|  |
| --- |
|  |
| Quick Start Guide for GitHub |
| Version 01.01.09 |



**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 |

# Table of Contents

[1. Purpose 4](#_Toc422218703)

[2. Definition(s) and Abbreviation(s) 5](#_Toc422218704)

[3. Reference(s) 5](#_Toc422218705)

[4. Overview 6](#_Toc422218706)

[4.1 What is GitHub? 6](#_Toc422218707)

[4.2 Why use GitHub? 6](#_Toc422218708)

[4.3 How it works? 6](#_Toc422218709)

[5. Getting started with GitHub 7](#_Toc422218710)

[6. File Structure for Hardware Data 9](#_Toc422218711)

[6.1 Retrieve Hardware data from Repository 10](#_Toc422218712)

[7. File Structure for Firmware data 14](#_Toc422218713)

[7.1 Retrieve Firmware data from Repository 15](#_Toc422218714)

[8. File Structure for Developer Release 18](#_Toc422218715)

[8.1 Retrieve Developer\_ Release data from Repository 19](#_Toc422218716)

# Purpose

* The purpose of this document is to understand the features of GitHub.
* As a developer working collaboratively in a team, it is essential to maintain the version control by GitHub repository
* 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** |
|  |  |
|  |  |
|  |  |

Table #1

# Reference(s)

|  |  |
| --- | --- |
| **Document No.** | **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 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.

# Getting started with GitHub

* Any person can request to admin 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>

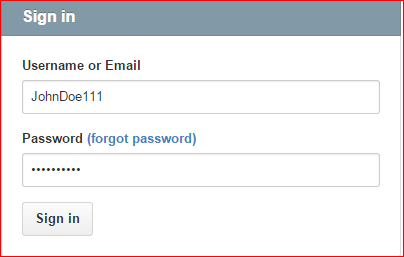


Figure #1

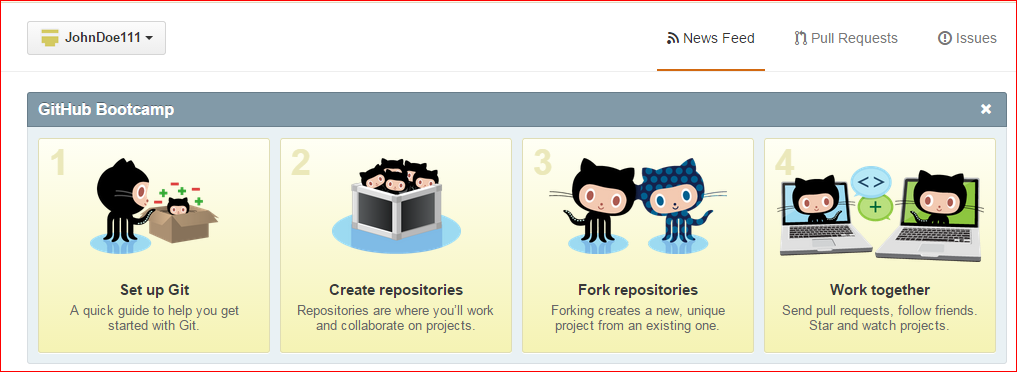
• Once done, you’ll see the welcome page shown below. Refer Figure #2. 

Figure #2

• Select “Sciter123” 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/sciter123/dashboard>

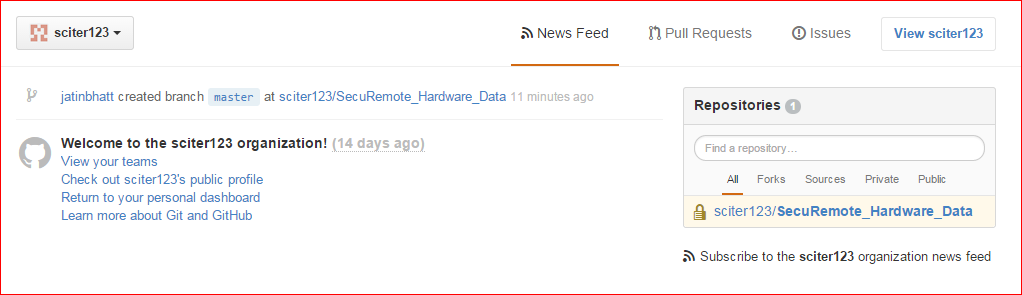


Figure #4

# File Structure for Hardware Data

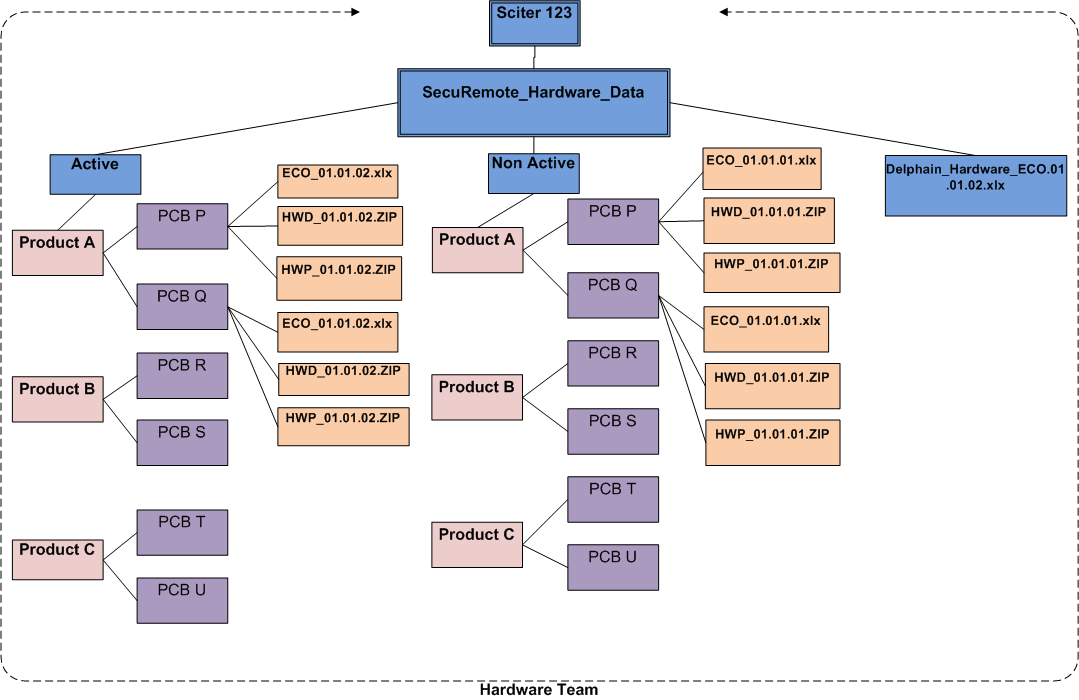


Figure #5

• **Note:** Here “Sciter 123” is an organization account into GitHub. Refer Figure #5.

• “SecuRemote\_Hardware\_Data” is the Repository into the “Sciter 123” 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.

## 6.1 Retrieve Hardware data from Repository

* User can retrieve the files from the “Repositories”.
* Let us see live example: <https://github.com/sciter123/SecuRemote_Hardware_Data>
* Find or select the Repository name from the listed Repositories. Refer Figure #6.

<https://github.com/orgs/sciter123/dashboard>

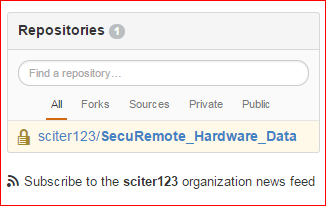


Figure #6

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

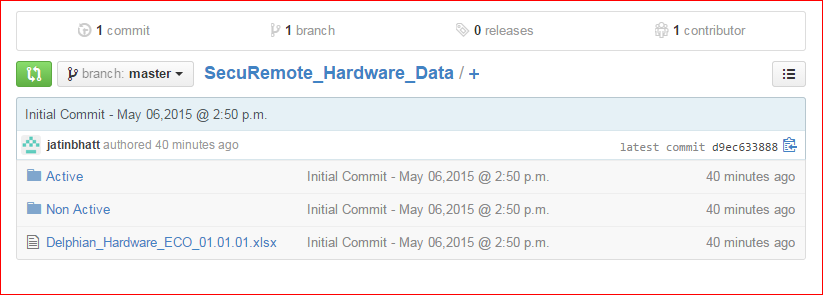


Figure #7

* Select “Active” folder. Refer #8. <https://github.com/sciter123/SecuRemote_Hardware_Data/tree/master/Active>

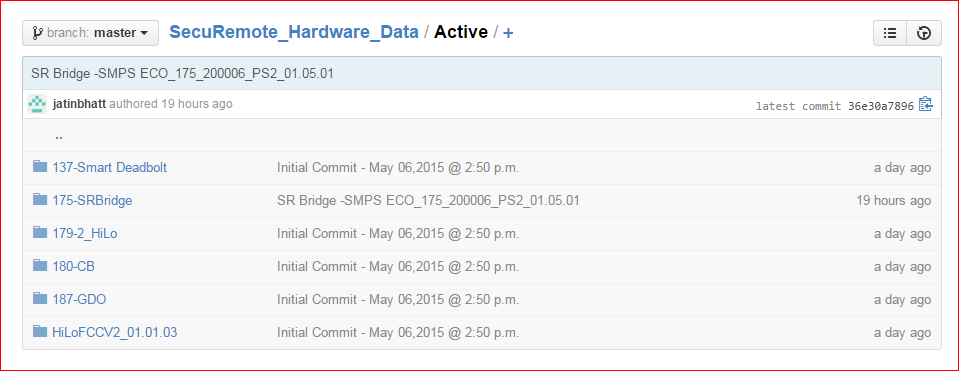


Figure #8

* Select “HiLoFCCV2\_01.01.03” folder. Refer Figure #9. <https://github.com/sciter123/SecuRemote_Hardware_Data/tree/master/Active/HiLoFCCV2_01.01.03>

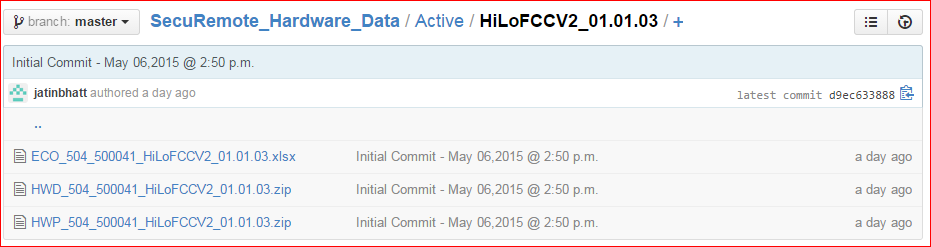


Figure #9

* Select any file out of three. For example “HWD\_504\_500041\_HiLoFCCV2\_01.01.03.zip“. Refer Figure #10. <https://github.com/sciter123/SecuRemote_Hardware_Data/blob/master/Active/HiLoFCCV2_01.01.03/HWD_504_500041_HiLoFCCV2_01.01.03.zip>

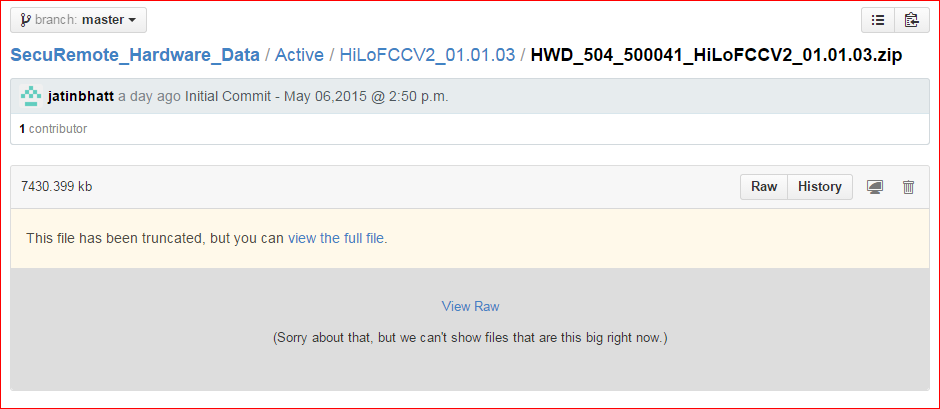


Figure #10

* 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/sciter123/SecuRemote_Hardware_Data/tree/master/Non%20Active>

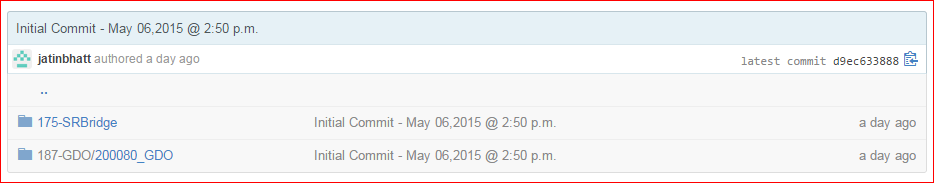


Figure #11

* Select “175\_SR Bridge” folder. Refer Figure #12. <https://github.com/sciter123/SecuRemote_Hardware_Data/tree/master/Non%20Active/175-SRBridge>

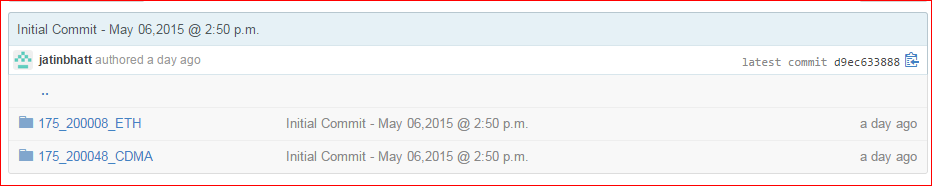


Figure #12

* Select “175\_200008\_ETH” folder. Refer Figure #13. <https://github.com/sciter123/SecuRemote_Hardware_Data/tree/master/Non%20Active/175-SRBridge/175_200008_ETH>

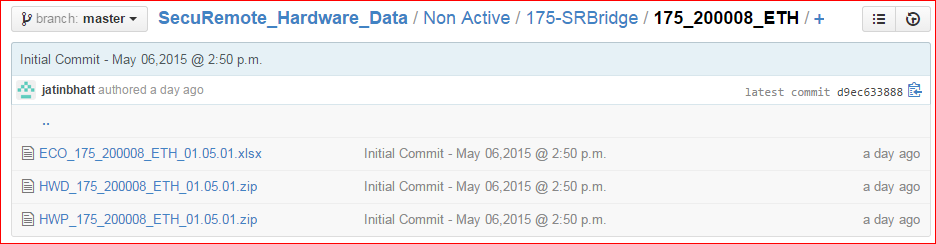


Figure #13

* Select any of them. They all are old version files. Support “. ECO\_175\_200008\_ETH\_01.05.01.xlsx “Refer Figure #14. <https://github.com/sciter123/SecuRemote_Hardware_Data/blob/master/Non%20Active/175-SRBridge/175_200008_ETH/ECO_175_200008_ETH_01.05.01.xlsx>

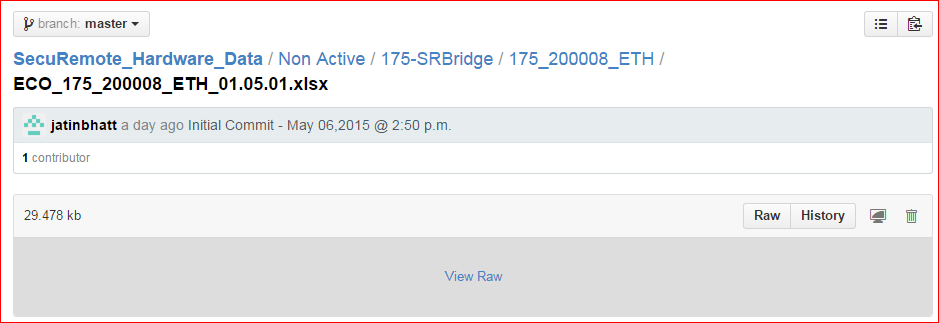
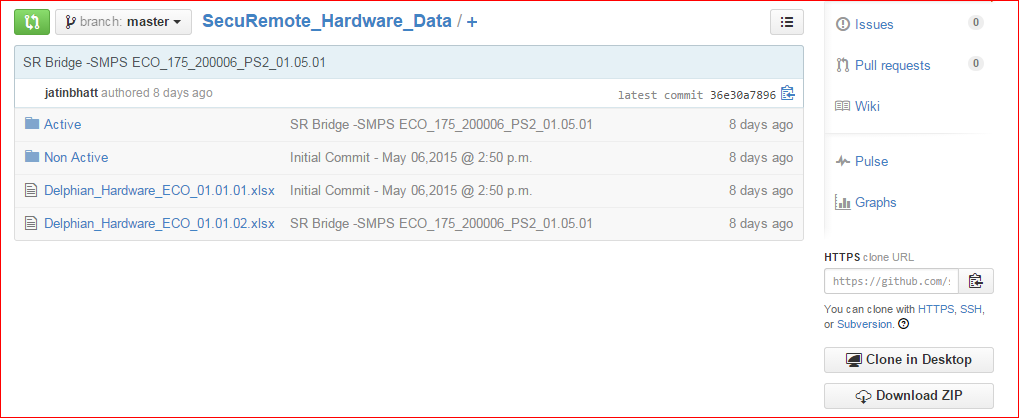


Figure #14

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

**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/sciter123/SecuRemote_Hardware_Data>

Figure #15

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

# File Structure for Firmware data

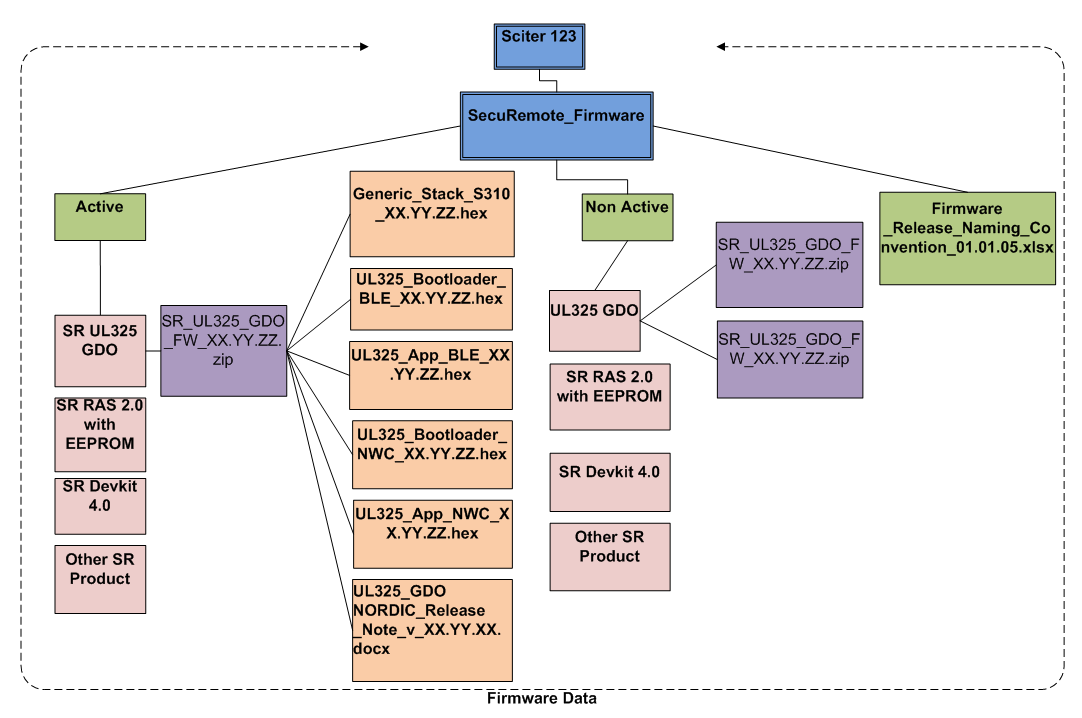


Figure #16

* **Note:** Here, “Sciter 123” is an organization account into GitHub. Refer Figure #16.
* “SecuRemote\_Firmware” is the other Repository into the “Sciter 123” 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.

## 7.1 Retrieve Firmware data from Repository

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

• Let us see live example: <https://github.com/orgs/sciter123/dashboard>. Refer Figure #17.

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

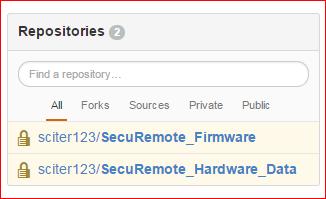


Figure #17

* Now select the “SecuRemote\_Firmware” Repository from the listed Repository. Refer Figure #17.
* After select specific Repository then it will display the three folders of that Repository. Refer Figure #18.

<https://github.com/sciter123/SecuRemote_Firmware>

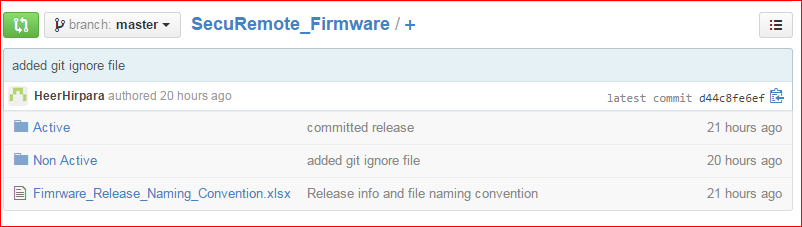


Figure #18

* Select “Active” folder from the list.
* It will display the folders for specific product. Refer Figure #19. <https://github.com/sciter123/SecuRemote_Firmware/tree/master/Active>

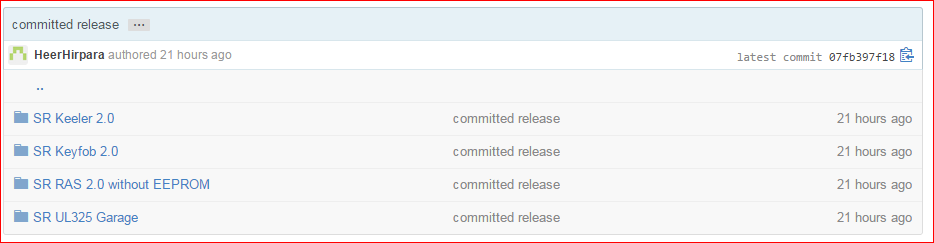
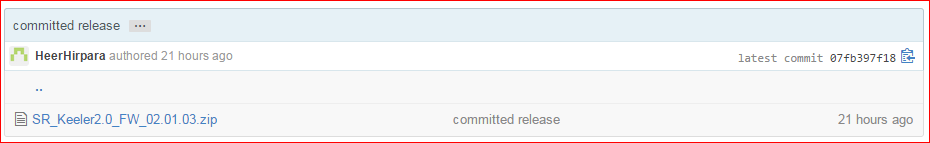


Figure #19

* Select any of them which user wants to get the information from that.
* For Example, select the first file from the list. Refer Figure #20. <https://github.com/sciter123/SecuRemote_Firmware/tree/master/Active/SR%20Keeler%202.0>

Figure #20

* Select .ZIP file for download. <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 #21.

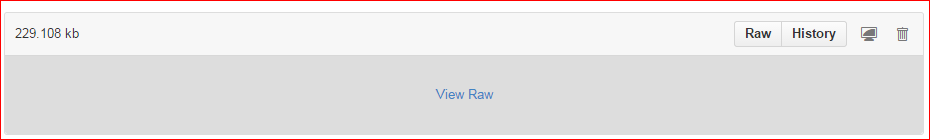


Figure #21

* Get that file from the Downloads folder into the command prompt.
* 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 #22.

<https://github.com/sciter123/SecuRemote_Firmware>

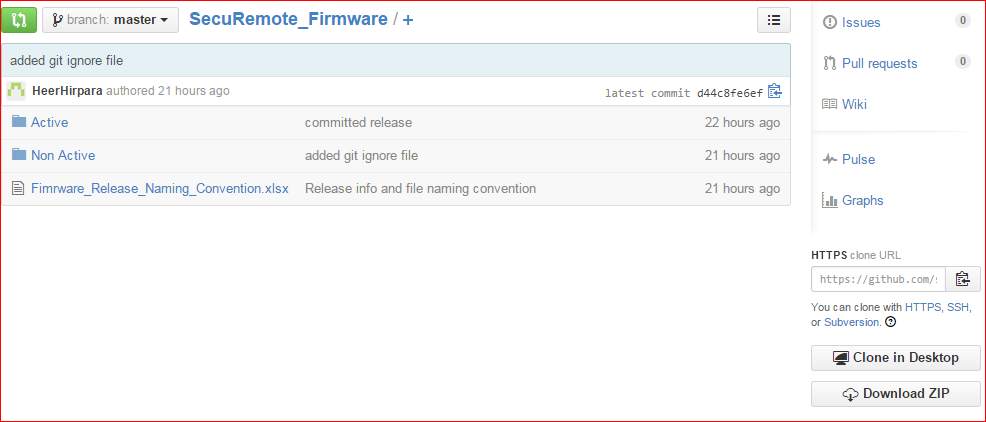


Figure #22

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

# File Structure for Developer Release

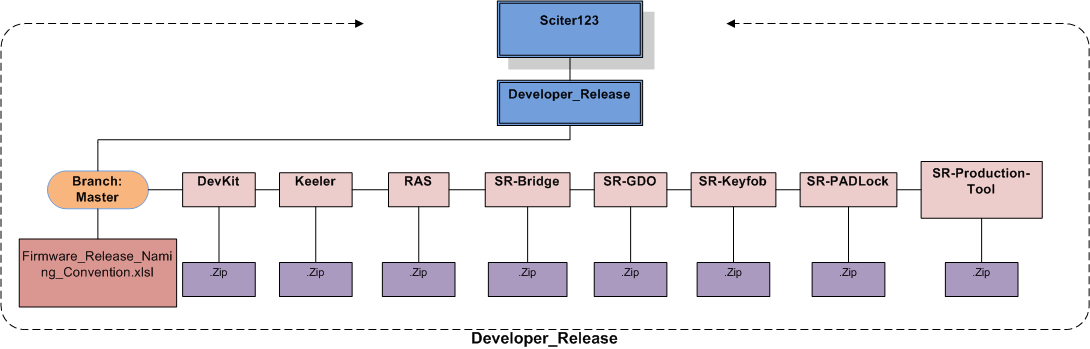


Figure # 23

* **Note:** Here, “Sciter 123” is an organization account into GitHub. Refer Figure #16.
* “Developer\_Release” is the other Repository into the “Sciter 123” 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.

## 8.1 Retrieve Developer\_ Release data from Repository

* User can retrieve the files from the “Repositories”.
* Let us see live example: <https://github.com/orgs/sciter123/dashboard>. Refer Figure #24.
* Select the Repository name “Developer\_Release” from the list. Refer Figure #24.

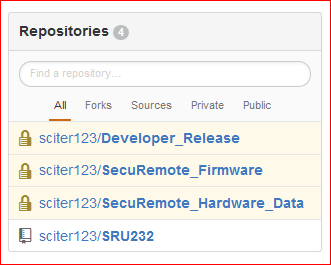


Figure # 24

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



Figure # 25

* Select Require File from the list. Refer Figure #26.

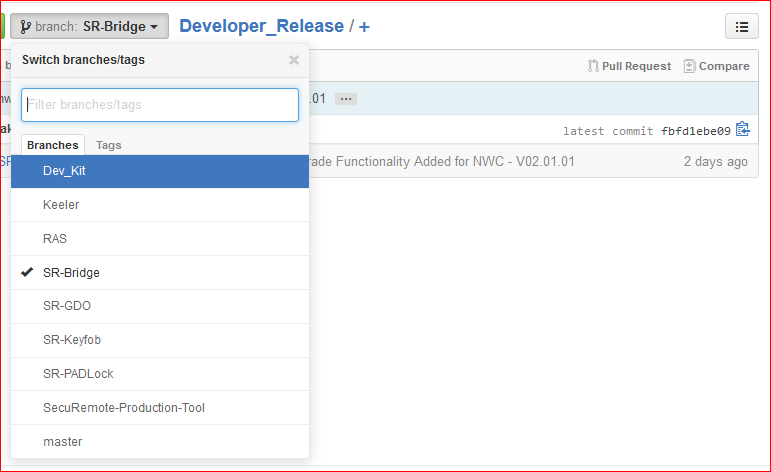


Figure # 26

* For Example: If user select Keeler from the list then there are two files is there. Refer figure #27.
* 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.

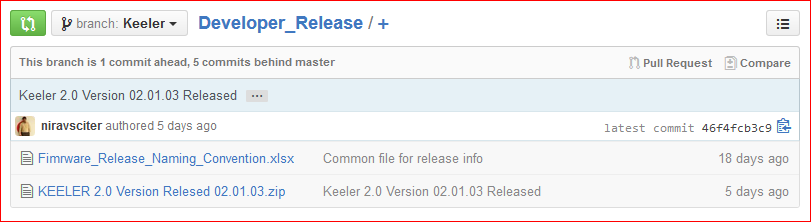


Figure # 27

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

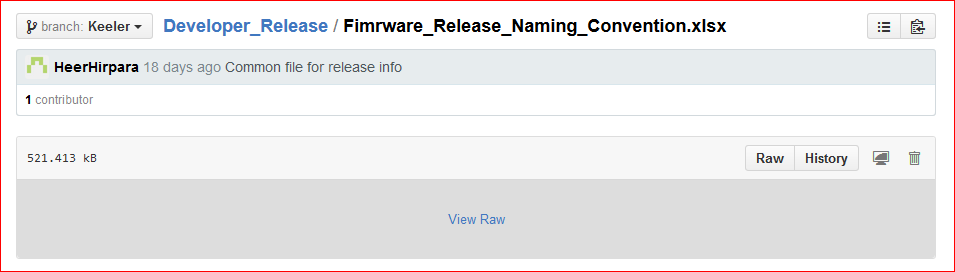


Figure # 28

* Click on “View Raw” and it will start the download that file. Refer Figure #28.
* Get that file from the Downloads folder into Command Prompt.
* Otherwise if user can download the all files then click on “Download ZIP”. Refer Figure #29.

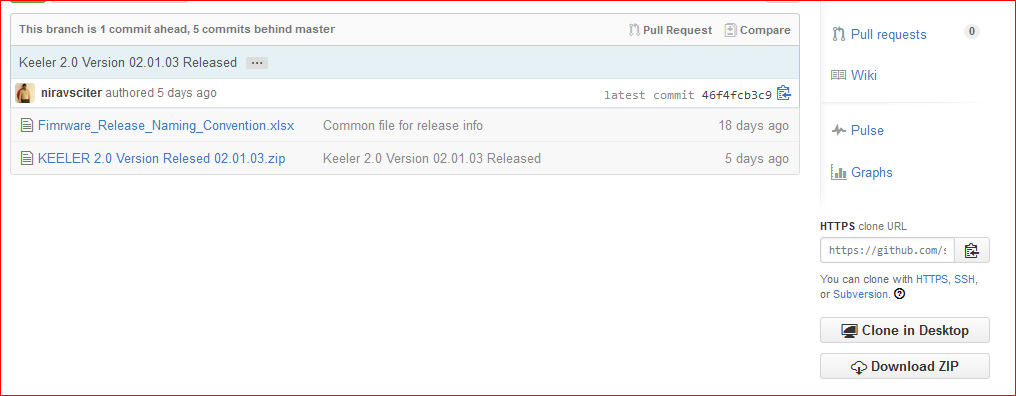


Figure # 29

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