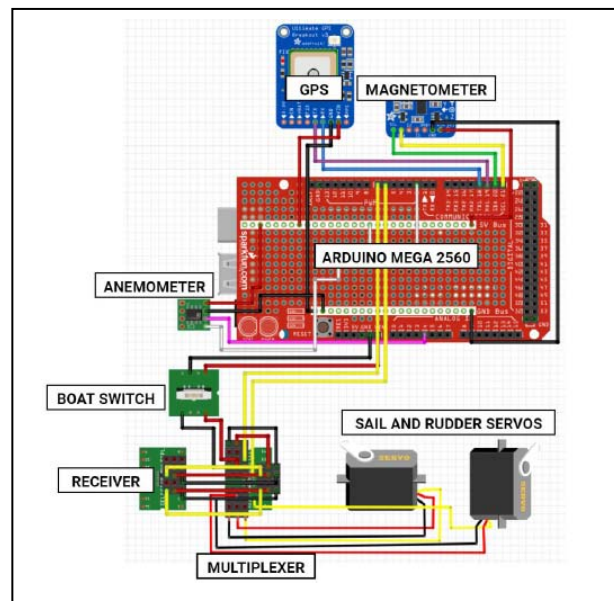


# Autonomous Hydrofoil Sailboat Manual

by

V. Tran, B. Zhao, J. Leung,  
M. Shi, and S. Bleveoet

Edited by: Drs. N. Sarigul-Klijn & Platzer



June 2018

**[ipgs@ucdavis.edu](mailto:ipgs@ucdavis.edu)**

**iPGS (innovative Power Generation Systems) Laboratory  
Department of Mechanical and Aerospace Engineering  
University of California, Davis  
2132 Bainer Drive  
Davis, California 95616**

iPGS is a research laboratory directed by Professors Sarigul-Klijn and Platzer to investigate Autonomous Energy Ship Concept based on Wind over Water. This concept enables the provisions of renewably generated electricity and renewably generated hydrogen.

**For further information contact founding directors Dr. Nesrin Sarigul-Klijn and Max Platzer  
via phone: (530)-752-0682, or email: [ipgs@ucdavis.edu](mailto:ipgs@ucdavis.edu)**

**How to cite this technical report as a source:**

1. V. Tran, B. Zhao, J. Leung, M. Shi, and S. Bleveoet, Edited by: Drs. N. Sarigul-Klijn & Platzer, 2018, *Autonomous Hydrofoil Sailboat Manual*, iPGS Technical Report TR 2018.06.01, University of California at Davis, 2018.

***©Copyright 2018 by the authors. All rights reserved.***

# Table of Contents

<b>Deployment Checklist</b>	<b>3</b>
<b>Electronic Wiring Connections</b>	<b>6</b>
Arduino, Actuation, and Sensor Connections	6
Power Supply Connections	8
<b>Raspberry Pi Operation</b>	<b>9</b>
Connecting to and Shutting Down the Raspberry Pi	9
Downloading New Arduino Libraries to the Raspberry Pi	11
Add New Arduino Code Scripts	12
<b>Troubleshooting</b>	<b>14</b>

# Deployment Checklist

Before heading to the testing location, make sure you have all the sailboat and controller components. The troubleshooting kit is recommended for last-minute troubleshoot.

*Table 1: Complete components list for testing*

Sailboat	Controller	Troubleshooting Kit
Boat hull	Charged battery and UBEC	Extra wires
Sail	Anemometer (with mount)	Electrical tape
Sail-to-boom string	External GPS antenna	Duct tape
Boom	Magnetometer	Small screwdriver
Mast (2 parts)	Arduino Mega (with Protoshield)	Wire cutter/ stripper
Rudder	Raspberry Pi	Digital multimeter
Keel	Power splitter	Extra wires (M-M, M-F, F-F)
	Ethernet cable	Scissors
	Transmitter	
	Receiver	
	Multiplexer	
	Enclosure cap and adapter	

Step 1: Bring all sailboat, controller, and troubleshooting kit components

Step 2: Power and connect all components

1. Battery is fully-charged
2. Battery, UBEC, Raspberry Pi (RPi), and Arduino are properly powered
  - a. RPi will have a blinking green light (OS running) and a constant red light (powered)
  - b. Arduino will have a constant orange light (pin 13 LED)
3. Connect the RPi and Arduino with the USB 2.0 to USB 3.0 cable
  - a. Check that RPi has registered the Arduino port through the software Arduino Integrated Development Environment (IDE)
4. Connect multiplexer to the Arduino, receiver, and servos.

5. Connect anemometer RJ11 connector to the RJ11 breakout board
6. Connect external GPS antenna to GPS
7. Connect magnetometer wires to Arduino. Power (red), GND (black), SDA (yellow with black tape flag), and SCL (yellow without flag). SDA is pin 20 and SCL is pin 21 on the Arduino.

Step 3: Carefully put all necessary components inside the enclosure in this order.

All components, except the RJ11 breakout board, have velcro markers inside to help with placement.

1. Battery against the wall on the port side of the boat
  - a. Leave the white tape ends of the power splitter on top for easy power disconnection.
2. Raspberry Pi against the wall on the starboard side of the boat with the USB cable end facing towards the center of the enclosure
3. Arduino Mega in the center of the enclosure at a diagonal. Power cable and USB end first will fit across the diagonal, minimizing the distance between the USB ends of the cable between the RPi and Arduino.
4. Multiplexer tucked in with the flat side against the ceiling on the port side edge of the opening.
5. Receiver tucked in next to the multiplexer with the flat side against the ceiling on the port side edge of the opening.
6. Anemometer RJ11 breakout board lays on top of the Arduino protoshield

Step 4: Check actuation

1. Turn on the remote control transmitter
2. Turn on the switch on the boat to power the servos. The blue receiver light should be on. This should be solid, as opposed to blinking, once it has connected to the transmitter.
3. Check that the rudder and sail servos move in manual mode
4. Check that the rudder and sail servos move as expected in autonomous mode

Step 5: Wirelessly connect to the RPi

Step 6: Uploading Arduino code

1. Open and upload the controller script with the Arduino IDE
2. Open the Arduino serial monitor to check that the sensors are getting proper readings
  - a. Latitude, longitude, speed, apparent wind direction, apparent wind speed

Step 7: Run the data logger

1. Open the dataLogger.py script on the Desktop
2. Check that the serial port listed in the script (variable “ser”) has the same name as the serial port listed in the Arduino IDE
3. Check that there is not currently a .csv file on the desktop named “boatTracker.csv”. If so, rename or delete it.
4. If an Arduino serial monitor is open, close it.
5. Run the data logger and check that the output does not say “Keyboard Interrupt”. This may have to be run a few times before it registers.

Step 8: Close the cap to waterproof the enclosure

1. Before putting the cap on, twist the cap (with the wires) counterclockwise 3-4 rotations to prevent the wires from being twisted while deployed
2. Place the cap on and carefully turn the cap clockwise to seal the enclosure, making sure that the O-ring does not leave its designated groove. In high temperatures, this may be more difficult due to thermal expansion.
3. Check that the wireless connection still works. If not, open the cap and check the connections.

# Electronic Wiring Connections

## Arduino, Actuation, and Sensor Connections

The figure below details the existing soldered connections and necessary wired connections for the Arduino.

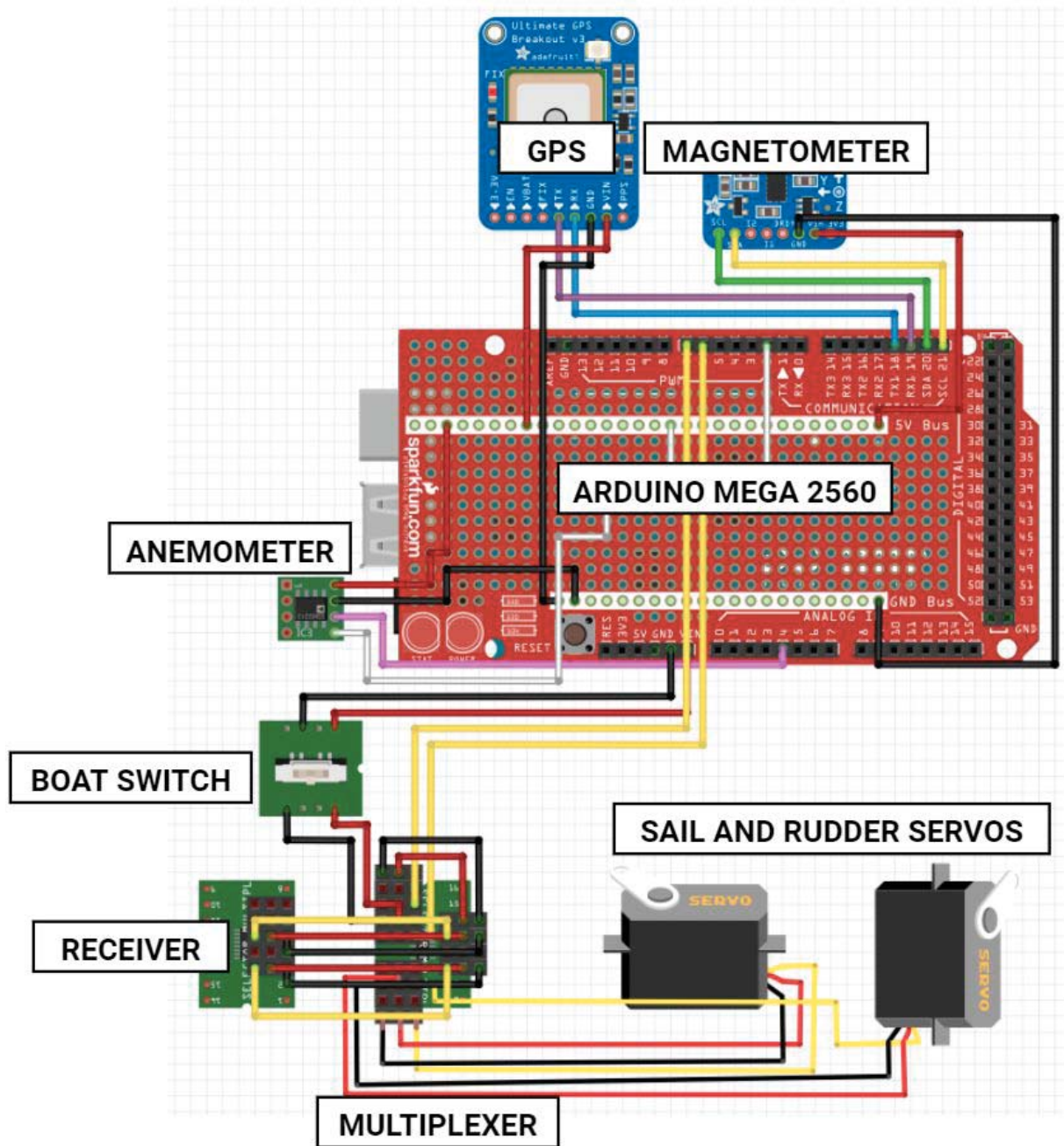


Figure 1: Soldered and wired connections of Arduino

The multiplexer pinout is detailed below in the figure below:

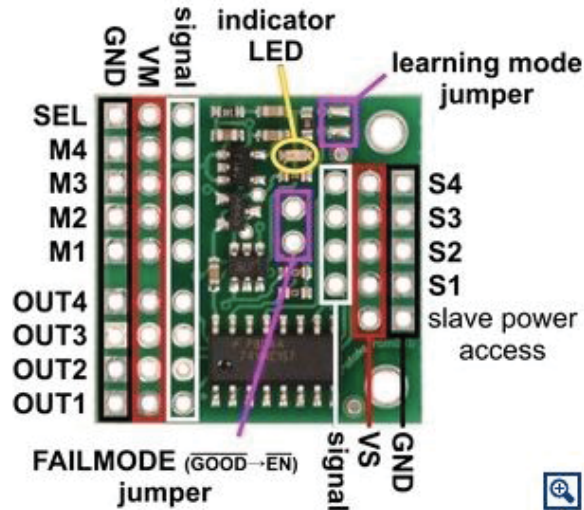


Figure 2: Multiplexer Pinout

The rows of the corresponding table show the connections. For example, the M3<sub>SIGNAL</sub> pin is connected to the Arduino D7 pin:

Table 2: Multiplexer wire connections

CONNECTED PINS	
Power	
SEL <sub>VM</sub>	S4 <sub>VM</sub>
SEL <sub>GND</sub>	S4 <sub>GND</sub>
Boat Switch <sub>VM</sub>	M2 <sub>VM</sub> or M4 <sub>VM</sub>
Boat Switch <sub>GND</sub>	M2 <sub>GND</sub> or M4 <sub>GND</sub>
Rudder Servo	
M1 <sub>SIGNAL</sub>	Arduino D6
S1 <sub>GND</sub> , S1 <sub>VM</sub> , S1 <sub>SIGNAL</sub>	Receiver CH1
OUT1 <sub>GND</sub> , OUT1 <sub>VM</sub> , OUT1 <sub>SIGNAL</sub>	Rudder servo cable connector



Sail Servo	
M3 <sub>SIGNAL</sub>	Arduino D7
S3 <sub>GND</sub> , S3 <sub>VM</sub> , S3 <sub>SIGNAL</sub>	Receiver CH3
OUT3 <sub>GND</sub> , OUT3 <sub>VM</sub> , OUT3 <sub>SIGNAL</sub>	Sail servo cable connector

## Power Supply Connections

To properly split the current from the battery to power the RPi and Arduino, follow the diagram below, making sure to check that the power and ground wires of the screw terminals on the barrel jack converters are connected to the correct terminal.

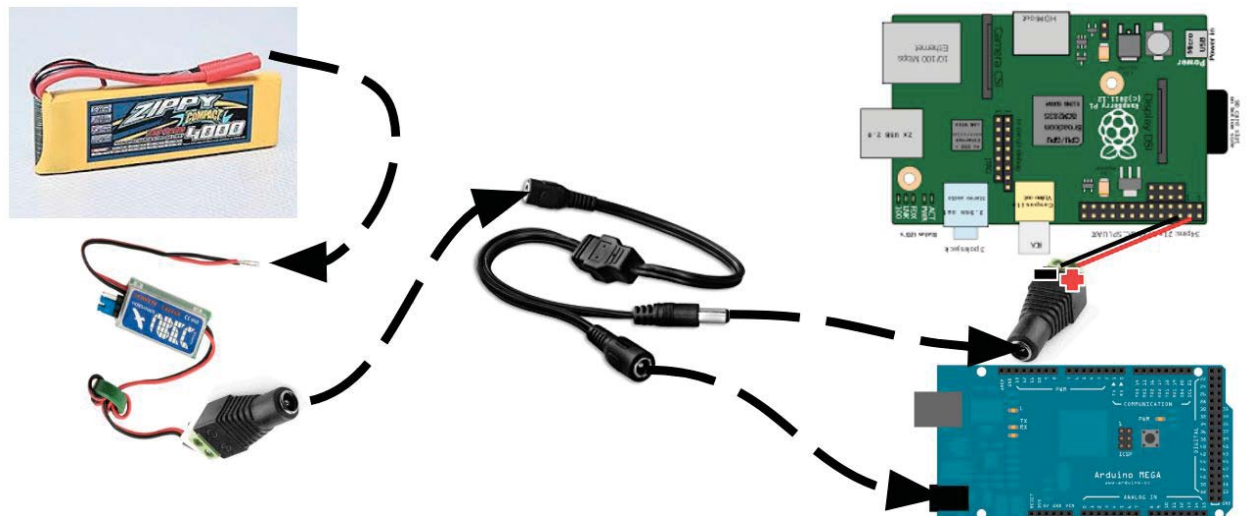


Figure 3: Power splitting connection diagram

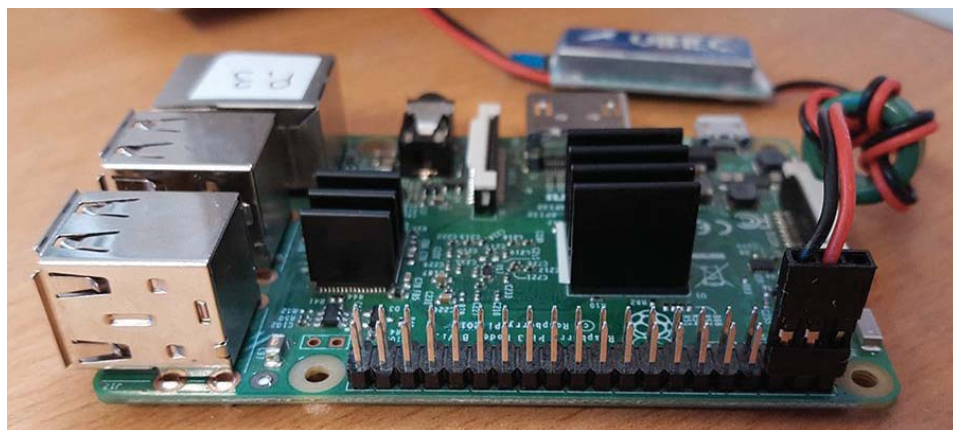


Figure 4: Power pins for RPi

Source: <https://www.modmypi.com/blog/using-a-ubec-with-a-raspberry-pi>

# Raspberry Pi Operation

Currently, the Raspberry Pi (RPi) is set up with the Raspbian Stretch operating system. All of the data (e.g. programs, files, and operating system) is stored on the SD card.

## Connecting to and Shutting Down the Raspberry Pi

Use these steps to connect to the Raspberry Pi for the first time and set-up a wireless connection.

### Step 1: Connect to the RPi

This can be done with SSH (<https://www.youtube.com/watch?v=IDqQIDL3LKg>) or by following the instructions below. For the latter, you will need to connect the RPi to a monitor with an HDMI cord and use a USB keyboard and mouse to interact with it.

1. Use the HDMI cord to connect the monitor and RPi.
2. Plug-in the USB keyboard and mouse.
3. Power the RPi with either the micro USB connector to the port or use the Li-po battery as detailed above.
4. See the Raspberry Pi Foundation's introduction article for more details. (<https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started/4>)

### Step 2: Connect the RPi to a wireless network.

From here, you should be able to see the RPi desktop and be able to connect to a new wireless network by

1. Clicking on the symbol with the two red crosses on the taskbar.



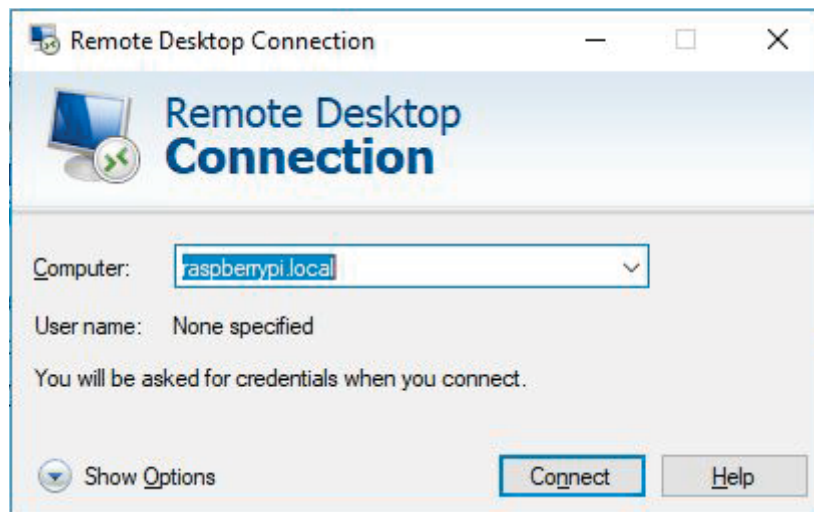
*Figure 5: Connection symbols on the taskbar*

2. Selecting the wireless network and type in the password. For deployment purposes, this was done using a cell phone hotspot.

### Step 3: Connect your remote device (i.e. Windows computer) to the RPi with a remote desktop

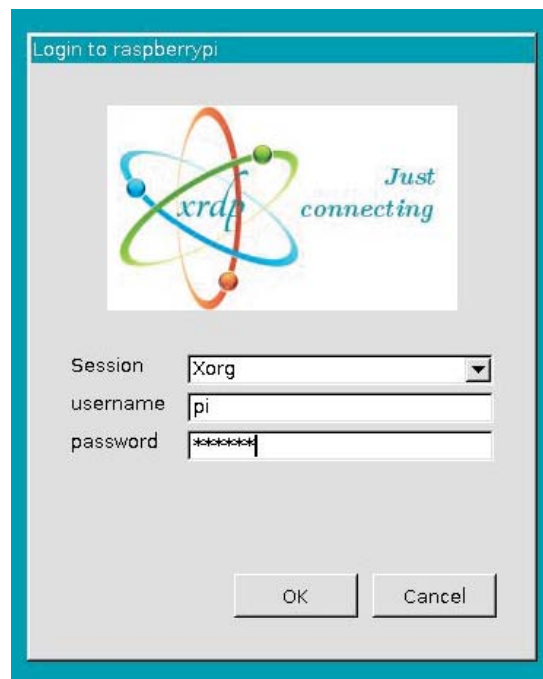
1. Connect your remote device to the same network (e.g. cell phone hotspot) as the RPi.
2. From your start menu on a Windows computer, search for "Remote Desktop Connection". This should be pre-installed on Windows computers.

3. Set the computer name as “raspberrypi.local” and click “Connect”. This should bring you to the RPi’s login screen. See the troubleshooting section if you’re having trouble connecting.



*Figure 6: Remote Desktop Connection login*


4. The username is the default “pi”, and the password is “emel185”.



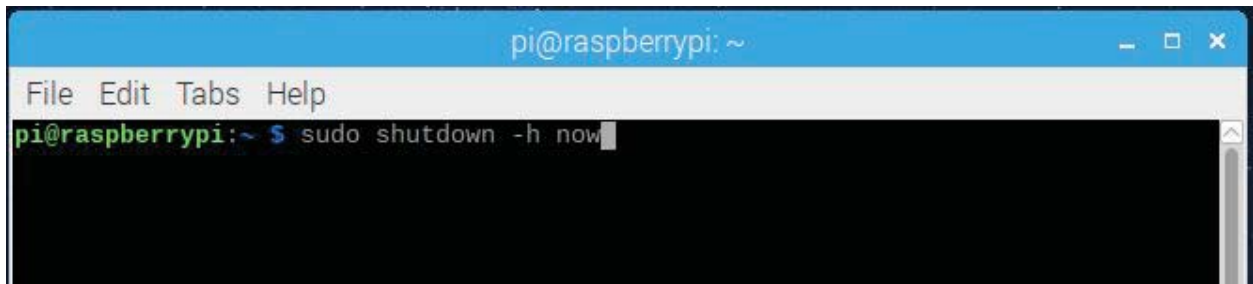
*Figure 7: RPi login information*

#### Step 4: Shutting down the RPi

This step is important to prevent data corruption.

1. Open a terminal window on the taskbar by clicking on the icon 

2. Type “sudo shutdown -h now” and press “Enter” to shutdown the RPi. This will disconnect your remote desktop.



*Figure 8: Terminal shutdown command*

## Downloading New Arduino Libraries to the Raspberry Pi

Currently, the magnetometer, GPS, and anemometer libraries are already added to the system. These instructions are for adding new Arduino libraries:

Step 1: Download and unzip the folders. If these are downloaded from the web, the folders will be located in the “Downloads” folder.

1. Find and open the folder.
2. Check that the .h and .cpp files are not in another folder or else the Arduino IDE won't find the library header. If the library folder name has any spaces, rename the folder, replacing the spaces with underscores to make a single, connected name.

Step 2: Move the library folder to the correct Arduino library folder

1. Using the terminal, change directories to find the folder using the “ls” command to list the content of the current directory and “cd” to change directory (e.g. cd /home/pi/Downloads). See the Linux commands documentation for more detail (<https://www.raspberrypi.org/documentation/linux/usage/commands.md>).
2. Still in the terminal, move the folder to the designated Arduino libraries folder using “sudo” for administrative permission and the “mv” command (e.g. sudo mv libraryFolderName /usr/share/arduino/libraries/). The library should now be available in the Arduino IDE.

# Add New Arduino Code Scripts

## Method 1:

- Open the Arduino IDE and start a new script. If you have pre-existing code, copy, paste, and save the code.

Method 2: Using GitHub through Terminal (go to Method 3 if you do not wish to use the Terminal with GitHub)

1. After creating a GitHub account, email [beezhao@ucdavis.edu](mailto:beezhao@ucdavis.edu) or [jjleung@ucdavis.edu](mailto:jjleung@ucdavis.edu) your username and email that was used to make the account
  - a. Either Bryan or Jordan will give you access to the GitHub to edit and upload files
2. *Using the Terminal (for LINUX/UNIX machines)*
  - Open up a terminal on your computer, and then navigate to your desired directory (e.g. typing “\$ cd ~/Desktop” without the \$, the \$ simply indicates the start of the line in Terminal)
  - Type in “\$ git clone <https://github.com/ucdsailboat/autonomous-sailboat> ” to save a copy of the repository into your desired location
  - Create your Arduino script using Method 1, and save it into your desired subdirectory:
    - For Arduino code related to hardware, save the script into the “Arduino Hardware Code” directory in the repository
    - To confirm that the script is in the correct directory, type “\$ ls ” to list all the files in your current directory
  - Then, to upload your Arduino script that you have created, type in: “\$ git add <filename> ” where <filename> is the name of your Arduino script (e.g. rudder\_controller.ino)
  - Then, enter “\$ git status ” to verify whether you have added a file in the queue
  - Type in “\$ git commit -m “Enter short message here” ” to commit your file that you have added
  - Finally, to push your local changes to the cloud, type in “\$ git push ” and ensure that you have correctly added your file(s) to the GitHub. More information about pushing your files to the git can be found at

<https://help.github.com/articles/adding-a-file-to-a-repository-using-the-command-line/>.

### Method 3: Using the GitHub website without Terminal

1. Navigate to GitHub website: <https://github.com/ucdsailboat/autonomous-sailboat>
2. After creating a GitHub account, email [beezhao@ucdavis.edu](mailto:beezhao@ucdavis.edu) or [jjleung@ucdavis.edu](mailto:jjleung@ucdavis.edu) your username and email that was used to make the account
  - a. Either Bryan or Jordan will give you access to the GitHub to edit and upload files
3. Download files directly or clone the repository onto the Desktop
4. For more info on using GitHub without the terminal, go to: <https://pixelpioneers.co/blog/2017/using-github-without-the-command-line>.

# Troubleshooting

## Step 2: Power and connect all components

Part 2 : If the RPi or the Arduino fails to remain powered constantly (inconsistent blinking of green light on the RPi or inconsistent orange light from the Arduino), the power may be an issue. A common problem encountered came from the screw terminals, shown in Figure 1. Move the red and black wires around to see if power is affected. Ensure that the red and black wires are screwed properly and secured by wrapping electrical tape around.



Figure 9: Screw terminals that power the Arduino and RPi.

Part 3: If the RPi fails to register the correct serial port after connecting the Arduino to the RPi, replugging the USB cable to another USB terminal may do the trick. A few iterations might have to be done for the RPi to register that a serial port is connected. If after a few tries this does not work, a possible solution is to use another USB cable.

Part 4: Most of the problems encountered here come from a lack of consistent power. If this is the case, ensure again that the Arduino and RPi are properly powered. Another problem may be that the wires need to be replaced. Refer to the wiring diagram to ensure that all the wires are connected properly.

Part 5: Check to see if the anemometer is getting any readings. Most problems came from a loose connection with the RJ11 and the RJ11 breakout board. If this is the problem, use electrical tape to ensure that the connection is secured.

## Step 4: Check actuation

If there's an issue with manual or autonomous mode, the issue may be a loose or broken wire. If the blue receiver light is not on, there may be a power issue upstream with the Arduino. This is likely due to a loose or broken wire.



## Step 5: Wirelessly connect to the RPi

If your computer can not find the RPi, wait for a few seconds for the RPi's operating system to fully boot up. You may have to try reconnect another few times. The team found that one of the latest Window's update prevented this connection, so you may need rollback the update by following this article.

(<https://www.howtogeek.com/235474/how-to-roll-back-builds-and-uninstall-updates-on-windows-10/>)

## Step 7: Uploading Arduino Code

Part 1: If there is a timeout issue with the connection between the RPi and Arduino, there are a few options to fix this.

1. Check that no other program is running serial communication (e.g. serial monitor or python data logger)
2. If the problem is still not fixed, on the Arduino protoshield, press the "RESET" button
3. If the problem is still not fixed, unplug the Arduino from the RPi and re-plug it in to reset the serial connection. If the serial port changes on the Arduino IDE, match the new serial port name from the Arduino IDE to "ser" on the data logging script before running the script.