

## Accessing the BeagleBone Black using SSH

1. Connect the BeagleBone Black to your computer with the USB-A (your computer) to Mini-USB (BBB) cable
2. SSH into the BeagleBone Black
  - Windows User
    - i. Install the SSH client Putty on your computer if you don't have it already
    - ii. Open up Putty and enter "192.168.7.2" as the Host Name and 22 as the Port #
    - iii. Select the connection type as "SSH" and click "Open"
    - iv. If you encounter a security alert, click "Yes" and continue (after the server's host key is cached in your registry, you won't have the alert pop up anymore)

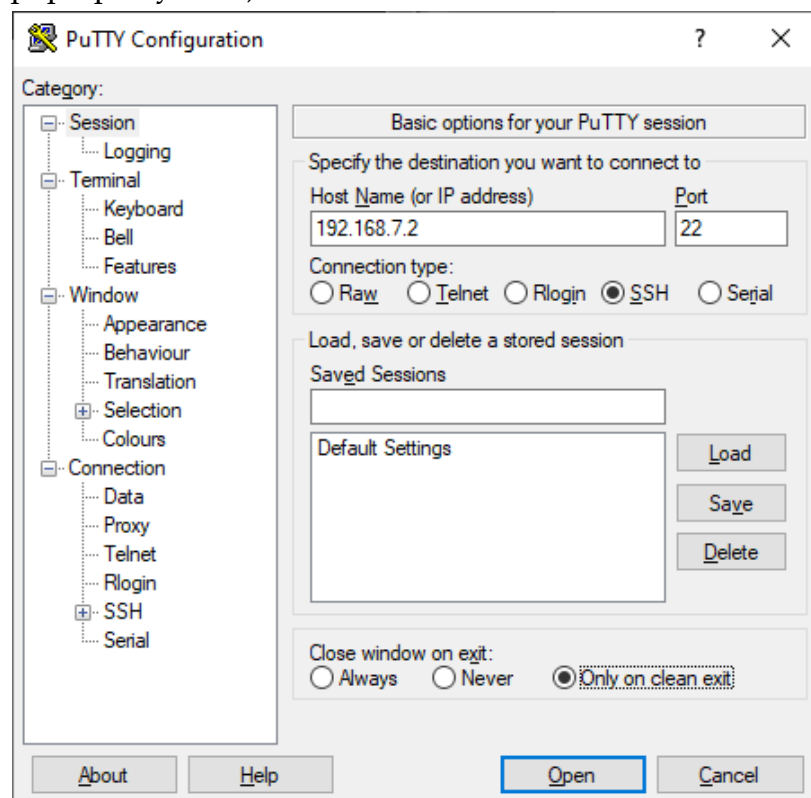


Figure 1: Your configuration should look something like this

- v. Login as: wracksweeper  
Password: wracksweeper
- Linux User
  - i. Open up a terminal window (CTRL + ALT + T)

- ii. Type in the command: “ssh [wracksweeper@192.168.6.2](mailto:wracksweeper@192.168.6.2)”
- iii. Enter the password: wracksweeper

**Note: Once you're in, the remaining steps will be the same for both Windows and Linux users**

## Overview of Everything on the Payload BeagleBone Black

If you do an “ls” to list your current directories contents, you'll notice a directory “wracksweeper” and a file “waves”

- wracksweeper directory – contains the source code, header files, Makefile, and README
- waves script – shell script to run the software

You can find the directory where the SD card is mounted on in “/media/Data”

In order to confirm that the SD card is mounted in that directory, type “df” to show file system disk space usage and look for the filesystem that is around 64GB (SD card storage) and you should be able to see that it's mounted on “media/Data”

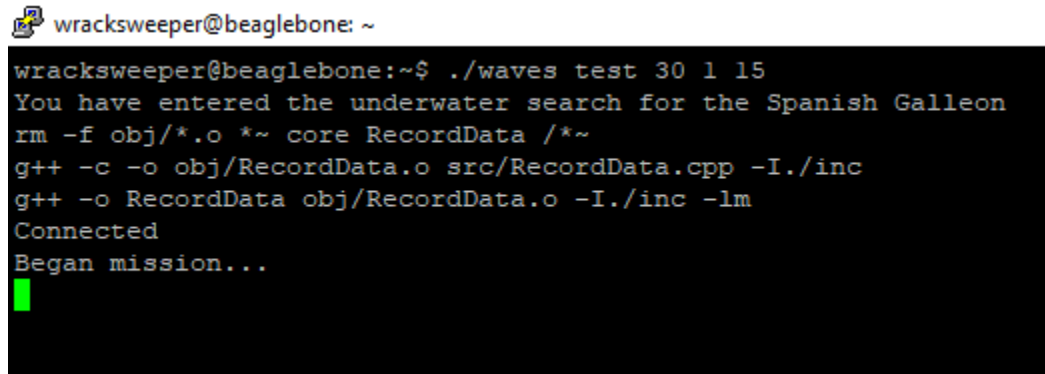
In that directory, there is a folder called “missions”, which is where all of the mission (.872) files will be stored

**WARNING: Make sure that the sonar board is connected to the BeagleBone Black via ethernet and that the transducers are connected to their respective ports on the sonar board, otherwise the program will run, but no mission file will be created.**

## Running the Sonar Software on the Payload BeagleBone Black

Due to only having single line command access for now on, the software is ran through a shell script that takes in arguments for the filename and different parameters

1. Change to the default directory, if you're not already there, by typing “cd”
2. Run the script with this syntax:  
./waves <filename> <range> <frequency> <start gain>



```
wracksweeper@beaglebone: ~  
wracksweeper@beaglebone:~$ ./waves test 30 1 15  
You have entered the underwater search for the Spanish Galleon  
rm -f obj/*.o *~ core RecordData /*~  
g++ -c -o obj/RecordData.o src/RecordData.cpp -I./inc  
g++ -o RecordData obj/RecordData.o -I./inc -lm  
Connected  
Began mission...  
█
```

Figure 2: Example execution of sonar software

3. We currently don't have a clean way of stopping the software, so in order to stop recording, you'll need to perform a CTRL + D to stop running the program
4. Your recorded mission file will be stored on the SD card located in the `"/media/Data/missions"` directory

## Future Improvements

In the GitHub repository, I pushed another folder containing future Linux code that has partially implemented ncurses to use a keypress as a command to stop recording. Using ncurses is not the most ideal way of implementing this, so I suggest using multithreading to have two processes running at the same time. The function `"getch()"` is the easiest to use, but the problem with it is that it waits for a user input, so it's not ideal if we're recording data at the same time while waiting for a user input. That's why we tried ncurses since it allows you to use `"getch()"` without waiting for a user input and still recording data at the same time.

Another minor improvement is when you'll have access to more than just single command line access, prompt the user for inputs on the filename and the parameters.