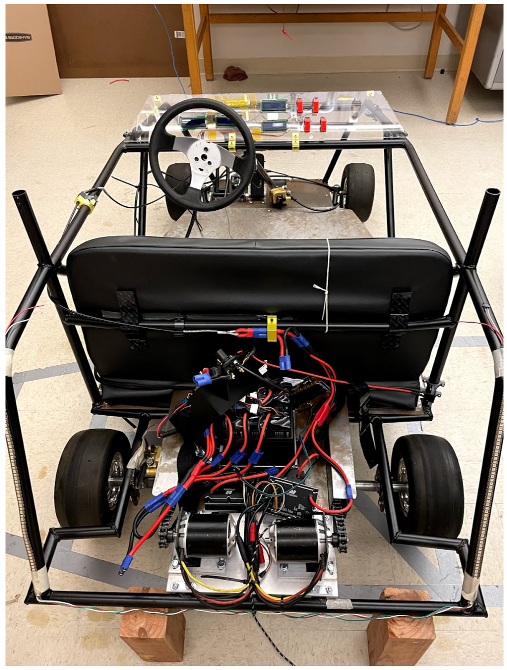
GoKart

Electric GoKart and Outreach Mid-Project Report
Fall 2021



ELECTRICAL AND COMPUTER ENGINEERING
COLORADO STATE UNIVERSITY

Electrical Engineering:
Nikola Durand, Vani Kapoor, David Neitenbach, Rico Barela

Computer Engineering:
Andie Groeling, Ryan Guidice

Vertically Integrated Program Students
Patrick Donovan(EE), Matt Gilmore(EE)

Supervising Professor
Olivera Notaros

Engineer in Residence
Doug Bartlett

Approved By: Doug Bartlett **Approval Date:** December 8th

Table Of Contents

Abstract

Electric GoKart is a fully electric small vehicle, built from the ground up, with a focus on highly modular and data collection to promote semi-autonomous actions. This year's iteration will include a new operating system, upgraded controls, and multiple new sensors. The GoKart will be developed by Electrical Engineering students focusing on controls and power systems, with Computer Engineering students focusing on operating systems, data management, and control programming. During this year's project development, it is important that we keep close records of our progress and decisions so we can use this vehicle to teach a younger audience about STEM* and how these fields intertwine.

Throughout the project the biggest constraint that the team has been working around is having no mechanical engineering students on the team. As stated above, the work around this was to set our focus on additions we can make to the Kart rather than much large improvements. The improvements made are revolving around our power management systems. The task at hand was to alter last year's team's design of the main system of the Go-Kart and through the process of schematic documentation and testing, the team has been able to diagnose and fix the issues that we began with at the start of the semester. This has given the Go-Kart a safer more efficient power system that can now be improved on in the coming semesters. Through part research and calculation, the EE* team has made the decision to add additional power systems to the front of the Go-Kart to show of PV* charging and to create a more stable system for us to be able to power the bulk of our control systems. These new additions will allow us to have a more stable and reliable system that is easy to debug in future years and a great talking point for the team and project to teach a younger audience about how the power systems would work. Further work and research has led the team to design an object detection system through machine learning techniques as well as implementing a fully functional touch screen user interface (UI) using a raspberry pi 4. More control research has been conducted as well to learn how we can use a CAN bus for more accurate control of our motors through the ESCs* which allows us to drive the Go-Kart.

The team also was tasked to have an organized plan and budget so that each of the above tasks could be accomplished. The use of team meetings and EIR* mentor check-ins proved the most helpful to keep the team on track and ensure that progress has been made in every aspect of the design. Budget management was taken on by one member in communication with the whole team to ensure that each purchase was not only feasible, but also was either completed smoothly or a solution was found to ensure the team's materials have been purchased. The difficulty arises as the supply chain shortage has greatly affected purchasing and ordering in many ways. These shortages have put the team off schedule with a lot of uncertainty when parts can arrive. This has been overcome by working and improving on the items that we have and learning more about what more we can do with all of the available material.

With the means stated above the team has completed some integral portions of the project this semester. New safety features have been implemented in the back portion of the Go-Kart such as chain guards and ensuring our ground connections will not cross any high potential. We successfully completed our machine learning module and are now working to fully integrate the objection detection technology into the control system of the Go-Kart. After debugging and careful consideration, the battery system has been rewired to ensure the safety of our electronics and functionality of the Go-Kart. Still in progress are the full implementation of our front solar panel battery charger which will be supporting the Raspberry Pi 4B, as well as creating a more stable 12V rail system for our other electronics.

Next semester the team plans on continuing to optimize the power system, optimizing the original fundamental design of the project and finishing the real-time control system we are implementing. Through this process the Go-Kart next semester will be a better optimized electric vehicle with plenty of new flashy features that really employ the student outreach aspect of this project.

*Appendix A

Introduction

This project acts as a second year continuation project from last year's team including students from both Electrical and Mechanical engineering departments. This project not only serves as a senior design project for six Electrical and Computer engineering students, but a project to be used for student outreach to teach students who are interested in STEM*. This project is growing from just a functional electric vehicle, this year's team is adding on modern day features that show off the skills and knowledge of electrical engineering topics.

With roughly \$6000 remaining from last year's team, as well as the additional \$200 per senior design student given to the team, we were able to start out in a comfortable place to purchase and buy any additions to the Go-kart that we needed. In addition to these funds, the CSU* ECE* Outreach team has granted us an extra \$3000 in budget to add to our existing budget from last year's team. Our full planned out budget can be found in Appendices B and C.

The design restraints this year really revolved around the fact that this year we were unable to collaborate with the mechanical engineering department. This constraint known early enough in the process allowed us to shift gears quickly and we were able to move on and begin on troubleshooting last year's design. At the same time as the troubleshooting, the controls team was easily able to begin their work on the machine learning and UI* aspects of the Go-Kart. Electronically, the biggest constraint for the project is to ensure that the project is safe enough to demonstrate to young kids, and ensuring that everything is sound enough for easy travel with the Outreach team. These constraints have been carefully considered and the team is always looking for new ways to ensure the safety of ourselves and our audience.

The following report is an in depth look and analysis of our progress this semester. The first section is a summary of the work completed before the project was handed off to us, but the following sections will be all that we have completed throughout the semester. The first section highlights the difficulties we encountered as well as the initial improvements we had to make to the Go-Kart. This was a large portion of the work that was done by our electrical engineering team who also take care of power improvements and management. After this section, this document will go into more detail of all of our technical aspects we are adding to the Go-Kart. The first subsection will highlight the work done for our UI*. This section will speak to the use of QT designer and how the design of the UI* has been completed through this program and the tasks completed and faced with this. The item will be looking at our Machine Learning model that has been implemented. This will cover the use of the Raspberry PI 4B and the programs and systems used to accomplish object detection. After this we will go over the new serial to CAN communication and will cover the reasoning behind the change and how that was implemented. The following sections will introduce the new front solar panel charging circuit.This will go over both the design and implementation. Next this document will detail all of the power management and any design that is used to power individuals systems on the Go-Kart. This section will speak to the changes made due to the difficulties we faced at the beginning of the project as well as the new solar panel charging system we are adding to the front of the vehicle.

*Appendix A

Difficulties Encountered

At the end of last school year, the previous team had left us with a running vehicle. The vehicle had working brakes and was able to be driven by the arduino code written onto the teensy. The Go-Kart had a working soft start and key switch for some safety features and this is all that we anticipated to be left with. The first need for the Go-Kart was for a recruitment event and the vehicle would not turn on. A few members in the team began this trouble shooting before the semester began and we could not solve the issue until the middle of the semester when we fully understood the error that was made. The soft start switches mentioned above had the ground tied to a high voltage which fried most of the existing electronics on the Go-Kart. Another large issue that led to a lot of what was stated above was the lack of documentation from last year's team. They had one singular schematic and it involved the whole system. However this did not include anything new that they had added after their final presentation in April. This led us down a path where we had to spend weeks on end trying to document everything that they had done so we were able to move on.

The team also had a lot of trouble with the ESC's* at the beginning of the semester. After discovering the power issues stated above. We realized that we had broken the microprocessor chips that were in these devices. The team was then tasked with the replacement of these chips which did not prove to be an easy task. Three different team members had an attempt at removing just one of the chips and finally when this was completed we still had a few more to replace. The task has been completed, but it is important to note the effort it had taken when one of these devices broke.

A large issue that we will be continuing to encounter is the supply chain shortage. This has caused a lot of issues with purchasing and receiving items since items we need are not only more expensive, but increasingly difficult to come by. This has set us back a few times in our schedule, but the team was able to overcome this issue by moving their attention to items we can work on.

*Appendix A

User Interface/ Dashboard Design

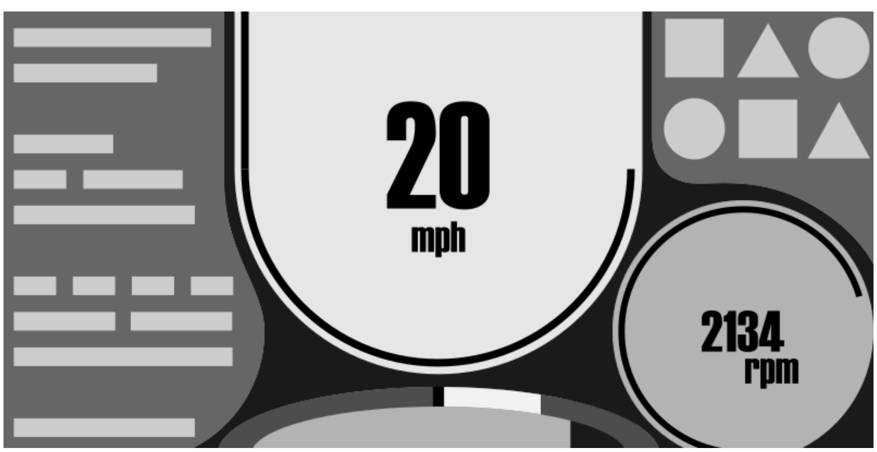


Figure 1: Mock-up of the User Interface

The user interface was first designed in inkscape to create the display that will be put onto the raspberry pi display. This design is then being developed in PyQT 6 designer which will give this design more functionality and work as the sole user interface controlling much of the components on the Go-Kart.

The purpose of this dashboard feature is to display many features that the team is deciding to implement and take measurements from. This first screen will include information such as speedometer, tachometer, and battery charge levels. Since this will be displayed on a touchscreen, the next biggest feature to be included is that students will be able to click the screen and then begin viewing the camera output as talked about in the next section. Along with selection options, this will also feature the ability to control all of the fun gimmicks that the team plans on adding next semester. We intend on including a color wheel to change the colors of the light strips that we are adding as well as displaying more data that is coming from new sensors that the team plans on adding next semester.

A key component of this dashboard feature is also to display when there will be any warning or faults in the control system. These faults will range from faults that the devices we have choses already give warning too as well as faults not yet fully chosen such as "battery critical" and motor and ESC* faults.

*Appendix A

Autonomous Pedestrian Detection (APD)

The Autonomous Pedestrian Detection (APD*) system is a machine learning model monitoring a webcam feed to perform object detection on a variety of vehicular classes, most notably pedestrians. This system was developed in partnership with Andrew Helmreich of the ML* on the Edge senior design team. The main goal of this feature is to provide a real-time demo of bounded-box object detection on the kart. While this data could theoretically be used to implement autonomous braking functionality (and we consistently evaluated the system's performance in this capacity), we decided this was too complex of a scenario to put our model into. Based on the overall goal of the kart, to demonstrate ECE* applications to students, we decided it was more important to have a live demo of this feature, rather than have it actually make decisions for the driver.

The base model of APD* is YOLOv5*, a machine learning model developed by Joseph Redmon and Ali Farhadi for real-time object detection. This model was selected due to its impressive performance in the real-time object detection area, where it consistently matches the accuracy of competing models while doing so in much less time. This was important for our application, as we theoretically want to keep the kart's stopping distance as low as possible in a braking scenario, and decreased model latency results in a shorter stopping distance. As for the data used to train the YOLOv5* model, we went with the KITTI* dataset. Developed by the Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago, the KITTI* dataset is heavily geared towards autonomous driving. With built-in classes for pedestrians, cars, cyclists, and other vehicle and traffic-oriented objects, this was the perfect choice as we didn't have to edit the dataset's built-in classes. We also added some of our own data using the Roboflow tool. By importing a few hundred of our own pictures of people, we used the tool to draw bounding boxes around them and label them as pedestrians, which added some extra data to the main class we were targeting with the APD* system.

To deploy this model, we are using a Raspberry Pi 4B with a Coral USB* accelerator and a Logitech C920X USB* webcam. The Raspberry Pi 4B was selected due to our familiarity with the Raspberry Pi line of single-board computers and the versatility they allow in these types of applications due to their small size, low power consumption, and relatively high performance. The CPU* of the Raspberry Pi 4B isn't quite powerful enough for our needs in machine learning inference performance though, so we purchased a Coral USB* accelerator to help in this category. By just plugging this accelerator in over USB*, making slight modifications to our deployment code, and re-compiling the model for Coral operations, we went from inference times per frame of ~3s to ~0.03s. Before we had the Coral, all 283 operations in our model's inference process had to be run on the Raspberry Pi's CPU*, but after re-compiling for the Coral, just 21 now run on the CPU*, with the remaining 262 operations being executed on the Coral accelerator. In addition to the Coral's custom Edge TPU* accelerating inference by being specifically designed for that application, the re-compilation also employs quantization on our model. Quantization approximates our original model from using 32-bit floating point numbers down to 8-bit integers, which only slightly hurts accuracy while drastically improving inference times. We also purchased a Logitech C920X USB* webcam to use as video input to our model due to its high resolution and good performance in autofocus and lighting correction.

Table #1: APD Performance at Various Resolutions

Input Resolution	Inference Time (s)	FPS*	CPU Ops*	TPU Ops*	mAP* (%)
256x256	0.021	47.62	0	285	45.7
320x320	0.032	31.25	21	262	53.7
384x384	0.048	20.83	21	263	54.2
448x448	0.092	10.86	21	263	56.4
512x512	0.163	6.13	21	263	58.1
640x640	0.275	3.64	40	246	61.3

As shown in Table 1, we evaluated several model resolutions for their performance in both inference time and mAP*. We settled on the 320x320 option as the best balance between FPS* and mAP*, as it maintains a respectable mAP score while maxing out the highest possible FPS* we can achieve with our deployment (since our webcam is capped at 30 FPS*). The 30 FPS* (due to the webcam limit) of this model means that if the kart was traveling at 20 MPH*, our system is able to make detections in each frame every 11.73 inches, which we believe is certainly fast enough for our demo. When testing this model in the lab and on footage we recorded from a car, the bounded boxes are fairly accurate, as shown in Figure 1. The image shows some inaccuracies in certain boxes, as well as missing a few cars, but it does notably well on the pedestrians and detecting the van in the background.

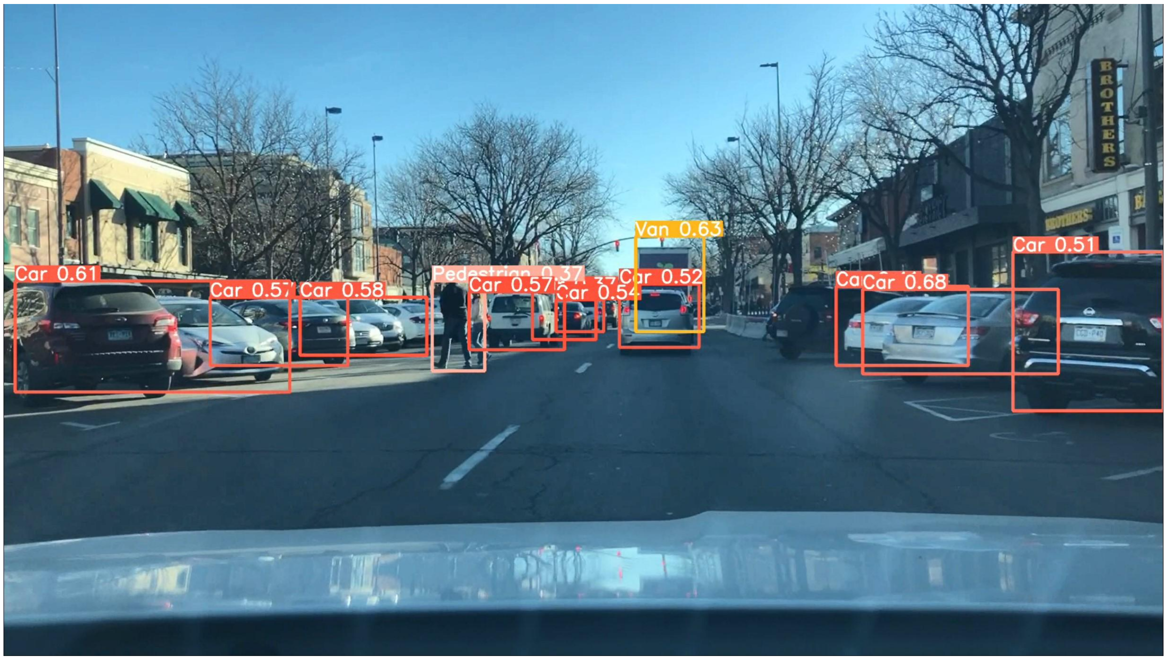


Figure 1: Bounded-Box Object Detection in Downtown Fort Collins

The final step for this feature is to integrate the live output into our new GUI* interface for the kart's display. We are able to display a live view of the model already, so this final step is not a major leap. We may also consider some slight tweaking of the model and its training after we mount the camera and system on the kart, depending on if the lower viewing angle on the kart impacts our performance.

*Appendix A

Component Communication and Data

Most component communication in our system falls into the category of time-sensitive or time-insensitive. Time sensitive parts could be operated using any of the protocols suitable for the component. For our system, we started using serial data and SPI* protocols for synchronous interfaces, time sensitive or not. With a single master, we would be able to have numerous slaves and a respectable data speed up to 10Mbps*. This was ideal because we could link sensors, controllers, and additional modules to one location for communication. I2C* was also being tested in conjunction to determine if it was a more economical solution. A typical I2C* bus would have a slower data transmission rate, but was still above minimum parameters to test. A great benefit of I2C* is its 7 bit or 10 bit address system, this would allow us to target specific devices on the bus. Compared to previously applied systems, I2C* was looking great. The need for only two wires and a device limit we wouldn't come close to made the protocol look like a good option. However, the required speeds to run the brakes, throttle, pedestrian detection, motor response, and other time sensitive functions were not there. On the test bench, devices were working at acceptable rates and data could be collected, parsed, and displayed. Upon install into the full scale system, noise became a major factor and rendered a considerable portion of the system inoperable. During this process, the team was already looking into other routes of transmission including UART* and CAN*. Because of the rapid increase in noise from the back of the kart to the front, we settled on trying an approach to CAN* transmission. This would allow for fast, time sensitive data to transfer very reliably and completely eliminate the risk of noise. There are already CAN* protocols in the motor controllers and we already have TEENSYv.4's in use that allow us to adjust component communication from serial to CAN* for use in this bus. In the case we establish communication with new components or systems and something is not working correctly, CAN* allows us to see the IDs of each message coming from the bus and allows us to change priority of IDs so we can see all data from them uninterrupted. There are multiple ways of parsing this data and having a central error diagnosis and configuration system makes this process a lot more manageable.

For non time sensitive electronics, the communication is mainly left to the designer. Currently, all of the non time sensitive systems are stand alone and do not communicate with the bulk of the vehicle. These systems include the radio, LED*s, and distance/motion sensors. Even though this means there is more work when trying to pull data from the system, it simplifies design and implementation. For the current state of the vehicle, this works well for us. Subsystems in progress follow the general rule that the communication is up to the designer. If there are components that do not have native CAN* communication already, we have the materials to transition them to can. As stated earlier, TEENSYv.4 is the primary tool used to change communication protocols, but there are other options available the team can invest in if needed.

There are a few things to be cautious of when integrating new systems onto the vehicle that need to be set up to communicate. Most importantly are wiring diagrams and power considerations. When wiring, we learned the hard way to be extremely cautious not to cross wires. There have been incidents of sending RxTx signals through power cables and incapacitating electronics on either end. Power is also something to consider, there are two power banks on the vehicle; one in front and one in back. The lead-acid battery in the front is used to power mostly time insensitive electronics, as more and more are attached we will need to evaluate if its performance is up to par and capable of its load. Without these considerations, there won't be any communication between anything.

*Appendix A

Front Solar Panel Battery Charger

The idea behind the solar panel charger first came from the decision to have a separate power system for the electronics being placed up front. The team felt as though the voltage rails needed if we were to over use the 5V supply voltage from the buck converter would not supply enough power to our control system which would then render the Go-Kart unusable in many aspects. To combat this issue the design came to add a simple 12V lead acid battery to the front of the vehicle that is easily accessible, and something we are able to buck down to a steady 5V voltage supply. The choice to use a lead acid battery in the front of the vehicle verses using the same LiPo* batteries that are used in the back are because of the way these batteries charge and how they react to overcharging. LiPo* batteries are very sensitive to overcharging, they have a threshold voltage that they cannot exceed and the same case for the voltage when discharged. This would cause the battery permanent damage and therefore we foresaw a far more need for replacement with these batteries. Lead acid is much more tolerable, they do not over charge to the point to where they would break and have no issue being recharged if the battery voltage drops too low. With this understanding the design became more feasible for the semester.

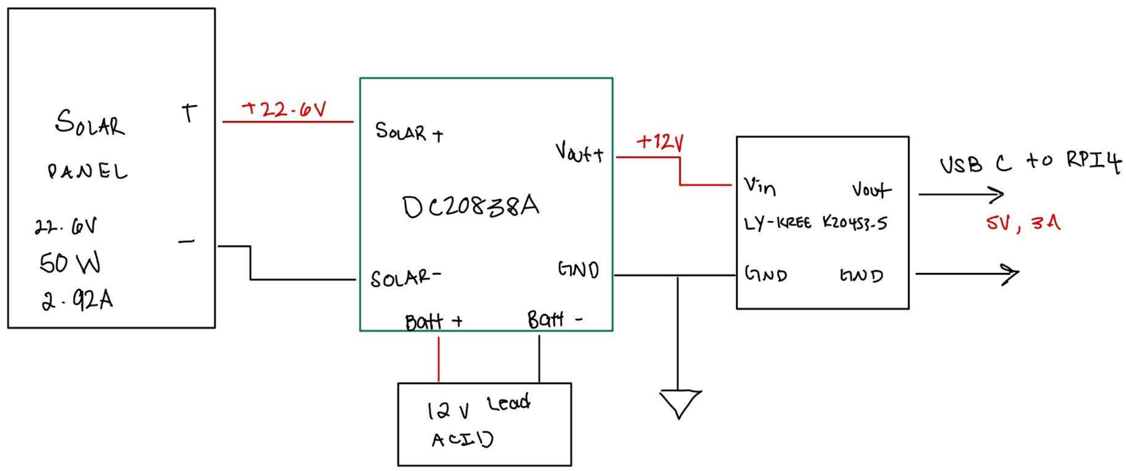


Figure 2: Solar Panel Charging Schematic

The above design includes a 22.6V 50W solar panel which was given to us by the Outreach team, a DC20838A BMS* a 12V-13V lead acid battery and a 5V regulator. The solar panel will be providing anywhere from a 18.75V to the max rated 22.6V input voltage to the BMS* which takes this voltage and supplies charging voltage back to the battery. As well as charging the battery, this device will also be outputting 12V which will serve as the input to our 5V regulator which will then be powering our Raspberry Pi 4B controller hub and 7inch display that goes with it.

This simple design is currently in the process of fabrication. Due to supply chain shortages, we have faced a lot of latency on our orders which has pushed back some of the assembly that has needed to take place. This circuit is placed on a plywood board which is fitted on mounts that we have created to create the look and feel of having electronics under the hood of the car. The solar panel that we have been given is flexible, therefore this will serve as the hood needed to hide the electronics until it is taken out for demonstration with the Outreach team.

Power Redesign

The design of the power systems in the kart had to change quite a bit from what it was last year. After documenting the wiring schematic from last year we found design choices that were causing issues with starting the kart this semester and also may have been causing issues for last semester's team.

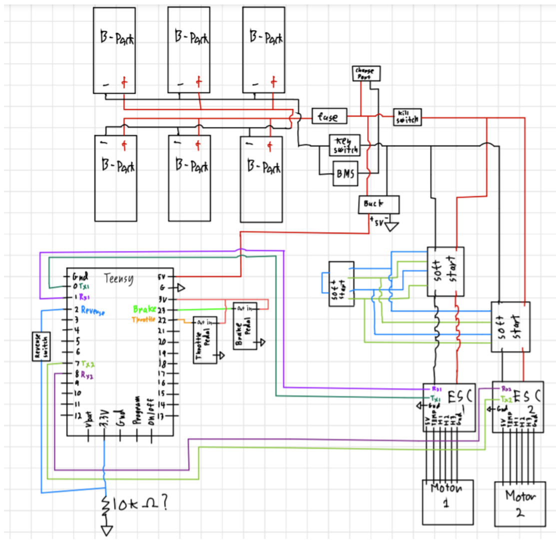


Figure 3: Wiring diagram of last semesters Go-Kart Electronics

From the wiring diagram and testing we found issues with grounding and connection points. The grounding issues were mostly due to the buck converter. With the buck converter the ground for the TEENSYv.4 controller and the rest of the battery power system were shared. This caused power voltage spikes to be present on the control side of the power system which broke the chips of the TEENSYv.4 as well as the Transistor on the throttle pedal. The buck converter could have also been the cause of signal noise issues from last year's team due to it being linked to the control ground. The duty cycle from the buck converter was linked to the signal ground of the TEENSYv.4, which we speculate contributed to unnecessary noise. The other grounding issues happened on the power side of the system with the power switch and the soft starts. The power switch being on the ground had a potential to have a current spike through the buck converter when switched off which could potentially break our controller chips. The problem with the soft starts took a lot of testing to find out what the problem was. It turned out the soft start used a mosfet with a PWM* to create a slow rising voltage differential between the positive and negative wires. The problem was that the mosfet was on the ground side making it so that the ground voltage for the ESCs* was higher than the control ground on the TEENSYv.4 until the soft starts hit steady state. This is a problem because the ESC* and TEENSYv.4 are supposed to share a ground and when the grounds have a voltage differential they cause a voltage larger than the ESC* and TEENSYv.4 chips could handle. Other than grounding difficulties we had a problem with the ESCs losing power at random times for no apparent reason. At first we thought that the connection problem was due to poor soldering but upon closer inspection it was the connector male plugs getting deformed with use that was causing them to be loose with the capability to become open circuited while plugged in.

To fix the grounding issues we had to completely redesign the power systems of the Kart. As mentioned earlier in the report we moved the control power system to a separate power source in the front of the kart to avoid all the issues with having the ground tied to the large voltage at the front of the kart. The redesign of the power side of the kart made it so things were less complicated and streamlined to avoid potential errors due to unneeded complexity.

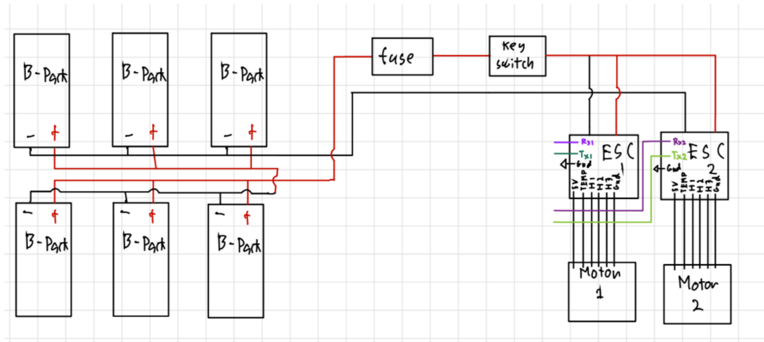


Figure 4: Redesigned Power system for the Motors

As seen in Figure 3 the battery is a lot less complicated than the previous system. The main changes made were the moving of the key switch, and the removal of the charge port as well as the soft starts. The key switch was moved so that the ground is never broken making for a more reliable circuit. The soft starts were removed to completely take out the ground voltage differential error and we have not had any problems since removing them. The charger port and BMS were removed as we could plug them into the batteries directly at any point to make the physical wiring less cluttered and easier to maneuver. The connection issue took a long time to find because the initial thought was that the wires themselves were faulty. As replacements were being made it showed that the male banana plugs in the connectors had the possibility of collapsing in on themselves and not putting pressure on the female banana plug. This was easily fixed by pushing the male plugs out with a screwdriver at all loose connection points.

*Appendix A

Standards

This team did not have to adhere to any standards. We operate at what is considered dangerous voltage; however, there are no standards that have needed to be followed unless the voltage is considered "high voltage".

Conclusion

Findings and Implementations

- The power systems were completely redesigned and implemented
- CAN communication was proven to be a viable way to control the ESCs
- Created UI that the user can interact with
- Created autonomous pedestrian detection that interfaces with the UI
- Created a control power system that is solar powered

Continuation

This semester brought a lot of challenges that were unexpected. Initially we were planning on just creating additions to the Go-Kart. This proved to be untrue as we had to redesign a lot of the previous team's work. The positives of this is that the issues that last year's team was having as well as the problems we are having have been resolved or are very close to being resolved. At the current stage we are in we are very close to getting the kart to run early next semester or over winter break. Once the kart can run, all of our individual parts of the project can be implemented and integrated with each other to have all of our goals met by the end of next semester. The rest of the project being finished before the end of the school year is well within scope.

Future Work

We still plan on continuing our shift from using microcontrollers as our main control system to single-board computers using our Raspberry Pi 4B and our ROCKPro64 boards. This will involve us using a SPI-CAN “middle man” for our devices that don’t support CAN directly to create two different controller hubs where we will be running all of our main control off of. We also plan on making more improvements to the power management system and optimizing the power coming from both the front and the back sides of the vehicle along with collecting data from these systems to know where our power levels lie. The next biggest step is to make the vehicle more appealing to younger audiences by adding devices such as sensors, bluetooth capable stereos, brake lights and underglow lights, and a redesigned dashboard with our 7-inch touchscreen attached to the raspberry pi.

This control system conversion to using CAN and assigning various kart system tasks to specific computing devices will take up the majority of the CE-sided work next semester. Our plan is to first get the APD system properly hooked into the new UI*, then deploy the UI and APD system on the Raspberry Pi and its touchscreen display. This will be mounted on a newly-cut acrylic dashboard. From there, we will be revising and editing the kart’s main control code on the Teensy to ensure all main kart functionality is working over the CAN bus. After that is done, we plan to add in the ROCKPro64 as a sensor hub that controls devices such as the lights. This will require careful planning, as the lights will be controlled on the UI running on the Raspberry Pi, which will send a CAN message to the ROCKPro64 to adjust the lights accordingly. We expect this process of integration to take the majority of the semester as we prepare for E-Days. While this is mostly CE-oriented, we expect plenty of collaboration with the EE side of the team to organize power for devices, wiring for communication between devices, and programming work to facilitate our complex communication strategy.

Other than design and additions to the Go-Kart, the team will be more heavily involved with the outreach aspect of the project and attending more events with the Outreach team.

References

[1]A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets Robotics: The KITTI Dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013, doi: 10.1177/0278364913491297.

[2]Ultralytics, “YOLOv5,” *GitHub*, Aug. 21, 2020. https://github.com/ultralytics/yolov5 (accessed Dec. 07, 2021).

[3]J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv*, vol. 1804, no. 02767, Apr. 2018, doi: arXiv:1804.02767.

[4]K. Yamazaki, “pytorch-armv7l,” *GitHub*, Nov. 29, 2021. https://github.com/Kashu7100/pytorch-armv7l (accessed Dec. 08, 2021).

Appendix A: Abbreviations

APD - Autonomous pedestrian detection

CAN - Controller Area Network

CPU - Central processing unit

CSU - Colorado State University

ECE - Electrical and Computer Engineering

EIR - Engineer in resident

ESC - Electronic Speed Controller

FPS - Frames per second

I2C - Inter Integrated Circuit

KITTI - short for KITTI Vision Benchmark Suite, an autonomous driving dataset

LED - Light emitting diode

Mbps - Megabits per second

ML - Machine learning

MPH - Miles per hour

Ops - short for Operations

PV - Photovoltaic

PWM - Pulse Width Modulation

SPI - Serial Peripheral Interface

STEM - Science Technology Engineering Mathematics

TPU - Tensor processing unit

UART - Universal Asynchronous Receiver/Transmitter

USB - Universal serial bus

YOLOv5 - version 5 of the You Only Look Once machine learning model

mAP - Mean average precision

Appendix B: Budget

Description	Quantity	Projected Unit Price
Initial		
Raspberry Pi 4B 4gb from pishop.us	2	$112.00
ROCKPro64 4gb from pine64.com	2	$79.99
ROCKPro64 20mm Mid Profile Heatsink from pine64.com	1	$8.99
ROCKPro64 12V 3A US Power Supply from pine64.com	1	$15.99
2-pack 32gb class 10 microSD cards from amazon.com	1	$1.95
RPi4B Aluminum Heatsink Pack from pishop.us	1	$3.95
3ft Micro-HDMI to HDMI cable from pishop.us	1	$2.45
HDMI cable from pishop.us	1	$64.95
Raspberry Pi 7” touchscreen display from pishop.us	1	$9.99
Raspberry Pi 15W power supply from amazon.com	1	$60.00
Teensy 4.0 amazon.com	1	$79.98
Microcontroller for solar panel ti.com	2	$99.99
Buck boost for solar panel ti.com	2	$119.95
ESC Chips	1	$22.99
8-40V to 12V 6A Regulator amazon.com	4	$12.59
6.3-22V to 5V Regulator amazon.com	2	$36.69
5 Pack Large Wire Connectors amazon.com	2	$17.40
ESC Debugger digikey.com	4	$46.21
6 Gauge Wire Black/Red 15 ft. amazon.com	1	$171.97

Coral USB Accelerator amazon.com	3	$64.52
Logitech C920x Webcam amazon.com	5	$99.00
Anti Spark Switch	3	$228.00
FSESC	3	$957.00
SPI-CAN Click	3	65.97

Total Spent = $2,385.81

Appendix C: Timeline

Date Range	**Objective**
8/30 - 9/5	● Define responsibilities of team members
9/6 - 9/12	● Complete FMEA and risk analysis documents ● Order controls parts ● Take group photo ● Started buck boost control code
9/13 - 9/19	● Finalize project goals ● Finish project website ● Write project plan ● Start analyzing last year's codebase and microcontrollers
9/20 - 9/26	● Start writing controls software ● Decide on type of ranging sensors (LIDAR/RADAR/ultrasonic) ● Begin work on solar charging system ● Mid-semester notebook check
9/27 - 10/3	● Began GUI development ● Started work on bluetooth LEDs
10/4 - 10/10	● Began pedestrian detection model development ● Redocumented missing wiring diagrams from previous team
10/11 - 10/17	● Revised project plan ● Radio system designed and mostly functional

10/18 - 10/24	• Held project management review with Susan Hunter • Got the kart working for the first time this semester
10/25 - 10/31	• Discovered cause of electrical issues • Finished current and voltage drivers
11/1 - 11/7	• Got initial versions of automatic pedestrian detection (APD) system working in development environment • Researched solar panel design concepts
11/8 - 11/14	• Got initial APD versions running the Raspberry Pi with the Coral USB Accelerator • Completed initial solar panel design
11/15 - 11/21	• Redid most of the power wiring from the battery • Drastically improved APD performance • Started parsing CAN data from the ESCs with IDs mapped
11/22 - 11/28	**Fall Break**
11/29 - 12/5	• Finalized APD resolution selection based on benchmarks • Ran APD over custom footage to test accuracy in tougher scenarios • Achieved reliable CAN data from the ESCs
12/6 - 12/13	• Created 12V and 5V rails with solar charger for the front of the kart • Tested all hardware for solar circuit • Present mid-project progress • Turn in mid-project report • Second notebook check

Acknowledgements

This project would not have been possible without the support from our virtually integrated students Matt, and Patrick. We really appreciate all of the hard work that they have put in this semester and have made some great additions to the Go-Kart.

We would especially like to thank Doug Bartlett for his continuous guidance and support that he provides weekly to the team. Doug has really pushed the team far and taught us so many skills that have brought the project to a different level. His drive to continue to teach us and push us to our full engineering potential has been something that the team really values and the skills we have learned from Doug are some that are unforgettable.

Another thanks is due to the Outreach and Olivera Notaros. The funding from the outreach team and the push to allow this project to continue another year has truly been amazing not only to the senior design projects, but the students who come to the Outreach events and learn from our team. We greatly appreciate the opportunities that we have been given with the continuation of this project.