Senior Design Final Presentation

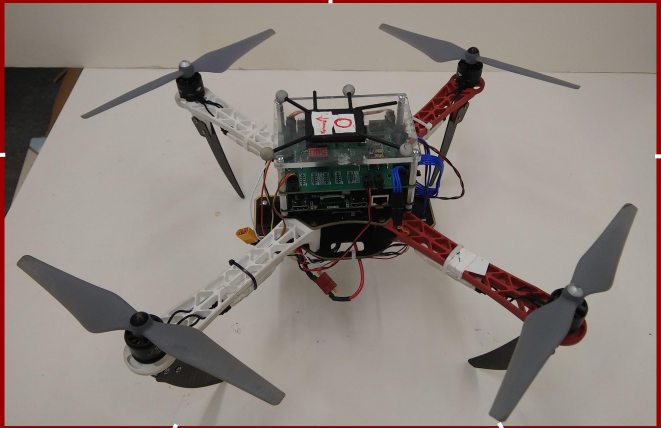
...

Team #20 - MicroCART

Concept

Demonstrate Department Capabilities

Data Analysis



Data Collection

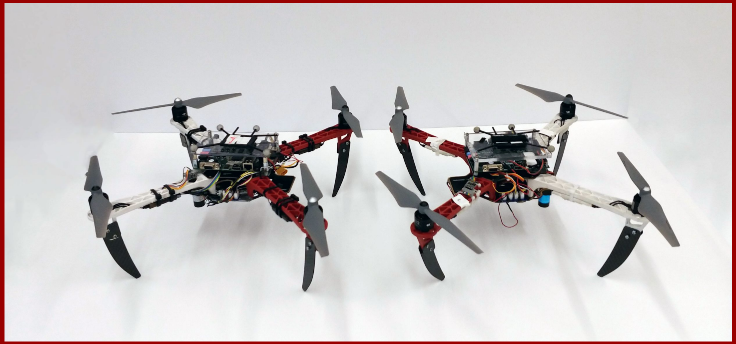
Promote Student Interest

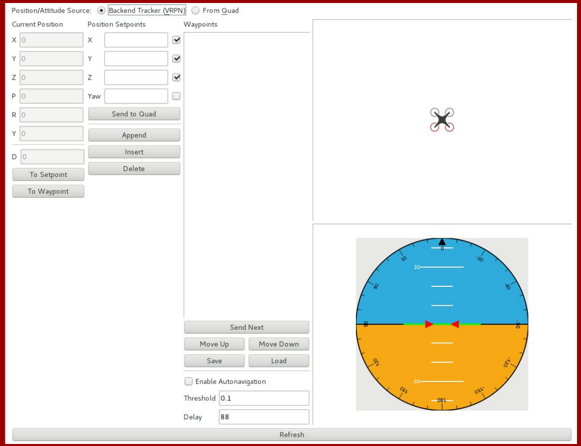
Controller Testing

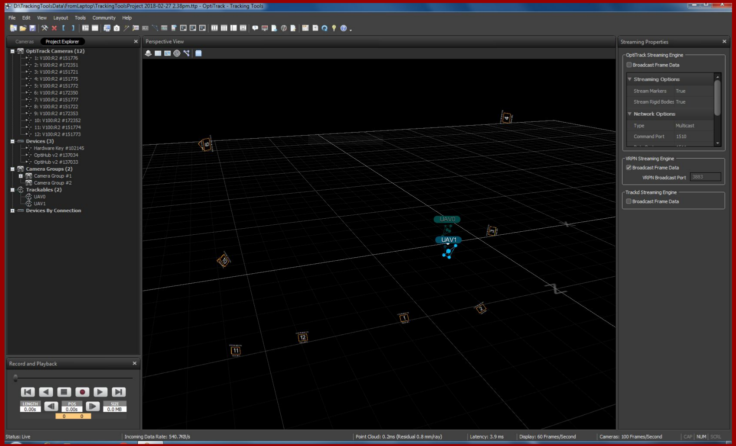
System Description

Three main project subsystems:

1. Quadcopter
2. Camera Tracking System
3. Ground Station

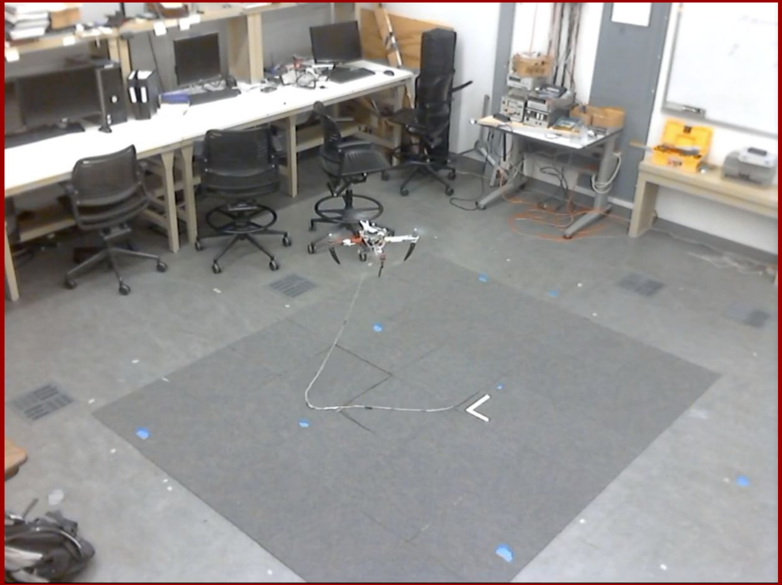




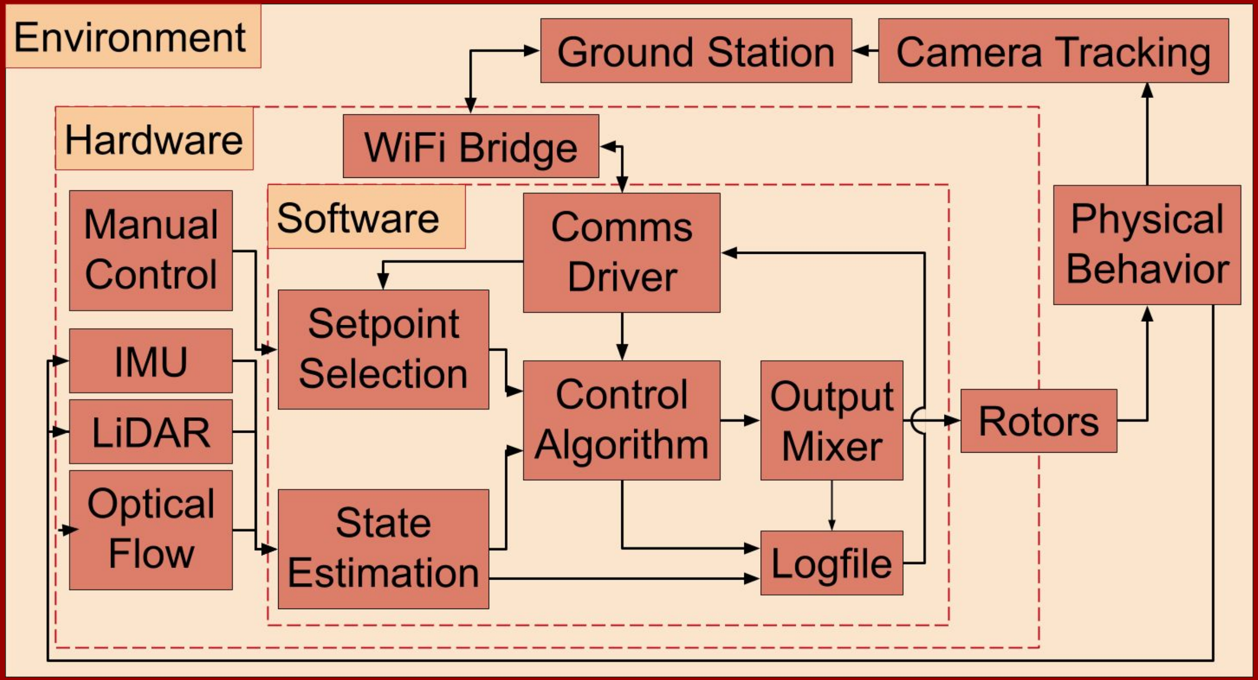


Operating Environment

- Indoor lab space in Coover 3050
- Object tracking system for position/orientation feedback
  - 12 IR Cameras
  - 100 Hz updates over local network
  - ~8 m^3 reliable tracking space

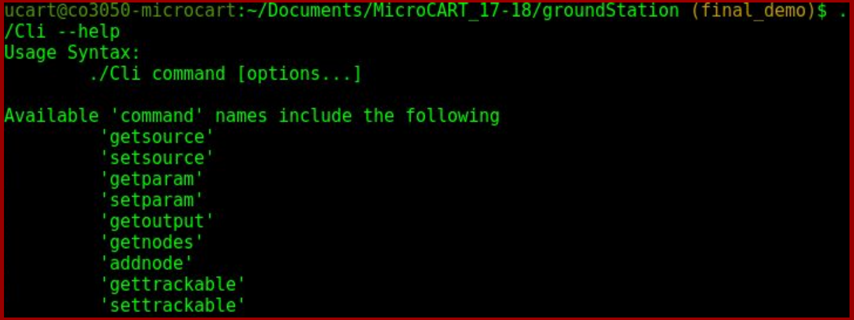


System Block Diagram

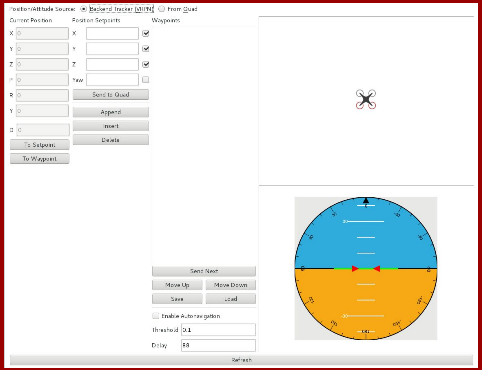


User Interface

- Command Line Interface
  - Provides scriptable interface to all backend functionality
  - Gives the user direct access to basic commands for the purposes of debugging or running basic scripts



- Graphical User Interface
  - Provides more intuitive interface for common operations
  - Allows the execution of available script files
  - Allows for the creation of graphical representations of flight data



Functional Requirements

- Quad Hardware/Software
  - Provide stable flight around a setpoint
  - Smooth autonomous movement
  - Safety Features
- Ground Station
  - Real-time data logging and visualization
  - Setpoint and Camera Feedback relay to quadcopter
- Controls
  - Autonomous and manual flight simulation using multiple controllers
  - Tools for controller verification
- Testing & Simulation
  - Automated test suite
  - Provide useful debugging tools

Non-Functional Requirements

- Quad Hardware
  - Get data from sensors within one control loop
  - Successfully control motors based on sensor data, so that flight is stable
- Quad Software
  - Real time sensor and actuator data transmission
  - Control loop should be able to operate at 200Hz or higher (<5ms)
- Ground Station
  - Must be able to process camera and quad data in no more than 10ms
- Controller Simulator
  - Produce data which can be compared to measurable values on the quad to verify correct controller behavior

Input & Output Specifications

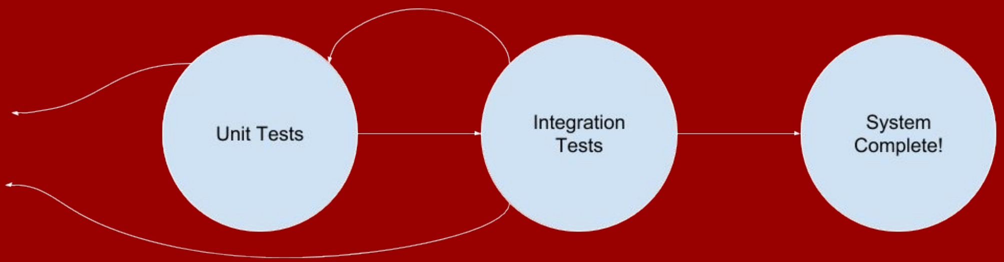
- User input at ground station
  - Script or GUI
- Commands and position data are sent to the quadcopter
  - Wifi→serial bridge with a custom messaging protocol
- Quadcopter outputs actuation to motor controllers
  - PWM @ 1 kHz
- The object tracking system observes the quad, and reports to the ground station
  - 12 IR cameras
  - VRPN Protocol over LAN

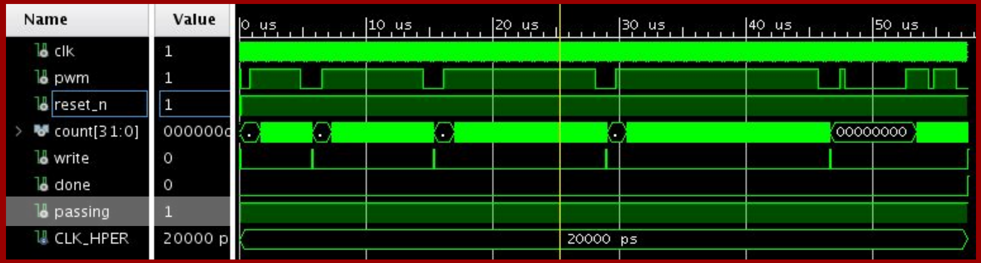
Hardware & Software Specifications

- Quad Hardware
  - 4 motors + controllers
  - Zybo Z7 control board + Shield PCB
    - FPGA: 4 PWM outputs, 6 PWM inputs, 2 I2C busses, 1 UART bus
  - MPU-9250 Inertial Measurement Unit, ESP8266 Wi-Fi bridge, LIDAR unit, Optical Flow Sensor
  - Power supply from 11.1 V battery to 5V for Zybo and sensors
- Quad Software
  - Baremetal control loop with data logging
    - 200 Hz, including sensors and output
  - Communication with ground station (<5ms latency)
- Ground Station
  - GUI created in C++ utilizing QtCreator
  - Backend, Frontend and Cli created in C

Testing Plan

- Hardware Tests
  - Unit Tests: Automatic
  - Integration Tests: Manual
- Software Tests
  - Automatic
  - GitLab CI run on virtual machine
  - Regression testing using virtual quad software
- Integration Tests
  - Manual (flights)
  - Testing Rigs (constrained flights)
- Debugging/Diagnostic Tools
  - Real Time Sensor Transmission



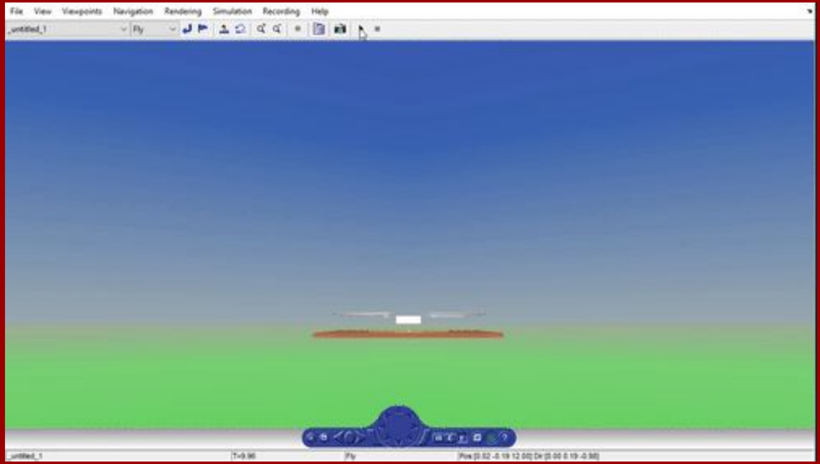


Project Additions

- Vivado Upgrade on Quad Zybo boards
- Custom PCB on Quad
- Real-time Ground Station Data Logging
- Controller Simulation
- Testing Mounts

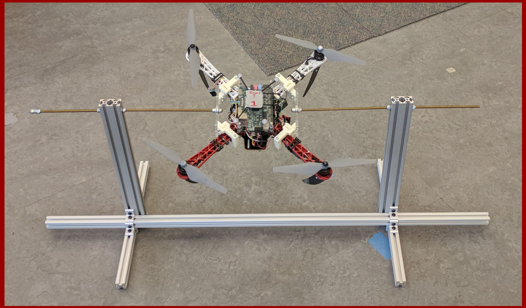
Simulations

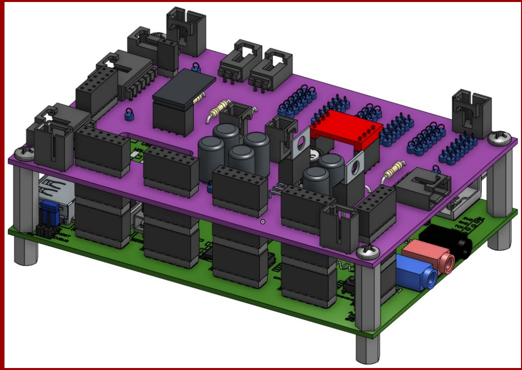
- Control Algorithms
  - Quad currently uses a PID controller
  - Last year’s team developed an LQR controller which has not yet been tested on the quad
- Both controllers can be simulated in Simulink under manual and autonomous flight modes
- Future controllers can also be tested in Simulink before being implemented on the quad
  - Reduces risk of physical damage from failed flights



Controller Characterization & Verification

- Testing mounts can be used to restrain the quad along 5 of its 6 degrees of freedom
  - Yaw, roll, and pitch can currently be isolated and measured to characterize the quad's response to inputs along a single degree of freedom
- Motor PWM signals can be read from test points on the Zybo PCB shield and compared to simulated PWM signals generated by the Simulink simulator
  - Allows unverified controllers to be tested without requiring flight tests





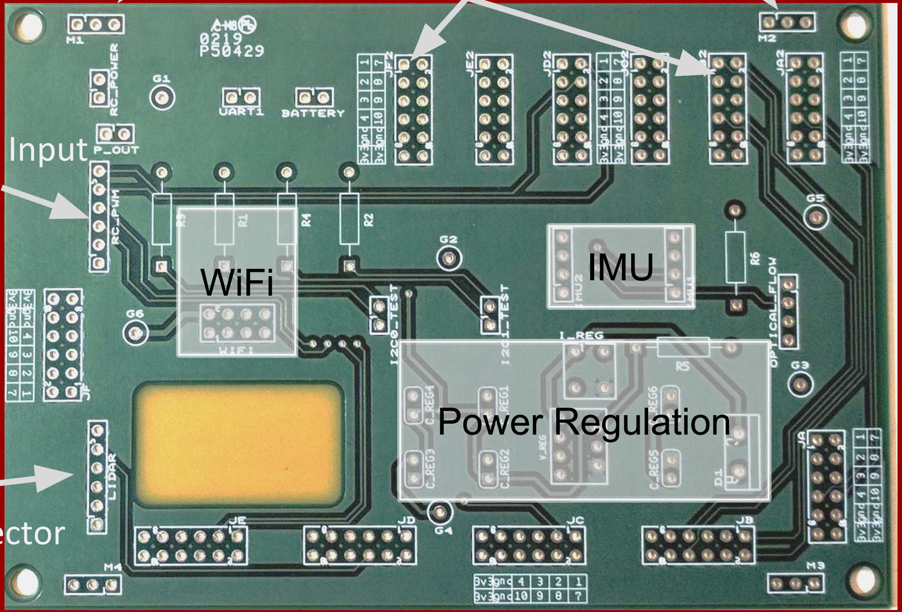
Custom PCB Shield

Design

- Created using MultiSim and UltiBoard
- Includes designated test points for all major signals (Motor PWM, I2C, RC inputs, ect.) to allow debugging while the quad is in operation

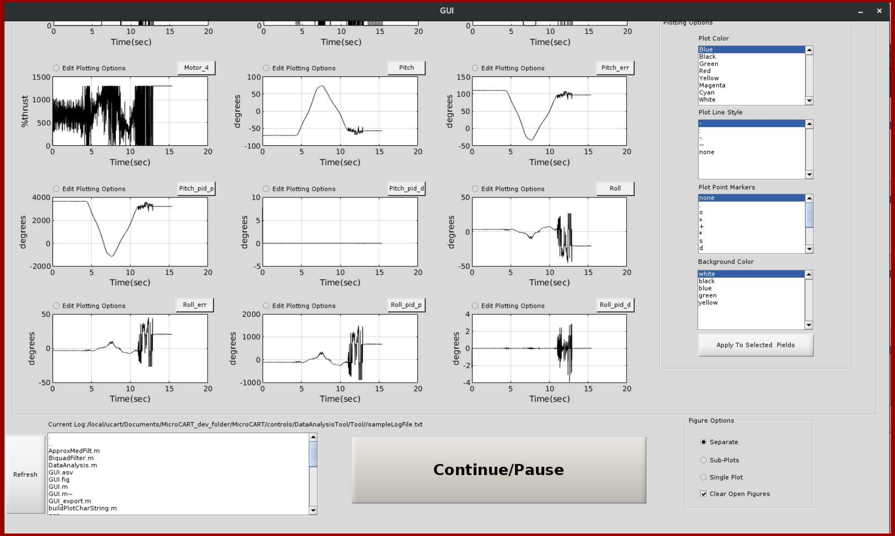
Integration

- Continuity checks ensured proper connections were made on the PCB
- Voltage checks and brief motor tests ensured that the voltage regulator was functional and all components were interfacing correctly



Real-Time Data Logging

- Allows for flight data from the quad (motor speeds, position, velocity, acceleration, etc.) to be transmitted to the ground station during flight time. The ground station can then display this data to the user as it comes in.
- This increases the viability of MicroCART as a research platform as users can now monitor flight-specific data coming from the quad during flight tests.



Questions?

Quad Software Flow