Operations Manual

Team 12

2nd Stage Development of an Autonomous Search and Rescue Unmanned Aerial Vehicle (UAV)

Members:

Matthias Clarke – matthias1.clarke@famu.edu

Devin Justice – dsj14b@my.fsu.edu

Trent Loboda – tl12g@my.fsu.edu

Cody Rochford – ctr12f@my.fsu.edu

Marcus Yarber – marcus1.yarber@famu.edu

Qinggele ‘Gale’ Yu – qy16b@my.fsu.edu

Faculty Advisor:

Dr. Farrukh Alvi

Sponsor:

Northrop Grumman

Instructor:

Dr. Nikhil Gupta

4/7/2017

Table of Contents

Table of Figures

1.0 Functional Analysis

The purpose of Team 12's project is to design an autonomous Unmanned Aerial Vehicle (UAV) that is tasked with finding a stranded hiker near a forested area using visual clues as well as Global Positioning System (GPS) coordinates. These objectives are outlined by the Association for Unmanned Vehicle Systems International (AUVSI) Student Unmanned Aerial Systems (SUAV) competition which the team hoped to attend however could not because of student schedule conflicts as well as inadequate budget.

The team was tasked with improving the design of the previous year's team which was a Vertical Take Off and Landing (VTOL) aircraft which also had the capability to transition to horizontal flight once airborne. The students accomplished VTOL capability of the aircraft after autonomous flight development but did not implement transition flight. Vertical flight impacted the overall speed and efficiency of the aircraft however, those aspects are secondary to autonomy according to the competition guidelines. The aircraft uses the PIXHAWK flight controller to fly the aircraft in both manual and autonomous flight modes. The PIXHAWK is integrated with multiple sensors including a GPS module as well as an airspeed indicator and compass which allow the aircraft to autonomous navigate its airspace to reach user specified coordinates. The top level electronics integration can be found in Figure 1.

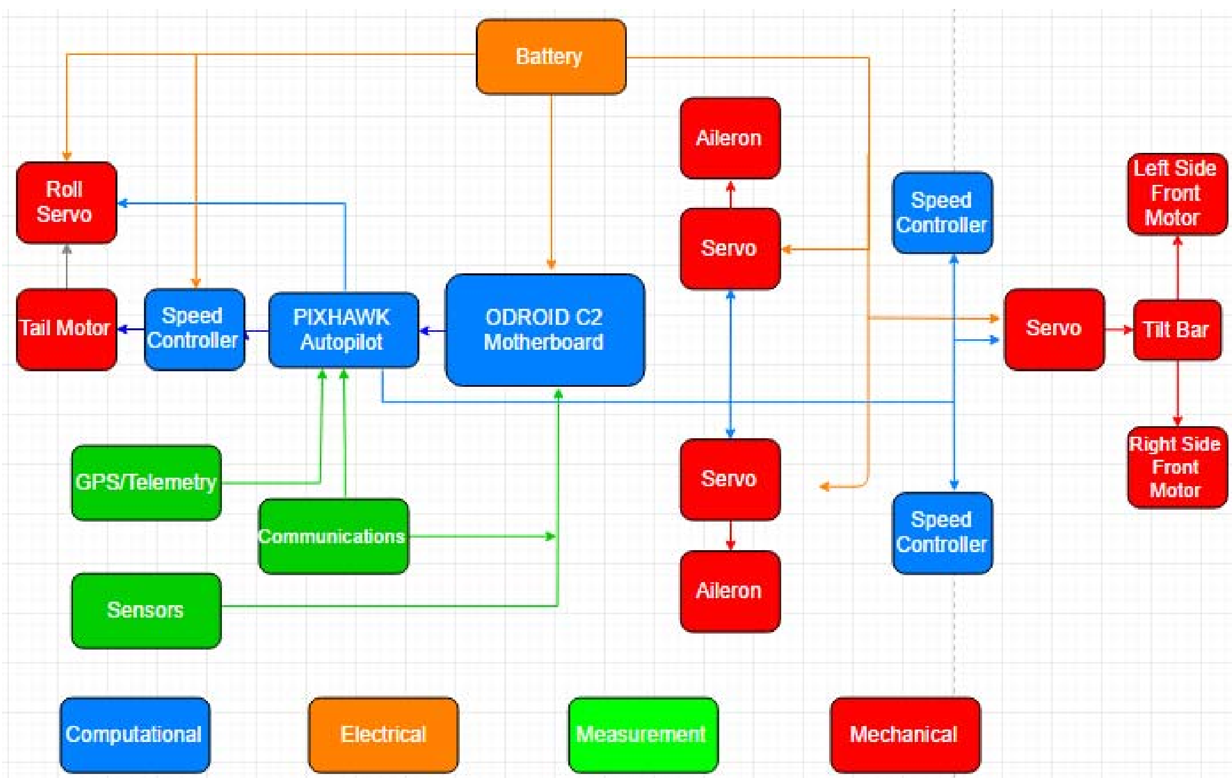

Figure 1. Top Level Electronics Design.

The radio controller is the main controller for the UAV. However, the UAV's autonomous functions can be programmed/altered via QGroundControl (QGC). QGC is a native software built to support the Pixhawk flight controller. QGC can be used as a telemetry display for the UAV during flight, a programmer to modify various parameters that control the UAV, and a mission planner to plan autonomous missions that can be flown via the flight controller.

Lastly, the payload delivery mechanism operation will be reviewed throughout the document in addition to the groundcontrol and the electronics. Components, assembly, and operation will be outlined and exploded CAD drawings will be included.

2.0 Product Specification

Because the design was being improved upon most of the components of the existing airframe were deemed viable for continued design. The motors that propel the aircraft are the Cobra 4510/28 Brushless motors, shown in Figure 1 below. The Cobra motors are 420kv motors that complete 420 revolutions for each voltage supplied to them by the battery. The continuous current of the motors is 35 which is important when considering the appropriate motor controller. The motors allow for 6 Cell LiPo batteries which are the ones used in the team's system. The motors are powered by Electronics Speed Controllers (ESCs) which draw power from the two 22.2V 5000mAh batteries integrated on the aircraft. The speed controllers are necessary because direct power is not applied from the battery to the motor however, it is precisely determined by the PIXHAWK flight controller which is linked to the ESCs.

The PIXHAWK is an advantageous flight controller because of its extensive open source development. It allows the controller to be adapted to almost any custom design and enable autonomous flight capabilities although, it can be a complicated process at times with which the students faced a lot of trouble with early on. The flight controller also features a 168 MHz / 252 MIPS Cortex-M4F which is better for this design, and 14 PWM / Servo outputs, which can accommodate all the servo, motor, telemetry, and sensor connections essential for proper autonomous functionality. Lastly, this controller is very stable as it came with the basic sensors needed such as the gyroscope, accelerometer, magnetometer, and barometer while still allowing for updates to the sensor package which the team included to improve the design. Finally, the aircraft's sensors are all digital which allows them to quick react to changing environmental conditions experienced throughout flight; this feature allows the aircraft to maintain stability. These servos are Hitec HS-5625MG servos which produce 131 oz-inches of torque, well above the minimum toque requisites to keep the aircraft aloft even at high speed.

The additional systems that the team added to the design to further accomplish the requirements of the competition are an updated GPS component, a new camera, and a Central Processing Unit (CPU) which is dedicated to processing the images that the camera takes throughout the flight to identify targets specified in the competition outline. The GPS component is the ZUBAX which is integrated with a GPS receiver, a high-precision barometric altimeter, and a 3-axis compass with thermal compensation. The reason the team decided to update the GPS was because the previous GPS component did not meet the minimum update requirement frequency of 15 Hz. The CPU is the ODROID C2 board which features a Quad Core Cortex A53 1.53GHz 64bit ARMv8 processor as well as 2GBytes of DDR3 32bit RAM and four USB 2.0 at a convenient cost of around $50. The new camera was chosen to replace the existing GoPro camera which was deemed not feasible per the system requirements as the wide-angle lens cause distortion of images at long distances which would affect target detection. Instead, the Canon PowerShot A2300 was selected because of its 16MP resolution, 5x optical zoom, 4x digital zoom, face detection feature which can help in the dynamic target detection, and its light weight at less than one pound.

3.0 Product Assembly

The full water bottle drop mechanism assembly is shown in figure 2. This shows all the parts in the mechanism with the exception of the method of attachment to the underside of the UAV. Figure 3 displays the exploded view of the assembly and Figure 4 displays the numbered exploded view.

Parts list as shown in Figure 4:

1. Assembly Base
2. Clip Arm
3. 8oz Water Bottle
4. Attachment Bracket
5. HiTEC HS-81 Micro Servo
6. Servo Rod
7. Clip Arm Pin
8. Servo Arm

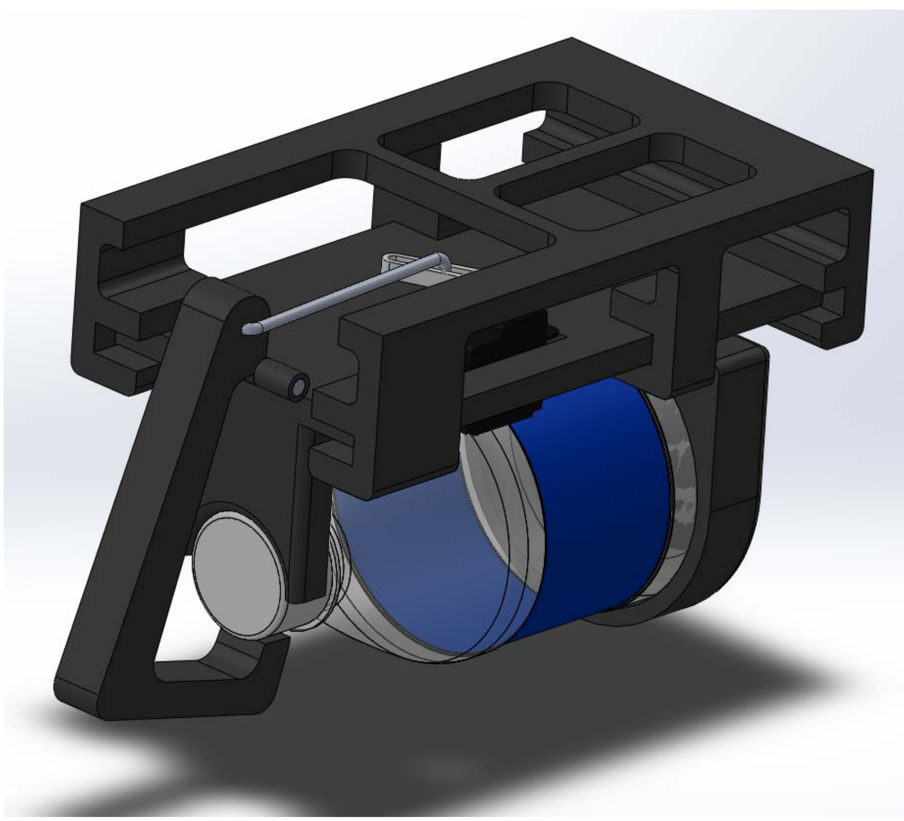

Figure 2. Full Assembly of Payload Delivery Mechanism.

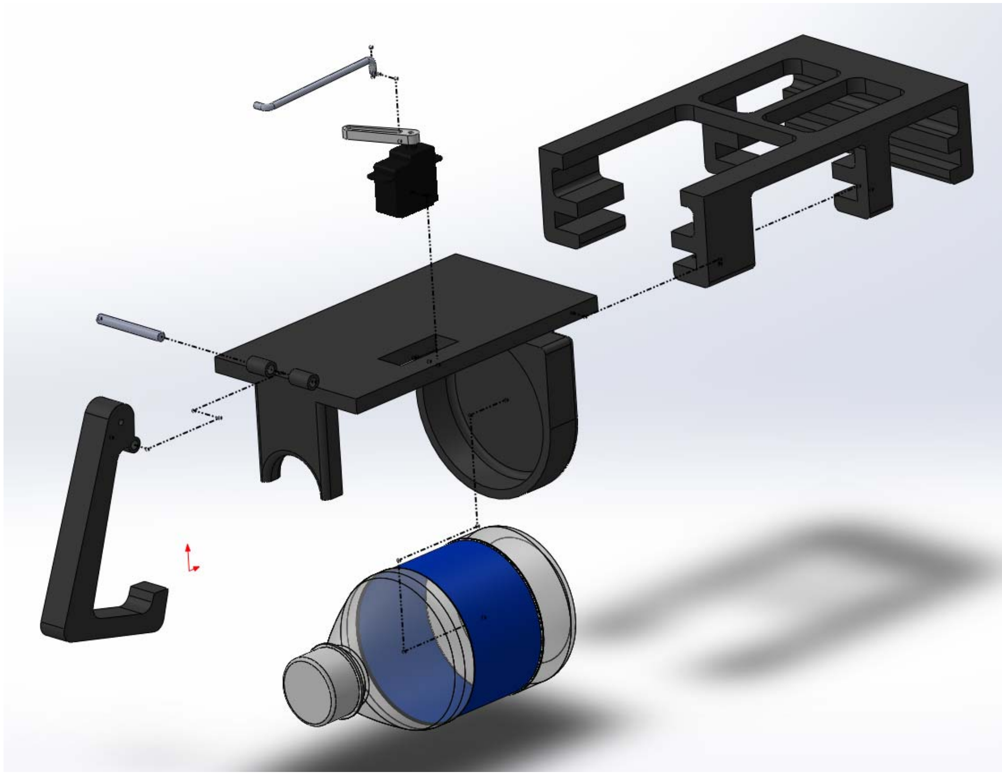

Figure 3. Exploded View of the Full Assembly.

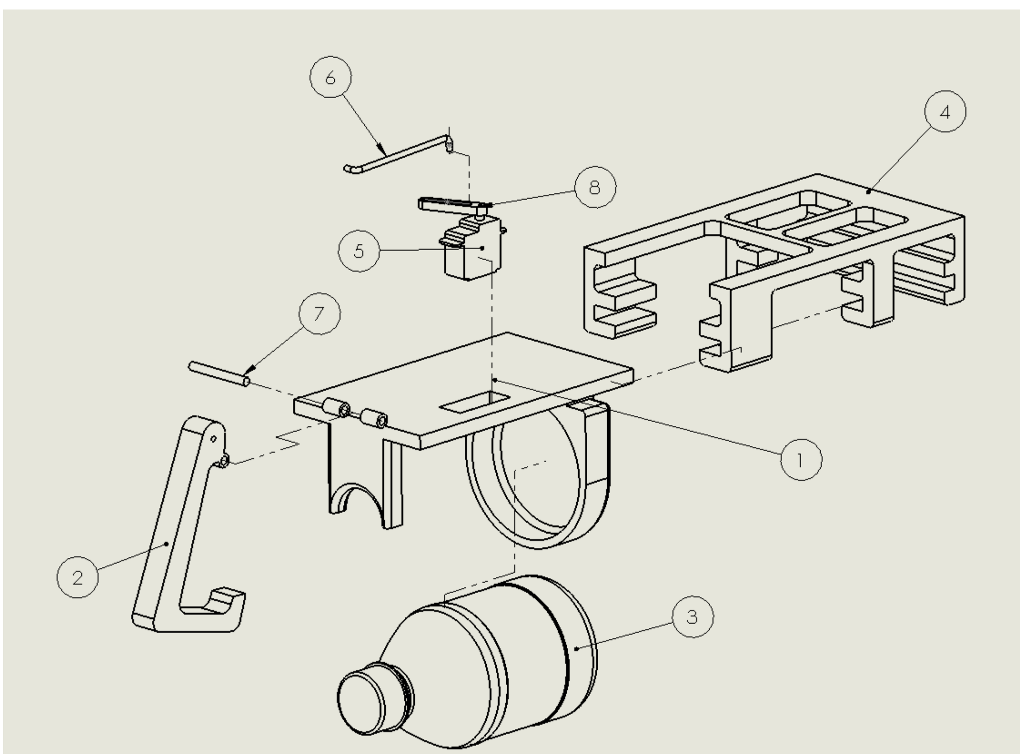

Figure 4. Exploded View of Parts.

4.0 Operation Instruction

The radio controller used for manual flight should always be available, powered on, and connected to the UAV during any flight.

A. Installation

QGC can be downloaded from the ground control website. QGC is available for a wide variety of operating systems including windows, android, and iOS. However, the recommended OS is Linux or macOS/OSX. Further installation instructions are available at the above site. (The remainder of the following instructions is based on a non-portable OS).

B. Connection

Once installed, creating a connection between the UAV and QGC can be done in two ways. On the ground, QGC can connect to the UAV via a micro USB cable (wired connection) that plugs into the side of the Pixhawk flight controller. (Arming of the UAV is generally not allowed while QGC is connected via USB). The preferred method to connect QGC to the UAV however is via the wireless 915MHz transceivers. The UAV has one transceiver onboard connected to the pixhawk, and the other connects to the computer running QGC via USB. Once both modules are plugged in and powered on, QGC will automatically detect and connect to the Pixhawk.

C. Making & Uploading Firmware

The Pixhawk flight controller is already programmed with firmware and is ready to fly. This section is included in case the firmware is ever lost and needs to be re-uploaded, or any updates to the firmware from the creators of the Pixhawk renders the UAV incompatible with the current firmware settings. Further documentation on the Pixhawk firmware can be found at pixhawk webisite. (Read completely before proceeding.)

1) Toolchain Installation/Upload: To build and compile the firmware on the operating computer, a few pre-installation tools need to be installed. The most recent version of Xcode needs to be installed before performing the following. Open a terminal and run the following lines of code (one at a time).

```
        xcode-select --install
        brew tap PX4/px4
        brew tap PX4/simulation
        brew update
        brew install git bash-completion genromfs kconfig-frontends
(part of previous line) gcc-arm-none-eabi
        brew install astyle cmake ninja
        # simulation tools
        brew install ant graphviz sdformat3 eigen protobuf
        brew install homebrew/science/opencv
        sudo easy_install pip
        sudo pip install pyserial empy pandas jinja2
(The following downloads editable code to the indicated directory)
        mkdir -p ~/src
        cd ~/src
```

```
        git clone https://github.com/PX4/Firmware.git
        cd Firmware
        git submodule update --init --recursive
(The following line of code is ran after editing the source files
discussed later to compile the software, and the line after that
uploads the created firmware to the Pixhawk connected via USB)
        make px4fmu-v2_default
        make px4fmu-v2_default upload
```

2) <u>Editing Source Files (Customizing Airframe)</u>: QGC can also be used to upload firmware to Pixhawk without doing all of the above. QGC comes with a variety of Airframes, but to create a custom firmware, the previous step and the following are used.

   a. In the created Firmware folder, find: /ROMFS/px4fmu_common/mixers/… In this folder "mixer" files can be created that control the various devices connected to the Pixhawk. Files of the type "Filename.aux.mix" control devices connected to the auxiliary ports and "Filename.main.mix" control devices connected to the main ports.
   b. In the created Firmware folder, find /ROMFS/px4fmu_common/init.d/… The files contained in this folder represent various airframes. Files can be created and deleted. Each airframe file must start with a unique id number followed by an underscore and the airframe name. They reference which mixer files should be used and set various parameter values for the Pixhawk.
   c. Once files are modified, the last two codes presented in part 1) can be used to compile and upload the firmware. This must be done every time the files are edited.

D. Creating A Geofence

Creating a geofence is recommended when using autonomy to ensure the UAV do not stray too far from the desired flight zone. In QGC find the screen represented by this icon displayed above. In the upper right corner click on "Fence." A box should show up under the checkbox and have a button in its lower left corner that says "Draw." Click this button to begin drawing a geofence. Geofences can be circular or polygonal. Multiple shapes can be drawn and overlapped. Max altitude geofences can also be created. A map background should also be display showing the current desired position. This map can be scrolled, zoomed, and changed. An internet connection is necessary to change map position and download new map tiles for offline use.

E. Autonomous Missions

In the same screen as above, in the upper right-hand corner, select "Mission." In Mission, a home position for the UAV can be specified. Consequently, a take-off point, various way points and rally points can be created on the map. Waypoints will be flown by the UAV in the order created. The altitude and loiter time of each waypoint can be modified. (Note: loiter time is generally used for multi-copter airframes, as fixed-wing airframes must continue to move.)

F. Switching Between Autonomy / Manual Control

Switching flight modes can be done via the QGC or the radio controller. The knob on the radio can be used to switch between up to six (6) or more flight modes. Flight modes include: *Manual*, *Mission*, *Takeoff*, *Landing*, and a couple more. Besides *Manual*, the remaining flight modes are autonomous. *Mission* performs the mission created in part E). *Manual* mode is set to mid-range, *Land* is set to the minimum, and *Mission* is set to the maximum.

G. Switching Between Multi-Copter / Fixed-Wing

Fixed-Wing Mode Not Recommended (Still Experimental). Switching between fixed-wing and multi-copter mode can be done via the Channel 5 switch on the radio. The transition between modes is completely autonomous (handles by the Pixhawk), but various parameters can be modified via QGC. Transition occurs when the front bar rotates approximately 90 degrees between each mode. Transition is based generally based on wind speed. Therefore, transition may not occur immediately if the UAV does not detect appropriate values for the mode be transitioned to.

H. Modifying Flight Parameters

A range of flight parameters may be modified via QGC (even during flight, although this is highly not recommended). In QGC, upper-left hand corner, find the screen represented by the icon above. The menu on the left can be used to calibrate various aspects of the UAV. Find "parameters" and click on it. This will display all the editable parameters. Edit these parameters wisely and avoid editing unknown/uncommon parameters.

I. Flight View

During flight, QGC can display a wide variety of telemetry and the UAV's live position on a map. Switch to the flight view (See Icon Above). On the right of the screen, a compact module displays Cardinal Direction, Roll/Pitch/Yaw, Altitude, and Speed. The set mission specs can also be seen. In the upper-menu (always shown), GPS location and satellite count, along with the battery percentage and radio connectivity can also be seen.

J. Mission Logs

At the end of each mission, flight logs may be saved via QGC. Click on the *Log* icon in the upper menu bar, select the log and click download. The information from the UAV will be downloaded to the specified save file.

K. Attaching Payload Delivery Mechanism

The Attachment Bracket is designed to allow a Velcro strap to pass through it and secure it to the underside of the UAV by threading it through the slot in the center of the fuselage.

L. Loading Mechanism

The Assembly Base houses the Clip Arm and Servo as they are affixed to their proper spots with a pin as shown in Figure 3. The Servo Rod and Servo arm are attached to the servo and then connected to the Clip Arm to complete the moving parts of the mechanism.

The Assembly Base with the remaining attached parts can simply be slid on the bottom track of the Attachment Bracket until the front of the Assembly Base is fully in the front slot of the bracket as shown in Figure 3. Note that the fit is very snug and requires a bit of hand pressure to properly seat the base into the bracket.

M. Attaching Servo Motor and Loading Water Bottle

The servo chord (not shown) is threaded through the slot in the fuselage and connected to the auxiliary 6 port on the PixHawk Flight Controller. This auxiliary port is programmed to the “flaps” switch on the DX8 Transmitter. Once the electronics are turned on and properly connected, toggling the flaps switch through the three positions will rotate the servo in 45 degree increments. The switch must be in the lowest (open) position for the Clip Arm to be open. Once open, the water bottle can be seated in the base and the switch can be moved to the highest (closed) position to rotate the arm and lock the water bottle in place. Ensure that all connections are secure and the bottle is properly sitting in the base before moving further in operating the UAV.

N. In Flight Operation

When it is desired to drop the water bottle:

- Ensure that the target area is clear of any persons and/or property
- Toggle flaps switch from closed to open position to release water bottle

Currently, the water bottle drop mechanism is operated manually using the DX8 Transmitter. This requires the operator to physically toggle the switch when the UAV is approaching the target for the water bottle to hit its target.

Future development can implement a formula that takes data from the confirmed target as well as the position and speed of the UAV relative to the target and calculate when exactly the bottle should be dropped.

5.0 Troubleshooting

• If the water bottle and Clip Arm is jammed:

Toggle “flaps” switch on DX8 Transmitter to open the arm and free the jammed water bottle.

If jam does not come free, shut down all electronics onboard the UAV and manually turn the servo arm and open clip to free the bottle.

• If arm opens and bottle does not fall out:

The base is designed to hold the bottle snug. If the bottle does not fall out when the arm is open, simply sand or file down the ridges on the bottle cap until it no longer sticks

• Servo arm default position is over extending the Clip Arm in either direction:

This may happen as the servo becomes uncalibrated over time. Take a Philips head screwdriver and remove the servo arm from the servo. Ensure that the “flaps” switch is toggled in the closed position and close the Clip Arm to secure the water bottle. Once it is in the correct position, screw the servo arm back onto the servo and test to make sure the servo fully opens the clip and releases the bottle.

• If any parts become damaged or broken

They must be reprinted using the 3D print files included with the UAV

6.0 Maintenance

A. Batteries

1) Store between 50-70% charge in LiPo storage bags.

2) Check voltage before use.

B. Wiring

1) Check for damage or kinks in the connections.

2) Ensure that connections are complete and none are partially connection.

C. Calibration

1) Calibrate on-board sensors after updating firmware.

7.0 References

[1] Kade Alley. (2016). Team 8 Operation Manual.

[2] Lorenz Meier, P. T. (2011). PIXHAWK: A System for Autonomous Flight using Onboard Computer. IEEE International Conference on Robotics and Automation.

[3] Pixhawk Home Page Retrieved from Pixhawk Autopilot: https://pixhawk.org/modules/

[4] Tedrake, A. J. (2015). Pushbroom Stereo for High-Speed Navigation in Cluttered Environments. IEEE International Conference on Robotics and Automation.

[5] Saurabh Ladha, D. K. (n.d.). Use of LIDAR for Obstacle Avoidance by an Autonomous Aerial Vehicle. IEEE Conference on Robotics and Automation.

[6] DJI Phantom 4: Finally an Obstacle-Avoiding, Object-Tracking Quadcopter. (n.d.). Retrieved from Makezine:http://makezine.com/2016/03/01/dji-phantom-4-finally-an-obstacle-avoiding- object-tracking-quadcopter/

[7] Competition Rules SUAS 2017. (2017).

[8] Luber, Wolfgang. "Dynamic Landing Loads on Combat Aircraft with External Stores Using Finite Element Models." *European Aeronautic Defence and Space Company – EADS* Web. 25 Nov. 2016.

[9] Witkiewicz, Wit. "Properties of the Polyurethane (PU) Light Foams." *Advances In Materials Science* 10th ser. 6.2 (2006): Web. 25 Nov. 2016.

[10] Callister, William D. *Materials Science and Engineering: An Introduction*. New York: Wiley, 2000. Print.

[11] Pixy Camera Detect the Colour of the Objects and Track Their Position. (n.d.). Retrieved from OpenElectronics: http://www.open-electronics.org/pixy-camera-detect-the-colour-of-the-objects- and-track-their-position/

[12] Rosebrock, A. (2015, May 4). Target acquired: Finding targets in drone and quadcopter video streams using Python and OpenCV. Retrieved from http://www.pyimagesearch.com/2015/ 05/04/target-acquired-finding-targets-in-drone-and-quadcopter-video-streams-using-python-and- opencv/

[13] Jangwon Lee, J. W. (2015). Real-Time Object Detection for Unmanned Aerial Vehicles based on Cloud-based Convolutional Neural Networks.

[14] K. Senthil Kumar, G. K. (2011). Visual and Thermal Image Fusion for UAV Based Target Tracking . InTech Open.

[15] Oleg A. Yakimenko, E. A. (2015). Autonomous Aerial Payload Delivery System “Blizzard”. 21st AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar. Dublin,Ireland.

[16] Chris Archer, O. Y. (2012). Enhancing SOF through UAV Pinpoint Payload Delivery. SOF Mobile Systems Focus Day. San Diego.

[17] Academy of Model Aeronautics National Model Aircraft Safety Code. (2014, January 1).