Electric Go-Kart

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

Computer Engineering:
Andie Groeling, Ryan Guidice

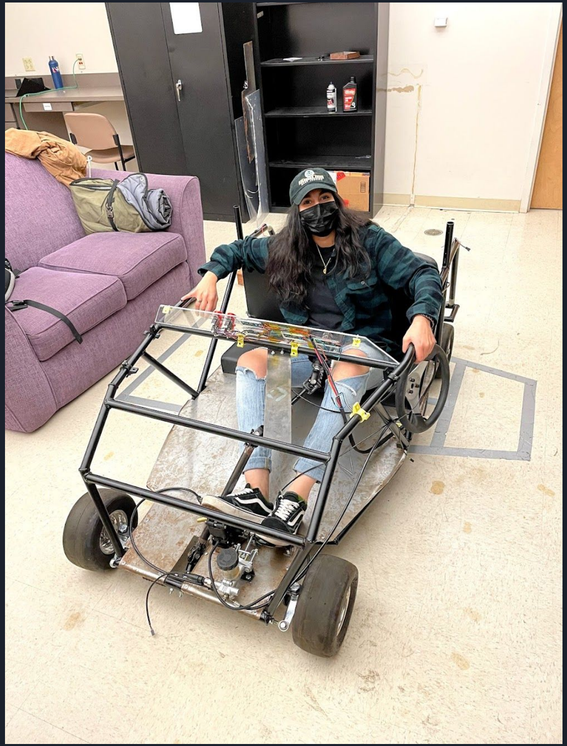
Vertically Integrated Program Students
Patrick Donovan(EE), Matt Gilmore(EE)

Supervising Professor
Olivera Notaros

Engineer in Residence
Doug Bartlett

Background

- Fully-electric go-kart
- Highly modular
- Part of Outreach team
  - Teaching students about engineering disciplines



Project Goals

Electrical Engineering:

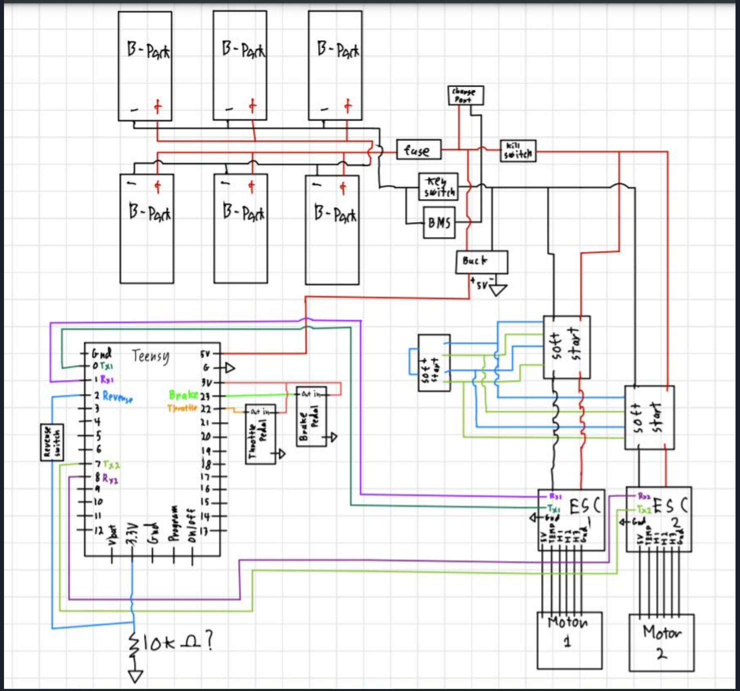
- Revisit kart's power management
- Implement solar charging
- Switch from UART to CAN

Computer Engineering:

- Redesign control systems
  - Raspberry Pi
    - Touchscreen dashboard
    - Support APD deployment
  - Switch to CAN
- Autonomous Pedestrian Detection (APD)

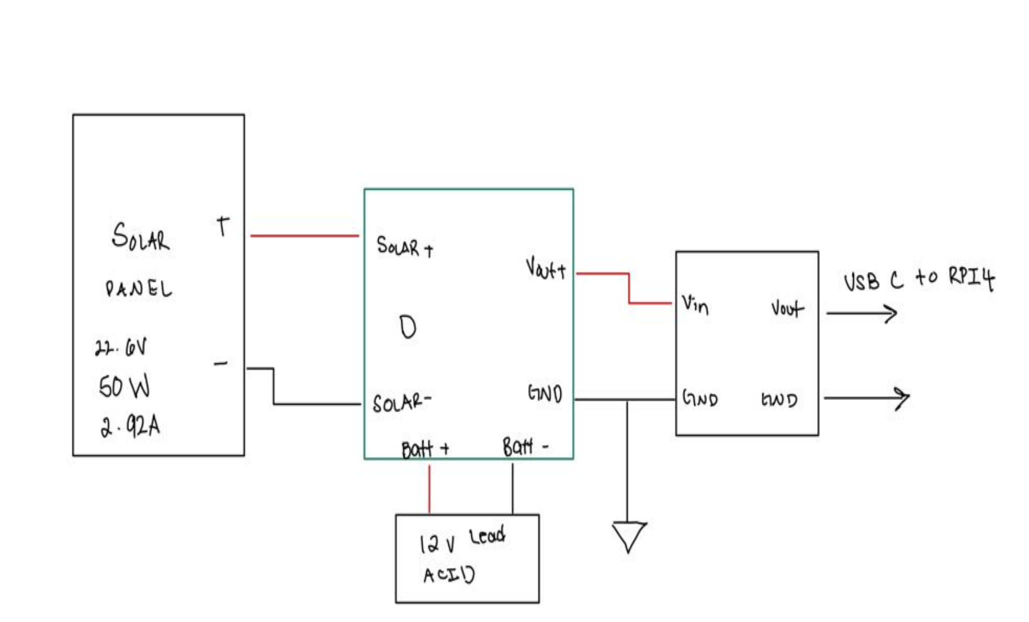
Continuation and Documentation

- Nothing really worked at the beginning
  - Couldn’t start the motors
  - Wiring was tangled and confusing
  - Lack of documentation
- Documentation:
  - Created ESC document
  - Redocumented wiring diagram
  - Labeled wires
  - Found grounding errors
  - Discoveries led to design changes



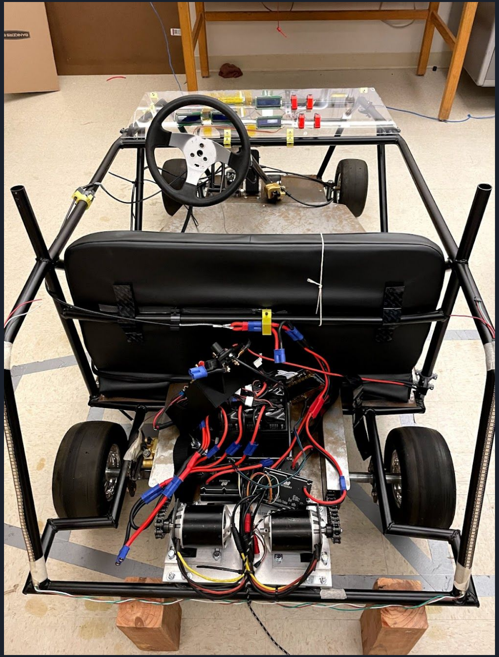
Solar Panel Battery Charger Design

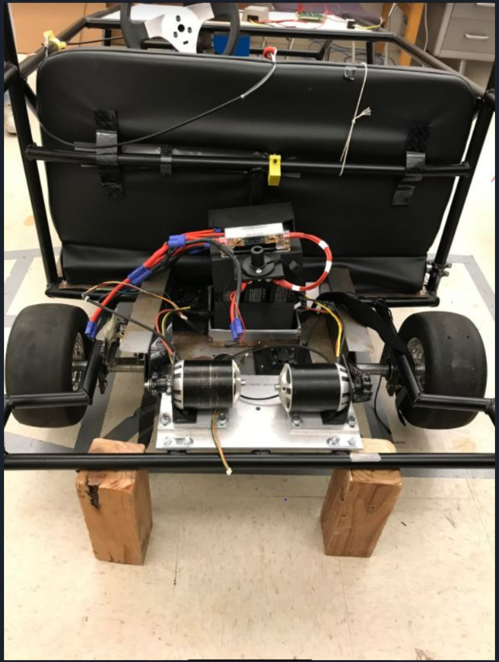
- Battery Voltage Rail system for Raspberry PI
    - How could we keep this system charged?
- Solar Panel
    - Provides the needed current for charging
- Lead Acid Battery
    - More charging forgiveness
- Implementation
    - 22.2V 50W Panel
    - 13V Battery
    - DC20838A-E BMS
- Progress:
    - Testing and Verification
    - Safety hardware



Wiring Modifications and Split Power Systems

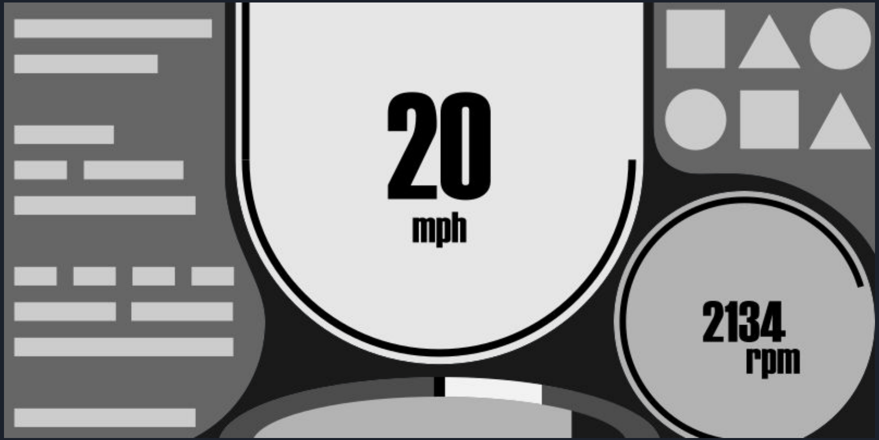
- Rewired the power system
  - Better organization
  - Took parts out
  - Changed paths
  - Fixed loose connections
- Divided power delivery into two systems (Front and Back)
  - To fix grounding errors
  - Reduce noise
  - To help implement solar panel charging





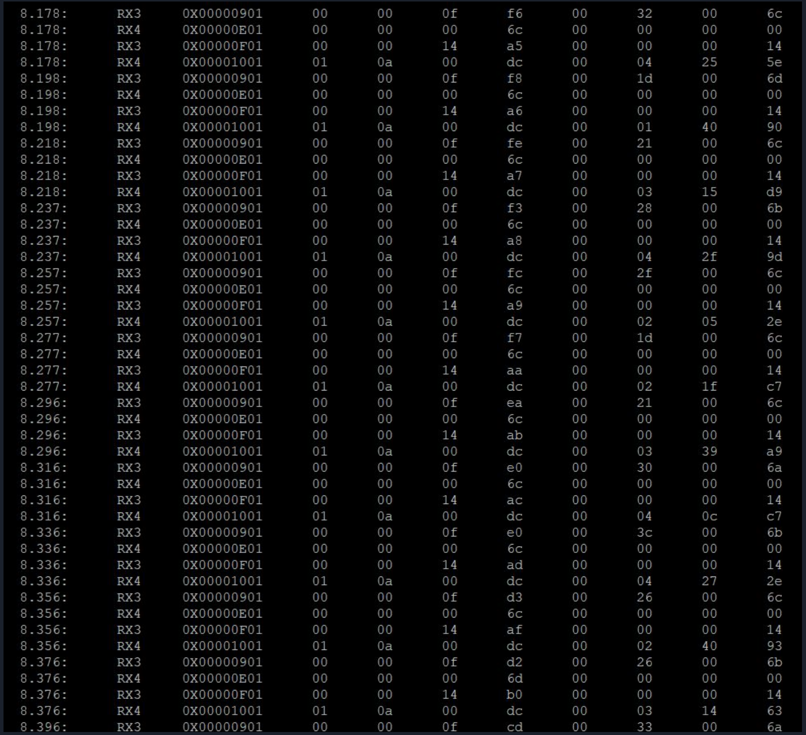
Touchscreen Dashboard

- Using PyQT 6
- Provides relevant information to driver from sensors
  - Speedometer
  - Tachometer
  - Camera with object detection
  - Battery Life
  - Regenerative braking
  - Warnings & Faults
- Also acts as a control panel for kart features
  - Lights
  - Camera



CAN and UART Communication

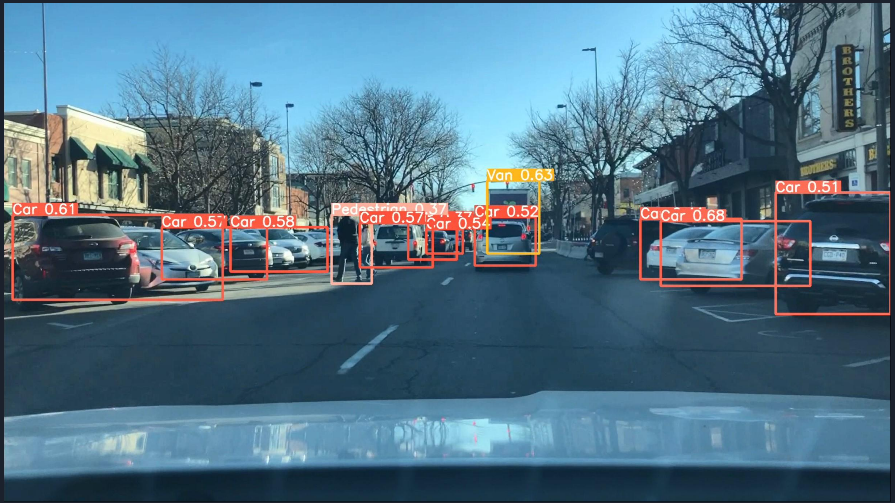
- Some bit rate is required by all units in CAN bus
    - Common master oscillator clock frequency not required
    - Non-return to zero, PLL synchronize receiver clock
- Some X number of UART buses for torque control
- One CAN bus can setpoint the value to the motor
- CAN detection from the ESC using puTTY and VESC tool
- UART communication devices
    - Over computer and peripheral device
    - Data stream not fast enough
- CAN asynchronous communication, real time performance, reliable
    - Devices not on CAN use peripheral devices to transition to CAN



```
8.178:    RX3   0X00000901   00   00   0f   f6   00   32   00   6c
8.178:    RX4   0X00000E01   00   00   00   6c   00   00   00   00
8.178:    RX3   0X00000F01   00   00   14   a5   00   00   00   14
8.178:    RX4   0X00001001   01   0a   00   dc   00   04   25   5e
8.198:    RX3   0X00000901   00   00   0f   f8   00   1d   00   6d
8.198:    RX4   0X00000E01   00   00   00   6c   00   00   00   00
8.198:    RX3   0X00000F01   00   00   14   a6   00   00   00   14
8.198:    RX4   0X00001001   01   0a   00   dc   00   01   40   90
8.218:    RX3   0X00000901   00   00   0f   fe   00   21   00   6c
8.218:    RX4   0X00000E01   00   00   00   6c   00   00   00   00
8.218:    RX3   0X00000F01   00   00   14   a7   00   00   00   14
8.218:    RX4   0X00001001   01   0a   00   dc   00   03   15   d9
8.237:    RX3   0X00000901   00   00   0f   f3   00   28   00   6b
8.237:    RX4   0X00000E01   00   00   00   6c   00   00   00   00
8.237:    RX3   0X00000F01   00   00   14   a8   00   00   00   14
8.237:    RX4   0X00001001   01   0a   00   dc   00   04   2f   9d
8.257:    RX3   0X00000901   00   00   0f   fc   00   2f   00   6c
8.257:    RX4   0X00000E01   00   00   00   6c   00   00   00   00
8.257:    RX3   0X00000F01   00   00   14   a9   00   00   00   14
8.257:    RX4   0X00001001   01   0a   00   dc   00   02   05   2e
8.277:    RX3   0X00000901   00   00   0f   f7   00   1d   00   6c
8.277:    RX4   0X00000E01   00   00   00   6c   00   00   00   00
8.277:    RX3   0X00000F01   00   00   14   aa   00   00   00   14
8.277:    RX4   0X00001001   01   0a   00   dc   00   02   1f   c7
8.296:    RX3   0X00000901   00   00   0f   ea   00   21   00   6c
8.296:    RX4   0X00000E01   00   00   00   6c   00   00   00   00
8.296:    RX3   0X00000F01   00   00   14   ab   00   00   00   14
8.296:    RX4   0X00001001   01   0a   00   dc   00   03   39   a9
8.316:    RX3   0X00000901   00   00   0f   e0   00   30   00   6a
8.316:    RX4   0X00000E01   00   00   00   6c   00   00   00   00
8.316:    RX3   0X00000F01   00   00   14   ac   00   00   00   14
8.316:    RX4   0X00001001   01   0a   00   dc   00   04   0c   c7
8.336:    RX3   0X00000901   00   00   0f   e0   00   3c   00   6b
8.336:    RX4   0X00000E01   00   00   00   6c   00   00   00   00
8.336:    RX3   0X00000F01   00   00   14   ad   00   00   00   14
8.336:    RX4   0X00001001   01   0a   00   dc   00   04   27   2e
8.356:    RX3   0X00000901   00   00   0f   d3   00   26   00   6c
8.356:    RX4   0X00000E01   00   00   00   6c   00   00   00   00
8.356:    RX3   0X00000F01   00   00   14   af   00   00   00   14
8.356:    RX4   0X00001001   01   0a   00   dc   00   02   40   93
8.376:    RX3   0X00000901   00   00   0f   d2   00   26   00   6b
8.376:    RX4   0X00000E01   00   00   00   6d   00   00   00   00
8.376:    RX3   0X00000F01   00   00   14   b0   00   00   00   14
8.376:    RX4   0X00001001   01   0a   00   dc   00   03   14   63
8.396:    RX3   0X00000901   00   00   0f   cd   00   33   00   6a
```

Autonomous Pedestrian Detection (APD)

- Worked with Andrew Helmreich (ML on the Edge)
- YOLOv5 model trained on the KITTI dataset
    - Some custom data added
- Deployed on Raspberry Pi 4B:
    - Coral USB Accelerator
    - 1080p USB Webcam
- Results:
    - Inference time (per frame): 32ms
    - FPS: 31.25
        - At 20mph, detect in 11.73in
    - mAP: 53.7%
- In-Progress: Live demo integrated into new UI



Budget

- Began with roughly $4500
  - Outreach gave us an additional $3000
- We have spent roughly $2,500
  - Replacement parts
  - New materials
- We still have 5,000 left
  - Projected spending was more than anticipated
- Anticipating more from fundraising and proposals next semester