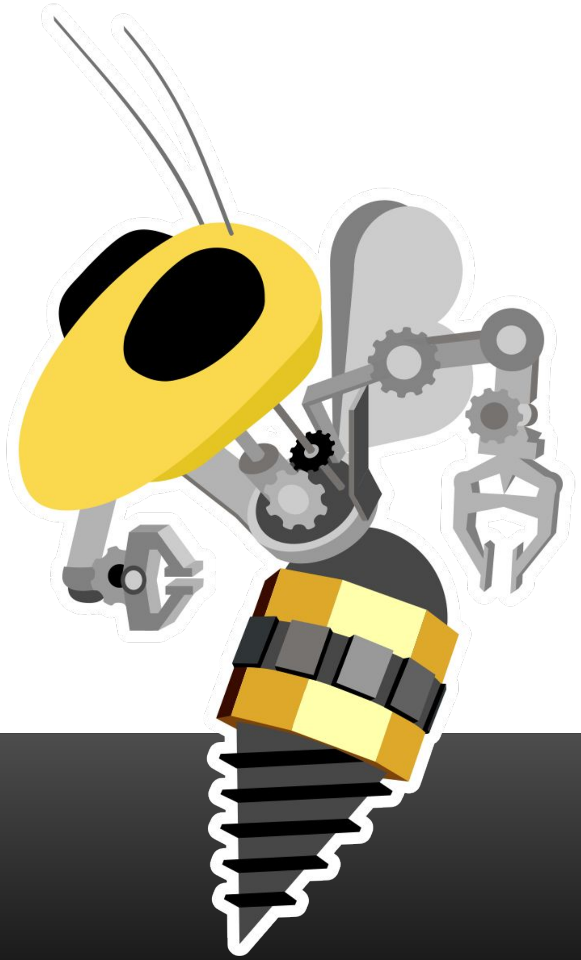
Welcome!

Firmware Training Week 2



RoboJackets

Competitive Robotics at Georgia Tech

www.robojackets.org

Last Week!

- Microcontrollers & Firmware
- Arduino, Part 1
- Prototyping

This Week!

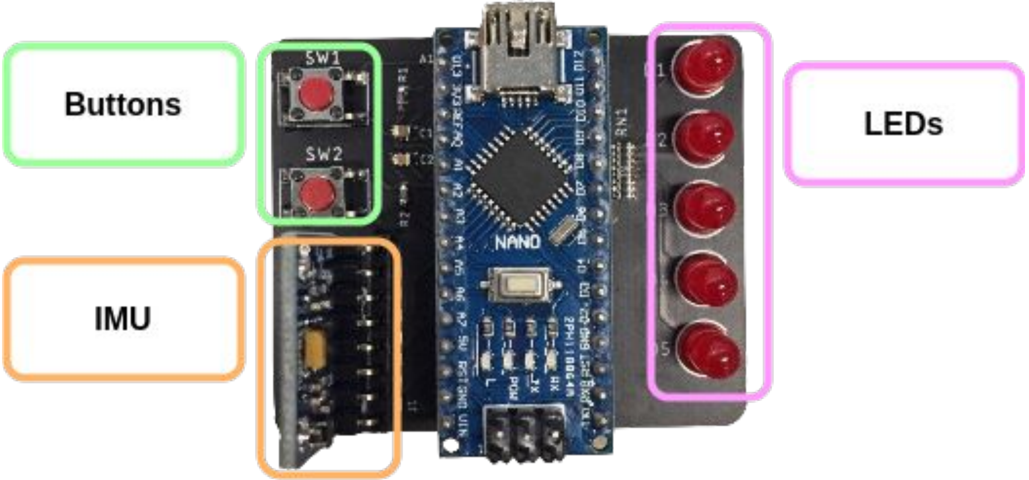
- Git / GitHub
- Arduino Reference
- Debugging
- Interrupts
- State Machines (will probably skip lol)

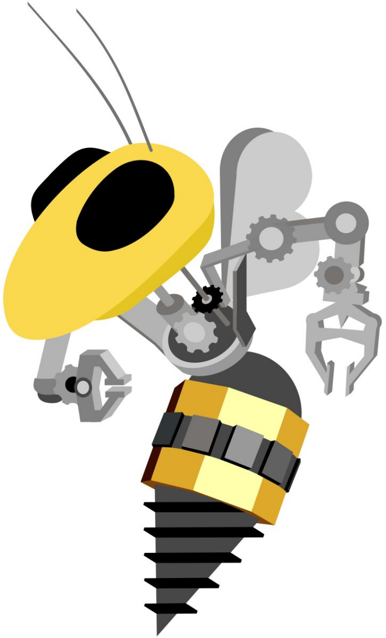
Firmware Training Board

What is this thing?

Firmware Training Board

No more breadboards

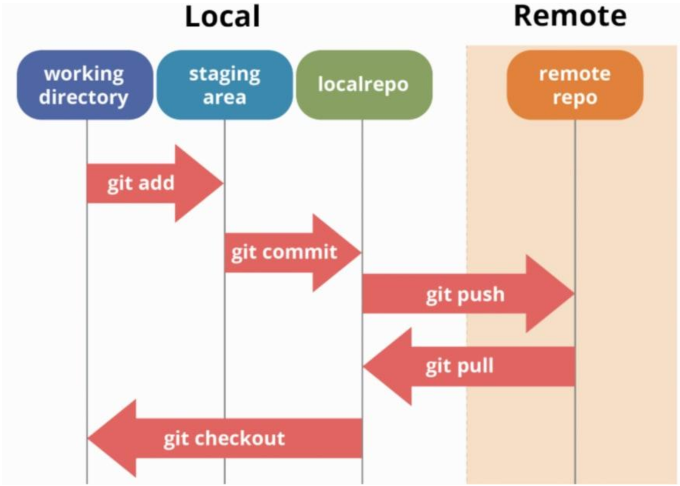




Git / GitHub

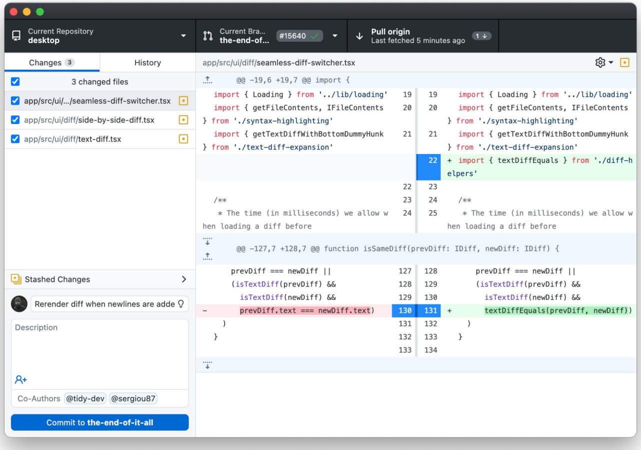
What is Git?

- Git = Version Control System (VCS)
  - Tracks changes to files over time
  - “Enables distributed, nonlinear workflows” - Wikipedia
- GitHub = website for hosting Git repositories



How do I use Git?

- Platforms
  - **GitHub Desktop (recommended)**
  - VS Code Plug-In
  - Command line



☑ 3 changed files

☑ app/src/ui/.../seamless-diff-switcher.tsx

☑ app/src/ui/diff/side-by-side-diff.tsx

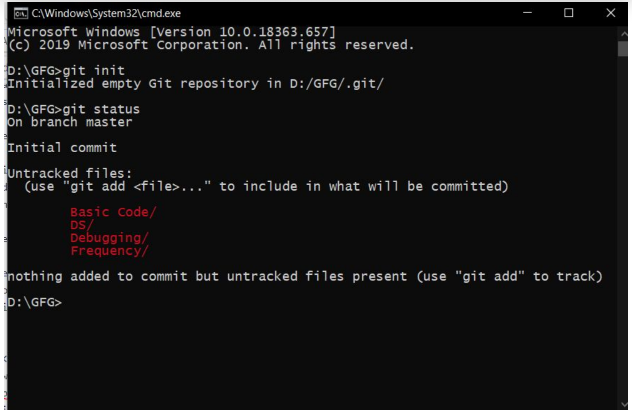
☑ app/src/ui/diff/text-diff.tsx

Rerender diff when newlines are adde

Co-Authors @tidy-dev @sergiou87

app/src/ui/diff/seamless-diff-switcher.tsx

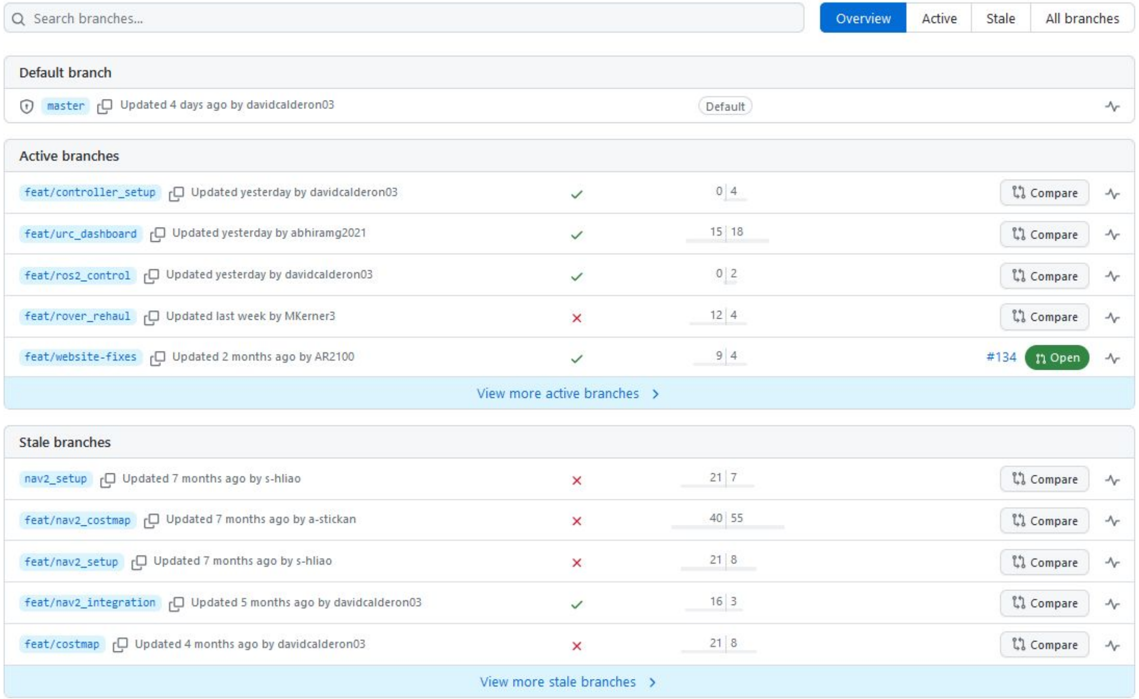
```
@@ -19,6 +19,7 @@ import {
  import { Loading } from '../lib/loading'            import { Loading } from '../lib/loading'
  import { getFileContents, IFileContents            import { getFileContents, IFileContents
} from './syntax-highlighting'                      } from './syntax-highlighting'
  import { getTextDiffWithBottomDummyHunk            import { getTextDiffWithBottomDummyHunk
} from './text-diff-expansion'                      } from './text-diff-expansion'
                                                    + import { textDiffEquals } from './diff-h
                                                    elpers'

  /**                                                 /**
   * The time (in milliseconds) we allow w             * The time (in milliseconds) we allow w
hen loading a diff before                           hen loading a diff before

@@ -127,7 +128,7 @@ function isSameDiff(prevDiff: IDiff, newDiff: IDiff) {
      prevDiff === newDiff ||                             prevDiff === newDiff ||
      (isTextDiff(prevDiff) &&                            (isTextDiff(prevDiff) &&
        isTextDiff(newDiff) &&                              isTextDiff(newDiff) &&
-       prevDiff.text === newDiff.text)             +       textDiffEquals(prevDiff, newDiff))
    )                                                   )
  }                                                   }
```



```
Microsoft Windows [Version 10.0.18363.657]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\GFG>git init
Initialized empty Git repository in D:/GFG/.git/

D:\GFG>git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Basic Code/
        DS/
        Debugging/
        Frequency/

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

D:\GFG>
```

How do I use Git?

- Core ideas
  - **Branch**



Default branch

master	Updated 4 days ago by davidcalderon03	Default	

Active branches

feat/controller_setup	Updated yesterday by davidcalderon03	✓	0 | 4	Compare
feat/urc_dashboard	Updated yesterday by abhiramg2021	✓	15 | 18	Compare
feat/ros2_control	Updated yesterday by davidcalderon03	✓	0 | 2	Compare
feat/rover_rehaul	Updated last week by MKerner3	✗	12 | 4	Compare
feat/website-fixes	Updated 2 months ago by AR2100	✓	9 | 4	#134 Open

View more active branches ›

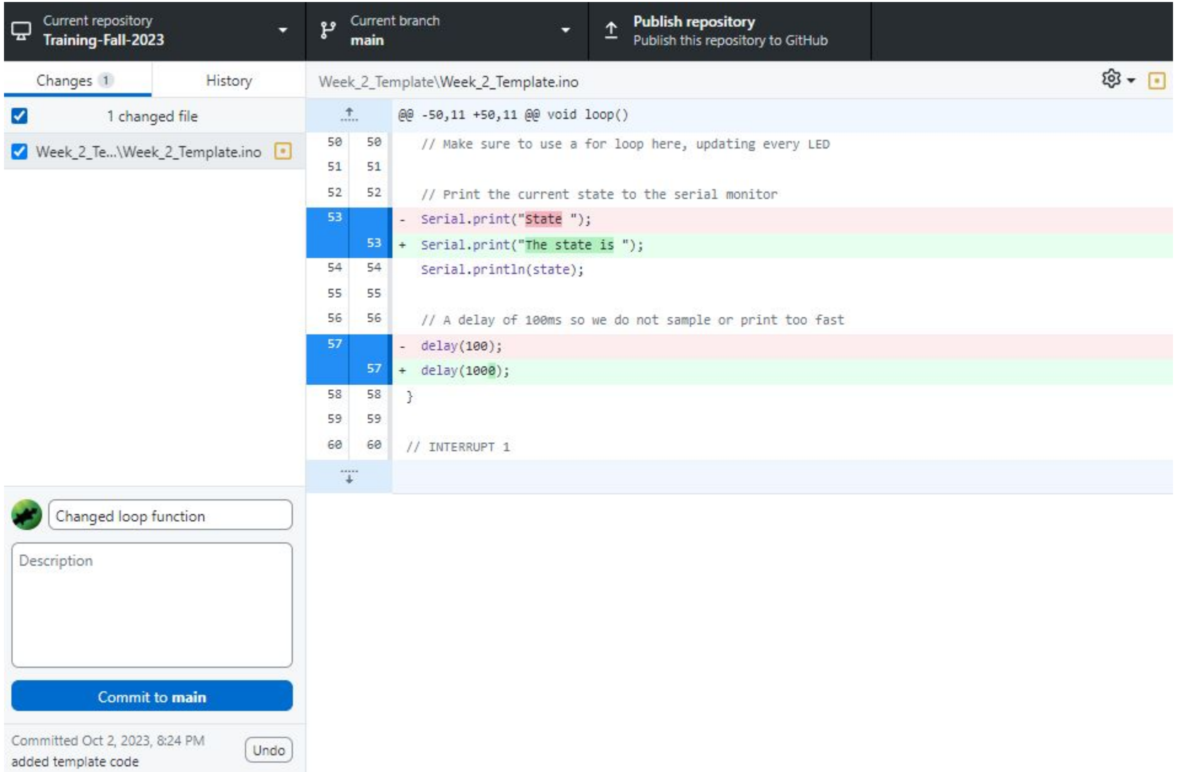
Stale branches

nav2_setup	Updated 7 months ago by s-hliao	✗	21 | 7	Compare
feat/nav2_costmap	Updated 7 months ago by a-stickan	✗	40 | 55	Compare
feat/nav2_setup	Updated 7 months ago by s-hliao	✗	21 | 8	Compare
feat/nav2_integration	Updated 5 months ago by davidcalderon03	✓	16 | 3	Compare
feat/costmap	Updated 4 months ago by davidcalderon03	✗	21 | 8	Compare

View more stale branches ›

How do I use Git?

- Core ideas
  - Branch
  - **Commit**



```
@@ -50,11 +50,11 @@ void loop()
   // Make sure to use a for loop here, updating every LED

   // Print the current state to the serial monitor
-  Serial.print("State ");
+  Serial.print("The state is ");
   Serial.println(state);

   // A delay of 100ms so we do not sample or print too fast
-  delay(100);
+  delay(1000);
 }

 // INTERRUPT 1
```

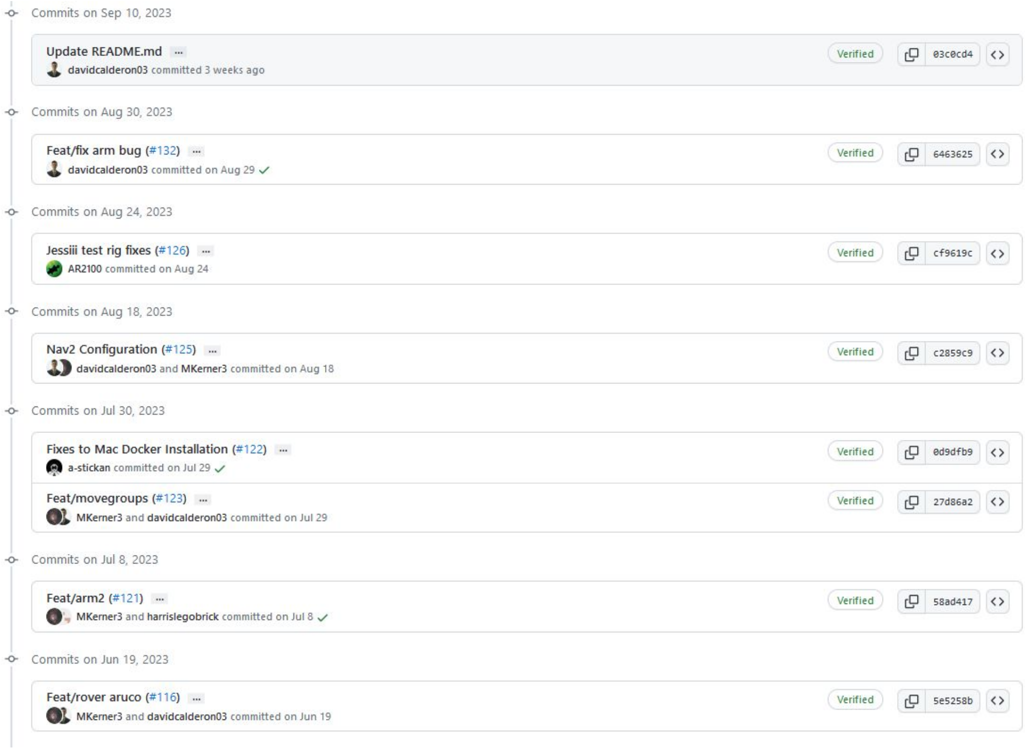
Changed loop function

Description

Committed Oct 2, 2023, 8:24 PM
added template code

How do I use Git?

- Core ideas
  - Branch
  - **Commit**
    - **Commit History**



Commits on Sep 10, 2023

Update README.md ...
davidcalderon03 committed 3 weeks ago
Verified 03c0cd4

Commits on Aug 30, 2023

Feat/fix arm bug (#132) ...
davidcalderon03 committed on Aug 29 ✓
Verified 6463625

Commits on Aug 24, 2023

Jessiii test rig fixes (#126) ...
AR2100 committed on Aug 24
Verified cf9619c

Commits on Aug 18, 2023

Nav2 Configuration (#125) ...
davidcalderon03 and MKerner3 committed on Aug 18
Verified c2859c9

Commits on Jul 30, 2023

Fixes to Mac Docker Installation (#122) ...
a-stickan committed on Jul 29 ✓
Verified 0d9dfb9

Feat/movegroups (#123) ...
MKerner3 and davidcalderon03 committed on Jul 29
Verified 27d86a2

Commits on Jul 8, 2023

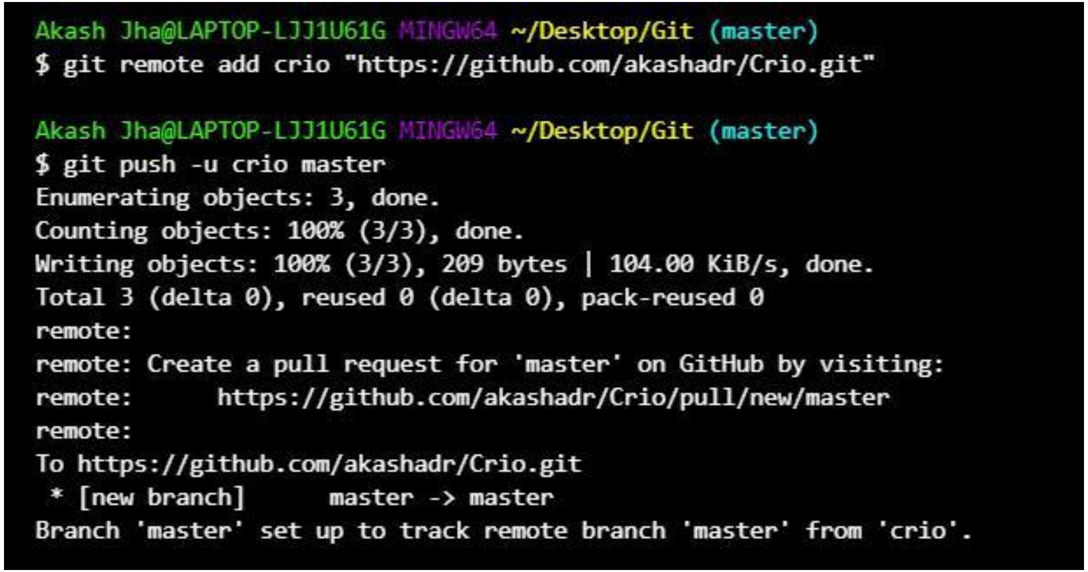
Feat/arm2 (#121) ...
MKerner3 and harrislegobrick committed on Jul 8 ✓
Verified 58ad417

Commits on Jun 19, 2023

Feat/rover aruco (#116) ...
MKerner3 and davidcalderon03 committed on Jun 19
Verified 5e5258b

How do I use Git?

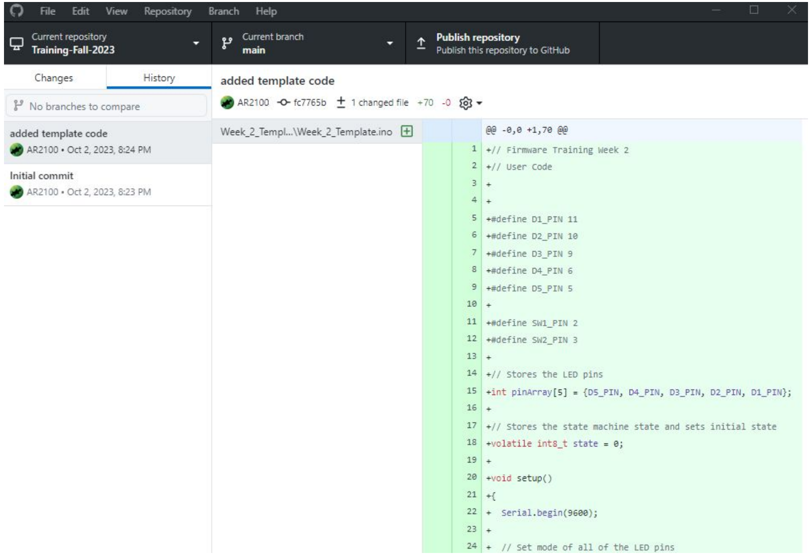
- Core ideas
    - Branch
    - Commit & Commit History
    - **Push / Pull**



```
Akash Jha@LAPTOP-LJJ1U61G MINGW64 ~/Desktop/Git (master)
$ git remote add crio "https://github.com/akashadr/Crio.git"

Akash Jha@LAPTOP-LJJ1U61G MINGW64 ~/Desktop/Git (master)
$ git push -u crio master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 209 bytes | 104.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/akashadr/Crio/pull/new/master
remote:
To https://github.com/akashadr/Crio.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'crio'.
```

Try Git Out!

1. Download GitHub Desktop
   a. Signing in with GitHub isn't required
2. Initialize an empty repository for RoboJackets Firmware Training
3. Download the **Week_2_Template.ino** file into your repository
4. Commit your changes!



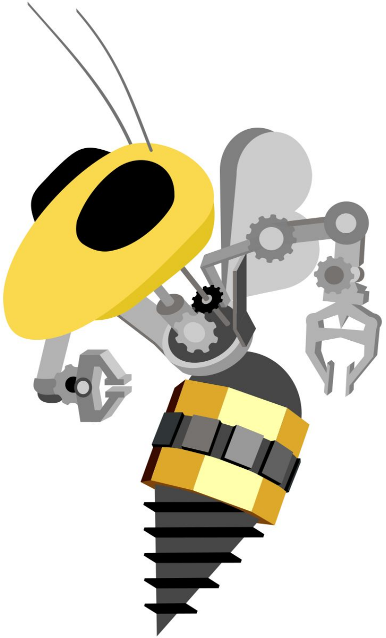
added template code
AR2100 • Oct 2, 2023, 8:24 PM

Initial commit
AR2100 • Oct 2, 2023, 8:23 PM

added template code
AR2100 fc7765b 1 changed file +70 -0

Week_2_Templ...\Week_2_Template.ino

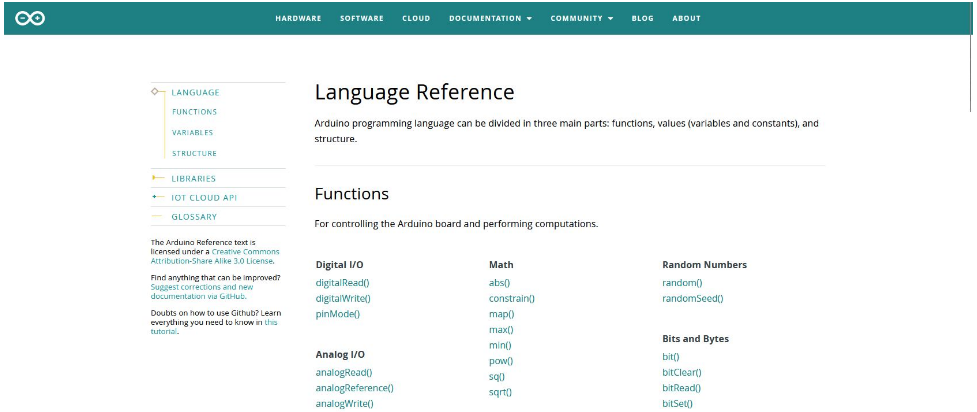
```
@@ -0,0 +1,70 @@
+// Firmware Training Week 2
+// User Code
+
+
+#define D1_PIN 11
+#define D2_PIN 10
+#define D3_PIN 9
+#define D4_PIN 6
+#define D5_PIN 5
+
+#define SW1_PIN 2
+#define SW2_PIN 3
+
+// Stores the LED pins
+int pinArray[5] = {D5_PIN, D4_PIN, D3_PIN, D2_PIN, D1_PIN};
+
+// Stores the state machine state and sets initial state
+volatile int8_t state = 0;
+
+void setup()
+{
+  Serial.begin(9600);
+
+  // Set mode of all of the LED pins
```



Arduino Reference

Use the Arduino Reference!

- https://www.arduino.cc/reference/en/
  - Q: How does this function work?
  - A: Refer to the documentation!



The Arduino Reference text is licensed under a Creative Commons Attribution-Share Alike 3.0 License.

Find anything that can be improved? Suggest corrections and new documentation via GitHub.

Doubts on how to use Github? Learn everything you need to know in this tutorial.

Language Reference

Arduino programming language can be divided in three main parts: functions, values (variables and constants), and structure.

Functions

For controlling the Arduino board and performing computations.

Digital I/O

digitalRead()
digitalWrite()
pinMode()

Analog I/O

analogRead()
analogReference()
analogWrite()

Math

abs()
constrain()
map()
max()
min()
pow()
sq()
sqrt()

Random Numbers

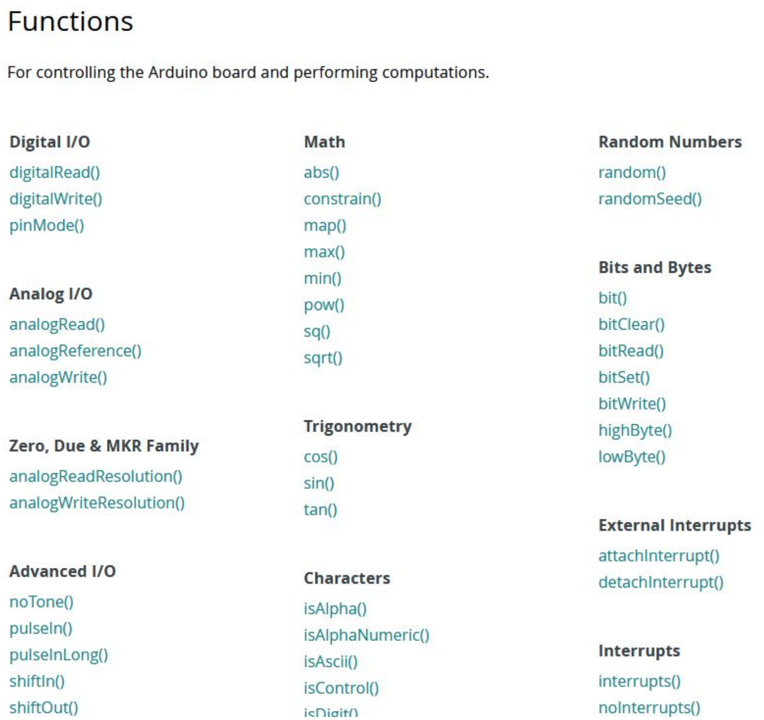
random()
randomSeed()

Bits and Bytes

bit()
bitClear()
bitRead()
bitSet()

Functions

- Arduino-specific functions
  - Interfacing with I/O
  - Helper functions



Functions

For controlling the Arduino board and performing computations.

Digital I/O

digitalRead()
digitalWrite()
pinMode()

Analog I/O

analogRead()
analogReference()
analogWrite()

Zero, Due & MKR Family

analogReadResolution()
analogWriteResolution()

Advanced I/O

noTone()
pulseIn()
pulseInLong()
shiftIn()
shiftOut()

Math

abs()
constrain()
map()
max()
min()
pow()
sq()
sqrt()

Trigonometry

cos()
sin()
tan()

Characters

isAlpha()
isAlphaNumeric()
isAscii()
isControl()
isDigit()

Random Numbers

random()
randomSeed()

Bits and Bytes

bit()
bitClear()
bitRead()
bitSet()
bitWrite()
highByte()
lowByte()

External Interrupts

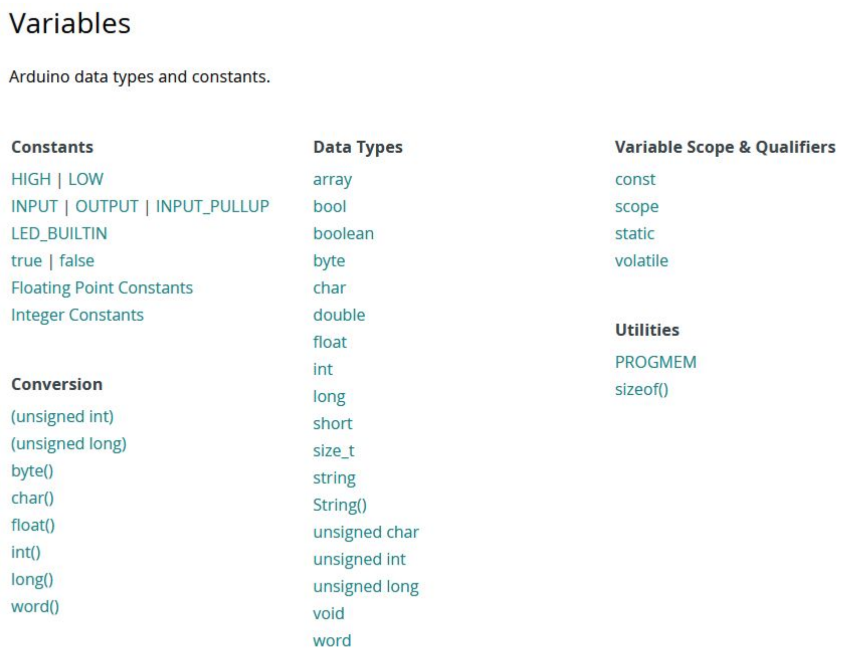
attachInterrupt()
detachInterrupt()

Interrupts

interrupts()
noInterrupts()

Variables

- Variables = data storage containers
    - Most data types are from C/C++
    - Some constants are Arduino-specific



Variables

Arduino data types and constants.

Constants

HIGH | LOW
INPUT | OUTPUT | INPUT_PULLUP
LED_BUILTIN
true | false
Floating Point Constants
Integer Constants

Conversion

(unsigned int)
(unsigned long)
byte()
char()
float()
int()
long()
word()

Data Types

array
bool
boolean
byte
char
double
float
int
long
short
size_t
string
String()
unsigned char
unsigned int
unsigned long
void
word

Variable Scope & Qualifiers

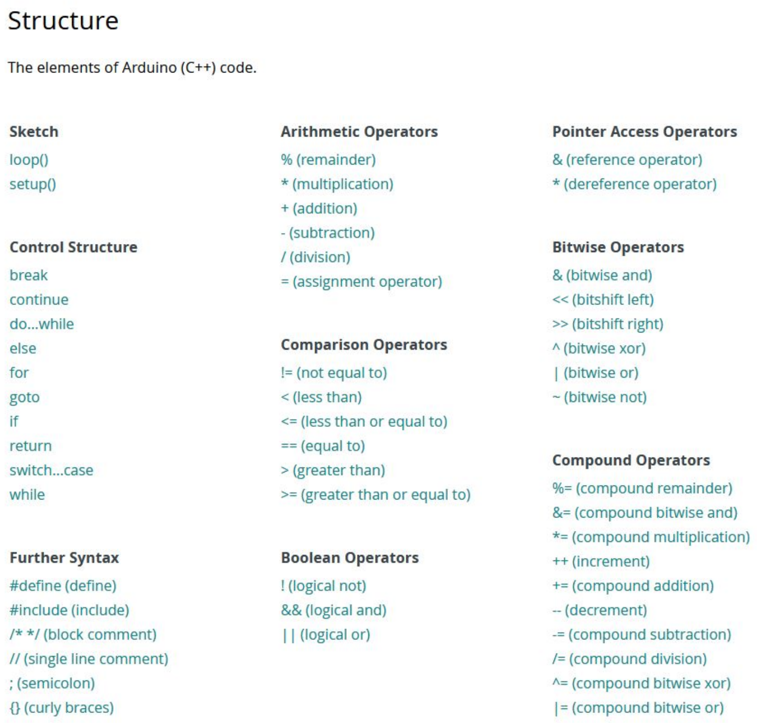
const
scope
static
volatile

Utilities

PROGMEM
sizeof()

Structure

- How to get work done!
  - setup() and loop() are Arduino-specific
  - Everything else is from C / C++



Structure

The elements of Arduino (C++) code.

Sketch

loop()
setup()

Control Structure

break
continue
do...while
else
for
goto
if
return
switch...case
while

Further Syntax

#define (define)
#include (include)
/* */ (block comment)
// (single line comment)
; (semicolon)
{} (curly braces)

Arithmetic Operators

% (remainder)
* (multiplication)
+ (addition)
- (subtraction)
/ (division)
= (assignment operator)

Comparison Operators

!= (not equal to)
< (less than)
<= (less than or equal to)
== (equal to)
> (greater than)
>= (greater than or equal to)

Boolean Operators

! (logical not)
&& (logical and)
| | (logical or)

Pointer Access Operators

& (reference operator)
* (dereference operator)

Bitwise Operators

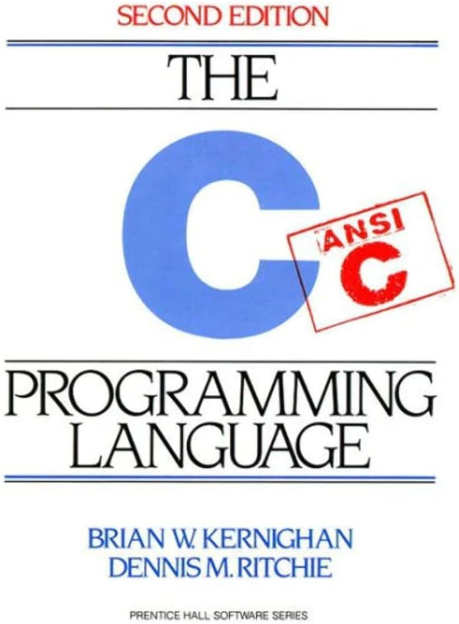
& (bitwise and)
<< (bitshift left)
>> (bitshift right)
^ (bitwise xor)
| (bitwise or)
~ (bitwise not)

Compound Operators

%= (compound remainder)
&= (compound bitwise and)
*= (compound multiplication)
++ (increment)
+= (compound addition)
-- (decrement)
-= (compound subtraction)
/= (compound division)
^= (compound bitwise xor)
|= (compound bitwise or)

A Note About Arduino / C / C++

- Arduino is based on C++
  - Most beginner sketches are written like C
  - Arduino Libraries take advantage of C++ features
- I can’t teach C or C++ in several short training sessions
  - C: “The C Programming Language” by K&R
  - C++: Huge language, not sure what to recommend



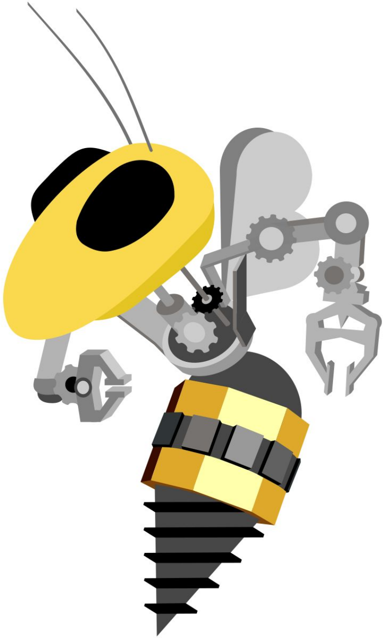
SECOND EDITION

THE C PROGRAMMING LANGUAGE

ANSI C

BRIAN W. KERNIGHAN
DENNIS M. RITCHIE

PRENTICE HALL SOFTWARE SERIES



Debugging

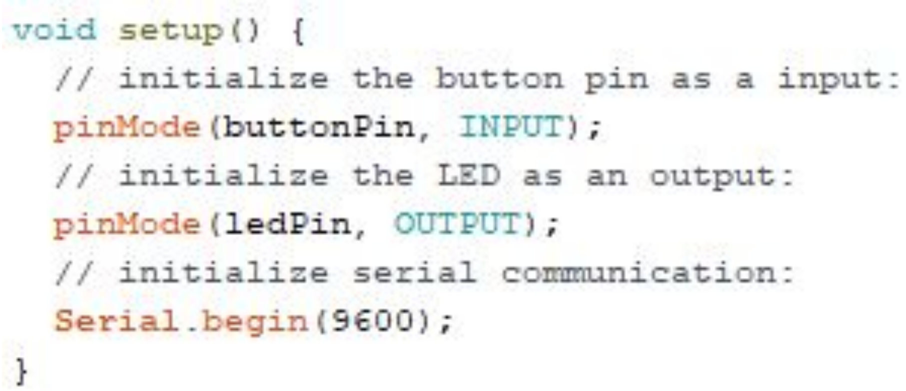
Debugging: When things go wrong…

- With firmware, you have to consider multiple sources of error
  - Faulty wiring
  - Incorrect hardware configuration
  - Logic Error
  - Memory Error
  - Data Parsing Error

OKAY, HUMAN.
HUH?
BEFORE YOU HIT 'COMPILE,' LISTEN UP.
YOU KNOW WHEN YOU'RE FALLING ASLEEP, AND YOU IMAGINE YOURSELF WALKING OR SOMETHING,
AND SUDDENLY YOU MISSTEP, STUMBLE, AND JOLT AWAKE?
YEAH!
WELL, THAT'S WHAT A SEGFAULT FEELS LIKE.
DOUBLE-CHECK YOUR DAMN POINTERS, OKAY?

Print Debugging

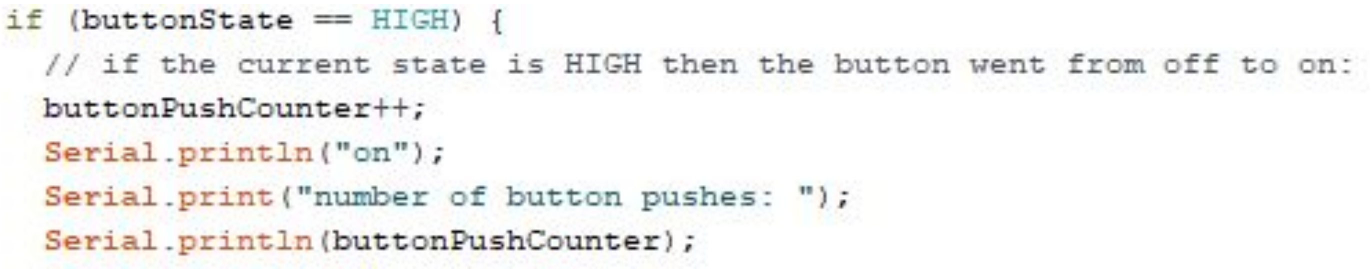
- Simplest form of Debugging
  - **Serial.begin(baud)**



```
void setup() {
  // initialize the button pin as a input:
  pinMode(buttonPin, INPUT);
  // initialize the LED as an output:
  pinMode(ledPin, OUTPUT);
  // initialize serial communication:
  Serial.begin(9600);
}
```

Print Debugging

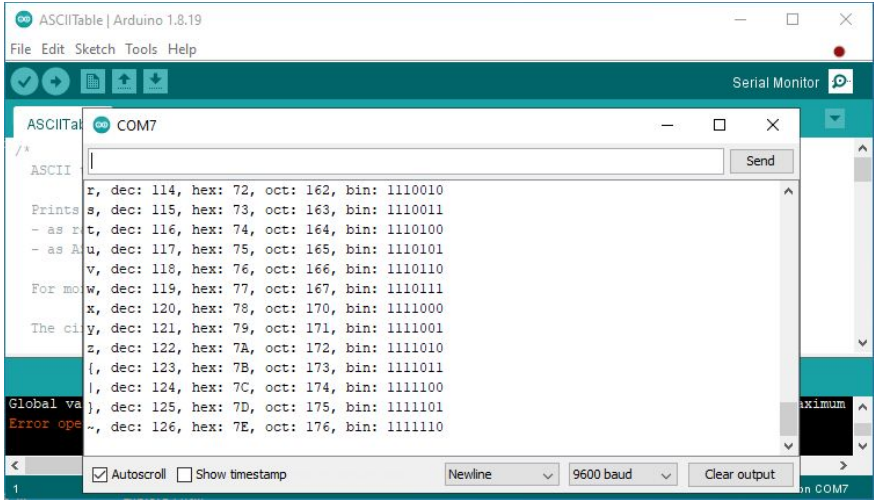
- Simplest form of Debugging
  - Serial.begin(baud)
  - **Serial.print() / Serial.println()**



```
if (buttonState == HIGH) {
  // if the current state is HIGH then the button went from off to on:
  buttonPushCounter++;
  Serial.println("on");
  Serial.print("number of button pushes: ");
  Serial.println(buttonPushCounter);
```

Print Debugging

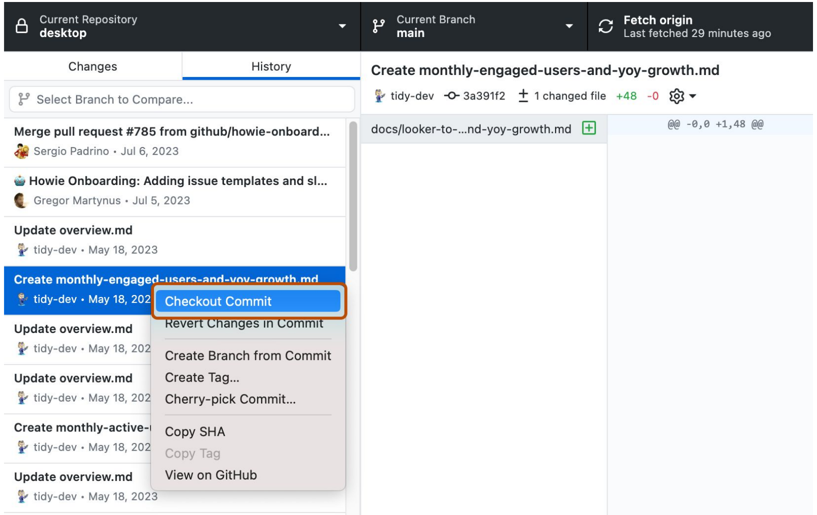
- Simplest form of Debugging
  - Serial.begin(baud)
  - Serial.print() / Serial.println()
  - **Serial Monitor**
    - **Baud rate must match!**



```
r, dec: 114, hex: 72, oct: 162, bin: 1110010
s, dec: 115, hex: 73, oct: 163, bin: 1110011
t, dec: 116, hex: 74, oct: 164, bin: 1110100
u, dec: 117, hex: 75, oct: 165, bin: 1110101
v, dec: 118, hex: 76, oct: 166, bin: 1110110
w, dec: 119, hex: 77, oct: 167, bin: 1110111
x, dec: 120, hex: 78, oct: 170, bin: 1111000
y, dec: 121, hex: 79, oct: 171, bin: 1111001
z, dec: 122, hex: 7A, oct: 172, bin: 1111010
{, dec: 123, hex: 7B, oct: 173, bin: 1111011
|, dec: 124, hex: 7C, oct: 174, bin: 1111100
}, dec: 125, hex: 7D, oct: 175, bin: 1111101
~, dec: 126, hex: 7E, oct: 176, bin: 1111110
```

Tips for Finding Bugs

- Trace carefully!
    - Use print statements to keep track of variable values
- Test Early, Test Often
    - Make incremental changes, test them out frequently
    - Comment out code
    - Use Version control!



Merge pull request #785 from github/howie-onboard...
Sergio Padrino • Jul 6, 2023

Howie Onboarding: Adding issue templates and sl...
Gregor Martynus • Jul 5, 2023

Update overview.md
tidy-dev • May 18, 2023

Create monthly-engaged-users-and-yoy-growth.md
tidy-dev • May 18, 202

Update overview.md
tidy-dev • May 18, 202

Update overview.md
tidy-dev • May 18, 202

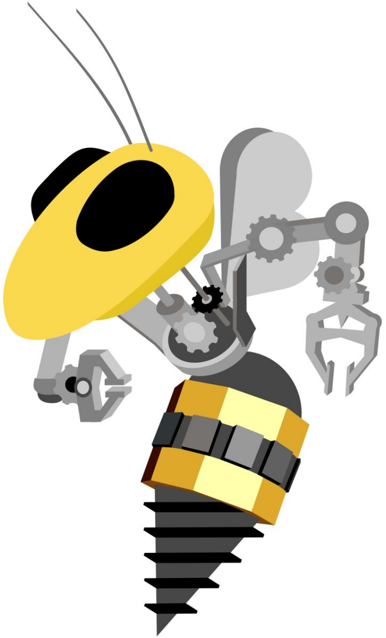
Create monthly-active-
tidy-dev • May 18, 202

Update overview.md
tidy-dev • May 18, 2023

Create monthly-engaged-users-and-yoy-growth.md
tidy-dev 3a391f2 1 changed file +48 -0

docs/looker-to-...nd-yoy-growth.md

```
@@ -0,0 +1,48 @@
```



Interrupts

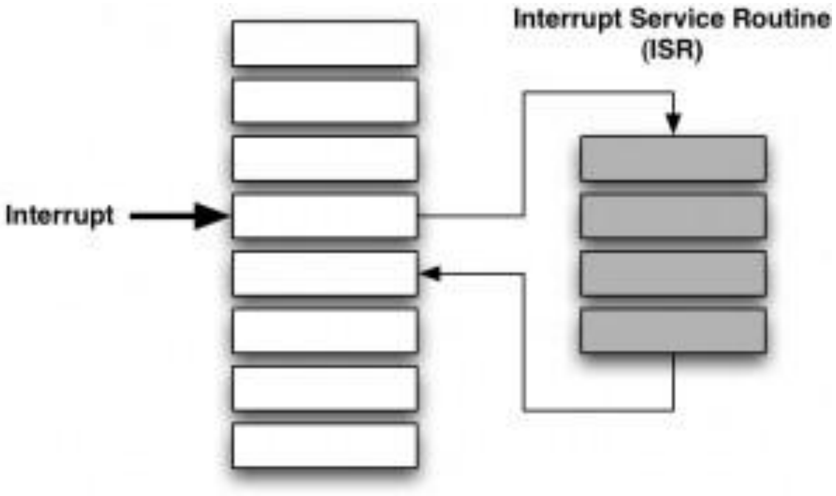
Hey Mom. Mom. Mom. Mom. Mo- WHAT!

What are Interrupts

- Mechanism built into processors to run a function when an event occurs
  - Can be hardware (a pin) or software (a timer)
- The function that gets called is known as an ISR (interrupt service routine)
- It stops (interrupts) the main code before returning back to the main code

Using Interrupts

- Allows us to not waste time checking if a device is ready (polling)
- Instead the device just tells us it is ready (interrupts)
- Returns to our normal code right after



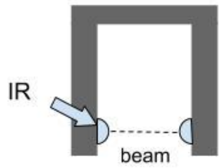
Arduino and Interrupts

- Arduino provides the attach interrupt method for this
  - Triggers an interrupt when the signal to an input pin changes
  - *attachInterrupt(pin, ISR, trigger mode)*
- Mode:
  - Rising/Falling/Change Edge trigger
  - Calls ISR on 0 to 1 change, 1 to 0 change, or both

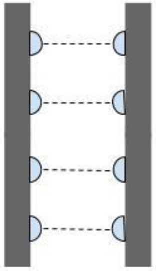
RoboCup Ball Speed Example

- Want to calculate the speed of the ball after the robot kicks ball
- IR sensors determine when the ball has passed
  - Using the time this occurs and distance between sensors can calculate average speed

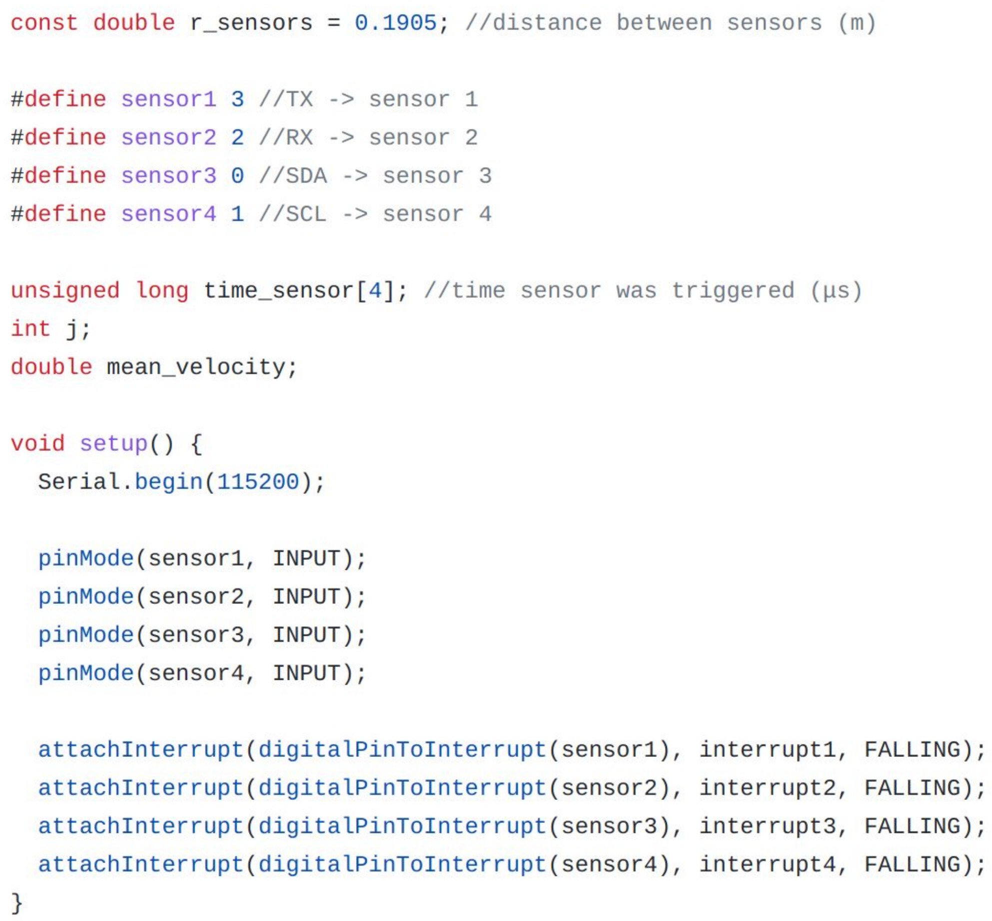
Front View



Top View

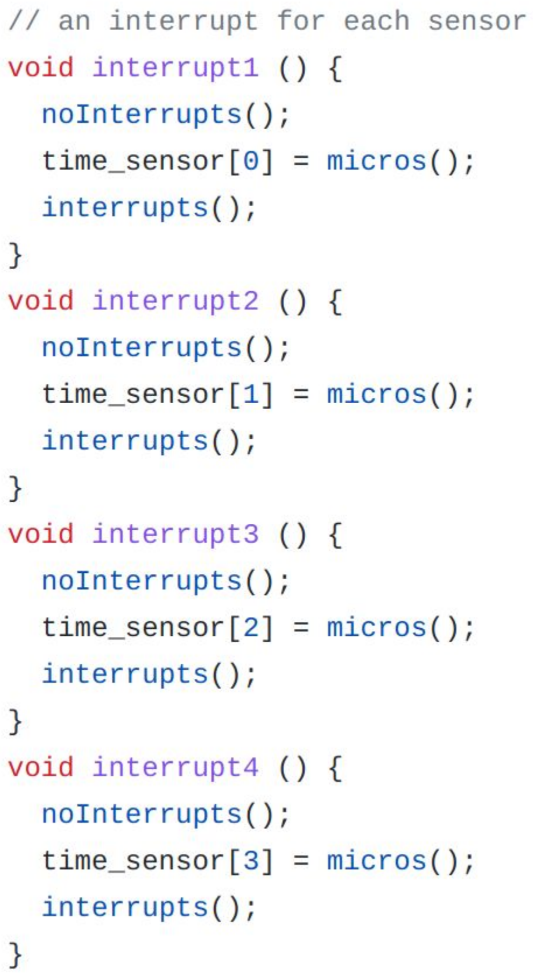


Code: Interrupt Setup

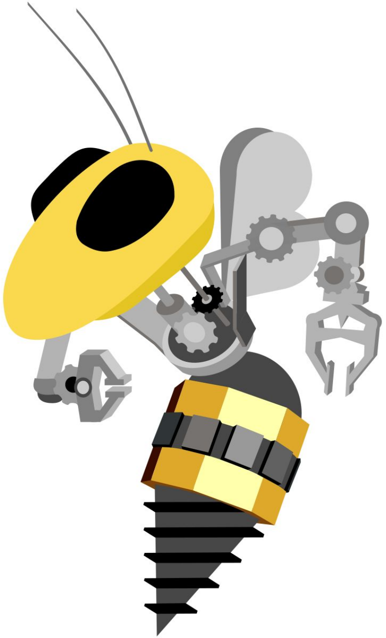


```
const double r_sensors = 0.1905; //distance between sensors (m)

#define sensor1 3 //TX -> sensor 1
#define sensor2 2 //RX -> sensor 2
#define sensor3 0 //SDA -> sensor 3
#define sensor4 1 //SCL -> sensor 4

unsigned long time_sensor[4]; //time sensor was triggered (µs)
int j;
double mean_velocity;

void setup() {
  Serial.begin(115200);

  pinMode(sensor1, INPUT);
  pinMode(sensor2, INPUT);
  pinMode(sensor3, INPUT);
  pinMode(sensor4, INPUT);

  attachInterrupt(digitalPinToInterrupt(sensor1), interrupt1, FALLING);
  attachInterrupt(digitalPinToInterrupt(sensor2), interrupt2, FALLING);
  attachInterrupt(digitalPinToInterrupt(sensor3), interrupt3, FALLING);
  attachInterrupt(digitalPinToInterrupt(sensor4), interrupt4, FALLING);
}
```

Code: Creating the ISR



```
// an interrupt for each sensor
void interrupt1 () {
  noInterrupts();
  time_sensor[0] = micros();
  interrupts();
}
void interrupt2 () {
  noInterrupts();
  time_sensor[1] = micros();
  interrupts();
}
void interrupt3 () {
  noInterrupts();
  time_sensor[2] = micros();
  interrupts();
}
void interrupt4 () {
  noInterrupts();
  time_sensor[3] = micros();
  interrupts();
}
```

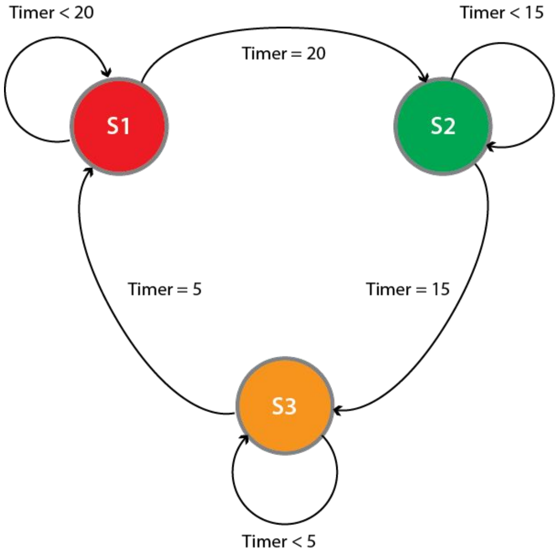


State Machines

“Ferst Driiive, walk sign is on to cross, Ferst Driiiive”

What is a State Machine?

- Stores a status, or state, at any given time
- Takes in inputs and gives outputs to interact with other devices
- Switches states based on current states and inputs



Traffic Light State Machine

Purpose of a State Machine

- Provide a way to use a sequence or history of inputs, not just current input values
- Restrict behavior of a system to certain actions based on a variety of inputs
- Provide a sort of memory - tied to history of inputs
  - Combination lock
  - Traffic signals && crosswalks
  - Robots!

Usages of State Machines

- Making sure things happen in the right order (startup)
- Controlling behavior of robots better - output based off of states (stable), not directly by inputs
- A CPU (and a Microcontroller)!

Types of State Machines

- **Moore state machine** - outputs are based only on current state
- Mealy state machine - outputs are based on the current state and the input (transitions)
- We typically use a Moore State Machine

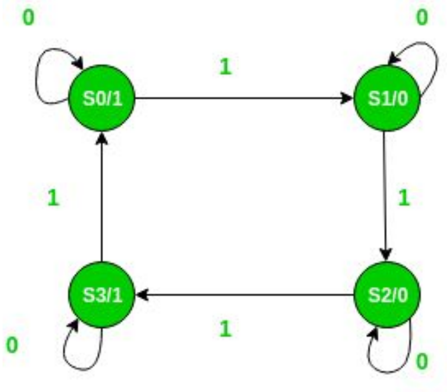


Figure - Moore machine

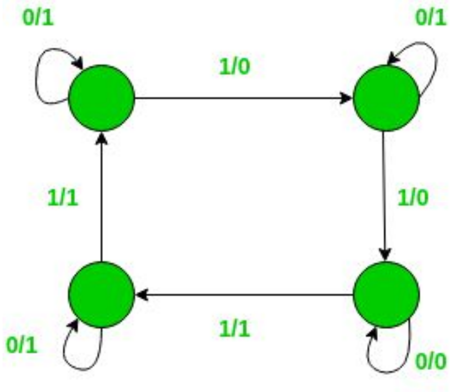
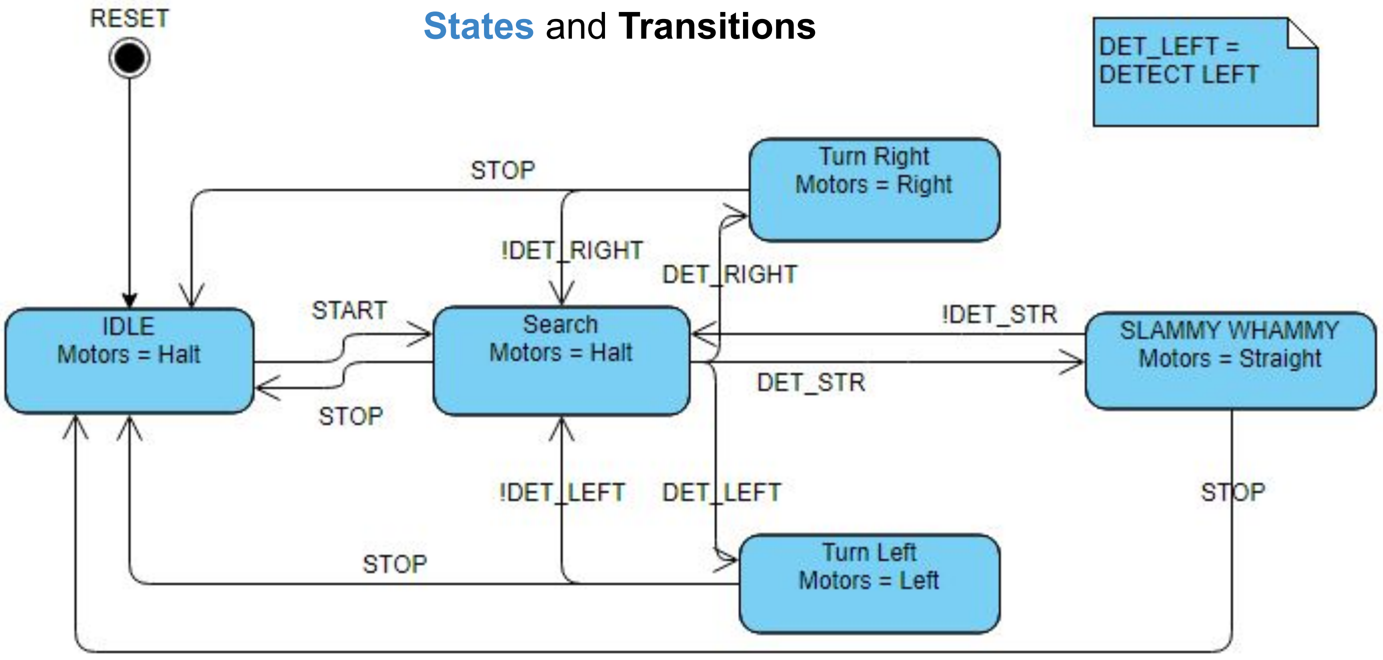


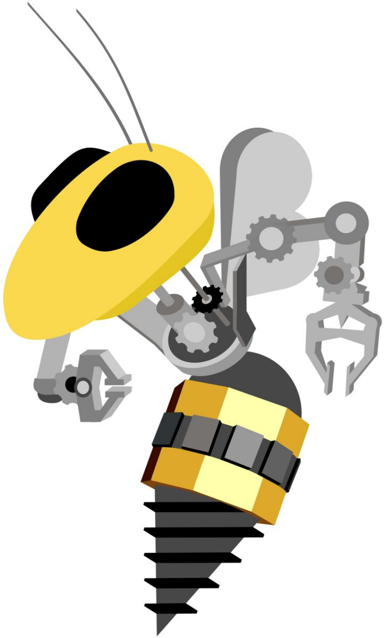
Figure - Mealy machine

Inputs and Outputs

- Inputs frequently include sensors or buttons
  - Prefer boolean state transition
- Outputs can be motors or LEDs
  - Behaviors (running motors, running code) of the robot

Example Diagram (RoboWrestling)





Lab Time

Lab Info

- Create a counter state machine
- Write interrupts for each button
    - One to count up
    - One to count down
- Implement state machine
- Display state using board LEDs