1. Git Definitions

1.1. Git

In software development, Git (/ɡɪt/) is a distributed revision control and source code management (SCM) system with an emphasis on speed.[3] Initially designed and developed by Linus Torvalds for Linux kernel development, Git has since been adopted by many other projects.

1.2. Branching

Branching, in revision control and software configuration management, is the duplication of an object under revision control (such as a source code file, or a directory tree) so that modifications can happen in parallel along both branches.

Branches are also known as trees, streams or code lines. The originating branch is sometimes called the parent branch, the upstream branch (or simply upstream, especially if the branches are maintained by different organizations or individuals), or the backing stream. Child branches are branches that have a parent; a branch without a parent is referred to as the trunk or the mainline.

1.3. Merging

Merging (also called integration) in revision control, is a fundamental operation that reconciles multiple changes made to a revision- controlled collection of files. Most often, it is necessary when a file is modified by two people on two different computers at the same time. When two branches are merged, the result is a single collection of files that contains both sets of changes.

2. Git Commands

1.1. Prerequisites

a) To be able to use Littelfuse's git repository, you need to provide the web development manager your public key

b) To generate a public key

- ssh-keygen -t rsa
- This will generate 2 files in your .ssh folder, the id_rsa (private key, protect this) and the id_rsa.pub (public key, you can share this to anyone)

c) Email the id_rsa.pub to the web development manager

1.2. Cloning a repository

a) git clone repository

1.3. Merging a branch

a) git merge <branch-name>

1.4. Fixing merge conflict

a) If you merge a branch to your current branch, git might not be able to auto merge some files. To fix that you can either do these possible solutions. To keep your current version use:

- git checkout –ours – <path-to-file-name>

b) To keep the version you are merging in, use:

- git checkout –theirs – <path-to-file-name>

1.5. Remove files from to be committed list

a) git reset <filename>

1.6. Checkout a branch

a) git checkout <branch-name>

1.7. Checkout a file from a different branch

a) git checkout <branch-name> <path-to-filename>

1.8. Look at the log of the current branch-name

a) git log

b) git log <branch-name> (you can use this to look at the log of the other branch-name

1.9. Look at the current status

a) git status

1.10. Remove a file from the index

a) git rm –cached <filename>

1.11. Rollback the last commit in remote

a) git reset HEAD^ --hard
b) git push origin <branch-name> -f

1.12. Reset your local branch to match remote

a) git reset –hard origin/<branch-name>

1.13. Copy a file from a commit from different branch

a) While on the test branch, do the following command. This will create a new branch with the commit-id as its name

```
git fetch <branch-name>:<commit-id>
```

b) Now to get the file based on that commit, do

```
git checkout <branch-name> <filename>
```