

Final Report:

Building a Drone from Scratch

F21-CPSC-65000-001

Taten Knight

Background

For this project, I chose to pursue building a drone from the ground up. The following are the general goals for the drone, with stretch goals marked with a '*' and very unlikely goals marked with '**'. The drone ought to be able to:

- Hover in place
- Have its movement controlled via Wi-Fi
- Avoid obstacles
- Navigate a living space autonomously*
- Create a 3D map of the living space via photogrammetry**
- Localize common objects in the 3D space via object detection and photogrammetry**
- Grab small objects and bring them to the pilot**

I chose to work on a drone because it seemed fun, and I saw an opportunity to use past knowledge from mechanical engineering and control systems courses and apply them here.

Materials

Some of these materials were bought specifically for the drone, while others were on hand from other projects.

- 80A Electronic Speed Controllers (4)
- Male and Female Bullet Connectors (50)
- 5S Lipo Battery (2)
- Lipo Balance Charger (1)
- 1 Male to 2 Female Lipo Battery Connector (2)
- MakerBeam Starter Kit (1)
- Infrared Proximity Distance Sensor (9)
- ESP32 Board (1)
- Male DC Power Connector to 9V Battery (3)
- MPU6050 (1)
- XING 2806.5 180 KV Brushless Motor (4)
- 7" 3 Blade Propellers (8)
- Servo Motors (4)
- GoPro Camera (1)
- Micro SD Card (1)
- Breadboard
- Wires
- 3D Printer

Methods and Technology

The ESP32 can run Micropython, which is a lightweight bare metal implementation of the Python programming language for microcontrollers. It provides access to the features of the board through use of specialized libraries. I chose to use this route because I am familiar with Python, and implementing a control scheme in this environment was simpler for me, versus using the Arduino environment and working with Java constructs. There was no overall design going into this, and my main focus was only to purchase parts that I believed would be essential to a drone flying successfully.

There are drone flight control boards that are built specifically for the control of a quad-copter drone, but there were two issues with this. One, I would like to build a drone with only two propellers, and two, a flight control board does all of the heavy lifting for the control systems, and I would like to design my own. This is why I chose an ESP32 board alongside an MPU6050 instead.

For control of the drone, the ESP32 and my desktop initialize a client-server socket connection on startup, with the client being the desktop and the ESP32 acting as a server. I wrote a very simple json object parsing scheme that allows me to type commands into the keyboard and change to speed of the propellers and the orientation of the servo motors. This will need to be updated in the future to allow for simpler controls, and to fit into a more formal control scheme.

The layout of the drone is simple. There is a lipo battery connected to two ESCs, which are in turn connected to the two propellers. The propellers are mounted on the servo motors via 3D printed brackets, and there is a shroud around the props to avoid injury. The servos and the ESCs are controlled via the ESP32. The MPU6050 is also connected to the ESP32, and streams orientation data to the desktop. There are three power sources for the project. The lipo battery for the motors, and two separate 9V battery sources. One powers the servos and the other powers the MPU6050 and the ESP32. This is because when the servos are moving, the power draw is too high and the ESP32 resets if on the same power source.

The frame of the drone is built with a combination of MakerSpace beams and 3D printed objects. The beams form a simple rectangular frame that other components can be easily bolted to. The 3D printed pieces include a bucket that forms the base of the drone and holds the ESCs, lipo battery, and 9V battery, another piece that holds the breadboard, shrouds for the drone propellers, and the aforementioned bracket that connects the brushless motors to the servos. The current state of the drone can be seen in Figure 1.



Figure 1. Current State of Drone (Disconnected)

Progress

As of now the drone can be controlled via the desktop, and the circuitry kinks have been worked through. I have done preliminary testing on all of the motors and the individual components. There was a major delay in testing due to an unfortunate accident early on, wherein I injured my hand when testing the brushless motor / propellor combination.

I created an issue by buying two motors that spin the same way for the drone, that issue being that the torque from the motors causes the entire drone to spin – the motors spin counterclockwise and the drone spins clockwise. Initial testing has been dedicated to solving this problem by using the thrust from the propellers at an angle to prevent spin. There was another accident wherein the drone propellor got caught on some fabric and wiring and destroyed itself, and caused another delay while waiting for the 3D printing of new parts, namely a new servo to motor bracket, and shroud for personal protection and safety.

Future Work

Now that there is a way for me to safely test the drone, next steps include preventing drone spin due to brushless motor torque, tying in the MPU6050 acceleration and orientation data to allow the drone to hover in place, and developing a control system that uses target MPU6050 orientation to promote forward and backward movement of the drone.

The control system will initially be a proportional control, with changes in actuator values being determined by the error between the current and desired state. The desire is to create individual goals that contribute to the overall motion of the drone. For example, the goal state for the pitch of the drone while moving forward might be 1 degree. Here the propellers will be angled forward slightly to allow the drone to move in the forward direction, so the servos will adjust to reach this goal state. With this angle, however, we lose some of the thrust in the upward direction. This will cause the drone to move down slowly if it is not corrected for. The other goal state may be 0 acceleration in the vertical direction, which can be measured via the infrared proximity sensors. This goal may be tied directly to motor speed and prevent the drone from falling. These two goal states and subsequent actuator adjustments will work in tandem to control different flight variables.

Conclusion

Although I didn't get as far as I would have liked to, I do believe that the drone is in a good place to pursue future goals moving forward. The biggest challenges thus far have been safety, as drone motors spin ridiculously quickly and I would prefer to keep my eyes/hands intact. With the creation of the propeller shroud I believe that it will be easier to iterate on control improvements. I will continue to pursue the completion of this project outside of this course, because I find it interesting, and the outcome to be extremely rewarding.

Notes

There are no citations for this report, as the only things that I have used as true resources have been tutorials on how to install software on the ESP32, create a socket connection, and get data via the MPU6050. Future citations may arise when designing the flight control system, as that process is potentially extremely complicated. Continuous updates around this project can be found on my [YouTube](#) channel and [TikTok](#) account.