High vs Low

Activity Overview:

The students' introduction to programming will cover the different parts of the Arduino Integrated Development Environment (IDE). Students will program a vibrating board connected to port 10 and negative expansion ports using alligator clips.

Learning Objectives:

Students will be able to:

- Connect the vibration board to pin 10
- Open, upload, edit & save Arduino sketches
- Identify and describe the purpose of a semi-colon (;)
- Identify the four parts of code being used
    - Comments
    - Variable declaration
    - Setup function
    - Loop function
- Add and manipulate delay command

Files/Materials Needed:

- Computer
- Engineering Journal
- LilyPad ProtoSnap Plus development board
- Micro-B USB Cable
- VibeOUTPUT.ino
- LilyPad vibration board (pictured below).

Code Vocabulary:

- Comments - describe the program for later reference for users to share
- Delay – Pauses the program for the amount of time (in milliseconds)
- High|Low – constants are predefined expressions in the Arduino language used in: reading and writing to a digital pin
- Pin – a physical connection (lead) to a point of microprossesor
- Void set up: a function that tells the LilyPad what the pins are and how they might behave
- Variable declaration - tells the Lilypad which pins you will be using in this code
- Void loop – contains loop function tells the LilyPad what the pins will do i.e. sense, turn on/off, wait, ect.

Syntax:

- camelCase – each word starts with a capital letter with exception of the first letter, functions and variable names are easier to read

- delay – delay(ms);
- semi-colon – used to end a statement
- HIGH – digitalWrite(pin, HIGH);
- LOW – digitalWrite(pin, LOW);

Lesson Instructions

"Sew" What: The Hook

Discuss these questions:

- When would a vibration alert be needed?
- What type of jobs should have a vibration alert rather than an auditory alert?
- What type of alerts (pattern) can be used to alert users?

Present New Information

Today we are going to learn who to turn on and off a vibrating component using commands in C++. Also, we'll look at the structures of the code we'll be using in Arduino.

Show Vibe Code and describe to the students the four parts of code:

```
/*
* Vibrate
* Turns on the vibe board for one second then off for one second, repeatedly.
*/

int vibe = ?;            // vibe pin is attached to ? on dev board

void setup() {           // NECESSARY PART OF CODE runs once
 pinMode(vibe, OUTPUT):  // sets vibeas an OUTPUT
}

void loop() {               // NECESSARY PART OF CODE repeats over and over
  digitalWrite(vibe, HIGH); // turns vibe ON
  delay(1000);              // waits 1 second or 1000 milliseconds
  digitalWrite(vibe, LOW);  // turns vibe OFF
  delay(1000);              // waits 1 second or 1000 milliseconds
}
```

The table below describes each section of the Code. A recipe analogy is used to help make a connection to something students may be familiar with.

	Section of Code	Lines	Arduino	Recipe Analogy

1	Comments	1-4	A brief description of what the program does	A brief description of the food to be made
2	Variable Declaration	6	Tells which pins and components will be used in the program	Tells which ingredients will be used in the recipe
3	Set-up	8-10	Tells what the pins “are” and what they “can” do	Tells what to cut or mix to do the cooking
4	Loop	12-17	Tells the pin what to do	Tells you how to do the final cooking

There are a lot of things going on in the code. We will just focus on a couple for this lesson. There is syntax – the set of rules that defines the combination of symbols that are considered to be a correctly structured document or fragment in that language and Arduino coding – language vocabulary – terms and commands used in Arduino environment. To begin, connect the (+) port of the Vibe board to Expansion Port 10/SCL on the LilyPad ProtoSnap Plus with an alligator clip. Next connect the (-) pin of the Vibe board to the (-) expansion port on the LilyPad ProtoSnap Plus as shown in the picture below.

We can turn off and on a pin by using the terms HIGH and LOW – notice all capitals. We change frequency and duration of vibration using delay();.

Apply Skills in Coding: A Guided Practice

Errors are friends

Together with the students you will be guiding them through a certain function or task in in coding. Balance direct instruction and having students try and explore.

1. With the students, open vibe_simple.ino sketch.
2. Verify the code by selecting the icon. This command goes through the code and checks for syntax and vocabulary mistakes.

   This error is intentional. Students are encouraged to use the error window and tools the Arduino IDE supplies to support correcting syntax and vocabulary errors. Notice the first line says: vibe_simple:6: error: expected pribary-expression before ‘?’ token

   The number 6 after the colon is important, suggesting there is an error in or around line 6. The verify command also suggests that the ‘?’ is a problem. Let’s fix it.

3. Identify the pin # on the development board.
4. Replace the ‘?’ with the right pin number. In this case, replace the ? with a 10.
5. Have students verify and upload the program with changed pin.

   The students vibe board will be vibrating: one second on, and one second off.

Explain the delay

1. Tell students about delay ();. delay is a native function in Arduino. Informing the computer to pause for the indicated number of milliseconds before moving onto the next line of code.

   1000 = 1 second: 1=1 millisecond

2. Together change the delay calue for line 14 to 500 (1/2 second) and line 16 to 1250 (1 ¼ seconds). It can be encouraged to change the values in the comments to match the code

3. Verify and Upload changes with the delay.

4. Allow students a shor opportunity to adjust and explore these values.
   - Ask students to change the values and see what happens.
   - "As the number gets larger what happens?"
   - "When the HIGH delay and the LOW delay are different what happens?"

Copy and Pasting Code

Tell students they can add lines of code to make the vibe board turn on and off in a pattern. You will be showing students how to copy and paste, tell students to notice the curly braces {}. These are explained more in depth in the next lesson. They need to keep the code inside the curly braces.

One of the easiest ways to copy and paste new code is to **move the curly brace down** by pressing return (enter) aka carriage return.

1. Place cursor in front of the curly brace on line 17.
2. Press return or enter to move the curly brace down enough lines to paste your new code.
3. Copy (control+c) and paste (control+v) new lines.
4. Allow students to experiment by placing delay(), digitalWrite(variable, HIGH), and digitalWrite(variable, LOW) in different arrangements to get different effects.

Save Your Code

1. Once you've created a vibrating behavior that you like, click on the downward-pointing arrow in the toolbar to save your code.
2. Choose a good name for your file (such as "vibe_simple xx v1"), and click on the "Save" button to complete the process. You can use your initals and version number. This will help in later projects. Vibe_simple_dv_v1

3. To make sure that the file should appear at the top of the pop-up menu that appears. All the programs that you save in Arduino will appear in this menu so that you can find and open them easily.

Assess Learning through Challenge

Provide students an opportunity to explore with the code in this lesson.

- Makes a beat from a faborite pattern or create a new beat. e.g. We Will Rock You.
- Make a vibe alert for an emergency. What is the vibration pattern? What is the emergency?
- Make a specific vibe alert for a phone call from a family member, friend, or other person.

Students are learning to control electrical signals with code. It's work stopping to think about how powerful this is. The LilyPad and the Arduino turn text that you wrote on your computer into behavior that happens in the physical world. Wow!