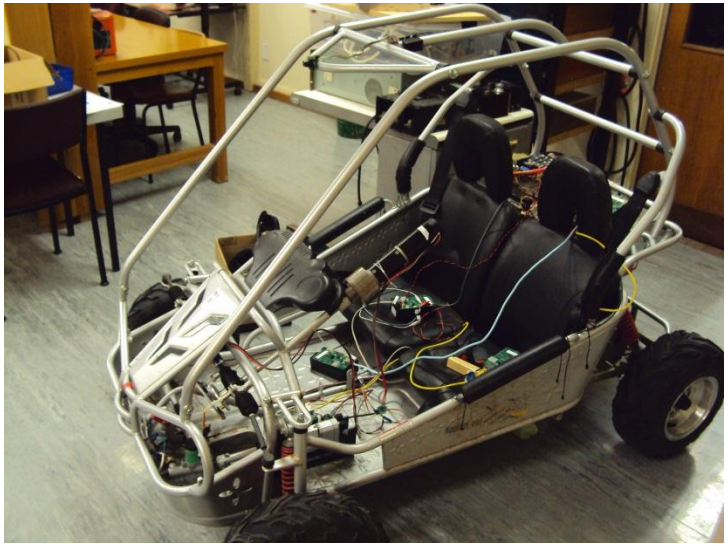
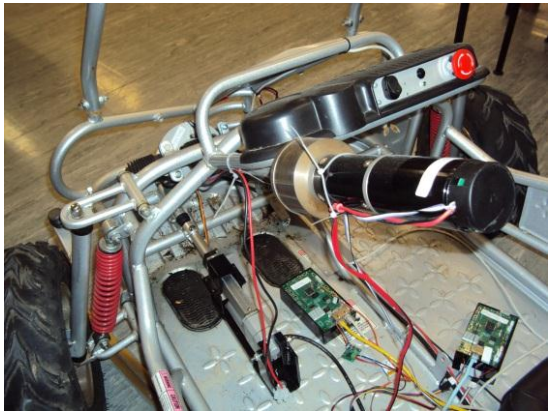
Drive-by-wire Go-kart



By Nissanka Weerekoon

System Overview

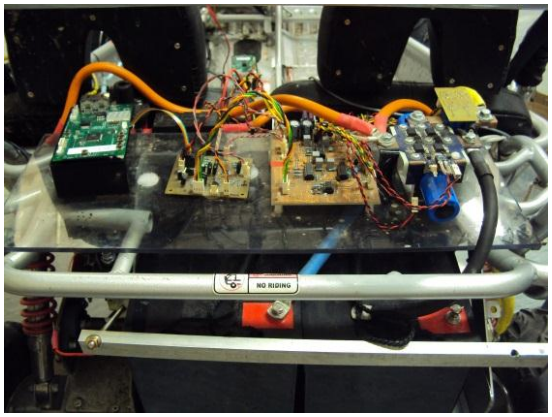
Steering



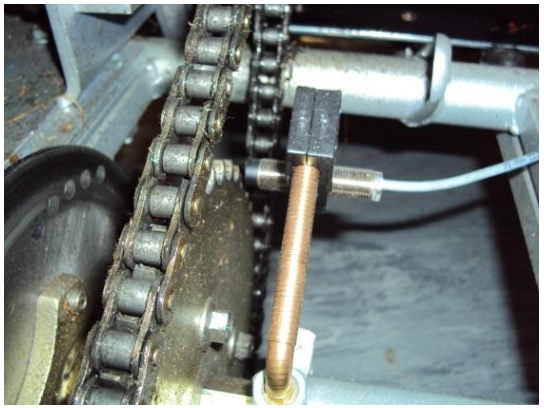
Comms Board

Brake

Motor Control



Speed Sensor



System Overview

Speed sensor

USB

Comms Board

Sensor board

CANBUS

Brake Board

Motor Board

Steering board

SPI

Driver Board

PWM Board

Driver Board

Brake Actuator

Potentiometer

Motor and Driving Electronics

Steering Motor

Encoder

Previous Work

2011- 4 students

- PCB design of CANBus boards
- Choose actuators
- Modify Go-kart chassis

2012- 1 student

- Python API for laptop control
- Bench testing of actuator control only

Contribution this year

1. Get existing system on to go-kart; show basic movement

2. Show integration with computer vision navigation

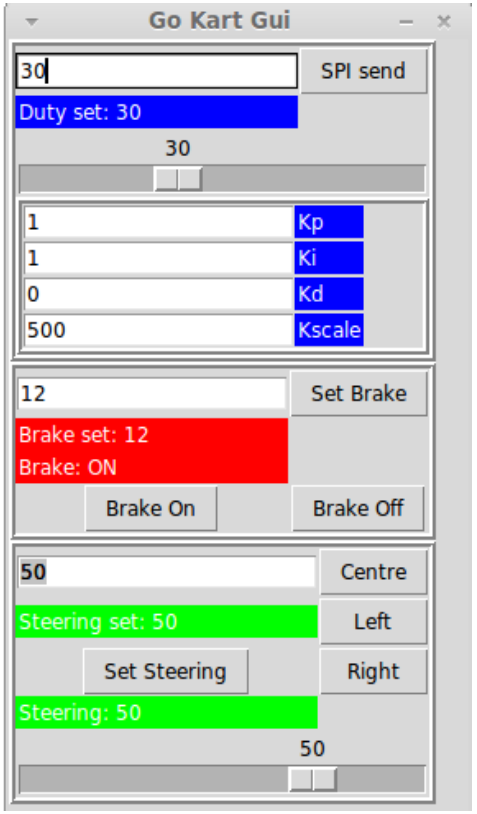
Drive-by-wire movement

Hardware:
Actuator bracket
Connectors
PWM board modification

Movement control
Laptop GUI

System Integration
Calibration – Brake, Steering, Speed
Motor control interface
- through PWM board

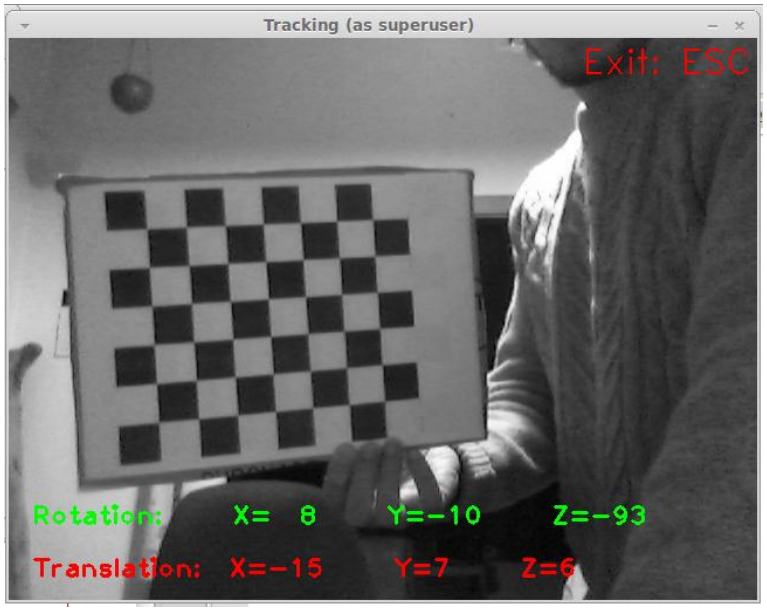
Proportional Speed control



Vision Integration

- Combining OpenCV checkerboard tracking algorithm (C++) with Python API
- Track the middle and maintain a set distance from checkerboard

Target	Go-kart response
Moves left	Steer left
Moves right	Steer right
Too close	Reduce speed, brake and stop if needed
Too far	Increase speed



Challenges Encountered

- There is a significant gap between bench testing and running on the go-kart
  - Grounding of boards and motor drivers
  - Interference from motor
  - Power sourcing and fusing

- Make minimal hardware changes/additions

- Latency of checkerboard tracking to actuator response – needs further testing

- Debugging error states including code from previous years (approx 3000 lines)

Future Work

Two possible paths…

Autonomous movement

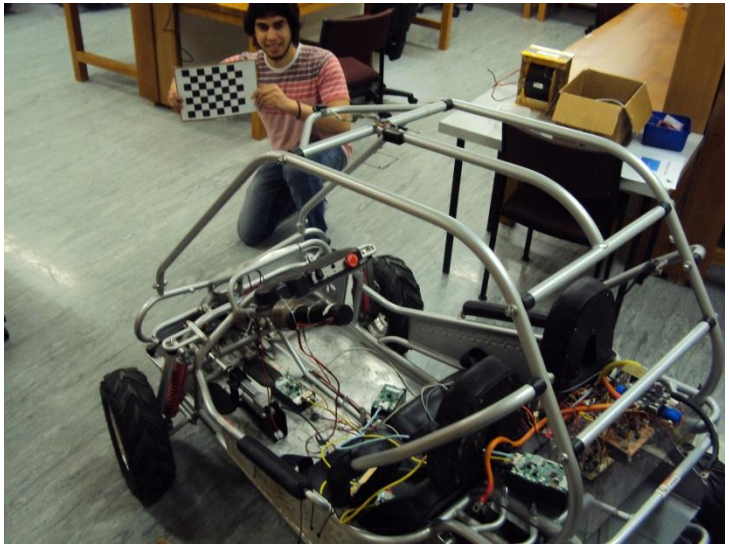
- Computer vision navigation
- Off-kart emergency stop
- Sensors for navigation (GPS, infrared..)

Rerun of current hardware

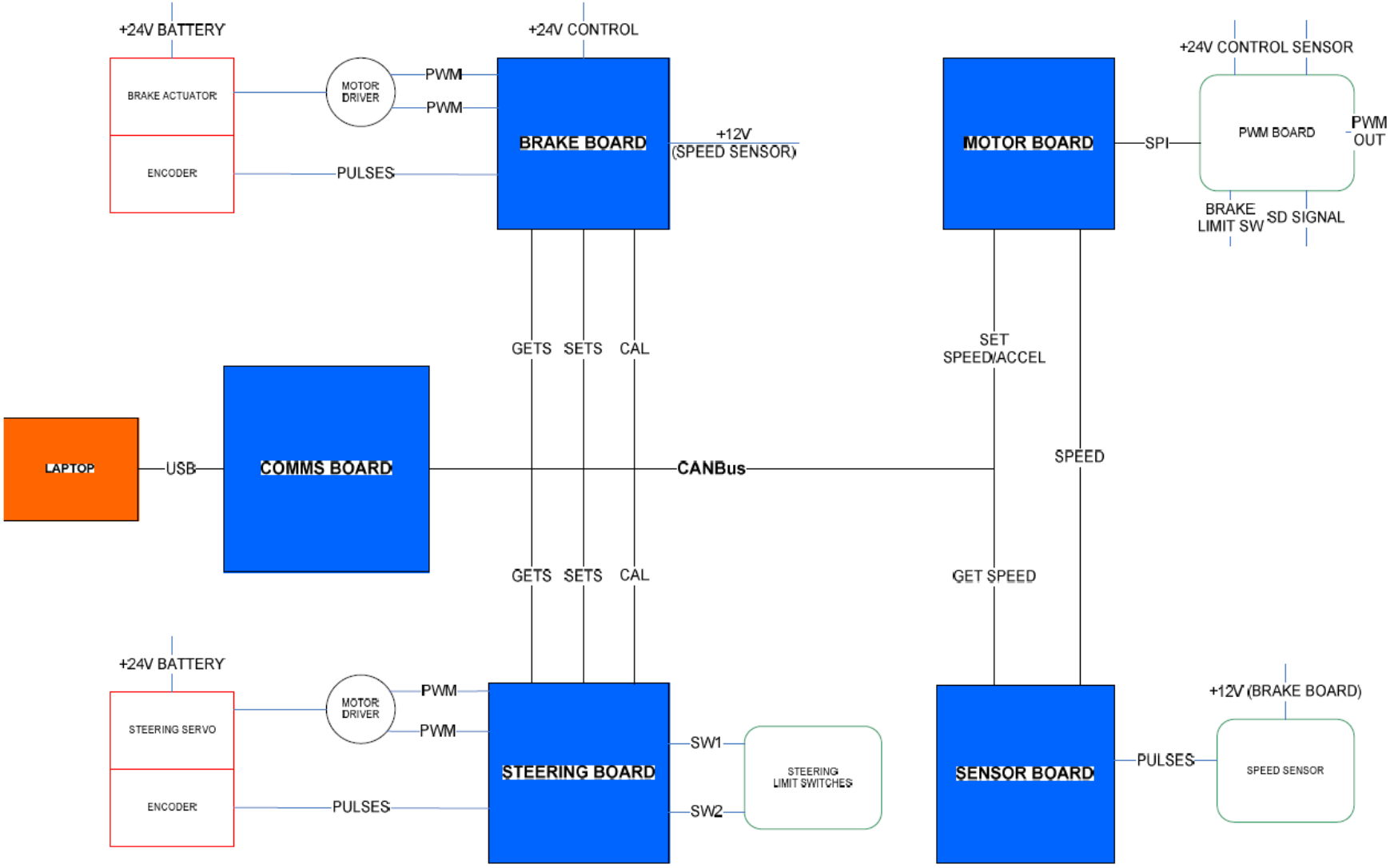
- Better connectors
- Wireless/Bluetooth
- Compatibility with new PWM boards

Thank you!

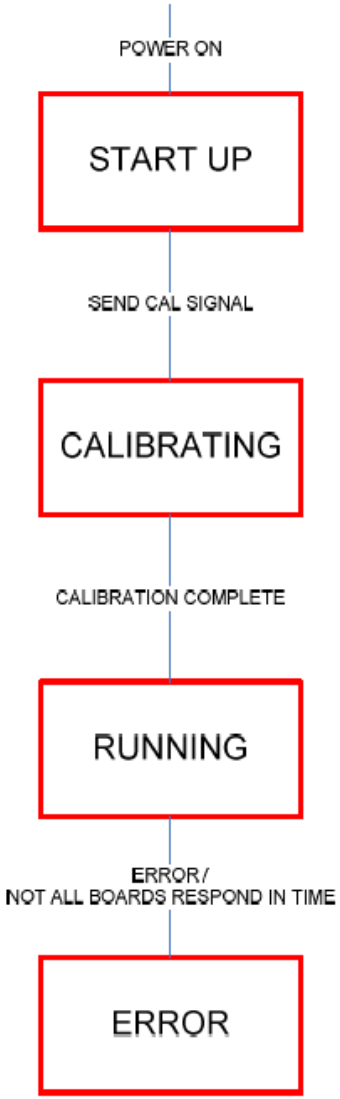
Any Questions?



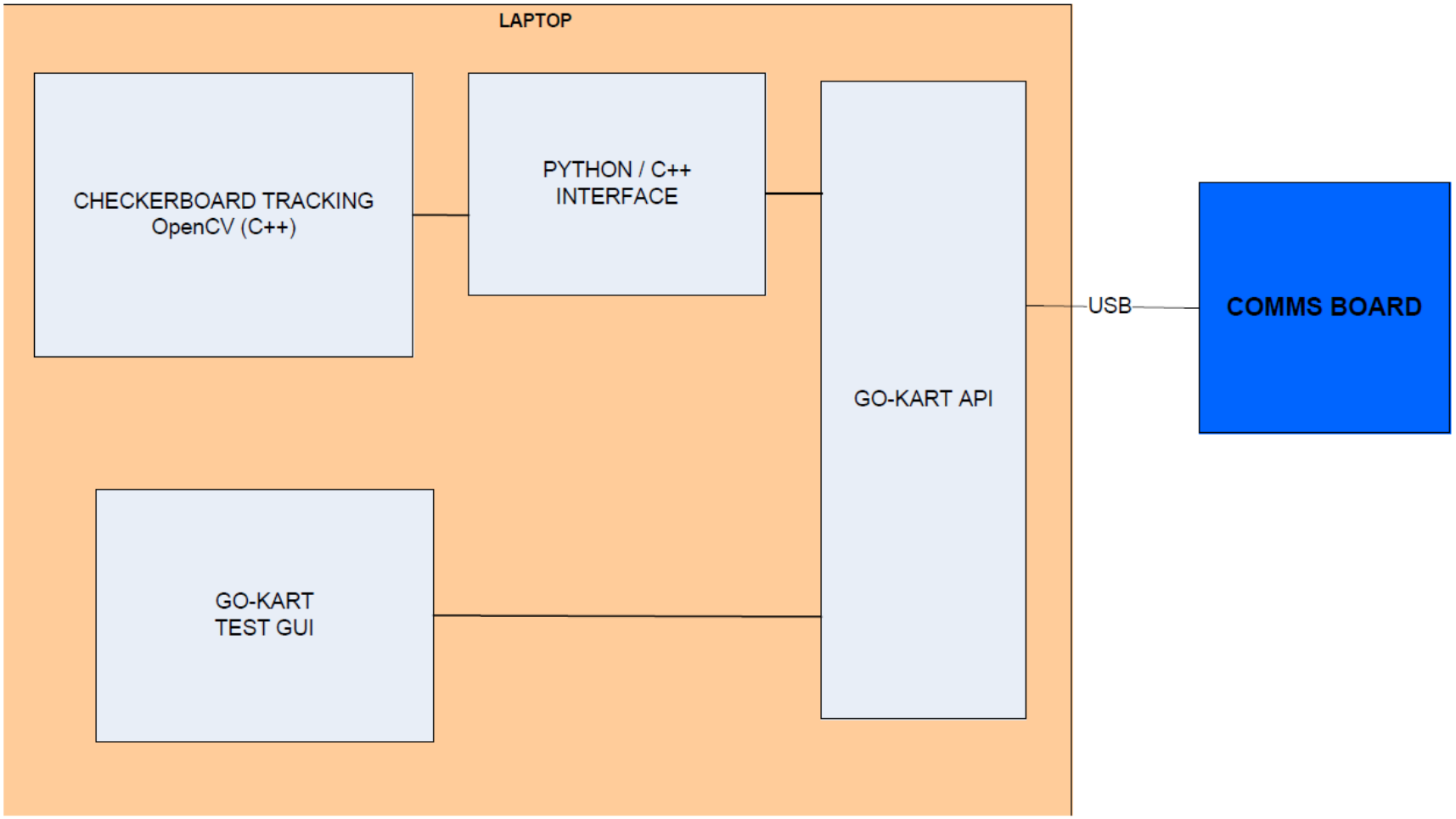
System Block Diagram



CANBus System States



Laptop Interface



Checkerboard Tracking

- Uses OpenCV computer vision library
- Finds the pose in 3D of a checkerboard of a given dimensions (number of squares)
- Can provide: rotation and translation vector – fully describes pose
- Uses
  - Thresholding
  - Intrinsic parameters of the camera
  - Planar homography techniques
- Developed separately in computer vision course