**Git Hub – Setup and Use Instructions** – 1/22/19, Version 1.1

1. Setup an account with git hub*. (A free account is fine…. no need for private repositories)*
2. Email your git hub **user name** to Russ. You will be added as a collaborator for repositories created on git hub. (email to [mjp763@zebra.com](mailto:mjp763@zebra.com) AND [RussC243@optonline.net](mailto:RussC243@optonline.net) )
3. Install Git for Windows from here:

<https://gitforwindows.org/>

This document only describes **Git Bash** which is one of the three tools that will be installed. A windows GUI is installed when you install git but it has many features you don’t need, so it can be more complex than learning how to type basic commands into Git Bash.

1. If you have never been exposed to Linux before, you will need to learn the basic commands to change directories, copy and paste etc. This link gives an overview of navigating around, cutting and pasting and other tips when using Git Bash.

<https://www.youtube.com/watch?v=oQc-2gsjgDg>

1. Create a folder on your local machine where you will keep your local repos. A name like “GitLocalRepos” is fine. You can use git bash or windows file explorer or command prompt for this step.
2. From git bash, cd into the folder you created in the last step or navigate to the folder with windows and right click->Git bash Here
3. Open this link

<https://github.com/RussC243>

1. You will see various repositories with names starting with WM6423. Select WM6423\_GitHub\_Test.
2. Go to the repo on git hub, click the green “Clone or Download” button then click the copy icon. This copies the URL of the repo to the windows clipboard.
3. **Clone the repo** to your local drive by typing git clone and pasting in the URL

$ git clone https://github.com/RussC243/WM6423\_GitHub\_Test

You should now have the repo on your local machine.

1. **Make changes to the local repo**. Normally this will be done with your IDE such as VS Code or Netbeans but for this example we can use Notepad. Add some text like “Hi this is Steve R testing”. Save the file.
2. With Git bash, cd into the WM6423\_GitHub\_Test folder. Notice (master) at the end of the line. This means you are on the **master branch**. For this example, working directly on the master branch is fine. Normally we will create a branch off the master branch and only push things to the master branch after the change was tested and other team members had a chance to review the changes.

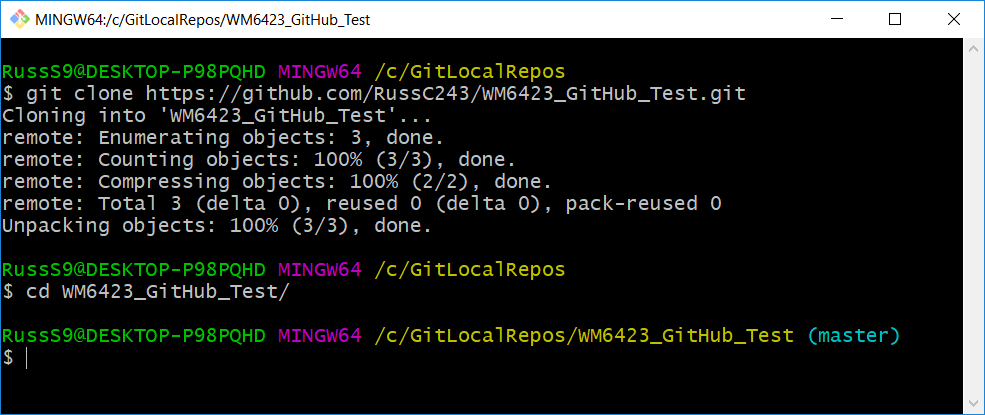
Git init

$ git clone https://github.com/RussC243/WM6423\_GitHub\_Test

git remote add origin <https://github.com/RussC243/WM6423_RoboRio_2019.git>

git pull origin master

in vs code – pull from



1. **Stage your changes** (the dot means all files). Staging just means selecting which files you want to push back to the repo on Git Hub.

$ git add .

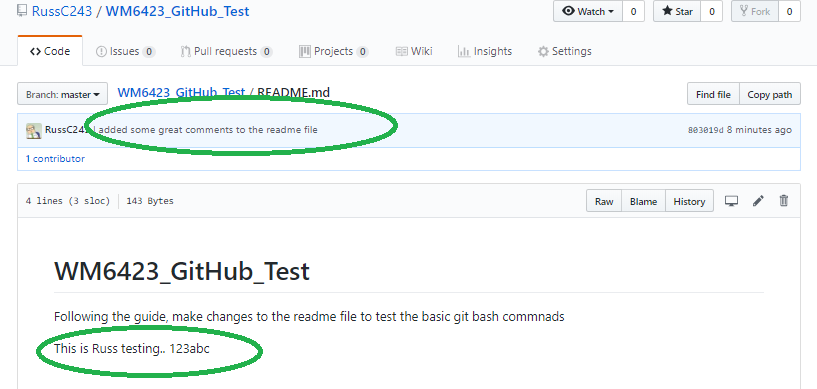
1. **Commit your changes.** The message should describe what you changed. Commit just means “Are you really really sure?”

$ git commit -m “I added come great comments to the readme “file”

1. **Push** your changes back to the repo. \**You won’t be able to do this step until you have been added as a collaborator*

$ git push origin master

1. Check the repo on Git Hub to verify your changes made it to the repo



**Instructions to merge a branch back to the master branch** – Scenario: Someone create a branch from the master, made changes to that branch, tested the changes and everyone deemed the changes needed.

Right click your local repo folder and select “Git Bash Here”

From git bash:

clone

$ git checkout master

$ git pull origin *branchName*

$ git merge --no-ff *branchName*

$ git merge --no-ff origin/*branchName*

$ git push origin master

**Instructions for Repository Owner -** Create your own repos from projects you have on your local drive)

1. Initialize a repository from existing code on your local drive
   1. Add a repository on the GitHub site (Don’t add readme file.)
   2. Open Git Bash

cd into the folder to be pushed or from file explorer, right click->git bash here

* 1. Initialize repository

$ git init

* 1. Configure git not to change line endings

git config --global core.autocrlf false

* 1. Add all files to staging area

$ git add -A

* 1. Commit

$ git commit -m "Initial Commit"

* 1. Specify remote repository URL

$ git remote add origin https://github.com/RussC243/WM6423\_RoboRio\_2019.git

* 1. Verify remote URLgit remote

$ git remote -v

origin https://github.com/RussC243/WM6423\_RoboRio\_2019.git (fetch)

origin https://github.com/RussC243/WM6423\_RoboRio\_2019.git (push)

* 1. Push to remote repository

$ git push -u origin master

JSON files for third party libraries:

1. Go to VSCode
2. Click on WPI Command Palette
3. Click on “Manage Vendor Libraries”
4. If Pheonic and navX.json files are missing (which they will be if you made a new project) click “install new libraries (online)” and add the following URLs for each vendor.

Pheonix -  <http://devsite.ctr-electronics.com/maven/release/com/ctre/phoenix/Phoenix-latest.json>

navX - https://www.kauailabs.com/dist/frc/2019/navx\_frc.json

**Setup VS code**

Pull from

Change files then

Save, stage, commit, push