

# Educational Experiences in Building a Custom UAV

Sean Stahly and John Hastings  
Dept. of Computer Science and Information Technology  
University of Nebraska-Kearney  
Kearney, NE 68849  
stahlysr@lopers.unk.edu, hastingsjd@unk.edu

## Abstract

Unmanned aerial vehicles (UAV) have become popular due to their far reaching applications, and as educational platforms because of their “coolness” factor. Our project sought to create a UAV in the form of a quadcopter which could be used as a platform for future UAV research. Rather than purchasing a prebuilt system, we structured this as a student-led experiential learning project in which the UAV would be assembled from parts sourced from a variety of suppliers and guides. This approach would provide deeper insight into the operational mechanics of drones and the programming required as opposed to simply using prebuilt UAV control software. Beyond the educational factor, we felt that having direct control over the components offered better scalability for future research directions. We created the UAV using a programmable flight controller which is operated through radio control (RC). The UAV is programmed through the use of an RC transmitter. The RC transmitter communicates with an RC receiver on the UAV which then passes the input into the flight controller. The flight controller converts that input into a usable form which controls the throttle of the motors on UAV, giving it controlled flight. The flight controller is a modified Arduino, and is programmed using the Arduino language, with several libraries also taken from arducopter, an open source project that is designed to equip Arduinos to work with UAVs. The RC input is mapped into a format usable for the motors using proportional integral derivatives (PIDs). By the end of the project, the UAV did achieve flight. Future work includes making the UAV fully autonomous with a long term goal to research a variety of UAV security related issues. Our paper describes how the project was structured as a student-led project, and provide details of the UAV build, along with lessons learned both from educational and practical perspectives.

# 1 Introduction

Unmanned aerial vehicles (UAV) have become popular due to their far reaching applications, and as educational platforms because of their “coolness” factor. Our project sought to create a UAV in the form of a quadcopter which could be used as a platform for future UAV research. Rather than purchasing a prebuilt system, we structured this as a student-led experiential learning project in which the UAV would be assembled from parts sourced from a variety of suppliers and guides. This approach would provide deeper insight into the operational mechanics of drones and the programming required (as opposed to simply using prebuilt UAV control software). Beyond the educational factor, we felt that having direct control over the components offered better scalability for future research directions. We created the UAV using a programmable flight controller which is operated through radio control (RC). The UAV is programmed through the use of an RC transmitter. The RC transmitter communicates with an RC receiver on the UAV which then passes the input into the flight controller. The flight controller converts that input into a usable form which controls the throttle of the motors on UAV, giving it controlled flight. The flight controller is a modified Arduino, and is programmed using the Arduino language, with several libraries also taken from arducopter, an open source project that is designed to equip Arduinos to work with UAVs. The RC input is mapped into a format usable for the motors using proportional integral derivatives (PIDs). By the end of the project, the UAV did achieve flight. Future work includes making the UAV fully autonomous with a long term goal to research a variety of UAV security related issues. The following sections describe how the project was structured as a student-led project, and provide details of the UAV build, along with lessons learned both from educational and practical perspectives.

## 2 Experiential Learning

This research was structured as an experiential learning project with funding through the Nebraska EPSCoR Undergraduate Research program. Kolb [1] defines experiential learning as the act of “learning through reflection on doing”. Kolb [2] suggests that experiential learning requires learners with four traits: the interest in trying new experiences (concrete experience), the ability to reflect on the experience (reflective observation), the analytical ability to abstract concepts from the observations (abstract conceptualization), and the problem solving ability to apply the new abstract ideas in practice (active experimentation). This approach can be successfully used with minimal instructor involvement given a learner with the necessary ability and self initiative, which was the case in this project, despite no prior UAV experience by the student researcher (or instructor). This project was completely student driven with involvement by the instructor only when needed and mostly just support in terms of discussion and final purchase approvals. The following sections details the process followed in the project is described in the next section.

## **3 Building a UAV**

This research project sought to build a programmable UAV from scratch due to student interest in the technology. With anything built from scratch, particularly using a new technology, preliminary research was necessary to ensure the project proceeded from a good starting point. The project went through several stages including, but not limited to, researching the type of UAV that should be built, studying how it could be built, sourcing parts, assembling the parts, building and configuring software, and testing and debugging the UAV programmed and airborne. The subsections that follow will describe these phases.

### **3.1 Choosing the Type of UAV**

When we decided to create a UAV, we first needed to decide what kind of UAV would be created. After looking over a few guides and recommendations, we decided upon building a quadcopter, or a UAV with four motors and propellers. This decision was primarily due to the fact that a quadrotor was supposedly easier to fly and manually program than a tri-rotor UAV. It was also slightly easier and quite a bit cheaper than a hexarotor copter. When compared to other styles, such as an airplane-like UAV, we chose to use a rotor-based UAV due to ease of use, and for the ability to hover.

### **3.2 Sourcing Parts**

Upon researching several guides on quadcopters and communicating with some hobbyists, we discovered what hardware exactly we would need, along with a few decent websites to order parts from. We ended up purchasing a frame, motors, electronic speed controllers (ESCs), a power distribution board, a battery, many propellers, an RC transmitter/receiver, and a flight controller. A majority of the parts were of no particular brand but were chosen because the selling company was fast, reliable, and relatively cheap. The exception to this rule was the flight controller. While there were numerous flight controllers that were fairly cheap, a majority of flight controllers are not programmable. In order to overcome this obstacle we ordered a flight controller that consisted of a modified arduino from 3DRobotics [3]. This flight controller would allow us to program it using the arduino language with libraries from the Ardupilot open source project [4]. After acquiring the hardware, it was now time for us to assemble the quadcopter.

### **3.3 Assembly**

Upon receiving the UAV components, the first task was to simply assemble the X-shaped frame. After this we attached the motors to the end of each arm. Next, we attached each motor to its own ESC, which were secured upon the UAV arms and secured with zip ties. The other end of these ESCs were attached to the power distribution board, which was then attached on the underside of the UAV's frame. This board then needed to be attached to both the battery and the flight controller. The flight controller then needed to be hooked up to the battery to receive power. Now while this would be all that the UAV would need to fly with a predetermined path, we still needed to hook up the RC receiver.

We simply used a few pin connectors to hook up the four incoming channels from the receiver to the flight controller. Due to the poor alignment of screw holes given the mix of sourced hardware, a majority of the parts were held on with zip ties. Progress during the initial assembly is shown in Figure 1. The assembly of the UAV was followed by an initial round of tests as described in the next subsection.

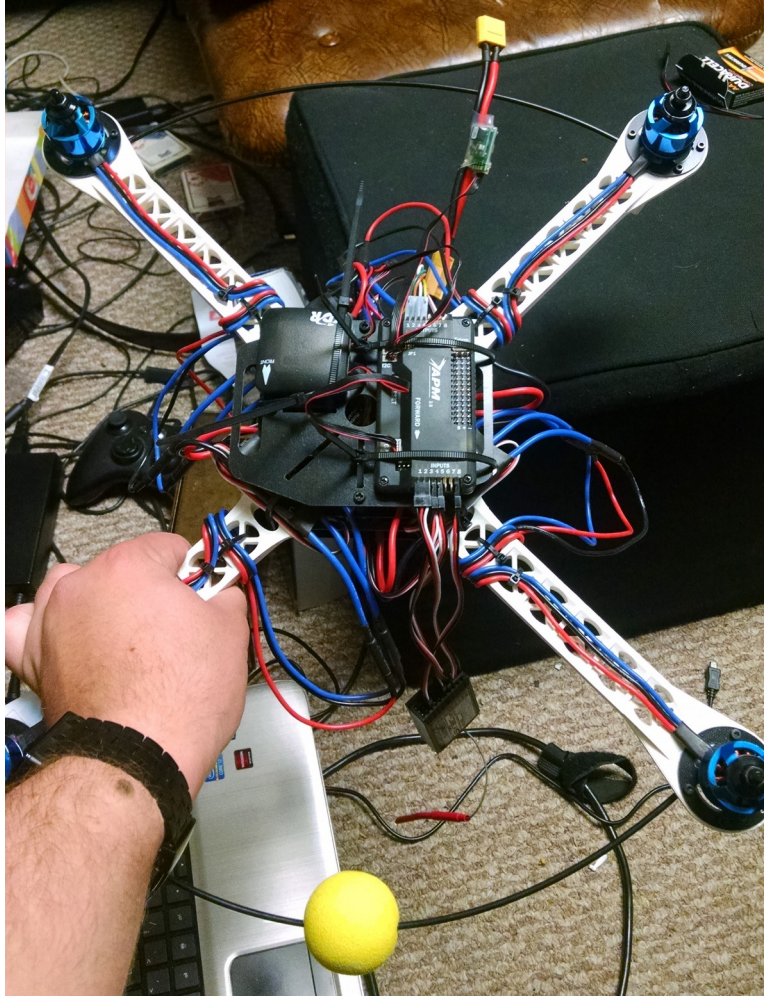


Figure 1: Progress During the Initial Assembly

### 3.4 Initial Tests

Before programming the flight controller, the UAV was first tested with some pre-built software. The Ardupilot [4] program was used. It is designed to load vehicle controlling software onto Arduino based vehicles. This includes vehicles such as RC cars, planes, and several different multi-rotor UAVS. After uploading the quadcopter software to the flight controller, the calibration process began. This included primarily rotating the UAV in a variety of directions, to sync up the onboard gyroscope to the software. However, the next portion of calibration required turning on the RC controller to measure the minimum

and maximum of each channel that would be used to pilot the UAV. During calibration, it was discovered that the RC transceiver would not send a signal to the flight controller. After swapping out several different pin connectors, in order to rule out defective connectors, communication still did not take place. The RC transceiver/receiver came with no instruction manual, or any documentation at all. However, an online forum described that the receiver requires a minimum of connecting six channels in order to properly function. After attaching the necessary additional pins, the rest of the calibration process performed quite smoothly. After this, the flight motors spun up quickly, just like expected. It was now time to attempt a test flight for our UAV.

### **3.5 Initial Test Flight**

The first requirement to fly was to attach a propeller to each motor, ensuring that each propeller was arranged to spin the opposite way from its adjacent propellers. After attaching these propellers, we went through the preflight safety checks, spun the motors up, and watched the UAV remain a ground based vehicle. After performing the calibration process a couple more times and attempting a few more test flights, no further progress was seen. During one of these tests, one person held the UAV above their head. It was noticed from this test that while the motors were on it was incredibly difficult to change the orientation of the UAV, showing that its stabilization functionality was working. This began a long search of Google, and delving into forums for several weeks searching for possible reasons for this issue. After yielding no results, the decision was made to go ahead and build our own software to run on the flight controller, with the goal to determine why the UAV could not get off the ground.

### **3.6 Building Custom Software**

The software was built using the Arduino IDE, along with the arduflyer libraries which were designed to work with Arduino-based flight controllers. The initial version of software was used to attempt to send a signal through the RC controller to the receiver and interpret the signal. This was fairly easy to read in and sync with each individual channel. The next step was to give power to our motors in order to activate them. Simply sending the raw output from our RC receiver to the motor was able to spin it up. It now came to the more complicated part of the programming, the actual control of the motors in order to allow the UAV to turn. While researching this portion of the programming, it was found that PID, or proportional integral derivatives, would need to be implemented in order to perform this task. This research uncovered a guide for a similar project [5] that gave detailed instructions on how to implement PIDs for Arduino-based flight controllers. After correctly mapping the controls, through this process and running the raw RC output values through an averaging filter, the UAV was ready for a test flight.

### **3.7 Successful Flight using Custom Software**

Testing the custom built software, without propellers attached, revealed a huge improvement in the speed of our motors over the prebuilt software. Further research showed that the ArduPilot Mission planner software would decrease the max speed of the motor for safety reasons, preventing the quadcopter from getting off the ground.

However, this limit was bypassed by implementing custom software. After attaching the propellers to the UAV, and ensuring that the battery passed the built-in safety checks, the throttle was increased. The UAV successfully lifted off the ground, and flew straight into a shed. After recovering the UAV, which suffered minimal damages, it was surmised that the UAV was overly front heavy. This was a difficult task to overcome, primarily due to the wide variety of parts from multiple vendors. The lack of a specific layout for the equipment, along with no good way to attach all of our components to the frame resulted in an improvised method of dozens of zip ties. This method was able to mitigate the lopsidedness of the load to improve balance somewhat, but could never be able to find a perfect solution. In addition, some fine tune adjustments to the custom software allowed for more accurate RC manipulation and adjusting the flight controller's output to the motors. Ultimately, the UAV was refined to a point that it was fairly stable and would fly, even though it had difficulty hovering in place.



Figure 2: Successful Outdoor Test Flight

## 4 Future Directions

When improving upon this project for the future, we would probably look to purchase a different frame, and perhaps replace a few other components that do not fit on standard boards. This would be so we can have a more balanced UAV to avoid the issue of the UAV moving slowly in a particular direction. In addition we would like to equip a GPS module so that we can incorporate some autopilot software onto the quadcopter. Moving further than an autonomous quadcopter, we would also like to explore the role that a UAV could play in cyber security, such as getting close to a network in order to sniff packets remotely.

## 5 Student Observations

As a student creating this project, I was able to learn quite a bit about electronics in general, particularly the communication between hardware, such as an arduino, and software. It has become apparent to me that it is much more difficult to attempt to debug

a hardware-based project rather than a solely software-based project. I was also able to work over a long period of time on a single project and see it grow over a long period of time. While this project did frustrate for a fairly long period of time, I do feel that I accomplished a fair amount and I was able to learn quite a bit through this project, which was the goal in the first place. In addition I was able to gain a perspective on a portion of the computing field that I would not have been able if I did not choose to perform a project similar to building a UAV.

## **6 Conclusion**

This research provided an experiential learning project for a undergraduate student researcher. With minimal faculty involvement, the student was able to fully manage the project from start to end, and ultimately produce a UAV which achieved successful flight. This was despite having no experience with any of the specific technologies used prior to the project. The student possessed Kolb's four traits described earlier in the paper, particularly an interest in the subject area, and aptitude. At the University of Nebraska-Kearney, students have demonstrated that given the necessary aptitude, and the freedom to formulate their own projects, successes such as that seen with this UAV research are possible.

## **References**

- [1] Kolb, D. A., & Fry, R. E. (1974). Toward an applied theory of experiential learning. MIT Alfred P. Sloan School of Management.
- [2] Kolb, D. A. (1984). Experiential Learning as the Science of Learning and Development. Englewood Cliffs, NJ: Prentice Hall.
- [3] 3D Robotics (2016). <https://3dr.com>. Accessed Feb. 1, 2016.
- [4] Ardupilot (2016) <http://ardupilot.com>. Accessed Feb. 1, 2016.
- [5] Owenson, G. (2016). Build Your Own Quadcopter Flight Controller. <http://blog.owenson.me/build-your-own-quadcopter-flight-controller>. Accessed Feb. 1, 2016.