# ECE 445

# Senior Design Lab

# Notebook

Fall 2024

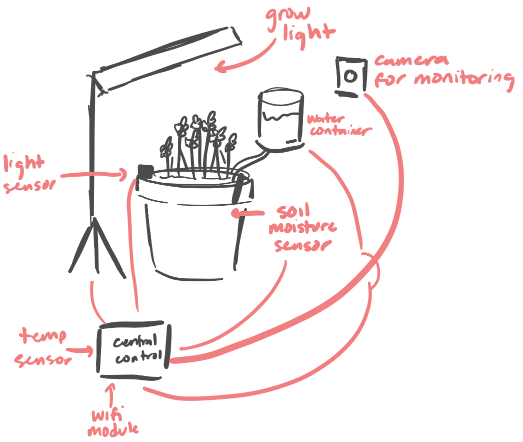
Matthew Paul

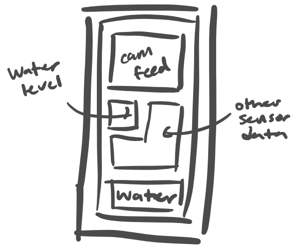
TA: Dongming Liu

Team 34

# 9/10 Ideation

## Remote Plant Monitoring System




- notifications when conditions are above/below threshold
  - water tank level, temp, moisture
  - ↳ can configure to auto water plant

**Problem:** Taking care of plants when away from home (Vacation, college, etc)

**Features:**

- toggleable grow light
- automatic watering system (schedule or directly in app)
- light, moisture, and temp sensors
- phone app for easy access → can potentially be used in conjunction with smart home system for temp control
  - ↳ cam feed

* requires someone with access to setup to refil water every so often

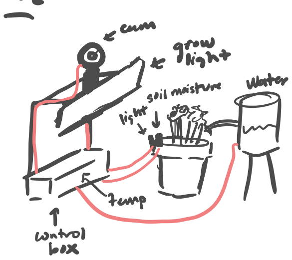
**Possible Features:**

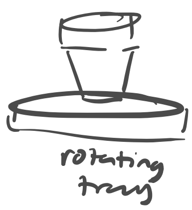
- multiple sets of sensors for different plants
- easy to take on and off of current setup

* 3 high level requirements

1) Maintain moisture level
2) remote access to sensor data and camera
3) maintain light level throughout the day

$V_2$

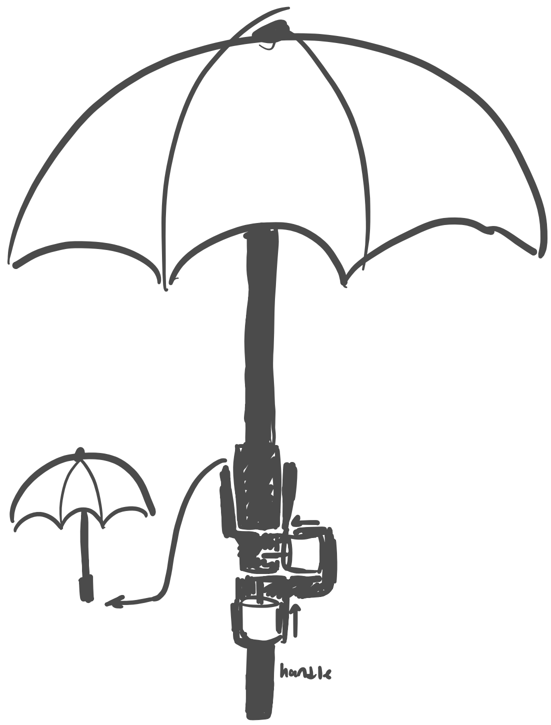



## PCB

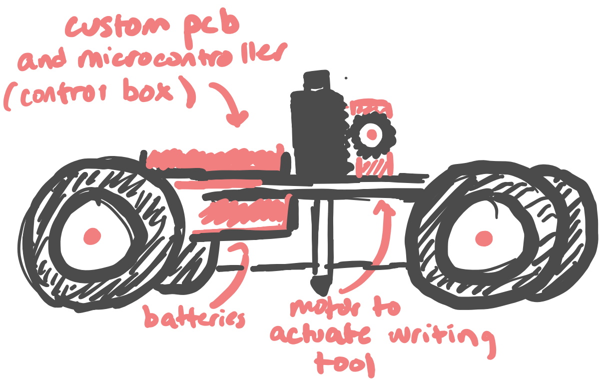
need
- microcontroller → STM popular
- dont make rasp pi main component
  - sensors go through pcb/microcontroller

Umbrella that rotates with wind

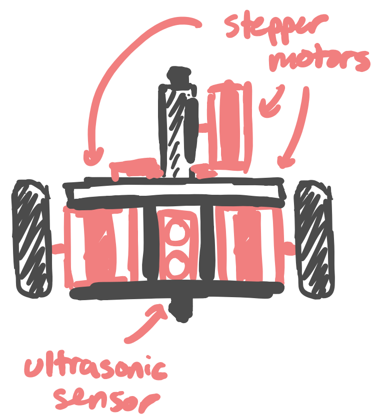


9/12 Initial Design for Proposal 

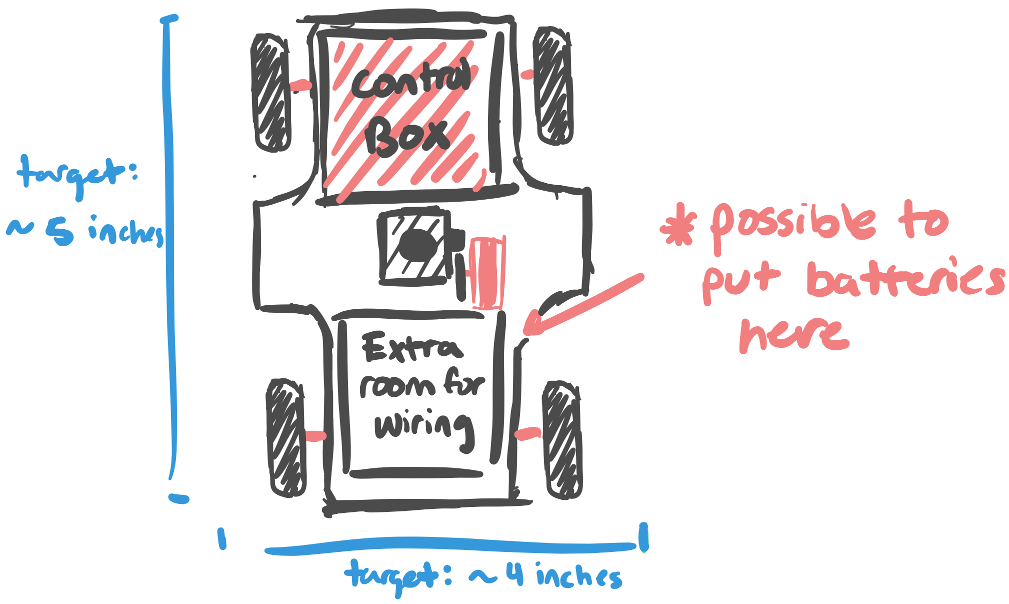
## Side View

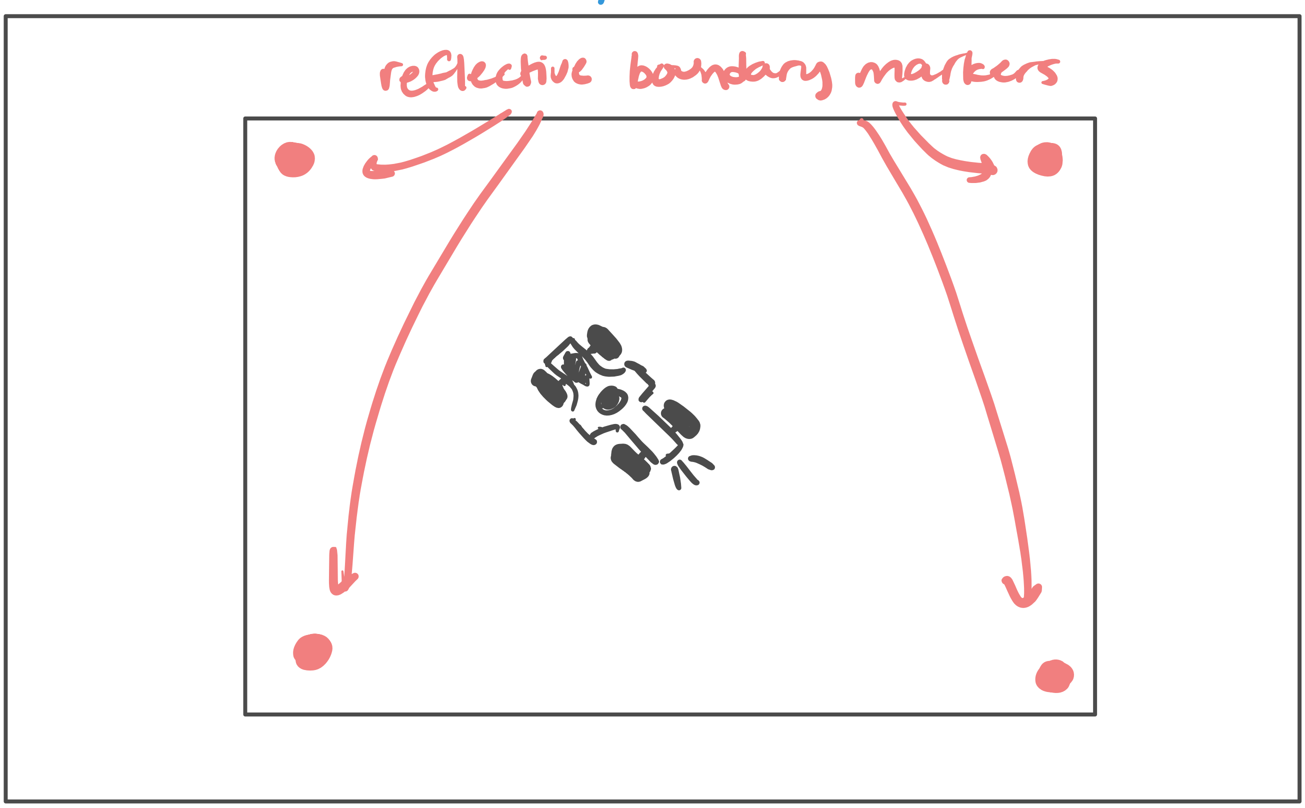



## Front View




## Top View



9/17 First Team Meeting

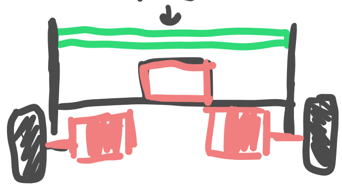
Top View

omniwheel

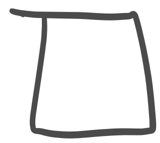
sensor

Side View

PCB

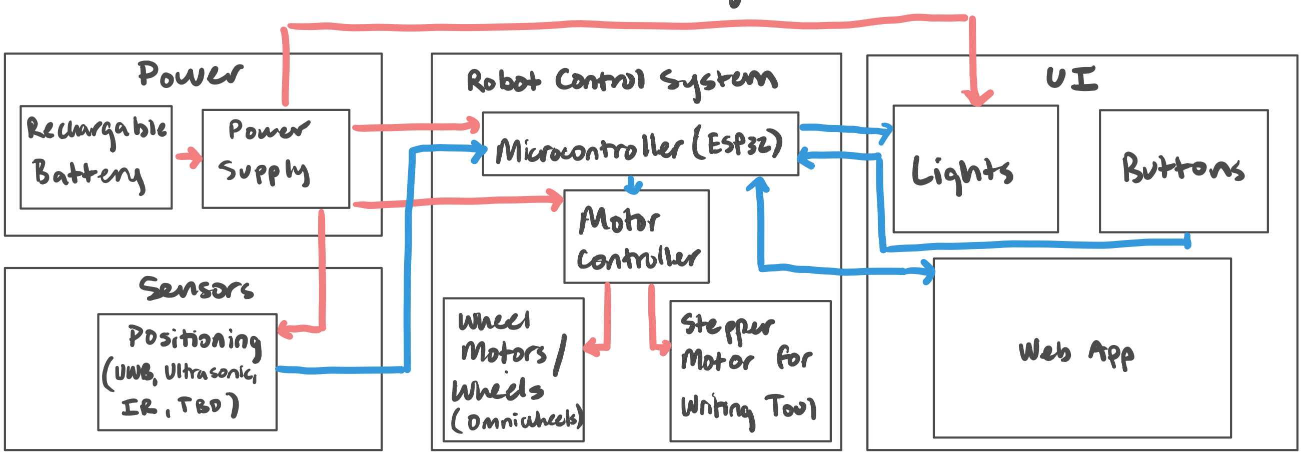


10cm



10cm

## Block Diagram




Power
Data

* ask about multiple pcbs

Fall 2021 Team 1

motor with feedback encoder *

photonic sensor

brushless - fine control

expensive ↳ hard driver

* no immediate feedback

maybe 6V

↳ go for 12V

- battery 12V

Camera

QR cod

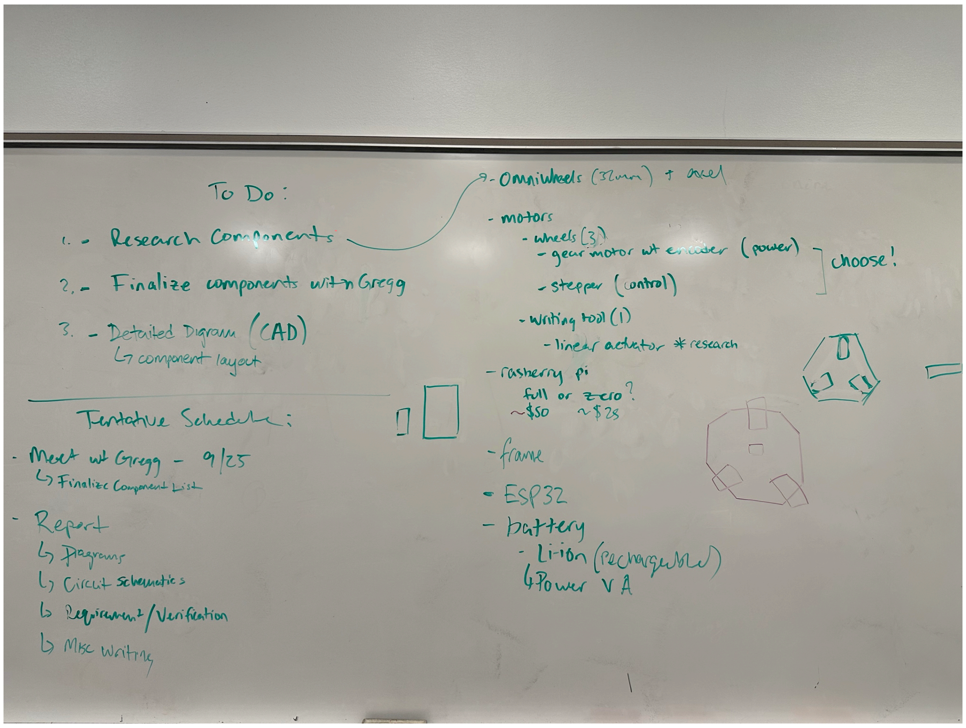
Rasppi

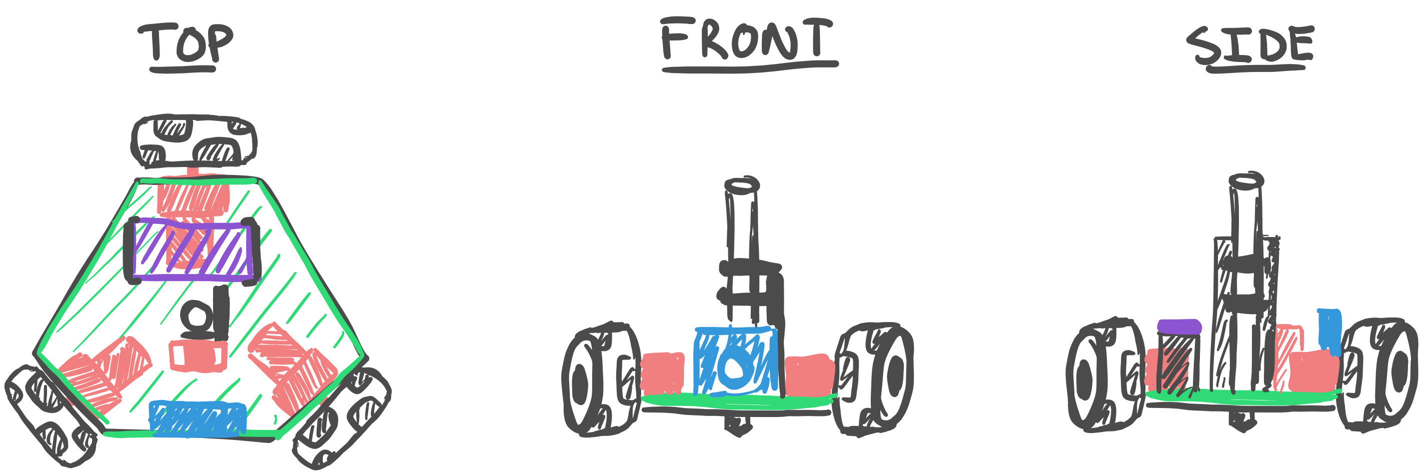
dev board / breadboard

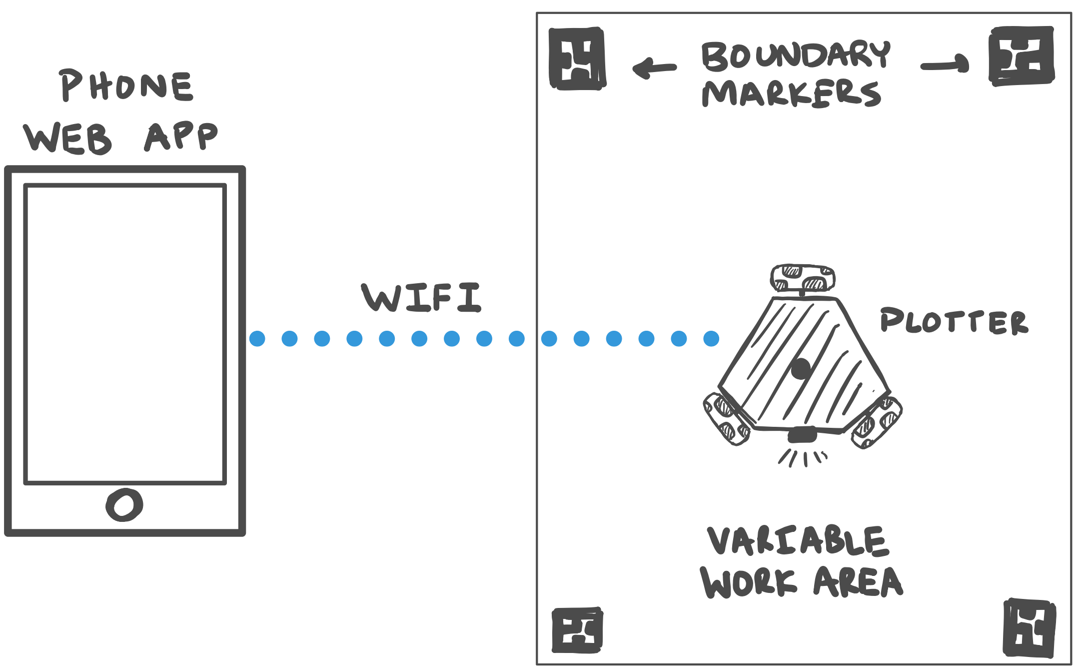
Team 34:

| F5 | 55 | 91Y23 | 49-21-41 |
| --- | --- | --- | --- |
| locker # | kit # | | combo |

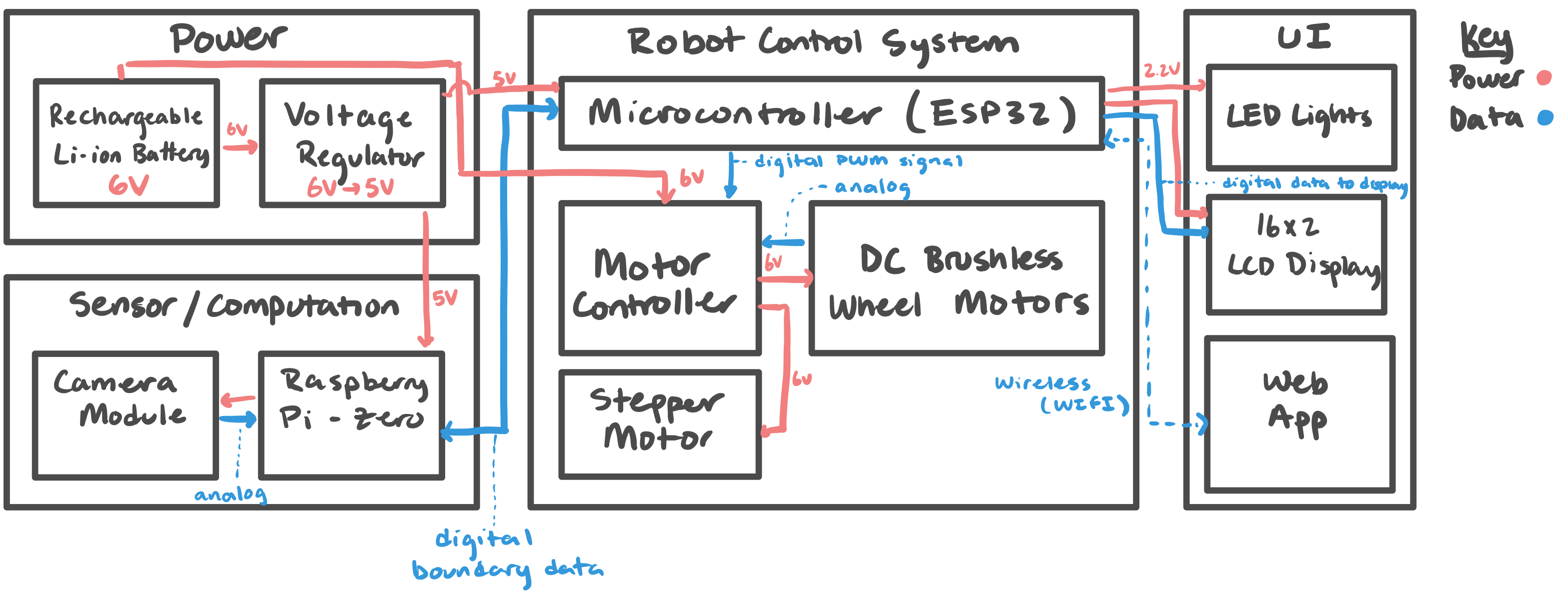
9/24 Brainstorming for Design Doc 



9/27 Block Diagram Rough Draft 




DC 12V

1 : 100

Gear motor

Gear box

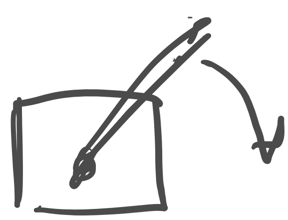
with encoder

stepper motor

omniwheels on us

linear for

mounting hubs



# 9/17 TA Meeting

## Design Doc Feedback

* look at datasheets for components

- Problem ✓
- Solution ✓
- Visual Aid ✓
- High level ✓
- Block Diagram ✗
    - label all lines took of
    - use software
- subsystem overview ✓
- sub system requirement ✓
- Tolerance Aanalysis
    ↳ proof that our design will work within tollerancy
        ↳ - motor torque calculations (load)

include few for
- battery powerful enough
- direation calculations

-1.5

rasp pi head
vs
esp 32

AMU inertia unit

for heading

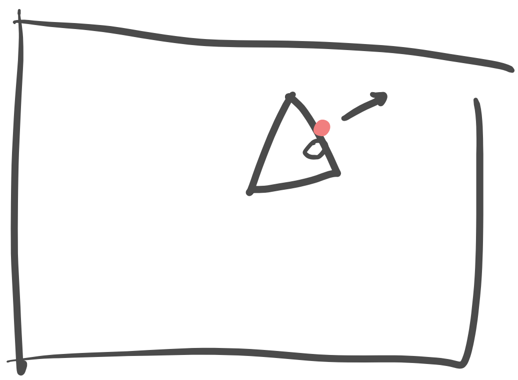
Ali expresss

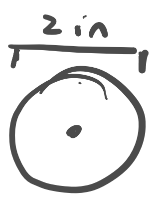
Ethics / Saftey
- Workes cited (Ref page)

SP 2024 Team 19 - TAs
FA 2023 Team 3 - Good example

9/25 Talk with Machine Shop

- provide dimensions / schematics
- give motors/wheels
- calculate needed motor specs






$2\pi r = 6.28$ in

150 rpm → $150 \cdot 6.28 = 942$ in/min

$= 15.7075$ in/sec

10/1 TA Meeting 3



* Sign up for peer review by tomorrow
* AMU inertia unit

Design Doc Rubric

R.V. - :(

Schematic (not perfect)

Complexity

C# for ESP 32

→ Tolerance

servo motor

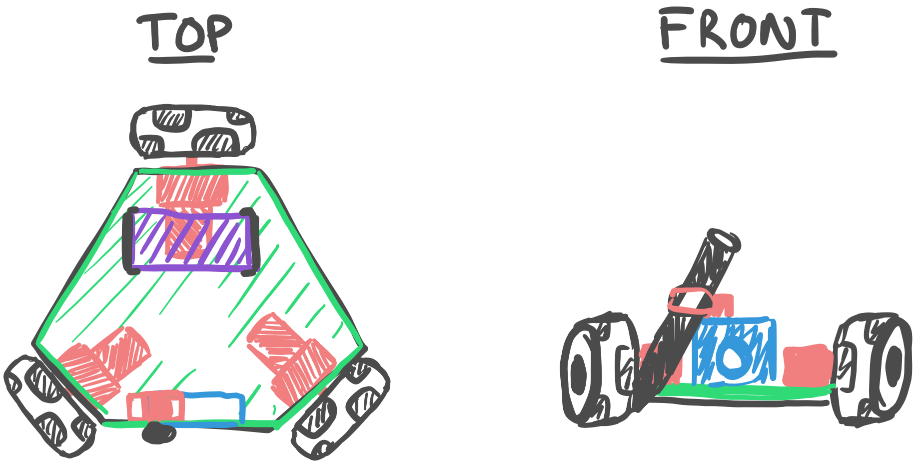
torque

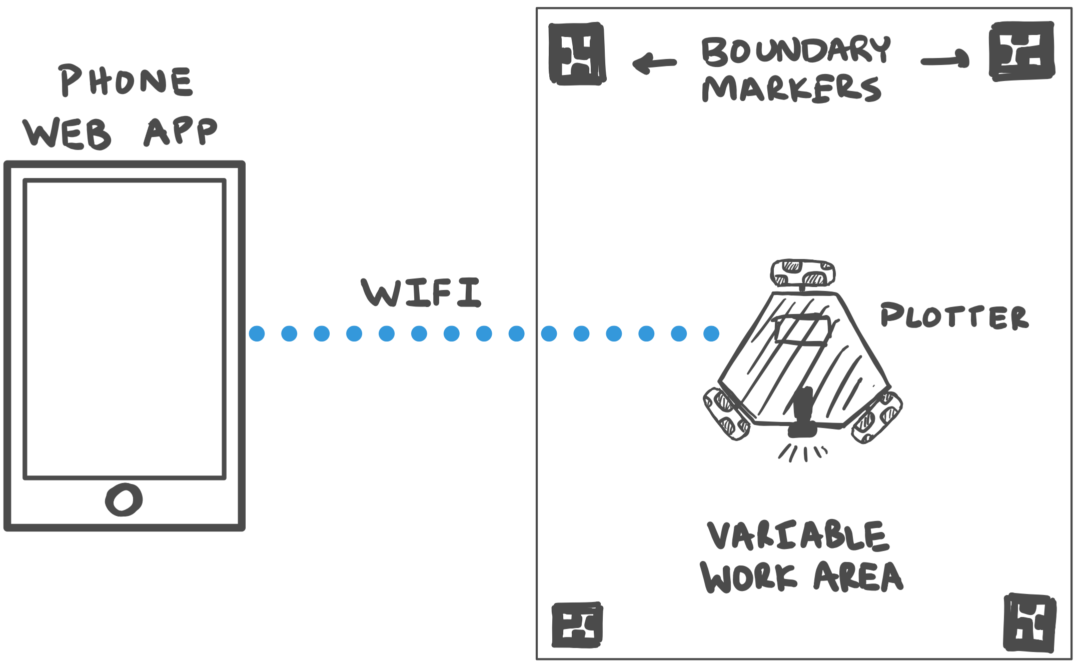
Design Review

Slide show

Regrade Friday

10/3 Updated Graphics for Design Doc

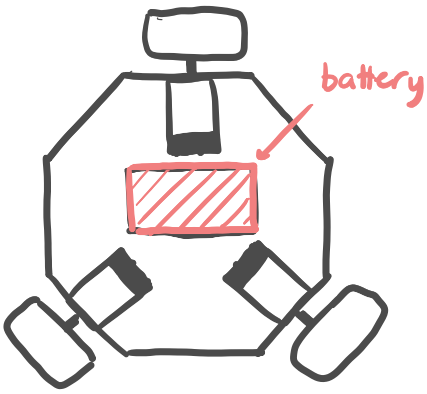


10/3 Brainstorming Workflow



Bottom View

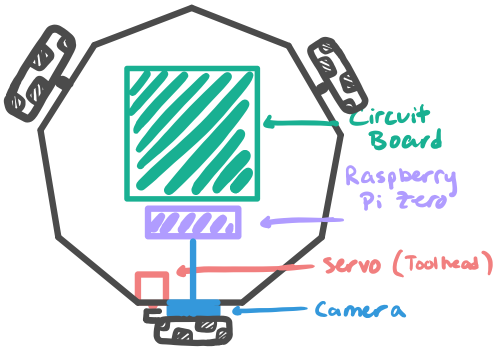



- Detailed Design
- Flesh out Software

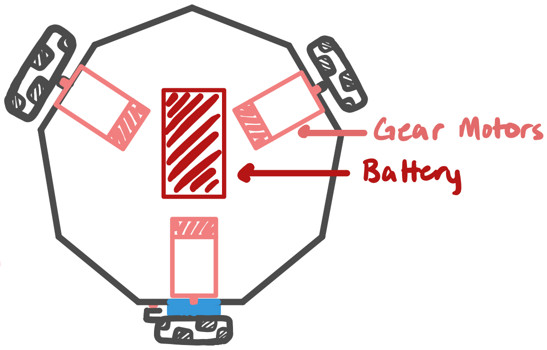
✓ add IMU to diagram and schematic

10/7 Updated desing for Design Review

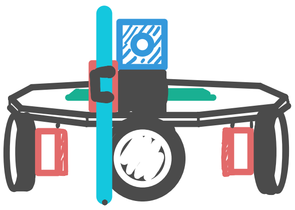
Top




Bottom




Side



10/11 PCB Review / Calculations 

# TPS62A01DRLR

*C3 optional

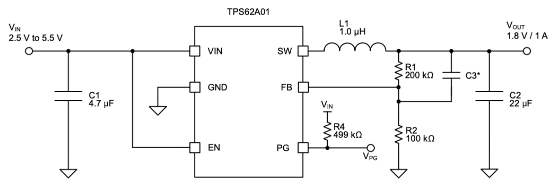



**Figure 8-1. TPS62A01 Typical Application Circuit**

$V_{out} = 3.3V$

3-3.6

The output voltage is set by an external resistor divider according to Equation 2.

$$R1 = R2 \times \left(\frac{V_{OUT}}{V_{FB}} - 1\right) = R2 \times \left(\frac{V_{OUT}}{0.6\,V} - 1\right)$$

$R_1 = R_2 \times \left(\frac{3.3}{.6} - 1\right) \Rightarrow R_1 = 4.5 R_2$

R2 must not be higher than 100kΩ to provide acceptable noise sensitivity. → $R_2 = 100k\Omega$

↳ $R_1 = 450k\Omega$

$10 = 2.2 \times \left(\frac{V_{out}}{.6} - 1\right)$

$R_1$ =

$R_2$

**Table 8-3. Matrix of Output Capacitor and Inductor Combinations for TPS62A01 and TPS62A02**

| $V_{OUT}$ [V] | L [µH][1] | $C_{OUT}$ [µF][2] | | |
|---|---|---|---|---|
| | | 10 | 22 | 2 × 22 |
| $0.6 \le V_{OUT} < 1.2$ | 1 | | + | ++[3] |
| $1.2 \le V_{OUT} < 1.8$ | | | ++[3] | + |
| $1.8 \le V_{OUT}$ | | +[4] | ++[3] | + |

standard

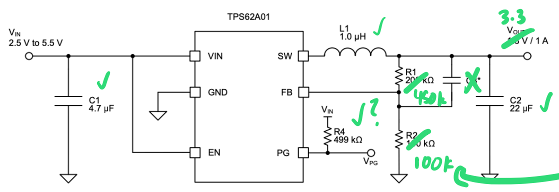
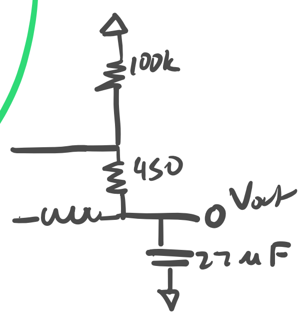
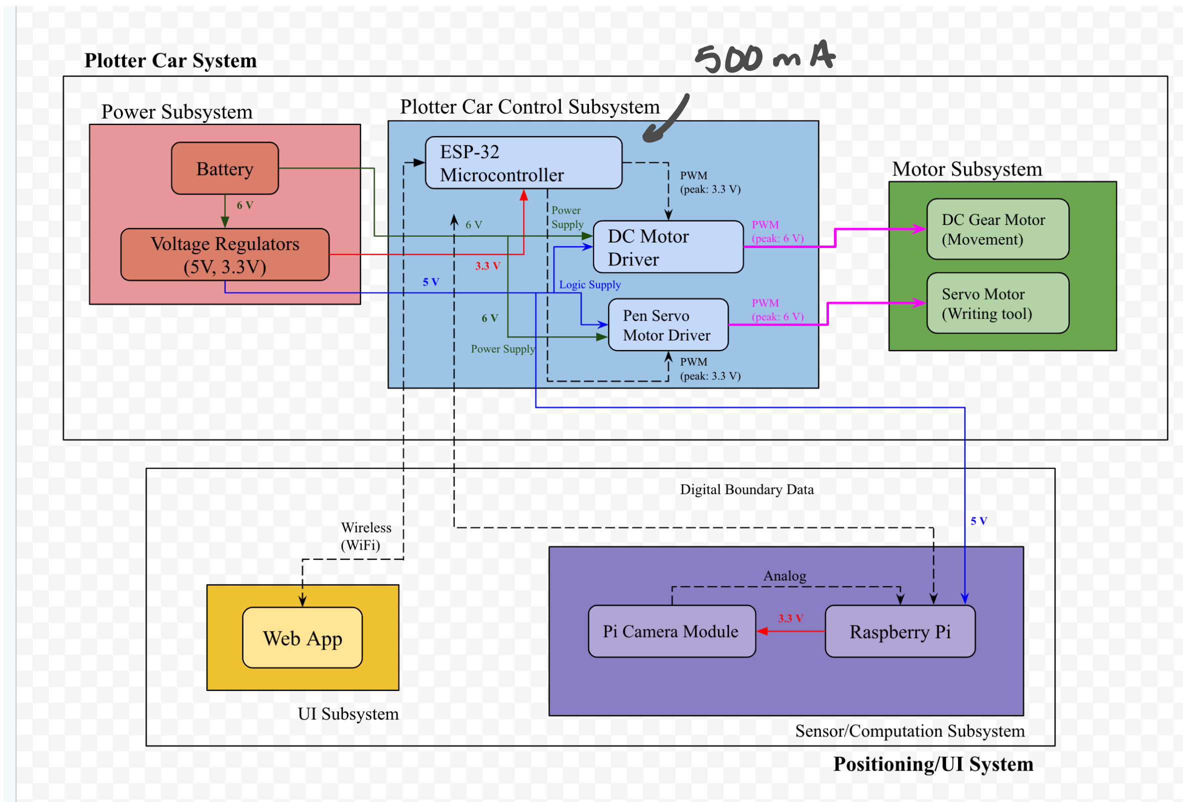



**Figure 8-1. TPS62A01 Typical Application Circuit**




ESP32 300mA



10/15 TA Meeting

- multiple orders are fine
- work on finishing PCB before anything else
- Start work on dev modules

# 10/21 Counts for Ordering Parts

| C | R | L |
|---|---|---|
| 10 uF | 22 kΩ | 1 uH |
| 1 uF | 100 kΩ | |
| 4.7 uF | 499 kΩ | |
| 22 uF | 1 kΩ | |
| 100 nF | 10 kΩ | |

| Strapping Pin # | Function | ESP32-S3 Module Default | Example Board Implementation |
|---|---|---|---|
| 0 | A value of 0 will put the chip into "Download Mode" which enables the flashing of the program memory. A value of 1 will boot from the SPI memory IC inside the module that contains the program memory. | Pulled up internally | Operate with button or programmer circuit |
| 3 | Switches the source for JTAG signals | Floating | Set with solder jumper but can be left alone. |
| 45 | Sets the voltage for the internal SPI memory IC | Pulled up or down internally | Pin is left unconnected to prevent the SPI memory voltage from changing. |
| 46 | Sets the boot mode and where the ESP32-S3 ROM sends messages. | Pulled down internally | A pull-up resistor is in the schematic and labelled as DNP to ensure the functionality of this pin is made available to the designer. In order to program normally, do not populate the pull-up resistor on the board. This will allow the internal pull-down to function and set the pin to 0. |

**Electrolitic**

IO
Power
ESP 32

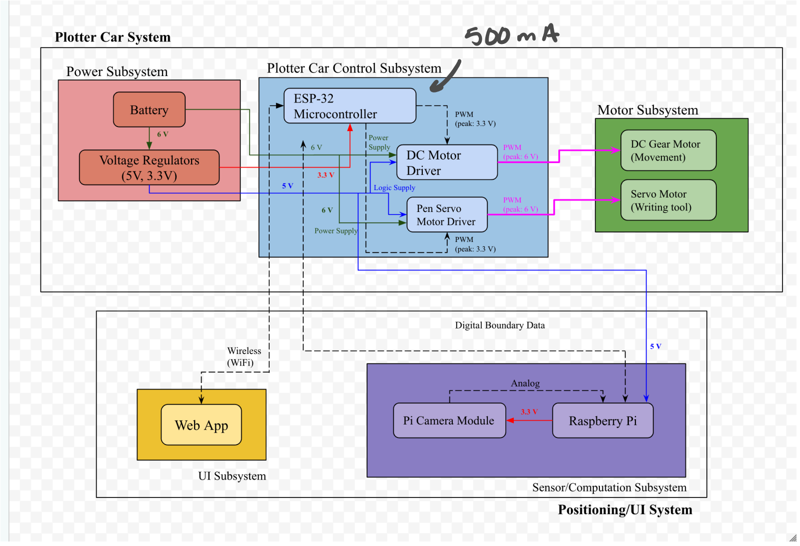
Motor Drivers
Sensors

C14, 15
11, 17
electrolitic
10 uF

10/22 TA Meeting

- ESP32 wifi edge of board
- vias next to dense cables
- test points (add footprints
- export uart pins

- order parts this week




- Order Parts
- Test motor drivers
- Test Raspi cam

## Week 10/27

- order parts
- web board mockup
- contact Gregg
- test motor (ESP, servo)

- create web app
  - ESP wifi interface
  - select shape
- Position Data
  - rasp. pi (camera)
  - IMU

- Move Robot
  - test motor drivers
  - wheels (gear motors)
    - program ESP
      - calculate motor functions
  - writing tool (servos)
    - program ESP

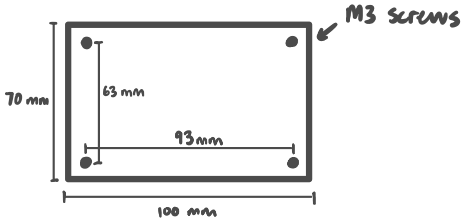
# 11/10 Feedback on PCB design

## PCB Changes :

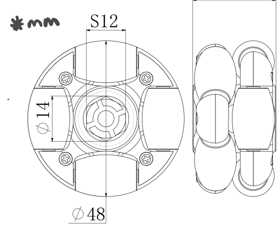
- ESP 32 to edge
- correct footprint for 6-SV and UART Bridge
- power connections for raspberry pi
- add test points to PCB
- add encoder connections for motor
- order buttons
- export pins for esp32 and UART
- add rasp pi GPIO pins

1, 2, 6, 7, 8, 21, 47

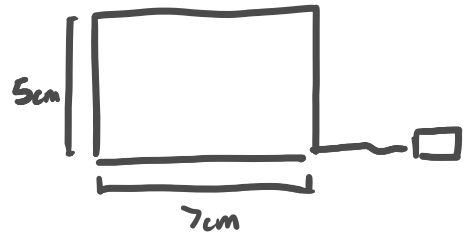
## PCB

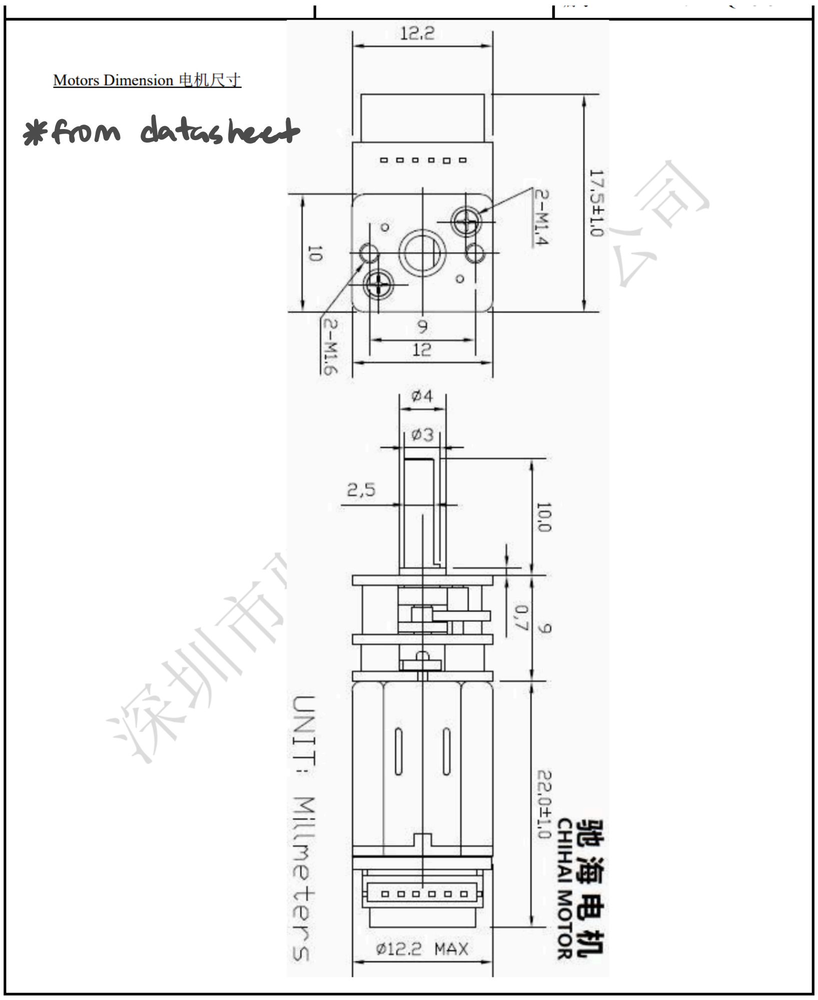



## Omniwheels
from datasheet

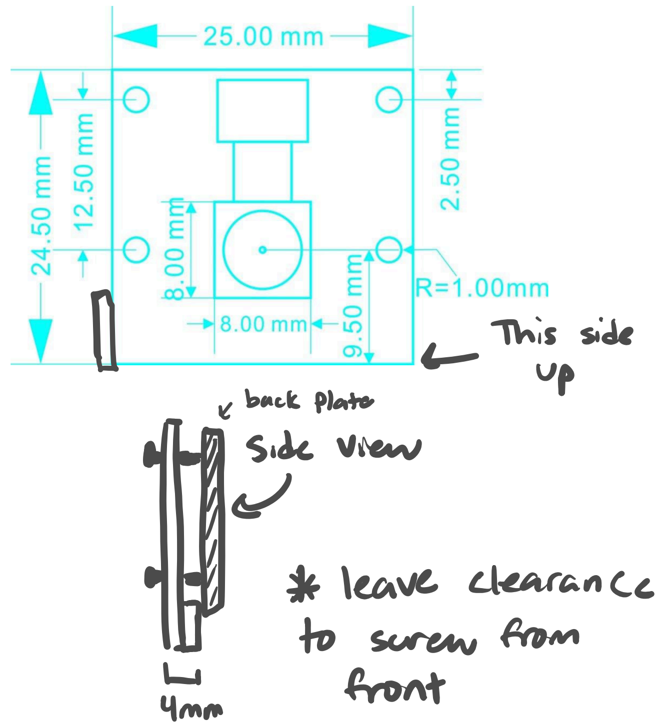



## Battery

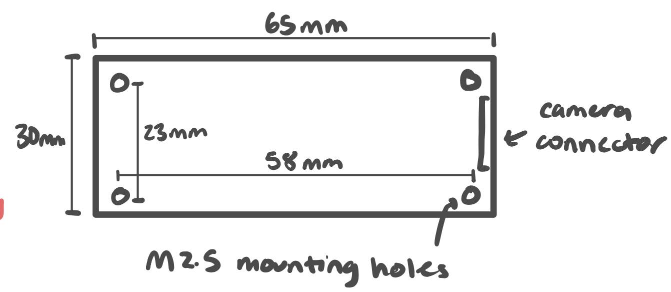


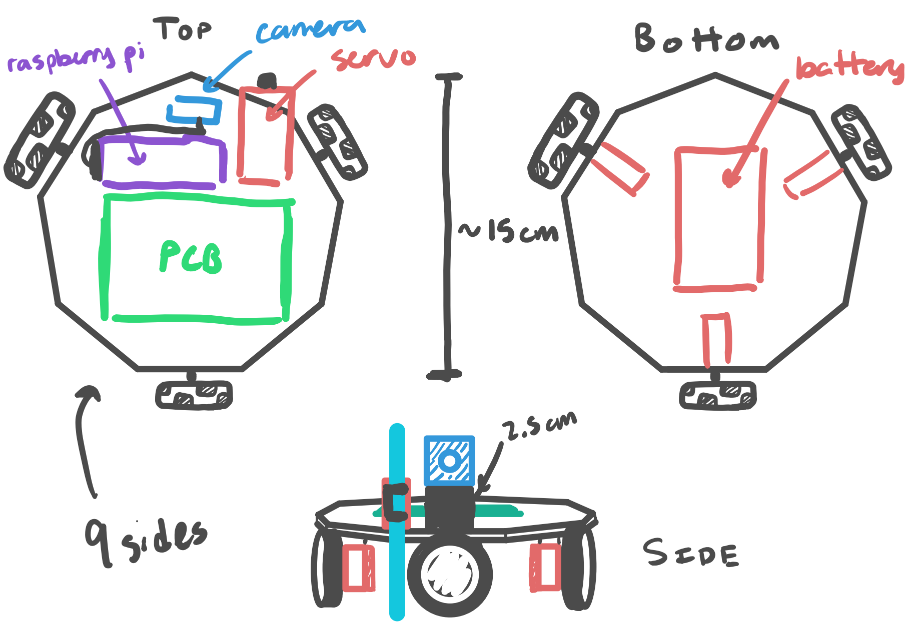

## Camera




## Raspberry Pi



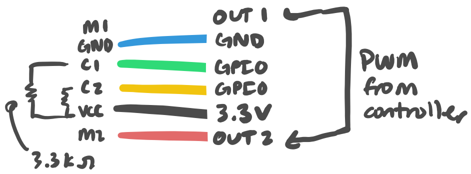

## General Layout



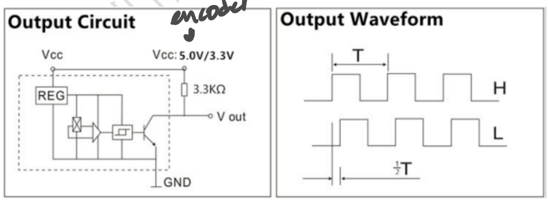

## Contact

Matthew Paul
@mjpaul3@illinois.edu
412-294-3074

# 11/18 Motor IO / Updated parts count




encoder Vcc




| C | R | Z |
|---|---|---|
| • 10u \|\| | • 1k \|\|\| | • 1u \| |
| • 1u \|\|\|\| | • 22.1k \| | ✓ |
| • 4.7u \| | • 47.5k \| | |
| • 22u \| | • 499k \| | |
| • .1u ~~\|\|\|\|~~ | • 22k \| | |
| • 10nF \|\|\|\| | • 100k \|\|\|\| | |
| ✓ | • 10k \|\|\|\| | |
| | • 3.3k ~~\|\|\|\|~~ \| | |
| | ✓ | |

# 12/4 Final Demo Notes



8:32 AM Wed Dec 4 ••• 2%

Final Demo Documents

## Power Subsystem

| Requirement | Verification |
|---|---|
| The 6V power output from the battery will be able to provide a stable output of 5V +/- 0.1V at least 400mA to the needed subsystems. | Using a digital multimeter, measure the output voltage of the battery, and on the test terminal of the regulator |
| The 6V power output from the battery will be able to provide a stable output of 6V +/- 0.1V at least 1.4A for the motors | Using a digital multimeter, measure the output voltage of the battery |

show voltmeter

* battery starts at 6.5V at full charge and stabalizes at 6V

## Control IO/Subsystem

| Requirement | Verification |
|---|---|
| Controller unit has a persistent WiFi connection and can connect to Web App with at most 1s of latency | Connect the USB of MCU to the laptop, write/upload a short program that verifies the wifi connection every second as an initial test. Write messages to arduino serial monitor. A test script can be written that measures the round-trip time (RTT) between sending a signal from the web app and receiving a confirmation response from the ESP32. This can be done using network tools like 'ping' or through custom logging on both ends that records the time taken for each message exchange. |
| ESP32 must generate an accurate PWM signal to send to the MDC | Connect the MCU pins to an oscilloscope and USB port to a laptop. Write a test function in Arduino that takes in inputs from the serial port and change commands. Then check the output waveform on the oscilloscope and verify expected results, and see if you can adjust the PWM. |

* show connection through web app

* through demo

## Motor Subsystem

| Requirement | Verification |
|---|---|
| ESP32 must provide an accurate PWM signal to motor controls such that the target location is within three centimeters of the actual position. | Perform controlled motor movement tests with the ESP32 providing PWM signals to the motors. Ensure the target position is reached within a 3 cm tolerance by comparing the actual and target positions across multiple trials through use of markers and distance measuring tools. |
| The ~~stepper~~ servo motor must be able to actuate quickly such that there is no ~~less~~ more than a .5 s delay between when the signal is sent and when the tool head is touching the writing surface | Measure the time delay between when the signal is sent to the stepper motor and when the tool head touches the writing surface using a digital oscilloscope and a contact sensor. |

* use ruler to check

hypothesis: at the time of designing the circuit, we did not know that the servo did not need to go through the motor controller

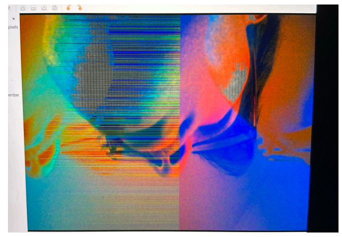
only a single PWM is needed

8:32 AM Wed Dec 4 ••• 2%

## Sensor/Computation Subsystem

| Requirements | Verification |
| --- | --- |
| The camera module can accurately calculate the robot's current position within five centimeters. | Test the camera module by comparing its calculated position with actual measurements using a precise ruler or external sensor. Ensure the difference does not exceed 5 cm across multiple trials. |
| The robot must be able to calibrate the camera's positioning and calculate the size of the boundary. | Run calibration routines where the robot detects boundaries, and verify the calculated boundary size by comparing it with manual measurements. Confirm that the robot can adjust camera positioning for accurate readings |
| Raspberry Pi must be able to communicate with the ESP32 via data line to relay the results of the positioning calculation. | Send positioning data from the Raspberry Pi to the ESP32 and log the communication. Verify successful data transmission by cross-checking the received data on the ESP32 with the original data from the Pi. |

hardware issues with camera
did tests to ensure it wasnt softw
↳ color channels misaligned



## UI Subsystem

| Requirements | Verification |
| --- | --- |
| The web app must have an interface that allows users to select both what shape they want to draw and its dimensions. | Manually test the web interface by selecting different shapes and entering their dimensions. Confirm that the shape and dimensions are correctly reflected in the app and transmitted to the backend for execution. Conduct automated unit tests to ensure proper shape and dimension selection functionality. |
| The web app must be secure with a login registered to the device and only allow one user to control the robot at a time. | Test the login functionality by attempting multiple simultaneous logins. Ensure that only one user can log in and control the robot at any given time. Perform penetration testing to verify the security of the login system and confirm that unauthorized access is prevented. |

* through demo

* ip

12/5 Updated digram for mock presentation