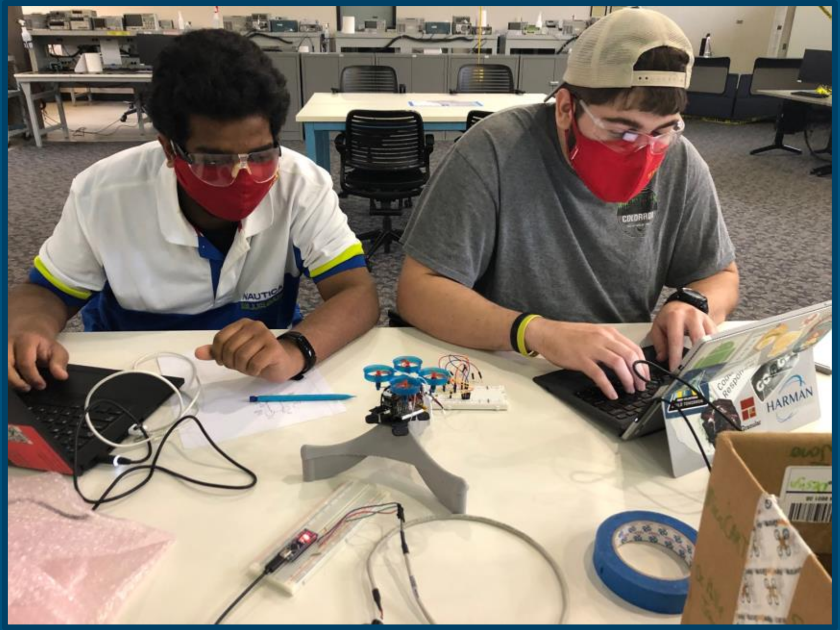
MicroCART
(Microprocessor Controlled Aerial Robotics Team)

sdmay21-27

Overview

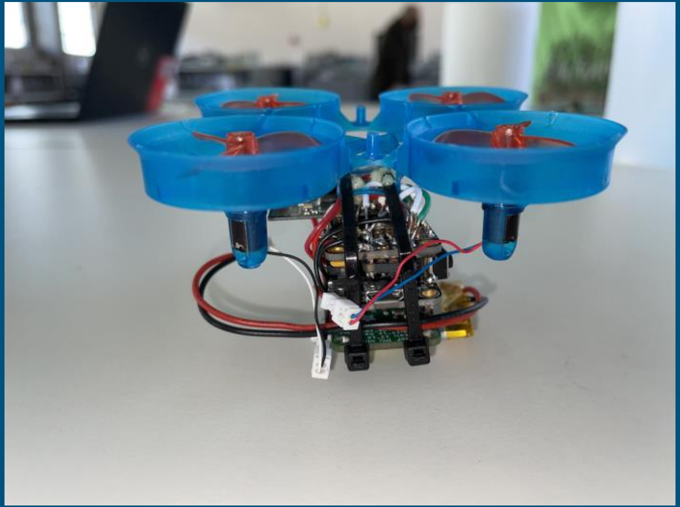
- CPRE 488 Labs
- Programmable Drone
- Test Control Algorithms
- Analyze Rotational Data



Students desperately trying to fix their code in lab

Requirements

- Programmable quadcopter (≤ 4.5 x 4.5 in)
- Re-programmable drone
- RF & Wifi Communication
- Testing station - Rotational data
- Ground Control program (WiFi)
- Cost Goal: ~$50 for the drone
- MCU: HW floating-point multiplication/division support

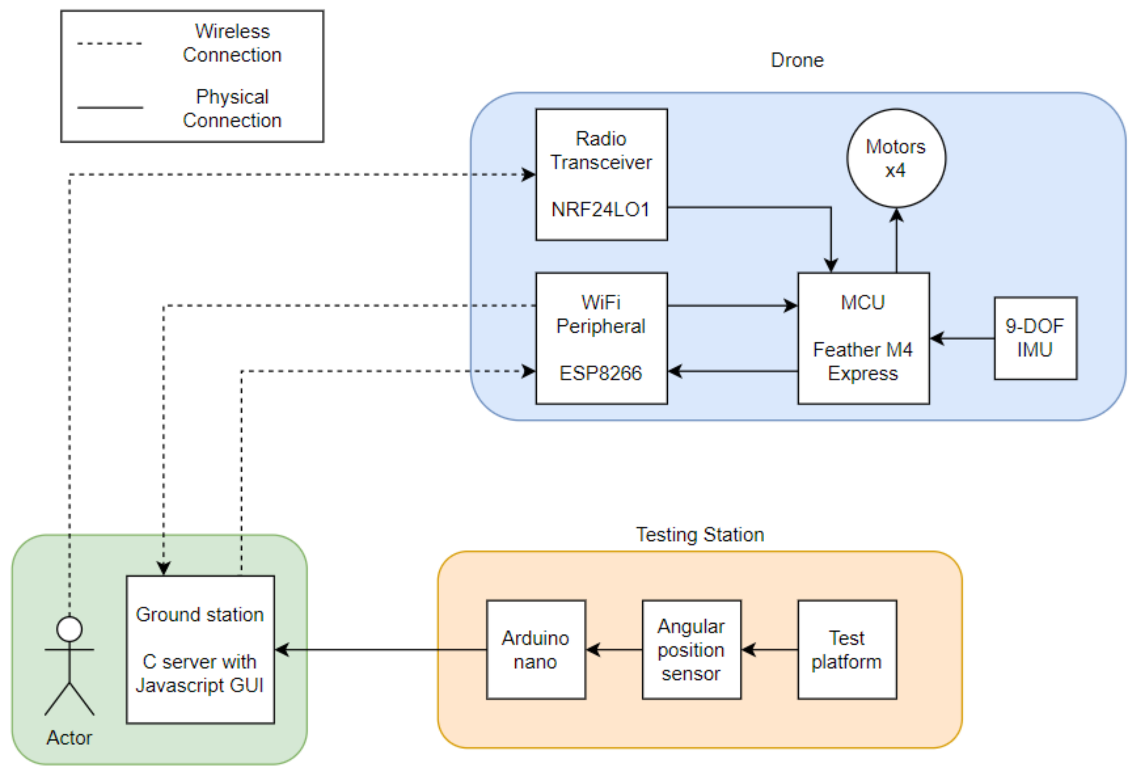


Getting Started

- Project needs
- Team composition
- Initial research
- Organization
- Planning

https://store.bitcraze.io/products/crazyflie-2-1

Block Diagram



Component:
Drone Hardware

Drone Hardware - Design

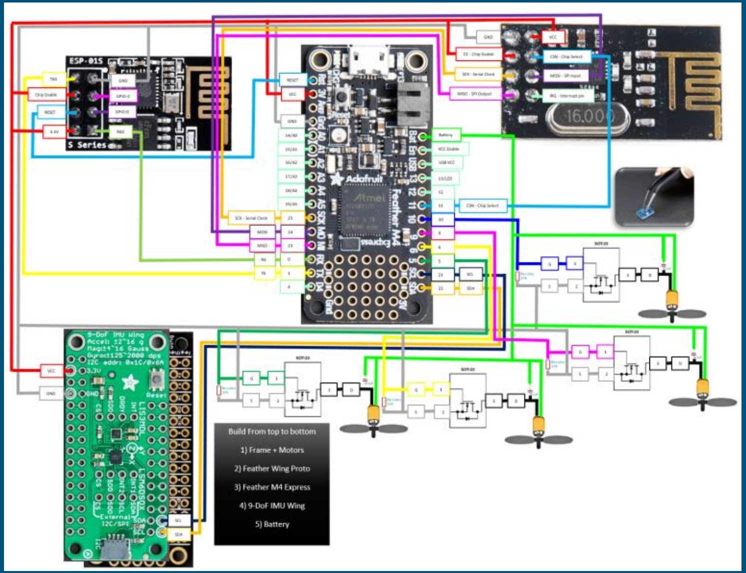
- 3 layers of modules/component on top of each other
- (From Top to Bottom) Frame + Motors, Feather Wing Proto, Feather M4 Express, 9-DoF IMU Wing &, Battery

Bill Of Materials:

- Microcontroller (Adafruit Feather M4 Express)
- Accelerometer + Gyro (9-DoF IMU Wing)
- WiFi Transceiver & Receiver (ESP8266 ESP-01)
- RF Transceiver & Receiver (NRF24L01+ 2.4GHz transceiver module)
- Feather Wing Proto
- Four 30 mm Propellers
- Four 6mm Brush Motor @KV: 19,700
- Four SOT-23 with Mosfets
- Four Mosfets N-CH 8V 6A
- Four 0.1 uf Capacitors
- Four 10k Resistors
- Four Schottky Diodes
- Two Frames (NewBeeDrone Cockroach Brushed Super-Durable Frame 65mm)
- 36 30 mm Long Connector Pins
- Lithium Battery at 3.7V and 2.5 AH

Drone Hardware - Implementation

- Research on the components of the drone
- Develop a pin assignment for the MCU
- Create Wiring schematic
- Build a prototype on breadboard
- Build a prototype using protoboards
- Design a PCB to replace the protoboard (Incomplete)

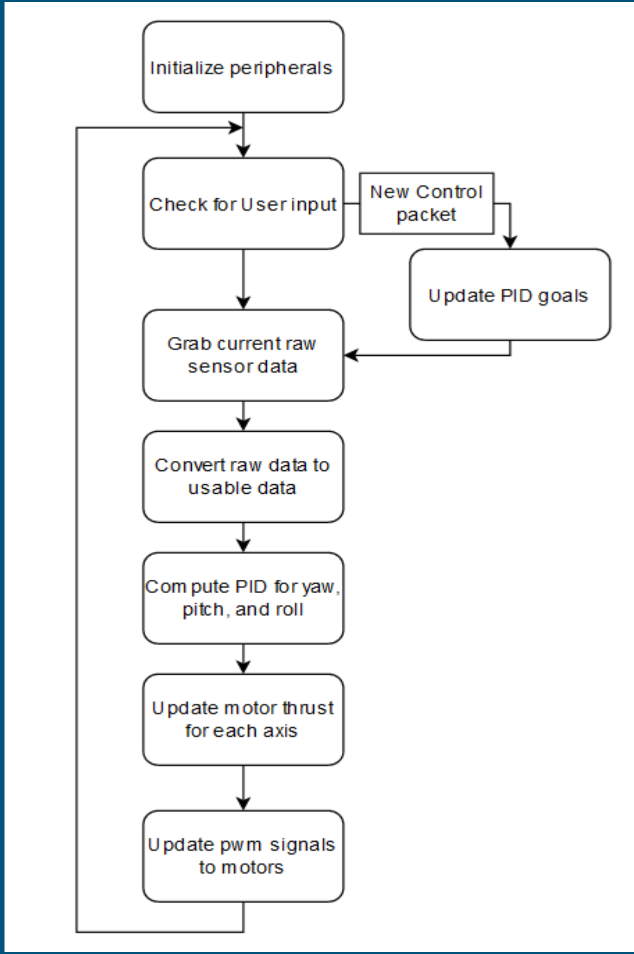


Component:
Drone Software

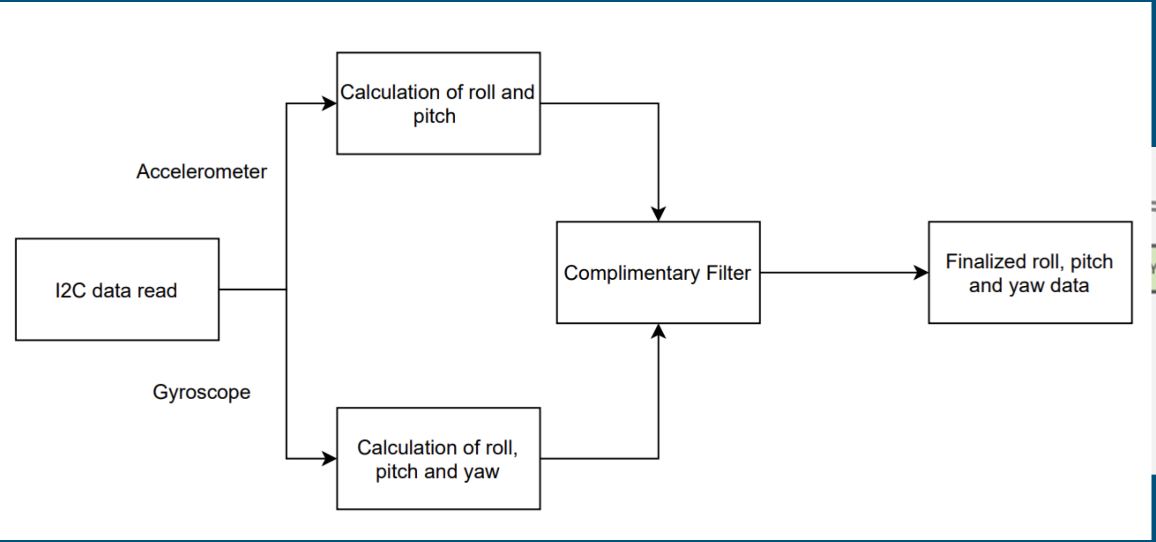
Drone Software - Design

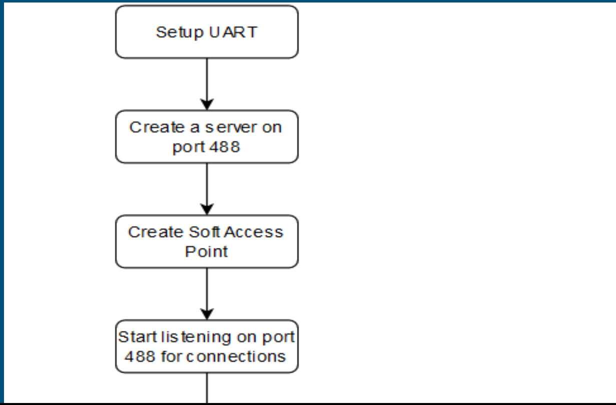
Design goals:

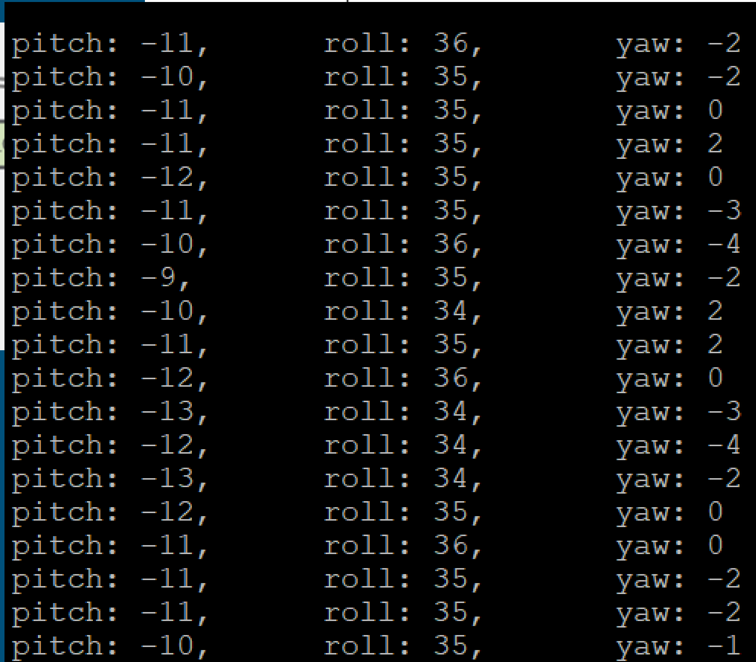
- Create modular code
- Generate PWM signals
- Use USART peripherals
- Use I2C peripherals
- Create PID control loop



Drone Software - Implementation

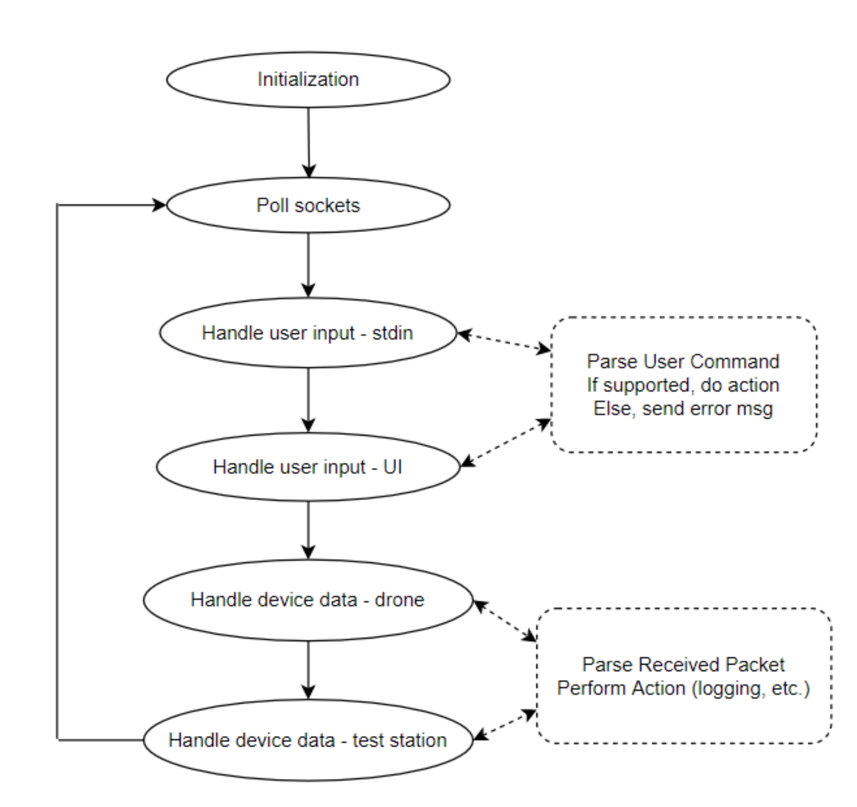






Component: Ground Control

Ground Control (C) - Design

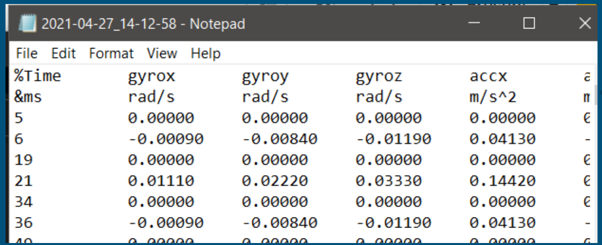


Command	Tag1	Tag2	Arg1	Example
connect	-d	-t	IP address	"connect -d 127.0.0.1"
disconnect	-d	-t		"disconnect -t"
log	-s	-q	File name (optional)	"log -s log.txt" "log -q"
ui				
status				
help				
quit				
custom	-d	-t	msg_string	"custom -d hi"
raw	-d	-t	msg_string	"raw -t hello"

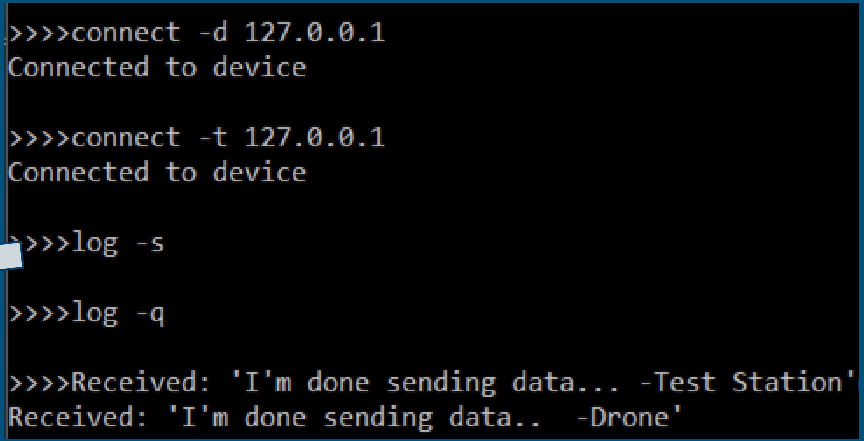
Field:	Start Char	Msg Type	Msg ID	Data Length	Data	Checksum
Bytes:	1	2	2	2	x	1

Ground Control (C) - Implementation

- Device connections
- Socket communication management
- User commands
- UI support
- Data packets
- Logging



%Time &ms	gyrox rad/s	gyroy rad/s	gyroz rad/s	accx m/s^2
5	0.00000	0.00000	0.00000	0.00000
6	-0.00090	-0.00840	-0.01190	0.04130
19	0.00000	0.00000	0.00000	0.00000
21	0.01110	0.02220	0.03330	0.14420
34	0.00000	0.00000	0.00000	0.00000
36	-0.00090	-0.00840	-0.01190	0.04130

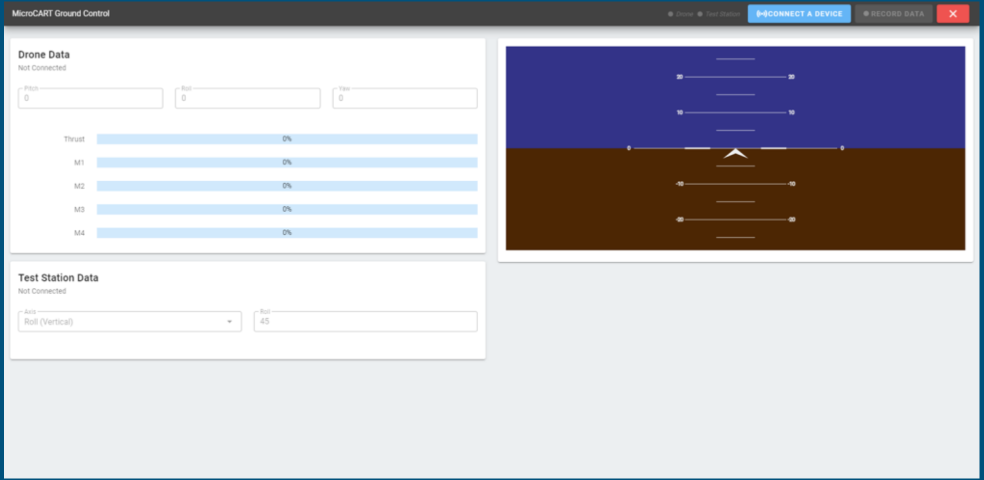


Ground Control (UI) - Design

- Dashboard UI
- Navbar manages additional features
    - Connections
    - Logging
- Is built alongside communication server

Ground Control (UI) - Implementation

- Dashboard
  - Attitude Indicator
  - Rotational values
  - Thrust values
  - Test Station value
- Connections
  - Can connect to drone and test station by IP address
- Logging
  - Displays all logged files
  - Can open files in MatLab

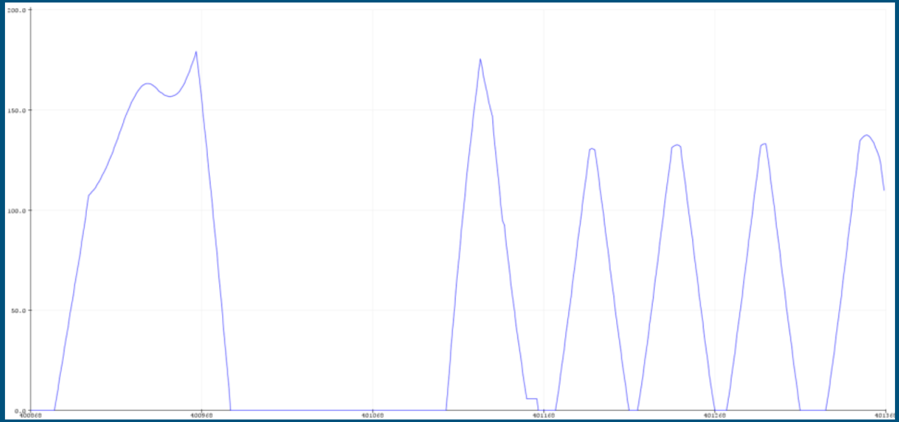


Component:
Test Station

Test Station - Design

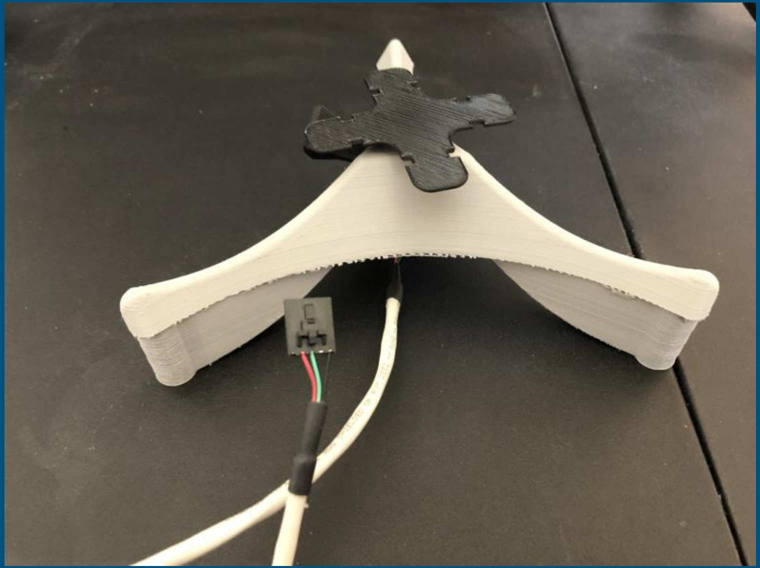
Design of the test station must achieve the following goals:

- Securement of a drone to a test platform
- Measurement of drone rotation in all three axes (Roll, Pitch, Yaw)
- Send these measurements to the ground station for interpretation
- Cheap to develop (3D Printing)



Test Station - Implementation

- Multiple platforms
- Raised base station
- Magnetic shaft encoder
- Arduino for ground station communication



Video Demo

The Experience

Challenges

Technical:

- Drone power distribution
- Drone component physical wiring
- Compiling for embedded devices
- Programming WiFi Chip
- Test Station 3D printing
- JavaScript and C communication

Non-technical:

- Getting started
- Task management
- Organization
- Email Communication

And of course, COVID restrictions

Lessons Learned

- Unified team vision
- Goal setting
- Following Agile process
- Get constant feedback

Source: https://www.quickbase.com/blog/how-to-make-scheduling-meetings-easier-and-more-productive

Conclusion

- Weaknesses + Improvements
- Preview of professional long term projects
- Client/Team communication
- Collaborating during COVID is difficult

Team


Alex Bjerke
Project Manager
Amith Kopparapu Venkata Boja
Embedded Software Lead
Alfonso Raymundo
PCD Design Lead
Grayson Goss
Technical / CAD Design Lead
Hannah Aisya Mohamad
Webmaster
Russ Paulsen
Test Station Assistant
Theodore Davis
Embedded Hardware Lead
Trent Woodhouse
High Level Software Lead

Questions?