



Workshop 1, Exercise 2

Change the Gemma code using Arduino software

Note. Gemma, NeoPixels, Adafruit are trademarks of Adafruit Industries. Arduino is a trademark of Arduino and its partners. All rights acknowledged.

The NeoPixels you connected in Exercise 1 are being flashed in different colours by the code below.

Activity 1. Understand the starting code. Work out which bit of the code does what, based on what you already know about the light pattern the NeoPixels are making.

Clue 1: The code controls up to 4 NeoPixel LEDs, which are numbered 0, 1, 2, and 3.

Clue 2: Colours are described using 6 characters, where the first two represent how much red, the next two represent how much green, and the final two represent how much blue. So the colour YELLOW would be ffff00 (a mix of red and green).

```
#include <Adafruit_NeoPixel.h>
// 24 Feb 2014, Workshop 1
#define PIN 1

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(4, PIN);

uint32_t color1 = 0xff0000, // red
color2 = 0x00ff00, // green
color3 = 0x0000ff; // blue

void setup() {
  pixels.begin();
  pixels.setBrightness(85);
}

void loop() {
  pixels.setPixelColor(0, color1);
  pixels.setPixelColor(1, color2);
  pixels.setPixelColor(2, color3);
  pixels.setPixelColor(3, color1);
  pixels.show();
  delay(200);
  pixels.setPixelColor(0, 0);
  pixels.setPixelColor(1, 0);
  pixels.setPixelColor(2, 0);
  pixels.setPixelColor(3, 0);
  pixels.show();
  delay(400);
  pixels.setPixelColor(0, color2);
  pixels.setPixelColor(1, color3);
  pixels.setPixelColor(2, color1);
  pixels.setPixelColor(3, color2);
  pixels.show();
  delay(200);
  pixels.setPixelColor(0, 0);
  pixels.setPixelColor(1, 0);
  pixels.setPixelColor(2, 0);
  pixels.setPixelColor(3, 0);
  pixels.show();
  delay(400);
  pixels.setPixelColor(0, color3);
  pixels.setPixelColor(1, color1);
  pixels.setPixelColor(2, color2);
  pixels.setPixelColor(3, color3);
  pixels.show();
  delay(200);
  pixels.setPixelColor(0, 0);
  pixels.setPixelColor(1, 0);
  pixels.setPixelColor(2, 0);
  pixels.setPixelColor(3, 0);
  pixels.show();
  delay(400);
}
```

Activity 2. Change the code. What changes would you like to make? You will need help to get started, but after that it is up to you. Careful to make only small changes at a time, and to save them only when you know they work. Look at the **suggestions** below if you are not need ideas.

IMPORTANT When you have changed the code you will need to upload it to the Gemma using a USB cable. The person leading the workshop will show you how to do that bit (but basically you do the following steps:

a) Change the code and click the tick at the top left of the Arduino software to 'Verify' that your code does not have any mistakes. The software will say 'compiling sketch' at the bottom, and then tell you if there are any problems.

b) Once your code is verified, you press the arrow next to the tick in the Arduino software. That sends the code to the Gemma **BUT** at the same time you have to tell Gemma to be listening. You do that by clicking the very small button on the Gemma itself, next to its little red light.

If all is well you will see blocks of code cascading down the screen at the bottom of the Arduino software. If not, you will get an error message. Note that you have to time it so that the Arduino stops compiling and starts sending within 10 seconds of you pressing the Gemma button!!!

If not, you may have to ask for help the first couple of times.

Here are some suggestions for making something interesting:

Suggestion 1. Change the colours being used. Here are some possible colour codes. Try them and see what you get:

ff99ff ffcc66 ff3399 ccffcc cfff66

As you will see, the full colour codes here all also have 0x at the start (e.g. 0xff0000 for red), so be careful.

Suggestion 2. Change the intensity of the LEDs. At the moment it is 85, but it can go up to a maximum of 255.

Suggestion 3. Change the DELAYS so that the LEDs are on for a longer or a shorter time, or off for a longer or a shorter time

Suggestion 4. Make the cycle of colours use more colours.

Suggestion 5. Your idea ...
