by bit.
- Lecture Focus: Book / Theory
- Lab component: Build a virtual computer!
- Desired result: Deep understanding by combining both.

Textbook

Patterson, Hennessy: “Computer Organization and Design, **ARM Edition**”, Elsevier, 2017.

ISBN-13: 978-0-12-801733-3 | ISBN-10: 0128017333

I will be using the ARM edition, there are many.

While I have my own content that largely drives the course forward Organization and Architecture are well understood subjects in CS and reading along in our text will substantially add to your experience!

I will occasionally use content from other texts, but will provide study material in slides or handouts when I do.

Week	Topics	Text (P = provided)	Quiz	Due
1	Introduction Abstraction / Systems	P		Lab 0
2	Comp Org. & Arch / Major Comp. Computer Systems Software	Chapter 1 Chapter 1		
3	Number Systems Mathematics in Bases	P		CP1 - Lab 1
4	Mathematics in Bases Q1		Quiz 1	
5	Numerical Representation Numerical Representation	P P		Full - Lab 1
6	LMC LMC / ISA Algorithms	P P		
7	ISA in Practice / System Calls Q2	2-2.5 2.5-2.6	Quiz 2	
8	Review Midterm			
SB				
9	Logic and Memory Design	P, A1,A2,A5,A7, A8,A9 4-4.4		Lab 2
10	CPU Datapath CPU Pipelining and Parallelism	4.5-4.12		
11	Q3 Memory and Caching	5-5.4, 5.6-5.8	Quiz 3	
12	Polling / Interrupts	P		Lab 3
13	Input and Output	P		
14	Text, Audio and Video	P,2.9,3.6,3.7		
15	Q4 / Final Review		Quiz 4	Project
F	Final Exam Week			

Assessment

Professionalism	50 points	5%
Quizzes	160 points 4 @ 40 points each	16%
Labs	165 points 1 @ 15, 3 @ 50 points	16.5%
Project	175 points	17.5%
Mid-Term	200 points	20%
Final	250 points	25%

Quizzes and Exams

- 4 Quizzes
  - 2 before midterm
  - 2 before final
  - Give you a very good feel for how the midterm and final will look.
- 2 Exams (Midterm and Final)
  - Final is **NOT** cumulative!

Course Policies

- Attendance - Yes.
- Tests - In person, in class.
- Available support: http://www.marist.edu/accommodations-accessibility/
- Late work:
  - -10% of grade from +01 second late to 24 hours late
  - -10% for every additional day.
  - After answers are posted, or graded work are handed back to the class, no late work will be accepted for credit (but will be graded somehow, sometime, if you seek feedback).

Office Hours

Email: brian.gormanly@marist.edu

Office Hours: HC 3003

- Monday 6:30 pm - 8:30 pm
- Wednesday 8:00 am - 11 am
- Email if you need to meet outside these hours, send at least 3 times you are available.

So?? Why are we here?

Think about how much changes in the technology world.

If you had been a student in the mid 20th century, the org and arch concepts you learned would still be serving you well today!

It is not an unfair assumption to believe that what you learn in this course will still be serving you well far into the future.

Bringing it together

During most of your CS studies you are learning high level programming languages like C, C++, and Java.

But these languages are geared (mostly) towards how we think, not how the computer computes.

They are translated by a compiler or an interpreter (or even sometimes both, like with Java).

In this course we are going to learn how computers really compute!

We will learn Assembler and machine language instructions (where the bits hit the silicon)

Some things change, others stay the same.

The new tech of today is based on the same foundation of architectural concepts that were developed in the 1940s

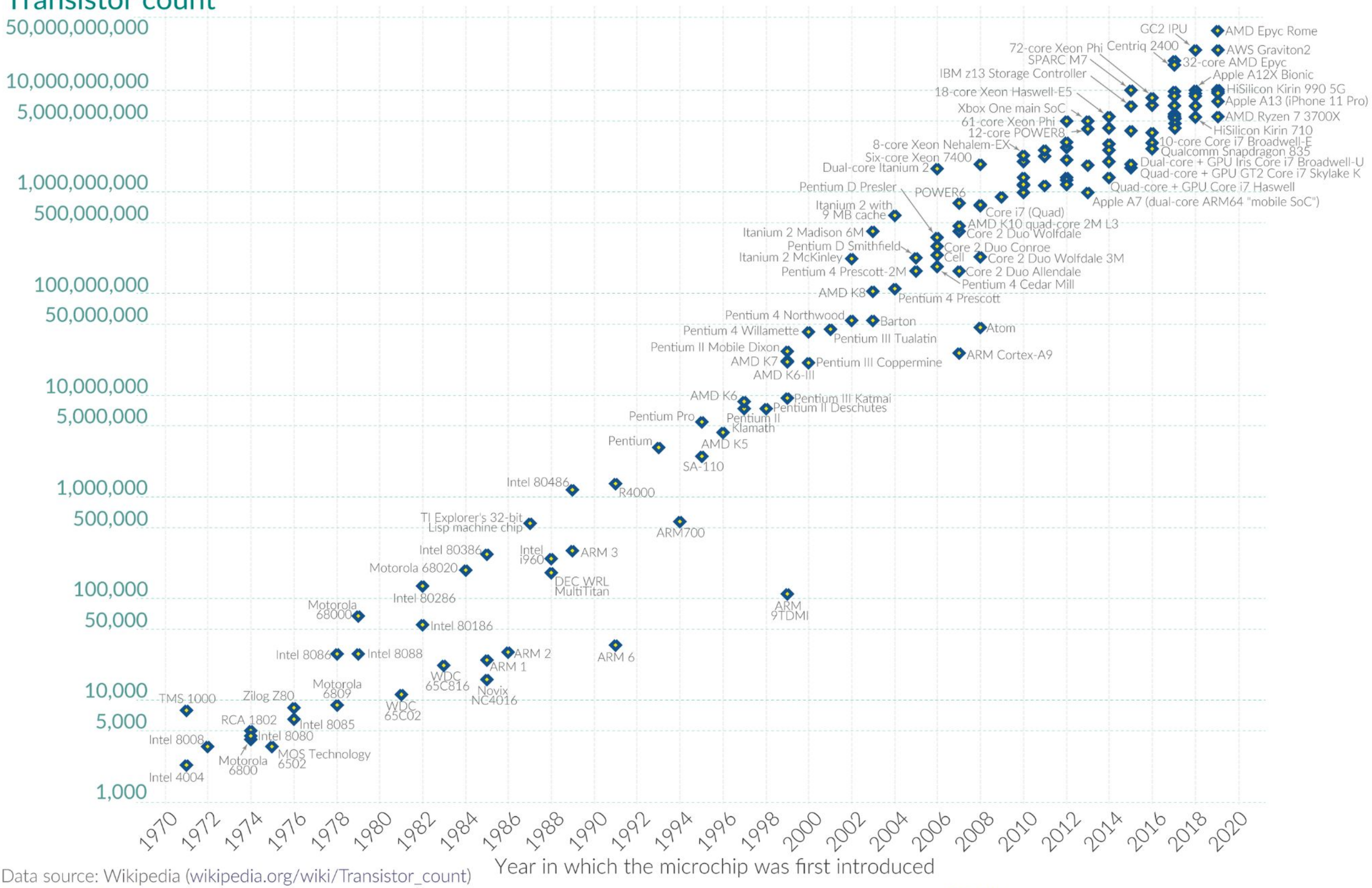
The instruction set in a modern personal computer or smartphone is nearly identical to that of computers built in the 1950s and 1960s.

Who knows what Moore’s Law is?

Moore's Law: The number of transistors on microchips doubles every two years

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



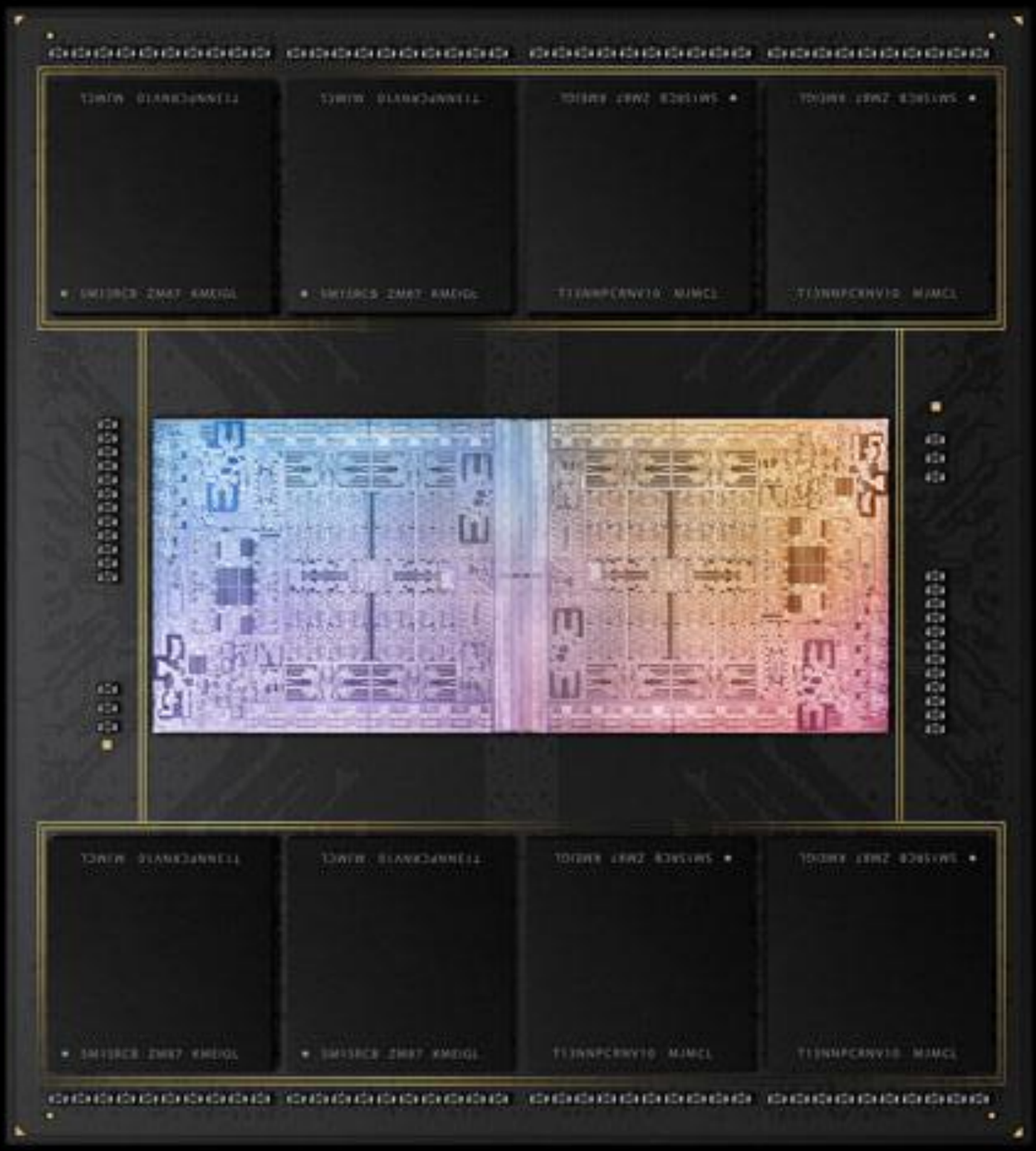
Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

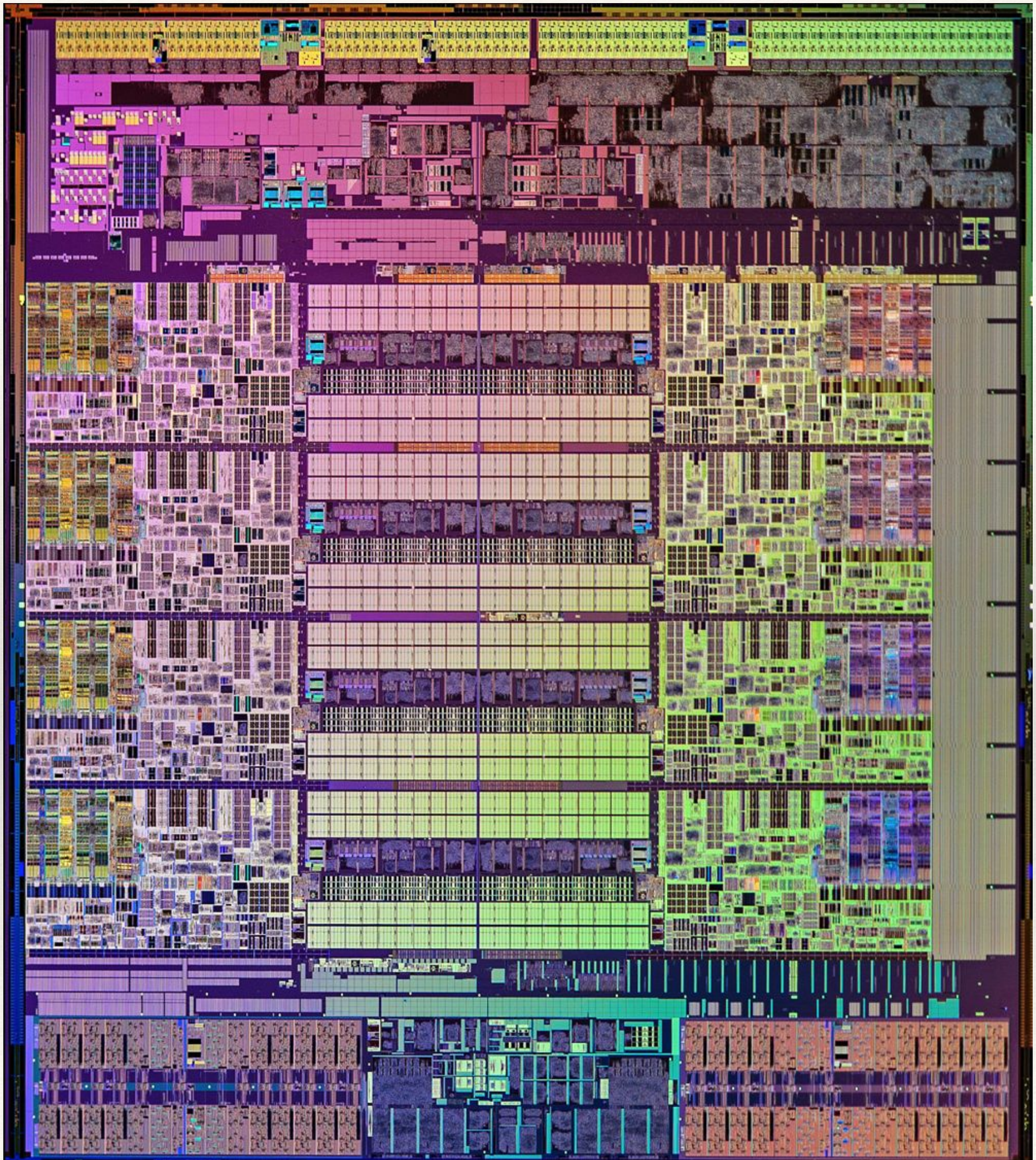
Other Things: Number Systems

```
      1 E 0 R3
    ___________
A ) 1 2 C 3
      A
    ___
      8 C
      8 C
    _____
        0 3
          0
      _____
        R 3
```

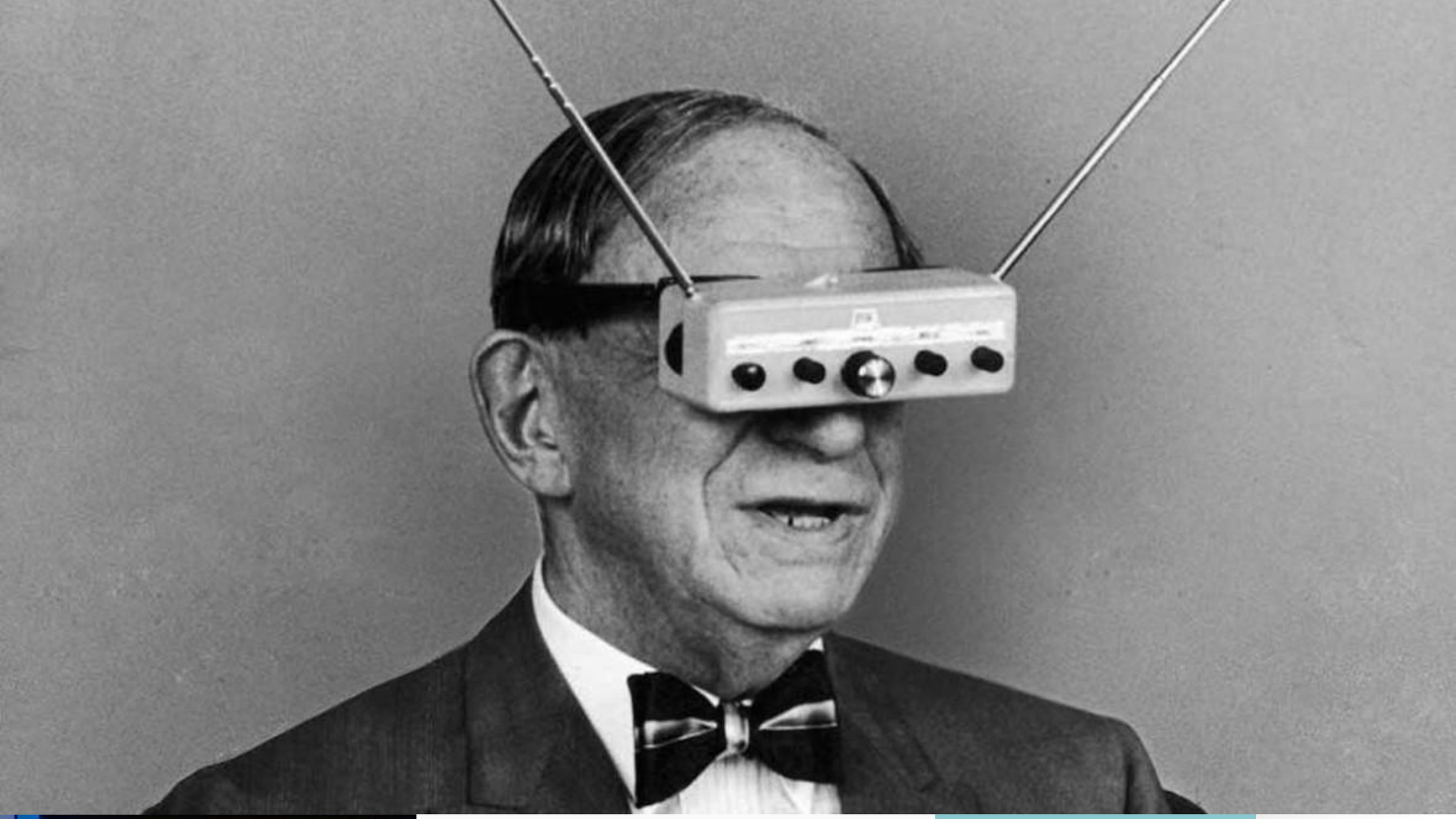
12 / 10 = 1.2 = 1

8C = 140 / 10 = 14(E)
E * A = 140 / 16 = 8 R 12 (C)

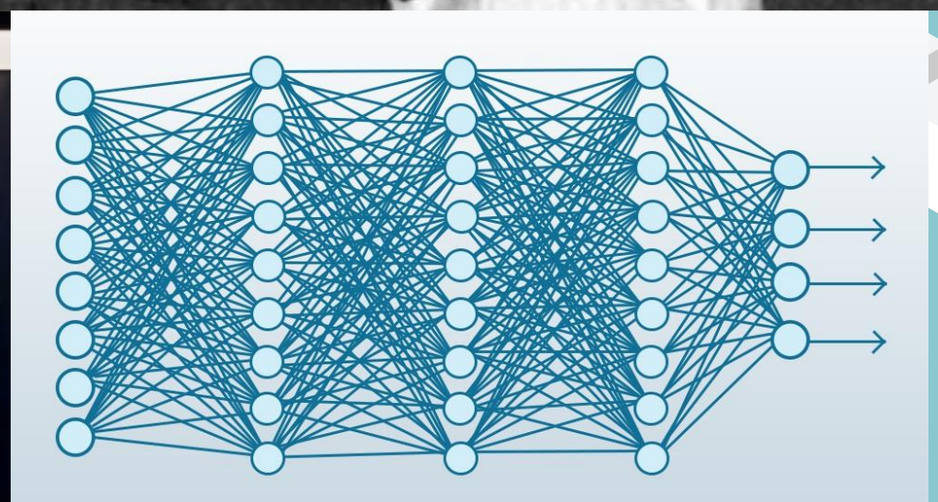


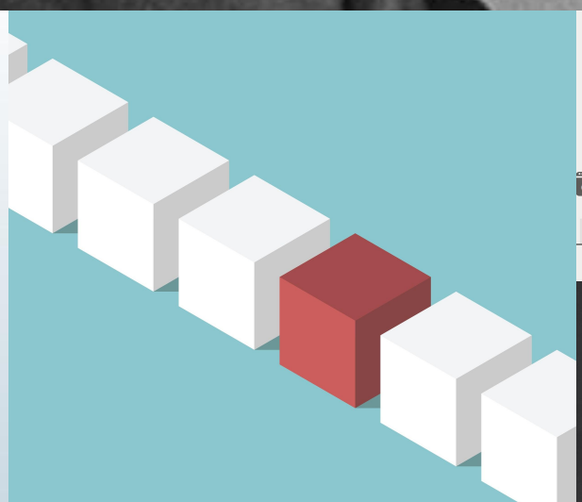


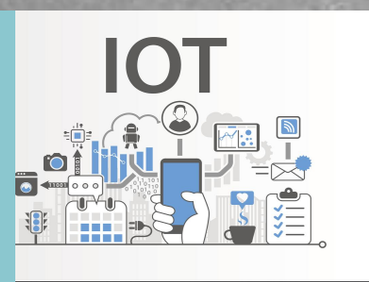
```
section .data                ; create constants here
    msg db "hello, world!", 0xA   ; Create a constant string of bytes (labeled msg)
                        ; the 0xA is a new line character added to the end

section .text      ; code goes here
    global _start  ; global is NASM specific, the linker will read the symbol (_start)
                ; to know where the entry point of the executable is

_start:    ; entry point of the executable

           ; To output text to the screen we are going to make a system
           ; call to the linux operating system (os specific!)
           ; The call we want to make is to sys_write
           ; Calls to sys_write require 3 parameters: a file descriptor,
           ; message to write and the message length
           ; Example call in C:
           ; sys_write(unsigned int fd, const char * buf, size_t count)
           ; To Do this in assembly, we need to load the parameters into
           ; the registers that the information will be expected in, and
           ; make the system call
           ; http://blog.rchapman.org/posts/Linux_System_Call_Table_for_x86_64/

mov     rax, 1 ; load value 1 in rax (register) this is the syscall number
mov     rdi, 1 ; Used to pass 1st argument (file descriptor)
mov     rsi, msg   ; Used to pass 2nd argument (message to write)
mov     rdx, 14; Used to pass 3rd argument (message length) Note: +1 for newline
syscall    ; Make the call to linux (sys_write) triggers software interrupt

           ; Now we are going to make another system call, this time
           ; to exit: sys_exit(int )
mov    rax, 60 ; load value 60 (syscall number) you can also use 0x3C
mov    rdi, 0  ; Used to pass 1st argument (error code)
syscall    ; Make the call to linux (sys_exit)
```




IBM Q
System One






IOT

Project

You will be building a **Virtual 6502** computer

You will be doing this using a really awesome high level language called TypeScript.

You no longer have a life, just Org & Arch.

The Labs are “checkpoints” on the path to completing the project. They largely represent major milestones in your journey to a working system.


commodore 64
POWER
RESTORE
RETURN
CLR HOME
INST DEL


VIDEO COMPUTER SYSTEM™
POWER
TV TYPE
COLOR
B-W
GAME SELECT
GAME RESET
ATARI


6502AD
4585 S


apple IIe
IN USE
disk II

Project

You will be working in TypeScript, because there is no such thing as a loosely typed virtual machine.

This project takes inspiration from Dr. Labouseur's GLaDOS project used in our Operating Systems class, but it is different in many ways.

You will be required to learn new skills on the fly. This project is designed to push your skills as a software engineer to heights all while using the act of creating a CPU to enforce understanding of how it works!

Labs

The labs are intended to reinforce particular topics as we discuss them in class by letting you apply those skills as we are discussing them.

They include:

- Git / Environment Setup
- Project Buildout
- Programming in Machine Instructions!
- Building a Virtual CPU

All labs are required for your project! (you cannot pass the project if you do not finish all labs!)

TypeScript Resources

- TypeScript Resources / Links
  - TypeScript
  - TypeScript Documentation
  - TypeScript tag on StackOverflow
- TypeScript Videos
  - What is TypeScript
  - Inside TypeScript
  - TypeScript Compiler Explained

Discussion

https://docs.google.com/presentation/d/14eR5ZXWRJG2qpYqwEmp4OdUwaEPbg4wcCe0DyPFfEVU/edit?usp=sharing

AI assist class rules

1. Any use of code that is not 100% written by you, if allowed for the assignment you are working on must be cited.
2. If you use any AI assist, include how you used it (use good commenting practices to explain in your code!) I want to hear about your experiences the good and the bad

Additional Course Resources

1. https://www.youtube.com/c/CodingCoach COVID and Hybrid classes have me making videos, so I intend to make the best of it. In addition to finding this classes videos there, you will most likely find content in the future as well which may prove useful.
2. Get acquainted with our Brightspace layout
3. ***Start learning about TypeScript and JavaScript***

Current Research and Open Source Work

I think the best part about being in technology and computer science in particular is much exciting change is on the horizon.

Some of my current research interests…

Agora

The Question:

When Microsoft stated the following during the height of the fervor and excitement surrounding ChatGPT:

"AI will fundamentally change every software category, starting with the largest category of all – search"[1]

This prompted a question.

Can LLM APIs be used to help researchers and academics find and evaluate literature as they write? Is there value to integrating such a feature into editors?

1: Source: https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/

agora Public

Unpin | Unwatch 6 | Fork 1 | Starred 65

main | 10 branches | 1 tag | Go to file | Add file | Code

briangormanly Merge pull request #401 fro... 1,033 commits

.github/workflows	i think this will fix the action	8 months ago
.vscode	fixe... exp...	6 months ago
client	Fixed linting issue	6 days ago
server	em... depev bug fix for creating a new user without sso	4 months ago
.env.example	added o...	4 months ago
.eslintignore	all b... 8	8 months ago
.eslintrc.json	ad...	8 months ago
.gitignore	removed ds-Store	last year
LICENSE	up... t... ...style.css and all bootstrap fr...	6 months ago
README.md	new... go and side menu adjustments	5 months ago
nodemon.json	switched back to openapi def and yaml	last year
package.json	added openai to .env	4 months ago

README.md

Agora

Star the repo to find it faster, and follow me on github!

About

An Open source, closed loop, user centric, cloud based learning and research platform

freeagora.org/

education | latex | research | notes | lms | learning-management-system | online-learning

Readme
BSD-3-Clause license
Activity
65 stars
6 watching
1 fork

Releases

1 tags

Create a new release

Packages

No packages published
Publish your first package

Contributors 19

+ 8 contributors

Languages

JavaScript 61.6% | EJS 32.4% | CSS 6.0%

Fabrication or Fact

Description: Testing the Viability of Li **Tags:** Start typing... research | generative ai | inform retriev

Research | 1st Draft

Research

Untitled

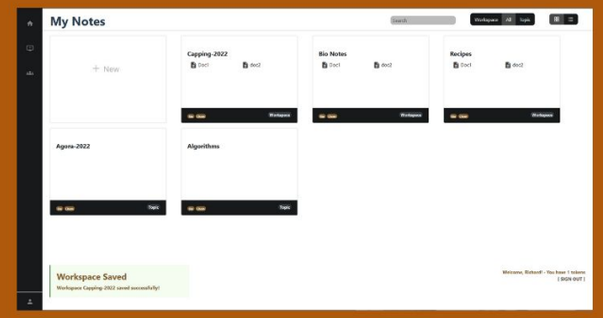
"Helvetica ... | 15px | Formats

Keywords: Generative Artificial Intelligence, Information Retrieval, Research Search, ChatGPT, Bard

Outline:

SIGN IN DASHBOARD ABOUT

to take notes, expectations



Agora for your notes

Notetaking, researching and learning,
For everyone.
Privacy guaranteed.

Agora creates a secure, private, online environment focused or and ownership of data within the system with infrastructure a managed and protected by the Agora Foundat

E YOUR ACCOUNT TODAY!

Workspace | Workspace

Planning 2
Testing a new Workspace for the bug
Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | week 9
planning | courses | Workspace

GR86 ToDo List
Collection of notes and todos for when the car arrives
Arrival to d | CANBus
gr86 | auto | Workspace

Agora
Agora Development Planning
Spring 23 De | {Meta} Docum | Design
agora | programming | documents | planning | Workspace

Algorithms
Course material for CMPT 435 Algorithms
General | Labs | Lecture Note
cmpt435 | algorithms | course | marist | Workspace

Fabrication or Fac
Testing the Viability of Literature Suggestions and Quality Evaluation Provided by LLM Integrations
Research | 1st Draft
research | generative ai | information retrieval | search | chatgpt | bard | Workspace

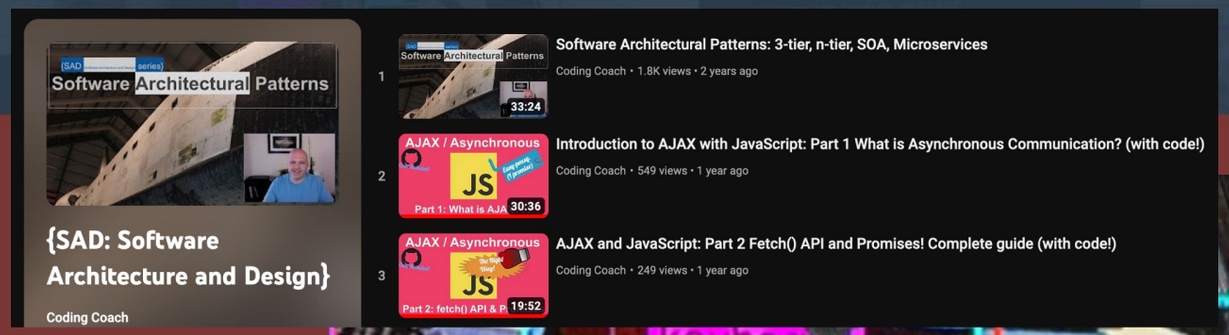
Models Evaluated

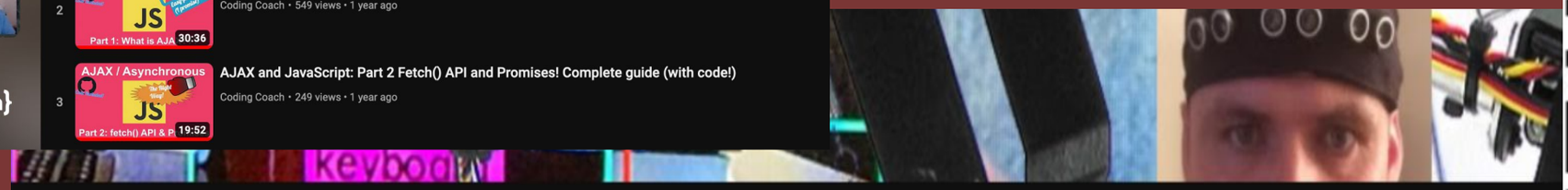
We investigated accuracy using the following models:

- GTP-3-text-davinci-003 (API: platform.openai.com)
- GPT-3.5-turbo
  - API: platform.openai.com
  - UI: chat.openai.com
- GPT-4 (multiple flavors)
  - GPT-4 (UI: chat.openai.com - Premium paid access) Data to September 2021
  - GPT-4 **with Bing** (UI: chat.openai.com - Premium paid access) Data to present
  - Bing Chat [**Prometheus** [1]] (UI - Product: www.bing.com - Public access)
- Google Bard (UI - Early access to pre-release product: bard.google.com)

Not the Same!

1: Source: Ribas, Jordi. Building the New Bing 2/21/2023. https://www.linkedin.com/pulse/building-new-bing-jordi-ribas.





Coding Coach

@CodingCoach 4.03K subscribers 105 videos

The Coding Coach is for anyone interested in learning Computer Science! ...

HOME VIDEOS PLAYLISTS COMMUNITY CHANNELS ABOUT

Popular videos ▶ Play all

Apples M1 Processor: The hardware behind the hype

45K views • 2 years ago

Robotics Programming: Building an Autonomous...

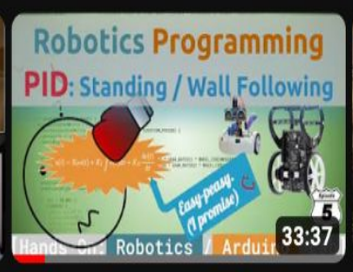
30K views • 2 years ago

Robotics Programming: Ultrasonic Sensor | Object...

10K views • 2 years ago

4.1 Database Decomposition Example: Step by step...

9.2K views • 2 years ago

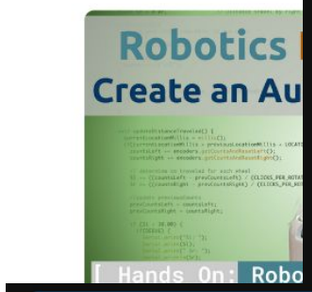


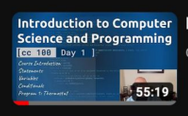
Robotics Programming: PID Algorithm - Wall Following...

8.2K views • 1 year ago

Database Connections and Connection Pooling

7.2K views • 2 years ago





1 Day 1: Introduction to Computer Science and Programming - Learn to Code!

Coding Coach • 1.8K views • 2 years ago

2 Binary Numbers: The Complete Introduction

Coding Coach • 391 views • 2 years ago

Robotics Programming: PID

AJAX and JavaScript: Part 2

Introduction to AJAX with

Robotics Programming:

Robotics Programming:

Robotics Programming:

MapCushion

- Dashboard
- Profile
- Maps
  - James I. O'Neill HS
    - 2nd Floor
    - 3rd Floor
    - 1st Floor
- Beacons
- Users
- Client Admin

Management Console

New Visitor

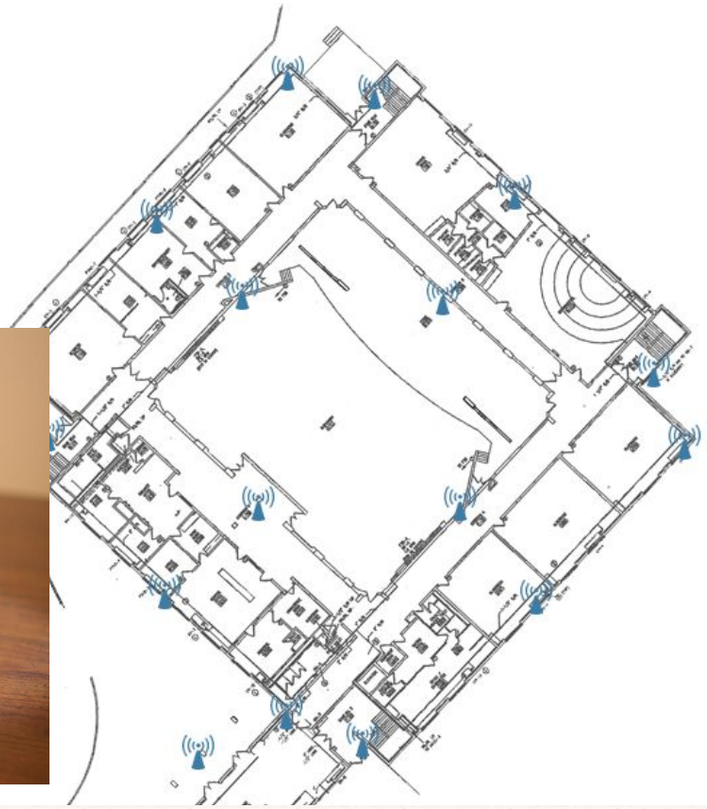
Name	Street Address	City	State
BRIAN,E GORMANLY	110 OAK RIDGE RD	HOPEWELL JCT	NY

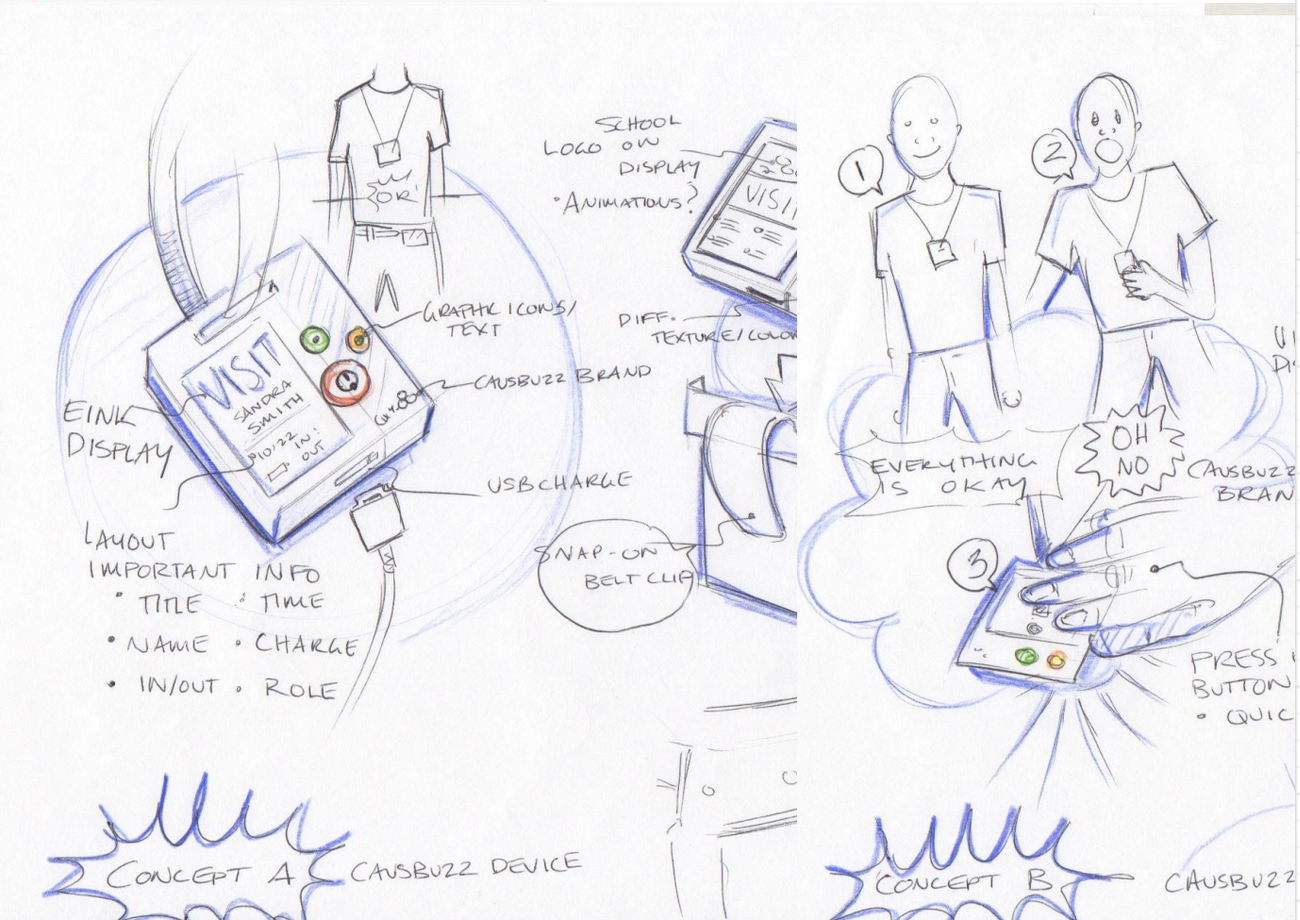
James I. O'Neill H

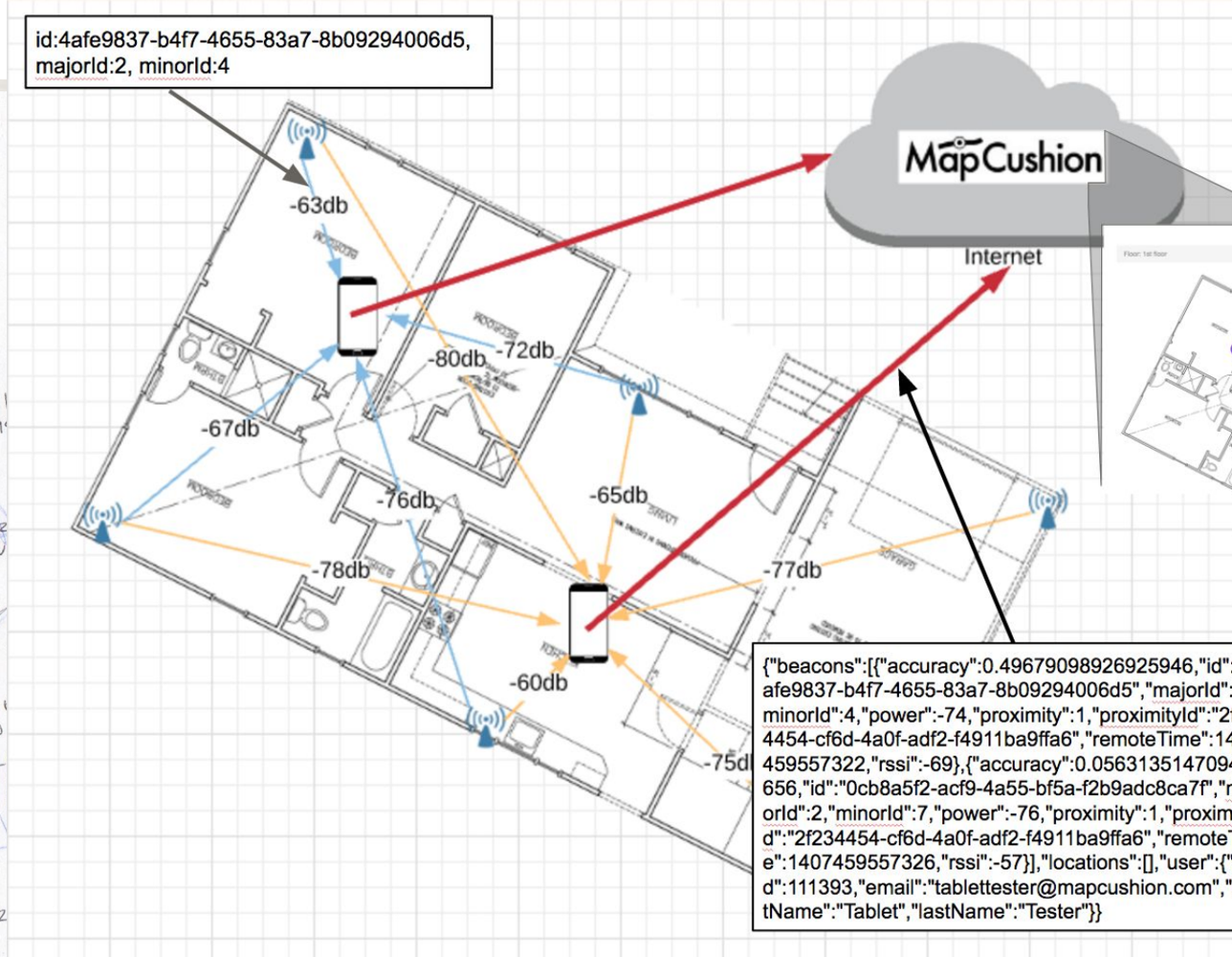
Jar

View Map for James I. O'Neill HS

Brian
beacon data
user data

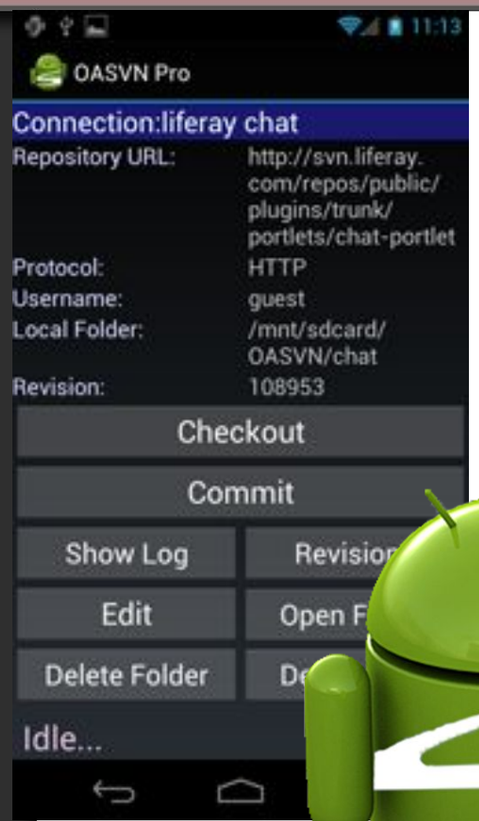






MIP Connector

Home
Logout
Collective Strategies
Administrator User
API User
SalesForce Configuration
Amplifund Configuration
MIP Configuration
Demo Tenant
Administrator User
API User
SalesForce Configuration
Amplifund Configuration
MIP Configuration
API Testing Tenant
Administrator User
API User
SalesForce Configuration
Amplifund Configuration
MIP Configuration
Demo Tenant
Administrator User
API User



API
Last Name:

API
Username:

New Password: [current encrypted, not shown]

github.com/briangormanly/4dflib

4DFLib

4DF (4th Dimensional Form) Library and ORM Tool

Author: Brian Gormanly bgormanly@gmail.com 4dflib.com Copyright © 2015-2017

0. Introduction

What is 4DF????
4DF is a Library that manages your applications interaction with the database, providing ORM, database abstraction and 4DF or 4th Dimensional Form data which means that all data is saved in every state in every table over time. Nothing is ever deleted or updated, and you do not need to anything to implement this behavior, it happens automatically. It also provides a basic service layer for accessing your data through time, allowing you to retrieve current and historical data and filtering by time ranges.

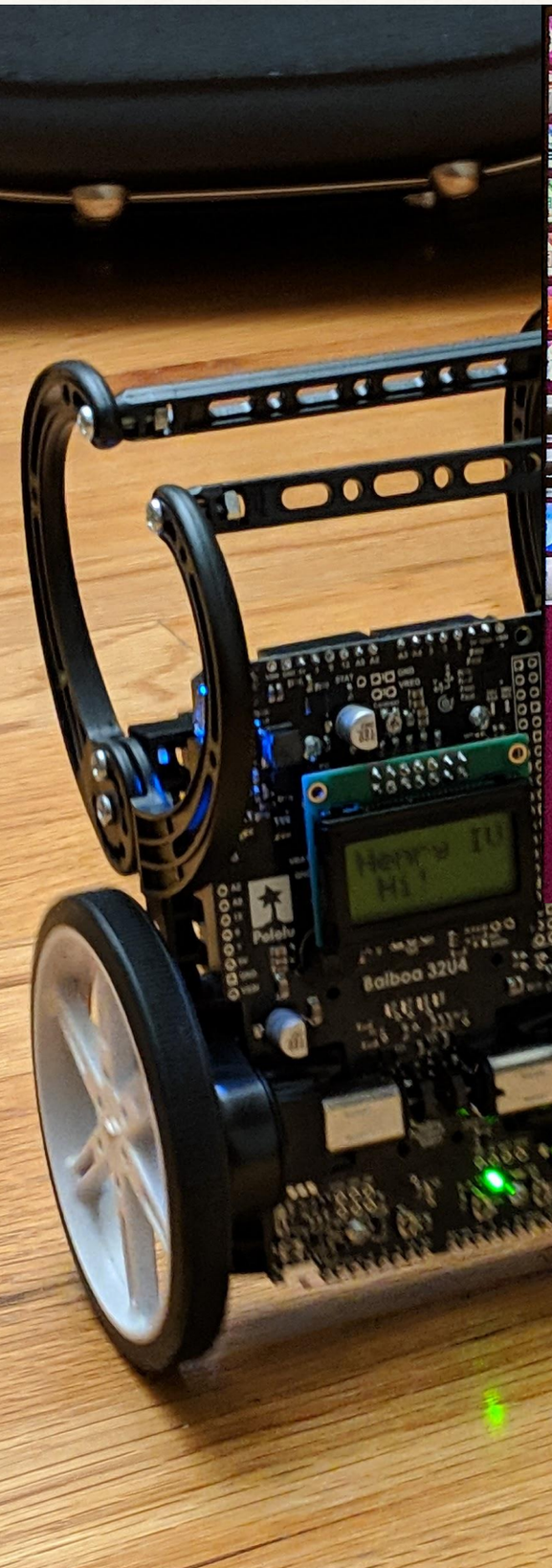
4DFLib currently works with HyperSQL (HSQLDB), PostgreSQL and MySQL, but many more are planned soon and you can easily implement your own database see: com.fdflib.persistence.impl.CorePersistenceImpl; and example connections in: com.fdflib.persistence.database and dynamic queries builders in: com.fdflib.persistence.queries

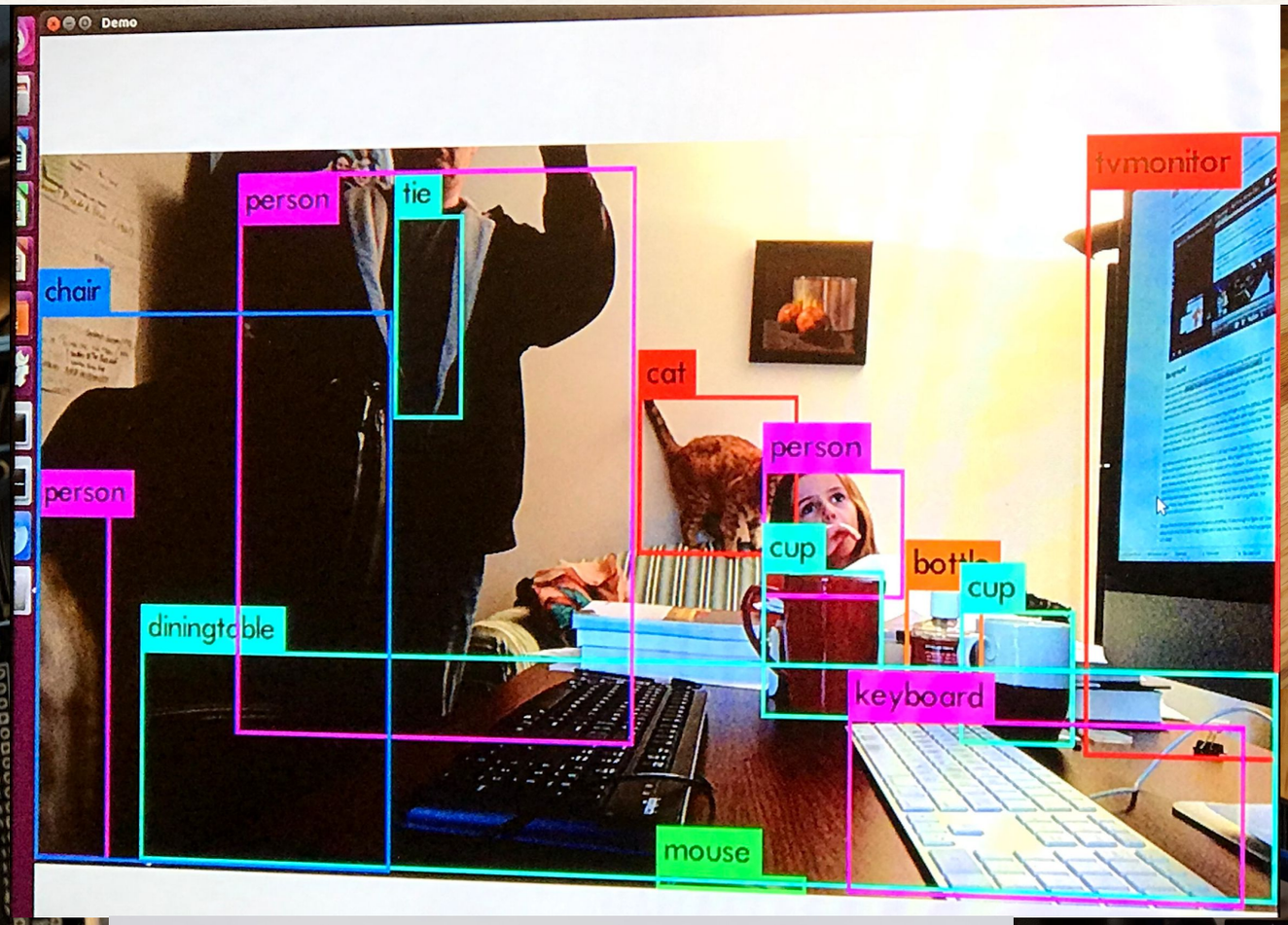
```
We welcome pull requests with new database implementations.  4DFLib can support connecting to both relatic
NoSQL persistence.
```

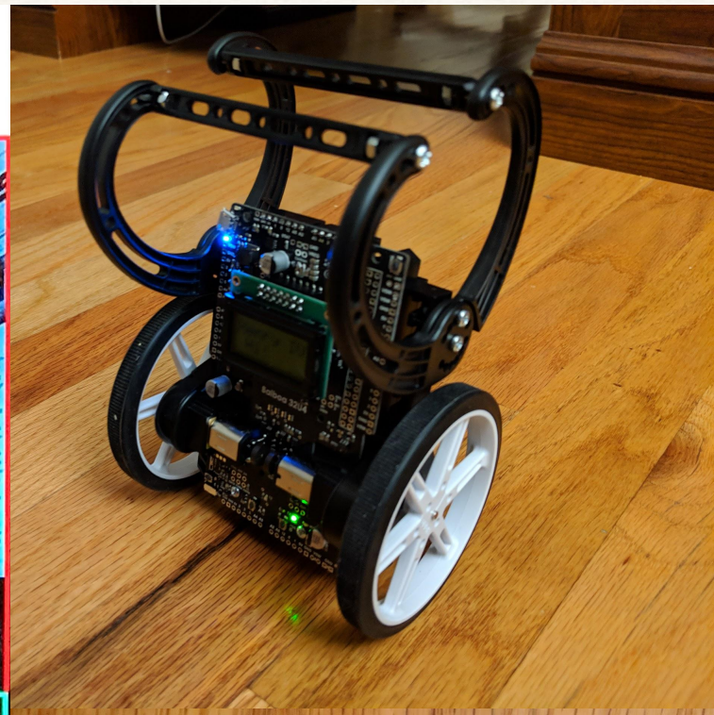
iECHO Online Faith Formation
Registration
General
School

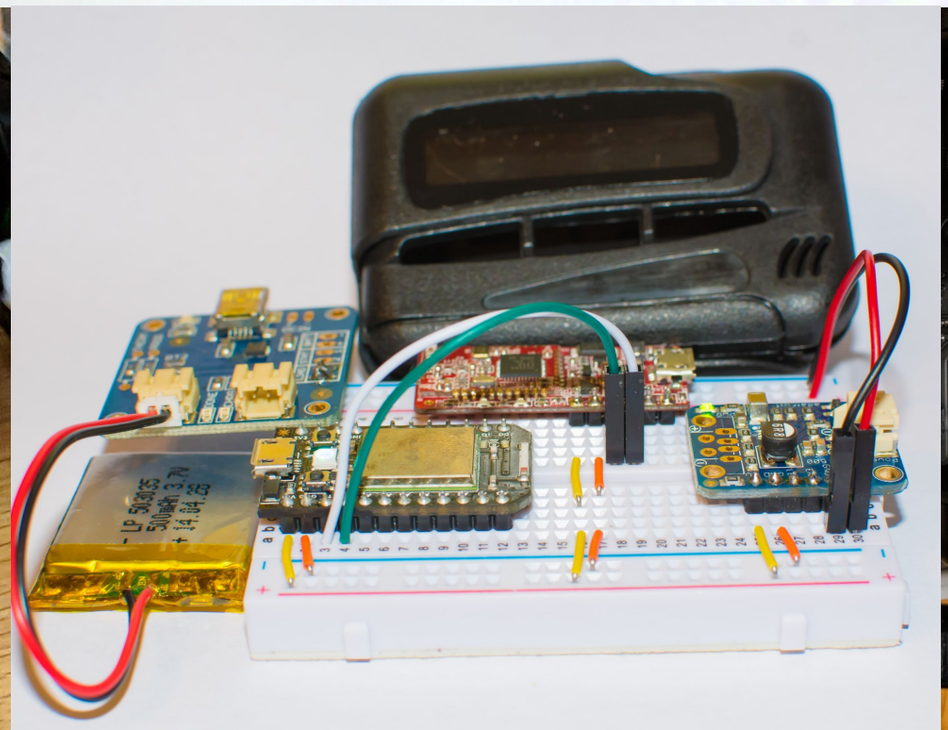
Find a Student
Search for...
Caitlin Waters S761 Select
caitlin waters S762 Select
Jamar Cummings S764 Select
Nicole Lamorte S765 Select
Tony DiMarco S766 Select
Cody Scalzo S767 Select
Kristen Lake S757 Select
Zachary Revella Select
Can't find what you're looking for? Add new here

Other Projects









Open Source and Other Projects

Github: https://github.com/briangormanly

Agora: https://github.com/briangormanly/agora

4DFLib: https://github.com/briangormanly/4dflib

OASVN: https://github.com/briangormanly/oasvn

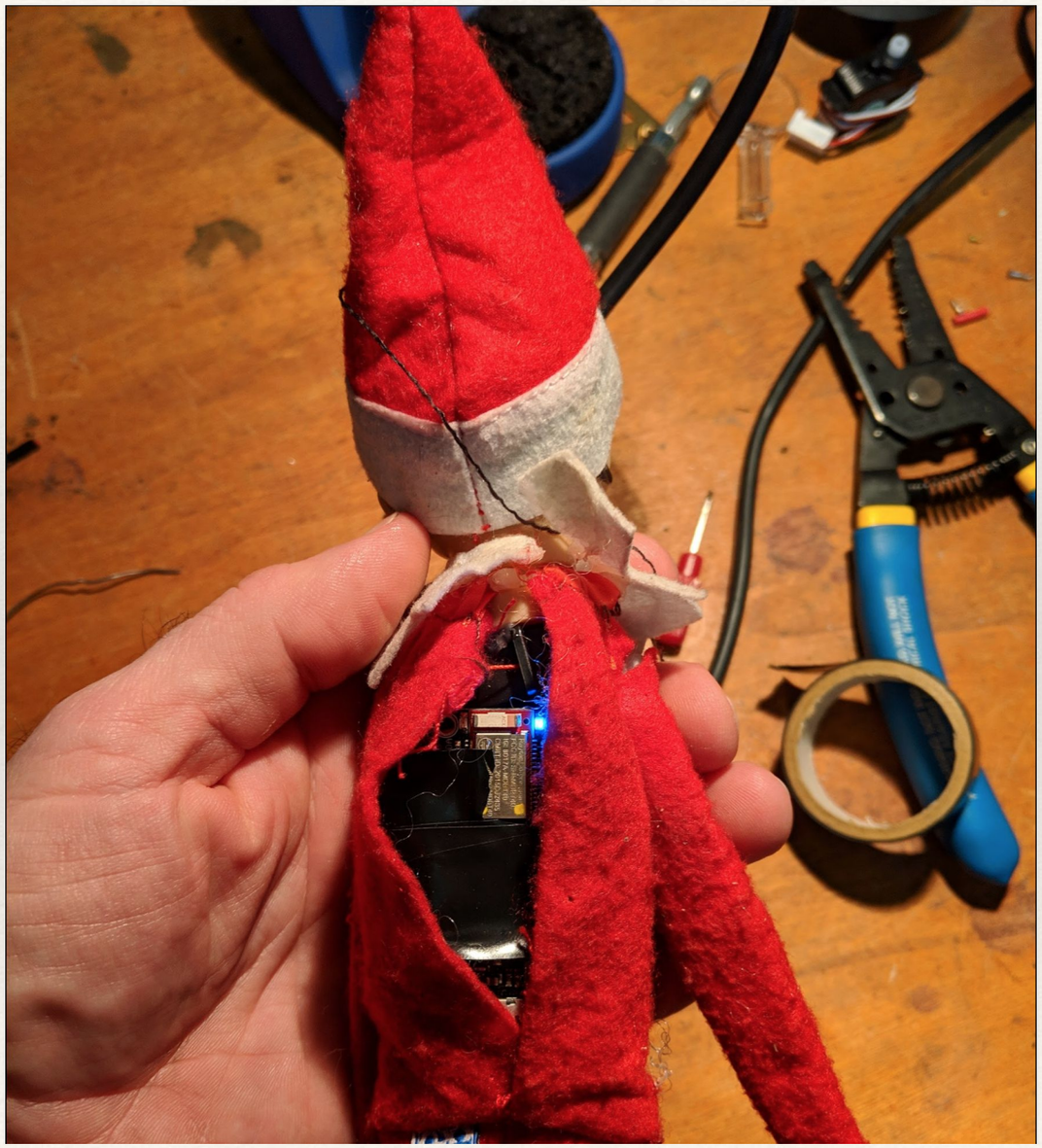
Outfox: https://github.com/briangormanly/outfox

TSIRAM a virtual 6502 computer: https://tsiram6502.abiggeek.com/

Coding Coach: https://www.youtube.com/c/codingcoach

Build a Better World!

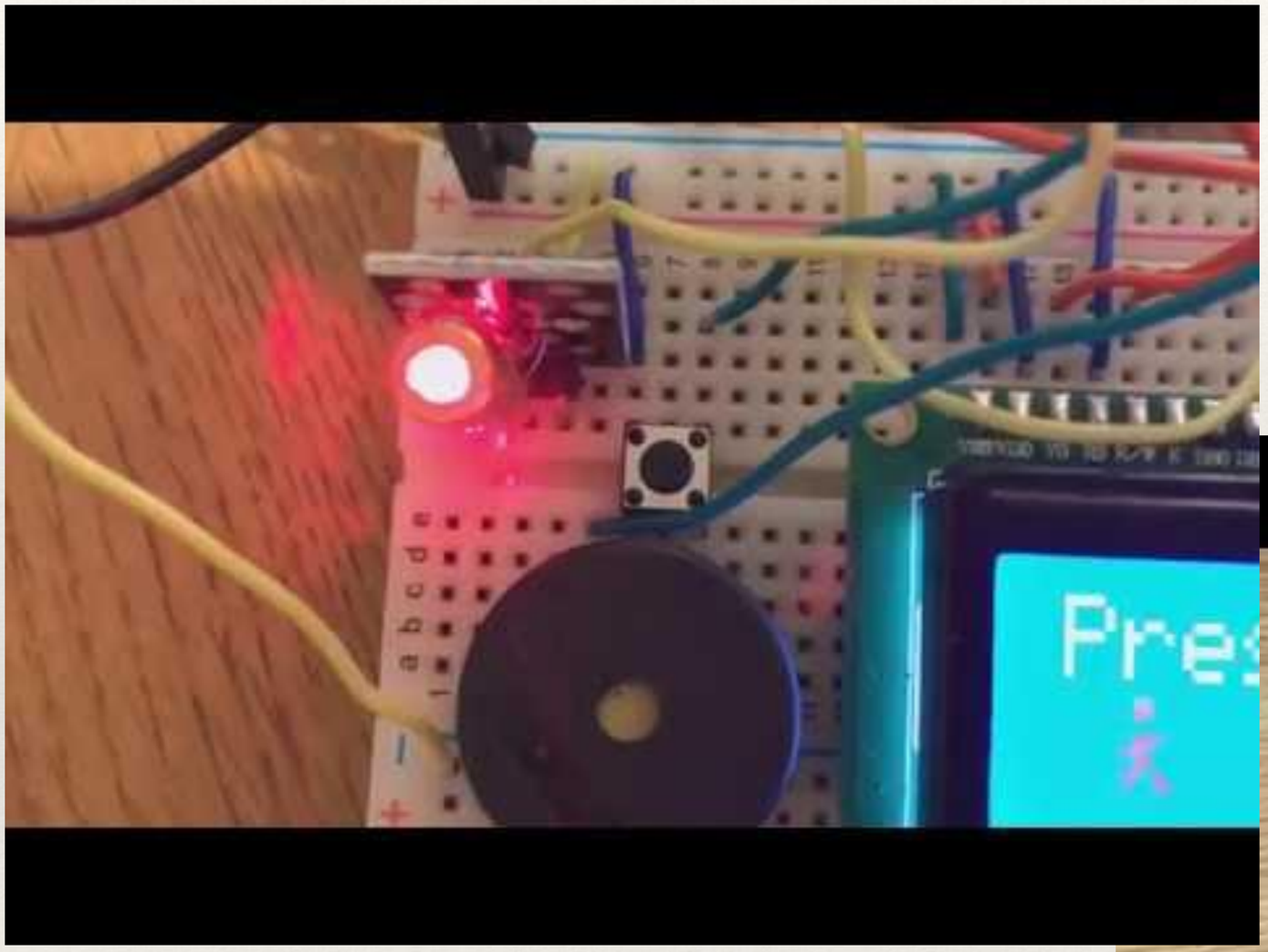
YouTube channel: https://www.youtube.com/c/CodingCoach

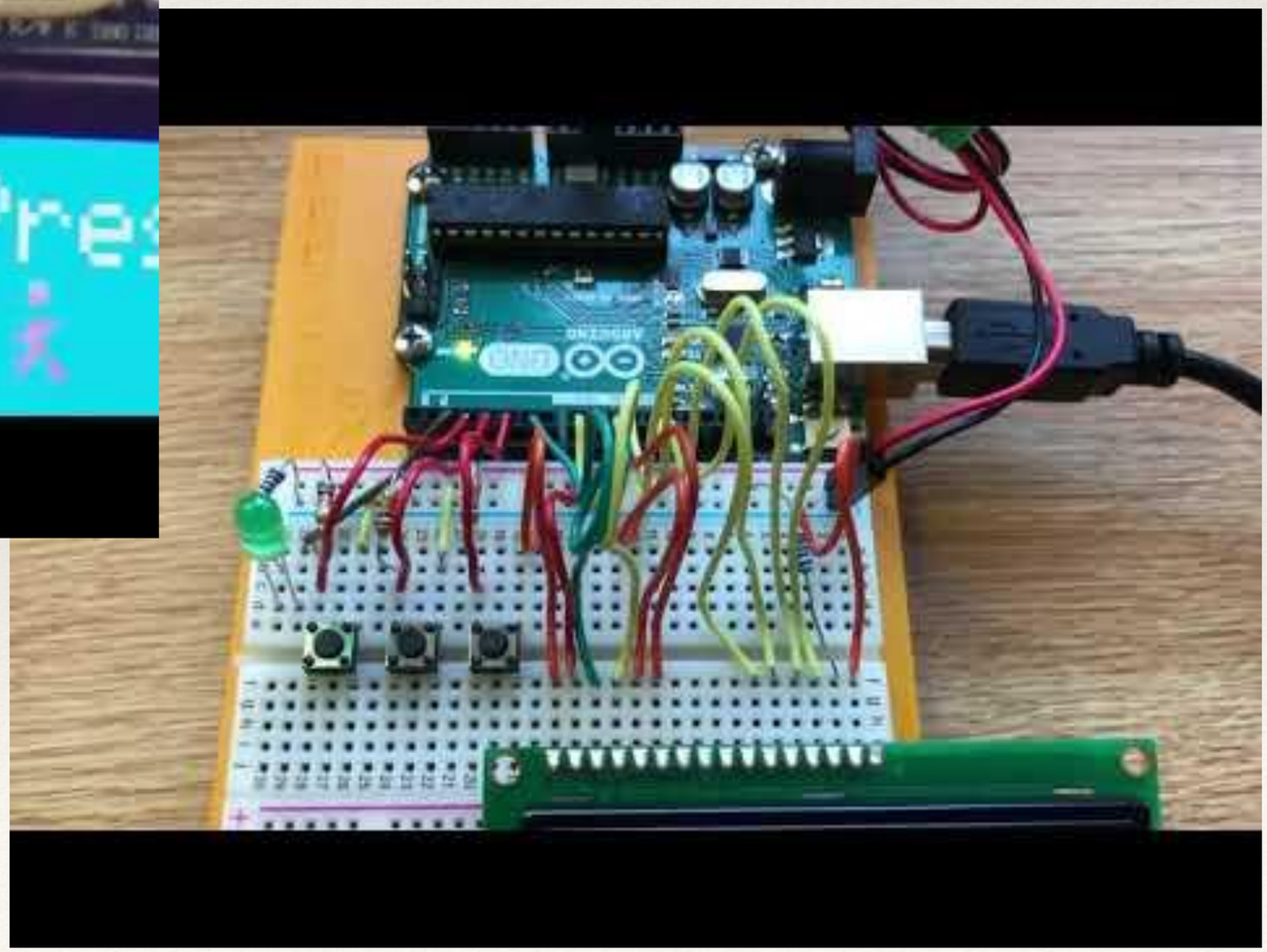


Connect with me | Linkedin: https://www.linkedin.com/in/gormanly/

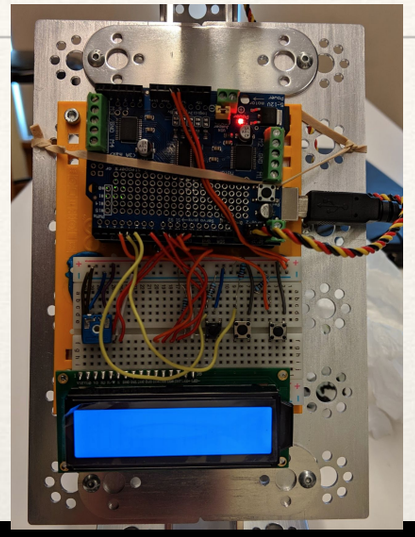
Questions?

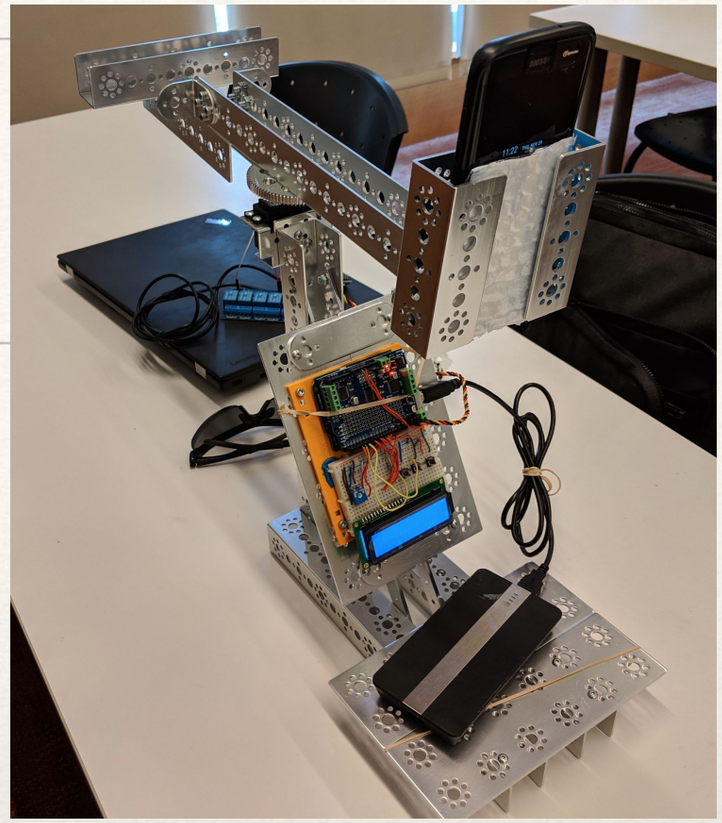
Project Hall of Fame

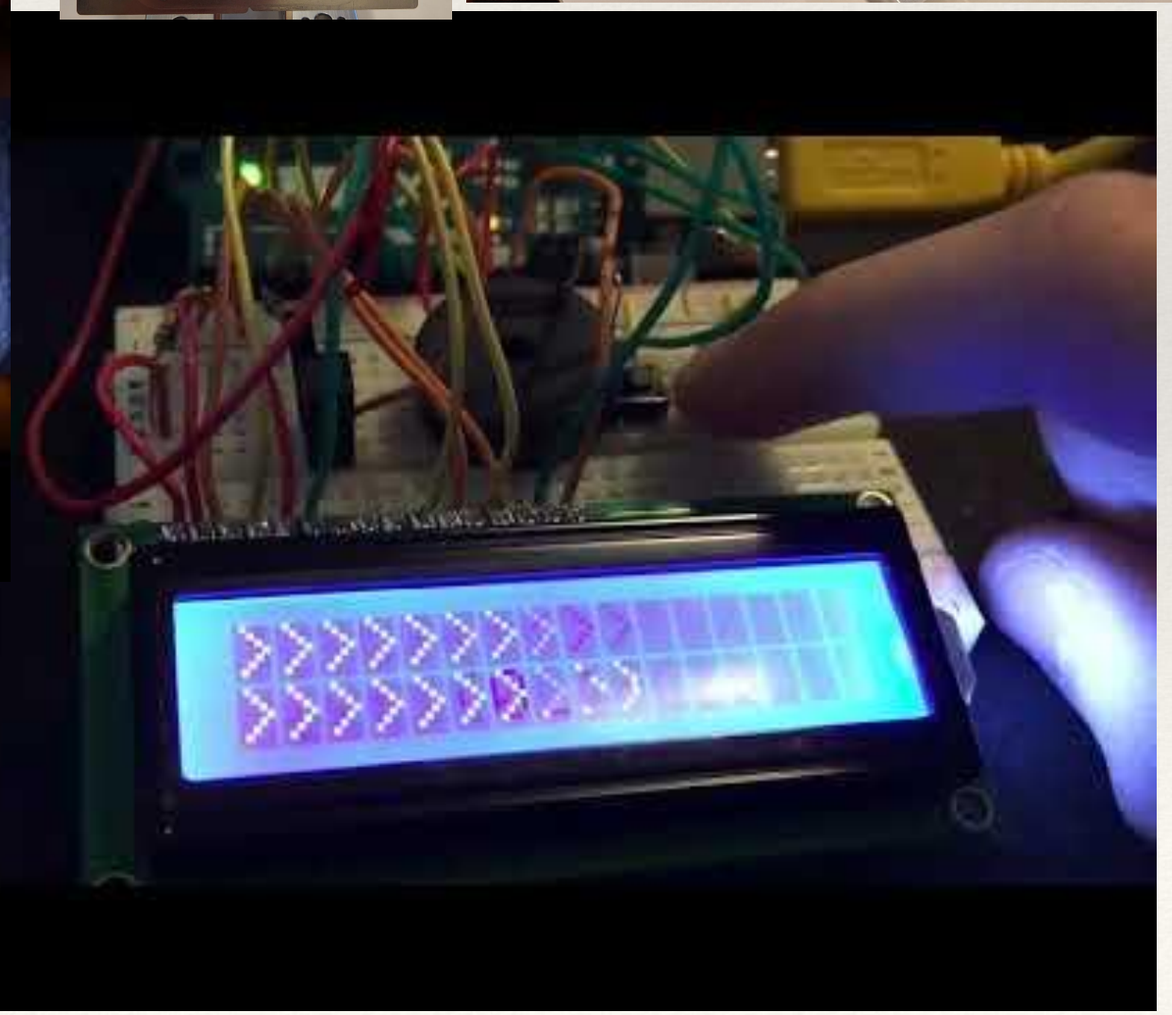


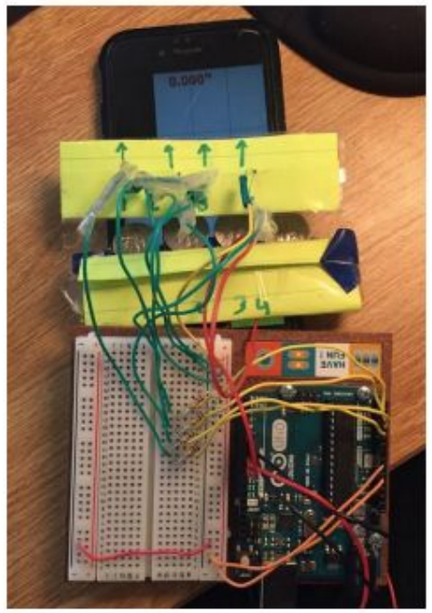


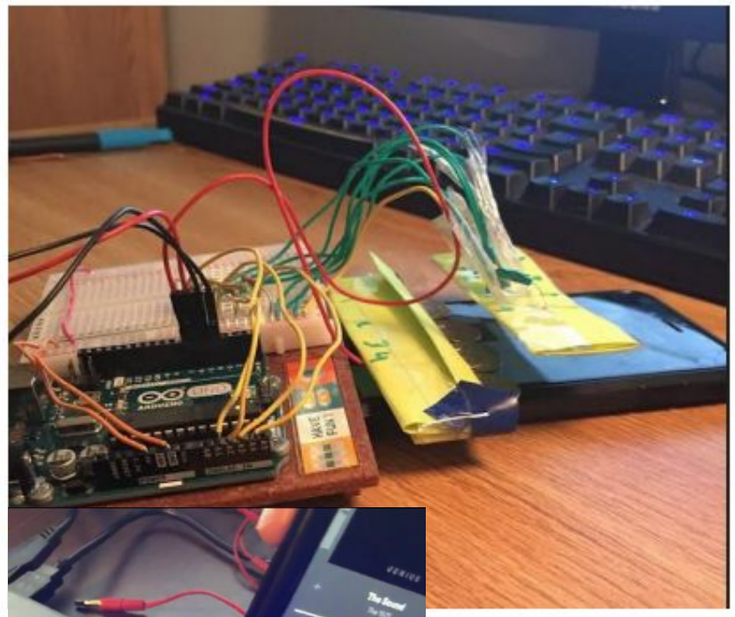
Project Hall of Fame

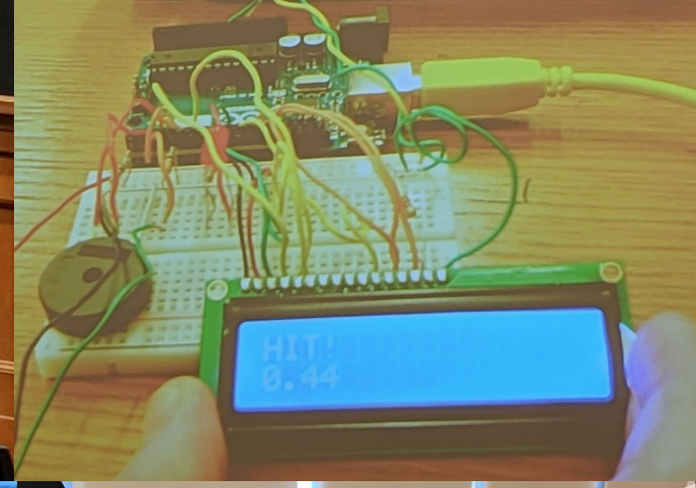




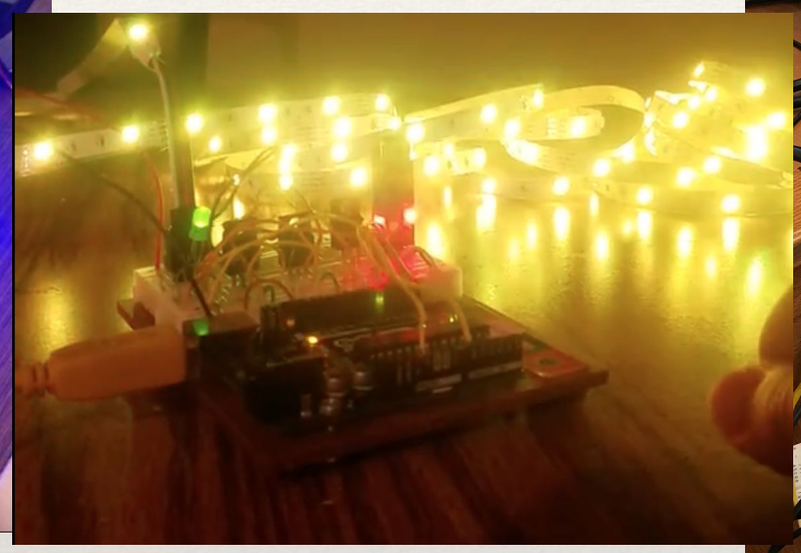


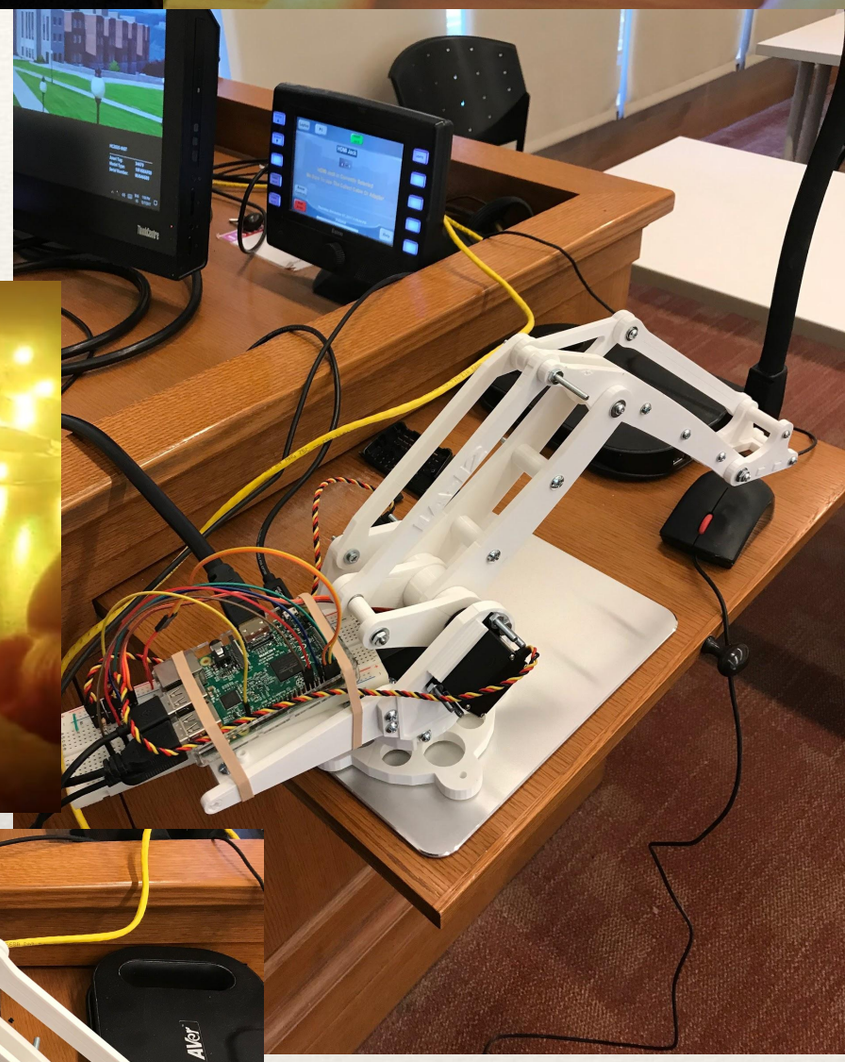


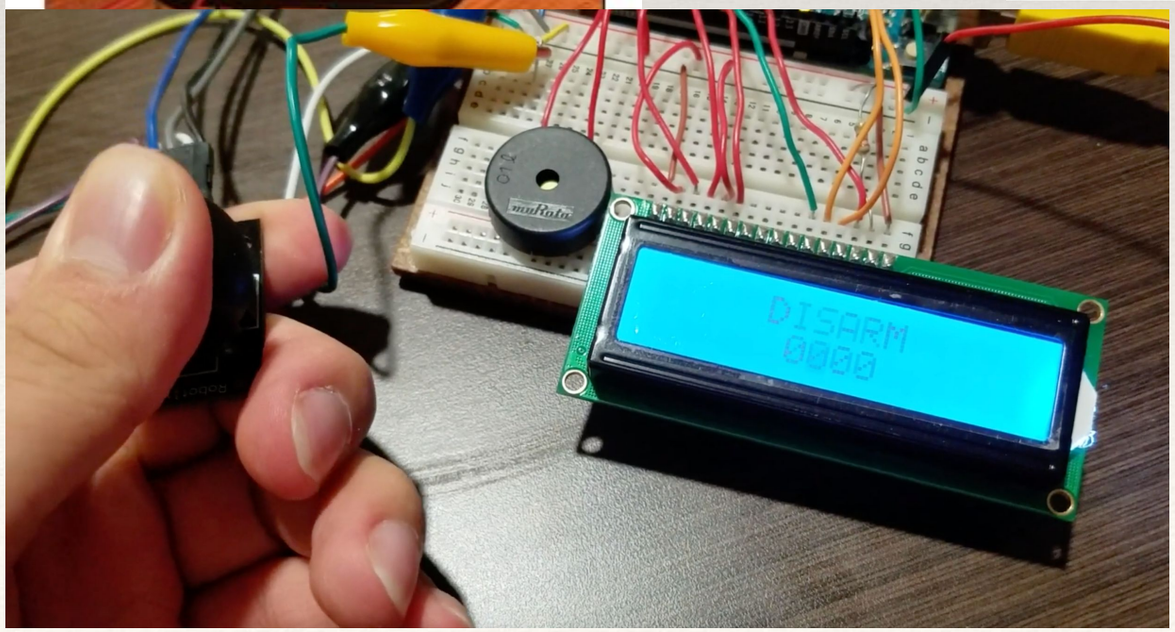


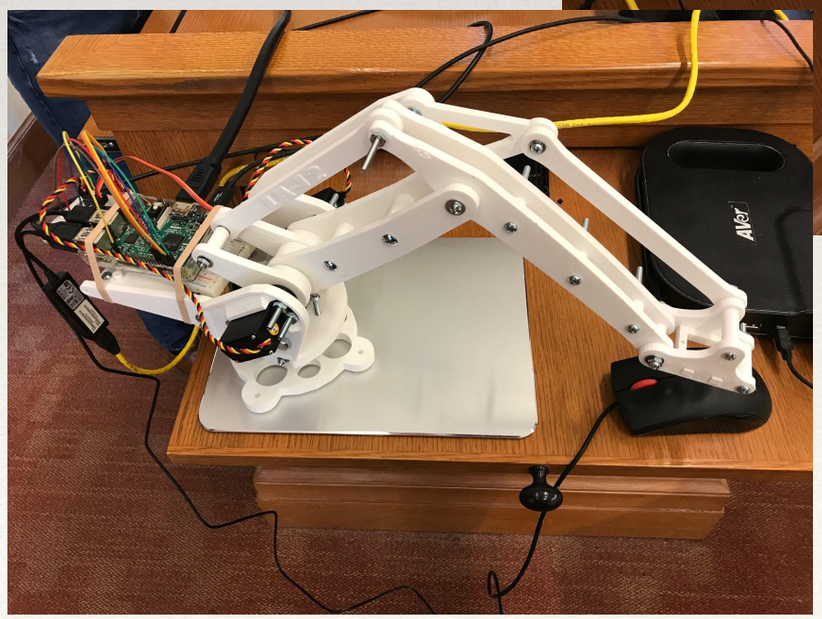



0.000"

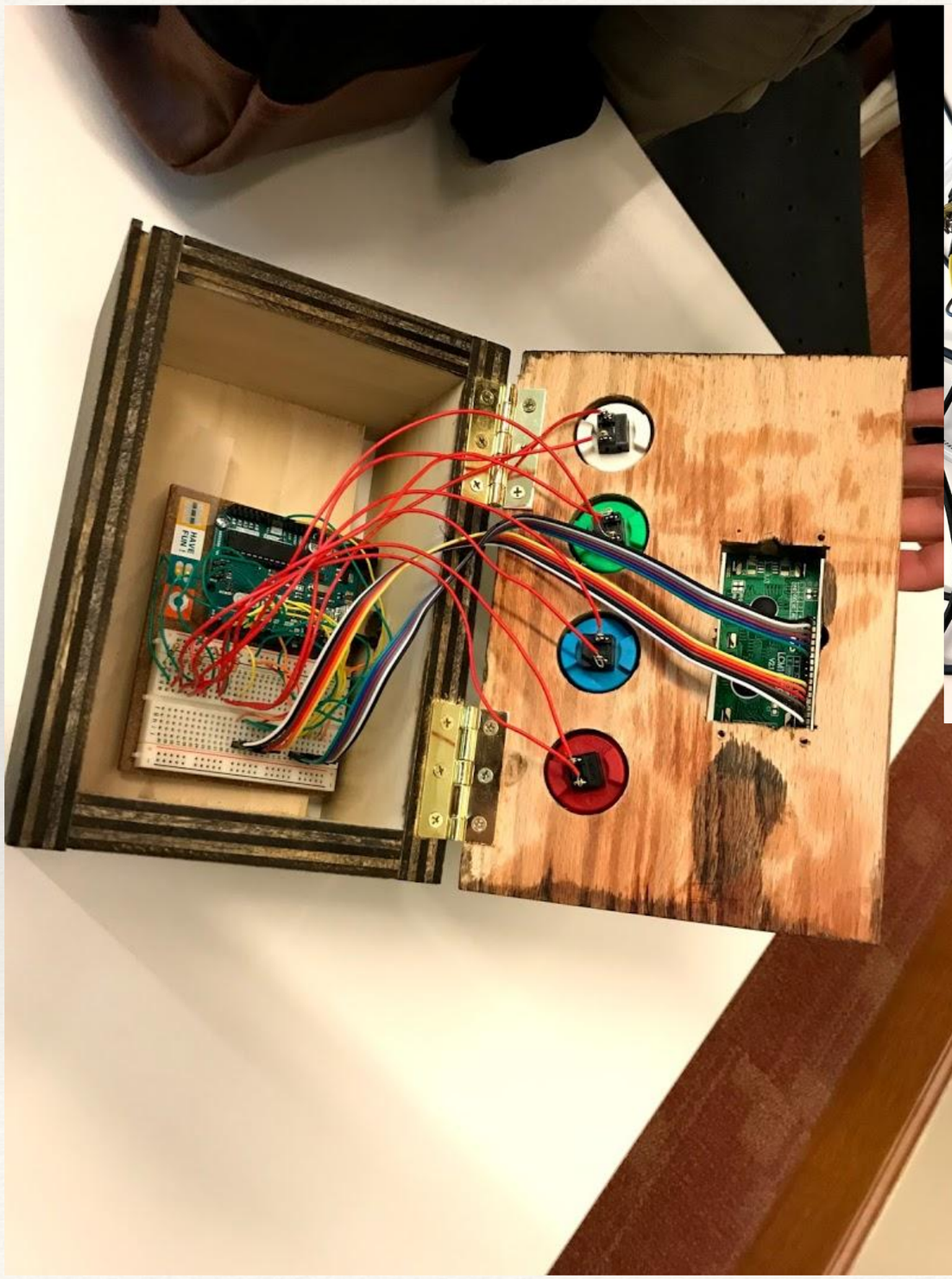


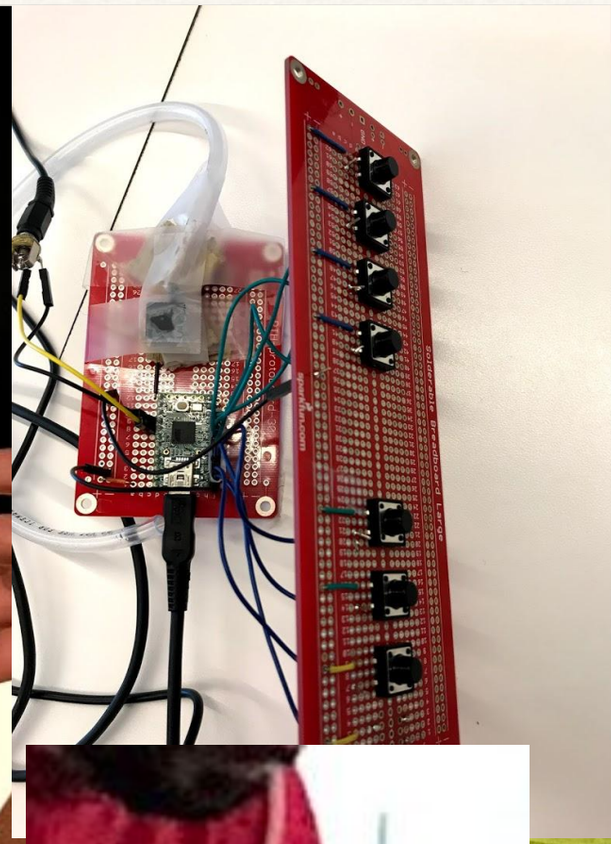


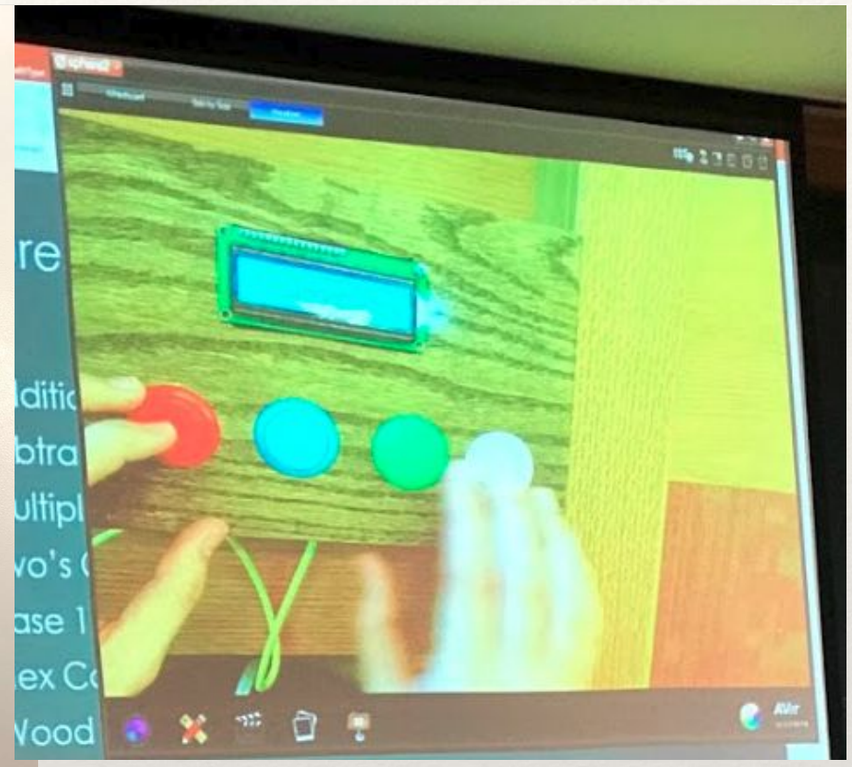

DISARM
0000

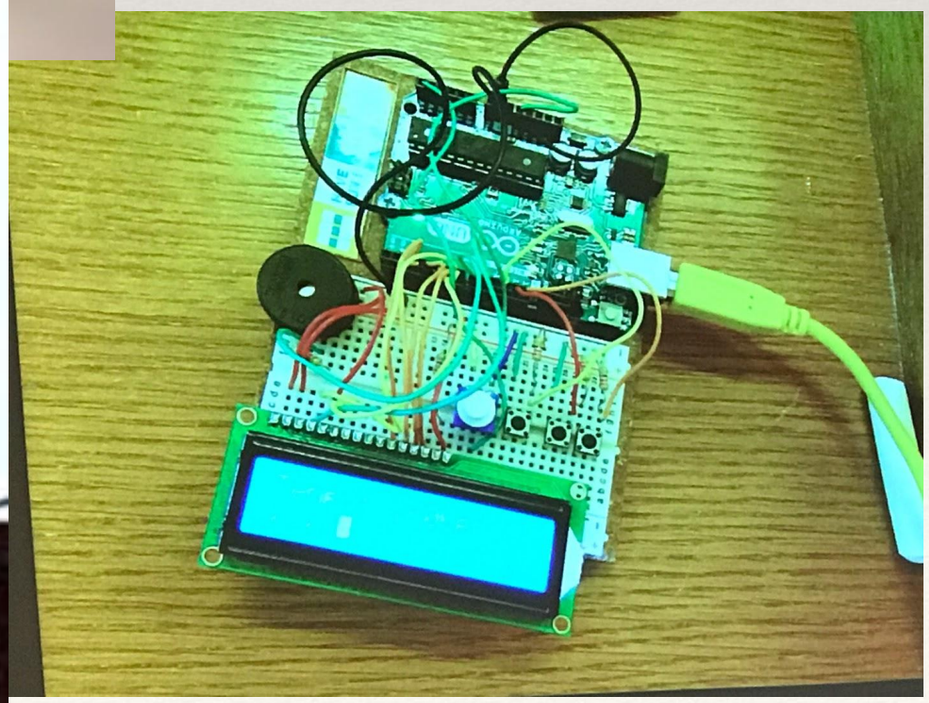


Project Hall of Fame









Project Hall of Fame