UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

Glove Controlled Drone

Electrical & Computer Engineering

Zachary Greening, Atsi Gupta, Aneesh Nagalkar

ECE 445 Group 11

Problem

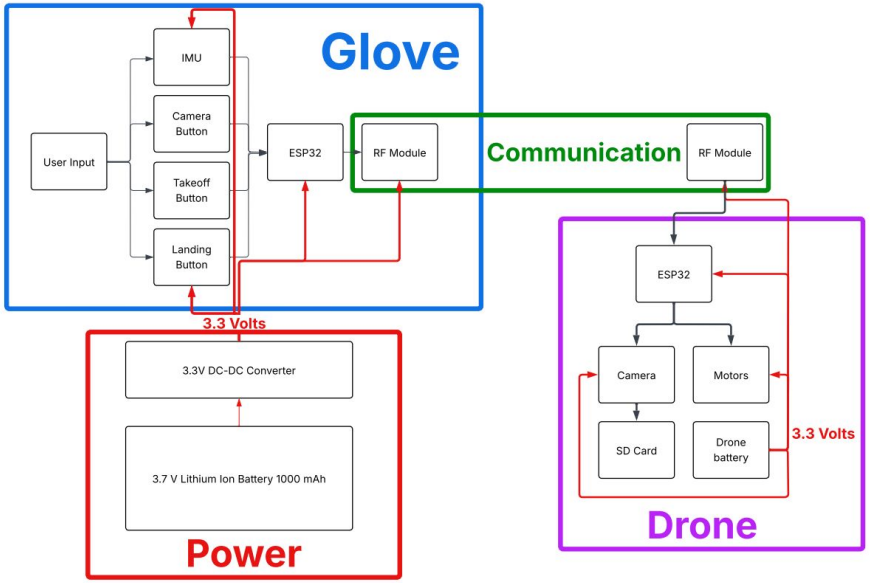
Our Solution:

High Level Requirements

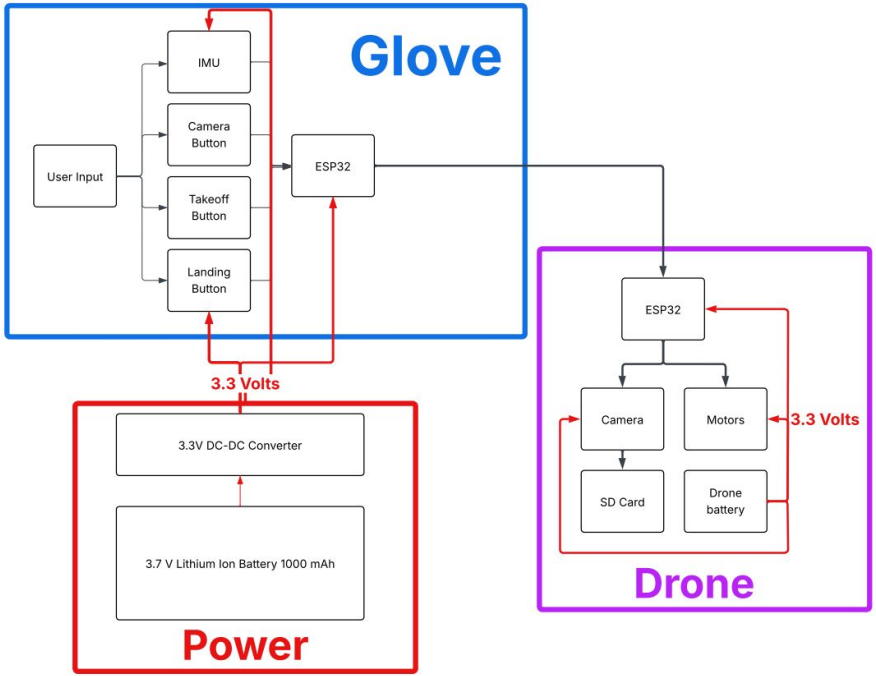
1. The drone responds in real time to glove commands with minimal delay.
2. The buttons make the drone hover or land within 15 milliseconds of being pressed.
3. Directional commands (forward, back, left, right) work 80% of the time over 20 trials.
4. If the camera is integrated, the system should be able to store low-resolution images to the sd card.

Design

Old



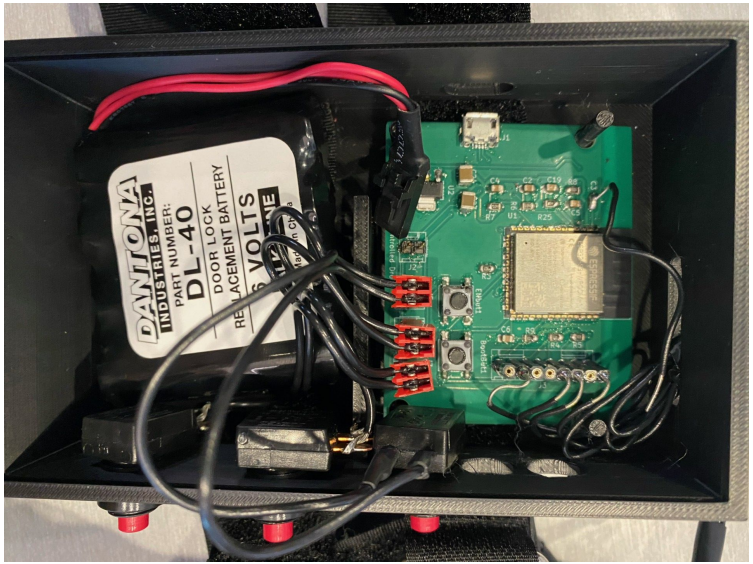
New



Glove Subsystem

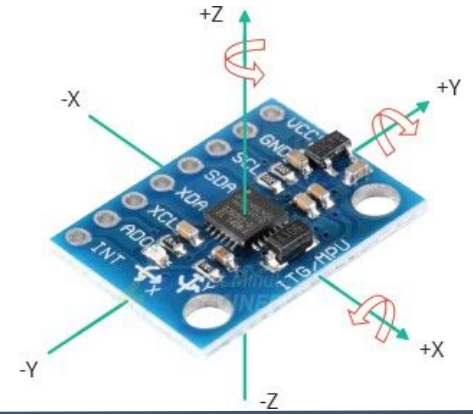
Glove Subsystem

- **Send commands to drone**
  - **Directional Commands**
  - **Takeoff/Land/Picture**

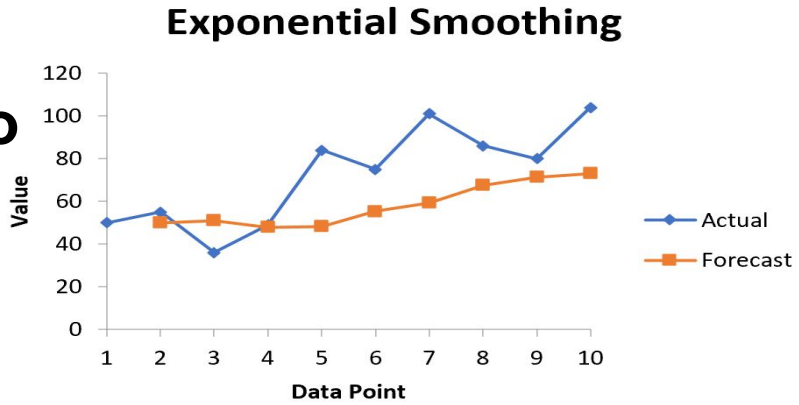


MPU 6050

- 6 DOF IMU
- ±2g accelerometer accuracy
- Efficient (~3.8mA max current draw)



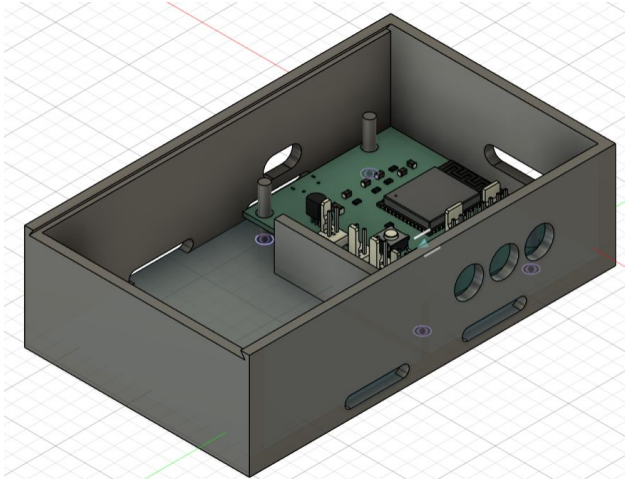
Calibration: 500 samples to determine relative origin

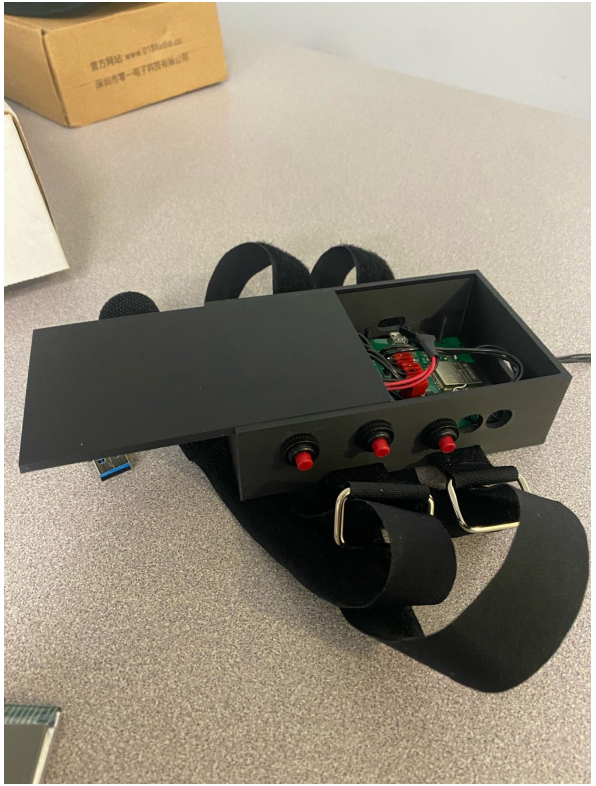


$$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\hat{y}_t$$

Enclosure

- Wearable
- Houses PCB and battery
- User buttons





Future Design Iteration:

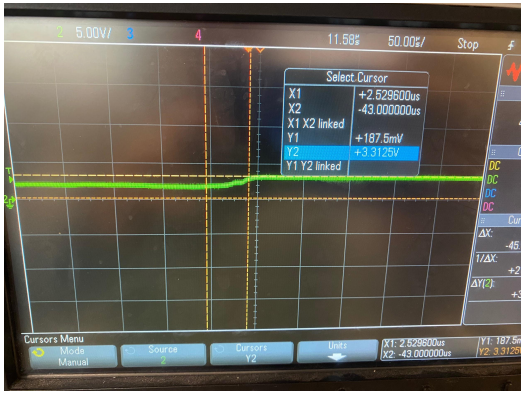
Buy smaller battery

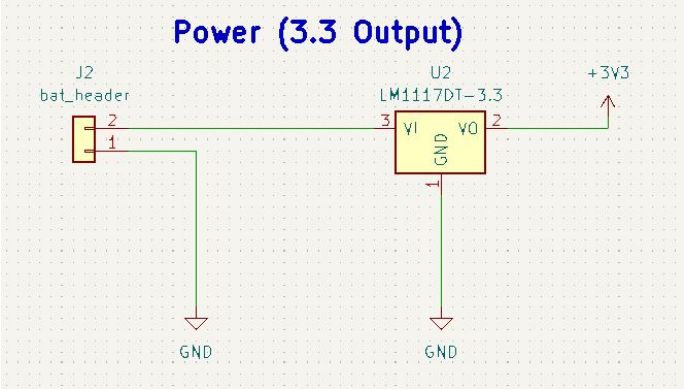
3.7 Volt

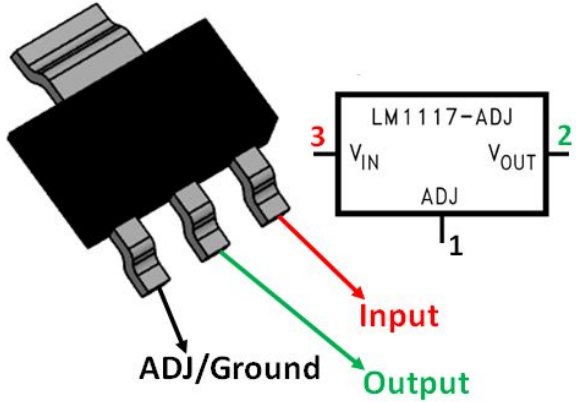
Power Subsystem

All components uses 3.3V

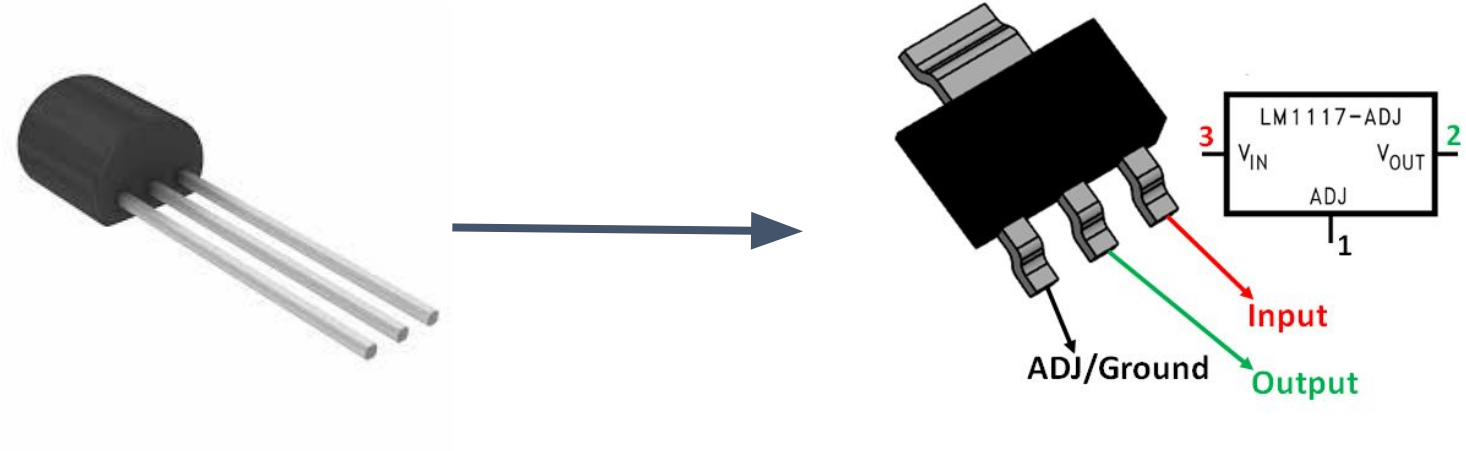
ESP32: active = 0.200 A (datasheet), peak 0.300 A
IMU: 0.004 A
Buttons: 0.001 A.
Typical total current: 0.200 + 0.004 + 0.001 = **0.205 A**.
Worst continuous estimate (ESP32 peak): 0.300 + 0.004 + 0.001 = **0.305 A**.







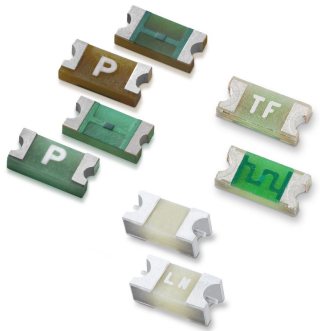
- **Design Changes**
  - **Changed from LP2950 to LM1117**
  - **ESP32 Antenna current spikes**



Future Improvements

Fortify Power Subsystem

Fuse



Buck Converter

Drone
Camera Subsystem

Drone + Camera Subsystem

01Studio Pydrone

- ESP32-controlled drone
- Pre-installed firmware
- Emits LAN
- 7-10 minute flight time

Xiao ESP32-S3 Camera Module

- Receives commands from glove
- Captures low-res photo
- Saves jpg to SD card

MicroPython

- Lightweight Python meant for microcontrollers
- Drone’s API written in micropython
- Limitations
    - Native packages
    - A lot of overhead

```
from drone import DRONE

#构建四轴对象
d = DRONE(flightmode = 0) #无头模式

#使用方法

#起飞
d.takeoff()

#降落
d.landing()

#四轴飞行器姿态控制
d.control(rol = 0, pit = 0, yaw = 0, thr = 0)
```

Drone Defect Journey

Drone unable to hover

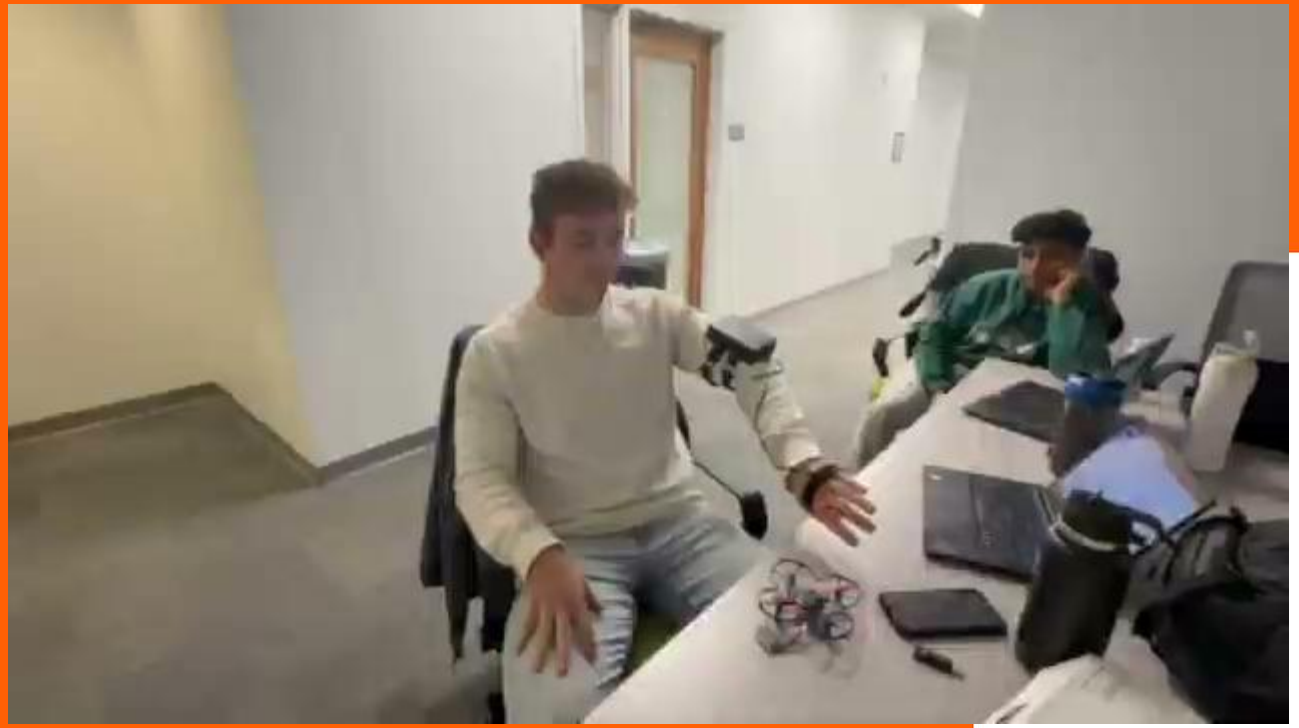
Built car

Diagnosed + Ordering Motors

Firmware Locked

Flight Controller

Result: Hover requirement could not be fully verified due to hardware defect

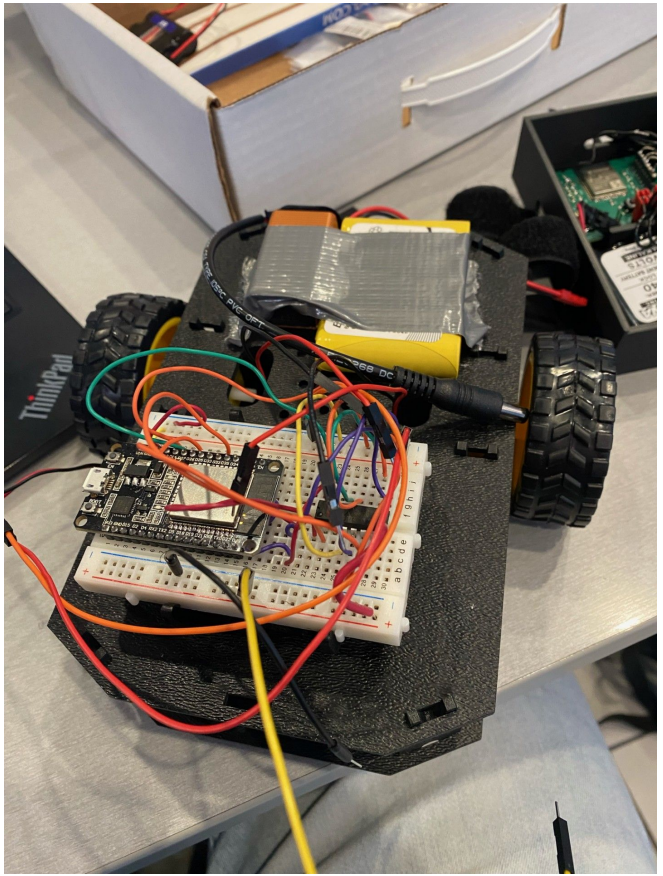


Future Improvements

Buy a different drone

RC Car

- Proof of concept
- Demonstrates glove functionality
- Maps controls to XY plane



Communication

Use ESP32 antenna instead of RF module

50 meters indoors
200 meters outdoors

UDP Advantages

Low latency
Handles packet drops

Glove sends 8-byte packet to drone

```
pkt = bytearray(8)
pkt[0] = 0
pkt[1] = control_to_byte(0)  # roll
pkt[2] = control_to_byte(0)  # pitch
pkt[3] = control_to_byte(0)  # yaw
pkt[4] = control_to_byte(0)  # throttle
pkt[5] = btn                 # button state
pkt[6] = 0
pkt[7] = 0
```

High Level Requirements

1. The drone responds in real time to glove commands with minimal delay.
2. The buttons make the drone hover or land within 15 milliseconds of being pressed.
3. Directional commands (forward, back, left, right) work 80% of the time over 20 trials.
4. If the camera is integrated, the system should be able to store low-resolution images to the sd card.

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Conclusion

Electrical & Computer Engineering