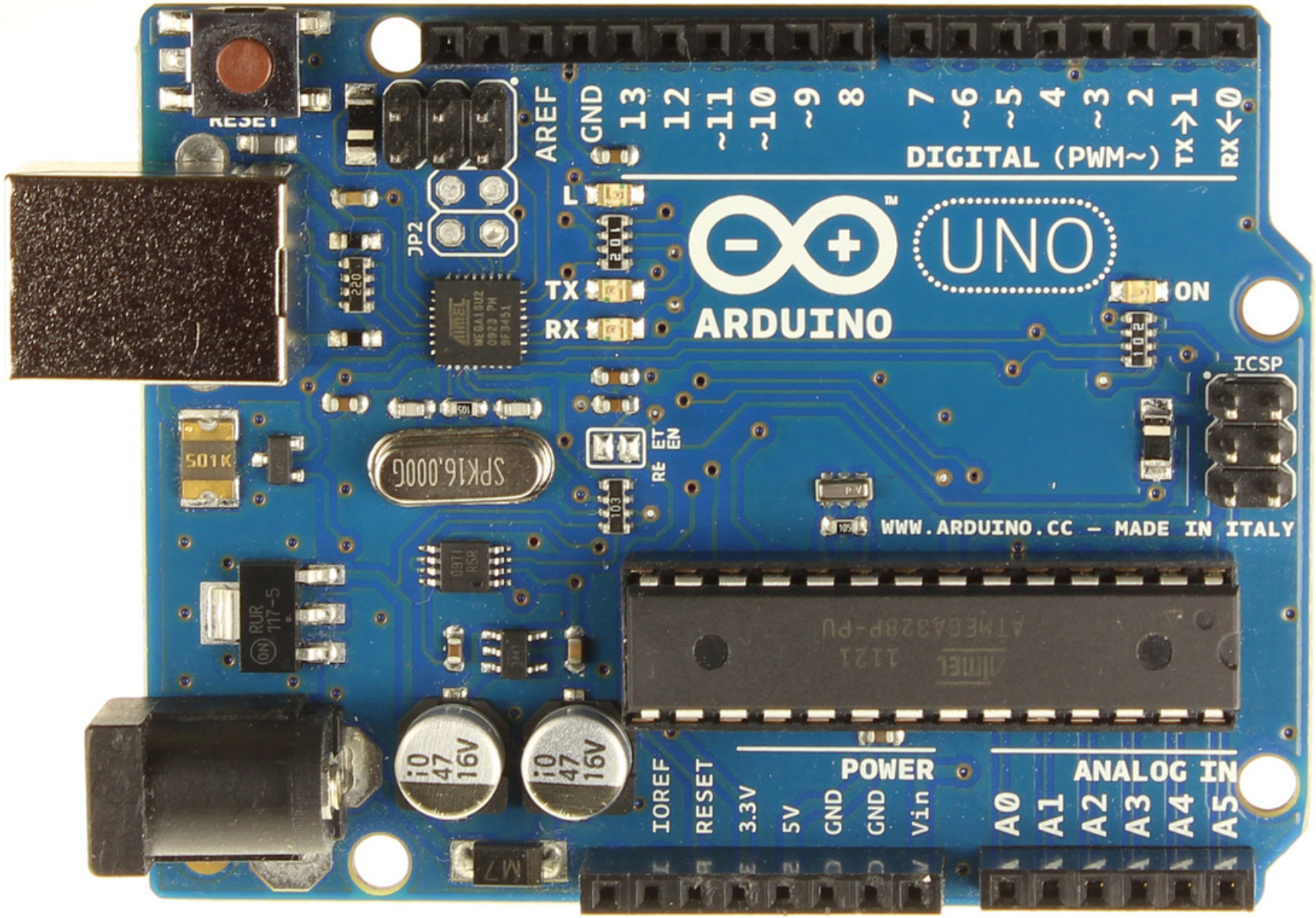
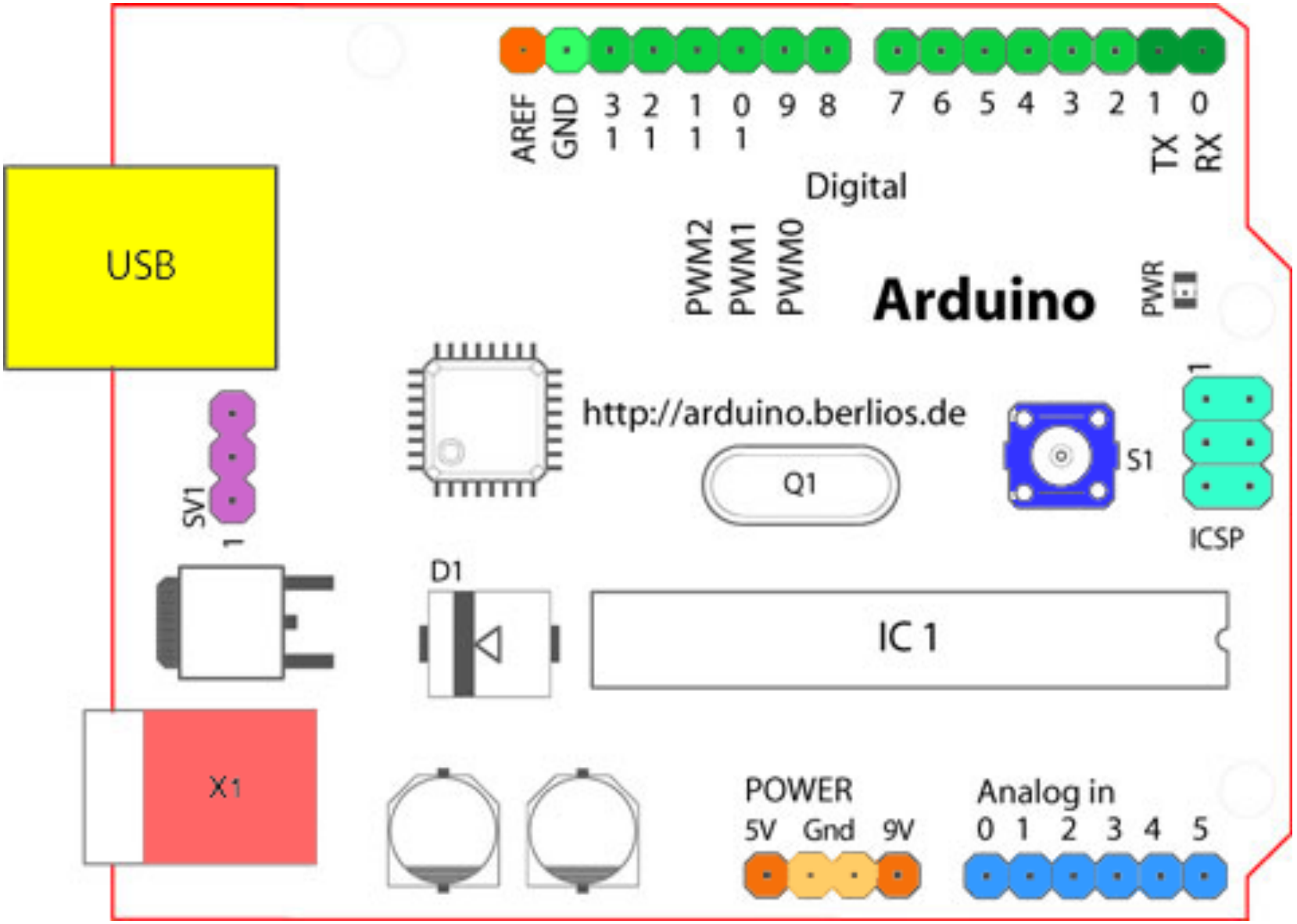
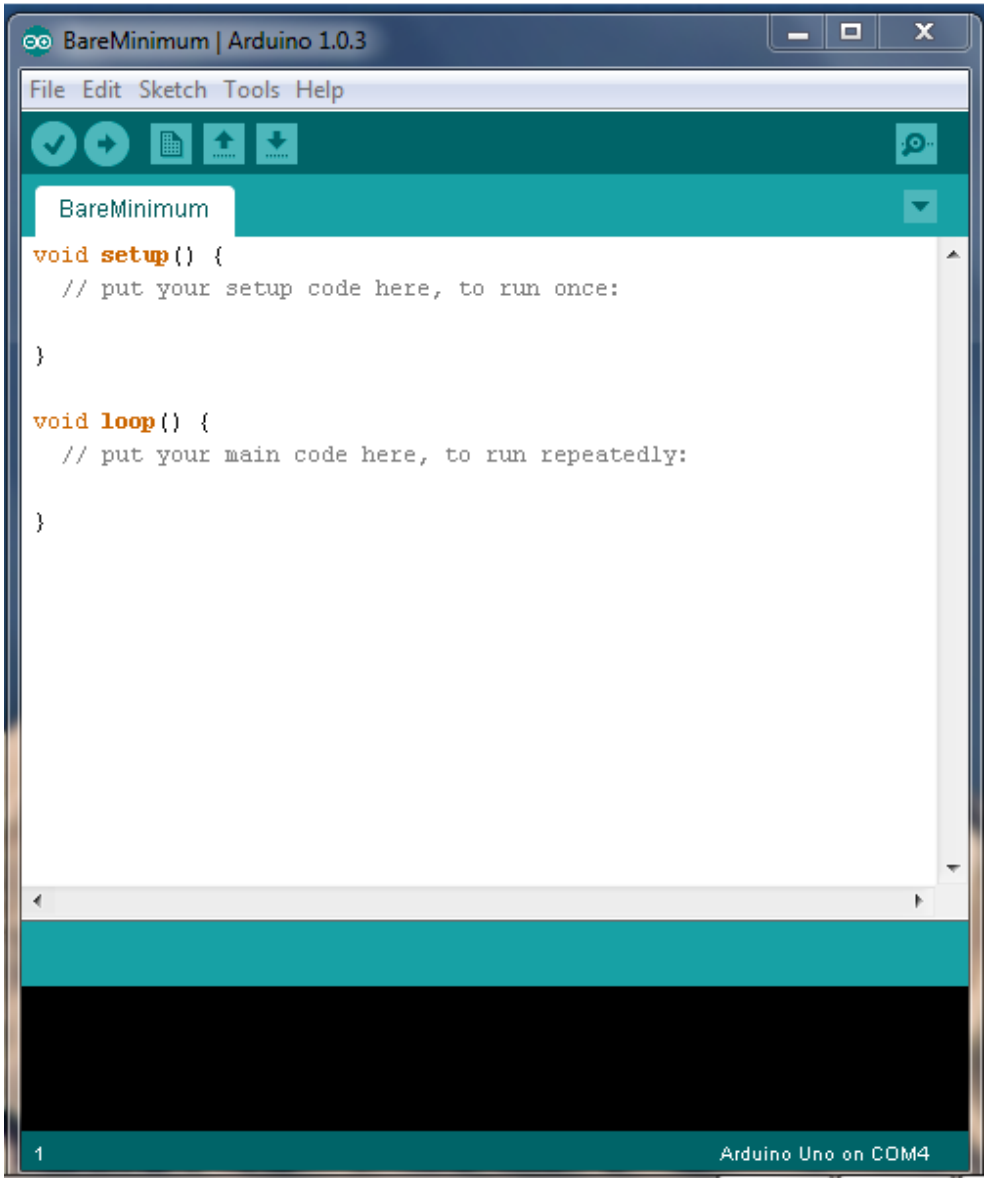

RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
JP2
TX
RX
ARDUINO
UNO
ON
ICSP
WWW.ARDUINO.CC – MADE IN ITALY
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5


AREF
GND
3
1
2
1
1
1
0
1
9
8
7
6
5
4
3
2
1
TX
0
RX
Digital
PWM2
PWM1
PWM0
Arduino
PWR
USB
http://arduino.berlios.de
Q1
S1
1
ICSP
SV1
1
D1
IC 1
X1
POWER
5V
Gnd
9V
Analog in
0
1
2
3
4
5

# Arduino IDE

- integrated development environment



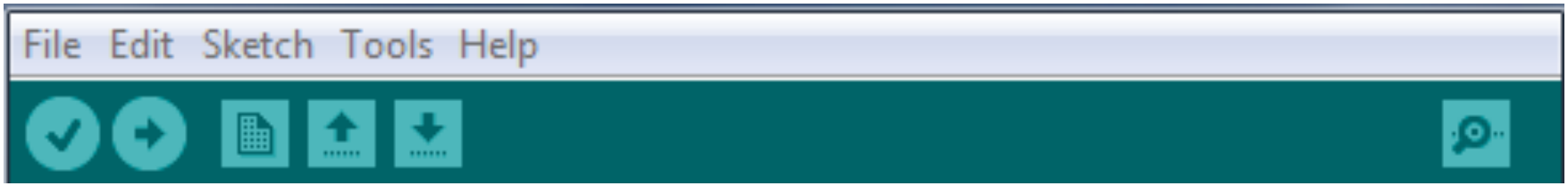
# Get going

- In the IDE goto:

  Tools > Board > Arduino/Genuino Uno

  Tools > Port > COM[!] Arduino/Genuino Uno

# Hello World




File > Examples > 0.1 Basics > Blink


Blink §
void setup() {
// initialize digital pin LED_BUILTIN as an output.
pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
1
Arduino/Genuino Uno on COM1

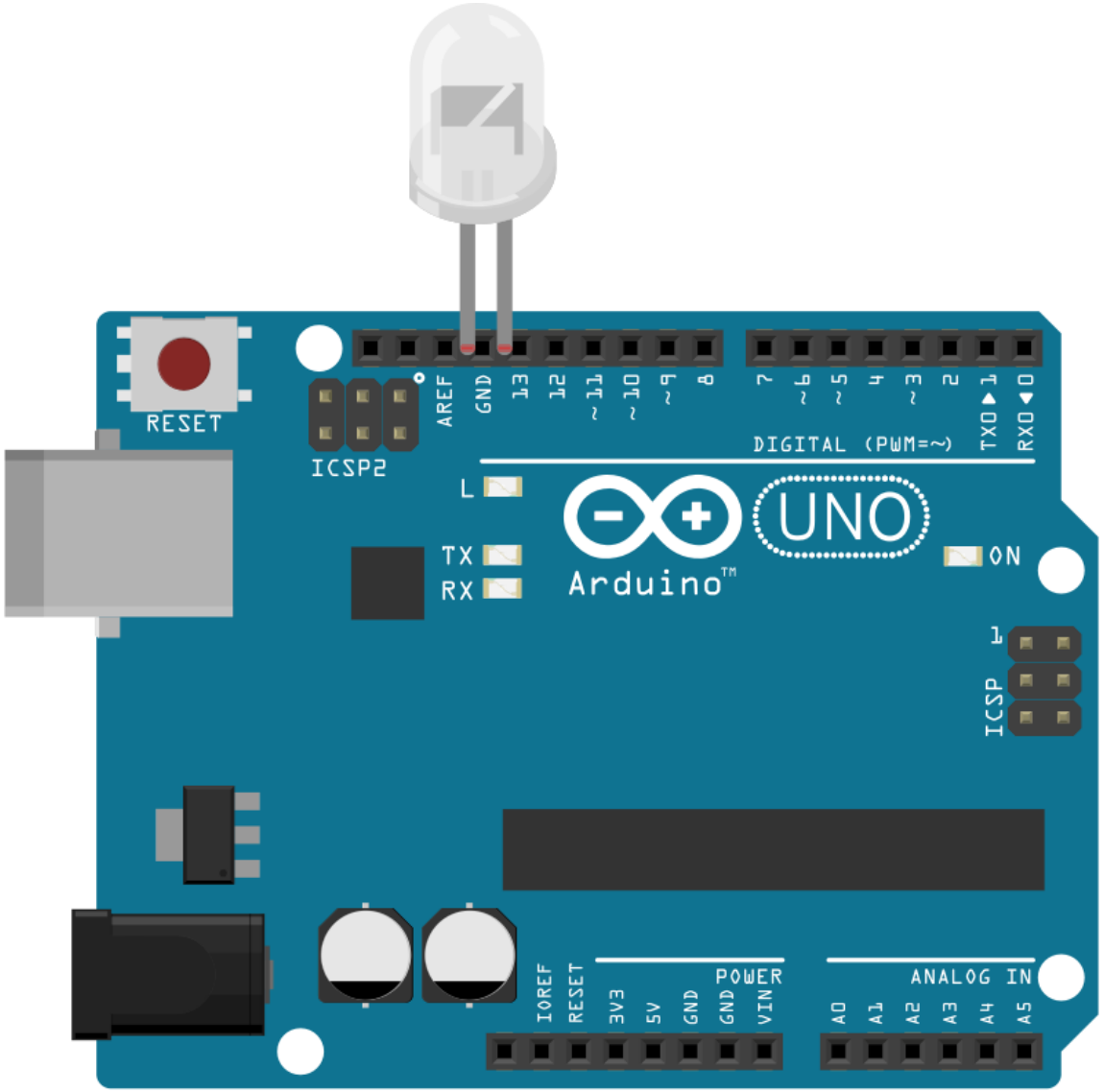

(anode)
+
(cathode)
GND

```
int ledPin = 13;    // LED connected to digital pin 13

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(ledPin, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                  // wait for a second
}
```

# Variables

- A variable is a way of naming and storing a value for use by the program
- E.g. data from a sensor or value that will be transformed by a mathematical calculation.

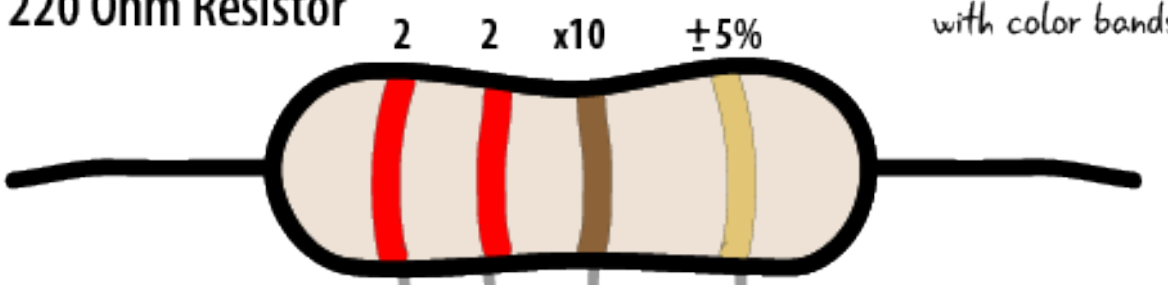
```
int ledPin = 13;
digitalWrite(ledPin, HIGH);  -> digitalWrite(13, HIGH);
```


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
Arduino™
UNO
ON
ICSP
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN

**220 Ohm Resistor**

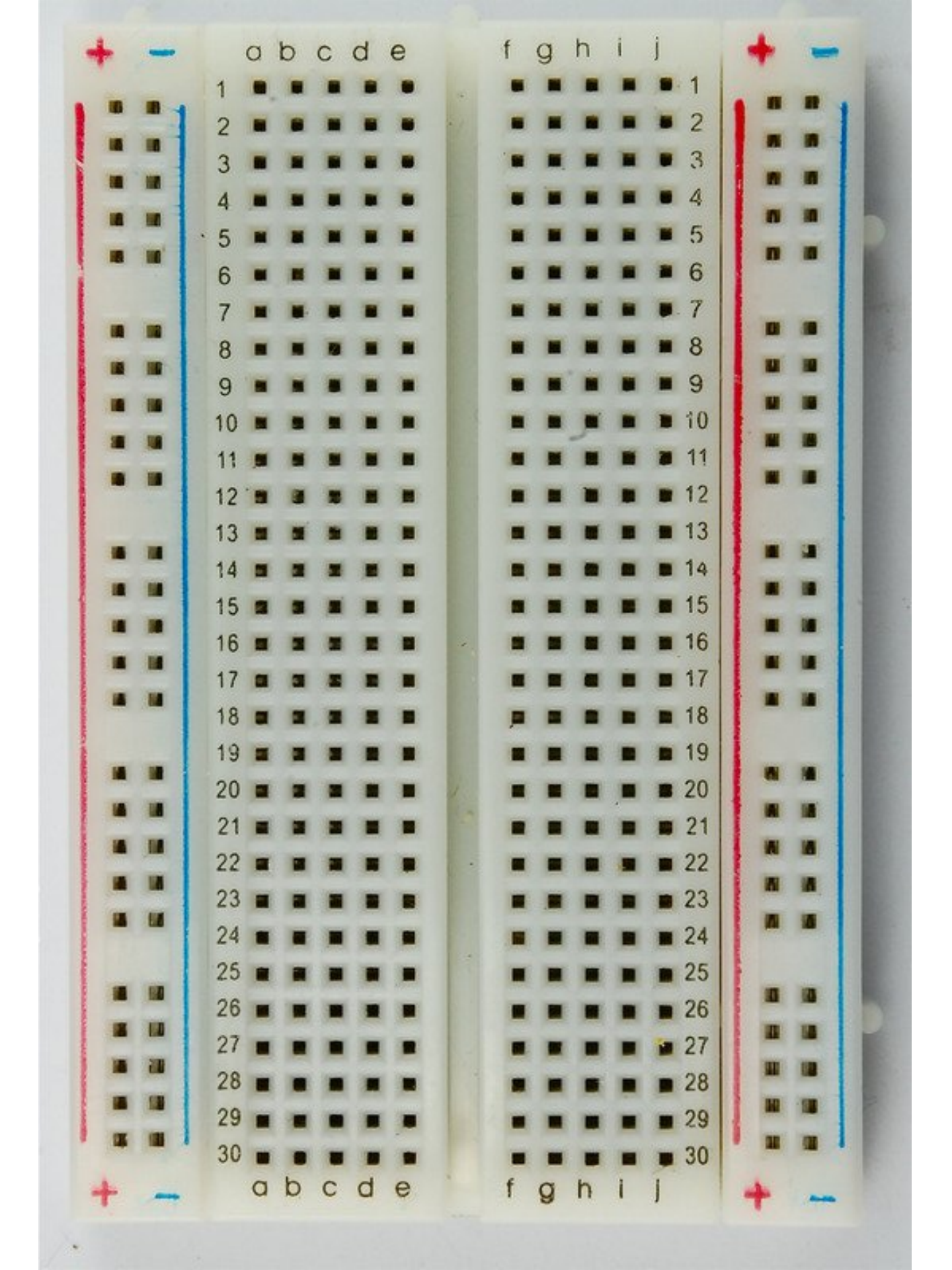
# 220 Ohm Resistor

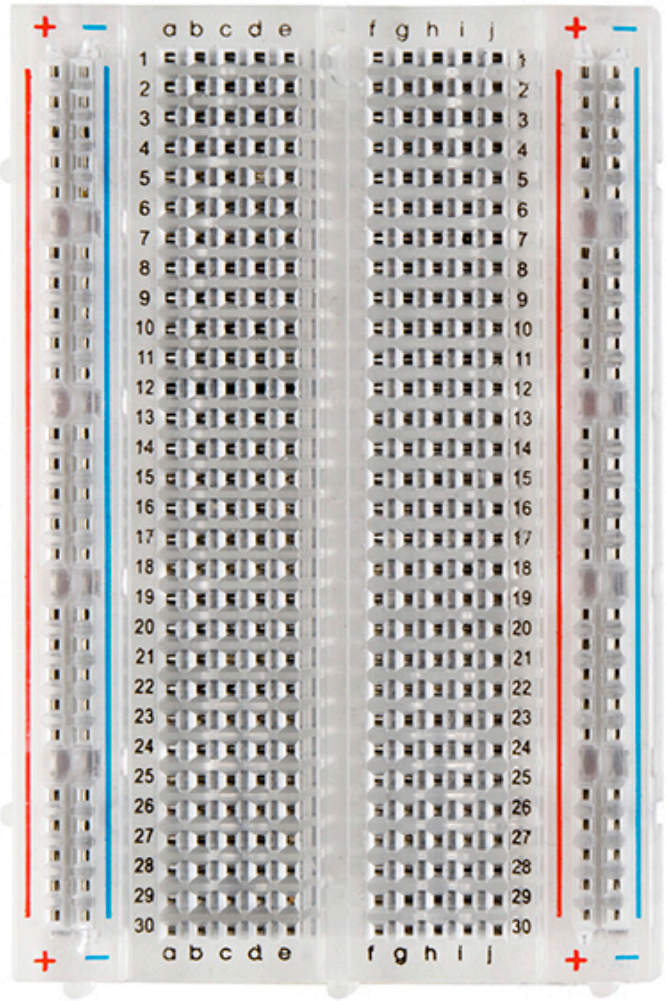
sample resistor with color bands

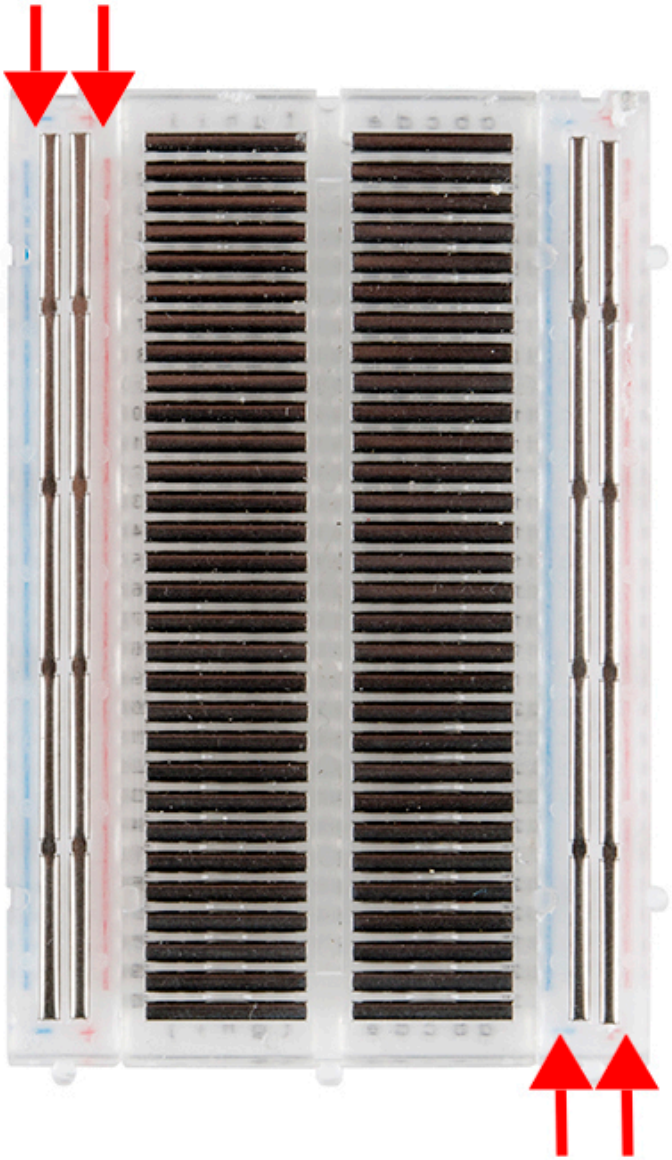


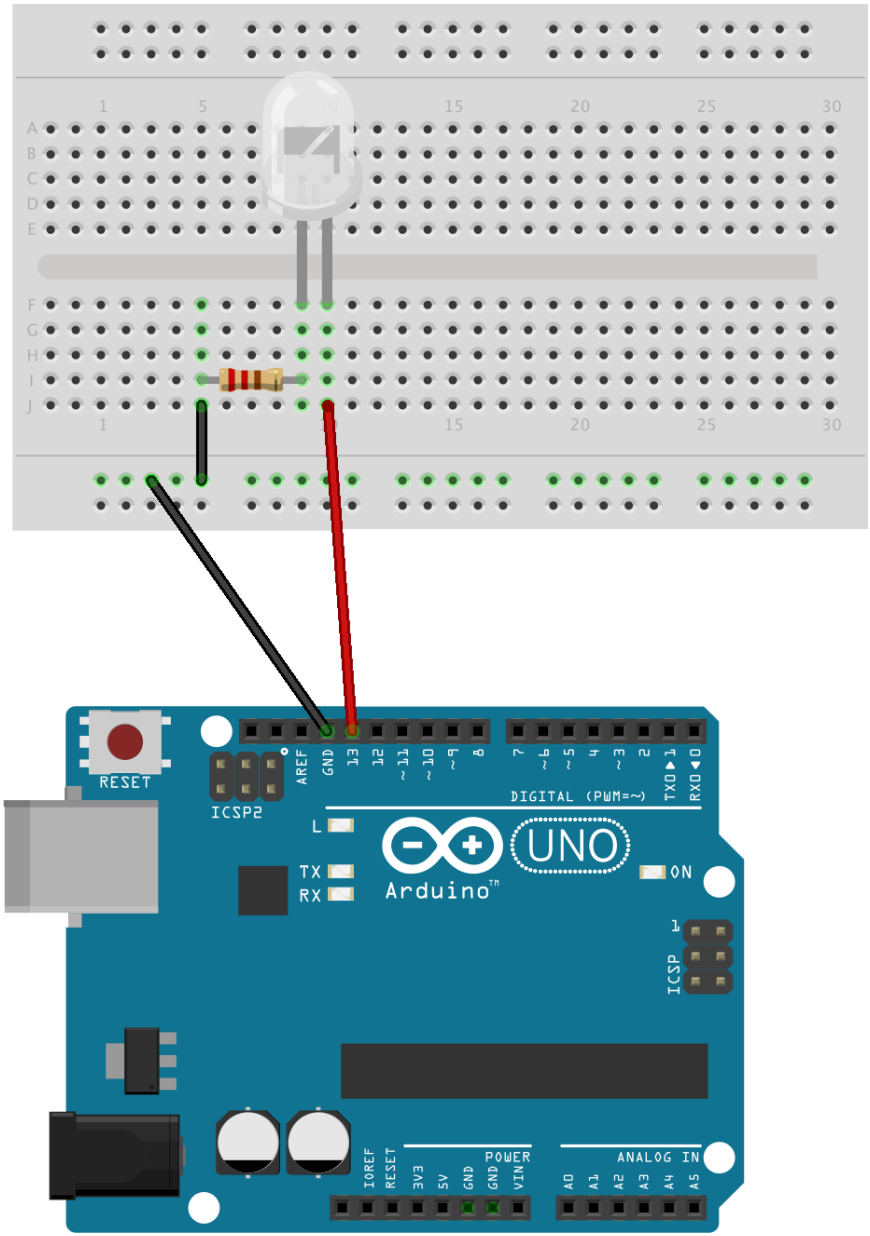

what the color bands represent on the resistor

| | 1st Digit | 2nd Digit | Multiplier | Tolerance |
|---|---|---|---|---|
| Black | 0 | 0 | 1 | 5% Gold |
| Brown | 1 | 1 | 10 | 10% Silver |
| Red | 2 | 2 | 100 | |
| Orange | 3 | 3 | 1,000 | |
| Yellow | 4 | 4 | 10,000 | |
| Green | 5 | 5 | 100,000 | |
| Blue | 6 | 6 | 1,000,000 | |
| Purple | 7 | 7 | | |
| Gray | 8 | 8 | | |
| White | 9 | 9 | | |


a b c d e
f g h i j
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
a b c d e
f g h i j

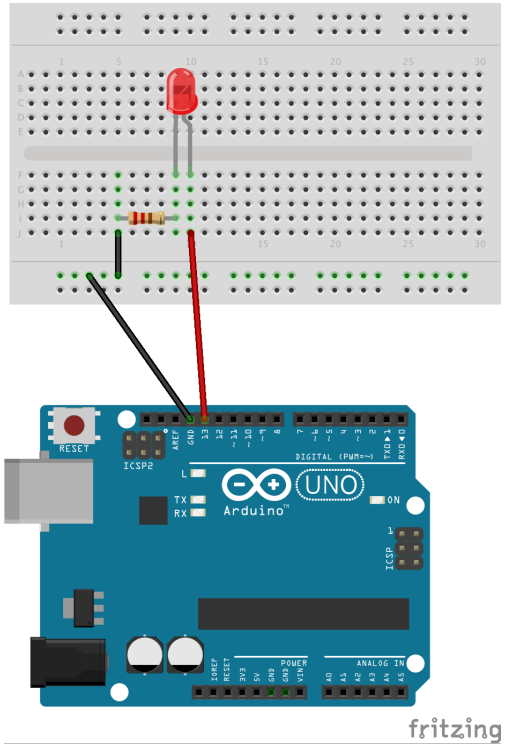

a b c d e
f g h i j
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30




1
5
10
15
20
25
30
A
B
C
D
E
F
G
H
I
J
RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino™
ON
ICSP
POWER
ANALOG IN
IOREF
RESET
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5

# Task

- Test the circuit
- Can you alter the blink time?
- Try different rates

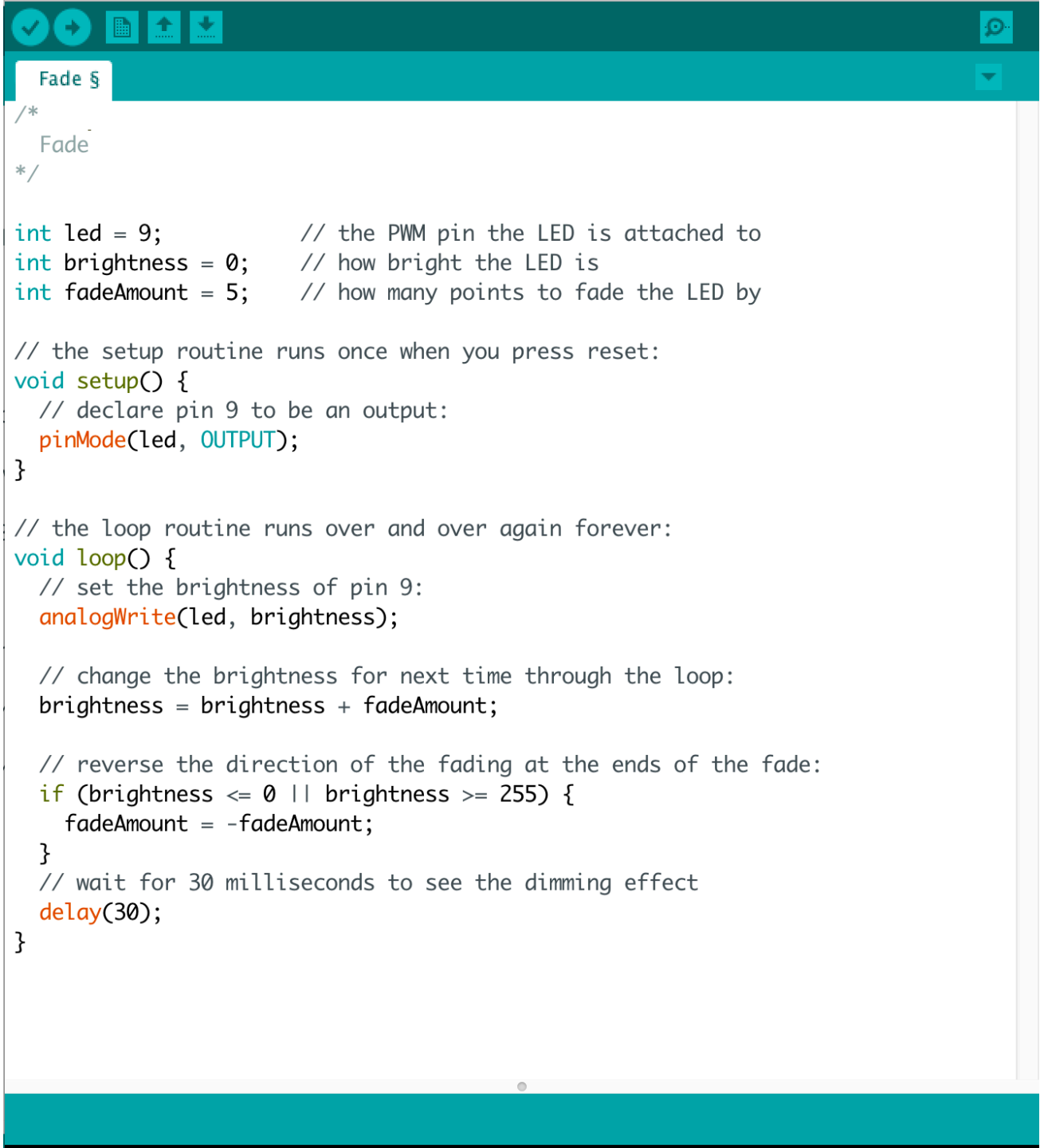


# Task

- Re-write the code to use pin 9 instead of pin13
- Re-wire the circuit to use pin 9 instead of pin13

# Fade

- From the Arduino IED open:
- File >  Example > 01.Basics > Fade
- Upload the Code
- What does it do?


Fade §
/*
Fade
*/
int led = 9; // the PWM pin the LED is attached to
int brightness = 0; // how bright the LED is
int fadeAmount = 5; // how many points to fade the LED by
// the setup routine runs once when you press reset:
void setup() {
// declare pin 9 to be an output:
pinMode(led, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
// set the brightness of pin 9:
analogWrite(led, brightness);
// change the brightness for next time through the loop:
brightness = brightness + fadeAmount;
// reverse the direction of the fading at the ends of the fade:
if (brightness <= 0 || brightness >= 255) {
fadeAmount = -fadeAmount;
}
// wait for 30 milliseconds to see the dimming effect
delay(30);
}

# analogWrite() v digitalWrite()

digitalWrite()

- Two states High or Low
- High - > 5v
- Low - > 0v

analogWrite()

- Set the intensity between 0 and 255

# if

- The if() statement allows you to make something happen or not, depending on whether a given condition is true or not. Eg:

```
if (someCondition) {
    // do stuff if the condition is true
}
```

# if-else

```
if (someCondition) {
    // do stuff if the condition is true
} else {
    // do stuff if the condition is false
}
```

# else-if

```
if (someCondition) {
    // do stuff if the condition is true
} else if (anotherCondition) {
    // do stuff only if the first condition is false
     // and the second condition is true
}
```

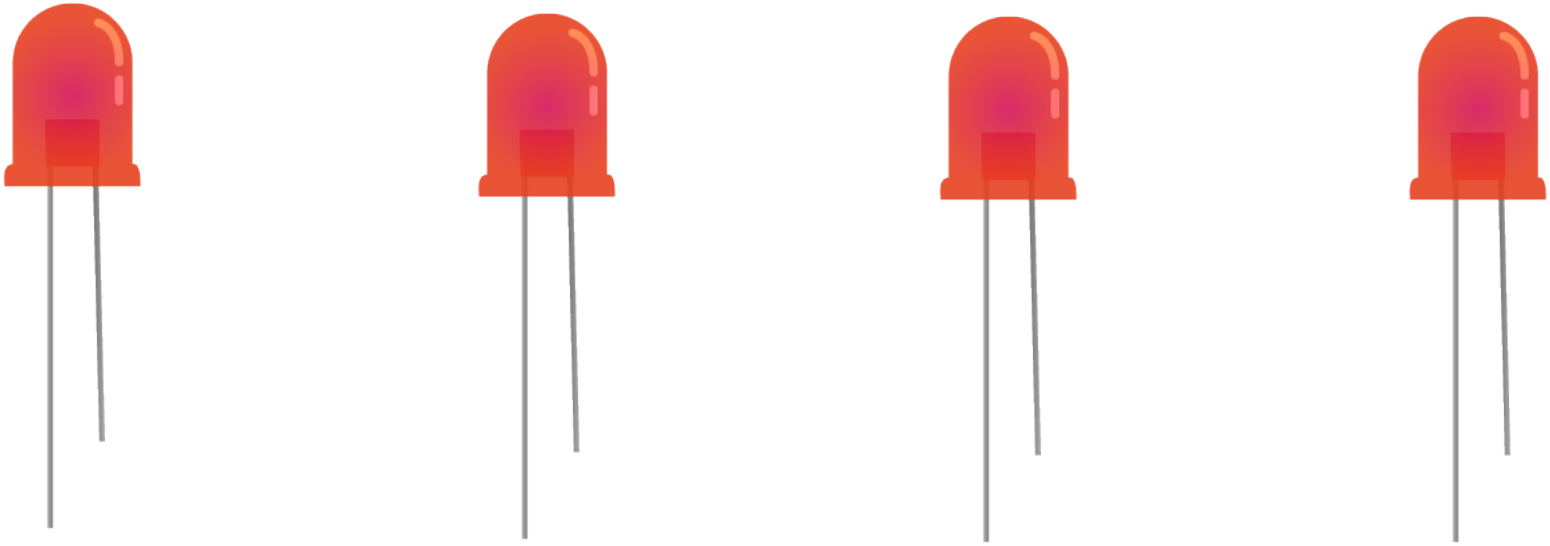
# if

```
if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
}
```

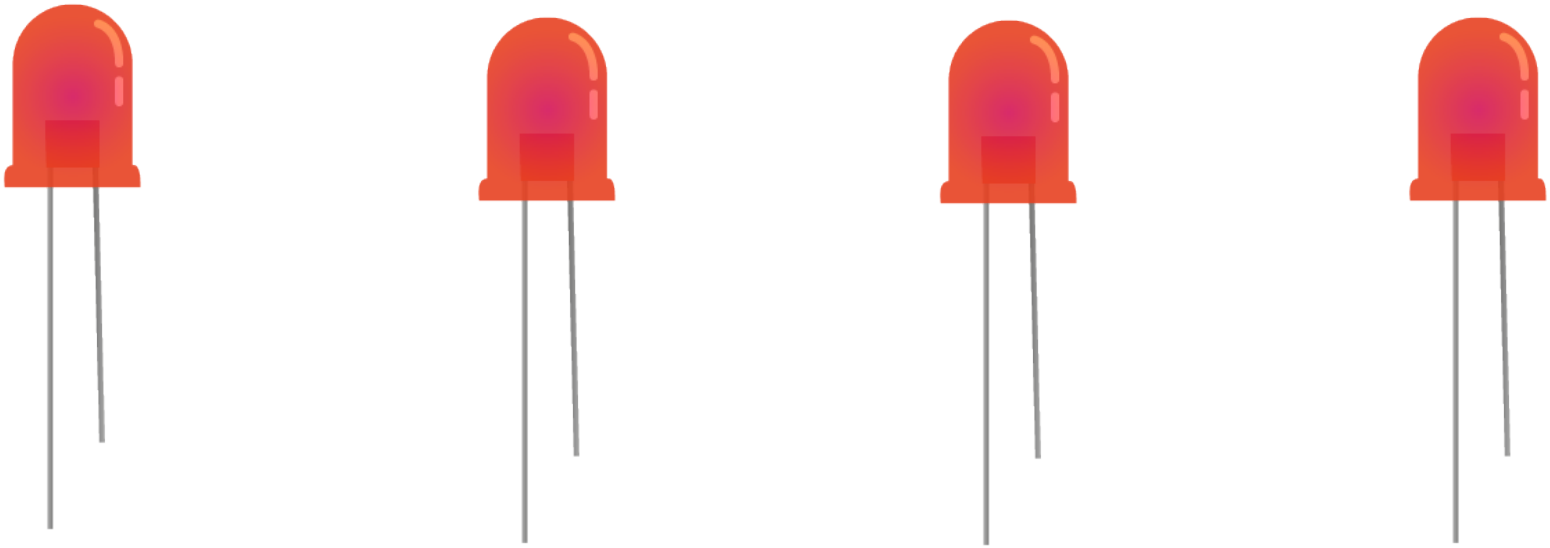
*Remember a – and a – make a +

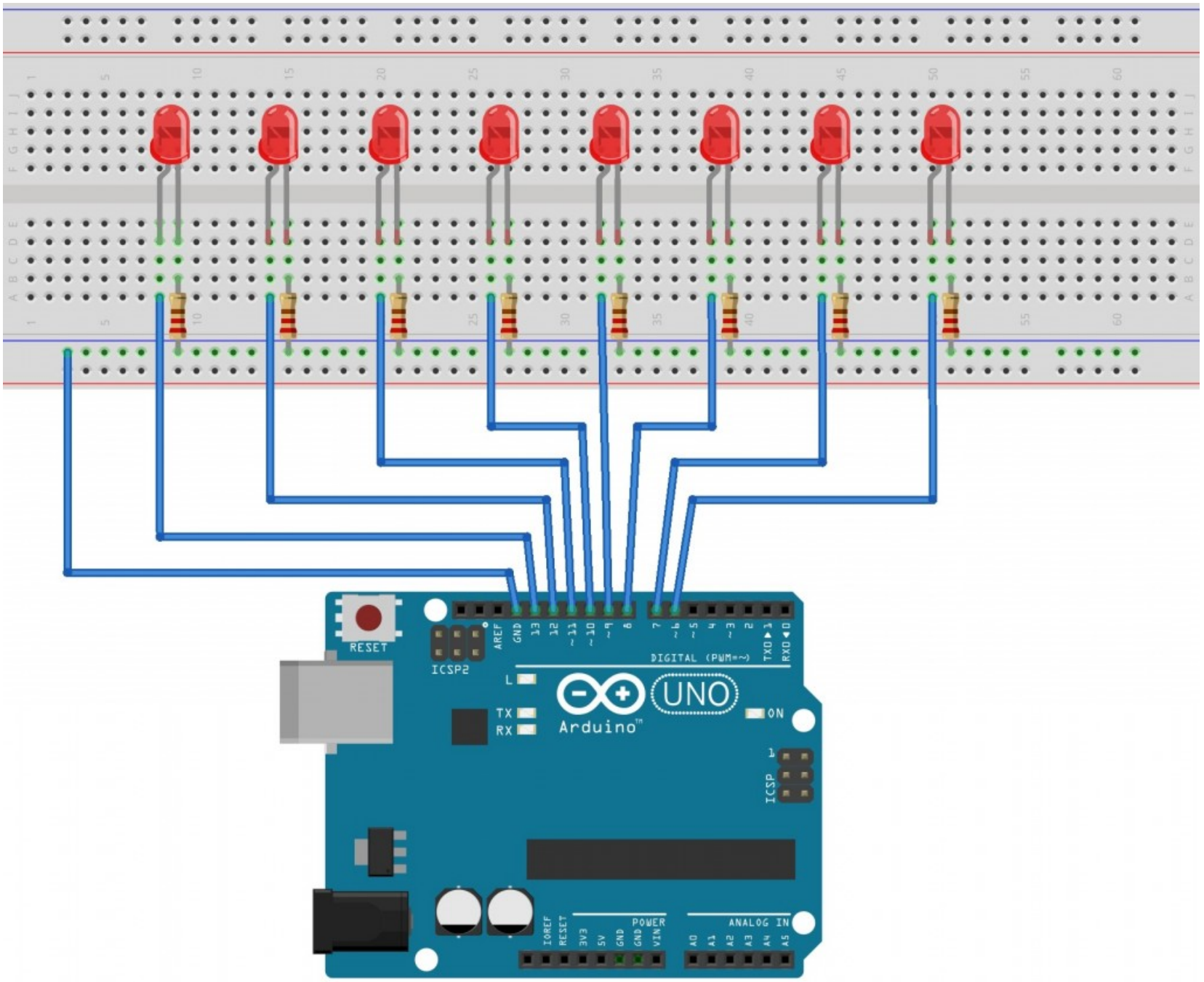
# 2 LED’s

- Wire the circuit to use 2 LED’s on pin 9 and pin 10
- Re-write the code to use pin 9 and pin 10



# 8 LED’s !!!




RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TXD▲1
RXD◀0
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino™
ON
ICSP
POWER
ANALOG IN
IOREF
RESET
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5

# The Code

- Click on the 8 LED code on RCADE
- Copy and past into the Arduino IDE
- Verify
- Upload to the Arduino

# Code Problem

- We have coded each pin individually
- But this gets unwieldy

Solution:

- Arrays
- Loops

# Arrays

- arrays[] – are used to make managing variables easier
- An array is a collection of variables that are accessed with an index number.

E.g.

int myPins[] = {2, 4, 8, 3, 6};

myPins[0] will contain 2

myPins[1] will contain 4

myPins[2] will contain 8

myPins[3] will contain 3

myPins[4] will contain 6

- Replace all this code: -

```
int ledPin6 = 6;
int ledPin7 = 7;
int ledPin8 = 8;
int ledPin9 = 9;
int ledPin10 = 10;
int ledPin11 = 11;
int ledPin12 = 12;
int ledPin13 = 13;
```

- with: -

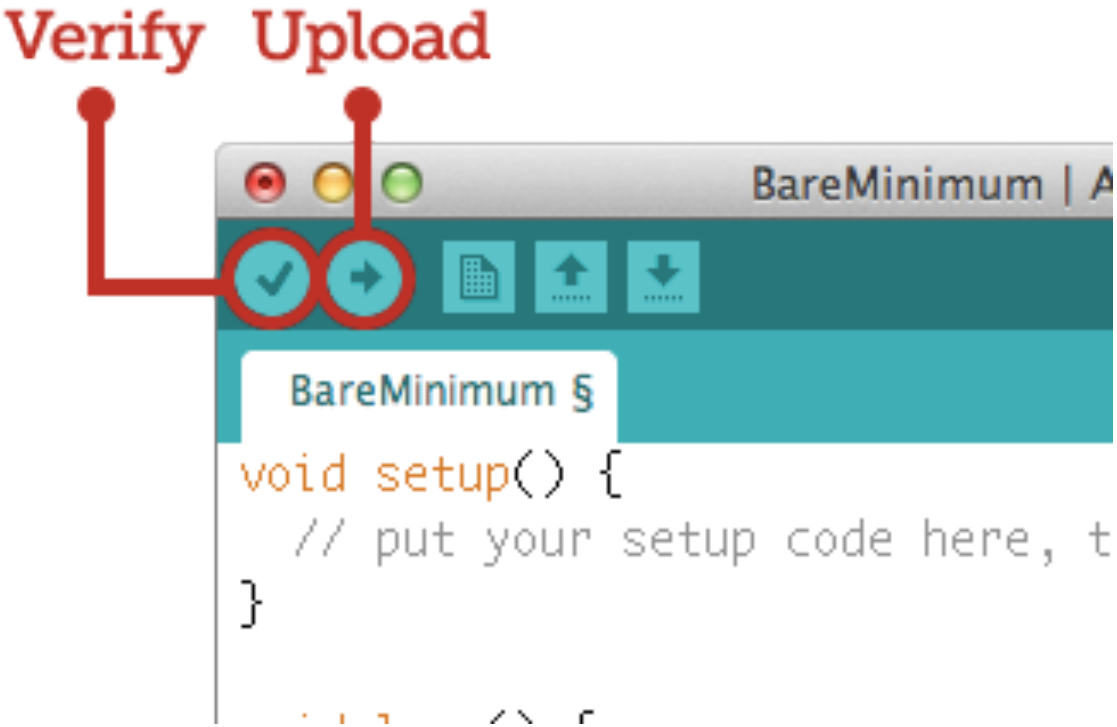
```
int ledPins[] = {6,7,8,9,10,11,12,13};
```

# for loop

- The **for** statement is used to repeat a block of statements enclosed in curly braces.
- An increment counter is usually used to increment and terminate the loop.
- The **for** statement is useful for any repetitive operation, and is often used in combination with arrays to operate on collections of data/pins.

```
void setup(){

   for (int i = 0; i < 8; i++){
   pinMode(ledPins[i],OUTPUT);
   }
}
```

```
void loop()
 {
    //Turns Each LED on one after another
    for (int i = 0; i <= 8; i++){
    digitalWrite(ledPins[i], HIGH);
    delay(100);
    }
    //Turn Each LED off one after another
    for (int i = 8; i >= 0; i--){
    digitalWrite(ledPins[i], LOW);
    delay(100);
    }
}
```

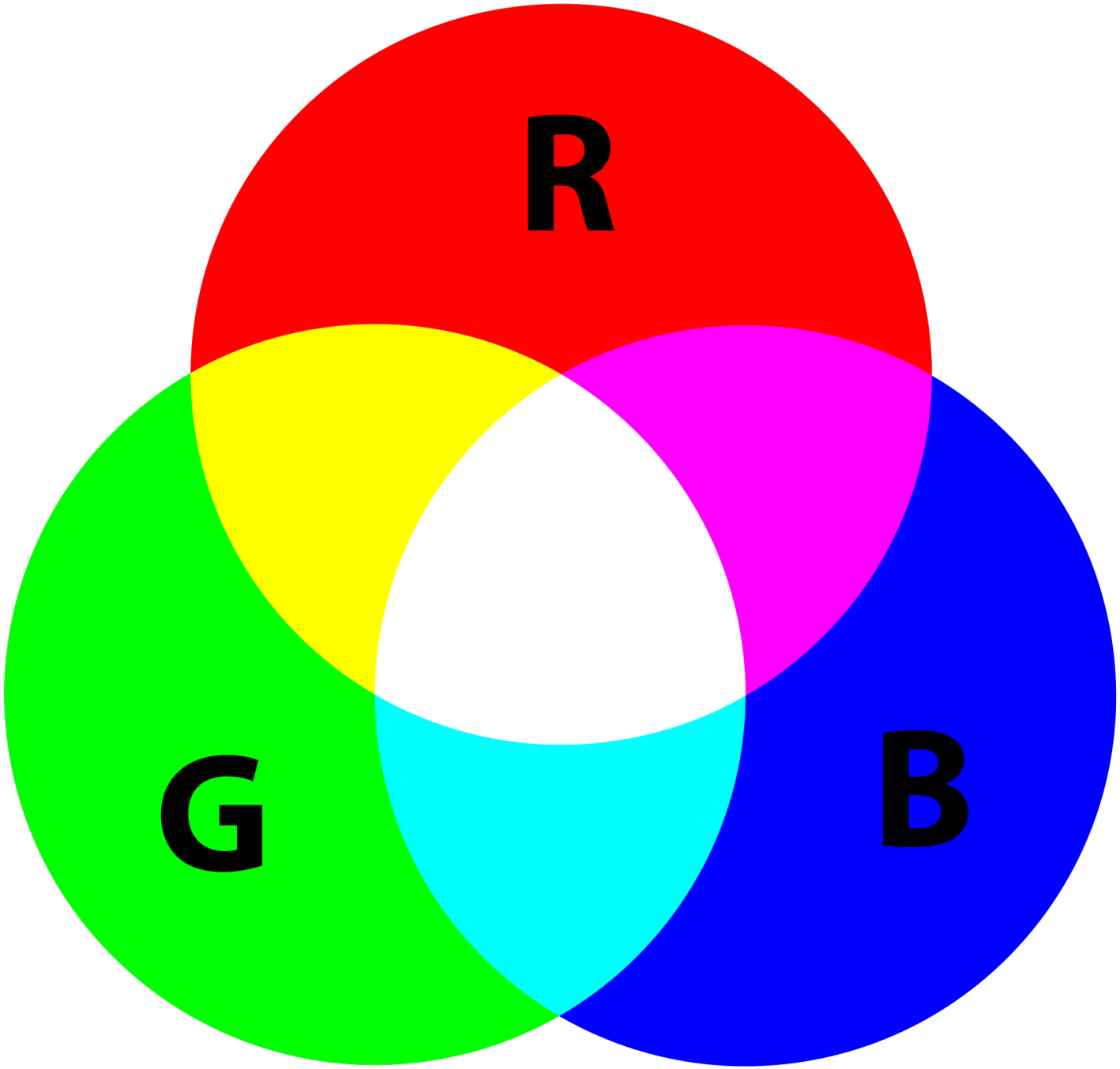


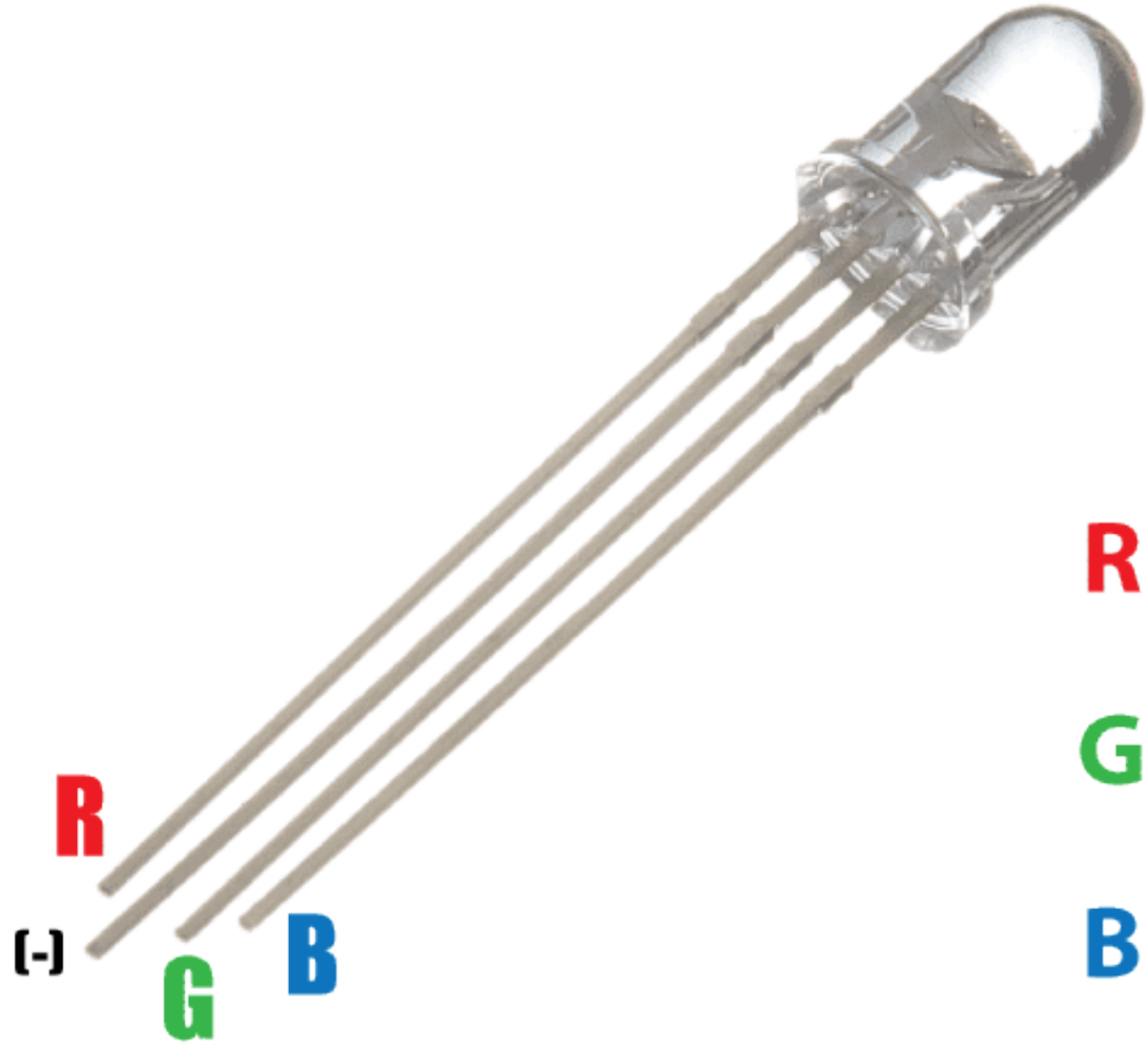

- Verify
- Find your bugs !
- Upload to the Arduino

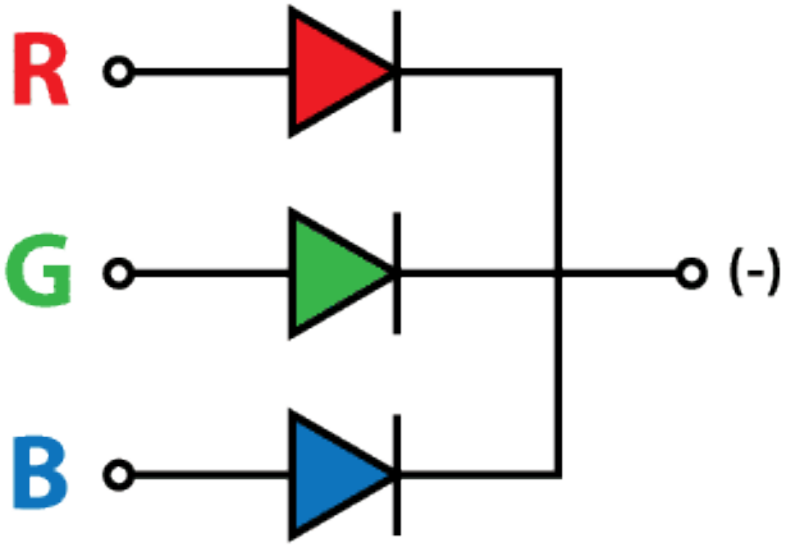
# Task

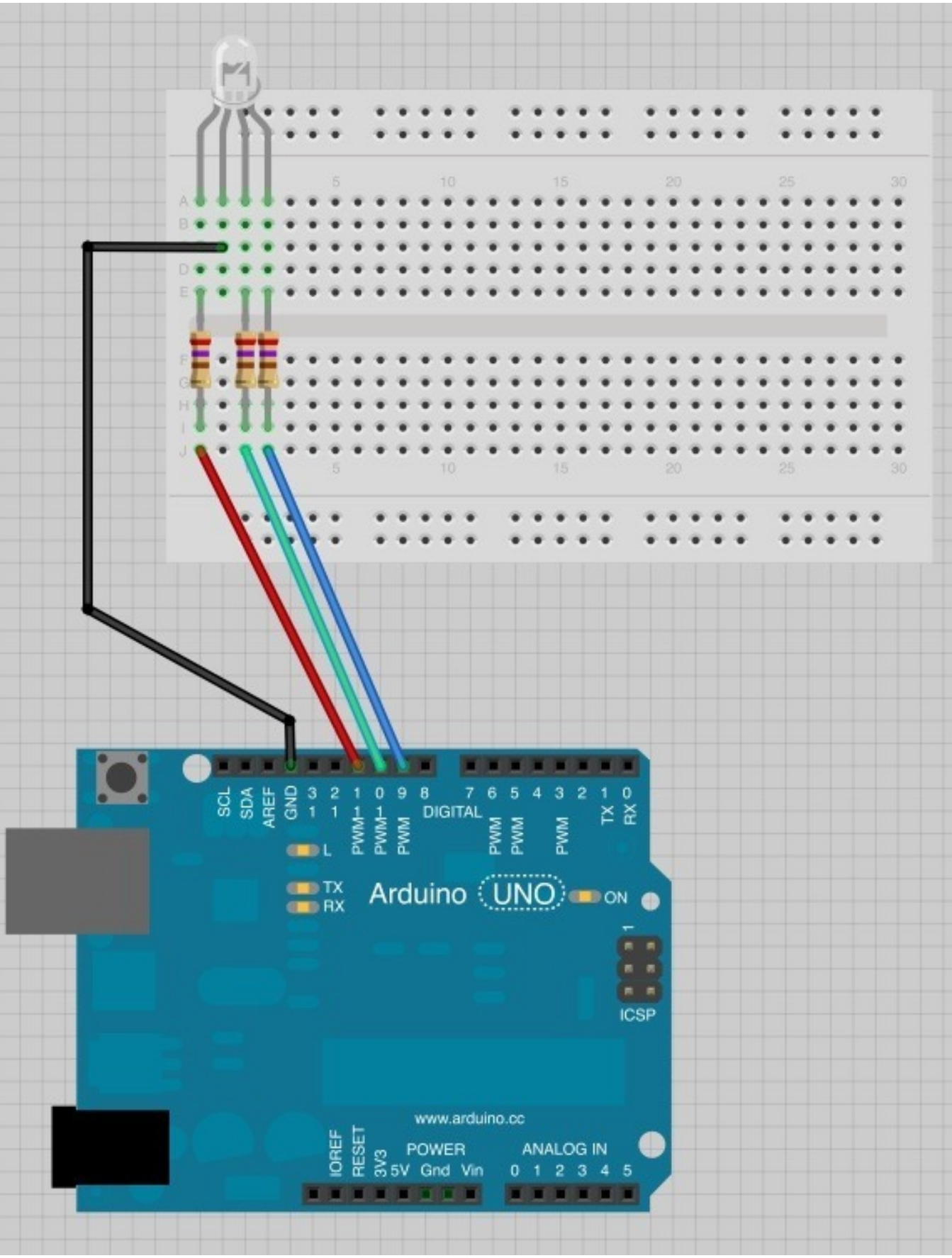
- Can you change the code to make a single LED travel forward then back?
  - Hint remember the fade code!

- Can you think of and code one other LED animation?


R
G
B


R
[-]
G
B

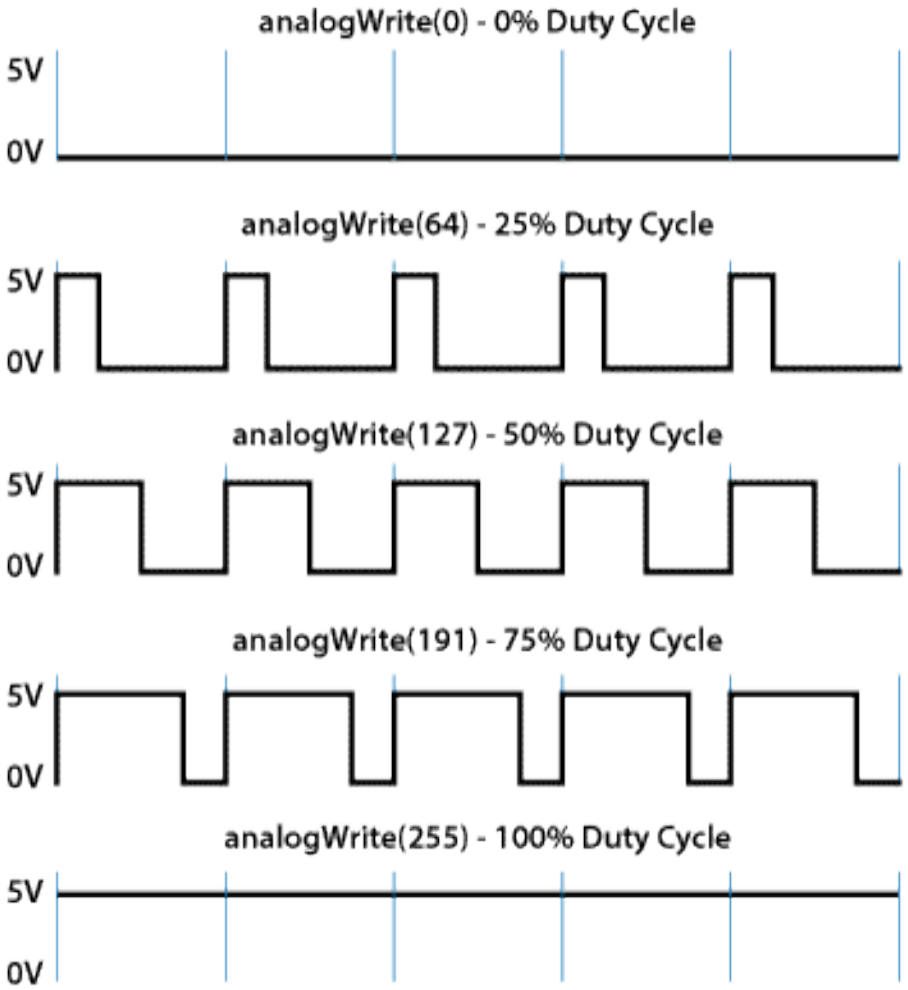

R
G
B
(-)

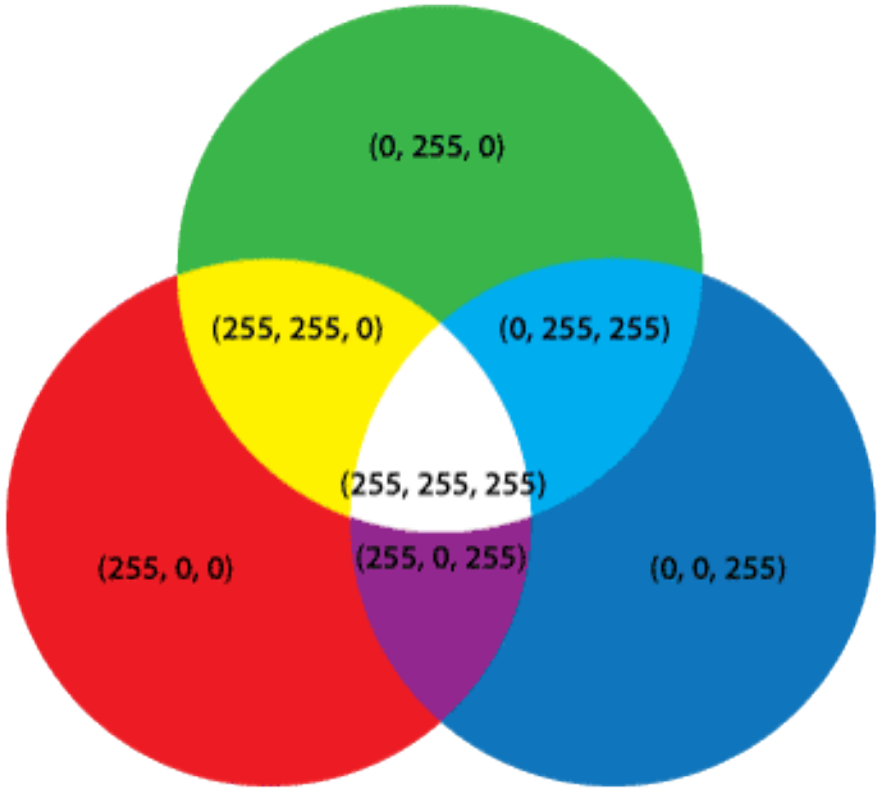

SCL
SDA
AREF
GND
DIGITAL
PWM
TX
RX
L
TX
RX
Arduino UNO
ON
ICSP
www.arduino.cc
IOREF
RESET
3V3
POWER
5V
Gnd
Vin
ANALOG IN
0 1 2 3 4 5

# The Code

- Download the RGB LED code from RCADE
- Upload to the Arduino

## PWM - Pulse Width Modulation



# Functions

- Segmenting code into functions allows a programmer to create modular pieces of code that perform a defined task and then return to the area of code from which the function was "called".
- The typical case for creating a function is when one needs to perform the same action multiple times in a program.

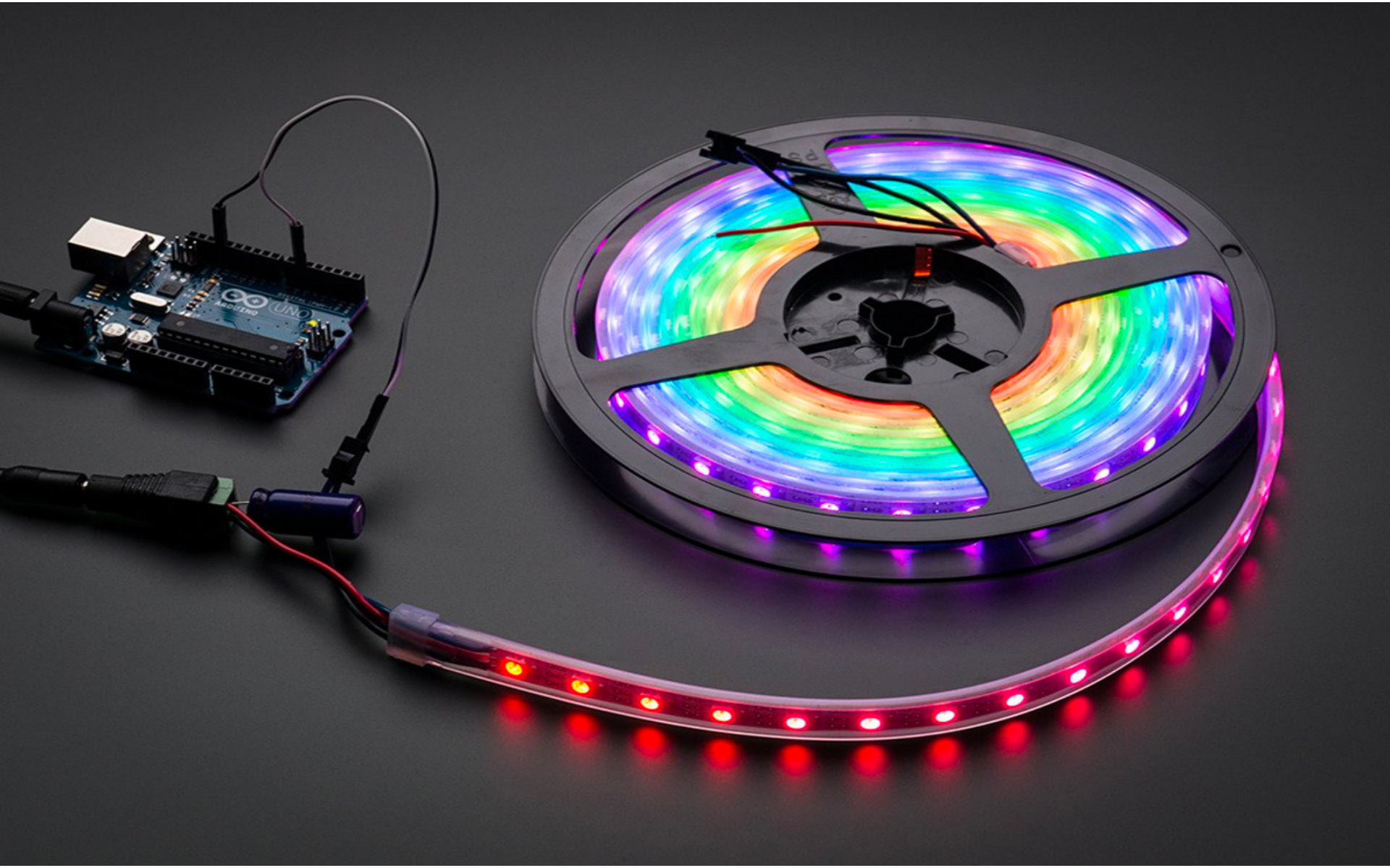
# Function example

```
void setColor(int red, int green, int blue){
    analogWrite(redPin, red);
    analogWrite(greenPin, green);
    analogWrite(bluePin, blue);
}
```

# Task

- Hack the RGB LED code to smoothly cycle through the colour spectrum

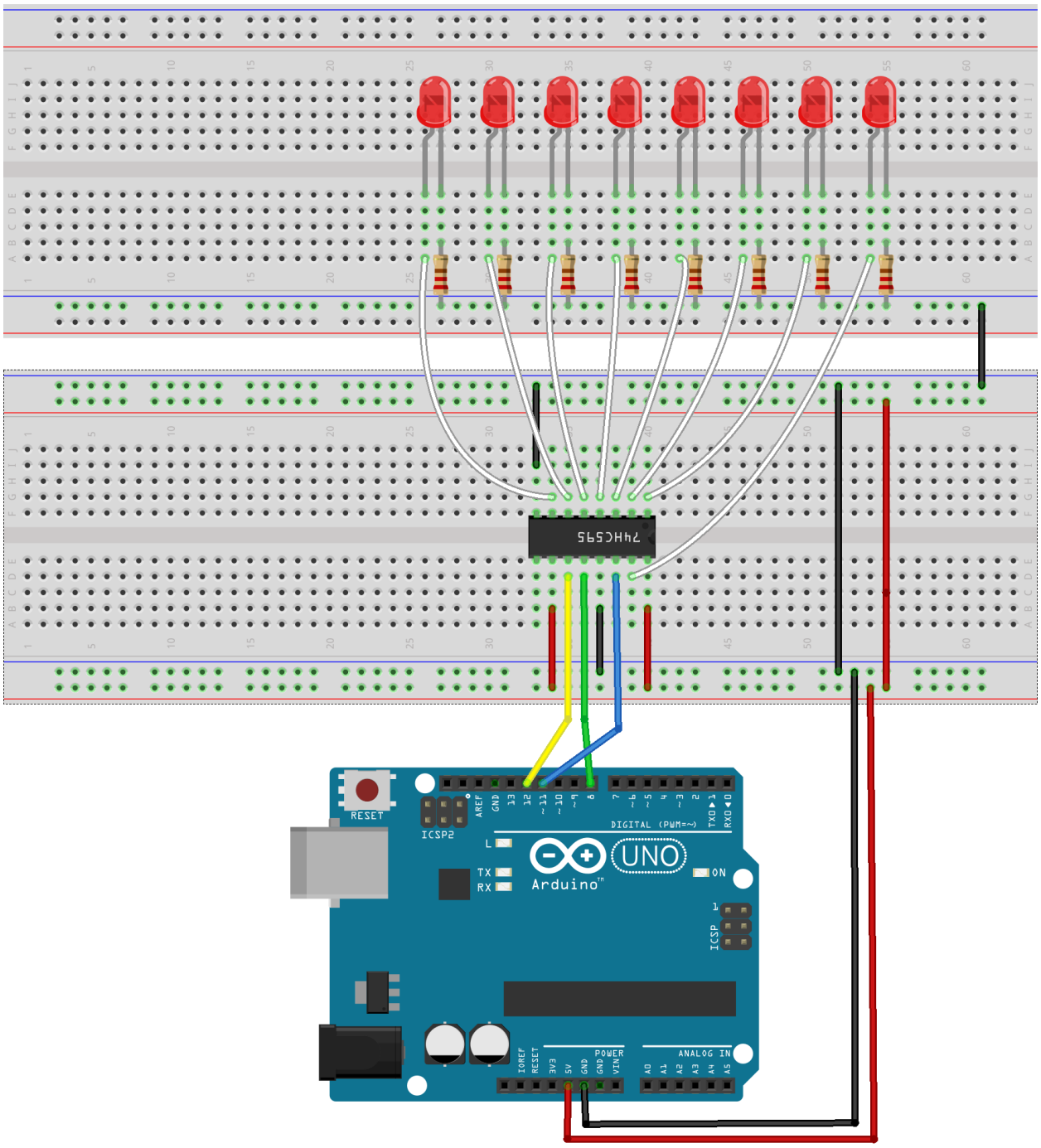
# Using lots of LEDS



https://www.youtube.com/watch?v=HO6xQMR8naw

# Doing It Yourself

- To control more LEDs than the Arduino has pins you need to use a chip.
- The best chip for the job is a shift register
- We will use the 74HC595
- The 74HC595 shift register allows us to control 8 LED’s with 3 Arduino pins.
- You can add a second shift register, doubling the number of output pins you have while still using the same number of pins from the Arduino


74HC595
RESET
ICSP2
DIGITAL (PWM=~)
UNO
Arduino™
TX
RX
ON
ICSP
POWER
ANALOG IN
fritzing

# The code

- Download the Shift register hello World from RCADE

```
//Pin connected to ST_CP of 74HC595
int latchPin = 8;
//Pin connected to SH_CP of 74HC595
int clockPin = 12;
////Pin connected to DS of 74HC595
int dataPin = 11;



void setup() {
  //set pins to output so you can control the shift register
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}

void loop() {
  // count from 0 to 255 and display the number
  // on the LEDs
  for (int numberToDisplay = 0; numberToDisplay < 256; numberToDisplay++) {
    // take the latchPin low so
    // the LEDs don't change while you're sending in bits:
    digitalWrite(latchPin, LOW);
    // shift out the bits:
    shiftOut(dataPin, clockPin, MSBFIRST, numberToDisplay);

    //take the latch pin high so the LEDs will light up:
    digitalWrite(latchPin, HIGH);
    // pause before next value:
    delay(500);
  }
}
```

# Useful References

- Arduino Language Reference https://www.arduino.cc/en/Reference/HomePage