Open Mobility

Inter-Semester Project Report

Fall 2024

Prepared by:

Tyler Blazoff

Joel Diccion

Trevor Dill

Alex Nigrine

Noah Schmidt

Malachi Strong

Table of Contents

Current Progress Image

Figure 1: The Assembled Board With Electronics on Top

Project Description

In short, Open Mobility is an open source, modular electric skateboard. Existing electric skateboards on the market are monolithic in design, meaning that user reconfiguration or remixing is impractical or impossible. Our goal is to design an open source product that users in the community can replicate, modify, add to, and use with relative ease at a modest price point. To aid in achieving this, as many of the most complicated parts as possible are bought off the shelf or found in the OSHE lab, with emphasis on general availability. We want each subsystem to be quickly swappable, and not tied down the individual components if availability changes.

The main priority for the Fall 2024 semester was to determine and acquire all of the major hardware components, as well as configure them to operate independently. The majority of this was achieved, and can be found in the *characterization data* section of the report. At our current progress, we are in a good position to fully integrate the major subsystems starting in Spring 2025.

Methodology in Brief

Control Methodology

After determining a general layout and parts list for the control slice, we researched different microcontrollers and receivers to find the best fits for the project. The Arduino Nano, ESP8266, ESP32, and the Stm32f103 were the main ones considered. After taking into account the properties we were looking for from a microcontroller, mainly price, size, ease of use, and CAN compatibility, the ESP32 was chosen.

The receiver requirements were not too stiff, as signal processing can be handled by the ESP32. In the name of open source friendly-ness we opted for a common RC hobbyist receiver and handheld controller package available at a low cost online. This receiver outputs a logic-level PWM signal.

Once we had the microcontroller and the receiver, we had to determine the best way to get the microcontroller to receive the output signal from the receiver. Using an oscilloscope, we could see that the signal was a PWM waveform and that changing the throttle changed the duty cycle. The first attempt at reading the voltage using the MATLAB function readVoltage produced inconsistent readings. After switching to the Arduino IDE, the function pulseIn gave much better results.

Part of the semester deliverables was the ability to turn on and off the slice. We found a switch in the lab, and after soldering some wires to it, used it to control whether or not the system receives power. There was also an issue when powering the ESP32 and reciever separately would cause them to have different grounds and would give out funny readings. Having the entire system on one power source, either from a laptop or a power generator, alleviates that problem.

Currently, the microcontroller is just reading in the signal and outputting to a laptop through the Arduino IDE, constantly updating numerical values between -100 and 100, and also to several LEDs on a breadboard, one lights up for high throttle, one lights up for neutral, and one lights up for low.

Battery Methodology

The biggest considerations for an electric skateboard battery are size, capacity, and cost. The physical size of the battery is primarily constrained by the dimensions of the board and the clearance from the ground. The capacity of the battery will determine the range of the skateboard, so a larger capacity is preferred. These two facts lead to the conclusion that energy density is extremely important for the skateboard. Lithium-ion is the best technology for energy density, making it an easy choice for the battery chemistry.

After determining what type of battery to use we were split on whether to use an off-the-shelf battery pack or assemble one ourselves. We decided to purchase a battery pack for a few main reasons: the risk of damage during assembly was significant - potentially leading to high costs and increased safety risks, we would be able to use the included BMS instead of buying a separate one or building one ourselves, and it is much more accessible to anyone who wants to recreate this product in the future.

After deciding to purchase a battery pack we had to determine what pack to buy. We created a Pugh Matrix; weights were assigned from one to ten, with ten being most important, and criteria values were assigned between one and ten based on how well the particular pack met our requirements. We came to the conclusion that the 6S2P Transparent Series battery pack from Moboards was our best choice primarily due to its low cost and adequate other specifications.

	12S2P	21700 Samsung 50S 12S4P	WowGo AT2 (36V 10S4P)	6s2p P42A Transparent Series	
Criteria					Weight
Price	6	1	5	8	7
Output	5	8	6	5	6
Size	5	5	5	7	5
Weight	5	5	5	6	3
Modularity	5	2	2	7	5
BMS	4	4	4	6	5
Capacity	6	8	7	5	4
Total	181	157	169	224	

Figure 2: Pugh Matrix for Battery Pack Selection

The other primary requirement for the battery module was to include under voltage and over current protection. This is largely redundant because the VESC is able to monitor and control those, however the point is to prevent injury to the user and damage to the battery and/or other components in the event of a catastrophic failure such as a short of the battery bus. We found the LM5069 IC from TI. The LM5069 will allow us to set a maximum and minimum voltage and a maximum current. Design was straightforward as TI provides a calculator to determine component values and placements [1].

Drive Methodology

The vast majority of personal transport vehicles (e-bikes, e-scooters, e-skateboards, etc) use brushless DC (BLDC) motors due to their efficiency and torque density, as such, this is the motor construction that makes the most sense for Open Mobility. These motors however, are difficult to control and require complex hardware and software for proper electrical commutation. For small vehicles, this is often accomplished using an electronic speed controller (ESC). ESC’s can be bought off the shelf for minimal cost due to their proliferation in hobbyist radio control, but they are typically very specific to their application; off the shelf ESC’s are usually designed and programmed at the at the factory with specific battery voltage, motor parameters, motor sensor type, and control limits. Modifying these parameters is either impossible (baked into the hardware), or requires extensive “hacking” experience. For these reasons, we researched and procured a Vedder ESC (VESC). This is an open source ESC designed by Benjamin Vedder, that allows for full control over the parameters listed above. Advantages of the VESC include:

- Control and telemetry via CAN bus, or control via PPM
- Sensorless control, or sensored control such has hall effect
- Field oriented control option
- Wide range of battery configurations accepted
- Wide range of motor sizes accepted
- Sophisticated programming controla and GUI via the VESC tool
- Totally open source, software is freely available via github
- Automatic motor configuration

These features allow for anyone who follows our procedure to use almost any motor or battery configuration they want, as well as any VESC as long as it follows the VESC protocol. Additionally, the auto setup feature allows for someone with little engineering experience to have access to this product.

Mechanical Methodology

Skateboard decks, including most electric skateboards, are typically manufactured out of wood composite (layers of wood sheet compressed and soaked with epoxy). This material is strong, lightweight, and relatively cheap to manufacture, making it well suited for its application. The only issue is that most individuals do not have the variety of tools or knowledge to manufacture a composite wood deck themselves; this leaves store-bought boards as the only option for the majority of people. Because we wanted to design a repeatable base product, we did not want the design to rely on the availability of a specific board deck, as such we went with a 3D printed deck design. With a 3D printed deck, the entire board was divided into subsections by subsystem which were designed to come apart completely. Each subsection of the deck can be seen clearly in the *Current Progress Image* section.

The mechanical theory behind this design is that metal rods run the length of the board and are under very high tension; this puts the plastic deck under compression. In this configuration, weight from the rider will materialize in the structure of the board as added compression on the top part of the plastic deck, and added tension on the rod, rather than a bending moment on the rod or plastic. This optimizes the force along the structural properties the rods and plastic are strong in. With this understanding, hardened steel was chosen for the rods due to a higher modulus of elasticity, and PETG was used for the plastic due to higher mechanical shock resistance, and environmental resistance (UV, heat, chemical).

With this design, whoever follows our procedures will have exactly the same board deck as us. The primary benefit to this is that anyone who wants to remix the project will do so without worrying about the uniqueness of their or anyone else’s board. Initially, we believed this to be more open-source friendly. After physical testing and characterization, it was found that while a 3D printed deck will work, it is not the most open source friendly option.

A better option would be to make a modular design that will work on *any* board, including a 3D printed one. This is less wasteful as well since un-powered boards can be recycled into a powered board. Changing this would not impact the overall benefit of a modular design, as the user can still configure their drive package for their use case unlike what is currently on the market. Another added benefit is this will make the project more price competitive, as a cheap wood board would be cheaper than the cost of materials for a 3D printed board.

Bill of Materials (BOM)

Module	Item	Link	Quantity	Cost
Battery	6s2p P42A Battery Pack	https://www.mboards.co/collections/batteries/products/6s-p42a-battery-pack-transparent-series	1	$198.53
Battery	Hot Swap Controller LM5069MM-2/NOPB	LM5069 data sheet, product information and support | TI.com	1	$3.89
Control	ESP-WROOM-32 Type C	Amazon.com: ELEGOO 3PCS ESP-WROOM-32 Development Board Micro-USB, 2.4GHz Dual Mode WiFi+Bluetooth Dual Core Microcontroller for Arduino IDE, Support AP/STA/AP+STA, CP2102 Chip : Electronics	3	$15.99
Control	Vanpro V2	Amazon.com: VANPRO Electric Skateboard DIY V2 Edition 2.4G Mini Wireless Remote Control Receiver Transmit 80 Meters Away Wireless : Sports & Outdoors	1	$19.99
Control	1K Ohm resistor	Found in lab	3	$0.30
Control	Breadboard	Found in lab	1	$6.57
Control	On/Off Switch	Found in lab	1	$0.68
Control	Wires	Found in lab	11	$11.99
Drive	ESC	https://www.makerx-tech.com/collections/x-esc/products/vesc4	1	$60.00
Drive	XT60 connector	Found in Lab	1	$1.60
Drive	XT90	Found in Lab	1	$3.90

	connector			
Drive	Motor	Turnigy Aerodrive SK3 - 6374-149KV Brushless Outrunner Motor (no longer available) or any other	1	~ $60
Mechanical	Trucks	https://www.tactics.com/caliber/caliber-ii-fifty-longboard-trucks	1	$52.95
Mechanical	⅜“ x 36” Threaded Rod	From store	3	$16.47
Mechanical	⅜” nuts	Found in Lab	6	$11.98
Mechanical	⅜” washers	Found in Lab	6	$6.97
Mechanical	M5 x 30mm machine screws	Found in Lab	8	$8.99
Mechanical	M5 Nuts	Found in Lab	8	$5.23
Total:				$486.03

Table 1: BOM

Tools Used

Tool	Use	Cost
3D Printer	Ender 3 S1 Pro	$249.00
Computer	Program ESP32 and VESC	$500
Oscilloscope (optional)	View waveforms and find issues - not required for assembly	$179.99
Multimeter	Verify voltages	$34.97
Heat gun (lighter works too)	Shrink tube over connectors	$16.99
Soldering station	Solder connectors	$139.98
Crescent wrenches	Tighten board rods and mount trucks	$27.99
Ratchet and Socket	Tighten board rods	$69.00
Drill	Drill holes for trucks and enlarge holes for rods	$155.46
Drill bits	Drill holes for trucks and enlarge holes for rods	$9.97
Allen Wrenches	Mount trucks	$12.99

Table 2: Tools Used

Assembly Instructions

These instructions are not entirely exhaustive, and primarily include steps that a maker would need to know to recreate our results. This assumes some maker experience, such as how to solder and how to heat shrink a tube on a wire.

Control Assembly

1. Place ESP32 chip on a breadboard.
2. Run a wire from the 3v3 pin to an open space on the breadboard.
    a. Connect this wire to the power in pin on the receiver
3. Connect a wire from the Vin pin to the power rail.
4. Connect a wire from the GND pin to the ground rail.
5. Connect a wire from the receiver ground pin to the ground rail.
6. Connect the signal pin on the receiver to pin 32 on the ESP32
7. Connect 3 LEDs to ground on the breadboard
8. Connect the positive end of each LED to pins 25,26, and 33 on the ESP32
    a. Do this using 1k resistors, with a wire as necessary.
9. Connect the power rail to a power supply.
10. Connect the ground rail to a power supply with a switch in series.
11. Use the below photo for further guidance.

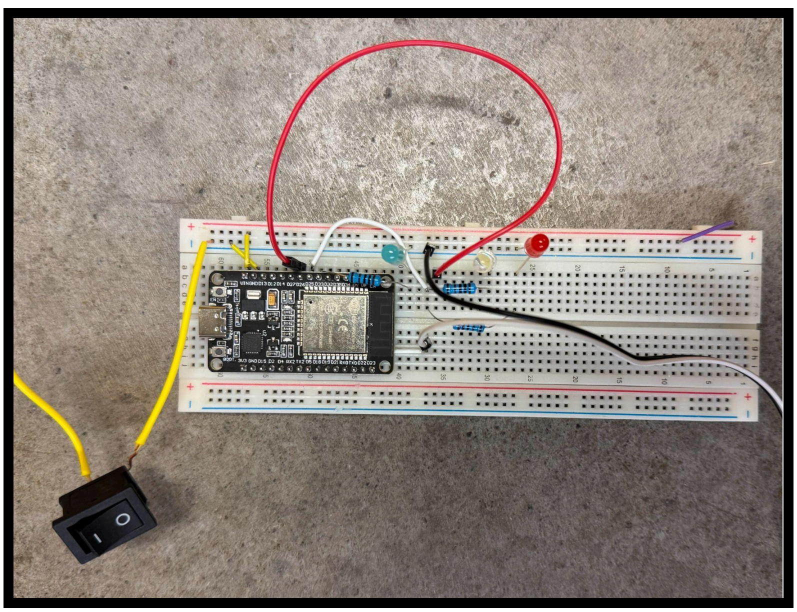

Figure 3: ESP32 on Breadboard to Receive Controller Signal

Battery Assembly

Make sure to familiarize yourself with Lithium Ion battery safety! There are many resources available online from the RC and skateboard communities. In general, never overcharge the battery, never exceed the charging and discharging current limit, and never let the battery drop below its minimum voltage. You should also store the battery at storage voltage when unused for a prolonged period; look up this term if you are unfamiliar with what your battery's storage voltage is.

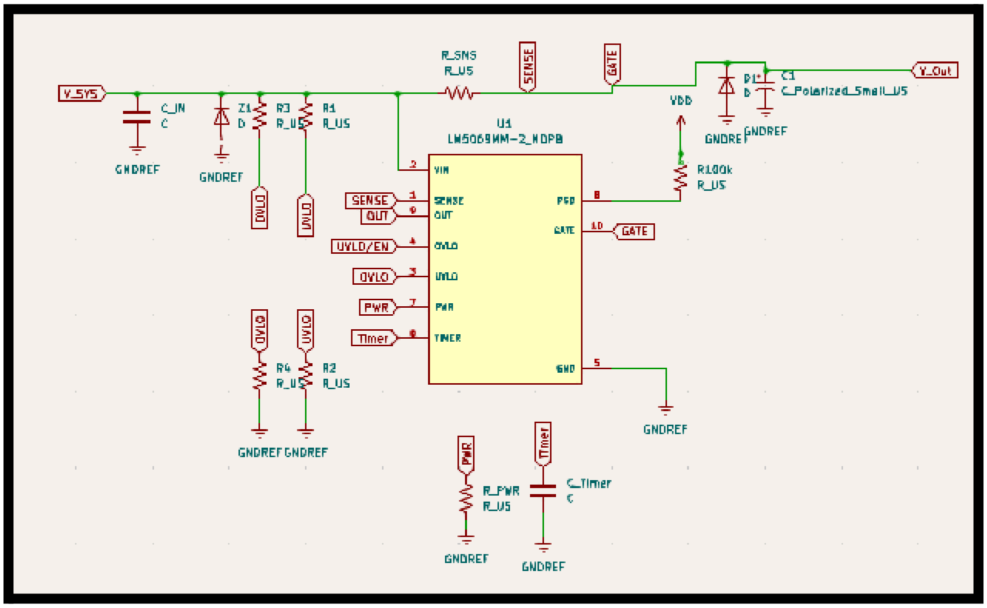

Figure 4: Under Voltage and Over Current Protection Circuit Diagram

1. Connect BMS and charging port to battery pack
    a. Place shrink tube over one side of each connector
    b. Connect red wire from battery pack to red wire of charging port
    c. Connect black wire of BMS to black wire of charging port
    d. Connect blue wire of BMS to black wire of battery pack
    e. Heat the shrink tube to secure connections
2. Check battery voltage using a multimeter
    a. Check manufacture specifications for minimum, maximum, and storage voltage
    b. You can check the voltage of each cell by probing from the ground pin to each other pin on the balance lead
    c. Take care to not short the leads

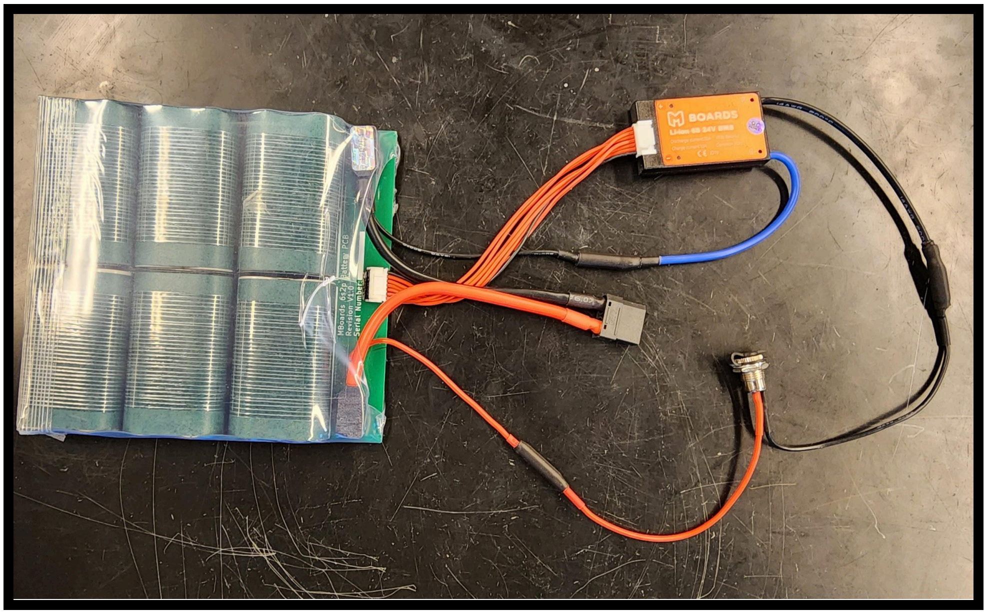

Figure 5: Battery Module; Showing Wiring Example

3. Solder together an adapter if needed, this will depend on the VESC and battery that you use. In this case, we needed to connect an XT90 male to XT60 female.
    a. Ensure that ground on one connector goes to the ground lead on the other connector. Doing this incorrectly will apply reverse voltage to the VESC, and will likely destroy it.
    b. The gauge of wire that you use must be at least as large in diameter as the wire on the batter and VESC if you plan on drawing maximum current.
    c. Tug test your solder joints, and make sure they are not close to shorting

Figure 6: Adapter to Connect VESC and Battery Pack

4. Connect battery pack to VESC using adapter

Drive Assembly

1. Download and install VESC software and follow these tutorials (They are high quality and go over everything you would need to do):
    a. https://www.youtube.com/watch?v=lDuV8cnPRmI&ab_channel=MBoards [4]
    b. https://www.youtube.com/watch?v=JVKFrdCgghQ&ab_channel=RBE-Motion [5]
2. Attach adapter made in *Battery Assembly* to end of VESC

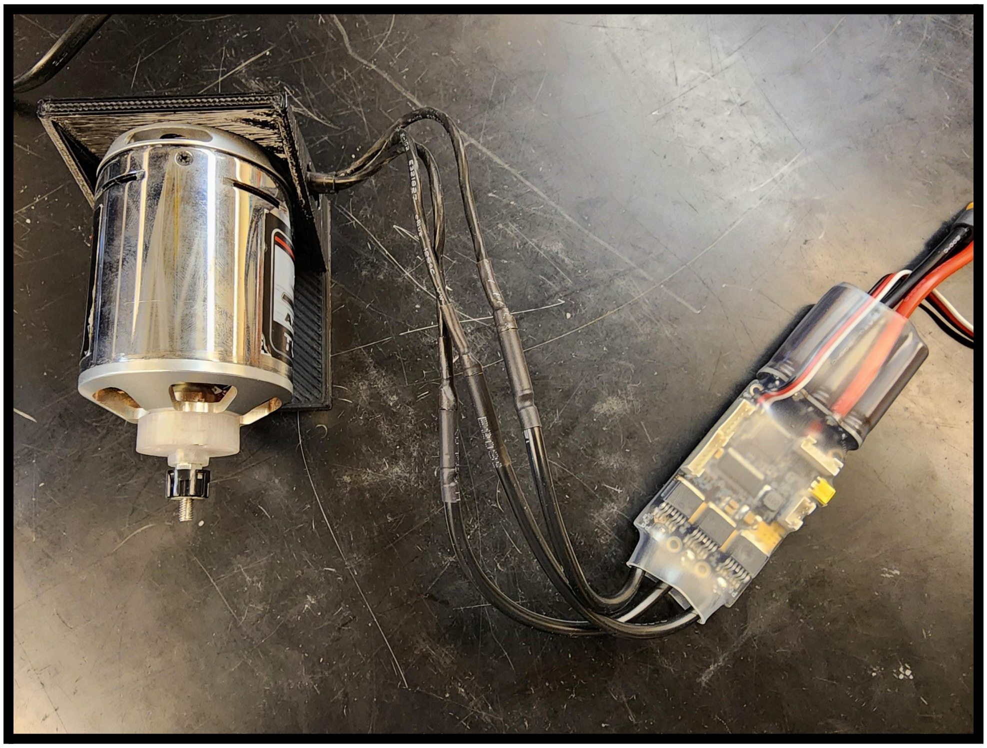

Figure 7: Motor Connected to VESC

3. Plug in controller receiver to receive signal, make sure the polarity is correct

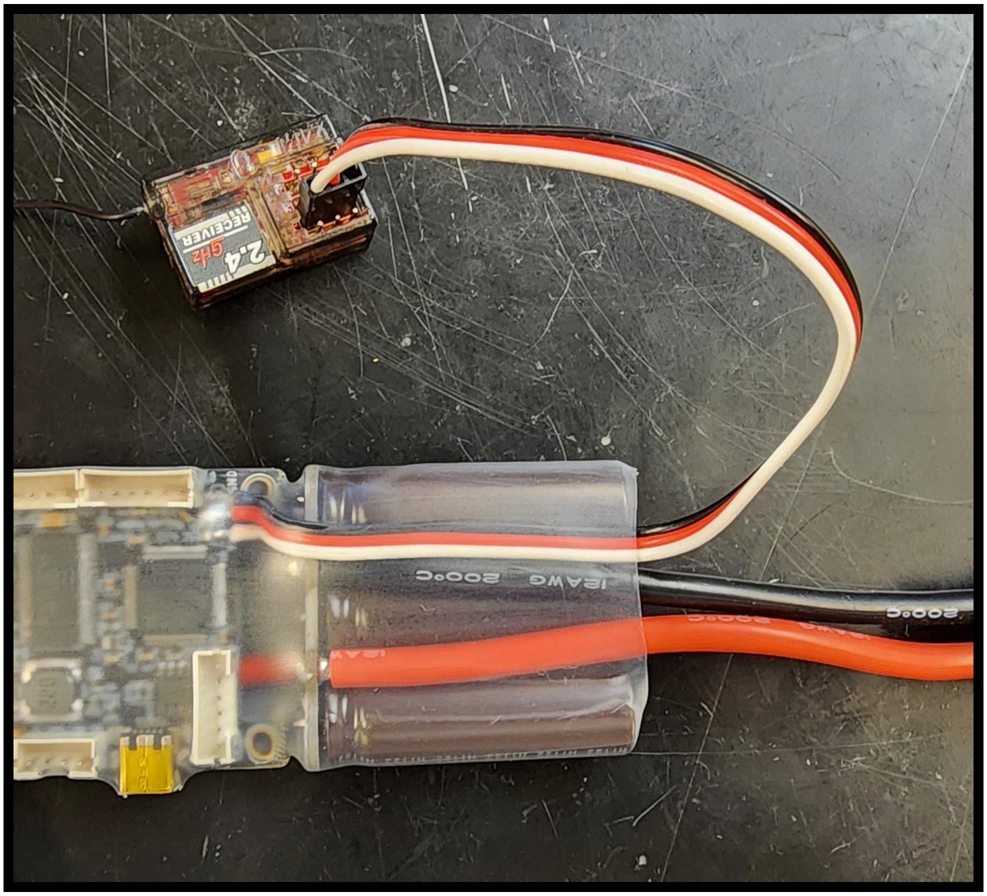

Figure 8: VESC Connected to Wireless Receiver

4. (for configuration) connect VESC to computer via USB
5. Plug the VESC into the battery pack for power using the adaptor if needed
    a. Once again, make sure polarity is correct

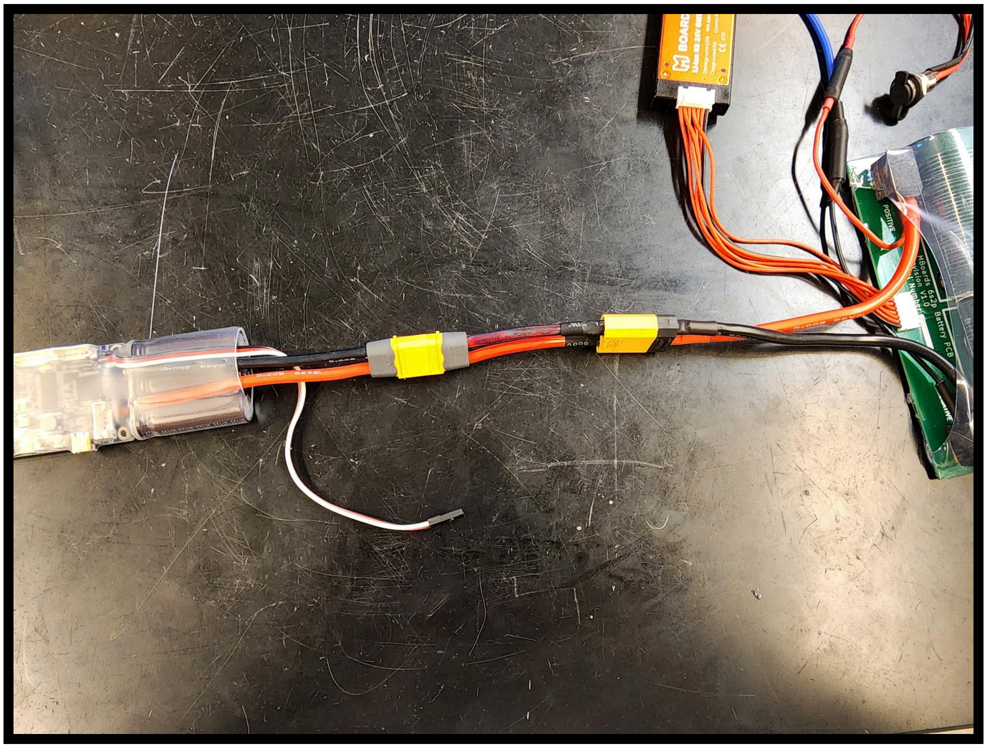

Figure 9: VESC Connected to Battery Pack Using Adapter

6. Open VESC tool on a computer and click "connect"

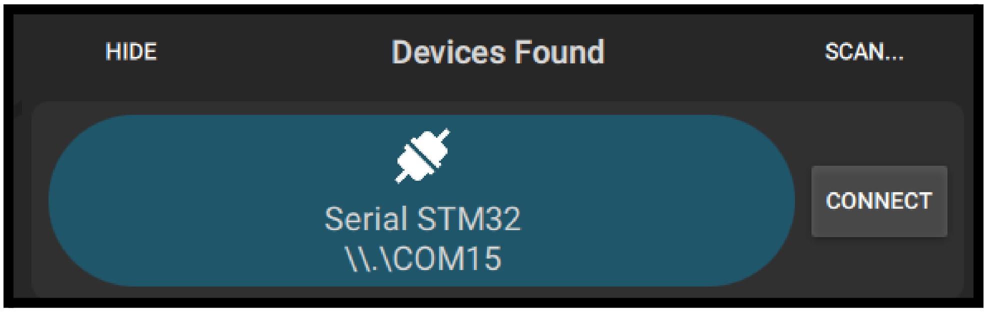

Figure 10: Connect Button in VESC Tool

7. Enable "real time data" and "real time app data" on the far right menu of the VESC tool homepage

Figure 11: Real Time Data Buttons

8. Click "setup input"

Figure 12: Setup Input Button

9. Click “next” when the wizard tool pops up
10. Select “PPM input, such as conventional RC receivers.” and click next

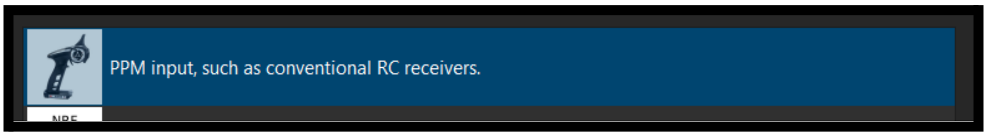

Figure 13: PPM Input Selection

11. Select “Reset” then push the throttle all the way forward then all the way back and click “Apply”
12. Test the throttle looking at the blue bars to ensure the range is correct, if not then try to reset the values again

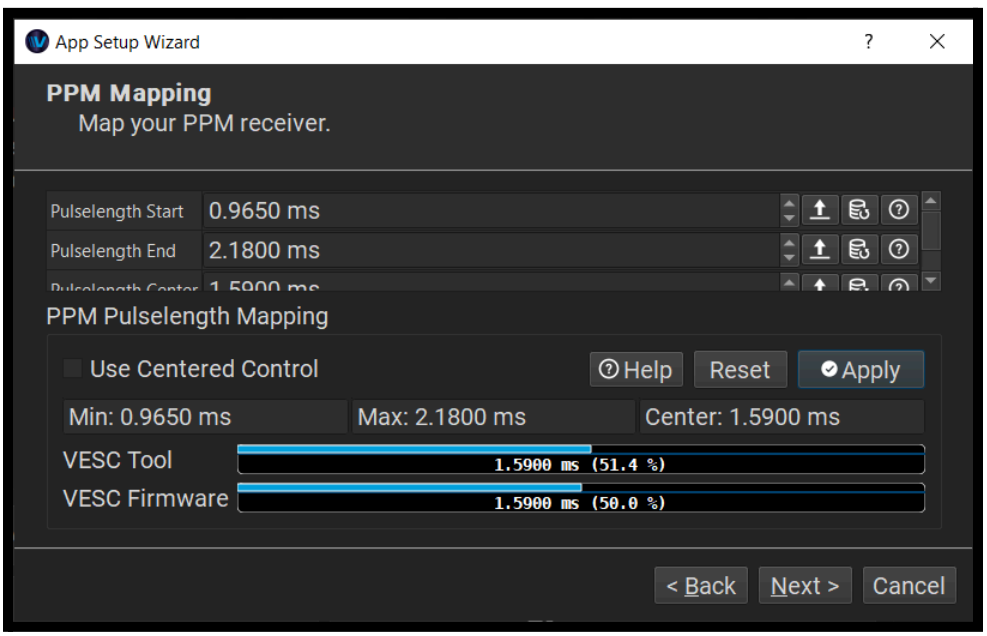

Figure 14: PPM Mapping Window

13. Click “next” until you get to the “conclusion” window, then click “finish”
14. On the left hand side of the screen select “PPM” under “App Settings” then click the “Throttle Curve” tab on top

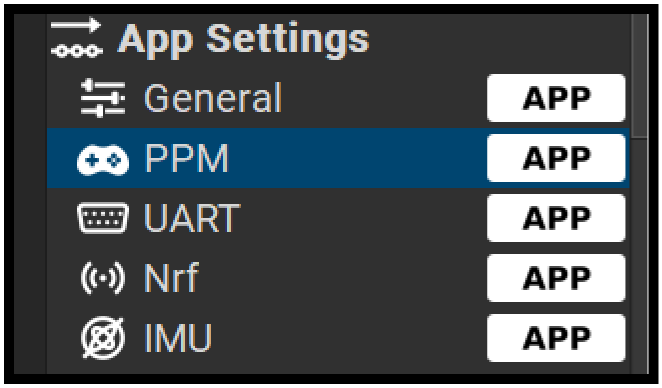

Figure 15: PPM Settings Selector

15. Select “Polynomial” as the Throttle Expo Mode and set Throttle Expo to -100%

Figure 16: Throttle Curve Control

16. Write the app configuration changes using the button on the right hand side

Figure 17: Write App Data Button

Mechanical Assembly

1. Clear out the holes from 3D printed pieces
    a. If the holes were printed the correct diameter then a piece of threaded rod works great here. Simply line it up with the hole and aggressively push it through and strings left from the 3D printing process.
    b. If the holes are too small then they must be enlarged with a 13/32” drill bit
    c. After inserting the rods, they should be able to slide through all pieces of the deck with minimal effort
2. Align the parts and push them together

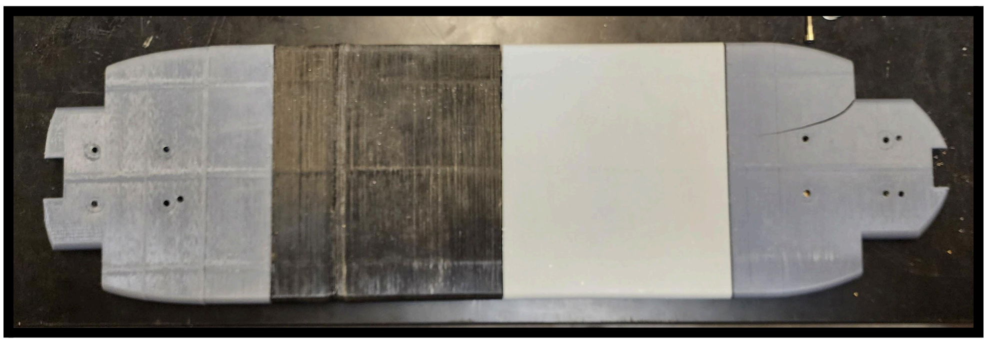

Figure 18: Board PartsLined up

3. Place the threaded rods though
4. Add washers and nuts

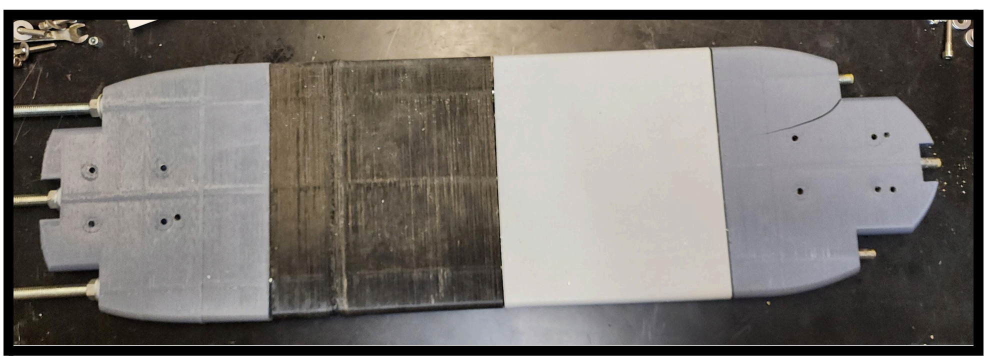

Figure 19: Threaded Rod inserted in Board Parts

5. Tighten evenly between the three rods, ensuring that tension on each rod remains close to equal
6. Center the trucks on the board and mark hole locations
7. Drill holes for the trucks to bolt on
8. Bolt the trucks on
9. Print the wheels
10. Use a dremel to clear out and enlarge the hole until the bearing fits snugly
11. Wrap electrical tape around the wheels to increase friction
12. Bolt the wheels onto the trucks

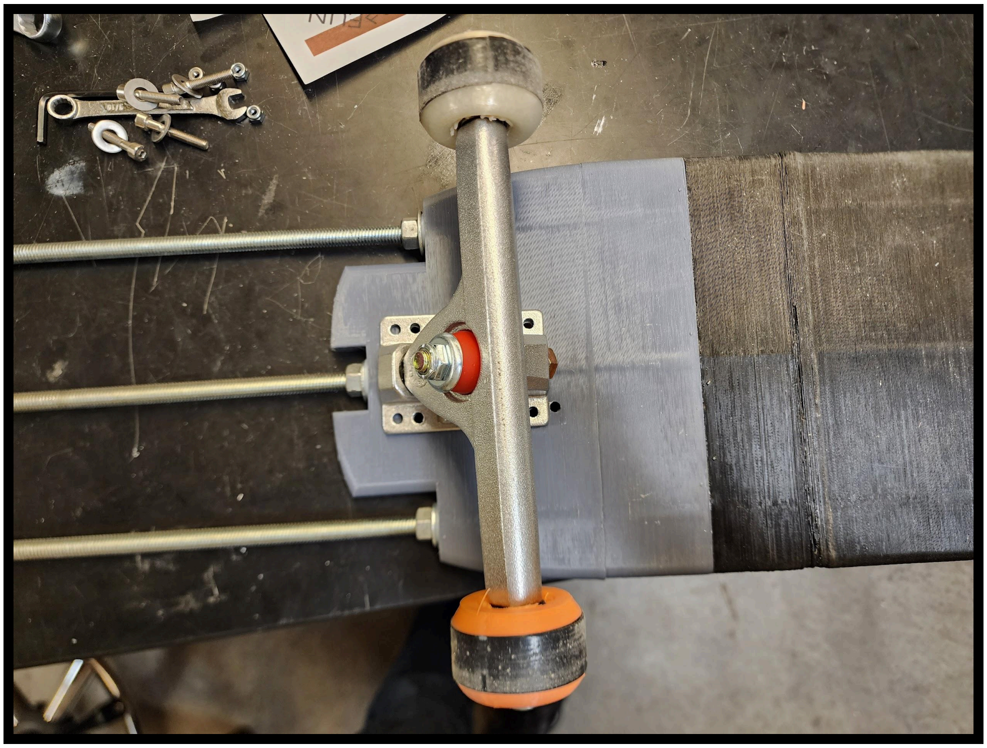

Figure 20: Trucks and Wheels on the Board

Characterization Data

Control Characterization

The control subsystem is to read a PWM input from the receiver that is bound to a handheld controller. The PWM input is to be filtered and output as a duty cycle value that can be output on a CAN bus in the future. Table 3 shows the approximate output duty cycle from the control slice from a given throttle position on the handheld controller.

Controller input	Full Break	Half Break	Neutral/No Throttle	Half Throttle	Full Throttle
Output Duty Cycle	-100%	-50%	0%	50%	100%

Table 3: Control Throttle Parameters

Battery Characterization

The battery is to supply ample current to the VESC, as well as manage changing via a battery management system (BMS). The battery was charged to an acceptable operating voltage, then partially discharged through operation. Each cell on the battery was measured via the balance lead. The cell voltages can be found in Table 4. These voltages show that the BMS is properly balancing the cells during charge and that the internal resistance of each cell is within tolerance relative to each other.

Cell	Raw Voltage (V)	Voltage (V)
1	3.96	3.96
2	7.92	3.96
3	11.9	3.98
4	15.88	3.98
5	19.83	3.95
6	23.80	3.97
Total	23.80	23.80

Table 4: Battery Cell Voltages

Drive Characterization

The VESC is to connect to the VESC tool on a computer via USB, as well as drive the motor and output telemetry to the serial bus. The VESC is also to be programmable for a drop in motor and battery, and limit operating conditions based on user parameters.

The VESC was tuned with a BLDC motor found in the OSHE lab, in this case a sensorless 6374 BLDC motor. The VESC was then tuned to the motor parameters and successfully operated the motor with expected performance.

It was also confirmed that real time telemetry was output to the serial bus and viewed on the VESC tool during operation. Current limits for forward operation and regenerative braking were also reached, and it was confirmed that the VESC limited the current successfully.

The control input to the VESC was PWM, when the device expects a PPM input. This caused issues when interpreting the duty cycle directly from the receiver, as such this makes it difficult to operate the motor. This will be resolved when the duty cycle is set via the CAN bus, as explained in *Control Characterization.*

Mechanical Characterization

The Individual pieces of the board fit together snugly. Interlocking portions help to stabilize the individual pieces against shear forces the threaded rods would be powerless to stop. The individual pieces are hard to push together by hand, but torquing down the threaded rods helps to put them together better. When the threaded rod is removed it is difficult to separate the pieces because of the strong friction fit.

Under testing, the board has been proven to support over 200 lbs, including bouncing. The threaded rods only undergo plastic deformation and return to their original shape after the load is removed. This indicates there is no permanent damage to the threaded rods and they will likely last the life of the board.

The rolling capabilities have been tested by a rider skating down the hallway. Relatively sharp turns (for the size and style of board) were undertaken to prove the efficacy of the design. The ride quality will improve as the cracked portion is redesigned and the wheels are replaced with softer and more uniform wheels.

The front part of the board developed a crack on two occasions. The first occasion was thought to be caused when the part was removed from the 3D printer bed so a second part was printed. The second part appeared to be free of defects. When tensioning the new part onto the board it developed a crack in the same style and mirror location from the first part. This indicates there is probably a design issue, not just a coincidence. A redesign of the part, removing the sharp 90 degree angle, may help to reduce the stress.

Figure 21: Person Riding Assembled Skateboard

Acknowledgements

- Dr. Shane Oberloier for his advisorship
- Michigan Technological University for the facilities and funding
- All OSHE members for their direct and indirect support

External Project Links

OSF Repository

The OSF repository contains all of the CAD files used in the board construction. Navigate to the Mechanical folder in the repository for these files. The motor test stand can be found in the Drive folder.

Link: https://osf.io/nszqb/

Github Page

The Github page contains the code loaded onto the ESP32 modules in the control section.

Link: https://github.com/Lad553/OSHE-Open-Mobility

Design Standards

The Open Source Hardware Association provides the following standards for Open Source Hardware (OSHW) projects [6]:

1. Documentation
2. Scope
3. Necessary Software
4. Derived Works
5. Free Redistribution
6. Attribution
7. No Discrimination against Persons or Groups
8. No Discrimination Against Fields of Endeavor
9. Distribution of License
10. License Must Not Be Specific to a Product
11. License Must Not Restrict Other Hardware or Software
12. License Must Be Technology-Neutral

We meet all of the OSHW requirements for an open source project.

References

[1] “LM5069QUICK-CALC Calculation tool | TI.com,” *Ti.com*, 2020. https://www.ti.com/tool/download/LM5069QUICK-CALC (accessed Nov. 13, 2024).

[2] B. Vedder, “VESC Project,” VESC Project. https://vesc-project.com/ (accessed December 7, 2024).

[3] “Make Moves: 3D Print Your Own Longboard,” LULZBOT. https://lulzbot.com/learn/make-moves-3d-print-your-own-longboard (accessed December 7, 2024).

[4] MBoards “VESC® Tool 2020 Tutorial - How to Program Vesc for DIY Electric Skateboards,” YouTube. https://www.youtube.com/watch?v=lDuV8cnPRmI&ab_channel=MBoards (accessed December 7, 2024).

[5] RB E-Motion “VESC Tool Setup & Programming Guide 2024: Black Mamba ESC Settings (Phase One Pt. 4),” YouTube. https://www.youtube.com/watch?v=JVKFrdCgghQ&ab_channel=RBE-Motion (accessed December 7, 2024).

[6] Open Source Hardware Association (2024, Aug. 31). OSHW, Definition of Free Cultural Works. Accessed on: Dec. 7, 2024. Available: https://freedomdefined.org/OSHW