WYRKSHOP MOBILE MAKERSPACE

LilyPad Protosnap Plus Coding

ESTIMATED TIME: 4-6 HOURS

LEARNING OBJECTIVES

- Discover how the Lilypad Protosnap Plus kit works
- Learn the anatomy of the LilyPad
- Create a customized code

LEVEL: 3

MATERIALS NEEDED:

Laptop

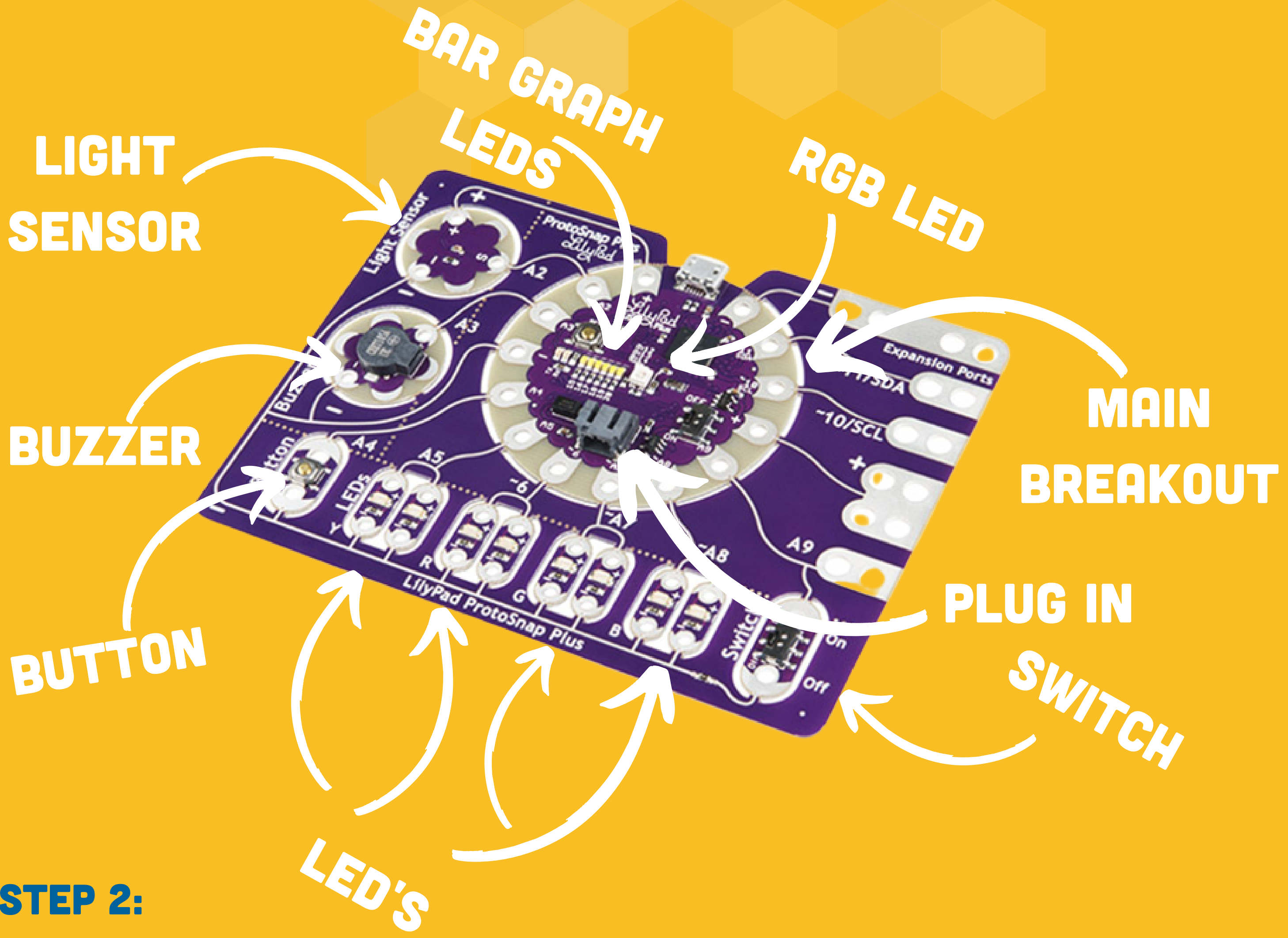
Lilypad Protosnap Plus Kit

Arduino Software

ACTIVITY OUTLINE:

STEP 1:

Download the Arduino software onto your laptop. While that is downloading, lets run through a bit of anatomy of the LilyPad.



STEP 2:

Go to: **shorturl.at/emyKO** to get the full Protosnap Plus set up guide. On this website, scroll down to "Setting up Arduino" and follow the guide, stopping at "Stitching into a project".

STEP 3:

Most Arduino codes will have text in a light grey color, this is not code. This text tells you what the actual code is doing. Here we will explain a bit of how to read basic code:

CODE APPENDIX:

- If you see code (such as int or tone) before void setup (), this is what we reffer to as a "**code appendix**". This is something that the code below void setup () and void loop () will reference, so you don't have to define the same thing over and over again.
- **analogWrite (pin, value)**- Similar to digital, however, analog values have a value anywhere from 0 to 255 brightness. 0 meaning that light is not on at all and 255 meaning maximum brightness.
- **delay(interval)**- coding language runs off milliseconds.

1000 MILLISECONDS = 1 SECOND

- **digitalWrite (pin, value)**- This confirms that the pin denoted will have either a high or low output.
- **If, else statements**- You can think of if else statements like if then statements in science class. If you do X then Y will happen, but if you don't do X then Z will happen.
- **Input**- This means something you do to get a reaction, in coding this usually means flipping a switch or clicking a button
- **int**- converts a value to a data type. In this project, this is usually what you classify your music notes and tempo. Such as int Note_C5= 523 or int tempo = 500. You may also see this for a button or switch.
- **Low/ High**- low means that there is a low voltage going to the pin meaning there won't be a voltage and no light. High output means that there is a voltage going to the pin meaning you will see a light or hear a buzzer.
- **Output**- This is what something is doing like shining a light or putting out a sound.
- **pin**- The arduino pin number to set the mode of
- **pinMode (pin, mode)**- Confirms the specific pin, inside the brackets that follow will denote the actual pin and if its an input or output
- **void loop ()**- This means any code below this point will run through the code and repeat forever or until something else happens
- **void setup()**- this tells the Arduino that the code starts here
- **{ }**- An open curly bracket must always be placed before a closed curly bracket.

STEP 4:

In the Arduino app, click on File> Examples> LilyPadProtosnapPlus> LPP_01_Blink to run the code, click the upload button with the LilyPad connected.

BLINK EXAMPLE BREAKDOWN:

First we can see a void setup () at the top. Meaning that the code is all below this point and there is no specifications of inputs.
Next we see pinMode (A5, OUTPUT); this means that at pin A5 on your LilyPad board the light will come one.

Take a look at your board, what light do you see at pin A5?

Next we see void loop ()
{
This means that this code will keep looping forever.
digitalWrite(A5, HIGH); - This means that this is a digital pin meaning that this LED can only turn on and off, no dimming.
delay(1000); - This LED will turn on for 1 second then go on in the code
digitalWrite(A5, LOW);
delay (1000); - The LED on pin A5 will turn off for 1 second.
}

BASIC COLOR MIXING EXAMPLE BREAKDOWN:

In the Arduino app, click on Files> Examples> LilyPadProtosnapPlus> LPP_02_BasicColorMixing.ino

This code is a bit longer than the last, but it does the same thing over and over again. Let's go through it together. This time we have some commands above the void setup (). This means that we have some int that we need to pay attention to.

int RGB_red= 12; - This is a built in RGB in the center of the circle LilyPad chip. This means that red is connected to pin 12.
int RGB_ green= 13; - Same as the last, except this is on pin 13 and it will output green.
int RGB_blue= 14; - The RGB blue pin is on pin 14

int redLED= 6; - If you look at the board, pin 6 is the red LEDs. *Side note the ~ means this is an analog pin.
int greenLED= A7; - This is defining pin A7 as the green LEDs.
int blueLED= A8; - Defining all blue LEDs on pin A8.

Bonus Question: Are the green and blue LEDs analog or digital?

BASIC COLOR MIXING EXAMPLE BREAKDOWN CONTINUED:

Void setup ()
{
pinMode(RGB_red, OUTPUT); - This is just telling us that these variables will have an output. Meaning those LEDs will turn on when commanded.

Void loop ()- This function will cycle through all the code, then loop back to the start and go again.

digitalWrite(RGB_red, LOW);
digitalWrite(RGB_green, LOW);
digitalWrite(RGB_blue,LOW): - This means that the pin defined (RGB_red, green and blue) will have low voltage, meaning they will not come on.

digitalWrite(redLED, LOW);
digitalWrite(greenLED,LOW);
digitalWrite(blueLED, LOW);
delay (1000); - These lights will not display because they have a low output for 1 second.

digitalWrite(RGB_red, HIGH);
digitalWrite(RGB_green, LOW);
digitalWrite(RGB_blue, LOW);

digitalWrite(redLED_HIGH);
digitalWrite(greenLED_LOW);
digitalWrite(blueLED_LOW);
delay(1000); - The red LED in the bottom rows and on the RGB will display, but nothing else will for 1 second.

Now, try to go through the next example, LPP_03_CustomColorMixing and LPP_04_Fading by yourself or a partner and see if you can decipher what is happening. Hint this is what we just went through, but using analog outputs of brightness

TUNE EXAMPLE BREAKDOWN:

int buzzer = A3; - Take a quick look at the LilyPad board, you should see that the buzzer is on pin A3. Do you think this will be an input or output?

int NOTE_C5 = 523 - We have found a new type of int! This one is corresponding to a music note C5. If you are curious how note C5 = 523, take a look at this website: http://www.arduino.cc/en/Tutorial/toneMelody

int tempo = 500; - Take a quick guess at what this int means. If you guessed that this is the tempo for the song notes listed in the code, then you are right! Even robotic music needs a tempo to sound like a song!

void setup ()
{
pinMode(buzzer, OUTPUT); - If you guessed above that this would be an output, then you were correct! A buzzer produces an output as in it gives something off.

Void loop ()- Take a wild guess at what this means.

tone(buzzer,NOTE_C5);
delay(tempo); - This is a new variable, but we can probably decipher it by now. This means that the buzzer will play note C5 at the tempo indicated above. We don't need to constantly define what tempo or the sound of each note because it is referenced above in the "code appendix".

Skip down to line 85 and 86:
noTone (buzzer);
delay (tempo*4); - This means that when the code gets to this line, it will not play and will delay at 500*4 which is 2 seconds.

BUTTON EXAMPLE BREAKDOWN:

int buttonPin = A4;
int switchPin = A9; - This is defining that the button is on pin A4 and the switch is on pin A9.

int buttonLED = A5; - Take a look at the LilyPad board, what LED light is on pin A5? If you said yellow, then you are correct. Before we even get down to the other code, take a wild guess at what will happen if you click the button.
int switchLED = A8; - Look at the LilyPad board, what LED is at pin A8? If you said blue, you are correct!

BUTTON EXAMPLE BREAKDOWN CONTINUED:

void setup ()
{
pinMode(buttonPin, INPUT_PULLUP);
pinMode(switchPin,INPUT_PULLUP); - This input is a new variable, lets break it down. Above we already defined what the button and switchpin were, so we won't go through those again. Make a quick prediction of what "pullup" may mean for a button and switch.
A pullup in this case would mean that the button is in the upright position (not pressed down), you can think of this as a HIGH variable.
Remember an INPUT is something that you have to do to get an OUTPUT, in this case you have to depress the button to get an output.

void loop ()
{
int buttonState;
int switchState;
buttonState = digitalRead(buttonPin);
switchState = digitalRead(switchPin); - Here we are defining that our button state (if its pushed in or not) needs to convert into a digital read (something will happen at the variable defined in the parenthesis).

if (buttonState ==LOW)
{
digitalWrite(buttonLED,HIGH);
}
else
{
digitalWrite(buttonLED,LOW); - Okay these are some new variables, lets break them down. The top chunk of code (the if statement) is saying that if the button is pressed (aka the buttonState is low) then the button controlled LED's will have a high output (aka they will turn on). The bottom chunk of code (the else statement) is stating that is the button is not pressed, the LED's won't come on.

LIGHT EXAMPLE BREAKDOWN:

int sensorPin = A2; - This is a new variable, take a look at the LilyPad board. Pin A2 is connected to a light sensor.

LIGHT EXAMPLE BREAKDOWN CONTINUED:

int redLED = 6;
int greenLED = A7;
int blueLED = A8; - What LED's do you think this code use? Hint: look at pins 6, A7, and A8.

void setup ()
{
pinMode(sensorPin, INPUT);
pinMode(redLED, OUTPUT);
pinMode(greenLED, OUTPUT);
pinMode(blueLED, OUTPUT); - So, if there is an input to the light sensor, then there will be an output of red, green and blue LED's. What do you think a light sensors input is?

Serial.begin(9600); - This initialized the serial monitor

void loop ()
{
int sensorValue; -This can be anywhere between 0 and 1023

senesorValue = analogRead(sensorPin); - This basically interprets the reading so that the serial monitor can understand it.

Serial.print("sensor value: ");
Serial.println(sensorValue);
analogWrite(redLED, sensorValue / 4);
analogWrite(greenLED, sensorValue / 4);
analogWrite(blueLED, sensorValue / 4); - Remember when we said that the serial monitor can be anywhere from 0 to 1023? analogWrite can only read values from 0 to 255, so we need to scale it (aka divide it) so that the analogWrite can read it.

BAR GRAPH EXAMPLE BREAKDOWN:

int sensorPin = A2;
int bargraphLED[6] = {15,16,17,18,19,20}; - This looks like a lot, so lets break this down. Normally when we see (6) it means that pin 6 will be involved, however, since this is square brackets [ ] this means that this is the number of LEDs involved. The specific LEDs are indicated within the curly brackets. These are the small LEDs inside the main chip.

BAR GRAPH EXAMPLE BREAKDOWN CONTINUED:

These LEDs are put into what is called an array meaning there are individual elements of the same type (aka 6 different LEDs). The array index is from 0 to 5; so if we call out bargraphLED[2] then the 17th light would come on.

void setup ()
{
int x;
pinMode(sensorPin, INPUT); - This initializes the sensor pins as an input, but without a pullup. This means that when the light sensor senses light, it will produce an output with the LEDs.
for (x=0; x <= 5; X++)
{
pinMode(bargraphLED[x],OUTPUT); - This is saying that LED 15 (variable x) is equal to 0, and x is less than or equal to 5 (aka LED 20); then LEDs between 0 and 5 will turn on depending on the amount of light coming through the light sensor.

THEREMIN EXAMPLE BREAKDOWN:

int sensorPin = A2;
int buttonPin = A4;
int buzzer = A3;
int bargraphLED[6] = {15,16,17,18,19,20}; -Do these int look familiar? If they don't go back to the light sensor, button, and bar graph examples.

int highestFrequency = 1047;
int lowestFrequency = 523; -These should also look familiar too, this is just like the music notes output in the buzzer example.

void setup ()
{
int x;
pinMode(sensorPin, INPUT);
pinMode(buzzer, OUTPUT);
pinMode(buttonPin, INPUT_PULLUP): - Pop quiz, what is this piece of code telling us? Well, its telling us that the light sensor will have an input (light in this case), and the buzzer will have an output of sound, and the button must be in the upward position.

for(x = 0; x <+ 5; x++) -If this does not look familiar, go back to the bar graph example. Go through the rest of the code and see if you can decipher it (use the grey text for help if you need it).

NIGHT LIGHT EXAMPLE BREAKDOWN:

The last example, finally! Go through this example on your own or with a partner. Don't worry, we have went though everything int his example, you can do this!

CREATE A CUSTOM CODE:

Now, lets have you make your own custom code. Take inspiration from the examples above, combine whatever you want together to create your own custom code. If you want to make your own custom coded song, google some inspiration or see if it has been done before.This will take a bit of practice, but you can do it!

If you enjoyed this lesson plan and want to go onto embroidering a beanie and installing the LilyPad, take a look at the Lilypad Beanie lesson plan.

FIX MY CODE:

I'm new to coding and I need some help! I wrote out the code below, but my Arduino IDE app says that there are some things wrong with it. Can you help me, I think it said there are 9 mistakes?

```
int blueLED = 6
int greenLED = 7;
void setup ()
{
pinMode(blueLED, INPUT)
pinMode(greenLED, OUTPUT);

void loop ()
{
int brightness;
for (brightness + 0; brightness <= 255; brightness = brightness + 1)
analogWrite(blueLED, brightness);
analogWrite(yellowLED, brightness);
delay ( );
}
else (brightness = 255; brightness >= 0; brightness = brightness - 1)
{
digitalWrite(blueLED, brightness);
analogWrite(redLED, dimness);
delay (5);
}
}
```