Firmware Training Week 1

Intro to Firmware and Programming

RoboJackets

Competitive Robotics at Georgia Tech

www.robojackets.org

Who am I?

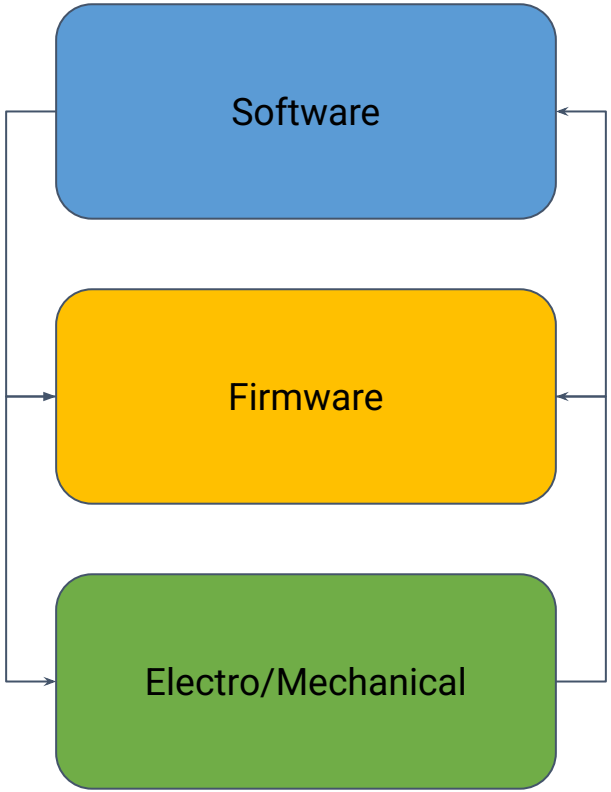
- Nate Wert
- 1st Year CS Master's Student
  - Perception & Robotics
- RoboCup

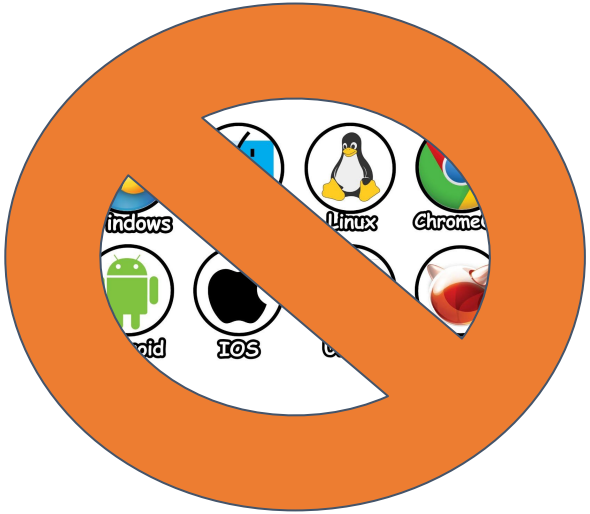
Agenda

- What is Firmware?
- General Arduino Programming (i.e. C/C++)
- Digital Outputs
- Lab Setup + Lab 1 (blinky)

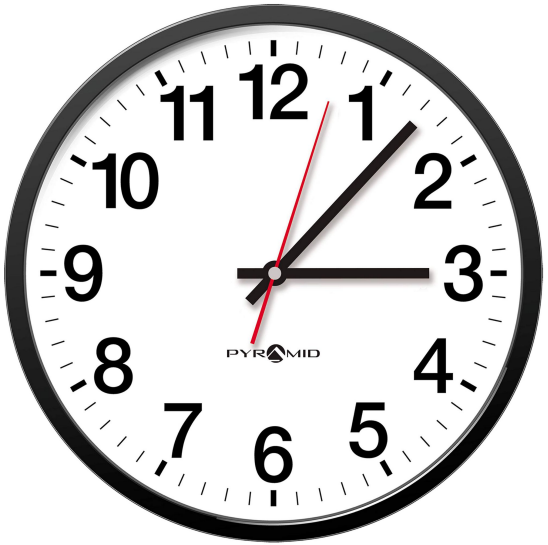
What is Firmware?

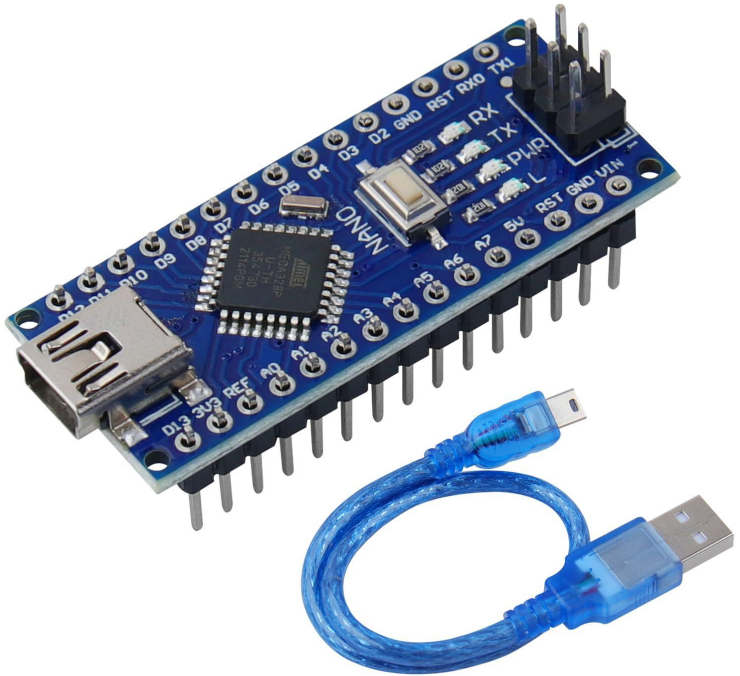

BostonDynamics


Software
Firmware
Electro/Mechanical


indows
Linux
Chrome
oid
IOS


SAFETY
FIRST


12
1
2
3
4
5
6
7
8
9
10
11
PYRAMID

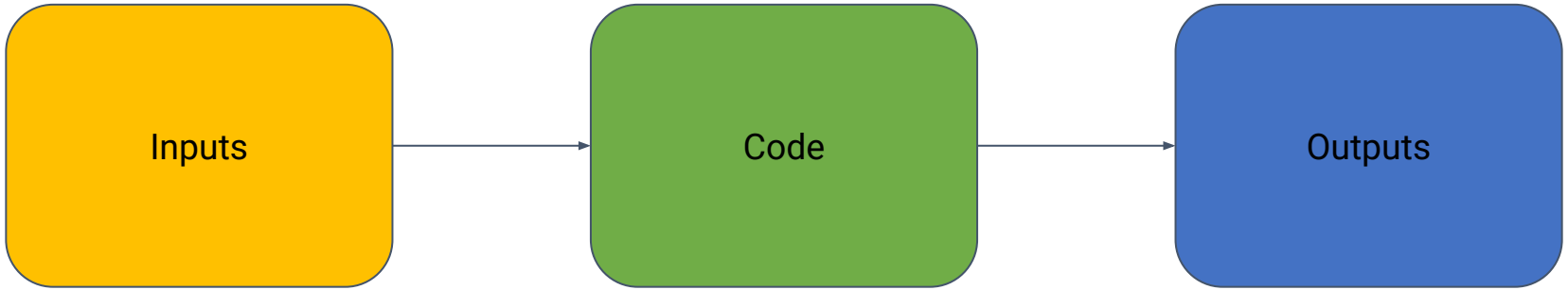

NANO
RX
TX
PWR
L
TXI
RXO
RST
GND
D2
D3
D4
D5
D6
D7
D8
D9
D10
A7
5V
RST
GND
VIN
A6
A5
A4
A3
A2
A1
A0
REF
3V3
D13

Training Format

Yapping

www.emmasaying.com

General Programming


Inputs
Code
Outputs

Numeric Data (according to <cstdint>)

Integers

Signed

- int8_t
- int16_t
- int32_t
- int64_t

Unsigned

- uint8_t
- uint16_t
- uint32_t
- uint64_t

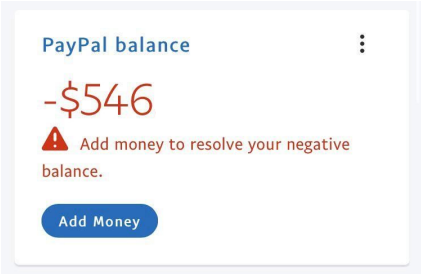
“Floats”

- float (f32)
- double (f64)

Boolean

- bool (true / false)

Signed

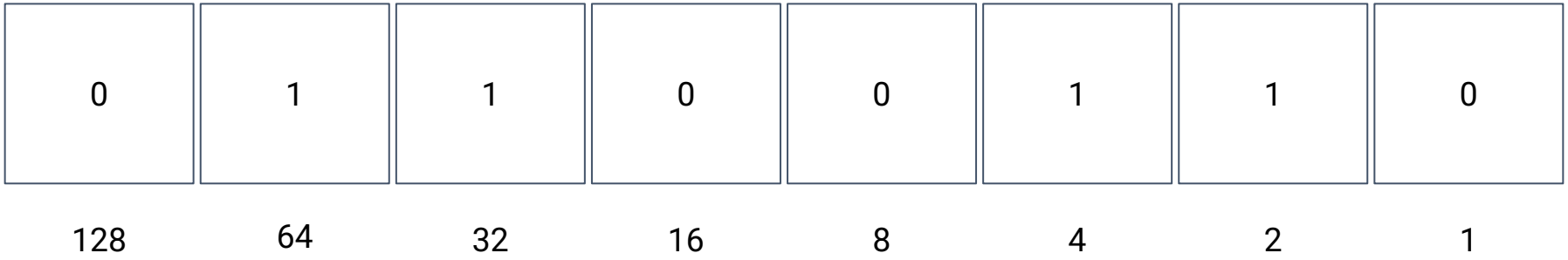


Unsigned

The “Byte”

- bit

The “Byte”



128(0) + 64(1) + 32(1) + 16(0) +
8(0) + 4(1) + 2(1) + 1(0)
= 102

* uint8_t

Signed Bytes

- bit

The “Byte”

1	1	1	0	0	1	1	0
+/-	64	32	16	8	4	2	1

$$-(1)(64(1) + 32(1) + 16(0) + 8(0) + 4(1) + 2(1) + 1(0))$$
$$= -102$$

* int8_t (signed magnitude)

Adding Two Unsigned Bytes

0	1	1	0	0	1	1	0

+

1	1	1	0	0	1	1	0

0	1	1	0	0	1	1	0

+

1	1	1	0	0	1	1	0

=

0	0	0	0	1	1	0	0

Adding Two Signed Bytes

0	1	1	0	0	1	1	0

+

1	1	1	0	0	1	1	0

Two's-Complement

0	1	1	0	0	1	1	0

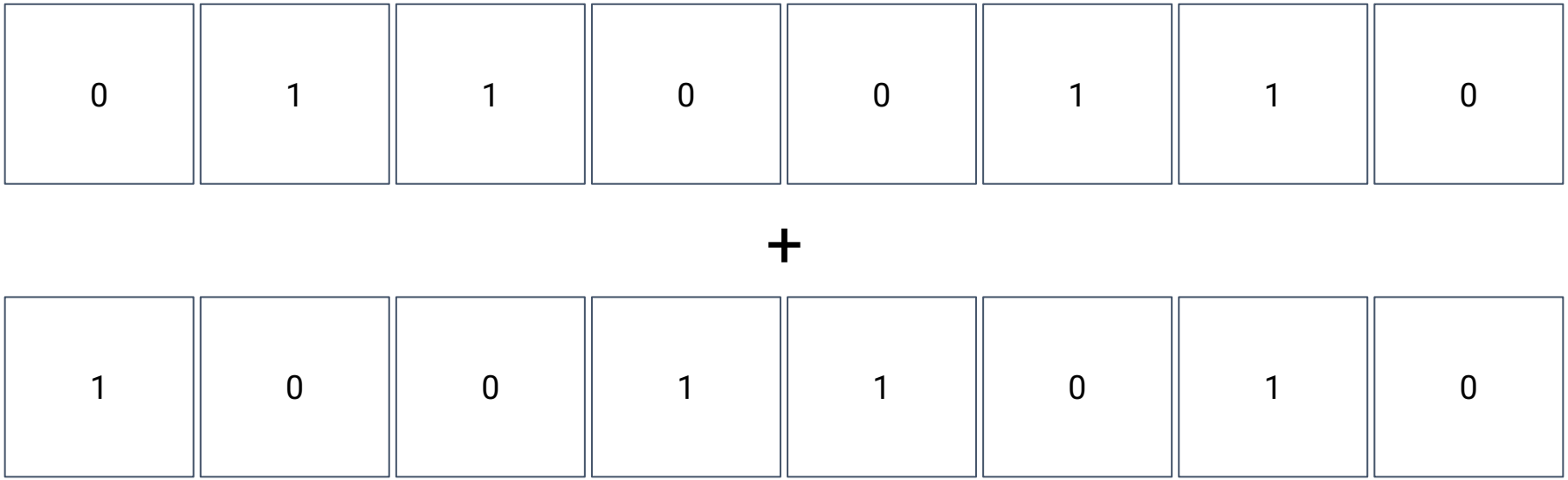
Flip

1	0	0	1	1	0	0	1

+1

1	0	0	1	1	0	1	0

Adding Two Signed Bytes

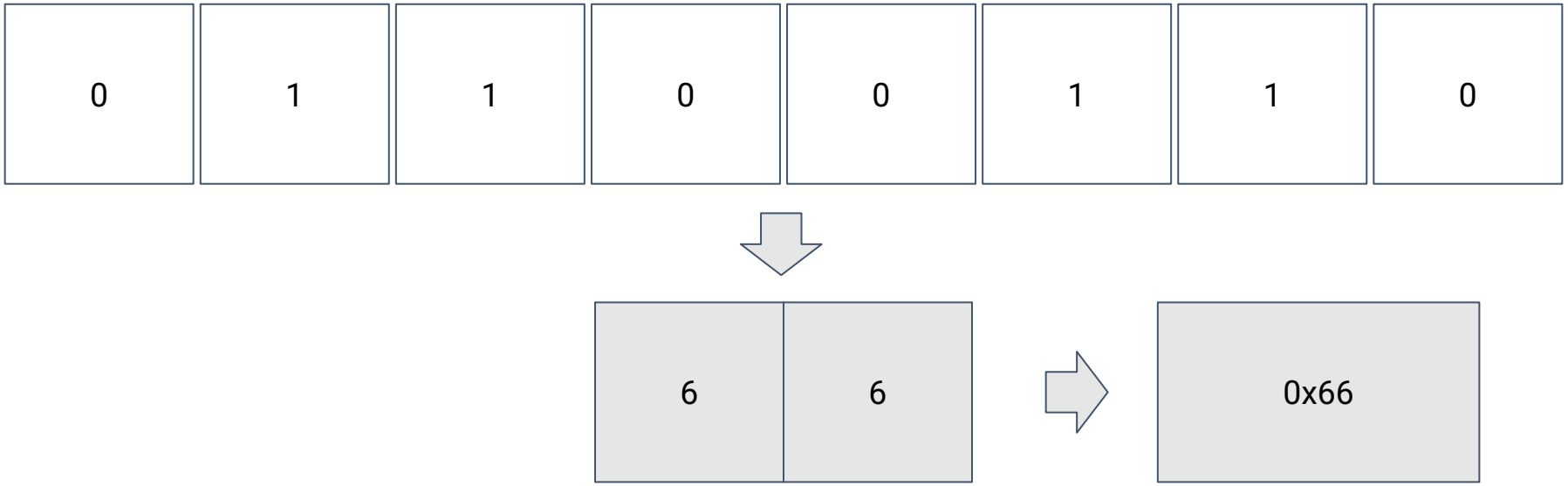


“Big” Numbers

How do I write 500?

255

The “Byte”



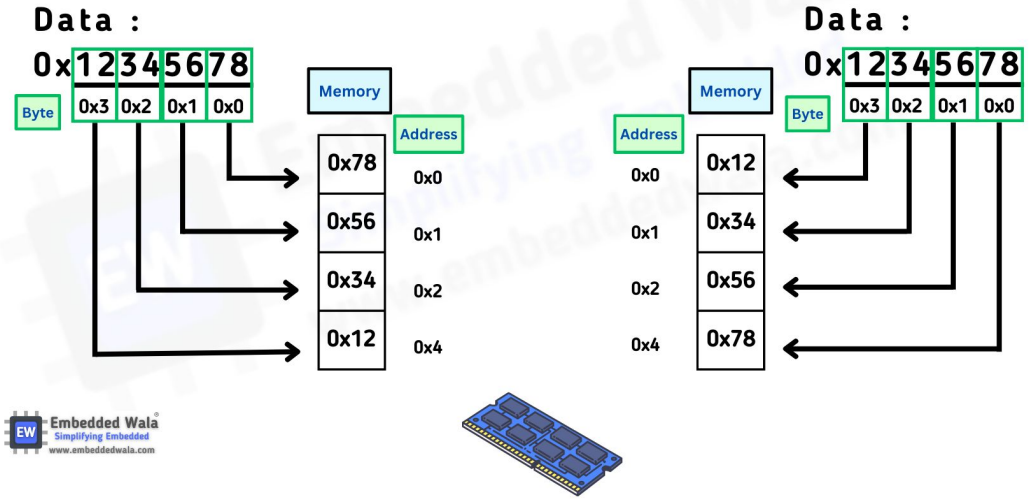
500

If 0x01F4 is 500, what is:

0xF4

0x01

Little Vs Big Endianness



500

Little Endian (LE)

Big Endian (BE)

Numeric Data (according to <cstdint>)

Integers

Signed

- Int8_t
  - -128
  - 127
- Int16_t
  - -32,768
  - 32,767
- Int32_t
  - -2^{31}
  - $2^{31}-1$
- Int64_t
  - -2^{63}
  - $2^{63}-1$

Unsigned

- uint8_t
  - 0
  - 255
- uint16_t
  - 0
  - 65,535
- uint32_t
  - 0
  - $2^{32}-1$
- uint64_t
  - 0
  - $2^{64}-1$

“Floats”

- float (f32)
  - Less Precision
- double (f64)
  - More Precision

Numeric Data (according to <cstdint>)

Integers

Signed

- Int8_t
  - -128
  - 127
- Int16_t
  - -32,768
  - 32,767
- Int32_t
  - -2^{31}
  - $2^{31}-1$
- Int64_t
  - -2^{63}
  - $2^{63}-1$

Unsigned

- uint8_t
  - 0
  - 255
- uint16_t
  - 0
  - 65,535
- uint32_t
  - 0
  - $2^{32}-1$
- uint64_t
  - 0
  - $2^{64}-1$

“Floats”

- float (f32)
  - Less Precision
- double (f64)
  - More Precision

Boolean

- bool (true / false)

Other Data Types

Characters (‘a’, ‘b’, …, ‘/’, ‘~’, …)

- char or uint8_t

Strings (“hello”, “goodbye”, “tschüss”, …)

- std::string or char[] or char *

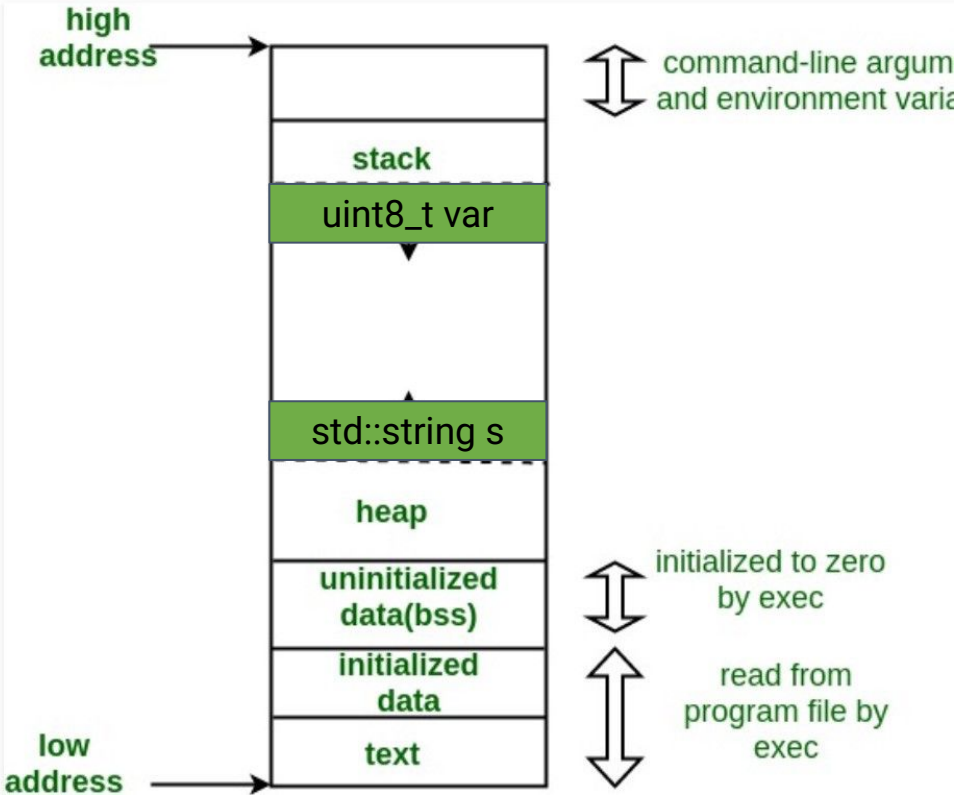
* Note: We will get to pointers, structs, and classes in a future week

Declaring Data

```
uint8_t v = 25;
v = 33;

// Will not fit
// v = 256;

std::string s = "Robojackets Training";
```

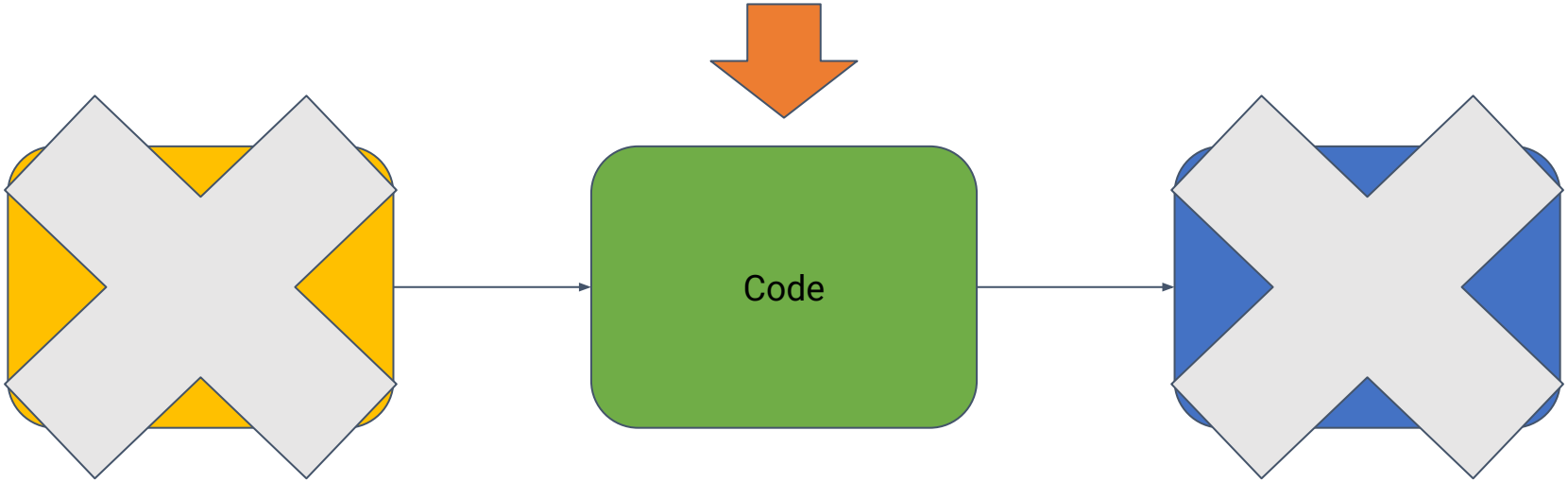

high address
command-line argum
and environment varia
stack
uint8_t var
std::string s
heap
uninitialized data(bss)
initialized to zero by exec
initialized data
read from program file by exec
text
low address

Modifying Data

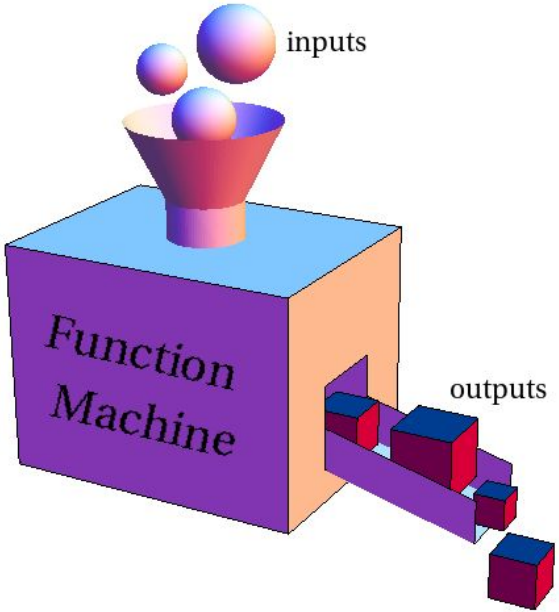
```
int64_t v = 333;

// Equivalent Statements
v = v + 20; // v = 353
v += 20 ; // v = ?

int32_t v2 = v;
v2 -= 30;
v *= 33;
v /= 100;
v2 = 100;
// What is v2?
```

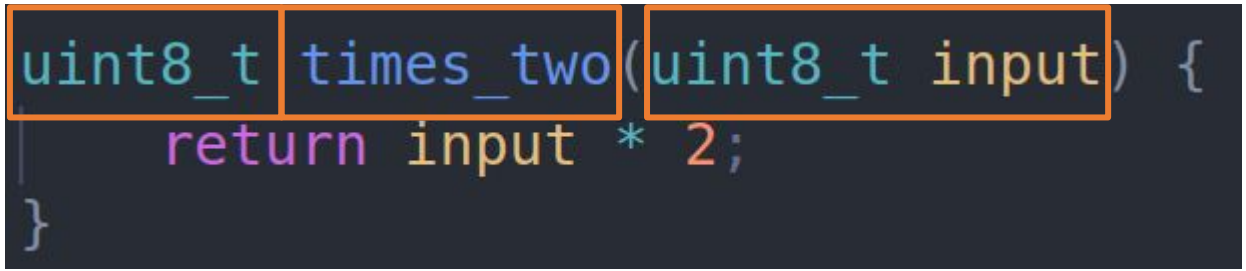

Code

Functions



Signature

Output Name Input(s)



Body

```
Dinosaur apply_magic(Arms arms, Legs legs, Tail tail)

double calculate_heat_death(double universe_age, float exp_decay)
```

Important Functions

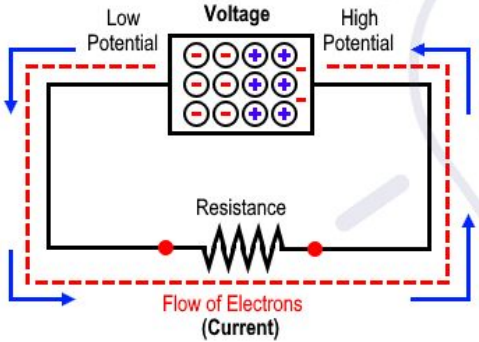
```
void setup()
```

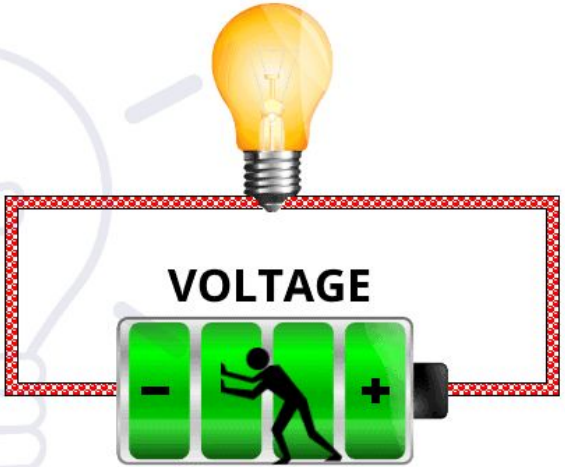
```
void loop()
```

Digital Outputs

WHAT IS VOLTAGE ?

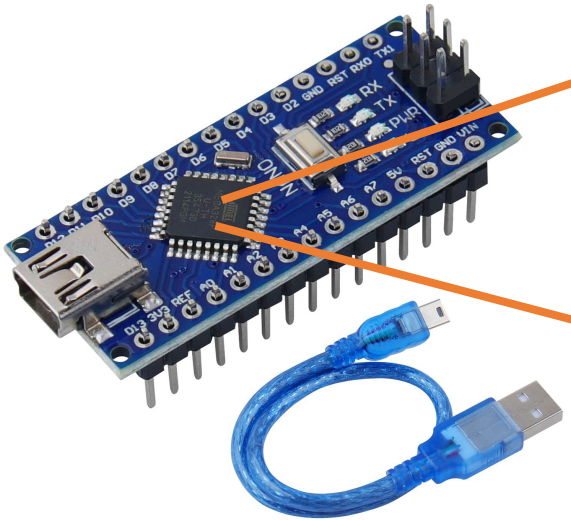
Its Unit, Formula, Types, Properties, Measurement & Application

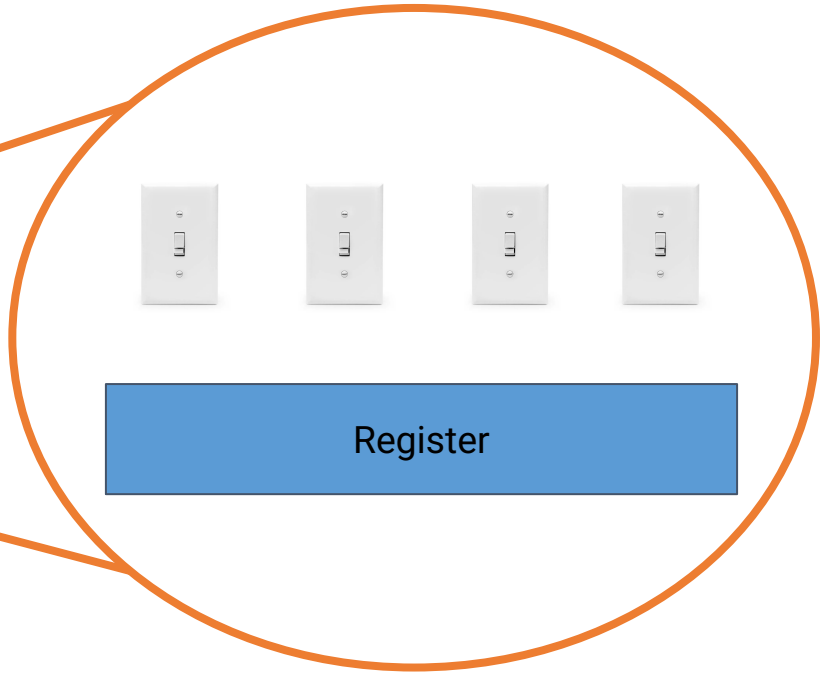


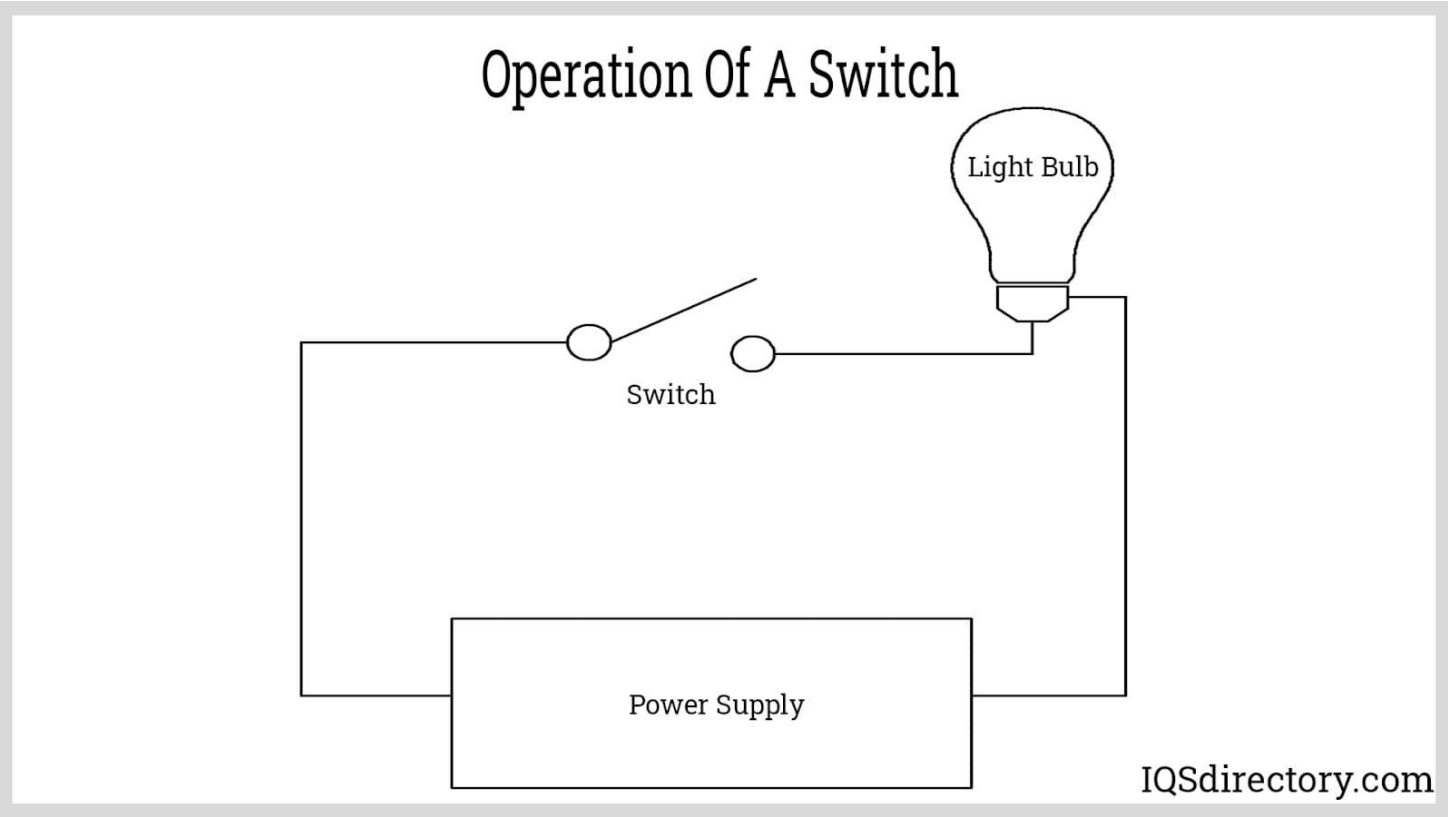


WWW.ELECTRICALTECHNOLOGY.ORG

Registers






Operation Of A Switch
Light Bulb
Switch
Power Supply
IQSdirectory.com

Use the following function to write a digital value to a pin:

```
digitalWrite(pin, value)
```

Parameters

The function admits the following parameters:

- `pin`: the Arduino pin number to be controlled.
- `value`: `HIGH` or `LOW`

Returns

The function returns **nothing**.

Syntax

Use the following function to set the behavior of a pin:

```
pinMode(pin, mode)
```

Parameters

The function admits the following parameters:

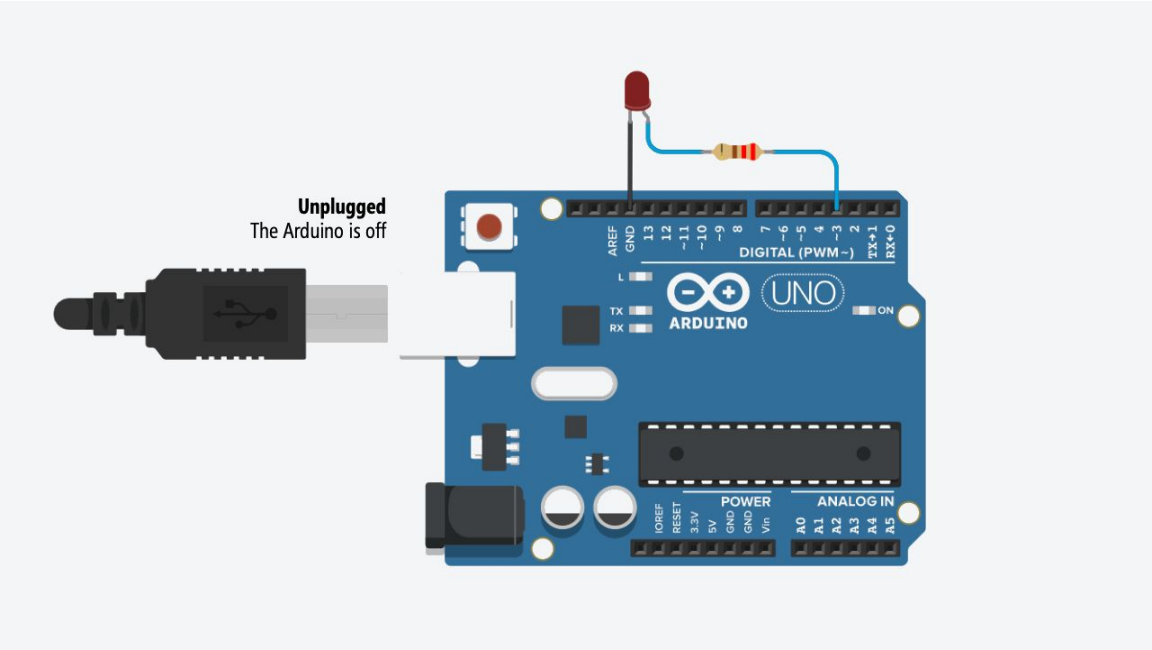
- `pin`: the Arduino pin number to set the mode of.
- `mode`: `INPUT`, `OUTPUT`, or `INPUT_PULLUP`. See the **Digital Pins** page for a more complete description of the functionality.

Returns

The function returns **nothing**.

Lab 1

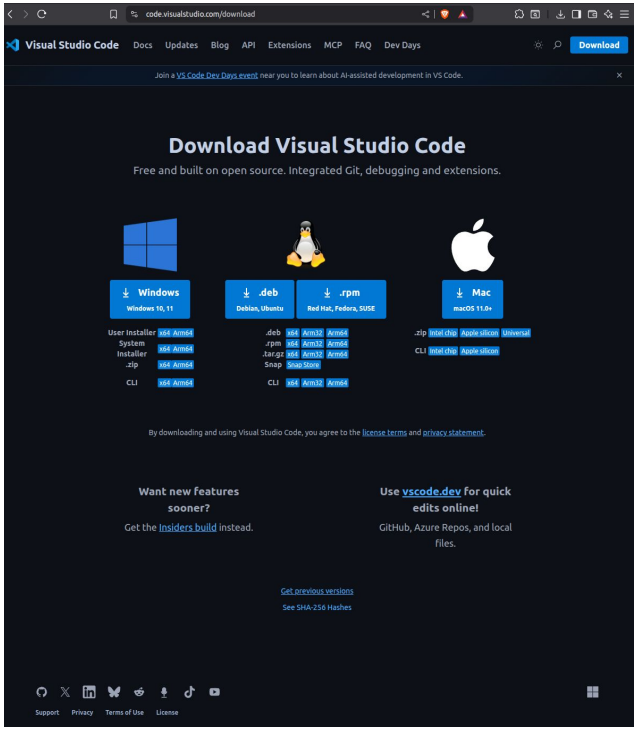
Goal


Unplugged
The Arduino is off
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

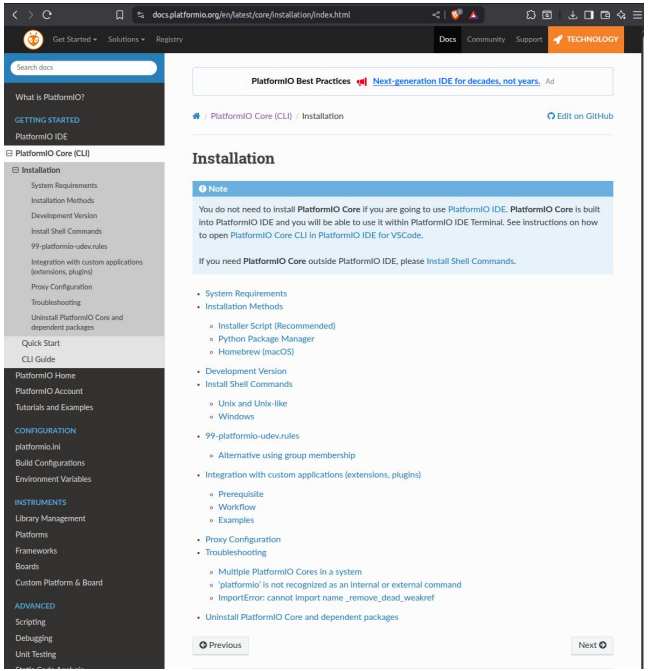
Setup

- VSCode
- PlatformIO CLI
- Wokwi for VSCode
- Git

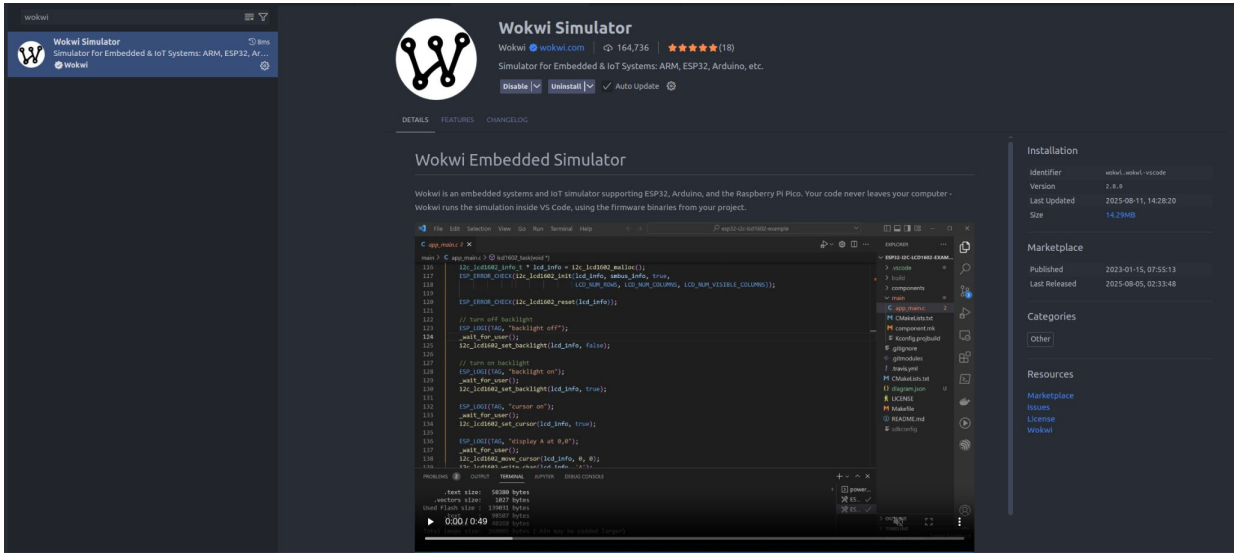
VSCode


code.visualstudio.com/download
Visual Studio Code
Docs
Updates
Blog
API
Extensions
MCP
FAQ
Dev Days
Download
Join a VS Code Dev Days event near you to learn about AI-assisted development in VS Code.
Download Visual Studio Code
Free and built on open source. Integrated Git, debugging and extensions.
Windows
Windows 10, 11
.deb
Debian, Ubuntu
.rpm
Red Hat, Fedora, SUSE
Mac
macOS 11.0+
User Installer x64 Arm64
System Installer x64 Arm64
.zip x64 Arm64
CLI x64 Arm64
.deb x64 Arm32 Arm64
.rpm x64 Arm32 Arm64
.tar.gz x64 Arm32 Arm64
Snap Snap Store
CLI x64 Arm32 Arm64
.zip Intel chip Apple silicon Universal
CLI Intel chip Apple silicon
By downloading and using Visual Studio Code, you agree to the license terms and privacy statement.
Want new features sooner?
Get the Insiders build instead.
Use vscode.dev for quick edits online!
GitHub, Azure Repos, and local files.
Get previous versions
See SHA-256 Hashes
Support
Privacy
Terms of Use
License

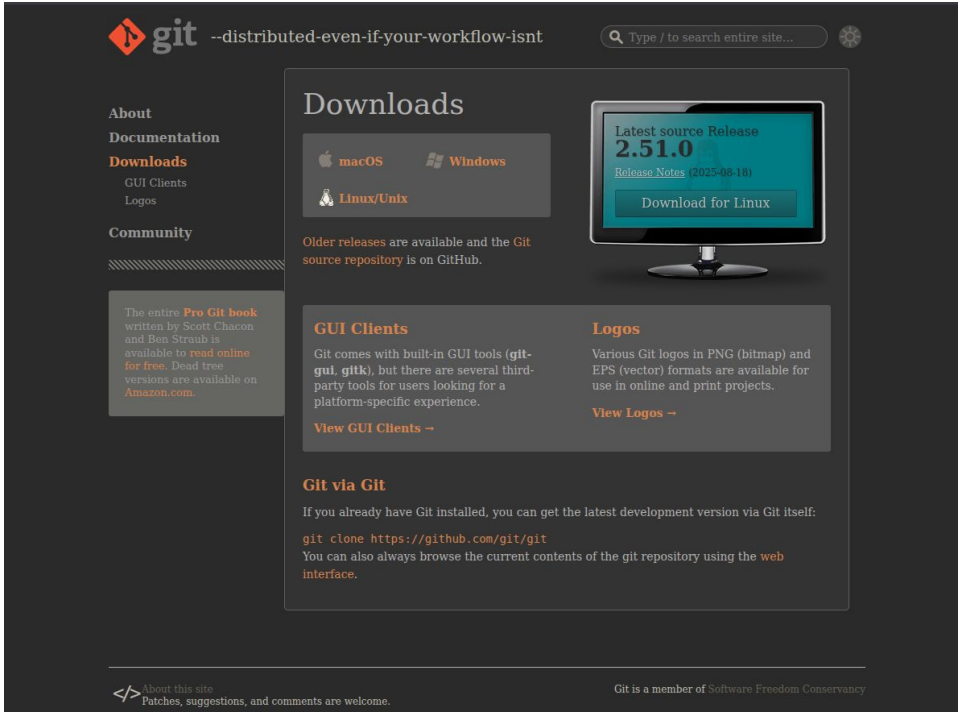
Platformio Core (CLI)



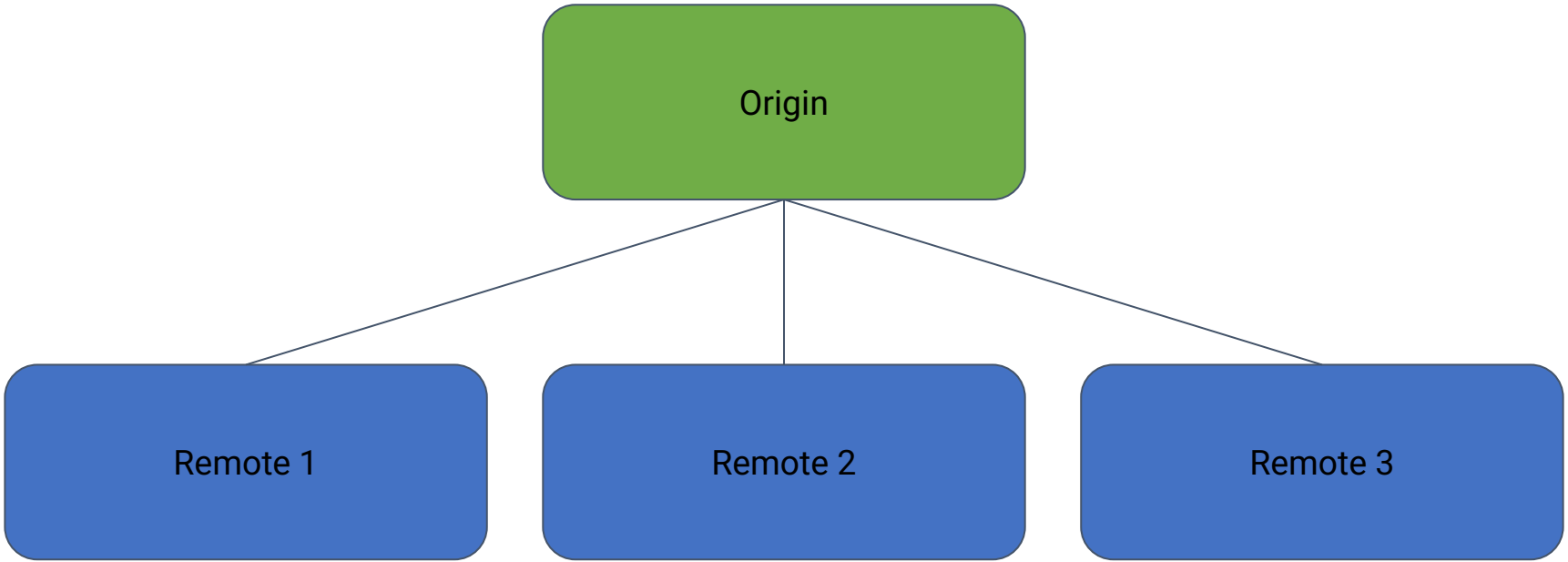
Wokwi


wokwi
Wokwi Simulator
Simulator for Embedded & IoT Systems: ARM, ESP32, Ar...
Wokwi
Wokwi Simulator
Wokwi wokwi.com 164,736 (18)
Simulator for Embedded & IoT Systems: ARM, ESP32, Arduino, etc.
Disable Uninstall Auto Update
DETAILS FEATURES CHANGELOG
Wokwi Embedded Simulator
Wokwi is an embedded systems and IoT simulator supporting ESP32, Arduino, and the Raspberry Pi Pico. Your code never leaves your computer - Wokwi runs the simulation inside VS Code, using the firmware binaries from your project.
Installation
Identifier
Version
Last Updated
2025-08-11, 14:28:20
Size
14.29MB
Marketplace
Published
2023-01-15, 07:55:13
Last Released
2025-08-05, 02:33:48
Categories
Other
Resources
Marketplace
Issues
License
Wokwi
0:00 / 0:49

Git

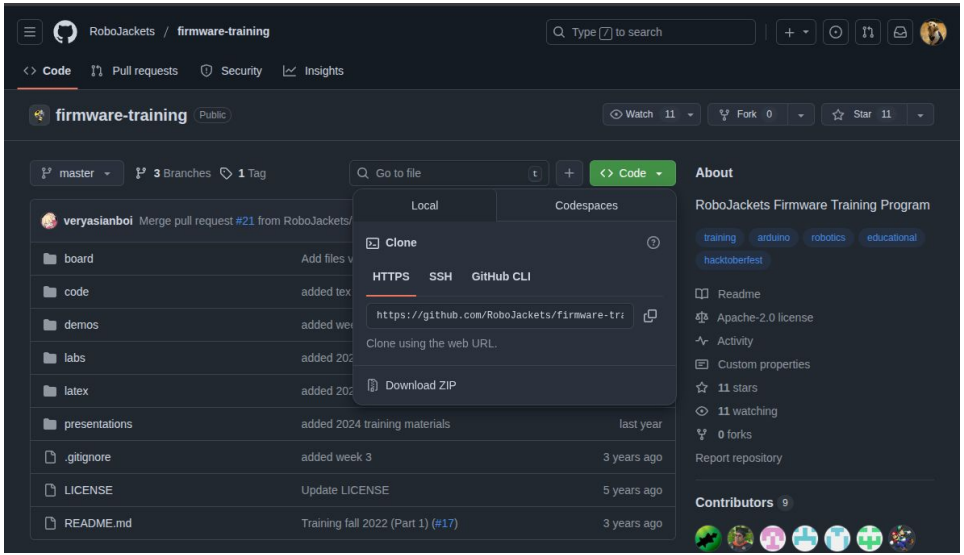

git --distributed-even-if-your-workflow-isnt
Type / to search entire site...
About
Documentation
Downloads
GUI Clients
Logos
Community
Downloads
macOS
Windows
Linux/Unix
Older releases are available and the Git source repository is on GitHub.
Latest source Release
2.51.0
Release Notes (2025-08-18)
Download for Linux
The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.
GUI Clients
Git comes with built-in GUI tools (git-gui, gitk), but there are several third-party tools for users looking for a platform-specific experience.
View GUI Clients →
Logos
Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.
View Logos →
Git via Git
If you already have Git installed, you can get the latest development version via Git itself:
git clone https://github.com/git/git
You can also always browse the current contents of the git repository using the web interface.
About this site
Patches, suggestions, and comments are welcome.
Git is a member of Software Freedom Conservancy

Git


Origin
Remote 1
Remote 2
Remote 3

Cloning

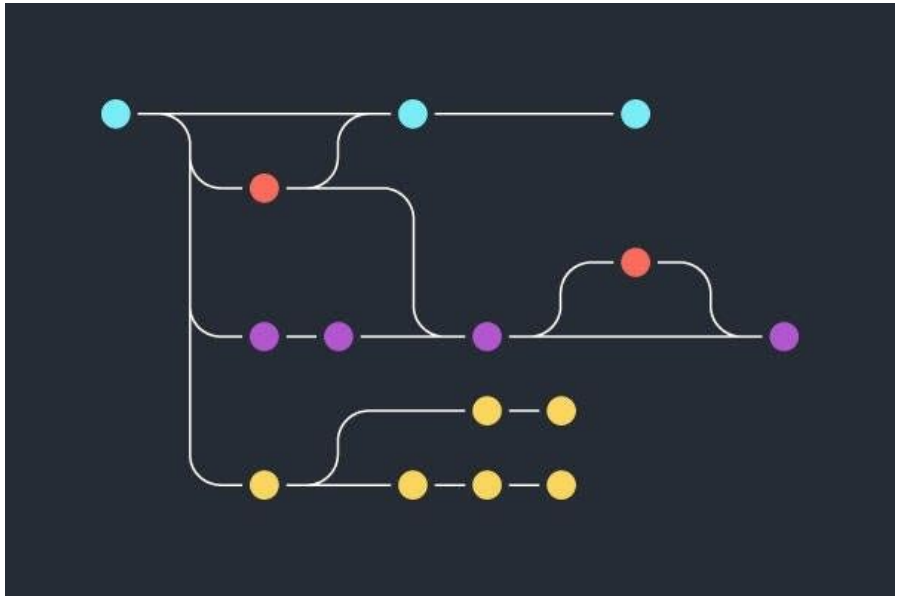
Cloning
(https://github.com/RoboJackets/firmware-training)



Cloning
(https://github.com/RoboJackets/firmware-training)

```
n8w3rt@n8w3rt:~$ git clone https://github.com/RoboJackets/firmware-training.git
```

Branching



Creating Branches

```
n8w3rt@n8w3rt:~/firmware-training$ git checkout -b nwert-firmware-training-2025
```

Seeing Branches

```
n8w3rt@n8w3rt:~/firmware-training$ git branch
  master
* nwert3-firmware-training-2025
```

Switching Branches

```
n8w3rt@n8w3rt:~/firmware-training$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
n8w3rt@n8w3rt:~/firmware-training$ git branch
* master
  nwert3-firmware-training-2025
```

Seeing Changes

```
n8w3rt@n8w3rt:~/robocup/firmware-training$ git status
On branch 2025-week-1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        labs/

nothing added to commit but untracked files present (use "git add" to track)
```

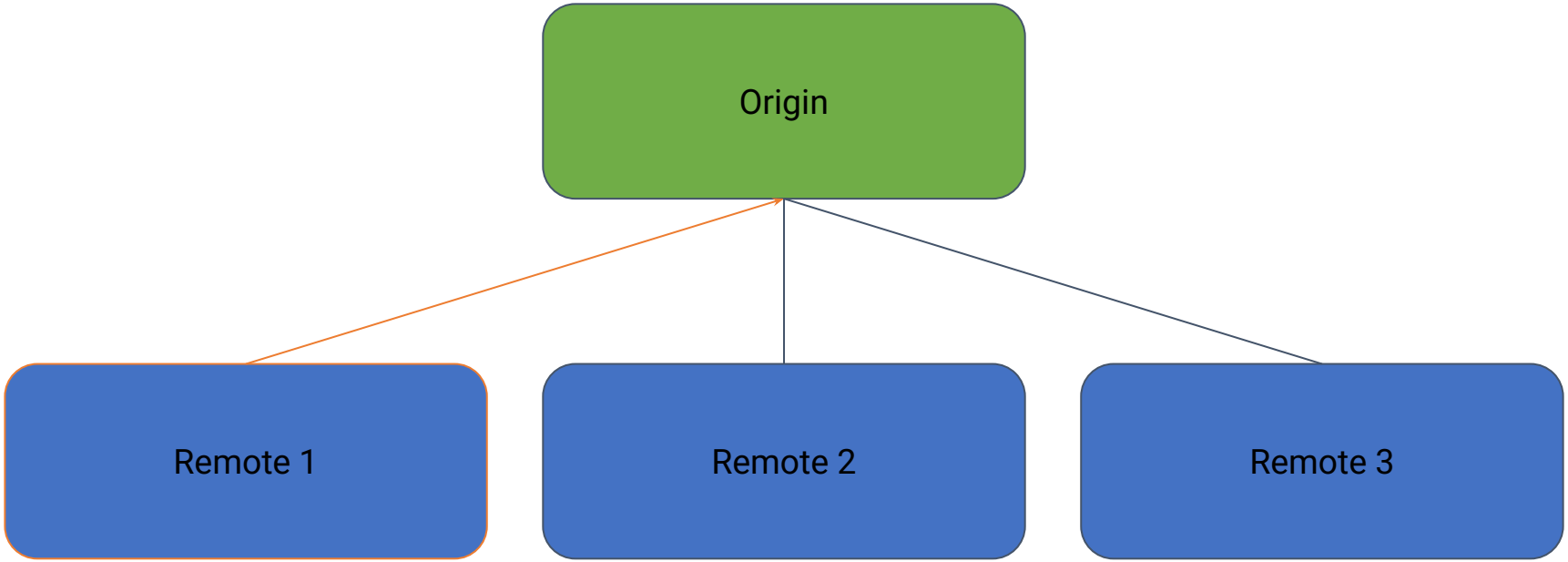
Adding Changes

```
n8w3rt@n8w3rt:~/robocup/firmware-training$ git add labs/*
n8w3rt@n8w3rt:~/robocup/firmware-training$ git status
On branch 2025-week-1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   labs/week-1/blink/.gitignore
        new file:   labs/week-1/blink/README.md
        new file:   labs/week-1/blink/diagram.json
        new file:   labs/week-1/blink/include/README
        new file:   labs/week-1/blink/lib/README
        new file:   labs/week-1/blink/platformio.ini
        new file:   labs/week-1/blink/src/main.cpp
        new file:   labs/week-1/blink/test/README
        new file:   labs/week-1/blink/wokwi.toml
```

Committing Changes

```
n8w3rt@n8w3rt:~/robocup/firmware-training$ git commit -m "Add week 1 lab training
material"
[2025-week-1 951f77a] Add week 1 lab training material
 9 files changed, 168 insertions(+)
 create mode 100644 labs/week-1/blink/.gitignore
 create mode 100644 labs/week-1/blink/README.md
 create mode 100644 labs/week-1/blink/diagram.json
 create mode 100644 labs/week-1/blink/include/README
 create mode 100644 labs/week-1/blink/lib/README
 create mode 100644 labs/week-1/blink/platformio.ini
 create mode 100644 labs/week-1/blink/src/main.cpp
 create mode 100644 labs/week-1/blink/test/README
 create mode 100644 labs/week-1/blink/wokwi.toml
```

Pushing Changes



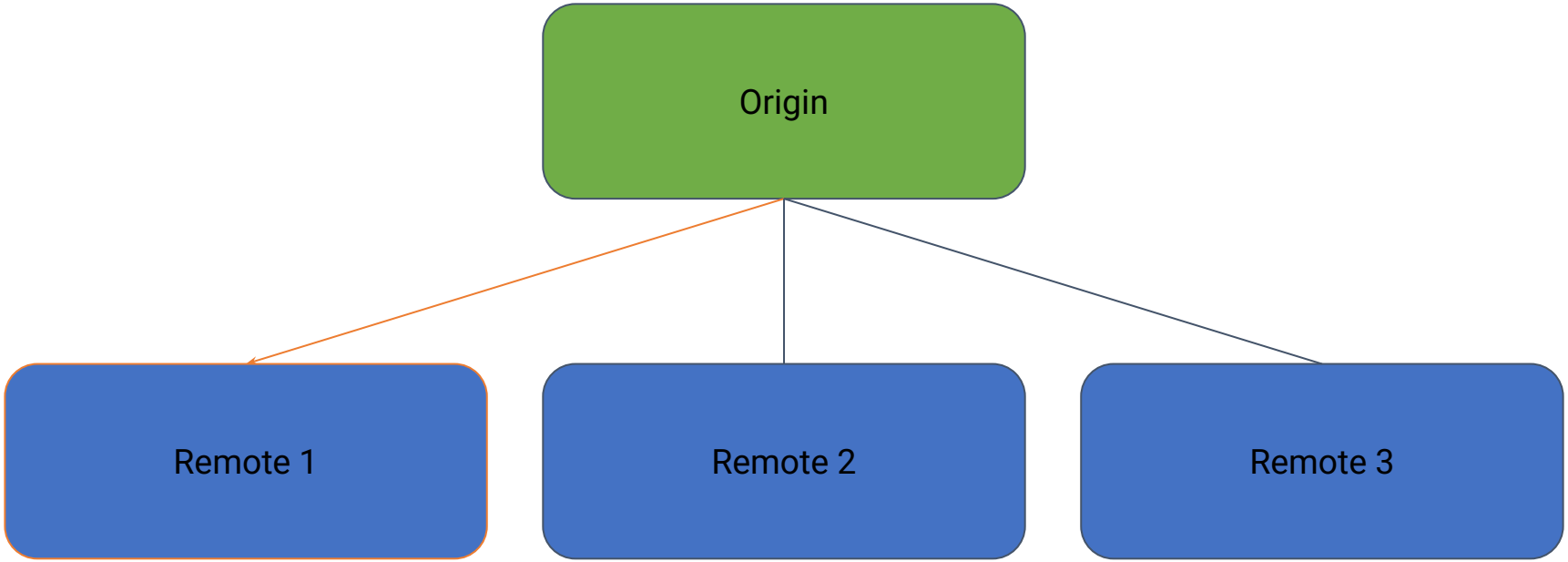
Pushing Changes

```
n8w3rt@n8w3rt:~/robocup/firmware-training$ git push
fatal: The current branch 2025-week-1 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin 2025-week-1

n8w3rt@n8w3rt:~/robocup/firmware-training$ git push --set-upstream origin 2025-wee
k-1
```

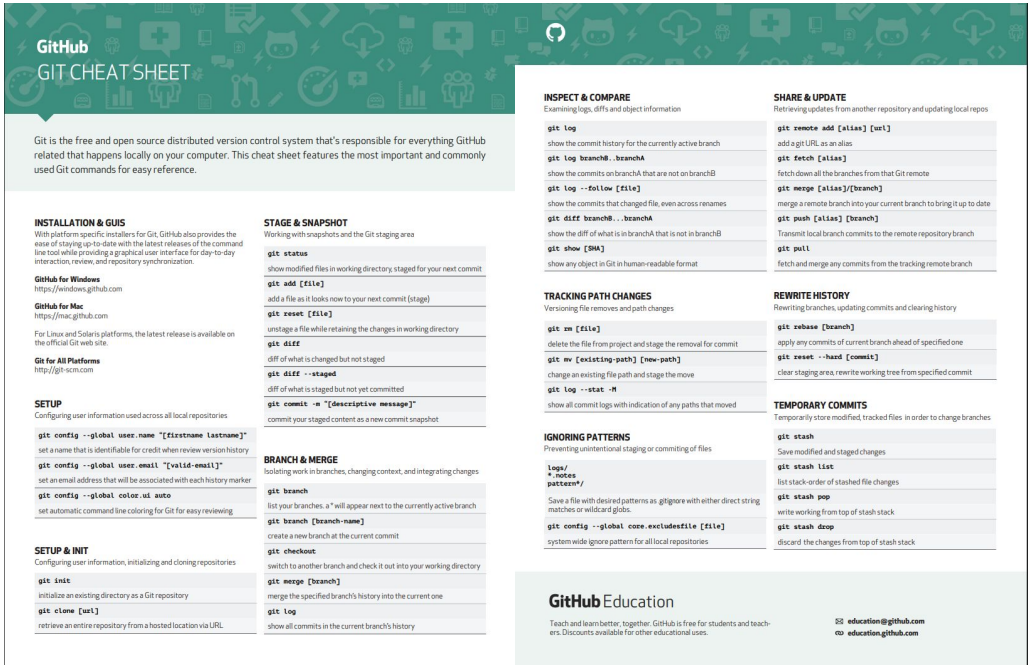
Pulling Changes



Pulling Changes

```
n8w3rt@n8w3rt:~/robocup/firmware-training$ git branch
  2025-week-1
* master
n8w3rt@n8w3rt:~/robocup/firmware-training$ git pull
Already up to date.
```

Cheatsheet



GitHub
GIT CHEAT SHEET

Git is the free and open source distributed version control system that's responsible for everything GitHub related that happens locally on your computer. This cheat sheet features the most important and commonly used Git commands for easy reference.

INSTALLATION & GUIS
With platform specific installers for Git, GitHub also provides the ease of staying up-to-date with the latest releases of the command line tool while providing a graphical user interface for day-to-day interaction, review, and repository synchronization.

GitHub for Windows
https://windows.github.com

GitHub for Mac
https://mac.github.com

For Linux and Solaris platforms, the latest release is available on the official Git web site.

Git for All Platforms
http://git-scm.com

SETUP
Configuring user information used across all local repositories

`git config --global user.name "[firstname lastname]"`
set a name that is identifiable for credit when review version history

`git config --global user.email "[valid-email]"`
set an email address that will be associated with each history marker

`git config --global color.ui auto`
set automatic command line coloring for Git for easy reviewing

SETUP & INIT
Configuring user information, initializing and cloning repositories

`git init`
initialize an existing directory as a Git repository

`git clone [url]`
retrieve an entire repository from a hosted location via URL

STAGE & SNAPSHOT
Working with snapshots and the Git staging area

`git status`
show modified files in working directory, staged for your next commit

`git add [file]`
add a file as it looks now to your next commit (stage)

`git reset [file]`
unstage a file while retaining the changes in working directory

`git diff`
diff of what is changed but not staged

`git diff --staged`
diff of what is staged but not yet committed

`git commit -m "[descriptive message]"`
commit your staged content as a new commit snapshot

BRANCH & MERGE
Isolating work in branches, changing context, and integrating changes

`git branch`
list your branches. a * will appear next to the currently active branch

`git branch [branch-name]`
create a new branch at the current commit

`git checkout`
switch to another branch and check it out into your working directory

`git merge [branch]`
merge the specified branch's history into the current one

`git log`
show all commits in the current branch's history

INSPECT & COMPARE
Examining logs, diffs and object information

`git log`
show the commit history for the currently active branch

`git log branchB..branchA`
show the commits on branchA that are not on branchB

`git log --follow [file]`
show the commits that changed file, even across renames

`git diff branchB...branchA`
show the diff of what is in branchA that is not in branchB

`git show [SHA]`
show any object in Git in human-readable format

TRACKING PATH CHANGES
Versioning file removes and path changes

`git rm [file]`
delete the file from project and stage the removal for commit

`git mv [existing-path] [new-path]`
change an existing file path and stage the move

`git log --stat -M`
show all commit logs with indication of any paths that moved

IGNORING PATTERNS
Preventing unintentional staging or commiting of files

```
logs/
*.notes
pattern*/
```

Save a file with desired patterns as .gitignore with either direct string matches or wildcard globs.

`git config --global core.excludesfile [file]`
system wide ignore pattern for all local repositories

SHARE & UPDATE
Retrieving updates from another repository and updating local repos

`git remote add [alias] [url]`
add a git URL as an alias

`git fetch [alias]`
fetch down all the branches from that Git remote

`git merge [alias]/[branch]`
merge a remote branch into your current branch to bring it up to date

`git push [alias] [branch]`
Transmit local branch commits to the remote repository branch

`git pull`
fetch and merge any commits from the tracking remote branch

REWRITE HISTORY
Rewriting branches, updating commits and clearing history

`git rebase [branch]`
apply any commits of current branch ahead of specified one

`git reset --hard [commit]`
clear staging area, rewrite working tree from specified commit

TEMPORARY COMMITS
Temporarily store modified, tracked files in order to change branches

`git stash`
Save modified and staged changes

`git stash list`
list stack-order of stashed file changes

`git stash pop`
write working from top of stash stack

`git stash drop`
discard the changes from top of stash stack

GitHub Education

Teach and learn better, together. GitHub is free for students and teachers. Discounts available for other educational uses.

education@github.com
education.github.com

Lab 1

- Clone robojackets-firmware
- Complete labs/week-1/blink/src/main.cpp
  - To build: `pio run`
  - To simulate: open diagram.json
  - To test: Grab an Arduino Nano