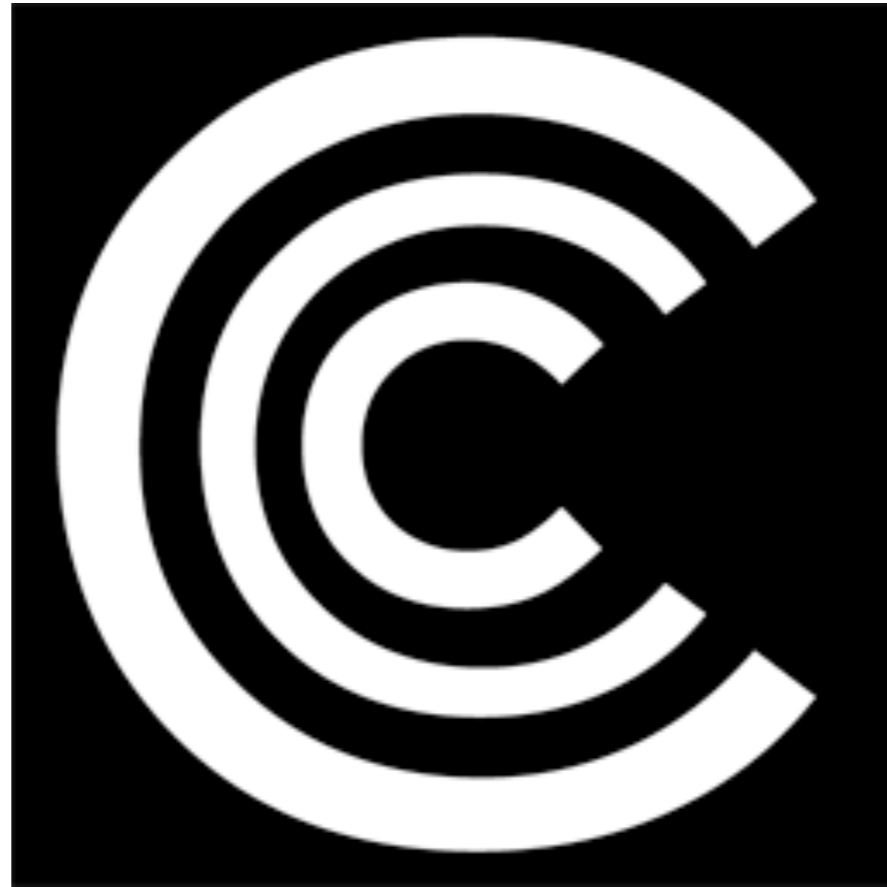


## Terms & Conditions

All Creative Computer Club Resources created by Matthew C. Applegate are licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License. Please note that some of these resources also contain images of software that is protected by copyright and are used under special agreement with these software companies, they are therefore are not covered by the Creative Commons License. The text is free to download, use, edit and redistribute, the images are free to download and use as is, unfortunately they are not available to edit and redistribute.



# Creative Computing Club

## Arduino Introduction

[www.pixelh8.co.uk](http://www.pixelh8.co.uk)

# Set Up

For this tutorial we used Arduino 1.0 some of the images and names may vary from version to version.

Download and install and setup Arduino as per the developers instructions.

Arduino will automatically generate a “Sketches” folder where your programs will be stored.

Now lets open Arduino and click on File and “Save”. Save the Arduino .ino as “Blink”

In the code you will occasionally see “ // “ these are notes and they are ignored by the computer but are there to help explain the process to you better. You can choose to type them in or not.

Arduino is probably one of my favourite ways to interface code with hardware. Like Processing, I have used innumerable times in just as many projects from games to interactive art installations. Its relatively easy to understand, easy to install and free.

For these examples I have used and Arduino Mega but all examples should be compatible with most boards..

Blue text are just notes and don't need to be copied in. Red text are changes to the previous example.

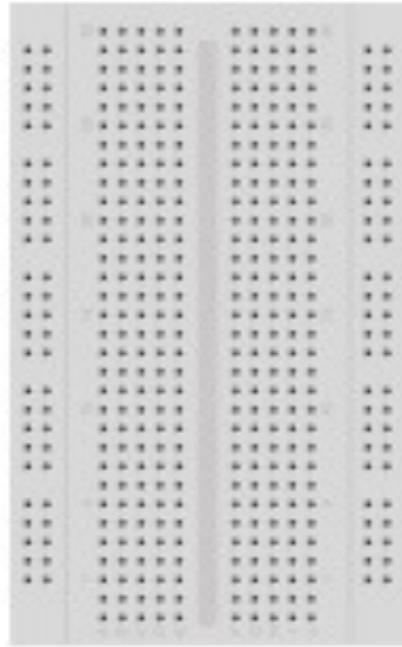
All images developed using [Fritzing](#). For more information, see the [Fritzing website](#).

Now lets get started.

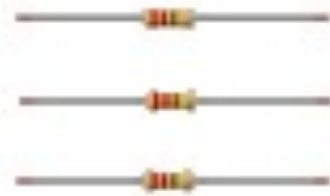
# Components



Arduino (Mega)



Breadboard



Resistors



LEDs

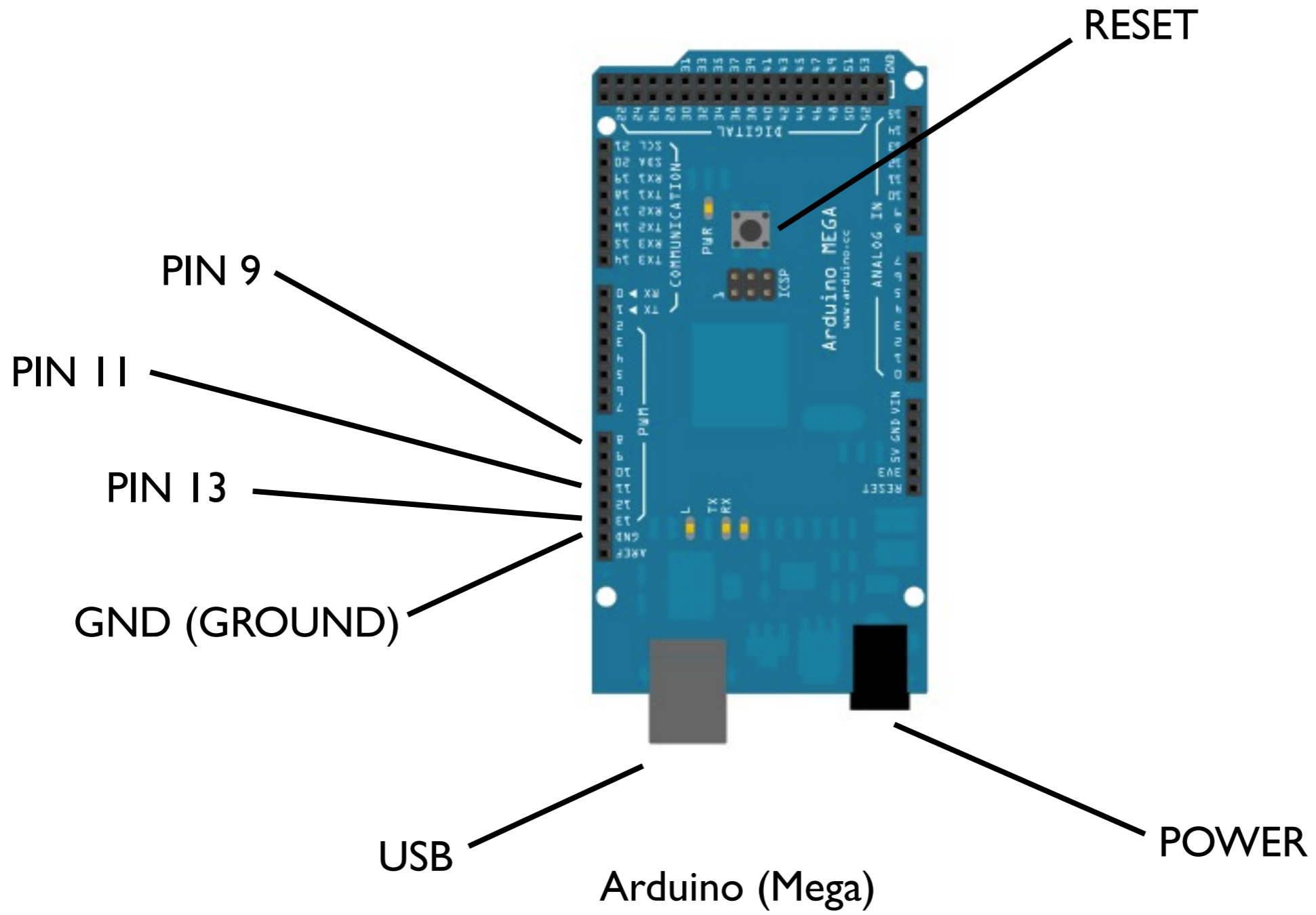


Potentiometer

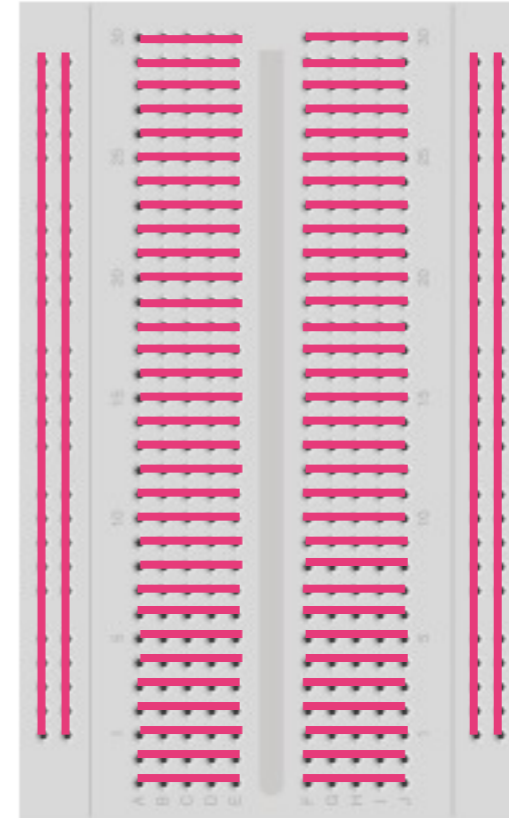
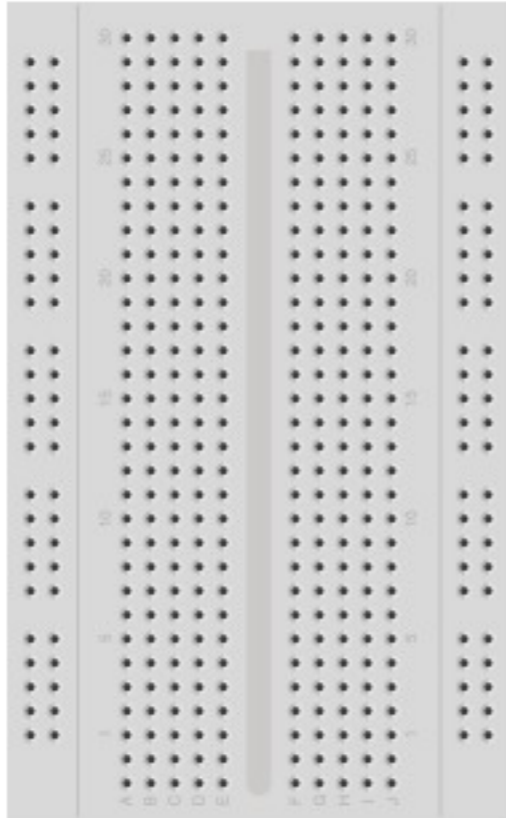


Buzzer

# Components



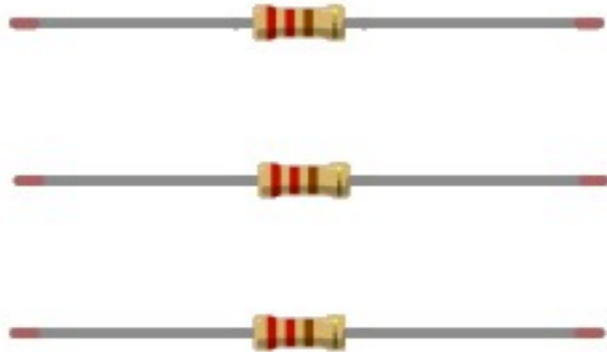
# Components



Breadboard

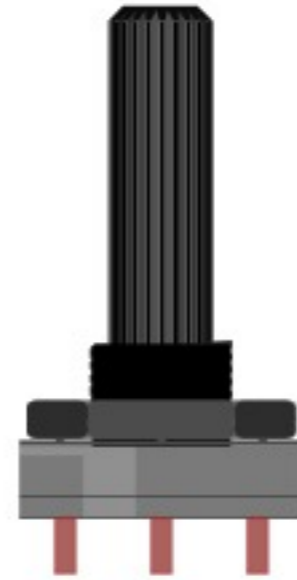
Breadboards allow you to connect components without soldering.

# Components



Resistors

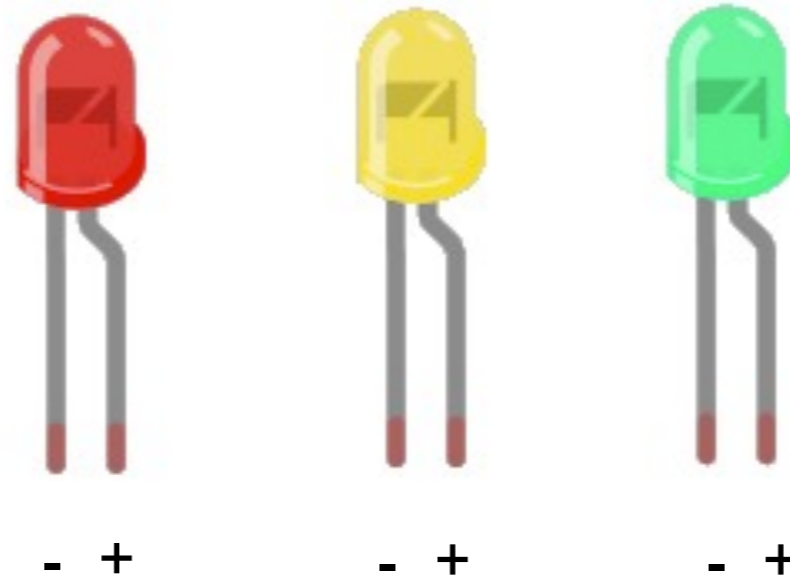
Resistors “resist” and restrict the flow of electricity.



Potentiometer (Variable Resistor) or (Pots)

These do the same as resistors but can vary the amount of restriction.

# Components



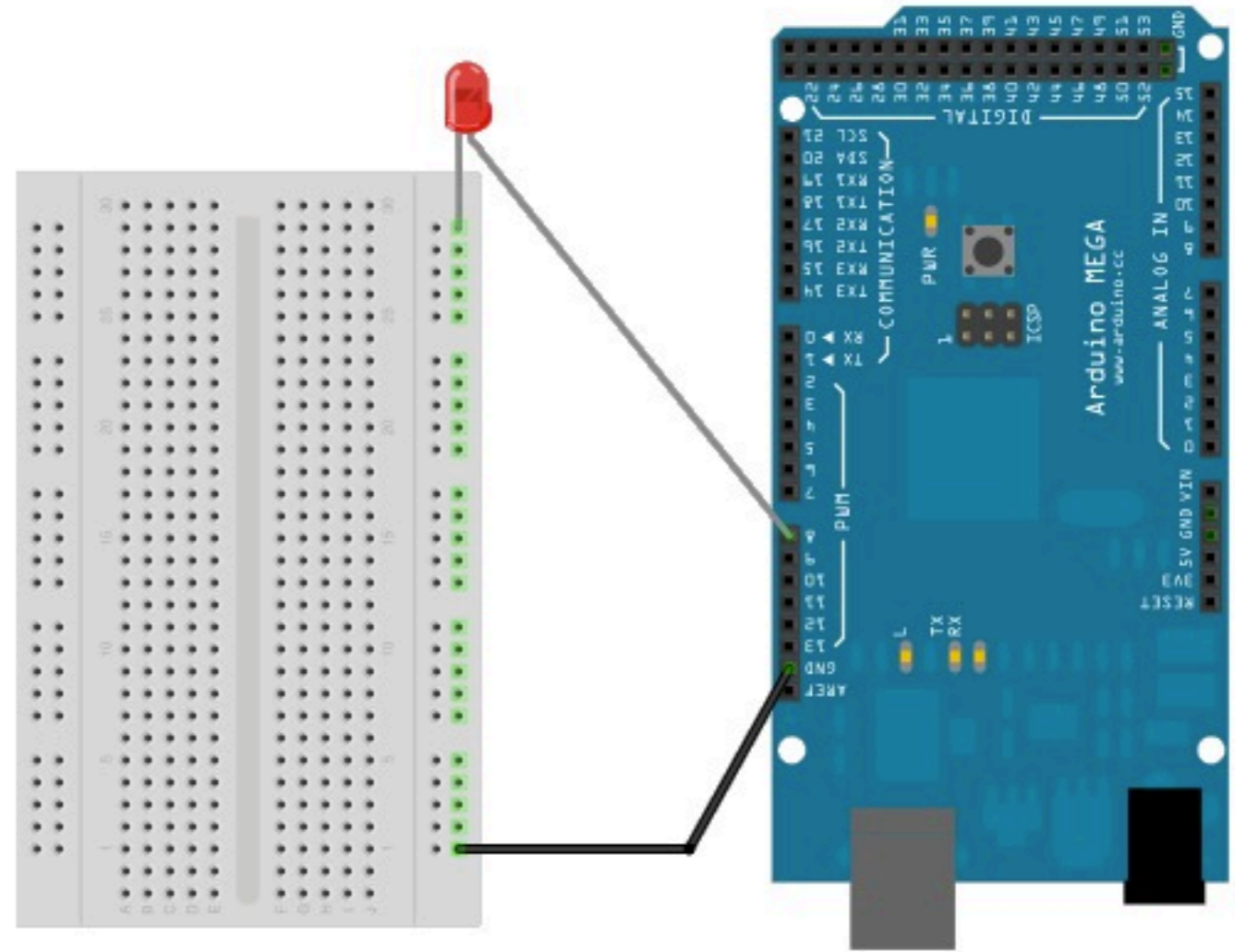
## LEDs

The long “leg” of the LED is the + positive where the power comes in, the short is the - negative.



# Blink

```
/*  
Blink  
Turns on an LED on for one second, then off for one second, repeatedly  
This example code is in the public domain.  
*/  
  
//The set up area runs once at the beginning of the program.  
void setup() {  
  // initialize the digital pin as an output.  
  
  pinMode(9, OUTPUT);  
}  
  
//The loop area loops again and again.  
void loop() {  
  digitalWrite(9, HIGH); // set the LED on //HIGH applies the current  
  delay(1000);           // wait for a second  
  digitalWrite(9, LOW);  // set the LED off //LOW removes the current  
  delay(1000);           // wait for a second  
}  
  
//End of program
```

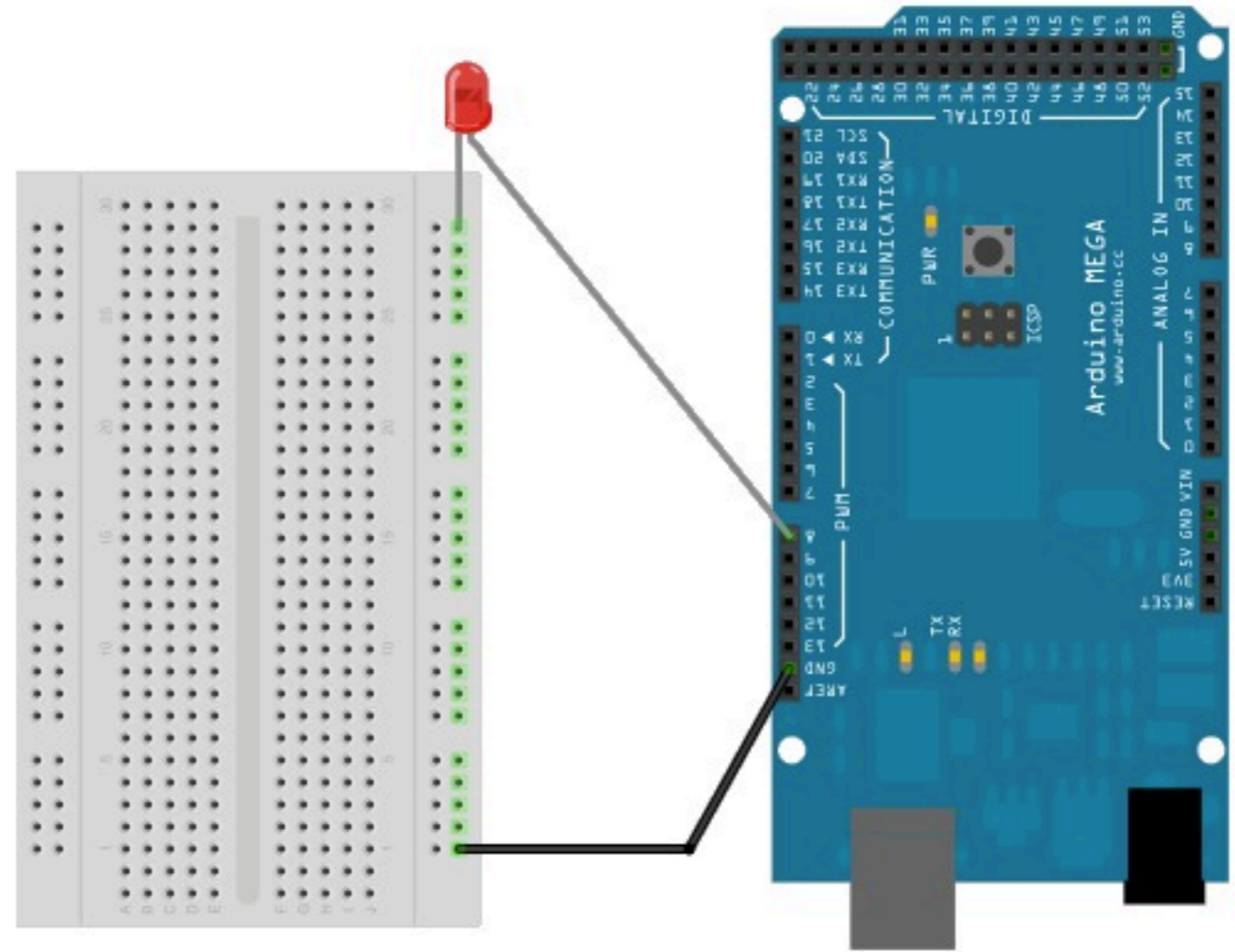


This example was adapted from the “BLINK” tutorial on the Arduino v  
The original can be found here <http://arduino.cc/en/Tutorial/Blink>

All images developed using [Fritzing](http://www.fritzing.com/). For more information, see the [Fritzing website](http://www.fritzing.com/).

# Blink V2.0

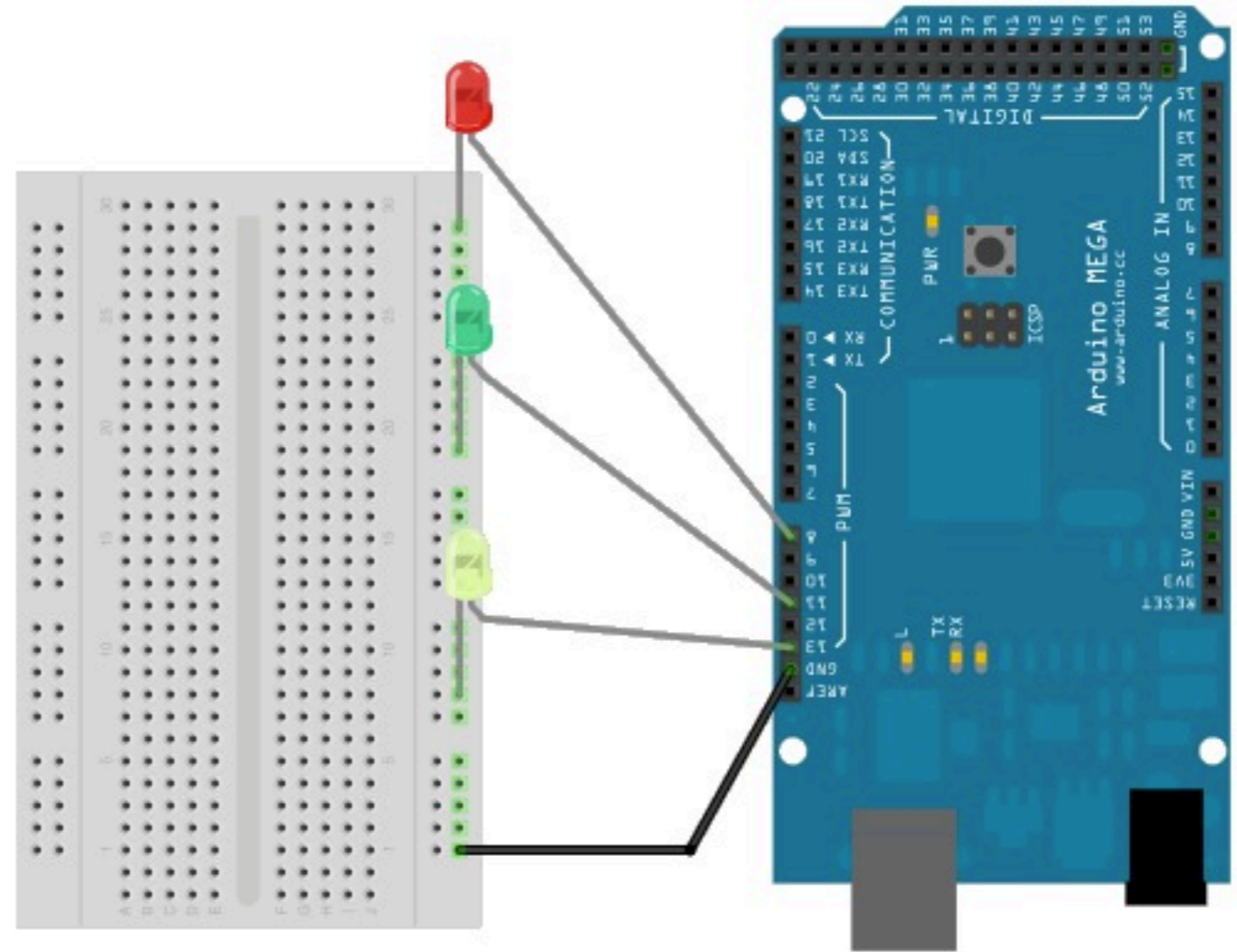
```
/*  
Blink V2.0  
Turns on an LED on for one second, then off for one second, repeatedly  
This example code is in the public domain.  
*/  
  
//We are creating a variable called red and giving it the value of 9.  
  
int red =9;  
  
void setup() {  
  pinMode(red, OUTPUT);  
  
}  
  
void loop() {  
  digitalWrite(red, HIGH);  
  delay(1000);  
  digitalWrite(red, LOW);  
  delay(1000);  
}
```



This example was adapted from the “BLINK” tutorial on the Arduino website.  
The original can be found here <http://arduino.cc/en/Tutorial/Blink>

# Blink V3.0

```
/*  
Blink V3.0  
Turns on several LEDs on, then off.  
*/  
  
//We are creating a variable called green and yellow  
  
//and giving it the value of 11 and 13.  
  
int red =9;  
  
int green = 11;  
  
int yellow = 13;  
void setup() {  
  
pinMode(red, OUTPUT);  
  
pinMode(green, OUTPUT);  
  
pinMode(yellow, OUTPUT);  
}  
  
void loop() {  
  
//This sequence will turn on the red, then turn it off, turn on the  
  
//green, then off and then yellow, then off and back to the beginning again.  
digitalWrite(red, HIGH);  
delay(1000);  
digitalWrite(red, LOW);  
delay(1000);  
  
//These control the Green  
digitalWrite(green, HIGH);  
delay(1000);  
digitalWrite(green, LOW);  
delay(1000);  
  
//These control the Yellow  
digitalWrite(yellow, HIGH);  
delay(1000);  
digitalWrite(yellow, LOW);  
delay(1000);  
  
}
```



# Blink V4.0

```
/*  
Blink V4.0  
Turns on several LEDs on, then off.  
*/
```

```
int time =1000;
```

```
int red =9;
```

```
int green = 11;
```

```
int yellow = 13;
```

```
void setup() {
```

```
pinMode(red, OUTPUT);
```

```
pinMode(green, OUTPUT);
```

```
pinMode(yellow, OUTPUT);
```

```
}
```

```
void loop() {
```

```
digitalWrite(red, HIGH);
```

```
delay(time);
```

```
digitalWrite(red, LOW);
```

```
delay(time);
```

```
digitalWrite(green, HIGH);
```

```
delay(time);
```

```
digitalWrite(green, LOW);
```

```
delay(time);
```

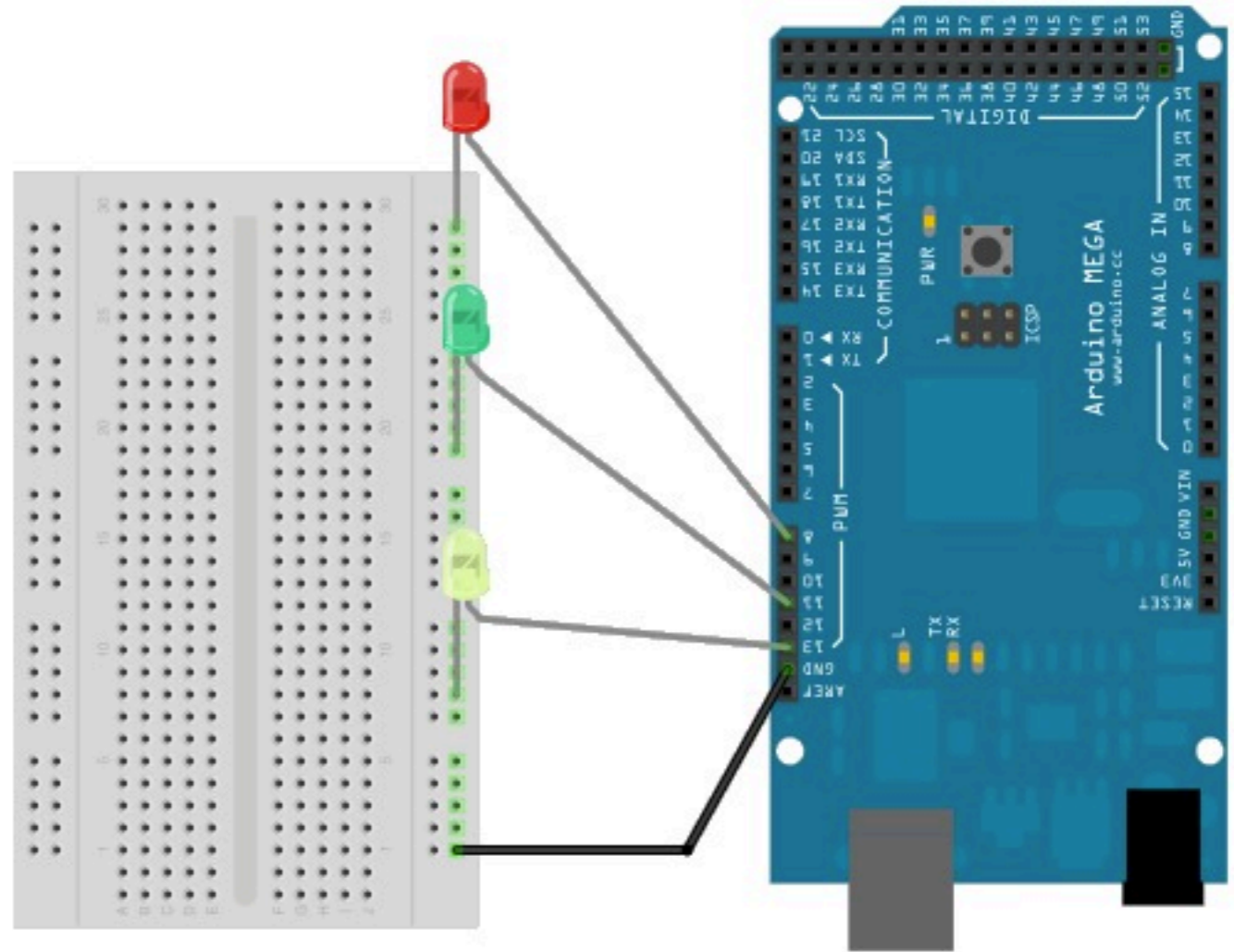
```
digitalWrite(yellow, HIGH);
```

```
delay(time);
```

```
digitalWrite(yellow, LOW);
```

```
delay(time);
```

```
}
```



## Experiment:

Change the int time = 1000 giving it different values, see how small a value you can give it before you stop noticing it turning on and off.

Move the areas of code which turn on and off the lights, and put them in a different order.

This example was adapted from the “BLINK” tutorial on the Arduino website. The original can be found here <http://arduino.cc/en/Tutorial/Blink>

[www.pixelh8.co.uk](http://www.pixelh8.co.uk)



# Blink V5.0

```
/*  
Blink V5.0  
Turns on several LEDs on, then off.  
*/
```

```
int time = 0;
```

```
int red =9;
```

```
int green = 11;
```

```
int yellow = 13;
```

```
void setup() {
```

```
pinMode(red, OUTPUT);
```

```
pinMode(green, OUTPUT);
```

```
pinMode(yellow, OUTPUT);
```

```
}
```

```
void loop() {
```

```
time = analogRead(A0);
```

```
digitalWrite(red, HIGH);
```

```
delay(time);
```

```
digitalWrite(red, LOW);
```

```
delay(time);
```

```
digitalWrite(green, HIGH);
```

```
delay(time);
```

```
digitalWrite(green, LOW);
```

```
delay(time);
```

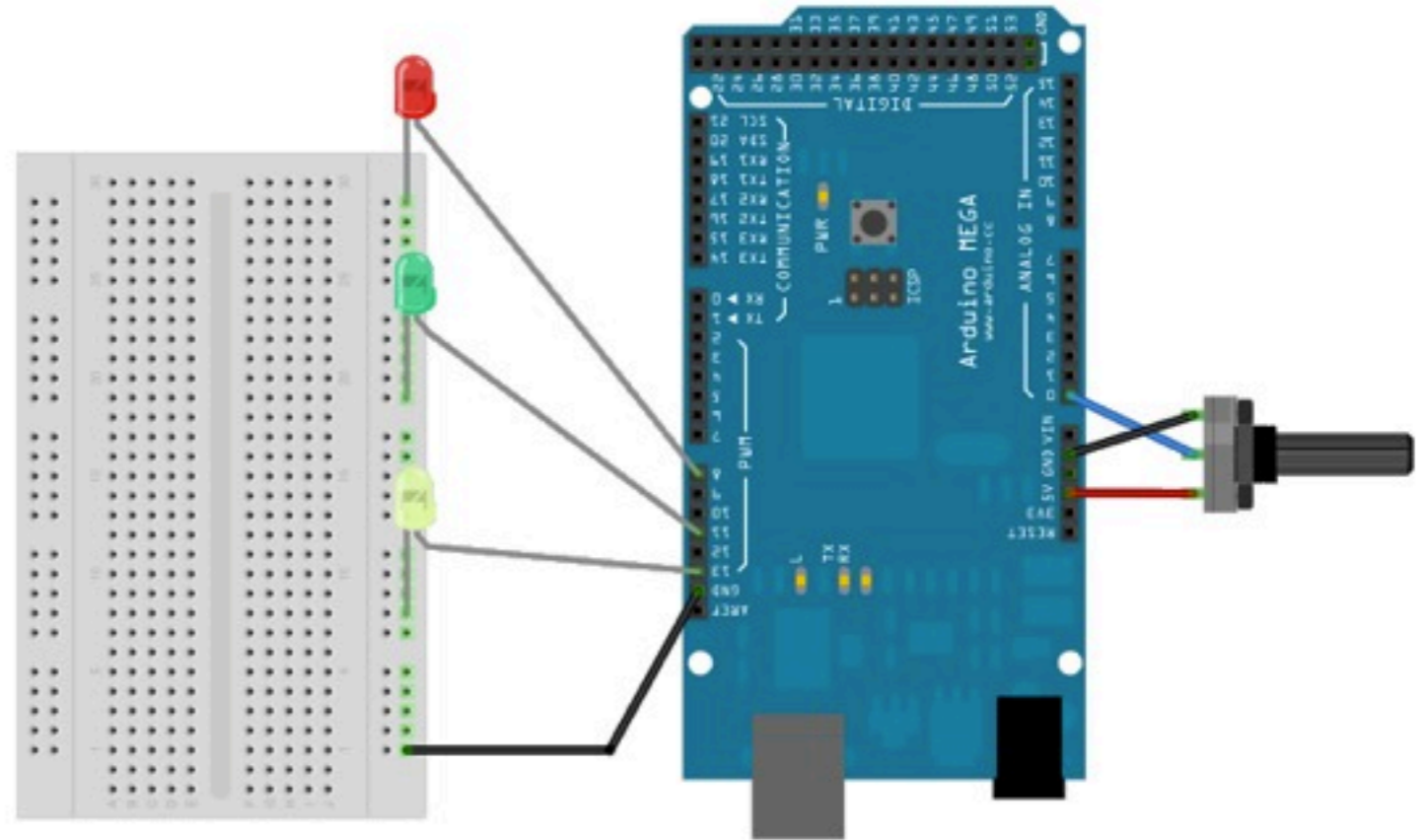
```
digitalWrite(yellow, HIGH);
```

```
delay(time);
```

```
digitalWrite(yellow, LOW);
```

```
delay(time);
```

```
}
```

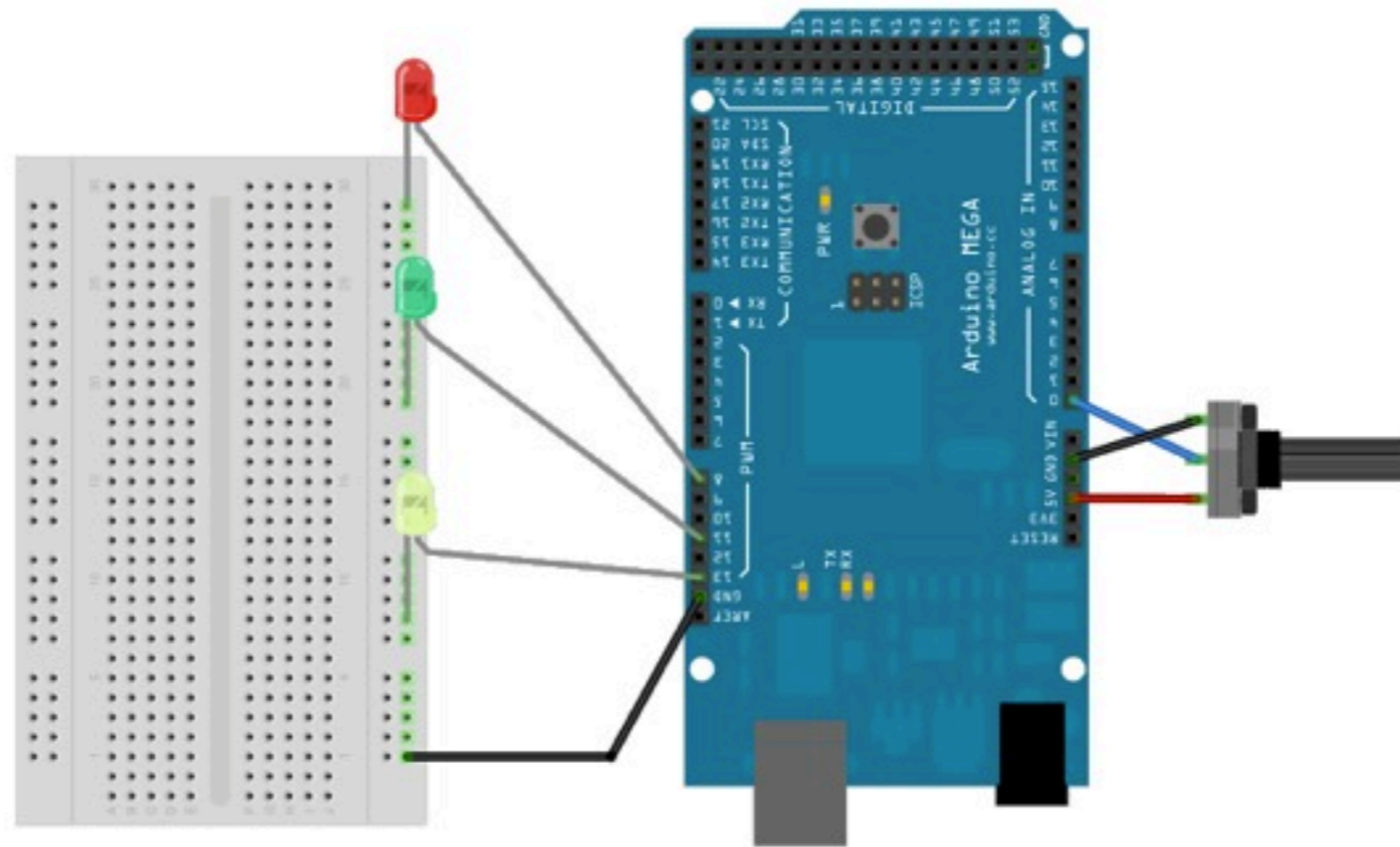


The Variable Resistor:

The Variable Resistor 0 to 5 volts, can provide a digital value between 0 and 1023. In our example that will give us the option of 0ms to 1023ms.

This example was adapted from the “BLINK” tutorial on the Arduino website. The original can be found here <http://arduino.cc/en/Tutorial/Blink>

www.pixelh8.co.uk



## Now what?

Now that you have learned a bit about Arduino and what it can do maybe start to think about what you could do with it.

Arduino is a very easy way of interfacing different things, it has a huge community and hundreds of examples online. This was purely an introduction to the wonderful Arduino development which is capable of so much more, the rest however is up to you.