

WHITEBOARD ASSISTANT

By

Morris Yu

Avneet Kaur

Saj Patel

ECE 445 Design Document - Fall 2026

Professor: Arne Fliflet

TA: Junyan Bai

Project #20

Table of Contents

1. Introduction
 - 1.1 Problem
 - 1.2 Solution
 - 1.3 Visual Aid
 - 1.4 High Level Requirement
2. Design
 - 2.1 Block Diagram
 - 2.3.1 External User Input Subsystem
 - 2.3.2 Control and Text Processing Subsystem
 - 2.3.3 Sensing Subsystem
 - 2.3.4 Writing Subsystem
 - 2.3.5 Power Subsystem
 - 2.3.6 Movement and Attachment Subsystem
 - 2.4 Tolerance Analysis
3. Cost and Schedule
 - 3.2 Schedule
4. Ethics, Safety, Societal Impact, and Engineering Standards
 - 4.1 Societal Impact
 - 4.2 Engineering Standards
 - 4.3 Safety
 - 4.4 Ethics
5. Citations
5. References

1. Introduction

1.1 Problem

Whiteboards are commonly used in classrooms, labs, and meeting spaces to explain concepts, share ideas, and work through problems. They provide a large, reusable surface where information can be added or changed during a discussion. However, someone still needs to physically write the content, even when that information already exists on a computer. During a presentation, this requires the presenter to turn away from the audience and spend time copying text onto the board. Rewriting the same information for different classes or meetings also creates repeated work that could be automated. These tasks can break the flow of a discussion and divide the presenter's attention between explaining the material and preparing the board. Large vertical whiteboards can also be difficult to use for people with limited reach or mobility, since writing requires them to reach different areas of the surface and maintain control of a marker. A device that writes user provided text would reduce these physical demands and make it easier to transfer digital information onto an existing whiteboard. Users could prepare the text from a laptop and have it displayed without manually copying each word.

1.2 Solution

We propose a portable, battery powered whiteboard assistant robot that attaches to a steel backed whiteboard using magnets and automatically writes user provided text. Users will send text wirelessly from a laptop or another compatible device to an ESP32 controller on the robot. The controller will represent each supported character as a series of strokes and convert those strokes into movement and marker control commands. Four motor driven wheels will move the robot across the board, allowing it to follow horizontal, vertical, and diagonal paths. In the current design, a separate steering motor will turn all four wheel modules together so that the robot can change its direction of travel while keeping the wheels parallel. Wheel encoders will provide feedback on wheel rotation, a steering angle sensor will report the wheel direction, and an inertial measurement unit will help monitor changes in the robot's body orientation. A spring loaded marker holder will maintain writing pressure while allowing small variations in the distance between the chassis and the board. A servo will retract the marker between separate strokes and letters to prevent unwanted connecting lines. Edge sensors will also be incorporated to stop the robot before it drives beyond the board. By combining magnetic attachment with onboard power, the design aims to simplify setup and allow the robot to be moved between compatible whiteboards without installing suspension hardware. It will use ordinary dry erase markers, allowing users to continue using the board manually alongside the robot.

1.3 Visual Aid

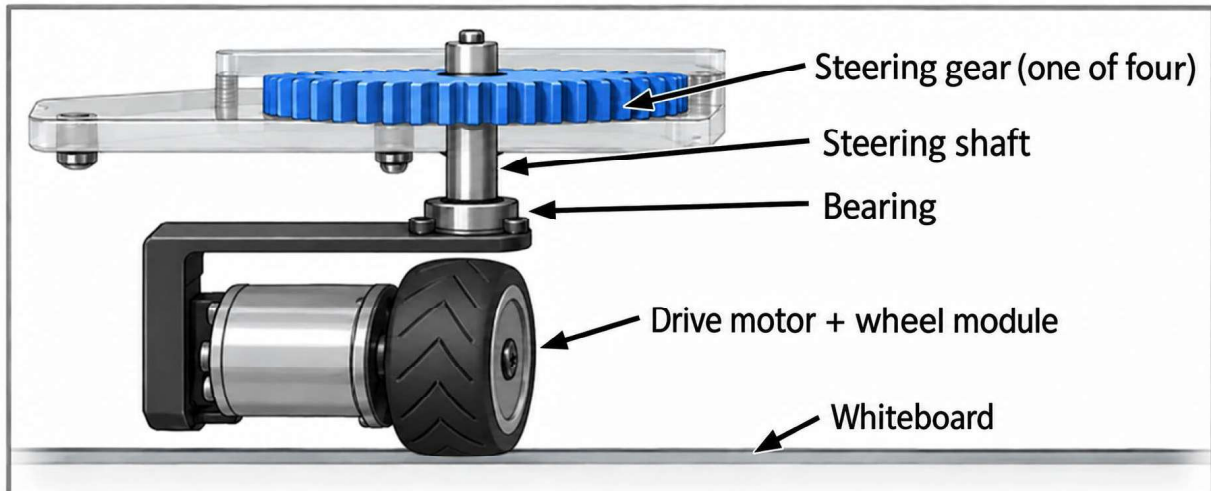


Figure 1: Proposed steering and drive module for the Whiteboard Assistant robot

Figure 1 shows one of the robot's four proposed steering and drive modules. The steering gear connects through a shaft to an L shaped bracket that holds the drive motor and wheel. Turning the gear rotates the entire module to change the wheel's direction, while the drive motor independently controls wheel rotation. The bracket positions the wheel beneath the steering axis and provides clearance so the tire can rotate without touching the bracket. In the complete assembly, a central gear driven by one steering motor will turn all four steering gears together, keeping the wheels pointed in the same direction.

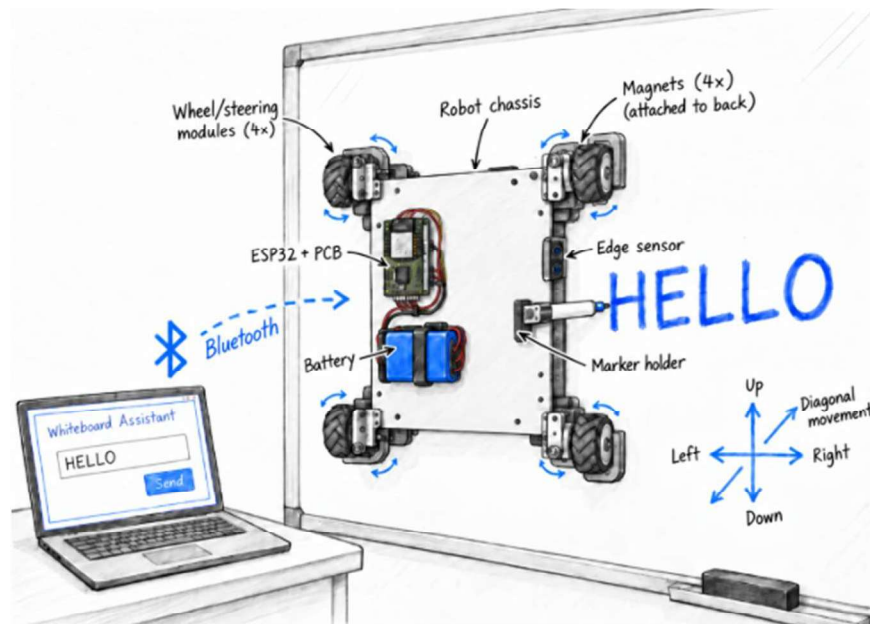


Figure 2: Overall operation of the Whiteboard Assistant robot on a vertical magnetic whiteboard using Bluetooth connection

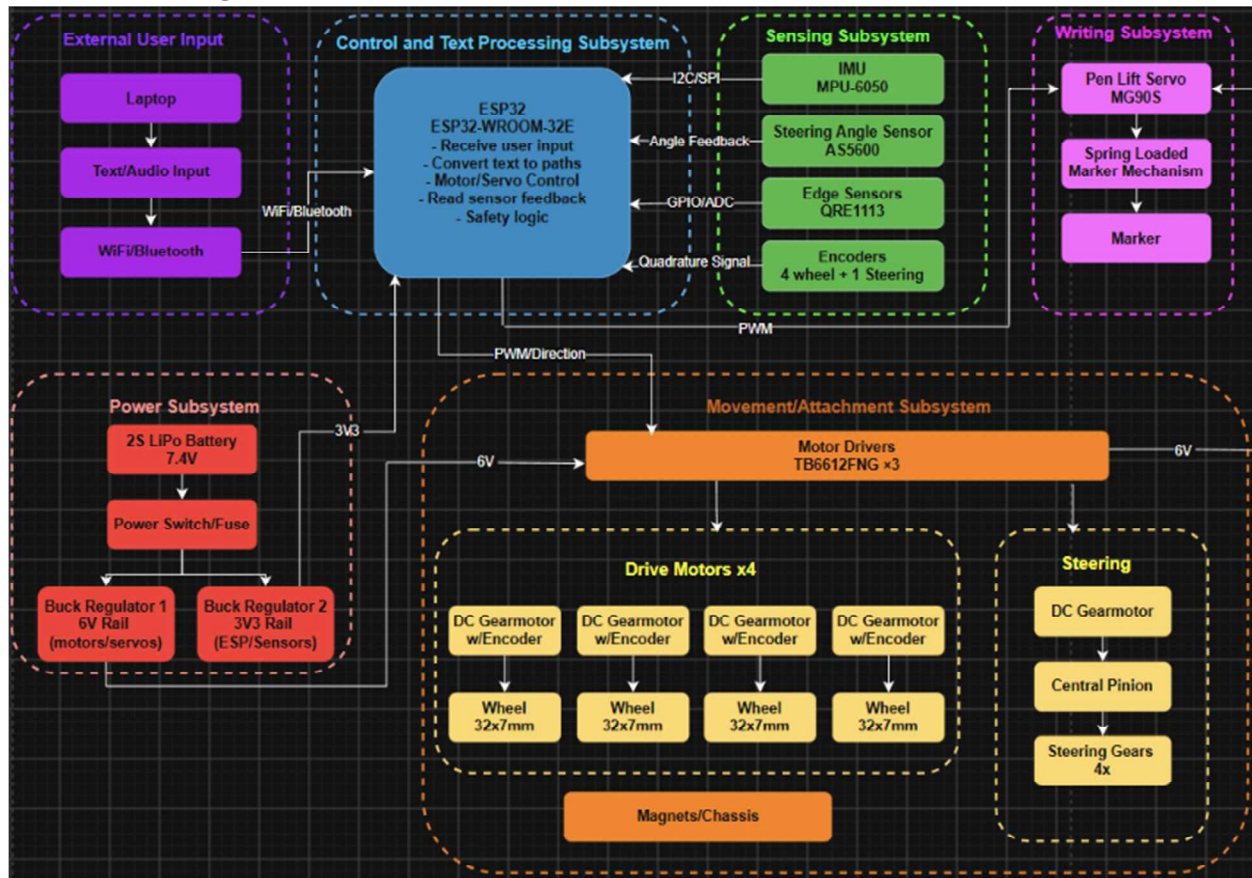
Figure 2 shows the overall operation of the Whiteboard Assistant on a vertical magnetic whiteboard. The user sends the text wirelessly from a laptop to the ESP32 controller, which converts the input into movement and writing commands. Four independently driven and steerable wheel modules move the robot across the board, while magnets mounted to the chassis keep the robot attached to the surface. The marker holder controls contact with the whiteboard during writing, and the edge sensor is mounted near the chassis perimeter to detect the loss of the board surface before the robot reaches the boundary.

1.4 High Level Requirement

- The robot shall remain attached to a vertical magnetic whiteboard without external support while performing commanded horizontal, vertical, and diagonal movements.
- The robot shall receive user provided text wirelessly from a laptop or another compatible device and complete the requested writing sequence without continuous user control.
- The robot shall write the uppercase letters A–Z so that a reader can identify each letter without being shown the input text, and shall lift the marker between disconnected strokes to prevent unintended connecting lines.

2. Design

2.1 Block Diagram



The block diagram divides the whiteboard assistant into six subsystems that work together to receive text, control movement, and write on the board. The external user input subsystem allows the user to enter text on a laptop or another compatible device and send it wirelessly to the robot through Wi-Fi or Bluetooth. The control and text processing subsystem uses an ESP32 to convert the received text into character paths, coordinate the drive motors and steering mechanism, and control when the marker contacts the board. The sensing subsystem provides wheel rotation, steering angle, and body orientation feedback, along with planned edge detection, so the controller can monitor movement and stop the robot when a board boundary is detected. The movement and attachment subsystem uses magnets to hold the chassis against the whiteboard, four independently driven wheels to move across its surface, and a central steering gear to turn all four wheel modules together. The writing subsystem uses a spring loaded marker holder to maintain contact with the board during a stroke and a servo to retract the marker between disconnected strokes or characters. The power subsystem supplies the robot from an onboard battery, regulates the voltage for the motors and electronics, and includes a master switch and fuse for power control and protection. These subsystems connect through defined power and signal interfaces, allowing them to be developed and tested separately before being assembled into the complete robot.

2.3.1 External User Input Subsystem

The external user input subsystem lets users enter text on a laptop and send it wirelessly to the robot. This supports the high level requirement that the robot receive a writing command and carry it out without continuous user control. The proposed interface is a simple web page with a text box, a submission button, and a message showing whether the robot accepted the request. The laptop will connect to a Wi-Fi network hosted by the ESP32, allowing users to operate the robot without an internet connection or a separate router. A browser based interface also uses equipment the user already has and avoids the need for a separate handheld controller.

The initial design will support uppercase letters A–Z and spaces, with a proposed limit of 32 characters per request. This keeps the first version focused on short messages while the team develops and tests the writing system. Each request must contain at least one letter. If the user submits an empty message, unsupported characters, or text that exceeds the limit, the interface will explain why the request was rejected. Typed text will be the main input method. Voice input could be added later by converting speech to text on the laptop before sending it to the robot.

When the user submits a message, the laptop will send it to the ESP32 through an HTTP request. The ESP32 will check the text and respond with an accepted, rejected, or busy status. An accepted response means that the robot has received and stored the text for processing; the writing itself may still be in progress. Displaying this response lets the user know whether the command reached the robot.

Interfaces with Other Subsystems

Interface parameter	Definition
Wireless connection	Direct Wi-Fi connection between the laptop and the ESP32 access point.
Request format	HTTP POST to /write, containing a JSON field named text.
Supported text	A string containing 1–32 ASCII characters, limited to uppercase letters A–Z and spaces, with at least one letter.
Controller response	A response containing a value of accepted, rejected, or busy; accepted responses also return the received text for confirmation.
Transfer to path processing	Only a complete, valid, accepted message is passed to the character-path generator.

Requirements and Verifications

Requirement	Verification
R1. While the robot is idle, the subsystem shall transfer supported text without missing or changed characters at indoor distances up to 3 m, with no failed transfers in a 30 request test.	Equipment: Laptop, ESP32 running the receiving firmware, tape measure, and serial logging connection. Procedure: 1. Disable motor operation and connect the laptop to the ESP32's Wi-Fi network. 2. Send ten messages at each nominal distance of 1, 2, and 3 m, recording the actual distances. Include all supported letters, spaces, and messages at the minimum and maximum lengths. 3. Compare each submitted message with the text stored by the ESP32. Results: Record distance, submitted text, received text, and pass/fail in a notebook table. Every received message must match its submitted text exactly.
R2. For every valid request in the R1 test, the interface shall display the robot's acceptance response within 2 seconds of submission.	Equipment: Laptop with browser timing tools and the R1 setup. Procedure: 1. Record when each request is submitted and when its acceptance message appears. 2. Calculate the elapsed time for each request. Results: Add response times to the R1 table and report the longest response time. Every result must fall between 0 and 2 seconds.
R3. The subsystem shall reject empty messages, messages containing only spaces, unsupported characters, and text longer than 32 characters without sending them to path processing.	Equipment: Laptop, ESP32, and serial logging connection. Procedure: 1. Submit an example from each invalid-input category through the web interface. 2. Send the same inputs directly to /write to check that the controller also rejects them. 3. Confirm that an error appears and that no writing command is added. Results: Record each input, the rejection message, and whether a writing command was added. Every invalid input must be rejected.
R4. If no controller response arrives within 2 seconds, the interface shall show that the request is unconfirmed and shall not report that it was accepted.	Equipment: Laptop, ESP32, and browser timing tools. Procedure: 1. Connect the laptop to the robot. 2. Make the controller unavailable, then submit a request. 3. Check the message displayed after the timeout. Results: Record the elapsed time and save a screenshot of the message. The interface must show that the request is unconfirmed, with no acceptance indication.

2.3.2 Control and Text Processing Subsystem

The control subsystem uses the ESP32 to coordinate the robot's movement and marker mechanism. It sends commands to the four drive motors, steering motor, and pen lift servo while reading feedback from the wheel encoders, AS5600, and MPU-6050. Before each stroke, it positions the wheels in the required direction, then adjusts motor output using encoder feedback as the robot moves. The marker stays raised during repositioning and lowers for writing strokes. If an edge or required sensor fault is reported, the controller disables motor drive and requests that the marker be raised. This supports controlled movement and writing without continuous user input.

The text processing subsystem converts an accepted message into paths the robot can follow. Each uppercase letter is stored as an ordered set of points with instructions for raising and lowering the marker. A single line font lets the robot draw each stroke directly, while spaces advance the writing position with the marker raised. The software scales these paths relative to the selected starting position and calculates the travel distance and direction for each segment. The accepted message remains stored until the sequence finishes, and additional writing requests receive a busy response while the robot is working.

Interfaces with Other Subsystems

Connected subsystem	Interface
External user input	Accepts the validated text string defined in Section 2.3.1 and returns accepted, rejected, or busy status.
Movement and attachment	Provides independent PWM and direction commands for four drive motors and one steering motor through the TB6612FNG drivers [5]. The proposed motor PWM frequency is 20 kHz $\pm$ 1 kHz, with duty cycle commands from 0 to 100%.
Wheel encoders	Receives two quadrature signals from each drive motor, giving eight encoder inputs. Counts per wheel revolution will be established for the selected decoding method.
Steering sensor and IMU	Communicates with the AS5600 and MPU-6050 over a shared I ² C bus, initially configured for 100 kHz $\pm$ 5 kHz.
Edge detection	Receives an edge indication that takes priority over normal writing commands. The electrical interface depends on the final edge sensor circuit.
Writing	Provides a servo control signal at a proposed frequency of 50 Hz $\pm$ 1 Hz. Pen up and pen down pulse widths will be calibrated using the assembled marker mechanism.
Power	Receives the proposed regulated 3.3 V $\pm$ 0.1 V logic supply and shares ground with the sensors, motor drivers, and servo. Motor and servo power are supplied separately by the power subsystem.

Requirements and Verifications

Requirement	Verification
R1. The controller shall generate the correct strokes and marker positions for every uppercase letter A–Z. For accepted messages of 1–32 characters, it shall keep the letters and spaces in the order entered by the user.	Equipment: Laptop, ESP32, serial connection, and a tool for plotting the character paths. Procedure: 1. Disable the motors. 2. Send each supported letter separately, followed by messages containing repeated letters, spaces, and the maximum supported length. 3. Record the generated points and whether the marker is raised or lowered for each movement. 4. Plot the paths and compare them with the approved character definitions. Results: Save the plots and record whether each test passes. Every character must follow the expected stroke order and marker positions, and each message must preserve the original text order.
R2. After accepting a valid message, the controller shall finish its command sequence without further input from the laptop. If another writing request arrives during that sequence, it shall respond with a busy message and continue the original task.	Equipment: Laptop, ESP32, and a simulated motor and sensor interface. Procedure: 1. Submit a valid message and supply simulated feedback showing that each movement has finished. 2. Send another request while the first task is running and check for a busy response. 3. Disconnect the laptop and continue supplying the simulated feedback. 4. Confirm that the original task reaches the complete state. Results: Record the controller’s state changes, its response to the second request, and the final completion status. The original task must finish without being replaced or requiring another laptop command.
R3. While the robot is moving, the controller shall check sensor feedback and update motor commands at least once every 50 ms.	Equipment: ESP32, simulated encoder and sensor inputs, and a logic analyzer or oscilloscope. Procedure: 1. Set a debug output to change state each time a control update finishes. 2. Run a writing sequence while changing the simulated feedback and sending wireless requests. 3. Capture the debug signal and measure the time between updates. Results: Save the waveform and report the longest interval. No measured interval may exceed 50 ms.
R4. When an edge or required sensor fault is reported, the controller shall disable the drive and steering motors and send a pen up command within 100 ms. It shall keep normal movement blocked until the fault is cleared and operation is deliberately	Equipment: ESP32, simulated fault inputs, and a logic analyzer or oscilloscope. Procedure: 1. Start a movement sequence with the motors disconnected. 2. Introduce each defined fault condition. 3. Measure the time from the fault indication to the motor disable output and the pen up command. 4. Check that movement stays blocked until the fault is cleared and a restart is requested. Results: Save the waveforms and record both response times for each fault. Both commands must occur within 100 ms, and

restarted.	movement must remain blocked until the restart conditions are met.
R5. The controller shall command the marker to rise before repositioning and shall lower it only for writing strokes. It shall prevent translation until the steering angle is within the approved tolerance.	Equipment: ESP32, simulated steering and movement feedback, and a logic analyzer or command log. Procedure: 1. Run a character with disconnected strokes. 2. Supply a steering angle outside the approved tolerance and confirm that translation remains blocked. 3. Supply an angle within the tolerance and allow the sequence to continue. 4. Check the order of the steering, marker, and movement commands. Results: Save a labeled timeline of the commands. The marker must be commanded up before repositioning, translation must wait for steering confirmation, and the marker must be commanded down only for when writing strokes.

2.3.3 Sensing Subsystem

The sensing subsystem provides the feedback needed to control movement and keep the robot on the whiteboard. It supports the requirements for coordinated travel and autonomous writing by reporting wheel rotation, steering angle, and body orientation. Each drive motor has an encoder so the controller can track its rotation and direction independently. The AS5600 measures the steering angle using a diametric magnet attached to a steering shaft, while the MPU-6050 monitors changes in chassis orientation.

The AS5600 will measure one wheel module's steering shaft, giving feedback after the steering gears rather than relying on steering motor rotation. The IMU will remain fixed to the chassis. Planned edge sensors will face the board near the chassis perimeter and report when the supporting surface is no longer detected. Their placement and detection thresholds still need testing on the actual board. Encoders and the IMU help monitor motion, but they do not directly measure absolute position or guarantee detection of wheel slip.

Interfaces with Other Subsystems

Interface	Proposed definition
Power	Sensors receive 3.3 V $\pm$ 0.1 V and share ground with ESP32.
Drive encoders	Two GPIO inputs per motor carry the encoder A and B signals, giving eight inputs total.
AS5600 and MPU-6050	Share the I ² C bus described in Section 2.3.2; initial bench connections use GPIO21 for SDA and GPIO22 for SCL.
Controller feedback	Supplies signed encoder counts, steering angle, body orientation, and sensor validity information at intervals no greater than 50 ms.
Edge detection	Supplies a surface present or edge indication. The final circuit and mounting positions remain to be selected.

Requirements and Verifications

Requirement	Verification
R1. Each encoder channel shall report rotation direction correctly and count within ± 1 count of a reference sequence containing 1,000 decoded counts.	Equipment: Quadrature signal generator, ESP32, and logic analyzer. Procedure: 1. Apply a known sequence corresponding to 1,000 decoded counts to each motor's encoder inputs. 2. Repeat with the direction reversed. 3. Compare the reference and recorded counts. Results: Record expected count, measured count, and direction for each channel. Every count must be within tolerance, with the correct direction.
R2. After calibration, the steering sensor shall report wheel angle within $\pm 1^\circ$ of a reference angle at 0° , 45° , 90° , 135° , and 180° .	Equipment: Assembled steering module, ESP32, and an angle fixture with reference uncertainty no greater than $\pm 0.5^\circ$. Procedure: 1. Set the zero reference. 2. Position the wheel at each test angle and record the sensor reading. 3. Repeat while approaching from the opposite direction, with the attachment magnets installed. Results: Record reference angle, sensor angle, and error. Every error must be within $\pm 1^\circ$.
R3. While stationary on a vertical board, the IMU orientation estimate shall measure chassis rotation within the board plane to within $\pm 3^\circ$ at 0° , $\pm 45^\circ$, and $\pm 90^\circ$ relative to its calibrated starting orientation.	Equipment: MPU-6050, ESP32, and an angle fixture with uncertainty no greater than $\pm 0.5^\circ$. Procedure: 1. Calibrate the starting orientation. 2. Rotate the chassis to each reference position and allow it to settle. 3. Record the reported orientation. Results: Compare reference and measured angles in a table. Every error must be within $\pm 3^\circ$.
R4. At the selected mounting height, each edge sensor shall distinguish the board surface from its absence in ten repeated transitions, reporting an edge before the nearest wheel crosses the board boundary.	Equipment: Actual whiteboard and frame, sensor mounting fixture, wheel position reference, and ESP32 logging. Procedure: 1. Move the fixture slowly toward each relevant board edge. 2. Record where the edge indication occurs relative to the wheel. 3. Return to the board surface and repeat ten times per sensor position. Results: Record detection positions and sensor states. It is important that every trial must detect the edge before the wheel crosses the boundary. It's okay if it reports too early, but it cannot ever report too late.

2.3.4 Writing Subsystem

The writing subsystem controls when the marker touches the board and maintains contact during each stroke. It supports the requirement to write identifiable letters without unwanted connecting lines. The proposed mechanism uses a sliding marker holder, a compression spring, and an MG90S servo mounted near the upper edge of the chassis. The spring pushes the marker toward the board and allows the holder to move slightly as the distance between the chassis and board changes. The servo retracts the holder through a cam or lifting arm, then releases it so the spring can provide writing pressure. This keeps the servo from rigidly fixing the marker position during a stroke. The holder will allow marker replacement and adjustment of the tip position.

Interfaces with Other Subsystems

Interface	Proposed definition
Controller	Receives a servo signal at $50 \text{ Hz} \pm 1 \text{ Hz}$, with calibrated pulse widths for pen up and pen down.
Power	Uses a separate servo supply with a common ground to the ESP32. The proposal uses a nominal 6 V supply; its allowable range must be confirmed for the selected servo before the power design is finalized.
Movement	The controller waits for marker retraction before repositioning and lowers the marker before starting a writing stroke.
Mechanical attachment	The holder mounts to the chassis with its sliding axis perpendicular to the board.

Requirements and Verifications

Requirement	Verification
R1. In the pen up position, the marker tip shall have at least 5 mm of clearance from the board throughout 20 repeated lift cycles.	Equipment: Assembled holder, servo, ESP32, regulated supply, and ruler or calipers. Procedure: 1. Set the marker to its normal writing position. 2. Command pen up and measure the tip clearance. 3. Repeat for 20 cycles. Results: Record the clearance for each cycle. Every measurement must be at least 5 mm.
R2. With the marker engaged, the mechanism shall produce uninterrupted ink strokes during ten 100 mm passes in each	Equipment: Whiteboard, assembled mechanism, controlled sliding fixture, ruler, and camera. Procedure: 1. Adjust the spring preload and marker

horizontal, vertical, and diagonal direction.	position. 2. Move the mechanism through each test direction while keeping the chassis mounting height fixed. 3. Inspect the resulting strokes. Results: Save labeled photographs and record any visible gaps. Every pass must leave a continuous stroke.
R3. The mechanism shall reach the required pen up clearance within 0.5 seconds of the command and leave no ink during subsequent repositioning.	Equipment: Assembled mechanism, ESP32, whiteboard, and camera recording at least 60 frames per second. Procedure: 1. Record a command indicator and the marker as a pen up command is issued. 2. Measure the time until the tip reaches the required clearance. 3. After 0.5 seconds, move the mechanism across the board. 4. Repeat ten times. Results: Record lift times and photograph the repositioning paths. Every lift must finish within 0.5 seconds, with no ink deposited during repositioning.

2.3.5 Power Subsystem

The power subsystem supplies the robot from an onboard 2S LiPo battery, supporting the requirement to complete a writing task without an external power connection. Separate regulators will provide a 3.3 V supply for the ESP32 and sensors and a nominal 6 V supply for the motors. Separating these supplies helps keep motor startup and load changes from disturbing the controller. The servo will use a supply compatible with its confirmed voltage rating. The battery connection will include a master switch, fuse, and protection against reversed polarity and excessive discharge. All subsystems will share a common ground, with motor current routed through suitable wiring and PCB traces rather than through the controller board.

Interfaces with Other Subsystems

Interface	Definition
Battery input	A 2S LiPo battery rated at 7.4 V nominal and 8.4 V when fully charged. The initial regulator test range is 6.6–8.4 V; the lower operating limit must be confirmed for the selected battery and regulators.
Controller and sensors	A regulated 3.3 V $\pm$ 0.1 V supply capable of delivering at least 500 mA continuously.
Motor drivers	A regulated 6.0 V $\pm$ 0.3 V supply, with an initial target of supporting 3 A peak demand.

Writing servo	A separate power connection with common ground. It may share the motor supply if the selected servo supports its full voltage range.
Protection	Battery power passes through the master switch and fuse. Final fuse, polarity protection, and battery cutoff components remain to be selected.

Requirements and Verifications

Requirement	Verification
R1. The logic supply shall remain between 3.2 and 3.4 V for loads from 0–500 mA across the proposed battery input range.	Equipment: Adjustable bench supply, electronic load, and DMM. Procedure: 1. Set the input to 6.6, 7.4, and 8.4 V. 2. At each voltage, increase the logic load from 0 to 500 mA in 100 mA steps. 3. Hold the maximum load for five minutes. Results: Record input voltage, load current, and output voltage. Every output reading must remain within the required range.
R2. The motor supply shall remain between 5.7 and 6.3 V during a proposed 3 A load pulse lasting 100 ms across the battery input range.	Equipment: Bench supply, electronic load with pulse mode, and oscilloscope. Procedure: 1. Test at 6.6, 7.4, and 8.4 V input. 2. Apply five 3 A pulses at each input, allowing one second between pulses. 3. Capture the output voltage during each pulse. Results: Save the waveforms and record minimum and maximum voltage. Every pulse must remain within the required range.
R3. Motor startup and servo movement shall not cause the logic supply to leave the 3.2–3.4 V range or reset the ESP32 during ten repeated operating cycles.	Equipment: Assembled electronics, oscilloscope, and controller reset log. Procedure: 1. Keep wireless communication active. 2. Run a cycle containing drive motor startup, steering, and marker movement. 3. Repeat ten times while monitoring the logic supply and reset log. Results: Save voltage traces and record any resets. Pass requires no resets and no voltage excursions outside the required range.
R4. Turning the master switch off shall disconnect battery power from both regulators.	Equipment: DMM and assembled power circuit. Procedure: 1. Remove the battery and check continuity through the switched power path in both switch positions. 2. Reconnect power and confirm that turning the switch off removes the battery feed to

	both regulators. Results: Record the switch state and continuity results. The battery feed must be connected when on and disconnected when off.
--	---

2.3.6 Movement and Attachment Subsystem

The movement and attachment subsystem holds the robot against the vertical whiteboard and moves it through the writing paths. Four Adafruit 4639 geared motors [3] will independently drive the 32 mm silicone wheels, with encoder feedback used to regulate their movement. Silicone tires were selected to provide traction against the smooth board surface. A Pololu 3068 steering motor [4] will drive a central pinion that meshes with four equal steering gears, turning the wheel modules together. This arrangement allows the robot to change its travel direction while keeping the wheels parallel. Each module uses the proposed L bracket to position the wheel beneath its steering axis, reducing the offset that causes scrubbing during steering. Chassis magnets will provide attachment force, with their size and spacing selected through testing on the actual board. Their force must provide enough wheel traction to support the robot without creating excessive steering resistance.

Interfaces with Other Subsystems

Interface	Definition
Power	Motor drivers receive the proposed $6.0\text{ V} \pm 0.3\text{ V}$ motor supply and share ground with the ESP32.
Controller	Three TB6612FNG carriers provide six motor channels, of which five are used for the four drive motors and steering motor. PWM and direction signals follow Section 2.3.2.
Sensing	Four drive encoders report wheel rotation, and the AS5600 measures a wheel module's steering shaft angle.
Writing	The chassis carries the marker holder, and movement begins only after the required steering and marker positions are established.

Requirements and Verifications

Requirement	Verification
R1. The magnetic attachment system shall support the complete robot on a vertical whiteboard for at least 60 seconds without detaching or receiving external support.	Equipment: Actual whiteboard, complete robot, scale, timer, and slack safety tether. Procedure: 1. Record the robot's operating mass with its battery and marker installed. 2. Place it on the board in its stationary operating state. 3. Observe it for 60 seconds, ensuring the tether carries no load. Results: Record mass, holding time, and any sliding or detachment. Pass requires attachment throughout the test without support from the tether.
R2. Under the installed magnetic load, all four wheels shall reach within $\pm 2^\circ$ of commanded steering angles of 0° , 45° , 90° , 135° , and 180° , without exceeding the driver's 1 A continuous current rating.	Equipment: Assembled robot, angle reference with uncertainty no greater than $\pm 0.5^\circ$, current measurement equipment, and controller log. Procedure: 1. Command each angle while the robot is attached to the board. 2. Measure each wheel's orientation and the steering current. 3. Repeat while approaching from the opposite direction. Results: Record wheel angles, errors, and current. Every wheel must meet the angle tolerance, and sustained current must remain within the driver rating.
R3. The robot shall remain attached and finish a commanded 300 mm horizontal, vertical, or diagonal movement within 5 mm of the intended endpoint.	Equipment: Whiteboard, ruler or calibrated board grid. Procedure: 1. Raise the marker and mark the starting position of a chassis reference point. 2. Command each movement in both travel directions, repeating each five times. 3. Measure the final reference point against the intended endpoint. Results: Record displacement and endpoint error for every trial. Every error must be no greater than 5 mm, with no detachment.
R4. At the selected maximum writing speed, an edge indication shall cause the robot to stop before any wheel crosses the board boundary.	Equipment: Whiteboard and frame, assembled robot, camera, board markings, and slack tether. Procedure: 1. Record the selected maximum speed and edge sensor positions. 2. Approach each relevant board edge with the marker raised. 3. Record where detection and stopping occur, repeating five times per approach direction. Results: Save video and record the remaining wheel clearance at each stop. No wheel may cross the boundary.

2.4 Tolerance Analysis

Steering Accuracy

The robot needs to follow its planned path closely enough to produce readable letters. Even a small steering error can move the marker noticeably off course during a long stroke or duration of movement. This analysis examines the proposed requirement of finishing within 5 mm of the intended endpoint after a 300 mm movement. If the robot travels the correct distance but moves in the wrong direction by an angle θ , its endpoint error is calculated as follows:

$$\text{Direction error} = 2 \times L \times \sin(|\theta| * 1/2)$$

Direction error	Endpoint error over 300 mm
$\pm 0.5^\circ$	2.62 mm
$\pm 1^\circ$	5.24 mm
$\pm 2^\circ$	10.47 mm

These results show that the earlier $\pm 2^\circ$ steering target is too loose for the proposed endpoint requirement. A tighter target of $\pm 0.5^\circ$ would leave some room for other errors, including wheel slip, incorrect distance estimates, and movement of the marker within its holder.

The AS5600 provides 4,096 angular positions per revolution, corresponding to approximately 0.088° per measurement increment [2]. However, its listed maximum angular nonlinearity is $\pm 1^\circ$, so small measurement increments alone do not guarantee the required accuracy. If the total direction error stays within $\pm 0.5^\circ$ and all remaining endpoint errors stay within 2 mm, the conservative error bound becomes:

$$\text{Total endpoint error} \leq 2 \times 300 \times \sin(0.25^\circ) + 2 \rightarrow \text{Total endpoint error} \leq 4.62 \text{ mm}$$

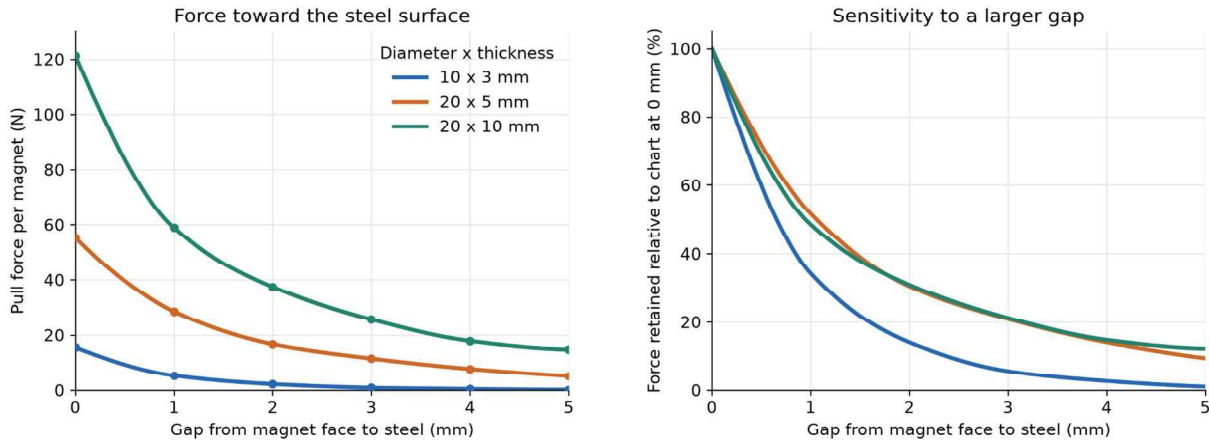
This would meet the proposed 5 mm requirement. These values are targets that still need testing. We will check all four wheel directions, including after steering reversals, and compare the robot's actual travel with its path.

Magnetic Attachment

The magnetic attachment system must keep the robot against the board while its wheels provide enough traction to support and move its weight. One important mechanical tolerance is the distance between each magnet and the steel backing. This gap includes the whiteboard's surface layers and any clearance beneath the chassis. We used three magnet sizes as a sample for the calculation [11][12][13].

N42 neodymium disc magnets: pull force versus gap

Axial magnetization | Approximate values digitized from manufacturer charts



A magnet that feels strong against bare steel may provide much less force when mounted beneath the robot. Thin steel can also reduce the available pull, so the final assembly must be tested on the intended whiteboard. The wheels' friction provides traction along the board. Assuming the magnets do not touch the surface, the available static friction can be estimated using:

$$\text{Available friction} = \mu \times (n \times F_{\text{magnet}} - F_{\text{pen}})$$

Here, μ is the wheel friction coefficient, n is the number of magnets, F_{magnet} is the pull force of each magnet, and F_{pen} is the outward reaction from the spring loaded marker.

For example, if we use a 1 kg robot with four 20 x 5 mm magnets [1], a friction coefficient of 0.40, a marker reaction of 1 N, a nominal gap of 1 mm with a tolerance of ± 0.5 mm, it would yield the following results:

Because force decreases as the gap grows, the worst case occurs at 1.5 mm. Using a rounded reference pull of 21 N per magnet gives:

$$\text{Available friction} = 0.40 \times (4 \times 21 - 1) = 33.2 \text{ N}$$

A design margin of two times the robot's weight would require:

$$\text{Required friction capacity} = 2 \times 1 \times 9.81 = 19.62 \text{ N}$$

Under these assumptions, the reference force would provide sufficient traction. Each installed magnet would need to supply at least approximately 12.5 N on the actual whiteboard, or roughly 60% of the reference chart value. This gives us a measurable target for evaluating the magnets and their mounting gap. We will verify this by measuring pull force with spacers of known thickness, weighing the complete robot, and measuring wheel traction on the intended board. The motors must also provide enough holding and climbing torque.

3. Cost and Schedule

Labor

Illinois ECE graduates in AY 2024–25 earned an average starting salary of \$87,429 in Electrical Engineering and \$119,064 in Computer Engineering [10]. Over 2,080 working hours per year, this corresponds to about \$42–57 per hour, so we assume \$50/hour. Each team member expects to spend about 10 hours per week over the 15-week semester, or 150 hours. Labor for each member is calculated as (\$/hour) x 2.5 x hours to complete.

Table 3.1: Labor cost

Member	Rate	Overhead multiplier	Hours	Total
Morris Yu	\$50/hr	2.5	150	\$18,750.00
Avneet Kaur	\$50/hr	2.5	150	\$18,750.00
Saj Patel	\$50/hr	2.5	150	\$18,750.00
Total labor				\$56,250.00

Parts

Table 3.2: Bill of materials

Description and specifications	Manufacturer	Part #	Qty	Unit cost	Total	Purchase link
ESP32 Wi-Fi/Bluetooth module for the PCB; 4 MB flash, 3.0–3.6 V	Espressif	ESP32-WROOM-32E-N4	1	\$4.99	\$4.99	Mouser
Drive gearmotor with magnetic quadrature encoder; N20, 6 V, 1:100	Adafruit	4639	4	\$12.50	\$50.00	Adafruit
Wheels, 32 x 7 mm silicone tire, pair; fits 3 mm D-shaft	Pololu	1087	2	\$4.45	\$8.90	Pololu
Steering gearmotor; 250:1 micro metal HPCB, 6 V, 130 RPM, 1.5 A stall	Pololu	3068	1	\$26.45	\$26.45	Pololu
Dual motor driver carrier (TB6612FNG); 4.5–13.5 V, 1 A continuous / 3 A peak per channel	Pololu	713	3	\$4.95	\$14.85	Pololu
Attachment magnet; 19.05 x 4.76 mm disc, N52, 8.7 kg pull	K&J Magnetics	DC3-N52	4	\$4.19	\$16.76	K&J
Magnetic steering angle sensor (AS5600); 12-bit, I ² C	Adafruit	6357	1	\$5.95	\$5.95	Adafruit

Diametric magnet for AS5600; 6.35 x 3.18 mm, N42	K&J Magnetics	D42DIA	1	\$0.46	\$0.46	K&J
IMU (MPU-6050); 3-axis accelerometer and gyroscope, I ² C, 3.3 V	Adafruit	3886	1	\$12.95	\$12.95	Adafruit
Marker lift servo; metal gear, 2.2 kg·cm at 6 V	TowerPro	MG90S	1	\$3.02	\$3.02	ReadyMadeRC
Compression spring assortment for marker holder; 200 pieces	Harbor Freight	67562	1	\$4.99	\$4.99	Harbor Freight
2S LiPo battery; 7.4 V, 1000 mAh, 25C	SparkFun	PRT-11855	1	\$16.95	\$16.95	SparkFun
3.3 V step-down regulator (D24V10F3); 1 A output	Pololu	2830	1	\$12.95	\$12.95	Pololu
6 V step-down regulator (D36V50F6); 6.5–50 V input, 5.5 A	Pololu	4092	1	\$39.95	\$39.95	Pololu
Reverse-polarity protection P-MOSFET; 30 V, 3.8 A, SOT-23	Diodes Inc.	DMP3098L-7	1	\$0.50	\$0.50	Mouser
LiPo balance charger; 2–4S, 30 W	SkyRC	e430	1	\$32.00	\$32.00	RCXX
Screw and standoff set; M2.5 nylon, 380 pieces	Adafruit	3299	1	\$16.95	\$16.95	Adafruit
3D-printed chassis, L brackets, steering gears, central pinion, marker holder; PLA	In-house	N/A	1	\$0.00	\$0.00	N/A
Custom PCB, 2-layer, 100 x 100 mm (department funded)	PCBWay	N/A	1	\$0.00	\$0.00	PCBWay
Total parts					\$268.62	

Lab Equipment and Shop Services

Verification uses the ECE 445 lab's oscilloscope, digital multimeter, bench power supply, electronic load, and logic analyzer at no cost to the team. All mechanical parts are 3D printed in-house in PLA, so no machine shop labor hours are required.

Grand Total

$$\text{Grand total} = \text{Labor} + \text{Parts} = \$56,250.00 + \$268.62 = \$56,518.62$$

3.2 Schedule

Week	Task	Person
October 5 to October 11	Verify power requirements and controller pin assignments and finish the schematic and begin PCB layout.	Morris
	Print the first chassis, steering gears, wheel brackets, and marker holder. Implement wireless text reception.	Saj
	Develop the drive and steering control loops. Calibrate encoder direction and steering angle readings.	Avneet
October 12 to October 18	Complete the PCB audit and submit the PCB order by October 15 at 4:45 p.m.	Morris
	Demonstrate wireless commands, motor operation, and pen movement using the prototype wiring.	Saj
	Demonstrate sensor feedback and basic motion control. Record the breadboard test results.	Avneet
	Complete the breadboard demo during October 12 to 14 and submit Teamwork Evaluation I by October 14.	Everyone
October 19 to October 25	Prepare PCB assembly parts and the wiring harness; test the battery and regulators under motor and servo loads.	Morris
	Assemble the mechanical prototype and implement character paths, pen lift commands, and repositioning logic.	Saj
	Test magnetic attachment, wheel alignment, and horizontal, vertical, and diagonal movement.	Avneet
	Integrate the prototype and make sure all systems work as intended with dev boards.	Everyone
October 26 to November 1	Assemble and test the PCB when it arrives and submit an audited revision by October 29 at 4:45 p.m. if needed.	Morris
	Connect the firmware to the assembled hardware and adjust mechanical parts identified during testing.	Saj

	Test with PCB and make sure all subsystems still work, but with a PCB this time.	Avneet
	Prepare a repeatable demonstration of the working subsystems.	Everyone
November 2 to November 8	Resolve any PCB and power issues found during the progress demo.	Morris
	Implement software to write the first letters and note any feedback for change and improvements.	Saj
	Test repeated writing attempts and identify the main sources of positioning error.	Avneet
	Complete the progress demo during November 2 to 4 and submit individual progress reports by November 4.	Everyone
November 9 to November 15	Finalize wiring, connectors, and circuit protection; verify operation using onboard battery power.	Morris
	Finish the wireless interface, input validation, and firmware fault handling.	Saj
	Carry out the requirements and verification procedures and organize the results into tables after making necessary adjustments and improvements to ensure the error is within acceptable range.	Avneet
	Submit the team contract assessment by November 13 and review any remaining problems together.	Everyone
November 16 to November 22	Check battery operation and power stability; resolve remaining electrical issues.	Morris
	Complete firmware and mechanical fixes; prepare the demonstration text and operating sequence.	Saj
	Repeat tests affected by revisions and confirm movement, writing, and edge detection behavior.	Avneet
	Complete the mock demo during the weekly TA meeting and the mock presentation during November 19 to 20. Also prepare the final report draft and presentation slides using the collected results.	Everyone
November 23 to November 29	Edit the electrical design, power analysis, cost information, and notebook records.	Morris

	Edit the firmware and mechanical design descriptions and prepare demonstration materials.	Saj
	Edit the verification results, error analysis, and discussion of project performance.	Avneet
	Fall break so no testing, only going to work on written reports and stuff that is accessible from home.	Everyone
November 30 to December 6	Prepare the robot, battery, and hardware for the final demonstration.	Morris
	Verify the final firmware and demonstration sequence; help record the project video.	Saj
	Organize the final measurements and support the demonstration and presentation.	Avneet
	Complete the final demo during November 30 to December 2 and the final presentation during December 3 to 4, and submit the video assignment by Dec 4.	Everyone
December 7 to December 10	Finish and submit the final paper by December 9. Complete the lab notebook, including final measurements and design changes, and submit it by December 10.	Everyone

4. Ethics, Safety, Societal Impact, and Engineering Standards

4.1 Societal Impact

The Whiteboard assistant is intended to reduce the amount of repetitive physical writing required during classroom, laboratory, and meeting activities. Conventional whiteboards require a presenter to repeatedly turn away from the audience to write information, which can interrupt the flow of a presentation. The proposed system allows text prepared on a computer to be transferred directly to an existing whiteboard while preserving the ability for the users to continue writing on the whiteboard manually. The system may also improve accessibility for users who have difficulty reaching or writing across a large vertical surface.

Several limitations must be considered. The robot is designed primarily for compatible steel-backed magnetic whiteboards, meaning it will not operate on every existing writing surface. Writing speed and quality may also initially be lower than manual writing. In addition, because the device accepts commands wirelessly, operation should remain restricted to authorized users so that unintended commands cannot interrupt a presentation or cause unexpected robot movement.

4.2 Engineering Standards

Several engineering standards and regulatory considerations are relevant to the Whiteboard Assistant. ISO 12100:2010, Safety of machinery - General principles for design - Risk assessment and risk reduction, provides a framework for identifying mechanical hazards and reducing the risk through design [7]. This is relevant to the robot because it contains moving wheels, gears, and a moving mechanism, and an automatically actuated marker. The project applies the risk-reduction approach by using the edge detection, software fault handling, controlled motor shutdown, and physical separation of moving components where practical. Because the proposed design uses a rechargeable lithium battery, IEC 62133-2:2017+A1:2021 is also relevant [8]. This standard addresses the safety requirements and testing for portable rechargeable lithium cells and batteries under the intended use and reasonably foreseeable misuse. The battery and power design should follow the same general safety principles, including protection against short circuits, improper charging and mechanical damage. The final device also communicates wirelessly through the ESP32 [6]. Therefore, the radio-frequency emissions and interference requirements applicable to Wi-Fi/Bluetooth devices must be considered if the design were developed into a commercial product. For the prototype, using an approved ESP32 module and maintaining the manufacturer's recommended RF layout and operating conditions reduces the likelihood of unintended radio-frequency behavior.

4.3 Safety

The primary safety concern is preventing the robot from detaching from the vertical whiteboard. The magnetic attachment system must support the complete robot while also providing the sufficient normal force for the wheels to generate traction. The design requires the robot to be attached

without support and requires commanded motion to be completed without detachment. A slack safety tether will be used during testing so that a failure of magnetic attachment system does not allow the prototype to fall onto a person or equipment.

A second hazard is the robot driving beyond the edge of the whiteboard. Edge sensors mounted near the chassis perimeter will detect when the supporting surface is no longer present. An edge detection takes priority over normal movement commands, and the controller will disable drive and steering motors and command the marker upward when a fault is detected. The final system will be tested to verify that no wheel crosses the board boundary at the selected maximum writing speed. Mechanical hazards are also present because the robot contains rotating wheels, gears, shafts, and a servo-actuated marker mechanism. Testing and adjustment should be performed with motor power disabled whenever practical, and fingers should be kept away from the steering gears and drive assemblies while the system is operating. The marker will be retracted before repositioning to prevent unintended marks and reduce the chance of excessive contact force against the board.

Electrical safety is primarily associated with the battery and the motor power system. The design includes power switch and fuse and proposes protection against reversed polarity and excessive battery discharge. Power-subsystem verification will check both voltage regulation and controller stability during repeated motor and servo operation.

4.4 Ethics

As the Whiteboard Assistant is intended to operate autonomously on a vertical surface, a mechanical, sensing, or a control failure could cause the robot to move or fall unexpectedly. The design therefore prioritizes attachment verification, edge detection, fault-triggered motor shutdown, controlled marker retraction, and protected battery operation. These decisions are consistent with the IEEE Code of Ethics, which emphasizes to place public safety, health, and welfare first [9].

The team must also accurately communicate the capabilities and limitations of the prototype. Successful laboratory tests will not fully demonstrate that the device is safe for every whiteboard, environment, battery, or operating condition. For example, magnetic holding force depends on the construction of the whiteboard and the magnet-to-steel gap, while the position accuracy can be affected by the steering error and wheel slip. Therefore, the reported results should include the actual conditions and measurement tolerances used during testing rather than claiming universal performance.

Accessibility is another ethical consideration. One motivation for this project is reducing the physical effort required to write across large vertical whiteboards, particularly for users with limited reach or mobility. The interface should therefore remain simple and should not necessarily require specialized hardware. At the same time, the project should avoid implying that the prototype provides a complete accessible solution, since the individual users may have needs outside of the scope of the design.

Finally, the users should remain in control of what the robot operates as the robot accepts a complete writing command and performs it autonomously, but invalid commands are rejected and new requests are blocked while an existing task is active. Safety faults override normal writing commands. These behaviors help prevent unexpected or conflicting motion and support a responsible operation of an autonomous electromechanical system.

5. References and Citations

- [1] Webcraft GmbH, “Disc magnet Ø 20 mm, height 5 mm, N42, nickel-plated, S-20-05-N,” supermagnete.de. [Online]. Available: https://www.supermagnete.de/eng/disc-magnets-neodymium/disc-magnet-20mm-5mm_S-20-05-N.
- [2] ams AG, AS5600 12-Bit Programmable Contactless Potentiometer, Datasheet, 2018. [Online]. Available: https://www.mouser.com/datasheet/2/588/AS5600_DS000365_5-00-1877365.pdf.
- [3] Adafruit Industries, “N20 DC Motor with Magnetic Encoder — 6 V with 1:100 Gear Ratio,” Product 4639. [Online]. Available: <https://www.adafruit.com/product/4639>.
- [4] Pololu Corp., “250:1 Micro Metal Gearmotor HPCB 6V,” Item 3068. [Online]. Available: <https://www.pololu.com/product/3068>.
- [5] Toshiba Electronic Devices & Storage Corp., TB6612FNG Dual DC Motor Driver IC, Datasheet, Oct. 2014. [Online]. Available: https://toshiba.semicon-storage.com/info/TB6612FNG_datasheet_en_20141001.pdf.
- [6] Espressif Systems, ESP32 Series Datasheet, ver. 5.3. [Online]. Available: https://documentation.espressif.com/esp32_datasheet_en.html.
- [7] International Organization for Standardization, Safety of Machinery — General Principles for Design — Risk Assessment and Risk Reduction, ISO 12100:2010, Nov. 2010. [Online]. Available: <https://www.iso.org/standard/51528.html>.
- [8] International Electrotechnical Commission, Secondary Cells and Batteries Containing Alkaline or Other Non-acid Electrolytes — Safety Requirements for Portable Sealed Secondary Cells, and for Batteries Made from Them, for Use in Portable Applications — Part 2: Lithium Systems, IEC 62133-2:2017+AMD1:2021, Jul. 2021. [Online]. Available: <https://webstore.iec.ch/en/publication/70017>.
- [9] IEEE, “IEEE Code of Ethics,” 2020. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>.
- [10] University of Illinois Urbana-Champaign, Department of Electrical and Computer Engineering, “Salary Averages,” ece.illinois.edu. [Online]. Available: <https://ece.illinois.edu/admissions/why-ece/salary-averages>.
- [11] Webcraft GmbH, “Disc magnet Ø 10 mm, height 3 mm, N42, nickel-plated, S-10-03-N,” supermagnete.de. [Online]. Available: https://www.supermagnete.de/eng/disc-magnets-neodymium/disc-magnet-10mm-3mm_S-10-03-N.

[12] Webcraft GmbH, "Disc magnet Ø 20 mm, height 5 mm, N42, nickel-plated, S-20-05-N," supermagnete.de. [Online]. Available: https://www.supermagnete.de/eng/disc-magnets-neodymium/disc-magnet-20mm-5mm_S-20-05-N.

[13] Webcraft GmbH, "Disc magnet Ø 20 mm, height 10 mm, N42, nickel-plated, S-20-10-N," supermagnete.de. [Online]. Available: https://www.supermagnete.de/eng/disc-magnets-neodymium/disc-magnet-20mm-10mm_S-20-10-N