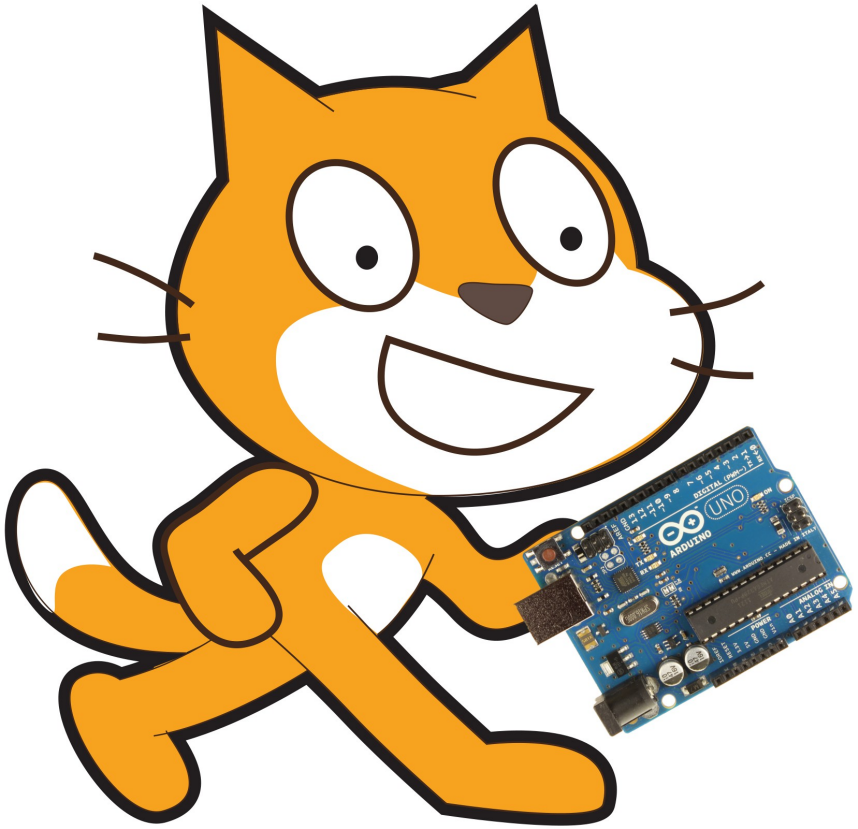
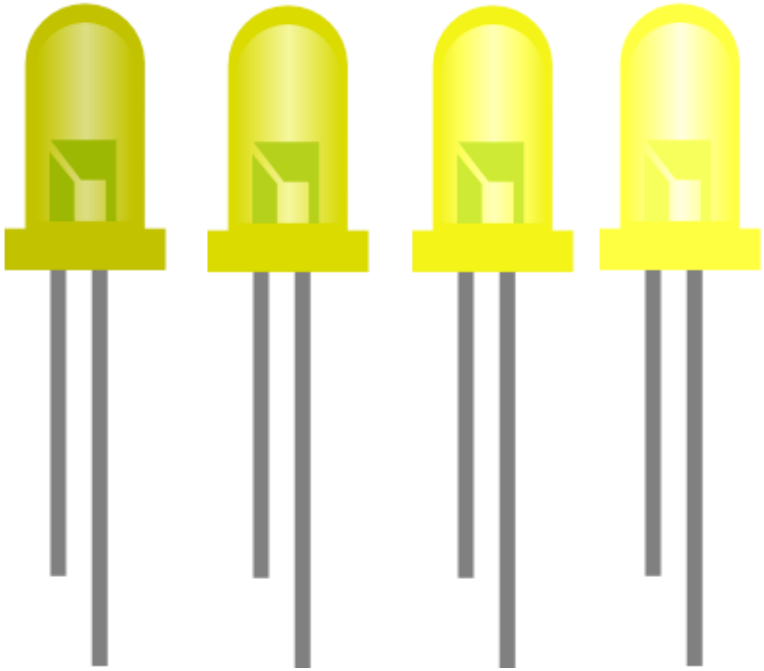
Controlling LED brightness Using Scratch V2.0





Created by Thomas Preece for the Technology Volunteers

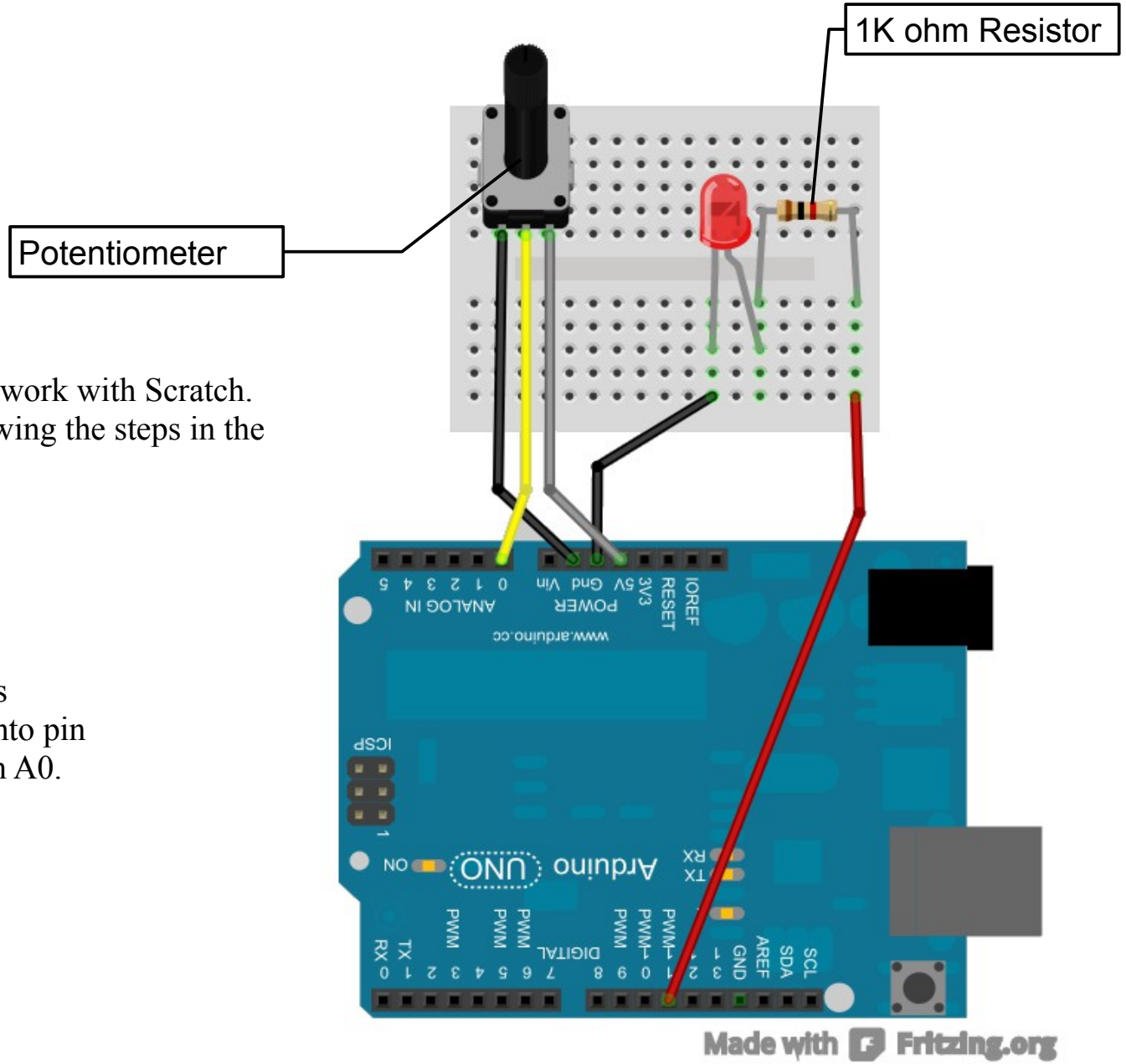
Warwick Volunteers

Step 1:

Make sure you have your Arduino ready to work with Scratch. You should have done this already by following the steps in the *'SetupYourArduinoForScratch'* document.

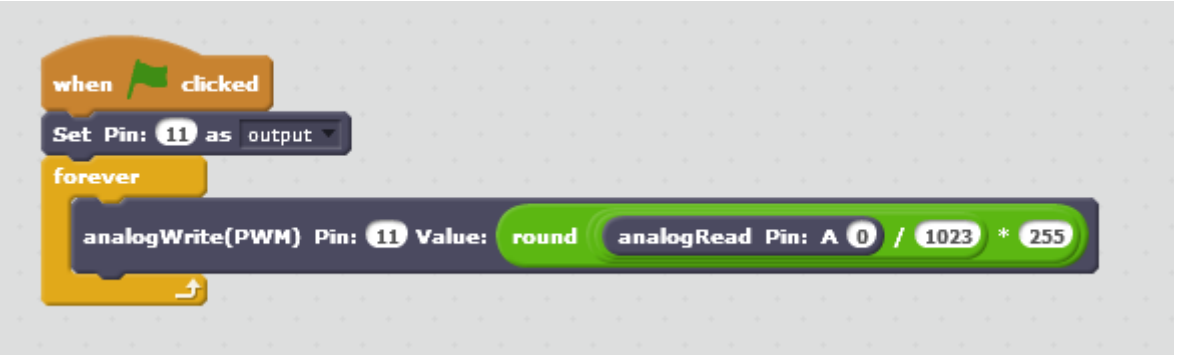
Step 2:

Create a circuit like the one to the right. It is important to note that the LED is plugged into pin 11 and the potentiometer is plugged into pin A0.



Created by Thomas Preece for the Technology Volunteers

Warwick Volunteers



Step 3:

Create a block like the one above in scratch. You have already seen the *'analogRead'* block and know that it returns a value from 0 to 1023. We then use the green blocks to transform the value *'analogRead'* returns into a value between 0 to 255. This is because the *'analogWrite(PWM)'* block requires a value from 0 to 255 instead of the usual 0 to 1023 we get from *'analogRead'* blocks.

The *'analogWrite(PWM)'* block works much in the way that the *'digitalWrite'* block works except with *'digitalWrite'* all we have is on or off where *'analogWrite(PWM)'* gives us a scale from 0 to 255 where 0 is off, 255 is fully on, 127 is half power and so on. This allows us to provide less power to our output than the full power that the Arduino is capable of. This is very useful for controlling the brightness of LED's or the speed of motors to name just two examples.

If you click the green flag and move the potentiometer you should see the brightness of the LED connected to 11 change.

It is also worth remembering that you can only use this block with pins that have PWM written next to them (hence why we moved the LED to pin 11 and didn't keep it on pin 13)

Extension:

Use loops and a variable to make the LED fade from on to off slowly.

Created by Thomas Preece for the Technology Volunteers

Warwick Volunteers

How does this all work?

In this example we use something called Pulse Width Modulation (PWM). Because the digital pins can only output 5V or 0V we cannot control the brightness of the LED by using a variable voltage, our only option left to control brightness is to use PWM.
PWM basically turns the LED on and off rapidly but varies how long it is on and off for. So say that it is on for the same amount of time as it is off then the light would be on for 50% of the time. We call this a 50% duty cycle and this corresponds to an *'analogWrite'* value of 127. Because the LED is being turned off and on so rapidly our eyes can't detect the flashing but because the LED is only on 50% of the time is appears to be half as dim as normal. Hence we can control the brightness of the LED by using different percentage duty cycles, i.e. different *'analogWrite'* values.
Again we can use a similar concept with motors. By turning a motor on only 50% of the time the motor appears to move slower. Again it doesn't appear to stop and start due to the rate at which we turn it on and off.

For more information on PWM see:
http://www.embedded.com/electronics-blogs/beginner-s-corner/4023833/Introduction-to-Pulse-Width-Modulation

Created by Thomas Preece for the Technology Volunteers