CS107E CURIS 2025

Elias Chikwanda, Daniel James, Joe Robertazzi

CS107E: RESEARCHING COMPUTER SYSTEMS FROM THE GROUND UP

Stanford University

Elias Chikwanda, Daniel James, and Joe Robertazzi

Mentor – Julie Zelenski

OVERVIEW

This summer we aimed to "hack" on the Mango Pi to **develop its libraries** and capabilities for students to use in the CS107E course

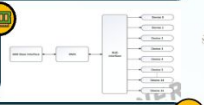
We **established APIs** for MangoPi serial communication protocols, allowing students to pursue more ambitious projects.

DRIVERS

SPI
Overhaled our SPI driver adding integration with DMA and allowing **asynchronous transfers** while establishing an easier to use API.

I2C
Further developed and tested an I2C driver module. Designed a **student-friendly** API.

DMA
Created a new driver for the DMA controller, unlocking **faster and asynchronous** APIs for other drivers.



I2S
Further developed the Mango Pi's I2S drivers, establishing a **more robust** API for students to use when working with mono/stereo audio for both microphones and speakers.

SHMC (SD CARD)
We successfully **ported** and **tested** the (SMHC) for SD cards. Exposed the block read and block write functions, effectively abstracting the underlying hardware layer.

```
mmc_bsp.c
mmc.c
mmc.h
util.c
util.h
```

DEMOS

POV SPHERE

To test the capabilities of our newly developed drivers, we decided to fix the POV sphere: a former student's CS107E final project.

The POV sphere consisted of a strip of SPI LEDs and a Hall effect sensor creating the illusion of a 3D sphere using a 2D circle.

HAT EXPLORATION

DOOM PORT

For the final few weeks of our "hacking", we decided to continue making examples of how our drivers could be used, especially across multiple devices at the same time.

With our DOOM port, we implemented an SPI LCD display, an I2C joystick, and I2S speaker to run the original DOOM game completely bare metal on the Mango Pi.

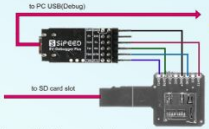
I2C HYDRA

One final test was making a giant "hydra" of our I2C devices including **RTCs**, **OLEDs**, and **accelerometers**, as I2C was the protocol that we had the **most devices working for.**

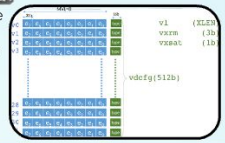
This demo also served as an **experiment** to explore the limits of our drivers and confirm that we could simultaneously communicate with multiple devices .

COURSE REVIEW

One of the biggest challenges for students when working on embedded systems is the lack of a debugger. Currently the course uses a software simulator to allow debugging, but this doesn't allow interacting with hardware components like screens or buttons. We researched the JTAG hardware debugging protocol in order to allow students to debug their code on device. **Future work:** the manufacturer's proprietary software only runs on Linux and Windows machines. Can we make it work on Mac?



In addition to debugging, we developed other aspects of the course to allow students to do more with the Mango Pi, whether that be guides explaining how to enable **Vector instructions** or bare-bones SD card drivers.



LEARNING

Designing an API forced us to think critically about which features to include to provide a solid foundation while leaving open options for further exploration by students.

Thorough testing and code review ensured the implementation was robust and superb quality so students can successfully use and learn from the code

KEY TAKEAWAYS

We gained a lot of experience in development for embedded systems, leveled up our debugging skills, and learned how to interpret datasheets of varying quality.

Alongside learning about the industry, we discovered how difficult and meticulous it is to develop new course materials and refine existing content.

Stanford University

CS107E: RESEARCHING COMPUTER ... FROM THE GRO...

...Elias Chikwanda, Daniel James, and Joe Robertazzi
Mentor – Julie Zelenski

OVERVIE...

This summer we aimed to "hack" on the Mango Pi to **develop its libraries** and capabilities for students to use in the CS107E course

We **established APIs** for MangoPi serial communication protocols, allowing students to pursue more ambitious projects.

DRIVERS

SPI
Overhaled our SPI driver adding integration with DMA and allowing **asynchronous transfers** while establishing an easier to use API.

I2C
Further developed and tested an I2C driver module. Designed a **student-friendly** API.

DMA
Created a new driver for the DMA controller, unlocking **faster and asynchronous** APIs for other drivers.

I2S
Further developed the Mango Pi's I2S drivers, establishing a **more robust** API for students to use when working with mono/stereo audio for both microphones and speakers.

SHMC (SD CARD)
We successfully **ported** and **tested** the (SMHC) for SD cards. Exposed the block read and block write functions, effectively abstracting the underlying hardware layer.

```
mmc_bsp.c
mmc.c
mmc.h
util.c
util.h
```

DEMOS

P...V SPHERE

...sphere using...

...OOM P...

For the final ..., we decided to con... how our drivers c... across multiple dev...

With our DOOM port, ... SPI LCD display, an I2C ... speaker to run the origina... completely bare metal on t...

I2C HYDRA

One final test was making a giant "hydra" of our I2C devices including **RTCs**, **OLEDs**, and **accelerometers**, as I2C was the protocol that we had the **most devices working for.**

This demo also served as an **experiment** to explore the limits of our drivers and confirm that we could simultaneously communicate with multiple devices .

...RSE REVIEW

...est challenges for students when working on ...systems is the lack of a debugger. Currently the ...ses a software simulator to allow debugging, but ...doesn't allow interacting with hardware components ...ke screens or buttons. We researched the JTAG hardware debugging protocol in order to allow students to debug their code on device. **Future work:** the manufacturer's proprietary software only runs on Linux and Windows machines. Can we make it work on Mac?

In addition to debugging, we developed other aspects of the course to allow students to do more with the Mango Pi, whether that be guides explaining how to enable **Vector instructions** or bare-bones SD card drivers.

LEARNING

Designing an API forced us to think critically about which features to include to provide a solid foundation while leaving open options for further exploration by students.

Thorough testing and code review ensured the implementation was robust and superb quality so students can successfully use and learn from the code

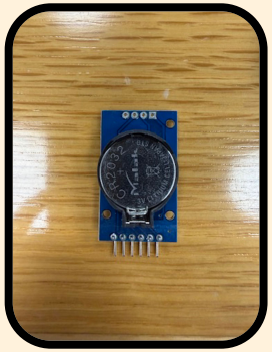
KEY TAKEAWAYS

We gained a lot of experience in development for embedded systems, leveled up our debugging skills, and learned how to interpret datasheets of varying quality.

Alongside learning about the industry, we discovered how difficult and meticulous it is to develop new course materials and refine existing content.

SUMMER OVERVIEW

WEEK 1



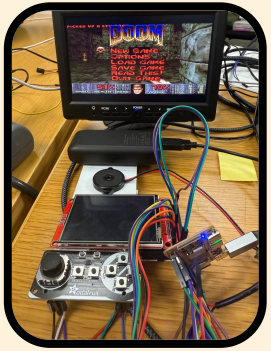
WEEK 3

WEEK 5

WEEK 7

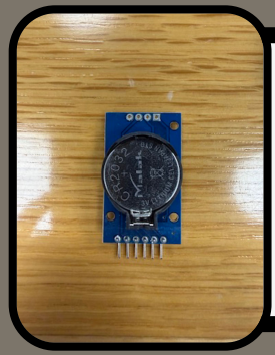
WEEK 9

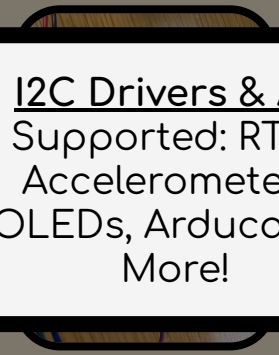
DEMO

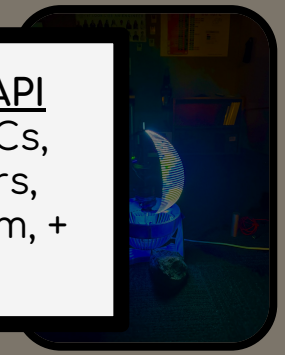


SUMMER OVERVIEW

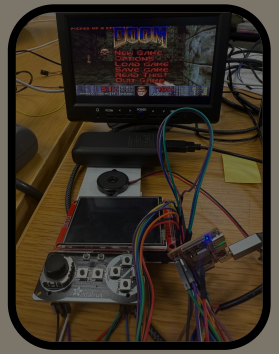
WEEK 1 | WEEK 3 | WEEK 5 | WEEK 7 | WEEK 9 | DEMO





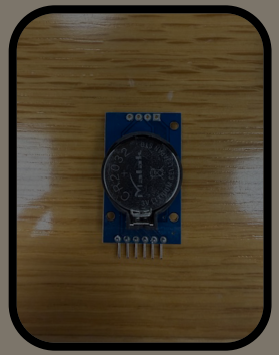


I2C Drivers & API
Supported: RTCs, Accelerometers, OLEDs, Arducam, + More!

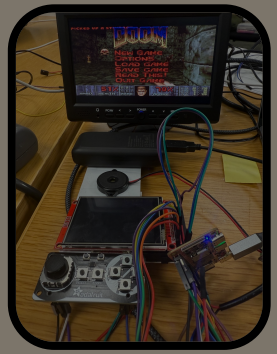


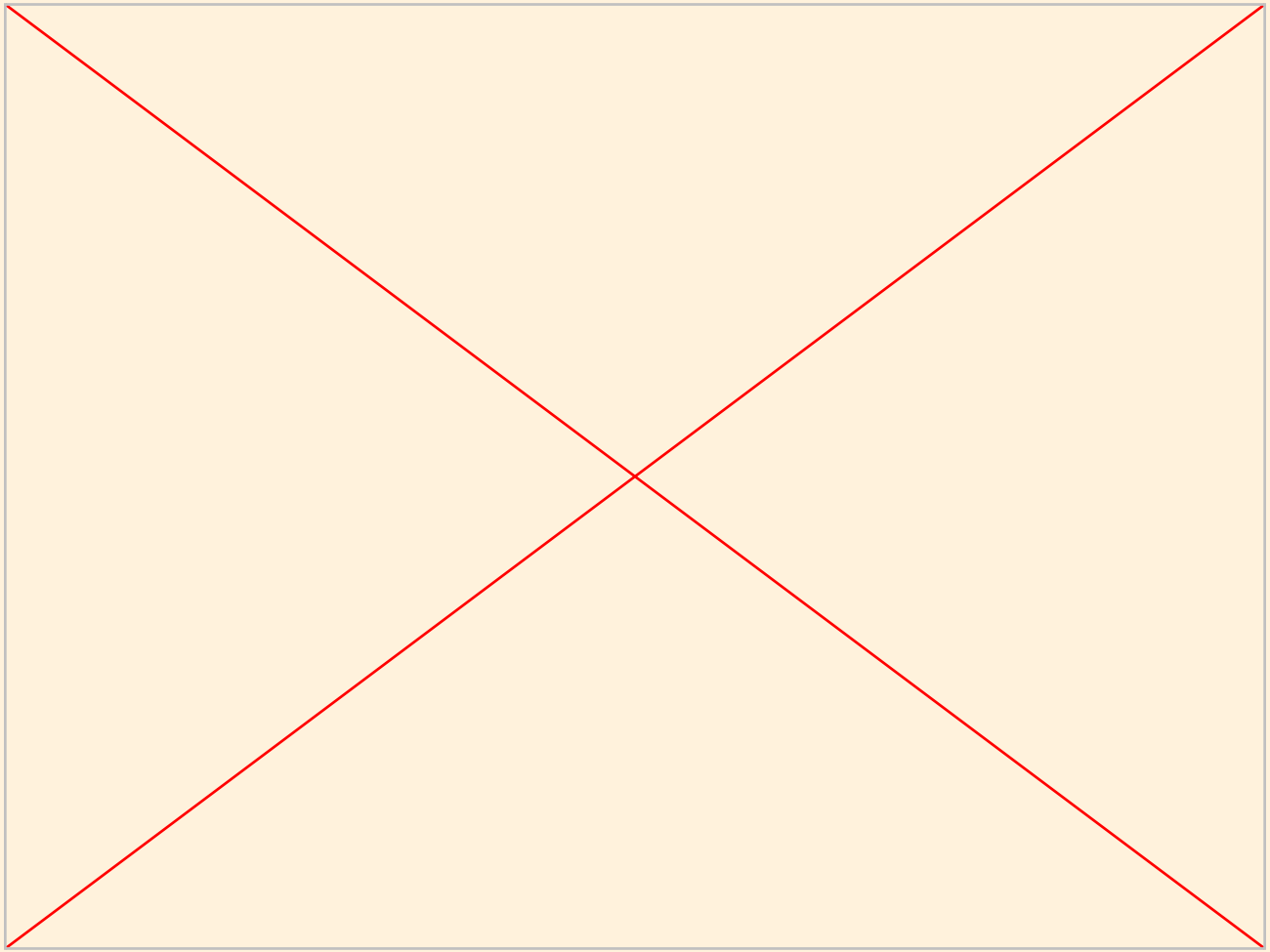
SUMMER OVERVIEW

WEEK 1 | **WEEK 3** | WEEK 5 | WEEK 7 | WEEK 9 | DEMO



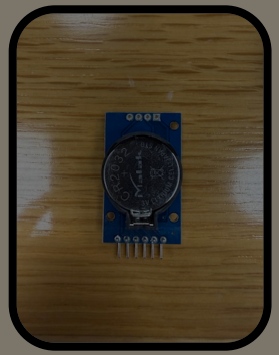
SPI+DMA Drivers & API
Supported: LCD screens, Asynchronous Data Transfers, High Speed Data Transfers!





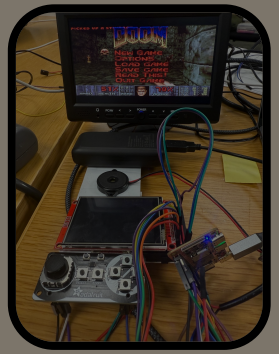
SUMMER OVERVIEW

WEEK 1 | WEEK 3 | WEEK 5 | WEEK 7 | WEEK 9 | DEMO



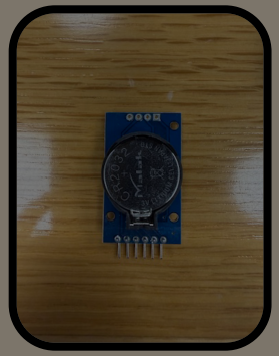
Project Porting

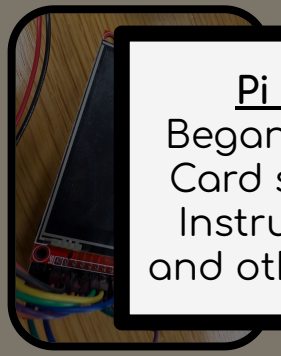
Ported old RaspberryPi projects to the MangoPi, including POV Sphere!



SUMMER OVERVIEW

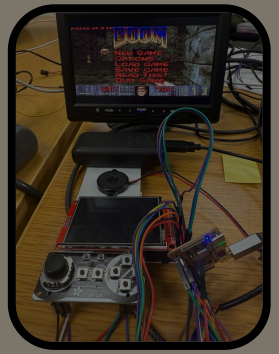
WEEK 1 | WEEK 3 | WEEK 5 | **WEEK 7** | WEEK 9 | DEMO





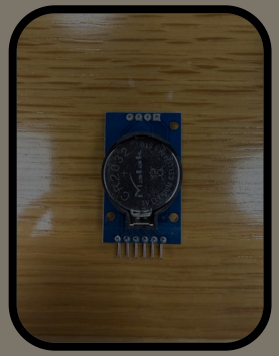
Pi Exploration

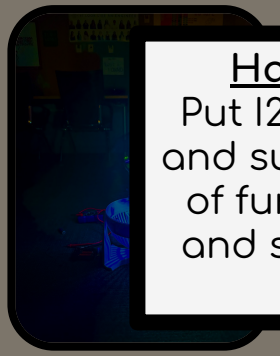
Began working on SD Card support, Vector Instruction support, and other Misc. areas...



SUMMER OVERVIEW

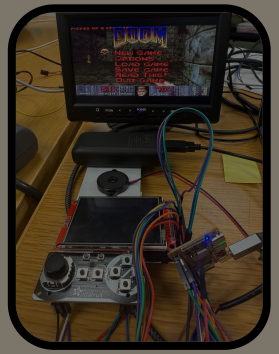
WEEK 1 | WEEK 3 | WEEK 5 | WEEK 7 | WEEK 9 | DEMO





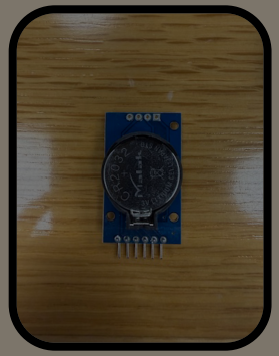
Hats + Research

Put I2C API to the test and supported a bunch of fun hats for the PI, and started personal projects



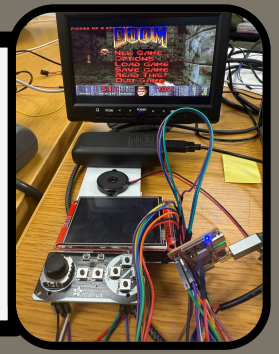
SUMMER OVERVIEW

WEEK 1 | WEEK 3 | WEEK 5 | WEEK 7 | WEEK 9 | DEMO



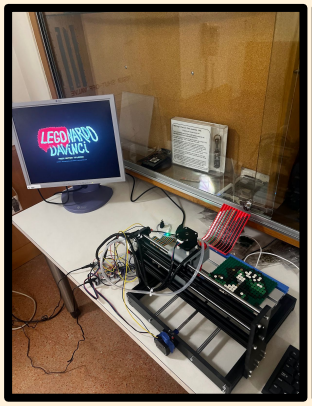
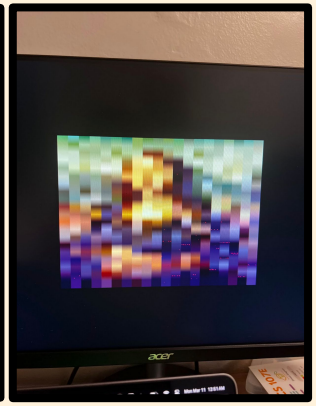
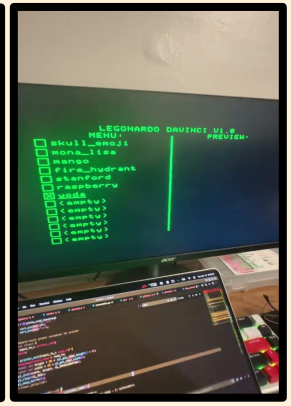
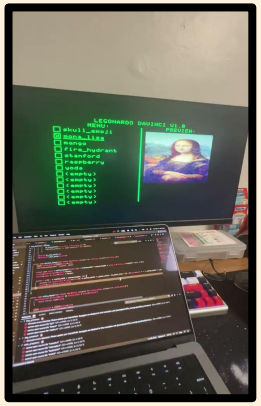
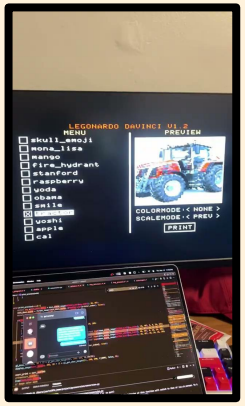
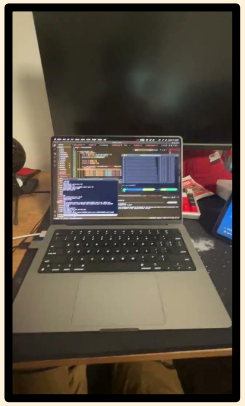
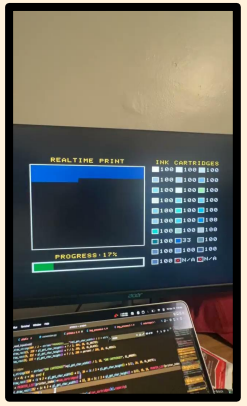
Demo... DOOM

Put everything together into a presentation for CURIS and a full port of DOOM to the MangoPi

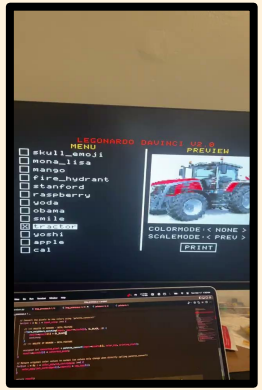
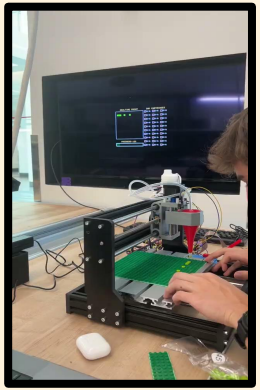
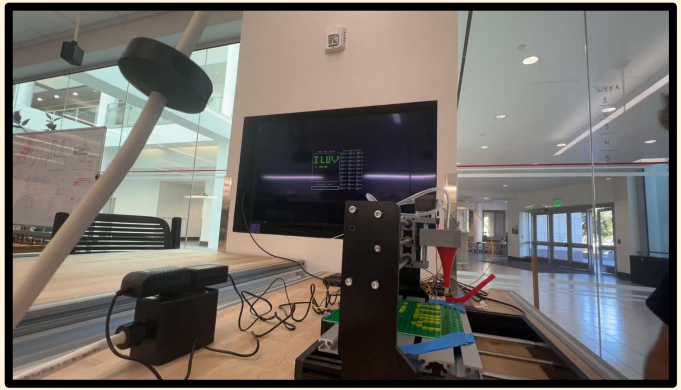
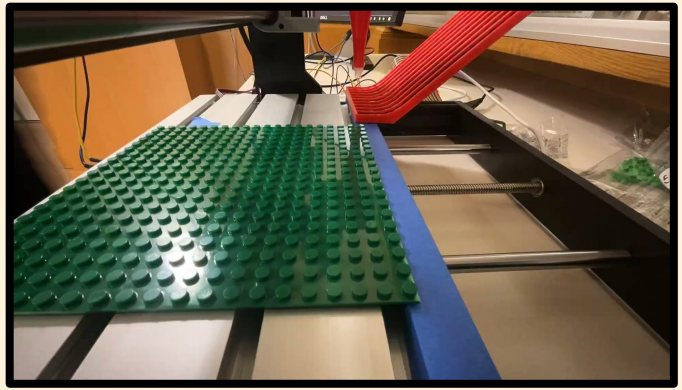


Project Info/Demos!

Joe's 107E Project

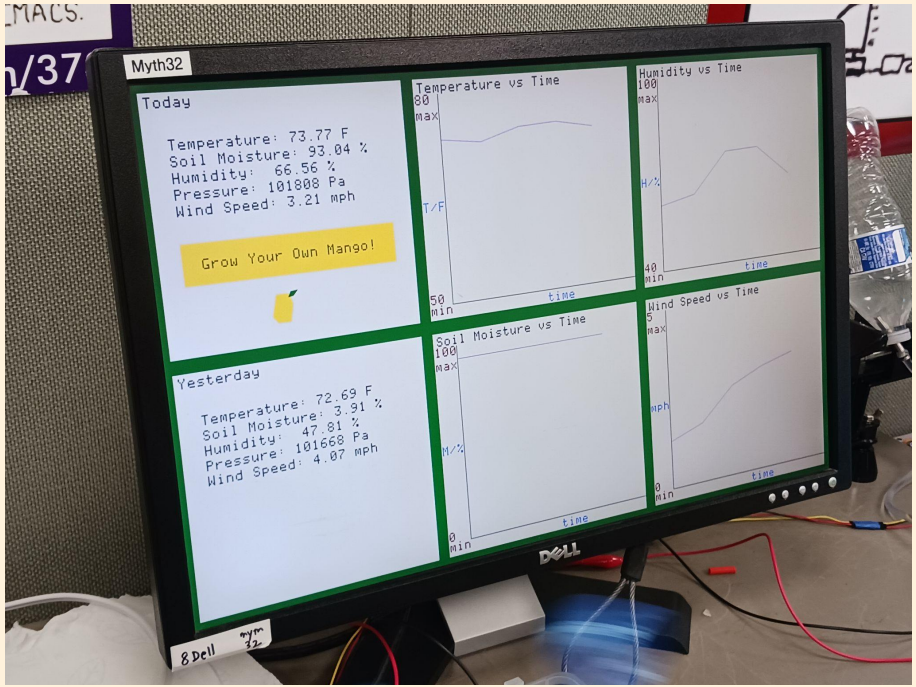








Joe's 107E Project

Elias' 107E Project

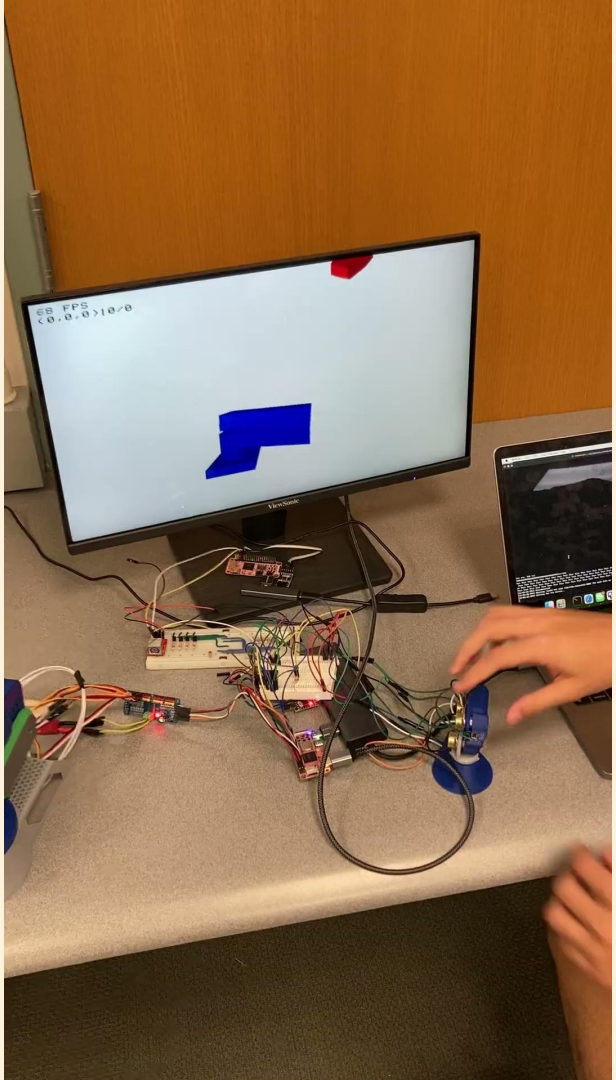
GROW YOUR OWN MANGO



Sensors Used

- BME280 Sensor (Temperature, Humidity, Pressure)
- Hall Effect Sensor
- Soil Moisture Sensor

Daniel's 107E Project


60 FPS
(0,0,0)|0/0