# What is it? What does it do?

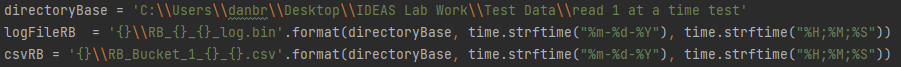
The compost monitor project is intended to track and record data on atmospheric indicators in a controlled environment to determine whether it is possible to make conclusions about the progress of compost with these data.

# How do I use the software?

The software is rather simple. There are three Python scripts – one for each of two Atlas Scientific sensors, and one for the RedBoard (Arduino clone) that powers and connects to the other sensors. These Python scripts are very similar to one another. They will all dump a binary log file and a .csv file every hour labelled for whichever hardware they’re connected to.

## Python Scripts

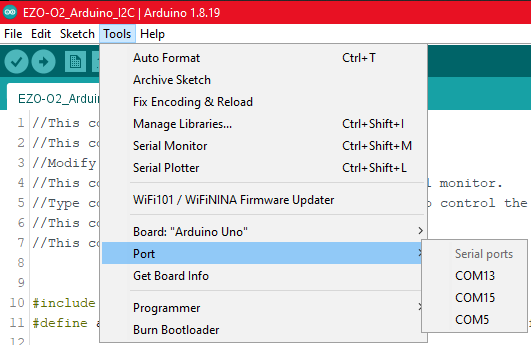
First, locate the sections in the python scripts that have file locations and names. They should look like this:



Make sure that directoryBase will place the files in the correct folder. Then, change anything else you deem fit. The .format method replaces the {} brackets in the strings in the order they appear with the corresponding order of variables in the method. Time.strftime() returns a string containing the date or time pieces that are specified. Keep in mind that using a colon in a Windows filename is not allowed, so the programs use semicolons instead by default.

### Locating Serial Devices

Using the Arduino IDE, it is easy to determine which device is which. Start with no USB devices connected and open the IDE. Then, connect your first sensor or logic board, open the Tools menu in the top left of the window, and scroll over the Port option. If Arduino detects your sensor, it will show up as ‘COM{}’ or ‘/dev/ttyUSB{}’ where {} is a number, depending whether you use Windows or Linux.



After determining the serial port name/number for a device, go back into the Python script for this device and make sure that the port name is set correctly. Here, we have the RedBoard connected as COM5, so the program will reflect this:



With these things adjusted, you should be ready to start logging data! Double-check that you have the correct settings now before causing yourself any unnecessary trouble.

## Plotting Program

The plotting software needs to be pointed to the directories where the data is stored. Once that is done, it is ready to run. **If you open the data in Excel, DO NOT SAVE ANY CHANGES.** Excel will alter the data files, and if the format of the data changes, the plotting program will be unable to import everything properly. For example, a date written to the .csv in MM-DD-YYYY HH:MM:SS format will show up in Excel as “MM/DD/YYYY HH:MM” by default. The raw data in these cells still includes the second data, and opening the file in Notepad shows the original MM-DD-YYY HH:MM:SS format. If you save changes in Excel, however, the .csv will now contain the data as you saw it in Excel. The Python program cannot recognize “MM/DD/YYYY” format and will fail to glean any usable data.