Portable Plotter Robot
Design Document

Team #34

Shinan Calzoni
Sagnik Chakraborty
Matthew Paul

TA: Dongming Liu
October 3, 2024
ECE 445

Table of Contents

1 Introduction

1.1 Problem

During the design process of any system, it is often necessary to create a visual aid that provides more detail and intuition regarding a particular idea. If one is not particularly artistically adept, this can prove to be a challenge. Although it is possible to utilize applications and electronic tools such as smart pencils or photo editing software, this is often impractical in certain collaborative settings where the working surface takes up a large amount of area (posters, large diagrams, etc.). Furthermore, current solutions for plotters that include fixed railings lack flexibility when it comes to the maximum size they can support.

1.2 Solution

What if there was a solution that met the criteria of its predecessors and exceeded it with the ability to draw over virtually any canvas? This is the mission of our project; our group aims to design a device that can support the illustration of basic shapes across virtually any canvas size with reasonable accuracy. The basic design is as follows: we plan to develop a vehicle with three omniwheels to allow for any lateral movement along a 2-D plane, equipping it with a camera and flexible border detection system. It will be controlled through a web application that allows the user to choose from a selection of shapes to draw and specify their dimensions and position.

1.3 Visual Aid

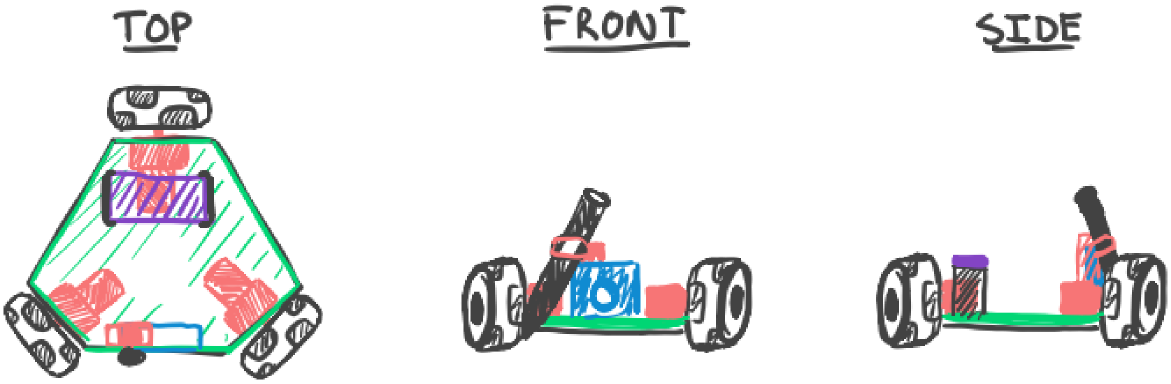

Figure 1: Device Appearance

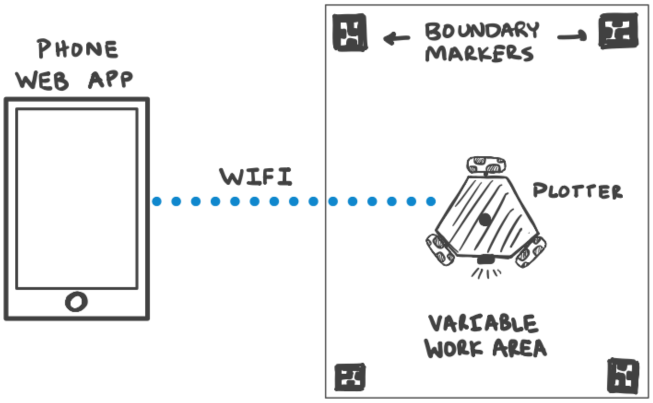

Figure 2: Solution in Context

1.4 High-Level Requirements

- The device can communicate with the web application maximum latency constrained to *one second* or less.
- The device footprint stays within the boundary dictated by corner markers and draws near the borders with a margin of *five centimeters.*
- Illustrations are accurate within a margin of error of *three centimeters* and the toolhead has the ability to actuate off and on to the canvas.

2 Design

2.1 Block Diagram

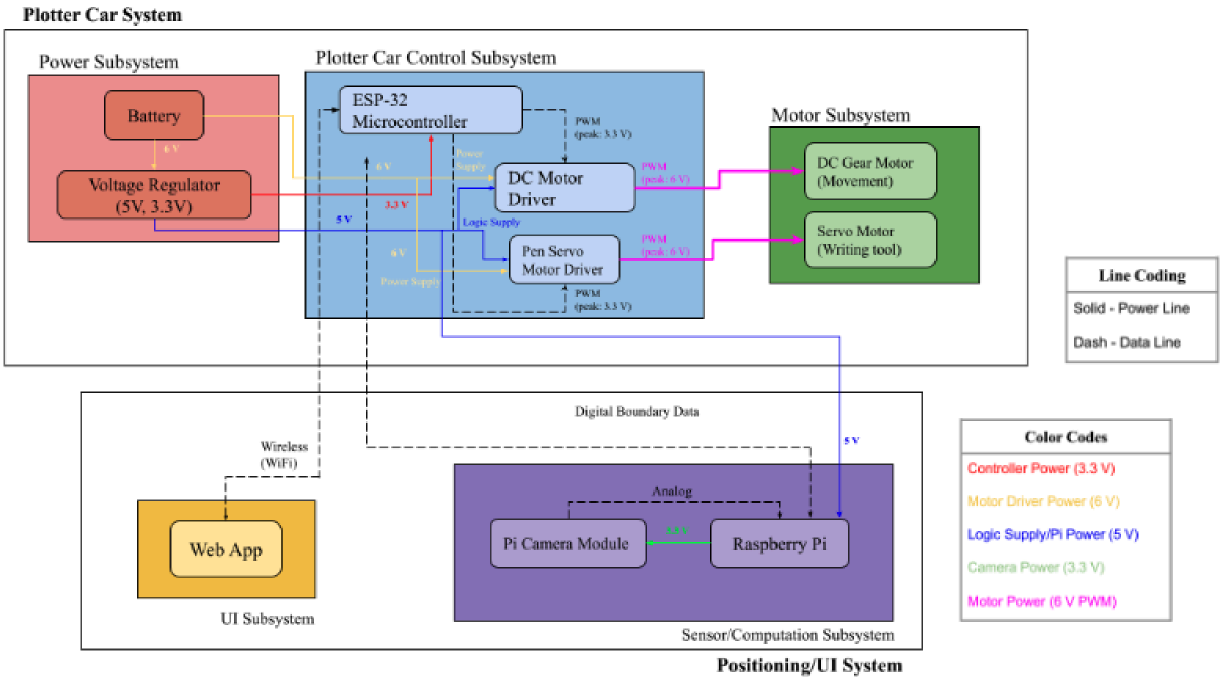

Figure 3: Full System Block Diagram

2.2 Physical Design

The area of our mounting board will be approximately 36 inches squared, not significantly bigger than the size of our PCB and shaped such that the design is triangularly symmetrical to support our three gear motors. Furthermore, the omni wheel diameter is 38 mm and the three DC gear motors will be placed symmetrically on the mounting board. At the edge of the mounting board, between two of the gear motors, will be the servo motor which controls the writing tool mechanism, as well as the Raspberry Pi Zero camera. Finally, the Raspberry Pi Zero and the rechargeable Li-Cd battery pack will be placed on the mounting board as well in an orientation that supports the most optimal weight distribution.

2.3 Plotter Car System Overview and Requirements

2.3.1 Power Subsystem

Our power system will need to safely and efficiently provide enough power to our motors and our ESP32 module. Since our motors are 6V, our main power supply unit will be a 6V Lithium Iron Phosphate ion rechargeable battery unit. This will provide most of the main power for our subsystems, as we can directly connect the motors to the battery. Now because our power supply and max voltage drops never exceed 2.7V of difference, we decided to simply use linear regulators to control the motor voltage. The LM1117DT Linear Voltage Regulator would suffice for setting up our power drops required, as they support a maximum dropout voltage of (V_{in} - V_{out}) = 5V

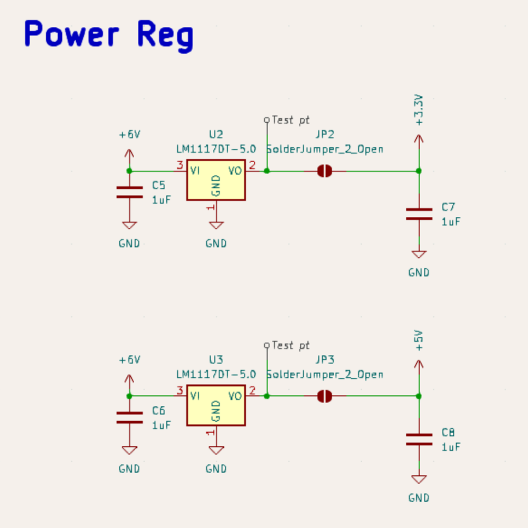

Figure 4: Power Circuit Schematic

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

2.3.2 Control/IO Subsystem

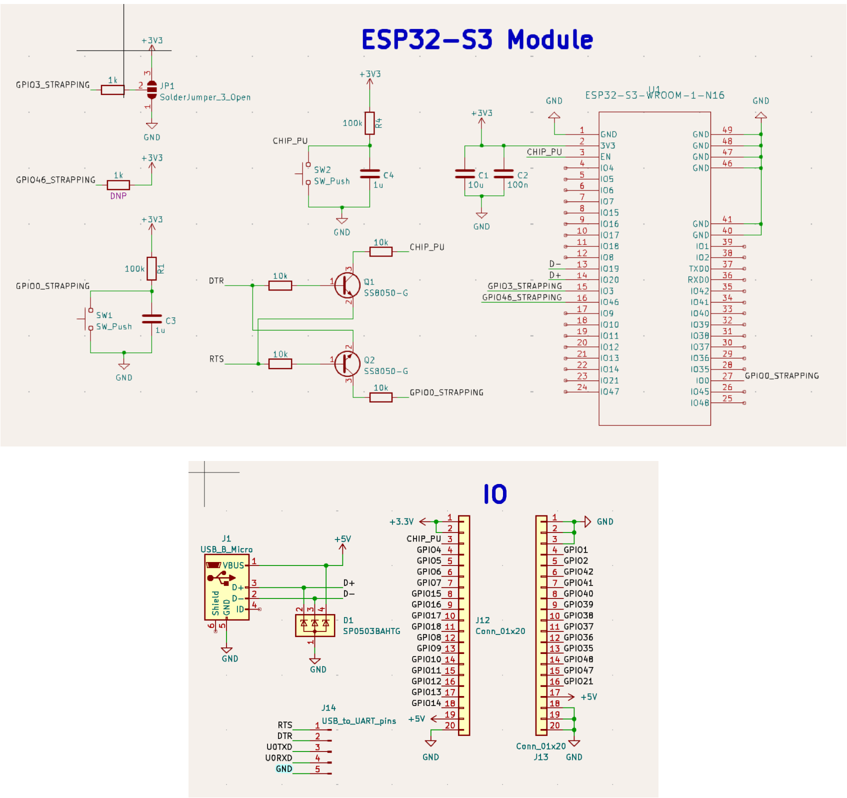

Figure 5: Control System and IO Circuit Schematics

The Control/IO subsystem for the plotter vehicle system is based on an ESP32-S3 microcontroller, which connects with numerous peripherals to properly manage inputs and outputs. The system contains a USB-to-UART interface (J5) for serial communication, allowing debugging and programming via the RTS, DTR, UOTXD, and UORXD signals. The ESP32-S3 module handles control signals for both DC motors and servo motors, and it generates the necessary PWM signals for motor control. The ESP32 communicates with external components via numerous GPIO pins and connectors such as J12 and J13, which increase its I/O capabilities. Strapping pins are also set to guarantee that the ESP32 boots into the correct modes and functions properly. The system also includes a Raspberry Pi, which, coupled with a Pi Camera Module, serves as the sensor.

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.

2.3.3 Motor Subsystem

The motor subsystem is responsible for controlling the movement and precise positioning of the plotter car, ensuring that both the DC gear motor for movement and the servo motor for the writing tool are managed effectively. The system receives PWM signals of a peak amplitude of 3.3V from the ESP32 microcontroller to control the motors, allowing for accurate motion and writing operations. The motors are powered by a 6V supply, with the ESP32 generating control signals to ensure the target location of the car is achieved within a 3 cm tolerance. Additionally, the stepper motor subsystem ensures rapid response times, minimizing delays between signal transmission and motor actuation, which is crucial for the plotter's precise writing and positioning tasks. For interfacing with the ESP32's control logic, we decided to use an L298N chip. This will require a logic drive voltage of 5V, which we directly get from the output of the linear dropout regulator discussed above. The 6V battery drives the max amplitude for the motors (which is 6V continuous delivery), and essentially acts as an amplifier for the motors to properly be controlled by the ESP32's output signal.

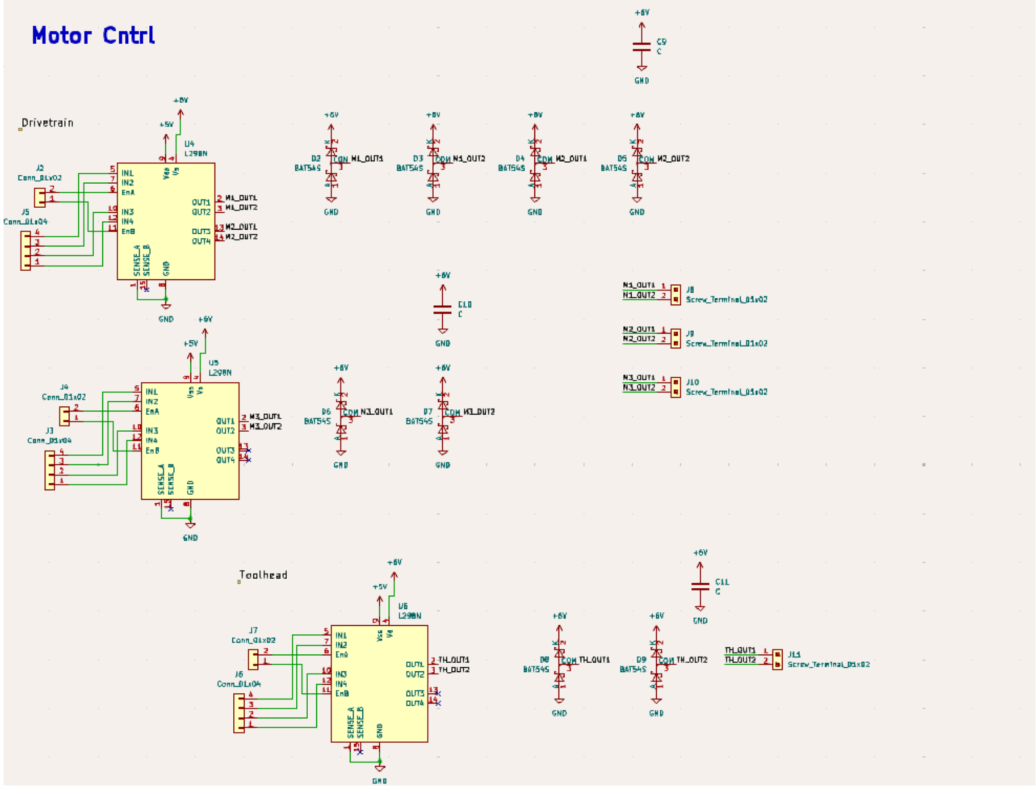

Figure 6: Motor Control Circuit Schematic

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 motor must be able to actuate quickly such that there is no less 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.

2.4 Positioning/UI System Overview and Requirements

2.4.1 Sensor/Computation Subsystem

An RPI-CAM-V2 Pi Camera Module will be integrated with a Raspberry Pi Zero to calculate the robot's current position through image processing. The camera captures the location and movement of the robot within the working area by interfacing with static boundary markers to gauge distance and calculate transformations for accurate positioning. The processed positioning data is transmitted from the Raspberry Pi to the ESP32 microcontroller using the I2C protocol over a data line, enabling real-time position updates. I2C is preferred for its faster, power-efficient, and more reliable wired communication compared to HTTP over WiFi, which might have high transmittance overhead. The camera operates with 3.3V power, and the Raspberry Pi communicates digital boundary data to assist in calibration and movement accuracy. The ESP32 uses this data to adjust motor controls via PWM signals, ensuring precise movement of the robot.

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.

2.4.2 UI Subsystem

The UI is a web-app that allows commands to be sent to the robot over a wireless connection (WiFi). It communicates directly with the ESP32 microcontroller in the Plotter Car Control subsystem, sending shape and dimension data, which the ESP32 uses to control the movement of the DC gear motor and pen servo motor through PWM signals. The web app is also responsible for updating the robot's position, which is relayed back from the ESP32 to ensure accurate plotting. This communication is vital for the overall coordination between the user's input and the robot's movements in the Motor subsystem. Additionally, the ESP32 integrates with the Power subsystem, which ensures the correct voltages are supplied to all components, and interacts with the Positioning/UI subsystem that uses the Pi Camera Module and Raspberry Pi for more advanced sensing and computation tasks.

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.
The LCD display needs to show the current position of the robot; data will be received directly from the ESP32 such that it is always the most updated location data.	Move the robot to various positions and observe the LCD display to confirm it shows the correct, real-time position. Cross-check the data displayed with logs from the ESP32 to ensure they match. Conduct stress tests by moving the robot rapidly to confirm the LCD consistently displays the updated position without delay.

2.5 Tolerance Analysis

2.5.1 Motors

First calculate movement speed of the portable plotter car:

Omni wheel diameter: d = 38 mm
Zero load motor rotations per minute: RPM = 200 rotations/minute

$$C = \pi d$$

$$C = \pi * 38\,mm * \frac{1m}{1000mm} = 0.11938\,m$$

$$speed = RPM * C$$

$$speed = 200\, rotations/min * \frac{1\, min}{60\, s} * 0.11938\, m/rotation = 0.49794\, m/s$$

The calculation above indicates that the speed of our car with no load is just under 0.5 m/s, which is approximately the top speed of a Roomba vacuum, for comparison.

Now to analyze the torque and force of our portable plotter, first define variables:

τ_w: Wheel torque (Nm)
r_w: Wheel radius (m)
μ_w: Wheel-floor static coefficient of friction
m: total mass
F_N: normal force
F_g: gravitational force
F_{FW}: Wheel-floor static friction force
g: acceleration due to gravity constant

Three DC gear motors are arranged in a triangular orientation, and the total force generated by the wheels can be summarized with the following:

$$F_{net} = F_1 + F_2 + F_3 \quad (1)$$

Furthermore, net torque can be derived as such:

$$\tau_{net} = \tau_1 + \tau_2 + \tau_3 = rF_{net} = r_w(F_1 + F_2 + F_3) \quad (2)$$

In order for the car to not slip and continue rolling, the following inequality must be satisfied:

$$\left|F_{net}\right| \leq \mu_w mg \quad (3)$$

Substituting net torque (2) into the above inequality (3):

$$\left|\tau_{net}\right| \leq \mu_w mgr_w \quad (4)$$

Given a radial static coefficient of friction of approximately 1 and a transverse static coefficient of friction of approximately 0.25, it is evident that transverse movement is the limiting factor when it comes to slip prevention. Furthermore, with our wheel radius of 17 mm and an estimated total mass of approximately 1-2 kg, in theory, the net torque exerted by the motors should not exceed 0.04165 Nm in the transverse direction in order to prevent slipping.

3 Cost and Schedule

3.1 Cost Analysis

Description	Manufacturer	Part Number	Quantity	Cost
Raspberry Pi Zero W	Raspberry Pi	SC1146	1	$22.99
Raspberry Pi Zero Camera	Raspberry Pi	RPI-CAM-V2	1	$12.79
ESP32-S3-WROOM-1-N16	Espressif Systems	ESP32-S3-WROOM-1-N16	1	$3.48
DC Motor w/ Encoder 6V	Adafruit	4638	3	$37.50
Omniwheel 38 mm	Nexus Robot	14166	3	$36.00
Ni-Cd Rechargeable Battery Pack (2-Pack)	GLESOURCE	GS0260SMP	1	$14.99
Linear Voltage Regulator	Texas Instruments	LM1117DT-5.0/NOPB	2	$3.80
Motor Driver	STMicroelectronics	L298N	3	$33.60
TVS DIODE	Littelfuse Inc.	SP0503BAHTG	1	$1.00
TRANS NPN 25V 1.5A	Comchip Technology	SS8050-G	2	$0.34
Inertial Measurement Unit	Bosch Sensortec	BMI323	1	$4.32
Total				$170.81
Total + Tax				$181.49

Labor: We will expect a salary of $40 / hour * 2.5 * 70 hours = $7000 per team member. Total labor costs amount to $21000.

Total Costs: The grand total cost will be $181.49 + $21000 = $21181.49.

3.2 Schedule

Week of	**Task**	**Person**
10/7	Design Review	All
	Design PCB (ESP32)	Matt
	Design PCB (Power System/Motor Drivers)	Shinan + Sagnik
10/14	Round 1 PCBway Orders	All
	Order Parts	All
	CAD Design for Machine Shop	Matt
	Start Working on Raspberry Pi Software	Matt + Shinan
	Start writing motor controller code	Sagnik
10/21	Round 2 PCBway Order	All
	Writing motor controller code	Sagnik
	Set up ESP32 wireless communication	Shinan
	Start Web App	Matt
10/28	Round 3 PCBway Order	All
	Motor Testing on PCB	Sagnik
11/4	Round 4 PCBway Order	All
	Individual Progress Reports	All
	Test App and Microcontroller Communication	Sagnik
11/11	Round 5 PCB	All
	Testing Web app with rest of design	Matt
	Testing Hardware Integration	All
11/18	Mock Demo	All
	Final Assembly	All
	Final Integration Testing with Web App	Everyone

11/25	Fall Break	All
12/2	Final Demo	All
12/9	Final Presentation	All
	Final Paper Due	All
	Submit Lab Notebooks	All

4 Ethics and Safety

Our project involves the development of a mobile drawing device equipped with a camera and web connectivity, designed to operate over various canvas sizes. While innovative, this technology raises several ethical and safety considerations that we must address to ensure responsible development and use.

Safety considerations:

- **Physical Safety**: The module will be mobile, and will operate in environments close to people. The robot will be designed to run with tabletop canvases in mind. This means that we will have protections in place to prevent the robot from misjudging the environment and rolling off the table. In accordance with the IEEE Code of Ethics, Canon 1, which emphasizes the imperative to "hold paramount the safety, health, and welfare of the public," we will integrate obstacle detection sensors and implement controlled speed limits to prevent accidents.
- **Electrical Safety**: The toolhead's actuation mechanism could pose risks as well. To uphold safety standards, we will design all mechanical and electrical systems to meet relevant industry regulations, ensuring they include fail-safes like emergency stop functions and overload protection, aligning with IEEE's commitment to public safety.

- **Honesty and Transparency**: We are committed to being upfront about the device's capabilities and limits. Whenever we must make a direct change to say the torque of the motors or potential force in the moving parts (e.g. the toolhead), we will be upfront about those changes, and the steps we took to make them safe. According to IEEE Canon 3, engineers must "be honest and realistic in stating claims or estimates." We will provide accurate information to users and stakeholders in order to establish reasonable expectations.
- **Accessibility and Inclusivity**: Ethical practice requires consideration of diverse user needs. As per ACM Principle 1.4, which encourages "being fair and taking action not to discriminate," we aim to make the device accessible to users with disabilities by incorporating features like audible alerts when the robot is too close to the edge and interfaces compatible with assistive technologies. We aspire to have as much control over the robot as needed.

Adherence to Professional Ethical Standards:

In developing this project, we are guided by the ethical principles outlined by both the IEEE and ACM:

Commitment to Public Welfare: Upholding the IEEE Code of Ethics, Canon 1, we prioritize the safety, health, and welfare of the public in all aspects of our project. Furthermore, in alignment with ACM Principle 1.6, we respect user privacy and are dedicated to implementing measures that protect personal data. As the plotter may rely on web applications for control and updates, it is vital to respect user privacy and protect sensitive data. We will adhere to best practices in data protection, ensuring that all communications between the plotter and the web app are secure and encrypted. In line with **ACM Principle 1.6**, which advocates for respect for privacy, we will safeguard any personal data collected during the use of the web app, ensuring transparency about data usage. The project must also consider the environmental impact of its materials and energy use. We will design the system to minimize power consumption, such as using the right voltage ranges for operation, continuous testing of our components, using recyclable materials where possible, and ensuring that the lithium-ion batteries meet sustainability standards. This commitment reflects the broader responsibility engineers have to the environment and society, as outlined in **IEEE Code of Ethics, Canon 5**, which calls for engineers to improve "the understanding of technology, its appropriate application, and potential consequences."

5 Citations

IEEE - IEEE Code of Ethics. Accessed October 4, 2024. https://www.ieee.org/about/corporate/governance/p7-8.html.

“The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society.” Code of Ethics. Accessed October 3, 2024. https://www.acm.org/code-of-ethics.