



ME 426 / ECE 590 - Ocean Engineering

Hybrid Engine Group - December 2017

Team: Dara Buggay, Henrik Cox, Kai Gaemin Lee, Andrew Liu,
Benjamin Liu, Charlie Niebanck, Shreya Singh

Mentors: Professor Martin Brooke, Professor Tyler Bletsch

Table of Contents

| | |
|--|-----------|
| Abstract | 3 |
| Introduction | 3 |
| XPrize Competition | 3 |
| Drone Background | 4 |
| Hybrid Engine System Background | 4 |
| Project Updates - August to December 2017 | 6 |
| Guideline for Future Teams | 8 |
| Appendix | 17 |

Abstract

An overall system consisting of a heavy lift drone and sonar pod has been in development as Duke University's entry for the Shell Ocean Discovery XPRIZE global competition. In order to

maximise the flight time of the heavy-lift drone, a hybrid engine has been designed to mount onto the drone, and charge the depleted batteries repeatedly in mid-air. A single battery on its own has the capacity to fly the drone for minutes, but including the hybrid engine, the drone has the potential to fly for multiple hours.

Introduction

In order to significantly enhance the flight-time capacity for the heavy lift drone, a hybrid system consisting of a DC motor coupled to a 2-stroke gas engine has been designed and manufactured. The main goals of designing such a system include: automatic engine ignition when the measured voltage of the drone's battery falls to 3.2V, controlling the charging voltage to 25V using servos on the choke and throttle, automatic engine cutoff when the battery charges to 3.8V, and balanced mounting on the fully-developed drone with minimal vibration caused by the running engine. The majority of recent updates on the project involve optimising the engine to run successfully, refining the electronic components used within the system, developing a dummy 'load' for long-term test runs, mounting servos to automate throttle and choke control, and designing a method to mount the hybrid engine to the drone.

XPrize Competition

Advancements in ocean technology has improved our understanding of the seas and underwater landscapes, but our knowledge has yet to scratch deeper than the surface. Modern devices make airborne digital mapping within the grasp of a general consumer, but underwater mapping carries it's own additional challenges and pressures. Finding a more efficient method of producing high resolution bathymetric maps of the ocean floor is a global task, and the Shell Ocean Discovery X-Prize challenges the design and manufacture of a semi-autonomous device launched from shore or air, which can map 500 km² of the ocean floor at depths of 2000m and 4000m in the span of a few hours. A grand prize of \$4 million is offered to the winning team and a NOAA bonus prize of \$1 million is available for a team able to find and identify the source of a chemical signal during the competition.

Drone Background

Present day ocean floor mapping requires capital investments over \$1 million, and daily operating costs exceeding \$60,000. In order to cover the vast scale of ocean floor required, compromises would need to be made in resolution, power consumption, and sensor weight, as only a fraction of government, academic, and private investment is made to deep ocean exploration. In order to maximise the efficiency and effectiveness of our bathymetric exploration technology, it was decided that airborne travel coupled with sonar underwater technologies would prove to be the ideal solution.

An eighteen-rotor heavy lift drone has been developed to carry, deliver, and retrieve a sonar diving pod used to collect data points for underwater landscape mapping. The pods are hermetically sealed and equipped with a 28 kHz sound system for SONAR, a variable buoyancy diving system, IMU and GPS based positioning sensors, WiFi and LoRa for module data communication, and a chemical signature sensor.

At predetermined locations, the drone will drop the sonar diving pod. The pod will sink to a measured depth and collect data of the surrounding area, and rise to the surface awaiting retrieval from a winch system on the drone. The pod and drone will then continue to the next location for further data collection.

In order to cover a large mapping area, it is expected that the drone would be able to fly for several hours, all at once or in stages. Gasoline engines would provide enough fuel to power the drone for the required time, but are not permitted by the competition as the sole form of energy resource. Electric motors have exceptional RPM flexibility and greater control, but the use of a battery alone on the drone would last for a few minutes. To overcome this obstacle, three gasoline powered generators will mount to the drone to create a hybrid engine system to charge depleted batteries.

Hybrid Engine System Background

The hybrid engine/generator system is composed of a DLE-60 Twin 60cc two-stroke model aircraft engine and a Turnigy 80cc Outrunner BLDC brushless motor. The brushless electric motor is powered by two [SIZE?] LiPo batteries in series, controlled by an arduino, and connected to an ESC. The motor is coupled to the engine with two turned steel shafts and a high strength lovejoy joint. The engine is connected to an ignition box, and runs current back through the motor, into a rectifier circuit, which will provide power outlets for a charging battery or dummy load. The aim of this system is to improve the flight time of the drone to several hours long. Other components that make up the system include: a battery eliminator circuit, servos, and an adafruit shield.

Previous mechanical challenges faced during hybrid engine development included machining proper shafts and couplings for the engine and motor to handle the generated torque, and developing a sturdy frame to be as lightweight and durable as required.

As inherited, the system was mounted in a sturdy and lightweight aluminum frame but was unable to run, and many of the electrical components required new connections or replacement after previous test runs left charred connectors. The engine and spark plugs were filthy and required carburetor cleaning fluid and replacing in order to combust the fuel within the engine. The previous team had upgraded the motor-engine coupling using keys and keyways so it was able to handle the required amount of torque to run the engine. The use of set screws in the coupling easily came loose with some testing, but did not seem to cause the engine to fail. The largest priorities to undertake included troubleshooting issues that stopped the engine from starting (and continuing to run), mounting the servos and automating the throttle/choke control, replacing and fixing many of the electrical components and circuitry, and further designing a mount on the drone for three of these hybrid systems.



Project Updates - August to December 2017

A. Electrical components

The majority of electrical components required replacing or upgrading for ease of use or lack of function. Our first ESC was tested using an arduino microcontroller with potentiometer to control current output to the ESC (and motor). This test resulted in the some of the capacitors on the ESC exploding and bursting the component into flames. Although the cause of this failure was never 100% conclusive, it is believed that a sheared connection on the rectifier circuit caused a current backfire through the ESC when the potentiometer was high enough. The second ESC had new connectors and heat guards soldered and heat shrunk on, and was reprogrammed with an ESC Programming Card to handle 12S (two LiPo batteries in series) and turned off the braking feature, which would otherwise cause abrupt stops to the motor when the potentiometer was turned back to zero. Discussion on reprogramming ESC's is found below in the 'guideline for future teams.'

At the start of the semester, we were using an arduino as the microcontroller for the ESC. This controlled the movement of the motor, which was required to start the engine. Initially the Adafruit shield was removed from this component and a potentiometer (twisting knob) was added to manually control the rotation of the motor. Once we ran our first test, resulting in the loss of our first ESC, the arduino with potentiometer was replaced with a commercial

microcontroller identified as a small orange box. This device has multiple modes of operation pre-programmed in, and a built-in potentiometer for manual motor control. Closer to the end of semester, when the engine started running fairly consistently, we swapped back to control the start of the motor using a pre-programmed arduino (without adafruit shield) powered by a small USB power pack product. This arduino is programmed to quickly start and run the motor for about 4-5 seconds before turning off. The reset button can be pressed in order to start the motor rotation program again.

As mentioned earlier, it was discovered that one of the connections of a diode on the rectifier circuit had sheared off. This required a quick diode replacement, with new connections screwed and heat shrunk into place. Finally, in order to allow for long test runs, a dummy output load was made using heaters to act as large resistors. These were drilled and secured in place into a bucket filled with water, and replaced the large LiPo battery for charging. By measuring the current over these output leads, we can find how much power is being transmitted to the charging battery (or during test runs, the water-heater setup). Ideally, 25V would be measured in the output leads to charge the battery sufficiently.

B. Starting/running the engine

The majority of work on this hybrid system consisted of troubleshooting engine issues that prevented it from starting. The engine itself required cleaning out using 'Gumout' carb/choke & parts cleaner, and a small spray of 'Prestone' starting fluid. The spark plugs were dirty from old fuel within the engine, which were replaced, and the engine emptied after using carburetor fluid cleaner. The fuel line was sized and was secured so that the vacuum was tight enough between the fuel tank and inlet. Multiple tests on the system were ran using the ESC to start the motor, or rope to hand start the engine, all while experimenting with the choke position and throttle amount.

Throughout all these tests, it was important to ensure the engine did not flood, and was emptied if it did. During a couple test runs, a couple of the set screws from the coupling came loose from the vibration. Hence, loctite blue was used to secure these into position. On the day of testing for the XPrize judges (round 1), the carburetor was replaced in order to improve the engine's likelihood of starting/running, and one of the exhausts were replaced, as a deteriorating gasket was causing some fuel leak when the engine tried to start.

C. Servo control

In order to demonstrate the automatic ignition process for the engine, two servos mounted on a 3D-printed holder were programmed to adjust the throttle and choke of the engine to maintain a voltage output of 25V to the charging battery (or water heater). The next step of this process will include attaching the metal rods (most likely after shortening them) that push the throttle and choke to the rotating servo fins. Code for the arduino controlling the servos exists (in google drive folders) and allows the servos to react to the output voltage accordingly.

D. Frame and mounting

Past designs have involved mounting 3 separate motor-engine systems to the drone in order to balance the drone. However, this adds an incredible amount of weight that exceeds the competition's guidelines, and was proven to provide less overall power and be less efficient than using a single hybrid engine system. With this in mind, the new placement of the singular motor-engine setup is in the center of the heavy-lift drone, above the winch and below the drone's automation platform. Similarly to this platform, the engine's frame and circuitry will sit upon its own platform, held by carbon fibre rods attached to the drone's sides. Images of the CAD layout are shown below. The next stage of this process will be manufacturing the platform and attaching rods to sit the hybrid engine system upon, and likely most the automation platform vertically higher on the drone to make more space.

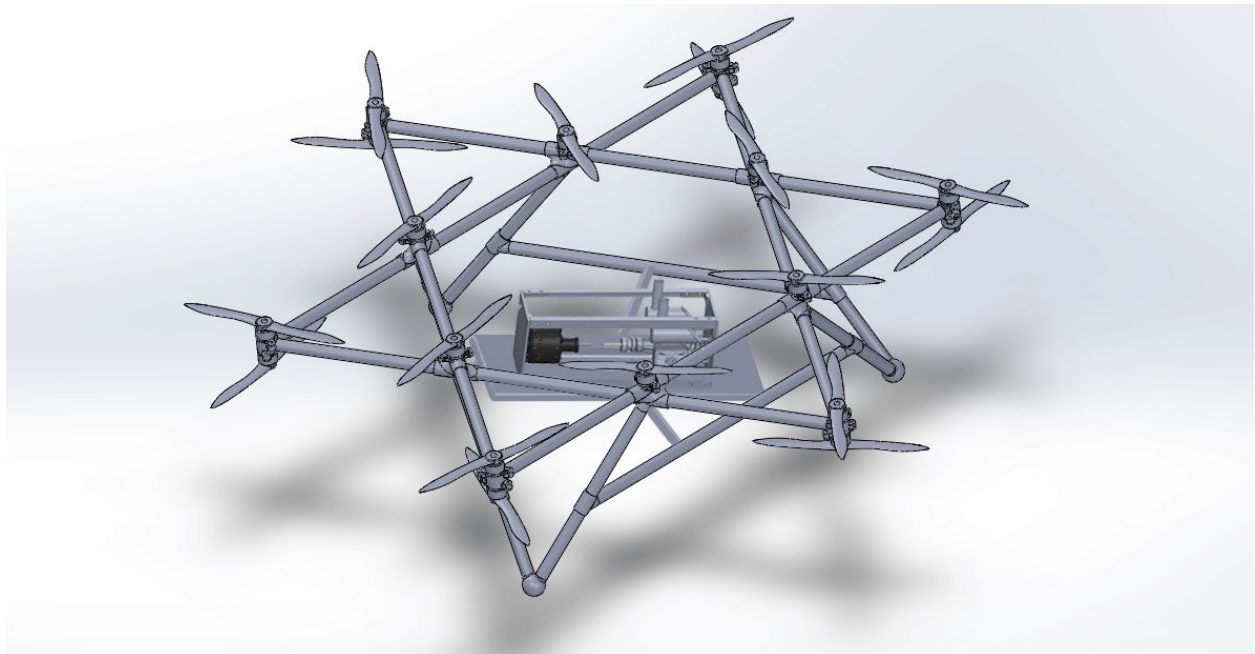


Figure : Solidworks models of full drone, and hybrid engine system (without circuits, leads or coupling), photoshopped into position together

Guideline for Future Teams

Frame Design - as written by the Spring 2017 team

“The current framing design consists entirely of a self-contained frame that can be directly attached to the redesigned landing gear. It is designed as a rectangular prism with parallel 6061 aluminum square faces (8”x8”x0.190”) on which the motor and engine will be mounted, connected on ‘alternating’ planes by 6061 Aluminum 90° angles (1”x1”x1/4”) and cross-braced with 6061 Aluminum bar (1”x1” cs). The connections are made with 4-hole L brackets. While carbon fiber was considered in the redesign process, the machinability, rigidity, and ease-of-connection between parts made aluminum the final choice. Further, there was limited proof supporting the notion that the use of carbon fiber would meaningfully change the mass of the system. Finally, the notion of self-containment is most important for mounting the sets on the drone. With an entirely self-contained frame, the system can be treated like a point mass; the mounting points are geometrically balanced to minimize the effects of gyroscopic forces from each system.”

- Hayes Griffin, Frank Jones, Ernesto de Losada, and Henry Meiring. March 24th 2017

Code

There are a number of scripts that have been written by previous year’s team and this year’s team to run and test the hybrid engine. They can be found in the GitHub link:

<https://github.com/tamran/Team-Hydra>

The components within the hybrid engine system that require code are:

1. The arduino microcontroller - starting the motor
2. Servos - controlling throttle/choke

ESC and reprogramming

The ESC was programmed multiple times during the testing of the hybrid engine using the Programming Card, pictured below. The left 3-pin connector on the bottom of the Programming Card is for the ESC receiver cable (the shorter of the two 3-pin female connectors of the ESC). The right 3-pin connector on the Programming Card is to supply power to the card. Connect the

<https://www.youtube.com/watch?v=a7kbMhBaUA4>

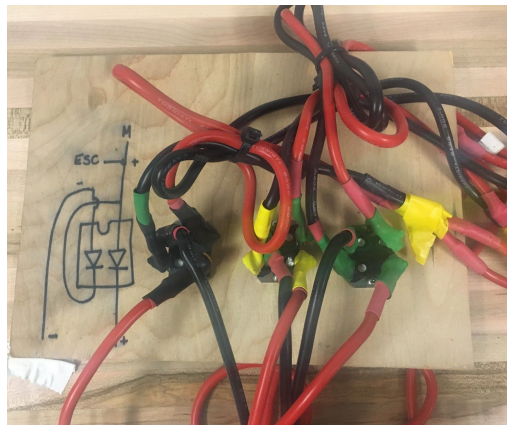


At times, it is necessary to reprogram the ESC being used. The primary settings made throughout the Fall semester included setting the cell count to 12, and turning OFF the braking feature. When reprogramming the card, remember to connect the ESC (using the longer of the two 3-pin female connectors of the ESC) to the motor rig (a single motor mounted in its own aluminum frame). When connected, reprogramming the ESC on the card should be accompanied by beeping sounds to confirm a setting has been saved onto the ESC. It is also

possible to start the ESC when connected to the battery and test on the single motor, in order to see the ESC and arduino microcontroller behavior.

Rectifier circuit

The rectifier circuit takes AC power generated from the motor/generator and converts it into DC power capable of charging a LiPo battery. As discussed earlier in the report - a single sheared connection on this rectifier circuit may cause the current to backfire when running the motor and cause the ESC to burst into flames. Ensure all connections are secure, and the output + and - leads are clearly labelled. Diodes are easy to replace when required. When mounting the circuit onto the drone, it is possible to cut the wood it's mounted on currently into a smaller shape, and screw this into the platform that will hold the engine. But for the sake of aesthetics, it is more likely preferable to replace the wooden platform.

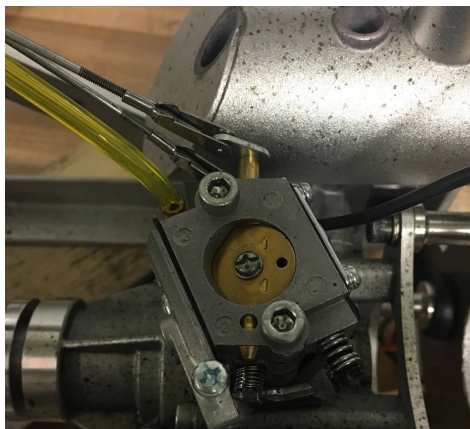


Exhausts

The exhausts discharge smoke and fumes when the engine is running (and when the fuel mix inside the carburetor is rich). Ensure these are secure, and the gaskets don't look withered and deteriorated to prevent leaking from joint connections. These are easily removed and replaced; if the exhaust starts to loosen when the engine runs, apply loctite blue to the inside screws to secure the joint more.

Metal Rods - Choke and Throttle

The choke and throttle are controlled by pushing/pulling against two metal rods attached to the carburetor. Our best success has been with an open choke to start the engine, and throttle activated after the motor has ramped up to full speed (controlled by code on the arduino). If the engine is having trouble to start, try starting the engine with closed choke and full throttle activated. This will create a rich fuel mixture in the carburetor (which may flood the engine and require emptying), and the engine can be run again with an open choke, and about 50% throttle.



Servos

In order to automate the engine ignition process, servos attached to a 3D printed holder will be mounted to the frame of the engine. These servos will connect to the metal rods and will be programmed with code to adjust the choke and throttle to start the engine, and run it to maintain a voltage output of 25V to the charging LiPo battery.

Carburetor & Spark Plugs

The carburetor includes the fuel intake (tapered nozzle with ridges), screws to adjust the fuel/air mix inside, and fins to control the throttle and choke. The cleanliness of the carburetor determines how smoothly the engine will run, and can be cleaned using carburetor cleaner found within the Foundry.

The spark plugs are found on the sides of the engine, and are removed (pull off cap, and use wrench to unscrew the plugs) to drain the engine if flooded. In order to test that the spark plugs do create a spark (at the right time, when the piston pushes against the outer wall of the engine), follow the process below:

1. Unscrew the spark plugs and push inside the caps pulled off earlier to reveal the spark plugs
2. Hold the plugs to the frame, so that they are in contact with metal. Tape them down into position
3. Turn off the lights for best visuals - hand turn the motor/engine coupling, and small sparks should show at the ends of the plugs.
4. If no sparks show, the plugs may require cleaning or more likely need replacing. If the sparks show, but the engine still has trouble starting, empty the engine in case of flooding, and determine whether a flooded engine was the issue.



Coupling

The coupling between the engine and motor is a love-joy machined by the Spring 2017 team. During our test runs, this coupling has handled the required torque well, and we've only experience a couple of loosening set screws from it before (these were secured into place using loctite blue). In the early stages of the semester, we using ratchet straps and rope to try and hand start the engine. The rope and straps were wrapped around this coupling as it provided the best space for rotation.



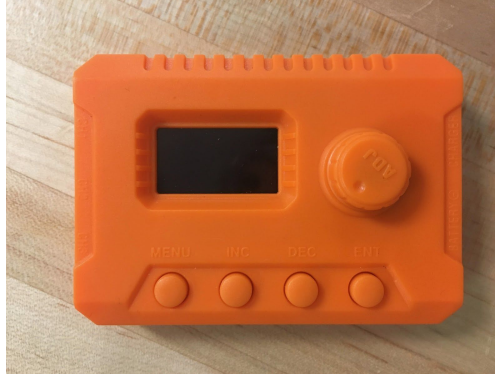
Ratchet Straps

Use the blue (smaller) ratchet straps. These were purchased specifically for the engine team as the larger yellow straps are far too large for this part of the project. For best results, tie down the engine to its own table.



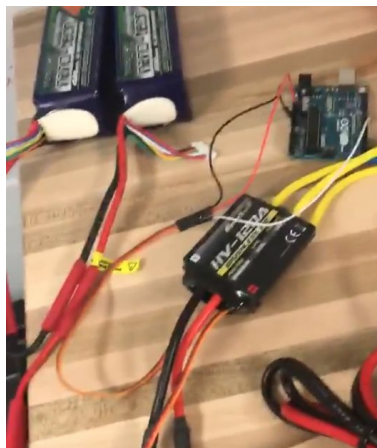
Orange box - manual microcontroller

The manual microcontroller can be used as you diagnose issues with the arduino and engine. Connect the longer of the two 3-pin female connectors of the ESC into one of the ports on the left side of the microcontroller, and the 7 multi-colored wires from a 6S LiPo battery to the right side of the microcontroller. When starting the motor, turn the potentiometer (knob) to 100%, but only for a few seconds maximum. Running the microcontroller will warm up the ESC and may cause melting/burning/exploding of the ESC if left on for too long.



Arduino - automatic microcontroller

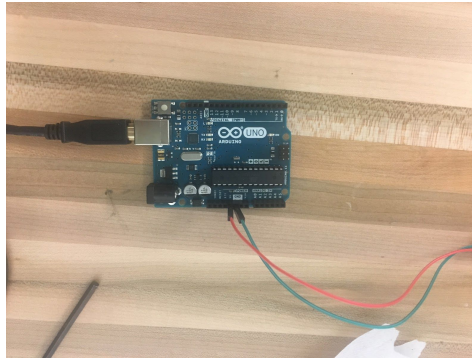
In the image below, the arduino is used as the automatic microcontroller to start the motor. It is preprogrammed with code to ramp up to full speed within a couple seconds, and shut off after a few more (this won't stop the motor though, as the braking feature on the ESC is turned off). It is connected to the longer of the two 3-pin female connectors of the ESC, and on the other side, is connected to a 5V USB power pack source. However, these USB portable battery chargers are not the ideal source of power. Although they have worked throughout the semester, it is highly encouraged to replace these as power sources, to both the arduino microcontroller and the ignition.



Ignition

The ignition box (not pictured) fires the spark plugs of the engine. This is powered by the arduino pictured below, which is simply used to take 5V from a USB power source and channel it to the ignition box. As stated earlier, these USB power sources should be replaced as soon as

possible, and in the case of the ignition box, it is ideal to replace the arduino with a 5V wire that only requires soldering to join the connection.



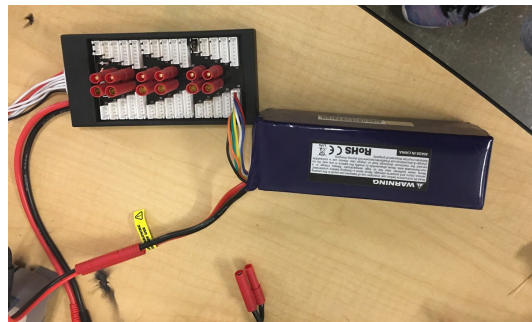
Water dummy load

In order to test the engine system for multiple hours at a time (or longer than a few seconds), we needed to manufacture a dummy load to replace the charging LiPo battery. The bucket is filled with water, and 6 large heaters acting as resistors are inserted into the lid on top. Each heater is connected to switches, used to activate the resistor and increase the load for the output. One switch includes all the black wires and is the grounded switch. It is possible to switch each heater on one at a time (two sets of two and two sets of one heater(s) are linked due to a lack of switches) while the engine is running, but so far the engine has not continued to run with more than two heaters switched on, as the resistance load is too high. This bucket's input leads are connected to the rectifier circuit's output wires.



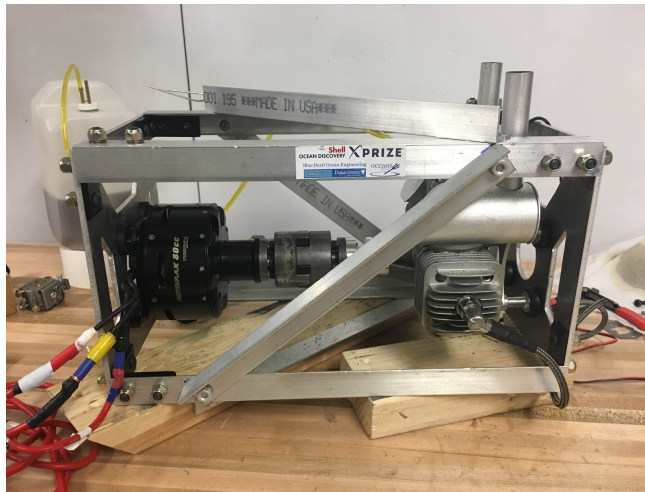
Batteries & Charging

Each battery is charged at the battery charging station within the Foundry. Specific instructions on how to use these stations are in videos found on the Hybrid engine team's google drive folders. Two 6S LiPo batteries are used to connect to the ESC when running the motor and engine, and a third is only required to connect to the orange box manual microcontroller if it is being used. A large, green, 12S LiPo battery will be connected to the output wires of the rectifier circuit in order to be charged while the drone is in mid-flight. The image below shows a 6S battery connected at the Foundry battery charging station.



Fuel Tank

The fuel tank is pictured below behind and to the left of the engine's main frame. The yellow tubing allows for clear visualisation whether the carburetor is receiving any fuel, and whether the tube's vacuum is secure enough for the engine to suck fuel. Throughout the semester, the fuel tank has been placed in the vertical position as pictured without issues. But if suction becomes an issue, it is best to secure the fuel tank horizontally to allow an easier flow of fuel to the carburetor.



Connecting everything together

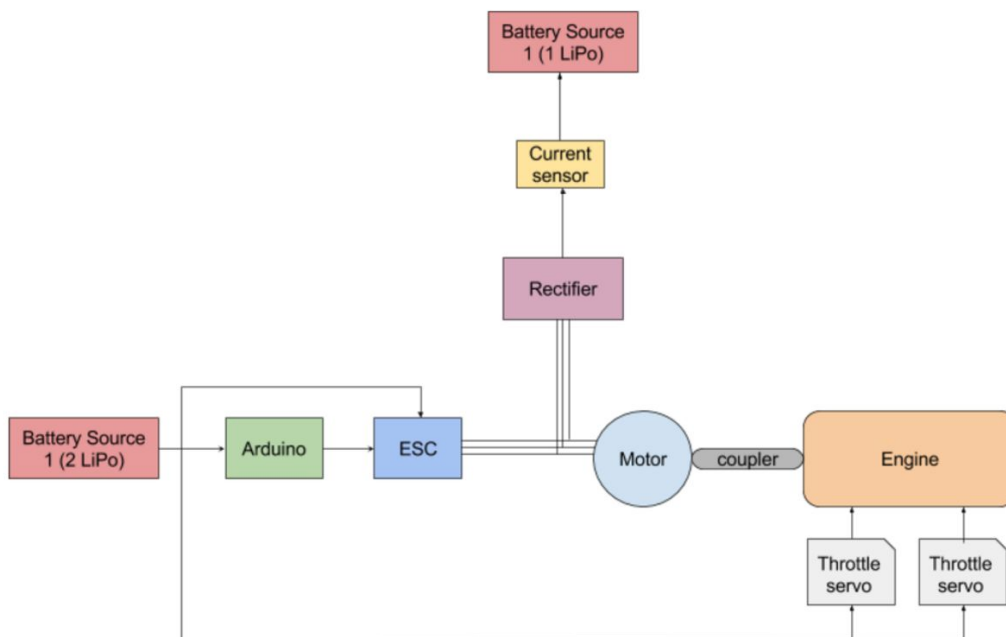


Figure : A schematic of the hybrid engine system and its components.

Killing the engine

When the engine is running, it is likely that pulling the arduino microcontroller, or the ESC, will cause the motor to spin sporadically. In this case, promptly remove the ignition box's power source and the two 6S LiPo batteries connected to the ESC to quickly stop the motor from running.