HIMALAYAN MAKERS GUILD
Foundation Activity 5
Blinking an RGB LED Using a Microcontroller

CONTENTS AND LEARNING OUTCOMES

Students will,

1. Create a circuit to control an RGB LED using an Arduino UNO microcontroller board.
2. Program the Arduino to make each colour turn on in sequence.

This activity should take **~1 hour (1.5 hours recommended)** to complete:

MATERIALS AND COSTS PER STUDENT

This activity assumes free access to computers capable of connecting to the Arduino with a USB cable and running two programs:

1. Arduino IDE: https://www.arduino.cc/en/Main/Software
2. BlocklyDuino: https://github.com/BlocklyDuino/BlocklyDuino (or a similar visual programming tool for the Arduino e.g. ArduBlock or S4A)

Both programs can be run in a web browser or downloaded for offline use. At least one computer and Arduino microcontroller board per three students is recommended.

Assuming one kit of parts per student:

Item	Qty.	Cost per Student[1]	Expendable[2]	Supplier
LED, RGB, Common Cathode	1	0.04	Y	AliExpress
Resistors, 220 ohm, ¼ W	3	0.02	Y	AliExpress
9V Battery Snap	1	0.16		AliExpress
Jumper cables, MM, 10cm	5	0.10	Y	AliExpress
Breadboard 400 point	1	1.49		AliExpress
Arduino UNO with cable	1	6.62		AliExpress
9V Battery, Ni-Mh, 450mAh	1	5.17		AliExpress
Total Cost per Student		**$13.61CAD**		

1. *Currency is CAD, 2017-06-10. Assuming one set of parts per student.*
2. *Likely to be broken or lost during the activity.*

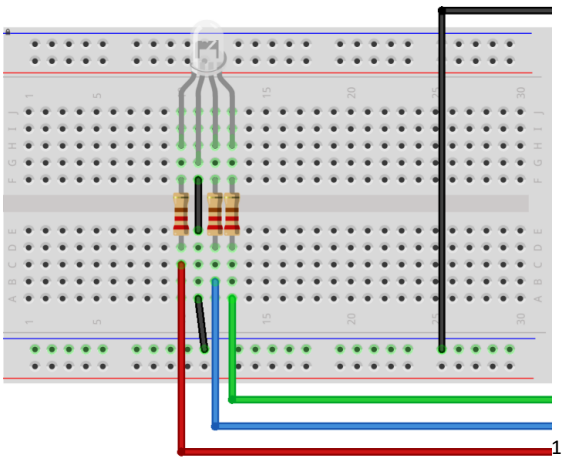
Each student should also get one printed copy of the activity handout.

LESSON

Bold text indicates directions or notes specifically for the instructor.

For additional information see:
www.jodyculkin.com/wp-content/uploads/2014/03/arduino-comic-2014.pdf

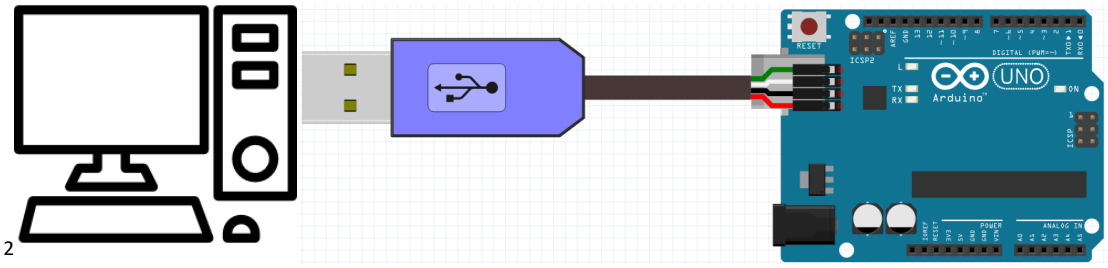
Before class: install the necessary software (see §Materials and Costs per Student) on the computers and confirm that the Arduino UNO is recognized by the computer when plugged in. As with the previous activity, assemble the LED circuits before the activity begins using three 220 ohm resistors, and leave it disconnected from the Arduino (the focus of the activity is controlling the LED with the Arduino, rather than building the circuit).



[1]Breadboard and Arduino images made using Fritzing.

ACTIVITY OVERVIEW (2 MINUTES)

In the last activity we connected an LED circuit to a digital output pin on the Arduino, then wrote a program for the Arduino so that it would blink the light on and off, and uploaded it to the Arduino.

 [2]

Today, we're going to:

1. Learn how to use multi-coloured RGB LEDs
2. Program the Arduino to control the RBG LED colours using multiple digital output pins.

REVIEW OF ARDUINO PROGRAMMING (15 MINUTES)

Provide students with a review of Arduino programming according to the following key points:

The Arduino microcontroller board acts as the brain of our circuit. We can write instructions for the Arduino, and it will follow the instructions repeatedly.

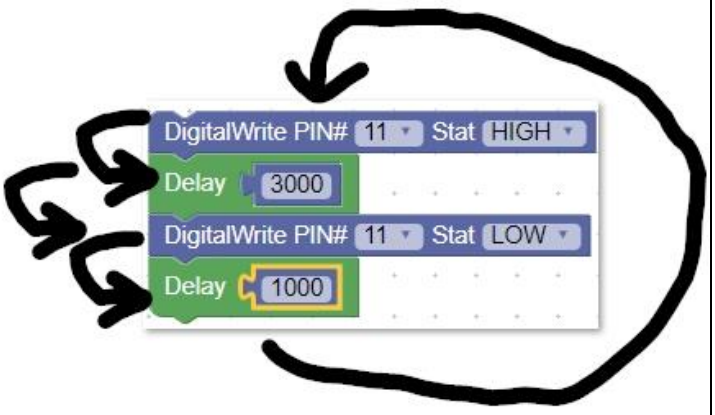
Let's look at the program we wrote last time. What are the two types of blocks we used? A: Delay blocks for making the Arduino wait, and DigitalWrite blocks to change the voltage on a digital output pin.

There are three important programming concepts that we used last activity to blink the LED circuit:

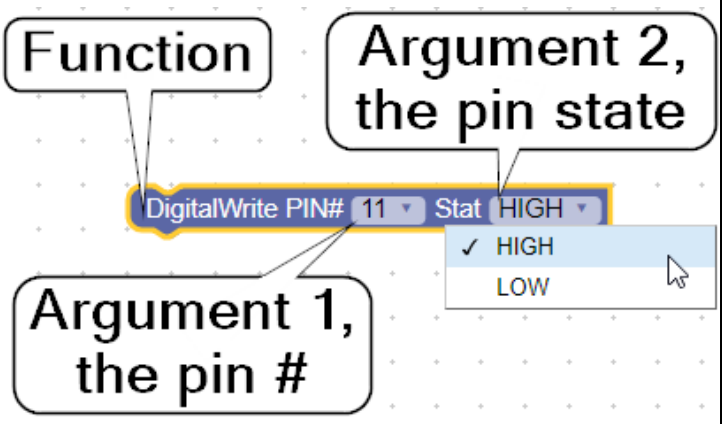
1. The Arduino will run the instructions from top to bottom. (This is known as sequential execution.) So, what does this program do? (Walk through each step)
   A: Makes pin 11 go HIGH (5V) for 3 seconds, then LOW (0V) for 1 second. So, if we attach PIN 11 to an LED circuit, the LED will blink.

[2]Icons made by catkuro from www.flaticon.com is licensed by CC 3.0 BY.

2. **Loops**: When the Arduino reaches the bottom of the instructions, it loops back around to the top and continues to run the instructions again. So, our program is a loop! After the light has been off (LOW) for 1 second (1000 ms), it will turn on again.



3. **Functions**: each block we use has a special function. In fact, we can call these blocks "functions." The first function (DigitalWrite) makes a pin go HIGH or LOW. We can select two options: the pin number we want to control, and the state we want to set it to (HIGH or LOW). These options are called **arguments**.



The second function (Delay) makes the Arduino wait a certain number of milliseconds. How many arguments does Delay have? A: one, the number of milliseconds to wait.

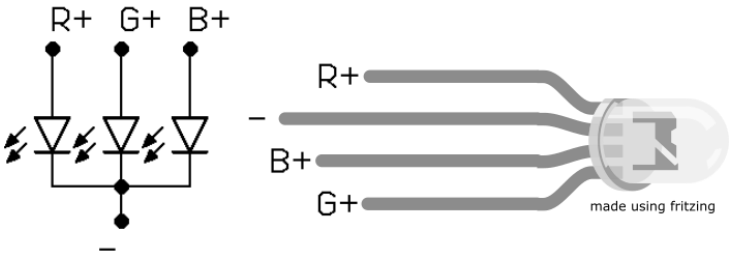
To program the Arduino, follow these 9 steps:

1. Open BlocklyDuino program and the Arduino IDE program.
2. Write instructions for the Arduino microcontroller using blocks in BlocklyDuino.
3. Click the "Arduino" tab in BlocklyDuino, select the code, and copy it
4. Go to the Arduino IDE and delete any code already there.
5. Paste the code from BlocklyDuino into the Arduino IDE.
6. Make sure the Arduino is connected to the computer using a USB cable
7. Click "Tools" on the top menu bar in the Arduino IDE, and make sure that "Arduino UNO" is selected under "Board".
8. Click "Tools" on the top menu bar in the Arduino IDE, go to "Port", and select the port that appears there after the Arduino is connected.
9. Click the arrow button  to upload the program to the Arduino

INTRODUCTION TO RGB LEDS (5 MINUTES)

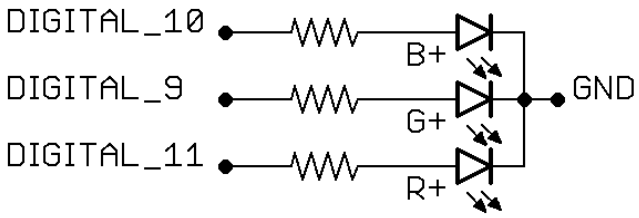
Introduce students to RGB LEDs with the following key points:

Today we're going to control a 3-coloured LED. The LED contains 3 small LEDs in one package, red, green and blue. So, we call it an RGB LED. The circuit diagram symbol and pin-out look like this:



Each of the 3 colours can be turned on separately, and they are a little different. To make sure they each turn on reliably, they need their own resistors. (**See §FAQ for more details**)

We'll connect each colour to a different Arduino digital output pin, for example pin 9, 10, and 11:

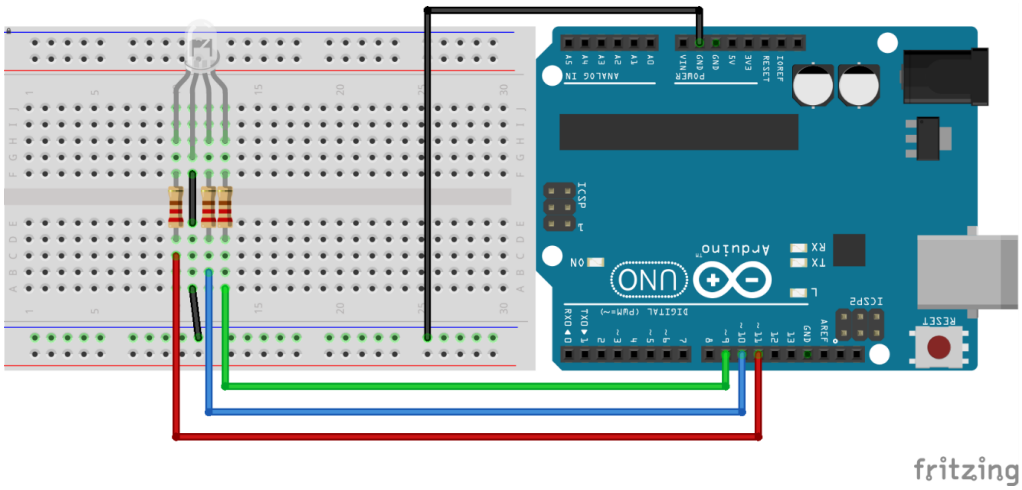


PROGRAMMING THE ARDUINO TO CONTROL AN RGB LED (20 MINUTES)

Distribute the Arduinos, USB cables, and RGB LED circuits. Ask the students to connect the circuits to the Arduinos, plug the Arduinos into the computers, and start writing their program using BlocklyDuino and the Arduino IDE.

Our goal is to make each of the three colours blink, one after the other. To do this, we will need to turn one on at a time, and make sure the other two are off, using the DigitalWrite and Delay functions.

Here is an example of the LED circuit connected to pin 9 (green), 10 (blue), and 11 (red):



DEBRIEF DISCUSSION (5 MINUTES)

Encourage a discussion among the students for them to share their thoughts on the activity.

Today we wrote and tested an Arduino program to control the colour of an RGB LED. Why is this important? What applications does this have? Some possible answers include:

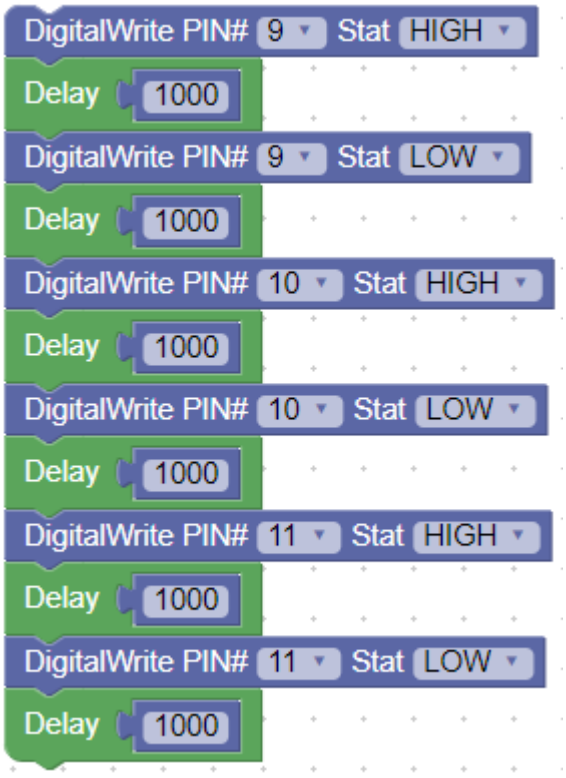
- If we control the brightness of red, green, and blue, we can make any colour! In fact, this is how modern computer monitors and TVs work. They're made of millions of tiny RGB lights packed tightly together.
- Controlling multiple digital outputs at the same time opens up new possibilities for doing things with the Arduino, not just controlling multiple LED lights. We could control many different outputs at the same time (motors, buzzers, switches, etc.).

What worked? What didn't work? Why didn't it work? What could we do next, or how could we make the circuit better?

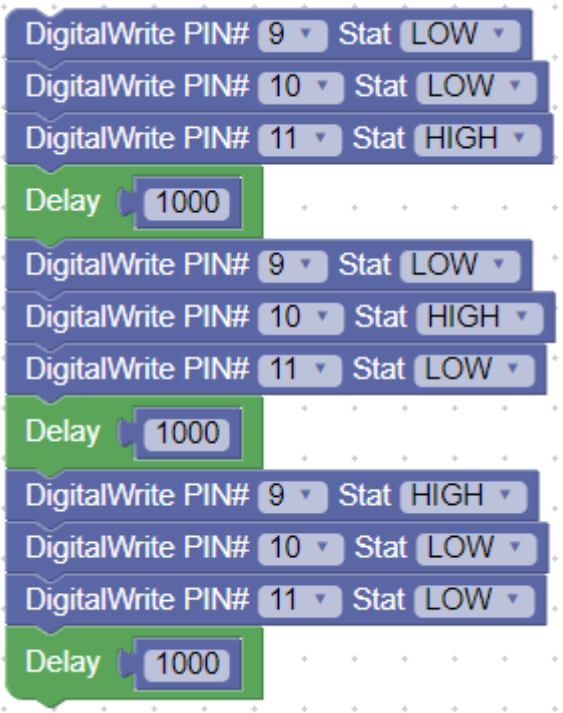
CHALLENGE AND EXPLORE

If a student completes the lesson early, evaluate their understanding by asking them to try the following:

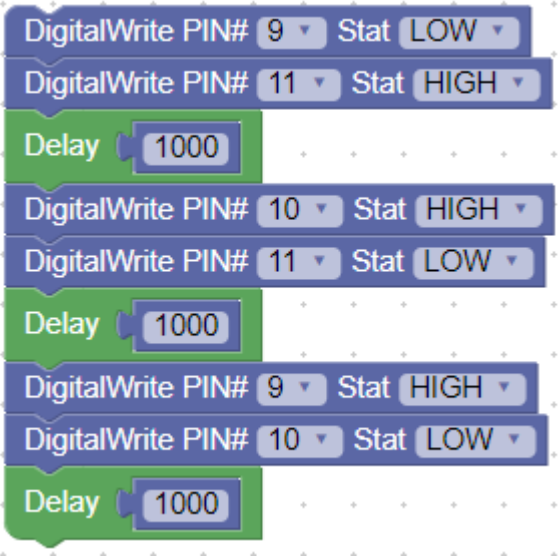
- If the light turns off between colours, can you make the colour change without any off-time in between?
    - Skill: controlling multiple digital IO pins
      Before (off between colours):



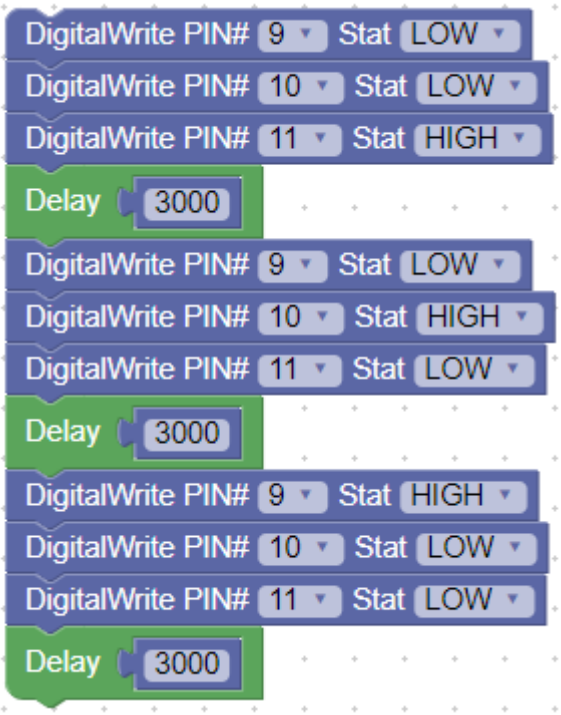
After (no delay between colours):



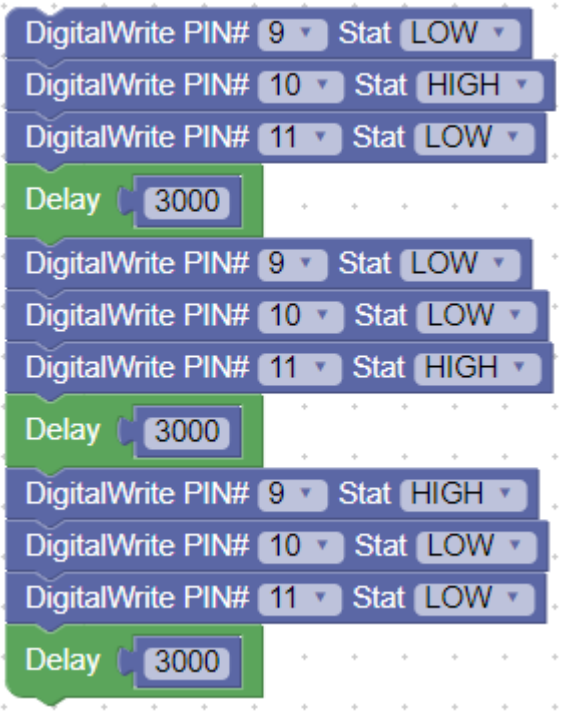
This will also work:



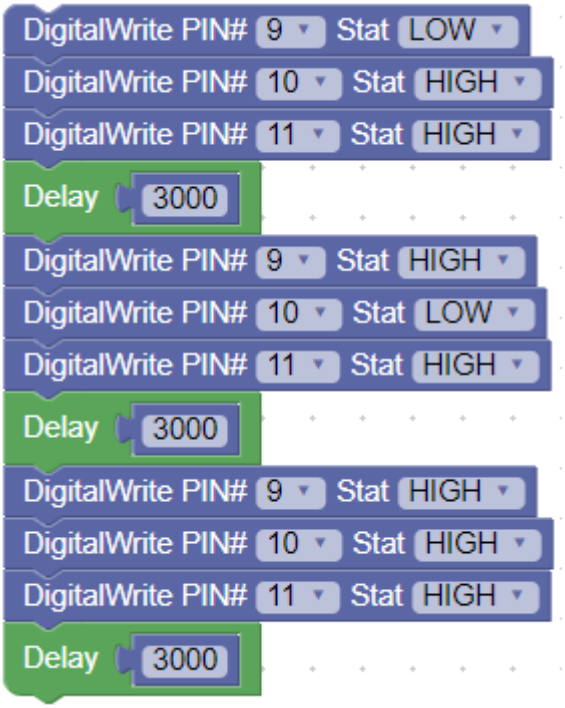
- Can you change the time that each light is on for?
    - Skill: understand function inputs
      e.g. each colour on for 3 seconds:



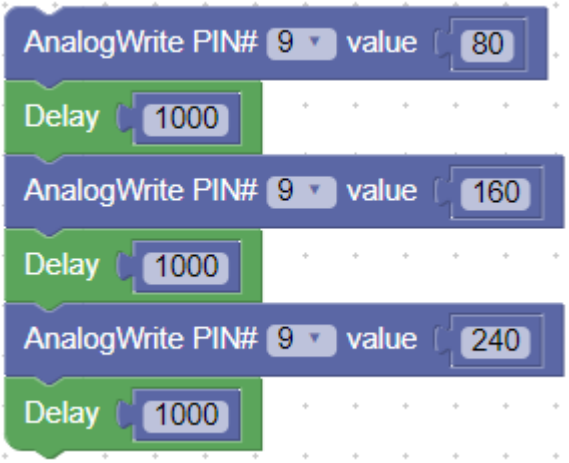
- Can you change the order of the light sequence to blue, green, then red?
    - Skill: RGB LED circuit wiring and corresponding representation of pins in the program; control multiple digital IO pins
      e.g. (depends on how pins are connected to the LED)



- Can you make the LED alternate between purple, orange/yellow, white?
    - Skill: control multiple digital IO pins; understand the RGB LED circuit and use it to mix colours (purple = RB, orange/yellow = RG, white = RGB)
      e.g. if pin 9 = green, 10 = blue, 11 = red

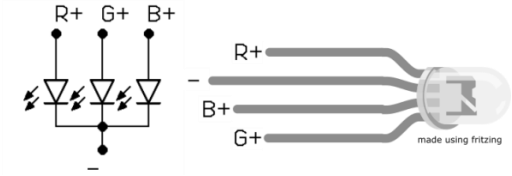


- Can you change the brightness of the colours? (introduce analogWrite() function)
    - Skill: analog output from "digital" pins with a tilde (~) in front of the pin number, to prime the students for discussing analog, digital, and PWM. The AnalogWrite block can take values from 0 to 255, with 255 being the brightest:



FREQUENTLY ASKED QUESTIONS

- Why is the code not compiling properly?
  - Make sure that all code was deleted from the Arduino IDE before pasting the new code from BlocklyDuino. Also, make sure that all the code was properly copied from BlocklyDuino!
- Why is the code is not uploading successfully?
  - Make sure the Arduino is connected, and that the correct COM port is selected (step 6-8 in §Programming the Arduino)
- Why isn't one of the colours turning off and on?
  - A: have you successfully uploaded the code using the USB cable?
  - A: do the pin numbers you used in your code match the pin numbers where the RGB LED circuit is connected?
  - A: check the polarity of the RGB LED and make sure that the (+) side is connected towards the pin, and (-) is connected to GND.



- Why do we use a separate resistor for each colour? What would happen if we just used one resistor on the (-) pin?
  - A: Try it out and see what happens! Each of the colours has a different turn on voltage, and the voltage to turn on the red LED is lower (1.8V) than green or blue (2.5V). Because of this, if only one resistor is used on the (-) pin the current will take the path of least resistance, which is through the red LED, and the other colours won't be able to turn on! However, when each has its own resistor, the resistors are in parallel and the path of least resistance for the current is to flow through all 3 resistors at once, causing all three LEDs to turn on.