Intro to Programming

(for Arduino)

Files will be here:

github.com/tommymcdonald

The Arduino Language

Arduino is a set of C/C++ functions that can be compiled directly to your board via the IDE.

All standard C/C++ functions and constructs can be used in sync with the Arduino language.

What is C?

C is crazy. That is what.

Variables

What is a variable? According to wikipedia:

“Variable (computer science) In computer **programming**, a **variable** or scalar is a storage location and an associated symbolic name (an identifier) which contains some known or unknown quantity or information, a value.”

???

Let’s look at an example in Arduino code...

```
int led = 13; ← VARIABLE!
void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

Change things

Hey, so you know what variables are, adjust the delays on the previous example.

1000 = 1000 ms == 1 second

There are several types of Variables

The most common types are:

- Int
- Float
- Double
- Bool
- Char

Int

Stands for “integer”, means any real (whole) number. Can be negative, just can’t be a fraction.

Fractions aren’t real.

ex. `int x = 12;`

Float

Floats just mean “floating point number”.

Simply put, a number with decimals.

ex. `float x = 12.11;`

Double

Just like a float, but bigger.

Big, huge decimals.

ex. `float x = 1234452.1332231;`

Don’t worry about this

The difference between a double and a float, is that:

Single precision (float) gives you 23 bits of significand, 8 bits of exponent, and 1 sign bit.

Double precision (double) gives you 52 bits of significand, 11 bits of exponent, and 1 sign bit.

Bool

“Bool” is short for “boolean”, which is a really fun way of saying “one or the other”. A boolean only has two values, true or false. Or 1 or 0, whatever you’re into.

Char

Char is short for “character”, and it is a representative of a 1 byte character of a-z, A-Z, and 0-9.

BORING

I know that was pretty boring, but let’s see what functions are and then start coding things!

Functions!

It’s getting real!

What is a function

Functions "Encapsulate" a task (they combine many instructions into a single line of code). Most programming languages provide many built in functions that would otherwise require many steps to accomplish, for example computing the square root of a number. In general, we don't care **how** a function does what it does, only that it "does it"!

Built in functions?

Setup() and Loop() are built in functions in the Arduino wrapper. They are written in C, but executed by the C compiler with Arduino.

Create your own program!

Create a Setup and Loop function with nothing in it! I promise you that it’s simple

Told you!

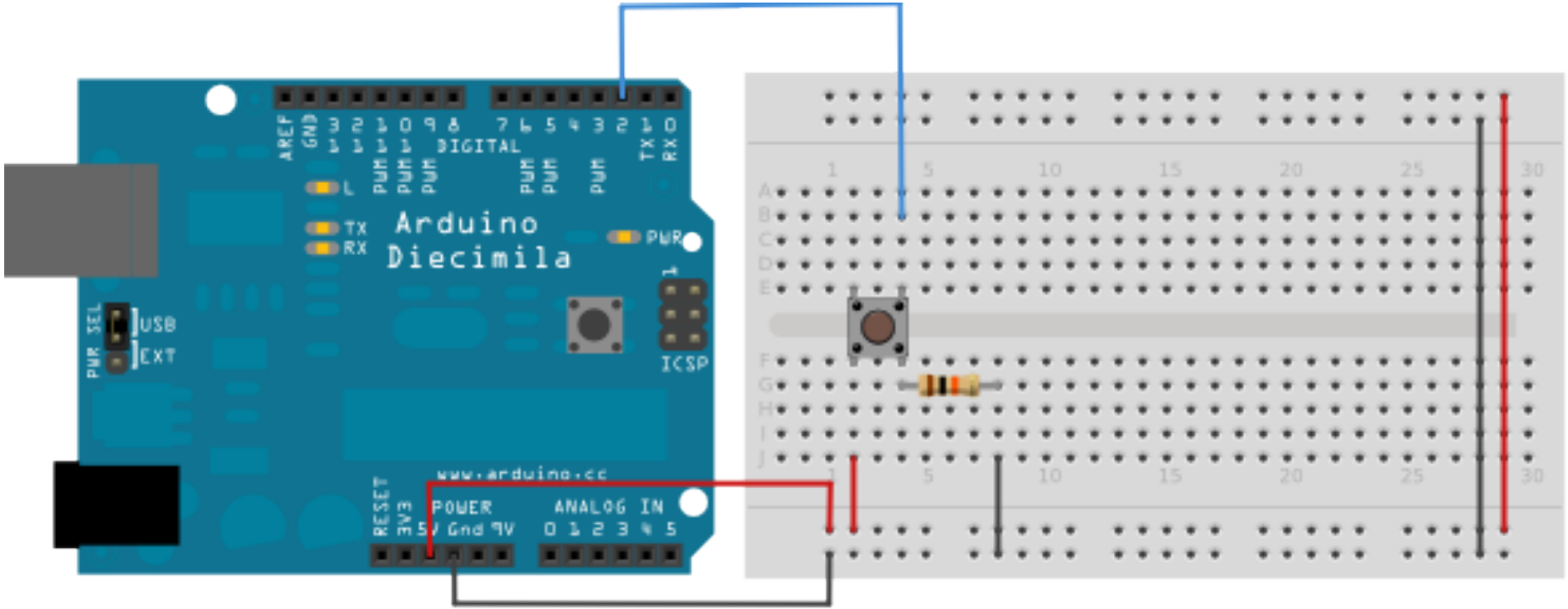
```
void setup() {
}

void loop() {
}
```

Let’s add something to that

So now we need to make something happen to our empty program...

But first, setup!



Let’s add something to that

So now we need to make something happen to our empty program…

So let’s make it blink!

```
const int buttonPin = 2;     // the number of the pushbutton pin

void setup(){

}

void loop(){

}
```

```
const int buttonPin = 2;     // the number of the pushbutton pin

void setup(){

}

void loop(){

}
```

This initializes our button variable, setting the pin on which we're outputting the button from.

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

void setup(){

}

void loop(){

}
```

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

void setup(){

}

void loop(){

}
```

This initializes our pin variable, setting the pin on the board which we're trying to output the LED from.

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin
// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup(){

}

void loop(){

}
```

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin
// variables will change:
boolean buttonState = 0;         // variable for reading the pushbutton status

void setup(){

}

void loop(){

}
```

Here we've set the state of our button to 0, which is "off".

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin
// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup(){
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
}
void loop(){

}
```

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin
// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup(){
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);

}
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
}
```

```
void setup(){
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

Fin