



BRIGHAM YOUNG UNIVERSITY
AUVSI CAPSTONE TEAM (TEAM 45)

Advice from 2019 AUVSI Team

| ID | Rev. | Date | Description | Author | Checked By |
|--------|------|-----------|-------------|--------------|------------|
| AF-012 | 0.1 | 3-19-2018 | Created | Kameron Eves | [CHECKER] |

Introduction

This document has been made to help the AUVSI 2019-20 team have a strong start and provide them with advice and knowledge about their potential resources. It is a loosely organized list of advice, as well as contact information for everyone who was previously on the team.

Best of luck! *AUVSI 2018-2019 Team*

Design and Development

“This is a very complex systems engineering project—more so (in my opinion) than almost every other Capstone project. Don’t rely on the pacing that the Capstone class demands; demand more of yourselves and don’t make excuses for not getting the plane up in the air. Look into different project management techniques that emphasize iterative testing and agile development. It all seems easier at first than it ends up being, so don’t rest on your laurels, even if you already have experience with ROS, ROSplane, ROSflight, aircraft design, etc.” - Andrew Torgesen

“As soon as the draft rules come out for the competition (August-September) I would check the point percentages and concentrate your time on the critical aspects of the competition that you need in order to compete. Only after you have those critical pieces working well I’d add additional bonus functionality.” - Derek Knowles

“Holding mock competitions early will be invaluable in your preparation” - Ryan Anderson

“Make your flight test time count. This is done primarily by testing your subsystem as best you can beforehand in the lab so you’re really ready to test in the air. We spent an inordinate amount of time setting up networks and trying to get software running correctly, while most of these wrinkles could have been ironed out in the lawn in front of the EB, and some of them even in the lab. We found that a “mock flight test” where we set up everything outside (where we had gps signal) but didn’t actually fly was extremely helpful and required negligible travel time.” -Ryan Anderson

“Keep and follow a checklist. There are way too many details for one person to keep straight 100% of the time. If something unexpected happens, add it to the checklist. It may seem unnecessary and monotonous, but it saved us from many crashes.” -Ryan Anderson

“When you run into an issue during flight testing, write down what the cause was, as well as the solution in a place where all team members can view it. Chances are that the issue

will resurface down the line, and everyone should have access to information for fixing the problem, should the need arise.” - Andrew Torgesen

“Do unit testing when you add new features; the imaging team was a great example of how to do that right this year, and their software was more mature and reliable in the end.” - Andrew Torgesen

“It is 100% worth your time to learn good software practices, and then use them. The vast majority of your work will be software development. I’m totally willing to come during capstone time every other Friday and talk about git, unit testing, continuous integration, etc as needed.” -Tyler Miller

“I think what would help with handing off from year to year is getting help from underclassmen. Many people are willing and want to help. They would learn a lot and could be on the future team. Then there wouldn’t be a blind hand off every year. If there is good leadership and delegation, having more help will not hold you back. They can work on the non crucial tasks, so you aren’t depending on their success. But they would learn a lot in the process and be in the loop.” -Brady Moon

“Consider dependencies between subsystems when planning schedule - for example the UGV drop relies on the full autopilot/planning stack before it can be tested.” - Jacob Willis

“Find ways to maximize the amount of time spent flying autonomously on flight tests. It takes a lot of time to travel to/from good testing areas, so make sure that the time there is very effective.” - Jacob Willis

“Find a good, repository friendly way to share code between the flight computer and personal computers without an internet connection. I think it should be possible to set the BRIX as an alternate remote for the odroid - that way git commits can be communicated without an internet connection.” - Jacob Willis

“Debug the simulator! Get everyone set up to run the full system on their laptops - this will help when gaps need to be filled in.” - Jacob Willis

“We had problems with the Ubiquiti until we turned off the 5GHz wifi on the wifi router. There are some guis you can access at the IP address of the lightbeam and bullet. These are useful for debugging network issues.” - Jacob Willis

“When doing mock competitions (or testing individual subsystems), make sure to keep track of your score. Imagining what you would have scored is not nearly as useful to you as figuring out exactly what you would have scored and recording it somewhere. This will give you a clear measuring stick to base your progress on.” - Tyler Critchfield

“I would make sure that by the end of the school year, the team has done multiple

mock competitions (with everyone participating in their anticipated roles), while timing everything just like the real thing. After finals end, people will leave for internships or graduate school and you'll likely have less time and manpower than you anticipate." - Tyler Critchfield

"In the early months of both semesters (Sep-Oct, Jan-Feb) you will likely have more time to devote to flight tests because your other classes will not be as busy yet. Take advantage of that time if you can, especially in the fall when the weather is nice." - Tyler Critchfield

"Flight test locations pros/cons:

- Rock Canyon Park:
 - close and quick to reach
 - has power and a covered pavilion
 - does not have much area to fly in and is near houses
 - There seems to be a downdraft coming out of the canyon (southeast corner of the park) that caused one catastrophic crash and almost another
 - Good for quick RC tests, not for autopilot - if possible avoid flying here due to safety concerns
- Springville (off the freeway)
 - Lots of space (just stay away from the buildings and freeway behind you)
 - Close-by
 - We had lots of connection issues that led to crashes
 - Technically within 5-miles of the Spanish Fork airport
 - We recommend no flying here just from our experience and proximity to the airport.
- Lehi Airspace (<https://www.google.com/maps/place/Utah+County+Radio+Controlled+Airplane+111.900469,701m/data=!3m1!1e3!4m8!1m2!2m1!1slehi+rc+airspace!3m4!1s0x874d7f21d54370cd:0111.9019458>)
 - No power
 - No car access unless someone else is already there
 - If the gate is locked you can walk everything in if you have a generator. We once did it with power from a car, but the extension cord is only so long and it was a weird set-up all together
 - More space than Rock Canyon, but still not a lot
 - Be careful of a river which would be bad to crash in (also lots of reeds, planes get lost in those frequently)
 - Nice area to take-off and land, with grass and asphalt (more similar to competition environment)

- Often is frequented by other RC enthusiasts. Two older gentlemen seemed to be there pretty much every morning. Pretty soon they didn't really like us being there taking up their time and space, and they let us know that there were rules we were breaking. One is that the pilot needs to have some sort of insurance. If you fly there, I think it's around \$20 or so, and could easily be covered by the budget. They also probably thought we were to blame for one of their crashes (even though at the time our system wasn't even on).
- We decided it wasn't worth the hassle to continue going there, but could be an option if Goshen doesn't work for some reason.
- Goshen/Elberta (39.983238, -111.990451, <https://www.google.com/maps/place/39%C2%B058'59.111.992484,706m/data=!3m2!1e3!4b1!4m14!1m7!3m6!1s0x874d07d44e19369b:0xa13b930b4d04edb111.9007716!3m5!1s0x0:0x0!7e2!8m2!3d39.9831504!4d-111.9902901>)
 - Further away (~40 min one-way)
 - No one else to worry about besides off-roaders and cows
 - No easy spot to land (rocky ground)
 - Lots of space to fly in, plenty to simulate the competition space
 - Our chosen spot to fly by the end of the year
- Most other areas that could be good around Provo are inside of the restricted airspace of an airport. Keep this in mind when looking for other spots to fly."
 - Tyler Critchfield

Airframe

"There is a good chance you will crash at some point. Choosing an off-the-shelf airframe made rebuilds extremely easy, since parts could simply be purchased and replaced. Even then, catastrophically crashed airframes that seem irreparable can often be fixed with a little TLC in the lab. During one flight test, we fell 100 feet straight down, broke the airframe into 5ish pieces, and had it airworthy again in just a few days. You just need to get really good with gorilla glue and water, fiber tape (get the 3M packaging stuff- it will change your life!), and making sure the hatch fits when you're done." -Ryan Anderson

"The 3M tape that Ryan mentioned can be found here: https://www.amazon.com/Scotch-Shipping-Strapping-Designed-8959-RD/dp/B001AFKV0S/ref=sr_1_5?keywords=3m+packing+tape+fi5" - Tyler Critchfield

"Airframes can take a rather large beating. Lots of tape and glue can fix most things. A heat gun can fix the rest. It is a lot faster and cheaper to repair than to make a new plane." -Brady Moon

“The MFD Nimbus Pro is advertised as having a larger wingspan than the MFD Crosswind. We bought both and they were identical.” -Ryan Anderson

“Before flying, make sure you do extensive range testing at each new location. A quick test before each flight couldn’t hurt either. A couple of our crashes during the year were due to a loss of connection, which if I remember right was a fault of the transmitter (something was loose). We actually crashed a couple of times into a tree, which saved the plane and would have been disastrous if it had crashed elsewhere.” - Tyler Critchfield

“Try experimenting with airspeed. See how it affects imaging and waypoint accuracy in flight tests. Flying slower may result in higher accuracy and image quality, but flying faster will help you finish the mission faster. According to the 2019 rules, you have 20 minutes to complete the competition for full points. If you decide to fly faster, you may not need the foam tail wedges we added.” -Ryan Anderson

“Try and make a better camera mount that guarantees the camera points straight down. Target geolocation was off and this could have contributed to why.” -Tyler Miller

“Buy some new batteries” - Jacob Willis

“On the subject of batteries, make sure someone is able to give you a safety training on their handling. If handled properly you shouldn’t have any problems, but it’s always better to be cautious. One time I unplugged the battery connection cord from the charger before disconnecting the battery, and there was a huge shock and I’m lucky I wasn’t hurt. I think someone else on the team may have had a similar experience. You probably want at least one person in charge of their care, and ideally two or three people who know how to charge and discharge them.” - Tyler Critchfield

“You may try testing a new airframe using R/C only (without all the expensive electronics inside). Note that a lighter airframe can have significantly different flight dynamics even if the center of gravity is in the same location. This took our pilot by surprise once and caused a serious crash.” - Ryan Anderson

Controls

Controls

“Look into pulling the latest ROSplane version from the MAGICC Lab into the BYU-AUVSI ROSplane fork. Or maybe just use the latest MAGICC Lab version.” -Derek Knowles

“ROSplane is good but not perfect. It might be because we were using an older version but there are a number of edge cases that were not covered by the code and led to undesirable behavior (e.g. when planning fillet paths, if the angle is too small or large between the waypoints, the halfplanes get set in incorrect locations). The vanilla ROSplane is also set up to just fly straight line segments or orbit segments which makes it a bit tricky to do a dedicated loiter” - John Akagi

“One of the biggest problems that I saw from both last year’s code and this years is that the airplane starts planning its path from where it is but by the time it has finished the path, the plane has moved. This can lead to the plane flying through areas it wasn’t intending.” - John Akagi

“Get a good simulation and simulation display up and running asap. Gazebo works fine for simulating flight but is terrible for understanding what the plane is actually doing. The missionPlanner/plotter.py was my very last minute attempt to at least get an idea of what the plane was doing.” - John Akagi

“Also might be a good idea to replace the Odroid on the plane or at least have an extra. It’s been through a few crashes” - Derek Knowles

“If you’re not going to fly straight-line paths (straight-line path manager), you have to make modifications to either the fillet/dubins path managers themselves or be smarter about how you specify waypoints in your path planner in order to avoid having your plane fly off the map without warning.” - Andrew Torgesen

“Do extensive testing in simulation (make sure the simulation works—it currently doesn’t work), and use good programming practices with git and unit testing as you add new functionality. It is worth the time and effort to integrate good programming practices.” - Andrew Torgesen

“Make sure you are not relying on one or two individuals to handle all things related directly to autonomous flight. Most team members should be proficient in ROS and should be able to debug ROSplane or other ROS packages should issues arise, as they will. We were hit hard by the fact that the controls subteam, consisting of three individuals, was away for most of the summer leading up to the competition.” - Andrew Torgesen

“Get up in the air as early in the development process as you can to tune the ROSplane gains. That will make your life a lot easier and facilitate testing for other subsystems that needs to get done.” - Andrew Torgesen

“ROSPlane and most of the other MAGICC lab stuff have migrated to Ubuntu 18.04 and ROS Melodic. It is probably a good idea to follow, but make sure you know how the whole system works and flies before switching” - Jacob Willis

Imaging

“I mainly worked on the image classification gui in the BYU-AUVSI/imaging repository. Suggested improvements can be found in the header of the main gui.py and inside of each tab file inside of imaging/client/lib. Tkinter is pretty basic and there a number of random bugs with the gui. When you have questions about design rationale, comments in the code, etc., please reach out to me” - Derek Knowles

“I built the server and some of the autonomous imaging stuff. A few suggested improvements are listed in github issues on the Imaging repo. The server is pretty stable and the items listed are enhancements. Please reach out to me and I can meet with you in person to explain how to use/improve it.” -Tyler Miller

“We never had a chance to extensively test target geolocation in flight. It would be worth testing and tuning. Day before competition target geolocation was off by 250+ft” -Tyler Miller

“Even though we were the only team to score an autonomous ODLC, there’s still a lot of work to be done on the autonomous system. One of our main holdups was having a good, large library of image data to test against. There’s a hard drive in our cupboard with a bunch of data that we either collected or generated this past year. It will require some organization/ categorization (ie: labeling images that have targets in them, etc), but it will be worth it if your looking to improve autonomous.” - Tyler Miller

“Again with autonomous, here’s what we wanted to improve, but didn’t have time: We wanted to add a class to our letter classifier for ‘no-letter’ (similar to the shape classifier’s ‘notarget’), and then use this to help reject false-positives. We also wanted to make a net for color classification, as we had a hard time getting this to work accurately (whether a simple net or a CNN is best for this, we’re not sure). Our strategy for orientation classification worked okay but not great, and it would be good to brainstorm ways it could improve. Jake Johnson was the main lead on autonomous development (he’s in the MAGICC lab) and I helped with creating the datasets and nets.” - Tyler Miller

“Find someway to access a computer with an NVidia card and CUDA installed on linux. It will save you so much time when training the nets” -Tyler Miller

UGV

“Progress on the UGV was slow. We made a good choice of parachute and a fairly reliable payload bay, but we didn’t have a good system for testing the UGV drop code until it was too late to make improvements to it. Having a plane that you can fly an autopilot

on and having code that is modular enough to not get stuck waiting for other people to finish will help you make better use of your flight test time.” - Jacob Willis

“The UGV electronics consisted of a battery, a 5V BEC, a Openpilot Revo board running custom code based on Airbourne.f4, a gps, and a 433 MHz transmitter. Turns out the transmitter is not FCC compliant, so it can’t be used in the competition - I’d recommend getting a set of the 3DR 915 MHz radios for communicating. The flight tents are only ~200 yards from where the UGV drop was, so range shouldn’t be a huge concern. Getting a good heading estimate was hard - GPS was too random and the magnetometer wasn’t reliable. I would recommend building a car that has encoders on it so you can at least drive a straight line far enough to get a good heading estimate.” - Jacob Willis

“The runway where the UGV drop was had lots of weeds growing in the cracks. It looked like they mowed it beforehand, but even then our little UGV would not have made it through the weeds.” - Jacob Willis

“The most successful UGVs had 4” tires on a platform that could land on either side. Only 3 made it to the drive target - this will likely be a major part of the competition next year as well.” - Jacob Willis

Competition

“Be fully prepared to fly when you arrive at the competition. You have minimal setup time and it’s crucial that you be in the air as soon as possible once it’s your turn to fly.” - Derek Knowles

“During setup, do not get casual or overly rushed with the checklist. Even though you have limited setup time and it may be tempting to take shortcuts, you may regret it in the end. We forgot to update two parameters on the flight line and, as a result, our telemetry was off by about 50 miles. This would have been avoided had we stuck to the checklist.” -Ryan Anderson

“Your time in Maryland will be much more enjoyable, less stressful, and more fulfilling if you check an UAS at the airport that is competition-ready. You’ll have some time minor improvements, but don’t expect to have time to fix major problems while you’re there.” -Ryan Anderson

“It seems that most teams spend a lot of time on webster field watching and supporting other teams. I think doing this does a lot to build a relationship between teams and universities and I wish that we had been more prepared so we could have spent some time making connections.

The judges did notice that we weren't around for the first two days of the competition."
- Jacob Willis

"Things got very rushed after we got out to the flight line. Roles had been defined, but not well practiced. The tent we were given had a table with three chairs. This was just enough to set up the groundstation, but we needed much more space for the imaging team. We drove our van out to the flight line, so we set up the imaging team in the van. This worked very well." - Jacob Willis

"-To avoid loss of power, we used the power from the van to run our network. This worked fine, though it would have been nice (and more professional) to have a backup supply. We used the competition power to keep laptops charged. There were no issues with the power (that I know of).

-We got the interop cable as soon as we arrived, so there was lots of time to get mission data, and plan paths.

- Our pilot was apprised of the waypoints we were flying, but didn't know where the search area was. This reduced the ability we had to find targets.

- Targets that were imaged were very easy to manually classify. The camera gave very clear pictures, even from our high altitude.

- The UGV drop is extremely close to the viewer stands, which makes it very important to have good control of the aircraft while doing this. We received an infraction for crossing that boundary, losing all of our autonomous flight points. In addition we had a low altitude infraction while attempting autonomous UGV drop because we set our altitude to 30m instead of the typical 40m we were using for the other waypoints.

- Because we weren't confident in our path manager, we loitered for three minutes before attempting to follow the waypoint path. This cut into our imaging time significantly.

- It is wise to keep the camera off unless over the search area. This both helps to reduce the number of empty images that must be manually processed and it reduces the likelihood of a false positive from the autonomous imaging system. To aid this, consider modifying and distributing the missionPlotter in Metis so anyone on the team can have a live update of where the plane is.

- The waypoint path might be slightly too long for keeping a telemetry connection. I think the judges were not impressed by the spotty telemetry we got at the farthest end of the path. It would be good to discuss this with Craig (MAGICC lab) and find out if there is a good way to do a combination of a close range omni antenna and a long range, high-gain antenna. This would require a different lightbeam antenna. Consider getting the Ubiquiti Rocke Prism AC along with one of Ubiquiti's long range antennas.

- It was very windy while we were there. This added significant disturbance to our flight and made the path follower and altitude hold controllers work hard to stay on track. Both of these could use more tuning.

- Make sure to be keeping track of the mission time. We used ~28 minutes. The breakdown was approximately:

- 1 minute - Takeoff
- 3 minutes - Autonomous loitering (to make sure we had Autonomous flight)
- 11 minutes - flying waypoints
- 5 minutes - attempting UGV drop
- 5-6 minutes - search path
- 1-2 minutes - wrapping up target submission and landing.

Since our imaging system was so good and our UGV so unreliable, it would have been much better for us to fly more of the search area than to even attempt the UGV drop.

There was a decision to perform manual search because the autonomous search was too long. Much of the area we searched was searched before - having a smarter way of planning the autonomous search, or having it plan wider paths that get ~70% of the search area fast might be a better approach.

- The judges mentioned they want to see more preflight testing being done by all of the teams.

- Many teams had their groundstation in a pelican case. We purchased a plastic tub for it and found it extremely helpful to keep things organized and plugged in. It is ready to go for next year.” - Jacob Willis

Travel

“The airframe box was oversized and cost upwards of \$230 each way to check on the airline. You might look into FedEx overnighting it.” - Ryan Anderson

“Last year’s team had difficulty finding a place to perform flight tests in Maryland before their competition time. Chaptico Park was relatively close, had plenty of space to fly, and was fairly deserted throughout our stay. There was a power outlet and a covered area with tables that was very convenient for setup.” - Ryan Anderson

“If you plan to go to chaptico park bring 150’ of extension cords” - Jacob Willis

“We transported our equipment in one large cardboard box and three suitcases. The large box had two fuselages and wings in it, while the suitcases had tools, electronics and other equipment. The box was oversize, so it was \$230 each way to fly it.” - Jacob Willis

“BWI is the closest airport to the competition, but we all wanted to see Washington DC, so it might be worthwhile to try to fly into Dulles or Reagan.” - Jacob Willis

“Capstone didn’t pay for any of the rental cars or hotel rooms before we arrived, so we had to put them on our credit cards. Make sure there are people on the team with high enough credit limits to do this (or, even better, make sure capstone pays for things before you leave).” - Jacob Willis

Other Subjects

“Make a solid effort on the paper and videos - they will take longer to make than you think they will. Luckily we had Brandon who knew how to put a video together, but otherwise we would’ve had a poor video to submit. This could be a good assignment for an undergrad who wants to help - they could be in charge of filming tests, taking pictures of candid work sessions, etc. Even set up an instagram account to build public interest. The video and paper will be much easier if you plan for them in advance (for instance, make sure everyone reads the requirements in September when the rules come out and review them periodically.” - Tyler Critchfield

“Be consistent with taking pictures and video as you develop the system so you have plenty of material to work with for the videos.” - John Akagi

“I think most of our success in the paper and video came from trying to model them after the highest-scoring papers and videos from the previous year. Pay attention to and follow exactly what the judges are looking for. Typically they want to see data - plots and tables - that show you’ve tested the system and have an idea of how it will perform.” - Tyler Critchfield

“To save some time, I would consider asking Capstone if your final report for the class can instead be the competition paper. Then you don’t need to write up multiple reports when everything is in crunch time.” - Tyler Critchfield

Appendix A: Contact Information

| Name | Email | Phone |
|--------------------|----------------------------|--------------|
| Akagi, John | akagi94@gmail.com | 858-231-4416 |
| Anderson, Ryan | rymanderson@gmail.com | 208-789-4318 |
| Critchfield, Tyler | trcritchfield@gmail.com | 206-939-8274 |
| Eves, Kameron | ccackam@gmail.com | 702-686-210 |
| Johnson, Jake | jacobcjohnson13@gmail.com | 801-664-7586 |
| Knowles, Derek | knowles.derek@gmail.com | 405-471-4285 |
| McBride, Brandon | brandon.mcbride4@gmail.com | 801-520-9165 |
| Miller, Tyler | tylerm15@gmail.com | 385-399-3472 |
| Moon, Brady | bradygmoon@gmail.com | 435-828-5858 |
| Olsen, Connor | connorolsen72@gmail.com | 385-230-3932 |
| Torgesen, Andrew | andrew.torgesen@gmail.com | 661-210-5214 |
| Willis, Jacob | jbwillis272@gmail.com | 208-206-1780 |

Appendix B: Help Index by Person

We are all committed to helping you out. Please reach out to one or multiple people when you have questions or issues.

| Name | Subteam(s) | Specialties |
|--------------------|--------------------|--|
| Akagi, John | Controls | Metis, Theseus, |
| Anderson, Ryan | Airframe, Controls | Aircraft aerodynamics, weight distribution, Groundstation, plane construction |
| Critchfield, Tyler | Airframe | Aircraft aerodynamics |
| Eves, Kameron | Airframe | Safety Pilot |
| Johnson, Jake | Vision | Autonomous target recognition |
| Knowles, Derek | Vision, UGV | Image classification GUI, Payload planner & release |
| McBride, Brandon | Vision, UGV | UGV drive, imaging server client |

| | | |
|-----------------------------------|-------------------------|---|
| Miller, Tyler | Vision | Interoperability, autonomous target recognition, imaging server |
| Moon, Brady | Controls | Metis, Path planning, RRT |
| Olsen, Connor Torgesen, Andrew | Vision, UGV Controls | Geolocation, UGV drive Groundstation, ROSplane, all things ROS, LaTeX, High-level software architecture |
| Willis, Jacob | UGV, Controls | UGV drop & drive, Avionics, Metis, ROSPlane |

Appendix C: Help Index by Resource

This is meant to list the various aspects of the AUVSI competition and provide you with a contact for each subject matter. See Appendix A for contact information for each individual.

BYU-AUVSI Github Repositories:

A6000_ros: Tyler Miller

(tools for using a6000 in ROs)

Airbourne_f4: Jacob Willis

This is a fork of the board support package used by rosflight. We used it for the UGV. Most of the code we wrote for the UGV is contained here.

auvsi_documentation_2019: Andrew Torgesen

(Source code for all reports and documentation)

Imaging: Tyler Miller

Imaging/autonomous: Jake Johnson

Imaging/client: Derek Knowles

Imaging/server: Tyler Miller

Interop_pkg: Tyler Miller

(Interoperability stuff as a ros package)

Inertial_sense_ros: Andrew Torgesen

(ROS wrapper for the INertialSense uINS2 GPS-INS sensor)

Metis: John Akagi

(Mission and path planner)

Metis/pathPlanner: Brady Moon

Metis/missionPlanner/payloadDrop.py: Derek Knowles

Metis/missionPlanner/payloadPlanner.py: Derek Knowles

ros_groundstation (AUVSI-SUAS-2019 branch) : Andrew Torgesen

(RQT plugin for interacting with ROSplane)

Rosflight: James Jackson (superjax08@gmail.com)

(ROS stack for the ROSflight autopilot)

Rosplane: James Jackson (superjax08@gmail.com)

(fixed-wing autopilot for ROS)

We used the “plane” branch on the odroid

Scripts: Tyler Miller

(scripts for ground station and onboard computers)

TechnicalDesignPaper: Andrew Torgesen

(Technical Design paper for the AUVSI competition)

Uav_msgs: John Akagi

(shared messages between ground station, interop, imaging, and autopilot)

Ugv_auto_drive: Jacob Willis

This was meant to be a higher level program that used Airbourne_f4. We ended up just developing in the Airbourne_f4 repo (not the best choice - it was just easier). There's not much here, mostly some random tidbits of code.

Capstone:

Documentation: Andrew Torgesen

Purchasing: Jacob Willis

Design & Development:

Airframe: Ryan Anderson & Tyler Critchfield

Competition Roles:

Team Captain: Andrew Torgesen

Ground Station: Andrew Torgesen

Safety Pilot: Kameron Eves

Antenna Pointing: Connor Olsen

Past Competitions Overview: Dr. Tim McLain

Appendix D: Passwords and IP Addresses

| Item | Password | IP Address |
|-----------------------|----------|-------------------|
| Cupboard Bicycle Lock | 1413 | |
| Odroid | odroid | 192.168.1.8 |
| Brix | Byuauvs1 | 192.168.1.10 |
| Imaging Server | | 192.168.1.10:5000 |
| Router | byuauvsi | 192.168.1.1 |
| Lightbeam | | |
| Bullet | | |

Advice from 2019 AUVSI Team
