|  |
| --- |
|  |
| Quick Start Guide for GitHub |
| Version 01.01.13 |



**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Remark** | **Date** | **Done By** |
| 01.01.01 | Initial Release | 6th May | Kinjal |
| 01.01.02 | Add file structure for hardware | 6th May | Kinjal |
| 01.01.03 | Add the detail regarding Hardware file structure | 7th May | Kinjal |
| 01.01.04 | Add figure reference | 7th May | Kinjal |
| 01.01.05 | Edit as format | 8th May | Kinjal |
| 01.01.06 | Change in Chapter #6 and Chapter #7 | 11th May | Kinjal |
| 01.01.07 | Add firmware team detail | 17th May | Kinjal |
| 01.01.08 | Update Chapter #5 | 18th May | Kinjal |
| 01.01.09 | Add Developer Release detail | 15th June | Kinjal |
| 01.01.12 | Added how to use GitHub Client, sync, commit | 7th July | Heer Hirpara |
| 01.01.13 | Some minor updates and correction | 7th July | Ashok Hirpara |
|  |  |  |  |

Table of Contents

[1. Purpose 4](#_Toc423699787)

[2. Definition(s) and Abbreviation(s) 5](#_Toc423699788)

[3. Reference(s) 5](#_Toc423699789)

[4. Overview 6](#_Toc423699790)

[4.1 What is GitHub? 6](#_Toc423699791)

[4.2 Why use GitHub? 6](#_Toc423699792)

[4.3 How it works? 6](#_Toc423699793)

[5. Install GitHub Client 7](#_Toc423699794)

[5.1 Download GitHub Client and Configure 7](#_Toc423699795)

[5.2 Accept invitation from Delphian-Systems 9](#_Toc423699796)

[5.3 Clone Repository 11](#_Toc423699797)

[5.3.1 Clone repository 11](#_Toc423699798)

[5.4 Add new file/directory to repository 12](#_Toc423699799)

[5.5 Synchronizing repository 16](#_Toc423699800)

[5.5.1 Making changes to files while multiple users are working on it. 17](#_Toc423699801)

[5.6 Viewing History 22](#_Toc423699802)

[6. Getting started with GitHub for Delphian Repository Structure 24](#_Toc423699803)

[7. File Structure for Hardware Data 27](#_Toc423699804)

[7.1 Retrieve Hardware data from Repository 28](#_Toc423699805)

[8. File Structure for Firmware data 33](#_Toc423699806)

[8.1 Retrieve Firmware data from Repository 34](#_Toc423699807)

[9. Developer Release 37](#_Toc423699808)

[9.1 Retrieve Developer\_ Release data from Repository 38](#_Toc423699809)

# Purpose

* The purpose of this document is to understand features of GitHub to share documents
* Maintain version control for documents as well as source code
* This document will provide the details of how to submit and retrieve content in the GitHub archival system.

# Definition(s) and Abbreviation(s)

|  |  |
| --- | --- |
| **Name** | **Abbreviation** |
| Repository | Folder (Directory) created by Administrator and shared with group of users who can read or write depending on permission given to user by Administrator. |
|  |  |
|  |  |

Table #1

# Reference(s)

|  |  |
| --- | --- |
| **Document #** | **Title** |
| 1 | <https://github.com/> |
|  |  |
|  |  |
|  |  |

Table #2

# Overview

The goal of Release Management is to take a holistic view of a change to an IT service and ensure that all aspects of a release, both technical and nontechnical, are considered together. Everyone has to be aware with release firmware/documents/source and version control process.

* 1. What is GitHub?
  2. Why use GitHub?
  3. How it works?

## 4.1 What is GitHub?

* GitHub is the portal to manage changes in a project without overwriting any part of it along with code review and management.

## 4.2 Why use GitHub?

• Suppose you and your team members are updating pages on the same website, then you can make your own require changes, save and upload them back to the website.

• It provides repository, pull-push facilities, source code management and multi access functionality.

• One of you are about to have your work overwritten and erased. A version control application like GitHub keeps that from happening. You and your team member can each upload your revisions to the same page, and Git will save two copies. Later, you can merge your changes together without losing any work along the way.

## 4.3 How it works?

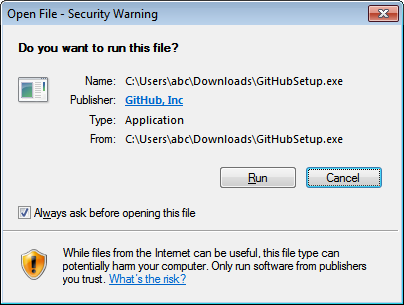
* GitHub makes Git easier to use in two ways: Firstly, if you download the GitHub software to your computer, it provides a visual interface. Secondly, creating an account on GitHub.com brings your version-controlled projects to the Web, and ties in social network features for good measure.
* One can browse other GitHub users’ projects, download those open source general public licensing copies or even suggest fixing the bugs for reference.
* **Please note that using browser, one can download repository content. GitHub Client is required if you want to modify the content of any repository.**

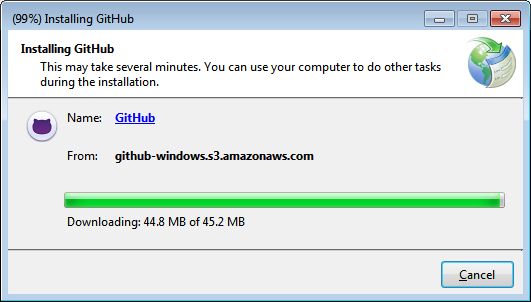
# Install GitHub Client

GitHub Client is required to be installed on PC so user can updates documents in repository.

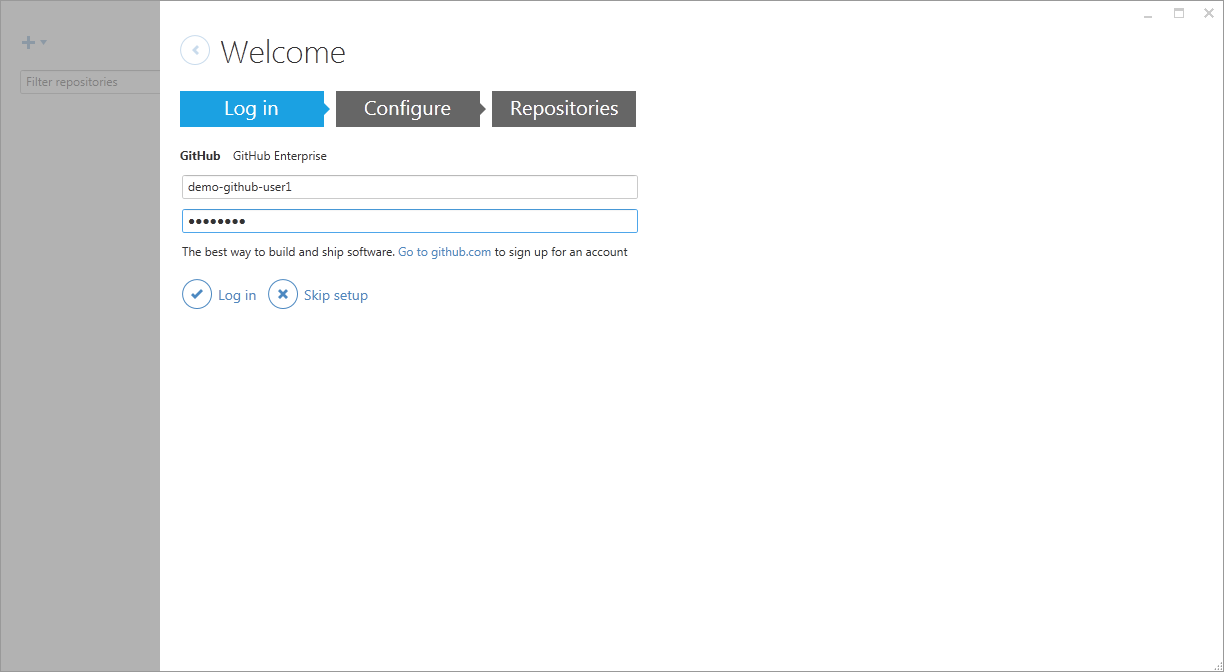
## Download GitHub Client and Configure

1. Download the latest version directly from <https://windows.github.com/> (For Windows) or <https://mac.github.com/> (For Mac) and click on it to install.



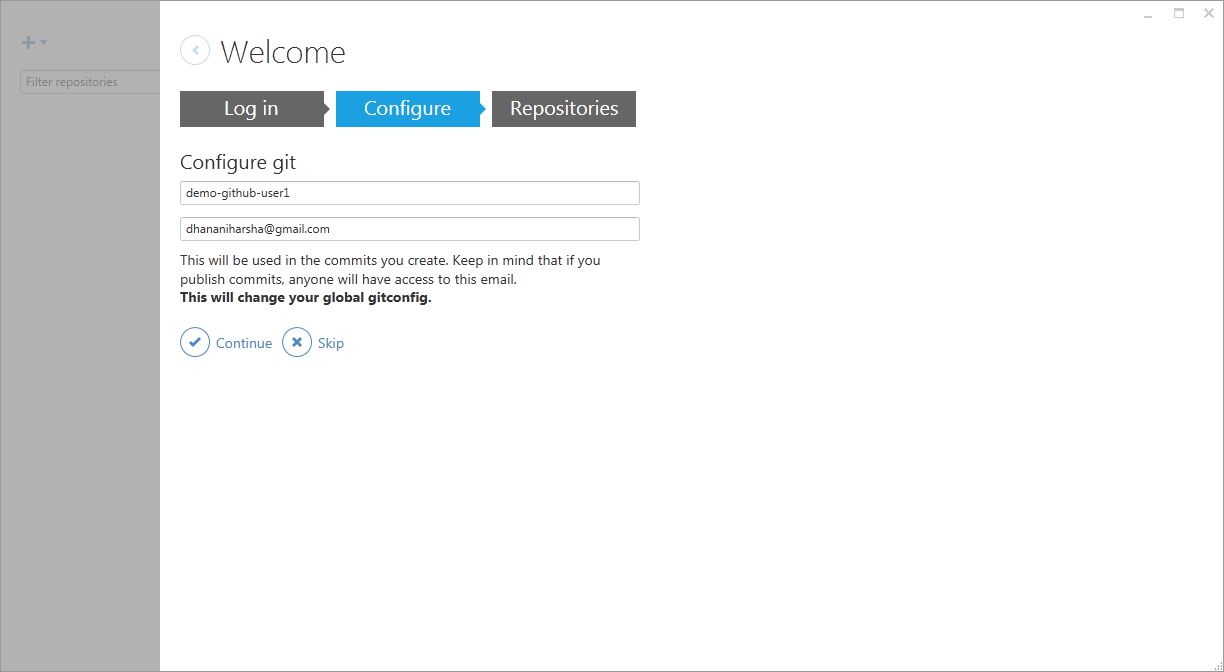


1. Launch GitHub using GitHub icon on desktop



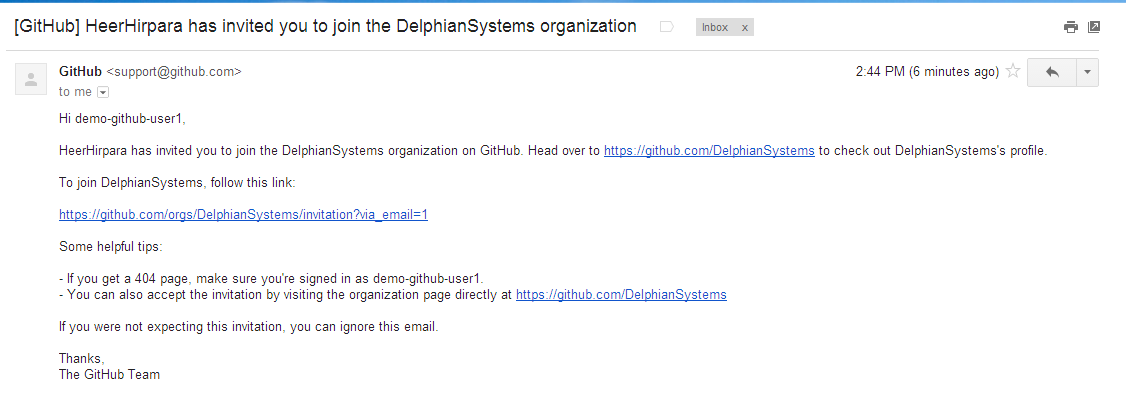
* **For New GitHub User: Please contact administrator at** [**heer.hirpara@delphiansystems.com**](mailto:heer.hirpara@delphiansystems.com) **to get new user information for login to GitHub**
* **For Existing GitHub User: login to GitHub using your login information**

1. Configure GitHub: Select Continue. It will give message – “ No Local repositories found”

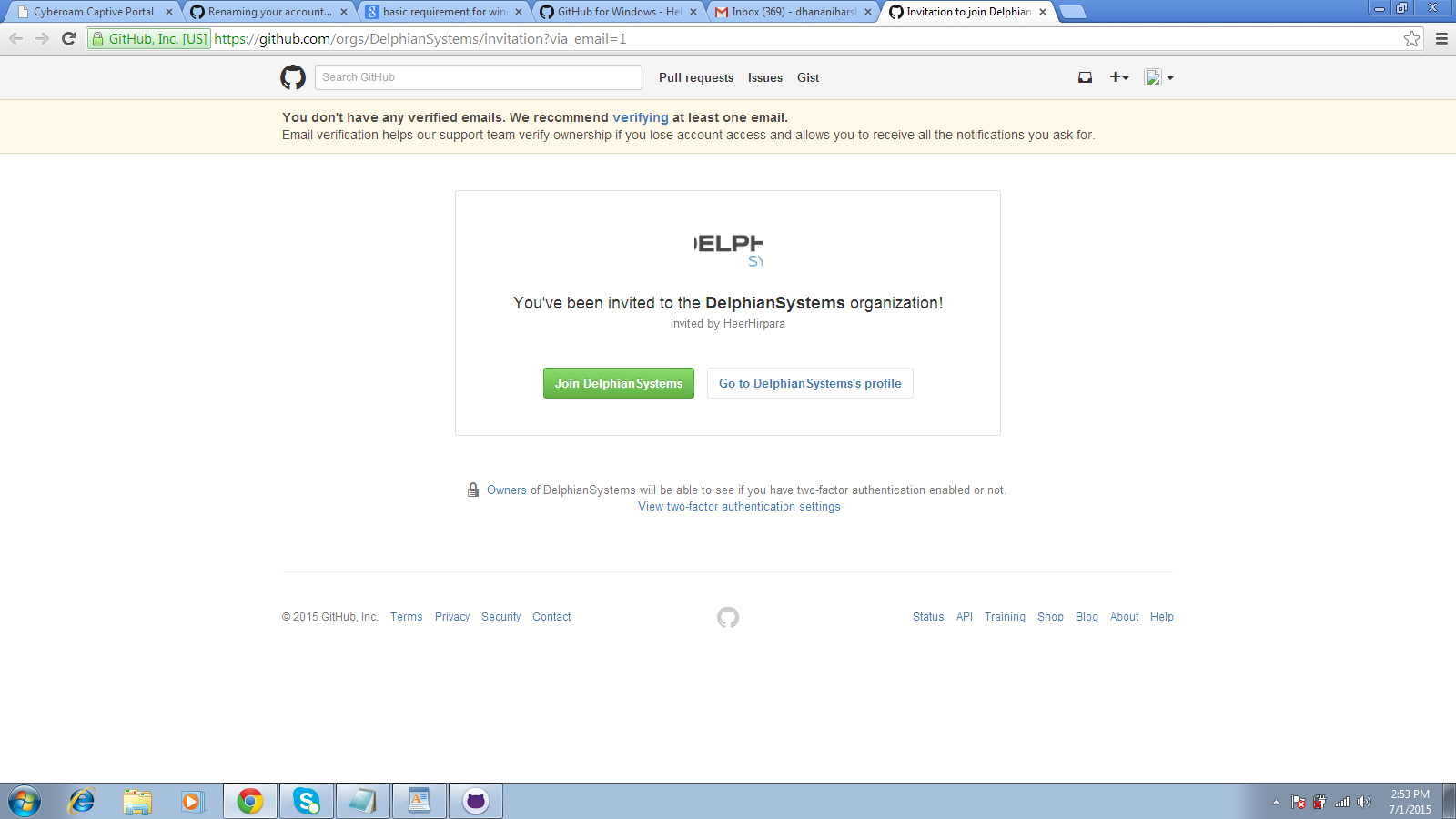


## Accept invitation from Delphian-Systems

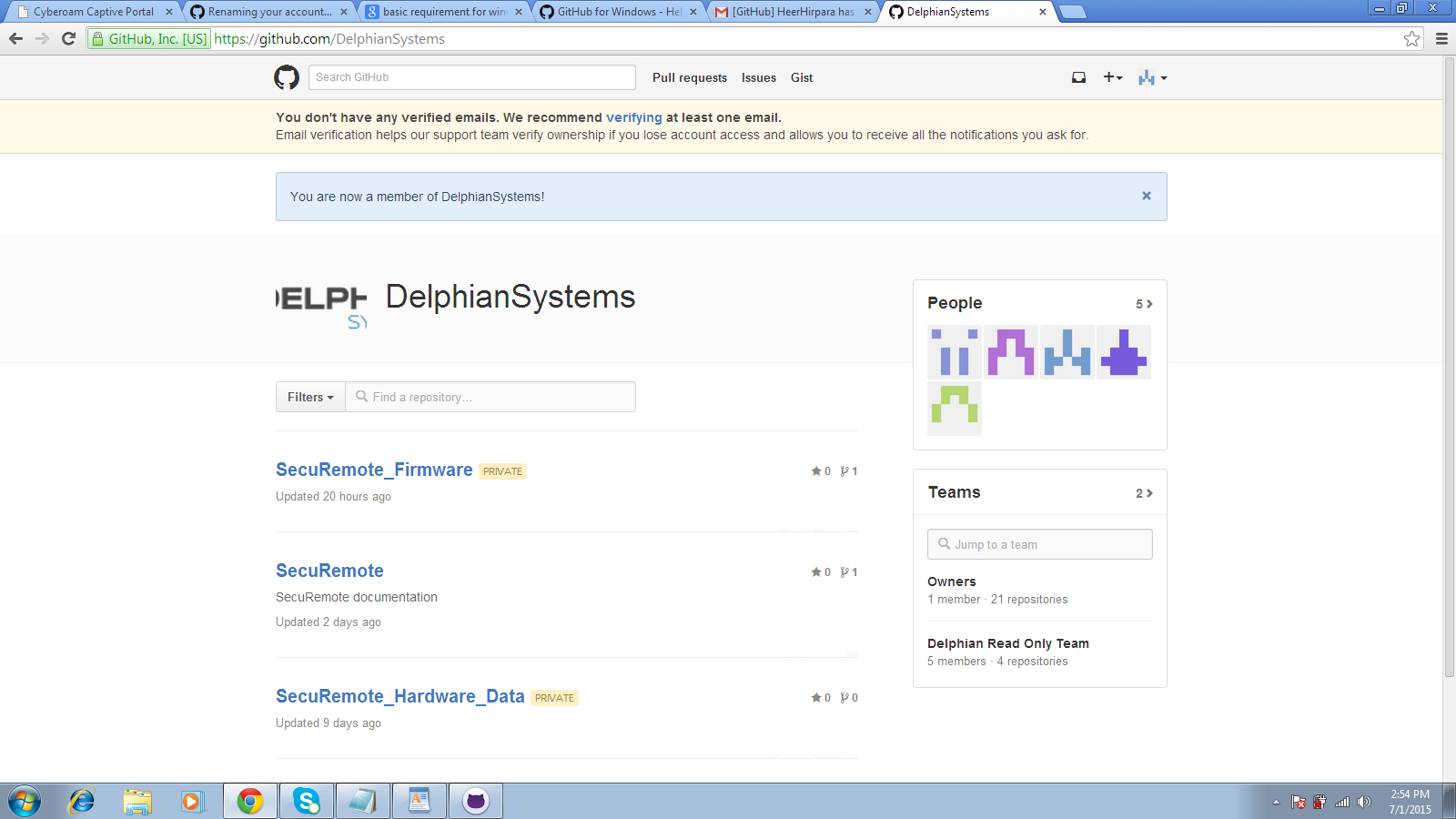
1. Once you have been a Git user, you can ask Administrator at [**heer.hirpara@delphiansystems.com**](mailto:heer.hirpara@delphiansystems.com) of Delphian Team to provide access to Delphian Repository. Admin will invite you to join Delphian Systems organization and you will receive email for the same. Please note that if you are provided account by Admin of Delphian Systems then you don’t have to process this step.



1. Click on the link to join the Delphian Organization. It will redirect you to GitHub online where you are supposed to click on Join Delphian Systems.



1. By joining the organization, it will show the entire repositories which are available to you.



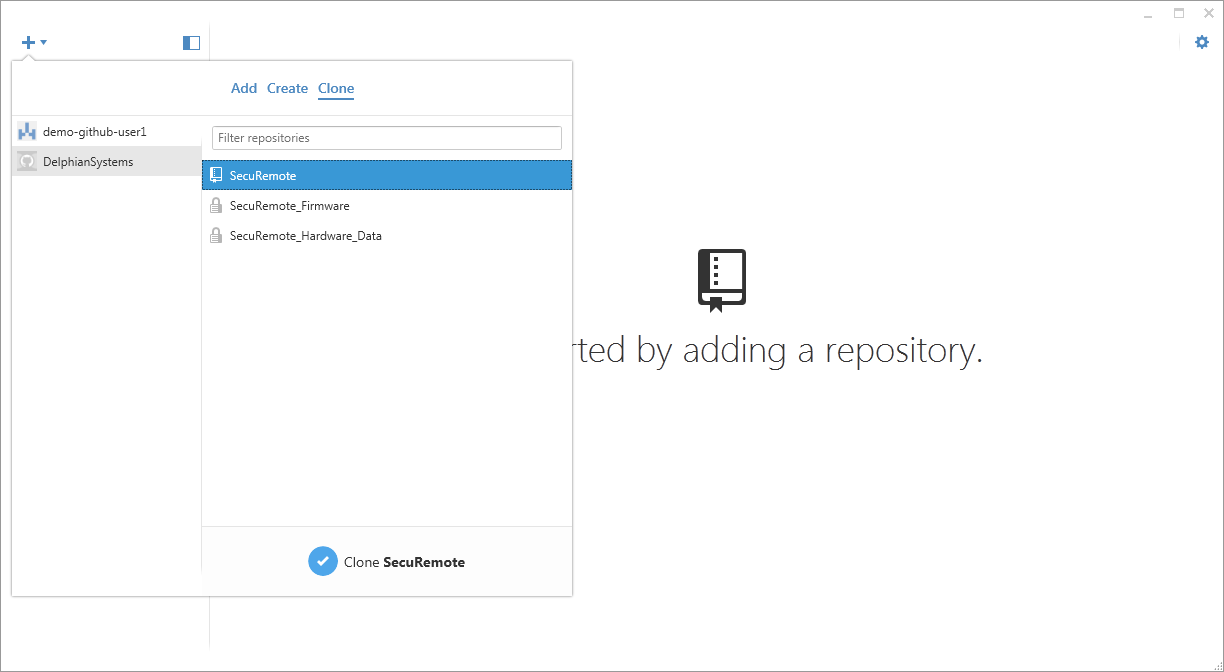
## Clone Repository

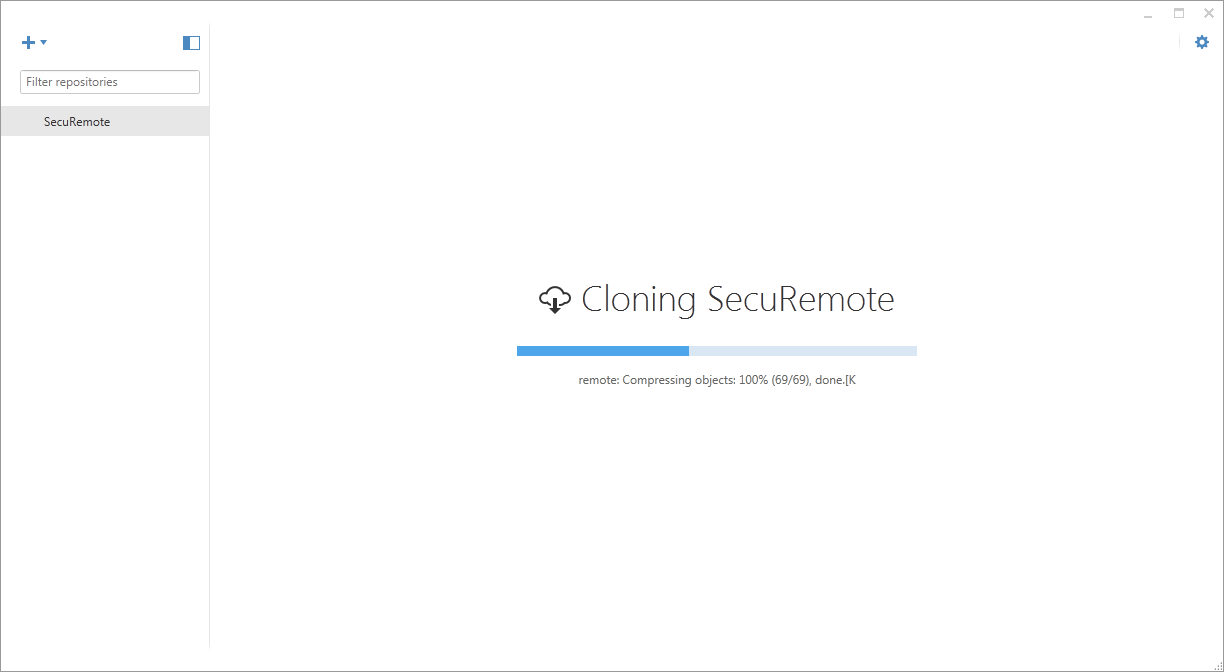
User can click on **+** add at the top of the program, and create a new repository locally.

On the left, you'll see your GitHub account, as well as any organizations you're a part of e.g. Delphian Systems. Clicking on a name will show you which repositories are available. Clicking on clone copy the repository to local drive on your computer. It is required to clone repository to update any document in this repository or add any new document to this repository.

### Clone repository

User must use this option to download all files & folders for this repository to local drive. Once repository is downloaded, user can add/delete/update files & folders within this repository and share updated files to other GitHub users by uploading them to repository.



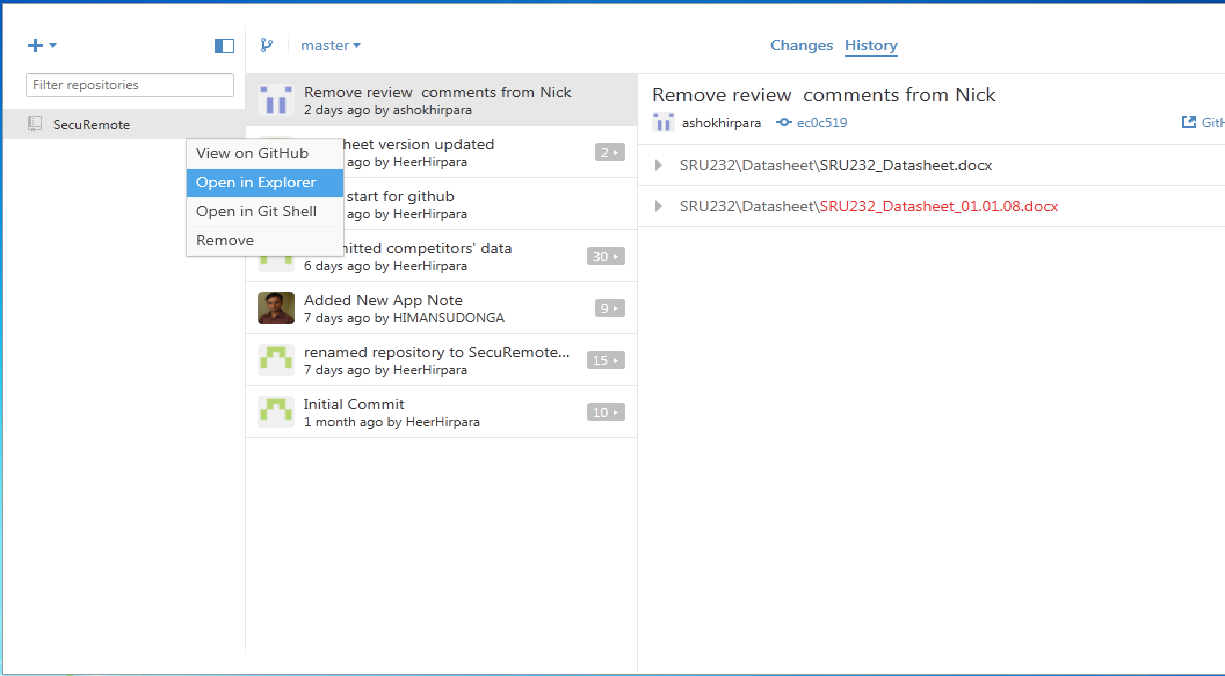


Note: “Add” and “Create” options are for personal repositories only. It cannot be shared with other users. To create new repository, please send request to Administrator.

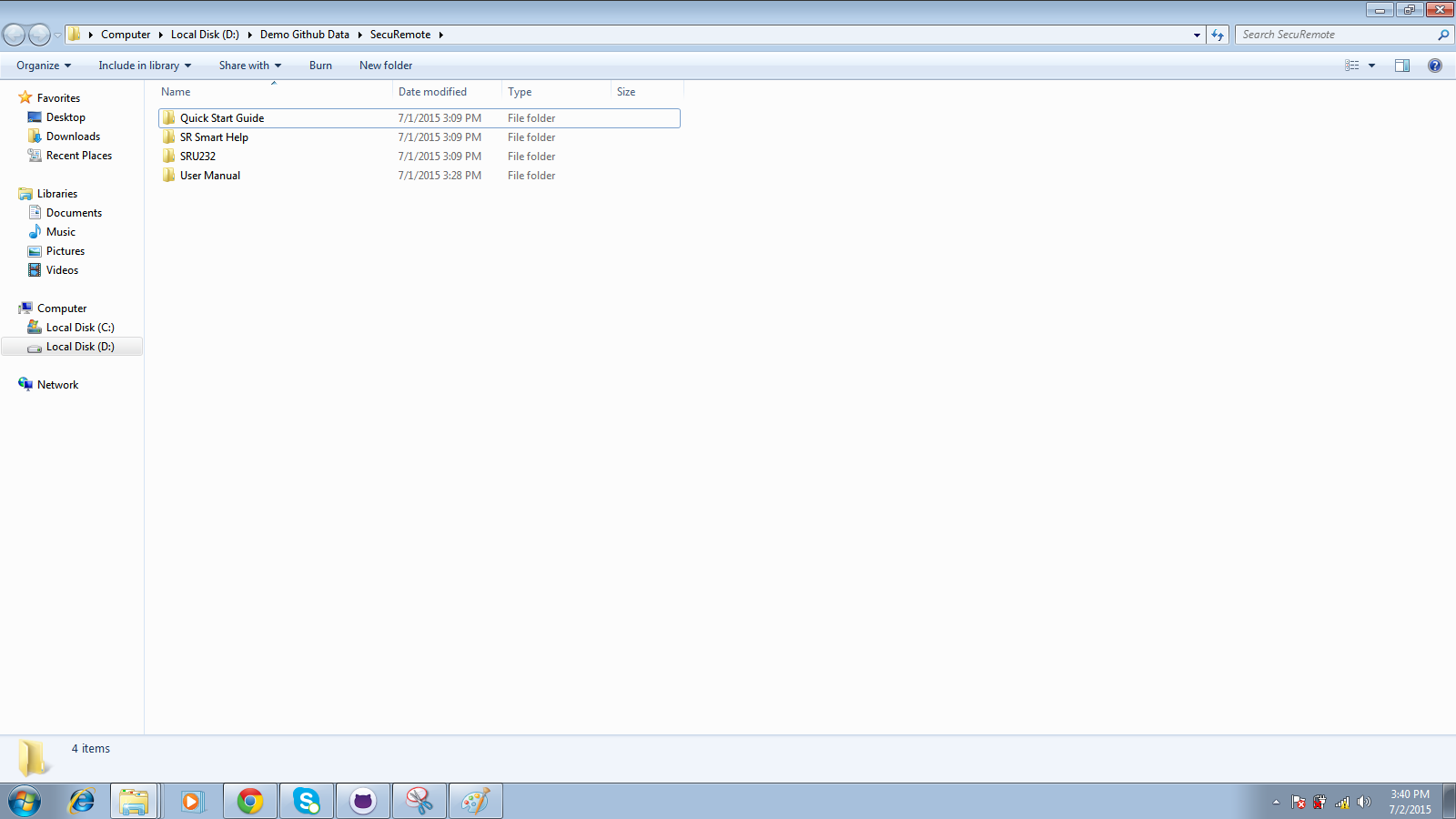
## Add new file/directory to repository

To add new file or directory to your repository,

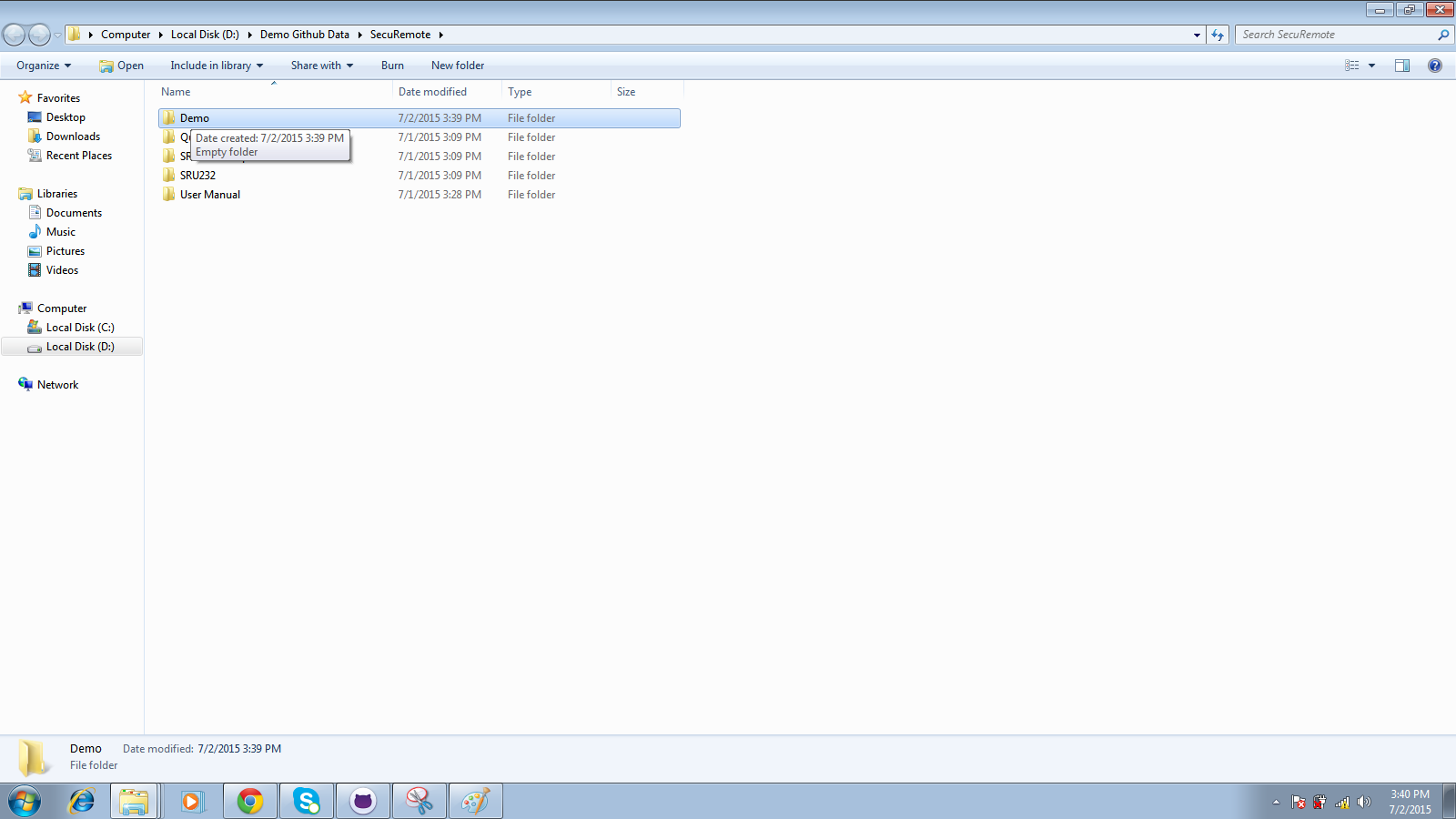
1. Right click on repository and click on “Open in Explorer”



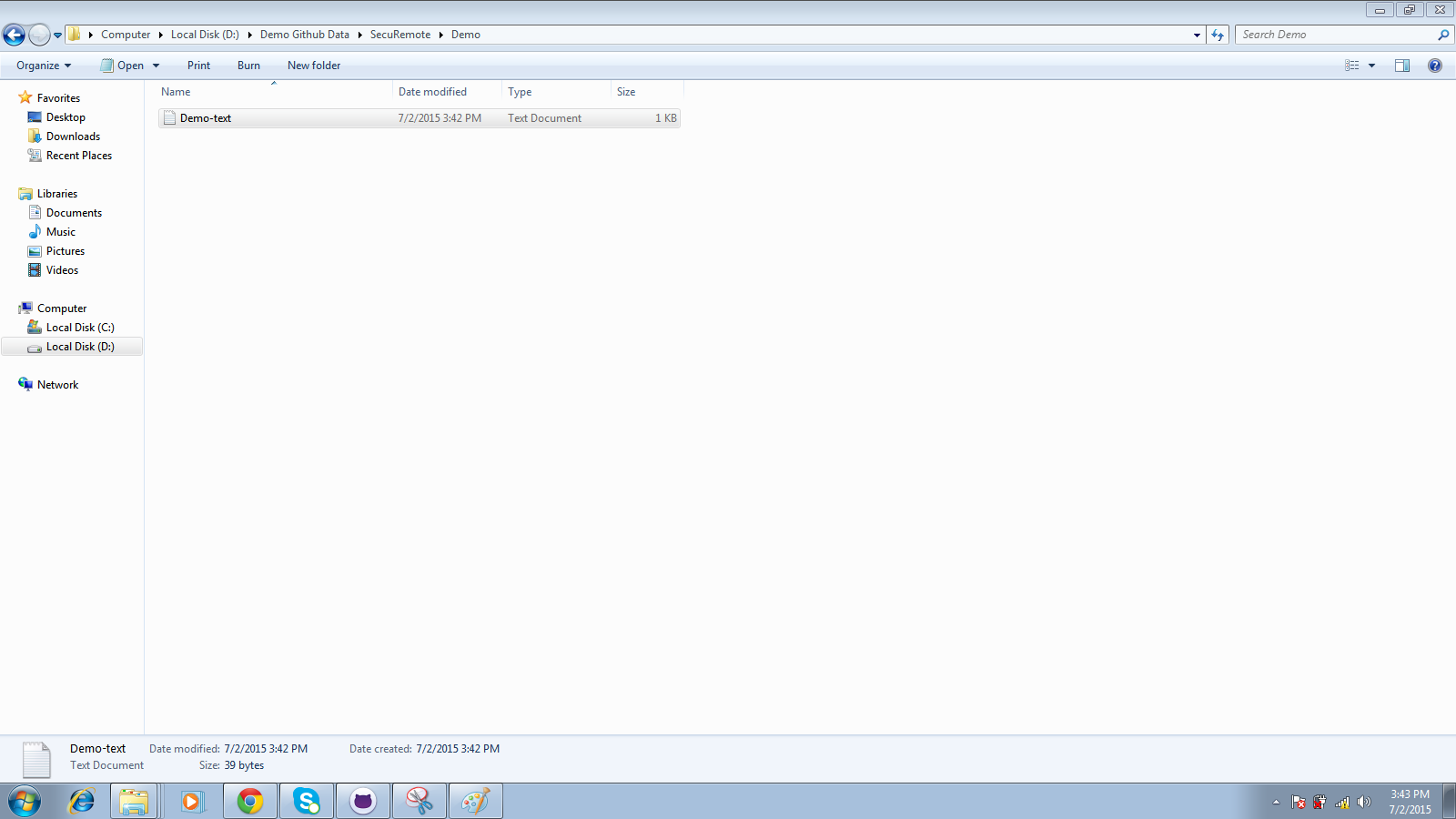
1. It will open physical location of repository on your local machine



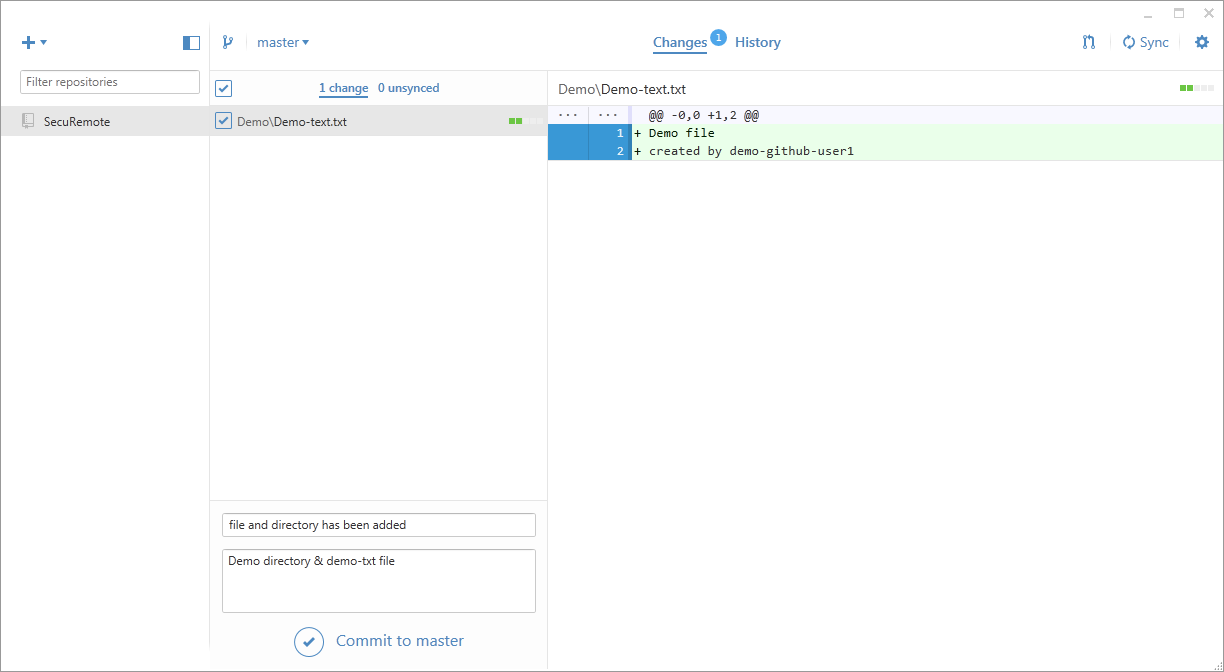
1. Create new folder/directory or can copy already exist directory that you want to add in repository



1. You can create new file inside directory. File can be of any type e.g. spreadsheet, doc file, notepad or any.

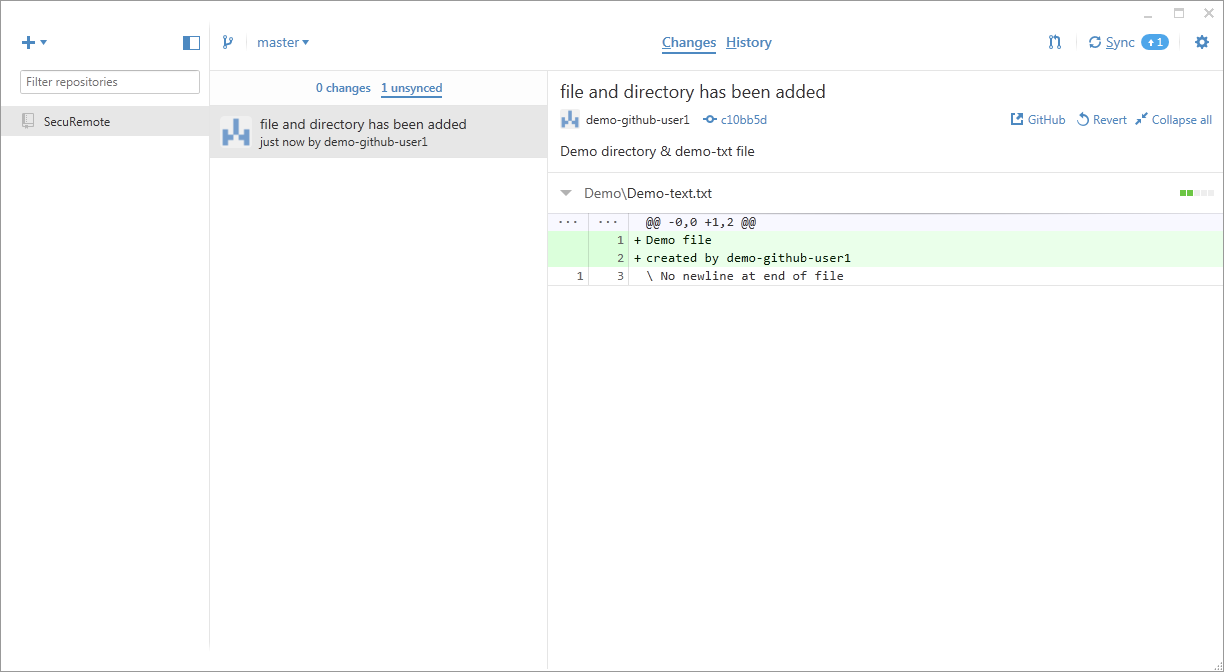


1. Once you have made any change in repository, you will see a notification of changes in GitHub Client



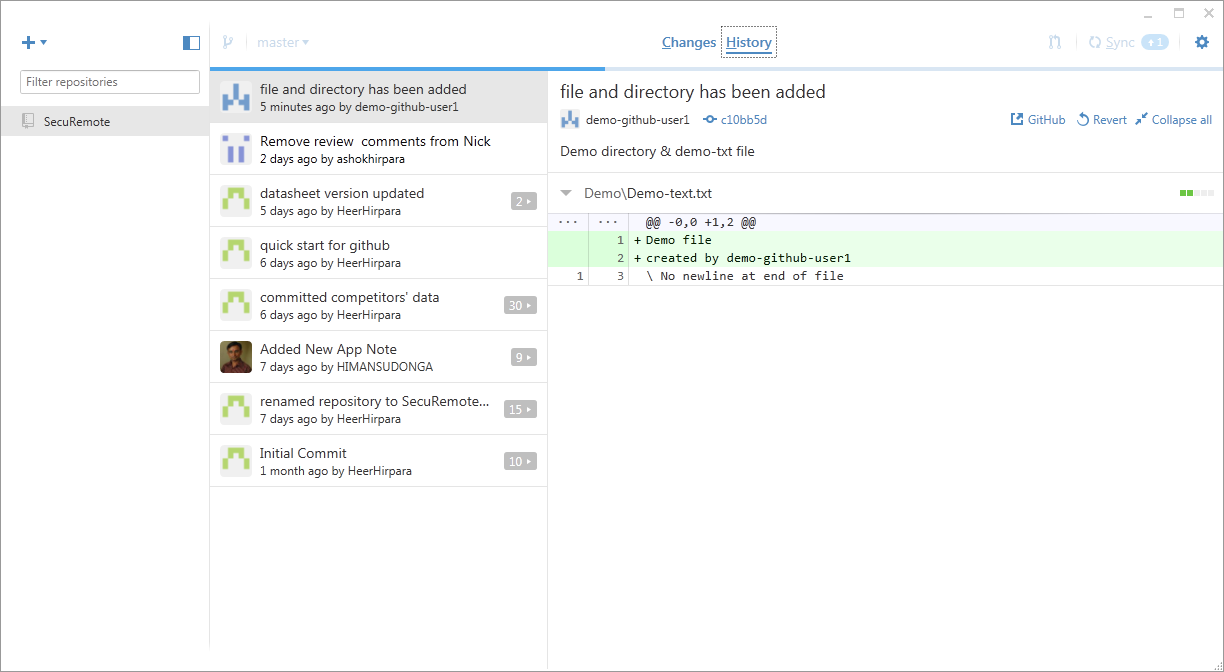
1. To commit these changes, type a commit summary (and, optionally, an extended description, if you want to provide more details of the changes you made) and click Commit.

At this point you'll notice that there are no longer any files to commit—instead, there’s now one unsynced commit, which you can view by clicking on the “Unsynced” header. This means that your changes have been committed with Git locally but not yet synced to your remote repository on GitHub.com.



1. When you're ready to share your changes, click the blue Sync button in the upper-right corner of the window to send those changes to your remote repository.

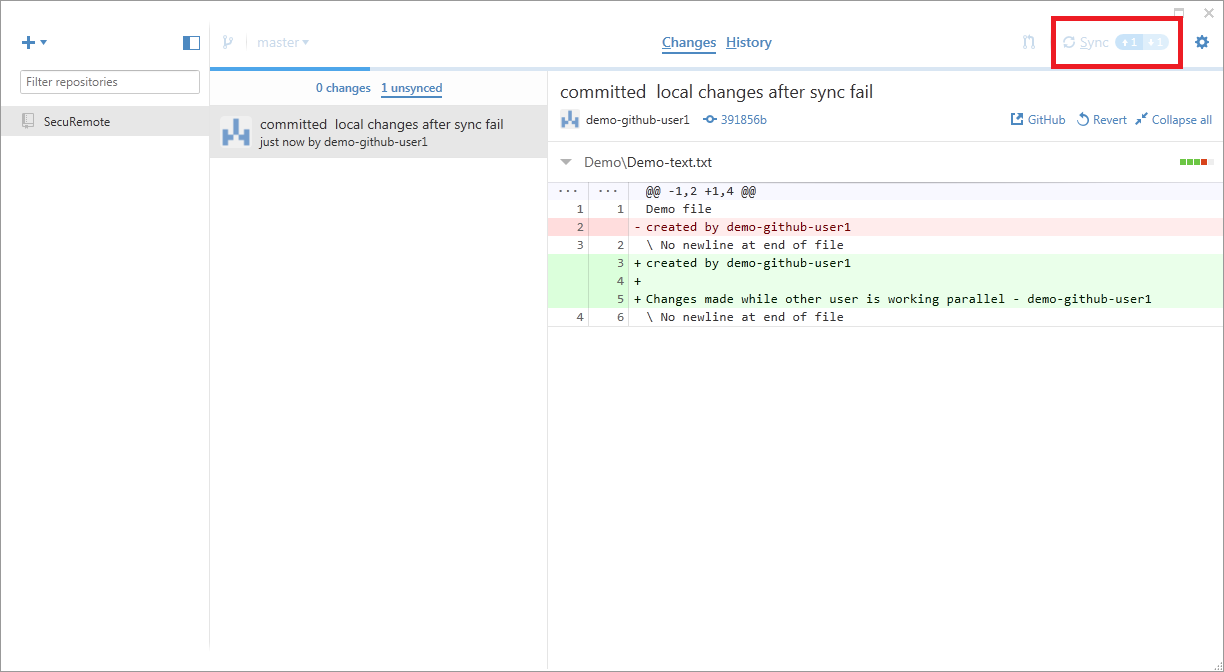
Sync will also pull in any changes made by others, so that you’re left with the latest version of all the files.



## Synchronizing repository

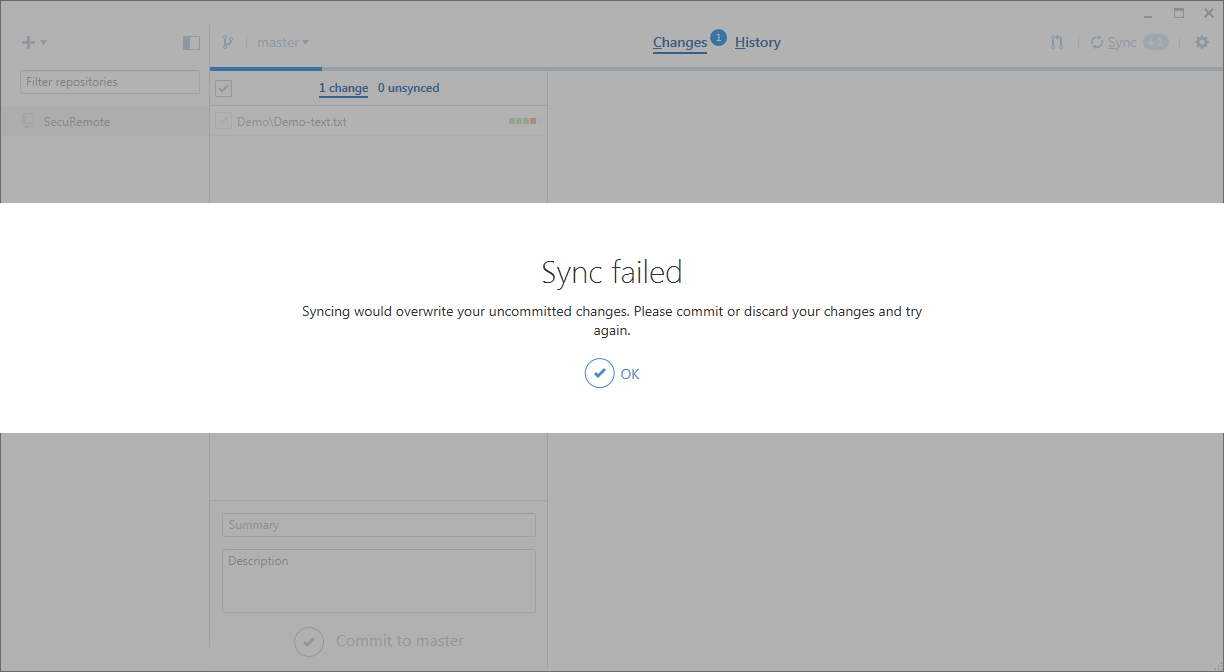
Whenever latest file is available on GitHub server, you will see blue *down-arrow* near to Sync button and if there is any local changes which are supposed to be shared on server, you will see blue *up-arrow* near to Sync button.

1. Click on Sync button to start the synchronizing.



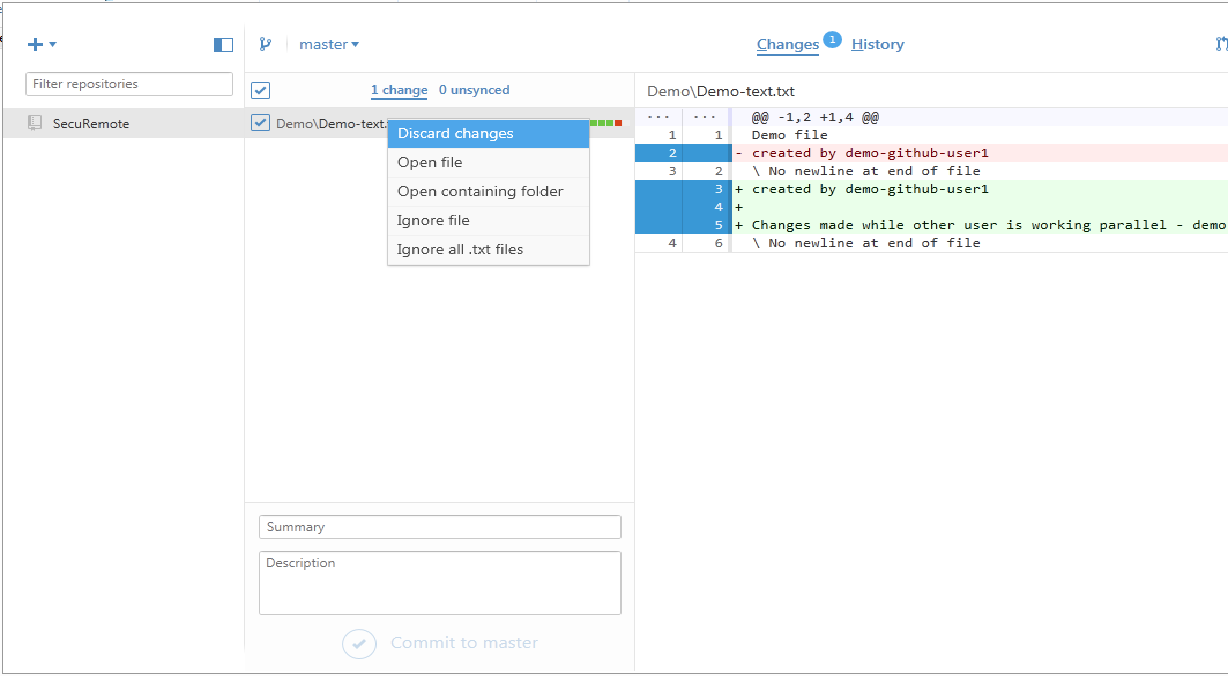
### Making changes to files while multiple users are working on it.

1. Whenever latest file is available on GitHub server (i.e. someone else has updated the file) you will see blue down-arrow near to Sync button.
2. At that time if you have made some changes which are not committed and you clicked on “Sync”, it will give an error.



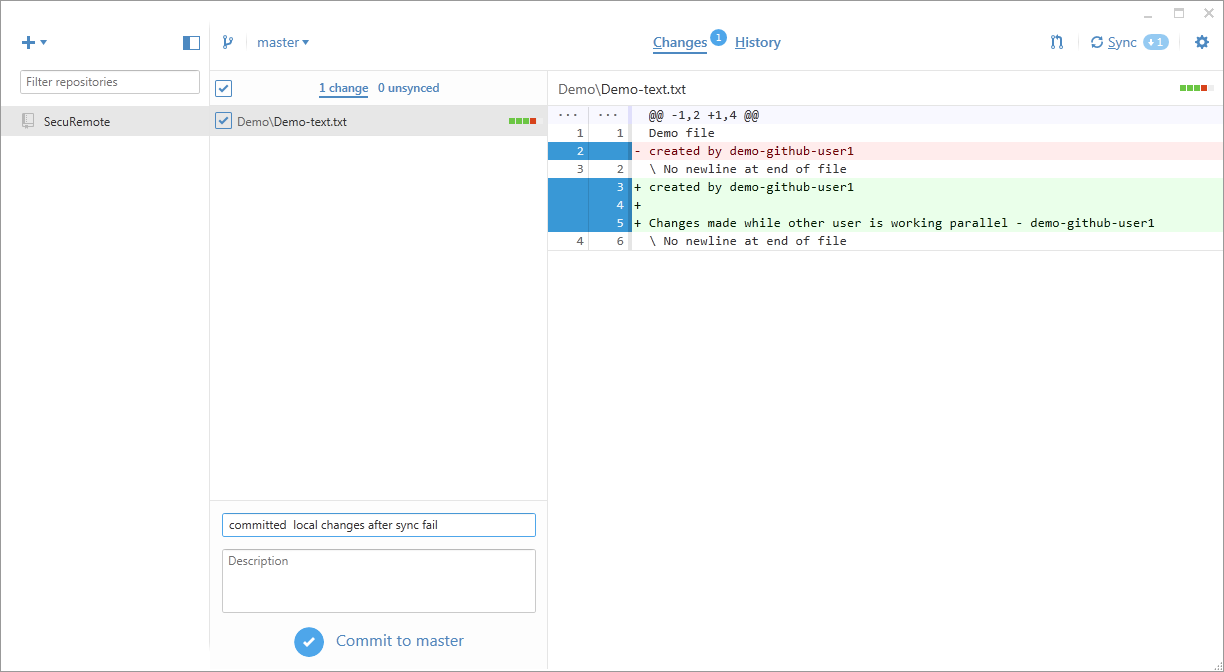
1. Here, you have two options.

First Option, Discard your changes if they are no longer required.

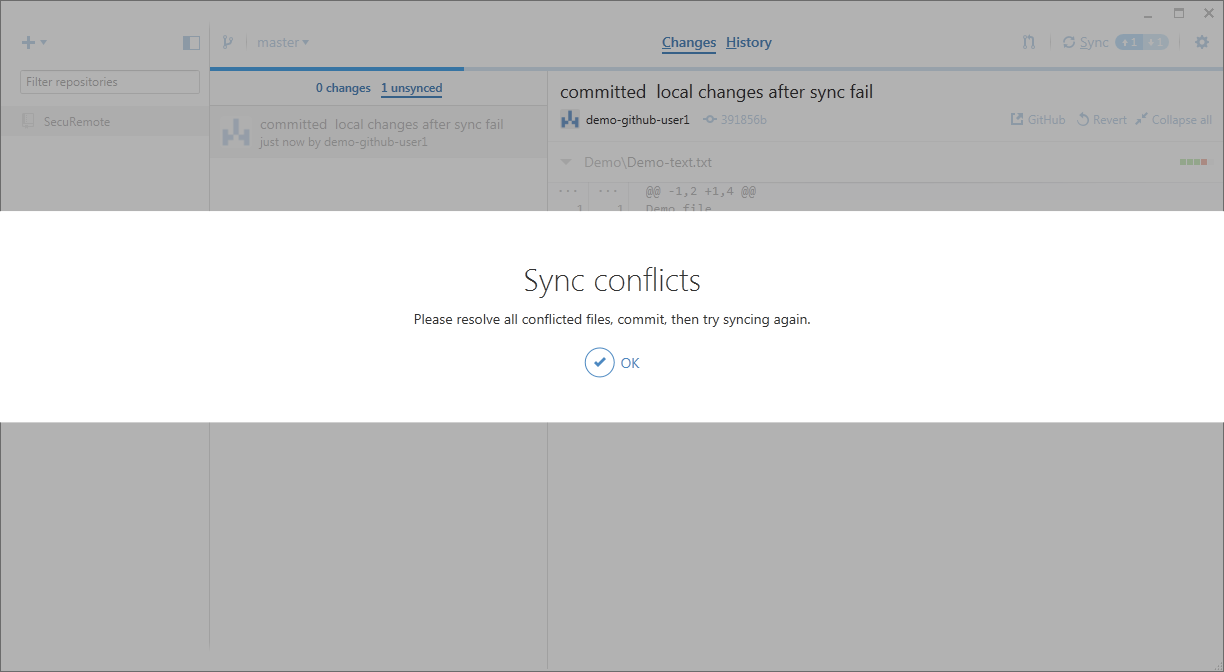


Second option, merge your changes to latest file by following below steps:

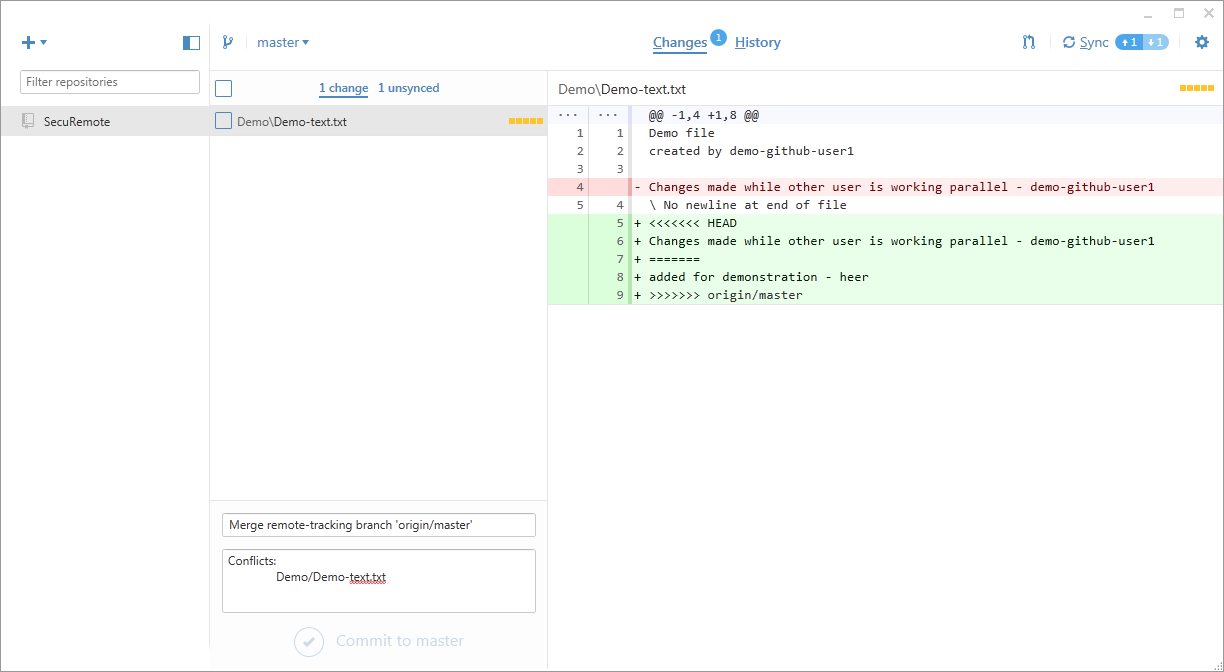
Commit your local changes.

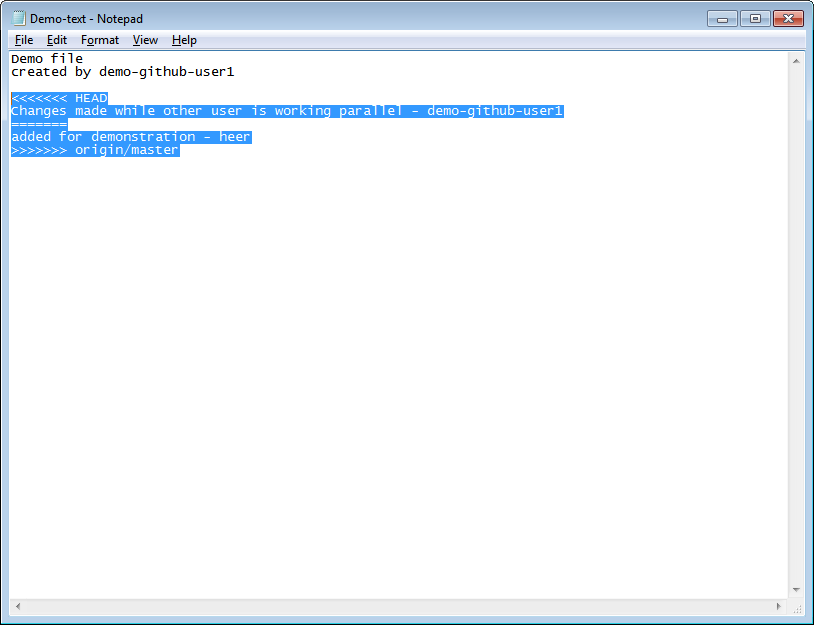


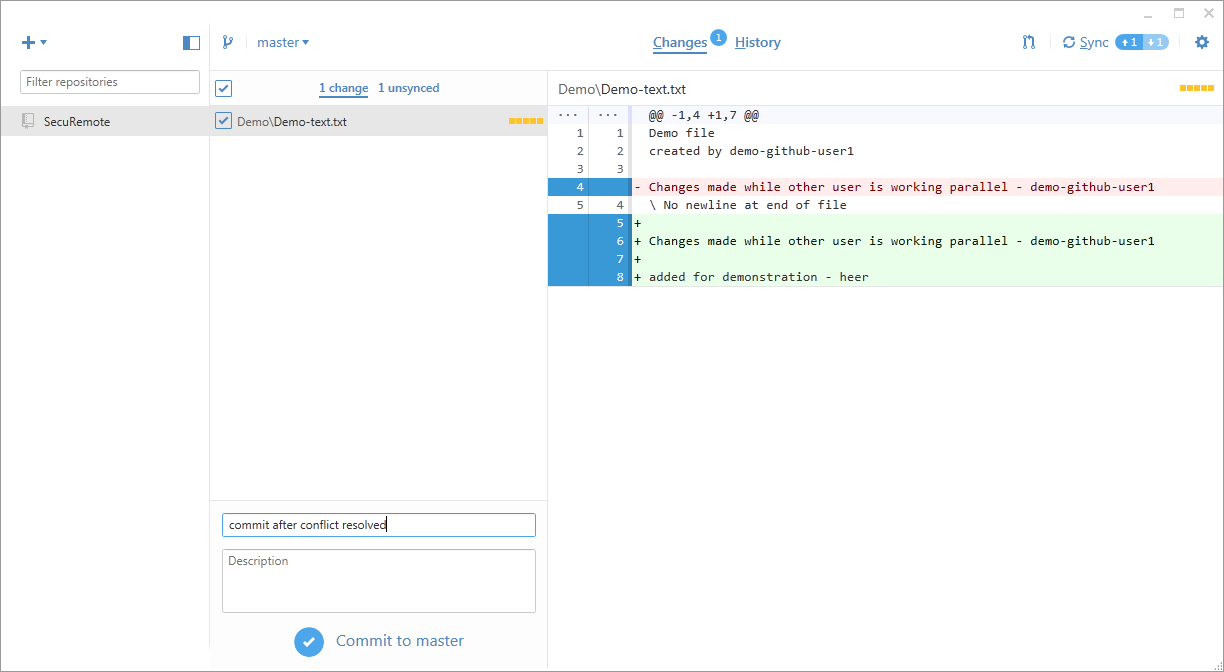
* + Sync with GitHub Server by clicking on “Sync” button. Here if there is no conflict with your changes & changes by other users, it will be merged automatically. If your changes are conflicting with changes from other users, it will give “Sync Error”



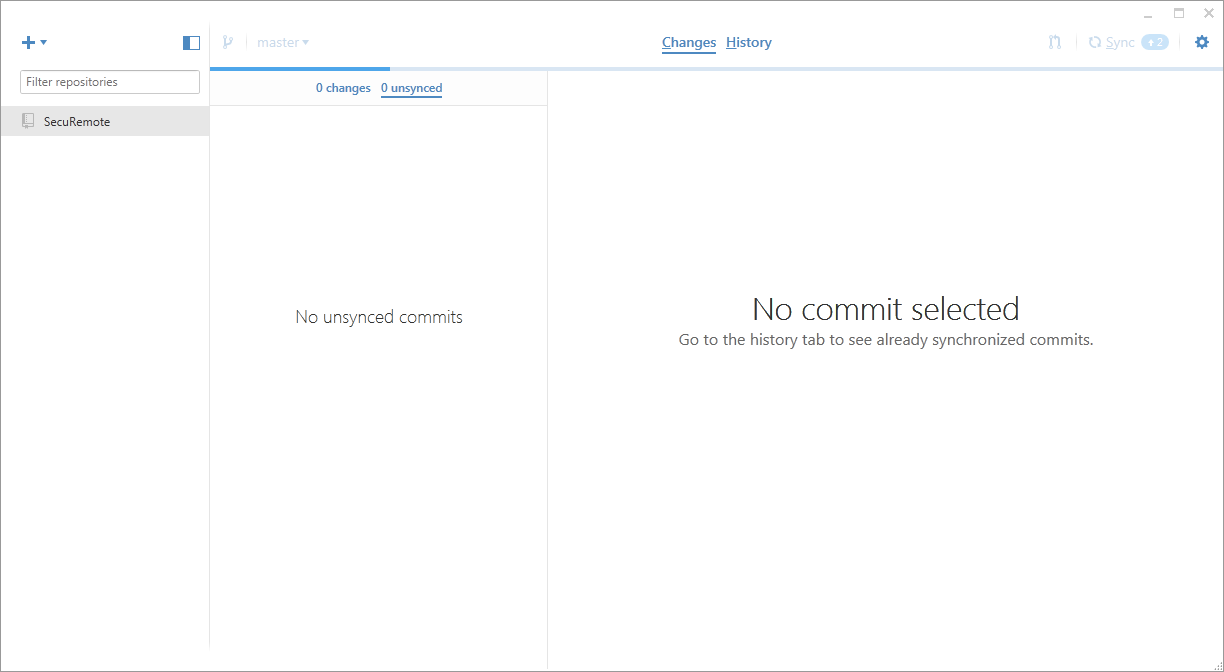
* + It will show you the conflicts in the file. The conflicted area will be started by “<<<<<<<< HEAD” notation and will end by “>>>>>>origin”. Both changes will be separated by “=======”.



* + You can view in file also
  + Commit after conflict has been resolved manually

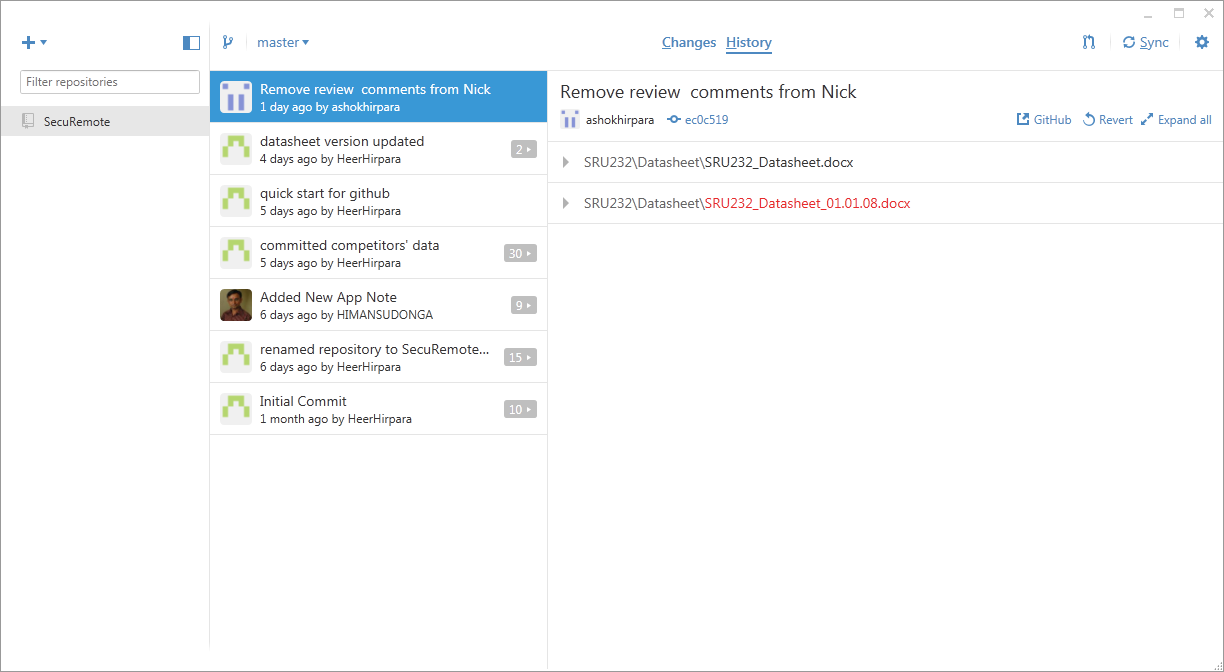


* + Synchronizing repository after committing changes and you’re done!!!

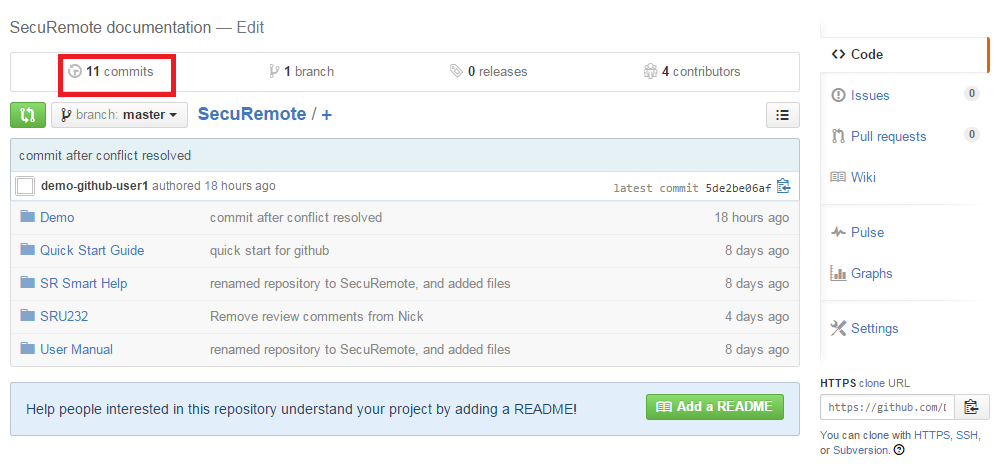


## Viewing History

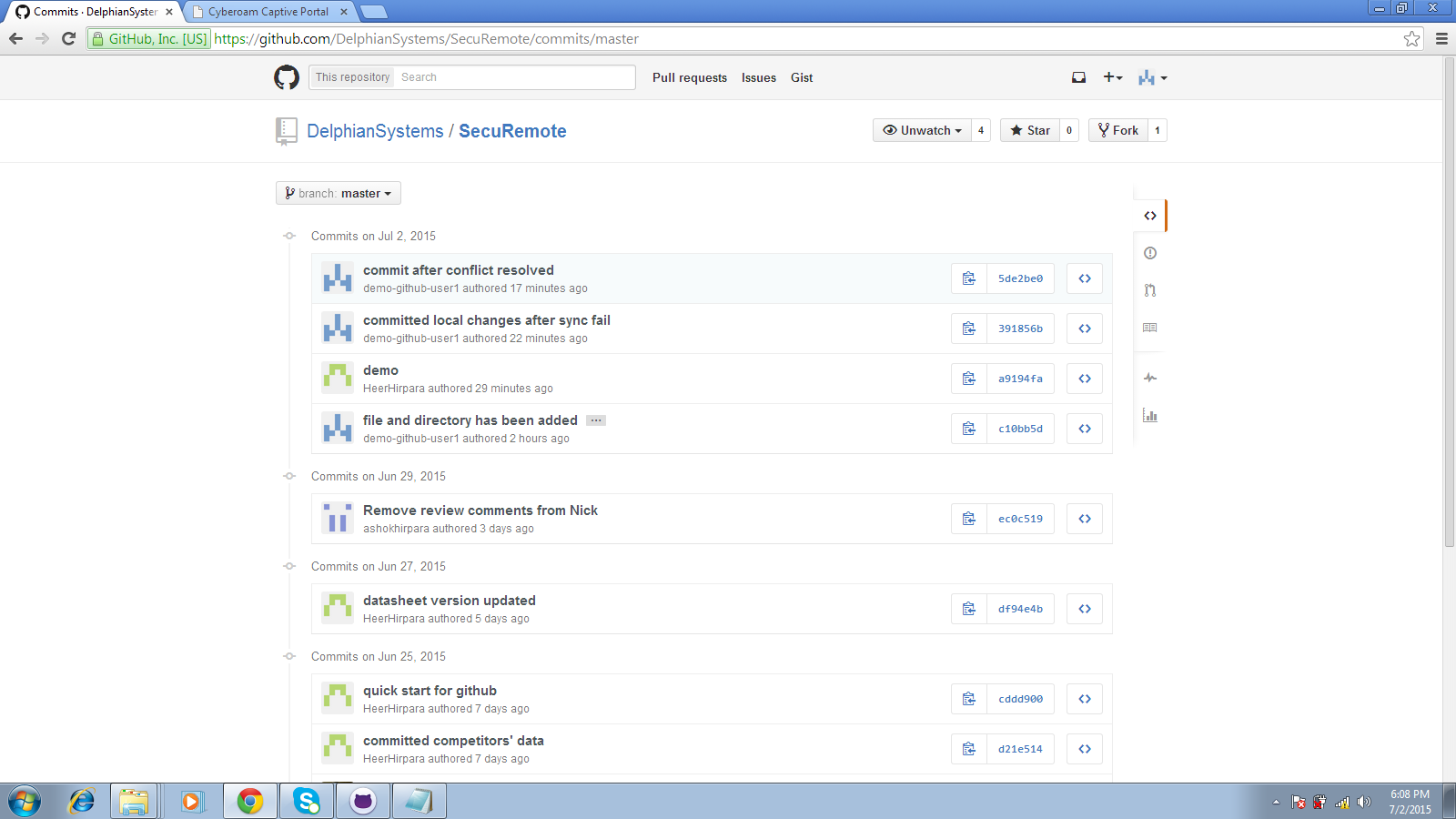
1. You can click on History to view the older changes that has been made to that branch.



1. In browser you can click on commit to see all the changes that has been made to repository till the date.



1. You can click on “<>” button to browse whole repository at that point.



# Getting started with GitHub for Delphian Repository Structure

* Any person can request to administrator at[heer.hirpara@delphiansystems.com](mailto:heer.hirpara@delphiansystems.com) for create GitHub account.
* Once account is created, admin will give the user id and password.
* Then admin can share folders to download along with privilege to read / write the content.

Even the permission can be assigned to user to upload content in folder.

* Once admin gives the user id and password, user can Sign in to [github.com](https://github.com/) with user id and password. Refer Figure #1.

<https://github.com/login>

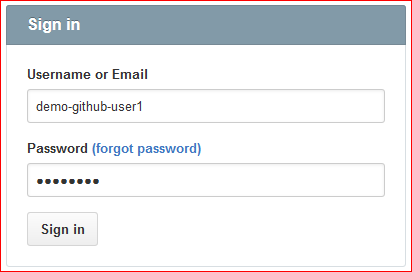


Figure #1

• Once done, you’ll see the welcome page with your user id. Refer Figure #2. <https://github.com/>

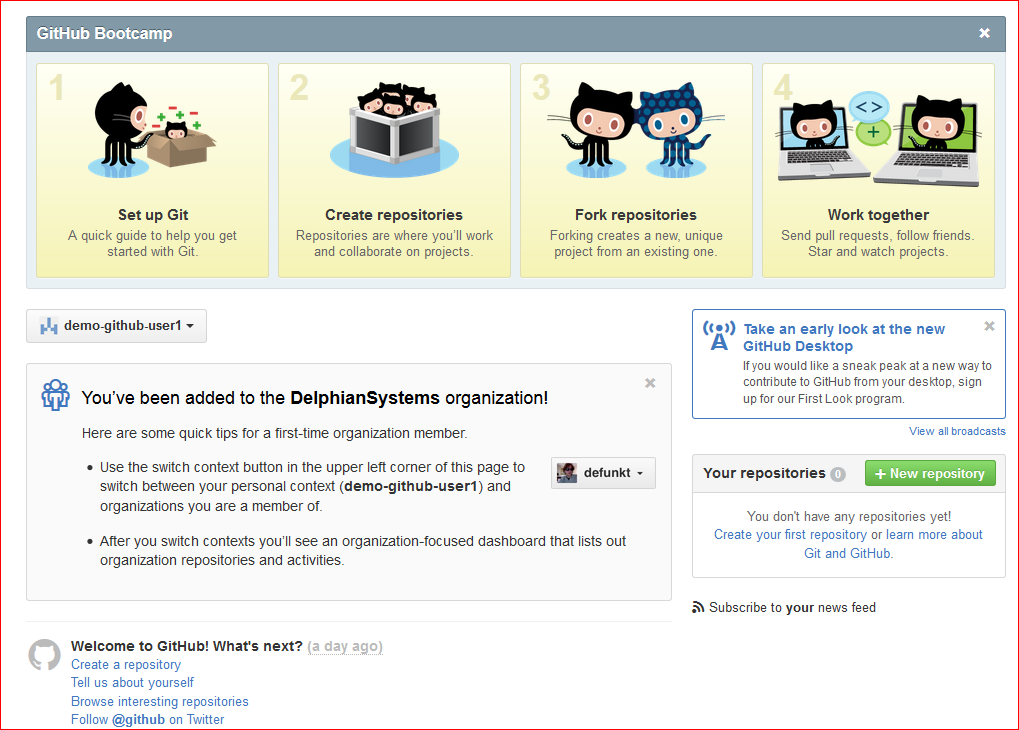


Figure #2

• Select “DelphianSystems” as an account from the Dashboard context. Refer figure #3.

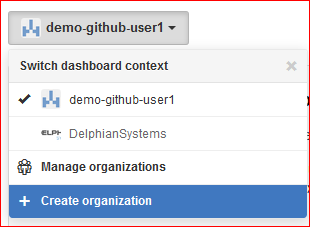


Figure #3

• “Welcome Page” will be display after selection of “Sciter 123” account. Refer Figure #4.

[https://github.com/orgs/DelphianSystems/dashboard](https://github.com/orgs/sciter123/dashboard)

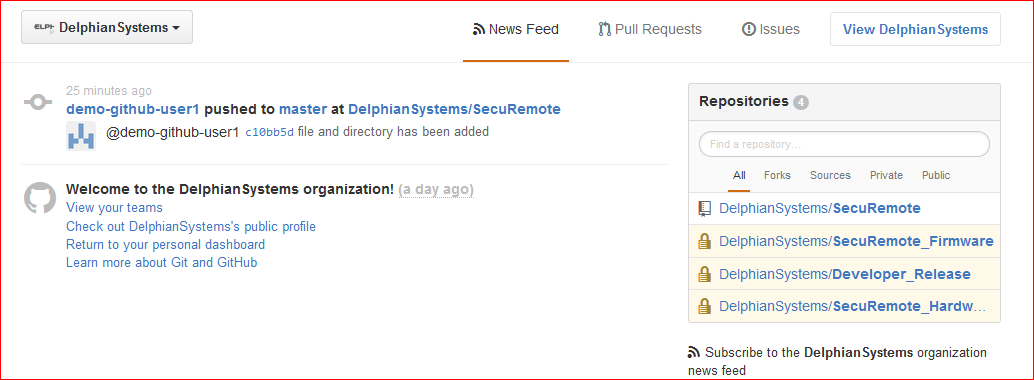


Figure #4

# File Structure for Hardware Data

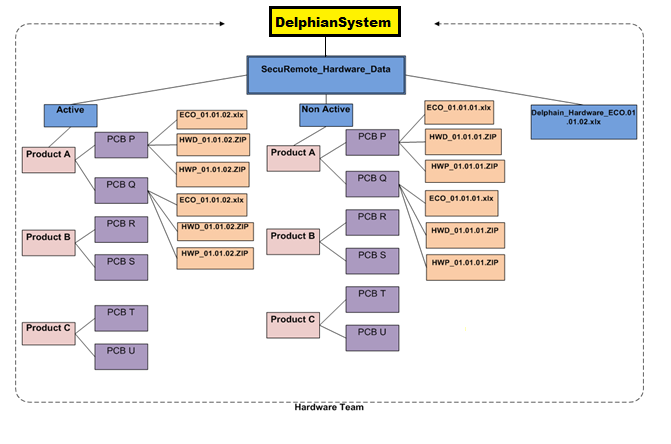


Figure #5

• **Note:** Here “DelphianSystem” is an organization account into GitHub. Refer Figure #5.

• “SecuRemote\_Hardware\_Data” is the Repository into the “DelphianSystem” organization account.

* This Repository has main three items:
* **Active folder**
* **Non Active folder**
* **Delphain\_Hardware\_ECO\_01.01.02.xlx**
* In Figure #5, “Active” and “Non Active” folder has main three Products: Product A, Product B and Product C.
* **“Active “folder contains only “New Release” files with updated version.**
* Product A has two folders: PCB P and PCB Q.
* PCB P has “Last Released” three files: ECO\_01.01.02.xlx, HWD\_01.01.02.ZIP and HWP\_01.01.02.ZIP. Select any require file to download and get it from “Downloads” folder.
* The same process is applicable for Product B and Product C folders too.

## 7.1 Retrieve Hardware data from Repository

* User can retrieve the files from the “Repositories”.
* Let us see live example: [https://github.com/DelphianSystems/SecuRemote\_Hardware\_Data](https://github.com/sciter123/SecuRemote_Hardware_Data)
* Find or select the Repository name from the listed Repositories. Refer Figure #6.

[https://github.com/orgs/DelphianSystems/dashboard](https://github.com/orgs/sciter123/dashboard)

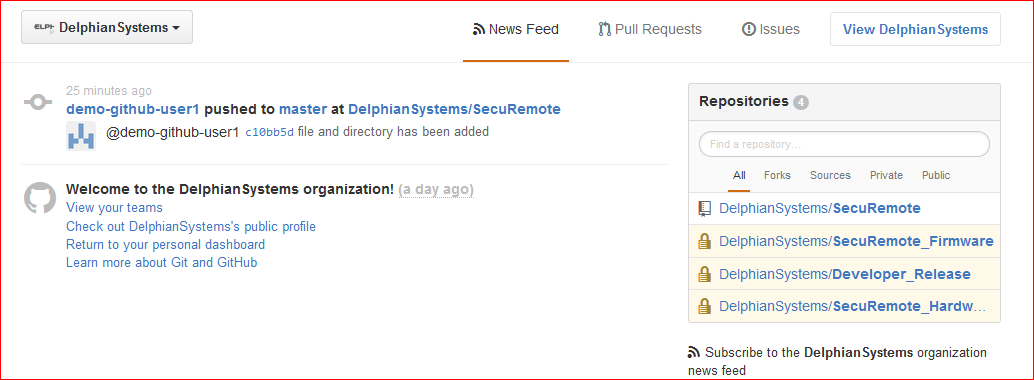


Figure #6

* After selecting the specific Repository from the list, it will display folders and files in the “DelphianSystems/SecuRemote\_Hardware\_Data” repository. Refer Figure #7. [https://github.com/DelphianSystems/SecuRemote\_Hardware\_Data](https://github.com/sciter123/SecuRemote_Hardware_Data)

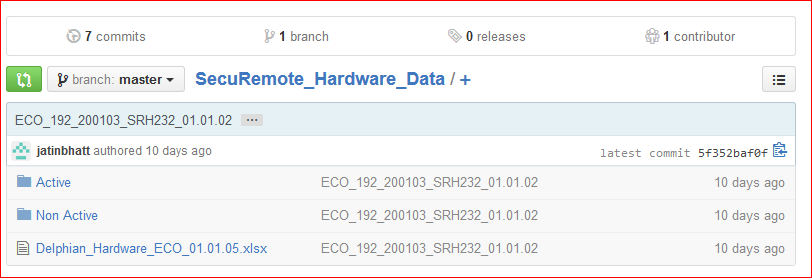


Figure #7

* Select “Active” folder. Refer #8. [https://github.com/DelphianSystems/SecuRemote\_Hardware\_Data/tree/master/Active](https://github.com/sciter123/SecuRemote_Hardware_Data/tree/master/Active)

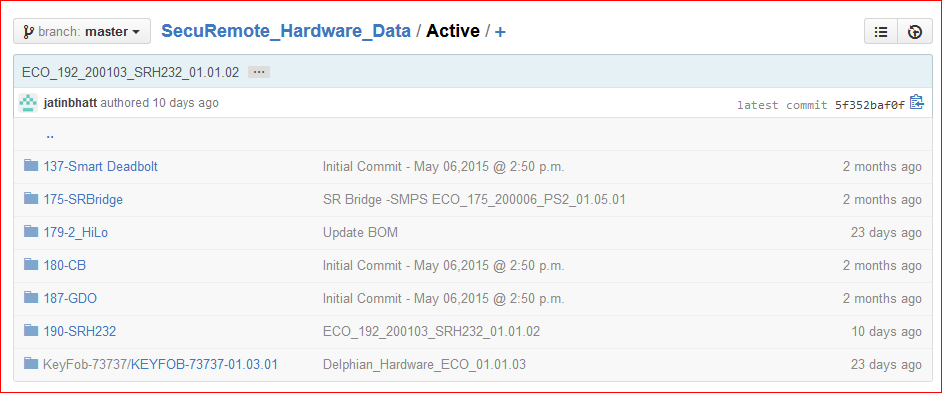


Figure #8

* Select “179-2\_HILo” folder. Refer Figure #9. <https://github.com/DelphianSystems/SecuRemote_Hardware_Data/tree/master/Active/179-2_HiLo>

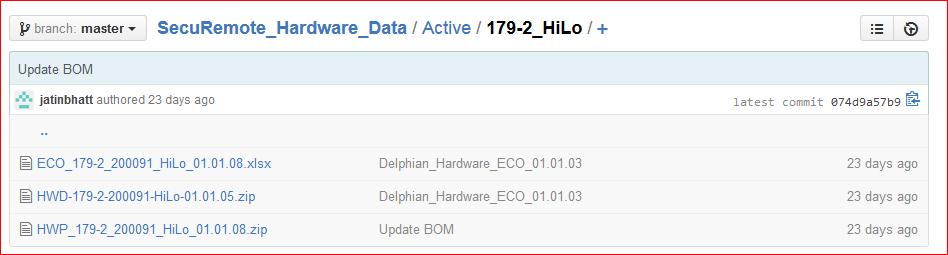


Figure #9

* Select any file out of three. For example “HWD-179-2-200091-HiLo-01.01.05.zip“. Refer Figure #10. <https://github.com/DelphianSystems/SecuRemote_Hardware_Data/blob/master/Active/179-2_HiLo/HWD-179-2-200091-HiLo-01.01.05.zip>

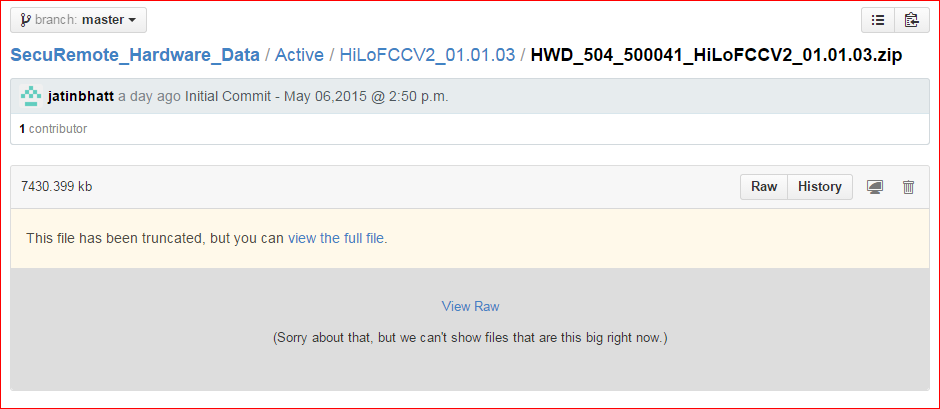


Figure #10

* Click on “View Raw”. It will download the selected .ZIP file.
* Above same process can be done in **“Non Active” folder but the difference is it contains old version of files.**
* In Figure #11, Non Active folder has the same product files as Active folder but it archives old version of contents.
* Let us see live example: <https://github.com/DelphianSystems/SecuRemote_Hardware_Data/tree/master/Non%20Active>

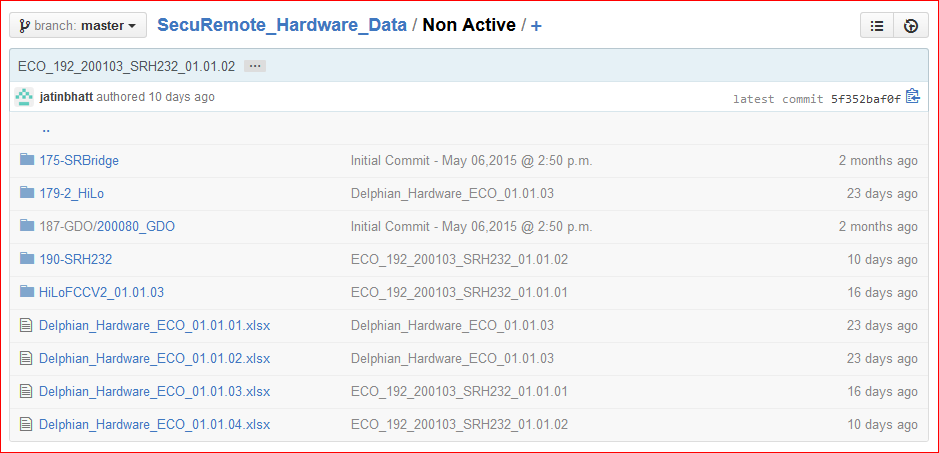


Figure #11

* Select “179\_HiLo” folder. Refer Figure #12. <https://github.com/DelphianSystems/SecuRemote_Hardware_Data/tree/master/Non%20Active/179-2_HiLo>



Figure #12

* Select “HWD-179-2-200091-HiLo-01.01.04.zip” folder. Refer Figure #13. <https://github.com/DelphianSystems/SecuRemote_Hardware_Data/blob/master/Non%20Active/179-2_HiLo/HWD-179-2-200091-HiLo-01.01.04.zip>



Figure #13

* Click on “View Raw”. It will download the selected file. Get that file from the Downloads folder.

**Note:**

* User can also download the all folders “Active”, “Non Active”, and “Delphain\_Hardware\_ECO\_01.01.02.xlx” into .ZIP file from Click on the “Download ZIP” Button from bottom right corner. Refer Figure #15. <https://github.com/DelphianSystems/SecuRemote_Hardware_Data>

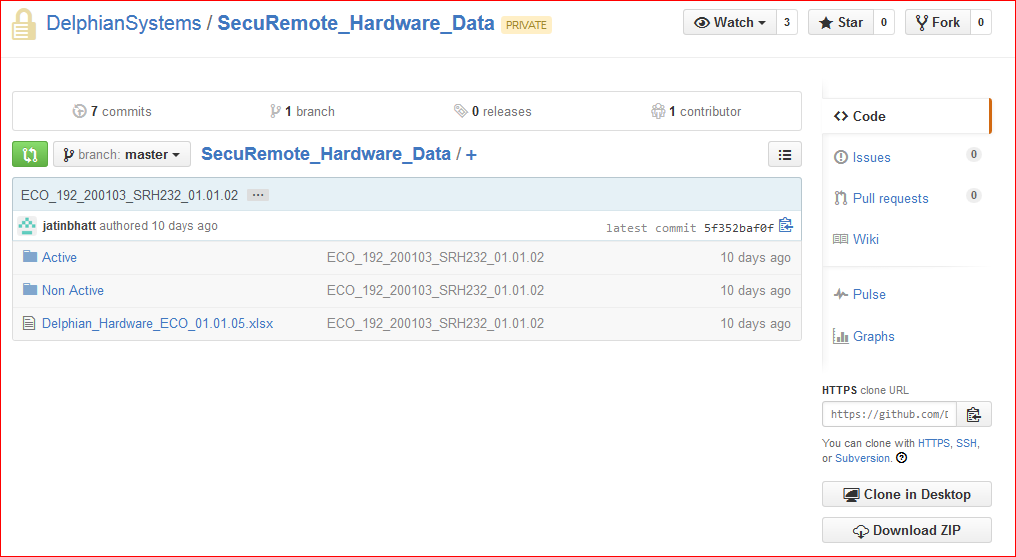


Figure #14

* It will download the all folders into .ZIP file. Get that file from the Downloads folder.

# File Structure for Firmware data

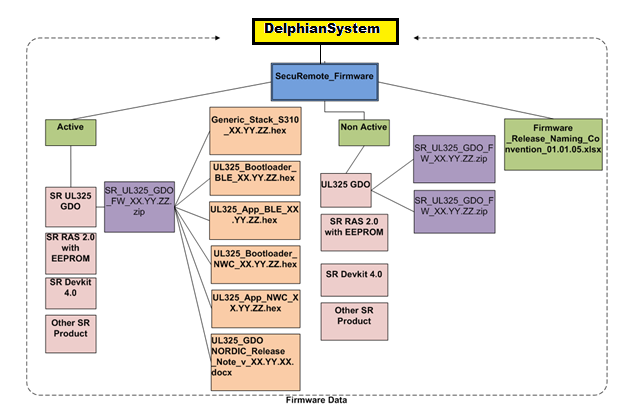


Figure #15

* **Note:** Here, “DelphianSystem” is an organization account into GitHub. Refer Figure #15.
* “SecuRemote\_Firmware” is the other Repository into the “DelphianSystem” organization account.
* This Repository has main three items:
* **Active folder**
* **Non Active folder**
* **Firmware\_Relaease\_Naming\_Convention\_01.01.05.xlsx**
* In Figure 15, “Active” and “Non Active” folder has main four Products: SR Ul325 GDO, SR RAS 2.0 with EEPROM, SR Devkit 4.0 and Other SR Product.
* **“Active” folder contains only “New Release” files with updated version.**
* Suppose SR UL325 GDO has “SR\_UL325\_GDO\_FW\_xx.yy.zz.zip” file with new version in Active folder.
* “SR\_UL325\_GDO\_FW\_xx.yy.zz.zip” contains six other files which are related with the SR UL325 GDO.
* Same as its applicable for other product.
* It also applicable for Non Active folder. But Non Active folder contains only old version files.

## 8.1 Retrieve Firmware data from Repository

• User can retrieve the files from the “Repositories”.

• Let us see live example: <https://github.com/DelphianSystems>. Refer Figure #16.

• Find or select the Repository name from the list. Refer Figure #15.

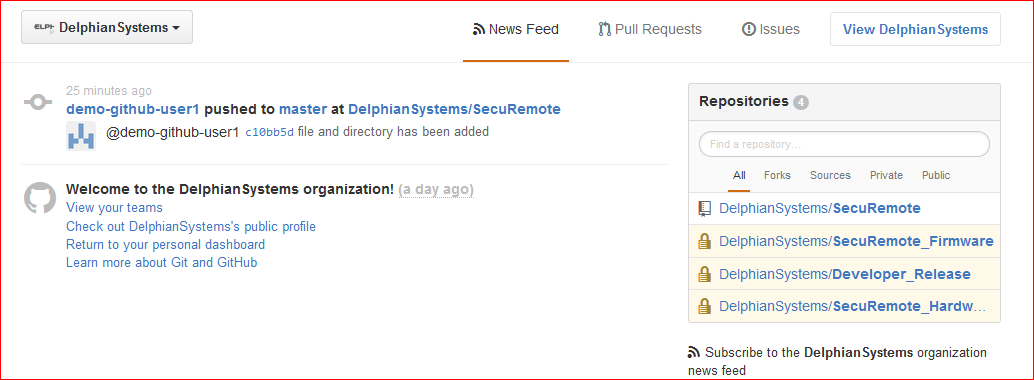


Figure #16

* Now select the “SecuRemote\_Firmware” Repository from the listed Repository. Refer Figure #16.
* After select specific Repository then it will display the three folders of that Repository. Refer Figure #18.

<https://github.com/DelphianSystems/SecuRemote_Firmware>

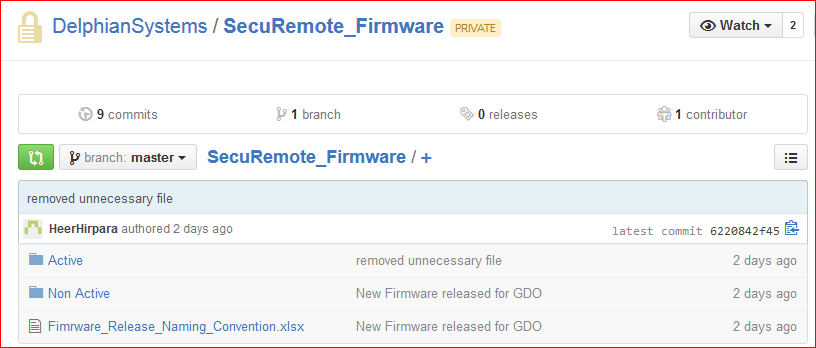


Figure #17

* Select “Active” folder from the list.
* It will display the folders for specific product. Refer Figure #18. [https://github.com/DelphianSystems/SecuRemote\_Firmware/tree/master/Active](https://github.com/sciter123/SecuRemote_Firmware/tree/master/Active)

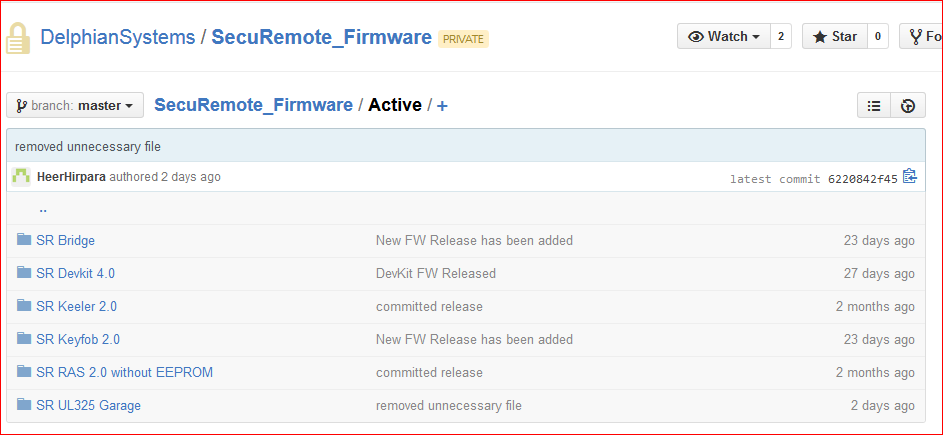


Figure #18

* Select any of them which user wants to get the information from that.
* For Example, select the first file from the list. Refer Figure #19. [https://github.com/DelphianSystems/SecuRemote\_Firmware/tree/master/Active/**SR**%20Keeler%202.0](https://github.com/DelphianSystems/SecuRemote_Firmware/tree/master/Active/SR%20Keeler%202.0)

Figure #19

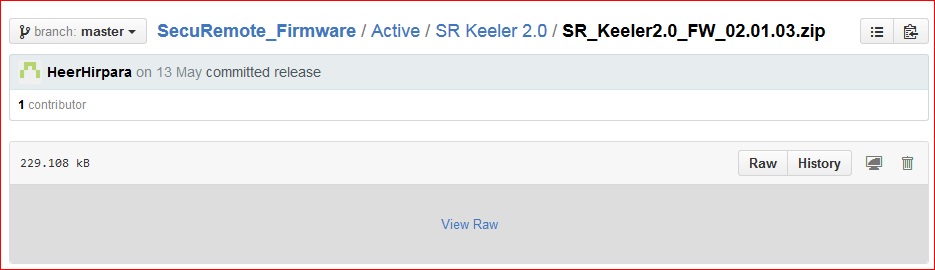
* Select .ZIP file for download. [https://github.com/DelphianSystems/SecuRemote\_Firmware/blob/master/Active/SR%20Keeler%202.0/SR\_Keeler2.0\_FW\_02.01.03.zip](https://github.com/sciter123/SecuRemote_Firmware/blob/master/Active/SR%20Keeler%202.0/SR_Keeler2.0_FW_02.01.03.zip)
* Click on “View Raw” and it will download the selected file. Refer Figure #20.

Figure #20

* Get that file from the Downloads folder.
* Same as **“Non Active” folder but the difference is that, in “Non Active” folder contains old version files.**

**Note:** User can also download the all folders “Active”, “Non Active”, and “Firmware\_Relaease\_Naming\_Convention\_01.01.05.xlsx” into .ZIP file from Click on the “Download ZIP” Button from bottom right corner. Refer Figure #21.

[https://github.com/DelphianSystems/SecuRemote\_Firmware](https://github.com/sciter123/SecuRemote_Firmware)

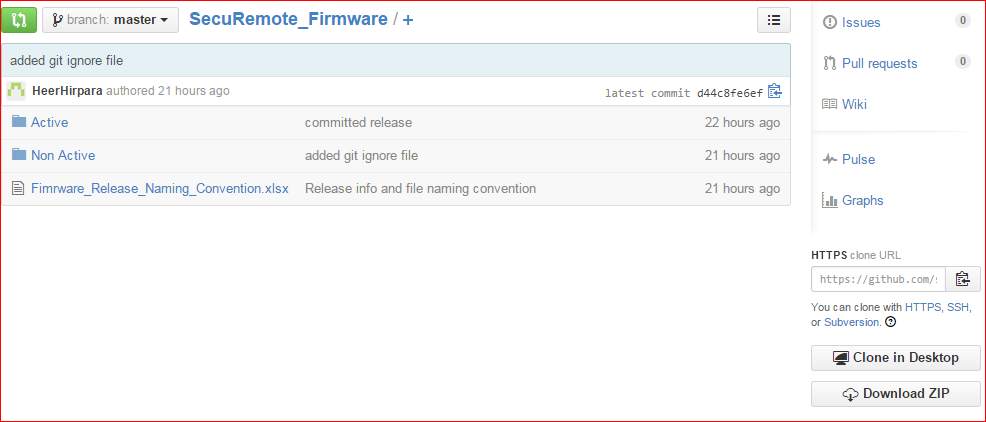


Figure #21

* It will download the .ZIP file for all folders.
* Get that file from the Downloads folder.

# Developer Release

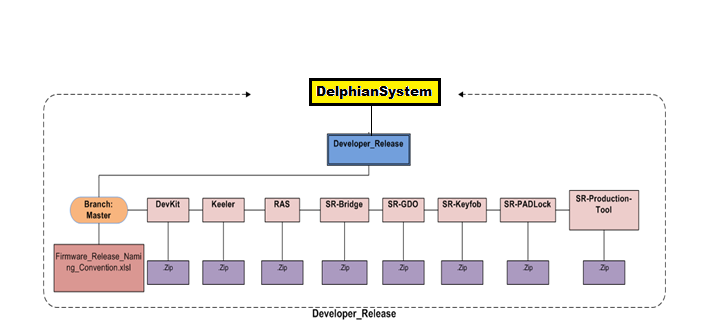


Figure #22

* **Note:** Here, “DelphianSystem” is an organization account into GitHub. Refer Figure #22.
* “Developer\_Release” is the other Repository into the “DelphianSystem” org.
* Here one Branch is there “Branch: Master”.
* “Branch: Master” has one common file “Firmware\_Release\_Naming\_convention.xlsl”. Which is common file for Release Information.
* Other files are in branches: DevKit, Keeler, RAS, SR-Bridge, SR-GDO, SR - PADLock, SR -Production-Tool.
* All files has sub files which have latest release data in .ZIP file.

## 9.1 Retrieve Developer\_ Release data from Repository

1. User can retrieve the files from the “Repositories”.
2. Let us see live example: [https://github.com/orgs/DelphianSystems/dashboard](https://github.com/orgs/sciter123/dashboard). Refer Figure #24.
3. Select the Repository name “Developer\_Release” from the list. Refer Figure #24.

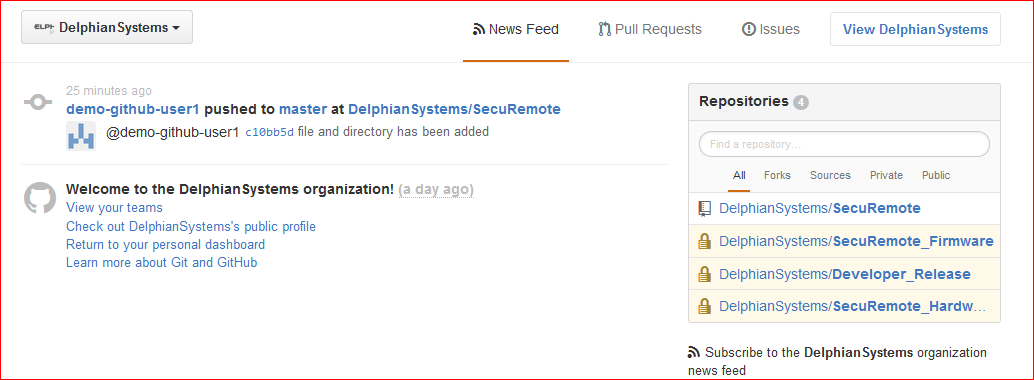


Figure #23

* Click on “Branch: Master”. Refer Figure #25.

<https://github.com/DelphianSystems/Developer_Release/tree/master>

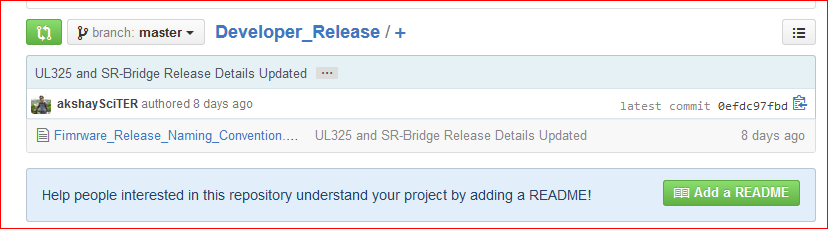


Figure #24

* Select Require File from the list. Refer Figure #25.

<https://github.com/DelphianSystems/Developer_Release/tree/Keeler>

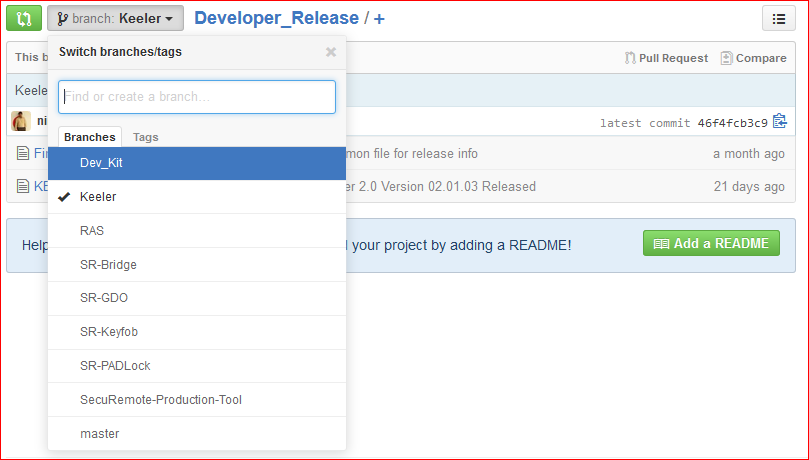


Figure # 25

* For Example: If user select Keeler from the list then there are two files is there. Refer figure #26.
* One is “Firmware\_Release\_Naming\_Convention.xlsx”.
* Other are .ZIP files which is for New Released Version files.
* **Note:** “Firmware\_Release \_Naming\_Convention.xlsx” File is common file for Release information.

<https://github.com/DelphianSystems/Developer_Release/tree/Keeler>

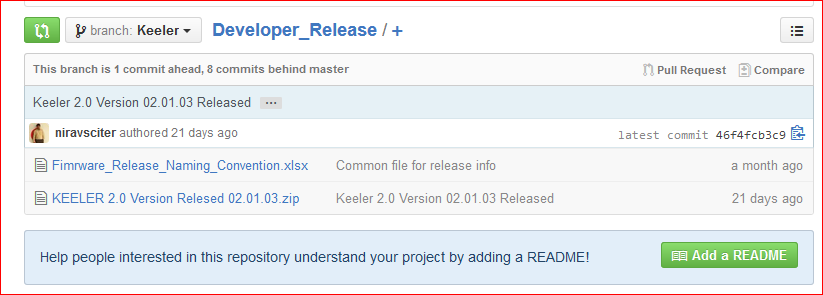


Figure # 26

* Click on required file from the list.
* Here, selected file is “Firmware\_Release\_Naming\_Convention.xlsx” file.

<https://github.com/DelphianSystems/Developer_Release/blob/Keeler/Fimrware_Release_Naming_Convention.xlsx>

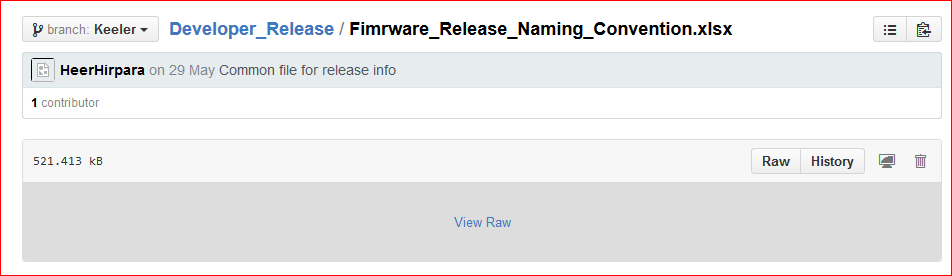


Figure # 27

* Click on “View Raw” and it will start the download that file. Refer Figure #27.
* Get that file from the Downloads folder.
* Otherwise if user can download the all files then click on “Download ZIP”. Refer Figure #28.

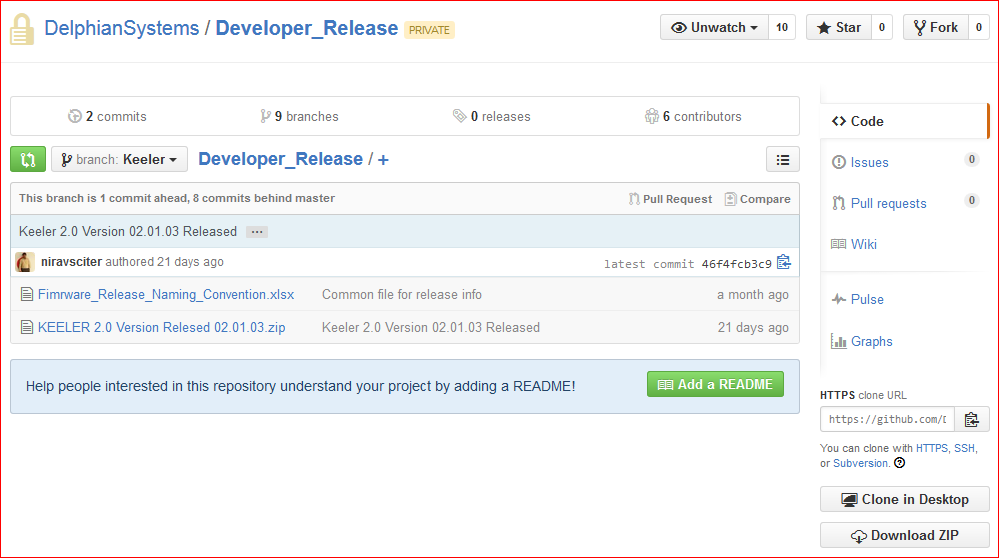


Figure # 29

* It will download all the files into .ZIP from selected Branch list.
* Get those files from the Download folder.