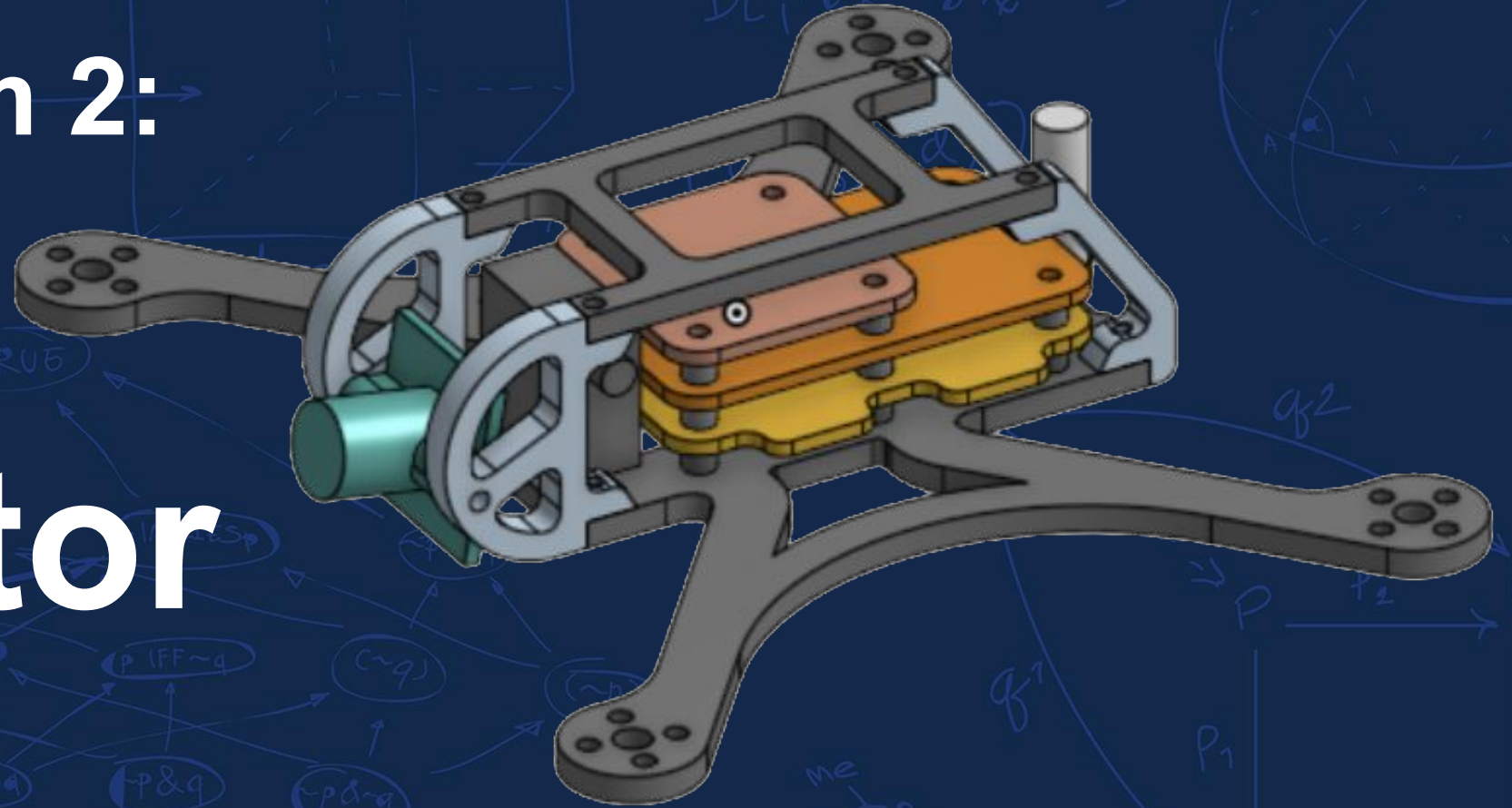




UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

ECE445 Team 2:

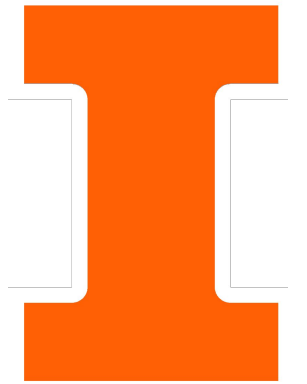
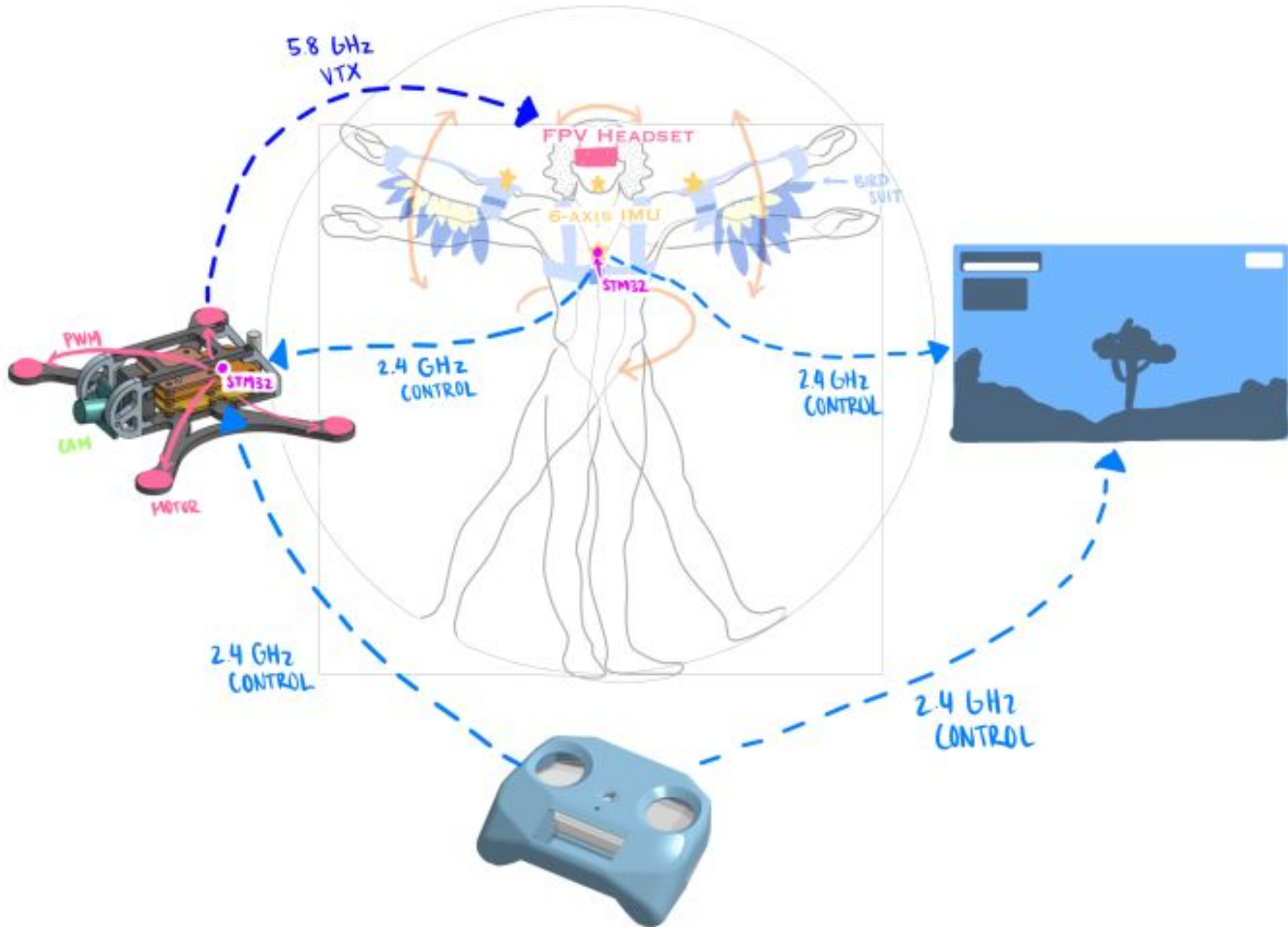
Bird Simulator



Anthony Amella

Emily Liu

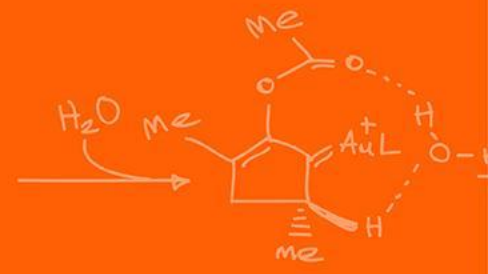
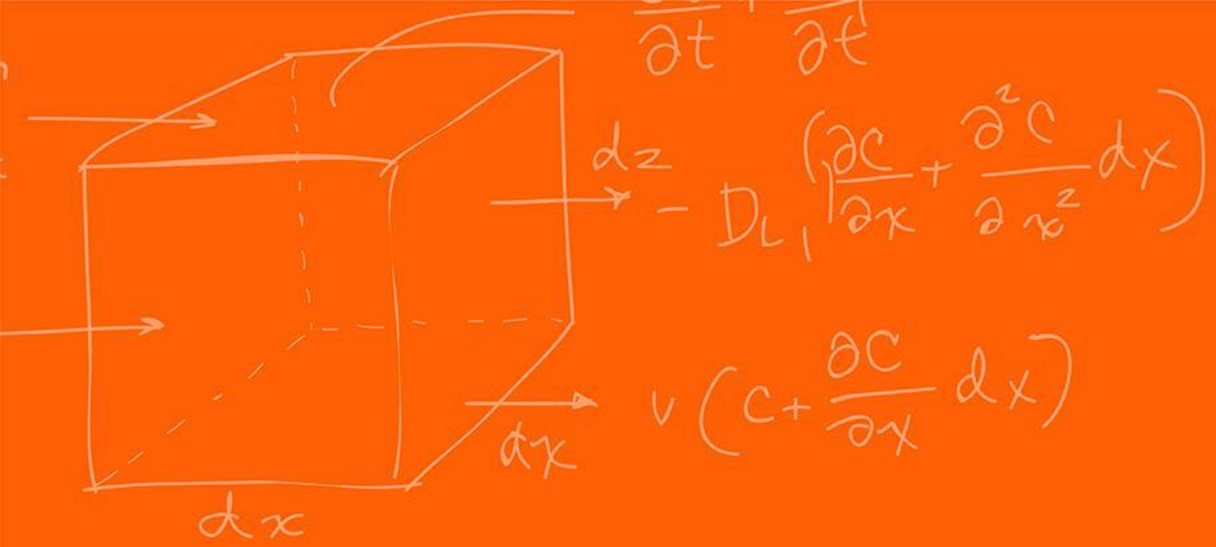
Eli Yang



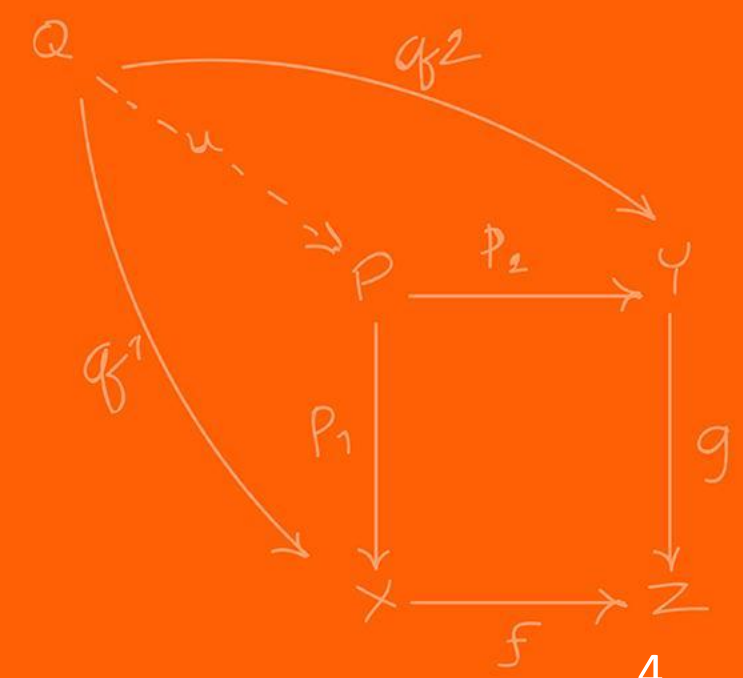
Objective

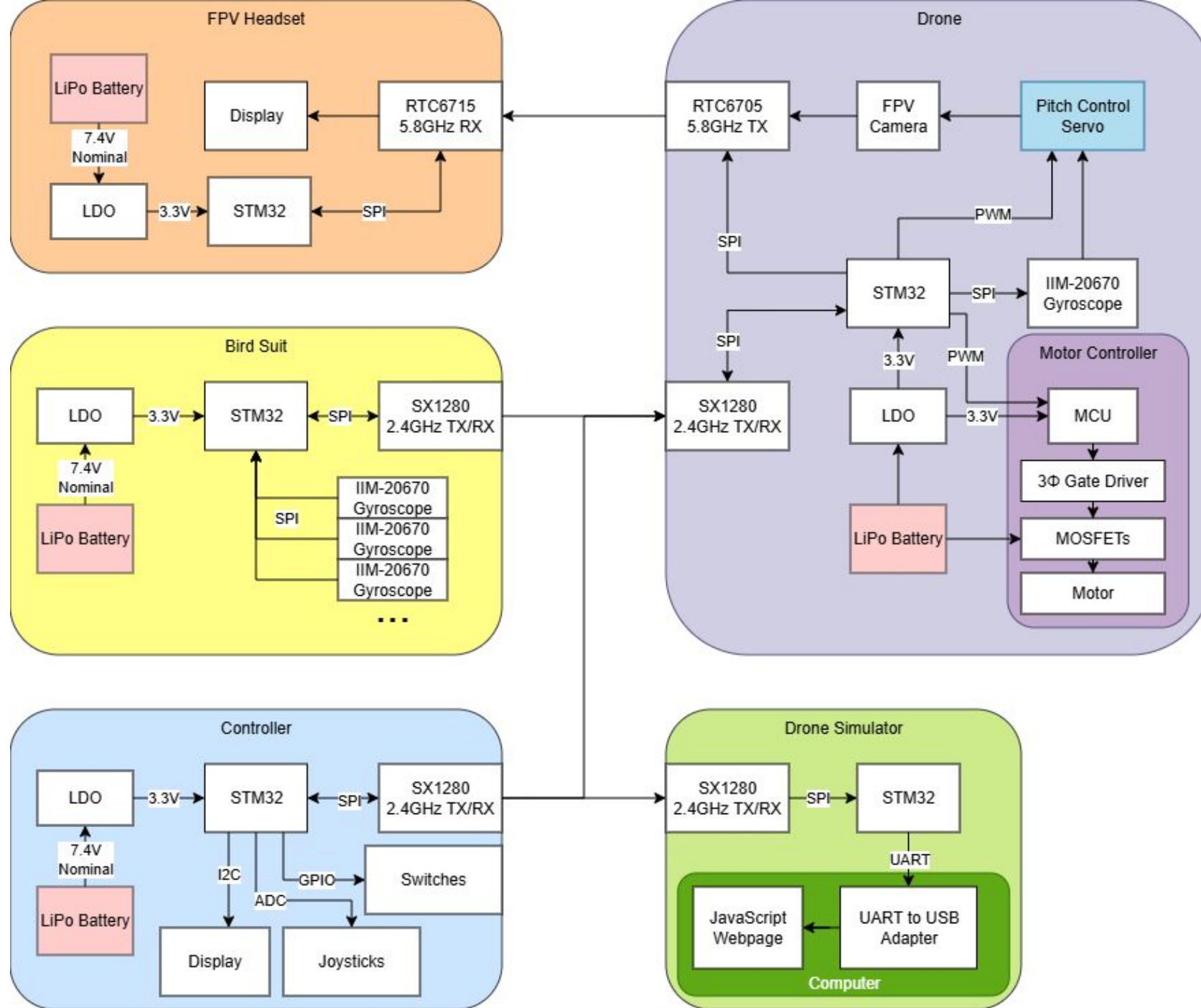
- Recreational
 - FPV for immersive flight experience
 - Does not incorporate whole body
- Operational Applications
 - Reduce cognitive load
 - Situational awareness
 - Reaction-based control



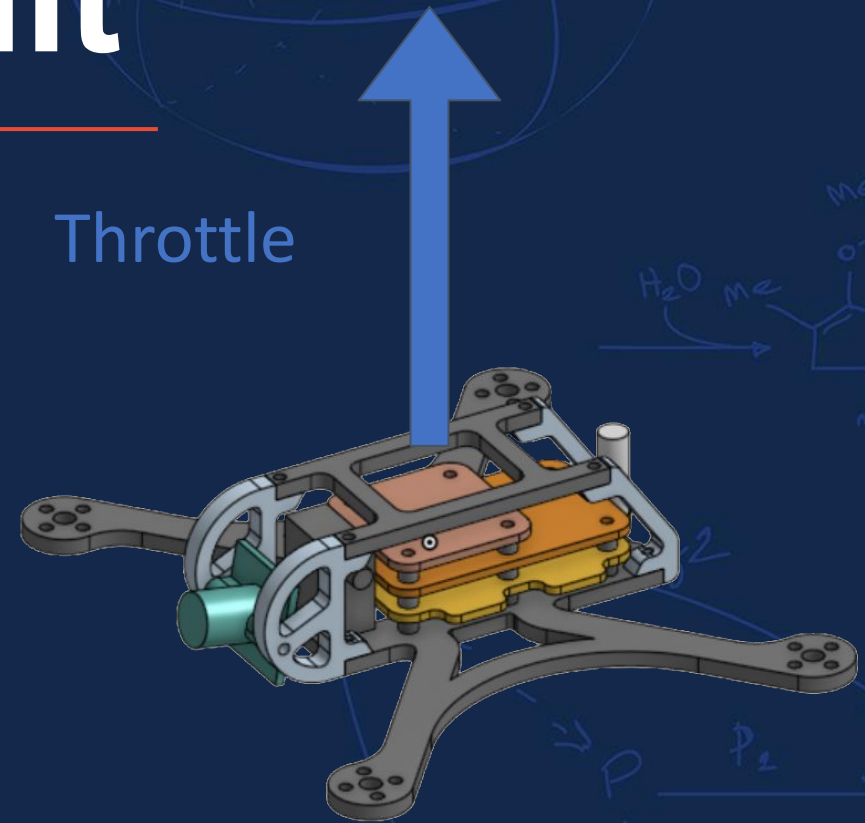
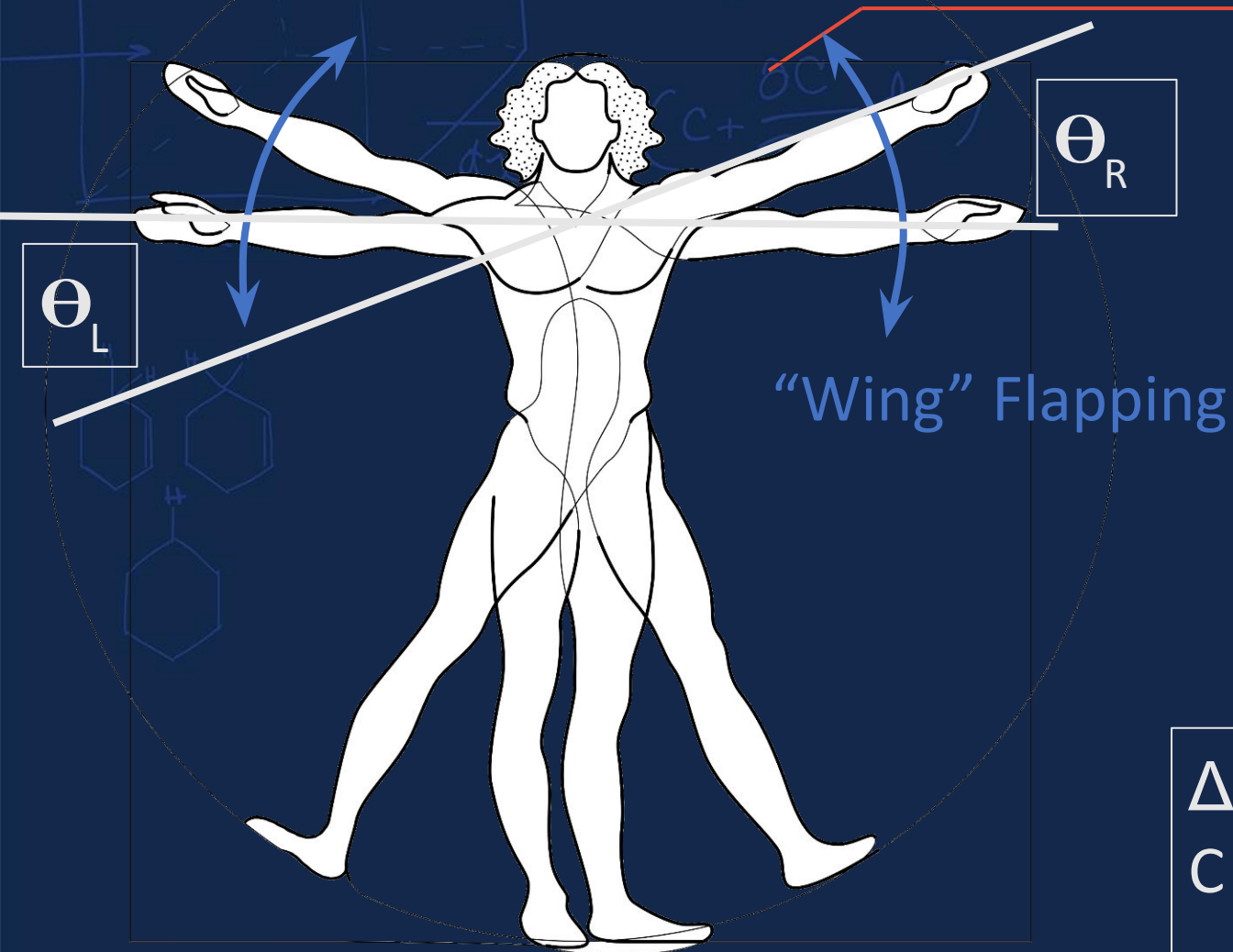


Design





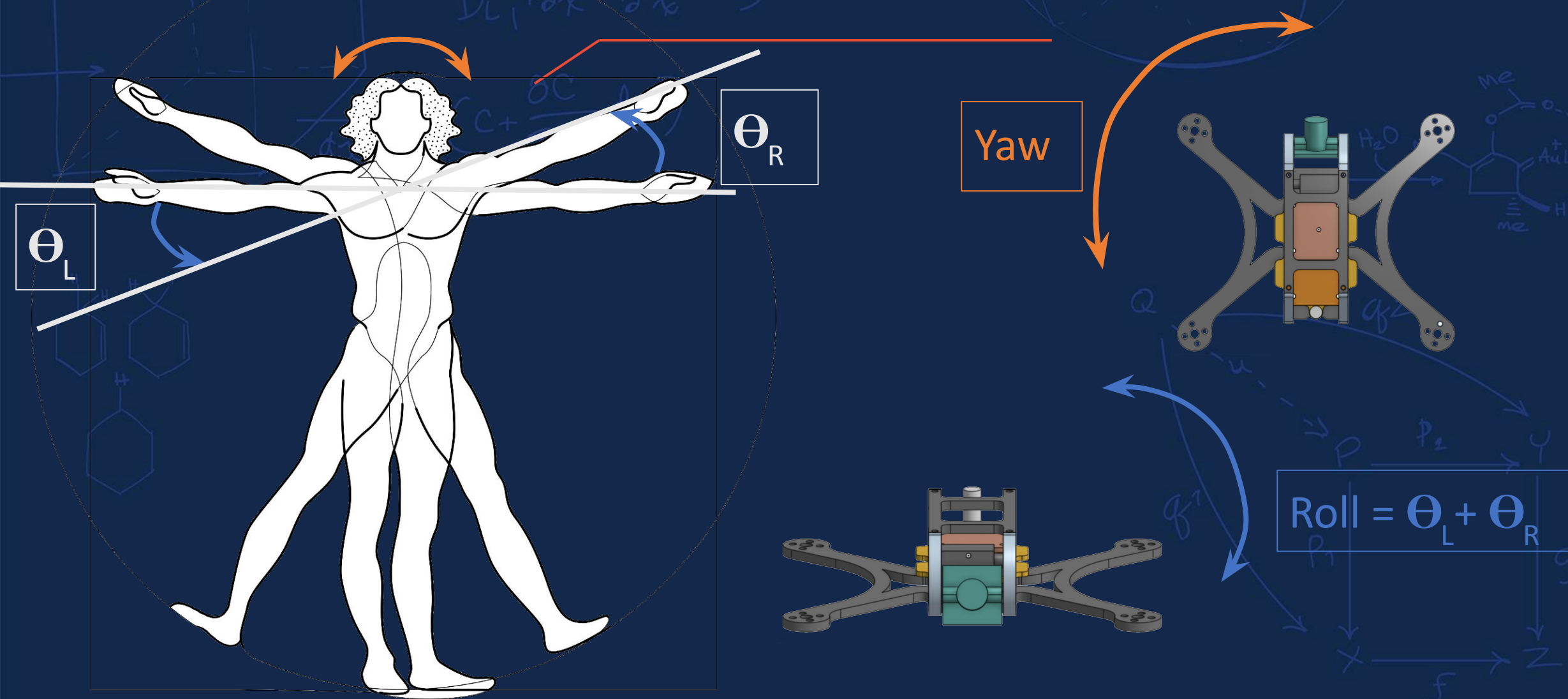
Bird Suit



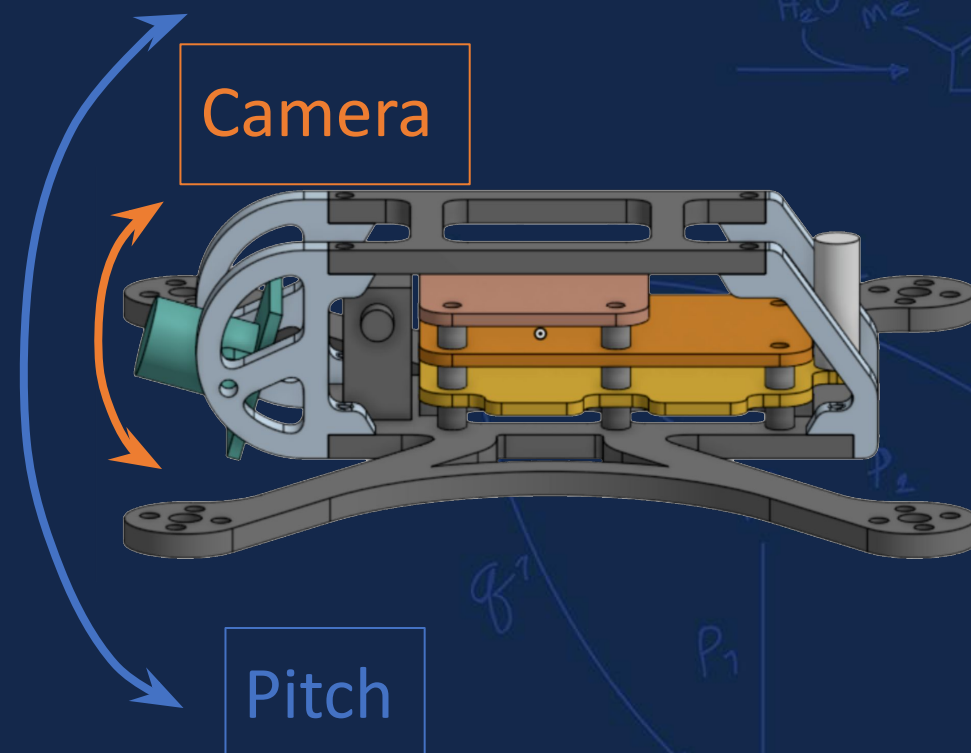
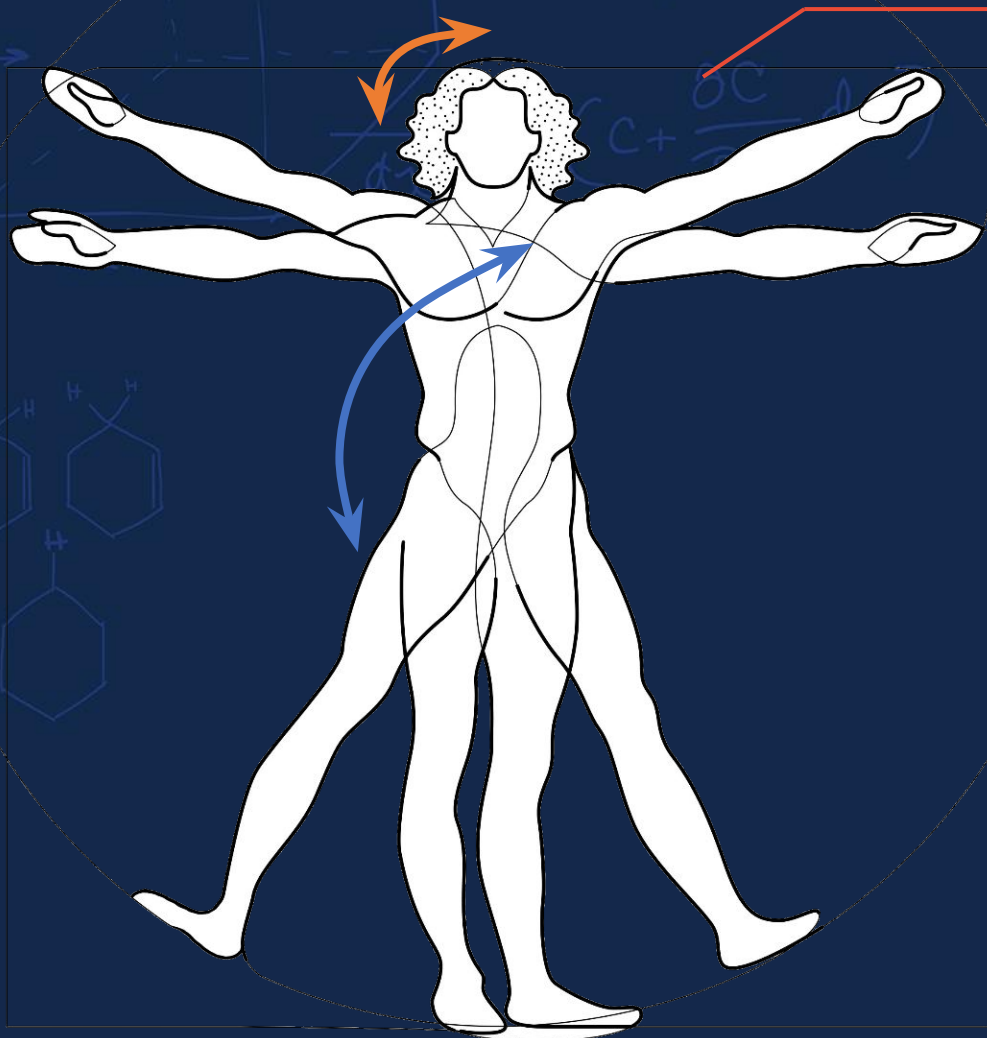
$$\Delta \text{Throttle} = \max(0, \Delta \theta_L - \Delta \theta_R) - C$$

$$\text{Throttle}' = \text{Throttle} + \Delta \text{Throttle}$$

Bird Suit

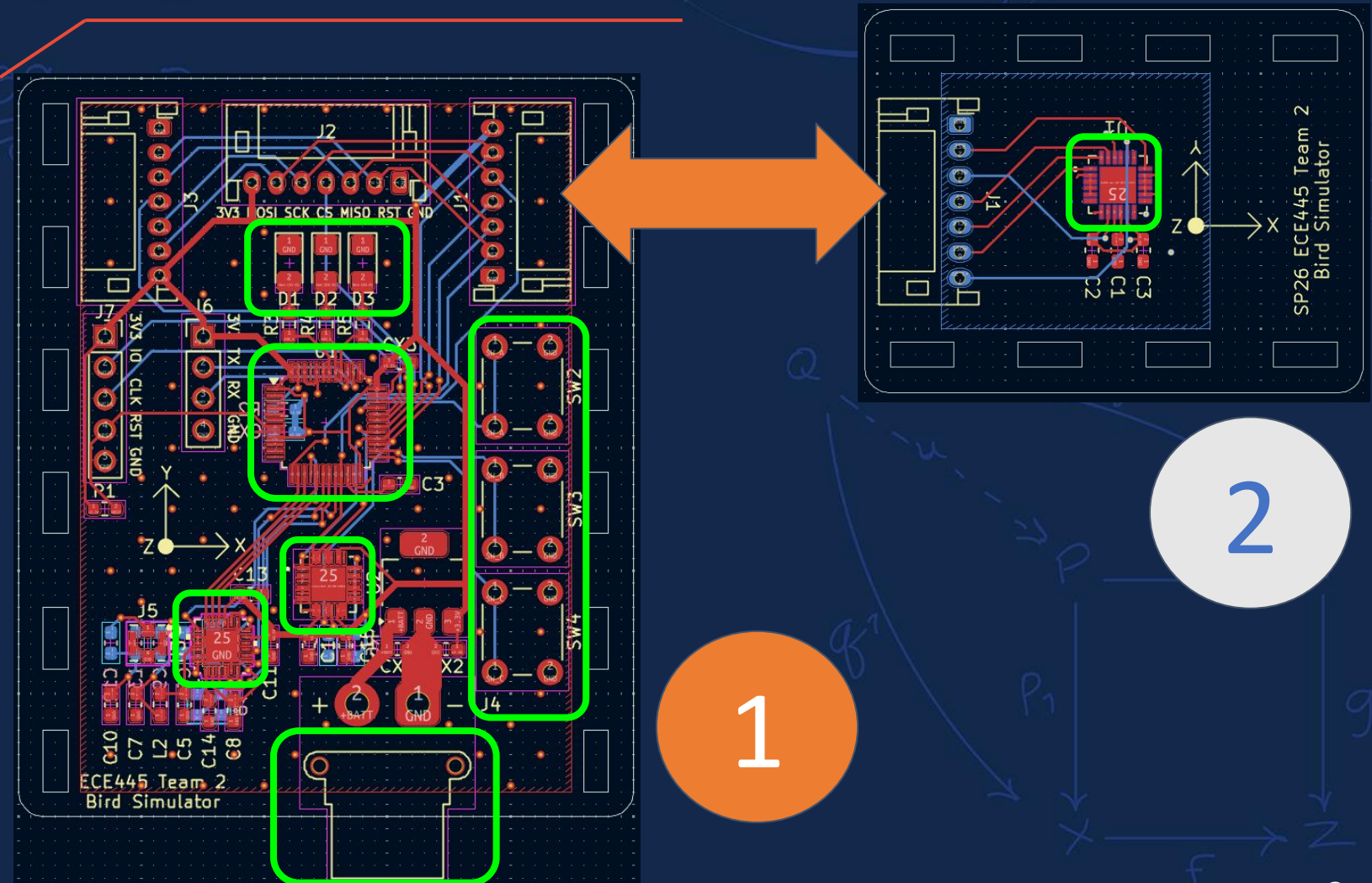


Bird Suit

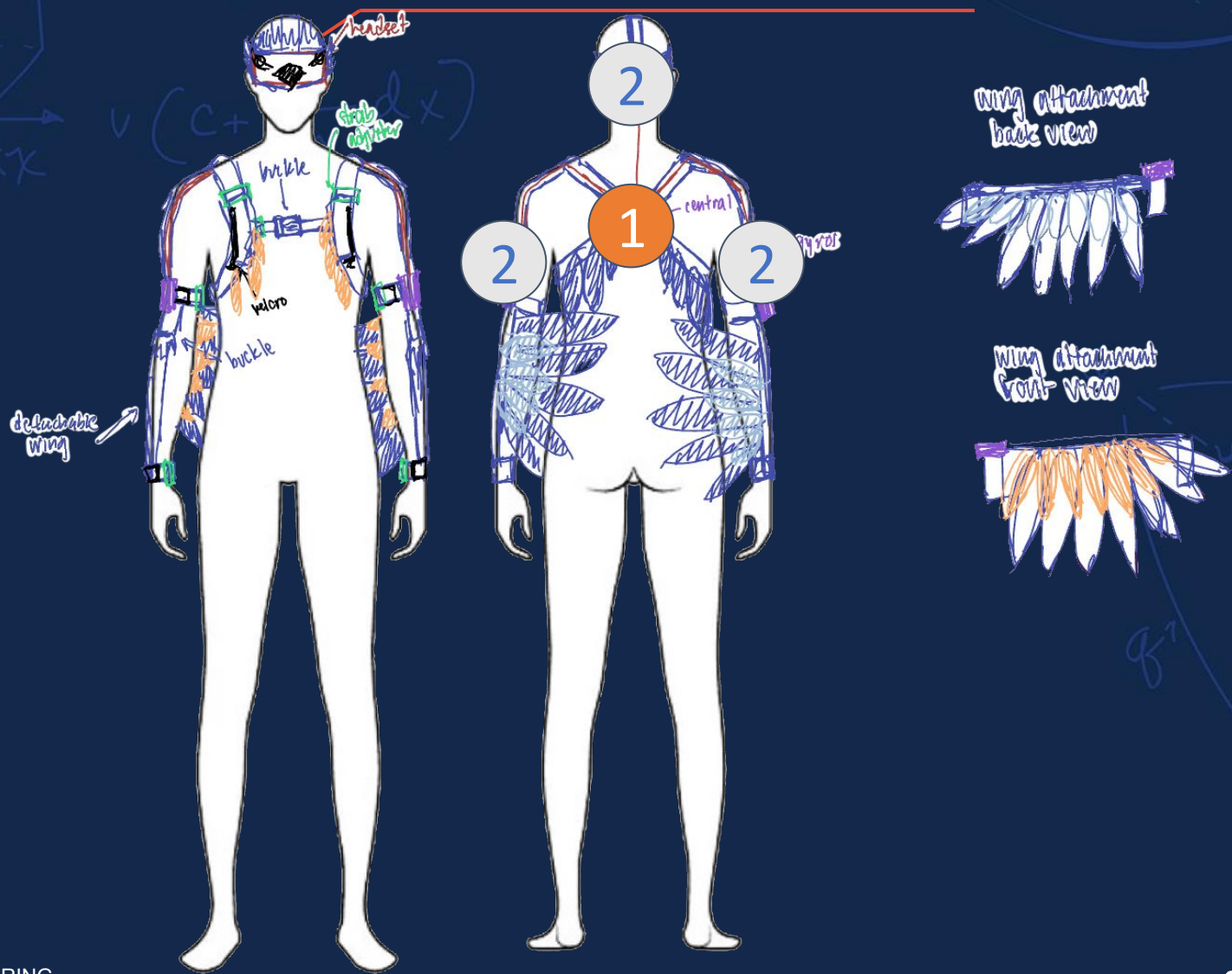


Bird Suit

- STM32F4 to process IMU data
- SX1280 to transmit flight data
- Four IIM-20670 IMUs to measure flight movement
 - One on main board, three auxiliary boards
- Battery connection
- Buttons and LEDs to allow for arming, status



Bird Suit



SX1280 2.4GHz Transceiver

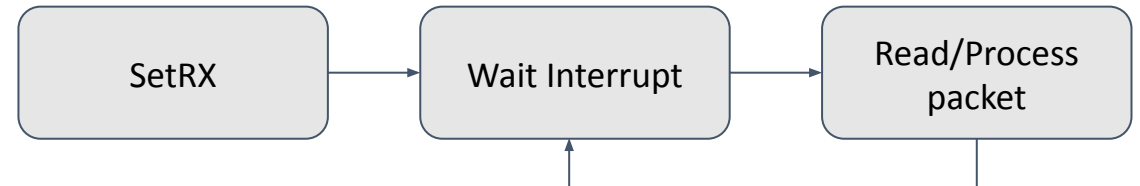
- SPI communication with STM32
- Command-based operation
- LoRa (Long Range) radio modulation
- Main communication between bird suit/controller and drone

```
module m = {
    .hspi = &hspi1,
    .nss = { SX_CS_Pin, GPIOB },
    .reset = { radioreset_Pin, GPIOB },
    .diol = { diol_Pin, GPIOA },
    .busy = { busy_Pin, GPIOA };
}
```

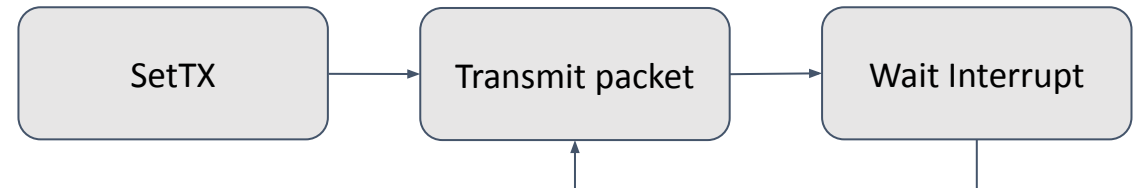
Send Command



RX Workflow

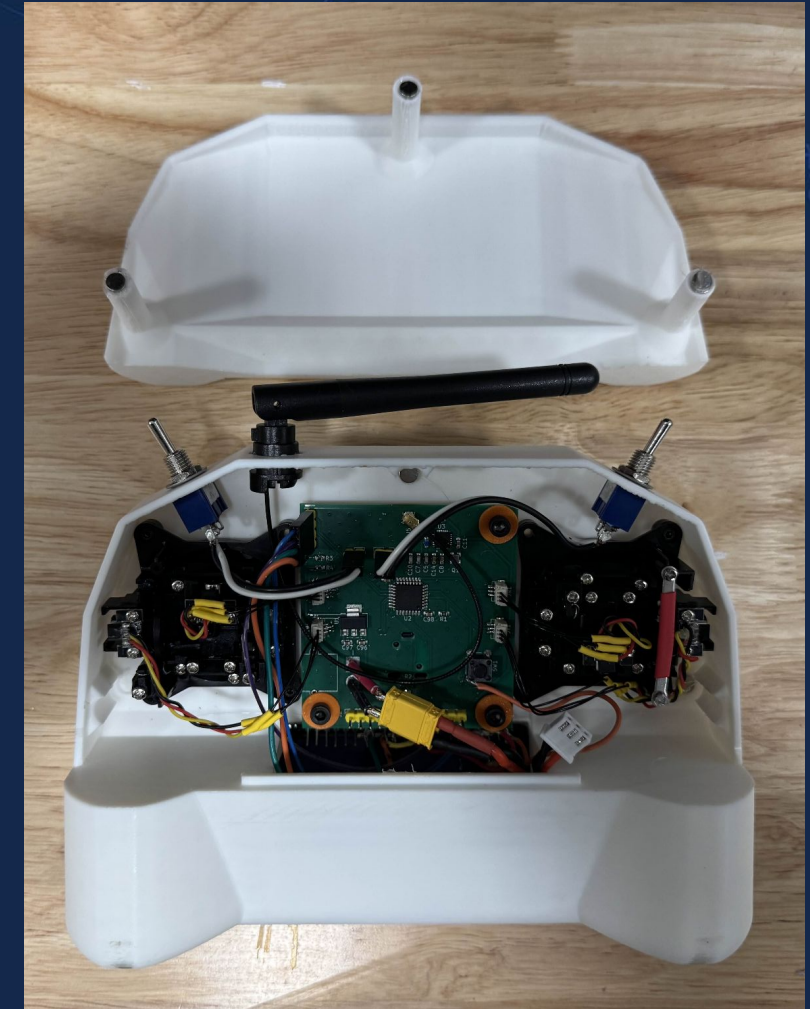
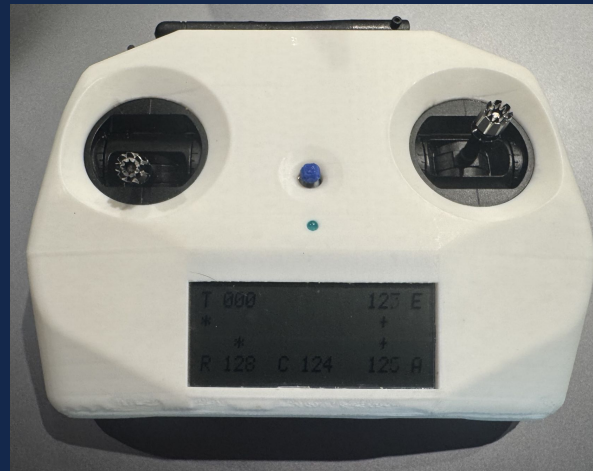
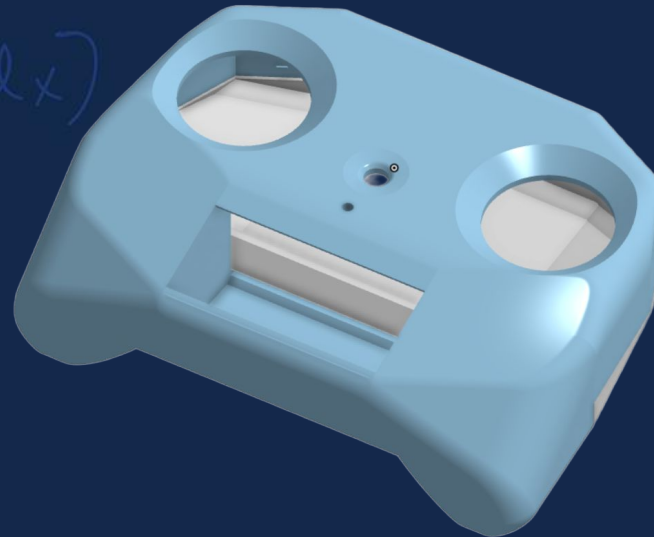


TX Workflow



Controller

- Two 2-axis joysticks to control throttle, roll, pitch, yaw
- STM32 to read sticks through ADC and transmit signals
- Switches to arm drone and control mode
- SSD1803 screen with custom driver to provide better user interface
- 5-way switch to control camera angle and tune parameters



Radio Packet

MODE	CAMERA	THROTTLE	YAW	PITCH	ROLL	ID	CRC
------	--------	----------	-----	-------	------	----	-----

1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

MODE |= 0x0F to arm

MODE |= 0xF0 for auto-levelling

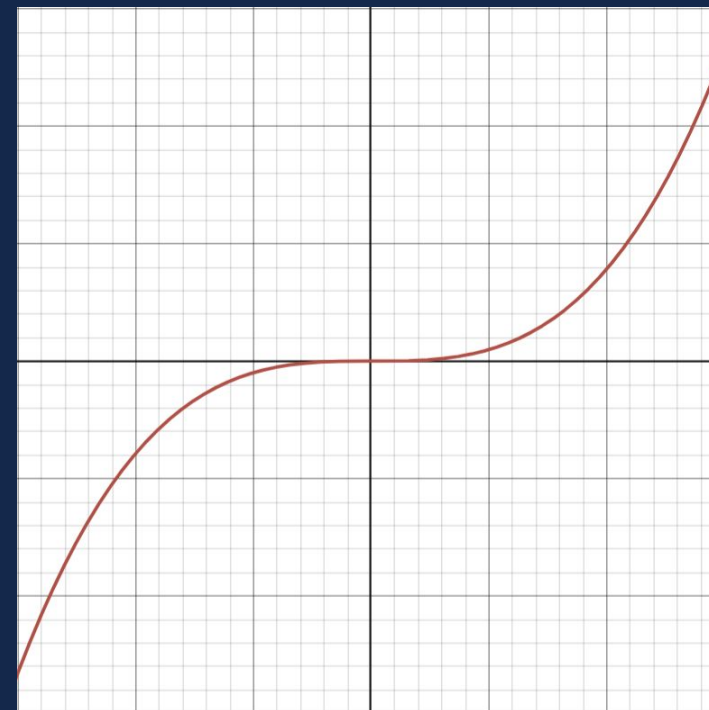
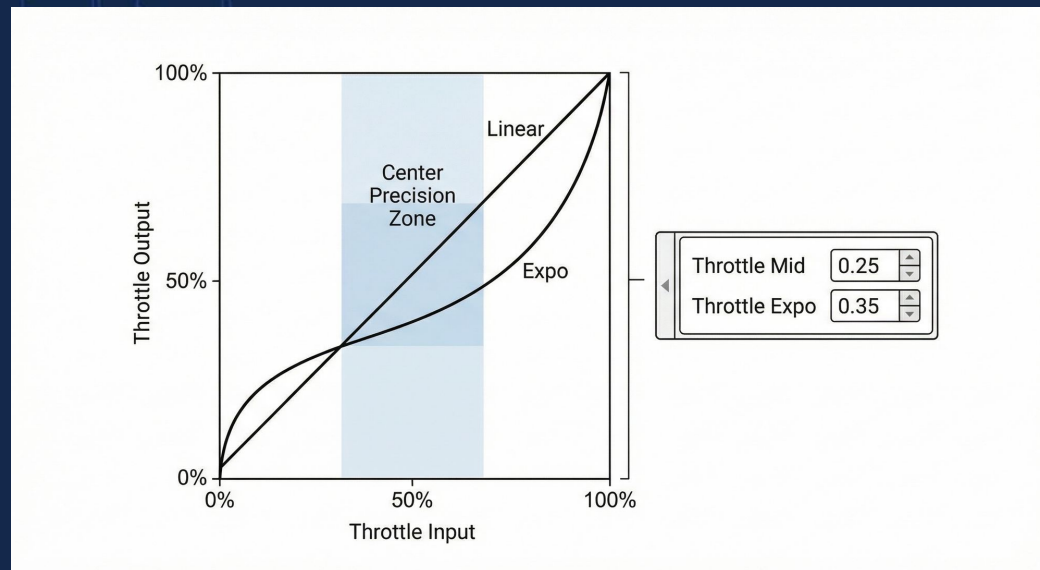
ID = 0x95 for controller

ID = 0x45 for bird suit
(0x95 overrides 0x45)

CRC = XOR of first 7 bytes

Expo Curve for Controls

- Throttle
 - Found hover point
 - Asymmetrical curve -> less sensitive around hover point (100/255)
- Yaw, Pitch, Roll
 - Symmetrical curve -> less sensitive around stick center (128/255)



Environment:

Racetrack

UART

Port: not connected

Connect

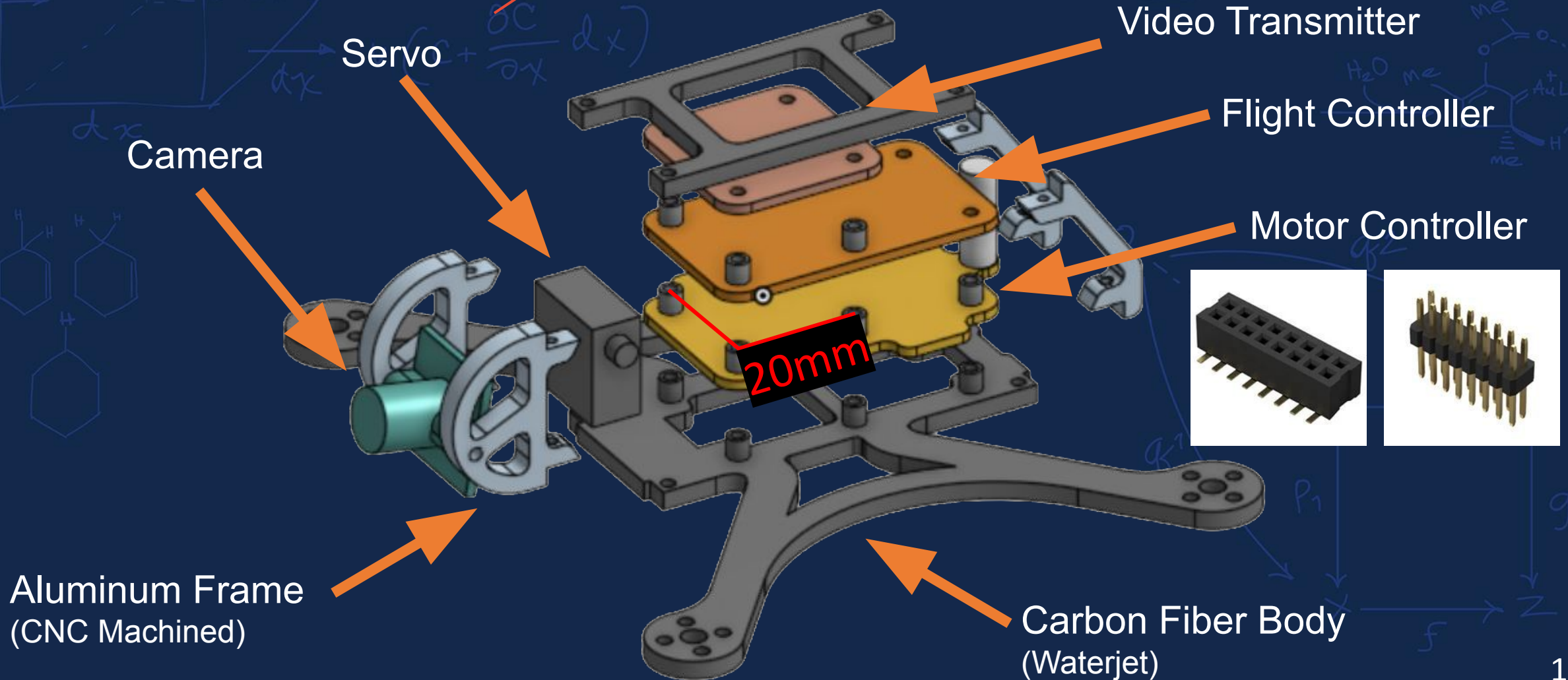
Disconnect

Respawn

Flight Simulator

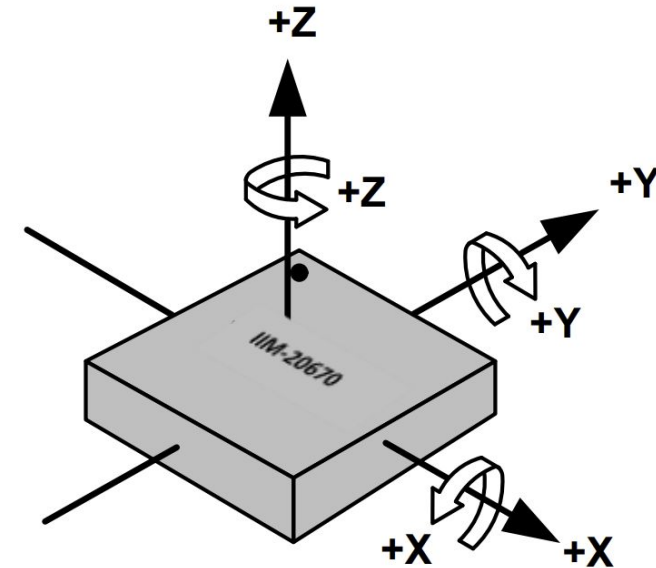
- JavaScript for flight implementation, [Three.js](#) for 3D rendering
- Physics based on Newton-Euler equations
- Receive control inputs through USB-to-UART adapter
- Use for initial testing of bird suit controls

Custom Mechanical Design



IIM-20670 6-Axis IMU

- SPI communication with STM32
- Register-based operation
- Accelerometer data scaled to represent linear acceleration in g's
- Gyroscope data scaled to represent angular velocity in degrees per second
- Drone PID control, suit movement capture

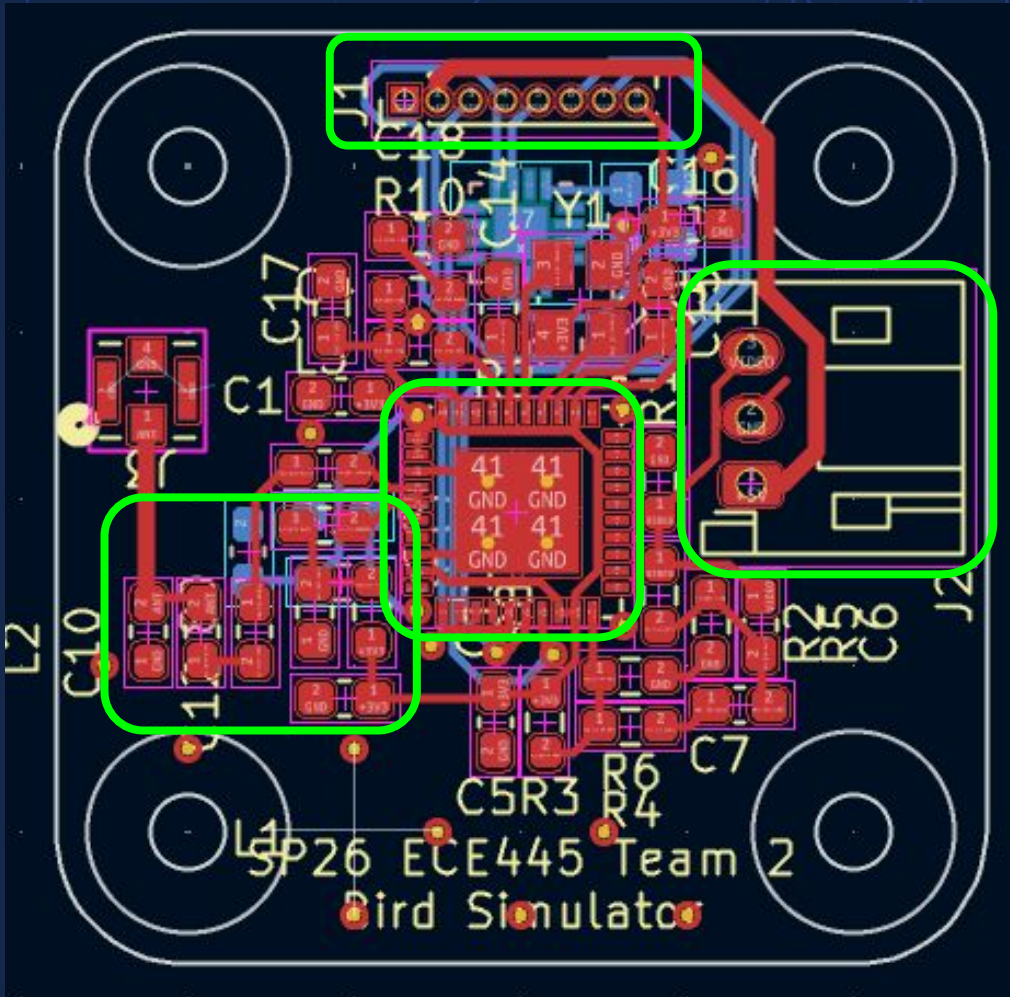


```
iim20670 iim = { .cs = {IMU_CS_Pin, GPIOB},
                .hspi=&hspi2 };
iim_scaleddata imu_data;
iim_status imu_status;
```

$$\omega_x = \frac{\text{gyro_x_data}}{2^{15}} * FS, \omega_y = \frac{\text{gyro_y_data}}{2^{15}} * FS, \omega_z = \frac{\text{gyro_z_data}}{2^{15}} * FS,$$

$$a_x = \frac{\text{accel_x_data}}{2^{15}} * FS, a_y = \frac{\text{accel_y_data}}{2^{15}} * FS, a_z = \frac{\text{accel_z_data}}{2^{15}} * FS,$$

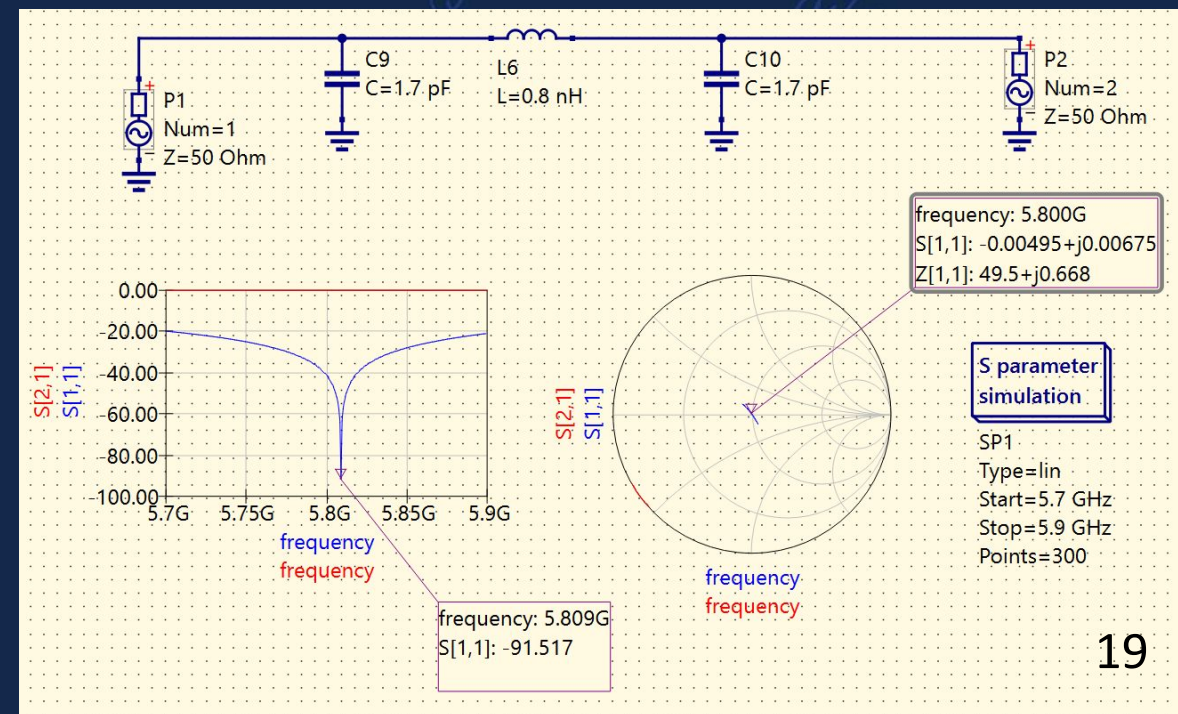
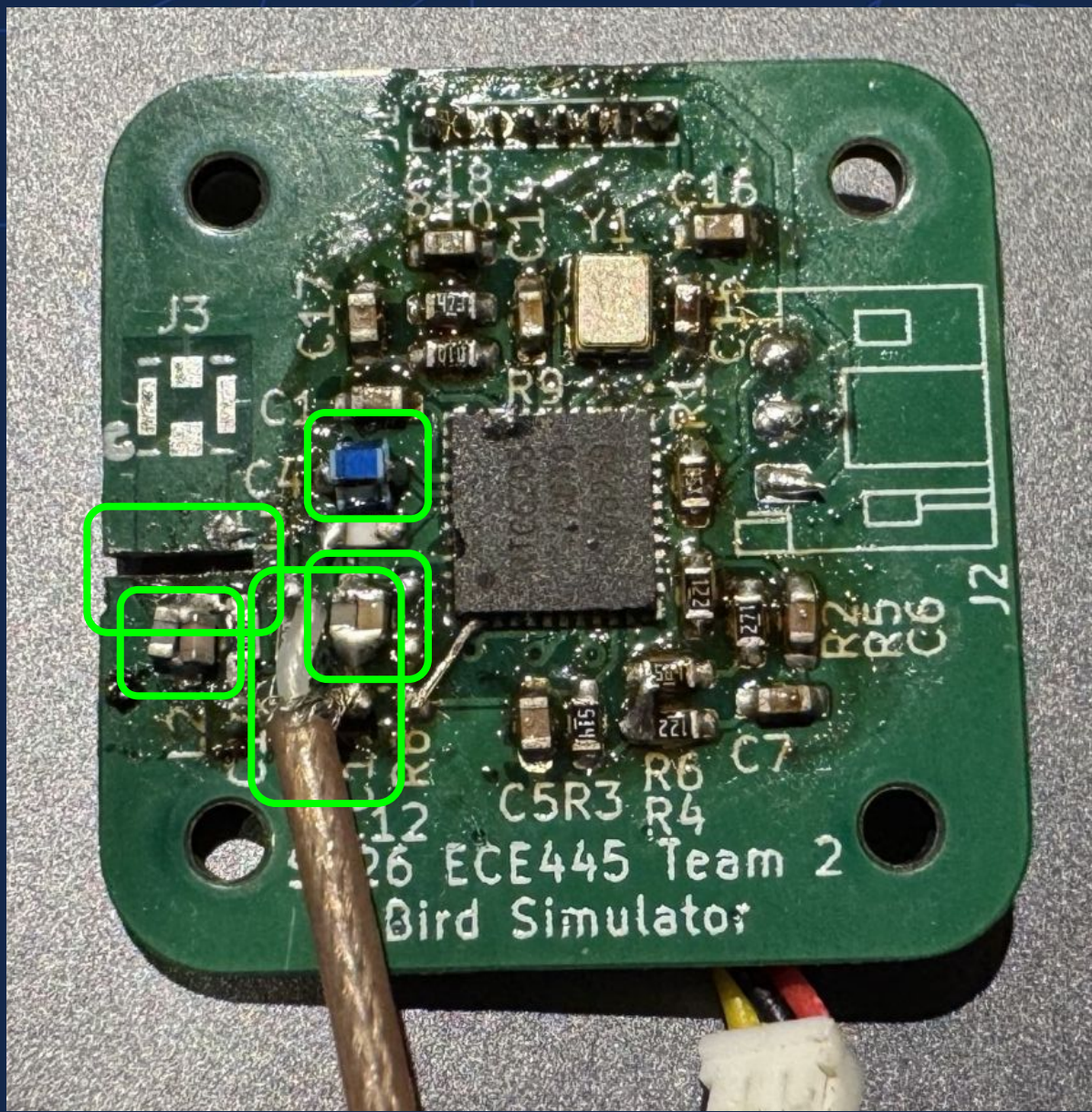
Video Transmitter (VTX)



- J1 to receive power and SPI signals from FC
- RTC6705 for 5.8GHz video transmission
- 50 Ω impedance matching
- Connector for external camera
- Software
 - Half-duplex SPI w/ 25 bit commands
 - GPIO “bit-banged”

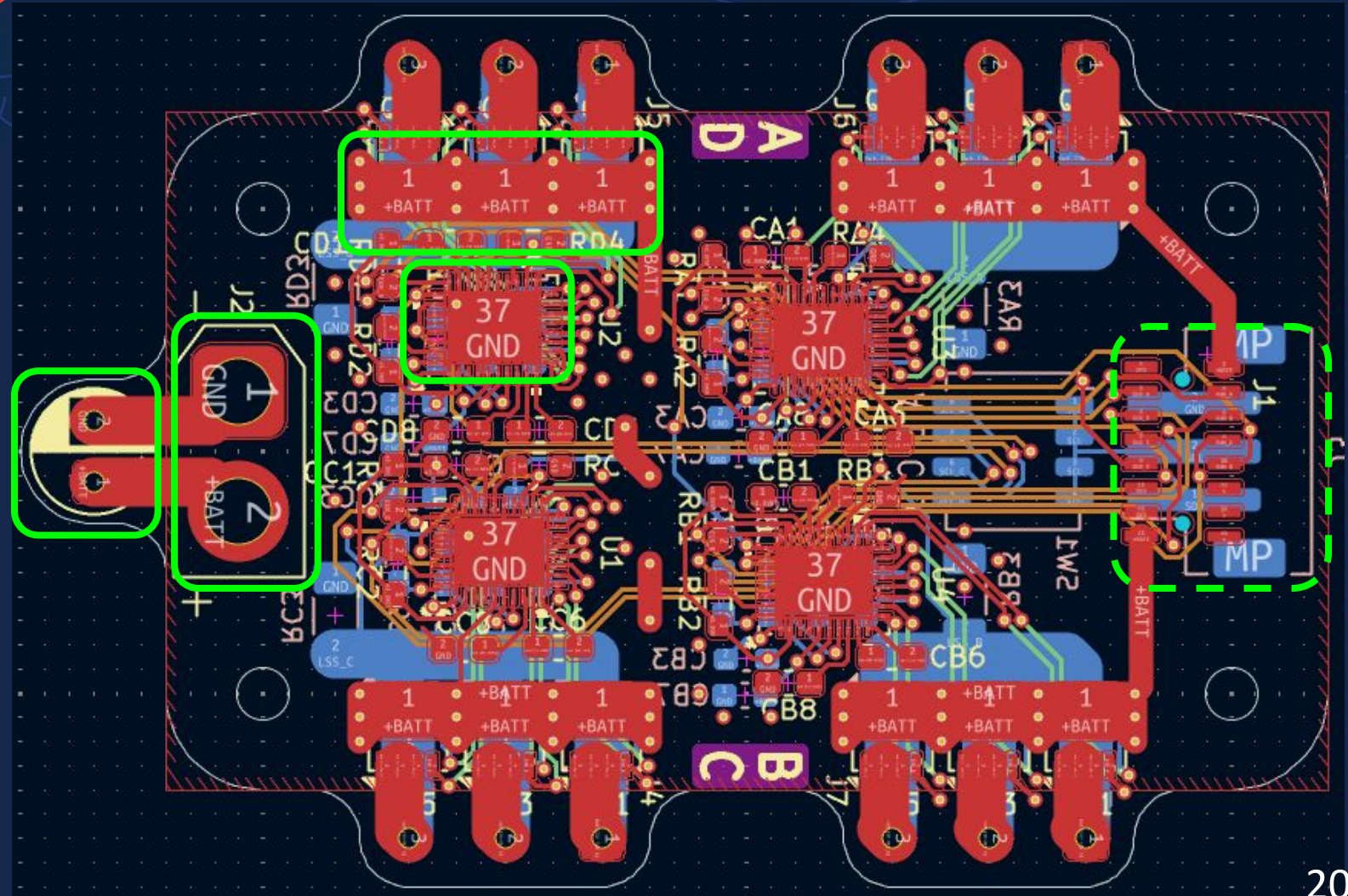
VTX Troubleshooting

- Transmission line trace too wide
 - Cut through trace
 - Solder antenna directly
- Pi network
 - Simulate using Qucs and ngspice
 - Acquired desired values w/ existing parts



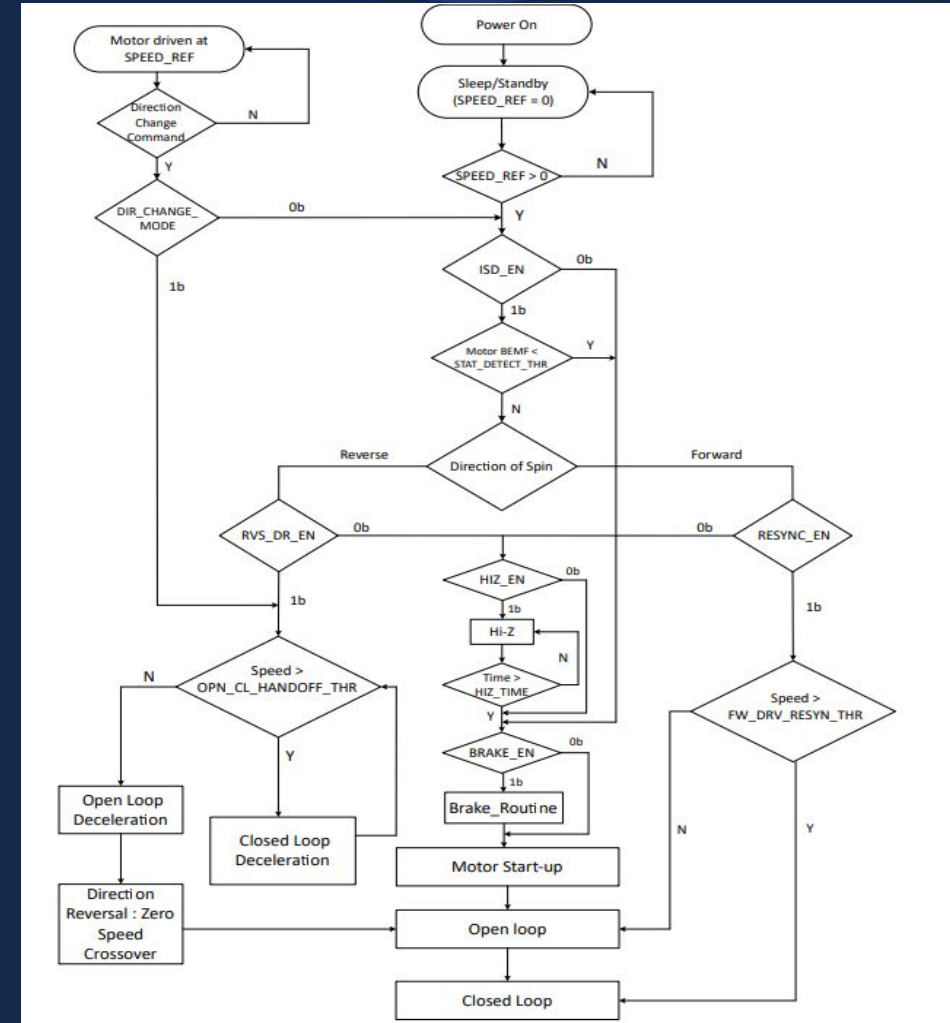
Motor Controller

- 4 × TI MCT8329A Sensorless Trapezoidal three-phase gate driver
- 24 SiSHA14DN MOSFETs for three-phase H-bridge driving
- External connection for I2C programming
- Direct XT30 battery connection
- 560 μ F capacitor
- 4 layers



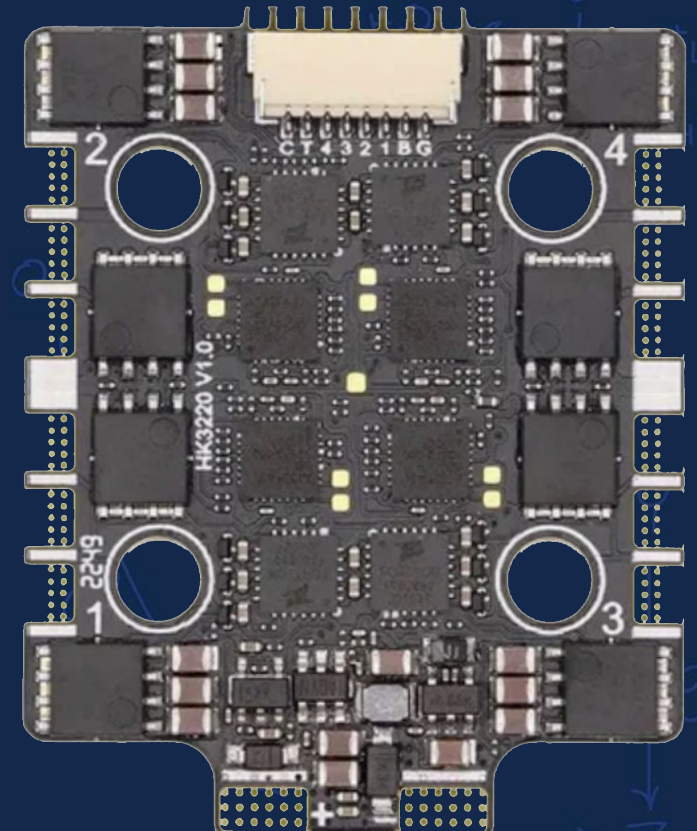
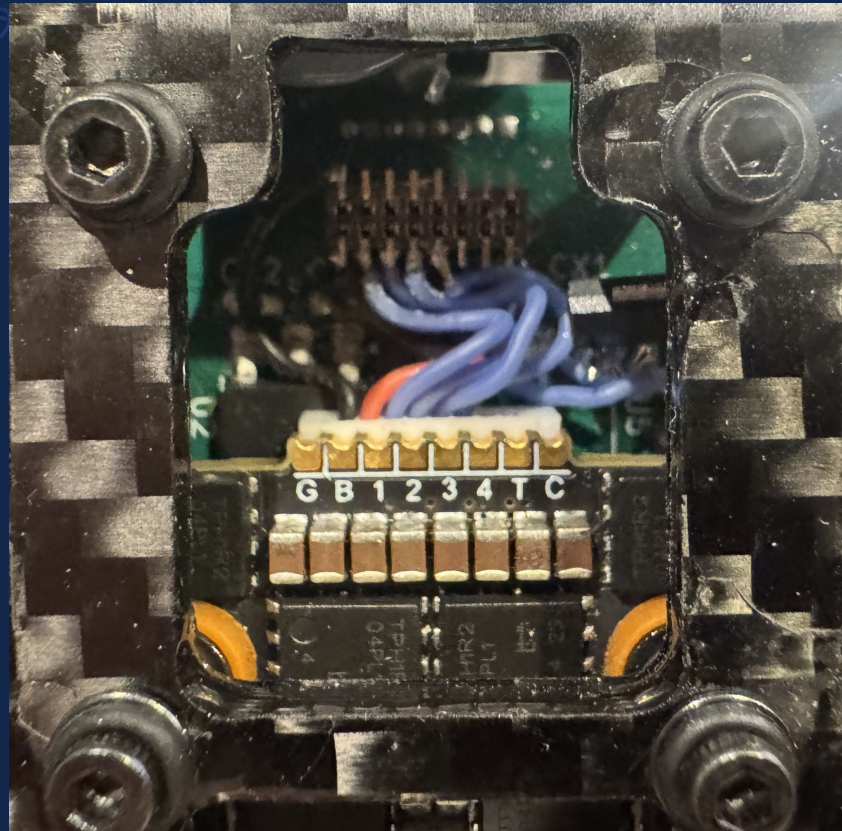
Motor Controller Programming I

- Programming individual gate drivers via I2C transaction protocol
- Control word + data payload for register access
- Motor control behavior configured through register tuning
- Final configuration is stored using EEPROM writes
- Device operates alone after EEPROM programming
- Motor startup, PWM frequency, brake time



Motor Controller Replacement ^I

- Off-the-shelf 4-motor driver
- Takes PWM input through the same connector as previous motor controller
- Passes battery power through to flight controller
- Same 20x20mm mounting grid, drop-in replacement



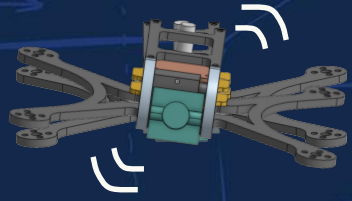
Motor Output

- Controlled by **50 Hz PWM** , **5-10% duty cycle**
- Speed measured by tachometer
- Speed range: **3.2k - 60k RPM**
- Motor 1 slower speed
 - Compensated in software

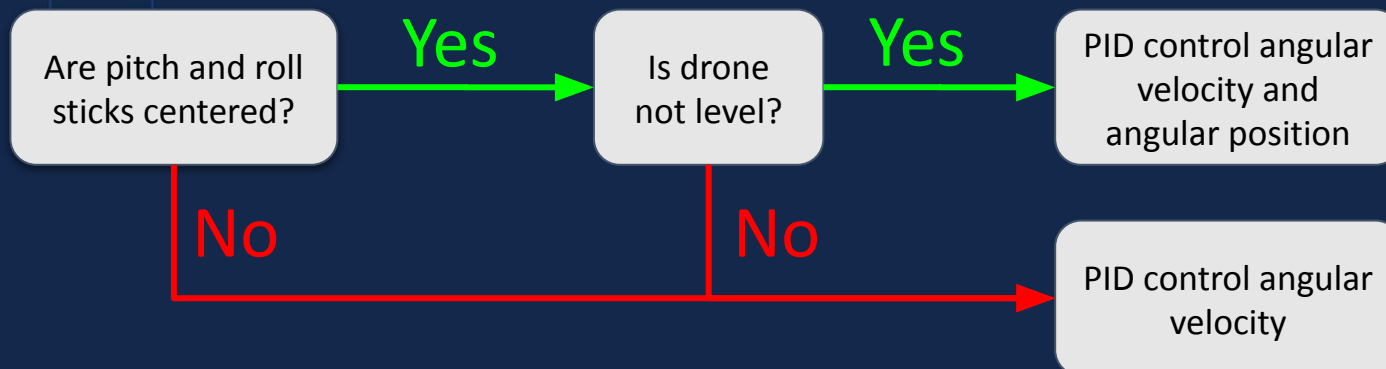
Motor 2 Speed (RPM)	Motor 1 Speed (RPM)
3250	3150
4600	4500
5700	5600
7100	7000



PID Tuning

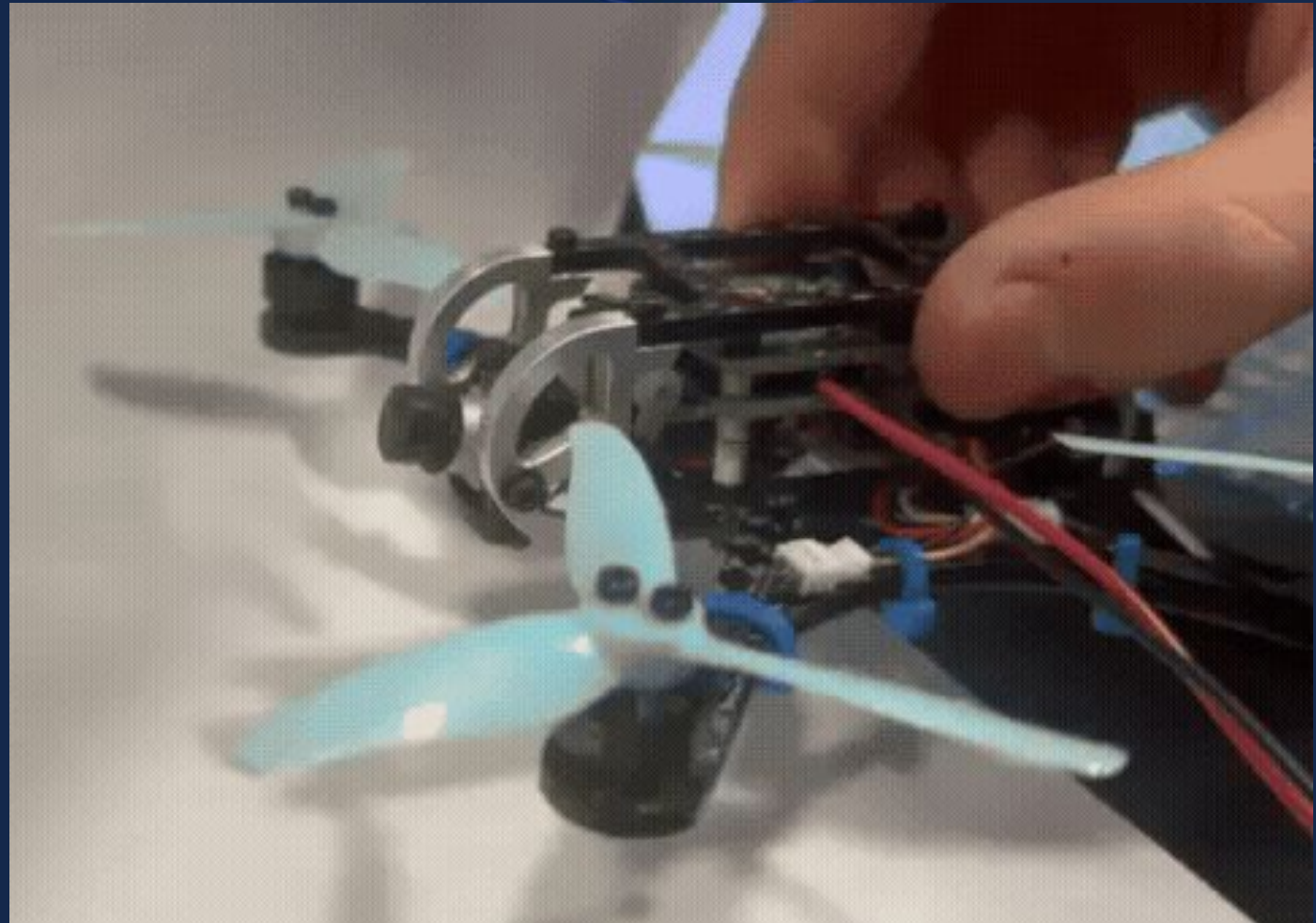


1. Tune K_p until axis overshoots and oscillates
2. Add K_d until overshoots stop
3. Add K_i if steady-state error is present
4. Tune low-pass filter if signal is noisy



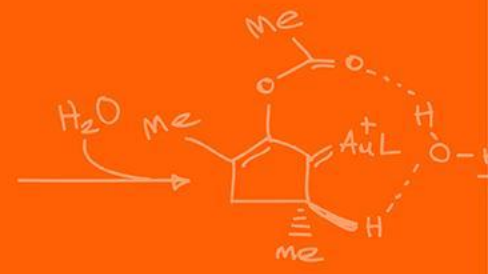
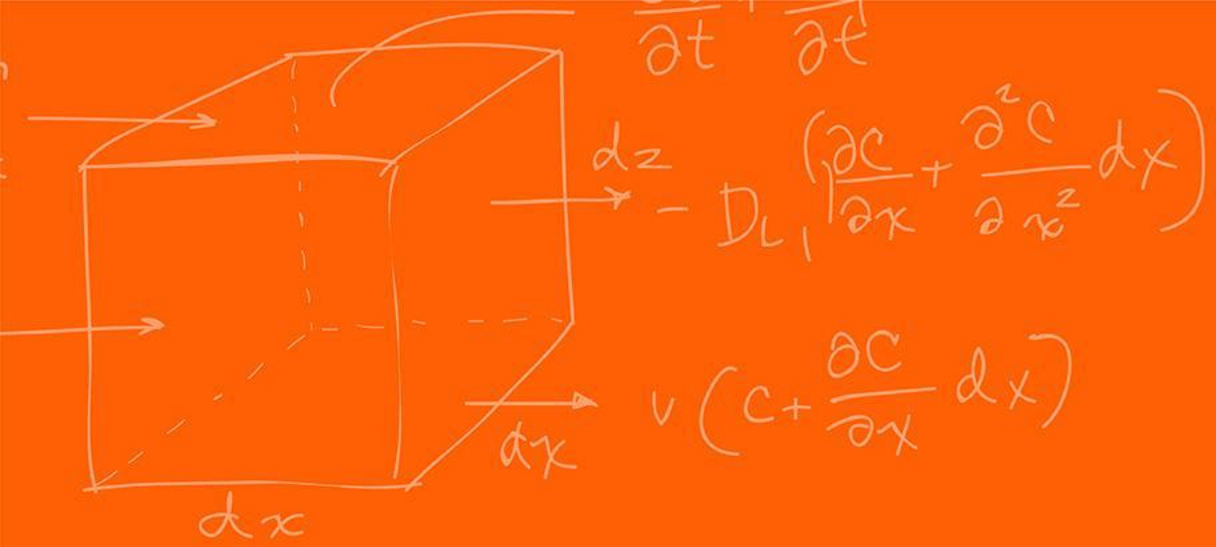
Camera Pitch Control

- Drone pitch found by integrating angular velocity
- Servo used to control pitch of camera through 5V LDO
- Bird camera input added to allow pilot to look up and down

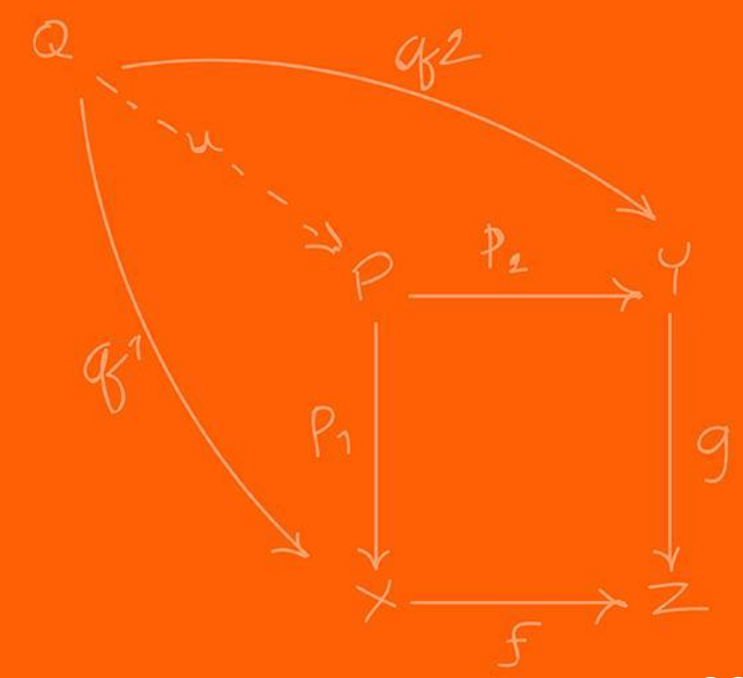


Latency

- FC:
 - Measured using HAL_GetTime()
 - SPI clock speed: **4 MHz**
 - **1-2 ms** per IMU read
 - **10 ms** per radio packet
 - Negligible time for calculations
 - Total: **12 ms** per packet received
 - Frequency: **~85 Hz**
- Simulator:
 - Measured using performance.now()
 - Frames update at least every **16.7 ms**
 - Frequency: **>60 Hz** (max display refresh rate)
- Suit:
 - SPI clock speed: **1 MHz**
 - 1 ms per IMU read x4 = **4 ms**
 - HAL_Delay(10) = **10 ms**
 - Total: **14 ms** per packet sent
 - Frequency: **~71.5 Hz**

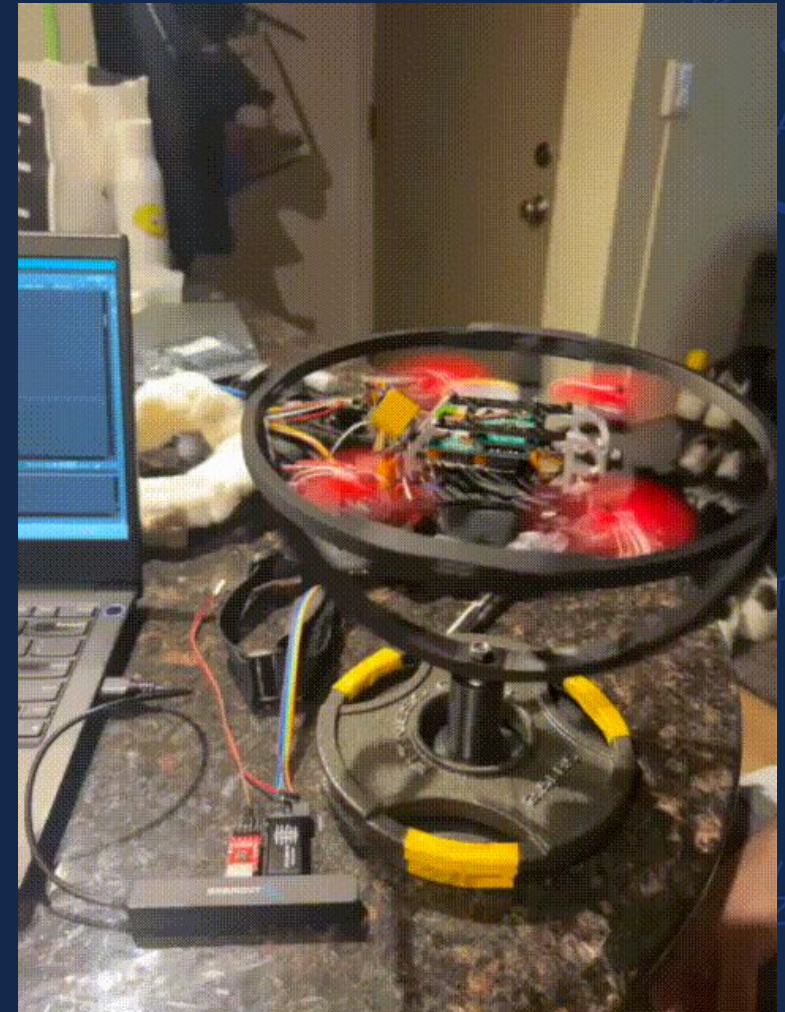


Results



Challenges

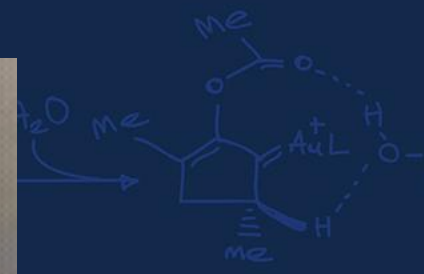
- Bird Suit SPI speed
 - Solved by increasing prescaler value while keeping timing within spec
- Bad radio packets received
 - Solved with CRC byte to ignore transmission errors
- Open-loop motor speed
 - Solved by measuring with tachometer, scaling motor values to match
- VTX improperly tuned
 - Solved by sawing board, simulating for correct parameters, and adjusting board



Successes



Successes



Successes



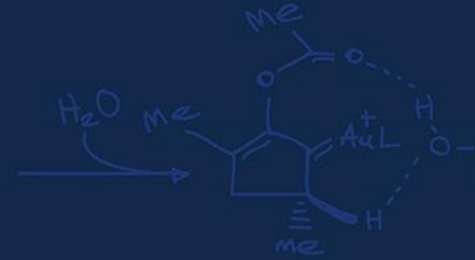


Arne Fliflet (prof)

Sound like a lot to do. Drones designed in this class have a habit of not getting off the ground.

Future Work

- Custom motor controller
- Adaptations for real world use
 - More sensitive and accurate motion tracking
 - Motion tracking tuned for detecting human reflexes
- General improvements
 - Better stabilization/autoleveling capabilities
 - Safety mechanisms for loss of control
 - Better impedance matching for RF chips



Drone Test Conditions

Condition	Requirement	Results
Motor Control	Motors spin up to 5000RPM, with variable, controllable speed	Custom motor controller is not functional in final assembly, but off-the-shelf controller meets requirements and is controlled by FC
Camera Levelling	Camera is kept near level with the ground	
Drone Flight Control	Drone measures pan, tilt, and roll, and adjusts flight controls to keep stable flight control	IMU data is measured correctly and flight control is stable in the air
Video Transmitter	Video signals are transmitted, and received with RSSI value >60dB@10m	
Mechanical Structure	Drone is fully functional after 10-foot drops	Drone has been dropped from 10 feet, and endured many harsh crashes while remaining functional
Emergency Shutoff	Motors are shut off if no control signal is received for 2 seconds	Motors shut off when neither controller nor bird suit is powered.