Guide to GitHub (Desktop Version) for MIL Electrical

# Introduction

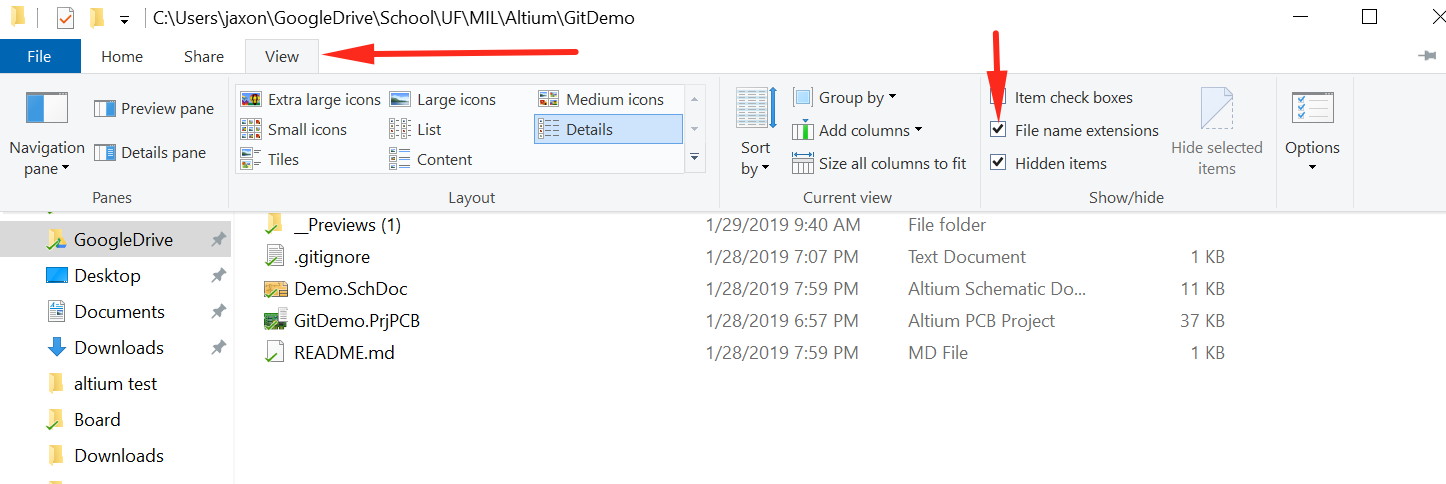
GitHub is an online platform for *version control*. This guide will explain one way to use GitHub that has been tested to work smoothly with Altium Designer.

# Step 1: Installation

While GitHub is a website, it is intended to be used in conjunction with software for your computer called Git. Follow these instructions to install Git.

Visit <https://desktop.github.com/> and download the installer. Run the installer. Leave everything as default.

Note: It will be useful to show file extensions in Windows. You can find the setting and enable this feature:



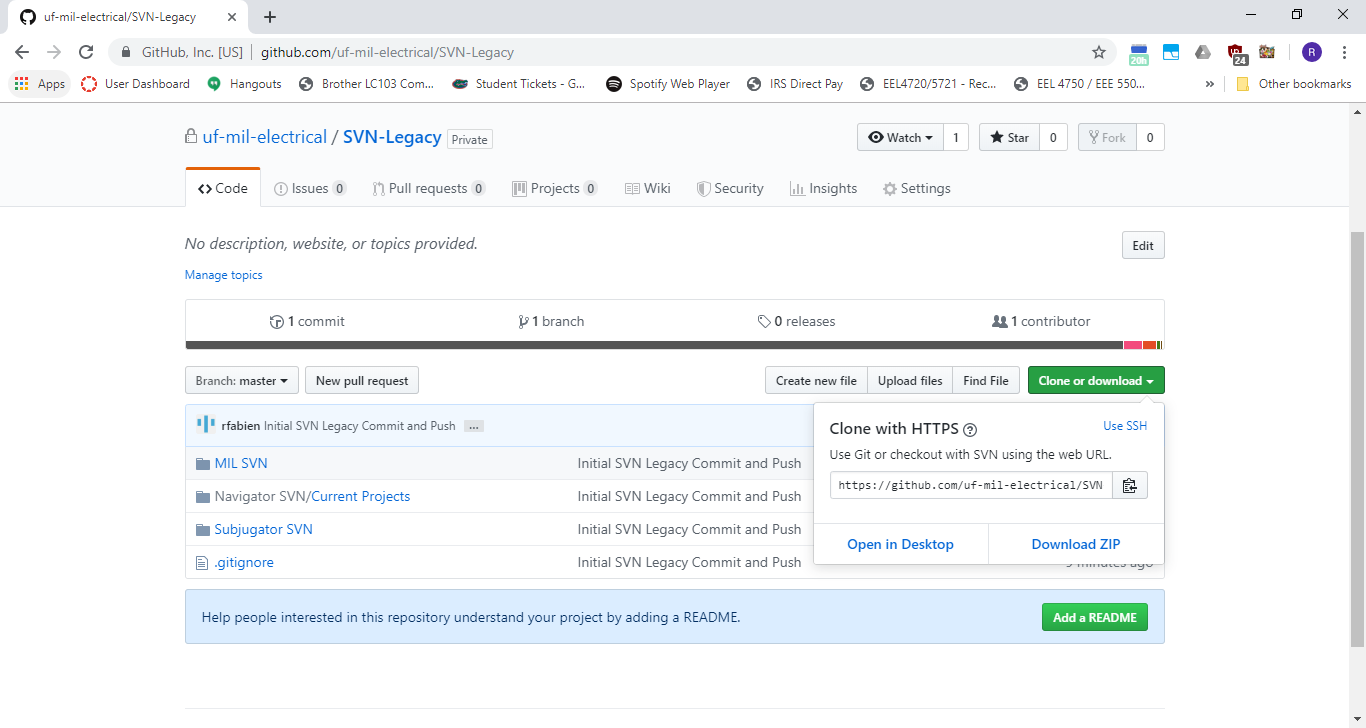
# 

# 

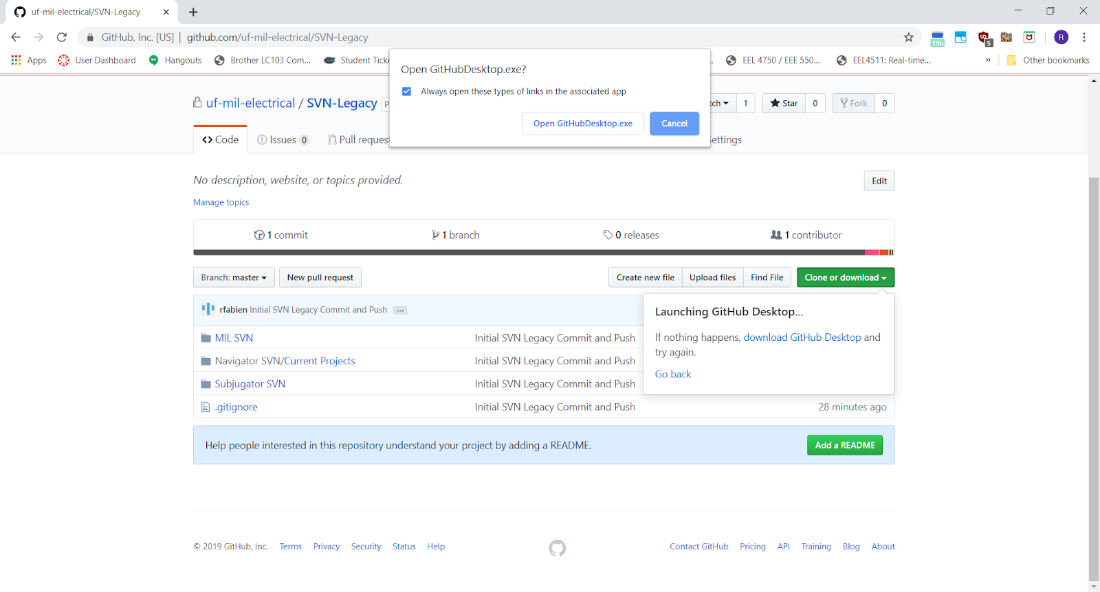
# Step 2) SVN Legacy Clone

Before you can start your own project you need to first obtain some files that may be useful. Namely, the PCB design libraries are located in a special repository named “SVN Legacy”. This step will show you how to set those files up in your work environment.

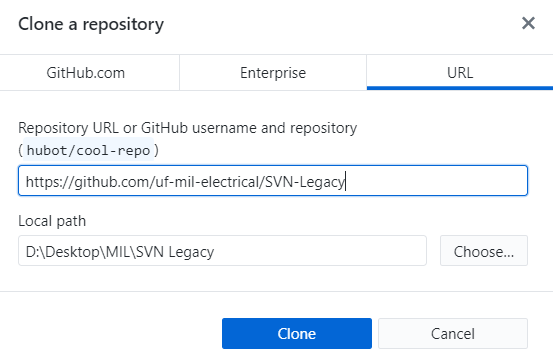
1. Create a folder named “MIL” on your PC
2. Create a subfolder named “SVN Legacy” under the “MIL” folder
3. Clone the SVN Legacy Repository to your local “SVN Legacy” folder
   1. Go to SVN Legacy Repo Site: https://github.com/uf-mil-electrical/SVN-Legacy
   2. Click the “Clone or download” drop down button and then click “Open in Desktop”



* 1. Click the “Open GitHubDesktop.exe” button and click the check box as well so this process does not have to be repeated in the future.



* 1. The GitHub Desktop app should now pop up with a “Clone a repository” window. Change the local path to the “SVN Legacy” subfolder and then click clone.



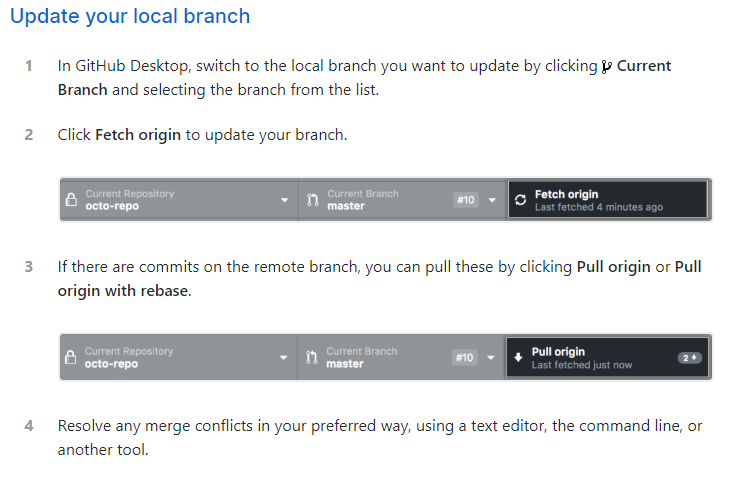


This may take some time the first time. This is normal

# Step 3) Clone a Project Repo

1. Create another subfolder in the “MIL” folder for your project. Name it something appropriate for your project.
2. Have an experienced member with admin privileges create a new project repo for you (from the template repo). If you’re not sure who that is, ask on slack.
3. Go the new project repo site. Ask the experienced member for the site.
4. Clone the new repo into your project subfolder. The cloning process should be the same as the SVN Legacy Clone step, but a different repo git site and appropriate local path.

# Step 4) Check the Repo for Updates

1. It is important that before doing any major work in any version control repo, that you **check for any updates that users might have made**. If you do not, then more extensive effort needs to merge the changes. In some extreme cases, the repository may even be broken and must be updated to an earlier revision and current work may be lost.
2. From the GitHub Syncing your branch Tutorial…
3. 
4. Because this is your first time working with the repo, there should be no changes to pull into your repository.

# Step 5) Using project Repo

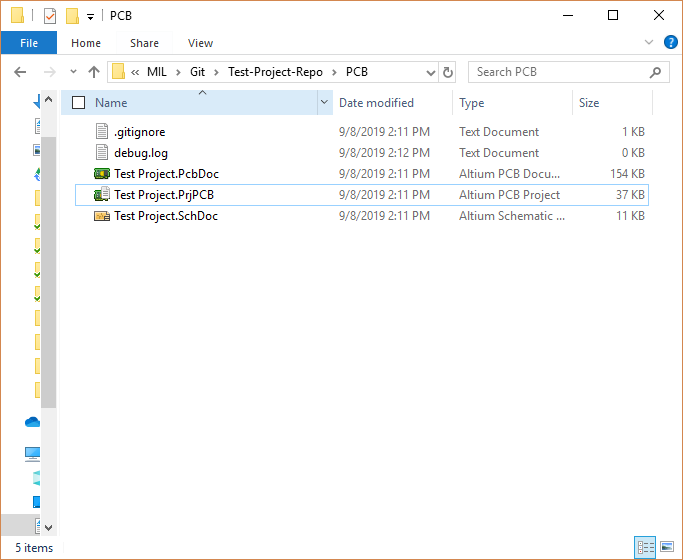
README

1. Populate the README file with a short description of your project and anything else necessary to use your repo as you go along. I recommend downloading and using Notepad++ or, whatever your favorite text editor is, to do this.

PCB

(Note: If you are familiar with setting up an Altium project, feel free to delete the current Altium Files in the directory and populate it with your own files)

1. Rename files in PCB folder with appropriate project name.



1. Double click the .PrjPCB file to use Altium with your new project file. Ignore any errors about files not found in Altium.
2. Relink the renamed schematic and pcb files to your project.
   1. Right click project in Altium
   2. Click “Add Existing to Project…”
   3. Select schematic and pcb file
   4. Click open.
3. Save your project.

FIRMWARE

1. If you have a ccs project at this point, feel free to add it to the firmware folder. Otherwise, you can leave it blank for now.

# Step 6) Pushing your changes

The GitHub site does a good job of explaining how to push your changes so I will just reference that here. Note: unless specified you should be working in the master branch.

Link: <https://help.github.com/en/desktop/contributing-to-projects/committing-and-reviewing-changes-to-your-project>

If the link does not work anymore, please reference the GitHelp directory of your project for some info.

# Step 7) Continual work

You should be all set now. The workflow with version control is to update -> make changes -> commit changes. If you follow that. There should be no issues. Please ask for help when needed.

For more advanced GitHelp please check this link: <https://help.github.com/en/desktop>

Note: If you used a practice repo for cloning, let an experienced member know when you are done so that the repo can be deleted or repurposed for another team member later.