ECE 445
Spring 2025

Senior Design Project Proposal

Project #1: Glove controlled quad-copter

Team Members: Atsi Gupta (atsig2), Aneesh Nagalkar (aneeshn3), Zach Greening (zg29)

TA: Wenjing Song
Professor: Cunjiang Yu

1. Introduction

1.1 Problem

Controlling drones typically requires handheld remote controllers or smartphones, which can feel unintuitive and demand significant practice to master. This steep learning curve limits accessibility for new users and prevents drones from being seamlessly integrated into areas like training, entertainment, or assistive technology. Existing remote-control methods also provide little user feedback and lack robust safety mechanisms, increasing the risk of crashes or improper handling.

1.2 Solution

Our project proposes a wearable gesture-control glove that enables intuitive, ergonomic drone operation. The glove will incorporate IMU and gyroscope sensors to capture the orientation and motion of the user's hand, translating gestures into commands such as forward, backward, strafe, yaw, and stop. An ESP32 microcontroller embedded in the glove will transmit these commands wirelessly to a drone equipped with an ESP32-based flight controller.

To improve upon previous iterations of gesture-control systems, our design will:

- Replace less precise flex sensors with IMUs for more accurate gesture tracking.
- Include a gesture-based emergency shutoff for safety.
- Optionally integrate haptic feedback to communicate drone status to the user (e.g., low battery, weak signal).
- Optionally integrate an ESP32-CAM on the drone to provide basic video feedback, enhancing user situational awareness.

The camera is not essential to the core glove-to-drone control system, but serves as a scalable feature that can expand the project's applications if successfully integrated.

1.3 Visual Aid

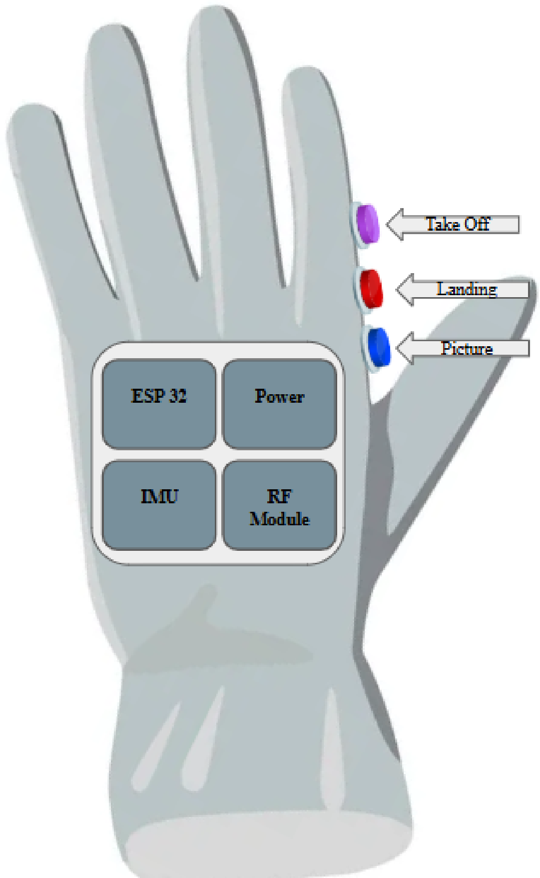

Figure 1: General layout of what goes on the glove.

1.4 High-Level Requirements List

To demonstrate success, our project must meet the following measurable requirements:

1. The drone responds in real time to glove commands with minimal delay.
2. Buttons work consistently for take off and landing.
3. Directional commands (forward, back, left, right, up, down) work consistently.
4. *(Stretch goal)* If the camera is integrated, the system should be able to store low-resolution images to the sd card.
5. *(Stretch goal)* Haptic feedback provides clear communication of system status to the user.

2. Design

2.1 Block Diagram

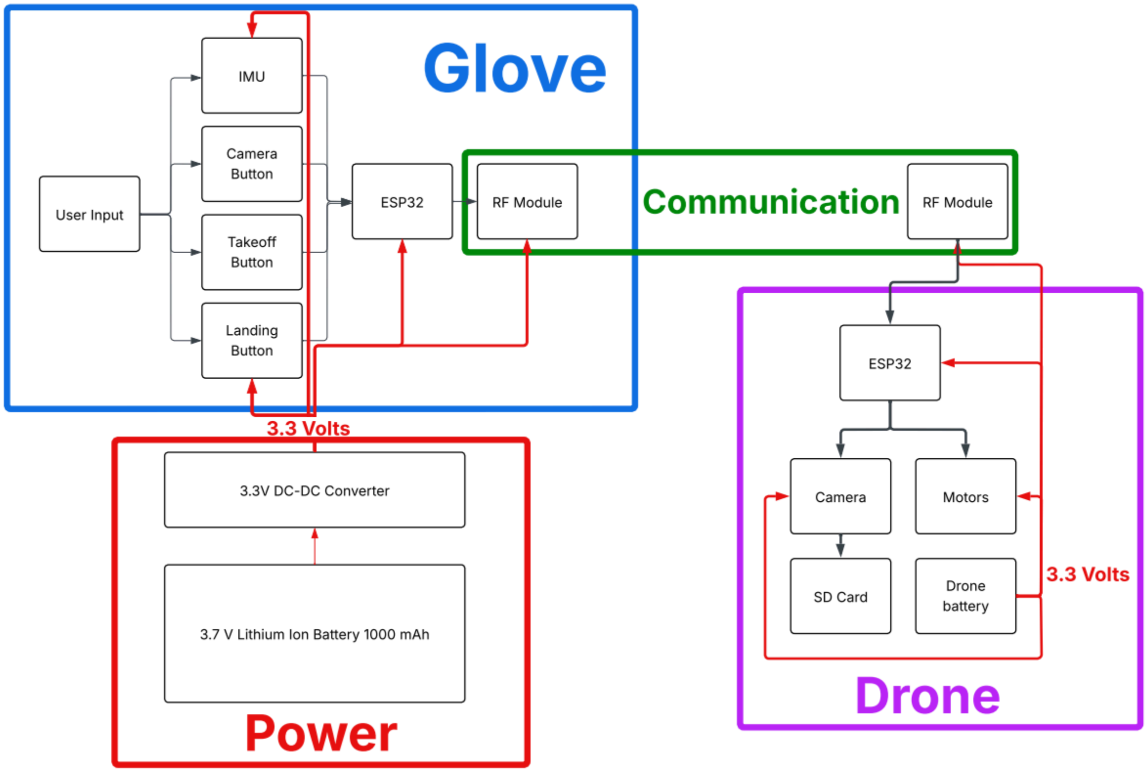

Figure 2: High level Block Diagram

2.2 Subsystem Descriptions and Requirements

2.21 System Summary:

A wearable glove equipped with IMU and gyroscope sensors will capture hand orientation and motion, converting gestures into control commands for a quadcopter. An ESP32 on the glove will transmit these signals wirelessly to an ESP32-based flight controller on the drone, which executes pitch, roll, yaw, and throttle inputs in real time. Optional modules, such as an ESP32-CAM for basic image capture and haptic feedback for status alerts, can be integrated to enhance usability. The electronics are powered by rechargeable Li-Po batteries, supporting stable hover and responsive gesture-based control.

2.22 Subsystem 1: Glove

2.22A: IMU

Description:
The glove uses an **MPU-6050 (6-DoF)** to sense hand tilt/rotation and map them to flight commands. Mounted flat on the back of the hand, readings are lightly filtered and calibrated; **pitch** drives forward/back, **roll** drives strafe.

Requirements

- Low-latency
- Reliable 3.3 V power and clean I²C wiring with pull-ups; startup bias calibration.
- Configurable dead-zones/gains for comfortable gestures and drift robustness.

Testing Plan

- Calibrate on a level surface; verify axes match mounting.
- Tilt/roll checks to confirm correct direction and sensible scaling.
- Hold-still drift/jitter check, then end-to-end integration to ensure smooth, predictable control.

2.22B ESP32

Description
The ESP32 (HiLetgo ESP-WROOM-32) runs the glove: it reads the MPU-6050 via I²C, maps pitch/roll to commands, and forwards them over the RF/UDP link. Programmed in Arduino IDE; spare GPIOs support buttons/UX. Powered at 3.3 V.

Requirements

- Stable 3.3 V power and proper SDA/SCL pull-ups.
- Proper communication with RF module
- Simple, configurable gesture mapping and debounced buttons.

Testing Plan

- I²C scan finds 0x68; stream raw IMU data to Serial.
- Calibrate biases at rest; verify axis/mapping directions.
- Send packets over RF/UDP; confirm end-to-end responsiveness.

2.22C Buttons

Description
Three momentary pushbuttons (panel-mount) on the glove provide discrete commands:

- Takeoff/Hover
- Land
- Capture Photo

Each button connects to an ESP32 GPIO configured as INPUT_PULLUP (idle HIGH, press pulls to GND to active-low).

Requirements

- Reliable active-low wiring to ESP32 GPIOs with common ground
- Debouncing (software preferred), plus long-press safety for takeoff/landing.

- Clear placement/labeling for quick access

Testing Plan

- Bring-up: Verify idle HIGH / pressed LOW on each GPIO (Serial print).
- Debounce check: Rapid taps register single events; long-press triggers only the intended safe action.
- Integration: Map to handlers—takeoff→hover, land→slow descend/disarm, photo→capture—and confirm correct behavior end-to-end.

2.23 Subsystem 2: Drone

Description

The drone requires a flight controller capable of receiving wireless commands from the glove and translating them into motor actions for stable flight. Our current plan is to use the PyDrone ESP32-based controller, as it is designed for programmability, includes exposed GPIO pins, and supports open-source firmware. These features provide flexibility to integrate additional modules, such as an ESP32-CAM, in later stages of development.

Key Requirements

Our system must support real-time command reception from the glove with latency under 200 ms to maintain responsive control. The drone must provide stable hover and basic flight control, including pitch, roll, yaw, and throttle. GPIO pins must be accessible for optional camera integration and future expansion. The firmware must be open-source and modifiable to implement custom command mapping and an emergency motor shutoff.

Testing Plan

We will validate a stable five-minute hover, confirm correct mapping of glove gestures to flight controls, and measure command latency to ensure it stays below 200 ms. If camera integration is attempted, we will verify that images can be captured or streamed without interfering with flight stability.

2.24 Subsystem 3: Communication

Description:

Foreseeing that the ESP32 might not have a long enough range to reach the drone upon flight, our group has opted to include an RF module to provide a long-range, wireless control link between the glove ESP32 and the drone's flight-control ESP32. We use the HiLetgo nRF24L01+ PA+LNA 2.4 GHz RF transceiver, which supports programmable data rates up to 2 Mbps and offers long-range, low-latency transmission. Its high-power front end and external antenna enable up to 1 km line-of-sight range at 250 kbps, while operating independently of any Wi-Fi camera streaming to prevent interference.

Key Requirements:
The RF module will enable commands to reach the drone with low latency and maintain reliable control over at least 50–150 m indoors and ~1 km outdoors. The module draws 100–130 mA in transmit bursts, so it requires a dedicated 3.3 V regulator to avoid voltage sag. This regulator is already included in our power design as seen below. Correct SPI wiring with CE and CSN pins and careful channel selection are essential to ensure stable, interference-free operation.

Testing Plan:
We will verify stable SPI communication and power supply, then measure end-to-end latency and range in both indoor and outdoor environments. Tests will confirm that simultaneous Wi-Fi video streaming does not affect command reliability and that the link remains robust under typical interference.

2.25 Subsystem 4: Power

Description:
The glove needs a controlled 3.3 V supply to power the ESP32, RF module, IMU, and three buttons. We chose a single-cell 3.7 V, 1000 mAh Li-Po with a 3.7 V to 3.3 V converter (1 A).

Key Requirements:
Provide stable 3.3 V power; supply the expected continuous and burst currents; include Li-Po protection (overcharge/overdischarge/short).

Components:
Battery: 3.7 V, 1000 mAh Li-Po.
Regulator: 3.7 V to 3.3 V converter, rated at 1 A.

Calculations:
Component current estimates (conservative):

ESP32: active = 0.200 A (datasheet), peak 0.300 A
nRF24L01: average = 0.040 A, TX burst = 0.120 A
IMU: 0.004 A
Buttons: 0.001 A.
Typical total current: 0.200 + 0.040 + 0.004 + 0.001 = **0.245 A**.
Worst continuous estimate (ESP32 peak + RF peak): 0.300 + 0.120 + 0.004 + 0.001 = **0.425 A**.
Output power at 3.3 V: 3.3 V × 0.245 A = **0.8085 W**.
P_worst = 3.3 V × 0.425 A = **1.4025 W**.
Battery energy: 3.7 V × 1.000 Ah = **3.7 Wh**.
Efficiency = 0.90 (estimate)

Typical battery draw: 0.8085 W / (3.7 V × 0.90) = 0.8085 / 3.33 = **0.2428 A**.
Worst battery draw: 1.4025 W / 3.33 = **0.4212 A**.
Typical runtime = 1.000 / 0.2428 = **4.12 h**.
Worst runtime = 1.000 / 0.4212 = **2.37 h**.
Apply conservative usable capacity (80%): typical = **3.3 h**, worst = **1.9 h**. These figures justify the 1000 mAh choice. Even under heavy continuous load the glove can still operate for almost two hours which will more than suffice.

Testing Plan:
Verify 3.3 V under idle, typical, and peak loads; run continuous RF transmit stress to confirm no brownout. Also measure real-world runtime to ensure >1 hour per charge.

2.3 Tolerance Analysis

The most difficult requirement for our project will be seamlessly integrating the camera onto the drone for taking pictures. This will be difficult because it is unclear whether we will be able to install the camera onto the existing ESP32 that comes with the drone. The ESP32-CAM module, for example, draws approximately 160–260 mA during active image capture at 3.3 V, which is roughly 0.85 W of additional power consumption. Our drone battery has a nominal capacity of 1200 mAh at 3.7 V. Assuming the drone flight time without a camera is 5 minutes, the additional camera load could reduce flight time by up to 10-15%, leaving us with closer to 4 minutes of flight time. This reduction must be tolerated within our system constraints, as it affects the window in which the user can capture images.

Furthermore, if we are able to install the camera on the drone, the process of the user sending a command to the camera to store the photo locally will also prove to be a challenging task. The ESP32 will need to handle camera data acquisition, file system writes to an SD card, and wireless command reception simultaneously. If we capture images at VGA resolution (640×480), a single JPEG image will typically take 50-100 KB of storage. At 10 pictures per flight, we will require a minimum of 1 MB of SD card space per session, plus overhead for the file system. We will have to tolerate delays of approximately 200-300 ms per image for image compression and write time, meaning the drone must hover stably during this period without significant drift to avoid capturing blurred or misaligned images.

3. Ethics and Safety

Our project follows the IEEE and ACM Codes of Ethics, prioritizing safety, honesty, and responsible design. Drones raise ethical concerns related to misuse, privacy, and airspace regulations. To address this, we will limit our system to hobbyist-level drones, comply with FAA rules (flying under 400 ft in uncontrolled airspace), and obtain approval before campus test flights.

Electrical Safety: All glove-mounted circuits will be insulated and tested to prevent shorts. Li-Po batteries will follow IEEE battery safety standards, using proper charging, protection circuitry, and enclosures to reduce risks of overheating or puncture.

Mechanical Safety: Propellers will be guarded, and flights limited to controlled test areas. A gesture-based emergency shutoff ensures immediate motor disablement in unsafe conditions.

Wireless Safety: ESP32 Wi-Fi communication will be tested for reliability. A fail-safe mode will cut motors if signals are lost.

Lab Safety: Work will follow UIUC lab policies and OSHA guidelines, including PPE use, safe soldering practices, and risk assessments during flight tests.

4. References

IEEE. "IEEE Code of Ethics." 2023. https://www.ieee.org/about/corporate/governance/p7-8.html ACM. "ACM Code of Ethics and Professional Conduct." 2023. https://www.acm.org/code-of-ethics OSHA. "Occupational Safety and Health Administration Regulations." 2023. https://www.osha.gov/ NFPA. "National Fire Protection Association Electrical Safety Standards." 2023. https://www.nfpa.org/ University of Illinois. "Division of Research Safety." 2023. https://www.drs.illinois.edu/