**Benthic Mapper**

**CSU Channel Islands & ESRM**

**Preface:** This guide is intended for use with the Benthic Mapper created at CSU Channel Islands for the ESRM lab. This guide will detail instructions on how to setup and use the specific Raspberry PI, Arduino, and mapping frame that were developed for the project. This guide, for the fist half, will assume that everything is ready and installed. The second half will detail how to start from scratch with the files provided on this git repository.

**Getting Ready:** To begin, make sure that the Arduino board and Raspberry Pi are connected via USB cable. Then, connect the Pi to the provided portable battery pack and plug that into the PI’s micro usb port. You can tell the PI is on if you see the leds light up on the PI and/or the Arduino, since the Pi is supplying the Arduino with power as well. Make sure the Arduino, PI and the battery pack all fit inside of the waterproofed box correctly, no hanging wires, and ensure the box is closed tight to prevent water getting in. The current safe depth to dive with the box is approximately 30ft.

Once the box is settled, ensure that each GoPro is setup correctly. There are 3 GoPros, all the same model. Ensure that their internal clocks are synced up as this will ensure more precise data later on. Next, make sure the repeating photo option is chosen (looks like a camera with a circle spinnging around it) and make sure the timing is set to 1 picture/.5 seconds, or lower. This will ensure the best overlay and a good amount of pictures with which to work with. Place a camera in each of the GoPro cases mounted on the benthic mapper.

Affix the left and right arms of the mapper to the middle section, with the cameras facing forward and try to line them up as best you can (don’t worry if it’s not perfect). Once attached, strap the box containing the PI and Arduino onto the plate on the middle piece and restrain it with Velcro straps. Make them tight so it will not fall off while swimming with the mapper.

**Connecting to the PI:** Once the mapper is setup, you’ll need tell the PI to begin recording data. To do this, you’ll need to access the pi.

\* username: pi, password: esrm1234, name of pi: esrmpi, IP address: 192.168.1.1

The method I have chosen for use in the field is VNC viewer. This allows you to access the PI even if you are in an area without an available monitor + HDMI cable setup, or without access to an Ethernet cable. For the purpose of this project, I will only go over how to connect to the PI using VNC viewer.

\* VNC Viewer setup & connection to PI: First and foremost, I have found these devices to work the best with VNC Viewer and the Ad-Hoc connection I created on the PI; Apple tablets, Ubuntu Linux devices, and Windows 7 (possibly 8) devices. These all have no issue connecting to the Pis Ad-Hoc connection, called “MyNetwork”.

- First, install VNC Viewer on your selected device if you haven’t already.

- Next, you’ll have to connect to “MyNetwork”. This seemed to work fine with Windows 7 and Apple tablets without any prior setup. On Ubuntu and possibly other devices, you will have to set a static IPv4 address, such as 192.168.1.2 (this usually works).

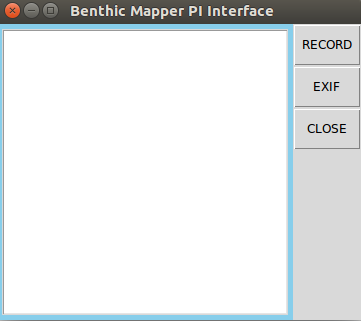
- After successful connection to “MyNetwork”, open VNC Viewer and enter in the address of the PI, 192.168.1.1

- You might be asked to allow this connection. Allow it.

- You will then be asked to enter a username and password. These are; username: pi, pasword: esrm1234

- Once you are connected, you should see a small screen displaying the desktop of the PI and can now remotely access all of its functions.

**Recording Data & Creating EXIF(geo) files:**

****

- To start recording data, open the folder in the desktop named “GUI”. This contains an executable file called gui\_ben\_pi.py Upon opening, it may ask how to be opened. Select an option that allows you to execute in terminal. This will be helpful if you want to see some of the background processes going on in the program. Once the GUI pops up, select the RECORD button. You will be asked to provide the # for the usb port. On the PI, this will usually be 0. Example: /dev/ttyUSB + #, select 0, → /dev/ttyUSB0. You’ll then be asked to enter a time. This is measured in minutes, so a value of 60 entered will be 1 hour. Once time is up, or if needed you can cancel the program (might impact some data), a text(.txt) file will be created with the time and date as the name, along with all relevant data recorded during the run.

- To tie the data from the pictures together with the data from the recording, select the EXIF button. You’ll be asked to enter the file path for the photos. You can get this by right clicking on the folder containing the photos and either “copy path” if that option is available, or, right click, open properties, and copy the location given. After entering in the photo path, you’ll be asked to enter the path to the data you just recorded (i.e. altitude, depth, etc). Same as before, except this time you must enter the path to the file itself, not the folder it’s contained in.

- An example of file paths:

# photo path: /media/alex/RUDOLPH\_USB/BenthicMapper/Santa Cruz

# file path: /home/alex/Desktop/BenthicMapper/PI/2018-05-13 19:29:46.728506.txt

- This will create a new file called newEXIF.txt

- The format of the file is currently: photoname.jpg,-,-,data(depth, altitude…) This is for Pix4D which needs txt files in this format.

- When you are done recording and compiling data, you can either close the program by clicking CLOSE or hitting X in the top right corner.

- Be sure to properly turn off the PI when you are done. Do this through the VNC Viewer connection by other selecting turn off from the menu, or enter *sudo poweroff* in the terminal.