General steps to making changes:

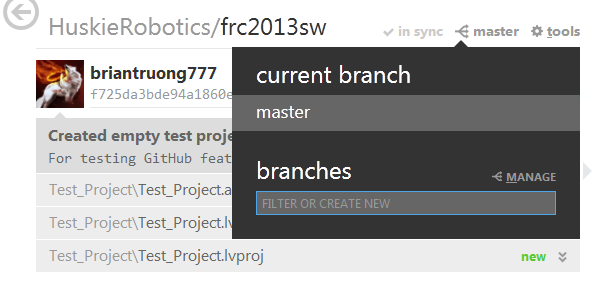
1. Make sure Labview projects are closed
2. Make a new branch or switch to a previous branch
3. Open project in Labview
4. **Do not switch branches** while project is open
5. Make changes to code
6. Save everything including the Labview project
7. Confirm you are on the correct branch
8. Commit changes
9. Sync if branch already existed or publish the new branch if you just created it
10. When ready for release, make a pull request
11. One of two possibilities depending if pull request is:
    1. Accepted: sub-branch is merged into master by an admin
    2. Rejected: development continues on sub-branch
       1. Can make further commits after making pull request to fix issues

Work Flow (as of 2013/01/12):

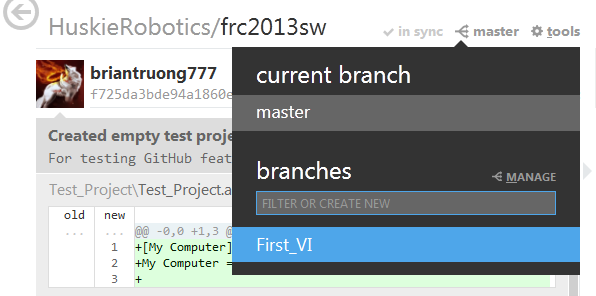
* Make pull requests to tell admin to merge from one branch to another
* Master branch should always be ready-for-release
  + The only commits made to master should be:
    - Merges from sub-branches
    - Universal changes affecting all sub-branches
    - Preparation for future branches
  + Only admin should make commits to master
  + All code added to master must be thoroughly tested
* Major sub-branches are each in charge of a certain sub-assembly (autonomous, drive, etc.)
  + A branch has only a certain set of files it can make changes to
  + These files are **never** in common among multiple branches
  + Major sub-branches are later merged into master by an admin
  + Only admins should make changes here
* Minor sub-branches are made for each person for work on a particular major sub-branch
  + Follow same rules as major sub-branch except owner can make changes
  + This way a single major sub-branches can have multiple people working on it
  + These minor sub-branches are intended to be temporary and will be deleted quickly
  + Pull requests are made to merge a minor sub-branch to a major sub-branch
  + A minor sub-branch should only be edited by its owner or an admin
* Each branch can test its own code regardless of development on other branches
* When code has been tested, a pull request is made to merge minor sub-branch into major sub-branch
  + Follow with online discussion on GitHub if necessary
  + Admin will then handle merging (or not if request is rejected)
* Any changes that will affect multiple branches must be made in master or major sub-branch
  + This will be done by an admin
  + These changes will be distributed by merging from master to all sub-branches if needed

Making a new branch:

1. Click the branch icon, ensure you are on the branch you want to branch off from, and type in a new branch name (don’t forget to publish the branch later)

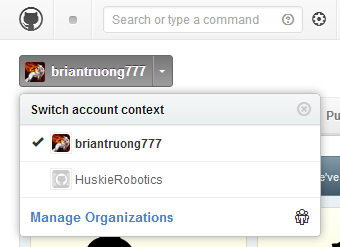


Switching to previous branch:

1. Click on the branch icon, and click on the particular branch

Making a pull request:

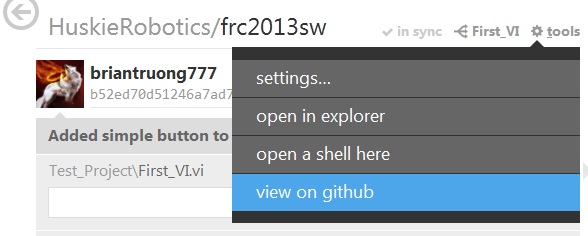
1. Go to github.com
   1. In the top right of the screen, click on your username to switch to the HuskieRobotics organization



* 1. On the right side of the HuskieRobotics organization page, click on the repository holding the season’s code



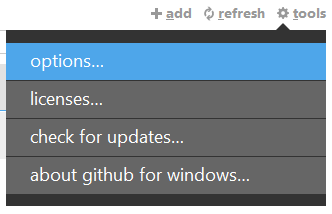
1. Alternatively in GitHub for Windows, click on tools, then “view on github” to go directly to the repository page



1. Click on the pull request button in the upper right of the repository page
2. Select your sub-branch in the right-side dropdown and master in the left-side dropdown (Merge from right-side branch to left-side branch)
3. Give the pull request a title and write a comment to describe overall changes since the creation of the branch

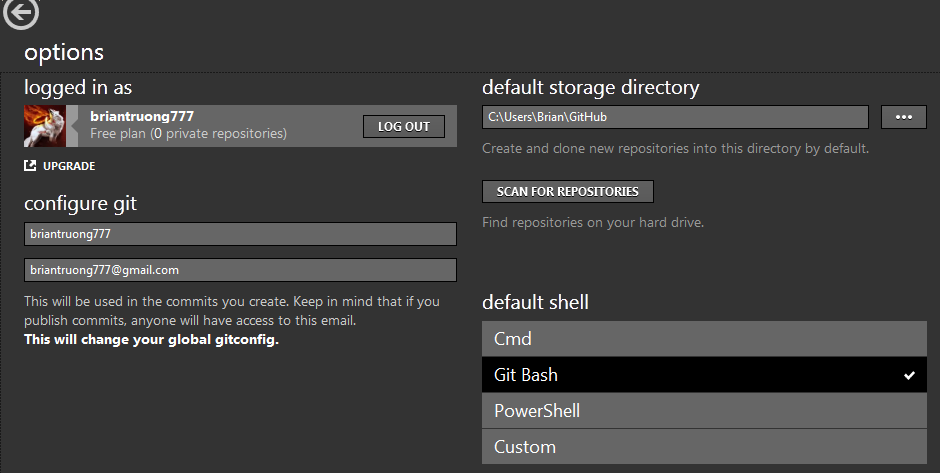
Setting up GitHub for Windows:

1. Download and install it from [windows.github.com](http://windows.github.com/)
   1. You do need an internet connection for the installer
2. Login with your GitHub user name and password
   1. You must login in order to adjust the following settings
3. Click on tools->options…

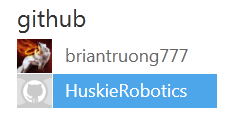


* 1. Make sure you are not currently in a repository
  2. You can also login here if you didn’t already

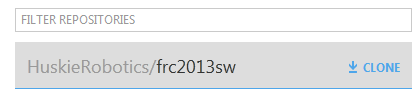
1. Configure a name and email to be attached to every commit you make
2. Select your default shell
   1. You can choose whatever you are familiar with
   2. Otherwise, I recommend Git Bash
3. Select default storage directory
   1. This is where all your local repos will go
   2. It’s a good idea to have a shortcut to this location for quick access
4. Click the update button
   1. Example of a correct setup:



1. Select the remote repo “HuskieRobotics” from the list of local repos under “github”



1. Click on the clone button to create a local repo



1. Click on the local repositories list to see your newly clone repo

