

# **Development of an Autonomous Platooning Campus Vehicle System**

A Technical Report submitted to the Department of Mechanical Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

**Patrick Dunnington**

Spring, 2022

Technical Project Team Members

Cameron Chiaramonte

Gilchrist Johnson

Nicholas Sofinski

Alex Wilson

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Tomonari Furukawa, Department of Mechanical Engineering

## Introduction

At the University of Virginia (UVA), there are many issues that hinder people’s ability to get around campus, or ‘grounds’, given the current methods of transportation. The UVA grounds consist of many steep hills and extensive distances between buildings. In addition, the current bus system, with sparse stops, still requires a significant walking distance to utilize. These inefficient methods are especially inaccessible to disabled or injured persons on the grounds. Thus, we propose a new mode of transportation as a solution: A system of platooning golf carts known as the Campus Vehicle System (CVS) (Figure 1).

Defined by the American Legislative Exchange Council (ALEC) as “a group of individual motor vehicles traveling in a unified manner at electronically coordinated speeds at following distances that are closer than would be reasonable and prudent without such coordination” (ALEC, 2018), platooning will allow the system to transport students to various destinations around the University of Virginia’s main campus while only utilizing one driver.



*Figure 1: Image of the completed Campus Vehicle System.  
In frame: Nicholas Sofinski (Gilchrist Johnson, 2023)*

As such, the purpose of this project is to modify two prototype Club Car golf carts for use within the CVS. To ensure proper development practices, *Product Design and Development* by Karl Ulrich, Steven Eppinger, and Maria C. Yang will be used as a reference throughout the design process of the system (Ulrich et al., 2020). Both carts will be made identical, with software settings determining which cart is the “leader” and “follower” of the platoon. As one of the provided golf carts had been previously modified, significant changes to the cart will be made to restore the cart to working order. A custom mechatronic system will be developed for both carts to enable autonomous capabilities within each cart. Then, ROS (Robotic Operating System) based software will be developed and implemented into both carts for them to be able to communicate and platoon effectively. Once the whole system is in working order, it will be tested thoroughly to ensure it is safe and reliable. The final goal is to successfully implement and demonstrate platooning with both carts on Engineer's Way, a section of the University of Virginia’s main campus.

This report will document the entire design and selection process that has been completed along with the progress that has been made throughout the 2022-2023 academic year. To properly understand the philosophy used within the design of the Campus Vehicle System, a review of the design philosophy will be included in this report (Ulrich et al., 2020). Additionally, a review of the previous work conducted on the project will be used to provide a timeline and give context to the project. The design process will then be discussed in detail, emphasizing the development process over the last two semesters. Finally, the report will present our final design for the Campus Vehicle System and give recommendations for future improvements and developments related to the project.

## Essential Knowledge

### Background & Previous Work

While the Campus Vehicle System (CVS) is currently being developed at the University of Virginia, this was not always the case. The project was first conceptualized at Virginia Tech by Dr. Tomonari Furukawa as a capstone project under the title Self-Driving Vehicle Team (SDVT) (Furukawa et al., 2018). This was the school's first senior design team aimed at autonomous driving. The goal of this project was to implement autonomy in a single Club Car golf cart through the Kairos Autonomi System (Furukawa et al., 2018). This system, sold as an all-in-one solution for implementing autonomy into industrial vehicles, was not as versatile as Dr. Furukawa had hoped. Thus, the focus of the project shifted to developing a custom system to replace the Kairos Autonomi. By 2019, the Kairos Autonomi was completely phased out, being replaced with a Robotic Operating System (ROS) based system. At this point, the development of the SDVT system would halt, as Dr. Furukawa moved from Virginia Tech (VT) to the University of Virginia (UVA). Here, the development of the project would continue under the new title, Campus Vehicle System (CVS) (Furukawa et al., 2021). Teams would continue the work started at VT, shifting focus to developing new methods to implement platooning into the system (Furukawa et al., 2021). This long development timeline has brought many different iterations of the CVS. While there has been a multitude of changes to the system, many of them are in relation to the braking and sensor methods used.

The first documented braking system was a linear actuator (Furukawa et al., 2018). While there isn't much documentation as to why the system failed, we speculate that electronic complications played a factor. As a result, the next team altered the braking system relay to include an open collector output and added a MOSFET for pulse width modulation (PWM)

(Furukawa et al., 2018). Even though the braking system worked, it could only be triggered to fully pull the brake down, having no option for gradual braking. To resolve this issue, the 2018 team developed a cable pulley system that pulled the brake pedal for gradual braking (Furukawa et al., 2018). Despite the high torque required, this system has worked well, and the method has been kept with subsequent teams. The final change to the braking system was changing the design to utilize a Clearpath Teknic MC Servo motor instead of the previously utilized DC brush motor (Teknic Inc., 2017). As this braking system has been proven to work well in past iterations, we will use it within our implementation of the CVS.

The sensor implementation into the CVS has a much more convoluted past than the braking system, with implementation methods utilizing various combinations of global positioning systems (GPS), inertial measurement units (IMU), ultrasonic sensors, infrared sensors (IR), LiDAR sensors, and cameras. In 2016, the SDVT used LiDAR for environment mapping with an IMU/GPS system to measure the cart's movement and position within that environment (Furukawa et al., 2018). The 2017 SDVT switched to cameras and ultrasonic sensors for object tracking and obstacle identification. As ultrasonic sensors have too much noise to accurately detect moving objects, the 2018 SDVT switched to utilizing LiDAR and an IR (Furukawa et al., 2018). Once again, reliability issues hindered the development of the system. When the project came to the University of Virginia, the system was once again redesigned, replacing the IR camera with an AI stereo camera to map the surrounding area and the cart's path to its objective (Furukawa et al., 2021). While the cart was able to map the surrounding area well with LiDAR, the combination of sensors proved ineffective with route mapping. Due to the wide range of sensor issues, our implementation of the CVS will be designed to only utilize the data from the speedometer built into the golf carts and feedback from the EPS.

## Design Philosophy

To aid the creation of the product, the method of product design outlined in *Product Design and Development* was used (Ulrich et al., 2020). This methodology is broken down into six steps: planning, concept development, system-level design, detail design, testing and refinement, and production ramp-up (Ulrich et al., 2020) (Figure 2).

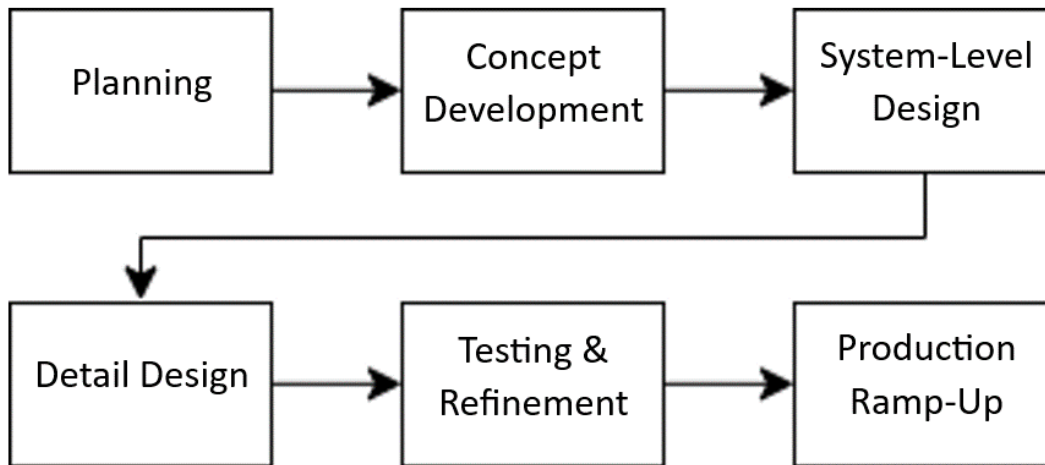


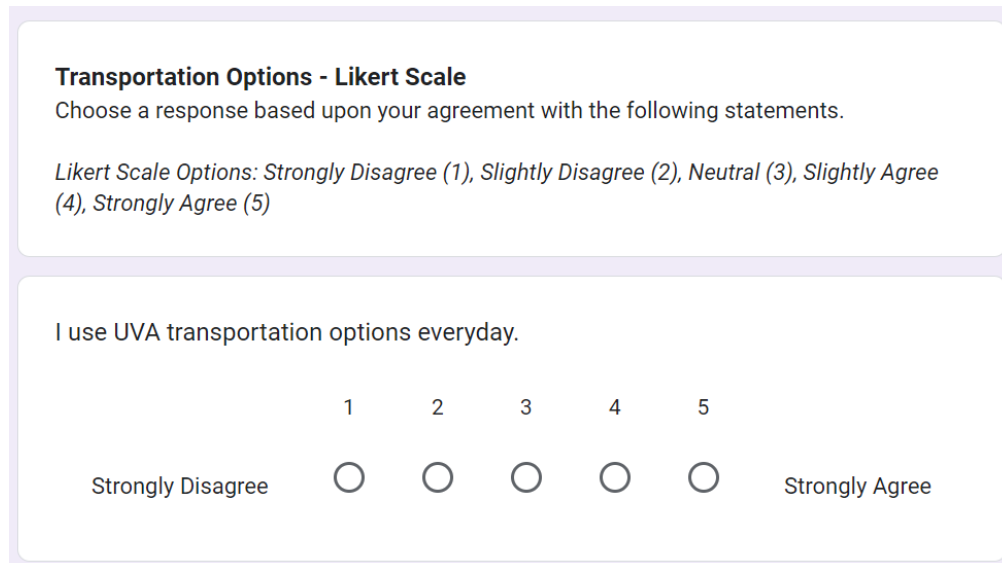
Figure 2: Product Design Flowchart, as defined in *Product Design and Development*. (Gilchrist Johnson, 2023)

The process starts with planning, where preliminary research is done, and constraints are set. Next is concept development, where the customer needs are formed and are iteratively refined into various concepts to be selected with respect to the functions required by the product. The product is then considered on the scale of the entire system, known as system-level design, with basic calculations being made to determine feasibility. Next, detail design identifies product specifications such as materials and geometry. The product goes into the testing and refinement phase, where prototypes are made and tested until a final design is chosen. The final stage, production ramp-up, marks the production of the final product. For this project, which is meant to introduce students to the concepts presented and not necessarily with a final design or multiple prototype iterations, concept development was the primary focus.

## Design Process

### Customer Needs & Technical Priorities

To design a product that meets the functions desired by the customer, a list known as the customer needs was created. The customer needs describe the product; when they are met, the product can compete with similar products on the market. This list is created by gathering information from customers and other patrons who will utilize the product. They identify their ideal product, and what they want it to be able to achieve. In terms of the campus vehicle, discussions occurred between the team and the capstone professor. To gather data from the customers, a survey was sent out to students at UVA asking about their current modes of transportation, and what improvements they wanted (Figure 3). A full reference of the questions on the form can be found in Appendix II.



**Transportation Options - Likert Scale**  
Choose a response based upon your agreement with the following statements.

*Likert Scale Options: Strongly Disagree (1), Slightly Disagree (2), Neutral (3), Slightly Agree (4), Strongly Agree (5)*

I use UVA transportation options everyday.

1      2      3      4      5

Strongly Disagree                        Strongly Agree

Figure 3: Screenshot of Customer Needs Form. (Gilchrist Johnson, 2023)

From the survey and the professor's statements, consequent needs were interpreted, and labeled as secondary or primary needs. These needs were then compiled into a finalized list (Table 1). By adhering to these needs, we can ensure that future progression on the cart is in line with an ideal end goal.

**Table 1**  
**List of Interpreted Customer Needs**

---

1	Platooning must be enabled for use as campus vehicles.
2	Inherit the best technologies developed/used in the past to focus on successful demonstration.
3	The integrated system should be simple to improve reliability.
4	Pedestrian and passenger safety is secured.
5	Any cart can be the leader or follower cart.

---

*Table 1: List of Interpreted Customer Needs. (Gilchrist Johnson, 2023)*

Since the customer needs are not quantitative measures, the target specifications are created. Looking at the needs and the product that is being created, the target specifications are chosen to represent important aspects of the cart. An ideal value is then created for each specification along with its functional unit. For example, the integrated system should be simple to improve reliability. This directly correlates to the customer's need to travel long distances and have a long battery life. To quantify this, a unit of miles traveled was chosen, with the ideal value being 35 miles. If this ideal value is met, then the product will be at its best performance. Since it is not possible to always meet the ideal value, a marginal range is also created. If the ideal value is not met, but is still within the marginal range, the system can still compete with other products. Using a quality function deployment spreadsheet, the technical priority of each specification was determined. First, each target specification was correlated to each customer need on a scale of 1-5, and each customer need was given an importance ranking from 1-5. From there, the sum of the product between each correlation and importance was taken, giving a value called technical importance. This value quantifies each specification and ranks them from most to least important.



After the importance was found, the difficulty of each specification was marked from 1-5. This difficulty is dependent on the team’s ability to complete each specification, which depends on factors such as whether we have the material necessary, or if we have the knowledge to complete it. This difficulty factor was then multiplied by the technical importance, which gave the technical priority. This priority calculated which technical specification would be the most difficult, and would therefore take the most time to complete, but would also be important to the design. For the campus vehicle, the highest priority is the advanced sensor suite (Table 2). This makes sense, as safety is a very big concern to the customer and to users, so a sturdy sensor suite is needed to ensure that platooning carts don’t cause accidents.

**Table 2**  
**List of Technical Priorities**

<i>Technical Priority</i>	<i>Target Specification</i>	<i>Ideal Value</i>	<i>Unit</i>
1	Advanced Sensor Suite	>90	%Surrounding Area Covered
2	Maximized Battery Life	35	Miles
3	High Reliability	0	Hours (Mean time between failure)
4	Platooning Signal Range	250	Meters
5	Control Response Time	0.1	Seconds
6	Cost Efficiency	15000	USD
7	Additional Platoons	4	# of Carts
8	Original Functionality	100	% Functionality Retained
9	Distance to complete stop	3	Meters

*Table 2: List of Technical Priorities. (Gilchrist Johnson, 2023)*

## Concept Generation & Selection

With the customer needs and target specifications in mind, the system that is being designed was then broken down into specific functions that need to occur. This is known as functional decomposition (Figure 4)

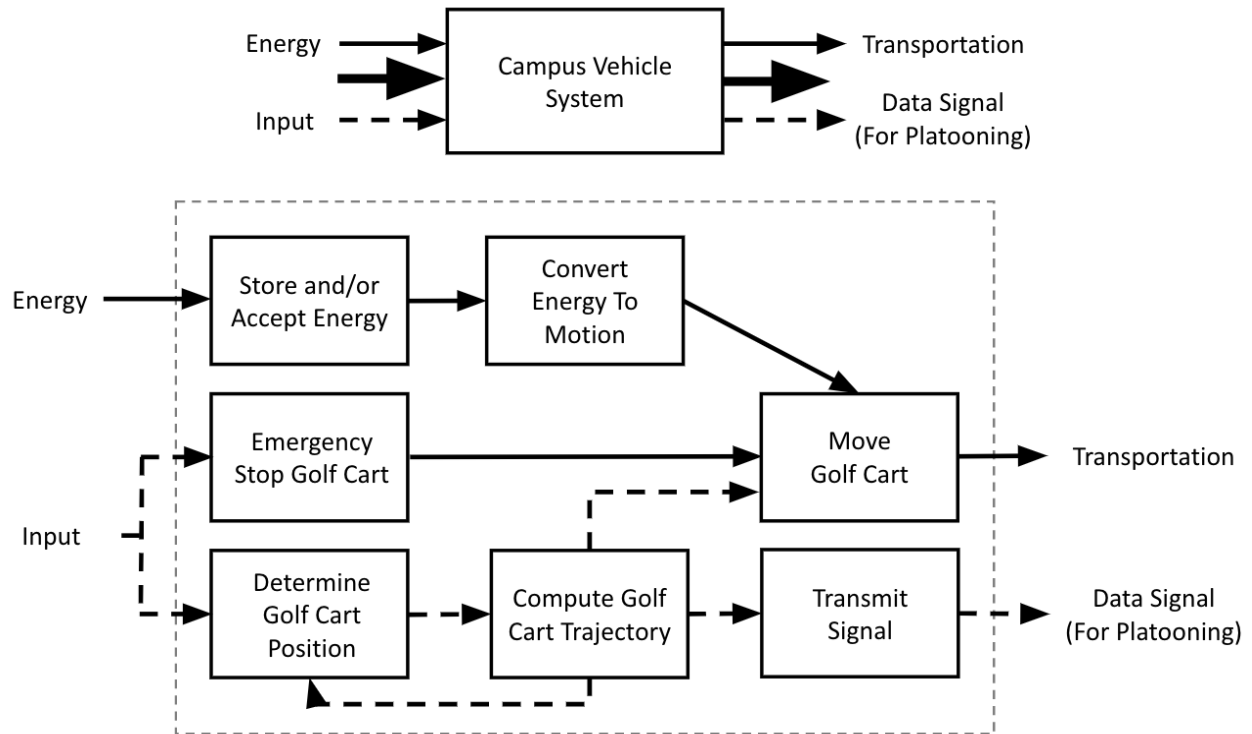


Figure 4: Functional Decomposition Diagram. (Gilchrist Johnson, 2023)

In the decomposition, the inputs to a single cart are given energy and some type of input. From here the energy and input travel through different functions that eventually transform them into transportation and data, respectively. Since either cart can be the leader or a follower, a general input is stated. For the leader cart, this input is from the user turning the wheel or stepping on the pedals, which then transmits the signal back to the follower cart. The follower cart then would have this signal as its input in the diagram. Given the diagram, a solution to each sub-function is required. To compare different solutions, a morphological chart is made (Table 3).

**Table 3**  
**Morphological Chart**

<i>Sub - Functions</i>	<i>Solutions</i>				
Store and Accept Energy	Battery				
Convert Energy to Motion	Motor	Actuator	Pulley System		
E-Stop the Golf Cart	Physical Button	Wireless Button	Controller	Voice	Physical Brake
Determine the Golf Cart Position	QR Code Reader	LIDAR	Odometry	Camera	Human Visual
Compute the Golf Cart's Trajectory	PC	Microcontroller	SOC	Human Calculation	
Send & Receive Data	5G WIFI router	Bluetooth	RF transmission	Camera	Binary LEDs
Move the Golf Cart	Motor				

*Table 3: Morphological Chart. (Gilchrist Johnson, 2023)*

This process is known as concept generation, as physical solutions are created that will help meet the target specifications and customer needs. From here, different solution sets were created to be used in concept selection. A solution set is a list of different solutions, where there is at least one solution for each sub-function. This set represents a possible system that could be for the campus vehicle.

Since the campus vehicle already has a system in place from previous years, concept selection was used to compare the current set to new sets that were created this semester. A screening matrix was filled out first (Table 4). This matrix follows a plus, minus, zero system. For each set, a given solution is ranked with a plus (better than the current solution for that sub-function), minus (worse than the current solution), or a zero (equivalent to the given solution).

**Table 4**  
**Concept Screening Matrix**

<i>Acceptation Criteria</i>	<i>Potential Solution</i>		
	<i>Solution 1</i>	<i>Solution 2</i>	<i>Solution 3</i>
Store and/or Accept Energy	+	+	+
Convert Energy to Motion	+	+	+
Emergency Stop	+	+	0
Golf Position Recognition	+	0	0
Compute Golf Cart Trajectory	+	+	+
Send/Receive Position Data Signal	+	-	0
Move Golf Cart	+	+	+
Sum +'s	6	4	3
Sum 0's	0	1	3
Sum -'s	0	1	0
Net Score	6	3	3
Rank	1	2	2
Continue?	Yes	Yes	Yes

*Table 4: Concept Screening Matrix. (Gilchrist Johnson, 2023)*

This matrix allows the user to weigh each sub-function in terms of its importance, and then rate a solution in each set from 1-5, with 5 being a good solution and 1 being a worse solution. The weight and rating are then multiplied together, giving a final score for each set. As an example, the carts knowing where they are in relation to each other and the surroundings were the most important (golf position recognition) in our design, so they were weighted as 25%

of the total selection criteria weight (Table 5). For each solution set generated from the morphological chart, the specific solution was then rated, as previously stated, from a 1-5, where a 5 means that solution effectively meets the criteria. For solution set 2, LiDAR was chosen for golf position recognition and was rated as a 5 because it is good at generating a map of the area surrounding the carts. The weight and rating are then multiplied together, giving a final score for each set. For the campus vehicle, the screening and scoring matrix reinforced the current solution set, meaning that the current system should still be used moving forward.

**Table 5**  
**Concept Scoring Matrix**

<i>Selection Criteria</i>	<i>Potential Solution</i>						
	<i>Solution 1</i>		<i>Solution 2</i>		<i>Solution 3</i>		
	<i>Weight</i>	<i>Rating</i>	<i>Weight</i>	<i>Rating</i>	<i>Weight</i>	<i>Rating</i>	<i>Weight</i>
Convert Energy to Motion	15%	5	0.75	5	0.75	5	0.75
Compute Golf Cart Trajectory	20%	5	1	5	1	4	0.8
Emergency Stop	5%	5	0.25	4	0.2	2	0.1
Golf Position Recognition	25%	5	1.25	5	1.25	5	1.25
Store & Accept Energy	5%	5	0.25	5	0.25	5	0.25
Send/Receive Position Data Signal	20%	5	1	2	0.4	3	0.6
Move Golf Cart	10%	5	0.5	5	0.5	5	0.5
<b>Total</b>	<b>100%</b>		<b>5</b>		<b>4.35</b>		<b>4.25</b>
<b>Rank</b>			<b>1</b>		<b>2</b>		<b>3</b>

*Table 5: Concept Scoring Matrix. (Gilchrist Johnson, 2023)*

## Final Design

The final design of the campus vehicle system integrates the existing club car components with a custom mechatronic system used to control the golf cart. The system can be split into 6 modules: Actuators, Club Car Components, DC-DC Converters, Electronic Control Unit (ECU), User Input, and Feedback (Figure 5).

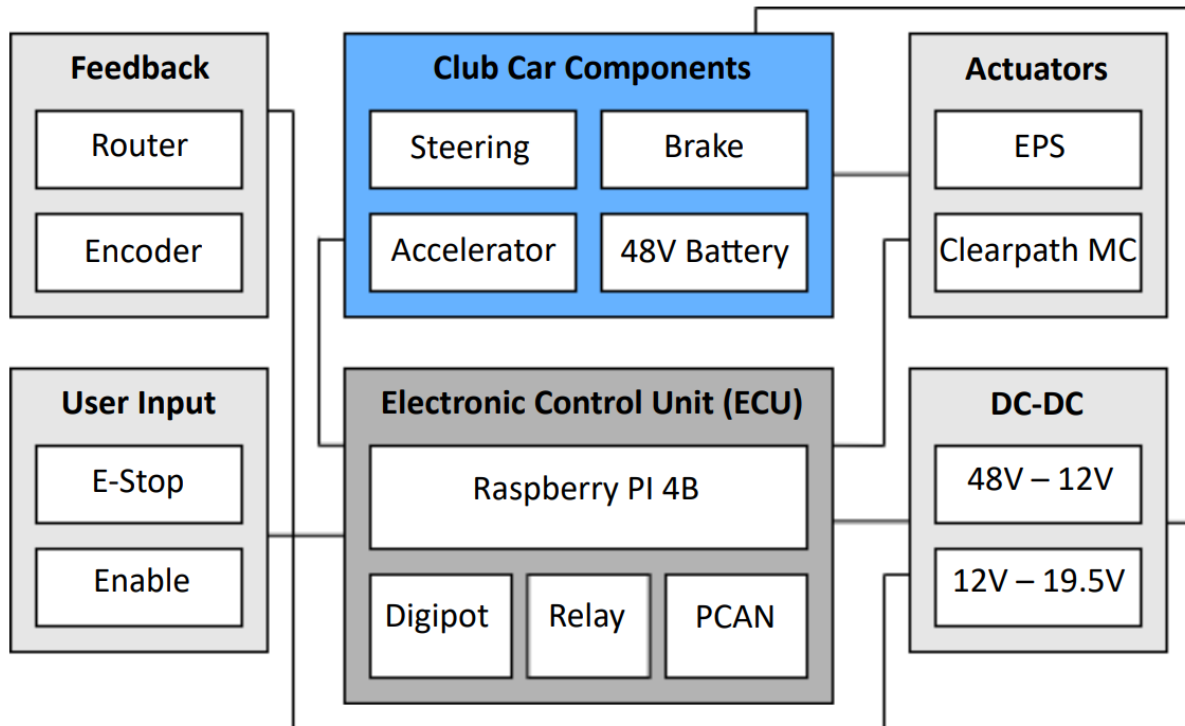


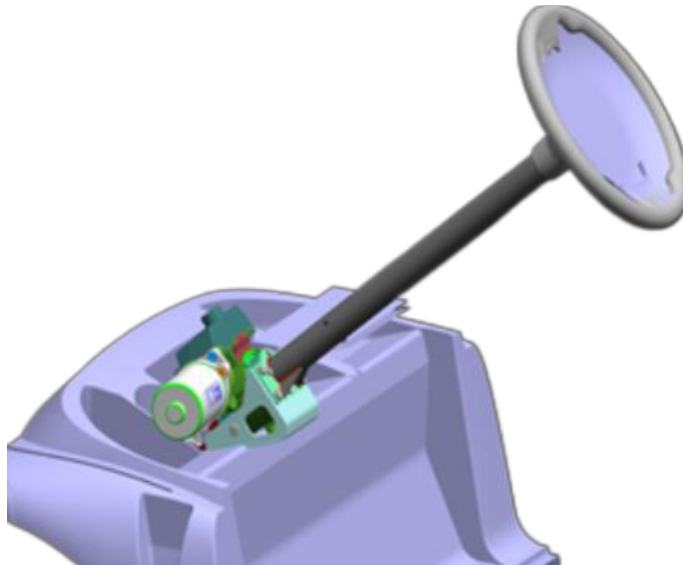
Figure 5: Overview of Final System Components. (Gilchrist Johnson, 2023)

The simpler modules can be summarized as follows. The Feedback module provides feedback to the system through a NETGEAR Nighthawk Wi-Fi router and speedometer attached to the gearbox. The User Input module allows users of the vehicle to enable the platooning feature of the vehicle as well as stop the vehicle with an emergency button. Finally, the DC-DC Converter module provides the necessary power conversions to run the ECU and actuators. For the more complex modules, refer to the following sections.

## Actuators

### *Electronic Power Steering*

The electronic steering for each Club Car is controlled by a Nexteer electric power steering system (EPS) (Figure 6). The EPS consists of an internal servo motor, an absolute encoder, and an internal computer that allows for communication to both subsequent parts.



*Figure 6: CAD Model of Nexsteer Electric Power Steering Module. (Credit: Furukawa et al., 2018)*

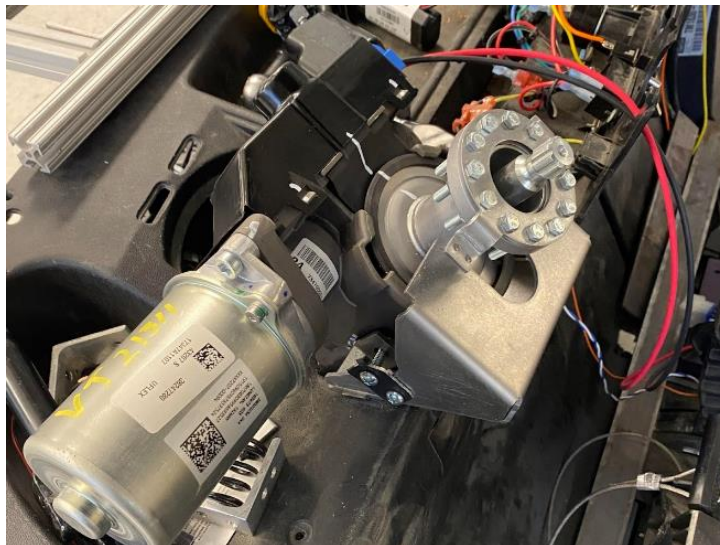
The EPS features CAN communication that requires a PCAN-USB converter to be used with the Raspberry Pi 4B's in the ECM. The PCAN-USB converter does not have internal resistors to comply with the standard CAN protocol. To account for this, 120  $\Omega$  of resistance is added between the CAN inputs on the Control Board. Additionally, the EPS requires 12V power from the DC-DC converters to power the internal components of the device. Finally, there is an ignition switch that also requires 12V of power to turn the EPS on, which is attached to a relay within the ECM. A full wiring diagram for this configuration can be found in Appendix XX.

Nexsteer did not provide any mounting hardware with the second EPS. Therefore, a custom mount was designed and sent to Protocase to be machined (Figure 7).



*Figure 7: Custom Nexsteer EPS Mount. (Cameron Chiamonte, 2023)*

The mount we received from Protocase had some minor fitment issues and was subsequently ground down using an angle grinder. Additionally, the mount was less stable than initially planned, so scrap metal pieces were cut and installed to stabilize the position of the mount (Figure 8). Finally, shaft couplings were attached to the gears on the sides of the EPS to connect the module to the drive and steering shafts.



*Figure 8: Custom Nexsteer EPS Mount (Credit: Cameron Chiamonte, 2023)*



## ***Braking***

The braking system uses a Clearpath MC Servo motor to control a pulley system attached to the brake. To attach the motor to the golf cart's frame, a custom mount was designed and sent to Protocase to be machined (Figure 9)



*Figure 7: Custom Clearpath MC Mount. (Patrick Dunnington, 2023)*

The design parameters for the frame were determined by the force it needed to withstand as the pulley was tightened and applied to the brake pedal. The shafts on which the pulley wheel sits were also removable to make replacing those wheels simple. Finally, the welds were boxed with thin plates at the bottom to maximize design strength. For the braking motor itself, a motor was chosen based on an estimated torque required to pull the brake pedal during operation. This had been determined by the motor on the previous cart, so the only task required was to install the motor onto the cart (Figure 10).



Figure 10: Installed Clearpath MC Brake Motor. (Patrick Dunnington, 2023)

Operation of the brake is conducted through the Raspberry Pi 4B's (Pi) general-purpose input/output pins (GPIO). First, the Clearpath MC motor was set to Move to Absolute Position (4) mode (Figure 11). In this mode, you can set the motor to respond to binary inputs from the Pi's GPIO pins. The 4 preset positions represent no brake, light brake, heavy brake, and full brake.

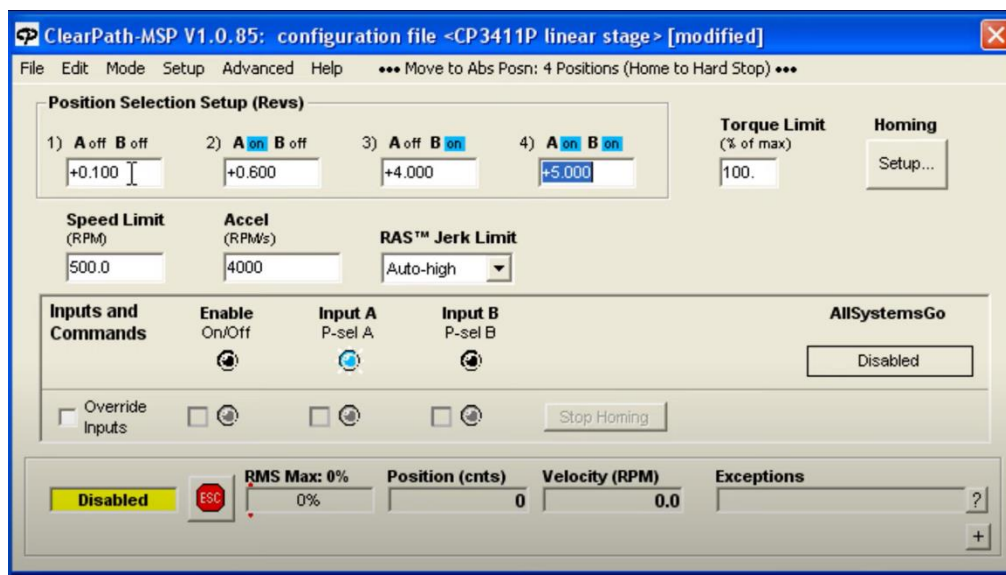


Figure 11: Screenshot of Clearpath MSC. (Credit: Teknic Inc., 2017)

## Electronic Control Module (ECM)

The electronic control module (ECM) serves as the main form of control within the CVS. Unlike previous mechatronic systems designed to be used with the cart, the entire ECM is designed to work off one power source (12V). This eliminates the need for additional external power converters and simplifies the wiring scheme (Figure 12).

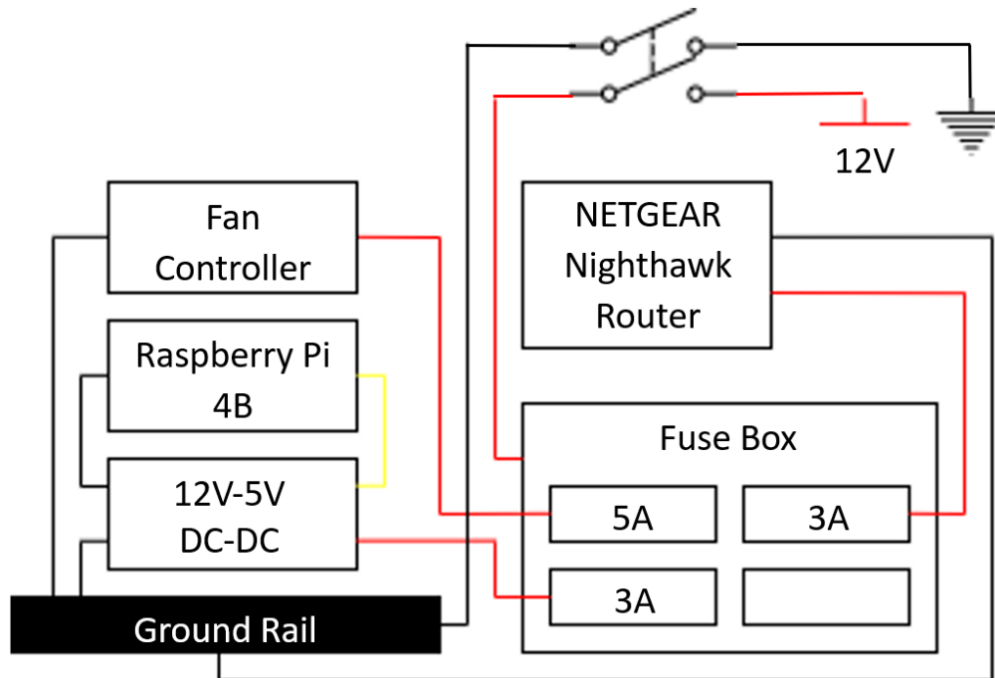


Figure 12: Power Diagram for the ECM. (Gilchrist Johnson, 2023)

The 12V input goes to a 4-way automotive fuse box. This provides protection to the internal components in case of a power surge. Attached to the fuse box are the fan controller, 12V-5V DC-DC converter (used to power the Raspberry Pi 4B), and NETGEAR Nighthawk Router. Finally, the entire system is connected to a double pole single throw (DPST) switch. By utilizing a DPST switch instead of a single pole single throw (SPST) switch, we reduce the risk of power surges. Additionally, it allows the power circuit to disconnect from the rest of the ECM, making troubleshooting easier.

All the inputs and outputs coming from the data pins go into a centralized control board. The control board contains all the connections as well as 3 relays and a digital potentiometer. Two of these relays (C93416) are used to switch between manual acceleration (NO) and autonomous acceleration (Figure 13).

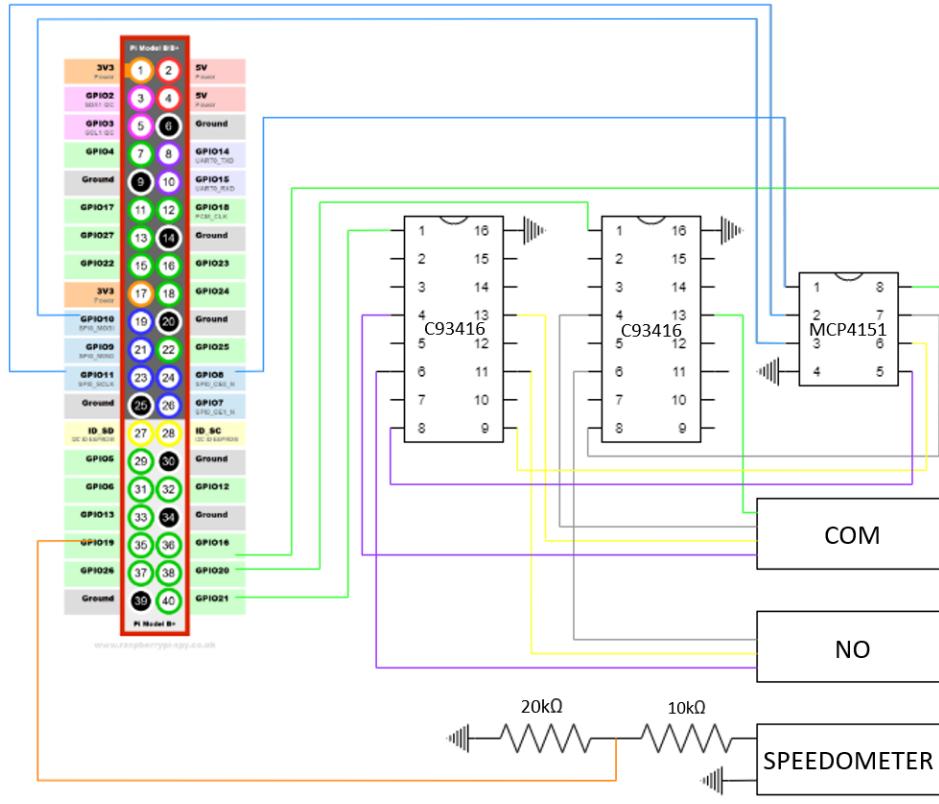


Figure 13: Wiring Diagram of Control Board. Direct IN/OUT Exempted. (Gilchrist Johnson, 2023)

When in autonomous mode, the acceleration is controlled through the MCP 4151 digital potentiometer (Digipot). Additionally, the speedometer input is connected to a voltage divider to step down the 5V high to a 3.3v signal readable by the Raspberry Pi 4B (Pi). Other inputs and outputs (brake, e-stop, enable) are directly fed by the Pi's general-purpose input/output (GPIO) pins. The third relay in the ECU controls the power going to the EPS module. This toggles the ignition on/off and allows the user to unlock the steering wheel when they drive manually. The control board itself has 12 connectors (Figure 14).

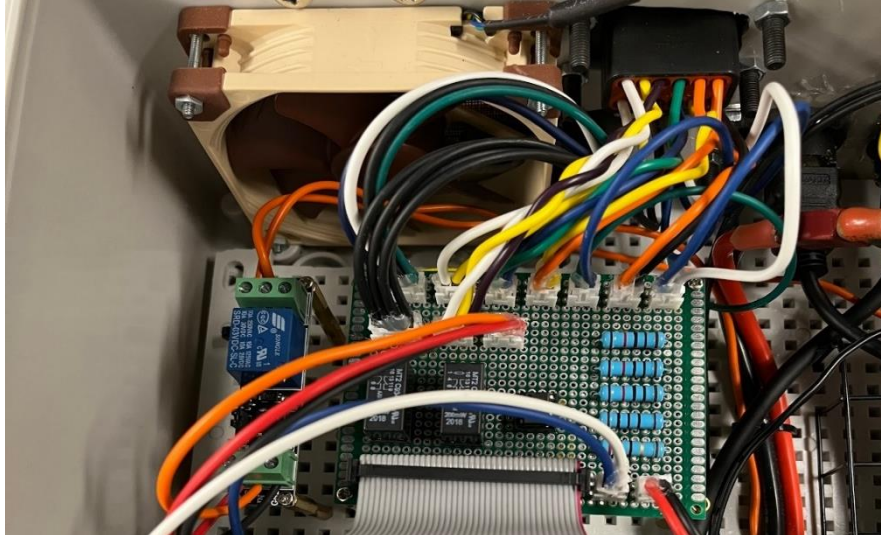


Figure 14: Plugged-In Control Board. (Patrick Dunnington, 2023)

Each of these connections (Brake GPIO, Brake GND, Acceleration COM, Acceleration NO, Accel-Enable, Ignition Relay, E-Stop, Enable, Encoder, CAN-COM, PCAN, and 5V Out) is then appropriately routed underneath the board to its corresponding GPIO pin using a ribbon connector between the board and the Pi (Figure 15). The ECU also contains a Peak PCAN-USB which allows the PI to communicate with the Nexteer EPS module. Finally, both ECUs are identical which means that the boxes are interchangeable between the 2 golf carts.

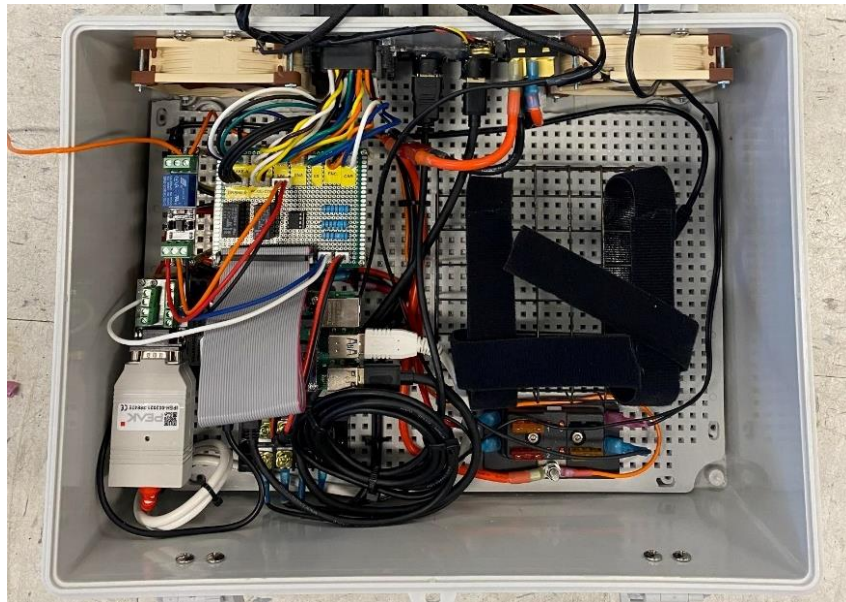


Figure 15: Picture of Final ECM. (Patrick Dunnington, 2023)



## ROS Packages

### *Base*

The Base package is used to operate the ECM. It contains three major folders: config, launch, and src. A full archive of how to use the package and the content found in each folder is contained within Appendix I. The config folder contains the yaml file “parameters.yaml”. This file contains various user-defined parameters to configure the rest of the package. Note that all parameters must be set for the package to run, as there are no preset parameters in the code. The launch folder contains two launch files, “base.launch” and “controller.launch”. The file, “base.launch” launches the entire system and imports the parameters from the “parameters.yaml” file. If “enable\_joy” is true in the “parameters.yaml” file, the joystick node will launch with the rest of the file. The “controller.lanch” file only launches the nodes required for joystick inputs and is mainly used for debugging. Finally, the src file contains the Python 3 code that operates ECM, such as “acceleration.py”, and “steering.py”. Unless otherwise stated each file was written solely by the 2022 Campus Vehicle Team. A brief explanation of what each file does is listed below:

- acceleration.py: Controls the acceleration through setting the digital potentiometer and relays on the control board.
- braking.py: Controls the braking through 4 GPIO pins setting input logic on the Clearpath MC brake motor.
- controller.py: Translates ROS “joy” messages to “cmd\_vel” messages. Overrides platooning commands and can be used to control the golf carts in a drive-by-wire configuration. Adapted from “henrychassiccontroller.py” by Jeronimo Cox.
- enable.py: Translates input from the autonomy switch on the cart to rosparam messages.

- `estop.py`: Translates input from the three e-stops on the cart to `rosparam` messages.
- `parameters.py`: Used as a reference class to import `rosparam` messages to the other Python files. “`check_static()`” is used to check parameters on startup. “`check_dynamic()`” is used to check parameters that change throughout operation of the golf carts, such as the e-stop.
- `PCANBasic.py`: Used as a reference class for CAN communication through an attached PCAN adapter. Required for the “`steering.py`” code. Adapted from “`PCANBasic.py`” from PEAK Systems.
- `pid.py`: Runs the PID controllers for the steering, braking, and acceleration. Adapted from the “`simple_pid`” class by Martin Lundberg.
- `Read_RPM.py`: Converts frequency from a speedometer to RPM. Used as a reference class for the “`encoder.py`” code. Adapted from the “RPM Monitor” example from the `pigpio` library.
- `steering.py`: Controls the Nexsteer EPS Modules through CAN communication. Also provides feedback for the PID in the form of the angle of the steering wheel. Adapted from “`steering.py`” from the 2021 Campus Vehicle Team.

### ***Platooning***

The platooning package allows the user to apply platooning to the Campus Vehicle System in an open-loop configuration through a custom-developed platooning algorithm (Figure 16) which is based upon an algorithm used previously by the 2018 Campus Vehicle Team. To derive the new platooning algorithm, an assumption is made that the encoder and velocity readings from the EPS and speedometer are reliable enough to use as inputs for the algorithm, eliminating the need for correction. Closed-loop platooning is possible but requires further integration of the correction means into the platooning algorithm.

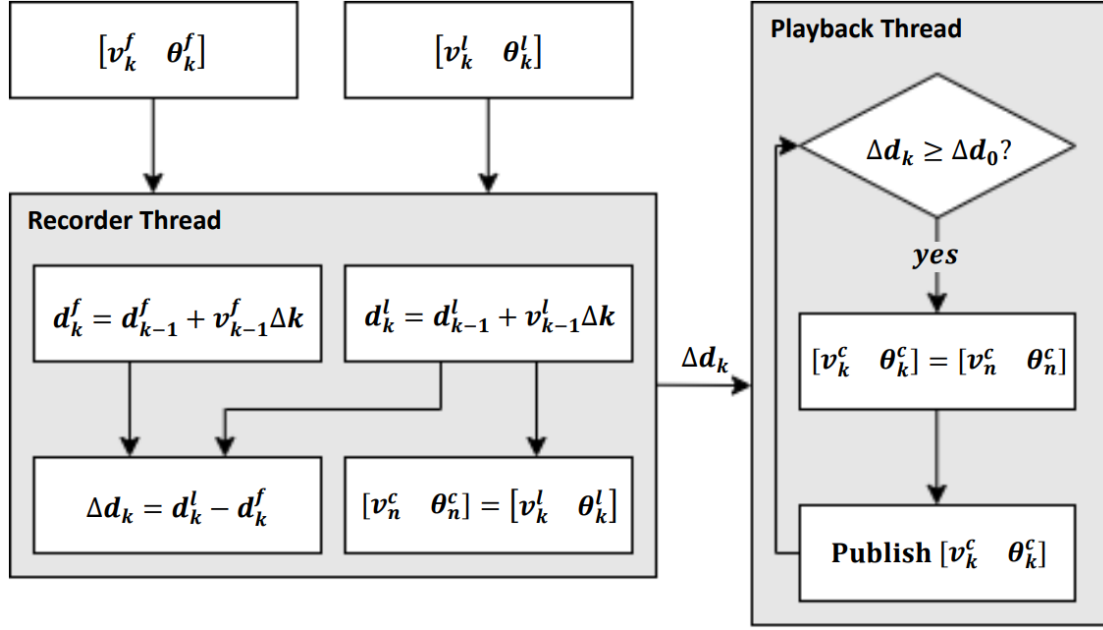


Figure 16: Platooning Algorithm Flowchart. (Gilchrist Johnson, 2023)

The algorithm starts with inputs from the EPS and speedometer as matrices that represent the trajectory of each cart,  $[v_k^f \ \theta_k^f]$  and  $[v_k^l \ \theta_k^l]$ . In the notation,  $k$  represents time,  $v$  represents velocity,  $\theta$  represents the angle, and  $f$  and  $l$  represent the leader and follower carts, respectively. This data is then used to calculate the distance each cart has traveled,  $d_k^f$  and  $d_k^l$ . The leader trajectory,  $[v_k^f \ \theta_k^f]$ , is then passed into a queue called the command queue, denoted by  $c$ . Finally, the distance between carts,  $\Delta d_k$ , is calculated using  $d_k^f$  and  $d_k^l$ . The algorithm is designed to utilize multithreading, ensuring the recording process is separate from the command publishing process. This design is meant to increase safety, as the calculation of the distance between carts is independent of the carts moving. Thus, the second thread waits for  $\Delta d_k$ . Once it receives it, it checks if that distance is greater than the platooning distance,  $\Delta d_0$ . If true, the algorithm pulls the next command from the command queue,  $[v_n^c \ \theta_n^c]$ , and publishes it as a twist message in ROS  $[v_k^c \ \theta_k^c]$ .



## Experimental Validation

### FEA

Two custom-designed mounts were used in the manufacturing of each cart: A mount for the Nexsteer EPS and a mount for the Clearpath MC motor. As previously described, Nexsteer only provided a mount for one of the provided EPS systems. One of the EPS systems was previously installed on one of the golf carts. Thus, the aim was to simply mirror that system on the second cart to have two interchangeable and equally capable carts. The only parameter that was required was that the electric power steering unit be the same module as the first cart, from Nexsteer. However, the model for the new cart was slightly different so the machined mounts that were installed on the carts were run through a simple finite element analysis to make sure that there would be no issues (Figure 17).

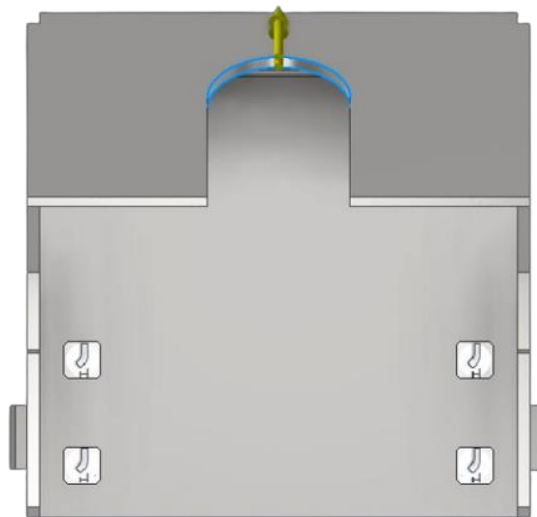
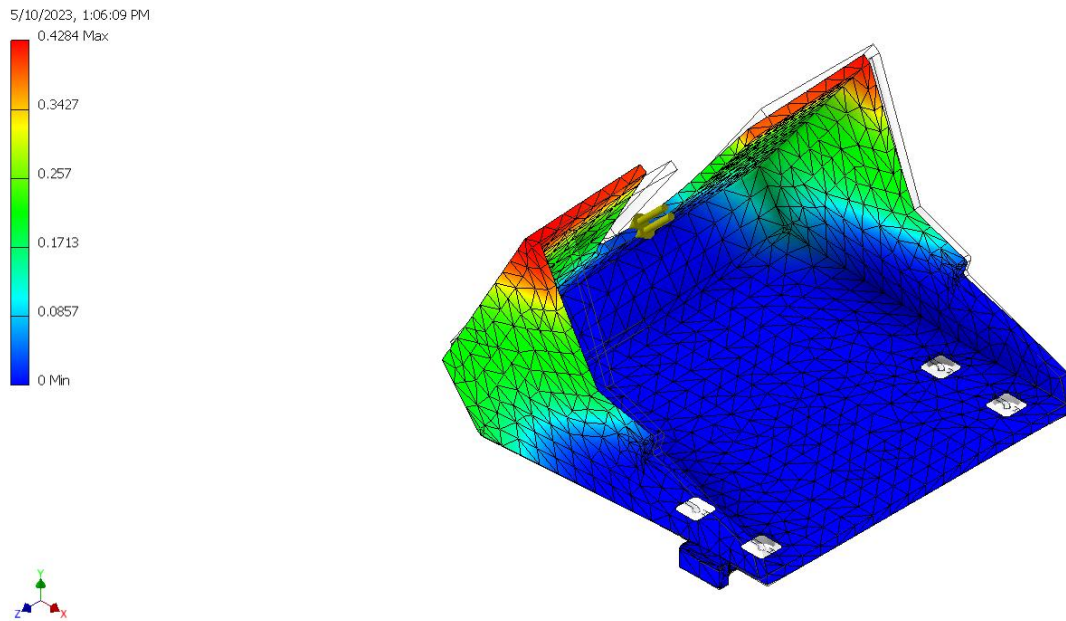


Figure 17: Top View of Mount. (Cameron Chiaramonte, 2023)

The top view of the mount shows the constraints in the four bolt holes along with the force applied on the neck of the mount. What is not shown is a fixed constraint on the bottom of the mount where it will be in contact with the dash of the golf cart. To allow for a correct

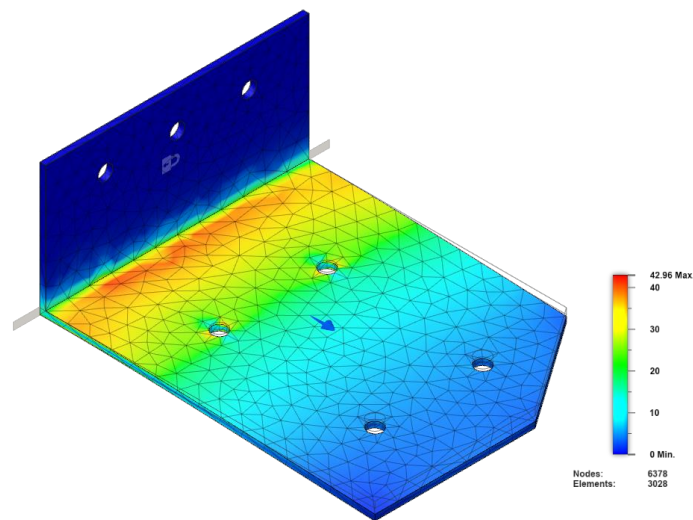
analysis the constraints in the bolt holes are pin constraints that only fix two degrees of freedom, tangential and axial. The third degree of freedom (radial) is left unconstrained. Moreover, this FEA was done in an absolute worst-case scenario where the entire weight of the Nexteer EPS is on the neck of the mount if someone were to pull down on it which would typically never happen (Figure 18).



*Figure 18: Nexsteer Mount FEA Deformation Results. (Cameron Chiaramonte, 2023)*

After meshing the mount and running the analysis the results show, in hundredths of an inch, that the maximum deformation at the top of the neck would be roughly 0.4, or 0.004in. This is completely negligible and was as expected. The parts are machined from cold rolled steel and act simply as a fixture to hold the EPS in place so there is no cause for failure from normal operation. Moreover, one way to improve upon this even further would be to copy the welded steering column cover from the original cart which prevents vertical and lateral movement of the steering column and therefore further limits any movement of the EPS.

A similar process was used for the brake mount. As one of the Clearpath MC motors was installed by a previous team, we had to design a mount to ensure both golf carts were interchangeable. To ensure the mount could withstand the forces applied to the motor, FEA analysis was conducted (Figure 19).



*Figure 19: Clearpath MC Mount FEA Deformation Results. (Patrick Dunnington, 2023)*

These studies were based on extreme scenarios to prove the plates would not fail even in extreme circumstances. For the studies, the top plate was fixed where it was attached to the cart and the bottom plate was fixed where it was attached to the top plate. While this is not exactly how the plates would perform, it provided a worst-case scenario at the corners where stress concentrations arise. The top plate would experience a stress along the side that attached to the bottom plate as the motor pulled on the brake. For the bottom plate the weight of the motor and the motor's torque to apply the brake created a moment in two directions. However, the maximum Von Mises stresses on both plates were below 50MPa.

## Platooning

The platooning system relies heavily on the data being read from the speedometers on the golf carts. Thus, we wanted to test the reliability of the encoders with regards to calculating the distance each cart travels (Figure 20). To conduct this test, distances of 0m, 2m, 4m, 6m, 8m, and 10m were measured on the road. Then, the golf cart was driven at a speed of 1.0m per second until the cart passed the measured goal. To minimize experimental error, each test was run three times.

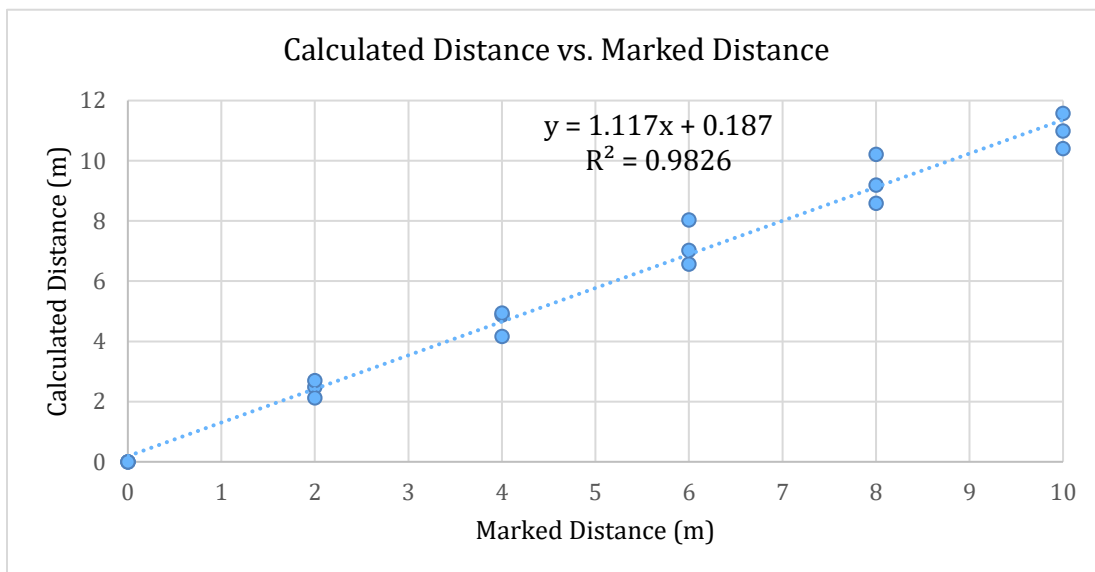


Figure 20: Experimental Calculated Distance vs. Measured Distance. (Gilchrist Johnson, 2023)

The results show a linear regression model of  $y = 1.117x + 0.187$  with an  $R^2$  value of 0.9826. This indicates a strong positive correlation between the two variables, with the calculated distance being approximately 1.117 times greater than the marked distance plus a constant value of 0.187m. This is within expected parameters, with the constant value of 0.187m being accounted for with experimental error.

## Conclusions & Future Work

Over the last two semesters, the campus vehicles team has made extensive progress towards achieving the final system of having autonomous platooning golf carts. The first major initiative undertaken by the team was cleaning up and revising the electronics on board the existing golf carts. Previously, the wiring and electrical components, while functioning, were messy and made it hard for new users to understand the design of the system. This was changed by taking the time to tidy up the wiring and also with the implementation of the electronics box. This box contains all the hardware needed to run the golf cart as an autonomous platooning vehicle, and only requires one plug for power and two plugs for data. The team was also able to add the braking and steering systems from the first cart onto the second cart. This will enable both of the carts to be used as the leader or the follower. After all the hardware changes were implemented, the software and sensor suite of the cart were improved upon. First the cart was configured to be remotely driven using a Logitech controller which proved to be successful. Then, using a combination of feedback code written by a previous capstone group as well as sensor data from AR tag tracking, the cart was able to successfully follow an AR tag mounted on the back of the leader cart.

The team has identified multiple potential areas for improvement in the golf cart system that future capstone teams could improve upon. One improvement to the system would be increasing its sensor suite. This would include adding LiDAR which would provide a 360 degree view of the area around the cart while it's driving through its environment. The addition of a GPS system would also increase the sophistication of the system while being relatively simple to implement. Improvement to the quality of the camera on the cart can also be made. The current camera is limited in its range capabilities which limits the following distance of the carts. A

camera with the ability to track at longer distances will make the system more flexible and increase safety. Another area that could be improved is with the platooning algorithm. Currently, the platooning algorithm takes data from the leader cart and replays it with a delay on the follower cart while using the camera for error correction. With the additional sensors, the location of the cart can be predicted more accurately, increasing the safety and reliability of the system. The last improvement to the golf cart system would be adding additional golf carts. While technically the system of two golf carts is a platoon with a leader and a follower, demonstrating that the idea works at a larger scale with more golf carts would prove that this system can be an effective solution.

## References

Club Car DS Installation Notes. (n.d.). Retrieved from

<https://www.hpevs.com/Site/images/jpeg/instructions/club-car-ds-installation-notes-6-21-12.pdf>

Furukawa, T., Arriaga, A., Bray, W., Chen, A., Chu, S., Cox, D., Cox, J., Ishanzai, A., Karas, M., Lee, D., & Voelker, A. (2018). Spring Final Report [Unpublished undergraduate thesis]. Virginia Polytechnic Institute and State University.

Furukawa, T., Breza, G., Chandler, J., Khatouri, S., Kim, Z., Rushton, C., & Singh, H. (2022). Modification of a Golf Cart to Autonomous Campus Vehicle [Unpublished undergraduate thesis]. University of Virginia.

Furukawa, T., Fontelera, A., Shin, J., Smith, W., Wellman, P., & Yocom, J. (2021). Modification of a Golf Cart to Autonomous Campus Vehicle [Unpublished undergraduate thesis]. University of Virginia.

Teknic Inc. (2017). ClearPath-SDSK User Manual. Retrieved

from [https://www.teknic.com/files/downloads/clearpath\\_user\\_manual.pdf](https://www.teknic.com/files/downloads/clearpath_user_manual.pdf)

Ulrich, K. T., Eppinger, S. D., & Yang, M. C. (2019). Product Design and Development (7th ed.). McGraw-Hill Education. (ISBN-13: 978-1260043655)

Vehicle Platooning for Safety and Efficiency Act. (2018, January 20). American Legislative Exchange Council. Retrieved October 27, 2022, from

[https://alec.org/model\\_policy/vehicle-platooning-for-safety-and-efficiency-act](https://alec.org/model_policy/vehicle-platooning-for-safety-and-efficiency-act)

## Appendix I

### Campus Vehicle System Operations Manual

#### I.I: Turning the Cart On

There are two switches on the cart. The first switch is a red switch underneath the back seat, behind the Electronic Control Unit (Electronics Box in the back). Turn it until the notch in the red switch shows green and it clicks (Figure A1) , indicating the system is on, providing power to the two power rails under the seat (Figure A2). These rails provide power from the batteries to the electronic box ( DC 12V), steering (DC 12V), braking (DC 48V), and monitor (DC 12V). Then press the power button on the back of the Electronics Box (Figure A3) to turn on the systems inside (Raspberry Pi, Router, PCAN steering communication, and NUC).

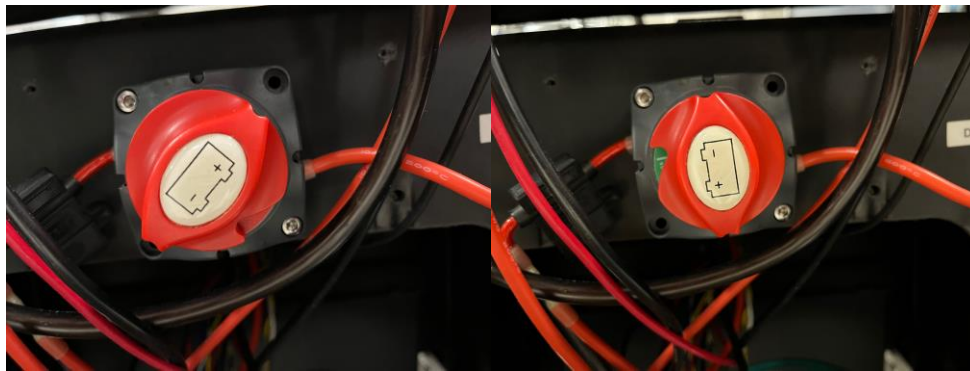


Figure A1: Switch on the back of the cart being turned on.

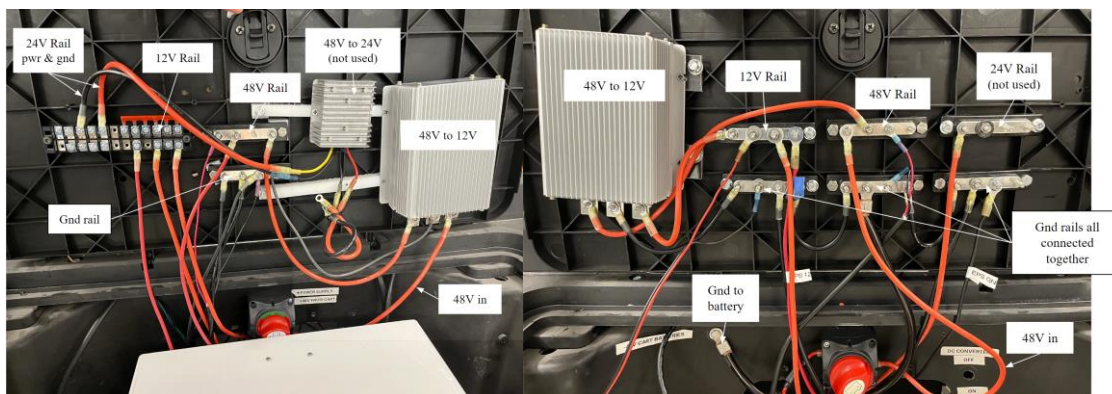


Figure A2: Labeled rails on the back of both Carts (788 on the left, 789 on the right)





Figure A3: Back of the Electronics Box

For the golf cart's electric motor, there is a run/tow switch underneath the front seats (Figure A4). Lift the front seats up, and the switch will be in the center against the back panel. Flip it up to run the motor. To verify that the switch is on, the battery light on the dashboard will be lit up. Ensure that when you turn the cart on, the red switch under the back seats is turned on before the box is switched on.

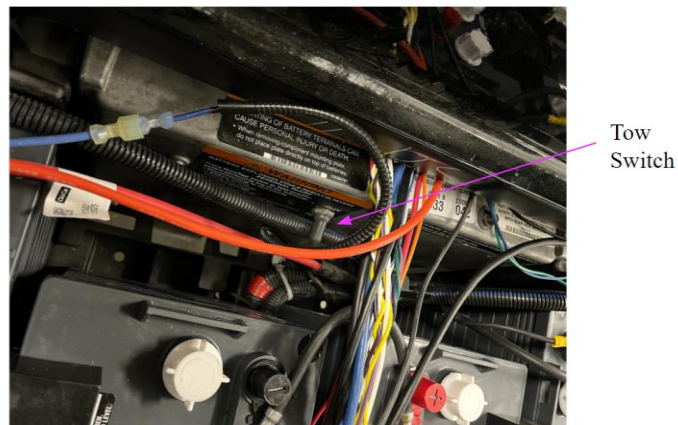
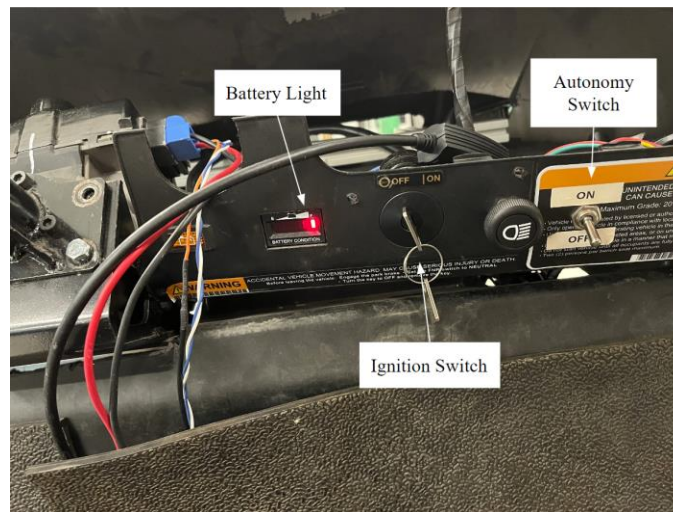


Figure A4: Tow switch under driver seat.

### **I.II: Normal Operation**

Once the cart is on, based on the steps above, make sure the autonomy switch on the front dash panel is off, and that the key is in the ignition and switched to on (Figure A5). Then the golf carts can be driven manually. Since this is a golf cart, there is a switch in between the driver and

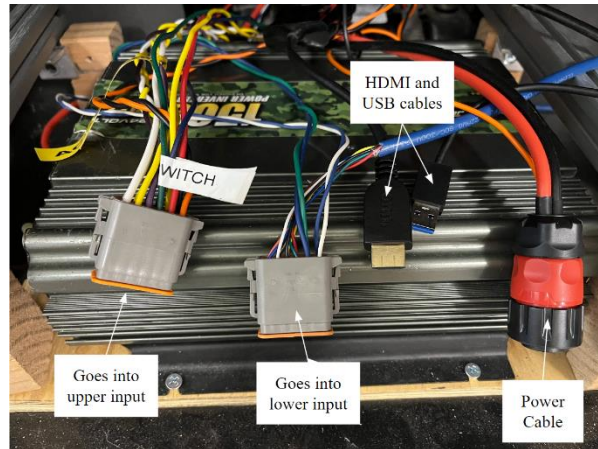
passenger seat to select forward (F), reverse (R), neutral (in between F and R). When the autonomy switch is turned on, the steering wheel will lock, and code needs to be run to achieve either joystick or platooning control of the cart. Turn the switch off again to unlock the steering wheel. Because of the braking design, the brake can be used manually in either mode in the case of an emergency, but in autonomous mode the servo motor under the cart can also control the brake pedal.



*Figure A5: Dashboard on the golf cart; Battery light is on indicating the tow switch has been turned on; Ignition switch is off so the cart won't move.*

### **I.III: Software Setup**

To implement platooning, make sure the keyboard and mouse are connected to the USB splitter and the USB splitter is plugged into the electronics box. Next, connect the HDMI cable from the monitor in the front to the HDMI port in the electronics box. Also, ensure that the two connectors consisting of sets of twelve wires each are plugged into the box (Figure A6). The connectors can only be physically plugged in one way; however, it is important to plug each connector into the correct input. One connector has a blue cord that splits into eight very small wires. This connector is plugged into the lower input on the box.



*Figure A6: Labeled cables in each cart; reference Figure 22 to know where to plug in each cord.*

The keyboard, mouse, and monitor should all be operational, and the Raspberry Pi will begin setting up on the monitor when the switch on the electronics box is on. The password to log in is “password” which should bring up the Ubuntu home screen. On this screen, there should be a desktop file named Setup File. Open this text document and a terminal window. In the terminal, copy and paste the first set of commands and run them. It may ask you for the password twice which is just the login password. Then, following the instructions on the Setup document, open a second terminal and run the next set of commands.

If this is done correctly, there should be multiple .py files initialized upon running the roslaunch command. These files allow remote communication, either from a remote control or talking between the carts (with a little more setup). To platoon, the carts must be on the same network, which is generated from the router inside of the box. Since the Raspberry Pi’s are running Ubuntu server 20.04 and not the desktop, you have to change the Wi-Fi connection from the command terminal. There are commands in the setup file to check what network the Pi is currently on, and how to change it. If changes are made to the router in the future, the password to the network is the same as what is labeled on the router. For admin access, the username is

always admin, and the password is “GolfCart” (if that doesn’t work then also try “golfcart”). The answer to both security questions is “golfcart”.

#### **I.IV: Turning the Cart Off**

To turn the cart off; first “^C” (ctrl+C) the programs running in the terminal and run “sudo shutdown -h now” from a terminal which will correctly shutdown Ubuntu. You can also shut down the Pi by going to the top right corner and choosing shutdown from the drop down menu. Once Ubuntu is shutdown, turn off the electronics box itself. Finally, flip the red switch behind the electronics box until it clicks, showing red in the notch on the switch, and turn the tow switch underneath the front seat off. To unplug the electronics box, be careful when removing the data inputs. Do NOT pull them by the wires, as the wires can come out (which requires force but is possible). Instead, pull it out by the gray part of the connector.

#### **I.V: Troubleshooting**

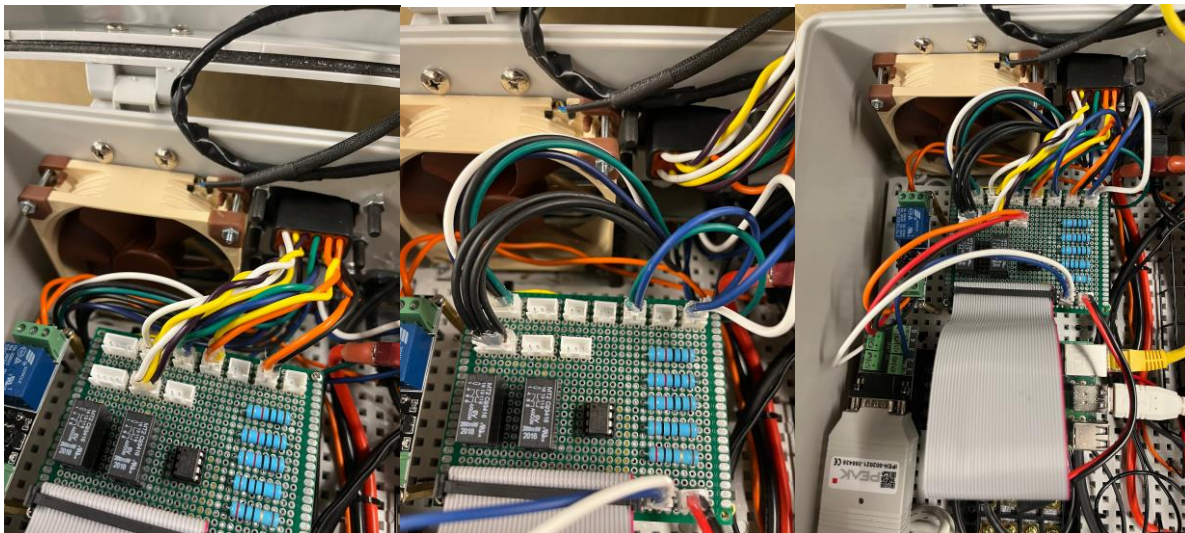
Due to the computing power required by the raspberry pi or possible alterations in current through the system, the pi can crash or glitch. If the sine converter is chosen to be set up, it draws a lot of power from the cart which can potentially cause, and has caused, systems to crash.

If the screen begins flashing/goes black, press the windows key repeatedly and eventually the screen will show normally again. If it freezes, then you will need to manually reboot the system by turning the box off and back on. It is VERY important that you keep a 3-amp fuse in the circuit with the raspberry pi, just in case the batteries or wires have problems. If the power is seeming to have trouble starting, check to see if there is a green light flashing (that indicates it is reading the SD card and is sending data to the monitor and is still working) if there is just a solid red LED on, the pi has power but is not communicating with the monitor or PCAN system.

Always have a backup of your code and the SD cards on the Pi. Win 32 Disk Imager is one software that can make a clone of a SD card.

Because of multi-threading and large quantities of code, as well as the Raspberry Pis running a faux desktop, they can have difficulty when running the full platooning system. It is recommended to run the platooning code off of a separate computer that can then send the ROS topics to the pis via wifi. There are two NUCs, one for each cart, that are running Ubuntu and have ROS set up, but do not yet have the platooning code installed. The NUC that runs off of 19.5V also has the capability of launching the Intel D435i camera, although upgrades to this system are recommended (especially using another camera such as a ZED camera).

Make sure you understand what connectors go into what spots on the board inside of each box (Figure A7). One of the boards is labeled, but it is important to follow these cords out of the box and to their respective systems so that you do not make any mistakes when plugging them back in.



*Figure A7: Connections for upper input (left) and connections for lower input (right).*

## Appendix II

### Customer Needs Survey Form

5/10/23, 2:49 AM

Customer Needs Form

## Customer Needs Form

For MAE 4610 - 2022-2023 Campus Vehicles Team. No personal data will be shared.

\* Indicates required question

---

### Transportation Options - Likert Scale

Choose a response based upon your agreement with the following statements.

*Likert Scale Options: Strongly Disagree (1), Slightly Disagree (2), Neutral (3), Slightly Agree (4), Strongly Agree (5)*

1. I use UVA transportation options everyday.

Mark only one oval.

Strongly Disagree

1

2

3

4

5

Strongly Agree



2. I would use additional transportation options \*  
given the ability to do so.

Mark only one oval.

Strongly Disagree

1

2

3

4

5

Strongly Agree

**Transportation Options - Multiple Selection**

Multiple Choice Options: Select all options applicable.

3. Select the transportation options you currently use to commute to (or on) campus: \*

Check all that apply.

- Walking
- VEO
- Personal Vehicle (Car, Bicycle, Scooter, etc.)
- Carpool
- Campus Bus
- Other: \_\_\_\_\_

**Transportation Options - Open Response**

*Type a response for the following questions.*

4. What do you use the given transportation options for?

---

5. What do you like about current transportation options?

---

6. What do you dislike about current transportation options?

---

7. How can current transportation options be improved?

---

---

This content is neither created nor endorsed by Google.





# Appendix II Club Car Wiring Schematic

