A Maker's World: Wearables with Arduino Gemma

Instructor: Natalia Baklitskaya (natalia.baklitskaya@gmail.com)

For this workshop, the laptop in front of you already has Arduino IDE installed. We have also pre-installed the IDE Adafruit boards support, Adruino Gemma Windows drivers, and Adafruit NeoPixel (programmable LEDs) library. If you would like to set these up on your personal or work computer outside of this workshop, please follow instructions outlines in Packet 3.

In this portion of the workshop, we will be working with an **Arduino Gemma**.

Arduino Gemma

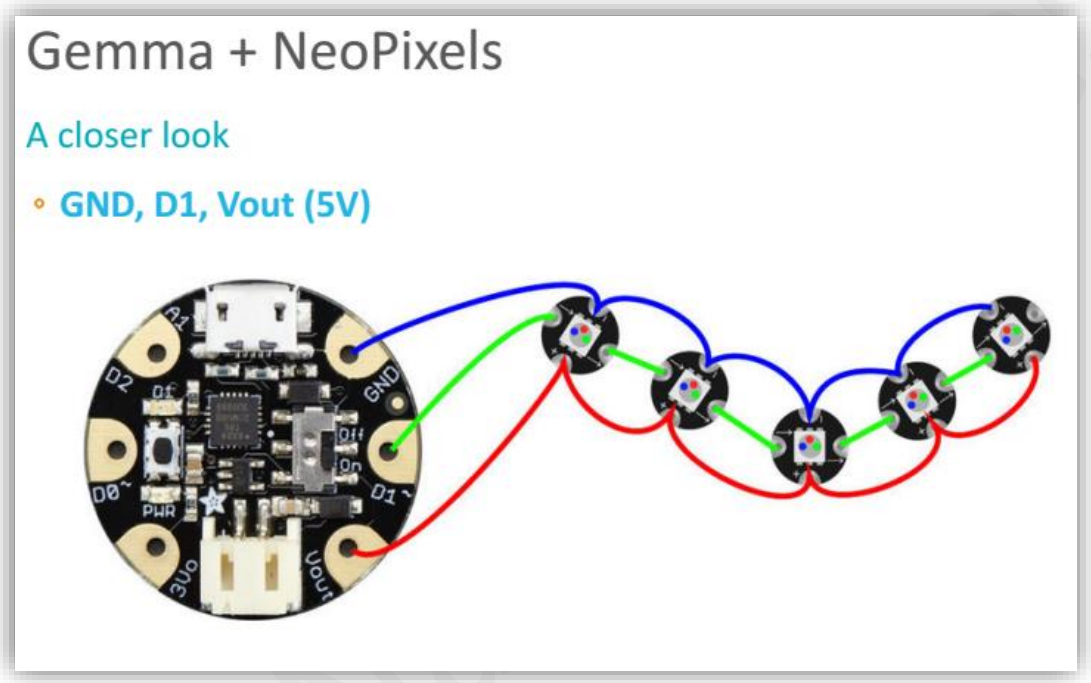
Arduino Gemma is a miniature wearable microcontroller board based on the ATtiny85. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a battery to get started on your wearable projects!

Arduino Gemma has much **fewer pins than an Arduino Uno**, and much **less space for your code**, but it is **compact**, which makes it great for wearables. It also **has a JST connector for a LiPo battery**, and **on/off switch**, and **nice big pads for its pins**, which are great for sewing through with conductive thread. It can also survive gentle washes (but don't stick this baby mictrocontroller in the dryer!).

INSTRUCTIONS

1. Pick up the Arduino Gemma microcontroller. Observe that it has been augmented with snaps for easy reuse between different wearable circuits.

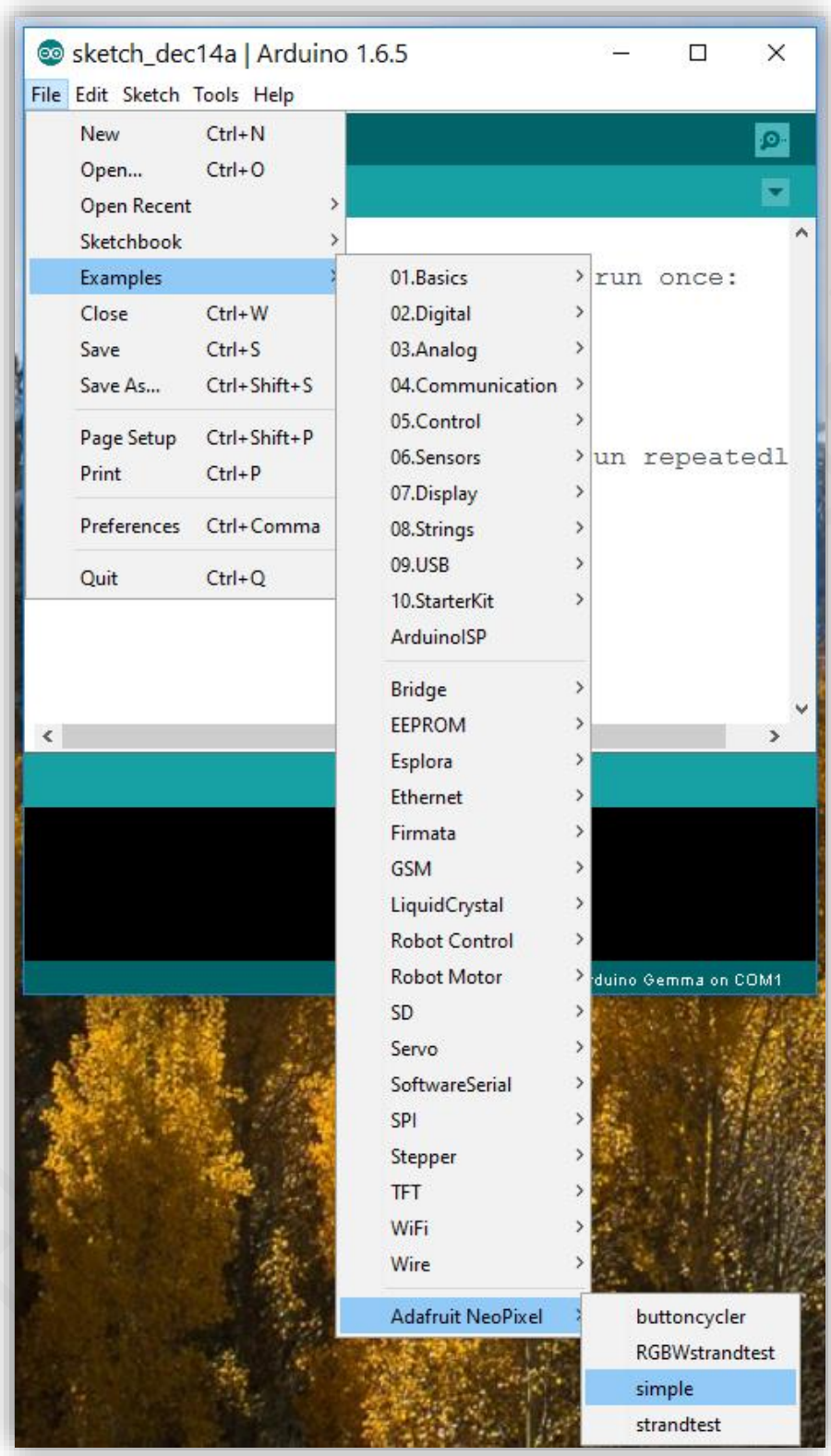
2. Plug the board into the circuit by snapping it into place to make the connections like in this circuit:



3. Locate the micro USB cable at the tables. Plug one end of it into Arduino Gemma, and the other into the laptop at your desk.

4. Open Arduino IDE by double clicking on the teal Arduino icon on your desktop.

5. Open ***File > Examples > Adafruit NeoPixels > simple***



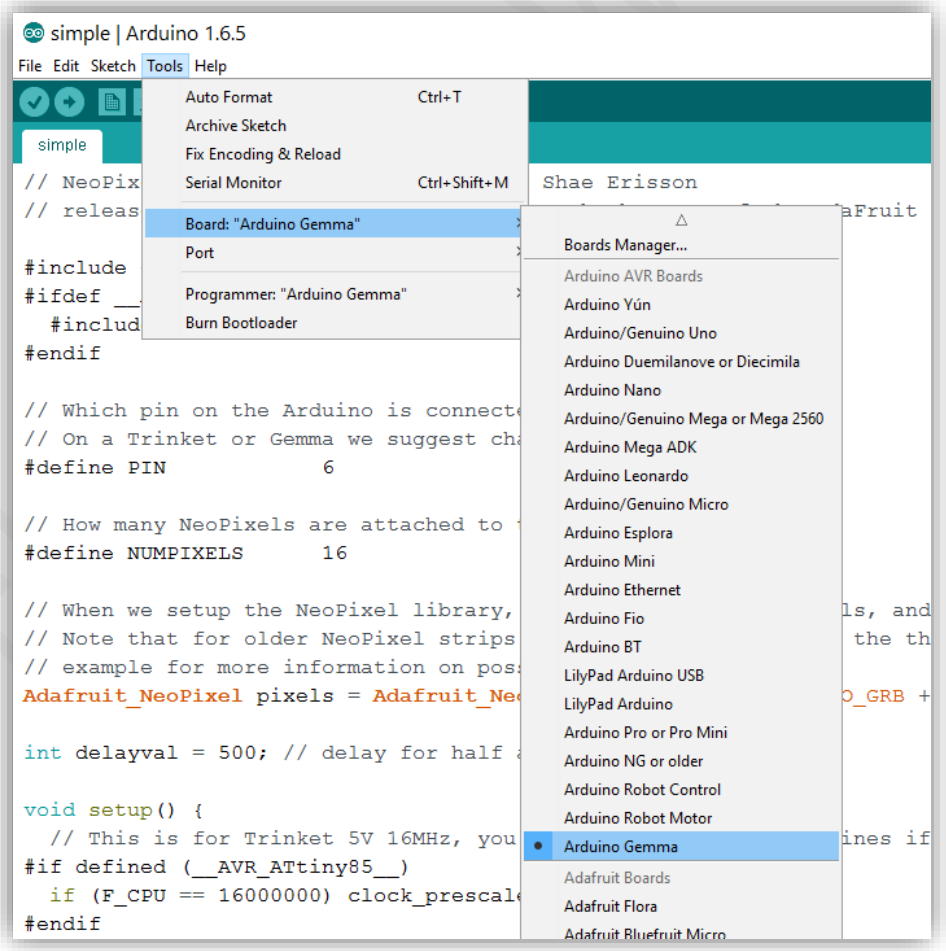
6. Modify the code to work with your microcontroller by:
    a. Adjust the value of **PIN** on line 11 to correspond to the output pin on your Gemma that is feeding the serial input to the NeoPixels.

```
// Which pin on the Arduino is connected to the NeoPixels?
// On a Trinket or Gemma we suggest changing this to 1
#define PIN            6
```

    b. Adjust the value of **NUMPIXELS** on line 14 to correspond to the number of NeoPixels connected to Arduino Gemma.

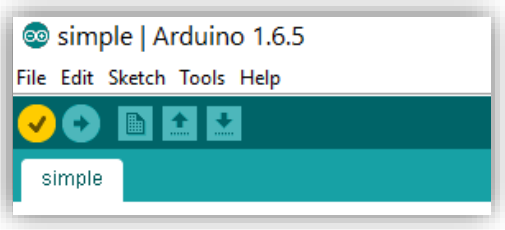
```
// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS      16
```

7. Change the Board and Programmer settings under Tools to 'Arduino Gemma'.



8. Save your program by going to **File > Save** (or hit **Ctrl+S**).

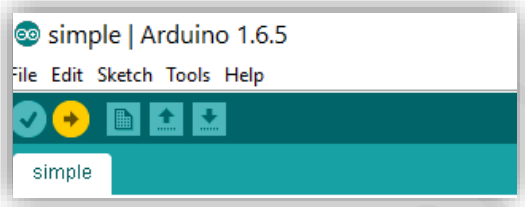
9. Compile your code by clicking the check mark in the upper left corner in your window. Check for errors at the bottom of the window.



10. Turn on your Gemma by flipping the switch on the board to 'On'.

11. Press the tiny reset button on the Gemma board and wait for it to start flashing a red light.

12. Upload the code to your Gemma while the red light is flashing on the board.



13. Both NeoPixels connected to your Gemma should be glowing **green**. Adjust your code to make the NeoPixels to first light up in **green** for 1 second, then **blue** for 1 second, then **red** for 1 second, all in a loop. Demonstrate your circuit to a TA once you done.
Hint: you may want to consider replicating the lines below and adjusting them for different colors.

```
// pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
pixels.setPixelColor(i, pixels.Color(0,150,0)); // Moderately bright green color.

pixels.show(); // This sends the updated pixel color to the hardware.

delay(delayval); // Delay for a period of time (in milliseconds).
```

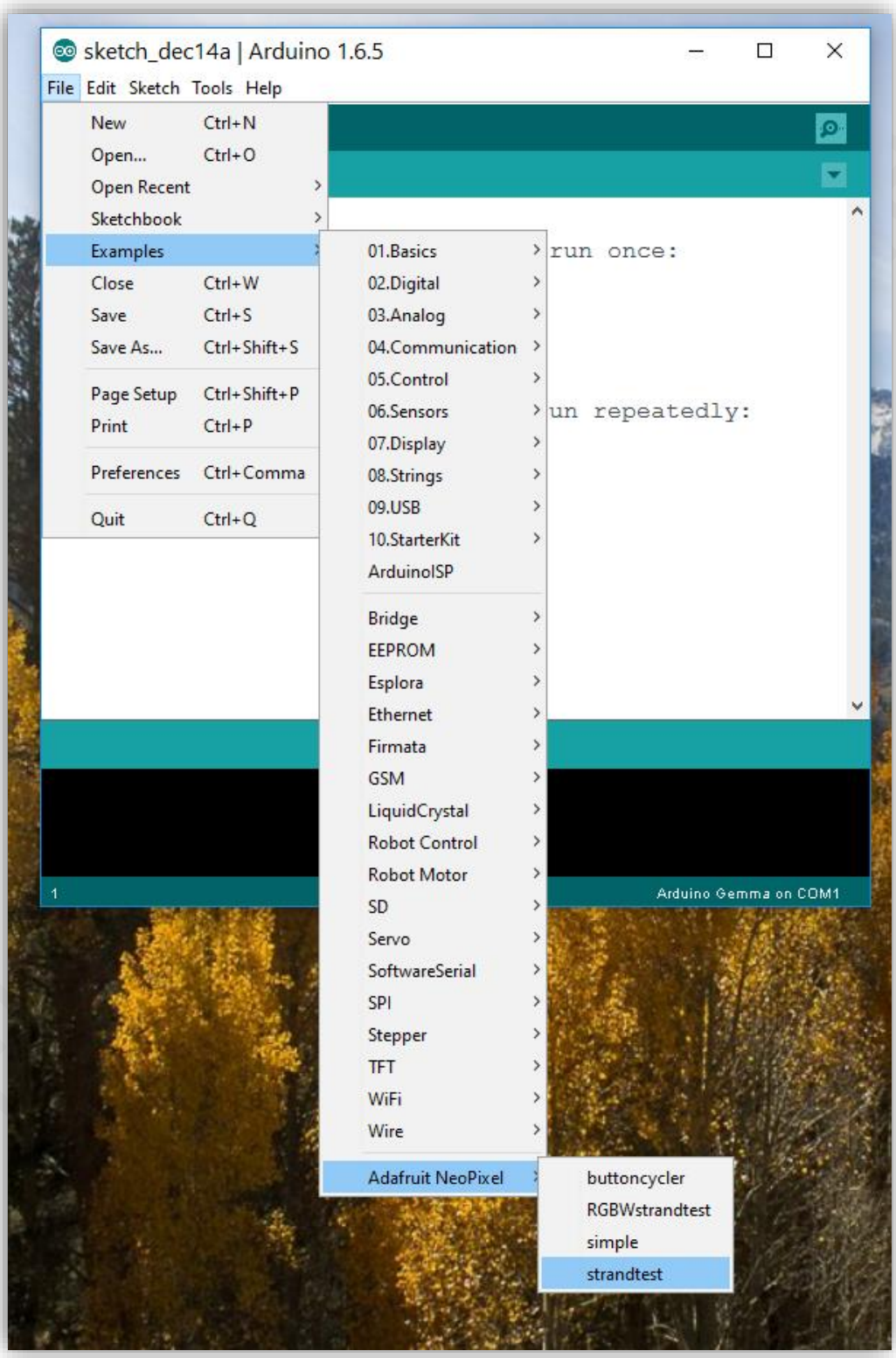
14. Now let's switch your circuit to a more compact power source. Connect the battery found at your table to the Gemma.

15. Unplug the USB cable from Gemma. Your circuit is now powered by a rechargeable battery!

BONUS

1. Connect your Arduino Gemma to the PC again using the USB cable.

2. Open ***File > Examples > Adafruit NeoPixels > strandtest***

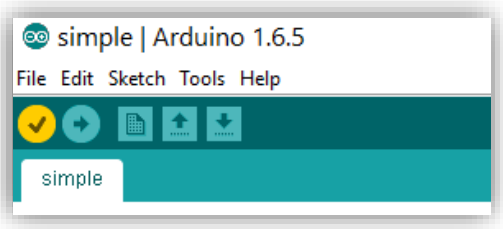


Workshop materials created by Natalia Baklitskaya (Natalia.Baklitskaya@gmail.com - ElectroNat Wearables).
Updated workshop materials can be found here: https://github.com/electronat

3. Adjust the value of **PIN** on line 6 to correspond to the output pin on your Gemma that is feeding the serial input to the NeoPixels.

4. Adjust the number of NeoPioxels on line 16 to correspond to the number of NeoPixels connected to Arduino Gemma.

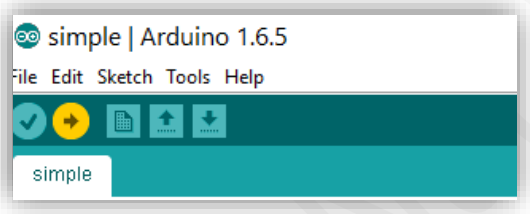
5. Compile your code by clicking the check mark in the upper left corner in your window. Check for errors at the bottom of the window.



6. Turn on your Gemma by flipping the switch on the board to 'On'.

7. Press the tiny reset button on the Gemma board and wait for it to start flashing a red light.

8. Upload the code to your Gemma while the red light is flashing on the board.



9. Observe the awesome patterns being displayed by your circuit. These get even cooler when you have more NeoPixels connected in sequence. Look through the code and look at the pre-programmed functions that are being used to create these patterns. You can modify these example functions and create your own patterns!