

ELECTRONICS 1

ELECTRONICS FOR INTERACTIVE MEDIA DESIGN
LES 4

EMMA PARESCHI

FROM THE LAST TIME

From class 2:

- How to count a switch
- For statement
- Array structure
- Generate sound: Buzzer
- Light sensor

Assignments:

- 1 - The Dice
- 2 - The buzzer

Today:

- RGB and “smart” LEDs
- Functions and libraries
- Soldering
- Review of the assignments

LIGHT - LEDs



LED



RGB LED



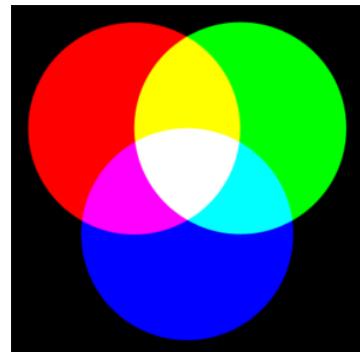
“SMART” LED



RGB LED



- It consists of 3 separate LEDs red, green and blue packed in a single case.
- The RGB LED can emit different colors by mixing the 3 basic colors red, green and blue.
- It has 4 leads, one lead for each of the 3 colors and one common cathode or anode depending of the RGB LED type.



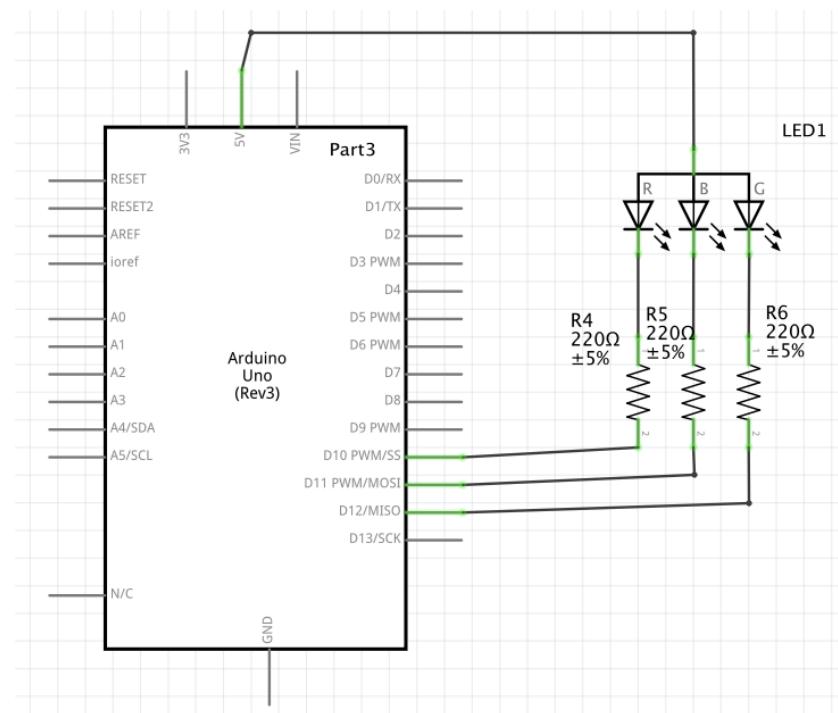
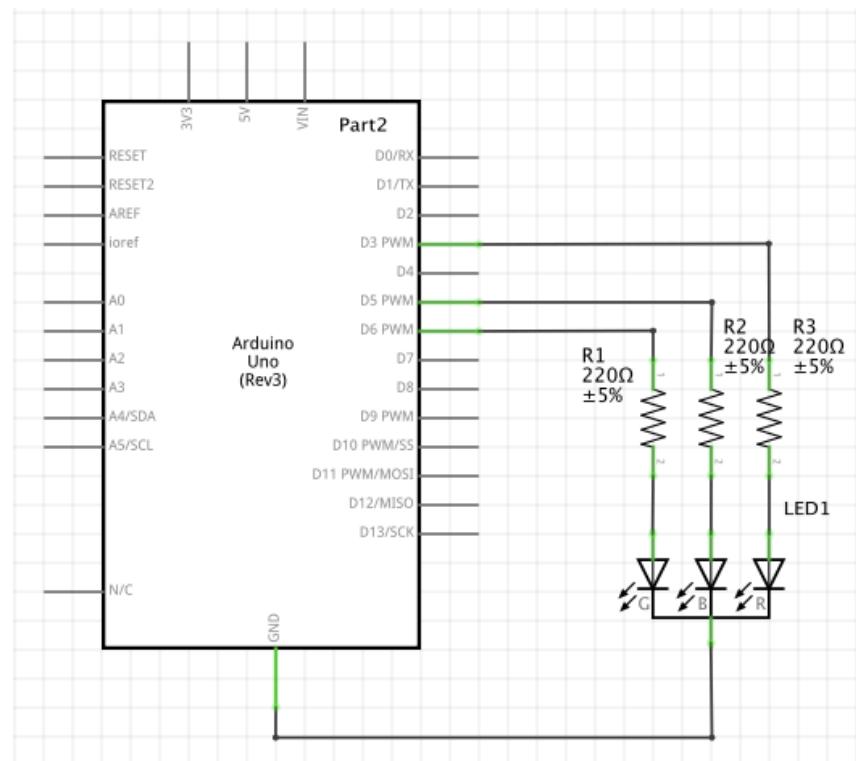
Common
Cathode (-)



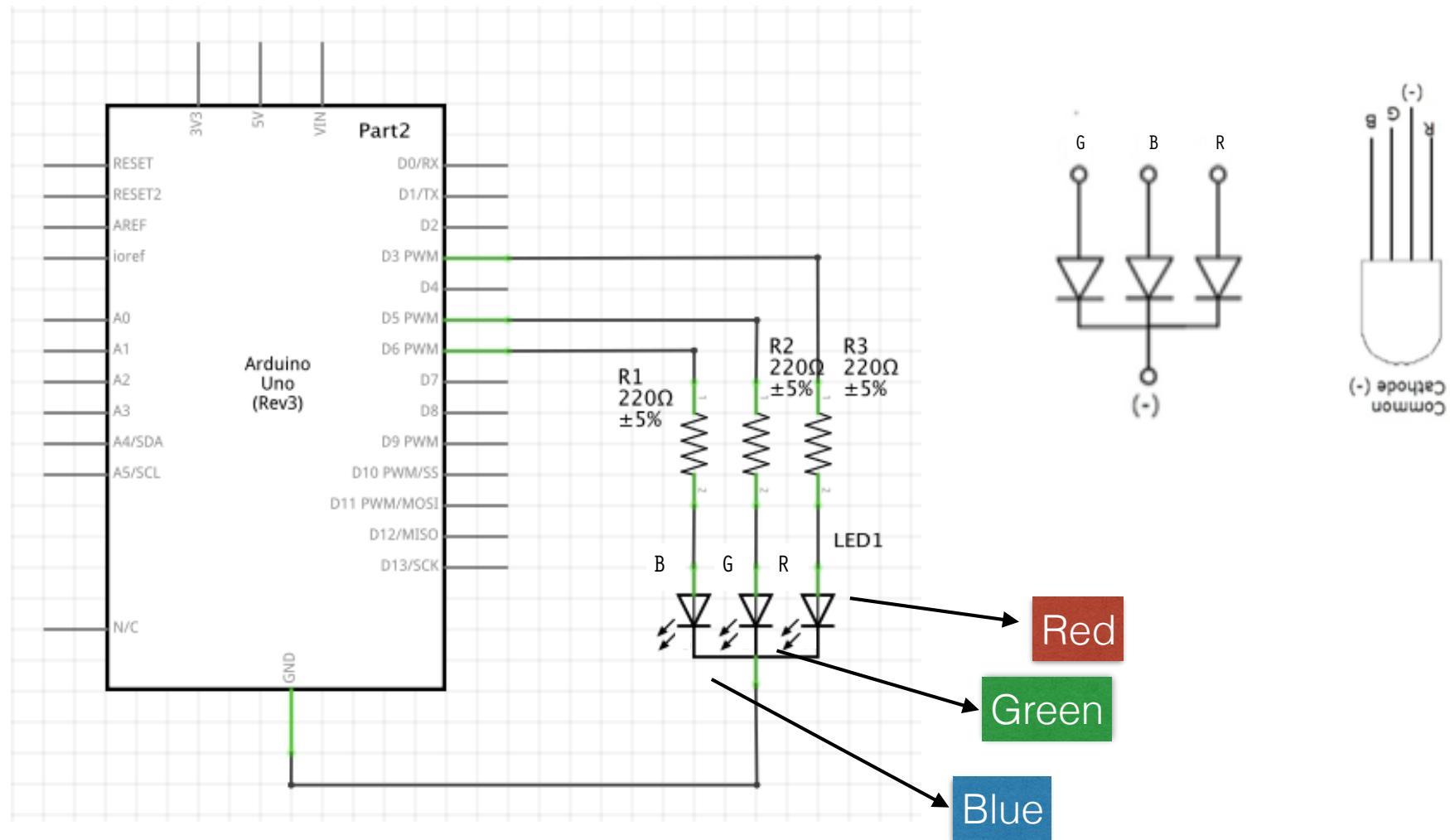
Common
Anode (+)



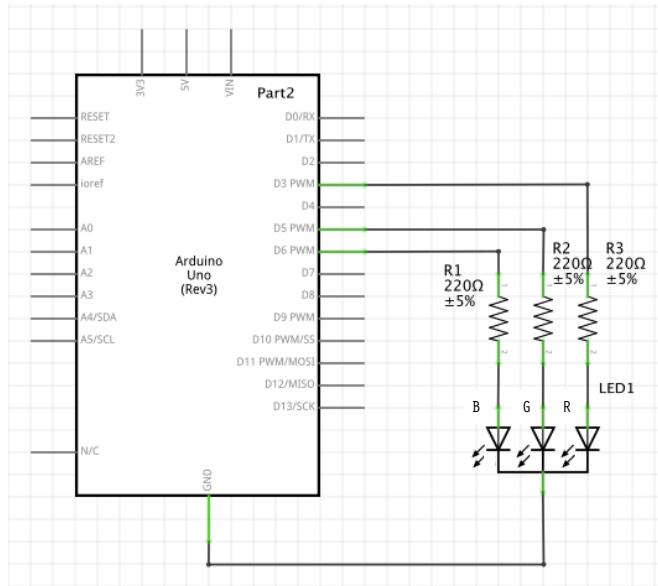
RGB LED



RGB LED



RGB LED



CODE: 01_RGB

```
int redPin= 3;  
int greenPin = 5;  
int bluePin = 6;
```

```
//Color Red  
analogWrite(redPin, 255);  
analogWrite(greenPin, 0);  
analogWrite(bluePin, 0);
```

Note:

- You can use the functions: digitalWrite and analogWrite.
- Use PWM pins and analogWrite to control the color.
- For standard RGB LEDs -> resistors = 2200hm
- RGB codes:
 - https://www.w3schools.com/colors/colors_rgb.asp
 - https://www.rapidtables.com/web/color/RGB_Color.html

FUNCTION

sketch_nov11d §

```
int redPin= 3;  
int greenPin = 5;  
int bluePin = 6;  
  
int wait = 1000;
```

```
void setup() {  
    pinMode(redPin, OUTPUT);  
    pinMode(greenPin, OUTPUT);  
    pinMode(bluePin, OUTPUT);  
}
```

```
void loop() {  
  
    //Color Red  
    analogWrite(redPin, 255);  
    analogWrite(greenPin, 0);  
    analogWrite(bluePin, 0);  
    delay(wait);  
  
    //Light off  
    analogWrite(redPin, 0);  
    analogWrite(greenPin, 0);  
    analogWrite(bluePin, 0);  
    delay(wait);  
}
```

CODE: 01_RGB

RGB_simple_function

```
int redPin= 3;  
int greenPin = 5;  
int bluePin = 6;  
  
int wait = 1000;
```

```
void setup() {  
    pinMode(redPin, OUTPUT);  
    pinMode(greenPin, OUTPUT);  
    pinMode(bluePin, OUTPUT);  
}
```

```
void loop() {  
    setColor(255, 0, 0); // Red Color  
    delay(wait);  
  
    setColor(0, 0, 0); // Light off  
    delay(wait);  
}
```

```
void setColor(int redValue, int greenValue, int blueValue) {  
    analogWrite(redPin, redValue);  
    analogWrite(greenPin, greenValue);  
    analogWrite(bluePin, blueValue);  
}
```

CODE: 02_RGB_FUNCTION

FUNCTION

```
sketch_nov11d §  
int redPin= 3;  
int greenPin = 5;  
int bluePin = 6;  
  
int wait = 1000;  
  
void setup() {  
    pinMode(redPin, OUTPUT);  
    pinMode(greenPin, OUTPUT);  
    pinMode(bluePin, OUTPUT);  
}  
  
void loop() {  
  
    //Color Red  
    analogWrite(redPin, 255);  
    analogWrite(greenPin, 0);  
    analogWrite(bluePin, 0);  
    delay(wait);  
  
    //Light off  
    analogWrite(redPin, 0);  
    analogWrite(greenPin, 0);  
    analogWrite(bluePin, 0);  
    delay(wait);  
}
```

CODE: 01_RGB

```
RGB_simple_function  
int redPin= 3;  
int greenPin = 5;  
int bluePin = 6;  
  
int wait = 1000;  
  
void setup() {  
    pinMode(redPin, OUTPUT);  
    pinMode(greenPin, OUTPUT);  
    pinMode(bluePin, OUTPUT);  
}  
  
void loop() {  
    setColor(255, 0, 0); // Red Color  
    delay(wait);  
  
    setColor(0, 0, 0); // Light off  
    delay(wait);  
}  
  
void setColor(int redValue, int greenValue, int blueValue) {  
    analogWrite(redPin, redValue);  
    analogWrite(greenPin, greenValue);  
    analogWrite(bluePin, blueValue);  
}
```

CODE: 02_RGB_FUNCTION

RGB LED - FUNCTION

```
//Color Red  
analogWrite(redPin, 255);  
analogWrite(greenPin, 0);  
analogWrite(bluePin, 0);
```

```
setColor(255, 0, 0); // Red Color
```

```
void setColor(int redValue, int greenValue, int blueValue) {  
    analogWrite(redPin, redValue);  
    analogWrite(greenPin, greenValue);  
    analogWrite(bluePin, blueValue);  
}
```

Instead to write three lines overtime you want to set a color:
We use a function: setColor

We use the function setColor inside loop and give to the function three parameters: the RGB values

We define the function setColor outside loop.

FUNCTION

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".

Anatomy of a C function

Datatype of data returned,
any C datatype.

"void" if nothing is returned.

Parameters passed to
function, any C datatype.

```
int myMultiplyFunction(int x, int y){  
    int result;  
    result = x * y;  
    return result;  
}
```

Return statement,
datatype matches
declaration.

Curly braces required.

ASSIGNMENT

RGB LED

- Read the page: <https://www.arduino.cc/en/Reference/FunctionDeclaration>

Ex 1

Write a sketch and call it: "MyMath.ino".

In MyMath create four functions that execute the four calculations:
and call the functions: MyMultiplication, MyAddition, MySubtraction,
MyDivision.

Share the code on your blog

Ex 2

- Using the code 02_RGB_rainbow: create your Color sequence and create a function called "MyRainbow". The function MyRainbow takes a parameter to change the speed of the rainbow. Document it in your blog.
- Extra: trigger the rainbow sequence with a sensor.

LIGHT - INTELLIGENT LEDs



Red, green and blue LEDs are integrated alongside a driver chip into a tiny surface-mount package controlled through a single wire.

They require a microcontroller (such as Arduino) and some programming. We provide some sample code to get you started.

RGB: WS2812, WS2811

RGBW: SK6812

NeoPixel -> [Adafruit](#)

LIGHT - INTELLIGENT LEDs

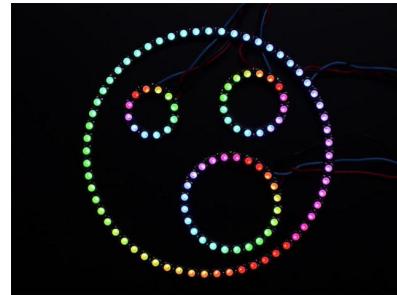
Different shapes, Form factors



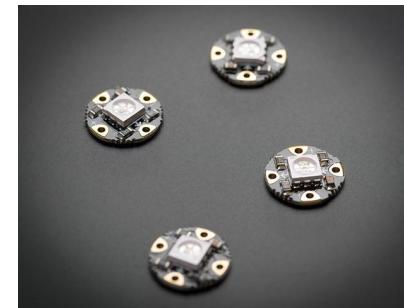
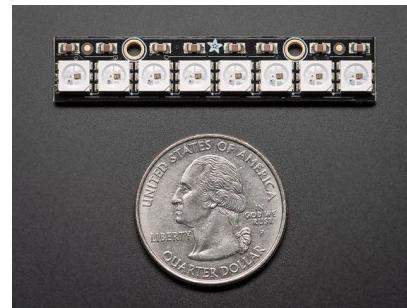
NeoPixel Strips



NeoPixel Rings



NeoPixel Matrix



WHAT YOU NEED TO SOLDER

SOLDERING STAND



TIN

SOLDERING SPONGE



SOLDERING IRON



DESOULDERING PUMP



DESOULDERING WIRE



STRIPPER WIRE



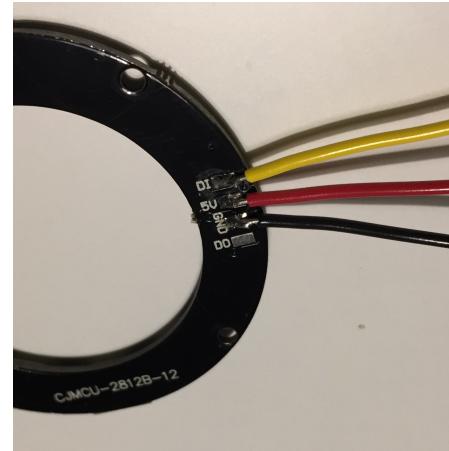
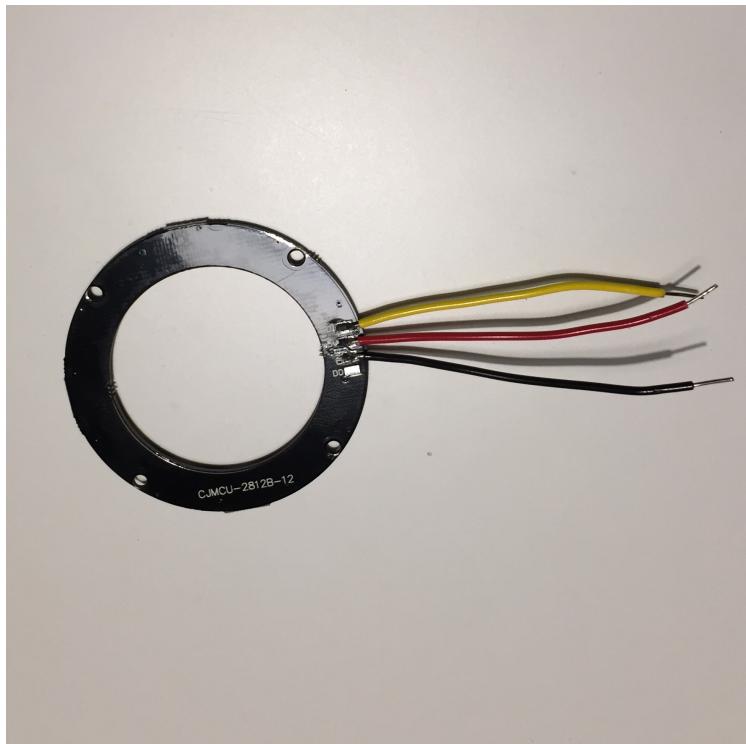
WIRE CUTTER



TWEEZERS



SOLDER

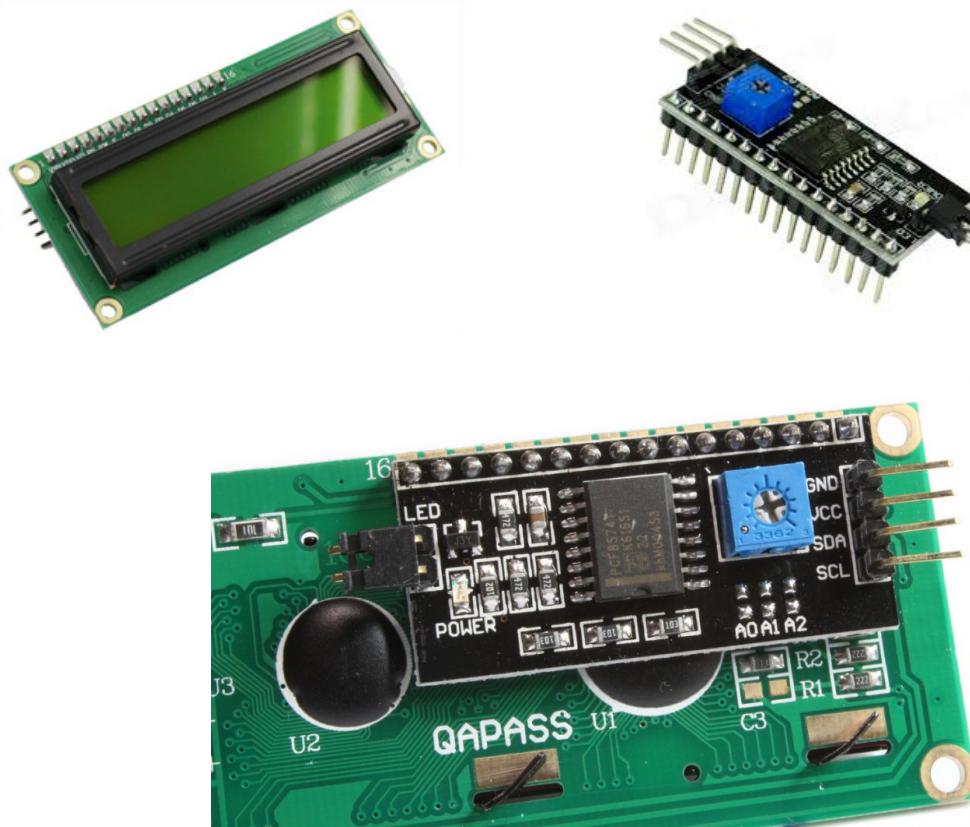


DI: YELLOW

5V: RED

GND: BLACK

SOLDER



ASSIGNMENT

NeoPixel

- Read the tutorial: <https://learn.adafruit.com/adafruit-neopixel-uberguide/the-magic-of-neopixels>. Read all the pages.

LICENCE

EXCEPT WHERE OTHERWISE NOTED, THIS WORK IS LICENSED UNDER:

[HTTPS://CREATIVECOMMONS.ORG/LICENSES/BY/4.0/](https://creativecommons.org/licenses/by/4.0/)

