SDP25

Team #15 Workspace Wizard

Meet the Team

Aryaman Ghura
CompE & CS

Knolling and Path Planning Algorithms

Ishaan Salian
CompE

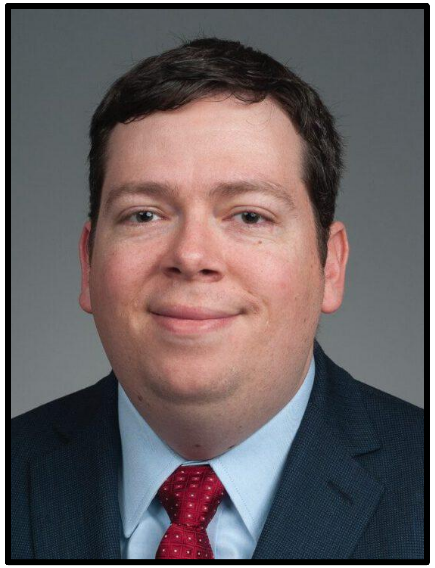
Mechatronics & PCB
Logistics

Mary Esenther
CompE

Power Delivery & Camera Processing

Kavya Manchanda
CompE

Processor & Camera
Budget



Professor Marco Duarte

Advisor

Problem Statement

- A cluttered desk can negatively affect productivity and mental health.
- Studies show that messy workspaces increase stress. [1]
- 57% of Americans associate messy desks with laziness. [2]

[1] "The Psychological Consequences of Clutter," Psychology Today, 2021.
[2] L. Alton, "The Negative Relationship Between a Messy Desk and Productivity," Inc.com, Feb. 16, 2017.

Project Goal

To create a distributed system that can organize your workspace autonomously by employing object detection and path finding.

Design Specifications for Project

Overhead Computer

- Detect human absence with 98% accuracy to start knolling.
- Map and find viable paths for object placement.
- ~~Align objects at ~90±5° angles~~ Orient Objects
- Identify objects to avoid, like liquids or monitors, with 98% accuracy.
- Recognize table boundaries to prevent overshooting.

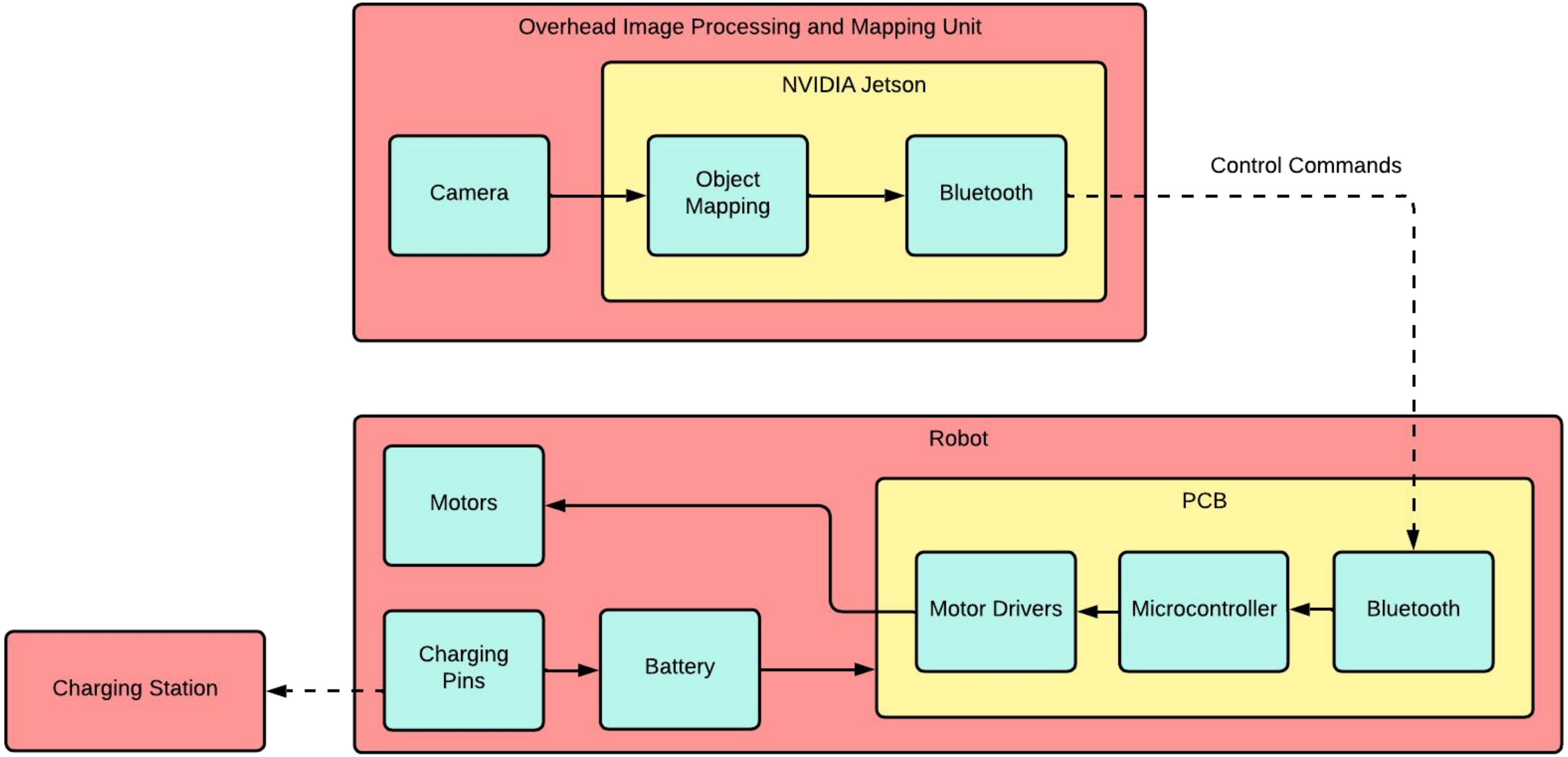
Robot

- Run for 30±5 minutes on one full charge.
- Knoll objects up to 4.5lbs, like a MacBook.
- Knoll up to 10 objects.

Project Testing Plan based on Design Specifications (Updated)

Design Specification	Testing Plans
Detect human absence	Robot movement starts only after 2 minutes of human absence in at least 98 out of 100 trials (reduced from 20 minutes for demo purposes)
Map viable paths for object placement	The robot should successfully navigate around obstacles to find viable paths without getting stuck, if there exists one.
~~Align objects at ~90±5° angles~~ Orient Objects	Orient objects such that their bounding boxes consume the least area
Identify objects to avoid	System should identify various forms objects (open cup, bottle, user-defined objects) with correct classifications in at least 98 out of 100 trials
Recognize table boundaries	No objects, nor the robot, should exceed the table boundaries
Run for 30±5 minutes on one full charge	Run robot for 30 minutes while monitoring battery levels
Move objects up to 4.5lbs, like a MacBook	The robot should successfully push objects weighing up to 4.5lbs
Knoll up to 10 (varying) objects	The robot should knoll all 10 objects accurately in at least 95% of trials, if knolling path exists

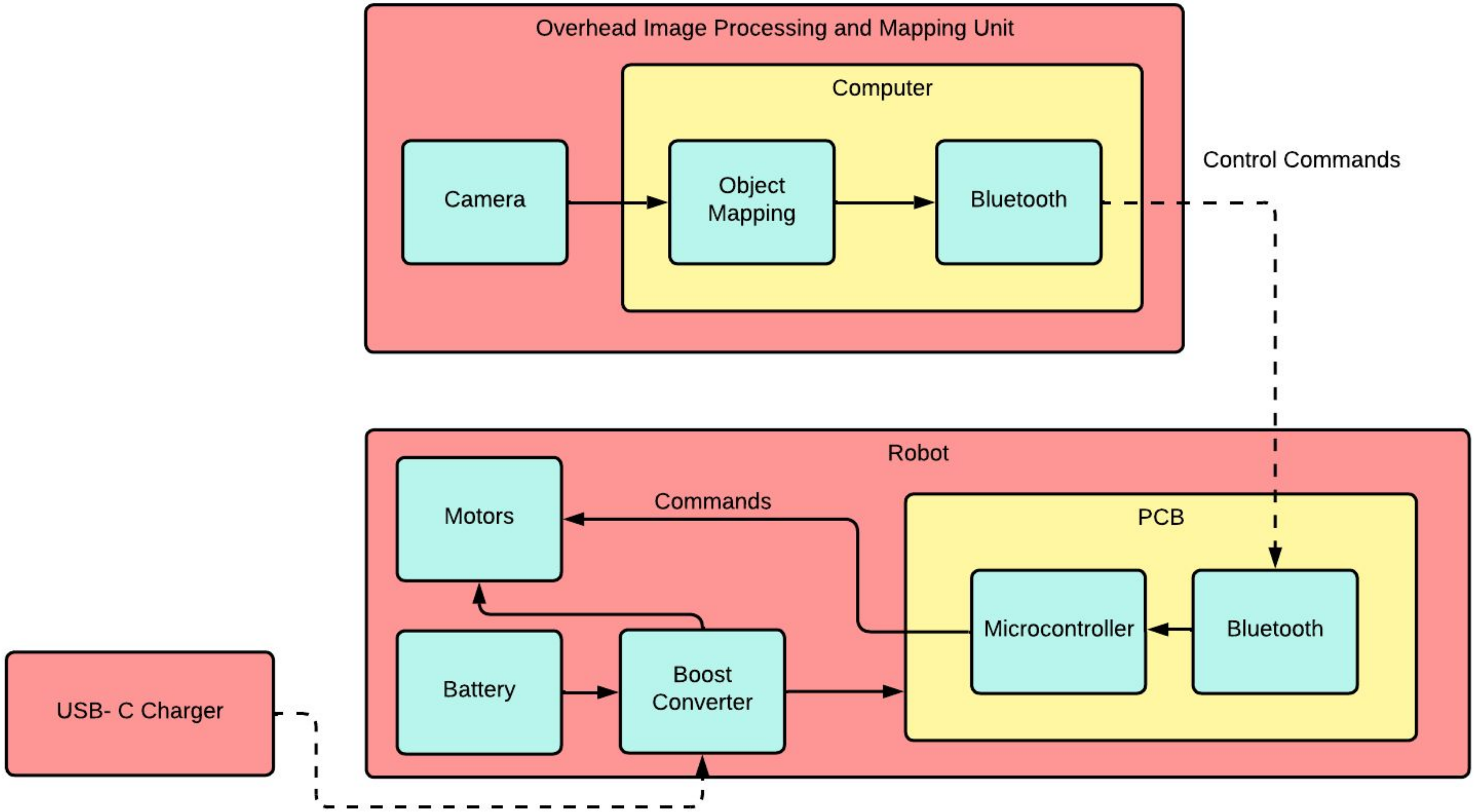
General Block Diagram (from MDR)



Overhead Computer:
NVIDIA Jetson Nano (Computer)
JETSON CAMERA (CSI Camera)
Edimax N150 (BLE)

Robot:
ESP32 (MCU+BLE)
TMC2209 (Drivers)
Stepper Motors,
22V Li-Po Batteries

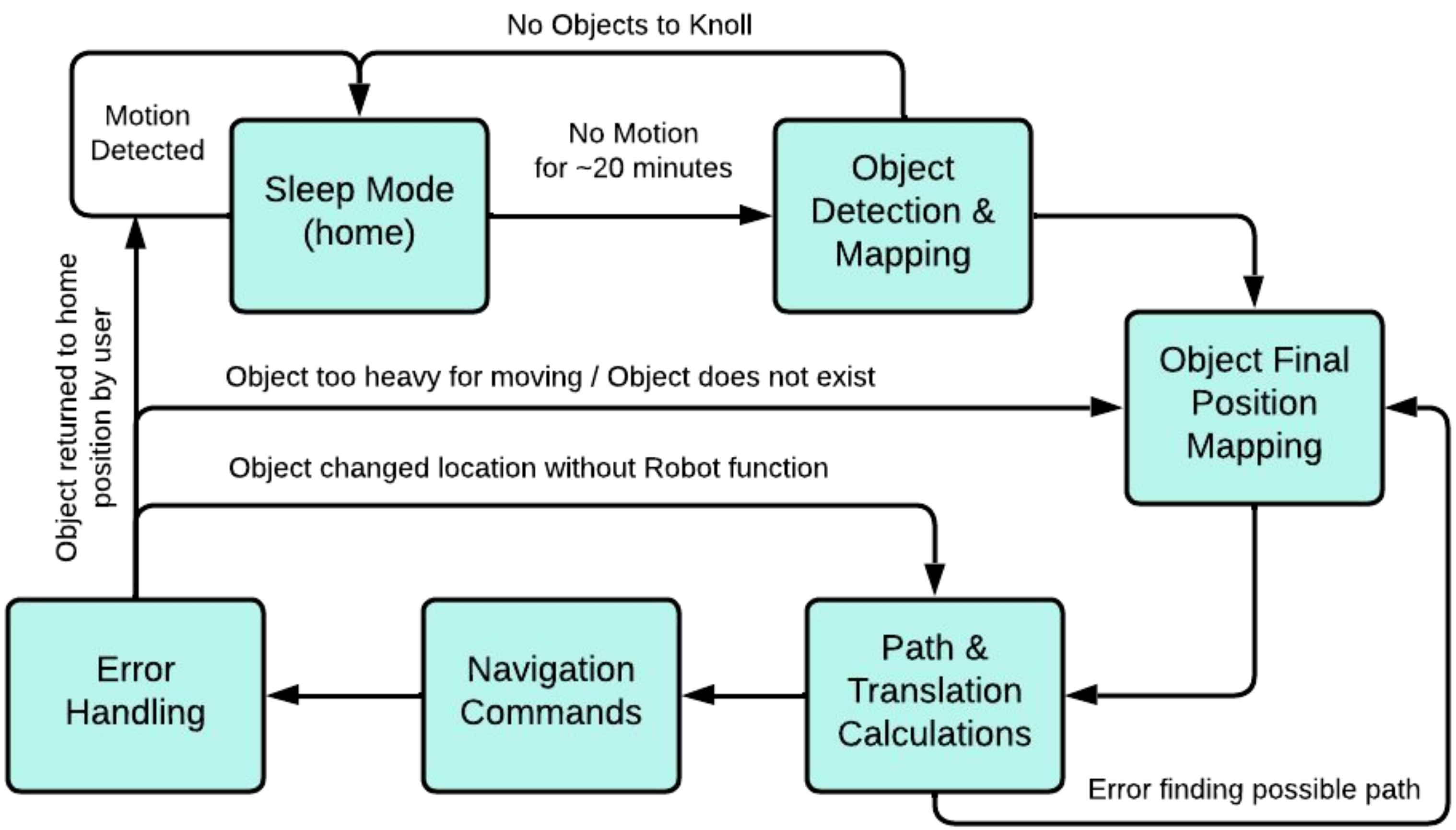
Updated General Block Diagram (CDR)



Overhead Computer:
NVIDIA Jetson Nano (Computer)
USB Camera
Edimax N150 (BLE)

Robot:
ESP32 (MCU+BLE)
TMC2209 (Drivers)
Servo Motors
3.7V Li-Po Battery (5000mAh)

Updated Software Diagram (MDR)



OpenCV, Media Pipe, DarkNet:
Image processing and object detection.

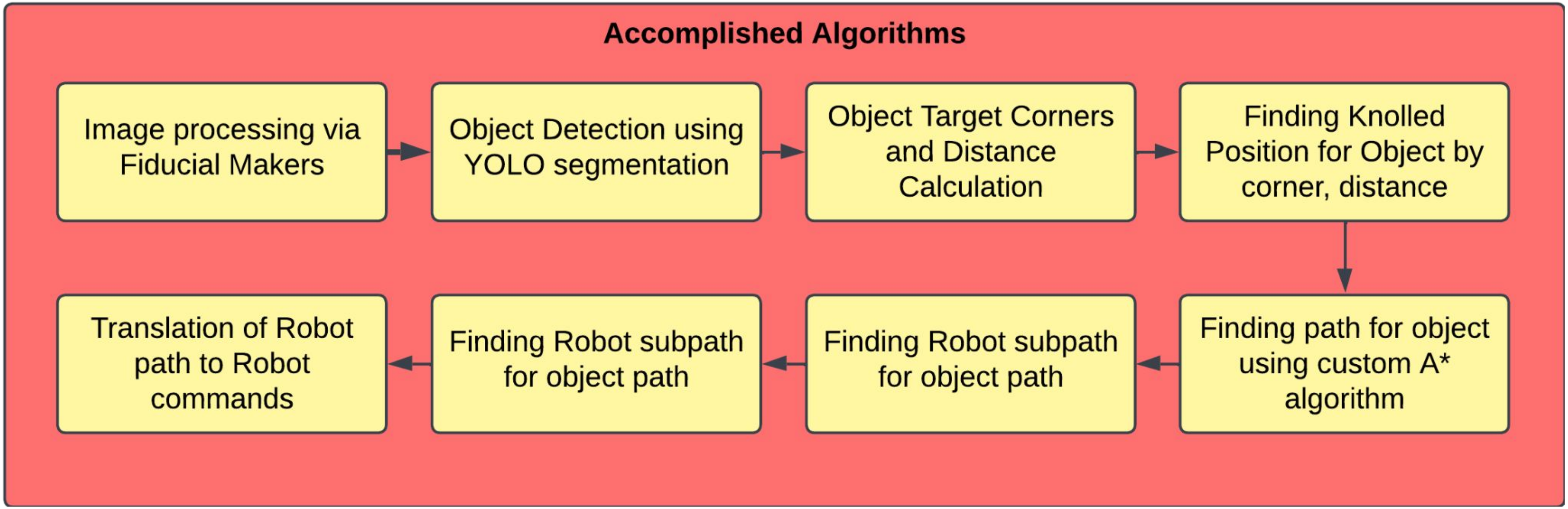
Pybullet:
Visualization and Testing

PCL (Point Cloud Library) (Hold):
2D mapping and object pose estimation.

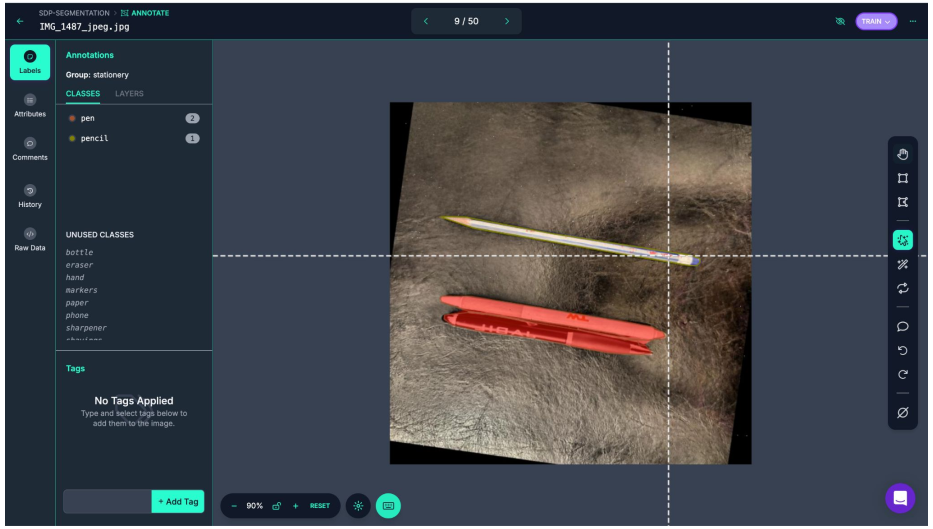
OMPL (Open Motion Planning Library)

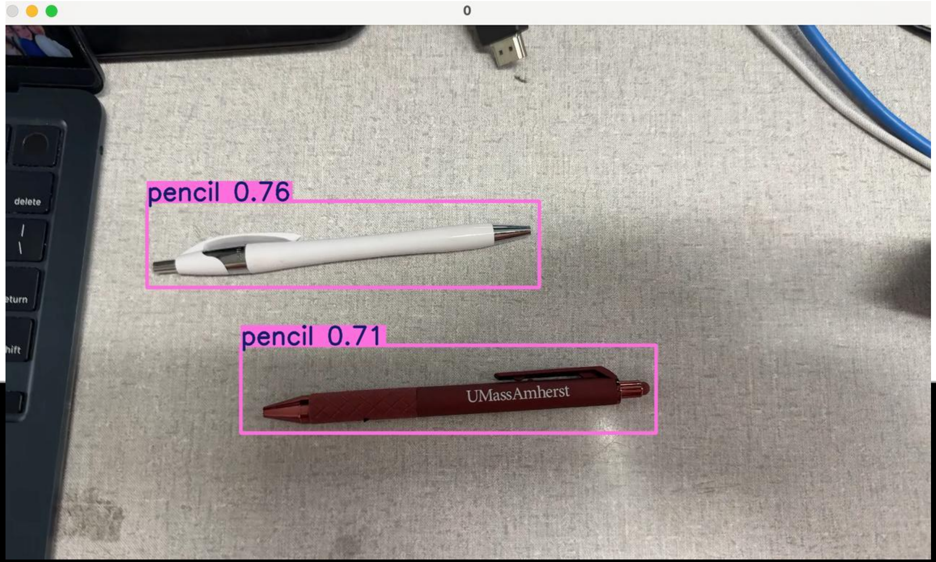
Networkx, Heap, Numpy, Math, Copy, Time

Detailed Software Diagram



Custom Trained Model





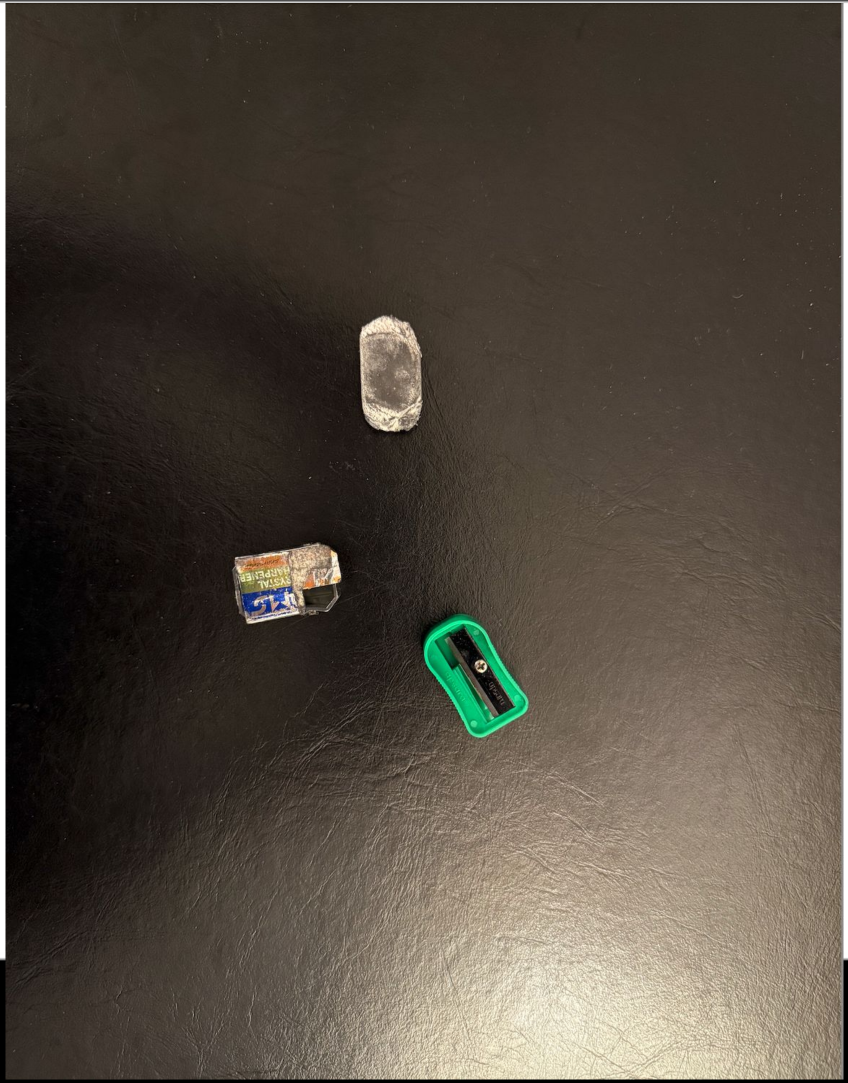
Objects we are identifying:

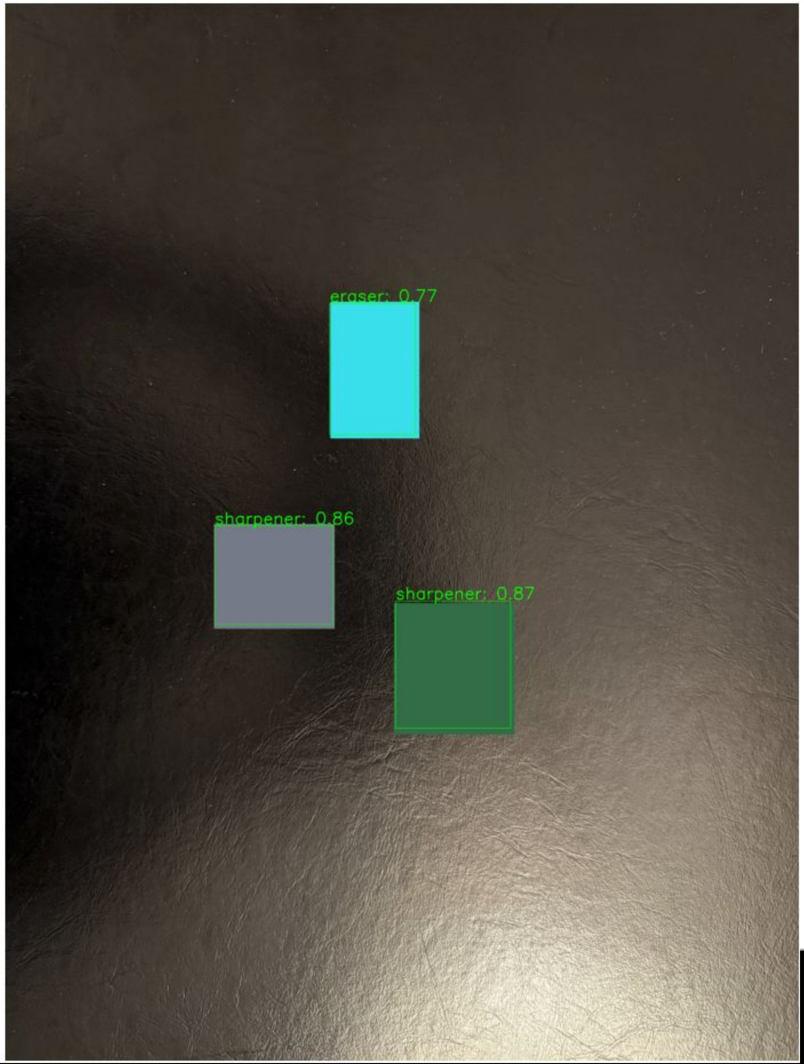
- Pencils
- Pens
- Erasers
- Sharpeners and shavings
- Bottle (un-knollable)
- Phone

Roboflow Universe to get dataset + own pictures

- Trained dataset with Ultralytics
- Using YOLOv8 for instance segmentation

Progress so far…





Tasks:

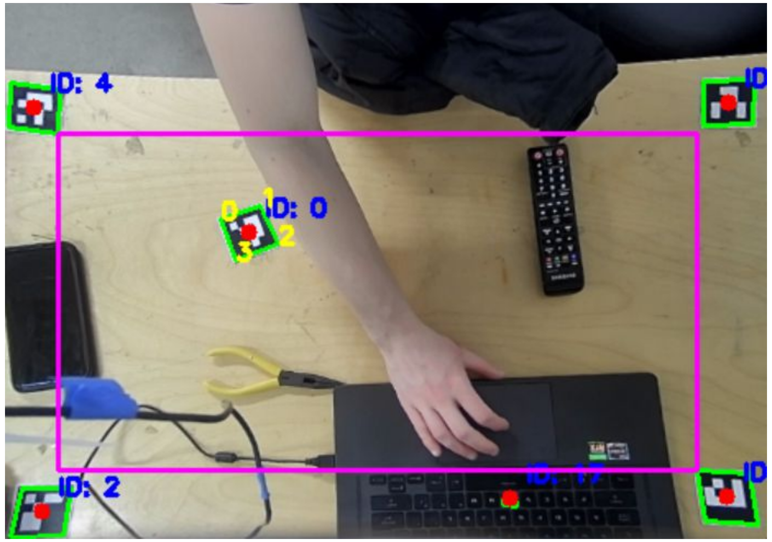
- Increase accuracy to 98%
- Provide clear boundaries for objects by annotating segmentation boundaries

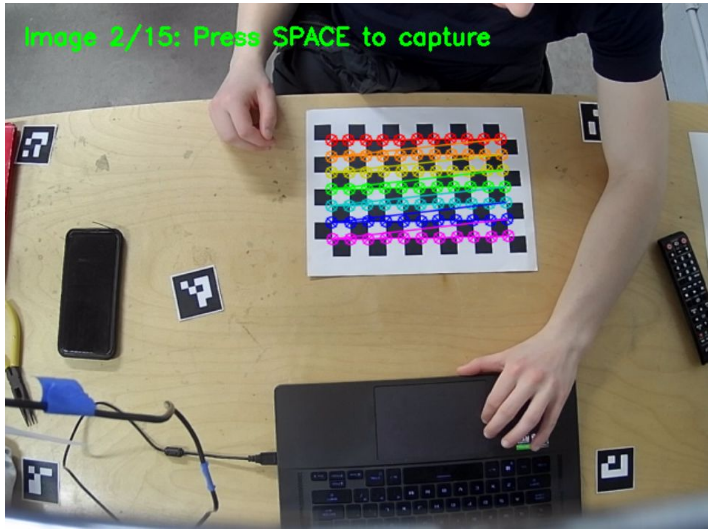
Fiducial Marker Tracking

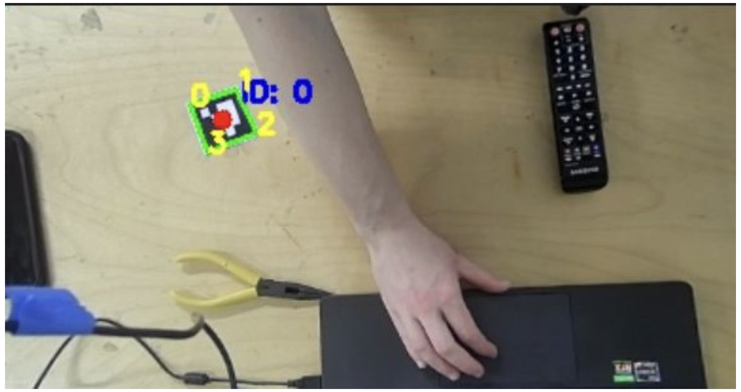
Step 1: Calibrate camera with chessboard images

Step 2: Crop image using outside fiducials and remove "fisheye" effect

Step 3: Print coordinates of the robot fiducial marker's corners





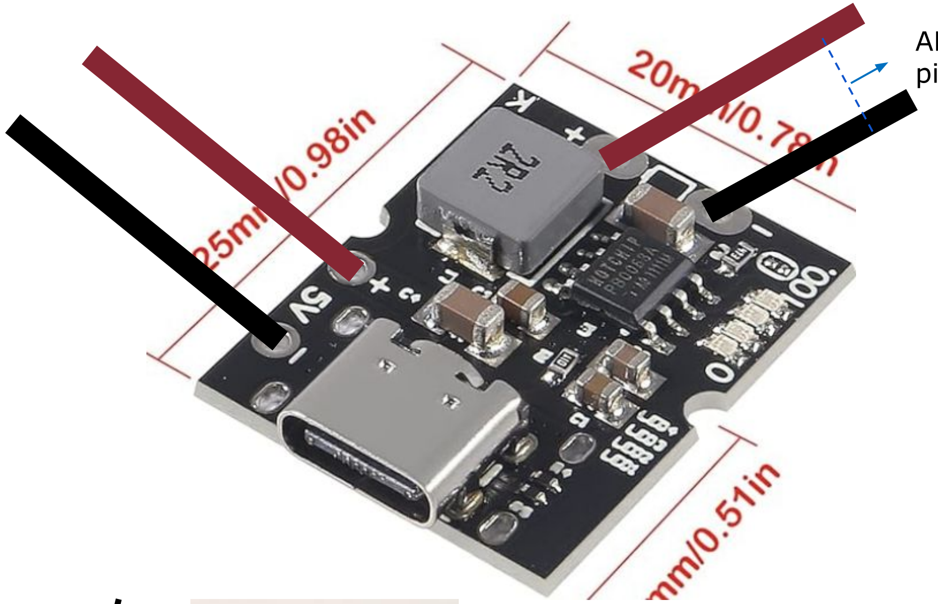


Simplified Power System

ESP32
(250mA max)
Motor 1
(600mA max)
Motor 2
(600mA max)

= **1.45A max**

Limit: 2A



ADC
pin

3.7V 5000mAh LiPo
Time = Capacity/Current
Time = 5Ah/2A = 2.5h mins

Expected:
5Ah/1.45A = **3.45h mins**

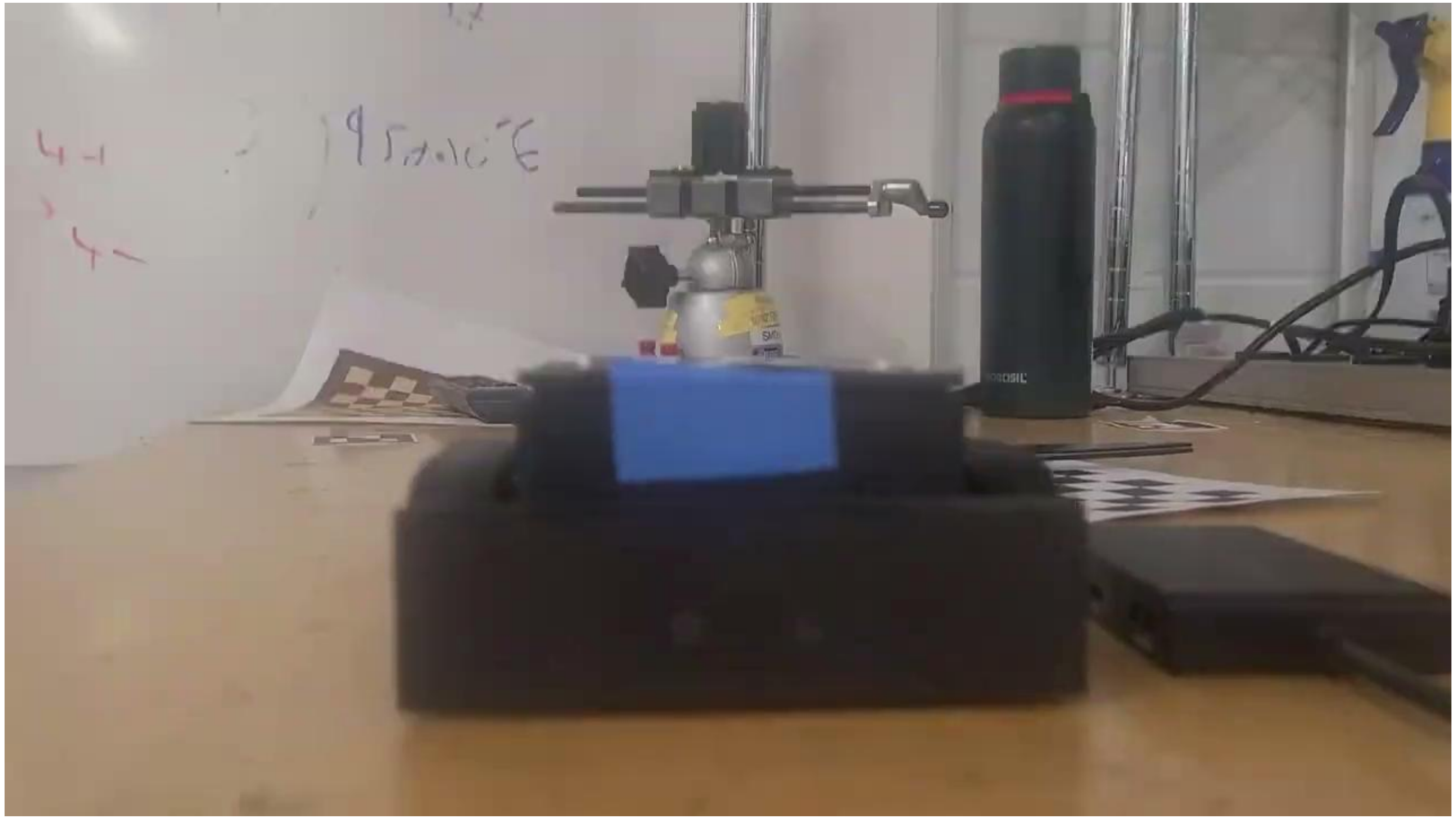
Charging:
5Ah/(charge current)

Expected:
5Ah/3A = **1.67h**

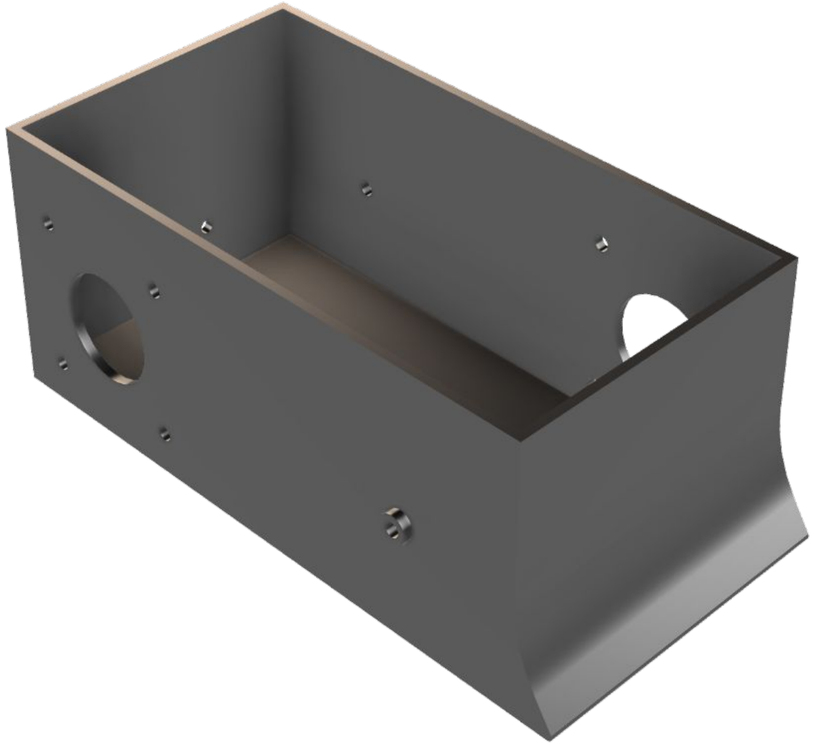
/

Voltage Tracking:
Battery terminals
feed into ADC pin of
ESP32

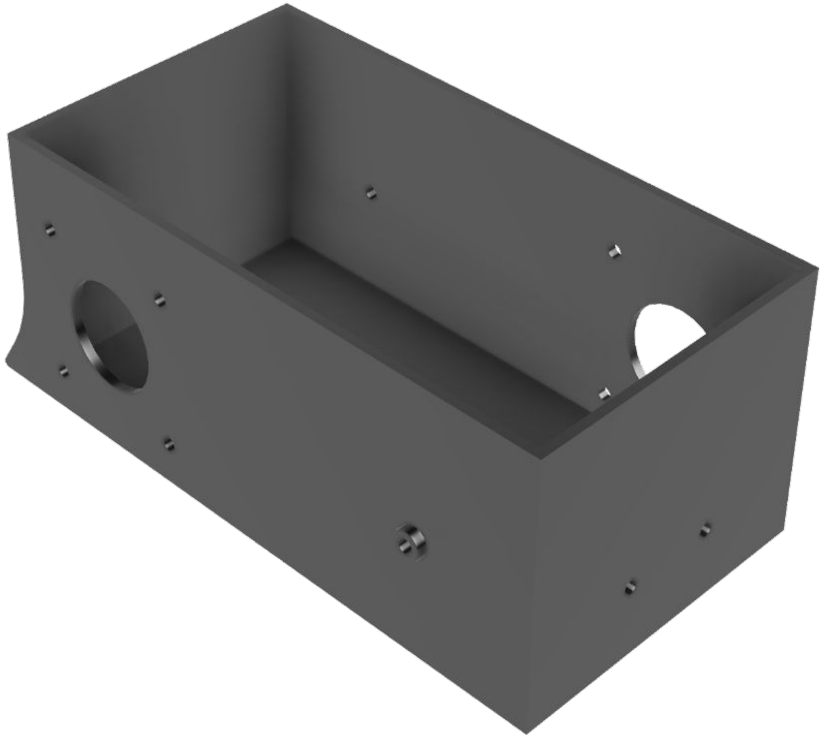
Simplified Power System: Demo



Design (from MDR)



Front

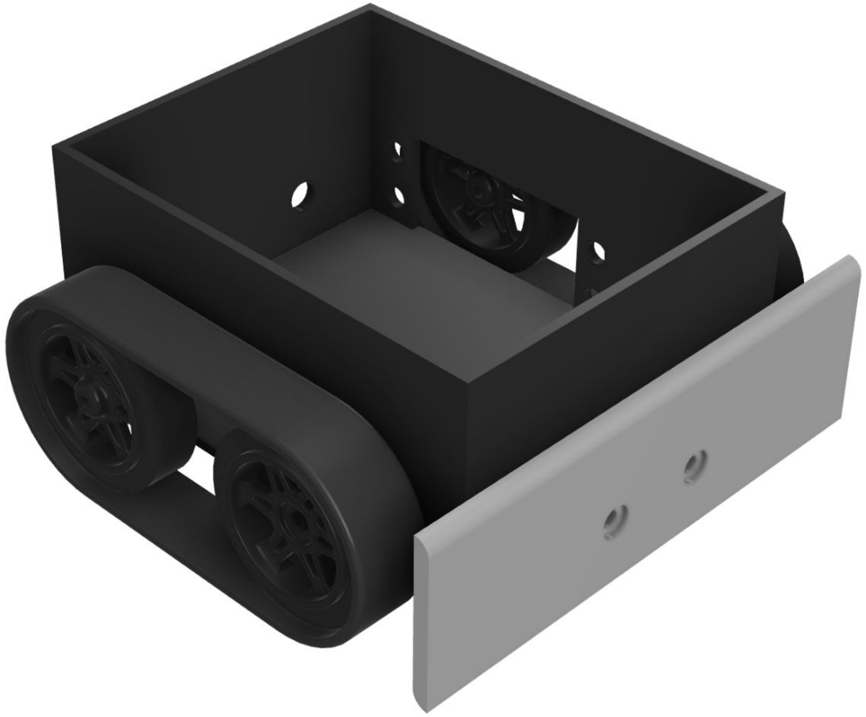


Back

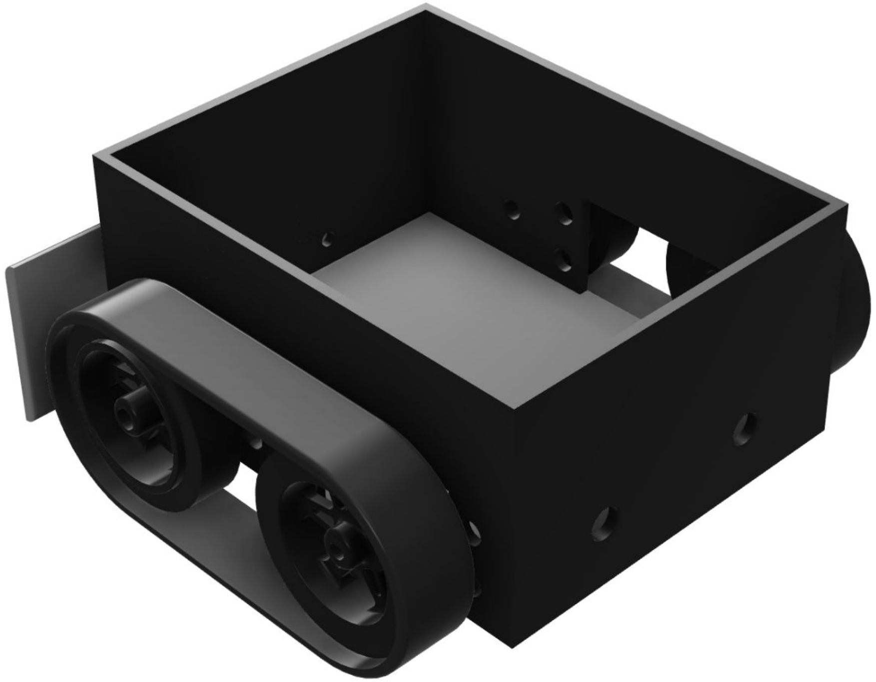
Shortcomings:

- Large size
- Motors placed diagonally
- Does not incorporate tracks

Current Design



Front

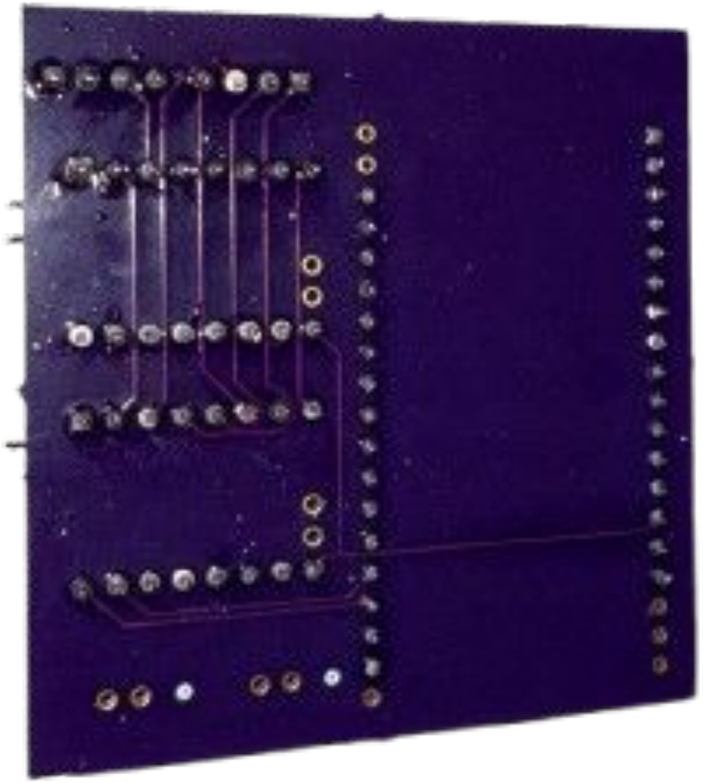


Back

Updates:

- Reduced footprint
- Better aligned motors
- Incorporates tracks
- Modular plow mechanism for alternate designs

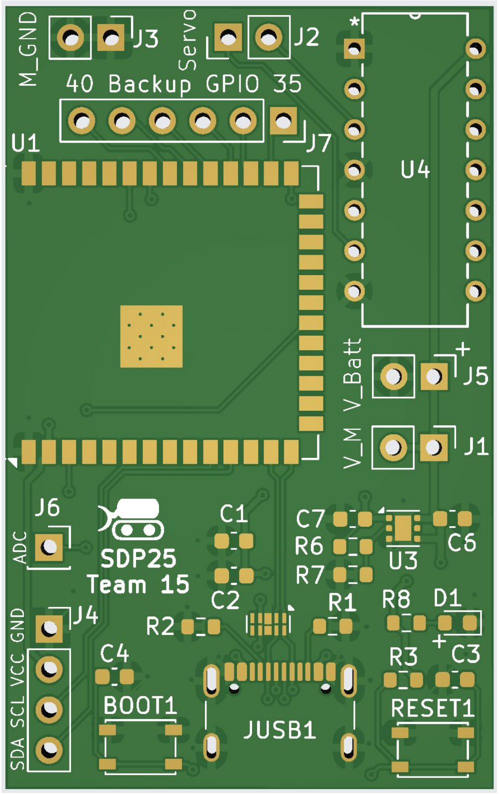
PCB (from MDR)



- ESP32 combines a powerful MCU with on chip BLE

Updated PCB (CDR)

Populated PCB



Unpopulated PCB

Updates:

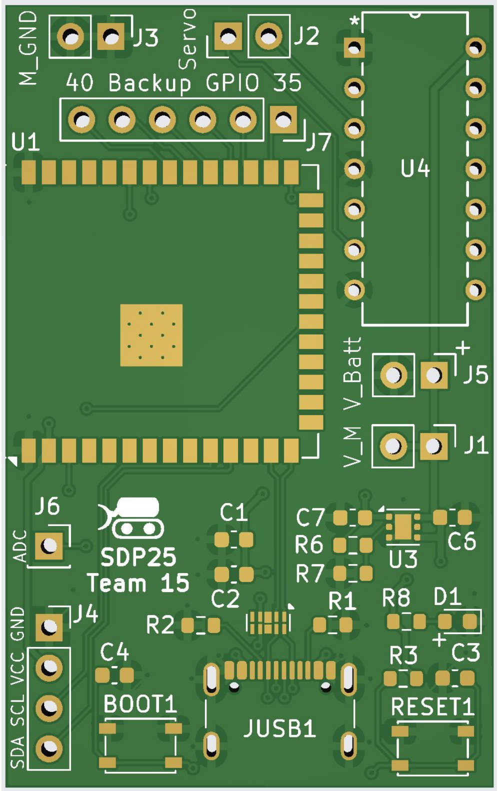
- Uses ESP32-S3 module, instead of the devboard.
- 5 backup GPIO pins.
- OLED display for human interaction.
- Level shifter for PWM signal, and ADC for battery % reading.
- USB-C for programming, with separate line for power from battery.

Problems with MDR PCB

- Selected the wrong ESP32 dev-board.
- Designed the PCB for 3.3V circuitry, but the design was later changed to use 5-7V unregulated input.
- Decided to use a single power supply regulated from 15V to 5-7V, eliminating the need for two separate power pins.
- Unable to test with the PCB; the demonstration will be conducted on a breadboard instead.

PCB Plan (for FPR)

Populated PCB



Unpopulated PCB

Plan:

- Add resistor circuit on PCB for ADC.

CDR Deliverables (Part 1)

Overhead Computer + Software

✔ - Meet all remaining MDR Deliverables.:
    - Ability to map each "knoll-able" object to a final "knolled" location virtually.
    - Ability to find viable path for robot to propel object from initial position to its final "knolled" location using the A* algorithm (desk will be divided into a grid for robot to find path).

✔ - Translation of calculated path to navigational commands for robot.
- Integration of all above software systems with one another and on the Nvidia Jetson.
- Demo of robot knolling three rectangular objects of weight below the robot's knolling threshold that are scattered across a study desk with ample space in between objects for robot to propel the object in any direction. The study desk will be initialized with a fourth object that must be ignored but considered during path planning.
- Finalized setup of the body of the processor and camera on top of the desk.

CDR Deliverables (Part 2)

Robot

- ✔ Robot (including revised PCB design) is able to navigate according to BLE commands from the Jetson.
- ✔ Next iteration of the robot body integrates suitable drivetrain.

Power Delivery

- ✔ Finalize charging mechanism.
- ✔ Successfully charge a LiPo (minimum 3.7V) using the finalized charging mechanism.

FPR Deliverables

Overhead Computer + Software

- Meet any remaining CDR Deliverables.
- Develop orientation algorithm
- Develop feedback control
- Optimize code for speed and efficiency
- Optimize the instance segmentation custom model for 98% accuracy

Robot + Power Delivery

- Revised PCB design with integrated power management subsystem.
- Improved movement accuracy.
- Integration of all above software/hardware systems with one another and on a centralized platform.

Project Expenditures (so far)

Tracks	$22.90	For the robot wheels	Legend:
PCB print	$5.03	Robot PCB	ROBOT
PCB shipping	$21.24	PCB Shipping Cost	HARDWARE
NVIDIA Jetson Nano	**	Overhead Computer	MISC.
MicroSD Card 64 GB	9.65	Overhead Computer storage	
Type-C USB 5V 2A Boost Converter Step-Up Power Module Lithium Battery Charging Protection Board LED Display	23.98	BMS for the robot	
3.7V Lipo Battery 5000mAh Rechargeable Lithium Polymer Battery Pack with JST PH2.0mm Connector for Electronic Device	28.98	Batteries for the robot	
PCB print (2)	58.6	Robot PCB	
PCB shipping (2)	15.5	PCB Shipping Cost	
Servo Motors (2)	**	Motors for movement of robot	
Total	$185.88		
Remaining	$314.12		

https://docs.google.com/spreadsheets/d/1IjUjBBgdBLAvl8chtcBwEpMHnhw8kTVnyTxhHfsBHh4/edit?gid=0#gid=0
** Acquired from M5

Timeline (To FPR)

PROJECT TITLE	Desk Organizing Robot	SDP Team No.	Team 15
Faculty Advisor	Professor Marco Duarte	DATE	3/11/25

Robot | Power
Combined | Software

PHASE	TASK	TASK OWNER	PROGRESS	WEEK 1 3/16 - 3/22	WEEK 2 3/23 - 3/29	WEEK 3 3/30 - 4/5	WEEK 4 4/6 - 4/12	WEEK 5 4/13 - 4/19	WEEK 6 (FPR) 4/20 - 4/26
				M T W Th F	M T W Th F	M T W Th F	M T W Th F	M T W Th F	M T W Th F
1	Robot								
1.1	Test Movement	IS	30%						
1.2	Finalize Design	IS	30%						
2	Power								
2.1	Add Power to PCB + Test	ME + IS	40%						
3	Finalization of Software								
3.1	Implementation of Feedback Control	ME + AG	5%						
3.2	Testing of Feedback Control	ME + AG							
3.3	Training of Objects	KM	80%						
3.4	Transfer to Jetson	KM	30%						
4	Integration Systems								
4.1	Integrating and troubeshooting	ALL	50%						

https://docs.google.com/spreadsheets/d/1pSwbcx0UXLtApVLdTTrH-0MRUfh0owMTRbfQQd-so74/edit?usp=sharing

Live Demo

Robot + Power Delivery

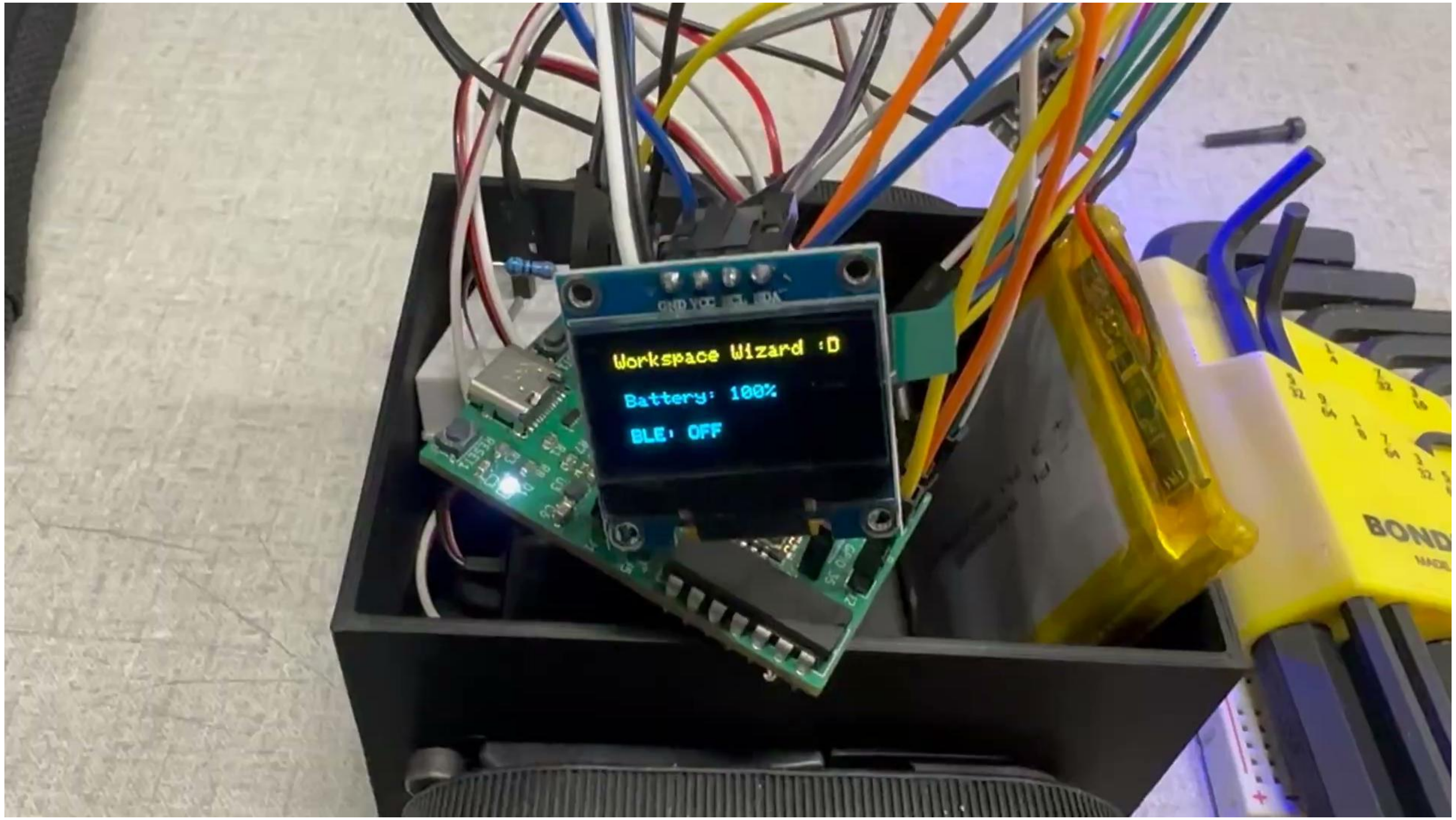
- System identifies and tracks fiducial markers.
- Robot receives **directional commands** based on initial position to navigate towards the target marker.
- Commands are transmitted via Bluetooth Low Energy.
- Robot interprets and executes movement commands (forward, backward, left, right) to approach the marker.
- System demonstrates responsiveness to marker placement, adjusting its path accordingly.

Overhead Computer + Software

- Demonstration of integrated image processing, object detection, robot detection, knolling algorithm, and path planning algorithm with live camera feed.

Thank you!
Any Questions?

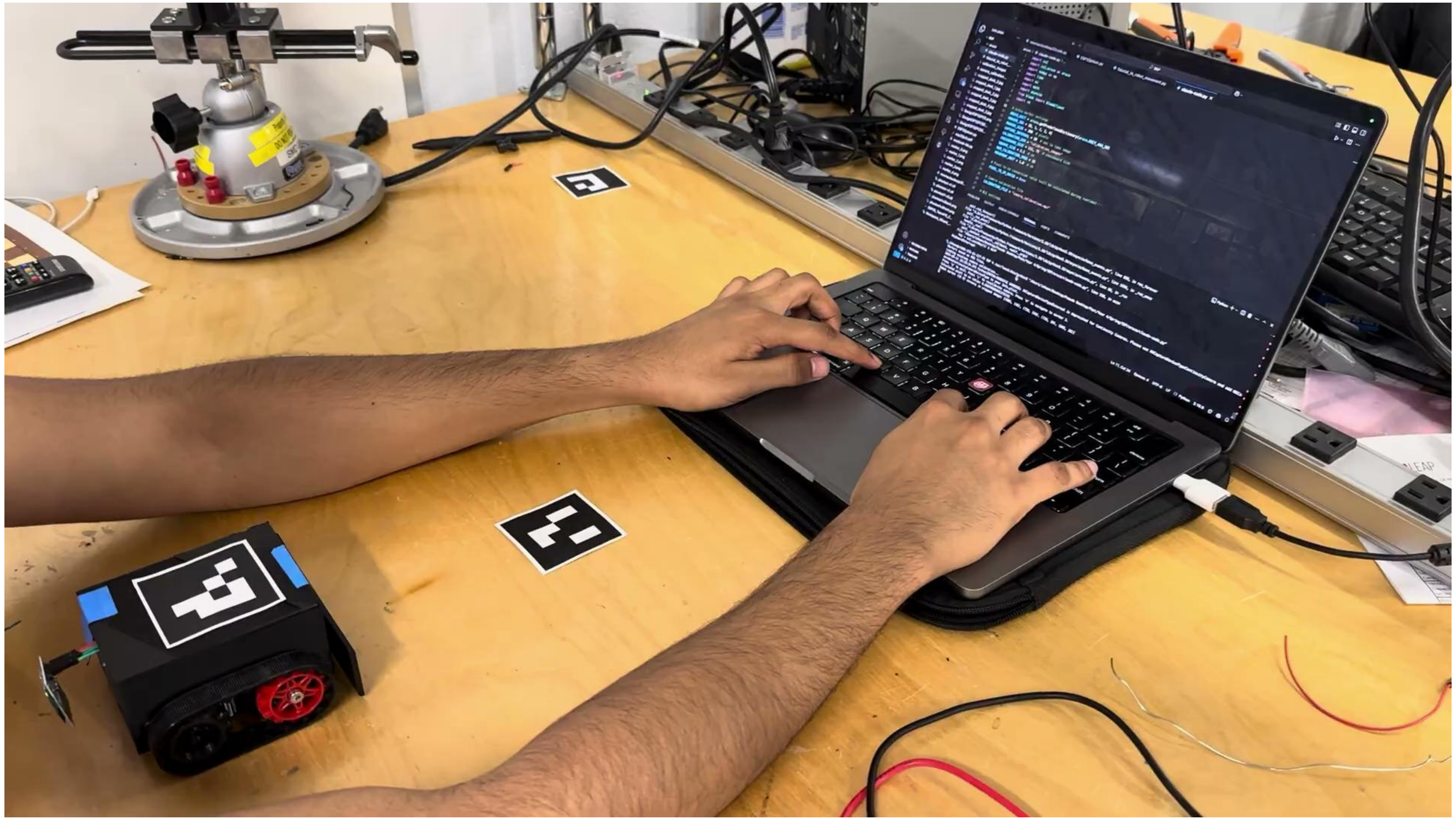
Demo Backup Video: BLE + Moving objects



- Python script connects to Robot, display updates.
- User inputs commands (eg: F,B,R,L).
- Commands are transmitted via Bluetooth Low Energy.
- Robot is able to push object without much error in movement
- Battery % and BLE status is always displayed
- Display can also show status like “finished”, or “stuck” for anyone near it

*Video opens in new browser tab, when using umass account

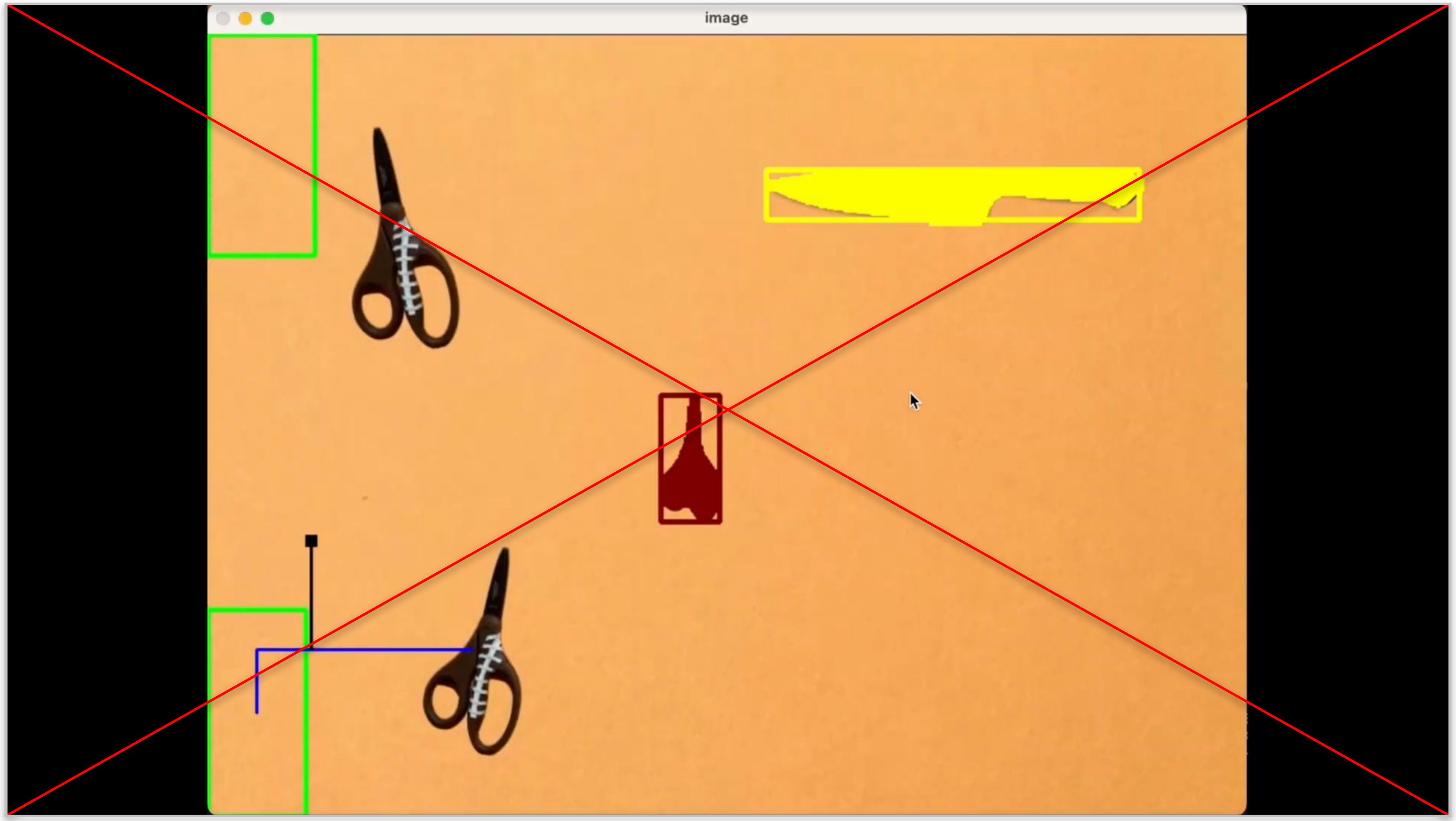
Demo Backup Video: Robot Path to Marker



- System identifies and tracks fiducial markers.
- Robot receives directional commands based on initial position to navigate towards the target marker.
- Commands are transmitted via BLE on custom python script.
- Robot interprets and executes movement commands (RIGHT, FORWARD, LEFT, FORWARD*14) to approach the marker.
- System demonstrates responsiveness to marker placement, adjusting its path accordingly.

*Video opens in new browser tab, when using umass account

Demo Backup Video: Knolling + Path Planning (Part 1)



Explanation on Next Slide

*Video opens in new browser tab, when using umass account

Demo Backup Video: Knolling + Path Planning (Part 2)

- Software identifies objects to be knolled (3 red objects) and prohibited items (1 yellow object)

- Software finds object path (in blue when available) from initial location to find location (in green bounding box) one object at a time.

- The robot commands for propelling object to final location are then provided. The robot is represented by a black square (sized down due to demo purposes and faster processing times). These steps are illustrated in the video in black straight line steps after each of which we plan on running error checking and correction by FPR.

- It may be noted that if an item is not boxed/highlighted in this video, it has been knolled and would be in the final green bounding box instead.

- **A similar demo was shown in CDR on live camera feed**, where the object and robot were moved to the new suggested location step by step, after which the camera updated the picture to account for the new object positions. It may also be noted that the objects and robot were moved by hand in the live demo to account for robot inaccuracy and current lack of error handling.

- The total recording time of the backup video is 13 minutes 53 seconds, which has been sped up in times when the path planning algorithm is being executed. The program ends upon completion.