



Git & Bricktronics

Software Training

Git

What is Git?

“**Git** is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.”

“Version Control System”

Systems used by teams to share code and track changes to a code base.

Goes above and beyond simply a shared storage device and makes it easy to keep up with change histories and who did what.

“Distributed”

A centralized VCS has a single server that hosts the one version to rule them all. Every team member pulls new code from that server and pushes code up to that server. If the server goes down, game over.

A distributed VCS allows for the transfer of code to many remotes, including other developer's computers. We have a central server via GitHub, but we can operate without the internet and even use a portable harddrive as a git server.

Useful Commands

- clone
- branch
- fetch
- merge
- pull
- push
- checkout
- stash
- blame
- log

Clone

- `git clone <remote url>`
- Creates a local copy of the remote repository linked to the remote

Branch

- Manage git branches
- `git branch -a` - show all git branches
- `git branch [new name]` - create a new branch
- `git branch -d [name]` - delete branch

Fetch

- `git fetch [remote] [branch]`
- Updates the history of remote repository in local working tree's log

Merge

- `git merge [remote] <branch>`
- Merges code from another branch or remote

Pull

- `git pull [remote] [branch]`
- Combines a fetch and merge operation into a single step

Push

- `git push [remote] [branch]`
- Sends local code up to remote repository.

Checkout

- `git checkout [remote] [branch]` - Switches to local copy of remote branch
- `git checkout [branch]` - Switches to local branch
- `git checkout -b [new name]` - Creates and switches to new branch

Stash

- Save a temporary copy of uncommitted changes
- `git stash` - save current uncommitted changes
- `git stash apply` - reapply latest stash
- `git stash drop` - erase latest stash
- `git stash list` - show list of current stashes
- `git stash show` - show changes in latest stash

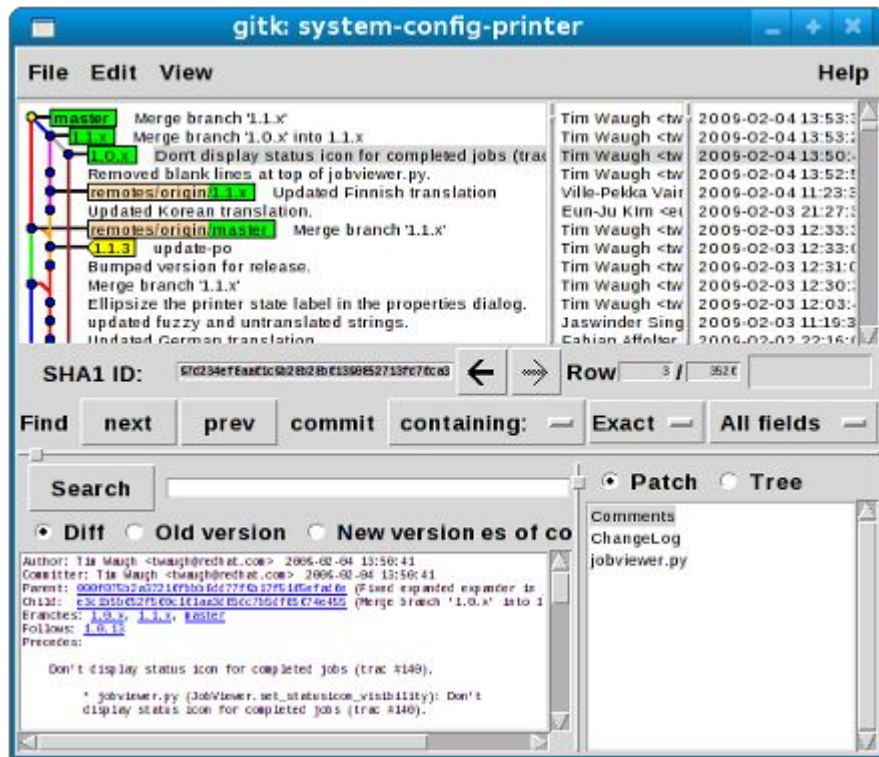
Blame

- `git blame [file path]`
- Show a line-by-line log of who wrote which parts of the file and when

Log

- `git log`
- Shows the history log of commits in this branch

Demo: gitk



Bricktronics

Motors

```
#include ...  
  
BricktronicsMotor motor{BricktronicsShield::MOTOR_1};  
  
void setup() {  
  BricktronicsShield::begin();  
  motor.begin();  
}  
  
void loop() {  
  motor.brake(); // Stop motor  
  motor.setFixedDrive(-150); // Run backwards  
  motor.setFixedDrive(150); // Run forwards  
}
```

Touch

Digital button switch that can be used for input, bumpers, etc.

Two states: pressed & released

```
#include ...  
  
BricktronicsButton button{BricktronicsShield::SENSOR_2};  
  
void setup() {  
  BricktronicsShield::begin();  
  button.begin();  
}  
  
void loop() {  
  if(button.isPressed())  
    // button pressed  
  else  
    // button released  
}
```

Light

Analog sensor for measuring light intensity or reflectivity

Gives light values on an 8-bit scale. Can turn on / off built-in light source.

```
#include ...  
  
BricktronicsLight ls{BricktronicsShield::SENSOR_1};  
  
void setup() {  
    BricktronicsShield::begin();  
    ls.begin();  
}  
  
void loop() {  
    auto raw = ls.value();  
    auto scaled = ls.scaledValue();  
    ls.setFloodlightAlways(true);  
}
```

Ultrasonic

Analog sensor for measuring distances using the speed of sound

Gives distances in mm

```
#include ...  
  
BricktronicsUltrasonic ultrasonic{BricktronicsShield::SENSOR_4};  
  
void setup() {  
  BricktronicsShield::begin();  
  ultrasonic.begin();  
}  
  
void loop() {  
  auto distance = ultrasonic.getDistance();  
}
```

Demo: Bricktronics Sensors