PCB Design Workshop - IEEE Git and Github Primer

Remy Ren

March 2023

Optional: Git/Github Primer (15-20 Mins.)

Git/Github is a great solution for keeping track of files (version history), storing files, and collaboration.

If enough people want, I'll go over a really basic primer on Git/Github using a command line interface (Terminal, Windows PowerShell, MobaXterm, etc.).

SHOUTOUTS TO TOM LYONS YO.

1. Create a Github account if you haven't already.
2. Create a new repository and name it. All other choices are optional. You should be on the *Quick Setup* page. Keep this tab in the background!
3. Open your command line interface (CLI) of choice. To access your repository, we will be using SSH keys as our method of choice. Don't worry! Copy/paste is your best friend and just be careful!
4. The setup is the same either locally or in your Halligan login. For today, as we'll be working on files locally, doing this locally might be best.
5. Enter the command: `ssh-keygen`
6. You should be prompted to "`Enter file in which to save the key (some address)`" Keep track of this address!
   (a) This directory may be hidden depending on your CLI. To view them later, you can enter this: `ls -la1`
7. Don't type anything and hit enter for both the previous "file" prompt.
8. Again, don't type anything and hit enter for both "passphrase" prompts.
9. Navigate to the address the key is stored at (check your CLI output at step 6).

10. Enter the command: `cat id_rsa.pub`
    Copy the entire text from `ssh-rsa.pub` to your computer name. This is your public key!

11. In Github, navigate to *Settings* → *SSH and GPG keys*.

12. Click *New SSH Key*, give it a relevant title ("laptop", "Halligan"), paste your saved RSA public key and add it.

13. In Github, go back to your new repository. Click on the green *Code* button, click on *SSH* and copy the SSH link.

14. In your CLI, find or create a directory you want to save your project in.

    (a) I recommend: Documents → KiCad, then making a new folder called "Tufts Projects".

15. In this directory, e.g. "Tufts Projects", use the following command with your copied SSH link for the underscore:

    ```
    git clone _
    ```

    For example, with an SSH link: `git@github.com:remren/IEEETuftsPCBWorkshop.git`, in the CLI enter: `git clone git@github.com:remren/IEEETuftsPCBWorkshop.git`

16. Any time you want to check the status of what Git is doing, use the following:

    ```
    git status
    ```

17. Once you've made changes to the directory you're working in, you want to use the command:

    ```
    git add .
    ```

    This adds all the files in the directory to a branch. The "." is to state all files in the folder.

18. If you want to commit your changes to the branch, use the command:

    ```
    git commit -m "type your commit message"
    ```

19. Once you feel your commited changes are all set, to upload the changes to the repository in Github, use the command:

    ```
    git push
    ```

20. Now you're all set to use Git and Github! Have fun and enjoy!

21. ADDITIONAL: If your local version is behind first check its status using "`git status`". To update your local version to the version in the repository, use:

```
git pull
```

22. OPTIONAL: To add a submodule (some other repository) from Github into your repository/directory, copy the *HTTPS* link from the desired Github repository, then do the following:

```
git submodule add https://github.com/project/project.git
```

23. OPTIONAL: If you used `git pull`, but you've included submodules, they might have not updated their contents. To force the update, use: `git submodule update --init --recursive`