GoKart

Electric GoKart and Outreach Project Plan

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

Summary

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.

Importance

This project is important to the Electrical and Computer Engineering department as a tool for Outreach. The GoKart will be used as a demonstration and show piece for schools. The completed and working GoKart is used by the ECE department to gain interest in STEM fields at CSU. The culmination of different skills and disciplines included in the design and fabrication of this vehicle shows the reach of engineering projects and the importance of inclusion and teamwork.

Revision History

Date	Comments	Version	Approved by
9/15/2021	**Make less technical**	**1.0**	**Doug Bartlett**
10/15/2021		**1.1**	

Problem Statement

GoKart is a second year project continuation with the goal of completing a well rounded teaching tool to gain interest in STEM from a younger audience and undeclared engineering students. There are already some projects developed by ECE outreach that can demonstrate some feats of engineering, but none have been as ambitious and as large in scale as Electric GoKart. This year's team is continuing development of the vehicle to be able to further demonstrate the capabilities of multiple disciplines working together. Through data collection and a number of sensors working simultaneously, this project will show the capabilities of our control systems, power systems, and autonomous machines, which will grow in scale as the project continues. This team consists of undergraduates concentrating in Electrical and Computer Engineering.

Electrical Engineering Objectives

Objectives for the Electrical Engineering team are split up into two subteams, power and control systems. This objective lists all the goals we will accomplish, some are required to ensure that Computer Engineers can complete their jobs. Some sensors and changes are tested on a separate vehicle, which is referred to as the “KitKart”, before they are integrated into the main GoKart. This minimizes risk of failure and lengthy troubleshooting or repair. The main focus is to ensure we can utilize the renewable energy sources implemented to build a flexible platform for learning. Electrical Engineering objectives are as follows.

- Step down voltage from batteries to controls
- Solar charging the batteries
- On board vehicle diagnostics and data
- Real time driving adjustments
- Optimizing power delivery from motors to wheels
- Upgraded power management and power collection from motors

Computer Engineering Objectives

Objectives for Computer Engineers are extremely important to many of the upgrades the senior design team is posed to develop this year. Because GoKart will be used around a wide audience of varying ages and technological knowledge, their program implementation needs to fit a wide variety of users. These objectives cover a range of processes and modes in the vehicle which will allow people of any age or skill to operate the vehicle safely and comfortably. Project redesigns are part of a larger picture to enable the use of more features and capabilities that will be available on the final product.

- Redesign control systems
    - Moving to Raspberry PI
        - Creating a location for data collection
        - Displaying data
        - Designing and implementing a graphic user interface
    - RockPRO64
        - Real-time operations on sensor data
        - Send all data through for real time processing, CAN data will be used and sent to a Raspberry PI
    - 7” touchscreen dashboard display
        - Shows data on multiple vehicle systems
        - Adjustable power for vehicle
        - Controls for added vehicle systems (lights, stereo)
- Low-level autonomous capabilities
    - Braking:
        - Software model for dealing with objects in front of kart

Electrical Engineering Research

- Solar panel power
  - Can feasibly charge the batteries from empty to full in a day
  - Look into dimensions of solar panel to put on cart
  - Power house has a lot of resources for testing and research
- Power Delivery
  - Switching to belts may make power delivery smoother for driver
  - Looks into effects of differing gear ratios
  - Look into how the escs switch current output and what would work the best
- Stepping down voltage
  - Two stage stepping for less noise
  - Wiring and shielding needs to be done to limit noise to controllers
- Buck-Boost Converters
  - Applying to solar panel charging output for extra or less voltage to the batteries.
  - Will be controlled by an MSP-EX432401Y Launchpad
    - Will be used to control current and voltage to the solar panels

This section covers some of the specifications required to operate the vehicle. Research has been ongoing for current vehicle systems and for new advancements and ideas. Most of the research is in power and control systems. It is important to ensure we stay within power and computational constraints, not doing so will result in extra expenses and time loss.

Computer Engineering Research

- CAN bus
  - Teensy CAN data generation
- UI framework & design
- Raspberry Pi 4B (4gb model):
  - Use for non-real-time applications, such as dashboard GUI
  - CPU and GPU powerful enough to handle GUI on 7” touchscreen display
  - Can install Raspbian or other Linux distros to fit our use case
  - Power requirements:
- RockPRO64:
  - Used for our real-time requirements
    - All sensor data will be interfaced either through serial or CAN
  - Power:

Frequency	Power cpu	Power vm
1296 MHz	2.64W	2.95W
1200 MHz	2.32W	2.69W
1008 MHz	1.90W	2.31W
816 MHz	1.62W	2.05W
600 MHz	1.45W	1.85W

408 MHz	1.33W	1.72W
Idle	1.10W	1.10W

Design Constraints

- Sensors
    - Must use noise resistant protocols
    - Must remain within budget
- Software
    - OS
        - Real-time
        - Low overhead
        - Easy to develop for
    - UI
        - User friendly
        - Uses screen space logically and efficiently
        - Flexible with growing/changing GoKart capabilities
    - Controller hub software
        - Hardware agnostic
        - Highly modular
        - Efficient handling of sensor data
        - Well documented
        - Developer friendly
- Compute Hardware
    - Real-time capabilities
    - Can run a complex graphical user interface
    - Can handle rapid communication with a multitude of sensors
    - Well documented
    - Easy to develop for
    - Remains within budget
    - Lots of I/O for sensors

Budget Justification:

The table shown below is a working description of our budget. It includes how much money we have received and from where it was given. It also shows how much we projected to spend on the items already purchased, the actual amount spent, the remaining budget, and the difference between the projected and actual expenditures.

Working Budget			
	Outreach	**Dept.**	**Total**
Total Received	$4116.79	$1200.00	$5316.79
Projected Spent	$606.47	$0.00	$606.47
Total Spent	$606.47	$0.00	$606.47
Total Remaining	$3510.32	$1,200	4710.32

The estimated budget is outlined below. This is itemized to represent our needs and goals for the project this year and all budget will be put towards the betterment of the vehicle. This is not including extra cost for damaged parts in need of replacement.

Description	Quantity	Projected Unit Price
Initial		
Raspberry Pi 4B 4gb	2	$55.00
ROCKPro64 4gb	2	$79.99
ROCKPro64 20mm Mid Profile Heatsink	1	$3.29
ROCKPro64 12V 3A US Power Supply	1	$8.99
2-pack 32gb class 10 microSD cards	1	$15.99
RPi4B Aluminum Heatsink Pack	1	$1.95
3ft Micro-HDMI to HDMI cable	1	$3.95
HDMI cable	1	$2.45
Raspberry Pi 7” touchscreen display	1	$64.95
Raspberry Pi 15W power supply	1	$9.99
Solar panel Charger (12-24V)	1	226.10
Lipo BMS litechpower.com	2	$53.99
400W Solar panel	2	$288.75
MSPEXPE401Y	1	$32.99
LiPo Batteries Amazon.com	4	$75.00
ESC/Maytech MTSVESC6.0 based 200A VESC	2	$490.99
Jumper wire kit	2	$10.00
MB1260 XL-MAXSONAR-EZL0 Ultra Ultrasonic sensors	4	$43.95
3 Meter NeoPixel Digital RGB LEDs for Under glow	1	$74.85
TTL Serial JPEG Camera with NTSC Video	3	$39.95
STM32F405RGT6 ESC Chips	5	$24.99
Total Projected Costs		**$3000.79**

FMEA Documentation

Process Step	Controls	Power Regulation	Power Regulation	Mechanical
Potential Failure Mode	Failure to Automatically Brake	Shorting of high voltage wires	Charge circuit failure	Broken Steering
Potential Failure Effect	Crash	Fire	Unable to charge Battery	Crash
SEV	6	10	2	6
Potential Causes	Sensor Failure, Control System Bug	Rubbing of wires, poor insulation	Blown capacitor, bad cell	Faulty weld, weak bolt, loose hardware
OCC	9	2	2	2
Current Process Controls	Poll for sensor data at high frequency and safety indicator for distance decreasing	Ensure no contact with metal, short wire lengths	Voltage sensing and display	Safety Switch to cut power in half
DET	4	10	2	4
RPN	216	200	8	48
Action Recommended	Build code in stages, check sensors before driving, and have indicators for data ranges. Include Shutoff switch and manual hand brake.	Use grommets and protective casings for high voltage wires		Simulating thoroughly in modeling software to identify weak points.

Process Step	Controls	Power Regulation	Power Regulation	Mechanical
Potential Failure Mode	Break the UI	Shorting of high voltage wires	Static shock to controller hub	Steering under/over performs
Potential Failure Effect	Inability to see GoKart data	Electrocution	Controls irreparably damaged	Unable to evade obstacle, possible rollover
SEV	6	2	7	7
Potential Causes	Breaking control loop within the UI	Wire cross, poor insulation, deterioration	Design/Operations failure	Jammed joint or quick tire rotation
OCC	2	3	5	1
Current Process Controls	Continuous QA testing of UI to find and eliminate bugs early in development	Most of total voltage shielded	None	Electronically controlled power system
DET	7	1	6	4
RPN	84	6	210	28
Action Recommended	Build code in stages, check sensors before driving, and have indicators for data ranges. Include Shutoff switch and manual hand brake.	shielding around the terminals	Isolate controller hub properly. Instruct team members on danger of static shock	Maintain practices of checking tires, steering column, and limiting power when using the vehicle

Risk Analysis Documentation

Risk Event	Probability	Impact(Hours)	Effect	Mitigation Plan	Person(s) Responsible
Not having a Mechanical Engineer	50	8	Drop aspects of the project	Assigned ME by certain date or not add any mechanical needed components	Olivera / ME department
Motor breaks	10	15	delay and cost	order a new motor	David or Rico
Raspberry pis don't ship or are on backorder	40	20	delay	wait until raspberry pi is available to purchase, get another Pine64 if past 4 weeks	Ryan and Andie
Batteries die	20	15	delay	Check continuity of batteries every other week	David or Rico
Software debugging takes longer than expected	100	30	delay	utilize git tools to effectively collaborate on software development	Ryan and Andie
Sensors break	50	5	delay/added cost	study datasheets and tutorials to effectively understand sensors prior to use, test prior to use and add protection	Ryan and Andie
Controller hub power delivery issues	30	9	delay	test and measure power supply to controller hub prior to first boot	Ryan and Andie
Parts become obsolete / inability to buy	70	18	delay/ redesign	Research equivalent or easily replaceable parts for all components ordered	Anyone on sub team who discovered the issue
Sensors are incompatible	20	5	delay	Research interfaces and libraries available for a sensor before ordering	Nikola and Vani
Computing power insufficient	30	10	delay, added cost	Use compute modules with significant computing power from the start	Nikola and Vani
Inconsistent PWM on motors	90	1	Extra wear to parts	Ensure that the motors are re-calibrated before each use	Nikola and Vani
Not getting budget approved	40	6	Less money to work with	More conscious purchasing and budgeting, cutting some projects	Everyone
Time Management falls behind	85	200	Systems are not completed or integrated	Use a Gantt chart to coordinate timing and objectives to keep all members on task	Everyone
Total Risk Hours		**357**			

Computer Engineering Timeline:

Computer Engineering Timeline:			
Task	**Date Start**	**Date End**	**Team Member**
Research Components	9/1	9/10	Ryan,Andie,Vani, Nikola
Order Raspberry PI and RockPRO64	9/7	9/7	Vani
Research Sensors: Ultrasonic Lidar, Sonar	9/7	9/15	Ryan,Andie,Vani, Nikola
Order Sensors	9/21	9/21	Anyone
Software Stack Architecture Design	9/21	9/21	Ryan,Andie,Vani, Nikola
Power Budget For Devices: Software Design	9/24	9/27	Vani, Nikola
Begin Coding UI with brake lights and old sensors	9/27	10/4	Ryan, Andie
Testing UI Code	10/18	10/25	Vani, Nikola
Distance detection	10/18	10/25	All
Begin new sensor integration	10/30	11/30	Ryan,Andie,Vani, Nikola
Send sensor data to raspberry pi display	10/30	11/10	Ryan, Andie, Nikola
Test GUI	11/1	11/5	Vani
Add sensors to front of GoKart	11/8	11/12	Anyone
Begin code for assisted braking	11/12	11/26	Ryan,Andie
Test code for assisted braking	11/26	12/3	Vani, Nikola
Make additions and changes to braking code	12/3	12/13	Ryan,Andie,Vani
Assisted braking code done	12/13	1/14	Ryan,Andie,Vani
Test Assisted braking	1/14	1/21	Anyone
Add underglow to Kart	1/21	1/25	Anyone
Add underglow controls to GUI	1/25	1/28	Ryan,Andie
Final Controls Validation	1/28	2/30	Ryan,Andie,Vani

Electrical Engineering Timeline

Electrical Engineering Timeline			
Task	**Date Start**	**Date End**	**Team Member**
Research Components and methods	9/1	9/20	Vani, Rico, David, Nikola
Order preliminary parts	9/10	9/20	David
Design Meeting: H-Bridge circuit integration	9/22	9/22	Vani,David,Rico
Design Step down voltage circuit	9/15	9/25	David, Rico
Order Voltage Converter components	9/25	9/25	David, Nikola
Design Solar Charging Circuit	9/25	10/15	David, Rico
Order Solar panel, controller, and converter	10/5	10/30	David
Design Controls for buck/boost converter	10/13	10/30	Vani, Rico, David
Project Management Review	10/25	10/25	Everyone
Test and iterate step down design	10/30	11/15	David, Vani, Rico
Test and iterate Solar Panel	11/15	12/15	David, Rico, Vani
Project Progress Review	12/3	12/3	Everyone
Project Presentation	12/3	12/3	Everyone
Design Solar Panel Wiring Harness	12/15	1/1	David, Rico, Nikola
Create Solar panel setup	1/1	1/15	David, Rico
Power Delivery Research	1/15	1/25	David, Rico, Nikola
Power Delivery Design	1/25	2/9	David, Rico
Power Delivery Parts Ordering	2/9	2/9	David, Rico
Implement Power Delivery and final tests	2/10	2/30	David, Rico, Vani