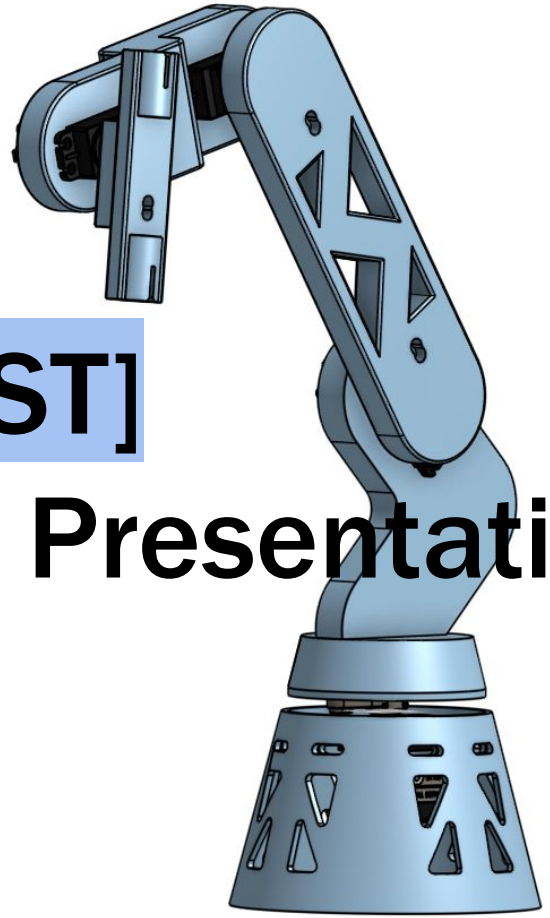
[ASSIST]

Hackathon Final Presentation



Who and What is Assist?

We wanted to create a multi-functional assistive feeding device.

Obi 3 Robotic Feeding Device – Adaptive Dining Robot for Independent Mealtime Support

Obi | SKU: E19667 | MPN: IFD-500-031

$12,000.00

- still requires a caregiver + controls
- Individual parts (for replacement) costs $150+ each
- you have to purchase a spork

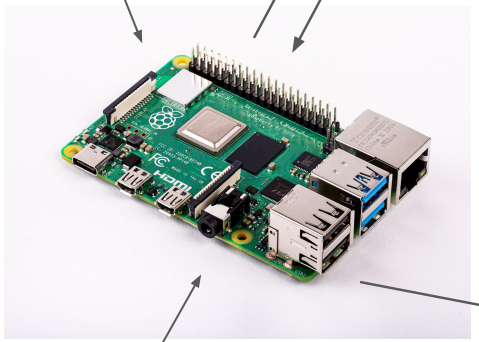
Eating should still be fun, comfortable, and high-tech. **So, we decided to create one for fun.**

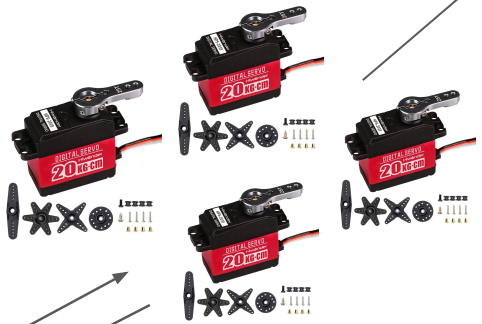
*https://especialneeds.com/products/obi-robotic-self-feeder?srsltid=AfmBOorwWxmGJZVBdWvLzQ8VTjXKro7_oZW1ZGEts-nOD7MvMD0wH4T

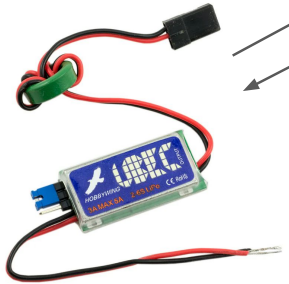
Simplified Top Down View

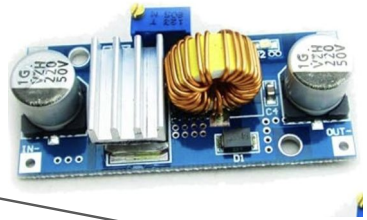
Robot

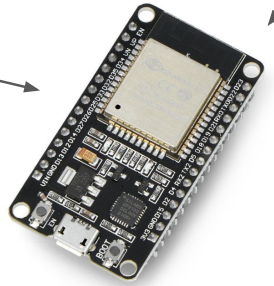
OpenCV + git

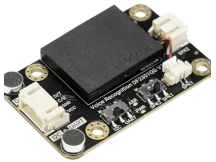








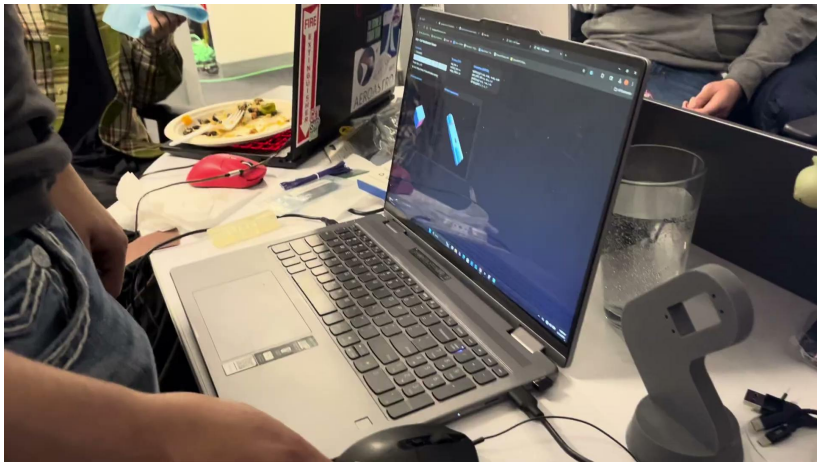


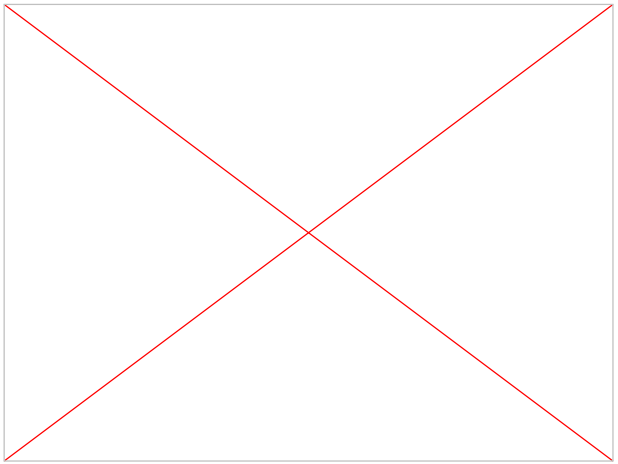


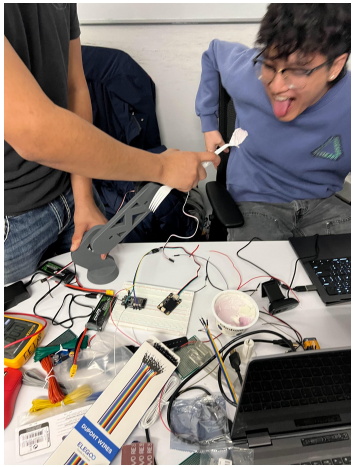
Power

Work In Progress

- Programmable robotic arm with several individually working sensors
- Working website showcasing the functionality of each sensor in action
- Somewhat trustworthy opencv (never got the chance to fully test this)
- Never completely soldered everything so we cannot test/demo







IMU + ToF WebSocket Viewer

Connection

● Connected

ws://192.168.4.1:81 | Connect | Disconnect

Packet rate: 25 Hz

☐ Axis remap (swap if cube orientation wrong)

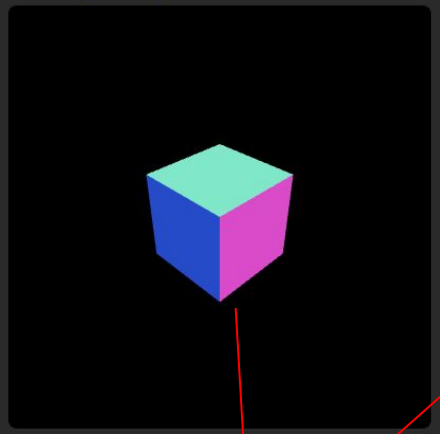
Distance (ToF)

raw_mm: 0
vertical_mm: 0.0
range_status: 255

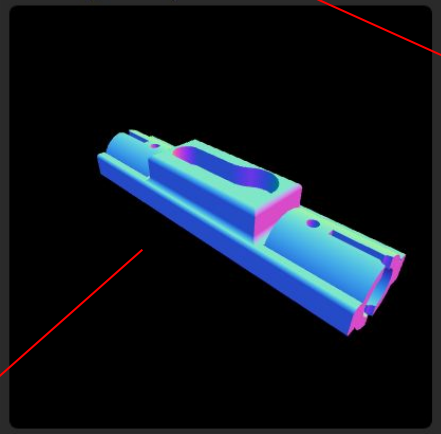
Orientation (BNO055)

quat w,x,y,z: 0.000, 0.000, 0.000, 0.000
yaw, pitch, roll: 0.0, 0.0, 0.0
gravity x,y,z: 0.00, 0.00, 0.00
cal sys,g,a,m: 0, 0, 0, 0

3D Cube (quaternion)



3D Model (quaternion)



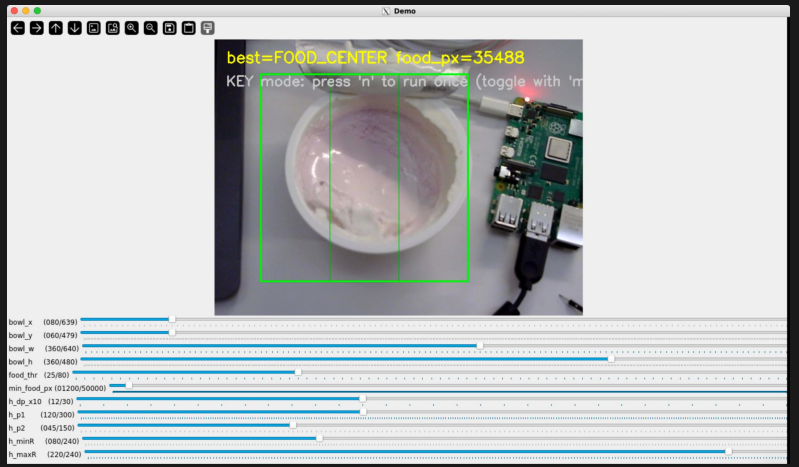
Raw Orientation Data

Raw Distance Data

Visual Model

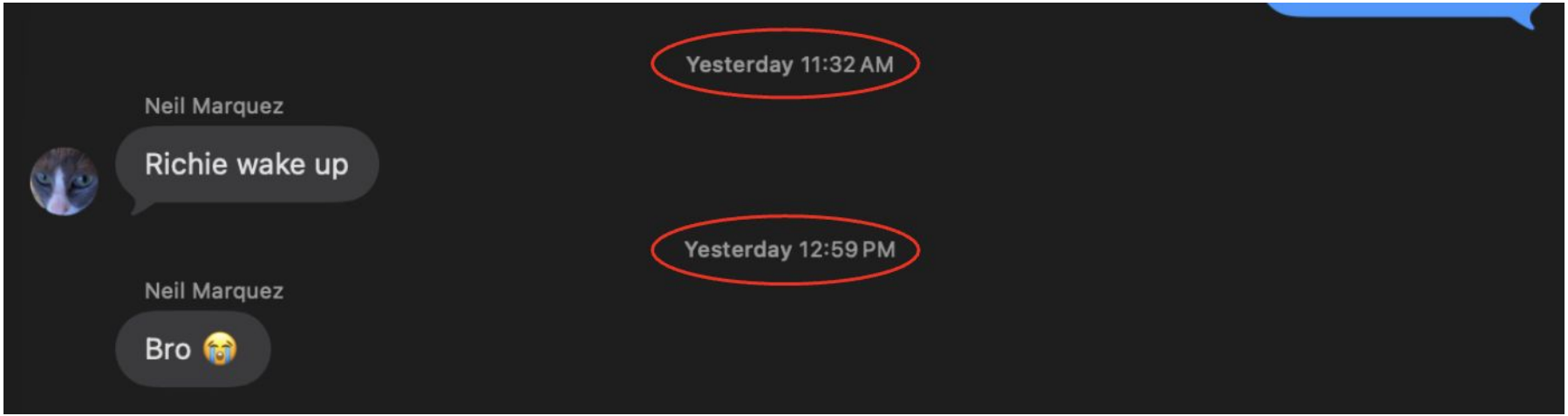
```
[kiuin@assistive-arm:~ $ cd ~/assist/rasberrypi/opencv                    ]
[kiuin@assistive-arm:~/assist/rasberrypi/opencv $ python3 brain_mac.py    ]
[ WARN:0@2.499] global cap_gstreamer.cpp:1777 open OpenCV | GStreamer warning: C
annot query video position: status=0, value=-1, duration=-1
No matching fbConfigs or visuals found
glx: failed to create drisw screen
Loaded baseline.

=== No-marker Bowl Zone Demo ===
Keys:
  b = capture EMPTY bowl baseline
  n = run ONE cycle (key mode)
  m = toggle key/auto mode
  f = try AUTO-FIND bowl circle (sets ROI)
  w = save config
  ESC = quit
```



Biggest Takeaways

- Designing for modularity
- Working with OpenCV, data collection, and calibration (ROI +baseline diff)
- Time-effective SLA printing
- Effective soldering
- ~~How to wake up on time~~



What’s Next?

- Finish soldering the electronics (crucial)
- Updating the OpenCV code
- Improvements in design, creating areas to place electronics safely
- Add in additional components + features including bite detection and maybe knife