Logo%20Main%20200

**Boston University**

**Electrical & Computer Engineering**

**EC464 Capstone Senior Design Project**

User's Manual



Bike for Blind

Submitted to

Eshed Ohn-Bar

H2X Lab

8 Saint Mary’s Street, 209

Boston, MA, 02115

eohnbar@bu.edu

by

Team 7

Bike for Blind

Team Members

Linden Adamson [lindena@bu.edu](mailto:lindena@bu.edu)

Jason Calalang [jasonc16@bu.edu](mailto:jasonc16@bu.edu)

Tara Gill [taragill@bu.edu](mailto:taragill@bu.edu)

Zhilang Gui [zgui@bu.edu](mailto:zgui@bu.edu)

Cole Resurreccion [coler@bu.edu](mailto:coler@bu.edu)

Raina Yin [ryin2@bu.edu](mailto:ryin2@bu.edu)

Submitted: 4/18/2025

#### Autonomous Trike User Manual

#### Table of Contents

[Executive Summary](#_heading=h.t1n3os1m0en5)

[1 Introduction 1](#_heading=h.n52fxkogyiwe)

[2 System Overview and Installation 3](#_heading=h.t4kn9rr4lf53)

[2.1 Overview block diagram 3](#_heading=h.dcpk02525kz2)

[2.2 User interface. 4](#_heading=h.l7ouew9nenl9)

[2.3 Physical description. 4](#_heading=h.r2igxqt3vurq)

[2.4 Installation, setup, and support 6](#_heading=h.z2mwbxprgbhy)

[3 Operation of the Project 7](#_heading=h.j99bdoyhk7ga)

[3.1 Operating Mode 1: Normal Operation 7](#_heading=h.1cqo981ewh3b)

[3.2 Operating Mode 2: Abnormal Operations 7](#_heading=h.392b6yrvx18k)

[3.3 Safety Issues 8](#_heading=h.5m2d3bp2zzw5)

[4 Technical Background 9](#_heading=h.p20bgvorrk98)

[4.1 Design 9](#_heading=h.kipwrf6z9eih)

[4.2 Hardware 9](#_heading=h.ks0zjgom5ph8)

[5 Relevant Engineering Standards 13](#_heading=h.ku3517h04xr4)

[6 Cost Breakdown 15](#_heading=h.wueoyfba3vx6)

[7 Appendices 17](#_heading=h.17n9fn7zni74)

[7.1 Appendix A - Specifications 17](#_heading=h.p2imd0f05c4s)

[7.2 Appendix B – Team Information 17](#_heading=h.vgcaws2qlue3)

# 

# Executive Summary

Despite advances in assistive technology, no widespread autonomous cycling solutions currently exist, excluding a community where a reported 87% seek better transportation options, especially as cities shift towards bike-friendly infrastructure1. We developed a semi-autonomous recumbent tricycle to provide visually impaired individuals with independent mobility and exercise opportunities.

Our tricycle combines computer vision, GPS-linked voice commands, and real-time obstacle detection to autonomously control steering and braking while the user pedals independently. The system leverages a Jetson Orin Nano, OAK-D RGB-D camera, VectorNav IMU, and ROS-based control to provide precise control and robust safety mechanisms, including emergency braking, light indicators, and haptic feedback.

Our client, Dan Parker, a blind racecar driver and Guinness World Record holder of “Fastest Speed for a Car Driven Blindfolded”, highlights the transformative potential of this technology and the demand. By bridging the gap in accessible mobility, our project lays the groundwork to expand transportation and fitness opportunities for the visually impaired community and others who are currently excluded and can take advantage of this alternative transportation.

# Introduction

Despite advancements in assistive technologies, no independent, autonomous cycling solutions exist specifically for individuals with visual impairments. This limits their ability to navigate independently and participate in the outdoor exercise of cycling. Our client, Dan Parker, is a blind race car driver who races using guidance systems and currently advocates for expanding accessible mobility options. His experience highlights both the demand and the possibility of assistive navigation technologies. In support of this need, a recent survey found that 87% of visually impaired individuals expressed strong interest in an autonomous mobility solution that supports independent outdoor exercise and transportation. This is especially important in urban environments, which are adapting to bike-friendly infrastructure, as there has been a reported 25% or greater increase in cycling activity over the past four years2.

The project is a semi-autonomous recumbent tricycle designed to provide individuals with visual impairments with independent mobility and exercise opportunities. At our client’s request, the design prioritizes user-powered pedaling while the tricycle autonomously manages navigation, adjusting to rider-driven speed through an accessible interface. To autonomously navigate, the trike uses a Jetson Orin Nano–further referred to as the Orin–to make low-latency decisions based on real-time camera and sensor input. Images gathered from the OAK-D (an RGB-D camera) and a VectorNav IMU serve as inputs into a model computed by the Orin. The Orin communicates through ROS with motors that steer and brake for the user. The steering system uses a 24V DC motor connected to the central steering joint on the tricycle. Attached to the motor is a position encoder that sends position data back to the Orin for precise control. The brakes are controlled by servo motors attached to the wheel that activate similarly to the original brake line through nylon Micro Cord.

What makes our project special is its usability and safety for people with visual impairments. The bike has lights which serve as braking and turning indicators for the safety of the user and their environment. Haptic motors are also attached to the upper lumbar of the seat to alert the user when not to pedal. For user-friendliness, A 90dB buzzer is activated via a 433 MHz RF signal key fob that alerts the user audibly to the tricycle’s location. Our user interface also includes a voice-activated module that uses a USB microphone with CMU Sphinx speech-to-text API to process user commands. Selected by these commands, destinations are linked with GPS data and an offline map for navigation planning. Finally, audio feedback is integrated through vocal callouts for steering and braking decisions, as well as confirmation for destination setting.

Given the system’s autonomous, outdoor operation, safety has been a major priority throughout development. We identified four principal hazards: collision with obstacles (e.g., vehicles, pedestrians), sensor limitations, electrical conduction through the aluminum frame, and system failures resulting in loss of control. We addressed these concerns as follows: redundant braking systems, combined braking and turn-signal lights, emergency-stop functionality, and insulated electronic enclosures.

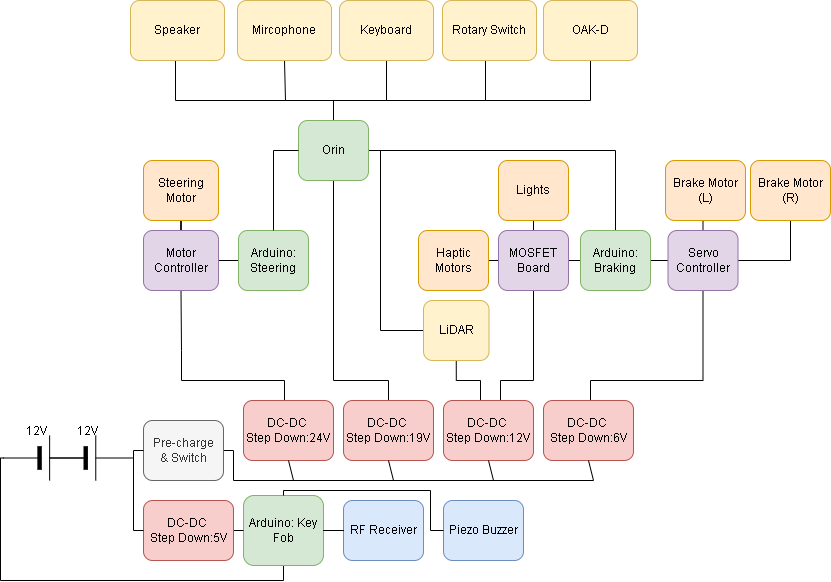
The remainder of this document details the technical design of the semi-autonomous platform and the results that show an accessible future for the visually impaired cycler.

# System Overview and Installation

## Overview block diagram



*Figure 2.1: Simplified block diagram of information flow in the trike. The Orin receives serial data through the sensors, and after processing that data, feeds the information into Nav2, which commands the controller to send serial data to the Arduinos that control the motor. One main safety feature–the Emergency Brake–directly overrides the controller if hazards are detected.*

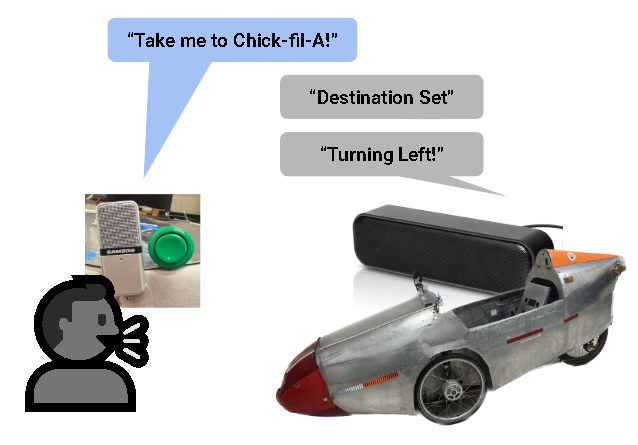
**

*Figure 2.2: Electrical diagram of the system that highlights the power distribution and all components in use.*

## User Interface.



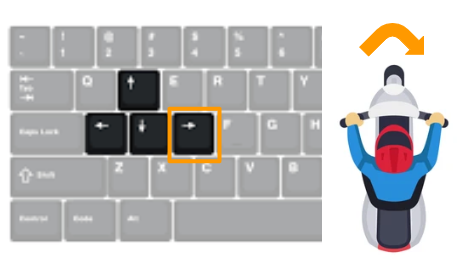
*Figure 2.3: Sound activating key fob to locate the bike.*



*Figure 2.4: The user will press an accessible button to activate microphone input to set destinations and the bike will respond with navigation updates.*







*Figure 2.5: Physical Control and Feedback. The user will operate the bike in one of the four modes described in detail later in the manual (left image). When braking, users will know to stop pedaling through vibrations in their seat (middle). In “Manual” mode, sighted users have the option to connect a keyboard to the Orin and steer with keyboard controls (right).*

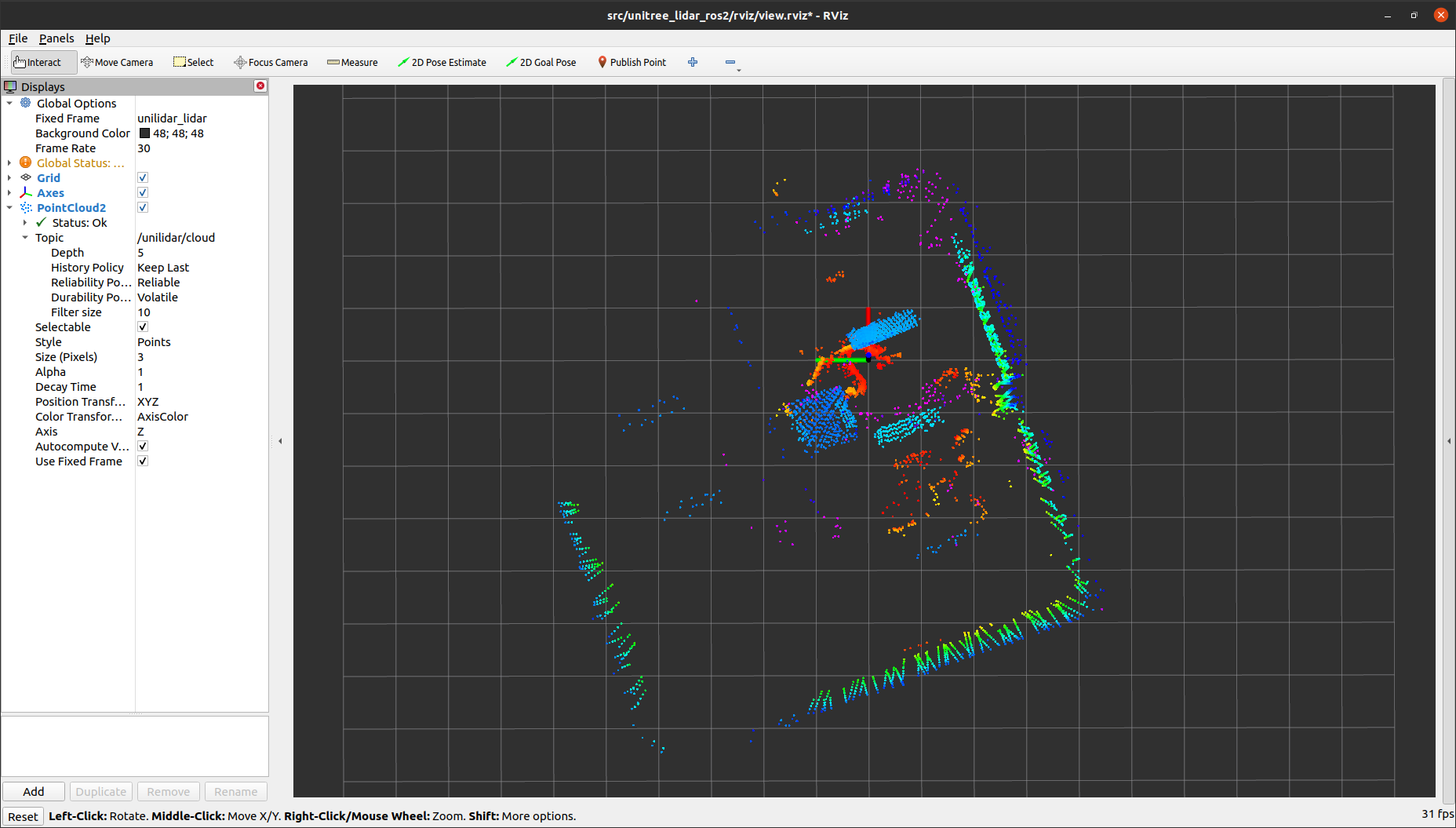
## Physical description.



*Figure 2.6: Side view of the trike. The OAK-D camera can be seen mounted on the front in the top middle of the bike.*

|  |  |
| --- | --- |

*Figure 2.7 and 2.8: Motorized steering (left), and braking mechanism on the left wheel connected to the drum bikes of the brakes (right).*

**

*Figure 2.9: Point cloud output of the LiDAR sensor in RViZ2*

## Installation, setup, and support

**Installation**

Each trike will come with the required software preinstalled and precalibrated. Ensure the sealed lead acid (SLA) batteries are charged and at the same state of charge by using a battery charger and battery balancer at 1A max charging rate. It is recommended that the user provides a map for the trike to use offline in autonomous mode. To do this, the user should use OpenStreetMap to download an offline version of the map onto a USB flash drive. The map file should be saved on the root level of the USB drive and named *map.osm.pbf.* This flash drive must be plugged into one of the USB ports on the Orin Nano before the trike is powered on. Upon booting, the trike's system will automatically detect the USB drive and load the map file for autonomous navigation. If a USB file is not loaded, but there is a valid network connection through ethernet or WiFi, a map of the local area will be downloaded.

**Setup**

1. Place the trike outside in a location with a valid GPS signal.
2. Ensure the trike is in park mode, dictated on the rotary switch as “(P)”.
3. Turn on the trike using the power switch on the left. Visually impaired users may need guidance or learned familiarity with locating the switch. Allow approximately one minute for the system to initialize. You will hear a voice notification when the trike is ready.
4. Rotate the mode selector switch and listen to the voice prompts until your desired mode is announced.You are now ready to begin pedaling.

**Support**

The trike is meant to be easily accessible in all situations for the visually impaired. For this reason, we opted to incorporate voice and haptic feedback into as many parts of the system as possible. To help find the bike, the user should activate the key fob which will play 90 dB sound from the bike. Once the bike is turned on, the user will be notified if a GPS signal is not found. In this case, the user should move to an area away from tall buildings or trees.

# Operation of the Project

## Operating Mode 1: Normal Operation

*3.1.1 Locating the trike:*

The user presses the button on the key fob to emit a distinct beeping noise. A visually impaired user will locate the trike’s position audibly.

*3.1.2 Preparing for use:*

The user should enter the trike by first stepping over one of the walls to the cockpit and then sliding their legs into position on top of the pedals. Flip the “On” switch. Wait for the voice confirmation that the system has initialized (see “Setup”, Section 2.4).

*3.1.3 Usage of modes:*

The mode selector offers four settings.

* *Park:* The brakes will engage until a different mode is selected.
* *Neutral:* All drive and brake systems disengage. Ideal for moving the trike from outside without powering it off.
* *Manual:* The steering and braking are controlled using an attached keyboard controller. There is no autonomous intervention for pedaling.

Notes: Emergency safety features remain enabled all times; however, the user must stay vigilant as when riding a conventional bicycle

* + W → brakes
  + S → disengage
  + J → steer left
  + L → steer right
  + K → recenter steering
* *Autonomous:* Activating this mode will engage both the autonomous navigation software suite and the emergency braking system.

1. Voice feedback will prompt the user for a destination.
2. After the user speaks their answer, the system confirms via voice feedback.
3. Pedal freely; users should keep their hands on the handlebars with manual-brake levers.
4. Use the manual brake for speed control or unexpected obstacles.
5. Haptic feedback (seat vibration) signals emergency braking. A user should be ready to stop pedaling immediately.
6. On arrival, the system will prompt through voice feedback to switch to “Park”.

In “Manual” and “Autonomous” modes, turn-signals flash automatically when steering. The brake lights illuminate only when the mechanized brakes engage (not for manual braking).

*3.1.4: Turning off the trike:*

The user should select the “Park” mode, flip the switch to Off, and exit the vehicle.

## Operating Mode 2: Abnormal Operations

If the trike is stuck in “Autonomous” mode–e.g., emergency brakes engage continuously because the trike is too close to an obstacle–switch to Neutral (recommended) or Manual. Neutral is preferred since the pedal gearing prevents backward motion and allows the user to push the trike free.

## Safety Issues

This semi-autonomous trike shares roads with vehicles and pedestrians. Safety is our top priority because the rider’s life is at stake. Given the system’s autonomous, outdoor operation, we identified five principal hazards: collision with obstacles, sensor limitations, electrical conduction through the aluminum frame, and system failures resulting in loss of control. We addressed these risks through redundant braking systems, combined braking and turn-signal lights, emergency-stop functionality, and insulated electronic enclosures. However, safe operation still requires user awareness and caution.

The first hazard is collision with obstacles, such as vehicles, pedestrians, and static objects. Our computer vision system predicts and avoids collisions, but the user must regulate their speed manually, especially when going downhill, where the trike can accelerate quickly. High speeds can exceed the response limits of the collision detection and braking system. Manual braking is available in both “Manual” and “Autonomous” modes and must be used proactively when necessary. The second hazard is sensor limitations. Our cameras perform poorly in low-light conditions, poor weather conditions, and may be affected by motion blur at high speeds. For this reason, users must never operate the trike after dark, even under street lighting, and must monitor their speed at all times. The third hazard is electrical conduction through the aluminum frame. Although all electronic systems are housed in insulated enclosures to minimize this risk, users should avoid operating the trike in rain or snow, both to prevent electrical hazards and to maintain computer vision accuracy. This also ties into environmental exposure. Water ingress into electronic enclosures or prolonged exposure to extreme temperatures could lead to unpredictable behavior. Users should avoid using the trike in wet conditions and store it indoors when not in use. The fourth hazard is system failure resulting in a loss of autonomous control. In the event of unexpected behavior, such as loss of navigation or braking automation, the user must immediately revert to manual control by using the mechanical brakes and stopping the trike. By following these procedures and understanding the system’s limitations, users will ensure both a safe ride and reliable system performance.

# Technical Background

# ***Design***

A velomobile is a trike that provides aerodynamic and protection advantages over a typical human-powered vehicle. This design provides greater stability than a bicycle, complemented by an aluminum frame for protection in minor accidents. It is also a recumbent trike, eliminating the need for balancing and mounted handlebars. Moreover, the velomobile has high weight tolerance and ample room for mounting components compared to a regular tricycle, making it an advantageous choice for this application.

# *Hardware*

This platform uses a Jetson Orin Nano Super to fulfill the computational needs of the hardware. To bridge connections between electromechanical components and send/receive data, multiple Arduinos are connected with different functions:

* Braking Arduino: This Arduino is connected to a servo driver powered with 6V from the battery that activates the servos connected to the left brake and right brake. This arduino is also connected to a 12V MOSFET board that activates the corresponding brake lights under the Arduino’s control, as well as the haptic motors that send feedback to the user when to brake.
* Key Fob Arduino: This Arduino is independent of the Orin and provides the function of receiving signals from the fob that the user can use to locate the bike when pressing the button on the fob. This arduino is always powered to allow users to discover the bike when not in operation (main power is off).
* Steering Arduino: This Arduino is connected to the NFP-GFP-555-EN (steering motor) and the motor controller and directs all steering from the Orin’s inputs. The motor controller acts as a gate for the 24V powered from the battery to our DC brushless motor, and while it can modify speed with PWM, the controller runs at full speed at all times to account for the slow speed of the motor. It also controls turn signal lights on the sides of the trike.

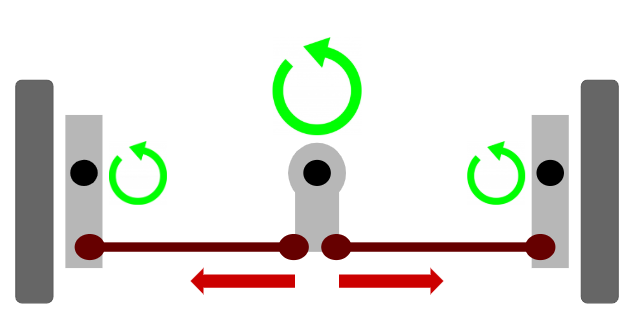
While the Arduinos and Orin are used for computation and signal flow, other components are implemented for accessibility and data collection from the environment. On the trike is a VectorNav VN-200 with a GNSS and IMU, OAK-D Stereo camera for depth images, and Unitree L1 LiDAR for SLAM. The VN-200 INS system was used over a typical IMU to estimate a more accurate position and velocity by using the GNSS. The OAK-D allows us to measure the depth of the object in the image, which is useful for our emergency brake feature. Finally, LiDAR lets us create a 3D map of the environment which is sent to our navigation server to create a path. There is also a microphone and speaker equipped for additional accessibility features.

* 1. ***Mechanical***

The original type of brake used on our platform is drum brakes: brake pads that push against the inside of the wheel’s hub when the brakes are activated. In order to offer users the safety of brakes independent of power and software logic, the braking system does not interfere with how the mechanical brake line operates but instead provides an alternative method to activate them by rotating the same lever the brake line pulls on when constricted.

The trike’s braking system uses high-torque servo motors to apply braking force. Each servo is mounted adjacent to the wheel so as to rotate in alignment when turning. Durable nylon Micro Cord is connected from the servo horn to the end of each lever. When a braking command is issued, the servo rotates, increasing tension in the Micro Cord and expanding the brake pad against the inner hub. The operation relies on basic principles of mechanical advantage (pulley effect) and tension forces transmitted through the cord.

The original steering method of our trike is an indirect steering mechanism where the driver would pivot the handlebar around a point in the center of the cycle (see Figure 2.7 for location). Any rotation around this point would shift two bars laterally which in turn pull on bars that rotate the wheels.



*Figure 4.1: Bird’s eye view representation of steering on the platform. The black dots represent pivot points and the dark gray rectangles represent the wheels.*

The platform’s steering control is achieved using a direct-drive configuration. A motor is mounted directly onto the steering joint of the trike. The motor applies torque directly to rotate the front wheels according to the steering angle output from the navigation system. This approach follows principles of rotational mechanics where the applied motor torque produces immediate angular displacement. While our direct-drive system can be vulnerable to high stall currents, we used the motor’s built-in encoder to continuously monitor steering angle and minimize unnecessary torque application. In all tests, power usage remained within safe limits, and the system showed no signs of overheating or mechanical failure.

To provide non-visual feedback to the rider, the trike integrates haptic motors for alert signaling. The system uses 3.3V 12,000RPM ERM (Eccentric Rotating Mass) coin motors, which generate vibration through the rotational imbalance of an internal mass. The physics behind ERM motors allows for simple, reliable vibratory cues to be produced when route changes, obstacle warnings, or system errors occur. Haptic alerts were placed on the tricycle seat, ensuring the rider could quickly and intuitively respond to system prompts.

* 1. ***Electrical***

Our trike is fully powered by a 24V Sealed Lead Acid Battery system consisting of 2 12V 12Ah batteries connected in series. According to the C-rating of our batteries, a constant discharge of current from 6.96 - 7.52A at 25° C will take around an hour to discharge, leading to a design that prioritizes efficient use of our components that draw the most power:

* Jetson Orin: 5A maximum @ 19V
* Steering Motor: 21A maximum @ 24V
* Braking Servo Motors: 3.5A maximum each @ 6V

According to the battery’s data sheet, if all three of the above components ran at these currents under the same operating conditions, the battery would deplete within 5 minutes. This is because the maximum operating current of the two motors come from the stall current. Stall current occurs when the motors attempt to turn but cannot move due to outside forces. To deter our motors from ever reaching this stall current, we employ mechanical and software solutions. When it comes to our servo motors, the force applied will always be to a fixed location. This means that the torque produced will be the same every time as long as the resting and activated position remains the same and the load on the spinning wheels is not impeding on the servo turning. Our steering motor is at a higher risk of reaching stall current due to our direct drive steering mechanism. When sending the signal to turn the bike, there are many scenarios where the wheels could be physically locked, or the steering could be in a maximum position (ie. trying to turn left when the steering is already turned fully left). To curtail this we use the built in positional encoder that reads if the motor is in a predefined range. If it is not in this range, then the motor will receive the signal to immediately stop. If the wheels are locked while the motor is within the safe range, we have fuses in place as a backup to prevent stall current.

In order to keep current flow safe and reduce risk of damaging our battery, we utilize fuses at multiple points in the system:

* A 10A fuse before the 24V input to the motor controller
* A 10A fuse directly after the battery’s positive terminal
* A 5A fuse before the positive terminal input to the Orin

Current is also monitored and controlled by 3 of our DC-DC step down converters. Each of our step down converters consist of capacitors at the inputs. In order to reinforce safety, a precharge circuit is included to protect against inrush current by slowing letting the capacitors charge through a resistive load, before our main power switch is flipped.

* 1. ***Software***

The trike is built using ROS2 Humble using Python and C++. Our codebase consists of four ROS2 packages: vectornav, unilidar\_sdk, depthai-ros, and trike. Only the code in trike was created by us, with the rest provided by the companies that created the sensor. On startup, the Orin will run all of the necessary packages by using a cron file.

*Emergency Brake*

The emergency braking system takes input from the OAK-D stereo depth camera’s estimated depth image. This depth image is produced by a proprietary machine learning model onboard the camera, and is passed to the emergency braking Python script via ROS2. The script first aggressively scrubs noise from this data. From here, the script can calculate whether there is any object in front of the nose of the trike within a threshold of two meters, and immediately activate the brakes if so. This process occurs many times a second, for accurate, immediate braking.

*Arduino*

Using the Boost.asio library, the Orin Nano is able to open a serial port connected to the Arduino. It does this by using the serial number of the Arduino to ensure a correct connection. The controller node subscribes to the topics */control/brake* and */control/steer* and writes to the serial port with the data received. Using two separate Arduino’s allow for a quick connection and for control even if the Orin were to stall during computation. The Arduino receives the data and uses GPIO pins to write to the motor controllers and lights.

*Manual Mode*

Manual mode is achieved by connecting to the keyboard using the serial ID of the keyboard and using the evdev Python library. This allows the system to read the keyboard inputs without opening a terminal. When the corresponding input is used, the node will send a message to the ROS2 control topic.

*Autonomous Mode*

At a lower level, autonomous mode uses lane following using the OAK-D camera. It will segment the image, look for two white lines, create a midpoint, then try to stay along that midpoint by lining up a vertical line on the camera and the generated midpoint. A PID controller is used to accurately follow the generated line.

For higher level navigation, we use the Navigation2 stack from ROS2. The system works by receiving an offline map from the user. Nav2 uses this as a global costmap. We then implement SLAM with the LiDAR in combination with the GPS system to create a local costmap. A route will be selected on the global costmap which the local costmap will follow. This will generate a set of waypoints for our system to follow. The INS will also be used to monitor the change in position with a PID controller to accurately follow the waypoints.

The final layer of low-level steering control is the collision prediction and avoidance system. Through ROS2, this script takes RGB and estimated depth images from the OAK-D camera, as well as the trike’s relative position from the VectorNav IMU, and the desired steering angle from the higher-level autonomous software. The script first runs the RGB image through FastSAM, an image segmentation model by Ultralytics. This identifies around 80 objects surrounding the vehicle, and has the ability to track some of these objects through time in the successive frames passed into the model. FastSAM is run in Python as it makes heavy use of PyTorch, while the rest of the script is written in C++ for the necessary realtime performance. The script does heavy image processing to narrow down the 80-some objects, of which the majority are windows on surrounding buildings or road markings, to only the vehicles, pedestrians, and obstacles the trike will need to avoid in the near future. From here, the script identifies the 3D location of these objects and compares them to their previously measured positions, estimating their motion relative to the trike. Using this information, the script can keep the trike clear of stationary objects, and if necessary, swerve out of the way of oncoming objects.

# Relevant Engineering Standards

In designing the trike, we referenced and complied with a suite of industry and governmental standards across electrical mechanical, software, communications, and regulatory domains. These standards ensure safety, interoperability, and accessibility for our visually impaired users.

## 5.1 Electrical Design and Construction

***5.1.1 National Electrical Safety Code (NESC)***

We followed NESC to ensure our trike’s electrical subsystem safety to guarantee user safety and system reliability.

* Conductor sizing & ampacity

Following NESC Table 210.2, we selected wire gauges to handle the highest continuous currents with a 25% safety margin. For example, the 24V DC main bus feeding the steering motor (21A max) uses AWG 10 conductors, while lower-power sensor and control lines (<= 7 A) use AWG 16. This prevents conductor overheating and voltage drop under peak load.

* Overcurrent protection placement

Inline fuses are installed immediately downstream of the battery’s positive terminal and at each DC-DC converter input, per NESC Section 4 “Overcurrent Protection”. This ensures that any short or overload is isolated at the closest point, limiting fault energy and protecting both wiring and devices.

***5.1.2 Underwriters Laboratories (UL) Standards***

UL 508 specifies safety requirements for industrial control devices – such as motor controller, and power distribution assemblies – designing criteria for component ratings, electrical clearances, enclosure construction, and fault-testing to prevent fire and electric shock. In our trike, the precharge circuit inline fuses, and power distribution panels are built and tested to UL508 guidelines to ensure that components are safely isolated and interrupted fault currents without damage or hazard.

## 5.2 Mechanical and Structural Standards

Although our trike is a three-wheeled, recumbent vehicle, we applied ISO 8098 (Cycles – Safety Requirements for Bicycles) bicycle-safety tests toi its chassis, steering, and brakes.

Specifically, ISO 8098 specifies static and dynamic load tests on the frame and forks to verify that welds and tubing resist permanent deformation. We ran identical load fixtures on our aluminum chassis and confirmed less than 10mm permanent deflection under the load. ISO 8098 outlines two-wheel bicycle braking-distance tests (from 15 km/h to stop within <= 3.5m), and we adapted these to our three-wheel configuration. Our servo-actuated brake system consistently stopped the trike from 15 km/h in under 3.2m on dry pavement.

## 5.3 Software Design and Coding Standards

We follow ROS2 Humble standard practices to build a robust, maintainable trike control system where each node in our packages cleanly separates publisher and subscriber responsibilities. For example, obstacle detection code never directly handles brake commands but instead publishes detection results to be consumed by a dedicated control node. We also organize all configuration parameters in structured YAML files loaded via standardized Python launch scripts. By grouping related nodes, it allows easy overriding of defaults and supports dynamic reconfiguration without code changes, making our autonomous-trike startup both reproducible and flexible in the field.

## 5.4 Regulatory and accessibility Requirements

We designed the trike to meet all relevant regulatory and accessibility requirements. Our user interface, including voice prompts, haptic feedback and a 90 dB RF-activated locator buzzer, aligns with Americans with Disabilities Act (ADA) guidelines so that visually impaired riders can operate the system independently.

## 5.5 Street-Legal Status

Our semi-autonomous tricycle has been engineered to meet U.S. regulations for low-speed electric bicycles, so that it can be ridden on public roads and bicycle paths wherever conventional bicycles are allowed. The trike is equipped with a front white LED and a rear red LED tail light where both of which remain fully operational at speeds as low as 5 mph. It also features side reflectors plus a rear reflector that meet the requirements of 49 CFR 571.108.

# Cost Breakdown

| Project Costs for Production of Beta Version (Next Unit after Prototype) | | | | |
| --- | --- | --- | --- | --- |
| Item | Quantity | Description | Unit Cost | Extended Cost |
| 1 | 1 | Stock Aluminum | 19.96 | 31.33 |
| 2 | 3 | Haptic Motors | 0.49 | 1.47 |
| 3 | 1 | Solid Core 20 AWG | 16.99 | 16.99 |
| 4 | 1 | Wagos 28pc | 20.95 | 25.25 |
| 5 | 1 | M3 Lock Nuts | 5.99 | 6.36 |
| 6 | 5 | DC-DC Step Down | 10.49 | 52.45 |
| 7 | 1 | Barrel Jack | 9.99 | 12.8 |
| 8 | 1 | Speaker | 15.98 | 15.98 |
| 9 | 1 | Motor Controller | 29.54 | 29.54 |
| 10 | 1 | Unitree L1 Lidar | 249.99 | 249.99 |
| 11 | 1 | Adhesive Cable Wire Clips | 6.99 | 6.99 |
| 12 | 1 | Double-Sided Tape | 17.99 | 17.99 |
| 13 | 1 | Wago Inline | 25.9 | 25.9 |
| 14 | 1 | Wire Crimp Kit | 10.06 | 10.06 |
| 15 | 1 | Inline Fuses | 12.49 | 12.49 |
| 16 | 1 | 4-Channel MOSFET PWM | 8.49 | 8.49 |
| 17 | 2 | Arduino Screw Terminal Hat | 18.99 | 37.98 |
| 18 | 1 | Screw Terminal Block Connectors | 14.99 | 14.99 |
| 19 | 1 | Battery Charger | 54.99 | 54.99 |
| 20 | 2 | 12 V Lead Acid Battery | 44.99 | 89.98 |
| 21 | 2 | Arduino Uno R3 Case | 5 | 10 |
| 22 | 1 | 16 AWG 120ft Wire | 26.39 | 26.39 |
| 23 | 1 | 18 AWG 13.2ft Wire | 15.59 | 15.59 |
| 24 | 1 | NFP-36GP-555-EN Gear Motor | 45 | 45 |
| 25 | 1 | NVIDIA Orin Nano | 249 | 249 |
| 26 | 1 | Truck Trailier Pigtail Connectors | 9.99 | 9.99 |
| 27 | 2 | DS5160 60 kg-cm Servo Motor | 30 | 60 |
| 28 | 1 | Keyfob | 10 | 10 |
| 29 | 1 | Adafruit PCA9685 Servo Motor Driver | 7 | 7 |
| 30 | 1 | Velomobile | 2000 | 2000 |
| 31 | 1 | Oak-D | 249 | 249 |
| 32 | 1 | Vectornav VN-200 | 3500 | 3500 |
| Beta Version-Total Cost | | | | 6903.99 |

The total prototype cost is $6903.99. We calculated this using regular retail prices and shipping costs for each part. The budget includes key components, with several donated items significantly reducing costs. The client donated items that he would like us to implement the semi-autonomous system with (Velomobil, VN-200 IMU, Oak-D Camera, Orin Nano), yet we still include their market prices. He picked a trike with a shell around it for both mounting space and to maximize his safety while riding. The Orin nano was donated because it offers high processing power for real-time sensor data processing and machine learning. A computer with high processing power is key since the trike must detect objects in real-time for user safety. The VN-200 Rugged IMU was donated because it provides integrated GPS and inertia data, simplifying the system. This integration minimizes latency by directly providing precise localization and motion data without additional processing data. This reduces processing complexity and potential points of failure, thus also maximizing user safety. The donated Oak-D cameras ensure comprehensive environmental perception from multiple angles. Major costs include the motor controller driver, battery, and servo motor for braking, with the total estimated budget at $6903.99. However, subtracting the cost of donated items, the budget comes down to $905.99. Additionally, because an insulated circuit was a high priority, this is reflected in our cost and prepares our design closer to a finalized product as opposed to the alpha version. There are no comparable autonomous cycles that offer the same combination of real-time computing power, advanced computer vision, and flexible, human-powered drive system. The beta version reflects the unique capabilities of the project and the cost of achieving safe, reliable semi-autonomous operation within the project timeline.

# Appendices

## Appendix A - Specifications

| **Requirement** | **Value, range, tolerance, units** |
| --- | --- |
| **Mechanical** |  |
| Electronic Steering Speed | Full range of motion (±45° under 2 second) |
| Braking System | * Full spectrum of braking intensities * “Smooth” braking - no wheel skipping, stalling, jerking etc. * Activates within 0.2 seconds when obstacle is detected |
| Manual Braking in Autonomous Mode | Fully function |
| Total Component Weight | 130kg (Components and Rider) |
| **Autonomous Navigation** |  |
| Path Following | Less than .5 meter deviation from intended path |
| Obstacle and Pedestrian Detection | Automatic stopping and/or waypoint aversion |
| Autonomous Operation | Autonomous steering and braking, user input limited to location |
| Environmental Condition Adaptability | Clear guidelines on time-of-day and weather robustness   * Daytime Operation Only * Clear Weather Conditions * No Wet or Slippery Surfaces |
| **Device Element Function** |  |
| Depth Camera (OAK-D) | Depth accurate 0.7 - 12 meters: 75mm precision |
| **IMU**(VN-200) | Less than 0.04 mg Acceleration Bias |
| GNSS(VN-200) | Within 2.5 meter accuracy |
| Power(12,000mAh 24V) | 1.5 hours on Full Charge |
| **Safety** |  |
| Emergency Braking Override | Enabled to override neural network control in case of system failure |
| Robustness to Inference Failure | Automatic stopping if any piece of software fails |
| Haptic Feedback | Noticeable indication to user of braking and ride progress in active environment (>60dB, high wind, speed, bumpy terrain) |
| **Accessibility** |  |
| Entry accessibility | Easy entry without clearance issues |
| Tricycle Locator | Sound-emitting speaker activated by key fob within 5 meters |

## Appendix B – Team Information

Linden Adamson is a graduating Computer Engineer with a concentration in Machine Learning and a minor in Business Administration & Management.

Jason Calalang is a graduating Computer Engineering and Biomedical Engineering student.

Tara Gill is a graduating Electrical Engineering student with a concentration in Machine Learning, and is currently the secretary of our IEEE chapter.

Zhilang Gui is a graduating Computer Engineering student with a concentration in Technology Innovation.

Cole Resurreccion is a graduating Computer Engineer who will be working as a Navigation Software Engineer at Joby Aviation while pursuing a Master of Computer Science at Georgia Tech.

Yiran Yin is a graduating Computer Engineering student.

## References

[1] Lutz, C., & Schöttler, M. (2018). Autonomous Driving and the Demand for Mobility among Older Adults: The Case of Autonomous Vehicles in Germany. Transportation Research Record.

[2] M. Hurford, “Good News, Everyone! People Are Riding More than Ever,” Bicycling, Sep. 26, 2023.

## 