Logging to data.envirodiy.org

I have heard many times that the Aurdino revolution has made electrical gadgets possible for the masses. And it’s (mostly) all open-source so you have a direct window into seeing how it’s all done! While it’s true that you don’t have to be an electrical engineer to make your own electronics widgets, it’s not exactly a simple matter for the masses to use the Arduino spin on C++ code to combine sketches (Arduino jargon for the program you run) to connect to and run more than one gizmo. If DIY environmental monitoring is your goal, you will need to figure out how to conserve energy (e.g. putting the sensors to sleep), save information to an SD card, stream information to an internet portal, and put all of that functionality together into one sketch.

The folks at EnviroDIY have made all of this simpler for you by creating a Modular Sensor library that will allow you conserve energy, log to an SD card, stream to the web (to their free data portal!), and do this with an ever-growing list of research-grade environmental sensors. I’m going to make this my first attempt at providing a tutorial for end-users like me (I’m a scientist with a limited amount of coding background and an even more limited understanding of electronics and electricity).

The first resource that we will use is the Modular Sensors repository at <https://github.com/EnviroDIY/ModularSensors>. Here you will find a growing set of documentation about how the Modular Sensor library works. It’s full of jargon, but has lots of useful information for end-users. It contains a list of the sensors that are currently supported by the library. The documentation for each sensor has useful information including the voltage requirements of the sensor, the communication protocol (I2C, modbus, SDI-12, etc.), and wiring information (the red wire coming out of your sensor isn’t always voltage!).

The Getting Started section of the Modular repository leads you first to the “physical dependencies”. This is all of the hardware, including the box to keep your sensor station waterproof. The simplest starting point is to use the EnviroDIY Mayfly logger, available on Amazon. It’s the multi-tool of data loggers for environmental sensing. The documentation lists other compatible loggers. You will need other hardware, such as a radio module if you plan to stream to the internet, and of course your sensors. The Mayfly has a solar-charging circuit and SD-card reader onboard, which you will have to piece together if you use a different microprocessor/logger. Some students from University of St. Thomas did a post that covers a lot about setting up the hardware for a sensor station. <<<<link to other blog post that covers many aspects of the hardware setup>>>>.

Next you need to have all of the “library dependencies” to run the sketches in the Modular library. My best analogy for new users is that Arduino libraries are like printer drivers; in order to communicate with a printer, your computer needs the correct software to speak with every make and model of printer. If you have the wrong printer driver, sometimes nothing will happen and sometimes the printer will print gobbledygook. In this open-source world you are a DIY-er, so that means you have to make sure you have all of the right libraries to run your sketches.

There are a couple of ways to manage Arduino libraries, depending on the IDE you are using to program your logger. If you are using the Arduino IDE, having lots of dependencies is where the simplicity crumbles. I strongly recommend that as a part of the switch to the Modular Sensor library that you also make the switch to using PlatformIO (<https://platformio.org/>), or another another IDE that allows you to point to repositories for your library dependencies instead of maintaining a downloaded version of them (other IDEs exist, I don’t know them by name). You will have to install PlatformIO for Atom or VSCode text editor (or probably others that I don’t know by name; my screenshots are from Atom). For the sake of brevity, this tutorial will give instructions for using PlatformIO.

If you choose to stick with the Arduino IDE, you will need to install all of the library dependencies independently or use the carefully crafted repository of them here: <https://github.com/EnviroDIY/Libraries>. (NOTE: Trying to manage my libraries manually turned me into a frustrated mess in a hurry, which sent me sprinting to learn how to point to library repositories in PlatformIO.)

I tend to work with others, so I keep my sketches in repositories on GitHub, which allows you to track changes and versions. I manage this through GitHub Desktop, which monitors directories on my computer and keeps track of the changes I’ve made and allows me to “push” them to my GitHub repositories. If you choose this for your workflow, please note that your modular sketches will contain ID numbers for your sensors and your sites and you probably do not want those in a public repository. GitHub has fee-based private repositories, which is what I use for deployment sketches, but there are free private repositories available (Bitbucket and maybe others).

Let’s look at our first sketch: logging\_to\_EnviroDIY.ino. Designate a directory on your computer for this Arduino project (e.g. ~/Arduino/Monitoring). Download the three files contained at (<https://github.com/EnviroDIY/ModularSensors/tree/master/examples/logging_to_EnviroDIY>), namely logging\_to\_EnviroDIY.ino, platformio.ini, and ReadMe.md. If you choose to clone the Modular Sensors repository, I still recommend creating a new project directory because you will not have access to push edits back to GitHub and your personal work will make it messy for you to pull updates from that repository.

The .ino file is the Arduino sketch, and it must be in a folder with the same name, so make a folder and put all three files into that folder (~/Arduino/Monitoring/logging\_to\_EnviroDIY).

You won’t really need the ReadMe.md file, which is a GitHub “markdown” file that is best viewed online, but as a best practice you can add to this Read Me with your own notes and metadata about your project. Things to keep here include sensor addresses (when applicable), my notes on the libraries required for the sketch, UUIDs, the location of the sensor station, etc.

The platformio.ini file is an “initialization” file (I call it the “ini” file, pronounced like the belly button). This file tells PlatformIO the configuration information for your project, which we will set up. I like the practice in the Modular library of exclosing this with each example sketch because it tells you the configuration for that specific sketch. However, this is not the platformio.ini file that will be “running” and uploading your sketches, that .ini file is located in the root directory for your project.

Open Atom and you will get a welcome message for PlatfiormIO if everything is installed correctly. Under the File menu, select Add Project Folder, and navigate to your ~/Arduino/Monitoring/logging\_to\_EnviroDIY directory. You should now see this directory among your lists of projects in the left column of your window. Open all three of the files in your directory by double-clicking on them, and they will appear as tabs in the right column of your window.



Before we start editing the sketch, I want to give you a tour of the things on your screen.