

# Sprint Report #2

## Summary

Team Expeditus was able to adapt to setbacks, and make progress on other tasks and goals of the project. Specifically, the team was able to make progress on the landing software, as well as find and implement a visual simulation that models the Pixhawk flight controller. The team was also able to coordinate with our advisor and school faculty for funding and ordering of our UAV platform and components. The restructuring of tasks for Sprint 2, does have a knock-on effect for Sprint 3. Sprint 3 will focus heavily on the building and testing of the UAV and its off-the-shelf components. Additionally, our client/advisor has suggested a different AI approach than the Artificial Neural Network(ANN). We will pursue this development during Sprint 3. Lastly, after meeting with the CENG/EE team several times, it was determined that while there is an opportunity for collaboration, time frames for both teams will not support collaboration. Our team will continue, however, to work with the ME UGV team on the development of a landing pad functional for both teams.

## Team Work

- **Julian Brackins:** Worked on tasks relating to Autonomous Landing.
- **Jonathan Dixon:** Worked on tasks relating to Autonomous Landing.
- **Dylan Geyer:** Worked on tasks relating to ordering parts for the UAV,
- **Christopher Smith:** Worked on tasks relating to ordering parts for the UAV, as well as tasks relating to setting up a Simulation Environment.
- **Steven Huerta:** Worked on tasks relating to ordering parts for the UAV, as well as tasks relating to setting up a Simulation Environment.

## Completed Backlog

### Common Development Tasks

- **Setup Simulation Environment.**  
The team now has a working software simulation of the Pixhawk 4, the flight controller for this build, that utilizes both ROS and Gazebo.
- **Identify Parts Needed for UAV.**  
The team was supplied with funding source. The team needed to additionally coordinate with the SDSMT UAV Team to order correct parts, as well as parts that would be useful to both groups to ensure redundancy in the event of component failure.
- **Acquire parts needed for quadrotor**  
Received approval for the ordering of the parts. Parts ordered. Expected delivery date of 11/9/15.

**As a user, I want to communicate the waypoints to the UAV**

- **Review code that communicates with quadrotor.**

Software is available to access the flight controller through a GUI called APM Planner, available for Linux/Windows. Additionally, there is Mission Planner, available for Windows. Both will provide the ability of a user to communicate with the UAV.

- **Review code that allows a user to input waypoints.**

Both APM Planner and Mission Planner allow the user to input waypoints through the GUI.

**As an owner, I want the UAV to autonomously take-off from the landing pad.**

- **Review code that enables the quadrotor to autonomously take-off from landing pad.**

This will be handled by Mission Planner/APM Planner.

**As an owner, I want the UAV to autonomously navigate through a set of waypoints.**

- **Review previous implementation for navigating waypoints.**

This will be handled by Mission Planner or APM Planner

**As an owner, I want the UAV to autonomously return to the location of the landing pad.**

- **Review code that allows the autonomous return of the UAV to the landing pad.**

This will be handled by Mission Planner or APM Planner. The built in autonomy will bring the UAV to a position near the landing pad, where either Visual Homography, Artificial Intelligence, or combination of the two will be responsible for landing the craft. It is estimated that the craft will be within 10 meters of the designated area. Discussions with UAV team members provide an estimate of 5 meters from their observations.

**As an owner, I want the UAV to autonomously land on the landing pad without damaging the craft**

- **Review previous implementation for autonomous landing.**

The code was reviewed and is running. The software is correctly identifying the RGB lights and accurately reporting distance.

**As an owner, I want the UAV to autonomously land on the landing pad with the correct orientation.**

- **Review previous implementation for autonomous landing.**

As reported above, the software is running and is able to detect the lights. This detection will allow the UAV to orient itself to align correctly with the landing pad.

## **Uncompleted Tasks**

### **Common Development Tasks**

- **Build UAV**

This will be completed during Sprint 3. Waiting for UAV parts to arrive.

- **Test flight under manual control**

Testing will be completed by Sprint 3. Waiting for UAV to be built.

## Prototype

There is a prototype document for Sprint 2 (found here in the repository), where this same material will be covered in much greater detail. This is only a brief description.

- **Visual Homography Code**

The Visual Homography Code that was developed last year for the UAV Landing Project has been reviewed. The program successfully builds and much of it is likely to be reused towards providing the landing algorithm. The code can be found here in the repository. The code requires that OpenCV has been installed. A cmake file is contained within the directory, so that after running cmake and make the program can be run by ./tracker. The tracker is looking for RGB blobs, so it may be better for testing to have some primary colors about to test. This program is currently providing correct distance in centimeters.

- **Simulation**

To test our landing algorithms in simulation, it would be very useful to have something that approximates the Pixhawk flight controller to communicate with for the purpose of supplying instructions to the controller, as well as receiving flight data. The PX4 development team have provided both Software-In-The-Loop and Hardware-In-The-Loop simulation environments. This will require Linux 14.04, ROS Distro Indigo or Jade, and the installation of a few repositories. These are detailed well in the setup document provided by the group here. However, **catkin\_make** command did not build the meta-package correctly, as detailed in the instructions. Rather, **catkin build** will build the meta-package properly and the simulation will run with the assistance of an XBox controller (PS controllers will not work).

- **Ordering Parts**

Over the course of Sprint 2, the team met with our advisor and faculty for purpose of receiving funding to create a new UAV platform for this project. The team also met with members of the UAV team to receive assistance and guidance in purchasing hardware that would be compatible with UAV team hardware. In the event of a component not functioning, our team would be able to utilize a component from the UAV team. After building a parts list for a hexrotor, we received approval from faculty for our purchase. A complete order list will be detailed in the Prototype document.