

HIMALAYAN MAKERS GUILD

Foundation Activity 4

Blinking an LED Using a Microcontroller

CONTENTS AND LEARNING OUTCOMES

Students will,

1. Build an LED light circuit controlled by an Arduino UNO microcontroller board
2. Program the Arduino to make the LED blink on and off

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

<i>Materials and Costs per Student</i>	1
<i>Lesson</i>	2
<i>Activity Overview (2 minutes)</i>	3
<i>Introduction to the Arduino UNO (5 Minutes)</i>	3
<i>Digital Input and Output (10 minutes)</i>	4
<i>Programming the Arduino (10 minutes)</i>	6
<i>Connect the LED Circuit to the Arduino (5 Minutes)</i>	7
<i>Program the Arduino and Test (15 minutes)</i>	7
<i>Debrief Discussion (5 minutes)</i>	8
<i>Challenge and Explore</i>	8
<i>Frequently Asked Questions</i>	9

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)

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 Assorted 3mm 5mm	2	0.04	y	AliExpress
Resistors Assorted 2100pcs	1	0.01	y	AliExpress
9V Battery Snap	1	0.16		AliExpress
Jumper cables MM MF FF 10cm	2	0.04	y	AliExpress
Breadboard 400 point	1	1.49		AliExpress
Arduino UNO with cable	1	6.62		AliExpress
9V Ni-Mh 450mAh	1	5.17		AliExpress
Total Cost per Student		\$13.54 CAD		

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

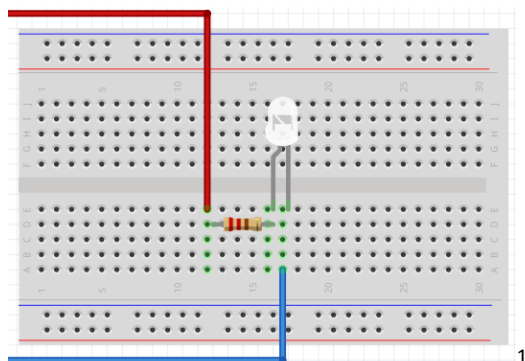
Bold text indicates directions or notes specifically for the instructor.

For additional information see:

www.jodyculkin.com/wp-content/uploads/2014/03/arduino-comic-2014.pdf

<https://learn.adafruit.com/ladyadas-learn-arduino-lesson-number-0?view=all>

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. Assemble the LED light circuit shown below using a 220 ohm resistor, 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 combined resistors in parallel and series to control the volume of a buzzer. When resistors were combined in series, less current would flow giving a lower buzzing volume. However, adding resistors in parallel gave the current extra pathways to flow through, so more current would flow and the buzzing volume increased.

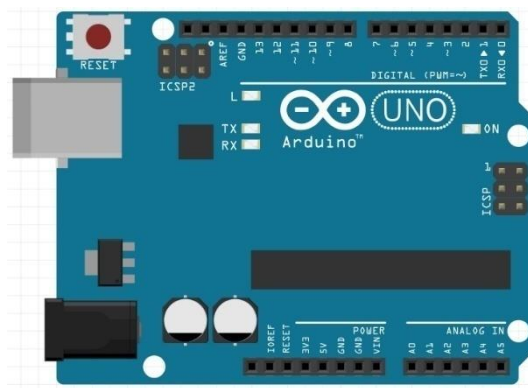
Today, we're going to:

1. Learn what microcontrollers are, and how to interact with them
2. Program an Arduino microcontroller board to make an LED blink

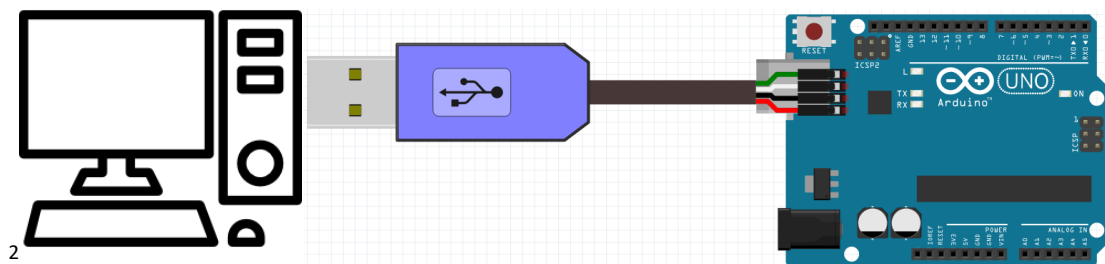
INTRODUCTION TO THE ARDUINO UNO (5 MINUTES)

Provide students with an introduction to the Arduino UNO using the following key points.

The Arduino UNO is an electronics prototyping board with a microcontroller on it. **(Draw a simple outline of the Arduino UNO):**

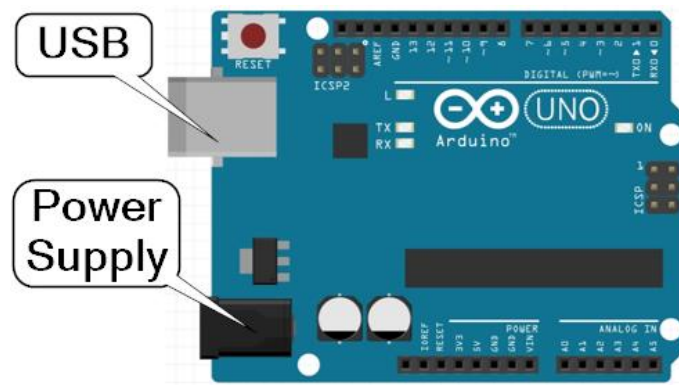


Microcontrollers are like small computers. They can act as the brain of our project, and we can program them with specific instructions. We write these instructions (otherwise known as code, or programs) on the computer, then upload them to the microcontroller through a USB cable.



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

We will power the Arduino either through the USB cable or by connecting a 9V battery to the power supply on the board.



We can attach sensors to the Arduino, then instruct the Arduino to do things when it senses a change. For example, we could attach a light sensor, then turn on an LED when the room becomes dark.

We can also make the Arduino blink an LED at a certain speed, by instructing the Arduino to wait for a set amount of time.

DIGITAL INPUT AND OUTPUT (10 MINUTES)

Computers can read information from the outside world and communicate information to the outside world. These are called input and output. Common examples include a keyboard and a monitor.



Microcontrollers like the Arduino also use inputs and outputs, by changing the voltage on its input/output (I/O) pins. Digital I/O values can be either HIGH or LOW.

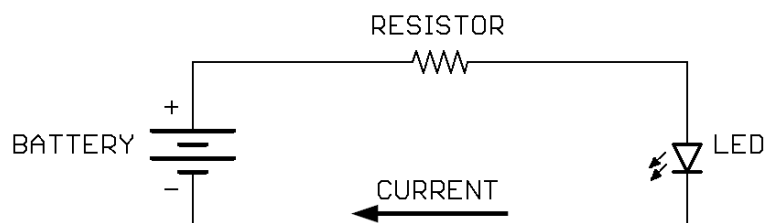


Previously, we talked about the (+) and (-) sides of a battery. The Arduino microcontroller provides +5V and GND (0V). So, for the Arduino, a digital HIGH state is represented by 5V and a LOW is represented by 0V.

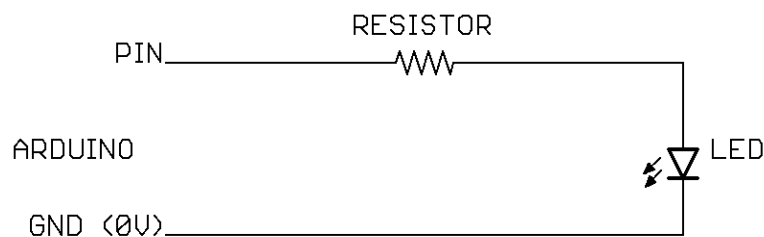
Battery	Arduino UNO	Digital Value
(+)	5V	HIGH ³
(-)	0V (GND)	LOW ⁴

We are going to use an Arduino to blink an LED light. Instead of connecting the LED circuit to the (+) and (-) sides of a battery, we will connect it to GND and a digital output pin of the Arduino. When the pin changes from LOW (0V) to HIGH (5V), the light will turn on.

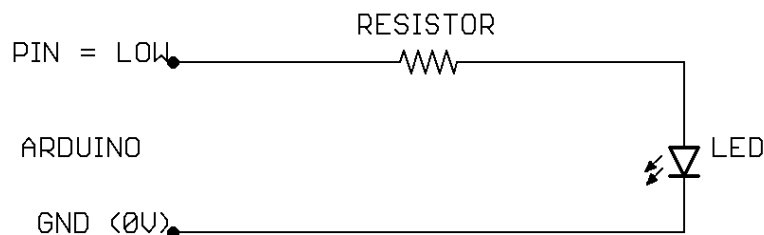
Before:



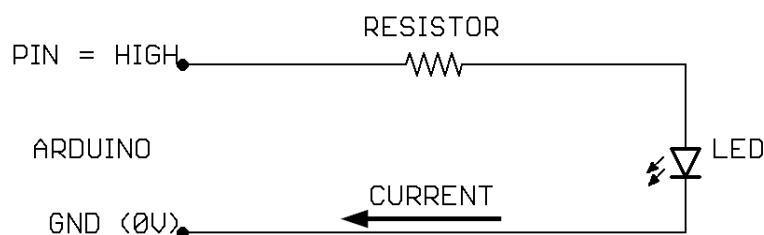
Today:



Pin set LOW (0V), no current flows:



Pin set HIGH (5V), current flows and LED turns on:



³ HIGH can also be referred to as 1 or ON

⁴ LOW can also be referred to as 0 or OFF

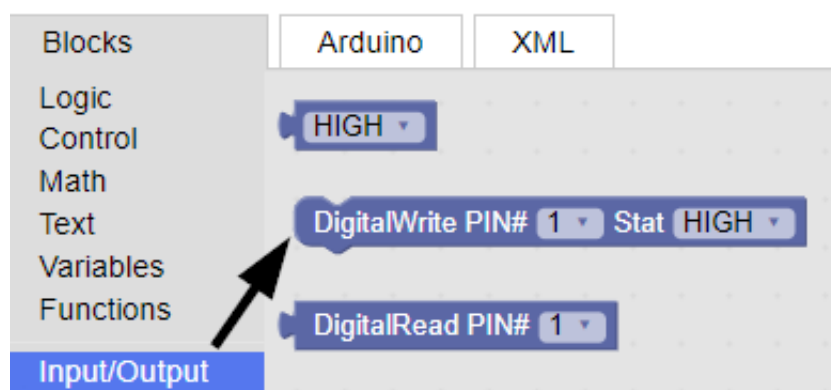
PROGRAMMING THE ARDUINO (10 MINUTES)

Instruct students on the fundamentals of programming the Arduino with the following key points.

The Arduino reads instructions from top to bottom. When it gets to the bottom, it repeats the instructions again from the top.

We will be using BlocklyDuino to write our instructions for the Arduino. Specifically, we will need two types of instruction blocks.

BlocklyDuino > web-based visual program



DigitalWrite allows us to set a pin as HIGH (5V) or LOW (0V). We need to set two options: the pin number and the value to write to it.

BlocklyDuino > web-based visual program

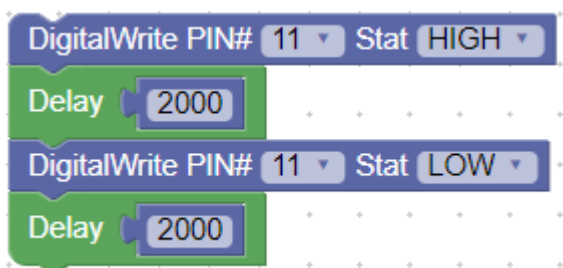


Delay will make the Arduino wait for a number of milliseconds. We need to select the number of milliseconds we want it to wait. There are 1000 milliseconds in one second.

So, which instruction do we use to make pin 11 become 5V, so it could turn on an LED?
(draw out the blocks on the board)




If we attach an LED circuit to pin 11, what blocks would we use to make it turn on for 2 seconds then off for two seconds?



If we don't have the last delay, what happens? A: since the Arduino follows the instructions from the top to bottom, then repeats, it would write LOW to pin 11, but then immediately

turn it HIGH again. This happens so fast (1000s of times per second) that we wouldn't even see the light turn off before the Arduino turns it back on again.

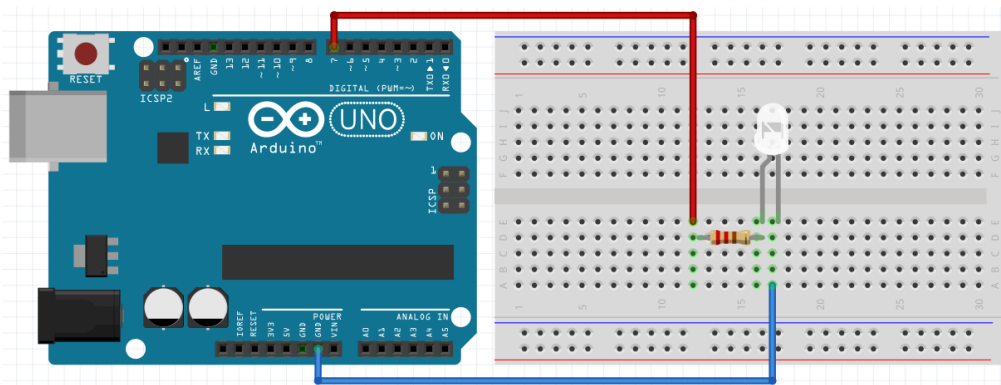
To write a program for the Arduino, and upload it to the board, follow these steps:

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

CONNECT THE LED CIRCUIT TO THE ARDUINO (5 MINUTES)

Distribute the LED circuits, Arduino UNOs, USB cables, and 9V batteries to the students.

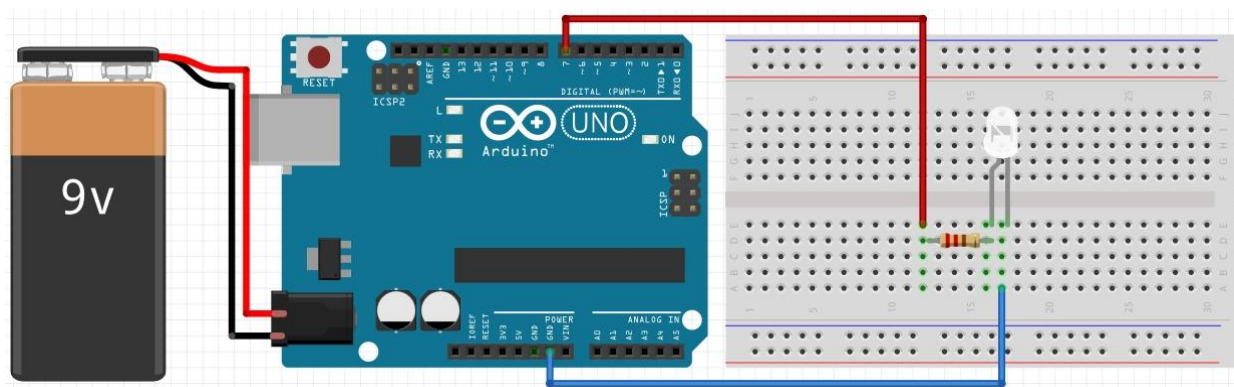
Connect the LED circuit to GND and one of the 14 digital pins on the Arduino, and then plug your Arduino into the computer using a USB. Here is an example of the LED circuit connected to Digital I/O pin 7:



PROGRAM THE ARDUINO AND TEST (15 MINUTES)

Assist students while they attempt to program and test the Arduino.

Use BlocklyDuino and the Arduino IDE to write a program that will blink the LED on for one second, then off for one second. When the program is uploaded to the Arduino, you can disconnect the Arduino from the computer and connect it to the 9V battery. Check to see that the Arduino is following the instructions that you uploaded.



DEBRIEF DISCUSSION (5 MINUTES)

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

Today we programmed an Arduino microcontroller board to make an LED light blink. Why is this important? What applications does this have? Possible answers include:

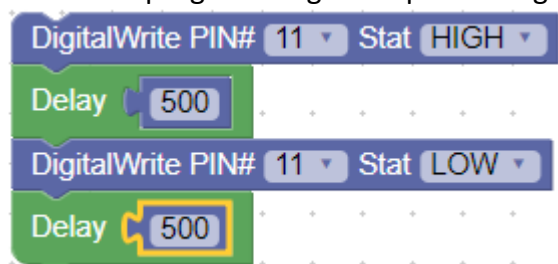
- Using microcontrollers, we can make our circuits respond to changes in time, or different sensors like light, sound, or temperature
- By writing instructions for computers or microcontrollers, we can make use of their amazing processing power. They can calculate and respond to things very quickly... millions of times each second!

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

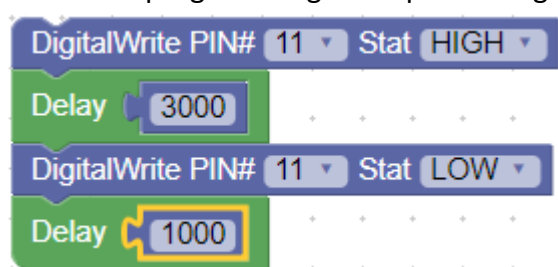
CHALLENGE AND EXPLORE

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

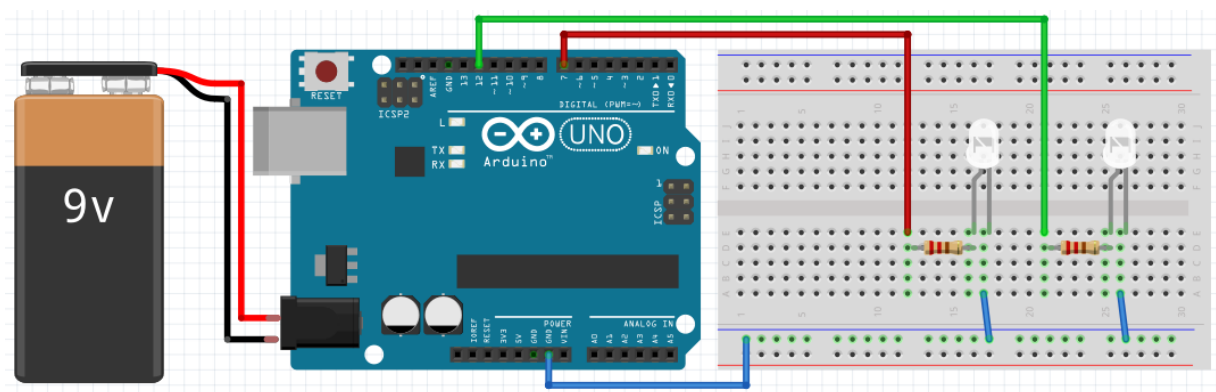
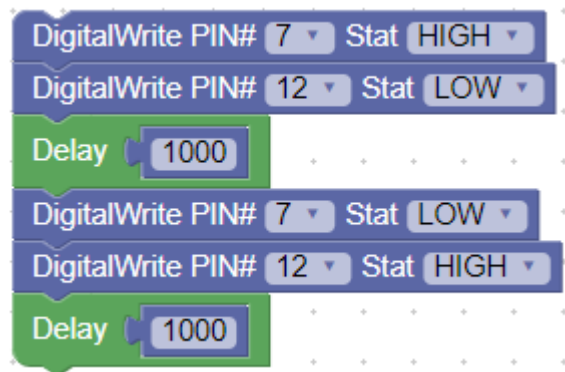
- Can you get the LED to turn on for $\frac{1}{2}$ second then off for $\frac{1}{2}$ second?
 - Skill: basic programming concepts - changing function inputs



- Can you make the LED turn on for 3 seconds, then off for 1?
 - Skill: basic programming concepts - changing function inputs



- Can you add a second LED to the circuit and make two LEDs alternate turning on and off?
 - Skills: microcontroller digital outputs; basic programming concepts; circuit building



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 my light turning off and on?
 - A1: have you successfully uploaded the code using the USB cable?
 - A2: does the pin number you used in your code match the pin number where the LED circuit is connected?
 - A3: check the polarity of the LED and make sure that the (+) side is connected towards the Arduino pin, and (-) is connected to GND.

