Vector Field Mapping and Analysis Using Finite Sensor Swarms

A Senior Honors Thesis

Submitted to the faculty

in partial fulfillment of the requirements for the

Degree Bachelor of Arts in Physics

by

Amanda K. Slagle

DARTMOUTH COLLEGE

Hanover, New Hampshire

May 2012

____
Kristina A. Lynch, Advisor

____
Robyn Millan

____
Hans Mueller

Table of Contents

Acknowledgements

I would like to thank my colleagues and friends in the GreenCube group, with whom I have had the pleasure of working for the last four years. DUCKS has been a group effort right from the start, and would not have happened without all of them: Ralph Gibson, Yorke Brown, Tim Smith, Kevin Rhoads, Dave Collins, Whitey Adams, Chris Grant, Jean Blandin, Todd Anderson, Casey Bradshaw, Max Fagin, Jon Guinther, Peter Horak, Ben Katz, Nina Maksimova, Stephanie Malek, Ha Nguyen, Dylan Sewell, Tom Whalen, Ellen Weburg, and Patrick Yukman.

I would also like to thank Frank Magilligan for his insight into the environmental science applications of river flow data.

Thanks to my committee members Robyn Millan and Hans Mueller for their patience and all their helpful comments and suggestions, which have greatly improved this thesis.

Finally, thank you to my advisor, Kristina Lynch. When I joined her lab as a freshman I had no idea how big a part of my life it would become, and I'm so glad that it did. I have had the chance to learn about everything from soldering and ham radio to vector field mapping and plasma physics, and Kristina has been there all along with guidance and encouragement. Kristina, thank you for everything. Working with you has been wonderful.

I. Abstract

Ionospheric plasma studies are complicated by the difficulty of distinguishing between spatial and temporal variations in structure. One solution to this problem is to use a swarm of sensorcraft to take multiple measurements from a variety of positions and times. In order to explore issues related to the construction and use of such a swarm for mapping and analyzing vector fields, a finite number (20) of GPS-enabled Arduino-based sensor units were floated down the Connecticut and Ompompanoosuc Rivers. The design of the units was modified from the GreenCube4 Arduino-based payload structure, and the teamwork and infrastructure of the GreenCube group enabled the rapid development of this GreenCube5 "Ducks" array. Position and attitude data from each were collected and assimilated to produce a 2-D map of river flow and vorticity, which involved converting the finite irregular grid of sensor data to a vector field. In addition to serving as a proxy for aurora, river flow and vorticity data have geophysics applications in the study of Large Woody Debris (LWD) transport.

II. Introduction

Auroral physicists study the ways that ions, electrons, and the electric and magnetic fields that accompany them move and change in the ionosphere. One method is to send sensors flying through these plasmas via a sounding rocket and take measurements all along the flight path. The trouble with this method is the difficulty in distinguishing variations that occur along the length of the flight path from those that occur throughout the time of the flight. A more comprehensive approach would be to use a large number of sensors (a swarm) that move through a variety of positions and times and make a map of the resulting data. That is the ultimate goal: to develop a swarm of payloads capable of mapping vector fields associated with plasmas. However, testing in the ionosphere is very inconvenient and expensive – it requires space-worthy payloads and rocket launches. This thesis focuses on what can be learned about swarms and vector field mapping and analysis using ground-based experiments.

1. River Flow as a 2-D Vector Field

One important question follows from the idea of a ground-based study: where is there a convenient and interesting vector field to substitute for a plasma? The GreenCube group has been using the atmosphere up to 30 km altitude as its laboratory for study for the last 5 years, using balloon-borne payloads. However, even balloon launches present logistical challenges for large numbers of payloads. Instead, the flow of a river, though not completely analogous to aurora, serves as an instructive proxy. Both are vector fields and both describe the behavior of fluids. Additionally, rivers offer many practical advantages for testing payload swarms. There are a variety of rivers with different flow patterns in close proximity to Dartmouth College. No special technology is required to deploy payloads into rivers. Relatively short range radios (1 mile) can be used because the ground station can follow the payloads closely in a canoe.

However, several significant differences between rivers and aurora should be noted. Payloads floating down a river move with the flow field, meaning that position and orientation measurements alone can be used to map the flow and vorticity. Conversely, payloads launched into aurora move through it, so instruments such as ion and electron probes are needed. A river is a neutral rather than a charged fluid. Finally, rivers are more or less time stationary for the several-hour duration of a mission, which simplifies the data analysis significantly. Nevertheless, much can be learned about using swarms of measurement points in this time-stationary flow field.

Besides providing a proxy for aurora, river flow measurements aid in the study of the transport of LWD, usually defined as logs at least 4 inches in diameter and 6 feet long that float down or become lodged in rivers. LWD is ecologically important because when it becomes trapped it creates low-velocity pockets of water, which are vital fish habitats (*"Large Woody Debris Fact Sheet" 2*). Decomposer species consume LWD, making it a base for the food chain. LWD also stabilizes riverbanks by restricting the movement of sediment there. It helps control erosion due to flooding by "absorb[ing] and redirect[ing]" forces associated with abnormally large flows (*"Large Woody Debris Fact Sheet" 2*). Several features of the payloads in this experiment were chosen to resemble LWD in the hope that they would move and become stuck in a similar fashion. Pieces of LWD tend to have one axis that is much, often more than an order of magnitude, longer than the other two axes. Replicating this exactly would have caused the payloads to be inconveniently large to transport and store. Instead, each payload was given one long fin and two short side-fins. An aluminum weight was added to each payload to make it sit low in the water like LWD.

Rivers provide a vector field that is not only reasonably similar to a plasma but also scientifically interesting due to its role in the transport of LWD. Now that a suitable vector field has been identified, all that remains is figuring out how to create a swarm of small, low-resource, easily mass-produced payloads with which to study it. Fortunately, the GreenCube team in the Lynch Rocket Lab has a history of using low-resource, inexpensive technology for science missions.

2. GreenCube's History with Low-Resource, Commercially Available Technology

A GreenCube is a sensorcraft containing a GPS, magnetometer, and radio system as well as up to six other sensors. The design is based on the CubeSat standard set by CalPoly and Stanford Universities. A GreenCube uses a 30cm x 10cm x 10cm housing containing all electronics and scientific instruments. Our GreenCubes have been launched on sounding balloons, achieving a burst altitude of 80,000 – 110,000 feet.

The team's first attempt at taking multipoint measurements was with GreenCube2 (2009), in which two payloads were launched from the same location 90 seconds apart. The objective of the mission was to identify gravity waves from temperature and wind speed fluctuations (as observed through payload velocity). As shown in Figure 1, the team was able to identify features that appeared at different phases in the data for both payloads and on both the uplegs and downlegs of the flights. Unfortunately with only two payloads it was not possible to determine whether the phase differences were due to spatial or temporal variation, or both. The mission for GreenCube3 (2010) included an attempt to use the real-time GPS coordinates transmitted by the payload to track it with a telescope. During GreenCube3 the team started experimenting with radio systems intended for networking, and with the use of smartphones as the basis for sensorcraft.

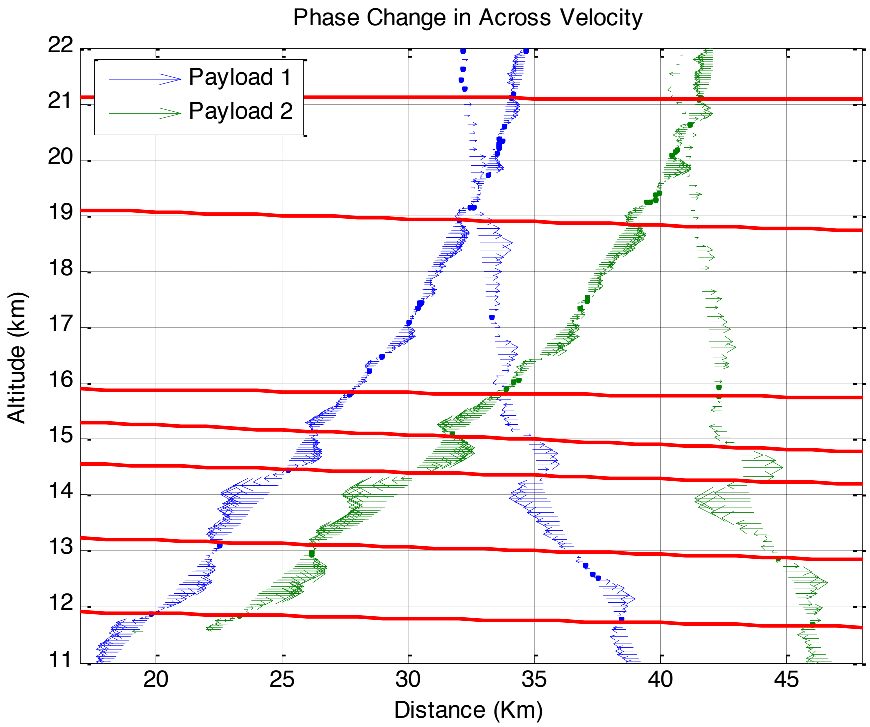


Figure 1: Magnitude of the component of payload velocity perpendicular to downrange direction. Shown on a plot of altitude vs downrange distance. Red lines are lines of constant phase. Velocities are approximately 5 – 15 m/s. Plot created by Sean Currey. (*Siddiqui slide 47*)

With GreenCube4 (2012) the group started looking forward to orbital missions. The team built a prototype solar panel with the objective of learning about solar power and using it for future missions. The GreenCube4 payload, “Dave” (shown in Figure 2), was the basis for the Duck payloads that were built and launched as part of this thesis. Dave uses DNT900 series radios in a simple star network configuration (see Methodology, section 5) similar to that which the Ducks have, but with a 10’s of kilometers range. Dave is also the first GreenCube payload to use an Arduino, an open-source electronics prototyping platform, as a control board. More information about Arduinos can be found on the Arduino website (http://www.arduino.cc) as well as in the Methodology section below. The Arduino replaced a custom control board that was designed and built in-house. This change makes the Duck payloads much easier to mass-produce than they otherwise would have been. Furthermore, the group's experience working with Arduinos for GreenCube4 greatly sped up the development process for the Ducks.

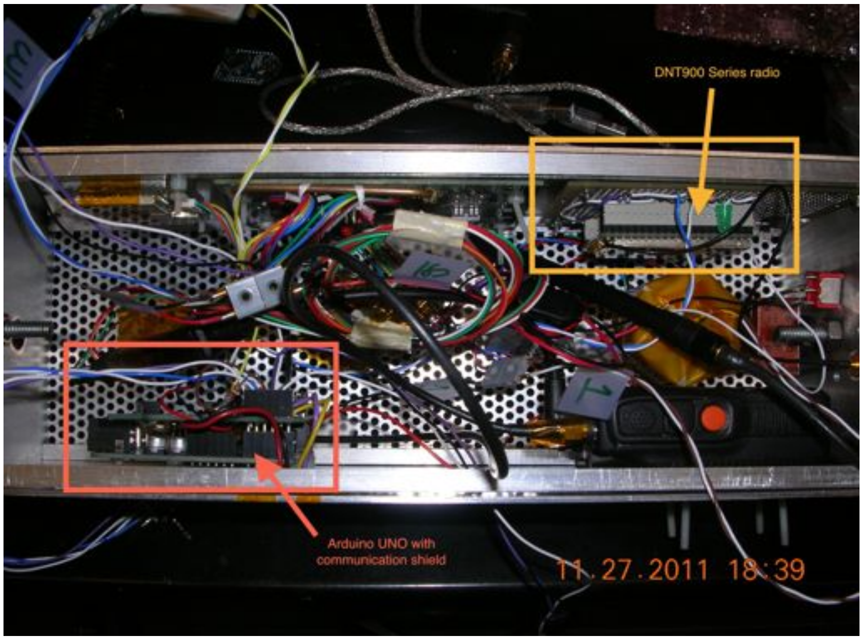


Figure 2: Inside of the Dave payload. Arduino UNO board with communication shield is located at the bottom left. The DNT900 series radio is at the top right.

The remaining sections will discuss in greater detail the Duck payloads, the missions in which they were used, the resulting data, and future directions for the project. Section III will describe the components that make up the Ducks, the pre-launch testing that was conducted, and the Ompompanoosuc and Connecticut River field work. Section IV will present the data and results from the two missions. Finally, Section V will discuss the next steps for investigating payload swarms.

3. Statement of thesis

The GreenCube team has built 20 “Duck” payloads out of low-resource, commercially available technology that can measure position, acceleration, and magnetic field. These are capable of taking measurements to map and analyze the flow fields of the Connecticut and Ompompanoosuc Rivers, the results of which can be used both to inform LWD studies and to explore issues related to the development and use of payload swarms.

III. Methodology

This section first gives a brief overview of a Duck payload, followed by detailed information about the instrumentation and radios and the testing that was performed with them. Next is a description of mission management for the project and the contributions of various team members. Finally, the fieldwork on the Connecticut and Ompompanoosuc Rivers is discussed.

1. Payload Overview

Each Duck payload contains an electronics stack, a 9-volt battery, a weight, and a foam cutout to keep the other parts in place housed within a waterproof Rubbermaid 1.2 Liter food storage container (Figures 3 – 6). The electronics stack consists of an Arduino UNO control board, a compass board for magnetic field and acceleration measurements, a GPS, and a radio for real-time data transmission (Figures 7 – 12). Further information about the individual components of the electronics stack can be found in the sections below. Additionally, each Rubbermaid container has fins attached to three sides for stabilization and to entrain spinning in eddies. The total cost per payload, including all parts and materials as well as student labor involved in the construction, comes to slightly less than $250. See the Appendix for a breakdown of cost by component.

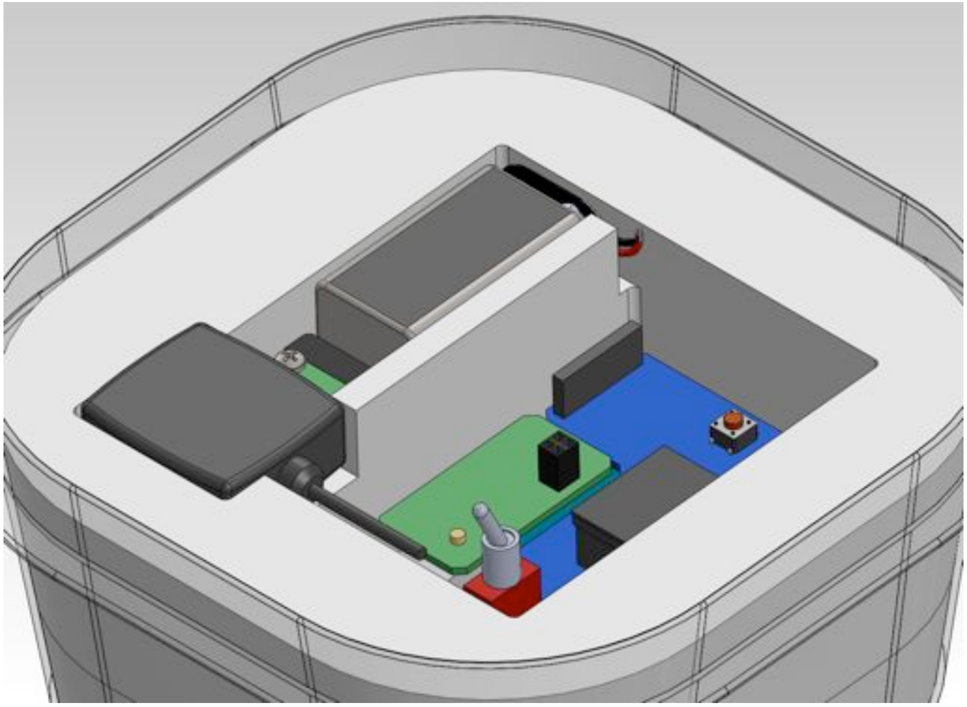


Figure 3: Rubbermaid container with foam cutout housing electronics stack, GPS antenna, battery, and on/off switch. The weight sits below the foam. Drawing by Max Fagin.

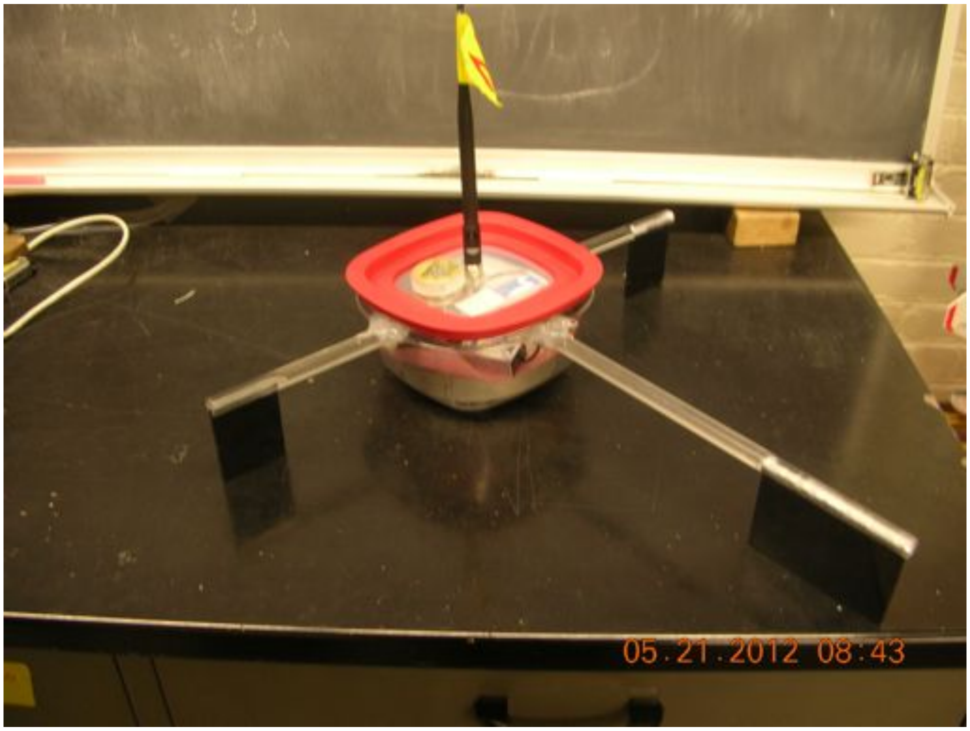


Figure 4: Fully assembled payload.

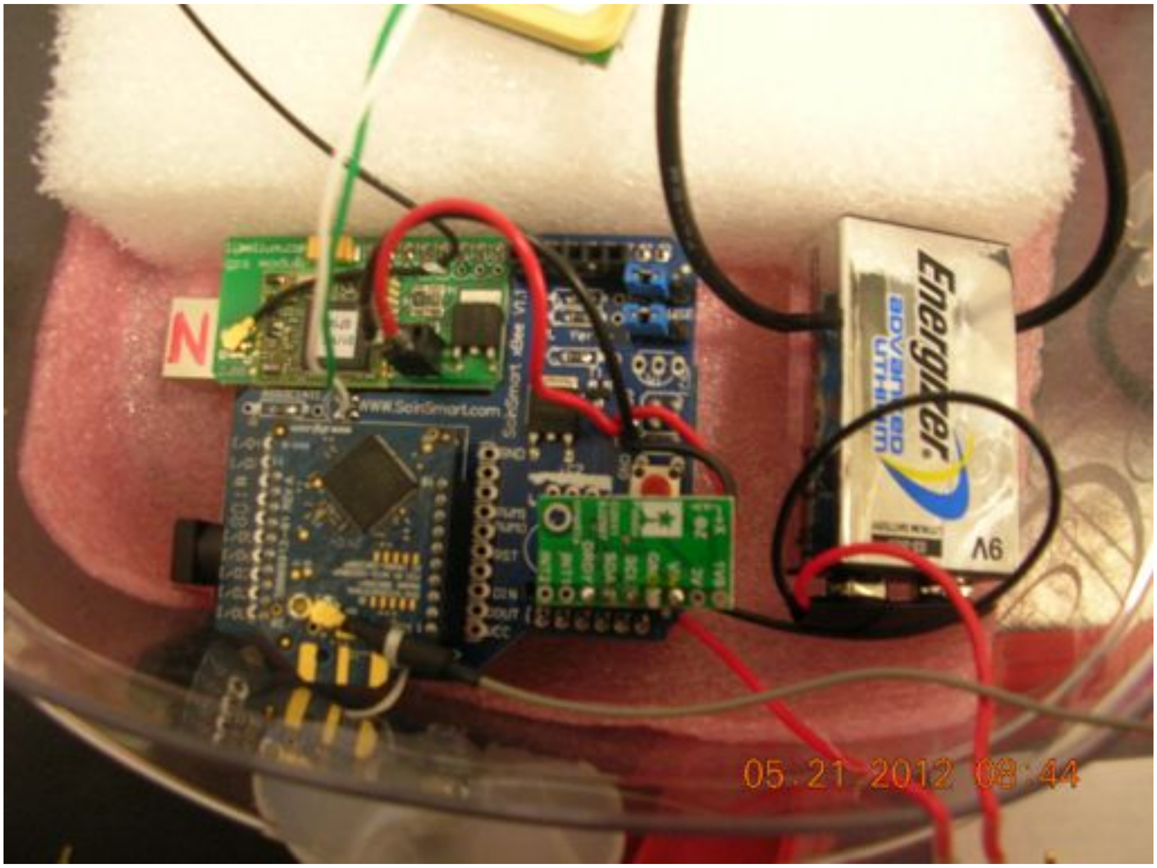


Figure 5: Closeup of Duck electronics stack. Communication shield, Xbee radio, GPS module, compass board, and 9V battery are visible. See sections below for more detailed information about these components.

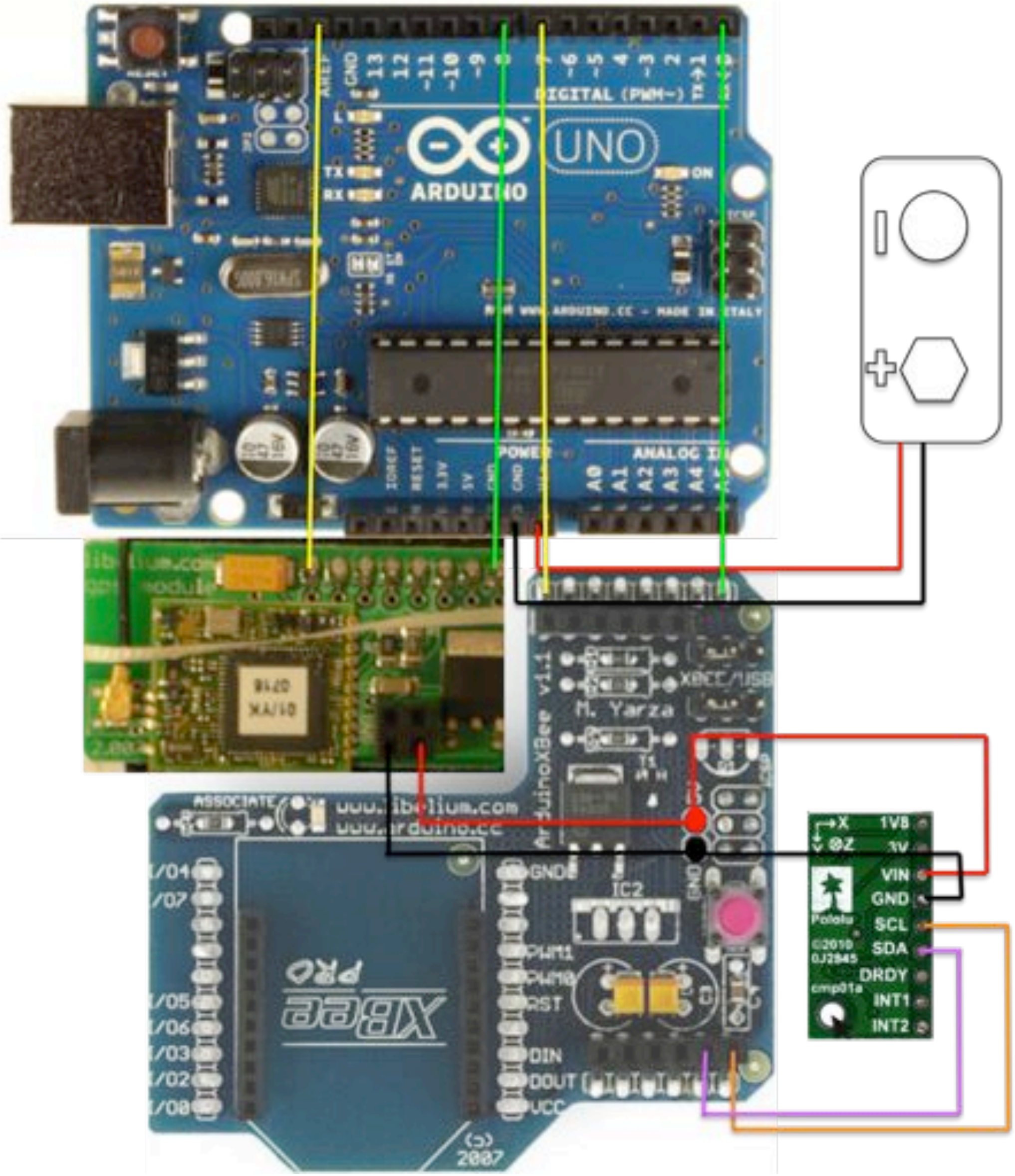


Figure 6: Electronics configuration diagram. Red and black lines depict physical power and ground wires. All other colors are guides showing the proper alignment of pins in headers. Note the 5V and GND through-holes on the communication shield which are used to supply power and ground to both the compass board and the GPS module. Also note that for the compass board, everything except VIN, GND, SCL, and SDA is left unconnected.

2. *Arduino UNO*

The Duck payloads use the open source Arduino UNO as a control board. The Arduino UNO uses an Atmel processor programmed in the Arduino language, which is a wrapper for the C language. It is a great advantage that Arduino is open source; one can learn to use it in a very short amount of time due to the wide variety of excellent tutorials and advice available from the Arduino community. The Arduino UNO contains a microprocessor and several analog and digital inputs for sensors. Many companies make “modules,” such as the GPS module used in the Duck payloads, designed to plug into these inputs. Tutorials and example Arduino code for most of these modules are readily available, particularly through websites such as Adafruit.

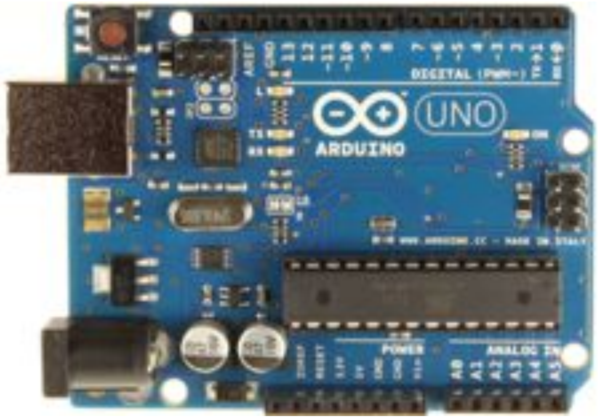


Figure 7: Arduino UNO R3, the control board for the Duck payloads. Image from www.arduino.cc

3. GPS

The GPS module, a Vinotech A1037A from Libelium, is capable of outputting a variety of standard National Marine Electronics Association (NMEA) sentences containing slightly different information. For the Ducks, all sentence types except GGA were suppressed from within the Arduino code in order to make processing the data easier. A GGA sentence looks like this:

$GPGGA,171034.000,4346.480,N,07215.306,W,1,05,2.4,0134.7,M,-32.5,M,, B

After $GPGGA, the first number is a time stamp, the second number is latitude, and the third number is longitude. The other number of interest is the 05, the second number after the longitude. This is the number of satellites that the GPS module is in contact with. The "B" at the end is not a standard part of a GGA sentence. It is a payload identifier (each of the 20 Ducks payloads has a different letter from A to T) that the Arduino is programmed to add to every sentence.

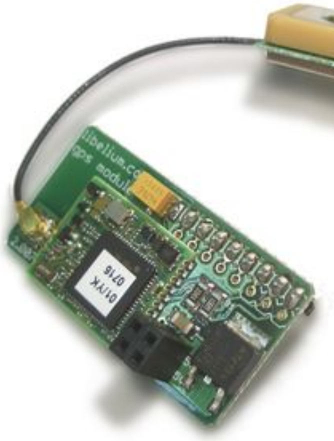


Figure 8: the GPS module used in the Duck payloads. Image from www.cooking-hacks.com

At minimum, the GPS units must be able to resolve points at distances some fraction of the width of the rivers under investigation. The narrower river, the Ompompanoosuc, is approximately 15 - 35 meters wide in the section where the payloads were launched. To test the resolving power of the GPS units, several were carried around on the Dartmouth College Green, as shown in Figure 9. Each color represents a different GPS. A single person holding one payload in each hand at arm's length made the brown and orange tracks. The inaccuracy of the two tracks at the very beginning, particularly where the orange payload appears to start off inside of Rauner Library, is common for the first few data points after a GPS module acquires a position lock.

GPS coordinates reported by these modules are typically accurate to within a few meters. The best results are obtained on clear days when the payload is located away from buildings, trees, and other objects that can interfere with reception. The more satellites a GPS module can communicate with, the better. Under non-ideal conditions, when the GPS module has difficulty establishing contact with enough satellites to acquire a position fix, the team has had some success with removing the antenna from the Rubbermaid container until a position lock is acquired.

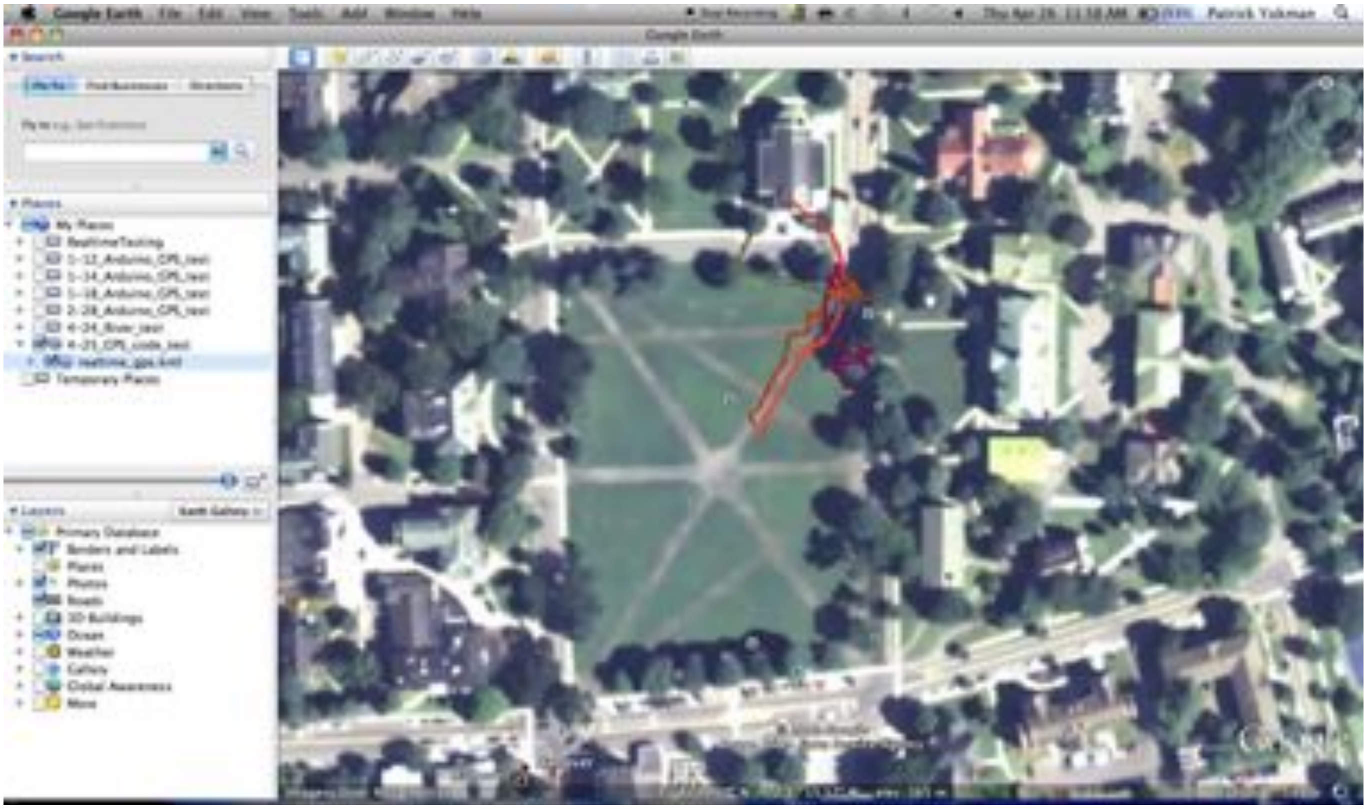


Figure 9: GPS plot from testing on the Green. Shows resolution of GPS units.

4. Compass Board

The Duck compass board is an LSM303DLM from Pololu. In the field, the compass board measures the earth's magnetic field and acceleration due to gravity. Because the direction and magnitude of both these quantities are known, this information can later be used to determine the orientation of the payload. Earth's magnetic field has a magnitude of approximately 0.5 gauss, so the total variation in magnetic field observed in any axis of the magnetometer should be roughly no greater than ±0.5 gauss. The maximum magnetometer gain setting is 1100 Least Significant Bits(LSB)/gauss, which means that it can resolve magnetic fields that differ by 1/1100 of a gauss. This corresponds to an input field range of ±1.3 gauss, and thus is appropriate for the Ducks. Magnetic fields outside of the ±1.3 gauss range have been observed in the lab due to the close proximity of a variety of metal and magnetic objects, but never in the field. NOAA's magnetic field property calculator (http://www.ngdc.noaa.gov/geomagmodels/IGRFGrid.jsp) was used to find the average magnetic field observed in Hanover, NH by a payload sitting with its vertical axis pointing down and the other two axes aligned with north and east. The average is approximately 0.498 gauss in the vertical direction, 0.186 gauss in the northward direction, and -0.049 gauss in the eastward direction.

A typical magnetometer/accelerometer data sentence looks like this:

```
A X: -0001 Y: -0012 Z: -1038 M X: -0017 Y: +0474 Z: +0001 1971 J
```

The first three numbers are the X, Y, and Z components of the acceleration reading. The Z reading of about -1000 indicates an acceleration of 1g in the downward direction, i.e. gravity. The second three numbers are the X, Y, and Z components of the magnetometer.

It is apparent from this example sentence that the vertical axis is Y. The last number is a counter that shows how many of these mag/accelerometer sentences have been sent since the payload began transmitting. Finally, the “J” at the end is a payload identifier like the one seen earlier in the GPS data.

The compass board is mounted on the rest of the electronics stack via a header, which keeps it level and square to the stack. The electronics stack itself is held firmly in place in a uniform orientation in each payload by the foam cutout (Figure 3). Error in magnetic field measurements due to differences in compass board orientation and fabrication remains to be quantified and is still an open project.

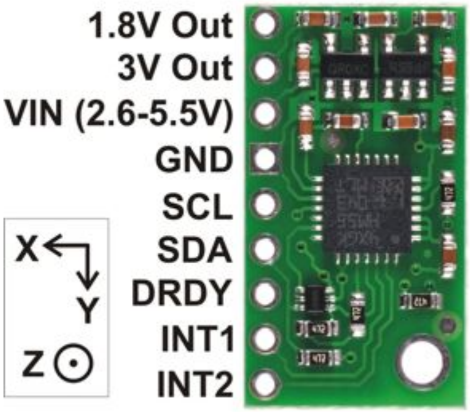


Figure 10: Duck compass board. Image from www.pololu.com

5. *Radios*

Each Duck contains an Xbee Pro 900MHz radio (Figure 11). Considerations for choosing these radios included cost, range, compatibility with Arduino, and networking capabilities. Under ideal conditions the range of an Xbee Pro is approximately one mile, which is far too short for a rocket or orbital mission or even for a balloon borne GreenCube (for which we use the much longer range DNT900 series radios) but perfectly adequate for a river-based one. Xbee radios are commonly used with Arduino, and several inexpensive “communication shield” boards exist which provide an interface between the two. We use the Arduino Xbee Shield developed by Libelium (Figure 12). Furthermore, Xbees are capable of operating in a “transparent mode,” in which they simply transmit all data sent to them from the Arduino with no handshaking required. The ability of Xbees to interface with an Arduino without any handshaking made the process of learning to operate them much quicker and easier than it otherwise would have been.

Figure 13 shows a representative example of data received by a ground station radio. Notice that GPS sentences are interleaved with sensor data sentences and that multiple payloads are sending data at once. Payloads can be distinguished from each other by the identifier letter located at the end of each sentence.

Groups of Xbee radios can be arranged as part of either a star or mesh network. A star network contains one central node to which all others are connected. This kind of network is very efficient because packets (units of data; in our case, a set of either GPS or sensor information – see GPS section for an example of a packet) only ever traverse one node, and ideally packets are never repeated. In a mesh network, each node is connected to all the others. These networks are very reliable and would allow payloads that get far away from the ground station to send their data by going through intermediate nodes. The major disadvantage of a mesh network is that it massively increases the volume of data flowing into the ground station compared to a star network. The Ducks use two star networks because even in the star configuration, the buffer of the receiving node fills up when it is communicating with 20 other nodes at once. Why this happens is unclear and is still under investigation.

The team has worked on adjusting the radios' baud rates to maximize our telemetry budget. The rate of communication between two radios is always 230400 baud, but the rate at which a radio communicates with a host device, such as a ground station computer, is adjustable. The highest standard rate to which the radios can be set is 115200 baud. When the baud rate for communication between the receiving radio and the ground station computer is less than that at which the radios communicate with each other, there is a danger that the receiving radio's buffer will overflow and cause data to be lost. Given the size of our data packets (approximately 100 characters, which is 100 bytes) and our data cadence of one hertz, this should not be a problem with 20 transmitting payloads at 115200 baud. Missing and incomplete data packets were seen during lab testing with 20 transmitting payloads per receiving radio but not with 10. The team is still in the process of testing to determine the exact nature of this limit.

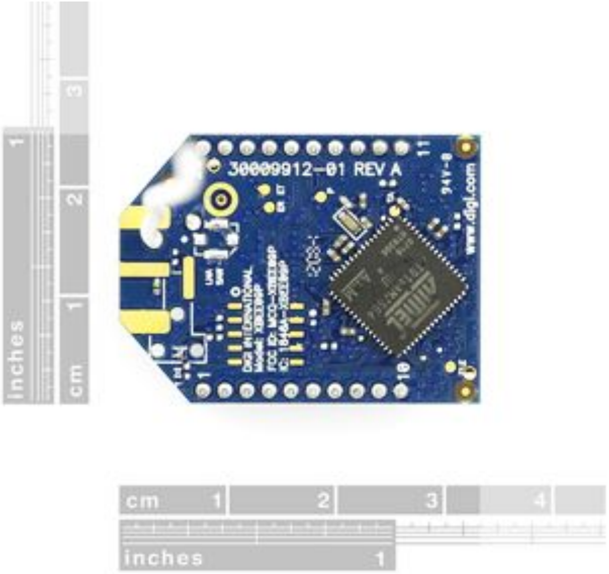


Figure 11: Xbee Pro 900 radio. Image from www.esawdust.com. Note that Duck payloads have an external antenna connected to the Xbee via UFL connector rather than a wire antenna like the one pictured here.

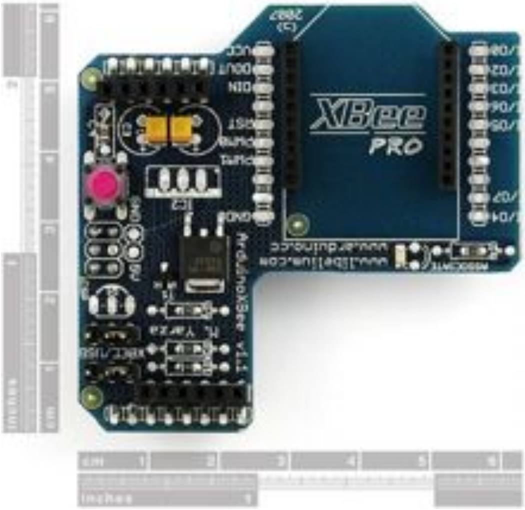
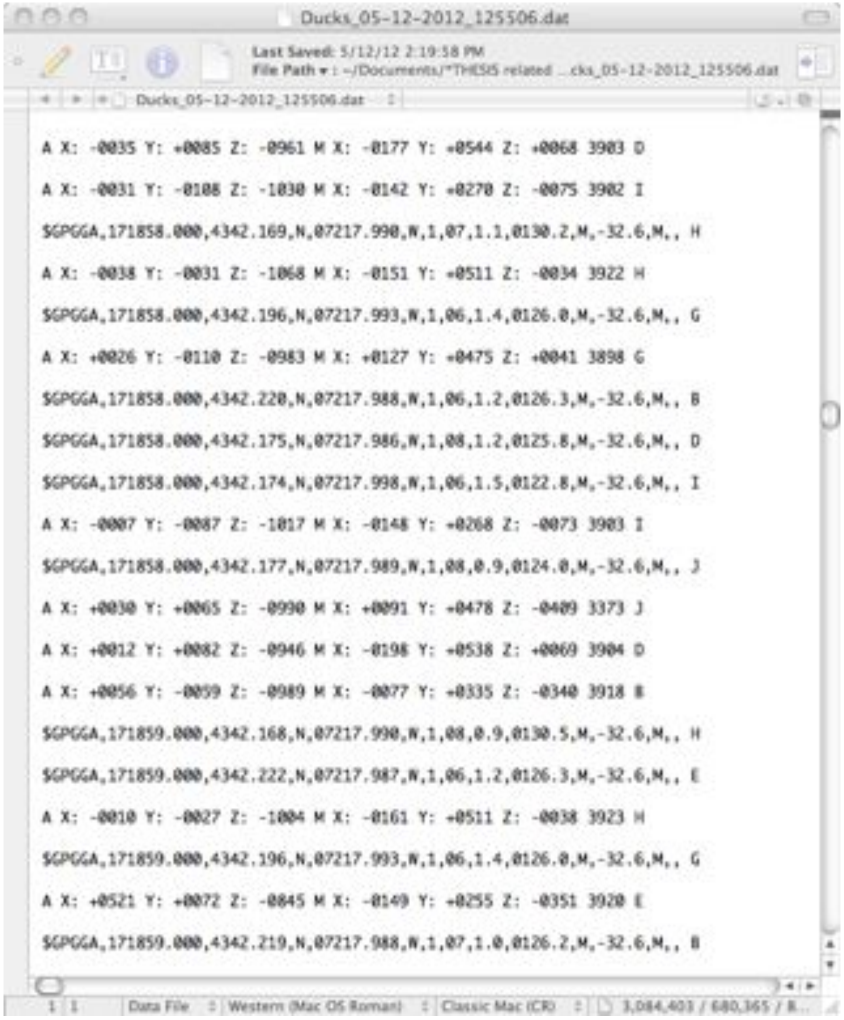


Figure 12: ArduinoXbeeShield. www.robotbox.net


Ducks_05-12-2012_125506.dat
Last Saved: 5/12/12 2:19:58 PM
File Path ▾ : ~/Documents/*THESIS related ...cks_05-12-2012_125506.dat
Ducks_05-12-2012_125506.dat
1 1
Data File
Western (Mac OS Roman)
Classic Mac (CR)
3,084,403 / 680,365 / 8...

```
A X: -0035 Y: +0085 Z: -0961 M X: -0177 Y: +0544 Z: +0068 3903 D

A X: -0031 Y: -0108 Z: -1030 M X: -0142 Y: +0270 Z: -0075 3902 I

$GPGGA,171858.000,4342.169,N,07217.990,W,1,07,1.1,0130.2,M,-32.6,M,, H

A X: -0038 Y: -0031 Z: -1068 M X: -0151 Y: +0511 Z: -0034 3922 H

$GPGGA,171858.000,4342.196,N,07217.993,W,1,06,1.4,0126.0,M,-32.6,M,, G

A X: +0026 Y: -0110 Z: -0983 M X: +0127 Y: +0475 Z: +0041 3898 G

$GPGGA,171858.000,4342.220,N,07217.988,W,1,06,1.2,0126.3,M,-32.6,M,, B

$GPGGA,171858.000,4342.175,N,07217.986,W,1,08,1.2,0125.8,M,-32.6,M,, D

$GPGGA,171858.000,4342.174,N,07217.998,W,1,06,1.5,0122.8,M,-32.6,M,, I

A X: -0007 Y: -0087 Z: -1017 M X: -0148 Y: +0268 Z: -0073 3903 I

$GPGGA,171858.000,4342.177,N,07217.989,W,1,08,0.9,0124.0,M,-32.6,M,, J

A X: +0030 Y: +0065 Z: -0990 M X: +0091 Y: +0478 Z: -0409 3373 J

A X: +0012 Y: +0082 Z: -0946 M X: -0198 Y: +0538 Z: +0069 3904 D

A X: +0056 Y: -0059 Z: -0989 M X: -0077 Y: +0335 Z: -0340 3918 B

$GPGGA,171859.000,4342.168,N,07217.990,W,1,08,0.9,0130.5,M,-32.6,M,, H

$GPGGA,171859.000,4342.222,N,07217.987,W,1,06,1.2,0126.3,M,-32.6,M,, E

A X: -0010 Y: -0027 Z: -1004 M X: -0161 Y: +0511 Z: -0038 3923 H

$GPGGA,171859.000,4342.196,N,07217.993,W,1,06,1.4,0126.0,M,-32.6,M,, G

A X: +0521 Y: +0072 Z: -0845 M X: -0149 Y: +0255 Z: -0351 3920 E

$GPGGA,171859.000,4342.219,N,07217.988,W,1,07,1.0,0126.2,M,-32.6,M,, B
```

Figure 13: Data received by one of the ground station radios during a launch

6. Payload Summary

In summary, the GPS and compass board collect data and send it to the Arduino data input pins. Arduino code collects these data in a buffer, adds the payload identifier, and streams the data out into the Xbee radio. The radio transmits (along with all the other ducks) to the ground station (essentially a listening duck), which takes the serial data stream and writes it to a file in the form of ASCII characters, shown in Figure 13. Once the components for the Duck payloads were well-defined, the next step was to build and test one complete example.

7. High-Level Testing

Once one duck was fully built, it was taken to the Connecticut River and allowed to float in the water while attached to a tether for testing. The interior of the payload remained dry, even when the payload was flipped upside down. This made the team confident about the waterproofness of the duck design. Unfortunately some payload interiors did get wet during the missions; see the **Field Work** section (9) below for more information. The upright duck floated with the water level just below its lid, confirming that the weight's mass was appropriate. The GPS module was able to communicate with multiple satellites while enclosed in the payload, even with some water sitting on top of the payload. Finally, the ground station continued to receive data from the duck even when it was upside down with its radio antenna under water, an encouraging indicator of the ducks' ability to operate in non-ideal conditions.

When the team was confident in the design of the first duck, they set about building 19 more. A vehicle log was created to track the construction process. The group divided up the building work with different team members handling mechanical fabrication, mechanical assembly, foam cutout manufacture, and electronics assembly. See the next section for more on the division of labor among different team members. When all ducks were complete, a brief series of tests was conducted on each to confirm correct operation. Each duck was taken outside and allowed to run until it acquired a GPS position fix. Each duck was rotated through 180° in each axis to confirm magnetometer and accelerometer functioning. These tests were conducted with the payloads transmitting via radio to the ground station rather than with them connected directly through USB, and thus the radios were also tested. The next section describes the importance of teamwork not only for fabrication and testing, but for the success of the project as a whole.

8. Mission Management and Student Team

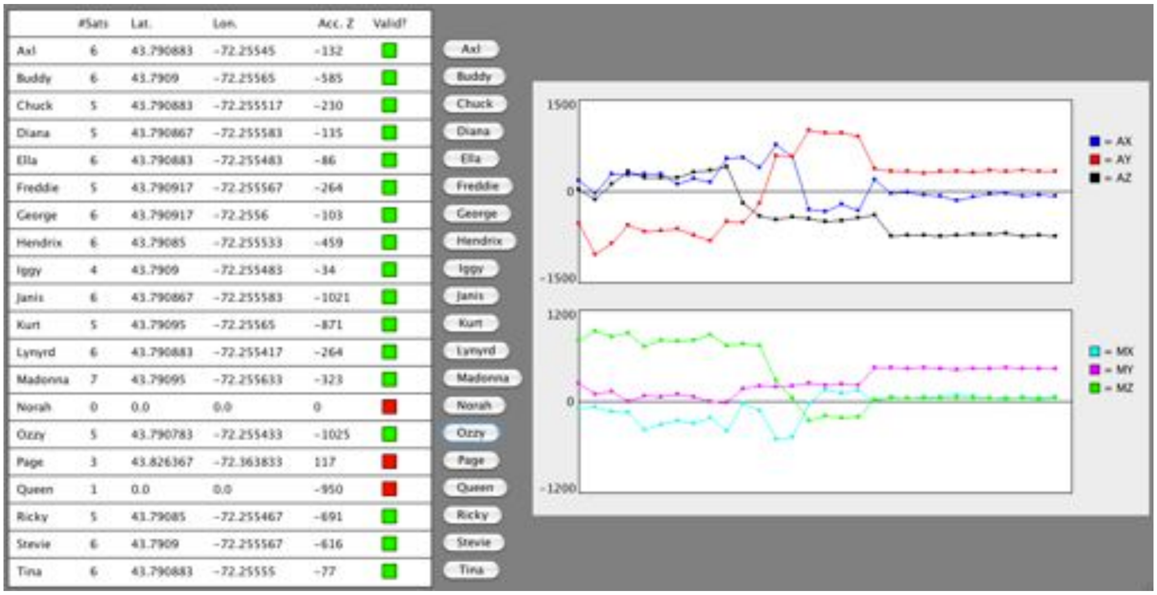
The GreenCube team has students at a variety of experience levels all of whom were able to contribute to the DUCKS project in different ways. The senior honors student (and author of this thesis) managed the team by coordinating the efforts of different students, creating launch procedures, and organizing the two launches, in addition to working on various aspects of the electronics construction and testing.

As part of a Presidential Scholars project, the junior-level student in the lab, Jon Guinther, has been studying computational aspects of vector field mapping from sparse measurements. He has been investigating problems that can arise during the process of sampling an underlying vector field with sparse measurements, plotting the data onto a regular gridded array, and reconstructing the field by interpolating from the gridded data.

An example of the kind of aliasing that can occur appears in magnetic field data from the Iridium satellite network, analyzed by the Johns Hopkins AMPERE group (http://ampere.jhuapl.edu). The group interpolated a map of magnetic field with the International Geomagnetic Reference Field (IGRF) removed based on measurements from the Iridium satellites. They then took the curl of the resulting vector field. The result was alternating rings of positive and negative curl around the pole, though there was no indication of such ringing in the raw data. In order to investigate these issues, Jon created increasingly complex vector fields, which he then "sampled" by making cuts through the fields and recording the values at intervals along the cuts. He then tried various methods for reconstructing the original vector field in order to extract information such as divergence and curl. His work interpolating vector field maps from dummy data served as the basis for much of the data analysis for the DUCKS missions. Jon will continue to expand on this work next year in the form of a senior honors thesis.

Several younger students have also adopted projects of their own within the DUCKS mission. Sophomore Tom Whalen has taken on the task of processing and plotting the magnetometer and accelerometer data, as well as converting the GPS data to velocities for vector field analysis.

Patrick Yukman wrote most of the Arduino code as well as developed a data display program that runs in real-time to make it easier for the ground station operator to see and understand the state of the Ducks during a launch. For each duck, the number of GPS satellites, latitude, longitude, and z component of the acceleration reading (which indicates whether the duck has tipped over) is displayed. There is also a “valid” box which appears green if a payload has several satellites, a GPS position lock, a negative z-acceleration value (indicating roughly that the payload is tipped less than 90 degrees from upright), and has sent data within the last 10 seconds. The user can click on the button next to a duck to display plots of its recent 3-axis magnetometer and accelerometer data. Patrick also worked with the author in most of the radio and GPS testing. Next year Patrick will complete a Presidential Scholar project investigating communications within payload swarms.



	#Sats	Lat.	Lon.	Acc. Z	Valid?
Axl	6	43.790883	-72.25545	-132	🟩
Buddy	6	43.7909	-72.25565	-585	🟩
Chuck	5	43.790883	-72.255517	-230	🟩
Diana	5	43.790867	-72.255583	-135	🟩
Ella	6	43.790883	-72.255483	-86	🟩
Freddie	5	43.790917	-72.255567	-264	🟩
George	6	43.790917	-72.2556	-103	🟩
Hendrix	6	43.79085	-72.255533	-459	🟩
Iggy	4	43.7909	-72.255483	-34	🟩
Janis	6	43.790867	-72.255583	-1021	🟩
Kurt	5	43.79095	-72.25565	-871	🟩
Lynyrd	6	43.790883	-72.255417	-264	🟩
Madonna	7	43.79095	-72.255633	-323	🟩
Norah	0	0.0	0.0	0	🟥
Ozzy	5	43.790783	-72.255433	-1025	🟩
Page	3	43.826367	-72.363833	117	🟥
Queen	1	0.0	0.0	-950	🟥
Ricky	5	43.79085	-72.255467	-691	🟩
Stevie	6	43.7909	-72.255567	-616	🟩
Tina	6	43.790883	-72.25555	-77	🟩

Figure 14: Screenshot of Darkwing DUCKS, real-time data display software created by Patrick Yukman, showing data from May 19 launch

Peter Horak will also conduct a Presidential Scholar project next year, about either solar panel design and use or the orbital mechanics of payload swarms (building on previous senior thesis work by Sean Currey), which he would investigate using Satellite Tool Kit (STK).

Even the first year members of the group have been vital to the success of the project. Women In Science Project (WISP) interns Ellen Weburg and Ha Nguyen tested various duck fin designs, and once one had been chosen they fabricated the fins for all of the ducks in the Sciences Division machine shop. Freshmen Nina Maksimova and Dylan Sewell completed the mechanical assembly. See the Appendix for a complete list of the team and their roles.

9. Field Work

The duck swarm was launched on two different rivers in order to capture different flow patterns. The first launch occurred on May 12, 2012 on the Connecticut River, between the Ledyard Boathouse and Gilman Island (see Figure 15). Since the Connecticut is relatively calm and slow moving, this first launch afforded the team an opportunity to practice deploying, tracking, and recovering the ducks with little risk of losing them. A total of 17 ducks were deployed and the mission lasted for 1.5 hours. The second launch occurred on May 19, 2012 on the Ompompanoosuc River between the Union Village Dam and the Connecticut River (Figure 16). For the second mission, 17 ducks were deployed into the water and one was kept with the ground station to record its GPS coordinates. A list of launch procedures can be found in the Appendix.

Many ducks leaked during the first mission. Afterwards the interior of each duck was classified as dry (no pooled water and no obvious precipitation), mostly dry (some precipitation but very little water collected at the bottom), or wet (a significant pool of water in the bottom). Four ducks were dry, six were mostly dry, and seven were wet. All five of the ducks that permanently stopped transmitting partway through the Connecticut mission were classified as wet. Several possible causes of leaks were identified: some Rubbermaid containers had small nicks in the tops from the fabrication process which may have allowed water in, glue used to anchor the fins may have interfered with the seal of some of the lids, poorly packed foam or wires inside the ducks could have prevented the lids from being properly seated, or the radio antenna seals may not have been completely watertight. After the launch, when all payloads had dried out, it was confirmed that all those that had gotten wet still worked. Before the second launch problem glue spots were fixed and special care was taken with packing.

The author noticed that in most wet payloads with foam cutouts (cutout shown in Figure 3, part 1 of this section), most or all of the water was diverted around the edges of the foam and pooled in the bottom of the payload without the electronics stack becoming noticeably wet. Nonetheless, wet payloads with foam cutouts were just as likely to have stopped working as wet payloads with makeshift packaging.

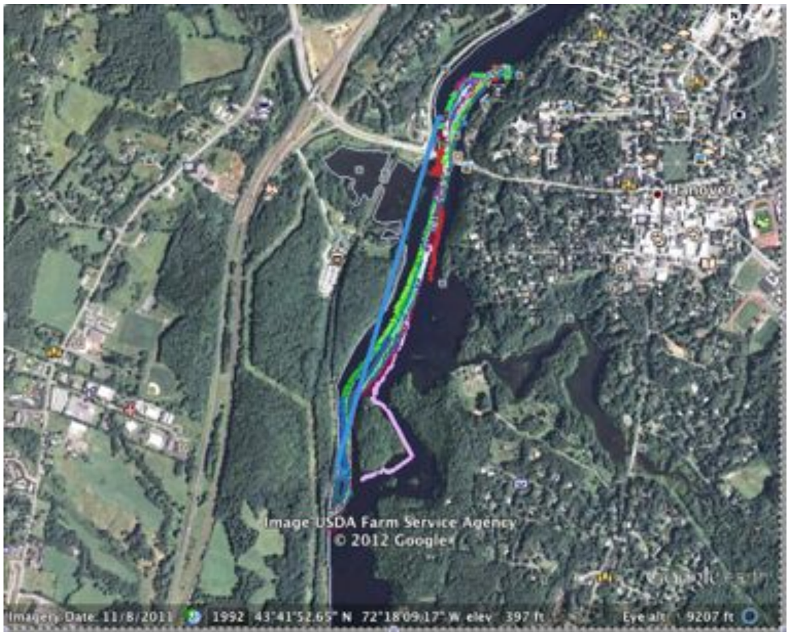


Figure 15: GPS tracks from payloads launched on 5/12/12. Ledyard Bridge crosses the river near the top of the image and the Dartmouth College Green can be seen to the right. Note that Google Earth creates these tracks by drawing a line from each data point to the previous one. Therefore, long straight lines such as the blue one on the left indicate missing data.

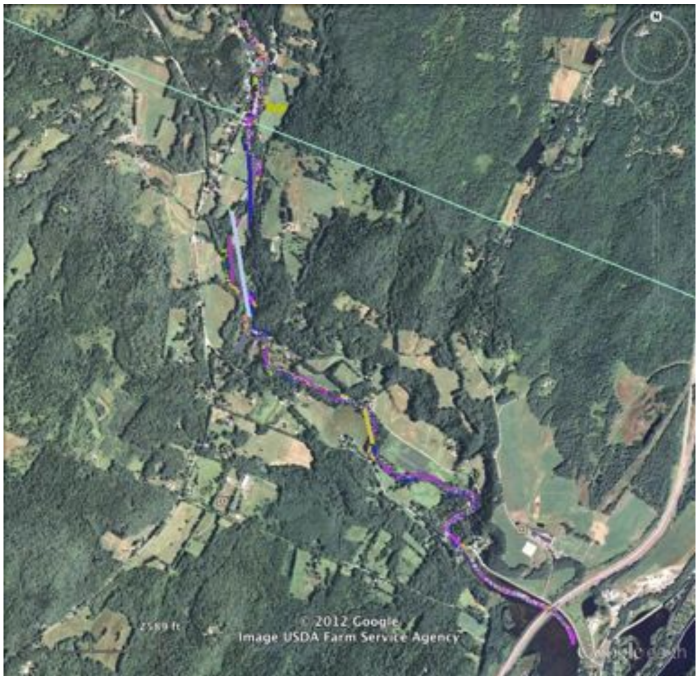


Figure 16: GPS tracks from payloads launched on 5/19/12. Tracks with large amounts of missing data have been removed. The green line running diagonally across the top of the image represents the border between Orange and Windsor counties in Vermont.

IV. Data

Launches were conducted on two different rivers to provide more opportunity for the payloads to experience a variety of flow fields. Ease of mission was prioritized for the first launch, so a relatively calm, deep, slow river was chosen. For the second launch, interesting data became a higher priority. A river with variable flow, many eddies, and narrow enough banks to allow for good payload coverage was selected.

The data presented below represent a group effort, particularly including Jon Guinther, Tom Whalen, Patrick Yukman, and the author. In particular, Jon Guinther's Presidential Scholar work is the underpinning for the vector field plots.

Figure 17 below shows the amount of dropped data typical in a launch. Notice short dropouts for individual payloads are common, and all payloads experience them at least occasionally. Long dropouts (10 minutes or more), however, are rare.

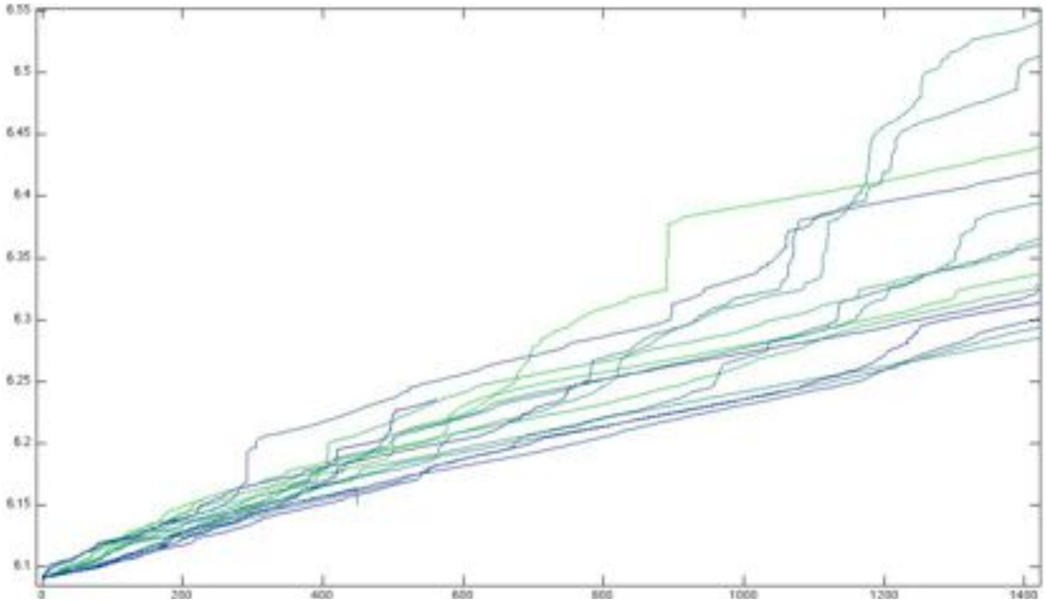


Figure 17: Index of GPS data sentence vs time for all payloads on the Ompompanoosuc. Any vertical jumps indicate dropped data.

Magnetometer and accelerometer data were evaluated to determine the payloads' headings and tilt angles. As shown in Figure 18, the baseline tilt for the payloads is approximately 10°. This happens because each payload has a long fin on one side that is not balanced by a fin on the opposite side. The payloads tilted many times by 180° or more during the Ompompanoosuc mission.

Figure 19 shows the headings of payloads A – J during the Ompompanoosuc launch. The wide variation among payloads and even for a single payload over time suggests that they have no preferred orientation with respect to the direction of flow. This is unsurprising considering that the two short fins together are approximately the same length as the long fin – the horizontal axes of the payload are the same.

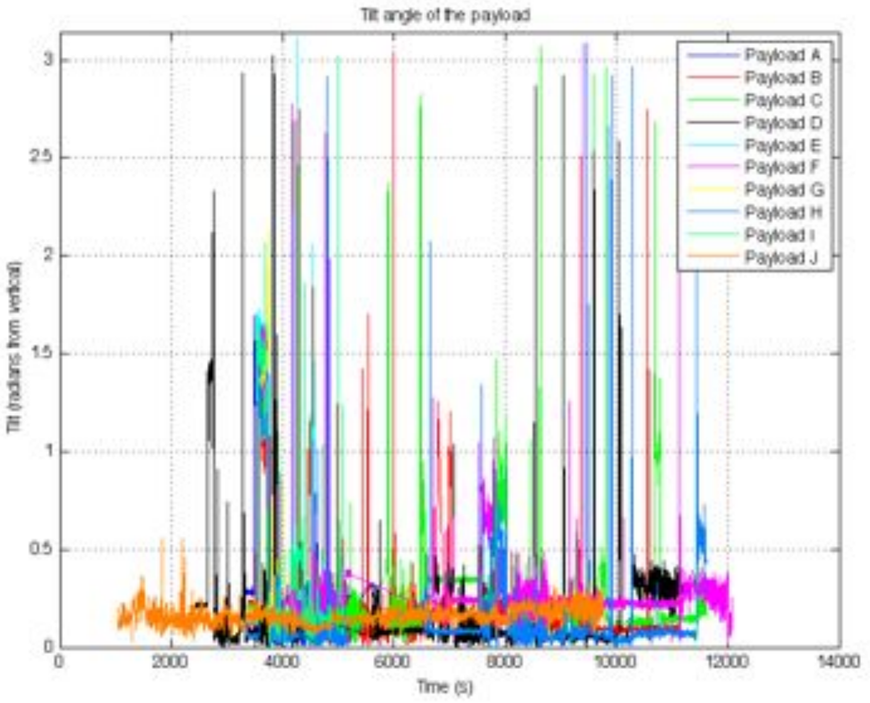


Figure 18: Tilt angles of payloads A – J during Ompompanoosuc mission. Plot by Tom Whalen.

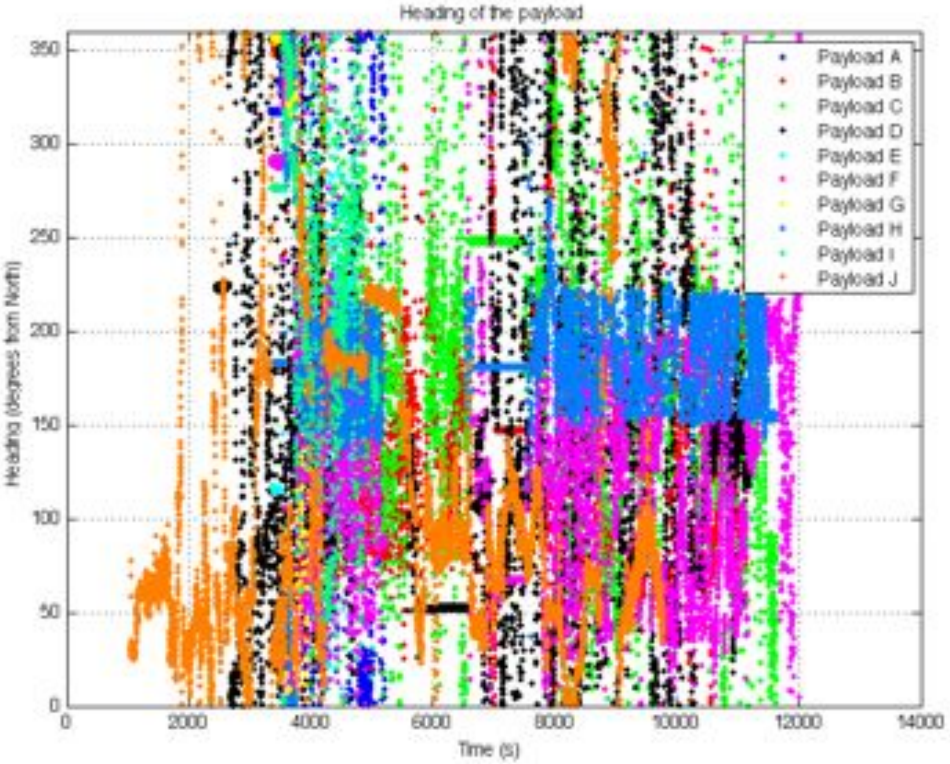


Figure 19: Heading of payloads in degrees from North. Plot by Tom Whalen.

As shown in Figures 20 and 21, jitter in GPS measurements was more pronounced in the Ompompanoosuc launch than in the Connecticut launch. This may be due to trees, houses, and other objects on the banks of the Ompompanoosuc interfering with GPS reception. The average number of GPS satellites "seen" by operational payloads on each mission was calculated: 6.5 for the Connecticut and 5.5 for the Ompompanoosuc.

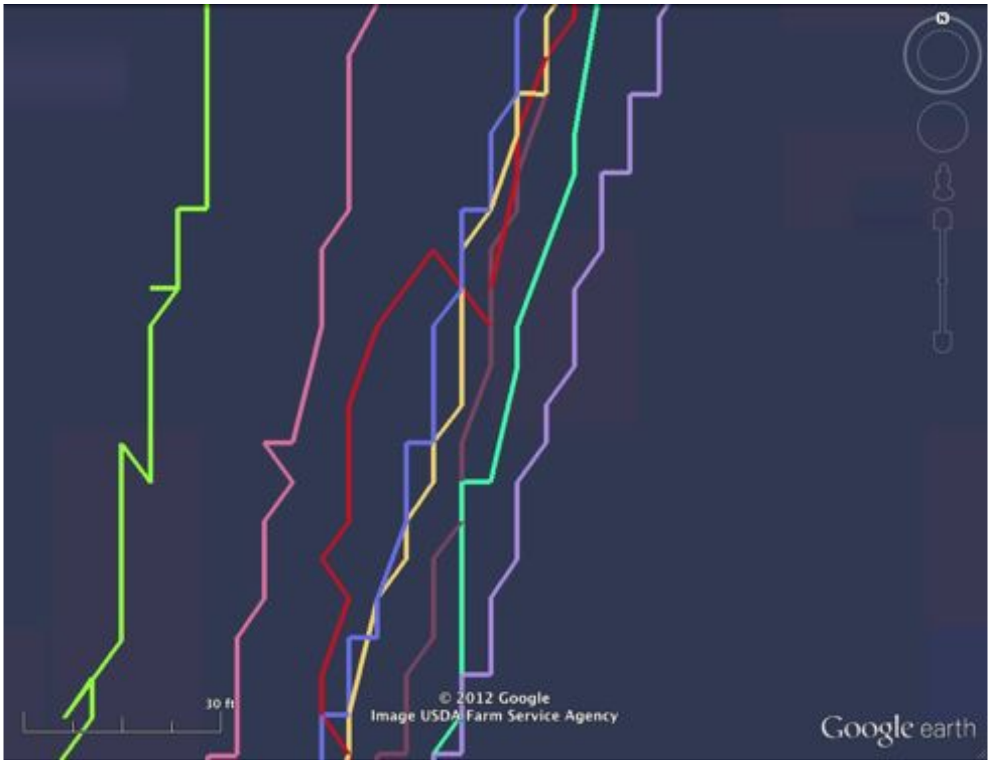


Figure 20: GPS jitter during Connecticut launch

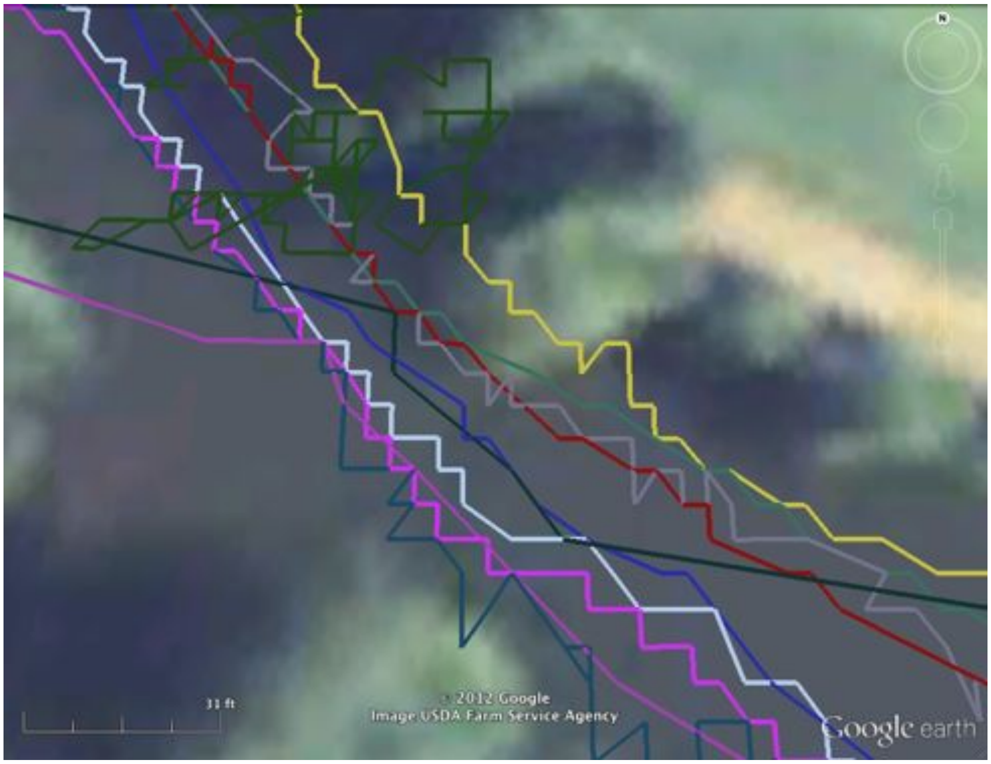


Figure 21: GPS jitter during Ompompanoosuc launch

A reasonably large data set was collected pre-launch as the payloads sat on the bank of the Ompompanoosuc while the team waited for the entire swarm to acquire a position fix. These data were used to quantify GPS jitter for two reasons: the measurements were taken on the same day and at the same location as the second set of launch data, and the payloads remained still over the course of data collection, so there is no true movement to be distinguished from jitter. Because position accuracy improves considerably after the first few minutes of data (see Methodology, section 3), the first 200 data points (about three and a half minutes) from each payload were discarded in the jitter calculation. The average position of each payload was calculated separately. Figure 22 shows the deviation from that average for payload "Axl" over a time period of 30 minutes. The average jitter was calculated for each payload using the expression $\langle\,[measurement - \langle\, position\,\rangle\,]\,\rangle$ and then the overall average was calculated from those values. The overall average jitter was 1.77 x10^{-5} degrees in the latitudinal direction and 2.65 x10^{-5} degrees in the longitudinal direction, which corresponds to 1.97 and 2.13 meters at the latitude of the launch site (43.8° N).

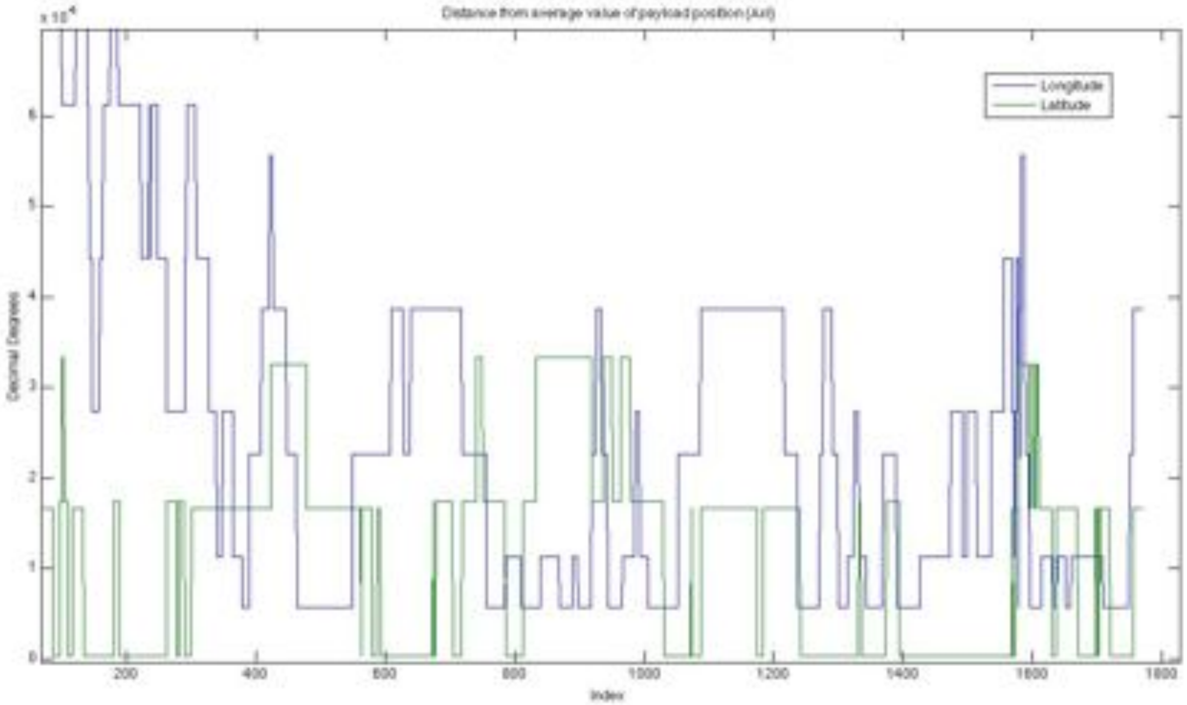
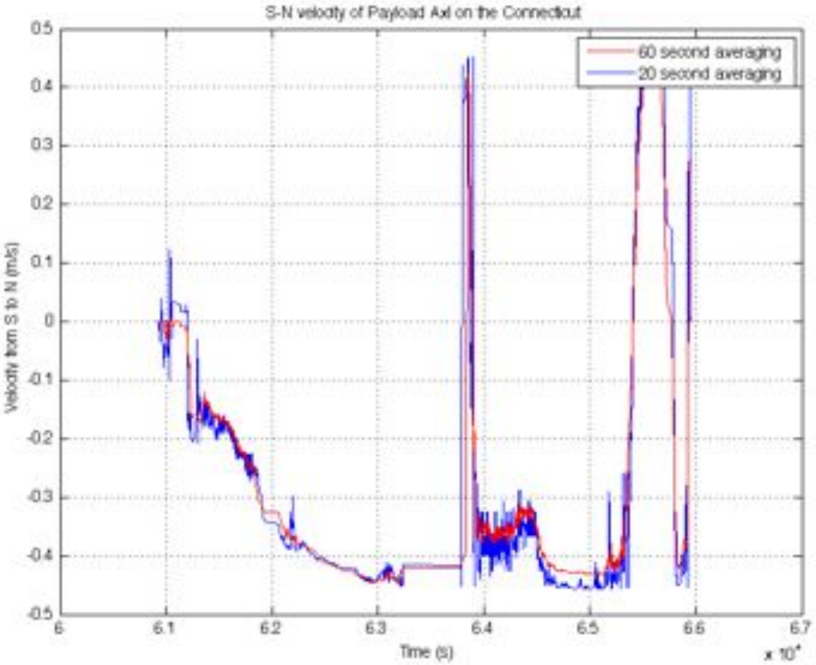
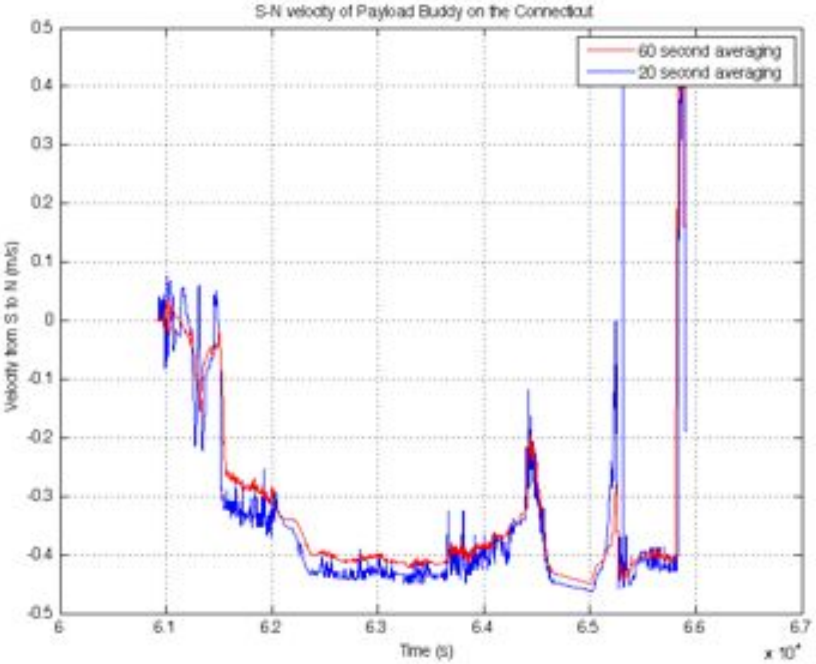


Figure 22: For each data point, the distance in decimal degrees between the recorded and average values for latitude and longitude. Y-axis units are decimal degrees x10^{-5}.

Position data for both missions were converted to velocities and then mapped. The amount by which the ducks move downstream in one second is comparable to amount of jitter in GPS measurements, so the velocities must be calculated using time periods longer than adjacent data points. Velocity for a point was calculated using the positions 30 seconds before and 30 seconds after it. Several other lengths of time averaging were considered, but a window of 60 seconds worked well. As Figures 23 and 24 show, velocities did not change dramatically when averaged over intervals of 60 seconds compared to intervals of 20 seconds, but the data were somewhat smoother with the longer time average.





Figures 23 and 24: South-North velocity over time of two payloads (“Axl” and “Buddy”) during the Connecticut mission. The payloads reverse direction at the end because they were collected and put into canoes, which then began paddling back upstream. In figure 23, the earlier reversal of direction occurs when the payload was picked up, moved, and then put back in the water. Plots by Tom Whalen.

Once a set of payload velocities had been created, they were plotted onto a 2-D grid of the river, the boundaries of which were set using data from the US Census Bureau (http://www.census.gov/geo/www/tiger/tgrshp2010/tgrshp2010.html), with the missing points interpolated. In some spots along the Ompompanoosuc River, the Census Bureau-defined river boundaries did not match the data. In the initial data analysis, data from these spots was thrown out. However, the data appears continuous and valid other than being outside the given river boundary, so in the future it would make sense to widen the river boundaries in trouble spots in order to encompass all the data. A condition of zero velocity along the boundaries was used in determining the interpolation. The velocity maps were created using the "TriScatteredInterp" function in Matlab, which by default performs a linear interpolation on a data set (the value of each point on the grid is determined by taking a weighted average of the two nearest points on either side). The north-south velocity was interpolated separately from the east-west velocity, and then the two were combined to determine the total magnitude of velocity.

Figure 25 shows velocity data for a section of the Connecticut before and after interpolation. The "before" velocities show little spatial variation in magnitude, which is reflected in the lack of prominent features in the interpolated map. Figure 26 shows a similar set of plots for the Ompompanoosuc River. The velocity for the inside of the first curve is not well-determined, since a relatively large space exists there from which no data were collected. The interpolated maps appear to provide a reasonably accurate representation of the river velocities as measured by the payloads.

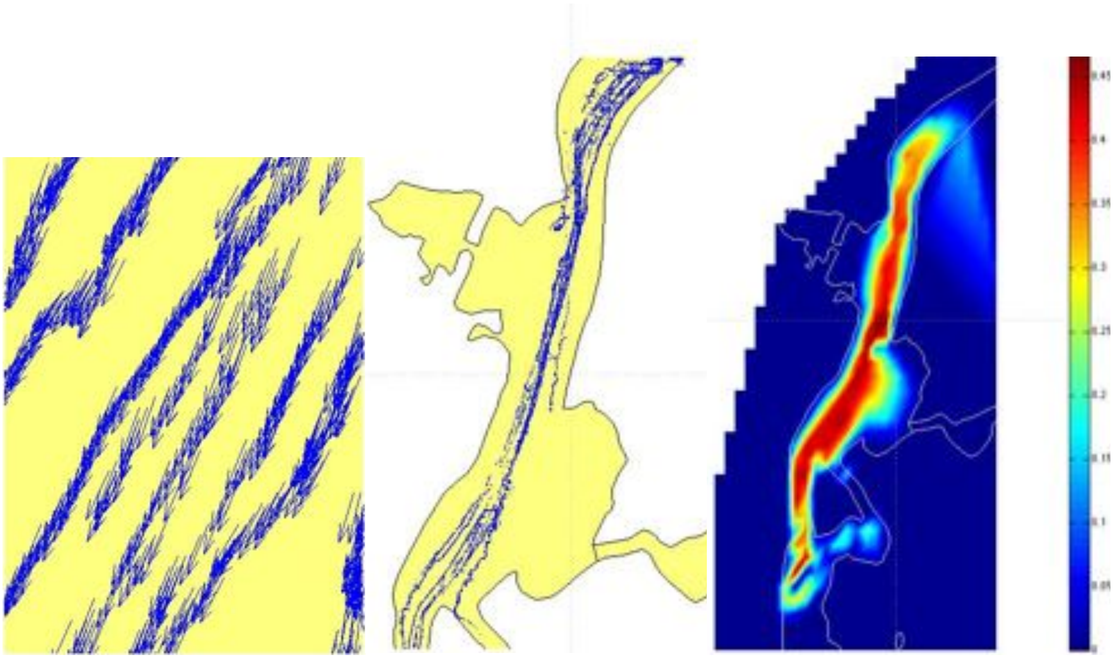


Figure 25: Left: a closeup plot of payload velocities on the Connecticut River, before interpolation. Center: a wider view of the same. Right: a map showing magnitude of velocity in m/s over the same stretch of river, interpolated from data shown on the left.

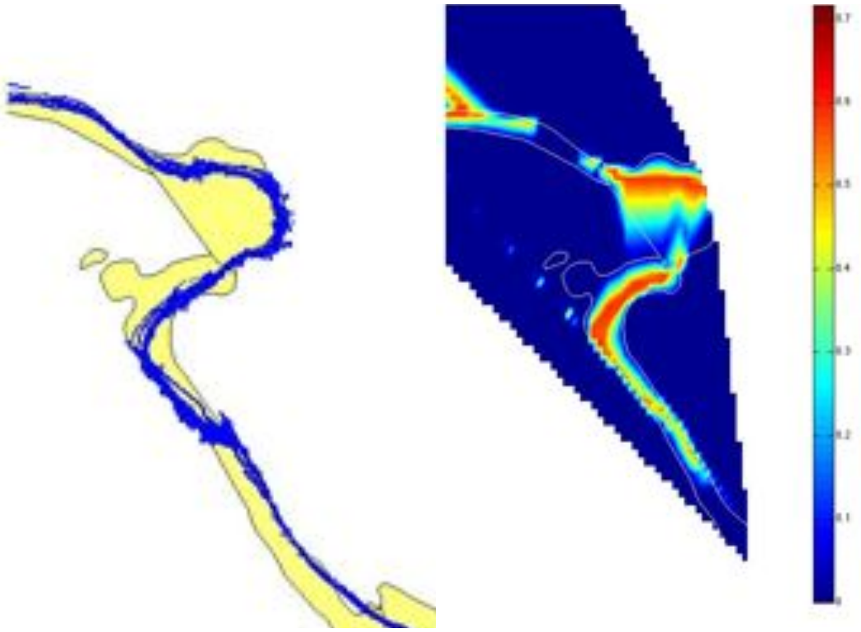


Figure 26: As above, a vector plot of velocities and an interpolated velocity map with units of m/s. Note that the payloads often move slightly outside the boundaries we have defined for the river. Data points outside the boundaries were not included in the creation of the velocity map, which explains the stretch of zero velocity near the top and the apparent slowdown in the middle of the first curve.

In order to assess the relative accuracy of the interpolated maps at various points on the rivers, the number of data points falling within each cell of the grid was counted (Figure 27). In areas of the map where very few data points were collected (such as the channel to the right of Gilman Island in the Connecticut), interpolated velocity values should not be trusted.

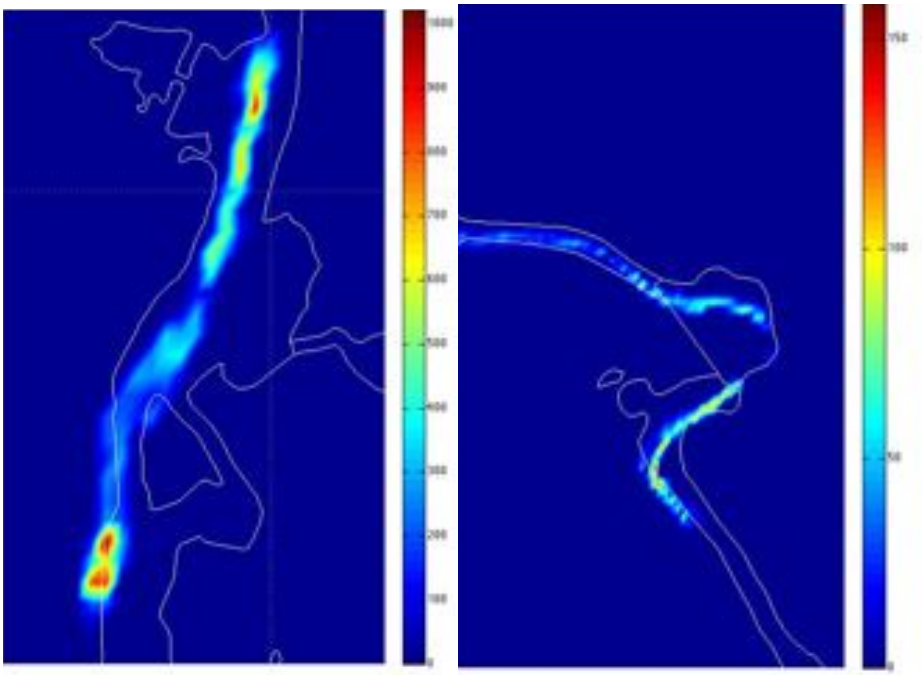


Figure 27: Number of data points falling within each cell of the grid onto which the velocity maps were interpolated. For the Connecticut (left), one cell is .0005 decimal degrees on a side. For the Ompompanoosuc, a cell is .0001 decimal degrees on a side.

On the Ompompanoosuc River, the flow was observed to follow a roughly sinusoidal pattern within the river channel in several places, as shown in Figure 28. This observation fits with the known tendency of river water to meander in sine-derived curves (*Leopold, A View of the River 65*).

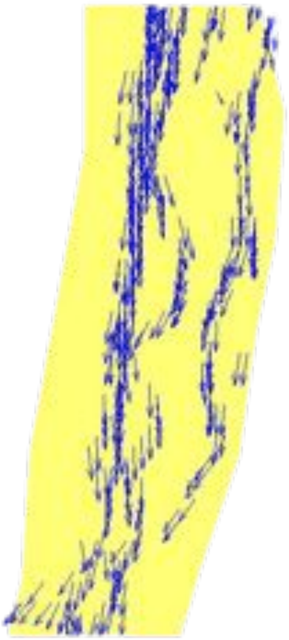


Figure 28: Sinusoidal variation in flow observed on the Ompompanoosuc

Next the divergence and curl of the interpolated velocity functions were taken. A river is an incompressible fluid, so it should have no divergence unless there is an upwelling. Curl should be nonzero in places with eddies or places where the river curves. Predictably, the divergence and curl plots for the Connecticut River were very uninteresting. As shown in Figure 29, neither plot deviates much from zero.

Divergence and curl were calculated for the Ompompanoosuc River velocity function as well, and were found to be an order of magnitude larger than those for the Connecticut. The divergence should be zero, as it was for the Connecticut. However, artificial divergences may appear in areas where data that fell outside the river boundaries were discarded. There should be nonzero curls in sections of the river with pronounced curves. Figure 30 shows the divergence and curl in one such section. Note the area of positive curl that matches the tracks of the payloads as they moved around the bottom curve.

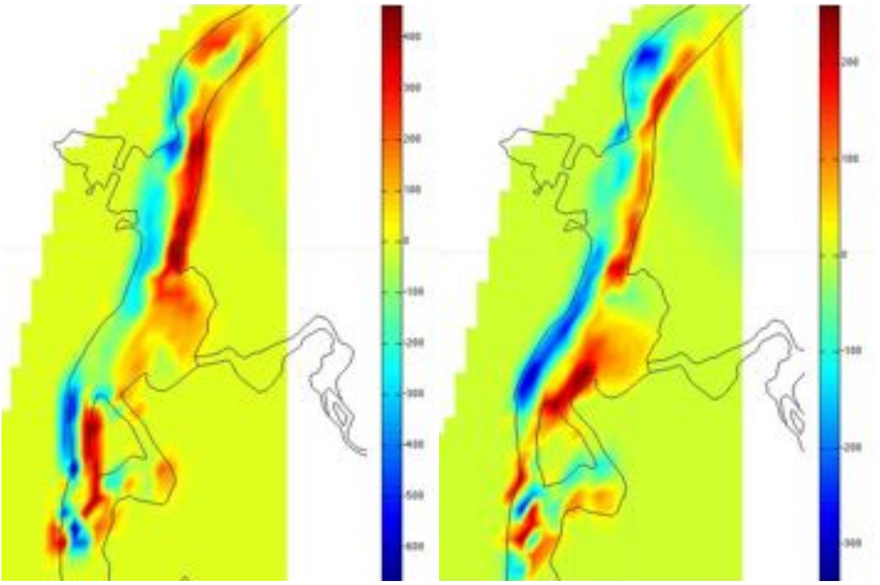


Figure 29: The divergence [Hz] (left) and curl [Hz] (right) of the velocity function for the Connecticut River. Overall, deviations from zero are small in both plots. See Figure 30 for comparison.

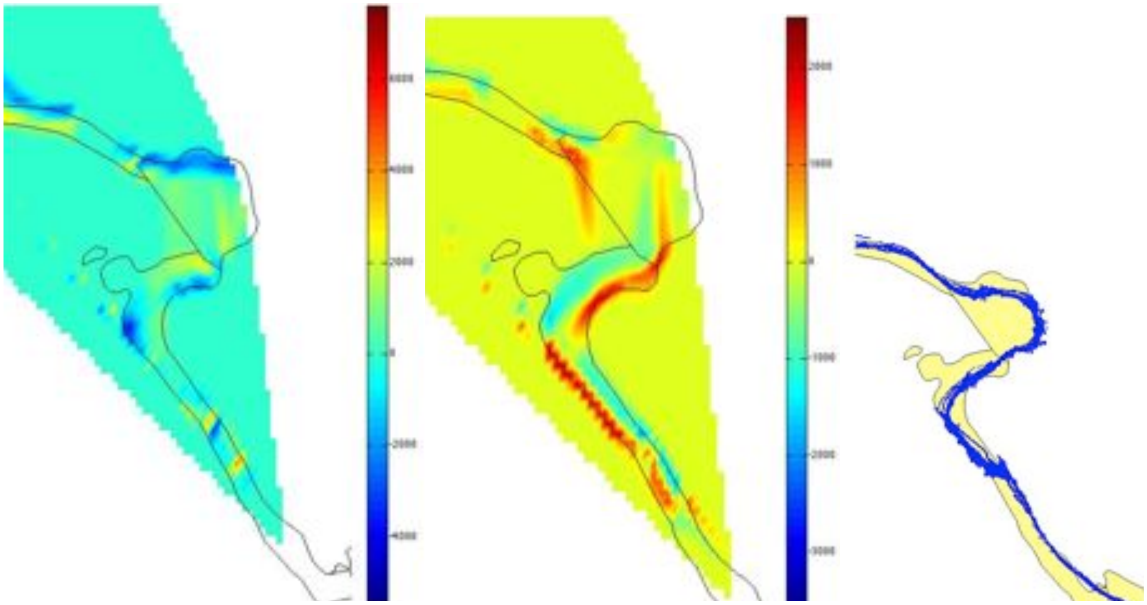


Figure 30: Divergence [Hz] (left) and curl [Hz] (center) in a curvy section of the Ompompanoosuc, with a velocity plot (right) for reference. Divergence is likely distorted by discarded data. Note the scales: deviations of div and curl from zero in the Ompompanoosuc case are an order of magnitude larger than for the Connecticut.

In summary, the magnetometer and accelerometer data from the ducks show that they tend to sit in the water at an angle of approximately 10° (although they tip frequently) and that they have no preferred heading with respect to the direction of river flow. GPS position measurements were converted into velocities, which in turn were used to create maps of the velocity field through each river. More work remains to determine exactly how well the divergences and curls of the velocity maps match predictions, but the initial results look promising. A useful plot to make in the future would be a cut of velocities across the width of the river, for comparison with theoretical models of flow velocity vs depth (*Leopold, A View of the River 168*).

1. *Open Projects*

Several open questions remain that should be addressed by the DUCKS team in the future. First, the magnetometers need to be calibrated. Timestamping of the magnetometer and accelerometer data should also be thought through carefully. At a data cadence of one sentence per second, timing of compass board sentences can be backed out with acceptable accuracy based on interleaving with GPS data. However, more sophisticated timestamping may be required at higher data cadences. Error and repeatability in compass board measurements should also be investigated more thoroughly.

The group should consult with Professor Magilligan about useful earth science data that can be collected using the DUCK infrastructure. The confusion about the telemetry buffer filling up still needs to be investigated further. Also, discrepancies in the river boundaries from the Census Bureau compared with the data from the Ompompanoosuc launch should be resolved. It would be useful to widen the boundaries such that all the data fall within them and can be included in analysis. More work also remains in evaluating the accuracy of the velocities calculated from the GPS measurements. Finally, the ducks went through interesting flow patterns surrounding bridge abutments on several occasions during the two completed missions, and the data from these areas should be studied.

V. Discussion

Many future areas of investigation follow from the present work with the ducks. One is using the river-ready duck payloads for further study of the transport of LWD. A mission that lasts for days or weeks would give a fuller picture of the long-term behavior of LWD. Since LWD is most beneficial to the environment when it becomes trapped, it would be useful to intentionally launch the payloads into a river with many obstructions and study how and where they get stuck. For future LWD experiments the group will want to redesign the payload fins to resemble LWD more closely; namely, one axis should be significantly longer than the other. Although the current design features short fins on opposite sides of the payload and a long fin on one of the other sides, the short fins together are the same length as the long one so there is no "long" axis. Another possibility would be to attach the electronics from one or several payloads to real LWD.

This summer the GreenCube group will continue to study the capabilities of low-resource payload swarms by collaborating with the Cornell Space Systems Design Studio on their Chipsats, fingernail-sized satellites that aim to exploit the "orbital dynamics of extremely small bodies" (*Atchison*). GreenCube will help them test the behavior of their satellites under two conditions: first hanging from a high-altitude balloon, and then tumbling through the atmosphere after falling from a balloon. These tests will also provide an opportunity to test the range of the software radios used by Chipsats.

In 2012-2013, the group will adapt the duck technology from use with rivers and balloons to small rockets. This change will require new, vibration-safe housing and higher power radios. The group intends to use the DNT900 series 900MHz radios (Figure

31) that have flown on several GreenCube balloon missions. These radios have a range of up to 40 miles and can use a star network configuration like that of the duck payloads. The team has developed a custom support board that supplies regulated power to the radio. It also contains several status indicator lights that blink when the radio is active and specifically when it is sending or receiving packets of data. The first series of launches will take place in Alaska and the rockets will reach an altitude of approximately 5km. Following those launches will be a Wallops Flight Facility test flight up to 200km. For the Wallops flight, the payloads would use local 900MHz radios to transmit to a main payload, which would then transmit to the ground via S-band radio.

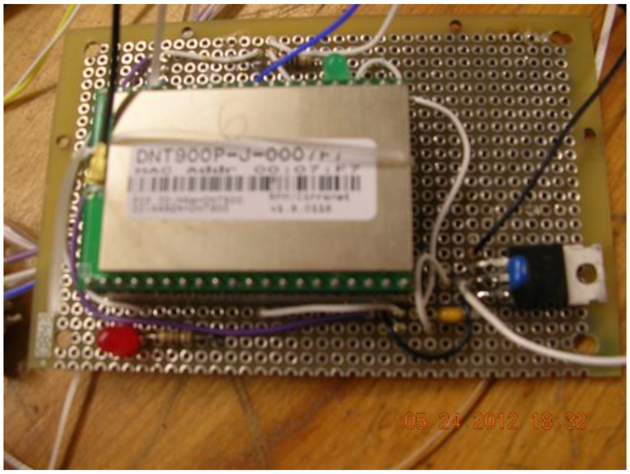


Figure 31: DNT900 radio with support board

In the longer term, several more adaptations are necessary to prepare the swarm for orbital missions. GreenCube students Peter Horak and Stephanie Malek have been working on a prototype solar panel that will allow the group to study issues related to running our electronics using solar power. In addition to transferring from single-use batteries to rechargeable solar-powered ones, additional sensors need to be incorporated for auroral physics experiments. Lynch Rocket Lab's Petite Ion Probe (PIP) instrument, a Faraday cup used to measure ion and electron temperatures, is a likely candidate.

Thanks to a dedicated team of students and their mentors, the DUCKS project grew from an interesting idea to a fully operational swarm of payloads with two completed launches behind it in the space of about seven months. Although many modifications to the ducks will be necessary as the project transitions into small rockets and beyond, the team has made great strides in developing an easily reproducible, low-resource electronics package for a swarm payload and in learning about mapping vector fields from sparse measurements. As the group continues to work towards truly multipoint ionospheric experiments, the knowledge base that they have built through DUCKS will serve as a strong foundation for future efforts.

VI. References

Atchison, Justin. "MII Overview." *Cornell University Space Systems Design Studio*. N.p., 2010. Web. 24 May 2012. <http://www.spacecraftresearch.com/MII/MII_overview.html>.

Leopold, Luna. *A View of the River*. Cambridge, MA: Harvard University Press, 1994. Print.

Leopold, Luna, M. Wolman, and John Miller. *Fluvial Processes in Geomorphology*. San Francisco: W. H. Freeman and Company, 1964. Print.

Siddiqui, Umair, Amanda Slagle, et al. "Dartmouth GreenCube: Multiple Balloon Measurements of Gravity Waves Over New Hampshire." Colloquium. Dartmouth College Physics Department, Hanover, NH. 20 November 2009.

"Large Woody Debris Fact Sheet." . Connecticut Department of Environmental Protection, n.d. Web. 23 May 2012. <•www.ct.gov/dep/lib/dep/fishing/restoration/largewoodydebrisfactsheet.pdf>.

VII. Appendix

1. DUCKS Team Member List

Team member	Role
Todd Anderson	Launch/Recovery Support
Casey Bradshaw	Ordering
Dave Collins	Electronics Guidance
Max Fagin	Mass production logistics
Ralph Gibson	Fabrication and Assembly Guidance
Jon Guinther	Vector Field Mapping, Communications
Peter Horak	Launch/Recovery Support
Ben Katz	Launch/Recovery Support
Kristina Lynch	Principal Investigator
Nina Maksimova	Mechanical Assembly
Stephanie Malek	Launch/Recovery Support
Robyn Millan	Co-Investigator, mass production guidance
Ha Nguyen	Fabrication
Dylan Sewell	Mechanical Assembly
Amanda Slagle	Mission Manager, Electronics
Tim Smith	Data Capture, Launch/Recovery Support
Tom Whalen	Data Analysis
Ellen Weburg	Fabrication Director
Patrick Yukman	Electronics, Real-time Data Display
Machine Shop	Fabrication Guidance

2. *Documentation*

The team maintains a google docs page and a wiki for documents about the DUCKS and other missions.

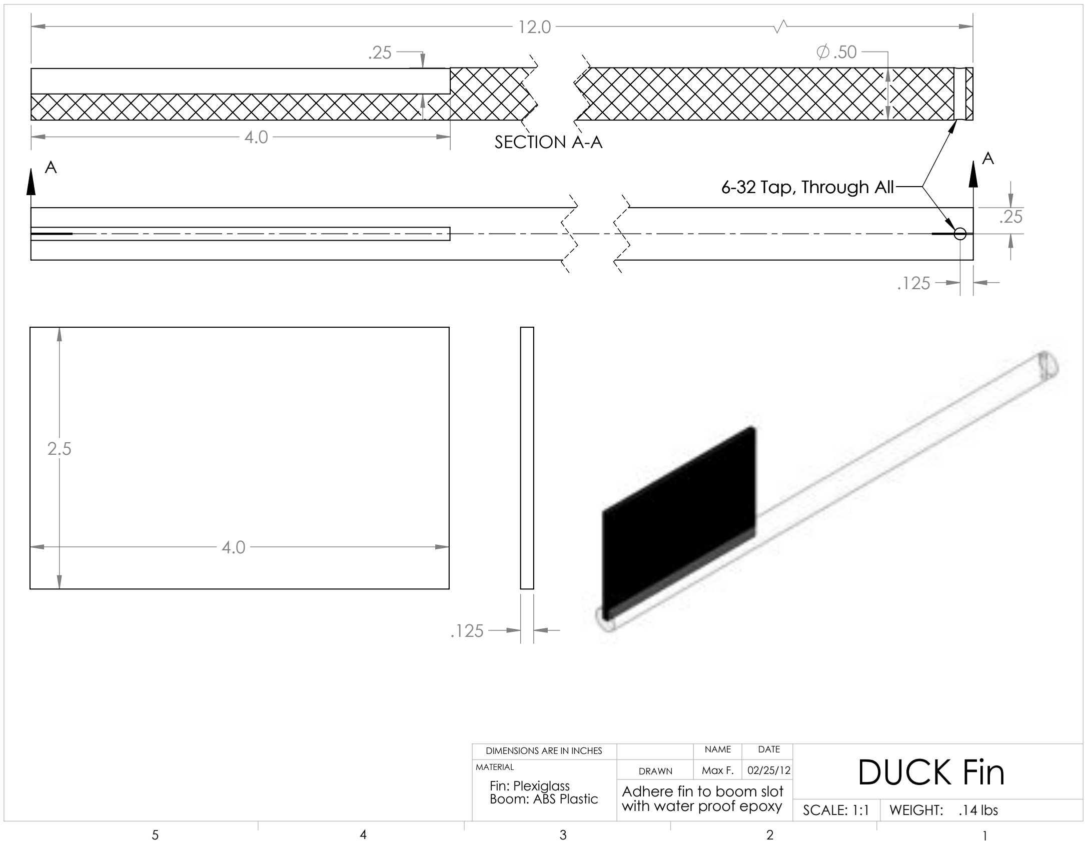
Google Docs: 317labdocs@gmail.com

This page is used for documents that require frequent editing. It houses the Mission Readiness Review document, some sections of which are reproduced in this appendix, as well as a Vehicle Log, which contains assembly instructions as well as assembly and anomaly logs.

Wiki: http://northstar-www.dartmouth.edu/~klynch/pmwiki-gc/

The wiki contains a data archive and current versions of the Arduino code, data capture, and data display software. There are also plots of launch data and general information about fluvial geomorphology, as well as a place for meeting notes and open items.

3. Duck Fin Specifications



Ellen Weburg's Additions and Revisions to the Design of DUCK Fins created by Max H. Fagin:

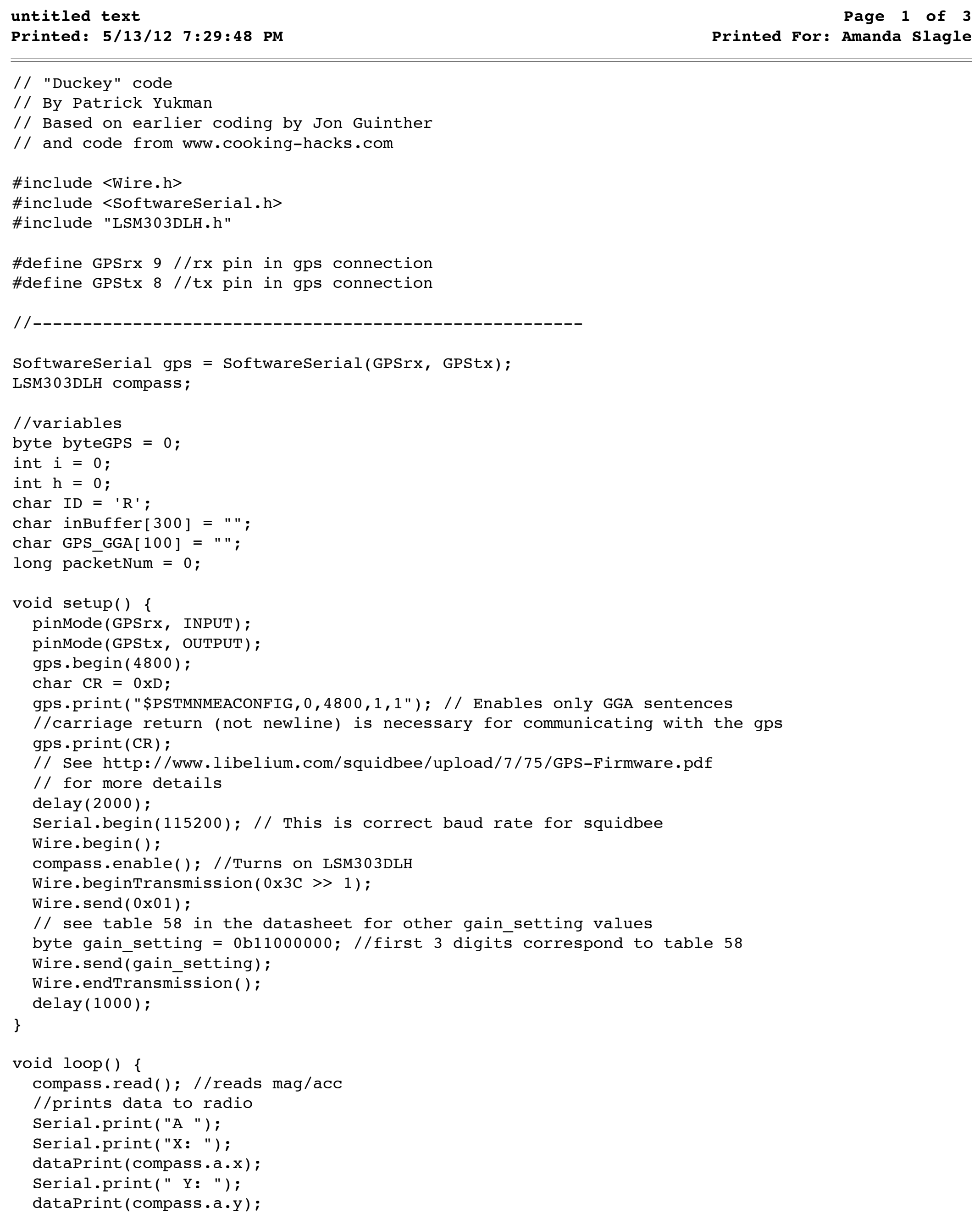
1. Instead of using a 6-32 tap, the hole .125 in. from the Tupperware end of the fin was drilled with a size 29 drill.
2. The length of the milled slot was made 4.062 in. not 4.00 in. to account for the curved shape of the mill.
3. A secondary fin with a total length of 6 in. was designed. Its fin slot was milled to a length of 2.062 in.

4. DUCKS cost analysis

The following table lists the costs for the parts that make up one complete duck. Note that where applicable costs reflect discounts for buying in bulk. For the ABS sheet, cost was determined by dividing the total cost by number of payloads built. Shipping fees have not been included. Costs for parts from Libelium have been converted from Euros.

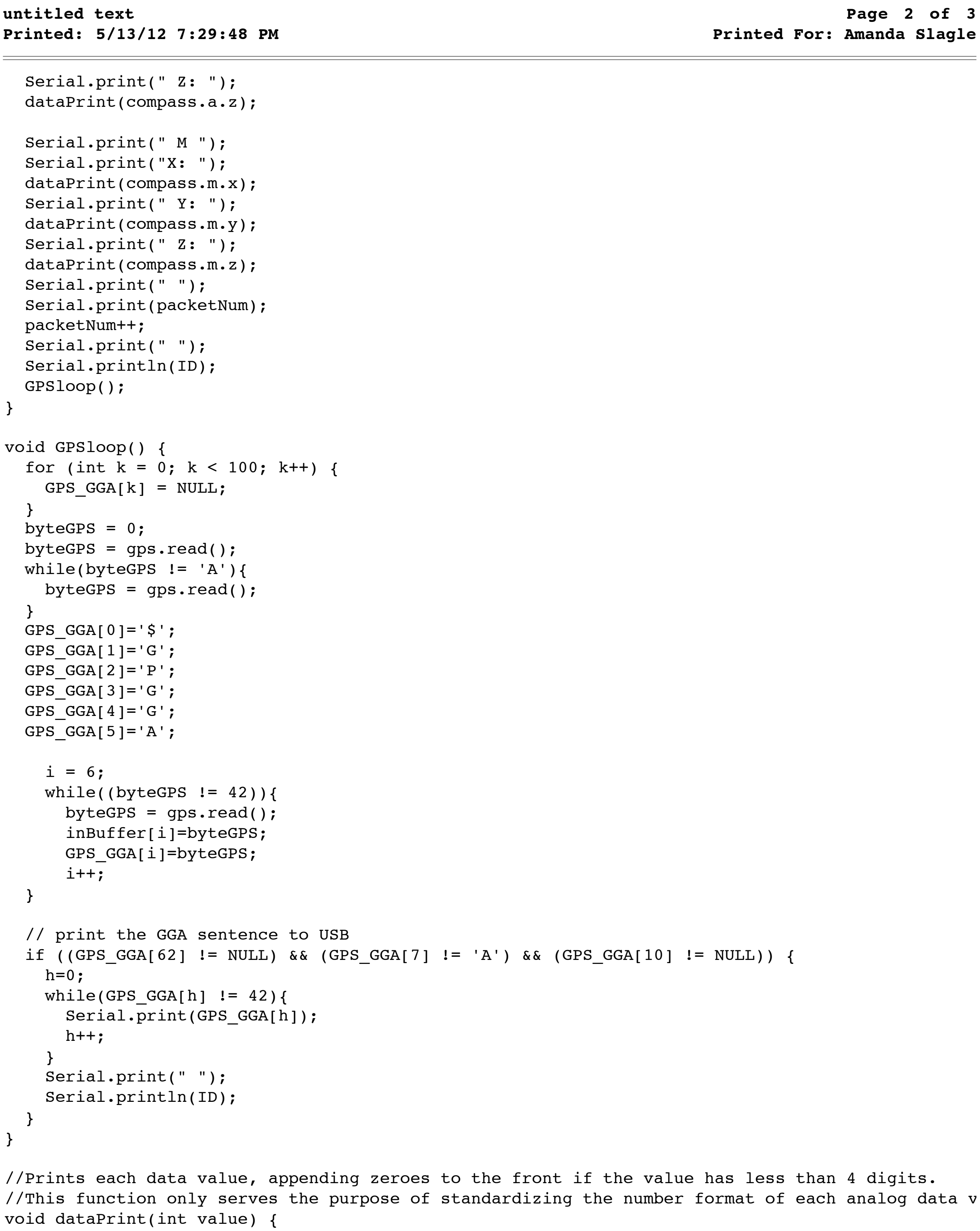
Part	Cost	Vendor
Arduino UNO	$26.96	Adafruit
Xbee shield module	$7.19	SainSmart
Xbee Pro 900	$39.00	Digikey
4.5 dBi SMA-RP antenna	$10.11	Libelium
Reverse SMA-UFL connector	$6.67	Digikey
3D compass and accelerometer	$26.96	Pololu
GPS module	$18.96	Libelium
External GPS antenna	$22.76	Libelium
5 mm LED	$0.47	Digikey
Toggle switch	$2.55	Digikey
9V Battery clip	$0.43	Digikey
9V lithium battery	$7.16	Newark
Rubbermaid 1.2L container	$6.72	Walmart
Aluminum	$10.81	McMaster-Carr
ABS sheet (black part of fin)	$1.41	McMaster-Carr
2 feet of plastic rod for fins	$3.00	Machine shop
Foam cutout	$0.00 (using Thayer school's rapid prototyper)	
Labor (~5 hours)	$50.00	
Total	$241.16	

5. Arduino Code



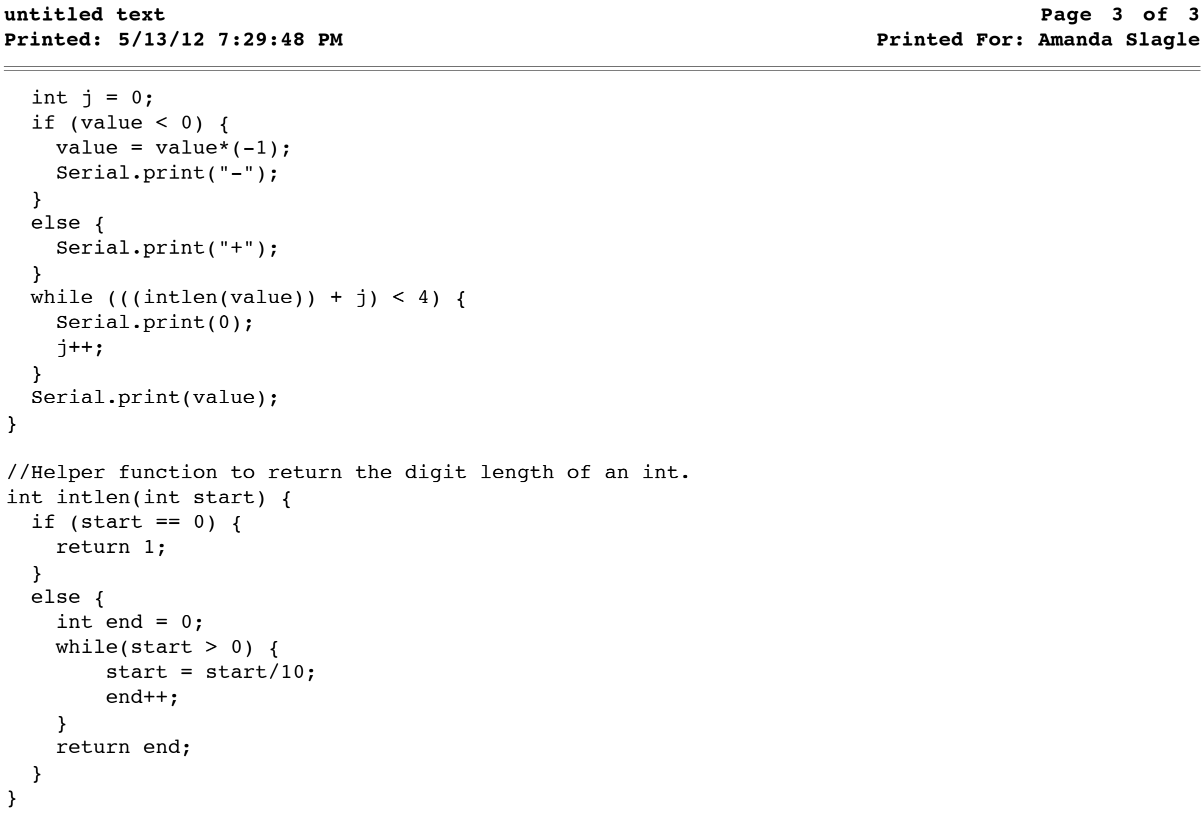
untitled text **Page 1 of 3**
Printed: 5/13/12 7:29:48 PM **Printed For: Amanda Slagle**

```
// "Duckey" code
// By Patrick Yukman
// Based on earlier coding by Jon Guinther
// and code from www.cooking-hacks.com

#include <Wire.h>
#include <SoftwareSerial.h>
#include "LSM303DLH.h"

#define GPSrx 9 //rx pin in gps connection
#define GPStx 8 //tx pin in gps connection

//--------------------------------------------------------

SoftwareSerial gps = SoftwareSerial(GPSrx, GPStx);
LSM303DLH compass;

//variables
byte byteGPS = 0;
int i = 0;
int h = 0;
char ID = 'R';
char inBuffer[300] = "";
char GPS_GGA[100] = "";
long packetNum = 0;

void setup() {
  pinMode(GPSrx, INPUT);
  pinMode(GPStx, OUTPUT);
  gps.begin(4800);
  char CR = 0xD;
  gps.print("$PSTMNMEACONFIG,0,4800,1,1"); // Enables only GGA sentences
  //carriage return (not newline) is necessary for communicating with the gps
  gps.print(CR);
  // See http://www.libelium.com/squidbee/upload/7/75/GPS-Firmware.pdf
  // for more details
  delay(2000);
  Serial.begin(115200); // This is correct baud rate for squidbee
  Wire.begin();
  compass.enable(); //Turns on LSM303DLH
  Wire.beginTransmission(0x3C >> 1);
  Wire.send(0x01);
  // see table 58 in the datasheet for other gain_setting values
  byte gain_setting = 0b11000000; //first 3 digits correspond to table 58
  Wire.send(gain_setting);
  Wire.endTransmission();
  delay(1000);
}

void loop() {
  compass.read(); //reads mag/acc
  //prints data to radio
  Serial.print("A ");
  Serial.print("X: ");
  dataPrint(compass.a.x);
  Serial.print(" Y: ");
  dataPrint(compass.a.y);
```



untitled text **Page 2 of 3**
Printed: 5/13/12 7:29:48 PM **Printed For: Amanda Slagle**

```
  Serial.print(" Z: ");
  dataPrint(compass.a.z);

  Serial.print(" M ");
  Serial.print("X: ");
  dataPrint(compass.m.x);
  Serial.print(" Y: ");
  dataPrint(compass.m.y);
  Serial.print(" Z: ");
  dataPrint(compass.m.z);
  Serial.print(" ");
  Serial.print(packetNum);
  packetNum++;
  Serial.print(" ");
  Serial.println(ID);
  GPSloop();
}

void GPSloop() {
  for (int k = 0; k < 100; k++) {
    GPS_GGA[k] = NULL;
  }
  byteGPS = 0;
  byteGPS = gps.read();
  while(byteGPS != 'A'){
    byteGPS = gps.read();
  }
  GPS_GGA[0]='$';
  GPS_GGA[1]='G';
  GPS_GGA[2]='P';
  GPS_GGA[3]='G';
  GPS_GGA[4]='G';
  GPS_GGA[5]='A';

    i = 6;
    while((byteGPS != 42)){
      byteGPS = gps.read();
      inBuffer[i]=byteGPS;
      GPS_GGA[i]=byteGPS;
      i++;
  }

  // print the GGA sentence to USB
  if ((GPS_GGA[62] != NULL) && (GPS_GGA[7] != 'A') && (GPS_GGA[10] != NULL)) {
    h=0;
    while(GPS_GGA[h] != 42){
      Serial.print(GPS_GGA[h]);
      h++;
    }
    Serial.print(" ");
    Serial.println(ID);
  }
}

//Prints each data value, appending zeroes to the front if the value has less than 4 digits.
//This function only serves the purpose of standardizing the number format of each analog data v
void dataPrint(int value) {
```



untitled text **Page 3 of 3**
Printed: 5/13/12 7:29:48 PM **Printed For: Amanda Slagle**

```
  int j = 0;
  if (value < 0) {
    value = value*(-1);
    Serial.print("-");
  }
  else {
    Serial.print("+");
  }
  while (((intlen(value)) + j) < 4) {
    Serial.print(0);
    j++;
  }
  Serial.print(value);
}

//Helper function to return the digit length of an int.
int intlen(int start) {
  if (start == 0) {
    return 1;
  }
  else {
    int end = 0;
    while(start > 0) {
        start = start/10;
        end++;
    }
    return end;
  }
}
```

6. Equipment Checklist:

___ duck payloads (20)
___ ~670g weight
___ foam cutout
___ electronics stack
___ gps antenna
___ xbee antenna
___ 9V battery
___ spare payload (Darkwing)
___ spare 9V batteries (5)
___ laptop containing:
___ Darkwing DUCKS software
___ TimCapture
___ CoolTerm
___ latest version of Java
___ Arduino software
___ latest version of Arduino code
___ Google Earth with recent cache of images
___ backed up on an external hard drive
___ laptop power cord (for use during pre-launch checks)
___ USB cables (3)
___ Ground station: Ground Control
___ Ground station: Major Tom
___ Cardboard carrying boxes (2)
___ Canoes (4) – rent 3 from Ledyard
___ Kayaks (3) – rent from Ledyard
___ Kapton tape
___ scissors
___ 5/16 wrench
___ Multimeter

7. Launch Procedures:

Procedure 1: Pre-Departure checks T-3 hours

1. ___ Run through equipment checklist and visually verify everything is in place.
2. ___ Install foam cutouts and o-rings.
3. ___ Tighten all nuts with a wrench.
4. ___ Check that GPS antennas are facing right side up.
5. ___ Take ground station and all payloads outside.
6. ___ Power up all payloads.
7. ___ Verify all payloads are transmitting
    a. ___ GPS data correct
    b. ___ magg/acc data correct
8. ___ Power down all payloads
9. ___ Replace batteries
10. ___ Power up all payloads one at a time and confirm data transmission, but do not wait for GPS lock. Power down when data transmission is confirmed.
11. ___ Power down all ground station equipment.
12. ___Confirm all payloads are off (no LEDs are lit or flashing).
13. ___ Run through equipment checklist again.

Procedure 2: Launch site T-30 min

1. ___ Acquire canoes and kayaks
2. ___ Lay out ducks on dock
3. ___ Set up ground station
    a. ___ connect laptop to outlet power if available
    b. ___ connect ground control and major tom to laptop
    c. ___ start timcapture
    d. ___ start darkwing ducks software
4. ___ Power up all ducks
5. ___ Confirm power up by checking LEDs
6. ___ Confirm that sensor data is being transmitted

Axl		Hendrix		Ozzy	
Buddy		Iggy		Page	
Chuck		Janis		Queen	
Diana		Kurt		Ricky	
Ella		Lynyrd		Stevie	
Freddie		Madonna		Tina	
George		Norah		(Darkwing)	

7. ___ Confirm gps position lock

Axl		Hendrix		Ozzy	
Buddy		Iggy		Page	
Chuck		Janis		Queen	
Diana		Kurt		Ricky	
Ella		Lynyrd		Stevie	
Freddie		Madonna		Tina	

George		Norah		(Darkwing)	

8. ___ Remove GPS antenna from Tupperware for any duck with no position lock after 15 minutes.
9. ___ Dispatch 2 kayaks to recovery site
10. ___ Restart TimCapture and Darkwing DUCKS (so that test/launch files will be separate)
11. ___ Load 10 ducks (Axl – Janis) into first canoe and 10 (Kurt – Tina) into second canoe.
12. ___ First canoe: Begin paddling across the river. Release one duck for every 1/10 of river width.
13. ___ Second canoe: Begin crossing the river and releasing ducks when first canoe is halfway across.
14. ___ Ground station canoe should stay put until all payloads have been launched.
15. ___ Other boats should go with the payloads.