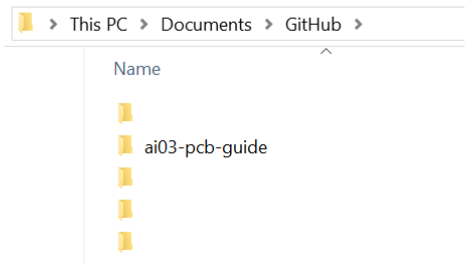
PCB Guide Part 2 - Beginning the project

By this point, you should have a repository readied for the project.

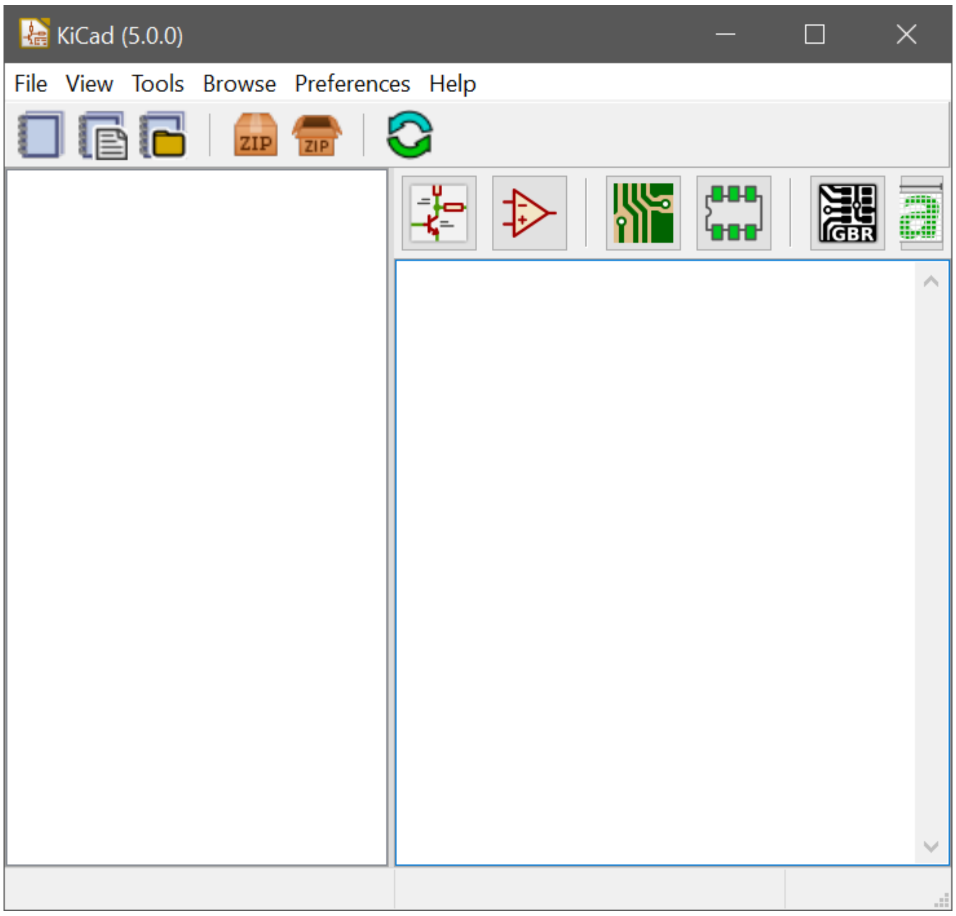
Step 3. Creating the KiCad project

The previous page's setup should have created a folder in your home/Documents/GitHub folder with the project name.

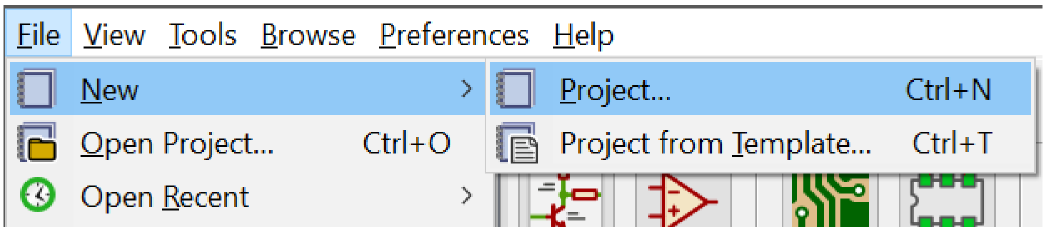


We will be doing all work here.

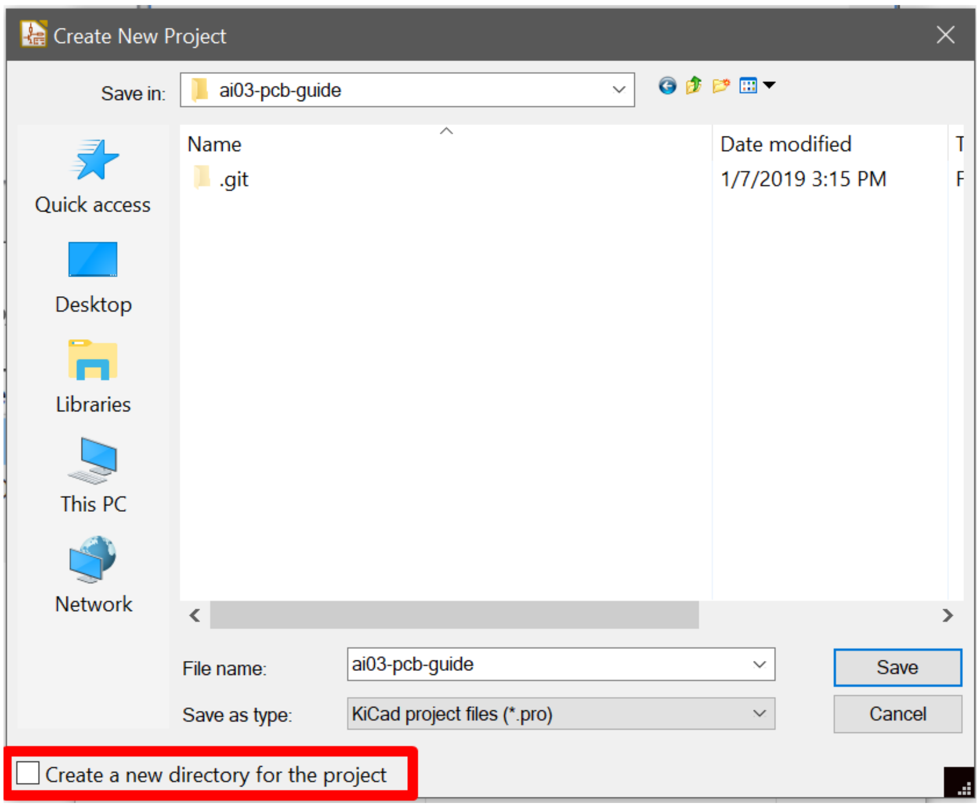
Launch KiCad, and you will be greeted with the main menu.



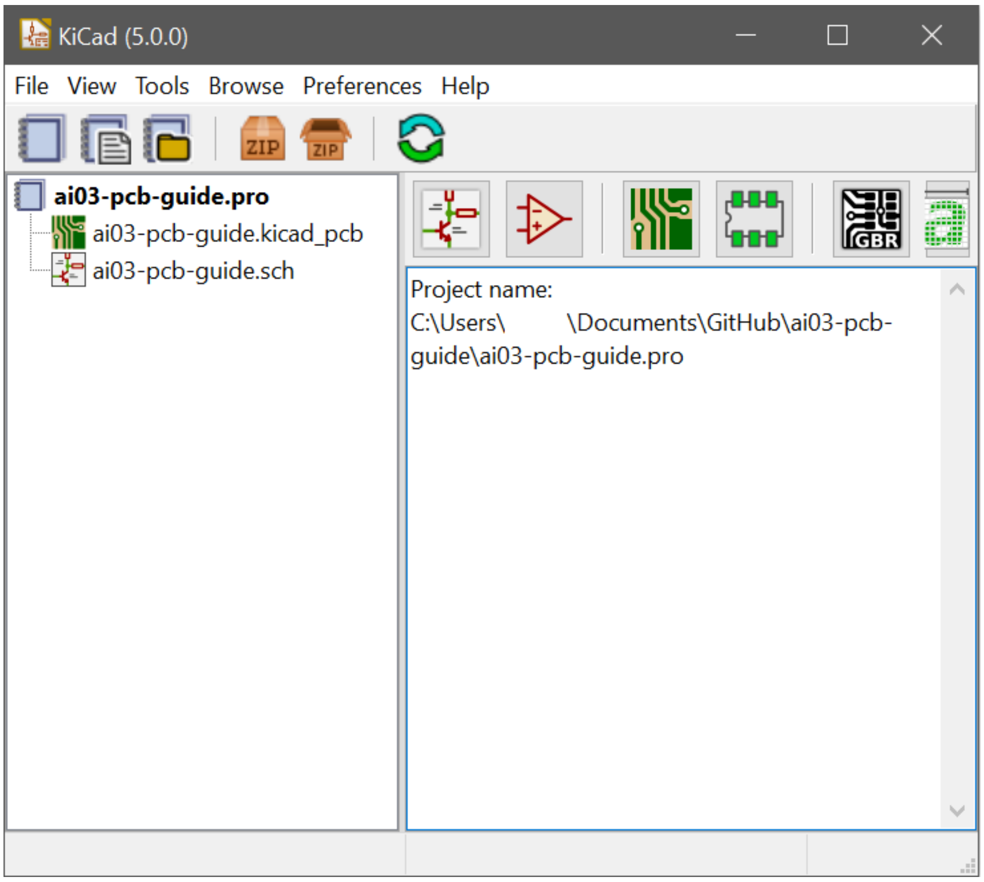
From the options, select New Project.


File View Tools Browse Preferences Help
New
Open Project... Ctrl+O
Open Recent
Project... Ctrl+N
Project from Template... Ctrl+T

Browse to your repository directory.
Make sure to uncheck "Create a new directory for the project". Otherwise, you will have your project in a useless subdirectory.



KiCad will generate the basic project files.



Project name:
C:\Users\ \Documents\GitHub\ai03-pcb-guide\ai03-pcb-guide.pro

The default project only has two files:

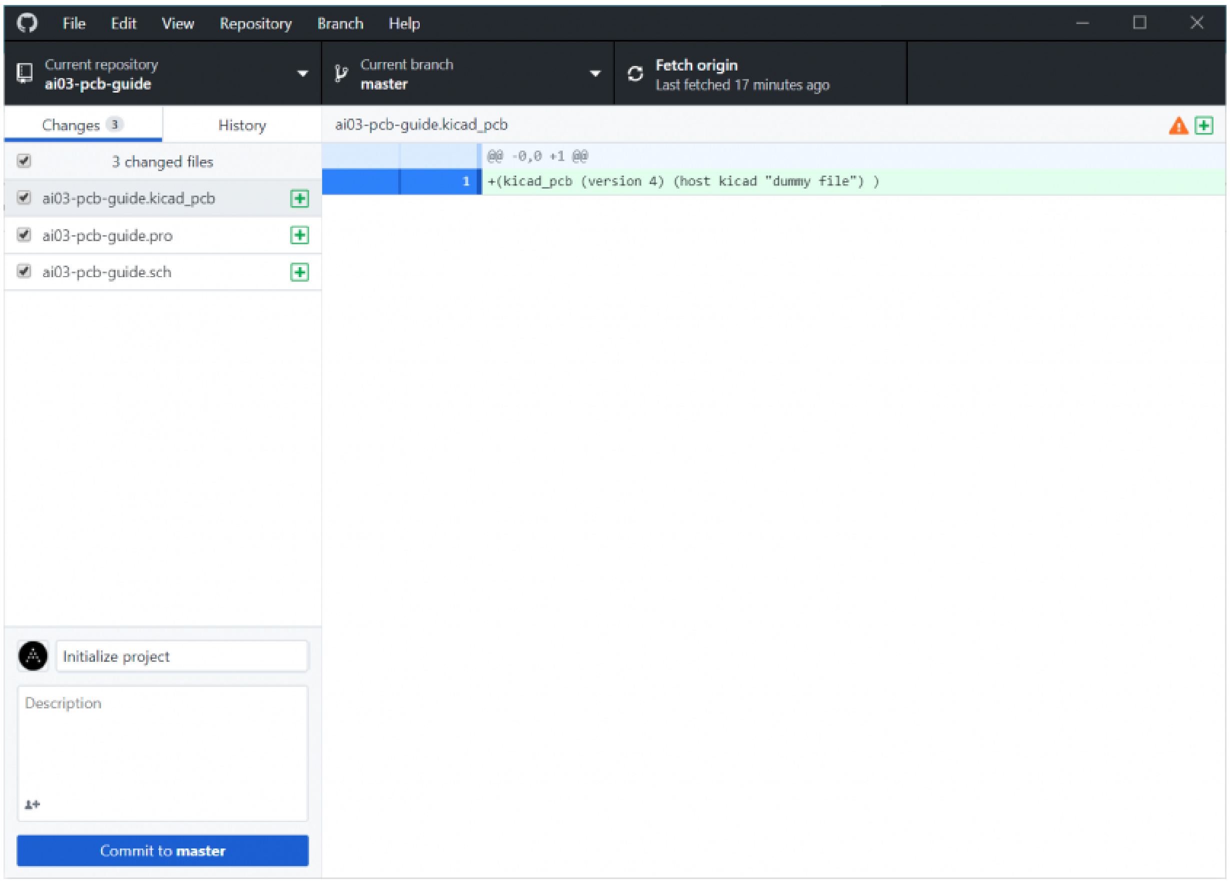
- The .sch file - Contains the schematic, or the electrical layout. This is where you will add the parts and wire them together, so KiCad knows what you're trying to accomplish.

- The .kicad_pcb file - Contains the physical layout. This is where you take the data from the schematic, and position and connect them all together for production.

The basic workflow is to create the schematic first so that the electric side is all set, then to lay it out on a physical PCB based on that.

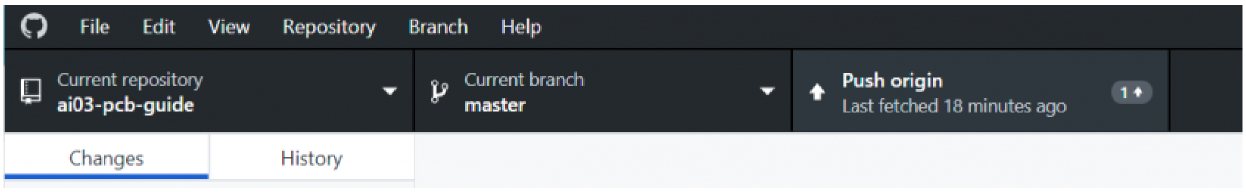
At this point, let's commit our changes.
Give the commit a name, commit, and push the changes.

This guide will occasionally remind you to commit after large milestones; however, it is recommended to do so often. Losing work is not an enjoyable experience.



ai03-pcb-guide.kicad_pcb

```
@@ -0,0 +1 @@
+(kicad_pcb (version 4) (host kicad "dummy file") )
```



Step 4. Add local libraries

KiCad is built around libraries - A collection of footprints, and often schematic components bundled with it. Footprints are the building blocks of PCBs - The sets of copper pads that the components attach to.

KiCad ships by default with support for common electrical components, such as resistors, capacitors, microcontrollers, etc.

It does **not** ship with exotic things, such as Cherry MX footprints, very specific USB connectors, etc.

Let's take care of that.

KiCad supports two library install types:

- Global - The library is added to the computer. This means you only have to install the library once. However, this has the massive downside of requiring that the library be installed to view the project on another computer. This can be a hassle, especially if having the project checked by others
- Local, or Project-Specific - The library is added only for the specific project. This means you will have to install the library each time; however, the project can be shared instantly without any worry about compatibility or missing libraries.

We will be using project-specific libraries.

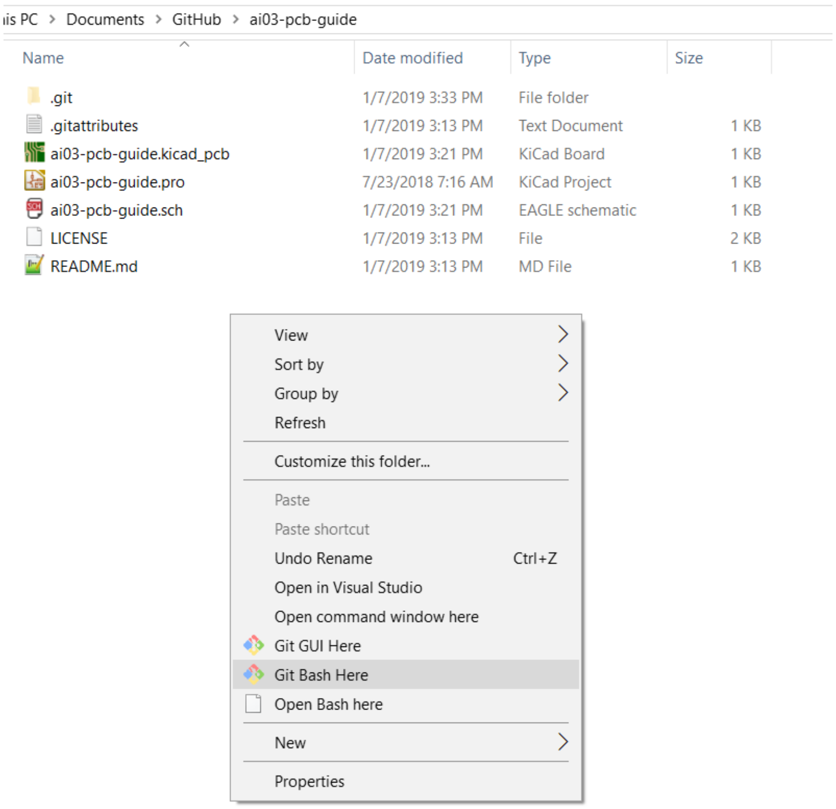
First, we will need to pick which footprint libraries to use.
In this guide, I will be making use of the following repositories:

- My MX-Alps hybrid footprint library - For switches.
  https://github.com/ai03-2725/MX_Alps_Hybrid.pretty
- My random keyboard parts library - Mainly for the USB connector.
  https://github.com/ai03-2725/random-keyboard-parts.pretty

We will be adding these to our project as Git Submodules - This allows us to pull new changes from them without having to manually download and overwrite the existing files.

Since the GitHub desktop application is fairly awful at managing submodules, we will opt for using the shell instead.

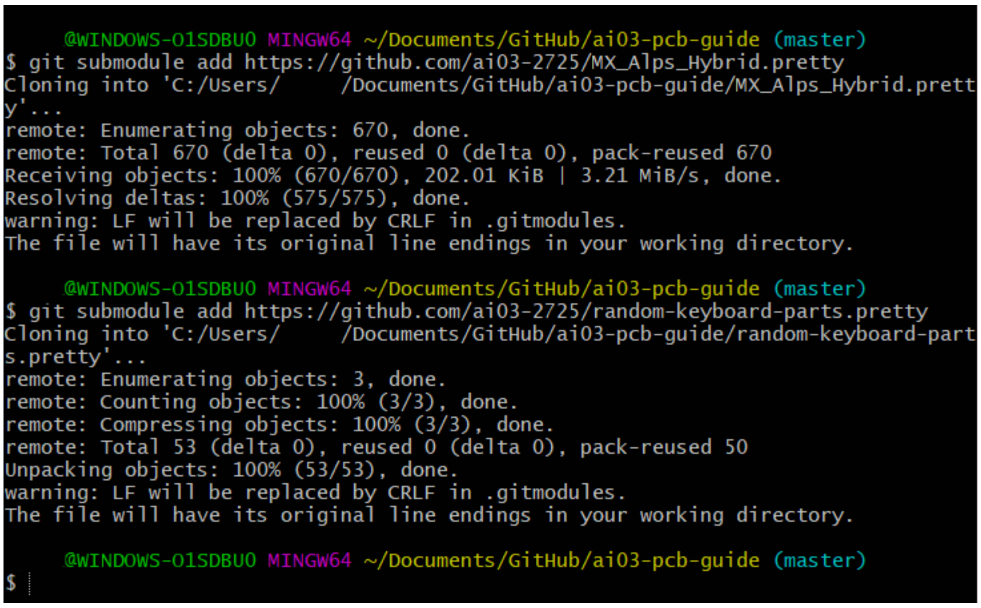
First, navigate to your repository folder, right click in an empty area, and bring up the git shell.


is PC > Documents > GitHub > ai03-pcb-guide

Name	Date modified	Type	Size
.git	1/7/2019 3:33 PM	File folder	
.gitattributes	1/7/2019 3:13 PM	Text Document	1 KB
ai03-pcb-guide.kicad_pcb	1/7/2019 3:21 PM	KiCad Board	1 KB
ai03-pcb-guide.pro	7/23/2018 7:16 AM	KiCad Project	1 KB
ai03-pcb-guide.sch	1/7/2019 3:21 PM	EAGLE schematic	1 KB
LICENSE	1/7/2019 3:13 PM	File	2 KB
README.md	1/7/2019 3:13 PM	MD File	1 KB

Then, run the following command:
git submodule add *URL-OF-REPO*

In this case, I will add the two repositories linked earlier.

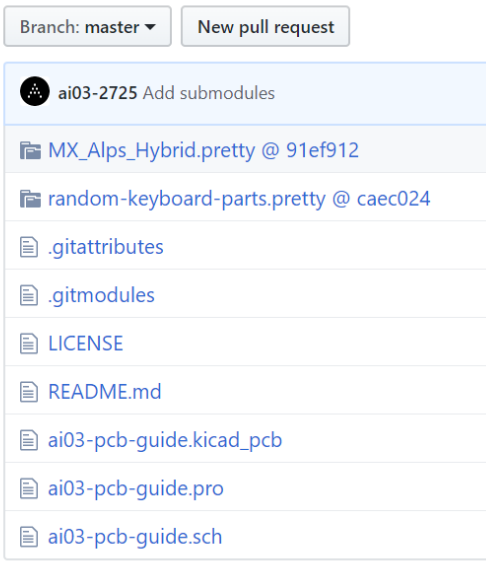

```
    @WINDOWS-O1SDBU0 MINGW64 ~/Documents/GitHub/ai03-pcb-guide (master)
$ git submodule add https://github.com/ai03-2725/MX_Alps_Hybrid.pretty
Cloning into 'C:/Users/     /Documents/GitHub/ai03-pcb-guide/MX_Alps_Hybrid.prett
y'...
remote: Enumerating objects: 670, done.
remote: Total 670 (delta 0), reused 0 (delta 0), pack-reused 670
Receiving objects: 100% (670/670), 202.01 KiB | 3.21 MiB/s, done.
Resolving deltas: 100% (575/575), done.
warning: LF will be replaced by CRLF in .gitmodules.
The file will have its original line endings in your working directory.

    @WINDOWS-O1SDBU0 MINGW64 ~/Documents/GitHub/ai03-pcb-guide (master)
$ git submodule add https://github.com/ai03-2725/random-keyboard-parts.pretty
Cloning into 'C:/Users/     /Documents/GitHub/ai03-pcb-guide/random-keyboard-part
s.pretty'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 53 (delta 0), reused 0 (delta 0), pack-reused 50
Unpacking objects: 100% (53/53), done.
warning: LF will be replaced by CRLF in .gitmodules.
The file will have its original line endings in your working directory.

    @WINDOWS-O1SDBU0 MINGW64 ~/Documents/GitHub/ai03-pcb-guide (master)
$
```

The GitHub desktop GUI seems to have issues with committing submodules, so let's do it from the command line:

git add *
git commit -m "Add submodules"
git push

The commands adds the new submodules to the commit, creates a commit with the given name, and pushes it to GitHub.

After the push finishes, you can see that the library repositories are linked in your GitHub repository.

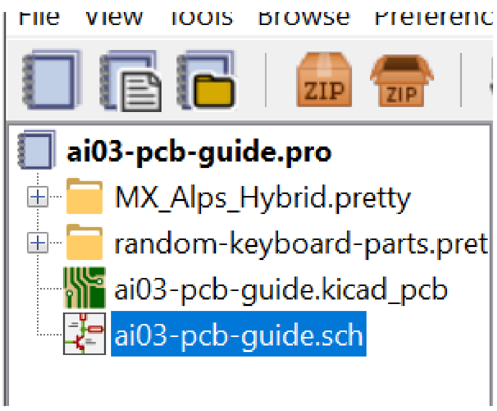


ai03-2725 Add submodules

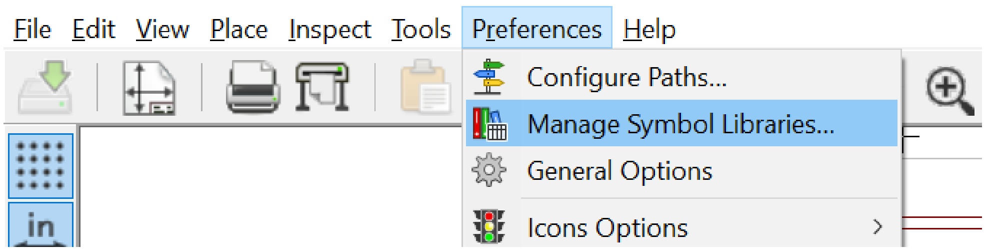
MX_Alps_Hybrid.pretty @ 91ef912
random-keyboard-parts.pretty @ caec024
.gitattributes
.gitmodules
LICENSE
README.md
ai03-pcb-guide.kicad_pcb
ai03-pcb-guide.pro
ai03-pcb-guide.sch

Now that the files are there, let's add them to the project.

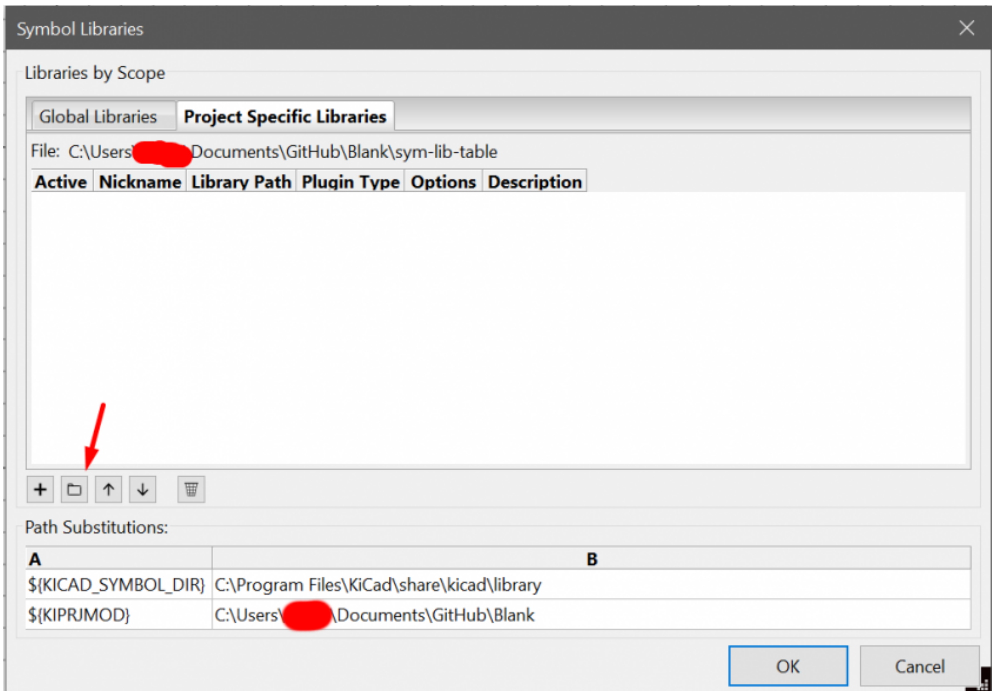
First, open up your schematic file,



Select Preferences -> Manage Symbol Libraries,



Select the **Project Specific Libraries** tab, select Browse Libraries (Folder icon),



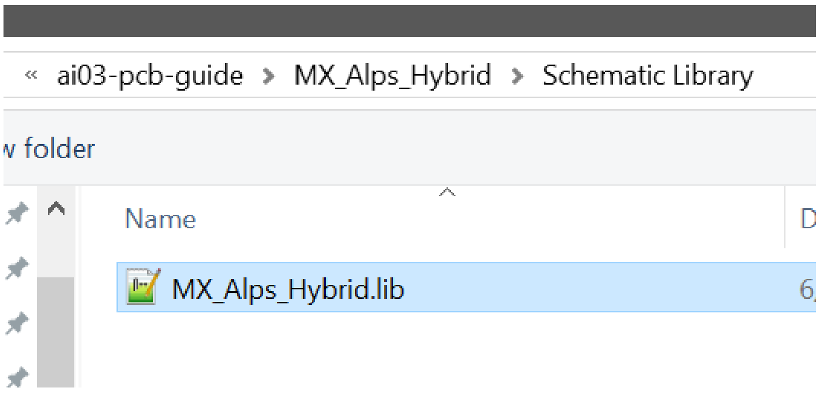
File: C:\Users[redacted]Documents\GitHub\Blank\sym-lib-table

Active	Nickname	Library Path	Plugin Type	Options	Description

Path Substitutions:

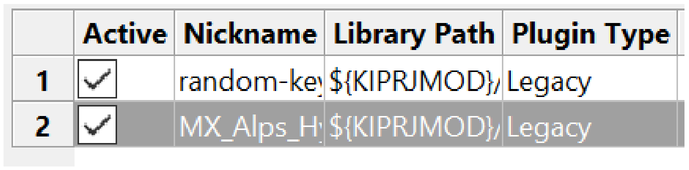
A	B
${KICAD_SYMBOL_DIR}	C:\Program Files\KiCad\share\kicad\library
${KIPRJMOD}	C:\Users\[redacted]\Documents\GitHub\Blank

And select the .lib files in each of the added libraries.


« ai03-pcb-guide › MX_Alps_Hybrid › Schematic Library

MX_Alps_Hybrid.lib

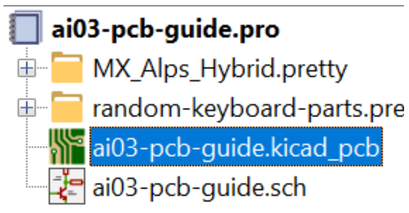
After checking that both are added, confirm the changes by pressing OK.



	Active	Nickname	Library Path	Plugin Type
1	✓	random-ke	${KIPRJMOD}/	Legacy
2	✓	MX_Alps_H	${KIPRJMOD}/	Legacy

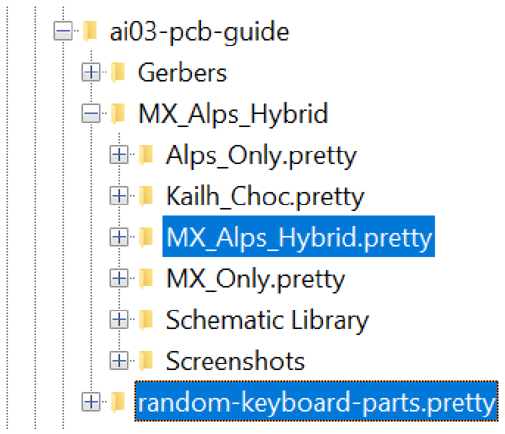
Close the schematic editor for now.

Open the PCB file,



```
ai03-pcb-guide.pro
  MX_Alps_Hybrid.pretty
  random-keyboard-parts.pre
  ai03-pcb-guide.kicad_pcb
  ai03-pcb-guide.sch
```

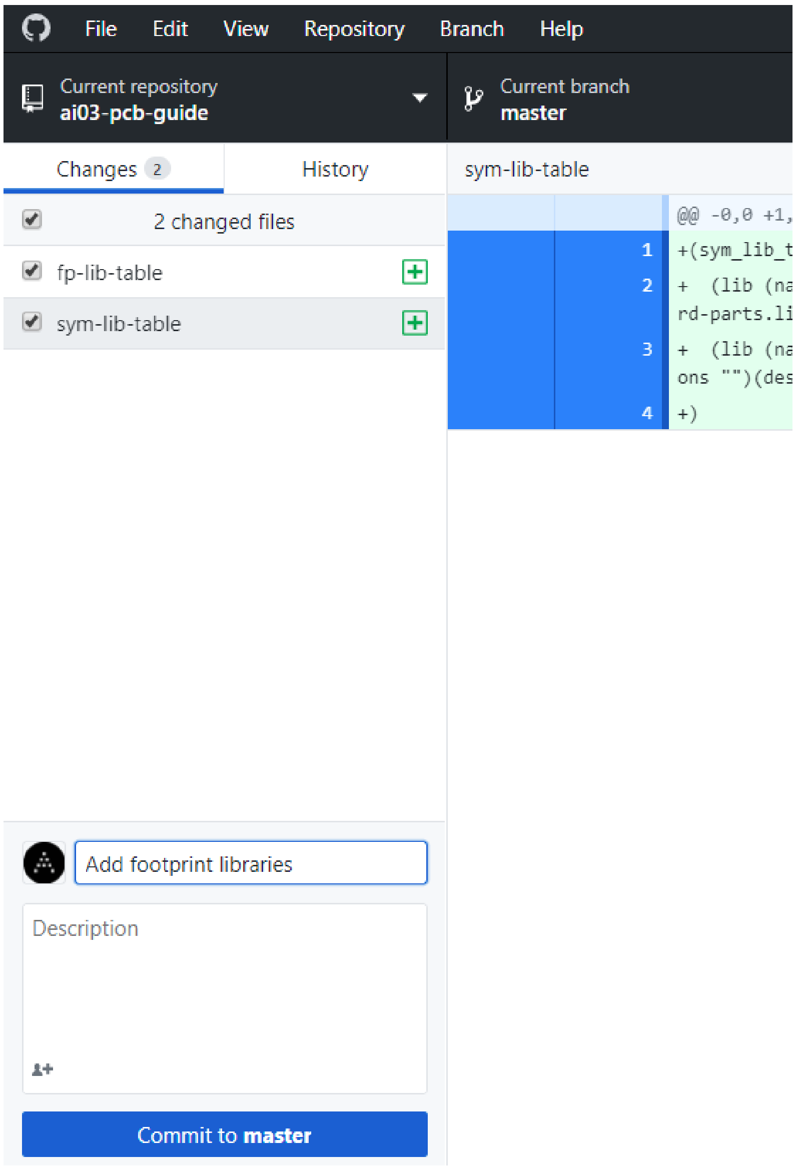
Go to Preferences -> Manage Footprint Libraries, switch to **Project Specific Libraries**, and browse for the added .pretty folders.
For this guide, I will be using the MX_Alps_Hybrid style footprints.



```
ai03-pcb-guide
  Gerbers
  MX_Alps_Hybrid
    Alps_Only.pretty
    Kailh_Choc.pretty
    MX_Alps_Hybrid.pretty
    MX_Only.pretty
    Schematic Library
    Screenshots
  random-keyboard-parts.pretty
```

Again, click OK to confirm changes, then close the PCB editor.

Commit these changes, and push the commit.
If both the fp- and sym-lib-table files are not showing up, make sure you hit OK when adding the libraries rather than simply closing the dialog.



```
@@ -0,0 +1,
+(sym_lib_t
+  (lib (na
rd-parts.li
+  (lib (na
ons "")(des
+)
```

Now you're all prepared for making your first keyboard!
Time to begin the schematic.

Revision #25
Created 7 January 2019 23:16:23 by ai03
Updated 10 December 2020 09:10:07 by ai03