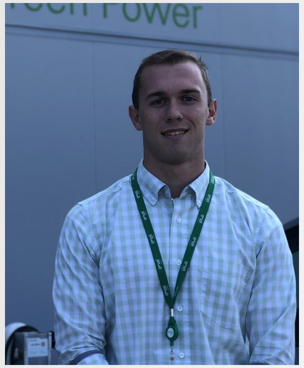
GyroGlove
Cumulative Design Review

Senior Design Project ‘21
Team #21

University *of* Massachusetts Amherst BE REVOLUTIONARY™

Meet the team



Bradley Spillert
CSE

Jacob Moynihan
EE

Son Pham
EE

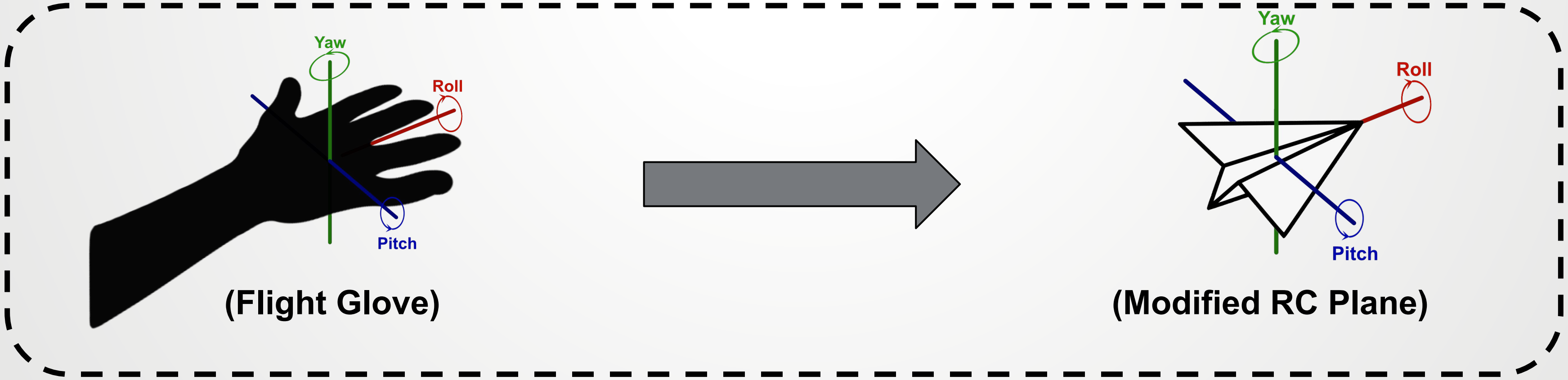
Professor Do-Hoon Kwon
Team Advisor

Problem Statement

Having a reliable flight control system is paramount for the success of any manned aircraft. While the current convention of using control mechanisms akin to analog sticks and steering wheels has proven to be timelessly effective, such mechanisms can often lack one valuable prospect: The immersion factor.

Problem Statement (cont'd)

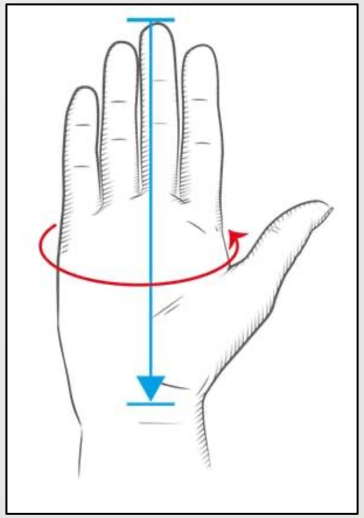
This is where GyroGlove comes into play. GyroGlove is a modern alternative to the classical stick-lever-wheel approach to flight control, ultimately allowing the pilot to control the aircraft simply by rotating their hand (orientation control) and curling their fingers (thrust control). These systems work elegantly in tandem to provide an immersive sensation of being "one with the aircraft", so to speak.



GyroGlove System

GyroGlove - System Specifications

- **Flight glove battery life exceeds RC plane battery life by at least 100%**
- **Aircraft rotation stays within ±5.0° of IMU rotation (±7.5° for yaw axis)**
- **IMU capable of detecting rotation between ±85.0° along each axis**
- **Flex sensor capable of generating dynamic output over 90.0° flex range**
- **Functional range of at least 100 meters**
- **Control latency of 20 milliseconds or less within the functional range**
- **Operating frequency of 2.4GHz**
- **Compliance with all FCC standards and regulations**



CDR Deliverables (as intended post-MDR)

1. Implementation of Kalman Filter & calibration mechanism (Bradley)
2. Custom PCB schematic design via Altium/KiCad (Jacob)*
3. Custom PCB construction via Altium (Son)*
4. Realization of Arduino Nano code on PCB-mounted MCUs (Bradley)
5. Implementation of finalized glove layout/assembly (Jacob & Son)
6. Implementation of finalized RC plane modifications/assembly (Jacob & Son)

*Jacob will design PCB schematics, Son will realize component layout & coordinate with fab-house.

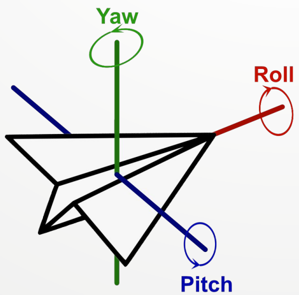
Subsystem Overview

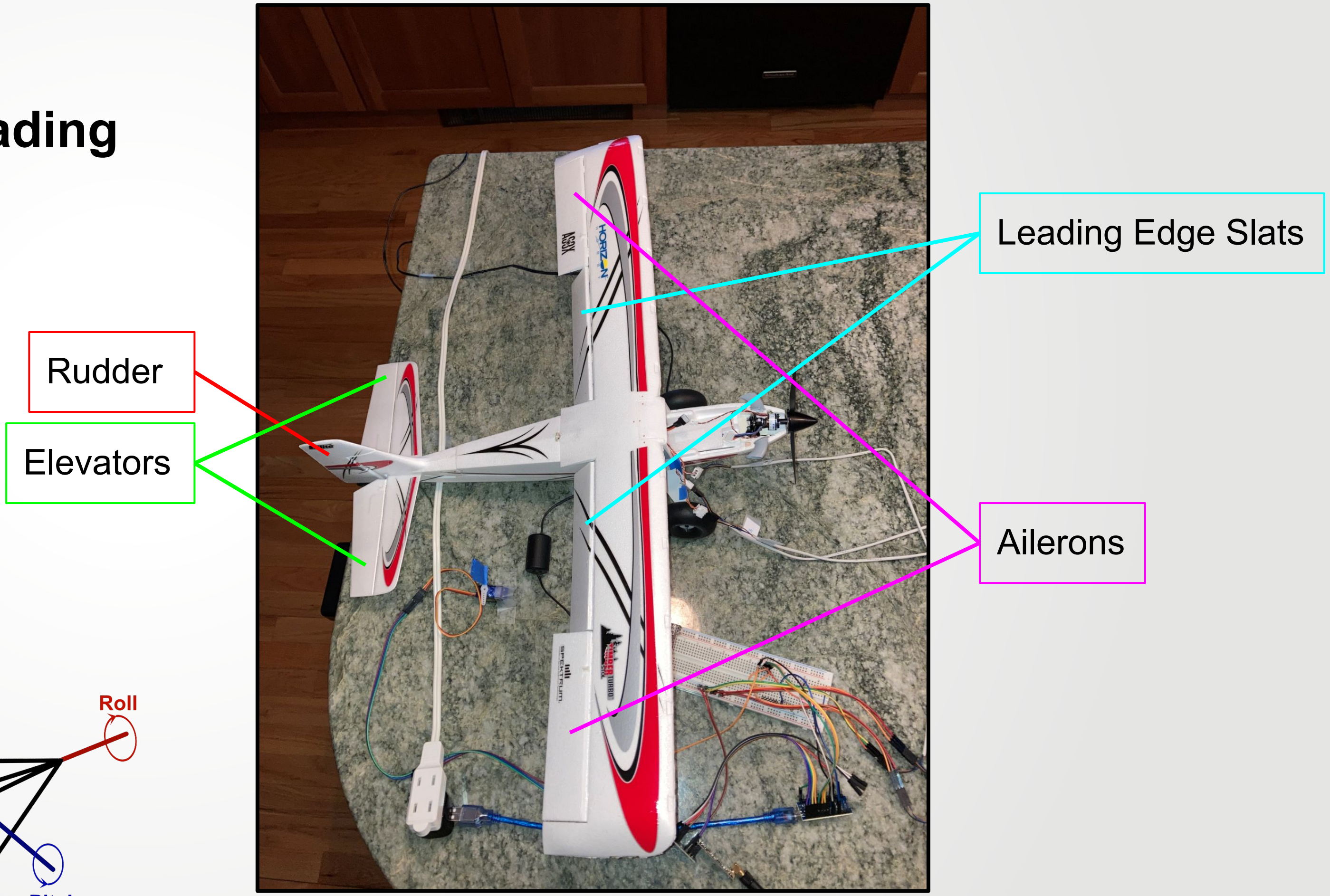
1. **Flight Glove: Captures/interprets movement of the pilot’s hand and encodes data for TX to RC plane**
   1.1. Rotational movement capturing: Using hand-mounted IMU
   1.2. Thrust control capturing: Using 3” flex sensor mounted along middle finger
2. **RC Plane: Receives encoded actuation data from Flight Glove and animates existing RC plane actuation hardware**
   2.1. Pitch/yaw/roll servo groups: Control the RC plane’s rotational orientation, as defined by glove IMU movement
   2.2. Engine/ESC: Controls the RC plane thrust, as defined by glove flex sensor tension

Axes of Movement & Actuation Hardware

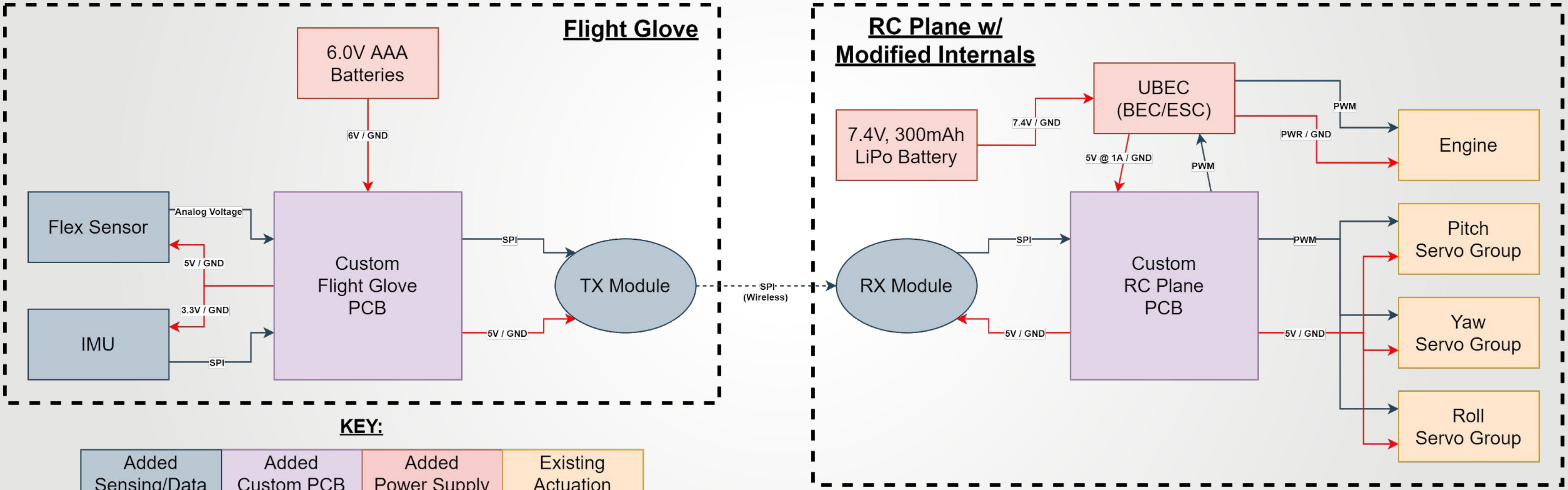
- **Pitch (controlled by elevators, leading edge slats)**
  - Actuation: 2x servos
- **Roll (controlled by ailerons)**
  - Actuation: 2x servos
- **Yaw (controlled by rudder)**
  - Actuation: 1x servo

** # of servos is unique to our particular demo aircraft. **





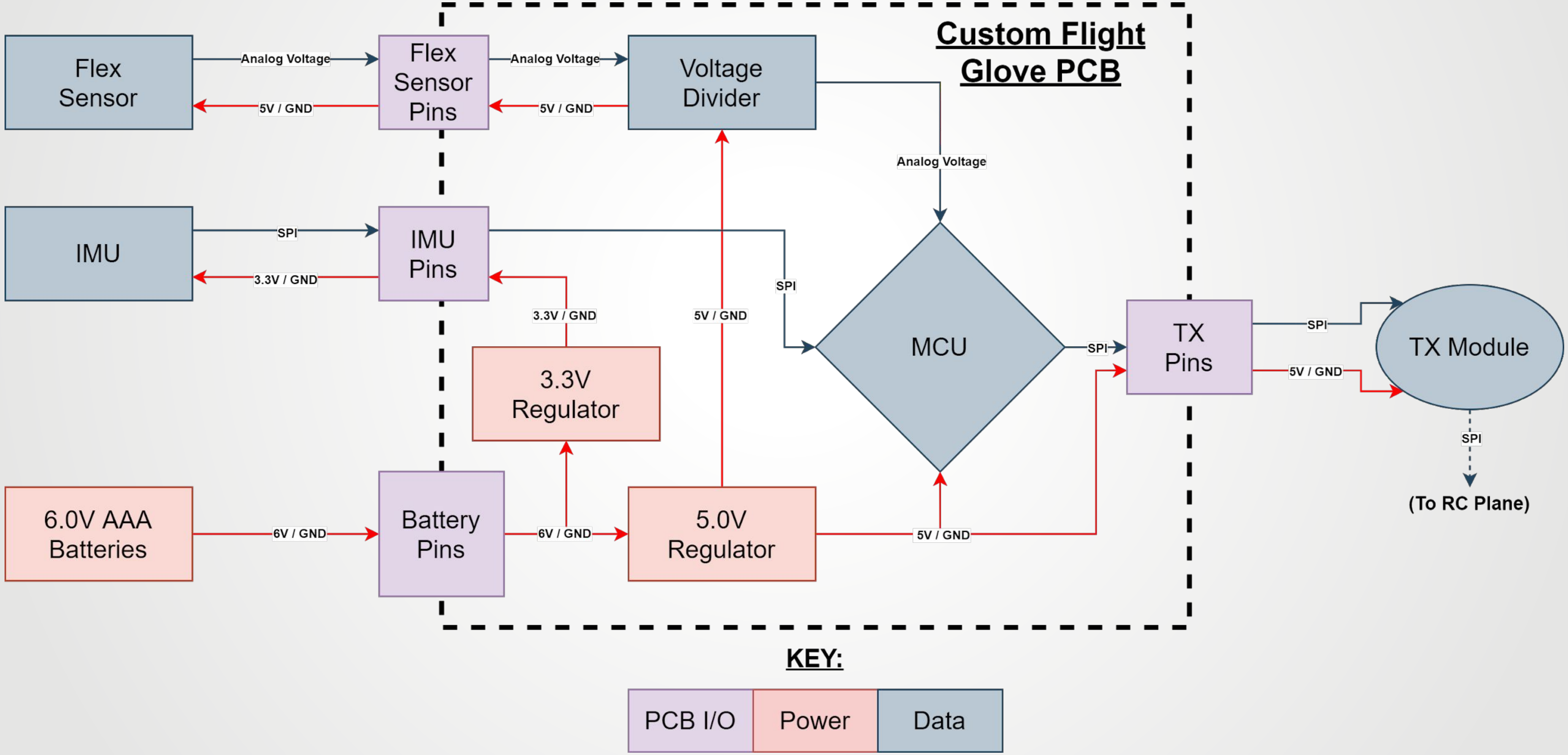
Hardware Block Diagram (Overview)



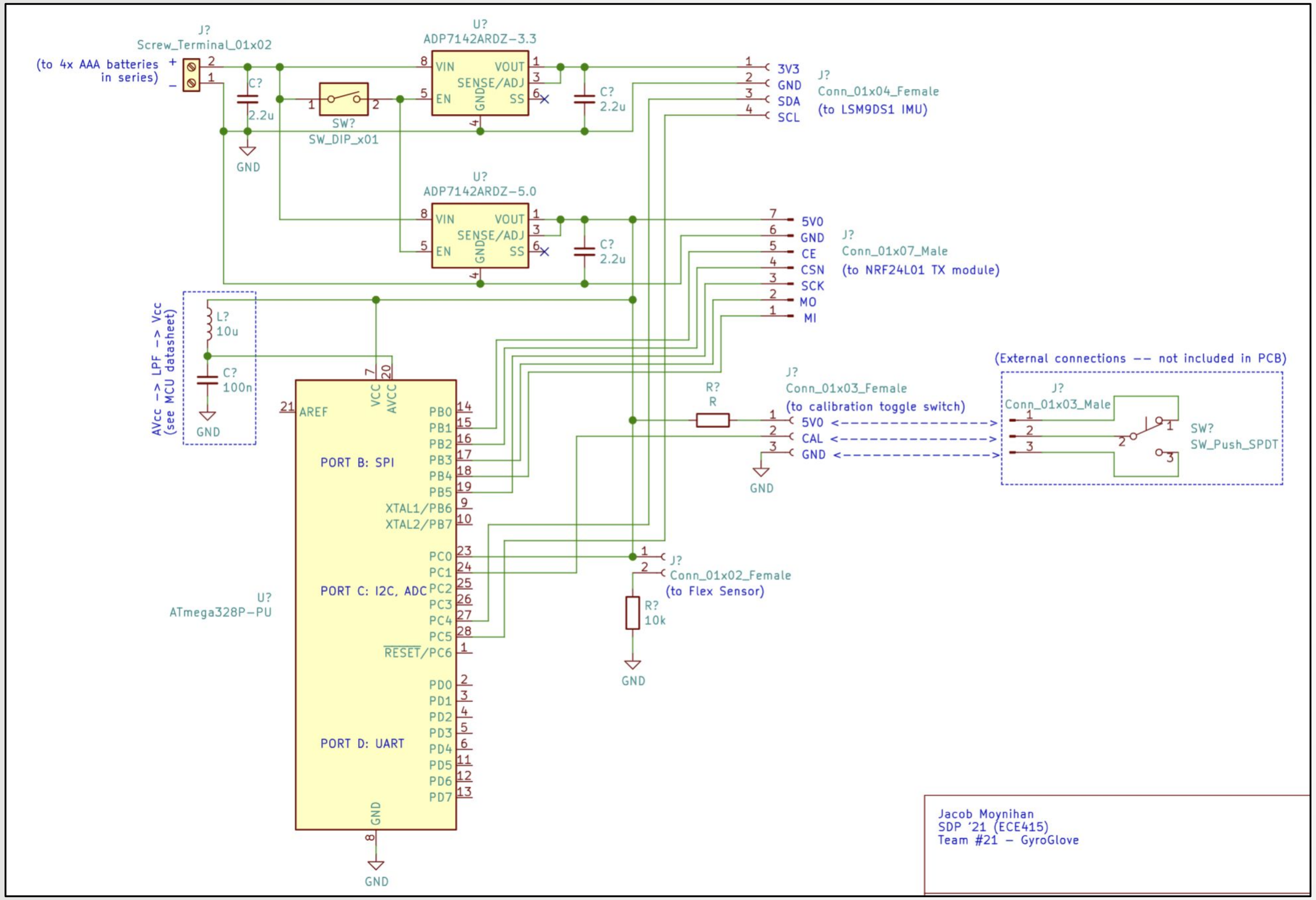
KEY:

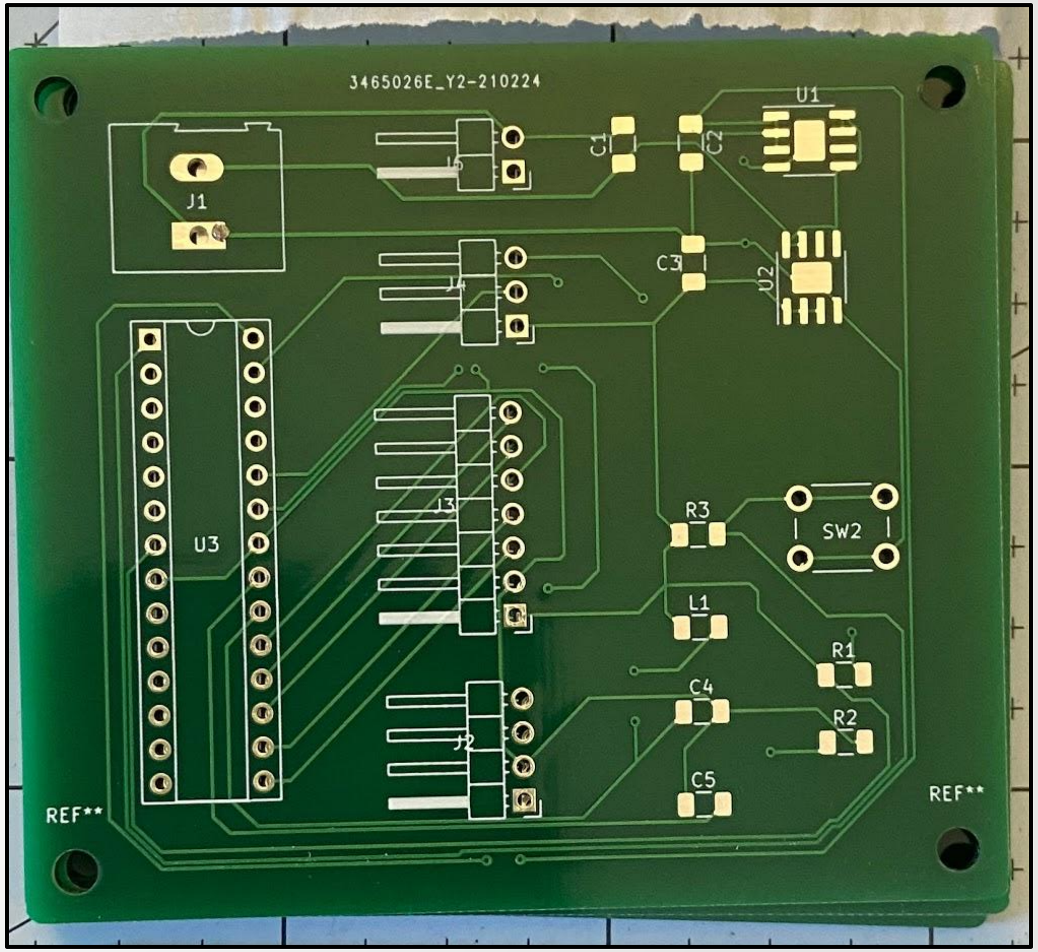
Added Sensing/Data Hardware	Added Custom PCB Hardware	Added Power Supply Hardware	Existing Actuation Hardware

Hardware Block Diagram (Flight Glove)

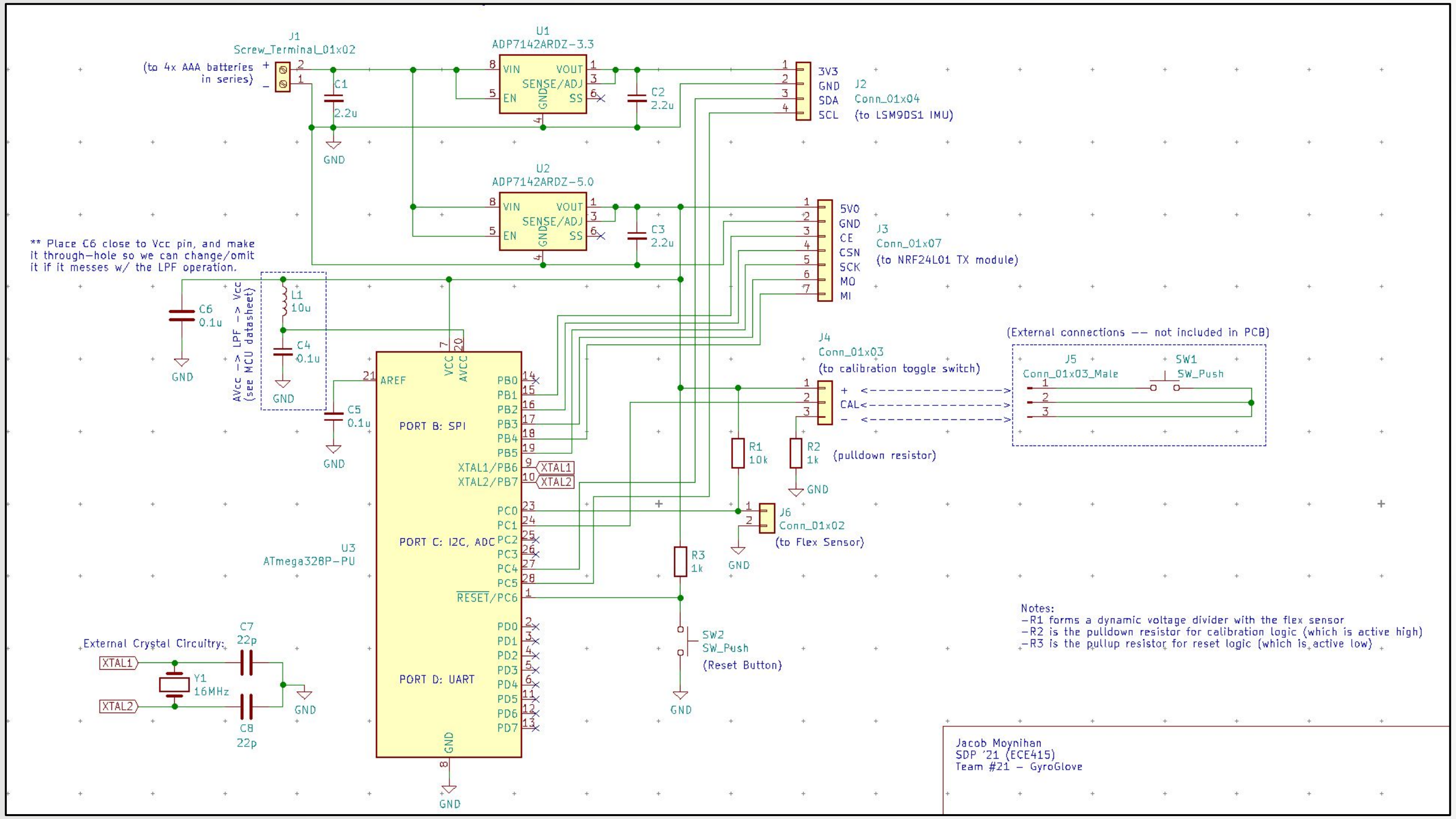


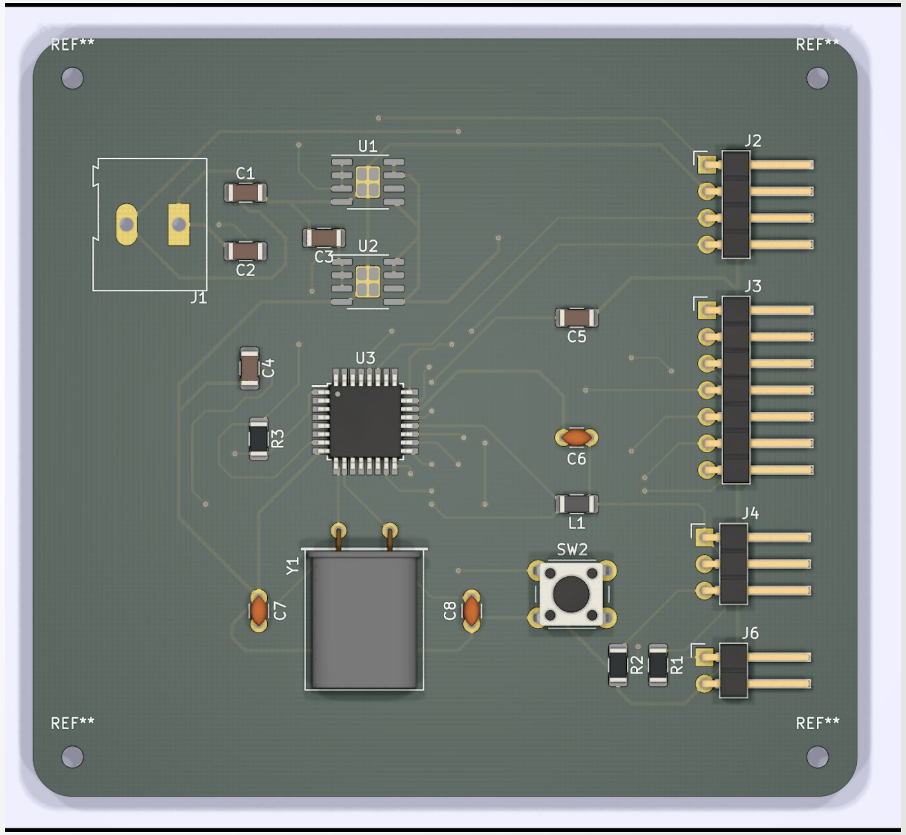
Custom TX PCB (Glove) - 1st Iteration



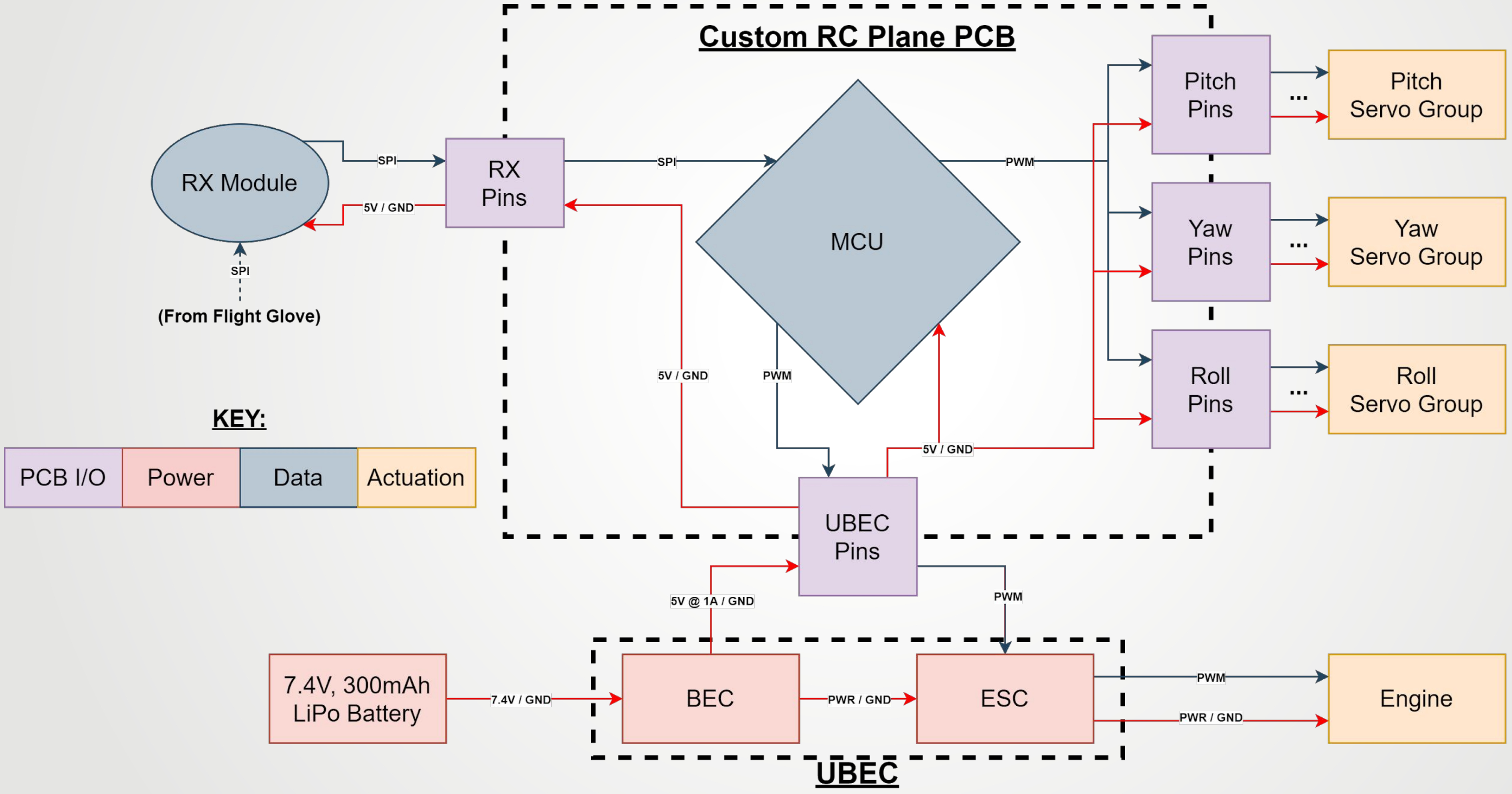


Custom TX PCB (Glove) - 2nd Iteration

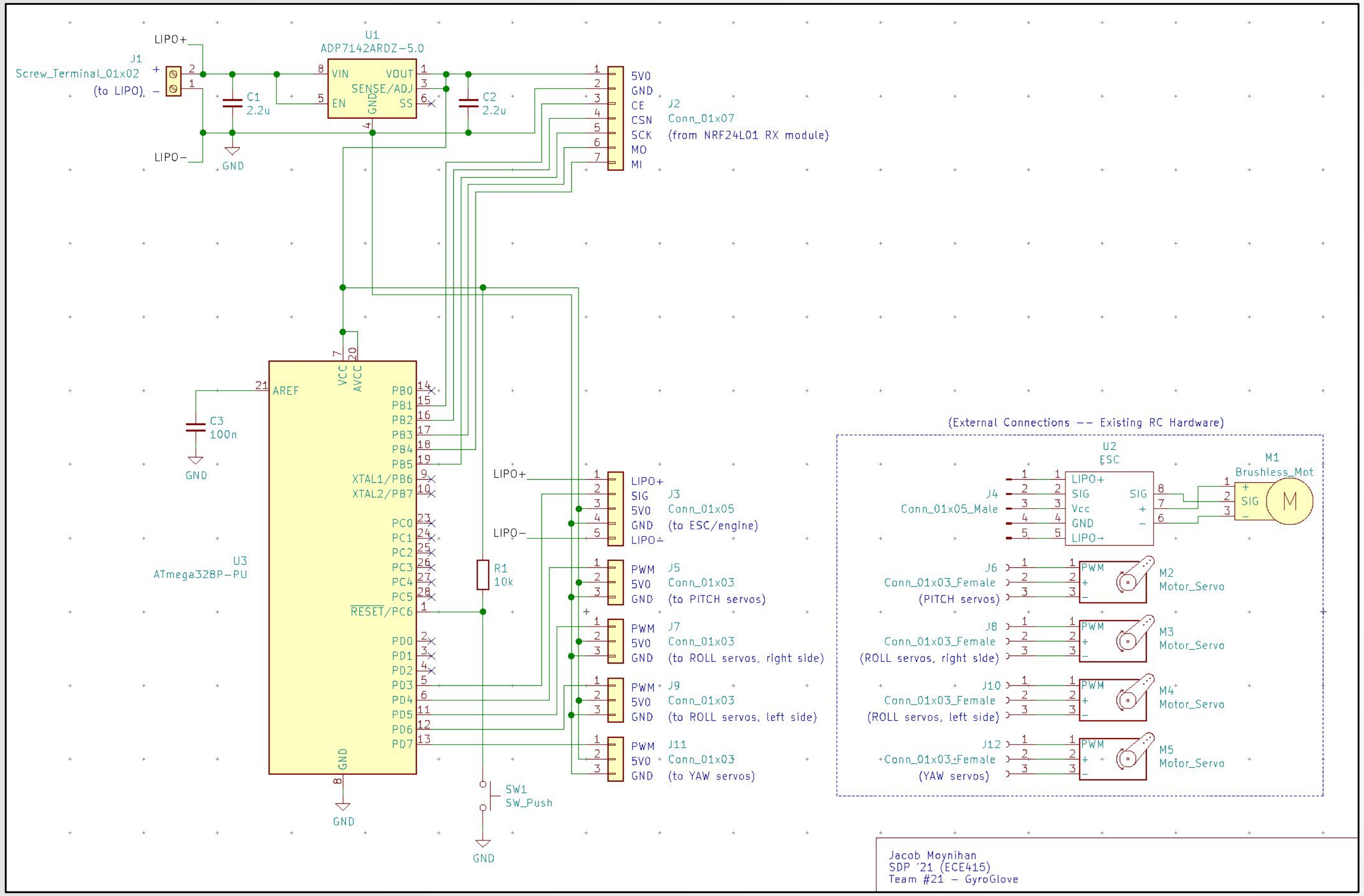


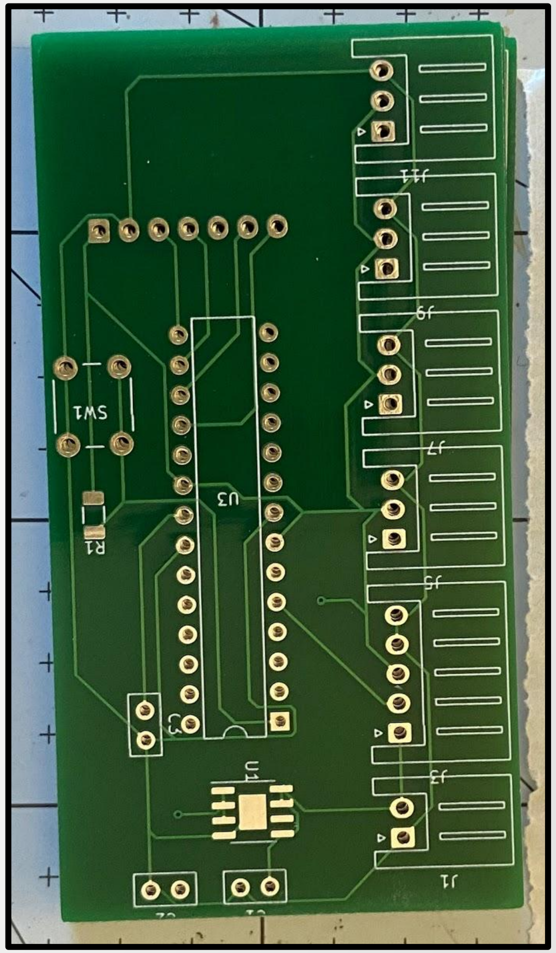


Hardware Block Diagram (Modified RC plane)

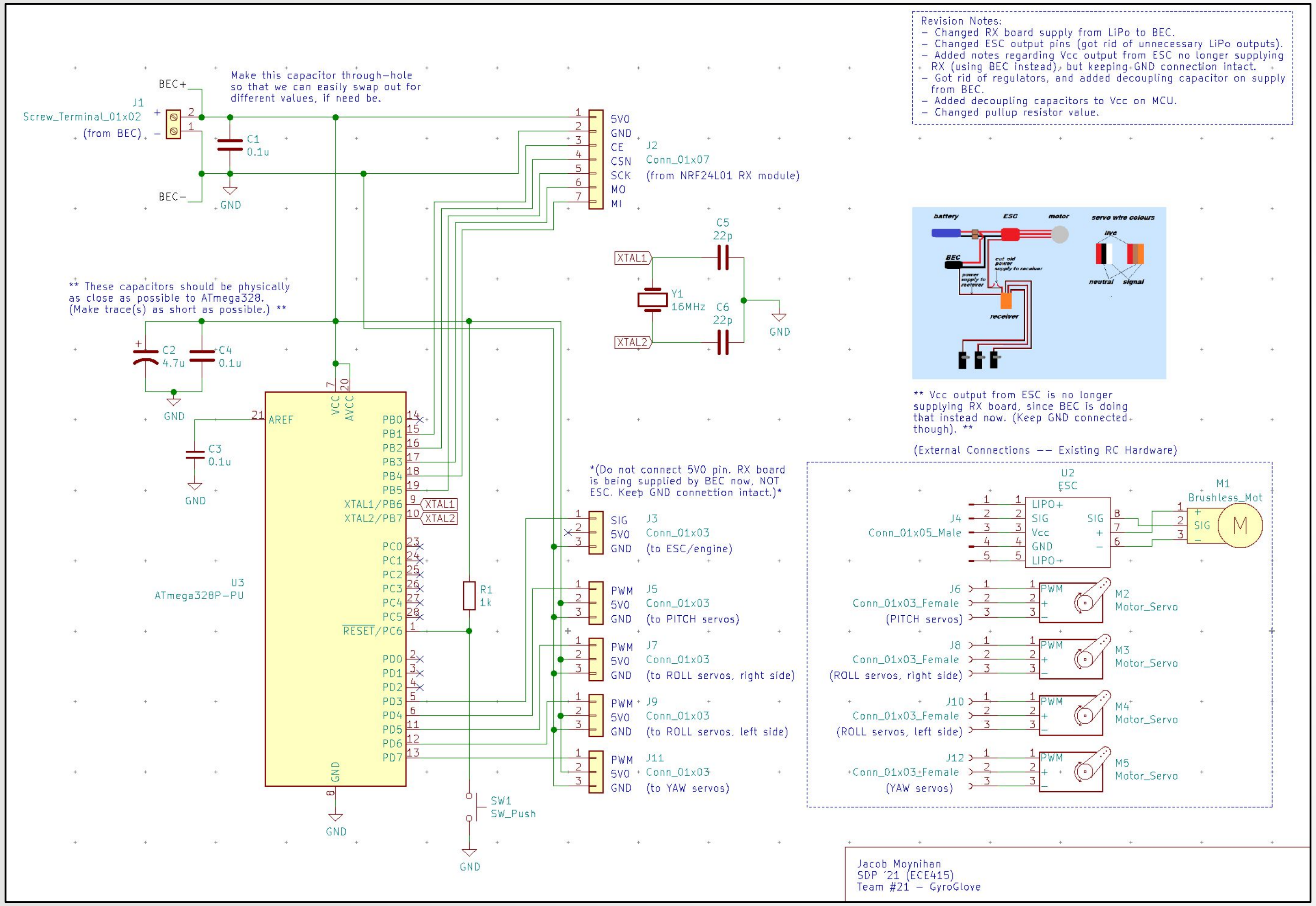


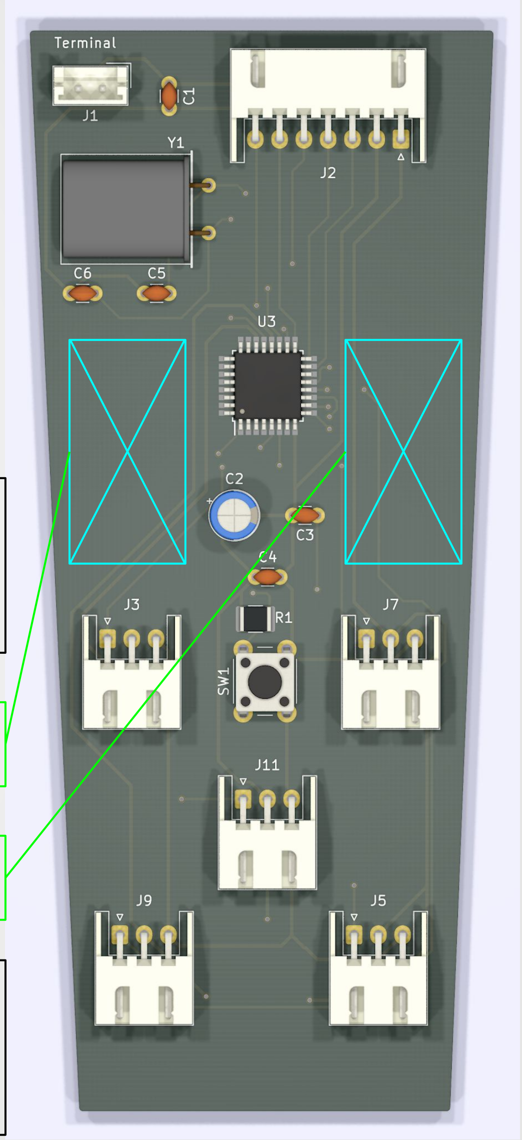
Custom RX PCB (RC plane) - 1st Iteration





Custom RX PCB (RC plane) - 2nd Iteration





Shape of the RX PCB conforms to RC plane interior.

Elevator Servo (mounted)

Yaw Servo (mounted)

**Both mounted servos actuate their respective control surfaces via mechanical cables.

LiPo Power Division & Delivery

- **We're using a "universal battery elimination circuit" (UBEC), which contains both a BEC and an ESC in one module/package**
  - The BEC (battery elimination circuit) allows us to siphon a safe amount of power from the LiPo battery to the PCB (5.0V @ 1-2A), while still allowing sufficient power through to the ESC/engine
  - The ESC (electronic speed controller) allows us to control the main engine by means of a low-power PWM signal sent from the MCU

**UBEC will be equipped with discharge protection.

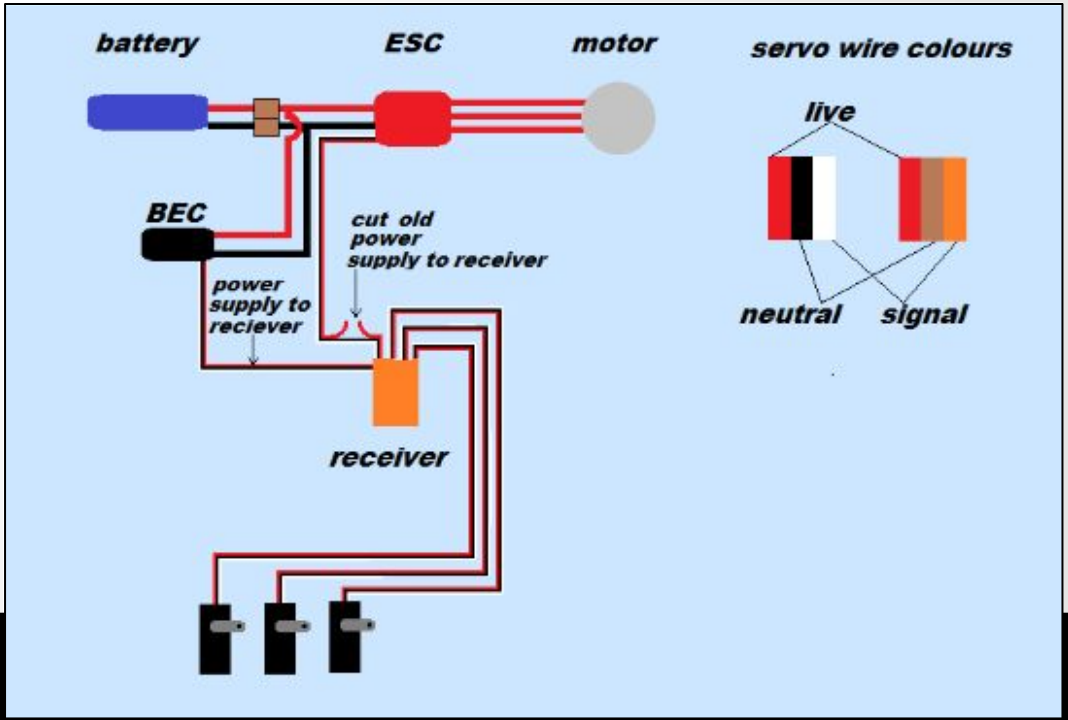
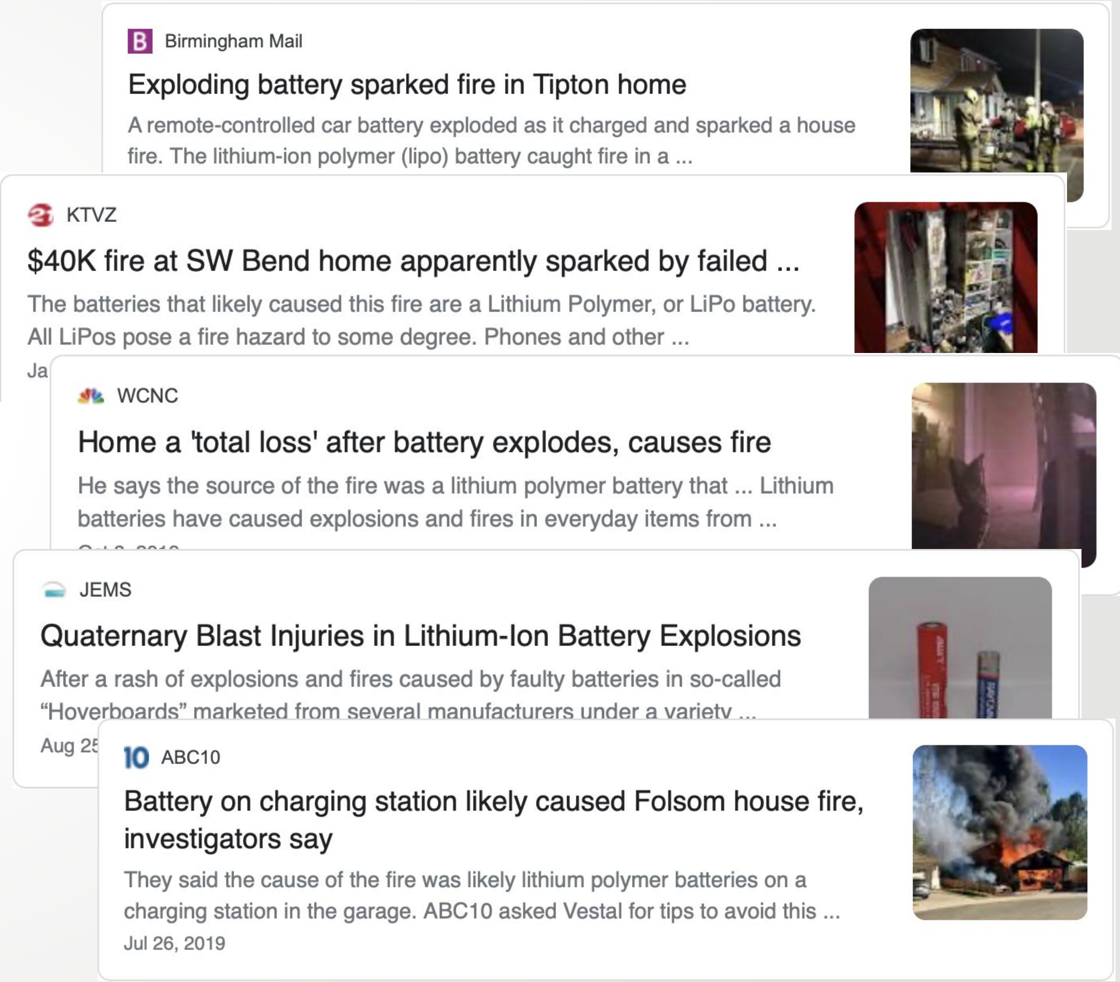


Image: *http://www.rcplanes4beginners.co.uk/bec-wiring-diagram.php*

LiPo Power Division & Delivery

- **Cells must be properly balanced and balanced charged**
  - No difference > 5-10mV
- **Charge and store in fireproof container**
  - Glass, metal bowl/box
  - Never leave unattended while charging
- **Never let cells get under/over 2.9V / 4.2V**
  - Nominal voltage 3.2-4.2V
- **Store with cells between 3.6V - 4.8V**
  - Never leave fully charged for more than 1-3 days
- **Never puncture**
  - dispose of old LiPo batteries properly
- **These are the the most important of many rules**


Birmingham Mail
Exploding battery sparked fire in Tipton home
A remote-controlled car battery exploded as it charged and sparked a house fire. The lithium-ion polymer (lipo) battery caught fire in a ...

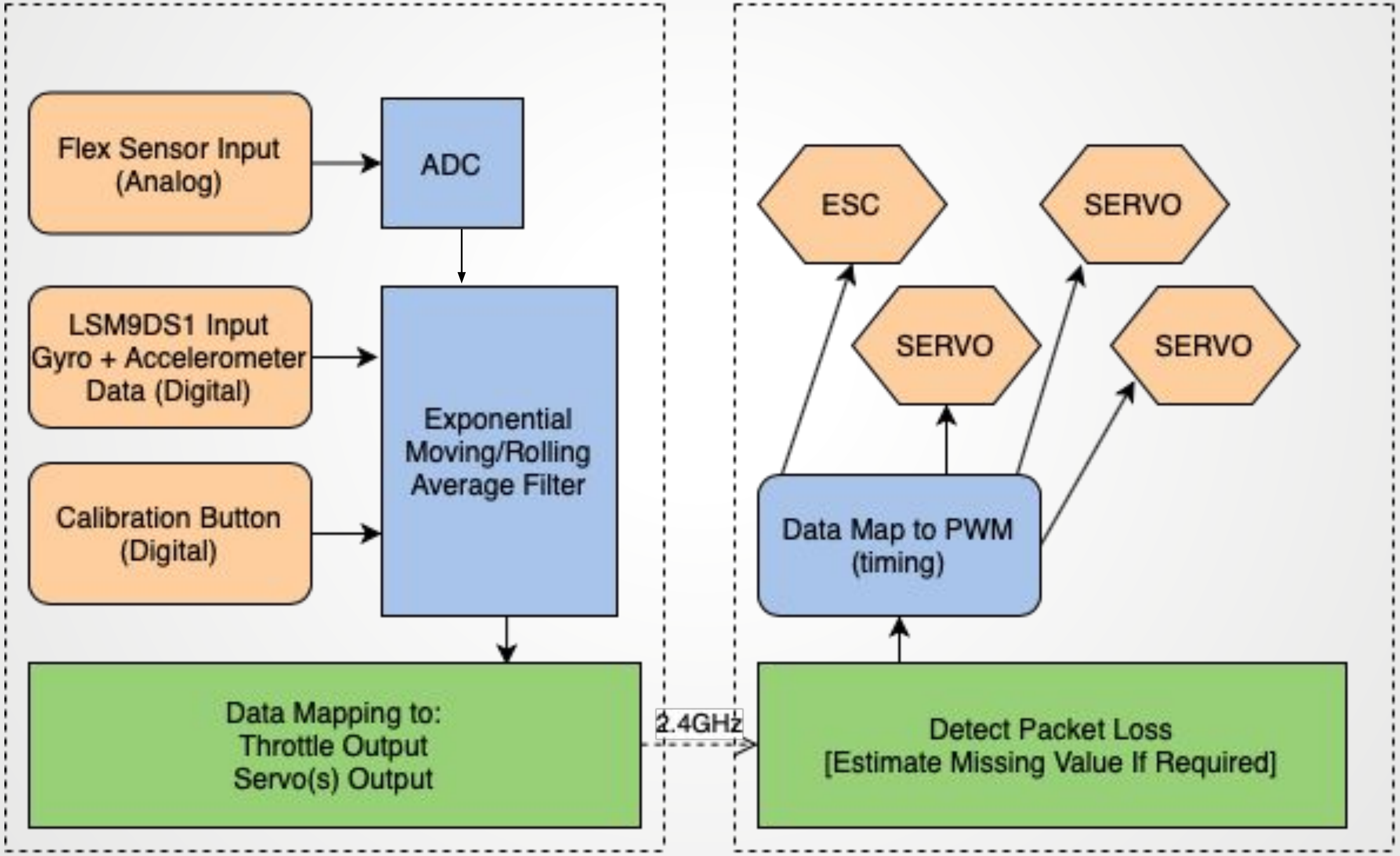
KTVZ
$40K fire at SW Bend home apparently sparked by failed ...
The batteries that likely caused this fire are a Lithium Polymer, or LiPo battery. All LiPos pose a fire hazard to some degree. Phones and other ...

WCNC
Home a 'total loss' after battery explodes, causes fire
He says the source of the fire was a lithium polymer battery that ... Lithium batteries have caused explosions and fires in everyday items from ...

JEMS
Quaternary Blast Injuries in Lithium-Ion Battery Explosions
After a rash of explosions and fires caused by faulty batteries in so-called "Hoverboards" marketed from several manufacturers under a variety ...

10 ABC10
Battery on charging station likely caused Folsom house fire, investigators say
They said the cause of the fire was likely lithium polymer batteries on a charging station in the garage. ABC10 asked Vestal for tips to avoid this ...
Jul 26, 2019

Software Block Diagram



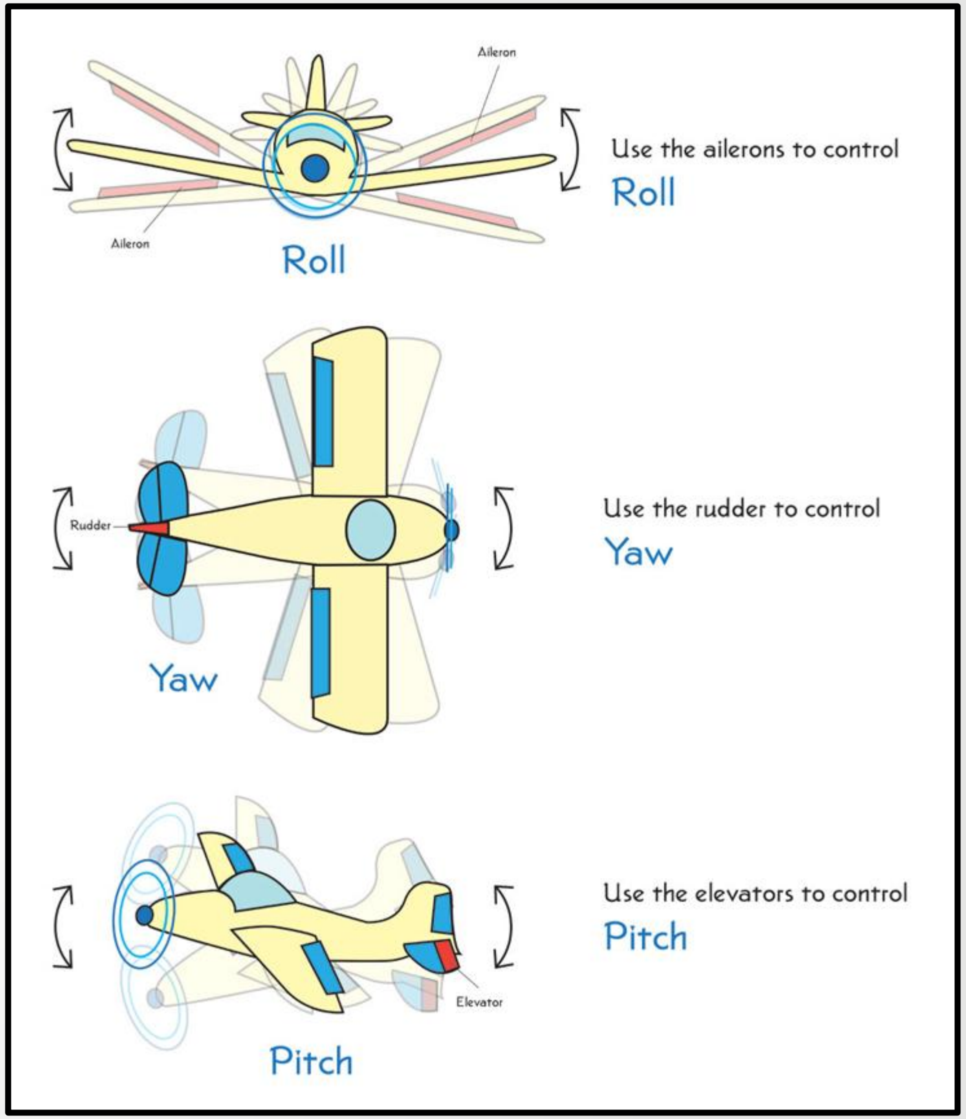
Accelerometer vs. Gyroscope

Gyroscope measures the rate of angular change (of the hand), therefore it must be continuously integrated

- **Used for Yaw**

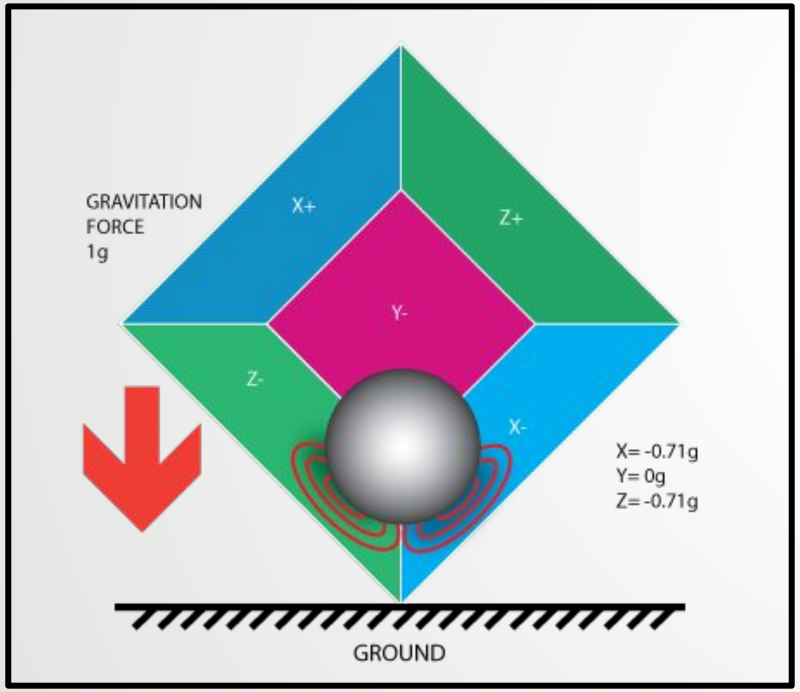
Accelerometer measures the acceleration vector on an axis, therefore it does not require any integration

- **Used for Roll (X) and Pitch (Z)**

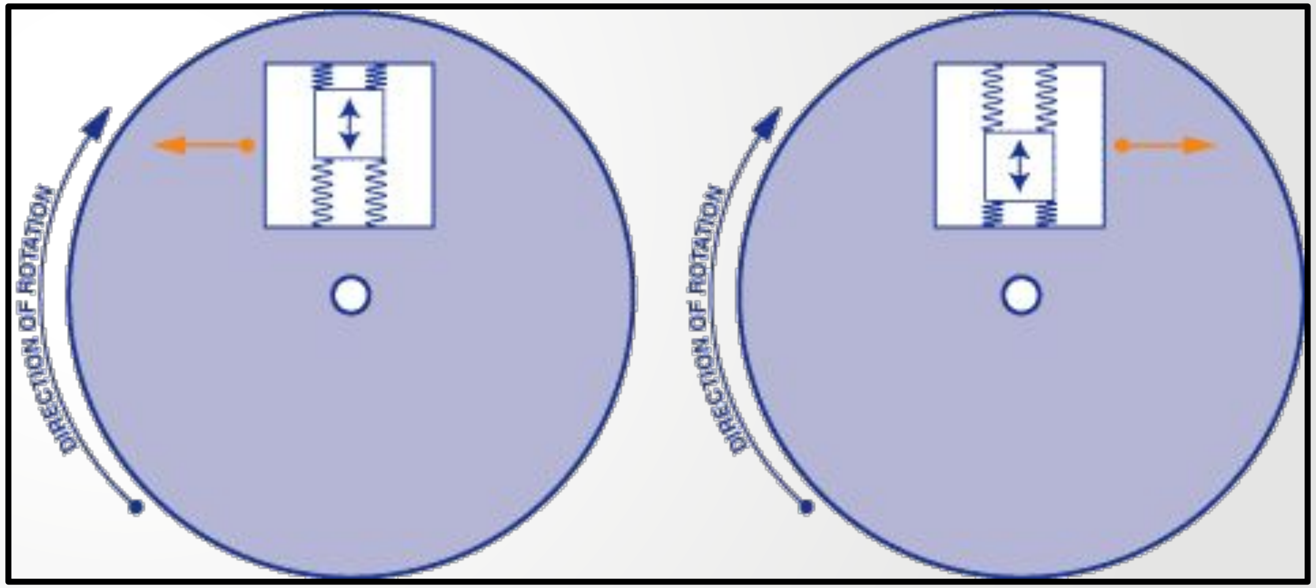


Accelerometer vs. Gyroscope

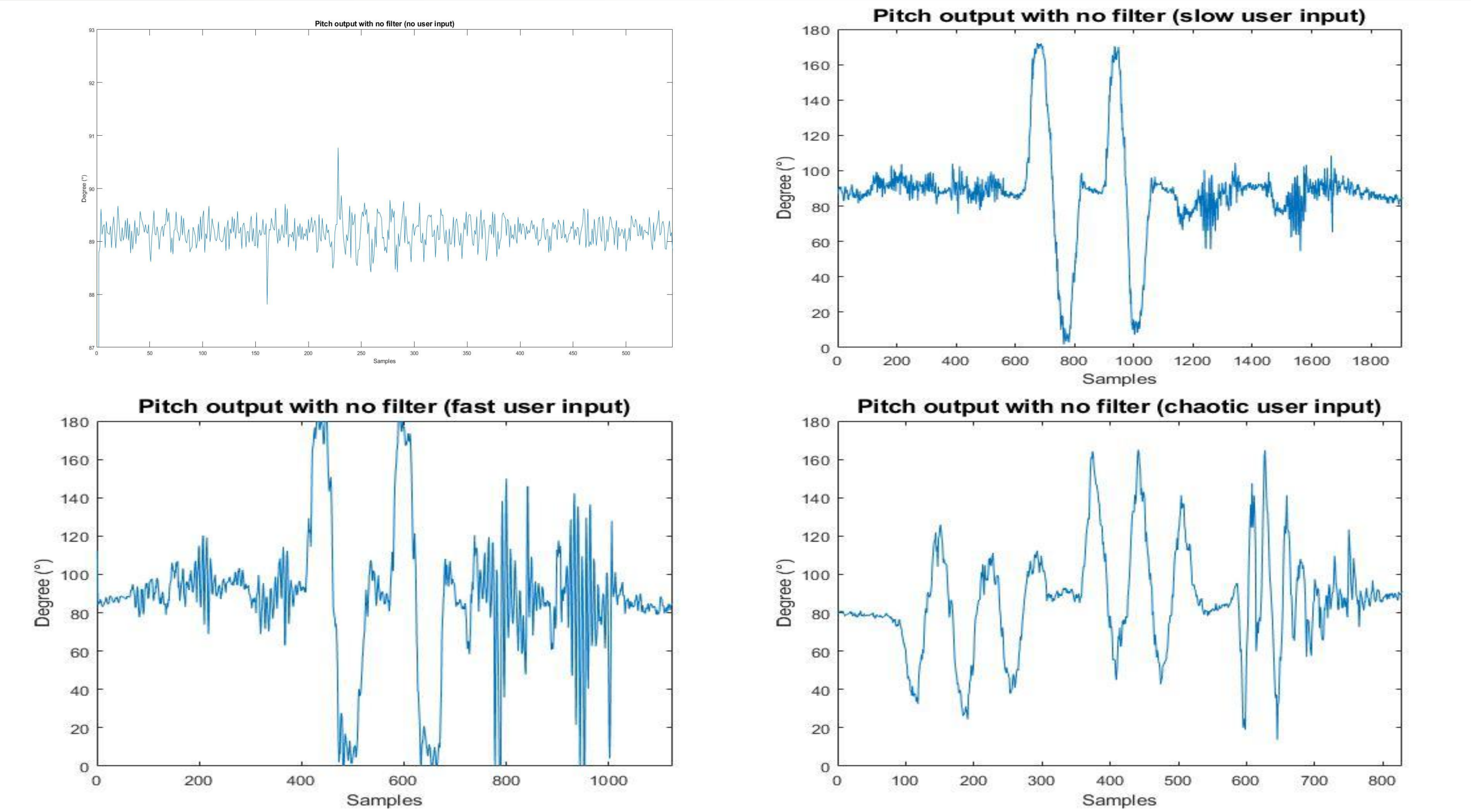
Accelerometer



Gyroscope



Filtering Design

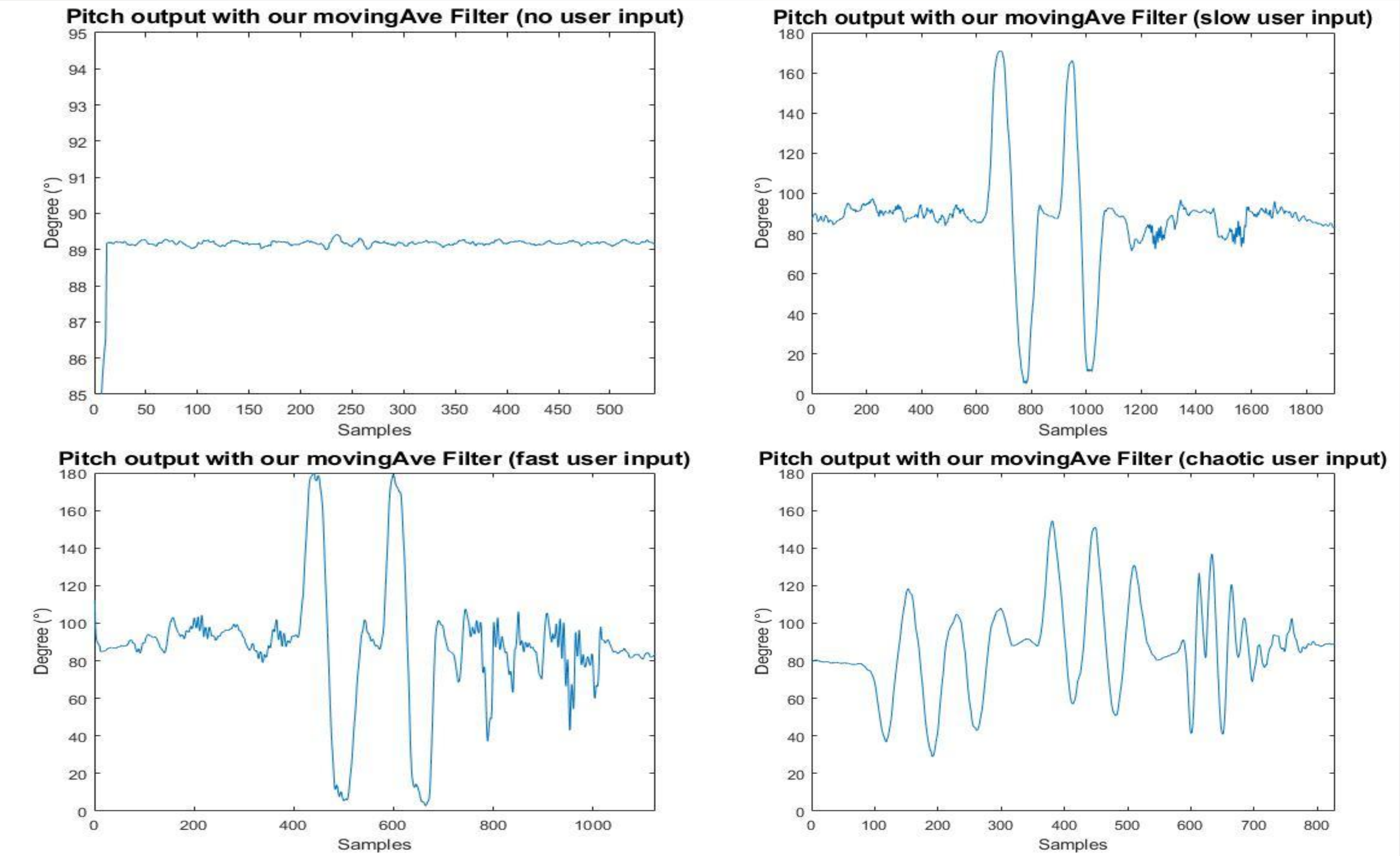


Sample Frequency: 100Hz

Speculation Criteria:

- **Eliminate noise as much as possible + create a curved shape with user input angle**
- **Fast respond to abrupt change**
- **Coding requirements: easy and not much space**

Rolling Average Filter

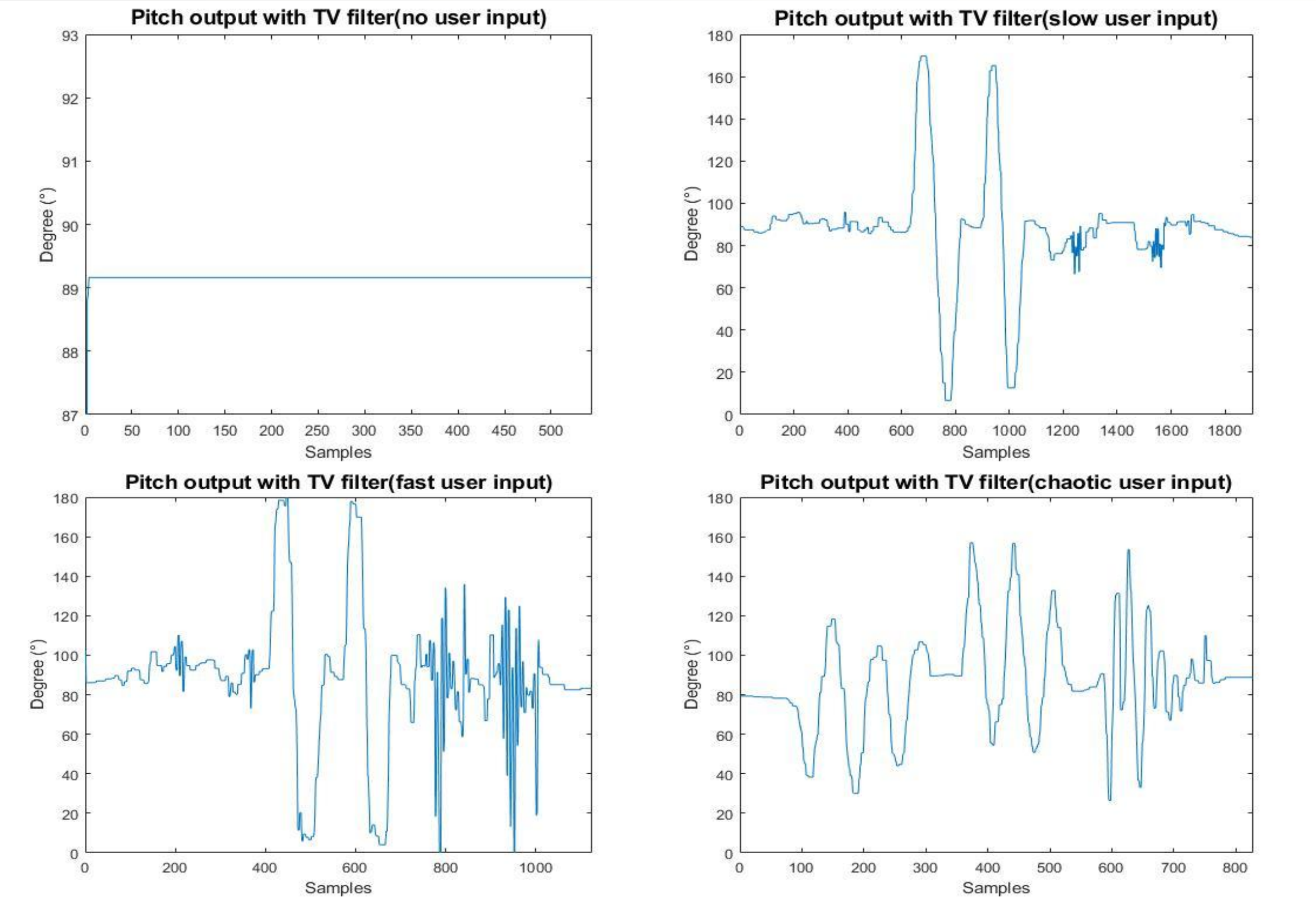


Comment:

- Our window size is 10 data points, including the current data point to do the approximation

- **The size <u>matters</u> because it can significantly change our system perception of turning angles**.

Total Variation Filter



Optimization Problem:

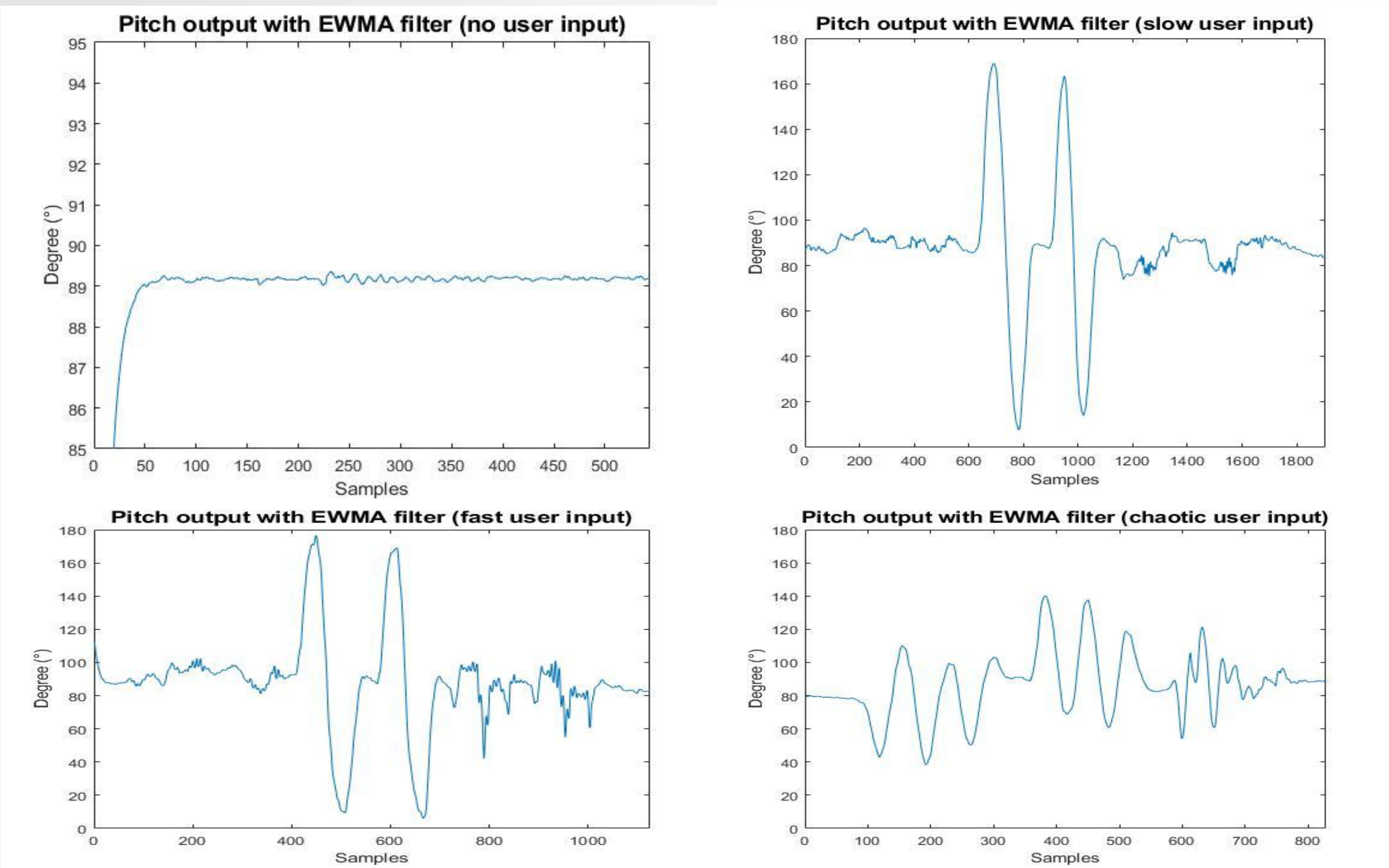
- **Solving the Cost Function J(x) using Iterative clipping algorithm**

$$J(\mathbf{x}) = \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \|\mathbf{D}\mathbf{x}\|_1.$$

Note : Dx measures how much the signal changes between values (Total Variation).

Reference for clipping solution : https://eeweb.engineering.nyu.edu/iselesni/lecture_notes/TV_filtering.pdf

Exponentially Weighted Moving Average (EWMA) Filter



EWMA equation :

$$S_t = ax_t + (1- a)S_{t-1}$$

Parameter :

x : is the value at time t (our measurement)

S : is the value of EMA at time t

a : weighting value. We set it to 0.1

Dropped Signal (Packet loss)

Dropped packets over transmission can lead to inaccuracy in the desired servo/motor position/control. We will implement a packet loss detection, where we replace the dropped packet with an estimated value from the previous two samples.

Window #											
1	2	3	4	5	6	7	8	9	10		
7	9	9	8	7	8	9	8	7	6	7.8	<-- Desired filtering result
7	9	9	8	7	8	0	8	7	6	6.9	<-- Filtering w/ packet loss
7	9	9	8	7	8	10	8	7	6	7.9	<-- Packet loss detection

Packet[7] = | Packet[5]-Packet[6] | / Window_Size[10]

* This is the equivalent to finding the tangent of the signal immediately before the packet loss is detected *

List of Hardware/Software (To Date)

Hardware

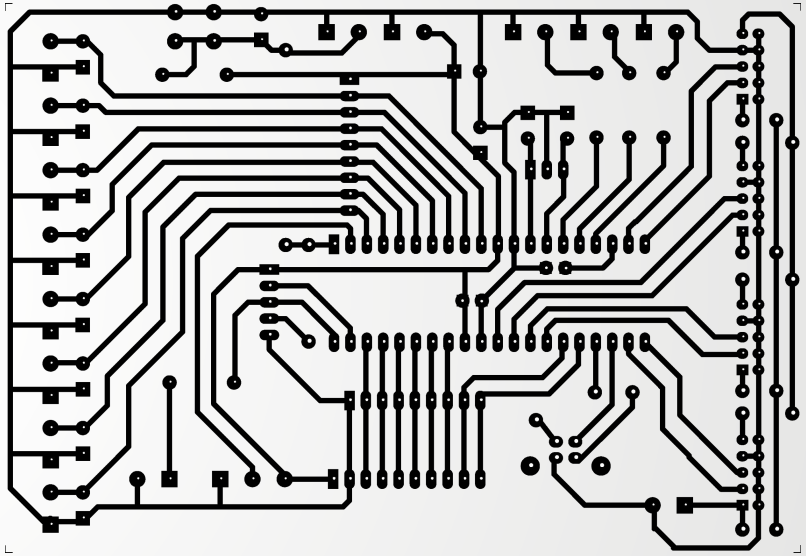
- Arduino Nano (x2)
- Arduino Uno [retired]
- NRF24L01+PA+LNA RF Module (x2)
- LSM9DS1 IMU Board
- GY-521 IMU Board [retired]
- SG-90 Servo Motor (x3) [retired]
- 3” Flex Sensor
- 10kohm resistor
- LED (placeholder for engine)
- 200 ohm resistor (for LED)
- Nitrile ESD Glove
- Breadboard Power Supply Module
- Breadboard, Jumper Wires
- JST-XH connectors (x3)
- E-flite RC Airplane UMX Turbo Timber

Software

- Arduino Uno : 1st iteration of main code [retired]
- Arduino Nano : rehauled main code, receiver & transmitter integration [retired]
- Arduino Nano : final prototype code, complete system integration [current]

Rough Cost Estimate (To Date)

- MDR expenses: $66.19
- Demo aircraft: $139.99
- LiPo Battery (2-pack): $17.68
- LiPo Balance Charger: $13.99
- 1st Iteration PCBs: $65.80*
- 2nd Iteration PCBs: $75.30*
- Connector/Crimping Kit: $36.99



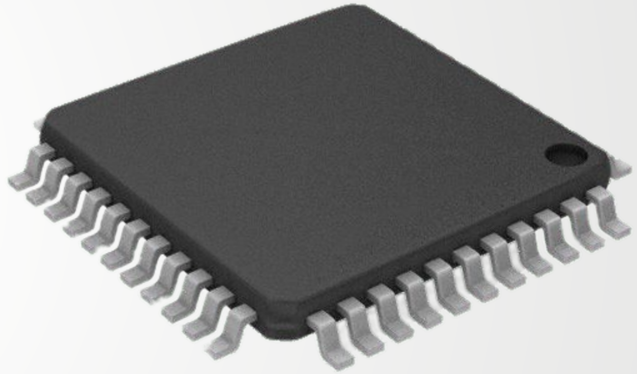
Total: ~ $415.94

Remaining: ~ $84.06

** Each iteration includes 5x TX PCBs, 5x RX PCBs, 1x TX Stencil, 1x RX Stencil

Rough Cost Estimate (Projected for FPR)

- CDR expenses: $415.94
- PCB parts: ~$20
- UBEC: $21.99
- PCB-mounted servos (x2): $35.98 ($17.99 ea.)



Total: ~ $493.91

Remaining: ~ $6.09

** Each iteration includes 5x TX PCBs, 5x RX PCBs, 1x TX Stencil, 1x RX Stencil

FPR Plan

- **Replace Arduinos/breadboards/excess wiring with custom PCBAs**
- **Ensure TX PCBA is powered by 4x AAA batteries**
- **Ensure RX PCBA is powered by LiPo battery**
- **Optimize plane's center of gravity (time permitting)**
  - This can mostly be done with the heavy LiPo positioning
- **Calibrate servo movements on the plane**

Post-CDR Responsibilities

Jacob

- finalize glove assembly, finalize RC plane assembly, finish purchasing, and help Bradley fine-tune ANSI-C code (eg. servo offsets, flex sensor flicker protection, etc.)

Bradley

- Finalize ANSI-C code, populate and solder second iteration of PCB

Son

- Finalize our PCBs design, manage team purchase and debug our code.

Demo