

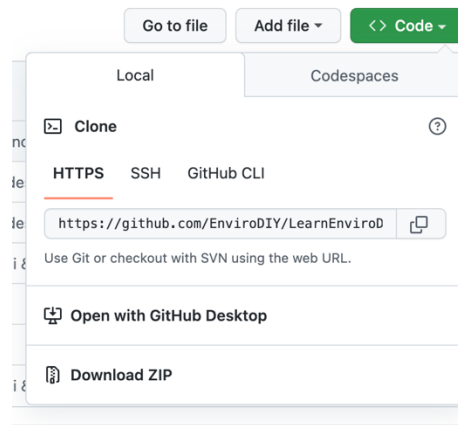
I have been working through the “Learn EnviroDIY Programming” tutorial and, first of all, I really want to thank y’all for putting it together. It is very necessary and helpful. As an end-user that wants to spin up on the modular sensors and Mayfly programming, I think I bring fresh eyes to this. So, in this document, I am going to point out some of the problems that I have encountered, in hopes that they can be addressed.

1. Provide a holistic map of the interactions between your Mayfly, GitHub online, Github Desktop, and Platformio. A good image would be most helpful. In particular, it is not clear how Platformio communicates with Github Desktop or online. Maybe this becomes clear after episode 8, but that is where I’m stuck right now.
2. Install GIT Client: This little bit of text (below) involves considerable work (which you note). However, it is absolutely not clear if Git Client must be running at all times in the background when using Github Desktop and Platformio. More importantly, because Git can be downloaded in so many ways, it is very hard to know what is the best option. For example, Git comes with XCode for mac. Does this mean that XCode must be running when I’m working on coding in Platformio if I want to commit edits to GitHub? Of is it actually working behind the scenes at all times, now that I’ve installed XCode? Is there a command line statement that I can run to make sure the Git is running? Providing this would be helpful.

- [Install Git Client](#)

- You will need this when you use PlatformIO in the next episode.
- Follow the instructions for your operating system. Note that these installations will require multiple steps, and will require you to do some command line installations (e.g. in Terminal in MacOS and PowerShell in Windows). This series of installations may take a long time, depending on your computer and internet connections.
- You will need to have administrator access on your computer to complete these installations.

3. Episode 4. What can you see in Platformio? So you have installed Git client (see above) and Platformio for VSCode. This section needs some basic explanation as to how, why or if Platformio will communicate with Github (online or desktop) other than the fact that Platformio needs Git to access libraries. Is it possible to make Git commits from Platformio? I think so, but I’m wondering how to do it and how/if Github and Platformio talk to one another.
4. Episode 5: Using Platformio. Creating a project of episode 5 sketches has you download the .zip file, move it into the ‘Arduino’ subfolder, extract it, and “Import Arduino Project”. In the screenshot below, from Github.com, if you click on the green “Code” button it appears that there are three ways to Clone a repo: Https, open with desktop, or download a .zip file. I think you try to explain the differences between these, but from the Github site it seems like all three of these will lead to a clone. However, it appears that only the first two will allow you to make a clone on your Github desktop or Platformio and have it update when modifications to code occurs. In my view, this method of cloning a repo should not be used because it just causes confusion. Or you should make it much more clear that downloading a .zip file



essentially breaks the connection between GitHub and the repo that you just downloaded. I think it would be better here if you showed how to bring the repo into both Github desktop and then connect it to Platformio, so that your readers would see how they connect more clearly.

5. Episode 6: Using PlatformIO Serial Monitor and Sensors. This episode was great. No problems!
6. Episode 8: Nicely done. Well explained.
7. Episode 9: Where it went off the rails for me.
  - a. “Clone the ModularSensors repository. Please explain why you have us open it in the Desktop version of Github. Also, it would be very appropriate to have us also get ModularSensors into Platformio at this point and maybe see how modifications to it might be passed back up the tree from Platformio to Github desktop and up to Github.com.
  - b. “Managing your sketches in our own Deployments repo.” Big problem. If you create a new repository on Github and clone it to Github Desktop, that’s all fine. Then you state “Clone your *deployments* repo (as we did earlier in this episode) and make it a project in PlatformIO (as in Episode 5).” However, if you follow the directions in episode 5 (by downloading the .zip file and selecting “Import Arduino Project), you will get an error because the Deployments folder is not an Arduino project, You must first add a file, Deployments.ino, into the Deployments project folder. Then it will be recognized as an Arduino project. Furthermore, when I do thi and I import the Arduino project, it will not name it “Deployments”, but gives it another name like: *23026-1900947-mayfly*. I had to go out of Platformio, rename the folder and then open it as a project to get it to show as “Deployments.”
  - c. Also in this folder, it seems like you would like us to move the files around (copying and such) using the Finder (mac) or FileExplorer (Windows), *not* Platformio. I base this on this text:

*Copy the simple\_logging sketch from ~/ModularSensors/examples/ by right-clicking on the simple\_logging folder. Paste this folder into the tutorials folder in your repository (~/Arduino/deployments/tutorials/).  
Now you can edit this sketch and save your edits to you own repository.*

Could you make it clear whether you want me to do this outside of Platformio or inside?
8. “Episode 10, Collect data from the Mayfly data logger’s built-in sensors.”
  - a. Can you clearly explain why the .ino sketch has #include statements with libraries that the platformio.ini file lacks? I understand that platformio.ini is the boss here. Should we include Arduino.h in the .ini file?
  - b. Also, when I tried to run this (aside from none of the libraries being found), I found that I could not specify the serial port and that I don’t know how that is done using platformio.ini, so my sketches wouldn’t load.
  - c. I am wondering, also, if the libraries couldn’t be found because of my “Git” install in ‘2’ above.