HyTech Racing Electrical Software Guide

GitHub Desktop

From Wikipedia, Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people.

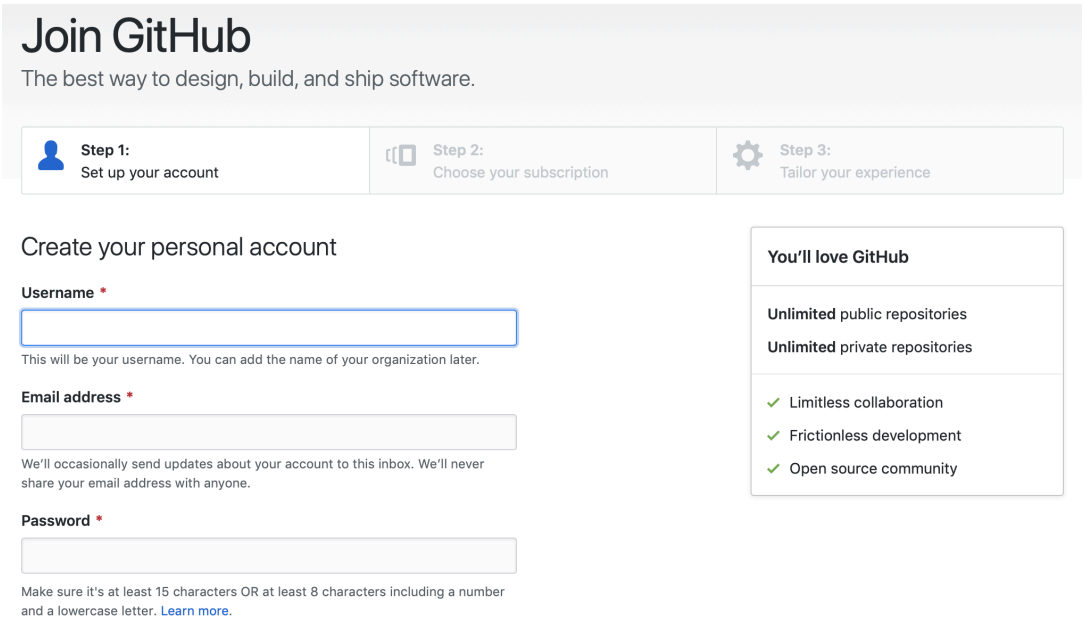
Git is the tool we will be using for *source control*, which means that we will be able to have many different versions of our project and can restore backup copies, add new features, and mix and match different versions freely.

It is currently the industry standard for software development. Think of it as a much more flexible Dropbox, or an improved version of the version history in Google Docs but where you are able to mix and match and create different “branches” or revert to a prior change at any time.

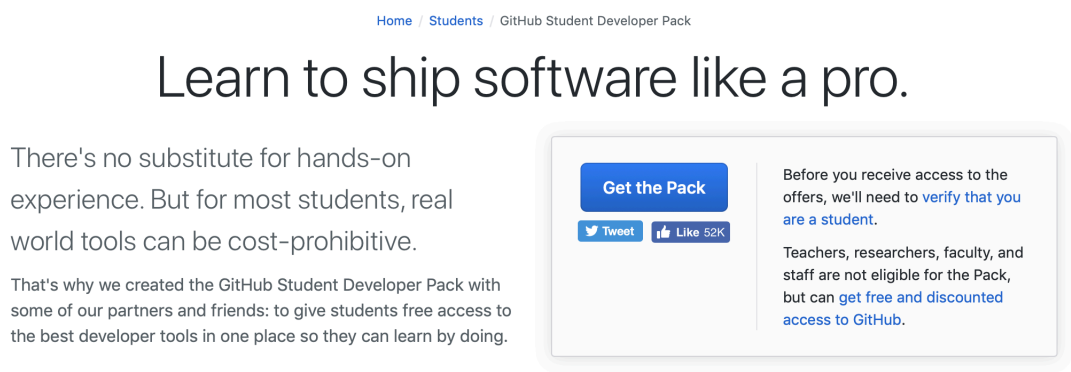
*This guide is for setting up and using **GitHub Desktop (a Graphical Tool for Git)** and aimed at people that are not familiar with Git/GitHub. If you know how to use Git/GitHub, you can use whatever method of interacting with HyTech repositories that you want. It is recommended to learn git via the **command line interface** because it is a more flexible and often faster way of doing most things.*

Creating a GitHub Account

Create a new GitHub account with your personal email. This will allow you to create an unlimited number of repositories (repos), which are basically like folders to hold your files for your projects. These personal repos can be public (open to anyone to see) or private (only you or a few others can see) This account also allows you to be a part of other organizations on GitHub, such as HyTech Racing.



Optionally, you can apply for a Github Student Developer Pack, which gives you related benefits. It is not required or necessary for the team, but you may wish to do so.



Github Desktop Installation

1. Download GitHub Desktop.
2. Run the installer file, exe (Windows) or dmg (MacOS) to install the application.

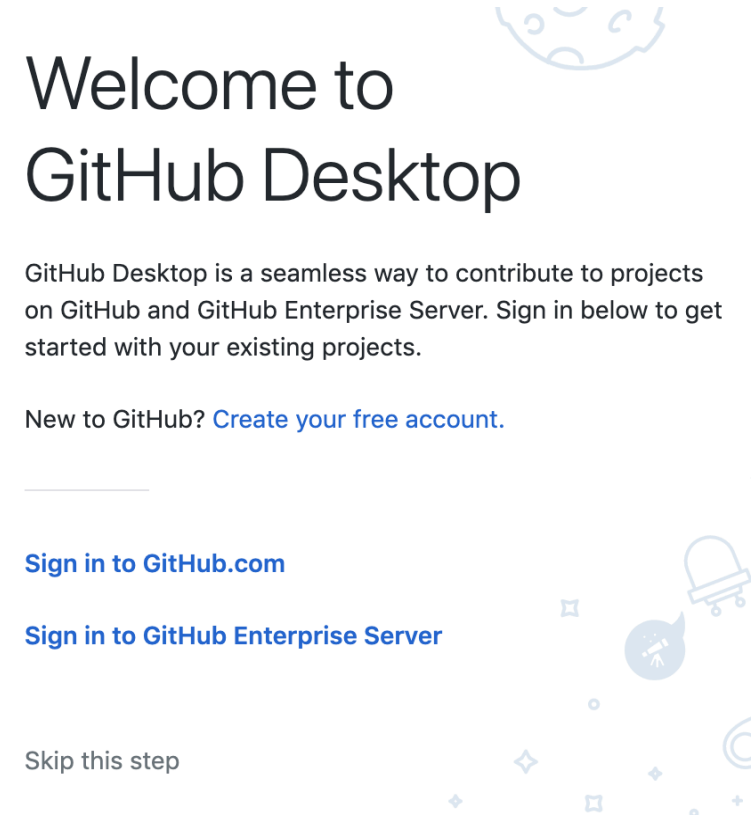
*GitHub **does not provide** a GitHub Desktop installer for Linux. You will just need to download the file from this GitHub Repository and install it.*

After installation, we need to clone HyTech repositories to our local machine. Cloning means to pull a repository and all its history to our local machine. Here, we will clone the training repository to your computers.

To get push access to other repositories, you will need to become a member of our Electrical Team on GitHub first. Push access means you can put updated files on the repository. Talk to an Electrical Team Lead to get access to the team on GitHub once you have paid dues.

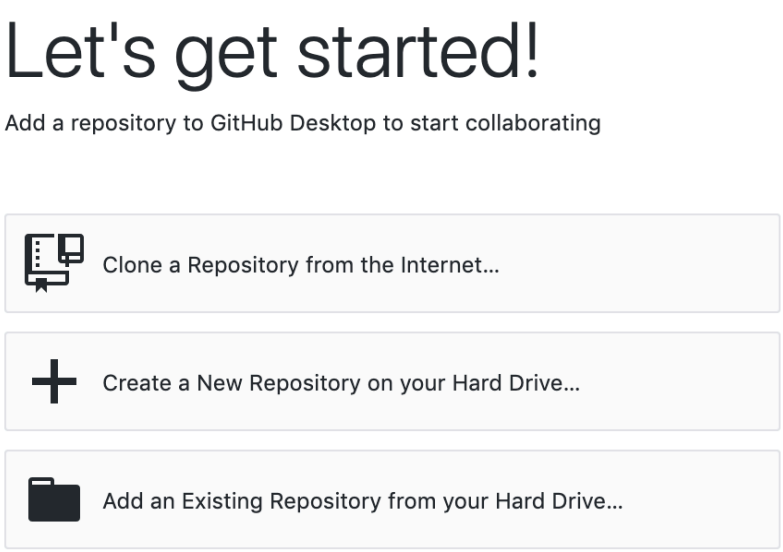
Cloning Repositories

1. Run GitHub Desktop application.
2. You should see a page asking you to login to GitHub, and follow the instructions to login.



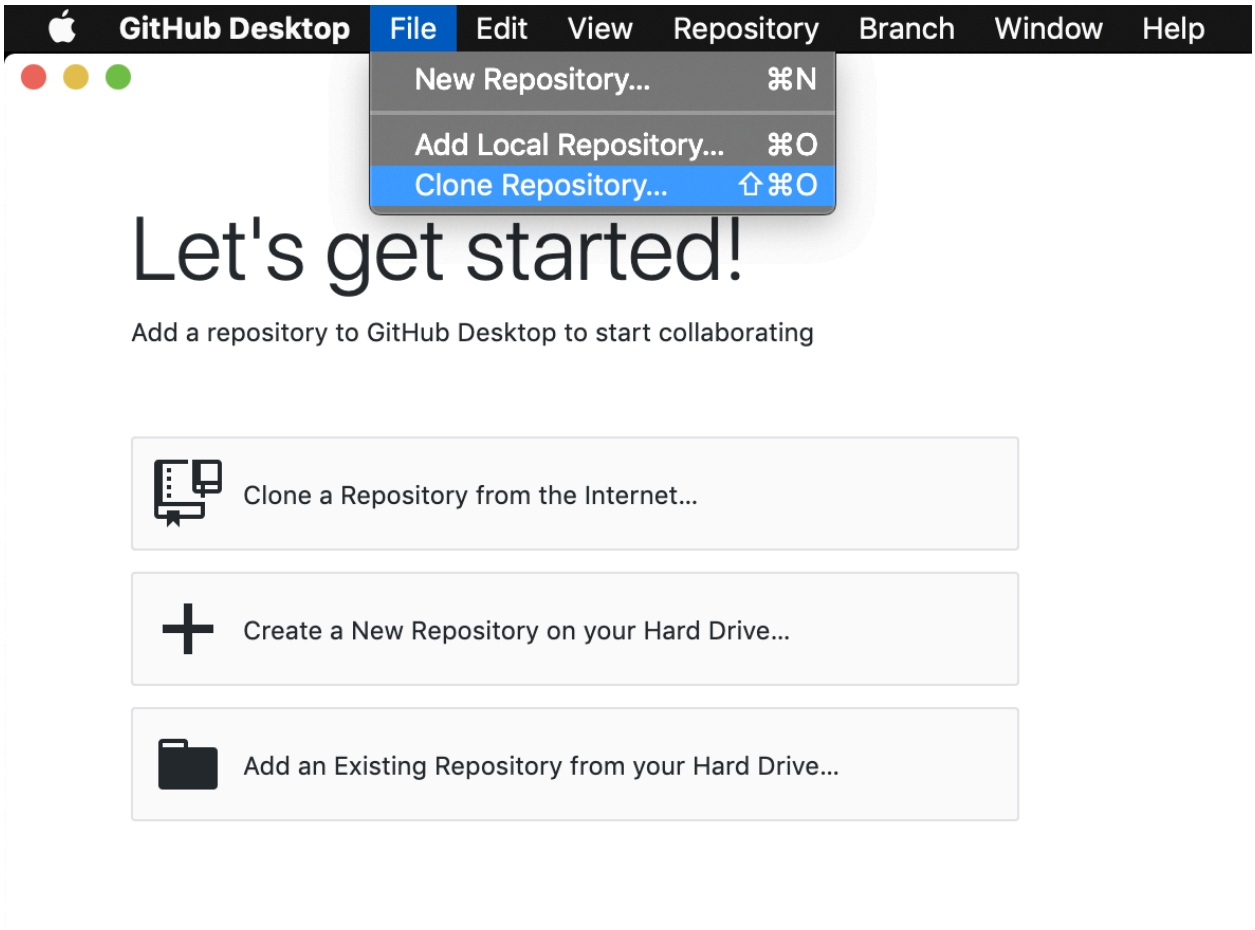
3. Accept the next prompt to select a username and email for your work on GitHub.

4. Click the **Clone a repository from the Internet…** button.



5. Choose the **URL** option:
    a. In the first field, you need to put the **URL** of the repository. As an example, clone HyTech Racing training repository by typing this URL (https://github.com/hytech-racing/training.git) into the field.
    b. In the second field, you need to specify the **location** on your local machine where you would like the repository files to be put.

6. If you want to clone more repositories, use **File->Clone Repository...**

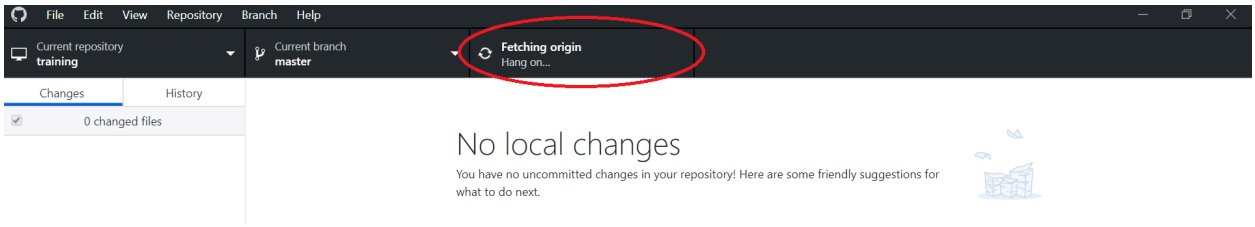


Creating Branches

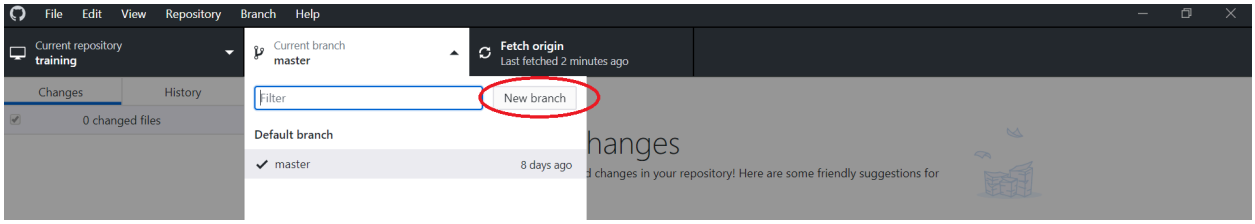
In order to start contributing to our repositories, you will need to create a branch. This allows different people to work on the same files without interfering with each other's work, working in parallel branches off of the same source. The "main" branch is called master, and you will usually branch off of that.

1. Before you create your own branch, it is usually helpful to get your local files up to date with the remote repository, which would be the GitHub repo. In order to do that, click **Fetch origin** at the top of the screen. This synchronizes your local copy of the chosen branch with the remote repository and gets changes that you

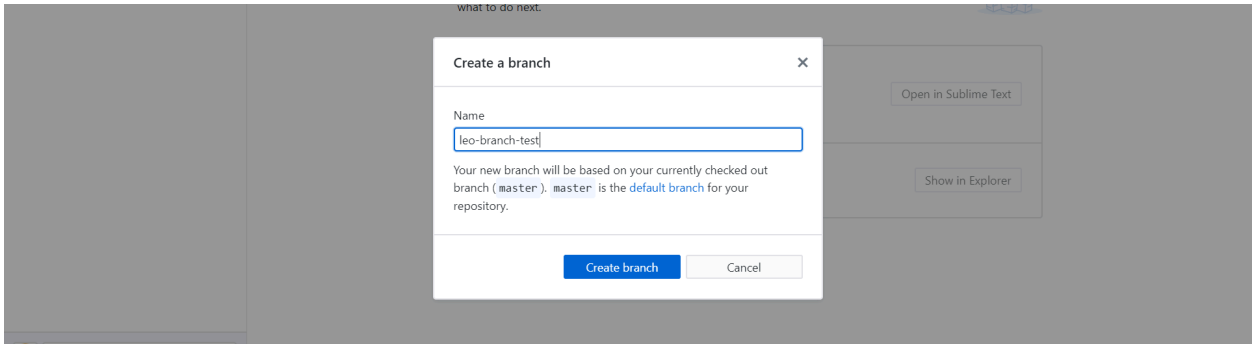
do not have locally.



2. Next to **Fetch origin**, you can see a dropdown menu that indicates the current branch and displays the list of all branches of the repository. Click on this dropdown menu and then click **New branch**.

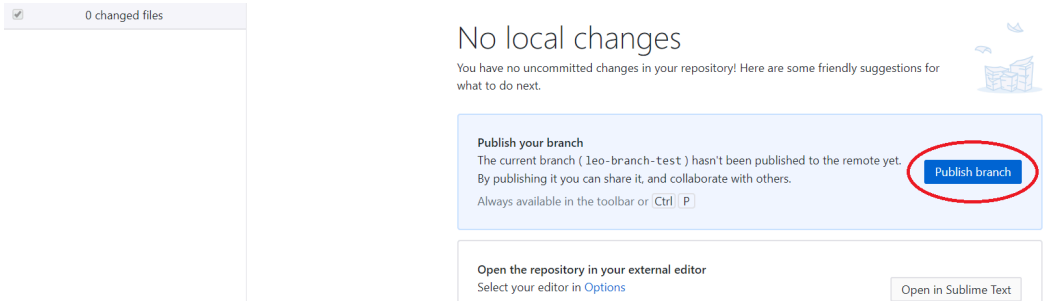


3. The application will ask you to give a name to your branch. It is good practice to give a descriptive name for your branch. For example, if you are working on updating libraries in EAGLE, you could name the branch “leo-branch-test”.



4. After you give your branch a name, click **Create branch**.
5. Finally, you can publish your branch, so progress is saved on the remote server in addition to your computer. It also allows other people to see what you are

working on and branch off of your branch. Click **Publish branch**.



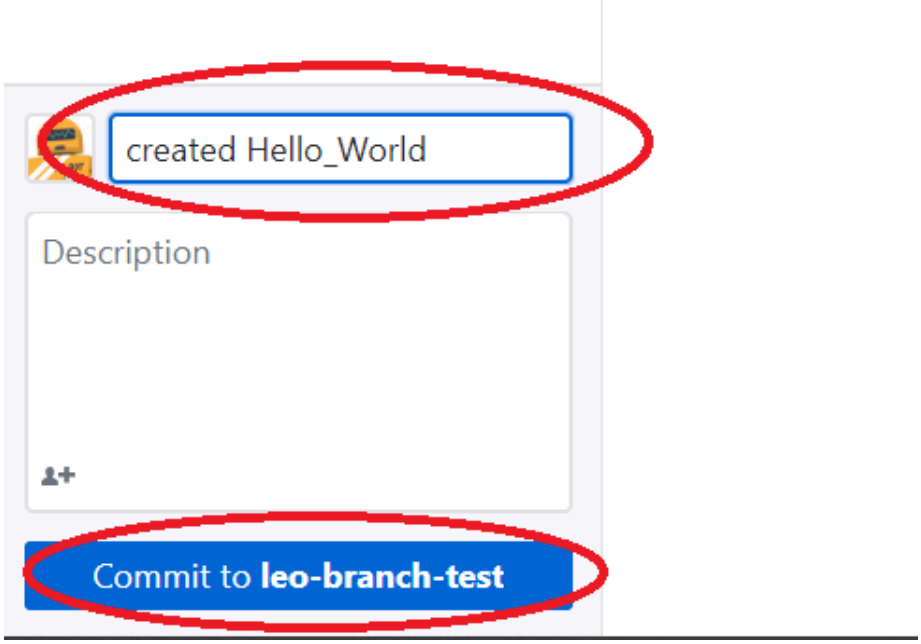
Adding, Committing, and Pushing Changes

After we have created our branch, and made some changes to existing files or added new ones, we need to publish those changes to our remote repository.

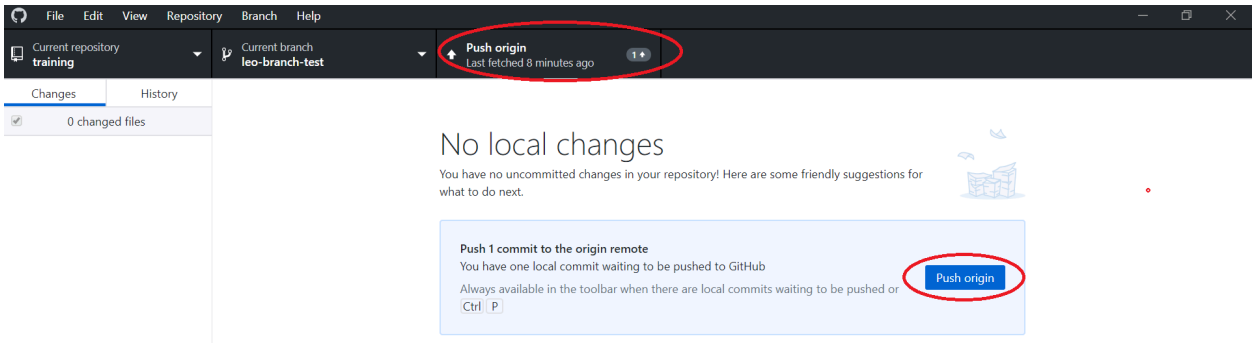
1. Here we have created a file with a simple “Hello World” program. First, we need to indicate that we want to **add** the file to our **commit**. Choose the files that you want to add to your commit by **checking** the appropriate boxes.

2. Add a short description to your commit in the top field, which will be used by others to figure out what your changes did. Then, press **Commit to**

<branch-name>.



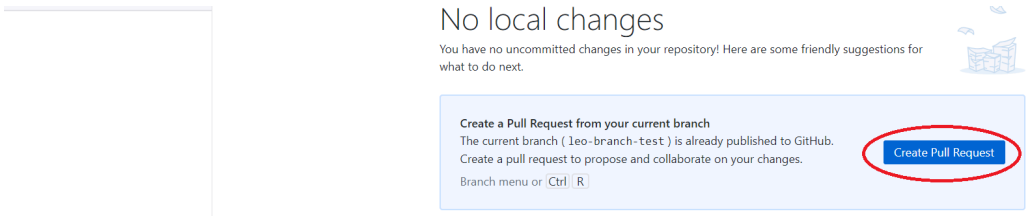
3. By committing our file, we save the current version of the branch on our local machine. Later, if we need, we can revert our commits to come back to a specific state of the branch at that exact moment in time.
4. Lastly, we can **push** our changes to the remote repository, so that all the changes are saved on the remote server as well. To do that, press either of the buttons that says **Push origin**.



Pull Requests

All HyTech repositories have a restricted *master* branch. This means that you cannot push your changes directly to *master*. This is done to protect the *master* branch from accidental changes and to make sure that code and designs on *master* work. If you think that your code or schematics need to be pushed to *master*, you need to create a **Pull Request** so that the changes can instead be **merged** into *master* instead of being directly added.

1. Press **Create Pull Request** (this can also be done in your browser , on the repository webpage on GitHub).



2. You will be transferred to your browser. In the provided fields, give your pull request a name and type a brief description of the changes. Then, press **Create pull request** in your browser to make a request.
3. Now, repository admins will review your pull request and either approve it or provide feedback for further changes.

Recap

The order of getting changes on a branch to the remote repository is:

1. Pull Remote
2. Make Changes
3. Add Changes
4. Commit Commit
5. Push Changes

The overarching process for a new feature involves making a new branch so it is as follows:

1. Pull Remote
2. Make Branch
3. Make Changes
4. Add Changes
5. Commit Commit
6. Push Changes
7. Make Pull Request to Master

Arduino

HyTech Racing uses Teensy Microcontrollers for our battery management system, vehicle controls, vehicle controls, battery management system, and many other things. These microcontrollers, which are based on the Arduino standard, can be programmed using the Arduino IDE.

Arduino IDE Installation

1. Download Arduino IDE.
2. Follow the guides below:
    a. Windows (Make sure **you do not install** the Windows App version of Arduino from the Windows Store).
    b. MacOS (Arduino IDE is bundled with Teensyduino, skip to the Teensyduino installation section).
    c. Linux (Follow the detailed steps).

Now you can program standard Arduino boards. However, to program Teensy microcontrollers, you need to install a special external utility, Teensyduino. Teensyduino allows to program Teensy microcontrollers using the Arduino IDE just like any other Arduino board.

Teensyduino Installation

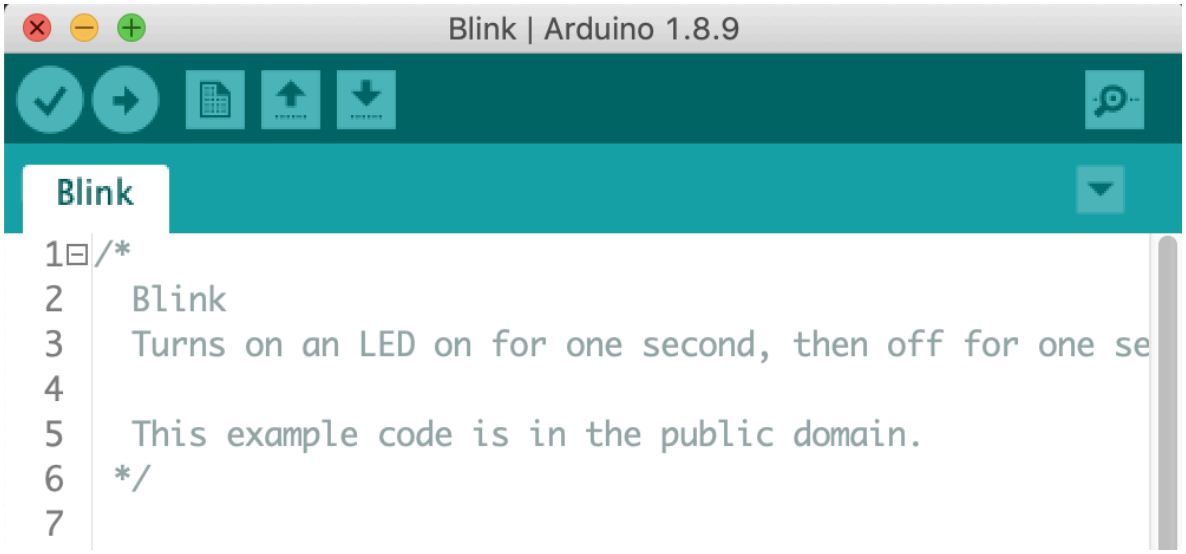
1. Open the Arduino IDE once.
2. Download Teensyduino.
3. Follow the instructions linked:
    a. Windows, make sure to select the correct directory containing Arduino on your system.
    b. MacOS, single installer bundles Teensyduino and Arduino together.
    c. Linux, take notes of the steps mentioned.

Programming Boards

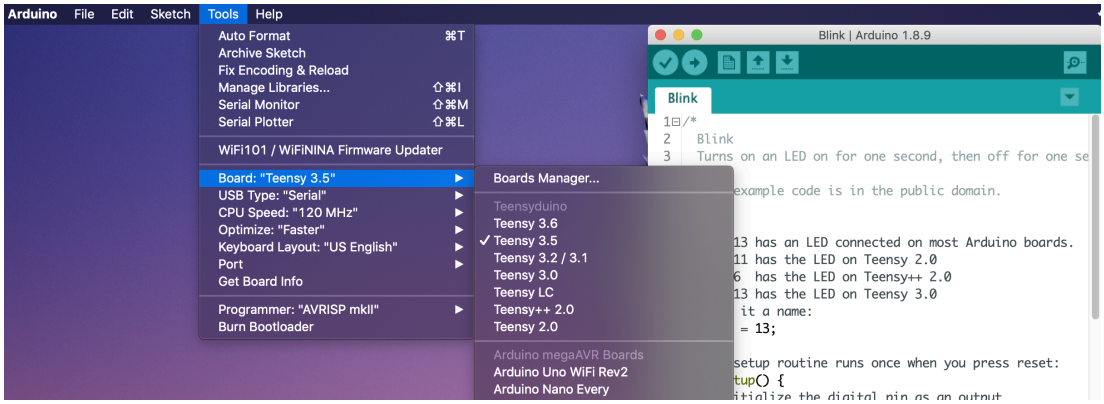
After you have installed Teensyduino, you can start programming microcontrollers.

1. Open an Arduino file with the Arduino IDE (it should have **.ino** extension).

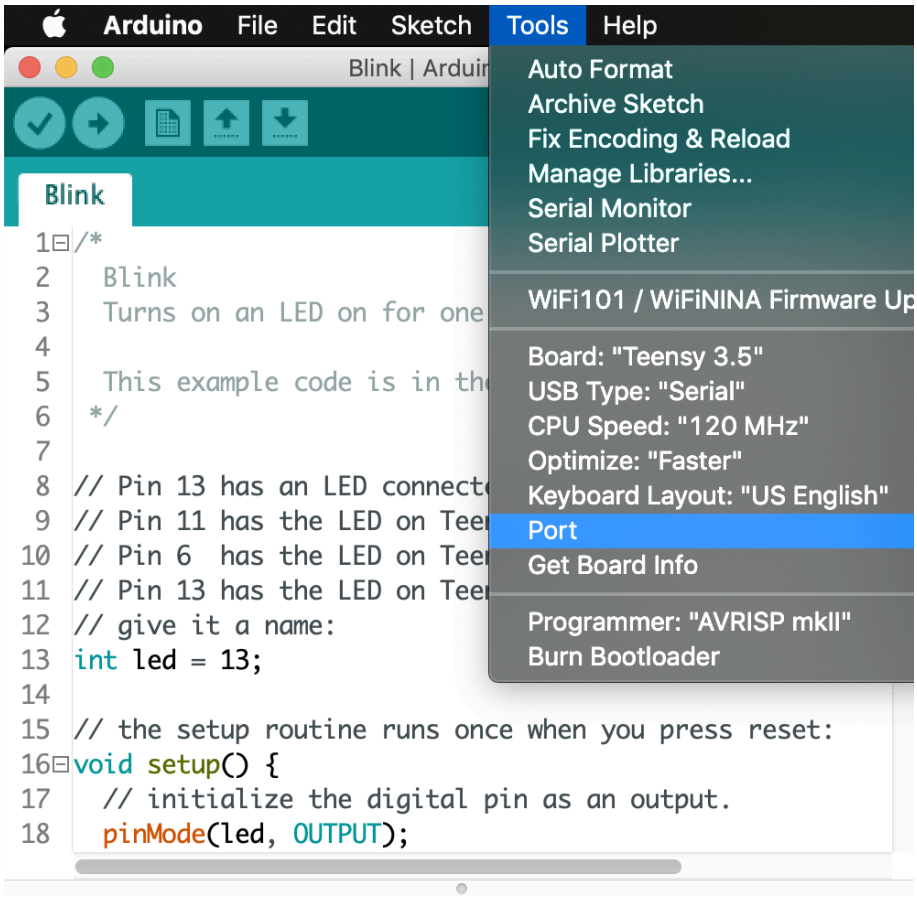
2. You can compile your program by pressing the checkmark button in the top left corner.



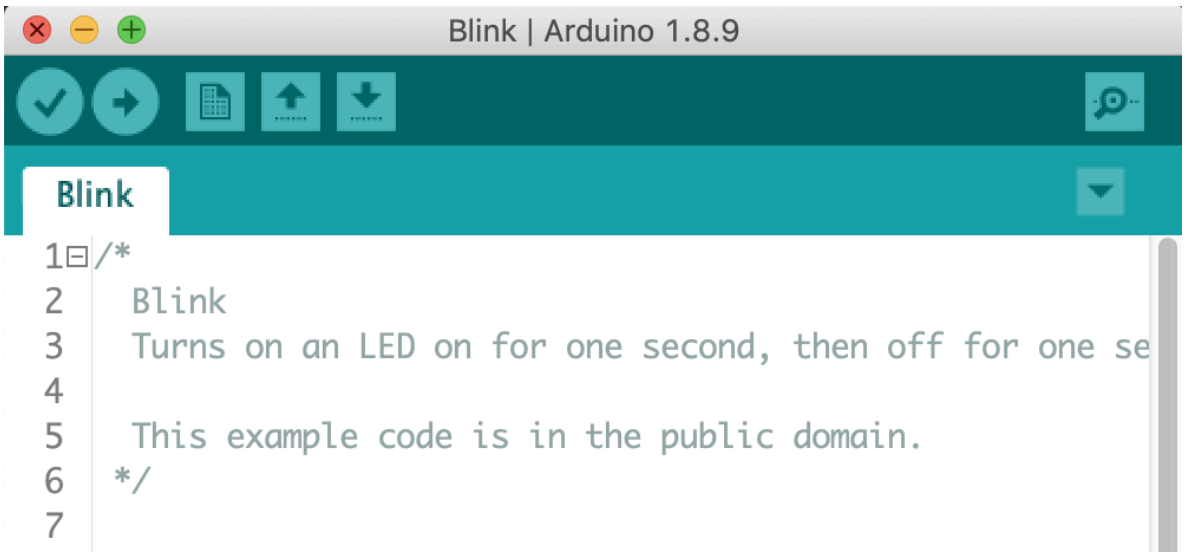
3. First go to **Tools->Board** and choose the board that you are programming.



4. Then go to **Tools->Port** and choose the port for your board.



5. Lastly, click the arrow button to upload the program to your board.

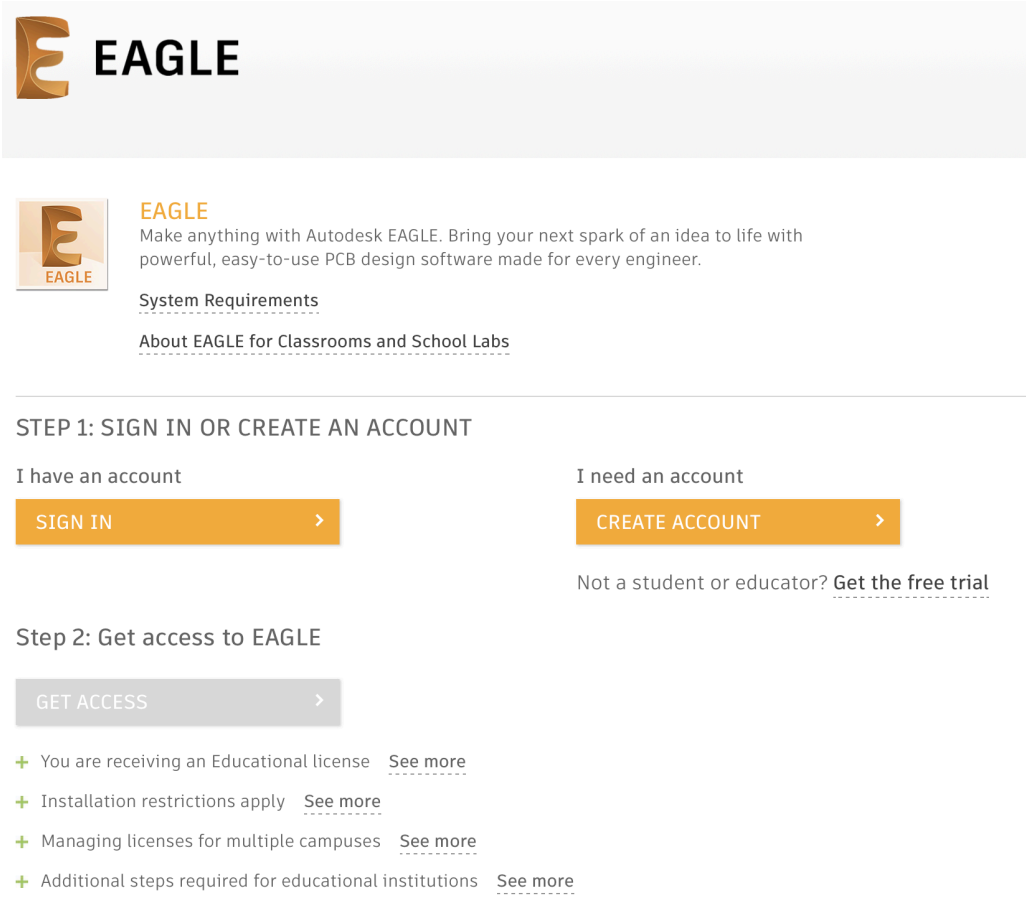


Autodesk EAGLE

HyTech Racing uses Autodesk EAGLE for designing schematics and PCB layouts.

Creating Autodesk Student Account

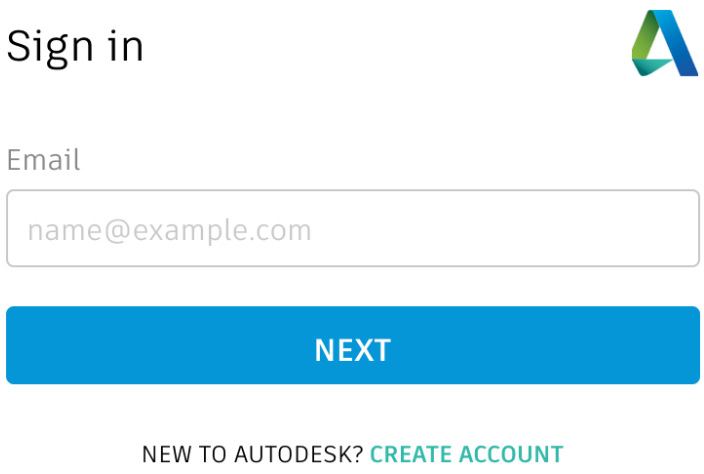
1. Go to the EAGLE Education Page.
2. You need to click “Create Account”, but be sure to **create your account with your @gatech.edu email** to ensure you can get the educational license.



3. Fill out the form and validate your account by checking your email.

EAGLE Installation

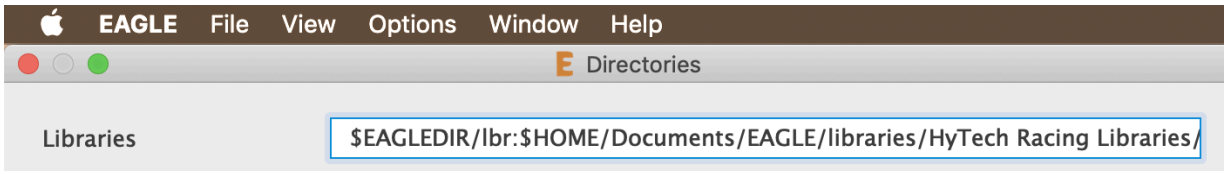
1. After creating an account, you will be redirected to a list of Autodesk products. EAGLE is not included on this page. Go to the EAGLE free download page instead.
2. Click Download.
3. Run the installer file, exe (Windows) or pkg (MacOS) to install the application.
4. Open the installed EAGLE application (you will have to answer yes to the prompt to create directories on your system)
5. Sign in with your Autodesk Account. This is necessary to activate the education version instead of the free version.



*Autodesk **does not provide** an EAGLE installer for Linux. You will just need to open the downloaded archive and run the file named "eagle". It would be recommended to move the contents of the archive to your documents and then setup a bash alias/adjust your path.*

EAGLE Setup

1. Using the Circuits Support Repository, clone the entire repository to somewhere on your computer using GitHub Desktop.
2. Using **File->Open**, go to where you cloned the repository and open all 4 libraries within the **EAGLE Libraries** folder to load them into EAGLE.
3. To permanently load them, use **Options->Directories** and put a semicolon ";" on Windows and a colon ":" on Mac/Linux and then put the path to the *EAGLE Libraries* folder from the GitHub Repository.



4. Now to open any EAGLE file, navigate to it using **File->Open**.