Bird Simulator

Anthony Amella

Emily Liu

Eli Yang

Design Document for ECE 445, Senior Design, Spring 2026

TA: Shiyuan Duan

24 February 2026

Project No. 2

Contents

1. Introduction

1.1 Problem

This project addresses the limitations of current FPV drone systems, which can create an immersive visual experiment, but lacks the ability to allow for physical control through the human body through motion cues or body orientation. This results in an experience with a realism factor for people who want an even more exhilarating experience.

Beyond entertainment, FPV drones are very useful in situations where it is not feasible to place humans directly in the environment, such as search-and-rescue operations, disaster response, or other hazardous settings. In these high-risk scenarios, rapid situational awareness is critical. Human reflexes naturally respond to perceived threats by instinctively moving their head or body to shift attention, for example, ducking when approaching an obstacle, or turning your head when there is movement in your peripheral vision. These reflexive movements are much faster than the conscious decision to move your hands to control the sticks to reorient the drone to face the threat. Combined with traditional stick control, human reflexes can be leveraged to create a more responsive tool valuable for applications where quick response time is essential.

1.2 Solution

To address this problem, we will design and implement a bird-inspired FPV drone system that allows the pilot to control flight using full-body motion rather than relying on a traditional controller. Human movements such as arm motion, head orientation, and torso tilt will be captured through multiple IMUs attached to a suit. A microcontroller will then read the data from the IMUs to convert them to instructions transmitted by a 2.4GHz transceiver that the drone receives and uses to maneuver in its environment. The pilot will be able to see what the drone camera sees through FPV goggles, which will receive analog video transmitted over 5.8GHz radio. A traditional drone controller will be made to override suit inputs and take over control in case the drone starts behaving erratically, which will also transmit signals with the 2.4GHz transceiver.

To allow development of the drone, suit, and video transmission in parallel, we will also build a drone simulator using JavaScript that can be run on a web browser. This simulator will provide a safe testing environment for validating controls and other algorithms without the risks of real-world flight, lowering the risk of injury and damage to hardware during development, and allowing more iterations for trial and error.

1.3 Visual Aid

Figure 1: Visual aid for bird simulator

1.1 High Level Requirements

- The bird suit must measure values from a set of IMUs, and transmit data over 2.4GHz to the drone, allowing for control similar to what can be done with a controller.
- The drone must receive 2.4GHz data from the bird suit or controller, and fly through the air with assisted control using the onboard IMU.
- The drone must use a camera to generate and transmit 5.8GHz FM video signals, to be received by either an off-the-shelf or custom FPV headset or screen.

2 Design

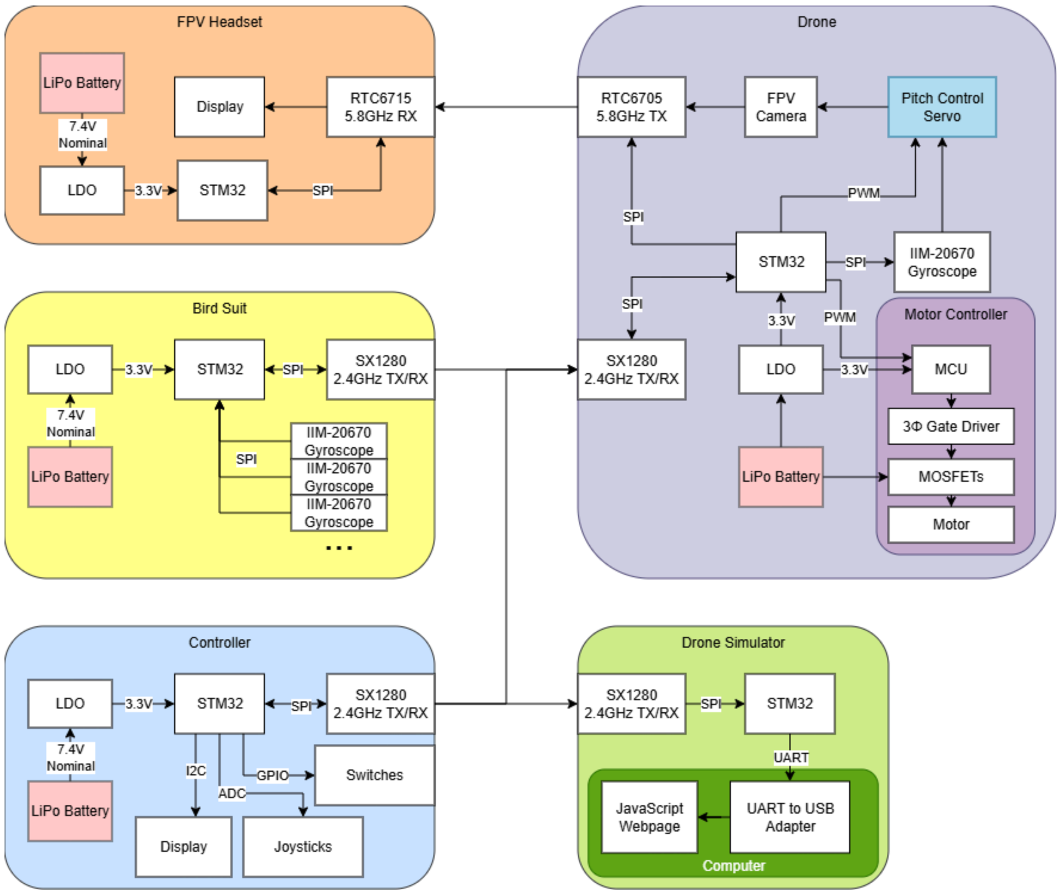

Figure 2: Block diagram for bird simulator

2.1 Subsystem 1: Drone

2.1.1 Flight Controller

The control of the drone will center around an STM32F410CBT3 microcontroller, which will communicate with the SX1280 to receive control signals from either the bird suit or the controller [3]. These signals will contain the raw inputs for the throttle, pitch, yaw, and roll of the drone. The STM32 will then use these signals, as well as the signals from the onboard IIM-20670 IMU, to generate four PWM signals for the four motor controller boards. The STM32 will also generate a PWM signal for the pitch control servo for the camera, allowing for the camera to stay level.

The flight controller has two connectors to attach to other boards. A 2x8 connector with 1mm pitch is used to communicate with the motor controller board. This connector carries battery power and ground to power the flight controller, since the battery will plug into the motor controller. The connector also carries SPI signals (MISO, MOSI, SCK), four chip select signals, used when configuring the gate drivers, as well as four PWM signals which will be used during flight to control motor speed. A 1x8 connector, also with 1mm pitch, is used to communicate with the video transmitter board. It carries 3.3v for the video transmitter chip, as well as 5v to power the camera. SPI signals are also sent, to communicate with the video transmitter, as well as the magnetometer on the auxiliary video board. These connectors will be lined up with matching connectors on the motor controller and video transmitter board, allowing for the boards to be stacked directly without any cables.

The flight controller board also contains connectors for programming the STM32, as well as communicating with the board over UART for debugging. A 3-pin servo connector is also present to provide PWM signals to control the camera levelling servo.

Two LEDs and a pushbutton are also present. The LEDs will be used to display the status of the drone, and the pushbutton will be used to zero the IMUs before flight, allowing for more stable flight control.

The footprint of the board relies on a 20x20mm grid, which is standard for existing drone parts. This will also allow for us to test flight with off-the-shelf components, such as a motor controller, while our own boards are tested.

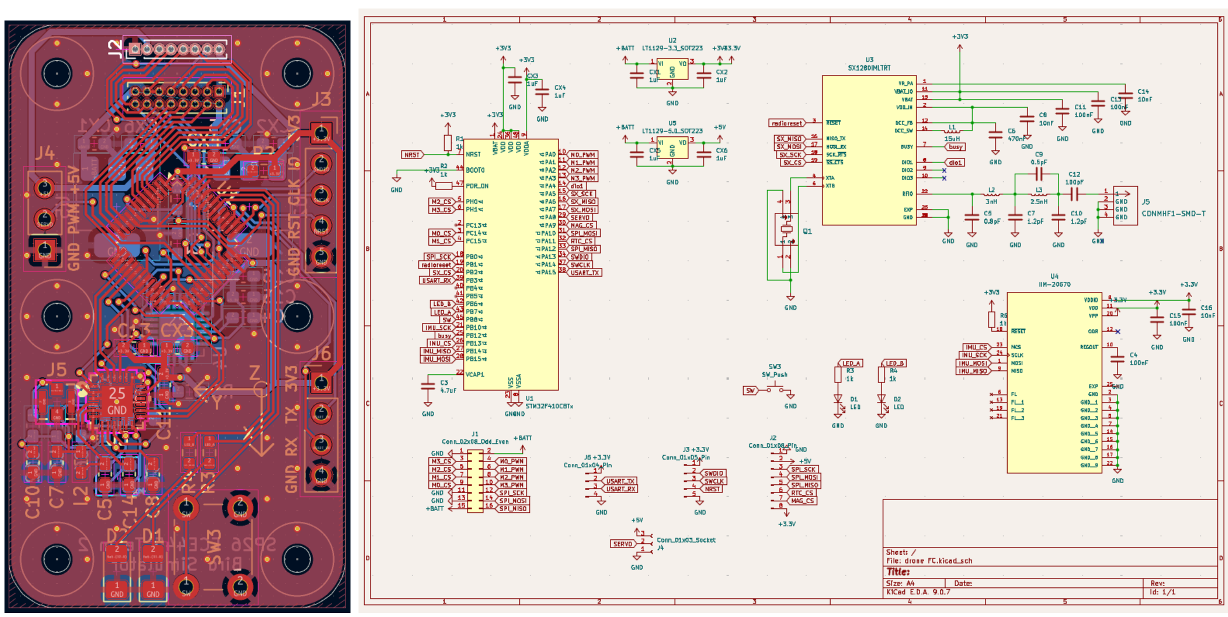

Figure 3: Schematic and PCB for drone flight controller

2.1.2 Motor Controller

The propulsion system on the drone consists of a secondary board, stacked under the drone flight controller. The motor controller board will take in power directly from the battery, which will then be passed on to the other boards on the drone. Four TI MCT8329A trapezoidal BLDC gate drivers will be present, each driving six N-channel MOSFETs to individually control the speed of each motor. The voltage divider is set up in a wye formation, which mirrors the 3-phase setup of the motor, which allows us to measure the individual phase voltages, essentially creating a back EMF measurement system. These gate drivers have non-volatile EEPROM, allowing for them to be configured on a bench and retain this configuration, so the flight controller microcontroller will only have to send a PWM signal for each of the gate drivers. Each gate driver will have a connector to communicate with a computer over I2C for this initial configuration.

2.1.3 Video Transmitter

We will use 5.8 GHz radio to transmit video data from the drone to FPV goggles using the RTC6705 transmitter module. This RF module handles amplifying, mixing, and modulating/demodulating signals and will be configured and programmed by the microcontroller through SPI. The camera outputs analog video to be transmitted by the RTC6705.

The VTX board will have the RTC6705, as well as a magnetometer, which will assist in the drone's autoleveling. This board will connect to the flight controller board from which it receives ground, +3.3V to power the RTC6705 and magnetometer, +5V to power the camera, SPI clock and data shared between RTC6705 and magnetometer, and two chip selects. The RTC6705's RF output is transmitted through an external antenna, which will be connected to the PCB through a U.FL connector. To ensure impedance matching and reduce reflections through the antenna feeder line, a pi network enhanced with filters is used, and the feeder trace width is calculated using a trace impedance calculator [9]. The enhanced pi network is based on existing designs including the RTC6705 [10].

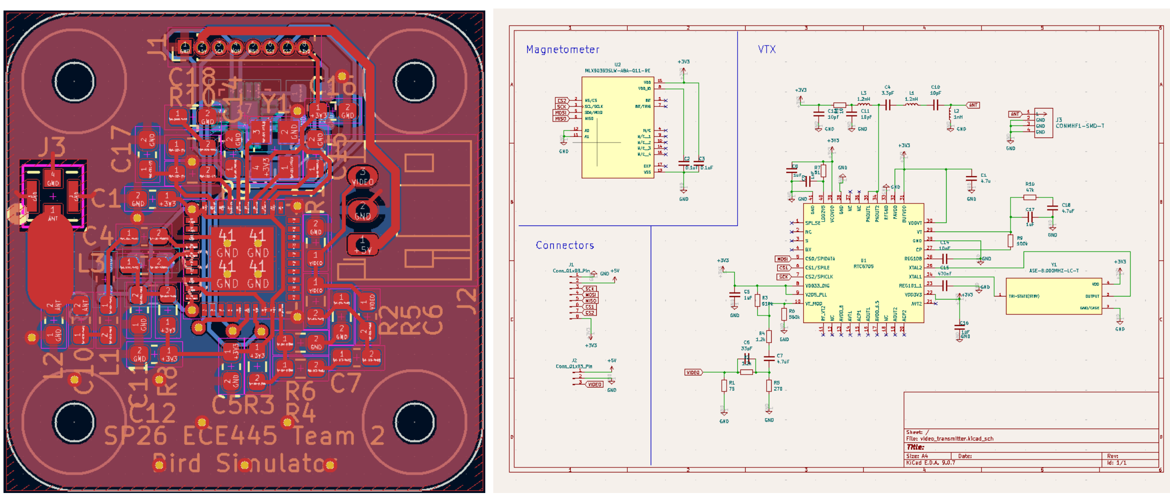

Figure 4: PCB and Schematic for video transmitter

2.1.4 Autoleveling

The drone will be equipped with an on board gyro (IIM-20670) to determine its orientation during flight. A MLX90393 magnetometer is also used for measuring absolute heading. This orientation information is used to generate motor commands that match the drone flight with the movement of the bird suit, as well as account for turbulence that may cause the drone to lose control. The autoleveling control will be implemented using a PID control loop on the microcontroller, which will calculate an error signal between the desired orientation and the current orientation, and generates corrective adjustments to the motor PWM signals.

2.1.5 Mechanical Design

The drone body will be made primarily from carbon fiber sheets, with a tentative thickness of 3mm. There will be one large body frame at the bottom of the drone, where the motors and PCBs attach, as well as a smaller top frame, which the battery will be attached to. Aluminum pieces will be used to connect the two carbon fiber frames, as well as hold the camera. The aluminum and carbon fiber parts will be manufactured with a waterjet.

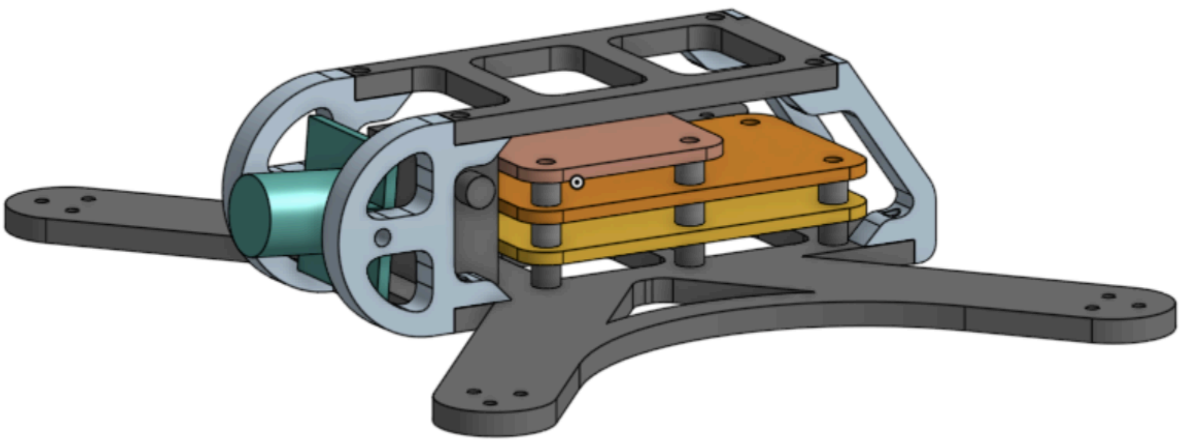

Figure 5: Drone frame CAD

(Yellow: Motor controller, Orange: Flight controller, Red: Video transmitter, Green: Camera, Blue: Aluminum, Gray: Carbon Fiber)

Each of the three PCBs will be designed with matching mechanical frames, so they can be stacked directly on top of each other, with a mounting grid that can also support potential off-the-shelf components. A servo is present directly behind the camera, which will be used to control the pitch of the camera.

2.1.6 Requirements and Verification

Component	Requirement	Verification
Motor Control	Motor is able to spin at a maximum speed of at least 5000RPM. The speed of the motor can vary according to a PWM signal by a factor of at least 5 (e.g. if the max speed is 5000RPM, the motor can spin as low as 1000RPM)	A slow-motion camera is used to capture the motor while it runs at the maximum and minimum speed. Camera frames are counted to calculate RPM
Camera Levelling	Camera is continuously kept within 10 degrees from level with the ground, across a maximum camera tilt range of at least 90 degrees.	Drone is held in the air, and moved across a range of acceptable angles. Camera pitch is measured with iPhone Measure app.
Drone Flight	Drone is able to measure pan, tilt, and	Drone is held in the air, and

Control	roll from IMUs, and individually adjust motor speeds to move towards a more level position, within a maximum range of +/- 30 degrees in the tilt and roll axes.	moved across a range of angles. A slow-motion camera is used to capture motor speeds, which must attempt to compensate for movement, such that the drone tends towards a level position.
VTX	Transmit a signal that can be received by the receiver with a RSSI value of -60 dB or stronger within a radius of 10 m	Read the RSSI value on the receiver headset and ensure that it is greater than -60 dB with the transmitter 10 m away
Mechanical Structure	Drone is fully functional after 10-foot drops on a rigid floor at any angle. No significant damage is done and all above requirements are still met.	Drone is held 10 feet in the air, verified with a tape measure, and dropped at a wide range of angles.
Emergency Shutoff	If the drone does not receive any control signal for 2 seconds, power will be cut off from the motors.	Some input device will initially control drone, before they are powered off. A timer is used to ensure the drone shuts off in time.

2.2 Subsystem 2: Bird Suit

There will be 4 IIM-20670 modules embedded in a wearable suit that will collect data to be combined and used to determine the motion and orientation of the user: one on each arm, one on the back of the head, and one on the back (in between the shoulder blades). The IIM-20670 modules can capture gyroscope and accelerometer information, which is communicated to the microcontroller through SPI [7]. Movements such as head rotation, wing flapping, body orientation will be translated to stick inputs on a drone controller. This information will be organized into a series of bytes, sent to the 2.4GHz transceiver chip (SX1280), and transmitted over radio to the drone. The byte order will be discussed in a later section (2.5 Subsystem 5: Controller). This subsystem is powered with a LiPo battery with a 7.4V nominal voltage, which will be stepped down to 3.3V by a LDO.

To extract meaningful body movement and orientation from the raw gyroscope and accelerometer data collected by the IIM-20670 modules, several mathematical processing steps are required. At a high level, we will need to use linear algebra to manipulate coordinate planes and transformations, filtering to combine information from multiple sources of information, and other scaling and mapping techniques to achieve a realistic response. We will assign drone input controls to human movement as follows:

1. Throttle: frequency and amplitude of wing flapping determines amount of throttle applied
2. Pitch: angular displacement of the torso in the sagittal plane from the neutral position determines the angle of the drone’s pitch
3. Yaw: angular displacement of the head in the transverse plane from the neutral position determines the rate of yaw

4. Roll: angular displacement of the arms in their extended position in the coronal plane from the neutral position determines the angle of roll
5. Camera pitch: angular displacement of the head in the sagittal plane from the neutral position determines the angle of camera pitch

2.2.2 Requirements and Verification

Component	Requirement	Verification
Wing flapping	Flapping both arms in phase will be measured by arm IMUs, which will be reflected by increasing the throttle sent. Each flap must give a clearly noticeable, positive change in vertical velocity according to the intensity of the flap.	Bird suit will transmit to bird simulator, which will clearly reflect drone flight based on IMU movement. Bird simulator allows for inputs to be more obvious, since there will be no wind or issues with flight control
Wing rolling	Keeping both arms parallel, and changing their angle from normal with the sagittal plane will cause linearly proportional changes in the roll of the drone	
Speed control	Leaning forwards or backwards with the body IMU changes the speed of the drone, proportionally to the angular displacement from the coronal plane	
Camera pitch	The angular displacement between the coronal plane and the head IMU causes the drone camera to pitch the same angle, with a maximum error of 10 degrees, within the range of motion of the camera.	Bird suit will transmit to drone. Head and camera angular displacement will be measured with iPhone Measure app.

2.3 Subsystem 3: FPV Headset

In the initial development phase, we will use off-the-shelf FPV goggles to receive the analog video transmitted by the drone's 5.8GHz transmitter. The RTC6705 uses standard 5.8GHz analog FM modulation [4], which is the same protocol that is used by many existing FPV goggles. If time permits, we will design our own FPV goggles, which involve the RTC6715 5.8GHz receiver [5], and either a composite-to-HDMI converter, and a small HDMI compatible LCD screen, or a composite video compatible LCD screen to reduce latency. Due to time constraints and complexity, this is a significant reach goal for this project, so no requirements will be listed.

2.4 Subsystem 4: Drone Simulator

This subsystem is a continuation of the Flight MP from CS418, which involved using JavaScript and WebGL to randomly generate terrain and take user input from the keyboard to move a camera around the terrain to simulate flight. To make the movement feel more drone-like, the controls will be remapped to a drone controller's sticks, and physics will be added to mimic real-world flight. Further development of the simulator will include a 2.4GHz receiver that can either receive information from the suit or controller which will be read by the computer running the simulation through a UART to USB adapter, and improved graphics by using existing 3D models rendered using Three.js. With the ability to easily modify the backend of this simulator, it is a better tool for the development of the bird simulator than a closed-source off-the-shelf flight simulator.

2.4.1 Requirements and Verification

Component	Requirement	Verification
Control	Given stick inputs from a drone controller, the simulator must receive each input and convert them to a value between -1 and 1, representing stick position.	Each command (throttle, yaw, pitch, roll) from the controller results in an appropriate camera movement in the simulator.
Latency	To ensure realistic and responsive control, the time between receiving a packet and calculating/rendering the simulated response should be less than 10 ms.	A timer will be added to the software to track time between when the simulator receives the packet to when it is finished processing the data.

2.5 Subsystem 5: Controller

The traditional drone controller consists of two switches, two joysticks, a display, and button/joystick, and is powered by a LiPo battery with a 7.4V nominal voltage, which will be stepped down to 3.3V by a LDO. The microcontroller reads the joystick values with an ADC to be converted to throttle, pitch, roll, and yaw values, which sends the data to the RF transceiver (SX1280) to transmit over 2.4GHz radio. One of the switches will arm the drone, and the other switch will cause the controller input to override suit input. The drone will not take any input from the suit unless the controller's arm switch is toggled, so that the drone has a backup control ready at all times. When override is toggled, the suit's transceiver will receive this signal and stop transmitting. The information is sent continuously in packets, which organize the data in this order, with each element being represented by one byte:

[Arm][Override][Throttle][Yaw][Pitch][Roll][empty][empty]

The controller is equipped with a LCD screen (DOGM204-A 4x20) that interfaces with the microcontroller over I2C that displays a cursor which can be controlled by the small button/joystick [8]. This can be used to adjust controls, parameters, etc. without having to reprogram the drone's flight controller. This controller was designed in Fall 2025.

2.5.1 Requirements and Verification

Component	Requirement	Verification
Control override	Controller signals will take	Bird suit will initially give signals.

	precedence over bird suit signals. If the controller is armed, the drone will ignore bird suit signals.	Once the controller is armed, the drone must behave according to exclusively controller inputs.

2.6 Tolerance Analysis

The most significant risk to successful completion of the project is the latency and accuracy of the IMU data processing. The microcontroller must be able to obtain data from each of the four gyroscope modules, process and translate the inputs, and transmit the data fast enough so that the drone receives enough information to maintain stable flight. This risk can be mathematically analyzed to demonstrate that it is feasible.

A drone feels responsive to a traditional joystick controller when data packets are transmitted with a frequency of greater than 50 Hz, or one packet every 20 ms. This means that for the drone to feel responsive and stable, the suit must transmit its data at this frequency or faster.

Since each gyro module provides 12 bytes of raw data, the microcontroller receives 48 bytes of data per polling. Assuming a SPI clock with a frequency of 10 MHz, the time to read 48 bytes is:

$$T_{poll} = \frac{1\ second}{10*10^{6}\ bits} \cdot \frac{8\ bits}{1\ byte} \cdot 48 = 28.8\ \mu s$$

which, compared to the time constraint of 20 ms, is negligible.

After polling the gyros, the raw data needs to be processed and translated to drone inputs. The processing consists mainly of filtering and manipulation of matrices and vectors, which can be executed very efficiently using optimized C libraries. The total processing time can be estimated to be on the order of microseconds, which again, compared to the time constraint of 10 ms, is negligible.

Finally, data will be transmitted by the RF module using LoRa modulation, which can transmit 8 bytes of data in 10 ms. The total time from polling the gyros, processing the data, and transmitting over radio is less than 20 ms, which means the drone will be able to receive input frequently enough to maintain responsive and stable flight. If the drone response feels sluggish, the time over air can be decreased by changing the SX1280 to use much faster GFSK modulation instead.

3. Cost and Schedule

3.1 Cost Analysis

3.1.1 Bill of Materials

Part Description	Mfr	Board	Price	Qty	Total
100pF Cap, 0603 SMD	Murata	Drone FC	$0.03	1	$0.03
4.7uF Cap, 0603 SMD	KEMET	Drone FC	$0.15	1	$0.15

100nF Cap, 0603 SMD	TDK	Drone FC	$0.02	4	$0.08
0.8pF Cap, 0603 SMD	Murata	Drone FC	$0.05	1	$0.05
470nF Cap, 0603 SMD	Samsung	Drone FC	$0.04	1	$0.04
1.2pF Cap, 0603 SMD	TDK	Drone FC	$0.07	2	$0.14
10nF Cap, 0603 SMD	KEMET	Drone FC	$0.02	3	$0.06
0.5pF Cap, 0603 SMD	Murata	Drone FC	$0.06	1	$0.06
1uF Cap, 0603 SMD	TDK	Drone FC	$0.10	6	$0.60
LED, 1206 SMD	Wurth	Drone FC	$0.25	2	$0.50
Conn 02x08, 1.00mm P	GCT	Drone FC	$4.59	1	$3.50
Conn 01x05 Socket, 2.54mm P	DuPont	Drone FC	$0.85	1	$0.85
Conn 01x03 Header, 2.54mm P	DuPont	Drone FC	$0.55	1	$0.55
CONMHF1-SMD-T	Hirose	Drone FC	$1.20	1	$1.20
Conn 01x04 Header, 2.54mm P	DuPont	Drone FC	$0.70	1	$0.70
Conn 01x08 Header, 1.00mm P	GCT	Drone FC	$1.22	1	$1.22
15uH Inductor, 0805 SMD	Coilcraft	Drone FC	$0.45	1	$0.45
3nH Inductor, 0603 SMD	Coilcraft	Drone FC	$0.30	1	$0.30
2.5nH Inductor, 0603 SMD	Coilcraft	Drone FC	$0.32	1	$0.32
52MHZ Crystal, NX2016SA	NDK	Drone FC	$2.80	1	$2.80
1k Resistor, 0603 SMD	Yageo	Drone FC	$0.01	5	$0.05
SW Push Button, THT	E-Switch	Drone FC	$0.40	1	$0.40
STM32F410CBTx MCU	STMicro	Drone FC	$4.50	1	$4.50
LT1129-3.3 LDO, SOT-223	Linear	Drone FC	$1.80	1	$1.80
SX1280 IMLTRT RF Chip	Semtech	Drone FC	$3.20	1	$3.20
IIM-20670 Gyro	InvenSense	Drone FC	$5.50	1	$5.50
LT1129-5.0 LDO, SOT-223	Linear	Drone FC	$1.75	1	$1.75
100nF Cap, 0603 SMD	TDK	Gyro	$0.02	2	$0.04
10nF Cap, 0603 SMD	KEMET	Gyro	$0.02	1	$0.02
Conn 01x07, 2.00mm P JST	JST	Gyro	$1.10	1	$1.10
IIM-20670 Gyro	InvenSense	Gyro	$5.50	1	$5.50
100pF Cap, 0603 SMD	Murata	Suit	$0.03	1	$0.03
4.7uF Cap, 0603 SMD	KEMET	Suit	$0.15	1	$0.15

100nF Cap, 0603 SMD	TDK	Suit	$0.02	4	$0.08
0.8pF Cap, 0603 SMD	Murata	Suit	$0.05	1	$0.05
470nF Cap, 0603 SMD	Samsung	Suit	$0.04	1	$0.04
1.2pF Cap, 0603 SMD	TDK	Suit	$0.07	2	$0.14
10nF Cap, 0603 SMD	KEMET	Suit	$0.02	3	$0.06
0.5pF Cap, 0603 SMD	Murata	Suit	$0.06	1	$0.06
1uF Cap, 0603 SMD	TDK	Suit	$0.10	4	$0.40
LED, 1206 SMD	Wurth	Suit	$0.25	3	$0.75
Conn 01x07, 2.00mm P JST	JST	Suit	$1.10	3	$3.30
Conn 01x02 XT30PW-F	AMASS	Suit	$1.50	1	$1.50
CONMHF1-SMD-T	Hirose	Suit	$1.20	1	$1.20
Conn 01x04 Header, 2.54mm P	Molex	Suit	$0.70	1	$0.70
Conn 01x05 Socket, 2.54mm P	TE Conn	Suit	$0.85	1	$0.85
15uH Inductor, 0805 SMD	Coilcraft	Suit	$0.45	1	$0.45
3nH Inductor, 0603 SMD	Coilcraft	Suit	$0.30	1	$0.30
2.5nH Inductor, 0603 SMD	Coilcraft	Suit	$0.32	1	$0.32
52MHZ Crystal, NX2016SA	NDK	Suit	$2.80	1	$2.80
1k Resistor, 0603 SMD	Yageo	Suit	$0.01	6	$0.06
SW Push Button, THT	E-Switch	Suit	$0.40	3	$1.20
STM32F410CBTx MCU	STMicro	Suit	$4.50	1	$4.50
LT1129-3.3 LDO, SOT-223	Linear	Suit	$1.80	1	$1.80
SX1280 IMLTRT RF Chip	Semtech	Suit	$3.20	1	$3.20
IIM-20670 Gyro	InvenSense	Suit	$5.50	1	$5.50
4.7u Cap, 0603 SMD	TDK	VTX	$0.12	1	$0.12
10pF Cap, 0603 SMD	Murata	VTX	$0.04	3	$0.12
10nF Cap, 0603 SMD	KEMET	VTX	$0.02	1	$0.02
470nF Cap, 0603 SMD	Samsung	VTX	$0.04	1	$0.04
0.1uF Cap, 0603 SMD	Murata	VTX	$0.03	2	$0.06
3.3pF Cap, 0603 SMD	TDK	VTX	$0.06	1	$0.06
1uF Cap, 0603 SMD	KEMET	VTX	$0.10	4	$0.40
33pF Cap, 0603 SMD	Murata	VTX	$0.05	1	$0.05

4.7uF Cap, 0603 SMD	KEMET	VTX	$0.15	2	$0.30
1nF Cap, 0603 SMD	TDK	VTX	$0.03	1	$0.03
Conn 01x08 Header, 1.00mm P	GCT	VTX	$1.60	1	$1.60
Conn 01x03, 2.00mm P JST	JST	VTX	$0.90	1	$0.90
CONMHF1-SMD-T	Hirose	VTX	$1.20	1	$1.20
1.2nH Inductor, 0603 SMD	Coilcraft	VTX	$0.35	2	$0.70
1nH Inductor, 0603 SMD	Coilcraft	VTX	$0.38	1	$0.38
75 Resistor, 0603 SMD	Yageo	VTX	$0.01	1	$0.01
47k Resistor, 0603 SMD	Yageo	VTX	$0.01	1	$0.01
1.2k Resistor, 0603 SMD	Yageo	VTX	$0.01	2	$0.02
510k Resistor, 0603 SMD	Yageo	VTX	$0.01	1	$0.01
270 Resistor, 0603 SMD	Yageo	VTX	$0.01	1	$0.01
560k Resistor, 0603 SMD	Yageo	VTX	$0.01	1	$0.01
51 Resistor, 0603 SMD	Yageo	VTX	$0.01	1	$0.01
10 Resistor, 0603 SMD	Yageo	VTX	$0.01	1	$0.01
100k Resistor, 0603 SMD	Yageo	VTX	$0.01	1	$0.01
RTC6705 VTX	Richwave	VTX	$6.50	1	$6.50
MLX90393SLW	Melexis	VTX	$7.50	1	$7.50
8.000MHZ XTAL	Abracon	VTX	$1.50	1	$1.50
MCT8329A	Texas Instruments	ESC	$4.00	4	$16.00
SiSHA14DN	Vishay Semiconductor	ESC	$1.01	24	$24.24

These part totals sum to $121.08. We plan to buy at least triple the required part quantities, to account for component loss, which brings the sum to $283.24. Accounting for extra parts that will be purchased in the future, we estimate the total parts cost to be $400.

3.1.2 Labor Costs

Assuming a reasonable salary, similar to what a graduate from ECE at Illinois might typically make, the total labor cost can be estimated by multiplying the hourly cost by the number of hours worked. The average ECE graduate from the University of Illinois has a starting salary of approximately $95,000 [11]. Assuming each of our team members works an average of 10 hours per week, across the 16-week semester, the total labor costs can be calculated as follows:

$$3 \times 10 \times \frac{95{,}000}{2{,}080} \times 16 = \$21{,}923$$

3.1.3 Grand Total

Summing the cost of materials and labor cost for each team member, the total cost is:

$21,923 + $400 = $22,323

3.2 Schedule

Week	Goal	Member
1-3 (1/19-2/2)	- Project idea formulation	All
4-5 (2/9-2/16)	- Details of overall project and individual subsystems finalized	All
6 (2/23)	- Design gyro, flight controller schematics and PCBs - Have boards and part orders ready for first round orders	Anthony
	- Design video transmitter schematic and PCB - Have boards and part orders ready for first round orders	Emily
7 (3/2)	- Begin writing firmware and drivers for RF transmitter, gyros, magnetometer, flight controller	Anthony, Emily
	- Design motor controller schematic and PCB - Have board and part orders ready for second round orders	Eli
8 (3/9)	- Have controller, receiver, and drone simulator ready for breadboard demo	Emily
	- Receive PCB and parts for VTX, suit, and flight controller - Assemble all boards, start testing and debugging	Anthony, Emily
9-10 (3/16-3-23)	- Receive, assemble, and test motor controller board	Eli
	- Identify issues with first design, make changes to schematic/boards if necessary and submit for fourth round orders - Individual subsystems roughly working	Anthony, Emily
11-12 (3/30-4/6)	- Improve on subsystem performance for progress demo - Assemble and test integrated system	All
13-14 (4/13-4/20)	- Have mostly working full system for mock demo - Refine final product	All
15-16 (4/27-5/4)	- Final demos and presentation	All

4. Ethics, Safety, and Societal Impact

4.1 Legal Issues

This project poses three potential legal issues; transmission on the 2.4GHz band, 5.8GHz band, as well as operating a flying object in public, outdoor airspace. While the SX1280 is FCC Part 15 certified, our use in a custom board without FCC testing and certification will require an amateur radio license [2]. Similarly, both of the 5.8GHz chips we are using for video are not FCC certified. To work around this, Emily will obtain a Technician Class Amateur Radio License (HAM License) under FCC part 97. This allows for legal use of custom radio boards without FCC certification [2].

Recreational flight in public airspace requires an FAA remote pilot license, which Eli currently possesses. This allows for flight of any object below 250 grams, without any FAA registration [6]. We expect to stay below 250 grams, but should we end up above this weight, registration of the drone is still relatively simple, inexpensive, and well within our ability. During any flight of the drone, Eli will be in control of either the bird suit or the controller, allowing him to take control of the drone at any point, should the flight pose any hazard.

4.2 Ethical and Safety Issues

In accordance with the IEEE Code of Ethics, our primary focus is keeping our project safe, both for us, as well as people in the areas where our drone would be flown. The primary safety risk is a loss of control of the drone in flight. During testing of the drone, rotating propellers and potential high velocity movement pose a safety hazard. Operating on 2.4GHz and 5.8GHz without FCC certification poses an ethical risk of interfering with other communications. The use of LiPo batteries in multiple parts of our system also poses safety risks.

4.3 Mitigation Procedures

To uphold IEEE Code of Ethics I.1 [1], we will implement an 'override' system, where a traditional controller can take control instead of the bird suit's inputs. This allows for a certified remote pilot to take control of the drone, and allow it to land safely. We are mitigating this by requiring a Technician Class Amateur Radio License (HAM License) under FCC Part 97 [2], ensuring we operate legally, within any power or frequency band limitations. To minimize physical risk while designing the drone, we are utilizing a drone simulator. This allows for us to validate the signals generated by the bird suit in a virtual environment, preventing hardware damage or injury from the drone flying erratically.

When testing drone functionality, we will maintain a safe distance from the drone and wear safety glasses. All flights will be tested in a controlled environment, whether that is indoors with safety nets and propeller guards or outdoors in open fields, far from people or public spaces. In the case of RX loss (no signal received for more than 2 seconds), the motors will be disabled, preventing further loss of control.

We will adhere to standard charging and storage protocols to prevent thermal runaway or fire hazards. This will be done by storing the batteries in battery storage containers when not in use, charging them only in open air under active supervision, and regularly monitoring the voltage of the batteries to prevent over- and under-voltage.

4.4 Societal Impact

This project is broad in its applications and potential to impact society. In this iteration, the bird simulator is purely for entertainment purposes, giving FPV pilots a new, more immersive way to explore environments using a drone. Beyond entertainment, this project explores how leveraging natural body movement and reflexes for control can lead to more intuitive and responsive ways for humans to operate remote systems. This enhanced-human machine interaction can have benefits from a safety perspective, reducing risk to humans in hazardous situations. Applications such as search-and-rescue, disaster response, and inspection of dangerous environments could benefit from faster, reflex-driven control that mimics natural human instincts, potentially reducing risk of damage and injury when reaction times matter.

4. References

[1] IEEE, "IEEE Code of Ethics," *ieee.org*, Jun. 2020. https://www.ieee.org/about/corporate/governance/p7-8.html

[2] "Federal Communications Commission FCC 25-60." Accessed: Feb. 14, 2026. [Online]. Available: https://docs.fcc.gov/public/attachments/FCC-25-60A1.pdf

[3] "SX1280/SX1281," *Mouser.com*, 2026. https://www.mouser.com/datasheet/2/761/sx1280_81-1107808.pdf?srsltid=AfmBOooDbVxLfQr3AzSAIDq3zJD_xqX2DoAVFrCTwrAqbstYnwREkYcq (accessed Feb. 14, 2026).

[4] "RTC6715 CMOS 5.8GHz Band FM Receiver," 2007. https://opendevices.ru/wp-content/uploads/2015/02/7058r.pdf

[5] "RTC6705 CMOS 5.8GHz Band FM Transmitter Product Description," 2007. Accessed: Feb. 14, 2026. [Online]. Available: https://wildlab.org/wp-content/uploads/2015/07/RTC6705-DST-001.pdf

[6] "FAA-G-8082-22 Remote Pilot -Small Unmanned Aircraft Systems Study Guide," 2016. Available: https://www.faa.gov/sites/faa.gov/files/regulations_policies/handbooks_manuals/aviation/remote_pilot_study_guide.pdf

[7] "IIM-20670 Datasheet SmartIndustrial™ 6-axis MotionTracking® MEMS Device for Industrial Applications." Accessed: Feb. 14, 2026. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2023/07/DS-000183-IIM-20670-v1.1.pdf

[8] "DOGM204-A," *Mouser.com*, 2026. https://www.mouser.com/datasheet/3/711/1/dogm204e.pdf?srsltid=AfmBOooq4qkRDof1eHDmkp5DgJaeL9aE2SEPhDB7MTqIqsARJZv0SLzp (accessed Feb. 14, 2026).

[9] "Controlled Impedance Calculator - JLCPCB," *Jlcpcb.com*, 2024. https://jlcpcb.com/pcb-impedance-calculator

[10] fishpepper, "tinyOSD & tinyFINITY – a tiny opensource video tx with full graphic OSD – fishpepper.de," *Fishpepper.de*, Mar. 21, 2019. https://fishpepper.de/2019/03/11/tinyosd-tinyfinity-a-tiny-opensource-video-tx-with-full-graphic-osd/ (accessed Feb. 27, 2026).

[11] G. E. O. of M. and Communications, "Salary Averages," *ece.illinois.edu*. https://ece.illinois.edu/admissions/why-ece/salary-averages