UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

AutoServe

Automated Room Service Robot

Electrical & Computer Engineering

Team 84

Nikhil Vishnoi, Ethan Jiang, and Johan Martinez

04/30/2025

Sections

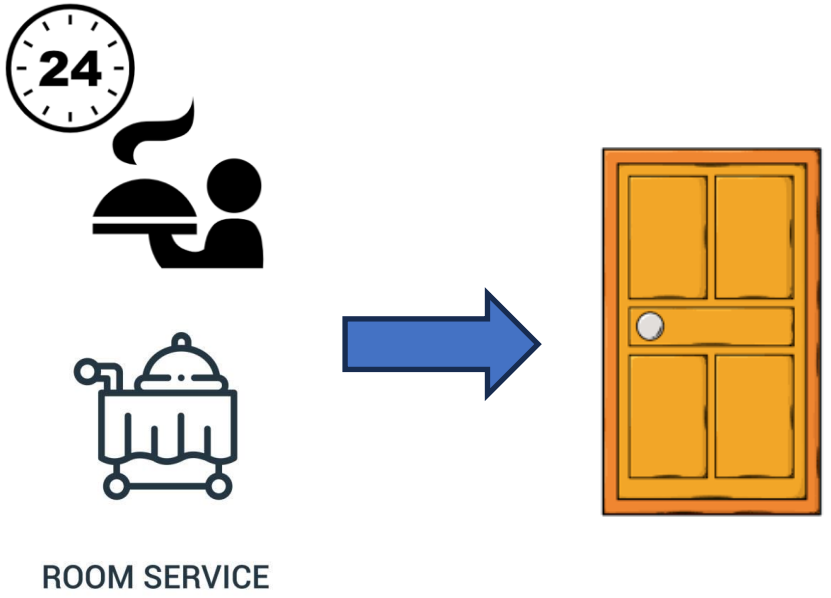
1. Introduction
2. Objectives
3. Initial Design
4. Final Design Overview
5. Tests & Verification
6. GUI
7. Issues Electrical/Mechanical
8. Conclusion & Future Work

Overview – Problem

Guests in hospitality-related environments frequently request low-value amenities (towels, toiletries, food items)

Operational impact:

- *Slow response times*
- *Increased labor cost*
- *Inefficient staff utilization*

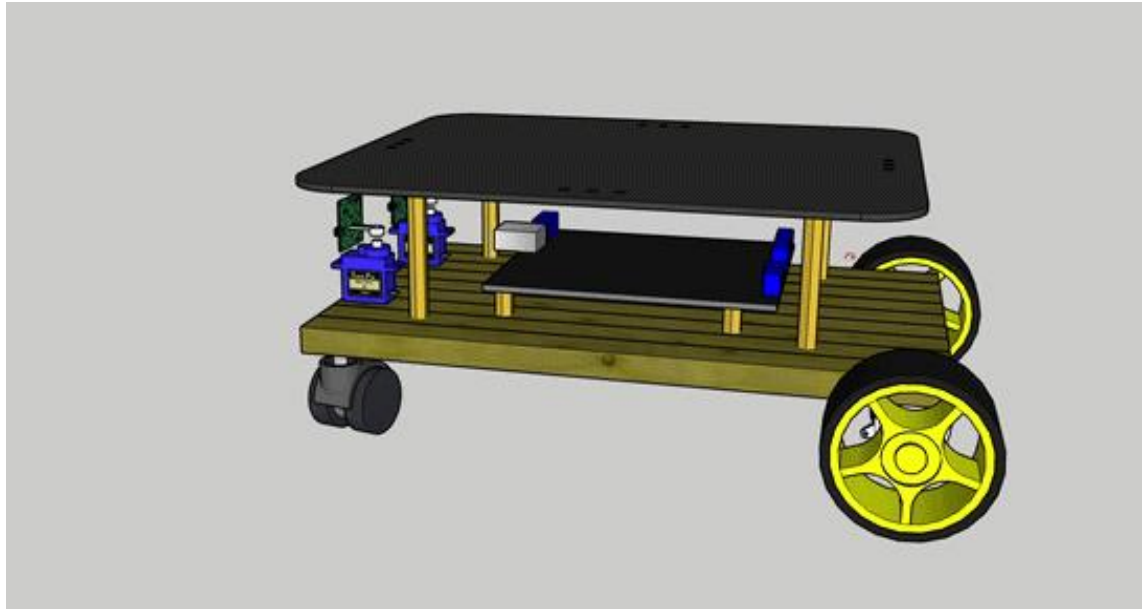


Repetitive tasks consume time that could be spent on higher-value service.

How can we fix this?

Our Solution

AutoServe – Autonomous Indoor Delivery Robot



Our Original 3d Model Mockup Sketch

Automates Indoor Delivery Tasks ✓

- No manual transport required

Autonomous Navigation ✓

- Reaches destination and returns to base

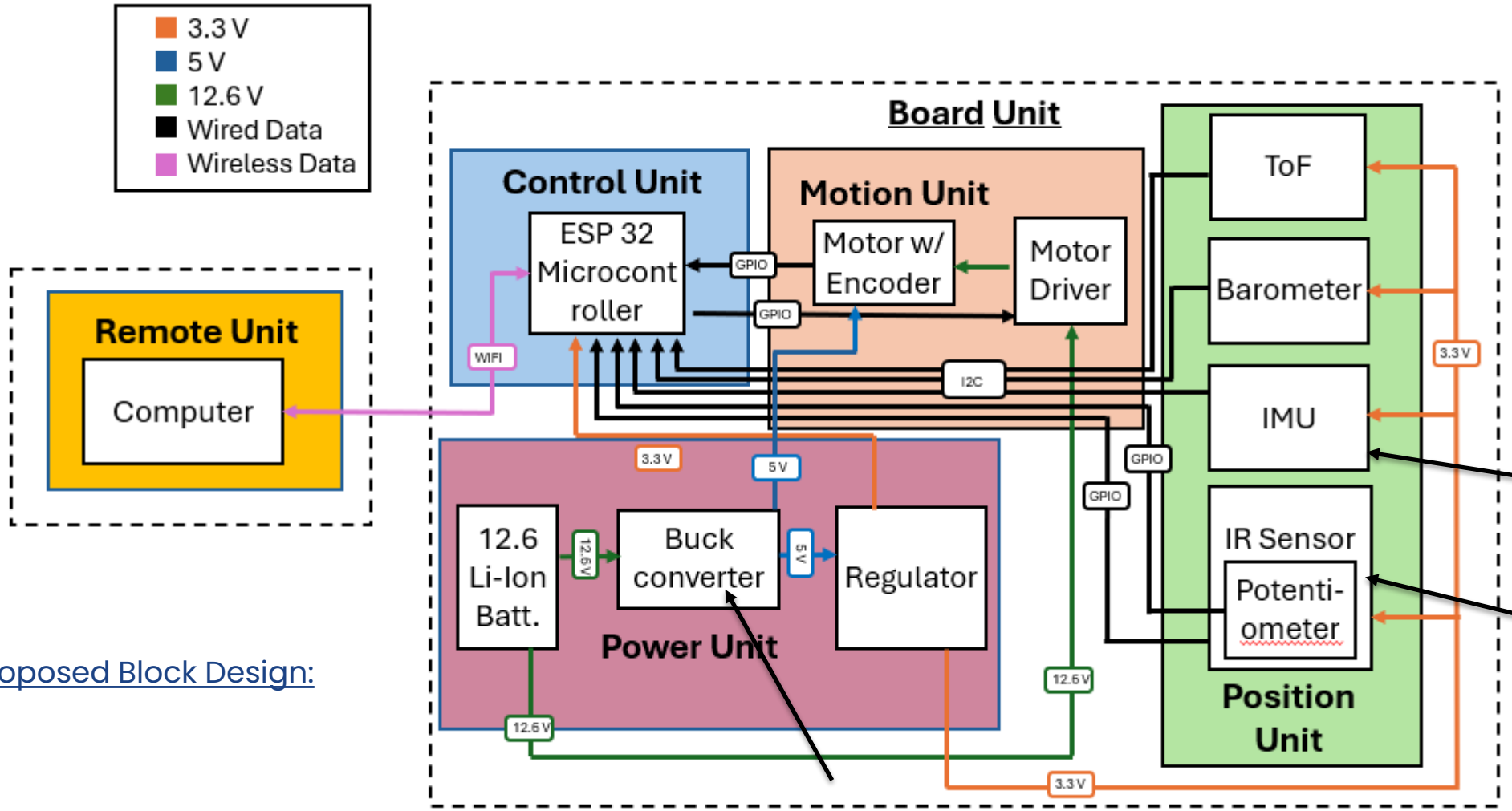
Integrated System ✓

- Wireless monitoring by staff
- Designed for multi-floor environments

Design Overview

- **Control Unit**
  - Controls main Board Unit and enables connection to a Remote Unit.
- **Remote Unit**
  - Python GUI to interact with the main board.
- **Sensor Unit**
  - Detect obstacles and stop Motion Unit from crashing the robot.
  - Track robot spatial position.
- **Motion Unit**
  - Drive the motors using encoder feedback.
- **Power Unit**
  - Maintain different voltage levels for different components (12V, 5V and 3.3V).

Design Overview – Initial Design



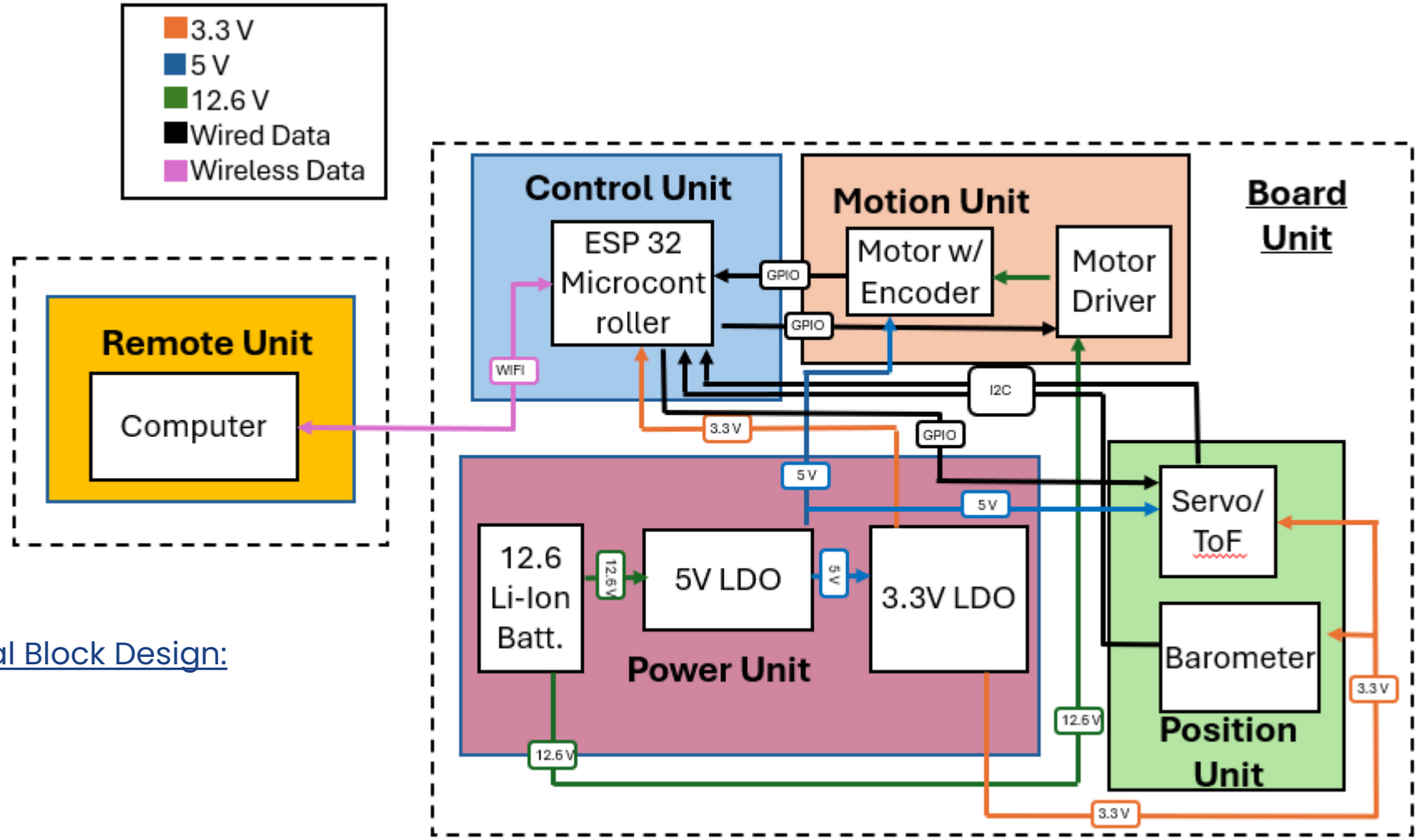
Did not work on PCB

Poor performance on breadboard

Initial Proposed Block Design:

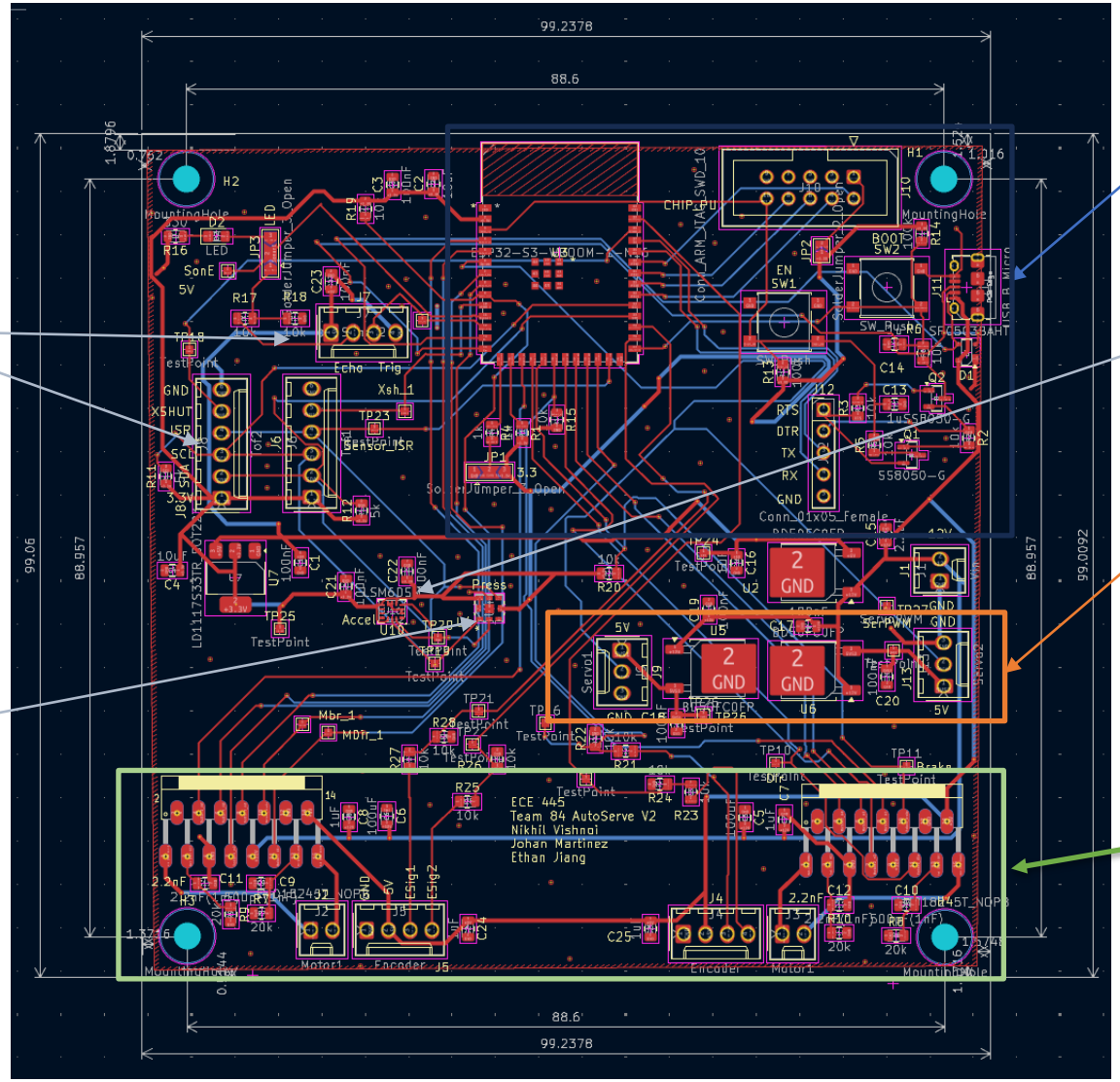
Did not need high current on 5V line

Design Overview – Final Design

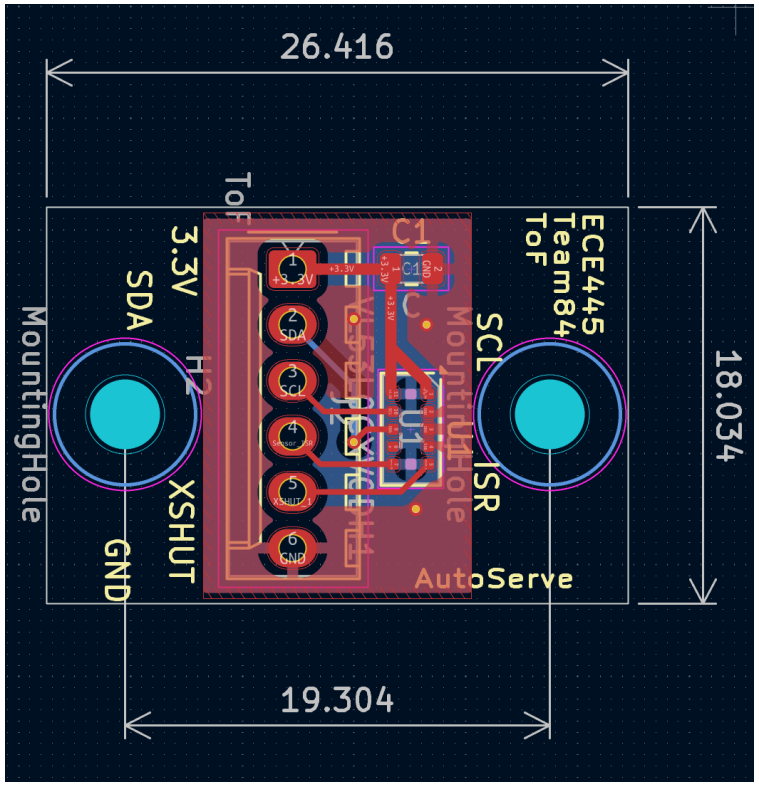


Final Block Design:

Hardware

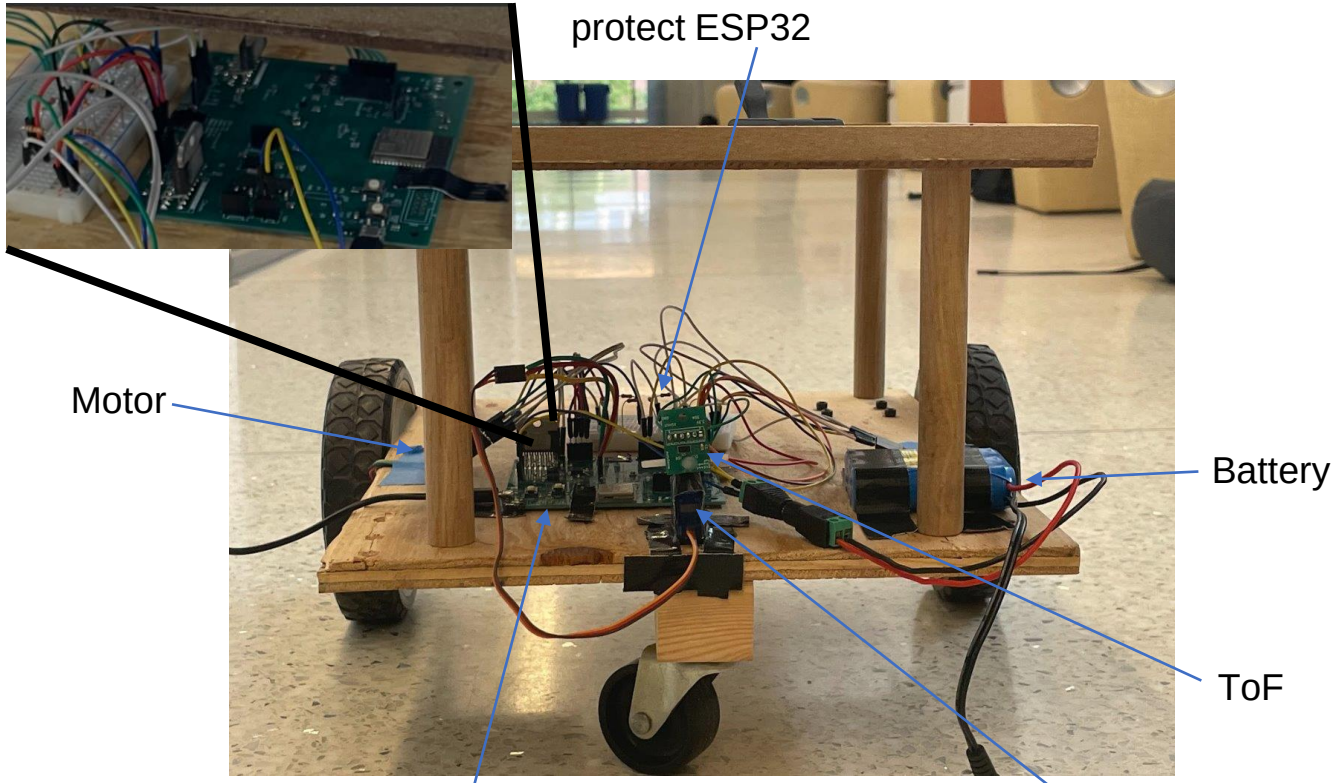


Main PCB



Time of Flight (VL53L0X) Breakout

Final Design



Front View

Top-Down View

Motion

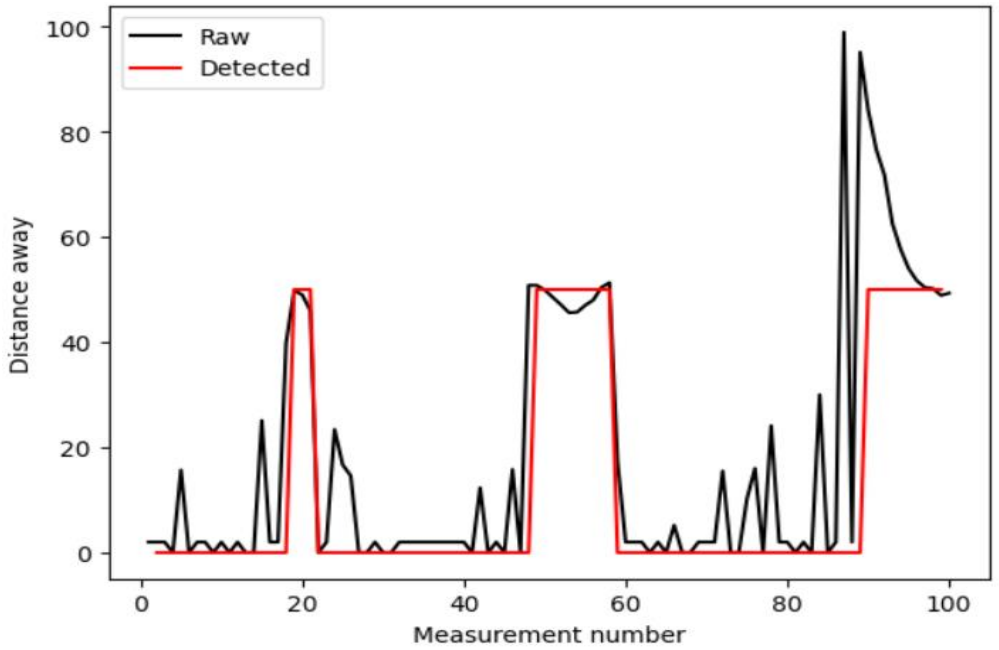
2 Brushless Motors - Separate Drivers

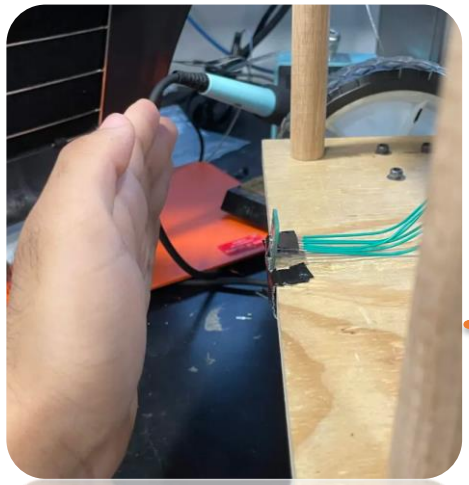
- 1 Caster Wheel
- PWM on Brake
- Motor Encoders for Control

```
> [MOTOR] FWD  36.0 deg  M1=150 M2=150
[FWD]  target=36.0 deg, step=10.0 deg, spd M1=150 M2=150
[STEP]  spinup wait 111ms (slower_spd=150)
[STEP]  ISRs armed  M1:ACTIVE M2:ACTIVE  target=15 pulses (10.0 deg)  spd1=150 spd2=150
[STEP]  stall windows  M1:80ms  M2:80ms
[STEP]  M1 stall-chk  abs=12 snap=0 delta=12 min=3
[STEP]  M2 stall-chk  abs=13 snap=0 delta=13 min=3
[STEP]  M2 TARGET  count=15 abs=15 target=15  t=80ms
[STEP]  M1 TARGET  count=15 abs=15 target=15  t=90ms
[STEP]  DONE  target=15  M1=15(err+0)  M2=17(err+2)  t=90ms
[FWD]  step 1 done (10.0 deg)
[STEP]  spinup wait 111ms (slower_spd=150)
[STEP]  ISRs armed  M1:ACTIVE M2:ACTIVE  target=15 pulses (10.0 deg)  spd1=150 spd2=150
[STEP]  stall windows  M1:80ms  M2:80ms
[STEP]  M2 TARGET  count=15 abs=15 target=15  t=50ms
[STEP]  M1 TARGET  count=15 abs=15 target=15  t=60ms
[STEP]  DONE  target=15  M1=15(err+0)  M2=17(err+2)  t=60ms
[FWD]  step 2 done (10.0 deg)
[STEP]  spinup wait 111ms (slower_spd=150)
[STEP]  ISRs armed  M1:ACTIVE M2:ACTIVE  target=15 pulses (10.0 deg)  spd1=150 spd2=150
[STEP]  stall windows  M1:80ms  M2:80ms
[STEP]  M2 TARGET  count=15 abs=15 target=15  t=40ms
[STEP]  M1 TARGET  count=15 abs=15 target=15  t=50ms
[STEP]  DONE  target=15  M1=15(err+0)  M2=17(err+2)  t=50ms
[FWD]  step 3 done (10.0 deg)
[STEP]  spinup wait 111ms (slower_spd=150)
[STEP]  ISRs armed  M1:ACTIVE M2:ACTIVE  target=9 pulses (6.0 deg)  spd1=150 spd2=150
[STEP]  stall windows  M1:80ms  M2:80ms
[STEP]  M2 TARGET  count=9 abs=9 target=9  t=20ms
[STEP]  M1 TARGET  count=9 abs=9 target=9  t=20ms
[STEP]  DONE  target=9  M1=9(err+0)  M2=11(err+2)  t=20ms
[FWD]  step 4 done (6.0 deg)
[FWD]  complete — 4 steps
[MOTOR] Done.
```

ToF Sensor

- Mounted Position:
  - Front-middle
- Consistent readings up to ~30 readings per second.
- Was mounted on a servo to sweep readings across a broad angle scan.

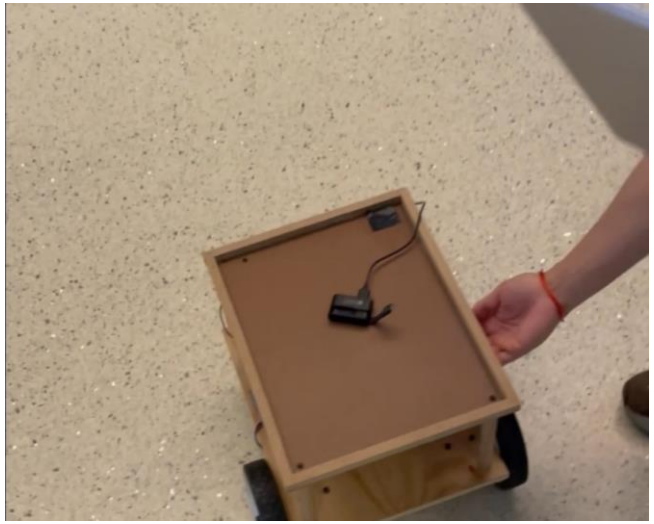


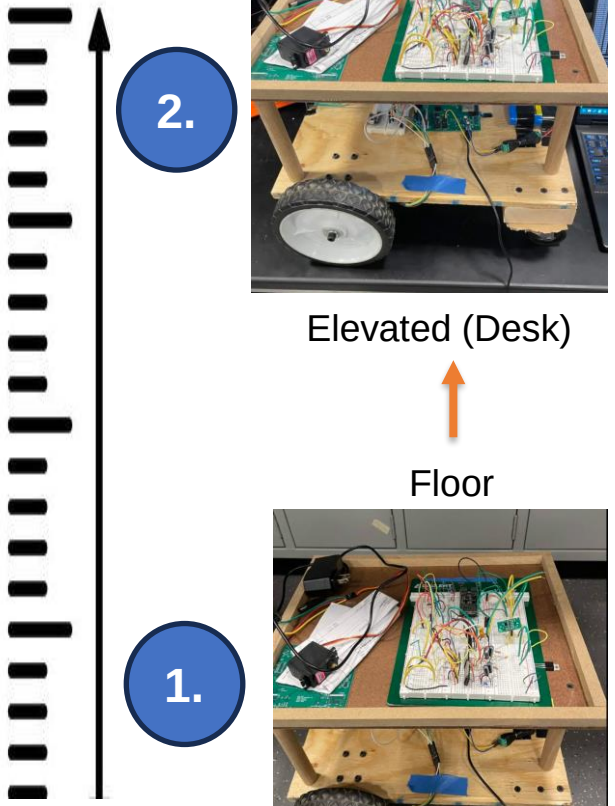


```
[TOF] 849 mm  (84.9 cm)
[TOF] 824 mm  (82.4 cm)
[TOF] 765 mm  (76.5 cm)
[TOF] 793 mm  (79.3 cm)
[TOF] 105 mm  (10.5 cm)
[TOF] 108 mm  (10.8 cm)
[TOF] 112 mm  (11.2 cm)
[TOF] 112 mm  (11.2 cm)
[TOF] 103 mm  (10.3 cm)
```

Height Sensor

- BMP Pressure sensor
  - Obtain height data
- It allows robot to detect several floors
  - Different maps





Elevated (Desk)

Floor


```
[ELEVATION] SAME | 0.8m | 981.15 hPa | 29.4C
[ELEVATION] SAME | 0.8m | 981.14 hPa | 29.4C
```

Desk Height

Floor Height

```
[ELEVATION] SAME | 0.0m | 981.24 hPa | 29.4C
[ELEVATION] SAME | 0.0m | 981.24 hPa | 29.4C
```

Baseline pressure set (floor)

```
> [ELEVATION] Setting baseline pressure...
[ELEVATION] Baseline: 981.24 hPa  Temp: 29.4C — monitoring...
```

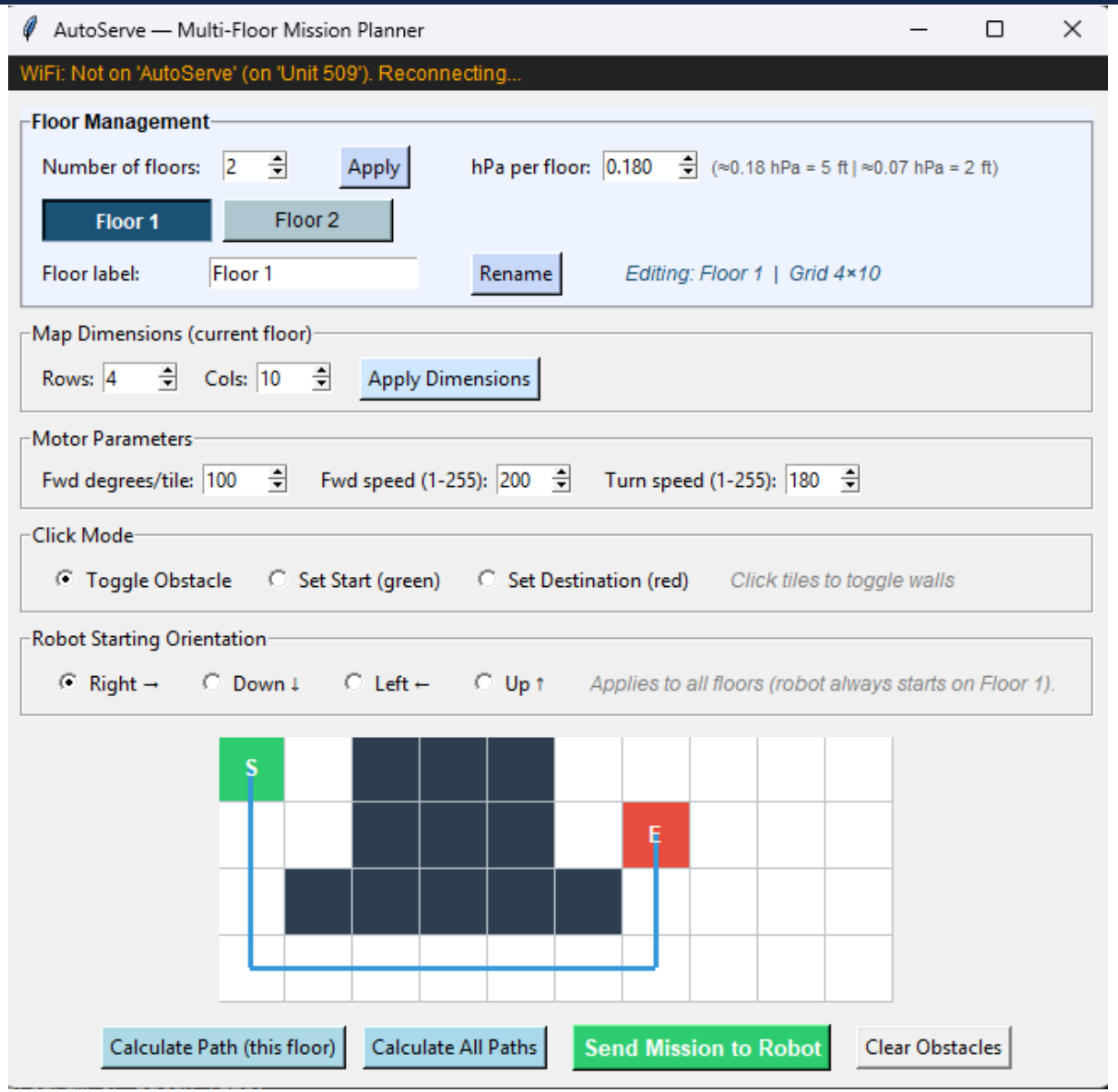
Real Time Operating System

- Each sensor/component uses its own thread waiting for broadcast.

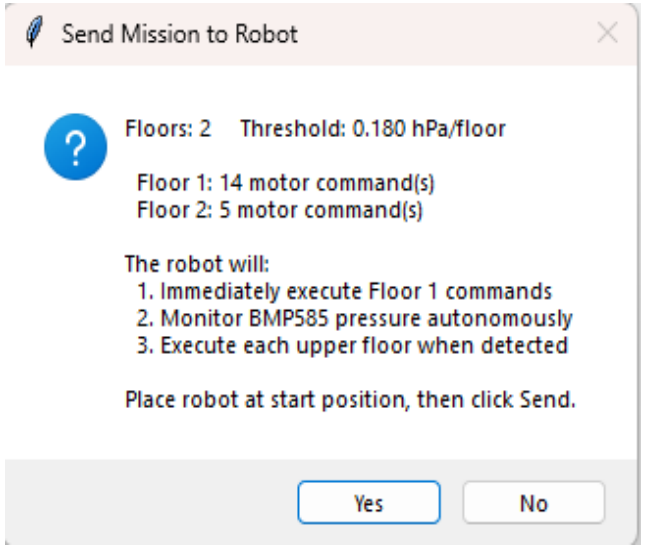
- Main listens to commands from telnet server: 192.168.4.1:23
  - Breaks up commands into steps
  - Broadcasts to wake up threads to make robot do different things concurrently

- Benefits:
  - Simpler code structure for each component (avoid one loop)
  - Allows components to start up other components and wait on their result (ex. the motor component calls the ToF to interrupt motion)

User Interface



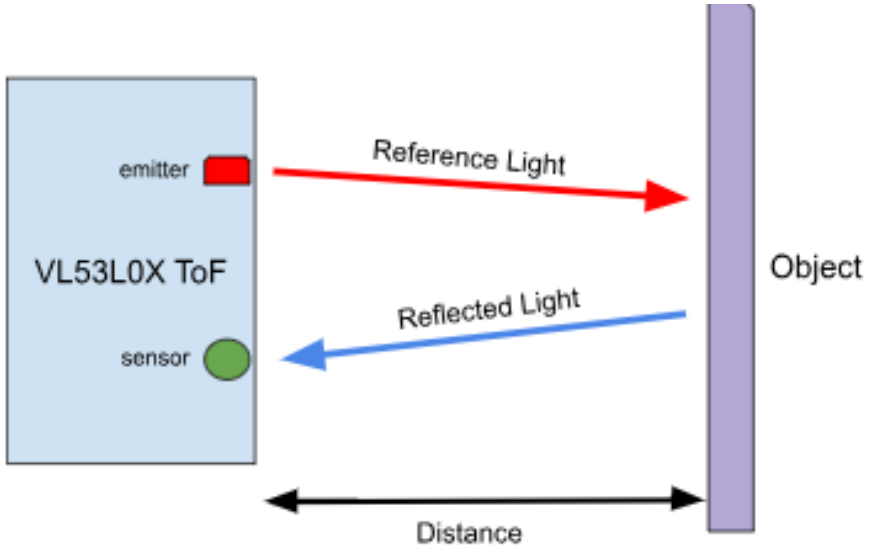
- Custom maps
- A* pathfinding
- Speed/Threshold Customization




```
Commands to send:
  fwd 10 150
  fwd 10 150
  fwd 10 150
  fwd 10 150
  -> sending: fwd 10 150
  done: fwd 10 150
  -> sending: fwd 10 150
  done: fwd 10 150
  -> sending: fwd 10 150
  done: fwd 10 150
  -> sending: fwd 10 150
  done: fwd 10 150
```

Terminal Output

Safety Measures

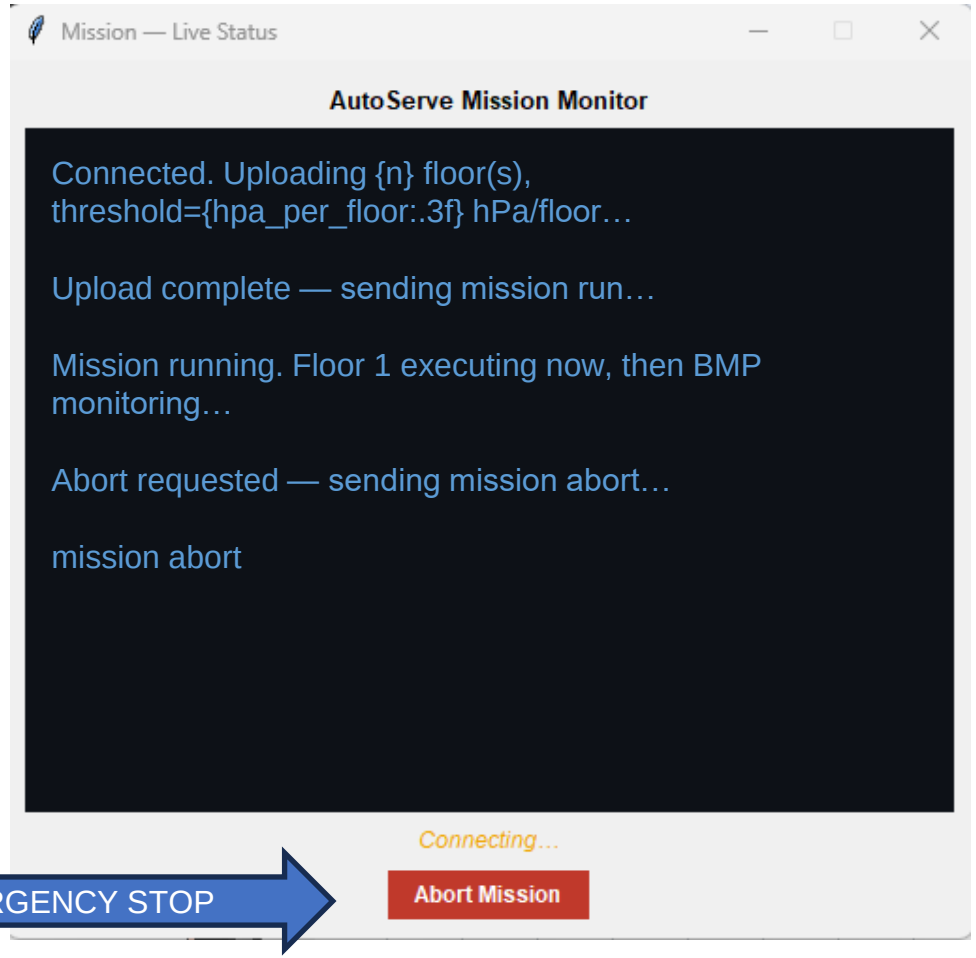


ToF Sensor

Obstacle Detection

“Halt Within Range”

Abort Mission Button



EMERGENCY STOP

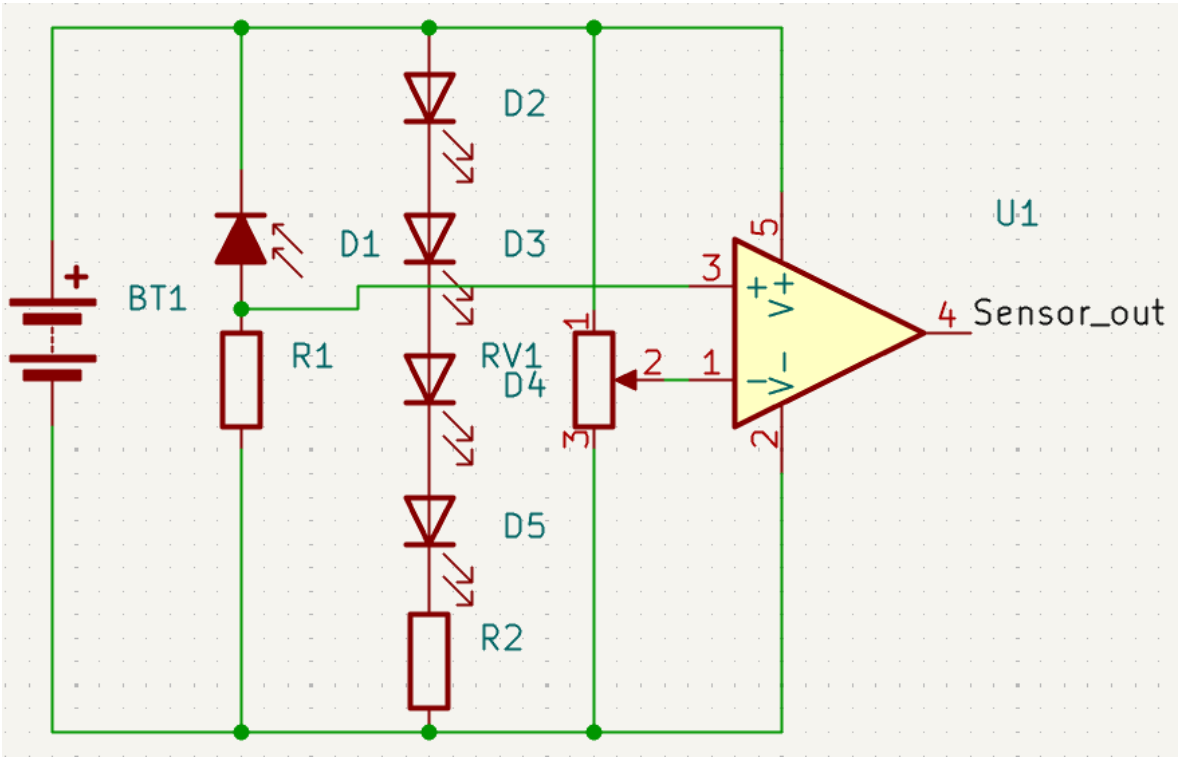
Challenges

Mechanical:

- Uneven wheel height
- Wheel inertia caused drifting

Sensing:

- Weak and noisy IR sensor signal ~ 3 mV
- Sonar could not detect fabric – bad for detecting people



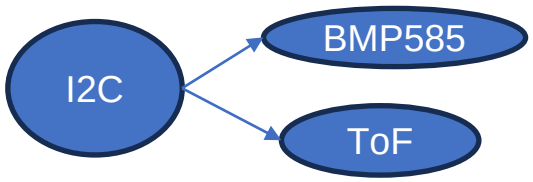
IR sensor circuit diagram

Conclusion

What We Learned

System Integration

•Design custom PCB with MCU control
•Interfaced multiple sensors



Motor Drivers
Servos
Encoders
(Etc.)

Software & Communication

•Remote interface Python + telnet
•Real-time wireless monitoring and control

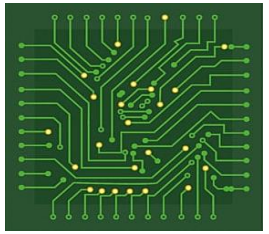
A*

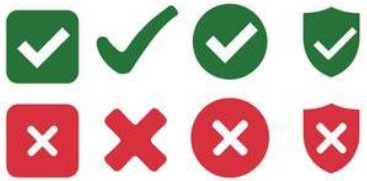
Real Engineering Process

What We Would Do Differently

Hardware Design

•Select higher-torque motors
•Design/Component Oversight
•Allocate more time for PCB validation and testing





Debugging

•Investigate Mechanical Issues
•Time Management for PCB design/integration

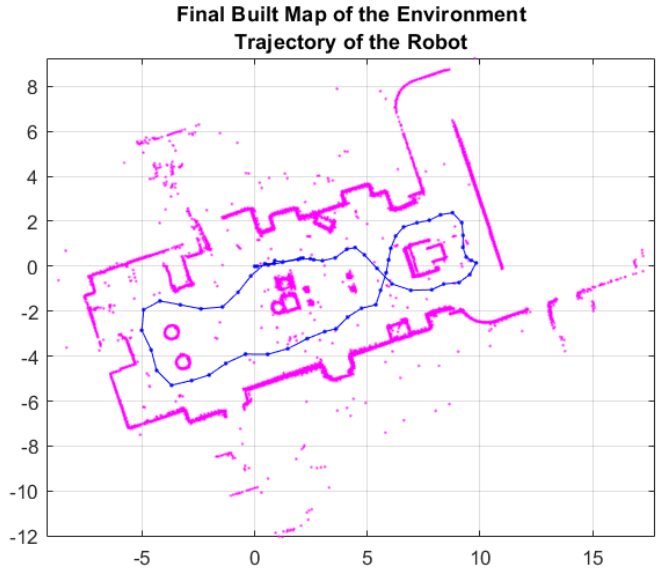
Future Work

- **Motors**
  - Better motors and motor drivers
  - Improve mechanical problems
  - Improve software control to allow for smoother motion
  - Position detection

- **UI**
  - Improved GUI features
  - Mobile app integration, voice control, etc.

- **LIDAR**
  - Spinning lasers to expand environmental data
  - Enable more powerful algorithms

- **SLAM**
  - Real time mapping
  - Re-pathing calculations when blocked



Example of Implement Simultaneous Localization And Mapping (SLAM) with Lidar Scans from MATLAB & Simulink.

Thank You!
Team 84 AutoServe

Questions?