CANADA LEARNING CODE

Morse Code with Arduino

By: Nabeela Merchant
Duration: 60 minutes

LEVEL	SUBJECTS	PROVINCES / TERRITORIES	TOOL
Grades 9-12	Science and Technology; Applied Design, Skills, and Technologies	Across Canada, BC	Arduino

Overview

In this activity, you'll learn how to set up an Arduino Uno microcontroller and program a light (LED) to display a message in Morse Code.

Prep Work

- Create an Arduino Create account: https://create.arduino.cc/editor
- Follow the instructions to install the Arduino plug-in. This will let the web browser communicate with the Arduino Uno. Read this article to learn more about the Arduino Create Editor or if you have any trouble installing the plug-in: http://bit.ly/arduino-editor-getting-started
- Work through the example project to get familiar with the hardware and software and ensure everything is working correctly: http://bit.ly/arduino-morse-code-eg
- Go over the slides the for the lesson: http://bit.ly/arduino-morse-code-slides

Key Coding Concepts

- Sequence
- Loops
- Functions

Terminology

Arduino

"Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

- Print the solutions sheet: http://bit.ly/arduino-morse-code-solution

Materials (per pair):

- Laptops with administrator permissions
- Arduino Uno
- An A B USB cable (USB printer cable) to connect the Arduino Uno to the laptop

Lesson

Use the following slides for this lesson:
http://bit.ly/arduino-morse-code-slides

Introduction
Robots are devices that can carry out a series of complex actions. They can be programmed by a computer and can respond to their environment.

Robots already exist in our daily lives, though we may not recognize it. Have you ever been to a store where the door magically opens as you approach it? There is robotics in the background that makes that magic happen.

Robots work in a simple 3 step process: input, processing, and output. This is quite similar to how humans work as well; we take in information, our brain processes it and makes a decision, and then our body reacts. So if you see a snake, your brain recognizes that it is dangerous and commands your body to run! Similarly, hardware and software can work together to do cool things and react to their surroundings.

We'll be using a microcontroller, called the Arduino Uno, to write a message in Morse Code. The Arduino Uno acts as the brain of our robot. Watch a snippet of this video to learn more: https://youtu.be/CqrQmQqpHXc?t=20s (0:20 - 1:22)

**Video also included in the slides*

You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing."
→https://www.arduino.cc/en/Guide/Introduction

Curricular Connections

Text-based coding, digital output devices, electronics, components of an electric circuit, electrical components, PCB (printed circuit boards), input/output devices, microcontrollers, communication, impacts of technology on societies

References

Arduino Create
https://create.arduino.cc/
Arduino Uno
https://store.arduino.cc/usa/arduino-uno-rev3
Arduino Reference
https://www.arduino.cc/reference/en/

Code Along
Open up the example project and show learners the code and physical output (blinking light).

Point out the main elements of the code: comments, setup, loop, and commands. Explain the purpose of each of the commands using the descriptions in the adjacent comments.

Check for understanding by asking learners what would happen if you change the first delay(1000) to delay(500). What would happen if you change the second delay(1000) to delay(2000)?

Demonstrate the effect of changing the value of the delays in the code. This should change the speed at which the light blinks. Don't forget to verify and upload the code each time you want to show learners the effect of the change.

Activity
In this activity, you'll spell a word in Morse Code using an LED.

Morse Code is a way to communicate text information using a combination of dots and dashes to represent letters. The dots and dashes could be broadcast using sounds or light. S.O.S. is a famous sequence of letters that sailors used as a distress signal.

For example, S in Morse Code is dot dot dot (...) and O is dash dash dash (---), which is used to spell SOS. (See slides for example gif)

Use the Solution Sheet (http://bit.ly/arduino-morse-code-solution) to guide learners through the following steps:

1. Choose a word with at least 5 letters and convert it to Morse code.
2. Connect the Arduino Uno to the computer.
3. Create an account with Arduino Create Editor + Download the Arduino plug-in.
4. Walk through the example project to make sure the hardware and software work.
5. Open up the Morse Code project and Add to Sketchbook

6. Run the existing code
7. Comment out existing code
8. Add first letter (replace S.O.S. with the letters of the chosen word)
9. Add remaining letters
10. End the word
11. Check your code
12. Verify and upload the code to the Arduino Uno.

Have a partner decode the pattern of blinking lights to ensure the learner has spelled their word correctly.

Assessment

- After reviewing the example project, co-construct success criteria with the class to be used to evaluate their final projects.
- Have learners research the following programming concepts and explain how they used them in their project: Functions, Loops, Sequence

Extensions

Program the built-in LED on the Arduino Uno to flash the same word in Morse code, but instead of using the functions dot, dash, end_of_letter, and end_of_word, have learners use the following commands:

- digitalWrite(LED_BUILTIN, HIGH); // turns the LED on
- digitalWrite(LED_BUILTIN, LOW); // turns the LED off
- delay(1000); // pauses the system. use appropriate delay values (in milliseconds)

A dot is a light that's on for 500 milliseconds. A dash is a light that is on for 1,500 milliseconds. The LED is off for 2000 milliseconds between letters and for 5000 milliseconds between words.

Morse Code: Solution Sheet

STEP 1: Pick a word to convert to Morse code

1. The word should have at least 5 letters.
2. Convert the word to Morse code on a piece of paper.

STEP 2: Connect the Arduino Uno

1. Using the A B USB cable (USB printer cable) connect the Arduino Uno to a USB port of a computer.

STEP 3: Create an account with Arduino Create Editor

1. Go to: https://create.arduino.cc/editor
2. Create a new account or sign-in to an existing account.
3. Follow the steps to download the Arduino plug-in, if prompted.
4. Go here to download the plug-in if **not** prompted: https://create.arduino.cc/getting-started/plugin?page=2

STEP 4: Walk through the example project

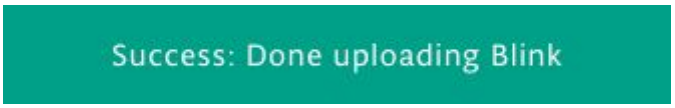
1. http://bit.ly/arduino-morse-code-eg
2. Have each learner go through the example project to make sure their hardware and software work.

STEP 5: Open the Morse Code project

1. Go to: http://bit.ly/clc-arduino-1
2. Select 'Add to my sketchbook'.

STEP 6: Run the existing code

1. Click the  button to verify the code.
2. Once verified, click the  button to upload the code to the Arduino Uno.
3. Check the progress bar at the bottom to ensure the code was successfully uploaded.



4. A light on the Arduino Uno should be flashing dot dot dot, dash dash dash, dot dot dot.

STEP 7: Comment out the SOS code for reference

1. Put // (2 forward slashes) in front of lines 33 to 46

```
31  void loop() {
32    // S in Morse Code
33    //dot();
34    //dot();
35    //dot();
36    //end_of_letter(); // end of letter
37    // O in Morse Code
38    //dash();
39    //dash();
40    //dash();
41    //end_of_letter(); // end of letter
42    // S in Morse Code
43    //dot();
44    //dot();
45    //dot();
46    //end_of_letter(); // end of letter
47    // End of word
48    //end_of_word();
```

STEP 8: Add the first letter

1. Find the Morse Code equivalent of the first letter of the chosen word (Arya). For example, A would be dot dash.
2. Use the functions dash(); and dot(); to code the first letter of the word. Each function should be called on a new line with a semicolon at the end of each line. Make sure all the code is within the closing curly bracket.
3. Add the function end_of_letter(); after the letter has been spelled out. This adds an extra pause to distinguish between letters.

```
46    //end_of_letter(); // end of letter
47    // End of word
48    //end_of_word();
49
50    // A in Morse Code
51    dash();
52    dot();
53    end_of_letter(); // end of letter
54  }
```

STEP 9: Add the remaining letters

1. Using a similar process to the one above, add the remaining letters to the code.
2. Don't forget to end each letter with the end_of_letter(); function.
3. Encourage learners to comment their code so they know when the letters start and end

```
50    // A in Morse Code
51    dash();
52    dot();
53    end_of_letter(); // end of letter
54    // R in Morse Code
55    dash();
56    dot();
57    end_of_letter(); // end of letter
58    // Y in Morse Code
59    dash();
60    dot();
61    end_of_letter(); // end of letter
62    // A in Morse Code
63    dash();
64    dot();
65    end_of_letter(); // end of letter
66  }
```

STEP 10: End the word

1. On a new line (still within the loop function) add the end_of_word(); function to add an extra delay to distinguish between the end of the word.

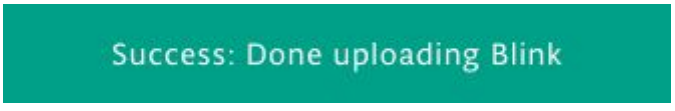
```
66    // End of word
67    end_of_word();
68  }
```

STEP 11: Final version of the code

```
50    // A in Morse Code
51    dash();
52    dot();
53    end_of_letter(); // end of letter
54    // R in Morse Code
55    dash();
56    dot();
57    end_of_letter(); // end of letter
58    // Y in Morse Code
59    dash();
60    dot();
61    end_of_letter(); // end of letter
62    // A in Morse Code
63    dash();
64    dot();
65    end_of_letter(); // end of letter
66    // End of word
67    end_of_word();
68  }
```

STEP 12: Run the code

5. Click the  button to verify the code.
6. Once verified, click the  button to upload the code to the Arduino Uno.
7. Check the progress bar at the bottom to ensure the code was successfully uploaded.



8. A light on the Arduino Uno should be flashing a word in Morse Code.

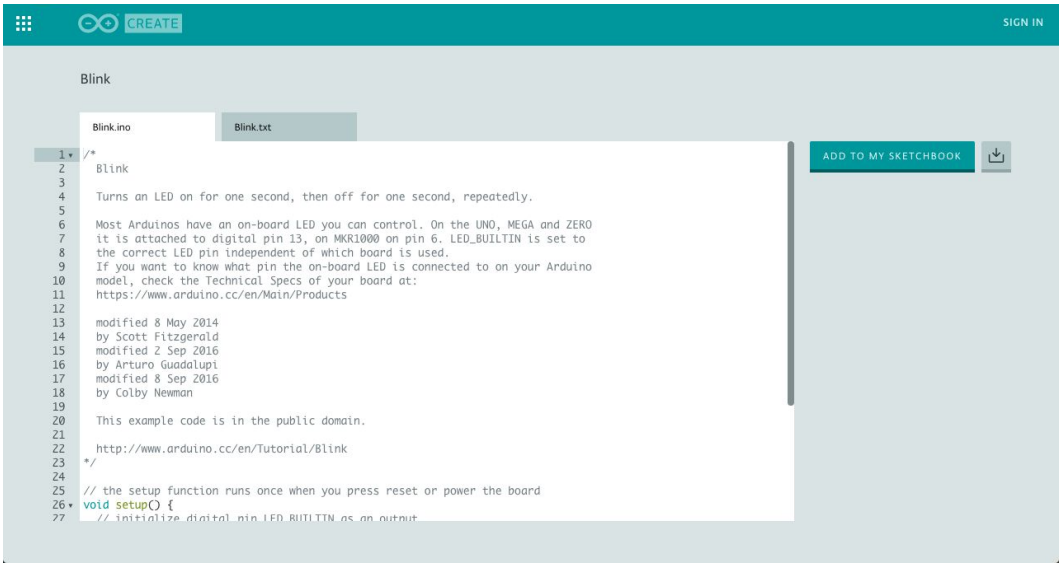
Morse Code: Example Project

STEP 1: Connect the Arduino Uno

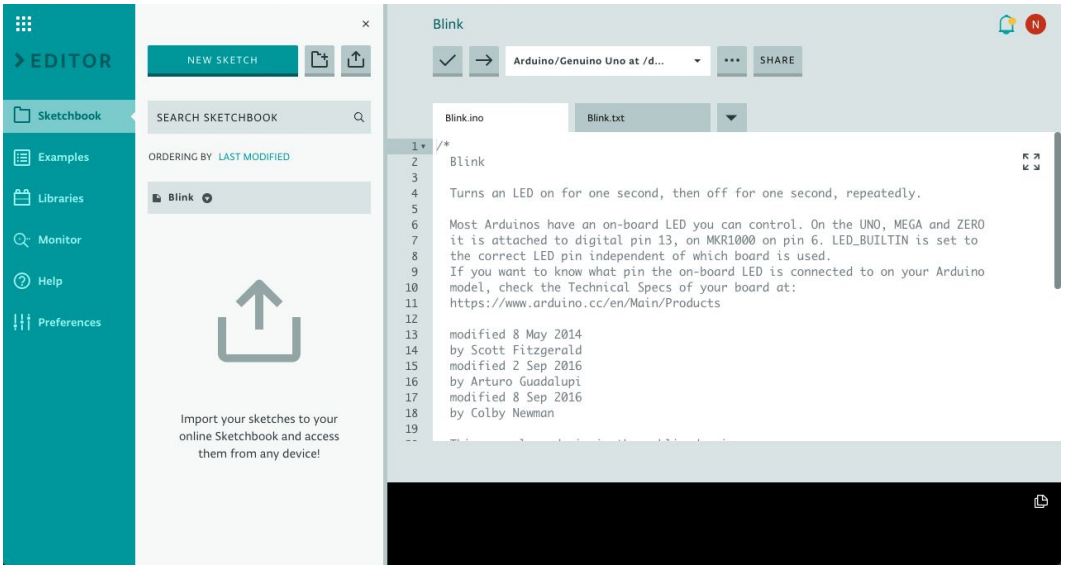
1. Using the A B USB cable (USB printer cable) connect the Arduino Uno to a USB port of a computer.

STEP 2: Open the example project

1. Open the example project: http://bit.ly/arduino-blink



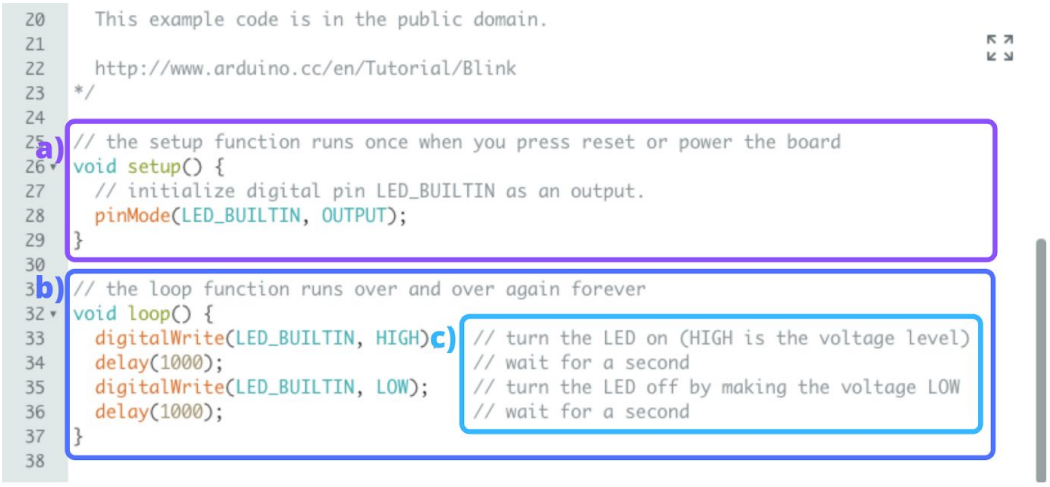
2. Sign into Arduino Create.
3. Click the “Add to my sketchbook” button.
4. The Arduino Create Editor should open with the example project, called Blink, open and ready to edit or upload.



STEP 3: Get familiar with the Arduino Create Editor

1. The 3 important parts of the Arduino Create Editor are a) the verify button, b) the upload button, and c) the editor.

2. The editor is the area where all the code is written. The 3 important parts of the editor are the a) setup function, b) loop function, and c) comments.



   a. The setup function runs once when you power or reset the board. This is where hardware can be initialised or set up for the first time.
   b. The loop function runs over and over again. Code that is placed in the loop function will continue to run so long as the board is powered. This is where the majority of the code is executed.
   c. Comments are text that does not affect the code. They an important part of the coding process and help programmers document and understand what is going on. Use 2 forward slashes

(//) to start a comment. Any text on the line after // will be ignored by the Arduino Uno.

3. Other important concepts include:
    a. Functions: functions are blocks of code that have names. setup and loop are examples of functions. The code that belongs to a function lives within the curly brackets that follow the function's name. Using the same format we can create functions of our own. These custom blocks of code are extremely useful as they make it easy to reuse code and reduce repetition.

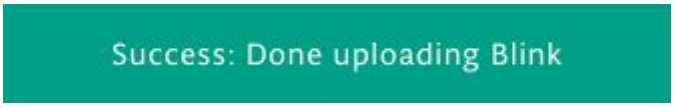
```
24  // the setup function runs once when you press reset or power the board
25  void setup() {
26    // initialize digital pin LED_BUILTIN as an output.
27    pinMode(LED_BUILTIN, OUTPUT);
28  }
```

    b. Semicolon: every command that is passed to the Arduino Uno needs to end in a semicolon (e.g., delay(1000);).

```
delay(1000);
```

STEP 4: Run the example project

1. Click the  button to verify the code.
2. Once verified, click the  button to upload the code to the Arduino Uno.
3. Check the progress bar at the bottom to ensure the code was successfully uploaded.



4. A light on the Arduino Uno should be flashing on for 1 second and off for 1 second.