r-one Python Setup for MacOS

NOTE: In addition to reading this document, look through the Windows Python setup document, because that contains even more useful information.

1 Setup

1. Install Sublime Text 2 or another text editor so that you can edit files: http://www.sublimetext.com/2
2. Install the VCP drivers from FTDI: http://www.ftdichip.com/Drivers/VCP.htm
3. Install Python 2.7: https://www.python.org/downloads/release/python-278/
4. Download `owl-tools` from the ENGI 128 website: https://www.clear.rice.edu/engi128/resources.shtml
5. Put `owl-tools` somewhere you can get to it (like Documents or your home folder)
6. Read this website on how to use the MacOS terminal/command line. http://blog.teamtreehouse.com/introduction-to-the-mac-os-x-command-line
7. Open up the terminal
8. Download the `.tar.gz` version of PySerial 2.7: https://pypi.python.org/pypi/pyserial
9. Uncompress the tar file somewhere you can get to it. You may need a program to help uncompress it, like 7-Zip.
10. Run the command `sudo python setup.py install` from the uncompressed folder in the terminal. This will install PySerial. It may be a good idea to get one of the TA's to check the installation at this point because sometimes things can go wrong.

2 Connecting To Your Robot

1. Open up the terminal
2. In order to connect to the robot, we need to run the python script that does this (it's called mcu.py and it is located in owl-tools). To do this there is a big long command to run. Here is it's structure:

   ```
   python <wherever you put your owl-tools folder>/mcu.py
   --serial=<The serial port the robot is connected on> <The thing you
   want to do on/to the robot>
   ```

   For example, I might do:

   ```
   python  /Documents/owl-tools/mcu.py
   --serial=/dev/tty.usbersial-A0749G ipm
   ```

   Which says:

   - Call the `python` program to interpret the script...

- The script is located in `<Home Directory>/Documents/owl-tools/` and is called `mcu.py`
- Give the script some information of how to connect and what to do with the robot, the serial port that the robot is connected to is called `/dev/tty.usbersial-A0749G`
- Finally, the command we want to run once we connect to the robot is called `ipm` which stands for "Interactive Python Mode" and is a way of talking to the robot that allows us to run small snip-its of code and run python on the robots.

3. **Finding out which serial port your robot is connected to:**

   Serial ports in Mac (and Linux) are found in a special folder called `/dev/`. In order to list all the serial ports that could be, run the command `ls` on that directory (you did read the How-To on the Mac Terminal, right!?). We can do better though, it turns out that robots always show up as `/dev/tty.usbserial-<SERIAL_NUMBER>` on Mac's (but not on Linux), so what we can do to just list the robots is to run the command:
   `ls /dev/tty.usbserial*`
   and it will list all the robots.

   But we can do even better. The terminal has Tab Completion, so what you can do is to start typing:
   `python ~/Documents/owl-tools/mcu.py --serial=/dev/tty.usbserial-`
   then hit Tab, and it will complete to
   `python ~/Documents/owl-tools/mcu.py --serial=/dev/tty.usbserial-A0749G`
   or whatever your robot's serial number is. Much better.

4. Running the interactive python mode (this is the normal, "connected" mode): Once we have typed
   `python ~/Documents/owl-tools/mcu.py --serial=/dev/tty.usbserial-A0749G`
   we just need to give the command at the end of the line. The command for interactive mode is just `ipm` so the full command is
   `python ~/Documents/owl-tools/mcu.py --serial=/dev/tty.usbserial-A0749G ipm`
   then hit enter and we've connected to the robot!

5. All of the interactive mode commands are the same between MacOS and Windows since they come from the robot, not the operating system.

In order to have a file live on the robot semi-permanently, we need to program them onto the robot, rather than just running them.

3 Running or Programming a File onto the Robot from Interactive Python Mode

See the windows instructions for doing these two tasks, taking into account the different file structures between Windows and MacOS and the different way to connect.

4 Programming a File onto the Robot from the Terminal

To do this, all we need to do is to replace the `ipm` command with a list of 1 or more files. For example, say we wanted to program `run_motors.py` on the robot, we might run the command
`python ~/Documents/owl-tools/mcu.py --serial=/dev/tty.usbserial-A0749G run_motors.p`

If we wanted to program multiple programs where `Lab4.py` is the main program and `velocity.py` is a library or other python file, we might run the command:
`python ~/Documents/owl-tools/mcu.py --serial=/dev/tty.usbserial-A0749G Lab4.py velocity.py`
Again, note that all the files have to be in the same folder as you when you run the command or they must be fully qualified with paths.
To run the program:

- Program the robot with the file(s)
- Push the reset button on the robot
- Connect to your robot
- At the `robot>` prompt, type `import mainpythonfile` and press enter. Example: `import PS07` (note that it is just `PS07` not `PS07.py`)
- To access a function from `mainpythonfile`, just use the function as normal (it has been imported into the current global namespace).
  Example: `function_in_mainpythonfile(0.5)`

5 Helpful hints/tips

1. When typing strings in python there is no difference between single quotes and double quotes as long as you remain consistent.
2. When you have an error in a file, the error message will give you a line number to look at in your code.
3. If you want to see what your program is doing, keep in mind that printing out your variables and watching how they change is a good way to understand what is happening in a given function.
4. Comments are your best friend. Commenting out a piece of code instead of deleting it, is often a good idea. Also, having comments explaining what a variable is or what you are trying to do, is good for when you leave your code and come back to it later. You can make a comment in Python by putting a `#` at the beginning of the line. For a section of code, there is a short cut in the menu to comment out a highlighted portion.
5. In the new version of the rone python connect library (i.e. `mcu`), it is not necessary to always type the serial port. If you have only one robot connected to your computer, you can leave out the serial port by just leaving out the `--serial=` option when connecting or programming. For example:
   `python ~/Documents/owl-tools/mcu.py ipm`
   will usually work. This can considerably speed up the connection process if you are trying to program multiple robots with the same program.
6. Files that are to be programmed onto your robot **must** use spaces and NO tabs.