

## r-one Python Setup

### 1 Setup

The files needed are available in the Resources page of <http://www.clear.rice.edu/engi128>. You will need to setup two files: Python 2.7.8 and PySerial

1. Download Python 2.7.8. When the download completes, run it and follow the installation instructions. Do not change the suggested install folder.
2. Download PySerial. Run the file and follow the instructions. It should find your python files.
3. Download the r-one Python software. **This is a zip file and must be unzipped before use.** Save it to the python install directory and unzip(extract) the files. It should create its own directory called "owl-tools". If it extracts and makes a mess, delete the new files, and try again. Selecting the "extract here" option should do the trick. Ask for help if this does not work.
4. Set the PYTHONPATH. On Windows 7/8, open the start menu and type "environment" into the search box. Click on "Edit environment variables for your account" which will bring up a window. Under the list of User variables, select "New" which will bring up another window. For Variable name, type PYTHONPATH . For variable value, put the entire path location to the rone folder containing owlpy.py. Example: C:\Python27\owl-tools. Your location will be different.
5. As with the GUI, you must find the com port number your robot connects to. Go to the start menu and search *device manager*. Open device manager. Plug in your robot and wait for *Ports* to show up. This may take a while if your computer needs to install driver software. When *Ports* shows up, expand the list and you should see the robot with COM# at the end. This is your com port. The com number will change with each different robot you plug into your computer.
6. Start IDLE from the start menu. At the prompt type: `import owlpy`. It should return with another prompt. Now type, `owlpy.connect()`, You should get a python prompt. Type something exciting, like: `1 + 1`. Congratulations, you just ran software on your robot!

### 2 Using IDLE

#### 2.1 Cycling through past entered commands

You can use previous commands from the prompt. You can access the previous command with `Alt + P` and the next one with `Alt + N`.

### 3 Using IDLE with your Robot

#### 3.1 Making sure IDLE can find your files

If you are in COMP 140 or have other versions of python installed on your computer other than Python 2.7.8, you will need to follow the instructions in the **Making IDLE Work With Multiple Version of Python** section below instead of this section. If you only have Python 2.7.8 installed, use the instructions below.

This is for each time you start IDLE. This is not a one time fix for IDLE locating your files. If you do not follow these steps (selecting "Edit with IDLE") then when you try to run a python file

or program a python file onto your robot, IDLE will not be able to find the folder that contains the python files you are looking for.

1. Open the folder with the python files you wish to run or program on the robot.
2. Right-click on a python file that you wish to open and select "Edit with IDLE". This will open the file and an IDLE window (This window will say Python 2.7.8" As the first line in the window, way above the >>> prompt. If it does not say this, see the section below **Making IDLE Work With Multiple Version of Python**).
3. Now IDLE will be able to find the files in that folder to run and program on your robot.

### 3.2 Making IDLE Work With Multiple Version of Python

If you have more than one version of python installed on your computer, you will need to perform some special steps *every time you wish to download or run a program on your robot*.

1. Open IDLE for Python 2.7.8. To do this go to:  
Start->All Programs->Python 2.7->IDLE (Python GUI)
2. Make sure that IDLE for Python 2.7.8 is actually the one running by looking at the top line in the main text box of the window. It should read Python 2.7.8 followed by some other junk. If it does not read this, you have done something wrong.
3. Type:  

```
import os
os.chdir('<FULL PATH TO FOLDER CONTAINING FILE HERE>') where
<FULL PATH TO FOLDER CONTAINING FILE HERE> is the location of the folder where
the file you wish to run is located. Example:
os.chdir('C:\Users\John Doe\Documents\ENGI128 Stuff')
```
4. Run or program the code to the robot as you normally would

### 3.3 Helpful hints/tips

1. In several of these step by step instructions, you need to type `import owlpy`. Each time you open IDLE, this step only needs to be completed once. For example, you do not have to import owlpy after connecting to your robot if you then decide to program your robot.
2. In IDLE, when typing strings there is no difference between single quotes and double quotes as long as you remain consistent.
3. When you have an error in a file, the error message will give you a line number to look at in your code. In IDLE, the line number is stated in the bottom right hand corner of the window (it displays the line number your flashing text cursor is on).
4. 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.
5. 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.
6. In the new version of the rone python connect library (i.e. `import owlpy`), it is not necessary to always type the COM port. If you have only one robot connected to your computer, you can leave out the COM port by typing '\*' instead of the COM port number. For example `owlpy.connect('*'), owlpy.program('*', 'file1.py', 'file2.py')`, and

`owlpy.connect()` will all work. This can considerably speed up the connection process if you are trying to program multiple robots with the same program.

7. Files that are to be programmed onto your robot **must** use spaces and NO tabs.

### 3.4 Connecting to your robot

1. Plug in your robot (with the robot turned on)
2. Type `import owlpy` at the `>>>` python prompt in the IDLE window
3. Type `owlpy.connect("serial port")` at the `>>>` prompt (Example: `owlpy.connect("com5")`, or if you have just one robot connected `owlpy.connect()`)
4. Blue text will appear and the prompt will be `robot>`
5. Type python commands to use the robot
6. This prompt is called Interactive Python Mode (IPM) and is where you will be spending a lot of time with the robots.
7. To close your connection to the interactive prompt on the robot and return to the IDLE `>>>` prompt, type `exit` and press enter. To close IDLE from the `>>>` prompt, press `Ctrl + d`.

### 3.5 Creating a new Python file

1. You will see a menu bar at the top of the IDLE window
2. In the File drop down menu, select `New Window` or press `Ctrl + n` which will open a very similar window that says `Untitled` at the top where `Python shell` was on the last window.
3. Save this file either by `Ctrl + s` or by clicking `save` in the File drop down menu. Be sure to type `".py"` at the end of your file name. If you don't, you'll see your code be all black instead of having syntax highlighting and the file will not be recognized as a python file.
4. Write your python code and save it. Now you can use this file on your robot.
5. Don't use tabs! Only spaces!

### 3.6 Running a file on your robot

- Running a file is a good way to debug. You can make a change in the file, and then simply re-run the file without having to reset the robot or reconnect to your robot.
  - *Be very careful with running files this way however. If a file is too large to fit on the robot, it will produce strange memory errors. If you ever think this is happening, use the program function described in the next section.*
  - If a function call is at the bottom of the file, when running the file, the function will run.
1. Connect to your robot using the steps above.
  2. At the `robot>` prompt type `run pythonfile.py`. Example: `run simon.py` (note that unlike running a programmed file, the `.py` ending is included).
  3. Note that the file that you are trying to run (or program) must be in the same folder that you ran IDLE from (or changed directories to using `os.chdir()`).
  4. To access a function from `pythonfile.py`, type `function(arguments)` at the `robot>` prompt. Example: `function_in_pythonfile()`

### 3.7 Programming your robot while in Interactive Python Mode (IPM)

1. A Single File
  - At the `robot>` prompt, type `program pythonfile.py` and press enter. Example:  
`program simon.py`

- Note: The file that you are trying to program (or run) must be in the same folder that you ran IDLE from (or changed directories to using `os.chdir()`).
  - Another Note: If your file name has spaces in it (shame!), you will need to escape the spaces by placing the file name in double quotes. Example: program "Bad Name.py"
  - Once the command has been entered, the program will be checked and programmed onto the robot. The robot will then be reset, and you will seamlessly re-enter IPM. This process usually takes about 2 seconds, and should not be interrupted.
2. Multiple Files
    - At the robot> prompt, type  
`program mainpythonfile.py pythonfile2.py pythonfile3.py`  
and press enter. Example: program PS07.py velocity.py leds.py
    - Note: `mainpythonfile.py` will be the file run when you press the mode button (red LED button) on the robot.
    - Once again, all the program files will be checked, then programmed onto the robot, which will reset and re-enter IPM.
  3. Running the Program
    - 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)`

### 3.8 Programming your robot from IDLE

1. A Single File
  - Type `owlpy.program("serial port","pythonfile.py")` at the `>>>` prompt.  
Example:  
`owlpy.program("com5","simon.py")`
2. Multiple Files
  - Type `owlpy.program("serial port","file1.py","file2.py",etc.)` at the `>>>` python prompt. Example:  
`owlpy.program("com5","drive.py","controller.py")`  
Note: "file1.py" will be the file run when you press the mode button (red LED button) on the robot.
3. Running 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)`

#### 3.8.1 Another Way to Run Pre-Programed Programs

1. Program the robot with the file(s)
2. Exit IPM if you are in it

3. Push the reset button on the robot **with the robot plugged in.** (Note: We don't know why this is required. It's a bug we are working on. Just roll with it.)
4. Push the mode button (red LED button) to start the program