ROSS

Raspberry Pi / Matlab GUI instructions  
(PixHawk, winch, CTD, …)

Jasmine Nahorniak  
June 17 2016  
(541) 740 5488  
jasmine@coas.oregonstate.edu

These instructions describe the basic steps needed to set up and monitor the communication between the various ROSS kayak components managed by the Raspberry Pi and monitored by the Matlab GUI. Currently, these components include the PixHawk, Winch & CTD. Work is in progress to also include the ADCP, GPS, and camera(s).

# Hardware

**Components Required**

* Ebox
* Antenna for Ebox
* Power source (Battery Box, Lithium polymer (Lipo) battery, or wall power adapter)
* Cable(s) to connect the Ebox to the power source
* Laptop
* Power source for laptop (wall or laptop battery)
* Antenna and board for laptop (must be paired with the Ebox antenna with a matching color sticker)
* USB cable to connect the antenna to the laptop
* Power source for antenna (wall power adapter or 12V lead-acid battery)

**Connections**

Ebox/Winch

1. Make sure the antenna is connected properly before powering on to prevent any damage to the system:
   1. Set the switch on the XTend daughterboard in the Ebox to “on”.
   2. Connect the antenna to the outside of the Ebox.
   3. Ensure the antenna is connected to the XTend board
2. Connect the Ebox to the Battery box, but do not power it on yet.
3. Connect the Winch to the Ebox and to the Winch Battery.
4. Turn the Winch on. *The Winch must be started before the Ebox.*
5. Turn the Ebox on using the Battery Box switch.

NOTE: The RPi, PixHawk, and Arduino must all be powered on at the same time in the Ebox to function correctly. If you need to power cycle a component of the Ebox, please power cycle the entire Ebox, do not just power cycle an individual component.

NOTE 2: The RPi has a UPS board attached – it takes 40 seconds to power off. Since the RPi supplies power to the PixHawk and Arduino, they also take 40 seconds to power off.

Laptop

1. Power on the laptop.
2. Make sure the laptop antenna is plugged into the laptop XTend board.
3. Make sure the two antennas (laptop and Ebox) are at least 18” apart at all times.
4. Do not plug the XTend USB into the laptop yet or the antenna will try to use the laptop as a power source.
5. Power on the XTend board (antenna) for the laptop.
6. After the antenna has powered on, plug the XTend daughterboard into a laptop USB port.

# Software

## RPi

The Raspberry Pi is configured to automatically start up the needed script on boot.

If you need to view the output from the Raspberry Pi, or restart the code, take the steps below. Note that this is only possible if the Raspberry Pi is within wifi range of the laptop.

1. Connect the laptop to the Raspberry Pi wifi (“ROSS”) – it should automatically connect when it sees it. It may take a minute or two after the RPi is powered on.
2. Using PUTTY (the icon is on the lower taskbar), load ROSS and connect to it. It may fail the first time (or 4!) if you try to connect before the RPi is ready. When it does connect, log in as “pi” with username “raspberry”.
3. cd /home/pi/kayak
4. To view the output from the code (if started on boot):
   1. sudo screen -R

To exit the screen without killing it, “Ctrl -a d”

To kill the screen and the program, “Ctrl-a k”

1. To view a log file of the status messages:
   1. cd /home/pi/kayak/kayak\_custom\_out/
2. To start the program if it’s not running;
   1. sudo screen /home/pi/kayak/kayak\_main.py
   2. To exit the screen without killing it, “Ctrl -a d”
   3. To kill the screen and the program, “Ctrl-a k”
3. To change the configuration (such as the kayak name or baud rates);
   1. vi /home/pi/kayak/kayak\_config.py
4. If you are running the kayak without a winch, you will need to edit the main script so it doesn’t try to connect to the winch. This is easily done near the top of the kayak\_main.py script. There’s a parameter that lists all of the components that need connections (pixhawk, modem, winch). Note, however, that you CAN run the code with just the Arduino/teensy connected even if the winch isn’t powered on.
5. The bulk of the code is located in /home/pi/kayak/kayak.py. This single file contains all of the key functions needed.
6. The /home/pi/kayak/10-kayak-rules.dev script is used to match a USB connection (eg the pixhawk) to a particular path (eg /dev/ttyPixHawk). This matching depends on knowing the model and serial number of the device being connected. If changes need to be made, you will need to edit this file. After editing, copy the file to /etc/udev/rules.d to implement the change. These setting must be correct in order for the scripts to be able to successfully connect to the devices.

## MATLAB

On the laptop, launch matlab using the icon in the taskbar at the bottom of the screen. At the command prompt, type;

cd c:\Users\ROSS\kayak-pc

kayak\_main(‘kayak1’)

The argument is the name of the kayak. This name is set in the configuration file on the RPi.

run the following

NOTE: If you are visiting a new location, you will need to download the map before you go (you’ll need an internet connection for the download). We currently have maps for (a) Willamette River, (b) Newport bay by the dock, and (c) Reser parking lot. To get a new map, take the following steps:

1. Open kayak\_config.m
2. Comment out the existing selected location using “%”
3. Add a new section for the new location, with new lat/lon extents. Try to select a region that is approximately square so the GUI view will be reasonable.
4. Save the config file.
5. From the matlab prompt, type;
   1. kayak\_googlemap\_grab

This will download the google map and store it on the laptop. This only needs to be done once for this set of coordinates. Next time you run the kayak\_main script with those exact coordinates in the configuration file, it will use the downloaded map.

### GUI

To launch the GUI:

cd c:\Users\ROSS\kayak-pc

kayak\_main(‘kayak1’)

There are currently three tabs on the GUI: navigation, ctd, winch.

To zoom or pan, use the small icons at the top of the figure window.

Just below the zoom/pan icons is a panel with the kayak name (KAYAK1) and the current kayak status (date, time, lat, lon). There are also two buttons (“Start Boat” and “Stop Boat”). These buttons stop and start the kayak. These buttons have been tested only in the lab, not in the field with the boat.

**NAVIGATION**

To load waypoints from a file, use the “load waypoints” button. The file must be in the following format:

44.2356 -123.5678  
44.2357 -123.5680

Where the first column is the latitude (decimal degrees North) and the second column is the longitude (decimal degrees East). Negative values are south/west. You may upload as many waypoints as the pixhawk can handle ☺

To select waypoints using the cursor on the map, use the “select waypoints” button. Your cursor will change to a cross symbol. Click once per location to select your waypoints. Double-click your final waypoint, or hit enter when done. Your new waypoints will turn blue and will be listed in the textbox in blue. At this point, the waypoints are only stored on the laptop – they have not been sent to the pixhawk yet. To redo your selection, simply hit the “select waypoints” button again and start over. To send your waypoints to the pixhawk, use the “set waypoints” button. The waypoints text box will go blank. After receiving the new waypoints, the pixhawk will send the new waypoints back to the laptop – this is your way of verifying that the pixhawk has received the new waypoints. The pixhawk waypoints will display on the map and will appear in black text in the textbox. If you wish to double-check the pixhawk waypoint values at any time, use the “get waypoints” button.

The number of the next waypoint to be visited will be shown in the Next WP text box, and the waypoint location dot will be colored red on the map. The next waypoint will change as the kayak moves from waypoint to waypoint. To change the next waypoint number at any time (if you want to skip a waypoint, for example), simply enter the desired waypoint number in the text box and hit the associated “set” button.

Any time a parameter in a text box is changed and the button “set” is pressed, the text box will first go blank while it waits for a response from the pixhawk. It will then be repopulated with the value of that parameter as received from the pixhawk. This ensures that the pixhawk has the value that you sent it. If it sends back the wrong value, try again – there may have been a communication issue. (I haven’t seen this problem yet, but it may occur at longer distances or on rough seas.)

The current kayak location will be shown in white on the map. It may not be obvious when you are stationary. The location will be plotted every time a new status message is received from the pixhawk. You can modify the number of track points to plot (the breadcrumb trail) by changing the “# Points to Display” value. The default is 20, which means you will only see the 20 most recent locations on the map. Decreasing the value (from 20 to 10 for example) doesn’t delete the data history – it simply doesn’t display them. Increasing the value (from 10 back to 20) will display them again. To completely delete a track, click the “delete track” button.

A maximum throttle (THR\_MAX) value of 30 is recommended for this kayak (kayak1). Faster speeds make the radius of turns much larger.

WP\_RADIUS is the waypoint radius. This is the acceptable distance (in meters) from a waypoint for the waypoint to have considered to have been reached. The default is 10 m.

GROUND SPD is the ground speed. I have started working on this parameter but it hasn’t been fully implemented or tested yet.

The final box in this list, that by default lists THR\_MIN (throttle min) can be used to set any pixhawk parameter. The correct name of the pixhawk parameter must be known to use this function.

## CTD

This tab is still in development. The leftmost figure is empty. The rightmost figure displays the maximum depth reached by each CTD cast as a function of time. This figure updates immediately after each CTD data download. This is a new feature that has been tested in the lab but has yet to be tested in the field.

## WINCH

This tab is also still in development. The figure is currently empty, but in the future it will display winch status information as a time series such as revolutions/min, revolutions, and direction (up/down/stationary).

The winch can be controlled from the boxes on the right. All stop buttons are immediate.

To requests casts, select the speed out (an integer representing a percentage of maximum speed, eg. 50), speed in, the depth to go to (expressed as number of winch revolutions from the surface, e.g. 50), and the number of casts you would like to perform. For continuous casts, simply enter a high number (e.g. 100). Once these instructions have been submitted using the “send command” button, the first cast will take place as soon as the winch status is ready (STATUS 1) if the CTD is not currently downloading. After each cast, when the CTD reaches the surface, Robert’s CTD download software is started – this software connects to the CTD wifi, downloads the data, and then parses it in ASCII format. The maximum depth reached is then calculated sent back to the laptop for plotting. Once the download is complete, the next cast will take place. This will continue until all casts have been collected. These operations take place on the RPi, independent of the laptop. To stop the casts at any time, use the stop winch buttons on the laptop matlab gui.